DIGITAL SIGNAL PROCESSING

Principles, Algorithms, and Applications

Fourth Edition

John G. Proakis
Dimitris G. Manolakis

Digital Signal Processing

Fourth Edition

John G. Proakis

Department of Electrical and Computer Engineering
Northeastern University
Boston, Massachusetts

Dimitris G. Manolakis

MIT Lincoln Laboratory
Lexington, Massachusetts

Upper Saddle River, New Jersey 07458

Contents

Preface

This book was developed based on our teaching of undergraduate- and graduate-level courses in digital signal processing over the past several years. In this book we present the fundamentals of discrete-time signals, systems, and modern digital processing as well as applications for students in electrical engineering, computer engineering, and computer science. The book is suitable for either a one-semester or a two-semester undergraduate-level course in discrete systems and digital signal processing. It is also intended for use in a one-semester first-year graduate-level course in digital signal processing.

It is assumed that the student has had undergraduate courses in advanced calculus (including ordinary differential equations) and linear systems for continuous-time signals, including an introduction to the Laplace transform. Although the Fourier series and Fourier transforms of periodic and aperiodic signals are described in Chapter 4, we expect that many students may have had this material in a prior course.

Balanced coverage of both theory and practical applications is provided. A large number of well-designed problems are provided to help the student in mastering the subject matter. A solutions manual is available for download for instructors only. Additionally, Microsoft PowerPoint slides of text figures are available for instructors on the publisher's website.

In the fourth edition of the book, we have added a new chapter on adaptive filters and have substantially modified and updated the chapters on multirate digital signal processing and on sampling and reconstruction of signals. We have also added new material on the discrete cosine transform.

In Chapter 1 we describe the operations involved in the analog-to-digital conversion of analog signals. The process of sampling a sinusoid is described in some detail and the problem of aliasing is explained. Signal quantization and digital-to-analog conversion are also described in general terms, but the analysis is presented in subsequent chapters.

Chapter 2 is devoted entirely to the characterization and analysis of linear time-invariant (shift-invariant) discrete-time systems and discrete-time signals in the time domain. The convolution sum is derived and systems are categorized according to the duration of their impulse response as a finite-duration impulse response (FIR) and as an infinite-duration impulse response (IIR). Linear time-invariant systems characterized by difference equations are presented and the solution of difference equations with initial conditions is obtained. The chapter concludes with a treatment of discrete-time correlation.

The z-transform is introduced in Chapter 3. Both the bilateral and the unilateral z-transforms are presented, and methods for determining the inverse z-transform are described. Use of the z-transform in the analysis of linear time-invariant systems is illustrated, and important properties of systems, such as causality and stability, are related to $z-$domain characteristics.

Chapter 4 treats the analysis of signals in the frequency domain. Fourier series and the Fourier transform are presented for both continuous-time and discrete-time signals.

In Chapter 5, linear time-invariant (LTI) discrete systems are characterized in the frequency domain by their frequency response function and their response to periodic and aperiodic signals is determined. A number of important types of discrete-time systems are described, including resonators, notch filters, comb filters, all-pass filters, and oscillators. The design of a number of simple FIR and IIR filters is also considered In addition, the student is introduced to the concepts of minimum-phase, mixed-phase, and maximum-phase systems and to the problem of deconvolution.

Chapter 6 provides a thorough treatment of sampling of continuous-time signals and the reconstruction of the signals from their samples. Our coverage includes the sampling and reconstruction of bandpass signals, the sampling of discrete-time signals, and A/D and D/A conversion. The chapter concludes with the treatment of oversampling A/D and D/A converters.

The DFT, its properties and its applications, are the topics covered in Chapter 7. Two methods are described for using the DFT to perform linear filtering. The use of the DFT to perform frequency analysis of signals is also described. The final topic treated in this chapter is the discrete cosine transform.

Chapter 8 covers the efficient computation of the DFT. Included in this chapter are descriptions of radix-2, radix-4, and split-radix fast Fourier transform (FFT) algorithms, and applications of the FFT algorithms to the computation of convolution and correlation. The Goertzel algorithm and the chirp-z transform are introduced as two methods for computing the DFT using linear filtering.

Chapter 9 treats the realization of IIR and FIR systems. This treatment includes direct-form, cascade, parallel, lattice, and lattice-ladder realizations. The chapter also examines quantization effects in a digital implementation of FIR and IIR systems.

Techniques for design of digital FIR and IIR filters are presented in Chapter 10. The design techniques include both direct methods in discrete time and methods involving the conversion of analog filters into digital filters by various transformations.

Chapter 11 treats sampling-rate conversion and its applications to multirate digital signal processing. In addition to describing decimation and interpolation by integer and rational factors, we describe methods for sampling-rate conversion by an arbitrary factor and implementations by polyphase filter structures. This chapter also treats digital filter banks, two-channel quadrature mirror filters (QMF) and M-channel QMF banks.

Linear prediction and optimum linear (Wiener) filters are treated in Chapter 12. Also included in this chapter are descriptions of the Levinson–Durbin algorithm and Schur algorithm for solving the normal equations, as well as the AR lattice and ARMA lattice-ladder filters.

Chapter 13 treats single-channel adaptive filters based on the LMS algorithm and on recursive least squares (RLS) algorithms. Both direct form FIR and lattice RLS algorithms and filter structures are described.

Power spectrum estimation is the main topic of Chapter 14. Our coverage includes a description of nonparametric and model-based (parametric) methods. Also described are eigen-decomposition-based methods, including MUSIC and ESPRIT.

A one-semester senior-level course for students who have had prior exposure to discrete systems can use the material in Chapters 1 through 5 for a quick review and then proceed to cover Chapters 6 through 10.

In a first-year graduate-level course in digital signal processing, the first six chapters provide the student with a good review of discrete-time systems. The instructor can move quickly through most of this material and then cover Chapters 7 through 11, followed by selected topics from Chapters 12 through 14.

Many examples throughout the book and approximately 500 homework problems are included throughout the book. Answers to selected problems appear in the back of the book. Many of the homework problems can be solved numerically on a computer, using a software package such as MATLAB®. Available for use as a self-study companion to the textbook is a student manual: *Student Manual for Digital Signal Processing with MATLAB®*. MATLAB is incorporated as the basic software tool for this manual. The instructor may also wish to consider the use of other supplementary books that contain computer-based exercises, such as *Computer-Based Exercises for Signal Processing Using MATLAB* (Prentice Hall, 1994) by C. S. Burrus *et al.*

The authors are indebted to their many faculty colleagues who have provided valuable suggestions through reviews of previous editions of this book. These include W. E. Alexander, G. Arslan, Y. Bresler, J. Deller, F. DePiero, V. Ingle, J.S. Kang, C. Keller, H. Lev-Ari, L. Merakos, W. Mikhael, P. Monticciolo, C. Nikias, M. Schetzen, E. Serpedin, T. M. Sullivan, H. Trussell, S. Wilson, and M. Zoltowski. We are also indebted to R. Price for recommending the inclusion of split-radix FFT algorithms and related suggestions. Finally, we wish to acknowledge the suggestions and comments of many former graduate students, and especially those by A. L. Kok, J. Lin, E. Sozer, and S. Srinidhi, who assisted in the preparation of several illustrations and the solutions manual.

JOHN G. PROAKIS
DIMITRIS G. MANOLAKIS

CHAPTER

1

Introduction

Digital signal processing is an area of science and engineering that has developed rapidly over the past 40 years. This rapid development is a result of the significant advances in digital computer technology and integrated-circuit fabrication. The digital computers and associated digital hardware of four decades ago were relatively large and expensive and, as a consequence, their use was limited to general-purpose non-real-time (off-line) scientific computations and business applications. The rapid developments in integrated-circuit technology, starting with medium-scale integration (MSI) and progressing to large-scale integration (LSI), and now, very-large-scale integration (VLSI) of electronic circuits has spurred the development of powerful, smaller, faster, and cheaper digital computers and special-purpose digital hardware. These inexpensive and relatively fast digital circuits have made it possible to construct highly sophisticated digital systems capable of performing complex digital signal processing functions and tasks, which are usually too difficult and/or too expensive to be performed by analog circuitry or analog signal processing systems. Hence many of the signal processing tasks that were conventionally performed by analog means are realized today by less expensive and often more reliable digital hardware.

We do not wish to imply that digital signal processing is the proper solution for all signal processing problems. Indeed, for many signals with extremely wide bandwidths, real-time processing is a requirement. For such signals, analog or, perhaps, optical signal processing is the only possible solution. However, where digital circuits are available and have sufficient speed to perform the signal processing, they are usually preferable.

Not only do digital circuits yield cheaper and more reliable systems for signal processing, they have other advantages as well. In particular, digital processing hardware allows programmable operations. Through software, one can more eas-

ily modify the signal processing functions to be performed by the hardware. Thus digital hardware and associated software provide a greater degree of flexibility in system design. Also, there is often a higher order of precision achievable with digital hardware and software compared with analog circuits and analog signal processing systems. For all these reasons, there has been an explosive growth in digital signal processing theory and applications over the past three decades.

In this book our objective is to present an introduction of the basic analysis tools and techniques for digital processing of signals. We begin by introducing some of the necessary terminology and by describing the important operations associated with the process of converting an analog signal to digital form suitable for digital processing. As we shall see, digital processing of analog signals has some drawbacks. First, and foremost, conversion of an analog signal to digital form, accomplished by sampling the signal and quantizing the samples, results in a distortion that prevents us from reconstructing the original analog signal from the quantized samples. Control of the amount of this distortion is achieved by proper choice of the sampling rate and the precision in the quantization process. Second, there are finite precision effects that must be considered in the digital processing of the quantized samples. While these important issues are considered in some detail in this book, the emphasis is on the analysis and design of digital signal processing systems and computational techniques.

1.1 Signals, Systems, and Signal Processing

A *signal* is defined as any physical quantity that varies with time, space, or any other independent variable or variables. Mathematically, we describe a signal as a function of one or more independent variables. For example, the functions

$$\begin{aligned} s_1(t) &= 5t \\ s_2(t) &= 20t^2 \end{aligned} \tag{1.1.1}$$

describe two signals, one that varies linearly with the independent variable t (time) and a second that varies quadratically with t. As another example, consider the function

$$s(x, y) = 3x + 2xy + 10y^2 \tag{1.1.2}$$

This function describes a signal of two independent variables x and y that could represent the two spatial coordinates in a plane.

The signals described by (1.1.1) and (1.1.2) belong to a class of signals that are precisely defined by specifying the functional dependence on the independent variable. However, there are cases where such a functional relationship is unknown or too highly complicated to be of any practical use.

For example, a speech signal (see Fig. 1.1.1) cannot be described functionally by expressions such as (1.1.1). In general, a segment of speech may be represented to

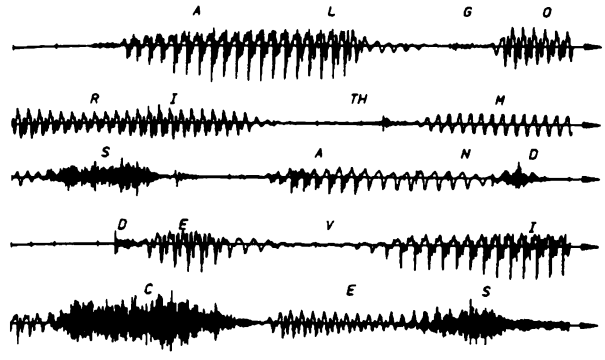


Figure 1.1.1
Example of a speech signal.

a high degree of accuracy as a sum of several sinusoids of different amplitudes and frequencies, that is, as

$$\sum_{i=1}^{N} A_i(t) \sin[2\pi F_i(t)t + \theta_i(t)] \tag{1.1.3}$$

where $\{A_i(t)\}$, $\{F_i(t)\}$, and $\{\theta_i(t)\}$ are the sets of (possibly time-varying) amplitudes, frequencies, and phases, respectively, of the sinusoids. In fact, one way to interpret the information content or message conveyed by any short time segment of the speech signal is to measure the amplitudes, frequencies, and phases contained in the short time segment of the signal.

Another example of a natural signal is an electrocardiogram (ECG). Such a signal provides a doctor with information about the condition of the patient's heart. Similarly, an electroencephalogram (EEG) signal provides information about the activity of the brain.

Speech, electrocardiogram, and electroencephalogram signals are examples of information-bearing signals that evolve as functions of a single independent variable, namely, time. An example of a signal that is a function of two independent variables is an image signal. The independent variables in this case are the spatial coordinates. These are but a few examples of the countless number of natural signals encountered in practice.

Associated with natural signals are the means by which such signals are generated. For example, speech signals are generated by forcing air through the vocal cords. Images are obtained by exposing a photographic film to a scene or an object. Thus signal generation is usually associated with a *system* that responds to a stimulus or force. In a speech signal, the system consists of the vocal cords and the vocal tract, also called the vocal cavity. The stimulus in combination with the system is called a *signal source*. Thus we have speech sources, images sources, and various other types of signal sources.

A *system* may also be defined as a physical device that performs an operation on a signal. For example, a filter used to reduce the noise and interference corrupting a desired information-bearing signal is called a system. In this case the filter performs some operation(s) on the signal, which has the effect of reducing (filtering) the noise and interference from the desired information-bearing signal.

When we pass a signal through a system, as in filtering, we say that we have processed the signal. In this case the processing of the signal involves filtering the noise and interference from the desired signal. In general, the system is characterized by the type of operation that it performs on the signal. For example, if the operation is linear, the system is called linear. If the operation on the signal is nonlinear, the system is said to be nonlinear, and so forth. Such operations are usually referred to as *signal processing*.

For our purposes, it is convenient to broaden the definition of a system to include not only physical devices, but also software realizations of operations on a signal. In digital processing of signals on a digital computer, the operations performed on a signal consist of a number of mathematical operations as specified by a software program. In this case, the program represents an implementation of the system in *software*. Thus we have a system that is realized on a digital computer by means of a sequence of mathematical operations; that is, we have a digital signal processing system realized in software. For example, a digital computer can be programmed to perform digital filtering. Alternatively, the digital processing on the signal may be performed by digital *hardware* (logic circuits) configured to perform the desired specified operations. In such a realization, we have a physical device that performs the specified operations. In a broader sense, a digital system can be implemented as a combination of digital hardware and software, each of which performs its own set of specified operations.

This book deals with the processing of signals by digital means, either in software or in hardware. Since many of the signals encountered in practice are analog, we will also consider the problem of converting an analog signal into a digital signal for processing. Thus we will be dealing primarily with digital systems. The operations performed by such a system can usually be specified mathematically. The method or set of rules for implementing the system by a program that performs the corresponding mathematical operations is called an *algorithm*. Usually, there are many ways or algorithms by which a system can be implemented, either in software or in hardware, to perform the desired operations and computations. In practice, we have an interest in devising algorithms that are computationally efficient, fast, and easily implemented. Thus a major topic in our study of digital signal processing is the discussion of efficient algorithms for performing such operations as filtering, correlation, and spectral analysis.

1.1.1 Basic Elements of a Digital Signal Processing System

Most of the signals encountered in science and engineering are analog in nature. That is, the signals are functions of a continuous variable, such as time or space, and usually take on values in a continuous range. Such signals may be processed directly by appropriate analog systems (such as filters, frequency analyzers, or frequency multipliers) for the purpose of changing their characteristics or extracting some desired information. In such a case we say that the signal has been processed directly in its analog form, as illustrated in Fig. 1.1.2. Both the input signal and the output signal are in analog form.

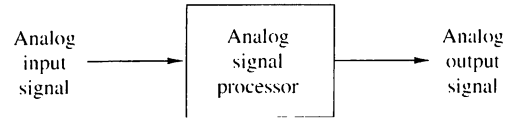


Figure 1.1.2
Analog signal processing.

Digital signal processing provides an alternative method for processing the analog signal, as illustrated in Fig. 1.1.3. To perform the processing digitally, there is a need for an interface between the analog signal and the digital processor. This interface is called an *analog-to-digital (A/D) converter.* The output of the A/D converter is a digital signal that is appropriate as an input to the digital processor.

The digital signal processor may be a large programmable digital computer or a small microprocessor programmed to perform the desired operations on the input signal. It may also be a hardwired digital processor configured to perform a specified set of operations on the input signal. Programmable machines provide the flexibility to change the signal processing operations through a change in the software, whereas hardwired machines are difficult to reconfigure. Consequently, programmable signal processors are in very common use. On the other hand, when signal processing operations are well defined, a hardwired implementation of the operations can be optimized, resulting in a cheaper signal processor and, usually, one that runs faster than its programmable counterpart. In applications where the digital output from the digital signal processor is to be given to the user in analog form, such as in speech communications, we must provide another interface from the digital domain to the analog domain. Such an interface is called a *digital-to-analog (D/A) converter.* Thus the signal is provided to the user in analog form, as illustrated in the block diagram of Fig. 1.1.3. However, there are other practical applications involving signal analysis, where the desired information is conveyed in digital form and no D/A converter is required. For example, in the digital processing of radar signals, the information extracted from the radar signal, such as the position of the aircraft and its speed, may simply be printed on paper. There is no need for a D/A converter in this case.

1.1.2 Advantages of Digital over Analog Signal Processing

There are many reasons why digital signal processing of an analog signal may be preferable to processing the signal directly in the analog domain, as mentioned briefly earlier. First, a digital programmable system allows flexibility in reconfiguring the digital signal processing operations simply by changing the program. Reconfigu-

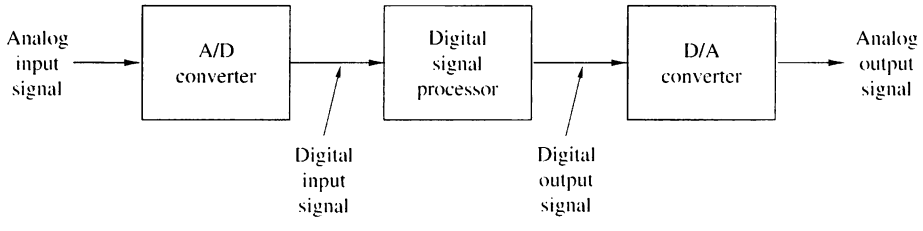


Figure 1.1.3 Block diagram of a digital signal processing system.

ration of an analog system usually implies a redesign of the hardware followed by testing and verification to see that it operates properly.

Accuracy considerations also play an important role in determining the form of the signal processor. Tolerances in analog circuit components make it extremely difficult for the system designer to control the accuracy of an analog signal processing system. On the other hand, a digital system provides much better control of accuracy requirements. Such requirements, in turn, result in specifying the accuracy requirements in the A/D converter and the digital signal processor, in terms of word length, floating-point versus fixed-point arithmetic, and similar factors.

Digital signals are easily stored on magnetic media (tape or disk) without deterioration or loss of signal fidelity beyond that introduced in the A/D conversion. As a consequence, the signals become transportable and can be processed off-line in a remote laboratory. The digital signal processing method also allows for the implementation of more sophisticated signal processing algorithms. It is usually very difficult to perform precise mathematical operations on signals in analog form but these same operations can be routinely implemented on a digital computer using software.

In some cases a digital implementation of the signal processing system is cheaper than its analog counterpart. The lower cost may be due to the fact that the digital hardware is cheaper, or perhaps it is a result of the flexibility for modifications provided by the digital implementation.

As a consequence of these advantages, digital signal processing has been applied in practical systems covering a broad range of disciplines. We cite, for example, the application of digital signal processing techniques in speech processing and signal transmission on telephone channels, in image processing and transmission, in seismology and geophysics, in oil exploration, in the detection of nuclear explosions, in the processing of signals received from outer space, and in a vast variety of other applications. Some of these applications are cited in subsequent chapters.

As already indicated, however, digital implementation has its limitations. One practical limitation is the speed of operation of A/D converters and digital signal processors. We shall see that signals having extremely wide bandwidths require fast-sampling-rate A/D converters and fast digital signal processors. Hence there are analog signals with large bandwidths for which a digital processing approach is beyond the state of the art of digital hardware.

1.2 Classification of Signals

The methods we use in processing a signal or in analyzing the response of a system to a signal depend heavily on the characteristic attributes of the specific signal. There are techniques that apply only to specific families of signals. Consequently, any investigation in signal processing should start with a classification of the signals involved in the specific application.

1.2.1 Multichannel and Multidimensional Signals

As explained in Section 1.1, a signal is described by a function of one or more independent variables. The value of the function (i.e., the dependent variable) can be

a real-valued scalar quantity, a complex-valued quantity, or perhaps a vector. For example, the signal

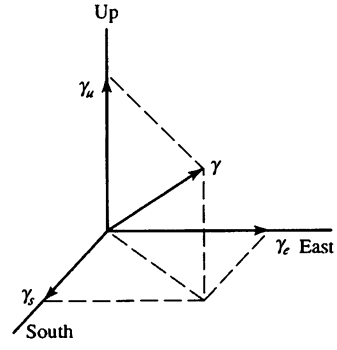
$$s_1(t) = A \sin 3\pi t$$

is a real-valued signal. However, the signal

$$s_2(t) = Ae^{j3\pi t} = A \cos 3\pi t + jA \sin 3\pi t$$

is complex valued.

In some applications, signals are generated by multiple sources or multiple sensors. Such signals, in turn, can be represented in vector form. Figure 1.2.1 shows the three components of a vector signal that represents the ground acceleration due to an earthquake. This acceleration is the result of three basic types of elastic waves. The primary (P) waves and the secondary (S) waves propagate within the body of



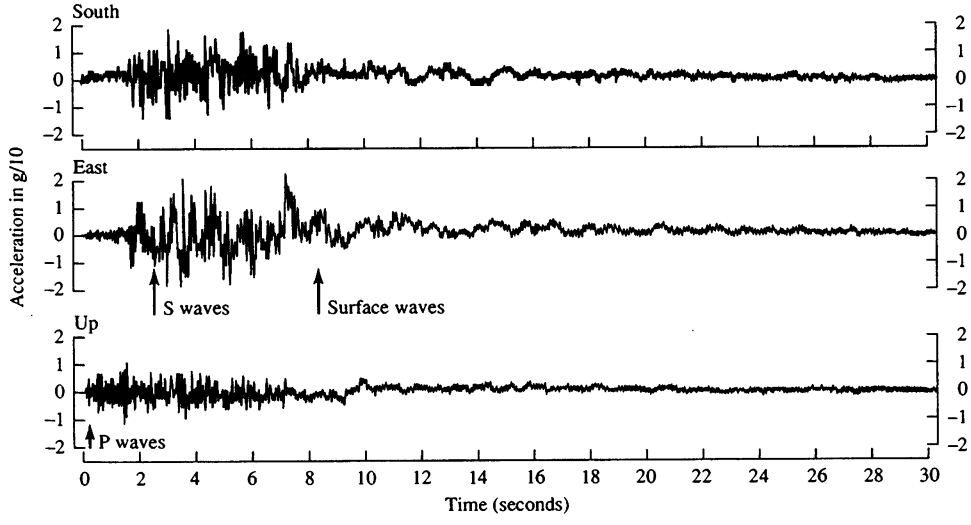


Figure 1.2.1 Three components of ground acceleration measured a few kilometers from the epicenter of an earthquake. (From *Earthquakes*, by B. A. Bold, ©1988 by W. H. Freeman and Company. Reprinted with permission of the publisher.)

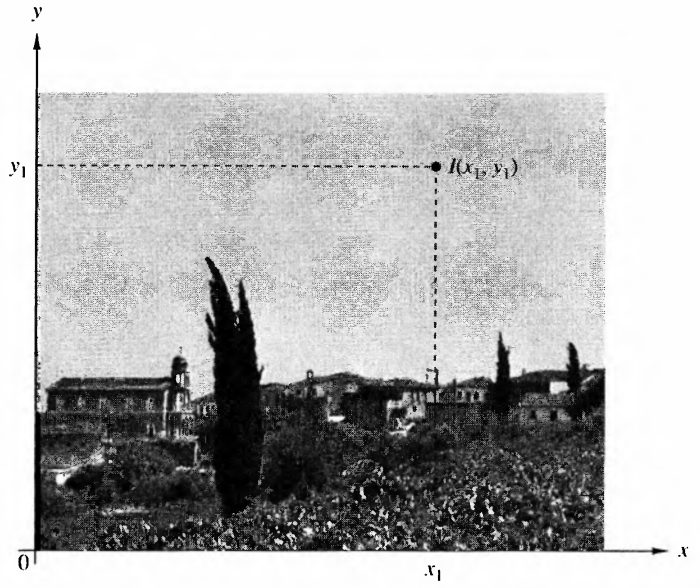


Figure 1.2.2
Example of a two-dimensional signal.

rock and are longitudinal and transversal, respectively. The third type of elastic wave is called the surface wave, because it propagates near the ground surface. If $s_k(t)$, $k = 1, 2, 3$, denotes the electrical signal from the kth sensor as a function of time, the set of $p = 3$ signals can be represented by a vector $\mathbf{S}_3(t)$, where

$$\mathbf{S}_3(t) = \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix}$$

We refer to such a vector of signals as a *multichannel signal*.In electrocardiography, for example, 3-lead and 12-lead electrocardiograms (ECG) are often used in practice, which result in 3-channel and 12-channel signals.

Let us now turn our attention to the independent variable(s). If the signal is a function of a single independent variable, the signal is called a *one-dimensional* signal. On the other hand, a signal is called *M-dimensional* if its value is a function of M independent variables.

The picture shown in Fig. 1.2.2 is an example of a two-dimensional signal, since the intensity or brightness $I(x, y)$ at each point is a function of two independent variables. On the other hand, a black-and-white television picture may be represented as $I(x, y, t)$ since the brightness is a function of time. Hence the TV picture may be treated as a three-dimensional signal. In contrast, a color TV picture may be described by three intensity functions of the form $I_r(x, y, t)$, $I_g(x, y, t)$, and $I_b(x, y, t)$, corresponding to the brightness of the three principal colors (red, green, blue) as functions of time. Hence the color TV picture is a three-channel, three-dimensional signal, which can be represented by the vector

$$\mathbf{I}(x, y, t) = \begin{bmatrix} I_r(x, y, t) \\ I_g(x, y, t) \\ I_b(x, y, t) \end{bmatrix}$$

In this book we deal mainly with single-channel, one-dimensional real- or complex-valued signals and we refer to them simply as signals. In mathematical terms these signals are described by a function of a single independent variable. Although the independent variable need not be time, it is common practice to use t as the independent variable. In many cases the signal processing operations and algorithms developed in this text for one-dimensional, single-channel signals can be extended to multichannel and multidimensional signals.

1.2.2 Continuous-Time Versus Discrete-Time Signals

Signals can be further classified into four different categories depending on the characteristics of the time (independent) variable and the values they take. *Continuous-time signals* or *analog signals* are defined for every value of time and they take on values in the continuous interval (a, b), where a can be $-\infty$ and b can be ∞. Mathematically, these signals can be described by functions of a continuous variable. The speech waveform in Fig. 1.1.1 and the signals $x_1(t) = \cos \pi t$, $x_2(t) = e^{-|t|}$, $-\infty < t < \infty$ are examples of analog signals. *Discrete-time signals*are defined only at certain specific values of time. These time instants need not be equidistant, but in practice they are usually taken at equally spaced intervals for computational convenience and mathematical tractability. The signal $x(t_n) = e^{-|t_n|}$, $n = 0, \pm 1, \pm 2, \ldots$ provides an example of a discrete-time signal. If we use the index n of the discrete-time instants as the independent variable, the signal value becomes a function of an integer variable (i.e., a sequence of numbers). Thus a discrete-time signal can be represented mathematically by a sequence of real or complex numbers. To emphasize the discrete-time nature of a signal, we shall denote such a signal as $x(n)$ instead of $x(t)$. If the time instants t_n are equally spaced (i.e., $t_n = nT$), the notation $x(nT)$ is also used. For example, the sequence

$$x(n) = \begin{cases} 0.8^n, & \text{if } n \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1.2.1}$$

is a discrete-time signal, which is represented graphically as in Fig. 1.2.3.

In applications, discrete-time signals may arise in two ways:

1. By selecting values of an analog signal at discrete-time instants. This process is called *sampling* and is discussed in more detail in Section 1.4. All measuring instruments that take measurements at a regular interval of time provide

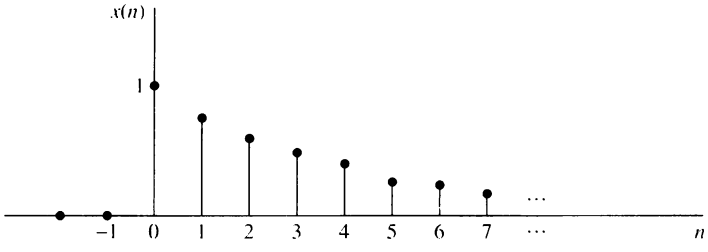


Figure 1.2.3 Graphical representation of the discrete time signal $x(n) = 0.8^n$ for $n > 0$ and $x(n) = 0$ for $n < 0$.

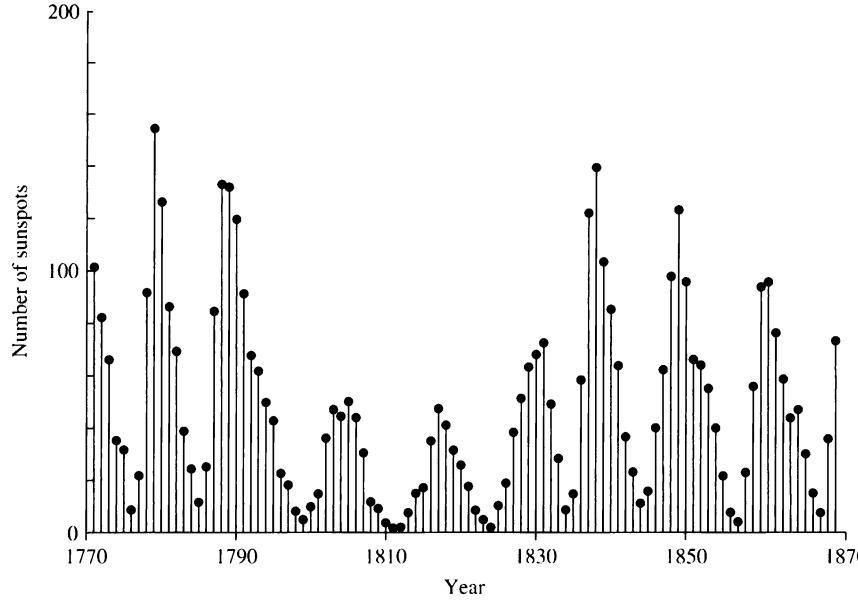


Figure 1.2.4 Wölfer annual sunspot numbers (1770–1869).

discrete-time signals. For example, the signal $x(n)$ in Fig. 1.2.3 can be obtained by sampling the analog signal $x(t) = 0.8^t$, $t \geq 0$ and $x(t) = 0$, $t < 0$ once every second.

2. By accumulating a variable over a period of time. For example, counting the number of cars using a given street every hour, or recording the value of gold every day, results in discrete-time signals. Figure 1.2.4 shows a graph of the Wölfer sunspot numbers. Each sample of this discrete-time signal provides the number of sunspots observed during an interval of 1 year.

1.2.3 Continuous-Valued Versus Discrete-Valued Signals

The values of a continuous-time or discrete-time signal can be continuous or discrete. If a signal takes on all possible values on a finite or an infinite range, it is said to be a continuous-valued signal. Alternatively, if the signal takes on values from a finite set of possible values, it is said to be a discrete-valued signal. Usually, these values are equidistant and hence can be expressed as an integer multiple of the distance between two successive values. A discrete-time signal having a set of discrete values is called a *digital signal*. Figure 1.2.5 shows a digital signal that takes on one of four possible values.

In order for a signal to be processed digitally, it must be discrete in time and its values must be discrete (i.e., it must be a digital signal). If the signal to be processed is in analog form, it is converted to a digital signal by sampling the analog signal at discrete instants in time, obtaining a discrete-time signal, and then by *quantizing* its values to a set of discrete values, as described later in the chapter. The process

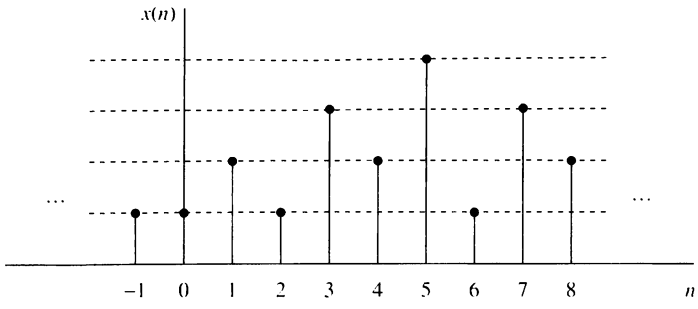


Figure 1.2.5 Digital signal with four different amplitude values.

of converting a continuous-valued signal into a discrete-valued signal, called *quantization*, is basically an approximation process. It may be accomplished simply by rounding or truncation. For example, if the allowable signal values in the digital signal are integers, say 0 through 15, the continuous-value signal is quantized into these integer values. Thus the signal value 8.58 will be approximated by the value 8 if the quantization process is performed by truncation or by 9 if the quantization process is performed by rounding to the nearest integer. An explanation of the analog-to-digital conversion process is given later in the chapter.

1.2.4 Deterministic Versus Random Signals

The mathematical analysis and processing of signals requires the availability of a mathematical description for the signal itself. This mathematical description, often referred to as the *signal model*, leads to another important classification of signals. Any signal that can be uniquely described by an explicit mathematical expression, a table of data, or a well-defined rule is called *deterministic*. This term is used to emphasize the fact that all past, present, and future values of the signal are known precisely, without any uncertainty.

In many practical applications, however, there are signals that either cannot be described to any reasonable degree of accuracy by explicit mathematical formulas, or such a description is too complicated to be of any practical use. The lack of such a relationship implies that such signals evolve in time in an unpredictable manner. We refer to these signals as *random*.The output of a noise generator, the seismic signal of Fig. 1.2.1, and the speech signal in Fig. 1.1.1 are examples of random signals.

The mathematical framework for the theoretical analysis of random signals is provided by the theory of probability and stochastic processes. Some basic elements of this approach, adapted to the needs of this book, are presented in Section 12.1.

It should be emphasized at this point that the classification of a *real-world* signal as deterministic or random is not always clear. Sometimes, both approaches lead to meaningful results that provide more insight into signal behavior. At other times, the wrong classification may lead to erroneous results, since some mathematical tools may apply only to deterministic signals while others may apply only to random signals. This will become clearer as we examine specific mathematical tools.

1.3 The Concept of Frequency in Continuous-Time and Discrete-Time Signals

The concept of frequency is familiar to students in engineering and the sciences. This concept is basic in, for example, the design of a radio receiver, a high-fidelity system, or a spectral filter for color photography. From physics we know that frequency is closely related to a specific type of periodic motion called harmonic oscillation, which is described by sinusoidal functions. The concept of frequency is directly related to the concept of time. Actually, it has the dimension of inverse time. Thus we should expect that the nature of time (continuous or discrete) would affect the nature of the frequency accordingly.

1.3.1 Continuous-Time Sinusoidal Signals

A simple harmonic oscillation is mathematically described by the following continuous-time sinusoidal signal:

$$x_a(t) = A\cos(\Omega t + \theta), \quad -\infty < t < \infty \tag{1.3.1}$$

shown in Fig. 1.3.1. The subscript a used with $x(t)$ denotes an analog signal. This signal is completely characterized by three parameters: A is the *amplitude* of the sinusoid, Ω is the *frequency* in radians per second (rad/s), and θ is the *phase* in radians. Instead of Ω, we often use the frequency F in cycles per second or hertz (Hz), where

$$\Omega = 2\pi F \tag{1.3.2}$$

In terms of F, (1.3.1) can be written as

$$x_a(t) = A\cos(2\pi F t + \theta), \quad -\infty < t < \infty \tag{1.3.3}$$

We will use both forms, (1.3.1) and (1.3.3), in representing sinusoidal signals.

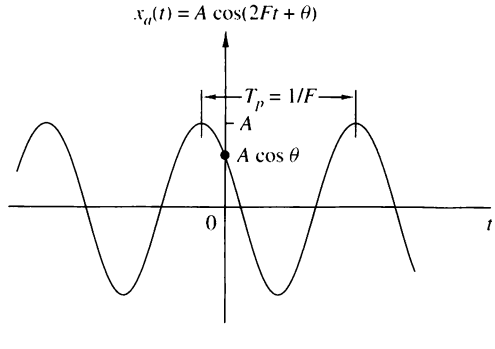


Figure 1.3.1
Example of an analog sinusoidal signal.

The analog sinusoidal signal in (1.3.3) is characterized by the following properties:

A1. For every fixed value of the frequency F, $x_a(t)$ is periodic. Indeed, it can easily be shown, using elementary trigonometry, that

$$x_a(t + T_p) = x_a(t)$$

where $T_p = 1/F$ is the fundamental period of the sinusoidal signal.

A2. Continuous-time sinusoidal signals with distinct (different) frequencies are themselves distinct.

A3. Increasing the frequency F results in an increase in the rate of oscillation of the signal, in the sense that more periods are included in a given time interval.

We observe that for $F = 0$, the value $T_p = \infty$ is consistent with the fundamental relation $F = 1/T_p$. Due to continuity of the time variable t, we can increase the frequency F, without limit, with a corresponding increase in the rate of oscillation.

The relationships we have described for sinusoidal signals carry over to the class of complex exponential signals

$$x_a(t) = Ae^{j(\Omega t+\theta)} \tag{1.3.4}$$

This can easily be seen by expressing these signals in terms of sinusoids using the Euler identity

$$e^{\pm j\phi} = \cos\phi \pm j \sin\phi \tag{1.3.5}$$

By definition, frequency is an inherently positive physical quantity. This is obvious if we interpret frequency as the number of cycles per unit time in a periodic signal. However, in many cases, only for mathematical convenience, we need to introduce negative frequencies. To see this we recall that the sinusoidal signal (1.3.1) may be expressed as

$$x_a(t) = A\cos(\Omega t + \theta) = \frac{A}{2} e^{j(\Omega t+\theta)} + \frac{A}{2} e^{-j(\Omega t+\theta)} \tag{1.3.6}$$

which follows from (1.3.5). Note that a sinusoidal signal can be obtained by adding two equal-amplitude complex-conjugate exponential signals, sometimes called phasors, illustrated in Fig. 1.3.2. As time progresses the phasors rotate in opposite directions with angular frequencies $\pm\Omega$ radians per second. Since a *positive frequency* corresponds to counterclockwise uniform angular motion, a *negative frequency* simply corresponds to clockwise angular motion.

For mathematical convenience, we use both negative and positive frequencies throughout this book. Hence the frequency range for analog sinusoids is $-\infty < F < \infty$.

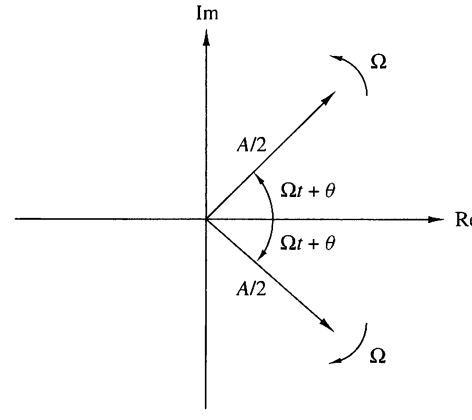


Figure 1.3.2
Representation of a cosine function by a pair of complex-conjugate exponentials (phasors).

1.3.2 Discrete-Time Sinusoidal Signals

A discrete-time sinusoidal signal may be expressed as

$$x(n) = A\cos(\omega n + \theta), \quad -\infty < n < \infty \tag{1.3.7}$$

where n is an integer variable, called the sample number, A is the *amplitude* of the sinusoid, ω is the *frequency* in radians per sample, and θ is the *phase* in radians.

If instead of ω we use the frequency variable f defined by

$$\omega \equiv 2\pi f \tag{1.3.8}$$

the relation (1.3.7) becomes

$$x(n) = A\cos(2\pi f n + \theta), \quad -\infty < n < \infty \tag{1.3.9}$$

The frequency f has dimensions of cycles per sample. In Section 1.4, where we consider the sampling of analog sinusoids, we relate the frequency variable f of a discrete-time sinusoid to the frequency F in cycles per second for the analog sinusoid. For the moment we consider the discrete-time sinusoid in (1.3.7) independently of the continuous-time sinusoid given in (1.3.1). Figure 1.3.3 shows a sinusoid with frequency $\omega = \pi/6$ radians per sample ($f = \frac{1}{12}$ cycles per sample) and phase $\theta = \pi/3$.

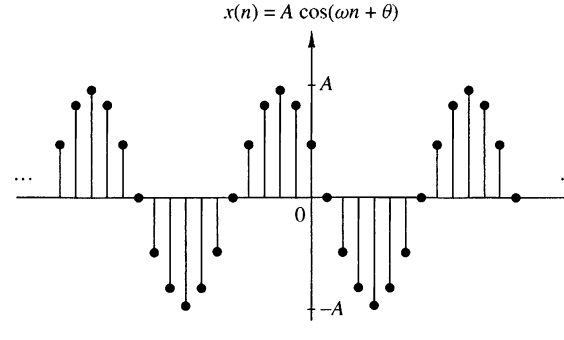


Figure 1.3.3
Example of a discrete-time sinusoidal signal ($\omega = \pi/6$ and $\theta = \pi/3$).

In contrast to continuous-time sinusoids, the discrete-time sinusoids are characterized by the following properties:

B1. *A discrete-time sinusoid is periodic only if its frequency f is a rational number.*

By definition, a discrete-time signal $x(n)$ is periodic with period $N (N > 0)$ if and only if

$$x(n + N) = x(n) \qquad \text{for all } n \tag{1.3.10}$$

The smallest value of N for which (1.3.10) is true is called the *fundamental period.*

The proof of the periodicity property is simple. For a sinusoid with frequency f_0 to be periodic, we should have

$$\cos[2\pi f_0(N + n) + \theta] = \cos(2\pi f_0 n + \theta)$$

This relation is true if and only if there exists an integer k such that

$$2\pi f_0 N = 2k\pi$$

or, equivalently,

$$f_0 = \frac{k}{N} \tag{1.3.11}$$

According to (1.3.11), a discrete-time sinusoidal signal is periodic only if its frequency f_0 can be expressed as the ratio of two integers (i.e., f_0 is rational).

To determine the fundamental period N of a periodic sinusoid, we express its frequency f_0 as in (1.3.11) and cancel common factors so that k and N are relatively prime. Then the fundamental period of the sinusoid is equal to N. Observe that a small change in frequency can result in a large change in the period. For example, note that $f_1 = 31/60$ implies that $N_1 = 60$, whereas $f_2 = 30/60$ results in $N_2 = 2$.

B2. *Discrete-time sinusoids whose frequencies are separated by an integer multiple of 2π are identical.*

To prove this assertion, let us consider the sinusoid $\cos(\omega_0 n + \theta)$. It easily follows that

$$\cos[(\omega_0 + 2\pi)n + \theta] = \cos(\omega_0 n + 2\pi n + \theta) = \cos(\omega_0 n + \theta) \tag{1.3.12}$$

As a result, all sinusoidal sequences

$$x_k(n) = A\cos(\omega_k n + \theta), \qquad k = 0, 1, 2, \ldots \tag{1.3.13}$$

where

$$\omega k = \omega_0 + 2k\pi, \qquad -\pi \leq \omega_0 \leq \pi$$

are *indistinguishable* (i.e., *identical*). Any sequence resulting from a sinusoid with a frequency $|\omega| > \pi$, or $|f| > \frac{1}{2}$, is identical to a sequence obtained from a sinusoidal signal with frequency $|\omega| < \pi$. Because of this similarity, we call the sinusoid having the frequency $|\omega| > \pi$ an *alias* of a corresponding sinusoid with frequency $|\omega| < \pi$. Thus we regard frequencies in the range $-\pi \leq \omega \leq \pi$, or $-\frac{1}{2} \leq f \leq \frac{1}{2}$, as unique

and all frequencies $|\omega| > \pi$, or $|f| > \frac{1}{2}$, as aliases. The reader should notice the difference between discrete-time sinusoids and continuous-time sinusoids, where the latter result in distinct signals for Ω or F in the entire range $-\infty < \Omega < \infty$ or $-\infty < F < \infty$.

B3. *The highest rate of oscillation in a discrete-time sinusoid is attained when* $\omega = \pi$ *(or* $\omega = -\pi$*) or, equivalently,* $f = \frac{1}{2}$ *(or* $f = -\frac{1}{2}$*).*

To illustrate this property, let us investigate the characteristics of the sinusoidal signal sequence

$$x(n) = \cos \omega_0 n$$

when the frequency varies from 0 to π. To simplify the argument, we take values of $\omega_0 = 0, \pi/8, \pi/4, \pi/2, \pi$ corresponding to $f = 0, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$, which result in periodic sequences having periods $N = \infty, 16, 8, 4, 2$, as depicted in Fig. 1.3.4. We note that the period of the sinusoid decreases as the frequency increases. In fact, we can see that the rate of oscillation increases as the frequency increases.

To see what happens for $\pi \leq \omega_0 \leq 2\pi$, we consider the sinusoids with frequencies $\omega_1 = \omega_0$ and $\omega_2 = 2\pi - \omega_0$. Note that as ω_1 varies from π to 2π, ω_2 varies from π to 0. It can be easily seen that

$$\begin{aligned} x_1(n) &= A \cos \omega_1 n = A \cos \omega_0 n \\ x_2(n) &= A \cos \omega_2 n = A \cos(2\pi - \omega_0) n \\ &= A \cos(-\omega_0 n) = x_1(n) \end{aligned} \tag{1.3.14}$$

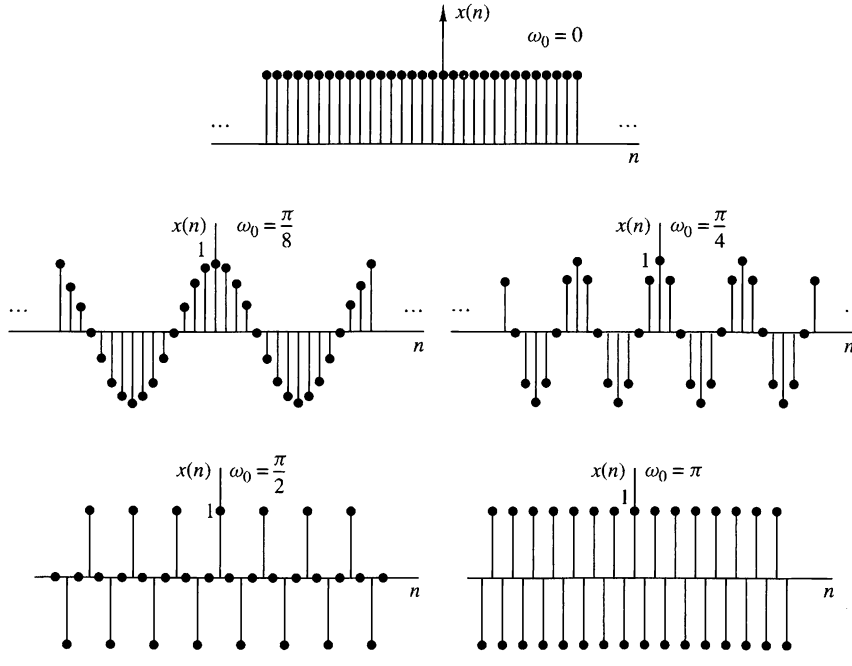


Figure 1.3.4 Signal $x(n) = \cos \omega_0 n$ for various values of the frequency ω_0.

Hence ω_2 is an alias of ω_1. If we had used a sine function instead of a cosine function, the result would basically be the same, except for a 180° phase difference between the sinusoids $x_1(n)$ and $x_2(n)$. In any case, as we increase the relative frequency ω_0 of a discrete-time sinusoid from π to 2π, its rate of oscillation decreases. For $\omega_0 = 2\pi$ the result is a constant signal, as in the case for $\omega_0 = 0$. Obviously, for $\omega_0 = \pi$ (or $f = \frac{1}{2}$) we have the highest rate of oscillation.

As for the case of continuous-time signals, negative frequencies can be introduced as well for discrete-time signals. For this purpose we use the identity

$$x(n) = A\cos(\omega n + \theta) = \frac{A}{2}e^{j(\omega n+\theta)} + \frac{A}{2}e^{-j(\omega n+\theta)} \tag{1.3.15}$$

Since discrete-time sinusoidal signals with frequencies that are separated by an integer multiple of 2π are identical, it follows that the frequencies in any interval $\omega_1 \le \omega \le \omega_1 + 2\pi$ constitute *all* the existing discrete-time sinusoids or complex exponentials. Hence the frequency range for discrete-time sinusoids is finite with duration 2π. Usually, we choose the range $0 \le \omega \le 2\pi$ or $-\pi \le \omega \le \pi$ ($0 \le f \le 1$, $-\frac{1}{2} \le f \le \frac{1}{2}$), which we call the *fundamental range.*

1.3.3 Harmonically Related Complex Exponentials

Sinusoidal signals and complex exponentials play a major role in the analysis of signals and systems. In some cases we deal with sets of *harmonically related* complex exponentials (or sinusoids). These are sets of periodic complex exponentials with fundamental frequencies that are multiples of a single positive frequency. Although we confine our discussion to complex exponentials, the same properties clearly hold for sinusoidal signals. We consider harmonically related complex exponentials in both continuous time and discrete time.

Continuous-time exponentials. The basic signals for continuous-time, harmonically related exponentials are

$$s_k(t) = e^{jk\Omega_0 t} = e^{j2\pi k F_0 t} \qquad k = 0, \pm 1, \pm 2, \ldots \tag{1.3.16}$$

We note that for each value of k, $s_k(t)$ is periodic with fundamental period $1/(kF_0) = T_p/k$ or fundamental frequency kF_0. Since a signal that is periodic with period T_p/k is also periodic with period $k(T_p/k) = T_p$ for any positive integer k, we see that all of the $s_k(t)$ have a common period of T_p. Furthermore, according to Section 1.3.1, F_0 is allowed to take any value and all members of the set are distinct, in the sense that if $k_1 \neq k_2$, then $s_{k1}(t) \neq s_{k2}(t)$.

From the basic signals in (1.3.16) we can construct a linear combination of harmonically related complex exponentials of the form

$$x_a(t) = \sum_{k=-\infty}^{\infty} c_k s_k(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\Omega_0 t} \tag{1.3.17}$$

where c_k, $k = 0, \pm 1, \pm 2, \ldots$ are arbitrary complex constants. The signal $x_a(t)$ is periodic with fundamental period $T_p = 1/F_0$, and its representation in terms of

(1.3.17) is called the *Fourier series* expansion for $x_a(t)$. The complex-valued constants are the Fourier series coefficients and the signal $s_k(t)$ is called the kth harmonic of $x_a(t)$.

Discrete-time exponentials. Since a discrete-time complex exponential is periodic if its relative frequency is a rational number, we choose $f_0 = 1/N$ and we define the sets of harmonically related complex exponentials by

$$s_k(n) = e^{j2\pi k f_0 n}, \qquad k = 0, \pm 1, \pm 2, \ldots \tag{1.3.18}$$

In contrast to the continuous-time case, we note that

$$s_{k+N}(n) = e^{j2\pi n(k+N)/N} = e^{j2\pi n} s_k(n) = sk(n)$$

This means that, consistent with (1.3.10), there are only N distinct periodic complex exponentials in the set described by (1.3.18). Furthermore, all members of the set have a common period of N samples. Clearly, we can choose any consecutive N complex exponentials, say from $k = n_0$ to $k = n_0 + N - 1$, to form a harmonically related set with fundamental frequency $f_0 = 1/N$. Most often, for convenience, we choose the set that corresponds to $n_0 = 0$, that is, the set

$$s_k(n) = e^{j2\pi kn/N}, \qquad k = 0, 1, 2, \ldots, N-1 \tag{1.3.19}$$

As in the case of continuous-time signals, it is obvious that the linear combination

$$x(n) = \sum_{k=0}^{N-1} c_k s_k(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \tag{1.3.20}$$

results in a periodic signal with fundamental period N. As we shall see later, this is the Fourier series representation for a periodic discrete-time sequence with Fourier coefficients $\{c_k\}$. The sequence $s_k(n)$ is called the kth harmonic of $x(n)$.

EXAMPLE 1.3.1

Stored in the memory of a digital signal processor is one cycle of the sinusoidal signal

$$x(n) = \sin\left(\frac{2\pi n}{N} + \theta\right)$$

where $\theta = 2\pi q/N$, where q and N are integers.

(a) Determine how this table of values can be used to obtain values of harmonically related sinusoids having the same phase.

(b) Determine how this table can be used to obtain sinusoids of the same frequency but different phase.

Solution.

(a) Let $x_k(n)$ denote the sinusoidal signal sequence

$$x_k(n) = \sin\left(\frac{2\pi nk}{N} + \theta\right)$$

This is a sinusoid with frequency $f_k = k/N$, which is harmonically related to $x(n)$. But $x_k(n)$ may be expressed as

$$\begin{aligned} x_k(n) &= \sin\left[\frac{2\pi(kn)}{N} + \theta\right] \\ &= x(kn) \end{aligned}$$

Thus we observe that $x_k(0) = x(0)$, $x_k(1) = x(k)$, $x_k(2) = x(2k)$, and so on. Hence the sinusoidal sequence $x_k(n)$ can be obtained from the table of values of $x(n)$ by taking every kth value of $x(n)$, beginning with $x(0)$. In this manner we can generate the values of all harmonically related sinusoids with frequencies $f_k = k/N$ for $k = 0, 1, \ldots, N-1$.

(b) We can control the phase θ of the sinusoid with frequency $f_k = k/N$ by taking the first value of the sequence from memory location $q = \theta N/2\pi$, where q is an integer. Thus the initial phase θ controls the starting location in the table and we wrap around the table each time the index (kn) exceeds N.

1.4 Analog-to-Digital and Digital-to-Analog Conversion

Most signals of practical interest, such as speech, biological signals, seismic signals, radar signals, sonar signals, and various communications signals such as audio and video signals, are analog. To process analog signals by digital means, it is first necessary to convert them into digital form, that is, to convert them to a sequence of numbers having finite precision. This procedure is called *analog-to-digital (A/D) conversion*, and the corresponding devices are called *A/D converters* (*ADCs*).

Conceptually, we view A/D conversion as a three-step process. This process is illustrated in Fig. 1.4.1.

1. *Sampling.* This is the conversion of a continuous-time signal into a discrete-time signal obtained by taking "samples" of the continuous-time signal at discrete-time instants. Thus, if $x_a(t)$ is the input to the sampler, the output is $x_a(nT) \equiv x(n)$, where T is called the *sampling interval.*

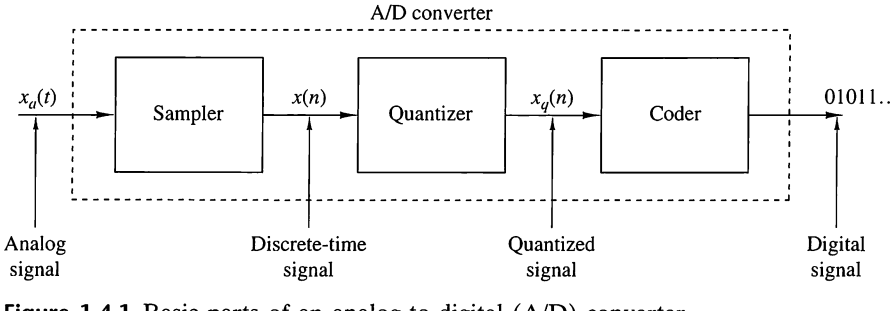


Figure 1.4.1 Basic parts of an analog-to-digital (A/D) converter.

2. *Quantization.* This is the conversion of a discrete-time continuous-valued signal into a discrete-time, discrete-valued (digital) signal. The value of each signal sample is represented by a value selected from a finite set of possible values. The difference between the unquantized sample $x(n)$ and the quantized output $x_q(n)$ is called the quantization error.
3. *Coding.* In the coding process, each discrete value $x_q(n)$ is represented by a b-bit binary sequence.

Although we model the A/D converter as a sampler followed by a quantizer and coder, in practice the A/D conversion is performed by a single device that takes $x_a(t)$ and produces a binary-coded number. The operations of sampling and quantization can be performed in either order but, in practice, sampling is always performed before quantization.

In many cases of practical interest (e.g., speech processing) it is desirable to convert the processed digital signals into analog form. (Obviously, we cannot listen to the sequence of samples representing a speech signal or see the numbers corresponding to a TV signal.) The process of converting a digital signal into an analog signal is known as *digital-to-analog (D/A) conversion.* All D/A converters "connect the dots" in a digital signal by performing some kind of interpolation, whose accuracy depends on the quality of the D/A conversion process. Figure 1.4.2 illustrates a simple form of D/A conversion, called a zero-order hold or a staircase approximation. Other approximations are possible, such as linearly connecting a pair of successive samples (linear interpolation), fitting a quadratic through three successive samples (quadratic interpolation), and so on. Is there an optimum (ideal) interpolator? For signals having a *limited frequency content* (finite bandwidth), the sampling theorem introduced in the following section specifies the optimum form of interpolation.

Sampling and quantization are treated in this section. In particular, we demonstrate that sampling does not result in a loss of information, nor does it introduce distortion in the signal if the signal bandwidth is finite. In principle, the analog signal can be reconstructed from the samples, provided that the sampling rate is sufficiently high to avoid the problem commonly called *aliasing.* On the other hand, quantization

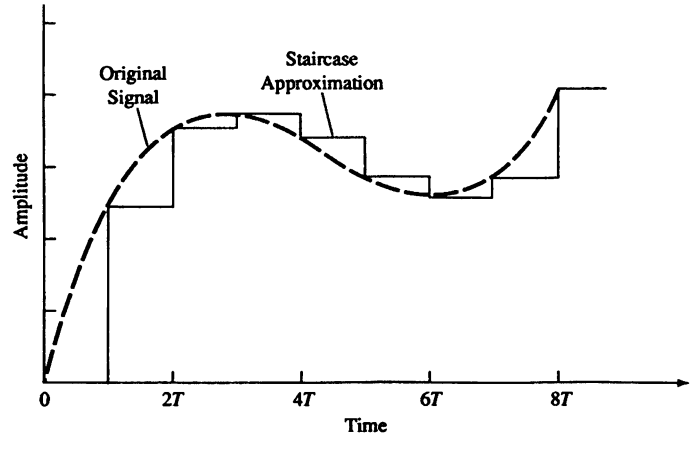


Figure 1.4.2
Zero-order hold digital-to-analog (D/A) conversion.

is a noninvertible or irreversible process that results in signal distortion. We shall show that the amount of distortion is dependent on the accuracy, as measured by the number of bits, in the A/D conversion process. The factors affecting the choice of the desired accuracy of the A/D converter are cost and sampling rate. In general, the cost increases with an increase in accuracy and/or sampling rate.

1.4.1 Sampling of Analog Signals

There are many ways to sample an analog signal. We limit our discussion to *periodic* or *uniform sampling*, which is the type of sampling used most often in practice. This is described by the relation

$$x(n) = x_a(nT), \qquad -\infty < n < \infty \tag{1.4.1}$$

where $x(n)$ is the discrete-time signal obtained by "taking samples" of the analog signal $x_a(t)$ every T seconds. This procedure is illustrated in Fig. 1.4.3. The time interval T between successive samples is called the *sampling period* or *sample interval* and its reciprocal $1/T = F_s$ is called the *sampling rate* (samples per second) or the *sampling frequency* (hertz).

Periodic sampling establishes a relationship between the time variables t and n of continuous-time and discrete-time signals, respectively. Indeed, these variables are linearly related through the sampling period T or, equivalently, through the sampling rate $F_s = 1/T$, as

$$t = nT = \frac{n}{F_s} \tag{1.4.2}$$

As a consequence of (1.4.2), there exists a relationship between the frequency variable F (or Ω) for analog signals and the frequency variable f (or ω) for discrete-time signals. To establish this relationship, consider an analog sinusoidal signal of the form

$$x_a(t) = A\cos(2\pi Ft + \theta) \tag{1.4.3}$$

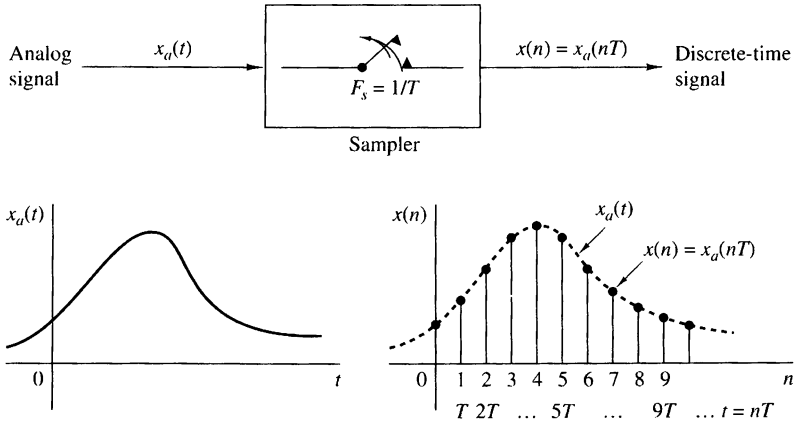


Figure 1.4.3 Periodic sampling of an analog signal.

which, when sampled periodically at a rate $F_s = 1/T$ samples per second, yields

$$\begin{aligned} x_a(nT) \equiv x(n) &= A\cos(2\pi FnT + \theta) \\ &= A\cos\left(\frac{2\pi nF}{F_s} + \theta\right) \end{aligned} \tag{1.4.4}$$

If we compare (1.4.4) with (1.3.9), we note that the frequency variables F and f are linearly related as

$$f = \frac{F}{F_s} \tag{1.4.5}$$

or, equivalently, as

$$\omega = \Omega T \tag{1.4.6}$$

The relation in (1.4.5) justifies the name *relative* or*normalized frequency*, which is sometimes used to describe the frequency variable f. As (1.4.5) implies, we can use f to determine the frequency F in hertz only if the sampling frequency F_s is known.

We recall from Section 1.3.1 that the ranges of the frequency variables F or Ω for continuous-time sinusoids are

$$\begin{aligned} -\infty < F < \infty \\ -\infty < \Omega < \infty \end{aligned} \tag{1.4.7}$$

However, the situation is different for discrete-time sinusoids. From Section 1.3.2 we recall that

$$\begin{aligned} -\frac{1}{2} < f < \frac{1}{2} \\ -\pi < \omega < \pi \end{aligned} \tag{1.4.8}$$

By substituting from (1.4.5) and (1.4.6) into (1.4.8), we find that the frequency of the continuous-time sinusoid when sampled at a rate $F_s = 1/T$ must fall in the range

$$-\frac{1}{2T} = -\frac{F_s}{2} \le F \le \frac{F_s}{2} = \frac{1}{2T} \tag{1.4.9}$$

or, equivalently,

$$-\frac{\pi}{T} = -\pi F_s \le \Omega \le \pi F_s = \frac{\pi}{T} \tag{1.4.10}$$

These relations are summarized in Table 1.1.

From these relations we observe that the fundamental difference between continuous-time and discrete-time signals is in their range of values of the frequency variables F and f, or Ω and ω. Periodic sampling of a continuous-time signal implies a mapping of the infinite frequency range for the variable F (or Ω) into a finite frequency range for the variable f (or ω). Since the highest frequency in a

TABLE 1.1 Relations Among Frequency Variables

Continuous-time signals		Discrete-time signals
$\Omega = 2\pi F$		$\omega = 2\pi f$
$\frac{\text{radians}}{\text{sec}}$ Hz		$\frac{\text{radians}}{\text{sample}}$ $\frac{\text{cycles}}{\text{sample}}$
	$\omega = \Omega T, f = F/F_s$	$-\pi \le \omega \le \pi$
		$-\frac{1}{2} \le f \le \frac{1}{2}$
	$\Omega = \omega/T, F = f \cdot F_s$	
$-\infty < \Omega < \infty$		$-\pi/T \le \Omega \le \pi/T$
$-\infty < F < \infty$		$-F_s/2 \le F \le F_s/2$

discrete-time signal is $\omega = \pi$ or $f = \frac{1}{2}$, it follows that, with a sampling rate F_s, the corresponding highest values of F and Ω are

$$\begin{aligned} F_{\max} &= \frac{F_s}{2} = \frac{1}{2T} \\ \Omega_{\max} &= \pi F_s = \frac{\pi}{T} \end{aligned} \tag{1.4.11}$$

Therefore, sampling introduces an ambiguity, since the highest frequency in a continuous-time signal that can be uniquely distinguished when such a signal is sampled at a rate $F_s = 1/T$ is $F_{\max} = F_s/2$, or $\Omega_{\max} = \pi F_s$. To see what happens to frequencies above $F_s/2$, let us consider the following example.

EXAMPLE 1.4.1

The implications of these frequency relations can be fully appreciated by considering the two analog sinusoidal signals

$$\begin{aligned} x_1(t) &= \cos 2\pi(10)t \\ x_2(t) &= \cos 2\pi(50)t \end{aligned} \tag{1.4.12}$$

which are sampled at a rate $F_s = 40$ Hz. The corresponding discrete-time signals or sequences are

$$\begin{aligned} x_1(n) &= \cos 2\pi \left(\frac{10}{40}\right) n = \cos \frac{\pi}{2} n \\ x_2(n) &= \cos 2\pi \left(\frac{50}{40}\right) n = \cos \frac{5\pi}{2} n \end{aligned} \tag{1.4.13}$$

However, $\cos 5\pi n/2 = \cos(2\pi n + \pi n/2) = \cos \pi n/2$. Hence $x_2(n) = x_1(n)$. Thus the sinusoidal signals are identical and, consequently, indistinguishable. If we are given the sampled values generated by $\cos(\pi/2)n$, there is some ambiguity as to whether these sampled values correspond to $x_1(t)$ or $x_2(t)$. Since $x_2(t)$ yields exactly the same values as $x_1(t)$ when the two are sampled at $F_s = 40$ samples per second, we say that the frequency $F_2 = 50$ Hz is an *alias* of the frequency $F_1 = 10$ Hz at the sampling rate of 40 samples per second.

It is important to note that F_2 is not the only alias of F_1. In fact at the sampling rate of 40 samples per second, the frequency $F_3 = 90$ Hz is also an alias of F_1, as is the frequency $F_4 = 130$ Hz, and so on. All of the sinusoids $\cos 2\pi(F_1 + 40k)t$, $k = 1, 2, 3, 4, \ldots$, sampled at 40 samples per second, yield identical values. Consequently, they are all aliases of $F_1 = 10$ Hz.

In general, the sampling of a continuous-time sinusoidal signal

$$x_a(t) = A\cos(2\pi F_0 t + \theta) \tag{1.4.14}$$

with a sampling rate $F_s = 1/T$ results in a discrete-time signal

$$x(n) = A\cos(2\pi f_0 n + \theta) \tag{1.4.15}$$

where $f_0 = F_0/F_s$ is the relative frequency of the sinusoid. If we assume that $-F_s/2 \le F_0 \le F_s/2$, the frequency f_0 of $x(n)$ is in the range $-\frac{1}{2} \le f_0 \le \frac{1}{2}$, which is the frequency range for discrete-time signals. In this case, the relationship between F_0 and f_0 is one-to-one, and hence it is possible to identify (or reconstruct) the analog signal $x_a(t)$ from the samples $x(n)$.

On the other hand, if the sinusoids

$$x_a(t) = A\cos(2\pi F_k t + \theta) \tag{1.4.16}$$

where

$$F_k = F_0 + kF_s, \qquad k = \pm 1, \pm 2, \ldots \tag{1.4.17}$$

are sampled at a rate F_s, it is clear that the frequency F_k is outside the fundamental frequency range $-F_s/2 \le F \le F_s/2$. Consequently, the sampled signal is

$$\begin{aligned} x(n) \equiv x_a(nT) &= A\cos\left(2\pi\frac{F_0 + kF_s}{F_s}n + \theta\right) \\ &= A\cos(2\pi n F_0/F_s + \theta + 2\pi kn) \\ &= A\cos(2\pi f_0 n + \theta) \end{aligned}$$

which is identical to the discrete-time signal in (1.4.15) obtained by sampling (1.4.14). Thus an infinite number of continuous-time sinusoids is represented by sampling the *same* discrete-time signal (i.e., by the same set of samples). Consequently, if we are given the sequence $x(n)$, an ambiguity exists as to which continuous-time signal $x_a(t)$ these values represent. Equivalently, we can say that the frequencies $F_k = F_0 + kF_s$, $-\infty < k < \infty$ (k integer) are indistinguishable from the frequency F_0 after sampling and hence they are aliases of F_0. The relationship between the frequency variables of the continuous-time and discrete-time signals is illustrated in Fig. 1.4.4.

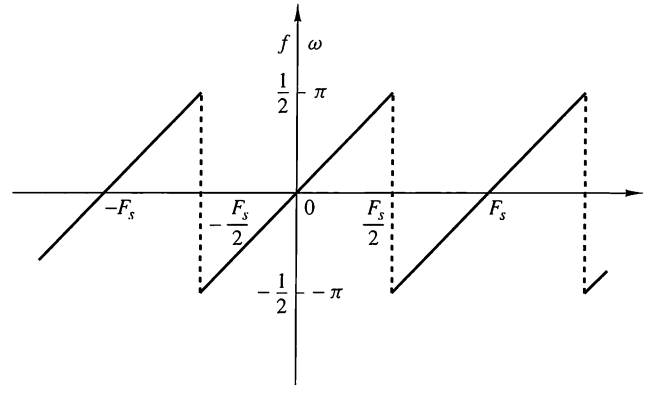


Figure 1.4.4
Relationship between the continuous-time and discrete-time frequency variables in the case of periodic sampling.

An example of aliasing is illustrated in Fig. 1.4.5, where two sinusoids with frequencies $F_0 = \frac{1}{8}$ Hz and $F_1 = -\frac{7}{8}$ Hz yield identical samples when a sampling rate of $F_s = 1$ Hz is used. From (1.4.17) it easily follows that for $k = -1$, $F_0 = F_1 + F_s = (-\frac{7}{8} + 1)$ Hz $= \frac{1}{8}$ Hz.

Since $F_s/2$, which corresponds to $\omega = \pi$, is the highest frequency that can be represented uniquely with a sampling rate F_s, it is a simple matter to determine the mapping of any (alias) frequency above $F_s/2$ ($\omega = \pi$) into the equivalent frequency below $F_s/2$. We can use $F_s/2$ or $\omega = \pi$ as the pivotal point and reflect or "fold" the alias frequency to the range $0 \leq \omega \leq \pi$. Since the point of reflection is $F_s/2$ ($\omega = \pi$), the frequency $F_s/2$ ($\omega = \pi$) is called the *folding frequency*.

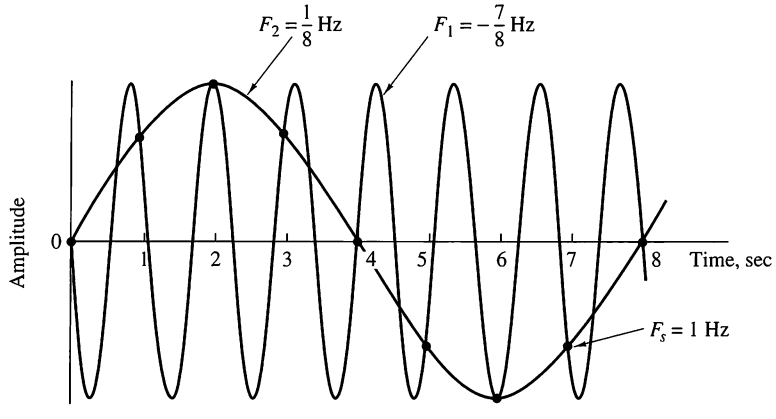


Figure 1.4.5 Illustration of aliasing.

EXAMPLE 1.4.2

Consider the analog signal

$$x_a(t) = 3\cos 100\pi t$$

(a) Determine the minimum sampling rate required to avoid aliasing.

(b) Suppose that the signal is sampled at the rate $F_s = 200$ Hz. What is the discrete-time signal obtained after sampling?

(c) Suppose that the signal is sampled at the rate $F_s = 75$ Hz. What is the discrete-time signal obtained after sampling?

(d) What is the frequency $0 < F < F_s/2$ of a sinusoid that yields samples identical to those obtained in part (c)?

Solution.

(a) The frequency of the analog signal is $F = 50$ Hz. Hence the minimum sampling rate required to avoid aliasing is $F_s = 100$ Hz.

(b) If the signal is sampled at $F_s = 200$ Hz, the discrete-time signal is

$$x(n) = 3\cos\frac{100\pi}{200}n = 3\cos\frac{\pi}{2}n$$

(c) If the signal is sampled at $F_s = 75$ Hz, the discrete-time signal is

$$\begin{aligned} x(n) &= 3\cos\frac{100\pi}{75}n = 3\cos\frac{4\pi}{3}n \\ &= 3\cos\left(2\pi - \frac{2\pi}{3}\right)n \\ &= 3\cos\frac{2\pi}{3}n \end{aligned}$$

(d) For the sampling rate of $F_s = 75$ Hz, we have

$$F = fF_s = 75f$$

The frequency of the sinusoid in part (c) is $f = \frac{1}{3}$. Hence

$$F = 25 \text{ Hz}$$

Clearly, the sinusoidal signal

$$\begin{aligned} y_a(t) &= 3\cos 2\pi Ft \\ &= 3\cos 50\pi t \end{aligned}$$

sampled at $F_s = 75$ samples/s yields identical samples. Hence $F = 50$ Hz is an alias of $F = 25$ Hz for the sampling rate $F_s = 75$ Hz.

1.4.2 The Sampling Theorem

Given any analog signal, how should we select the sampling period T or, equivalently, the sampling rate F_s? To answer this question, we must have some information about the characteristics of the signal to be sampled. In particular, we must have some general information concerning the *frequency content* of the signal. Such information is generally available to us. For example, we know generally that the major frequency components of a speech signal fall below 3000 Hz. On the other hand, television

signals, in general, contain important frequency components up to 5 MHz. The information content of such signals is contained in the amplitudes, frequencies, and phases of the various frequency components, but detailed knowledge of the characteristics of such signals is not available to us prior to obtaining the signals. In fact, the purpose of processing the signals is usually to extract this detailed information. However, if we know the maximum frequency content of the general class of signals (e.g., the class of speech signals, the class of video signals, etc.), we can specify the sampling rate necessary to convert the analog signals to digital signals.

Let us suppose that any analog signal can be represented as a sum of sinusoids of different amplitudes, frequencies, and phases, that is,

$$x_a(t) = \sum_{i=1}^{N} A_i \cos(2\pi F_i t + \theta_i) \tag{1.4.18}$$

where N denotes the number of frequency components. All signals, such as speech and video, lend themselves to such a representation over any short time segment. The amplitudes, frequencies, and phases usually change slowly with time from one time segment to another. However, suppose that the frequencies do not exceed some known frequency, say $F_{\max}$. For example, $F_{\max} = 3000$ Hz for the class of speech signals and $F_{\max} = 5$ MHz for television signals. Since the maximum frequency may vary slightly from different realizations among signals of any given class (e.g., it may vary slightly from speaker to speaker), we may wish to ensure that $F_{\max}$ does not exceed some predetermined value by passing the analog signal through a filter that severely attenuates frequency components above $F_{\max}$. Thus we are certain that no signal in the class contains frequency components (having significant amplitude or power) above $F_{\max}$. In practice, such filtering is commonly used prior to sampling.

From our knowledge of $F_{\max}$, we can select the appropriate sampling rate. We know that the highest frequency in an analog signal that can be unambiguously reconstructed when the signal is sampled at a rate $F_s = 1/T$ is $F_s/2$. Any frequency above $F_s/2$ or below $-F_s/2$ results in samples that are identical with a corresponding frequency in the range $-F_s/2 \leq F \leq F_s/2$. To avoid the ambiguities resulting from aliasing, we must select the sampling rate to be sufficiently high. That is, we must select $F_s/2$ to be greater than $F_{\max}$. Thus to avoid the problem of aliasing, F_s is selected so that

$$F_s > 2F_{\max} \tag{1.4.19}$$

where $F_{\max}$ is the largest frequency component in the analog signal. With the sampling rate selected in this manner, any frequency component, say $|F_i| < F_{\max}$, in the analog signal is mapped into a discrete-time sinusoid with a frequency

$$-\frac{1}{2} \leq f_i = \frac{F_i}{F_s} \leq \frac{1}{2} \tag{1.4.20}$$

or, equivalently,

$$-\pi \leq \omega_i = 2\pi f_i \leq \pi \tag{1.4.21}$$

Since, $|f| = \frac{1}{2}$ or $|\omega| = \pi$ is the highest (unique) frequency in a discrete-time signal, the choice of sampling rate according to (1.4.19) avoids the problem of aliasing.

In other words, the condition $F_s > 2F_{\max}$ ensures that all the sinusoidal components in the analog signal are mapped into corresponding discrete-time frequency components with frequencies in the fundamental interval. Thus all the frequency components of the analog signal are represented in sampled form without ambiguity, and hence the analog signal can be reconstructed without distortion from the sample values using an "appropriate" interpolation (digital-to-analog conversion) method. The "appropriate" or ideal interpolation formula is specified by the *sampling theorem.*

Sampling Theorem. If the highest frequency contained in an analog signal $x_a(t)$ is $F_{\max} = B$ and the signal is sampled at a rate $F_s > 2F_{\max} \equiv 2B$, then $x_a(t)$ can be exactly recovered from its sample values using the interpolation function

$$g(t) = \frac{\sin 2\pi Bt}{2\pi Bt} \tag{1.4.22}$$

Thus $x_a(t)$ may be expressed as

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a\left(\frac{n}{F_s}\right) g\left(t - \frac{n}{F_s}\right) \tag{1.4.23}$$

where $x_a(n/F_s) = x_a(nT) \equiv x(n)$ are the samples of $x_a(t)$.

When the sampling of $x_a(t)$ is performed at the minimum sampling rate $F_s = 2B$, the reconstruction formula in (1.4.23) becomes

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a\left(\frac{n}{2B}\right) \frac{\sin 2\pi B(t - n/2B)}{2\pi B(t - n/2B)} \tag{1.4.24}$$

The sampling rate $F_N = 2B = 2F_{\max}$ is called the *Nyquist rate.* Figure 1.4.6 illustrates the ideal D/A conversion process using the interpolation function in (1.4.22).

As can be observed from either (1.4.23) or (1.4.24), the reconstruction of $x_a(t)$ from the sequence $x(n)$ is a complicated process, involving a weighted sum of the interpolation function $g(t)$ and its time-shifted versions $g(t - nT)$ for $-\infty < n < \infty$, where the weighting factors are the samples $x(n)$. Because of the complexity and the infinite number of samples required in (1.4.23) or (1.4.24), these reconstruction formulas are primarily of theoretical interest. Practical interpolation methods are given in Chapter 6.

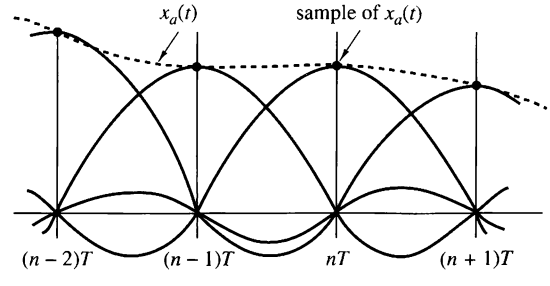


Figure 1.4.6
Ideal D/A conversion (interpolation).

EXAMPLE 1.4.3

Consider the analog signal

$$x_a(t) = 3\cos 50\pi t + 10\sin 300\pi t - \cos 100\pi t$$

What is the Nyquist rate for this signal?

Solution. The frequencies present in the signal above are

$$F_1 = 25 \text{ Hz}, \qquad F_2 = 150 \text{ Hz}, \qquad F_3 = 50 \text{ Hz}$$

Thus $F_{\max} = 150$ Hz and according to (1.4.19),

$$F_s > 2F_{\max} = 300 \text{ Hz}$$

The Nyquist rate is $F_N = 2F_{\max}$. Hence

$$F_N = 300 \text{ Hz}$$

Discussion. It should be observed that the signal component $10\sin 300\pi t$, sampled at the Nyquist rate $F_N = 300$, results in the samples $10\sin \pi n$, which are identically zero. In other words, we are sampling the analog sinusoid at its zero-crossing points, and hence we miss this signal component completely. This situation does not occur if the sinusoid is offset in phase by some amount θ. In such a case we have $10\sin(300\pi t + \theta)$ sampled at the Nyquist rate $F_N = 300$ samples per second, which yields the samples

$$\begin{aligned} 10\sin(\pi n + \theta) &= 10(\sin \pi n \cos\theta + \cos \pi n \sin\theta) \\ &= 10\sin\theta \cos \pi n \\ &= (-1)^n 10\sin\theta \end{aligned}$$

Thus if $\theta \neq 0$ or π, the samples of the sinusoid taken at the Nyquist rate are not all zero. However, we still cannot obtain the correct amplitude from the samples when the phase θ is unknown. A simple remedy that avoids this potentially troublesome situation is to sample the analog signal at a rate higher than the Nyquist rate.

EXAMPLE 1.4.4

Consider the analog signal

$$x_a(t) = 3\cos 2000\pi t + 5\sin 6000\pi t + 10\cos 12{,}000\pi t$$

(a) What is the Nyquist rate for this signal?

(b) Assume now that we sample this signal using a sampling rate $F_s = 5000$ samples/s. What is the discrete-time signal obtained after sampling?

(c) What is the analog signal $y_a(t)$ that we can reconstruct from the samples if we use ideal interpolation?

Solution.

(a) The frequencies existing in the analog signal are

$$F_1 = 1 \text{ kHz}, \qquad F_2 = 3 \text{ kHz}, \qquad F_3 = 6 \text{ kHz}$$

Thus $F_{\max} = 6$ kHz, and according to the sampling theorem,

$$F_s > 2F_{\max} = 12 \text{ kHz}$$

The Nyquist rate is

$$F_N = 12 \text{ kHz}$$

(b) Since we have chosen $F_s = 5$ kHz, the folding frequency is

$$\frac{F_s}{2} = 2.5 \text{ kHz}$$

and this is the maximum frequency that can be represented uniquely by the sampled signal. By making use of (1.4.2) we obtain

$$\begin{aligned} x(n) = x_a(nT) &= x_a\left(\frac{n}{F_s}\right) \\ &= 3\cos 2\pi\left(\tfrac{1}{5}\right)n + 5\sin 2\pi\left(\tfrac{3}{5}\right)n + 10\cos 2\pi\left(\tfrac{6}{5}\right)n \\ &= 3\cos 2\pi\left(\tfrac{1}{5}\right)n + 5\sin 2\pi\left(1 - \tfrac{2}{5}\right)n + 10\cos 2\pi\left(1 + \tfrac{1}{5}\right)n \\ &= 3\cos 2\pi\left(\tfrac{1}{5}\right)n + 5\sin 2\pi\left(-\tfrac{2}{5}\right)n + 10\cos 2\pi\left(\tfrac{1}{5}\right)n \end{aligned}$$

Finally, we obtain

$$x(n) = 13\cos 2\pi\left(\tfrac{1}{5}\right)n - 5\sin 2\pi\left(\tfrac{2}{5}\right)n$$

The same result can be obtained using Fig. 1.4.4. Indeed, since $F_s = 5$ kHz, the folding frequency is $F_s/2 = 2.5$ kHz. This is the maximum frequency that can be represented uniquely by the sampled signal. From (1.4.17) we have $F_0 = F_k - kF_s$. Thus F_0 can be obtained by subtracting from F_k an integer multiple of F_s such that $-F_s/2 \le F_0 \le F_s/2$. The frequency F_1 is less than $F_s/2$ and thus it is not affected by aliasing. However, the other two frequencies are above the folding frequency and they will be changed by the aliasing effect. Indeed,

$$F_2' = F_2 - F_s = -2 \text{ kHz}$$

$$F_3' = F_3 - F_s = 1 \text{ kHz}$$

From (1.4.5) it follows that $f_1 = \frac{1}{5}$, $f_2 = -\frac{2}{5}$, and $f_3 = \frac{1}{5}$, which are in agreement with the result above.

(c) Since the frequency components at only 1 kHz and 2 kHz are present in the sampled signal, the analog signal we can recover is

$$ya(t) = 13\cos 2000\pi t - 5\sin 4000\pi t$$

which is obviously different from the original signal $x_a(t)$. This distortion of the original analog signal was caused by the aliasing effect, due to the low sampling rate used.

Although aliasing is a pitfall to be avoided, there are two useful practical applications based on the exploitation of the aliasing effect. These applications are the stroboscope and the sampling oscilloscope. Both instruments are designed to operate as aliasing devices in order to represent high frequencies as low frequencies.

To elaborate, consider a signal with high-frequency components confined to a given frequency band $B_1 < F < B_2$, where $B_2 - B_1 \equiv B$ is defined as the bandwidth of the signal. We assume that $B << B_1 < B_2$. This condition means that the frequency components in the signal are much larger than the bandwidth B of the signal. Such signals are usually called bandpass or narrowband signals. Now, if this signal is sampled at a rate $F_s \geq 2B$, but $F_s << B_1$, then all the frequency components contained in the signal will be aliases of frequencies in the range $0 < F < F_s/2$. Consequently, if we observe the frequency content of the signal in the fundamental range $0 < F < F_s/2$, we know precisely the frequency content of the analog signal since we know the frequency band $B_1 < F < B_2$ under consideration. Consequently, if the signal is a narrowband (bandpass) signal, we can reconstruct the original signal from the samples, provided that the signal is sampled at a rate $F_s > 2B$, where B is the bandwidth. This statement constitutes another form of the sampling theorem, which we call the *bandpass form* in order to distinguish it from the previous form of the sampling theorem, which applies in general to all types of signals. The latter is sometimes called the *baseband form*. The *bandpass form* of the sampling theorem is described in detail in Section 6.4.

1.4.3 Quantization of Continuous-Amplitude Signals

As we have seen, a digital signal is a sequence of numbers (samples) in which each number is represented by a finite number of digits (finite precision).

The process of converting a discrete-time continuous-amplitude signal into a digital signal by expressing each sample value as a finite (instead of an infinite) number of digits is called *quantization*. The error introduced in representing the continuous-valued signal by a finite set of discrete value levels is called *quantization error* or *quantization noise*.

We denote the quantizer operation on the samples $x(n)$ as $Q[x(n)]$ and let $x_q(n)$ denote the sequence of quantized samples at the output of the quantizer. Hence

$$x_q(n) = Q[x(n)]$$

Then the quantization error is a sequence $e_q(n)$ defined as the difference between the quantized value and the actual sample value. Thus

$$e_q(n) = x_q(n) - x(n) \tag{1.4.25}$$

We illustrate the quantization process with an example. Let us consider the discrete-time signal

$$x(n) = \begin{cases} 0.9^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

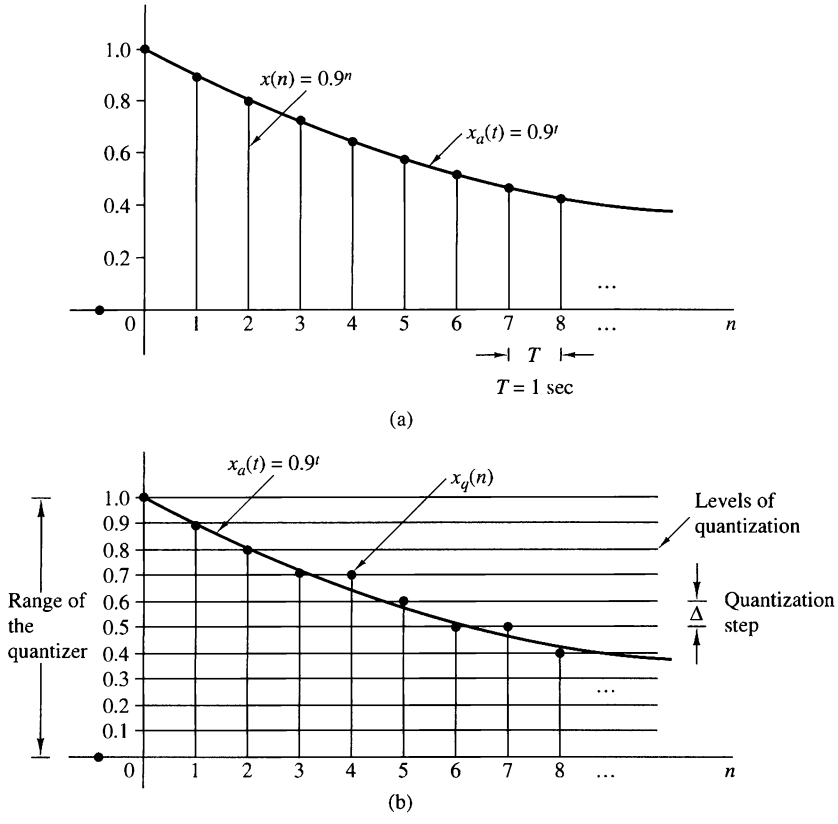


Figure 1.4.7 Illustration of quantization.

obtained by sampling the analog exponential signal $x_a(t) = 0.9^t$, $t \geq 0$ with a sampling frequency $F_s = 1$ Hz (see Fig. 1.4.7(a)). Observation of Table 1.2, which shows the values of the first 10 samples of $x(n)$, reveals that the description of the sample value $x(n)$ requires n significant digits. It is obvious that this signal cannot be processed by using a calculator or a digital computer since only the first few samples can be stored and manipulated. For example, most calculators process numbers with only eight significant digits.

However, let us assume that we want to use only one significant digit. To eliminate the excess digits, we can either simply discard them (*truncation*) or discard them by rounding the resulting number (*rounding*). The resulting quantized signals $x_q(n)$ are shown in Table 1.2. We discuss only quantization by rounding, although it is just as easy to treat truncation. The rounding process is graphically illustrated in Fig. 1.4.7(b). The values allowed in the digital signal are called the *quantization levels*, whereas the distance Δ between two successive quantization levels is called the *quantization step size* or *resolution*. The rounding quantizer assigns each sample of $x(n)$ to the nearest quantization level. In contrast, a quantizer that performs truncation would have assigned each sample of $x(n)$ to the quantization level below

TABLE 1.2 Numerical Illustration of Quantization with One Significant Digit Using Truncation or Rounding

n	$x(n)$ Discrete-time signal	$x_q(n)$ (Truncation)	$x_q(n)$ (Rounding)	$e_q(n) = x_q(n) - x(n)$ (Rounding)
0	1	1.0	1.0	0.0
1	0.9	0.9	0.9	0.0
2	0.81	0.8	0.8	−0.01
3	0.729	0.7	0.7	−0.029
4	0.6561	0.6	0.7	0.0439
5	0.59049	0.5	0.6	0.00951
6	0.531441	0.5	0.5	−0.031441
7	0.4782969	0.4	0.5	0.0217031
8	0.43046721	0.4	0.4	−0.03046721
9	0.387420489	0.3	0.4	0.012579511

it. The quantization error $e_q(n)$ in rounding is limited to the range of $-\Delta/2$ to $\Delta/2$, that is,

$$-\frac{\Delta}{2} \leq e_q(n) \leq \frac{\Delta}{2} \tag{1.4.26}$$

In other words, the instantaneous quantization error cannot exceed half of the quantization step (see Table 1.2).

If $x_{\min}$ and $x_{\max}$ represent the minimum and maximum values of $x(n)$ and L is the number of quantization levels, then

$$\Delta = \frac{x_{\max} - x_{\min}}{L - 1} \tag{1.4.27}$$

We define the *dynamic range* of the signal as $x_{\max} - x_{\min}$. In our example we have $x_{\max} = 1$, $x_{\min} = 0$, and $L = 11$, which leads to $\Delta = 0.1$. Note that if the dynamic range is fixed, increasing the number of quantization levels L results in a decrease of the quantization step size. Thus the quantization error decreases and the accuracy of the quantizer increases. In practice we can reduce the quantization error to an insignificant amount by choosing a sufficient number of quantization levels.

Theoretically, quantization of analog signals always results in a loss of information. This is a result of the ambiguity introduced by quantization. Indeed, quantization is an irreversible or noninvertible process (i.e., a many-to-one mapping) since all samples in a distance $\Delta/2$ about a certain quantization level are assigned the same value. This ambiguity makes the exact quantitative analysis of quantization extremely difficult. This subject is discussed further in Chapter 6, where we use statistical analysis.

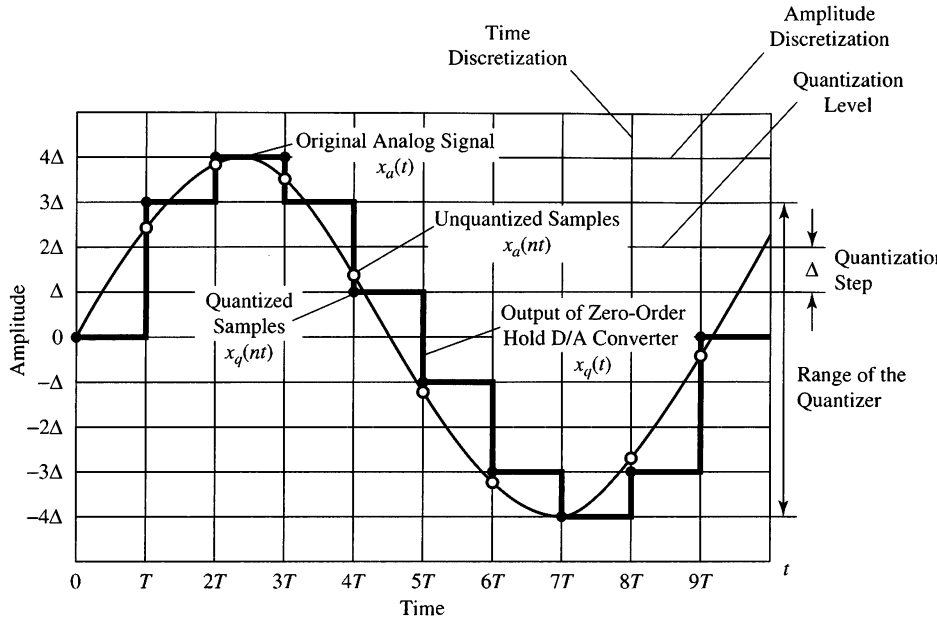


Figure 1.4.8 Sampling and quantization of a sinusoidal signal.

1.4.4 Quantization of Sinusoidal Signals

Figure 1.4.8 illustrates the sampling and quantization of an analog sinusoidal signal $x_a(t) = A\cos\Omega_0 t$ using a rectangular grid. Horizontal lines within the range of the quantizer indicate the allowed levels of quantization. Vertical lines indicate the sampling times. Thus, from the original analog signal $x_a(t)$ we obtain a discrete-time signal $x(n) = x_a(nT)$ by sampling and a discrete-time, discrete-amplitude signal $x_q(nT)$ after quantization. In practice, the staircase signal $x_q(t)$ can be obtained by using a zero-order hold. This analysis is useful because sinusoids are used as test signals in A/D converters.

If the sampling rate F_s satisfies the sampling theorem, quantization is the only error in the A/D conversion process.

Thus we can evaluate the quantization error by quantizing the analog signal $x_a(t)$ instead of the discrete-time signal $x(n) = x_a(nT)$. Inspection of Fig. 1.4.8 indicates that the signal $x_a(t)$ is almost linear between quantization levels (see Fig. 1.4.9). The

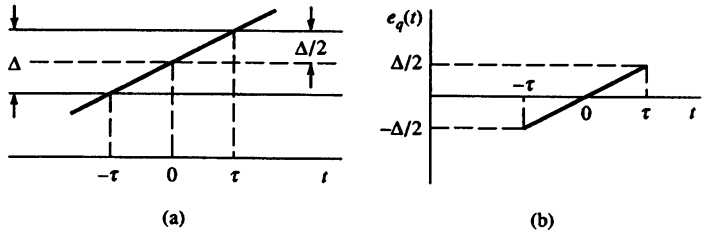


Figure 1.4.9 The quantization error $e_q(t) = x_a(t) - x_q(t)$.

corresponding quantization error $e_q(t) = x_a(t) - x_q(t)$ is shown in Fig. 1.4.9. In Fig. 1.4.9, τ denotes the time that $x_a(t)$ stays within the quantization levels. The mean-square error power P_q is

$$P_q = \frac{1}{2\tau}\int_{-\tau}^{\tau} e_q^2(t)\,dt = \frac{1}{\tau}\int_0^{\tau} e_q^2(t)\,dt \tag{1.4.28}$$

Since $e_q(t) = (\Delta/2\tau)t$, $-\tau \le t \le \tau$, we have

$$P_q = \frac{1}{\tau}\int_0^{\tau}\left(\frac{\Delta}{2\tau}\right)^2 t^2\,dt = \frac{\Delta^2}{12} \tag{1.4.29}$$

If the quantizer has b bits of accuracy and the quantizer covers the entire range $2A$, the quantization step is $\Delta = 2A/2^b$. Hence

$$P_q = \frac{A^2/3}{2^{2b}} \tag{1.4.30}$$

The average power of the signal $x_a(t)$ is

$$P_x = \frac{1}{T_p}\int_0^{T_p} (A\cos\Omega_0 t)^2\,dt = \frac{A^2}{2} \tag{1.4.31}$$

The quality of the output of the A/D converter is usually measured by the *signal-to-quantization noise ratio (SQNR)*, which provides the ratio of the signal power to the noise power:

$$\text{SQNR} = \frac{P_x}{P_q} = \frac{3}{2}\cdot 2^{2b}$$

Expressed in decibels (dB), the SQNR is

$$\text{SQNR(dB)} = 10\log_{10}\text{SQNR} = 1.76 + 6.02b \tag{1.4.32}$$

This implies that the SQNR increases approximately 6 dB for every bit added to the word length, that is, for each doubling of the quantization levels.

Although formula (1.4.32) was derived for sinusoidal signals, we shall see in Chapter 6 that a similar result holds for every signal whose dynamic range spans the range of the quantizer. This relationship is extremely important because it dictates the number of bits required by a specific application to assure a given signal-to-noise ratio. For example, most compact disc players use a sampling frequency of 44.1 kHz and 16-bit sample resolution, which implies a SQNR of more than 96 dB.

1.4.5 Coding of Quantized Samples

The coding process in an A/D converter assigns a unique binary number to each quantization level. If we have L levels we need at least L different binary numbers. With a word length of b bits we can create 2^b different binary numbers. Hence we have $2^b \ge L$, or equivalently, $b \ge \log_2 L$. Thus the number of bits required in the coder is the smallest integer greater than or equal to $\log_2 L$. In our example (Table 1.2) it can easily be seen that we need a coder with $b = 4$ bits. Commercially available A/D converters may be obtained with finite precision of $b = 16$ or less. Generally, the higher the sampling speed and the finer the quantization, the more expensive the device becomes.

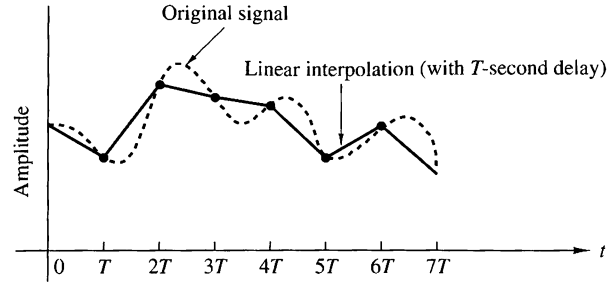


Figure 1.4.10
Linear point connector (with T-second delay).

1.4.6 Digital-to-Analog Conversion

To convert a digital signal into an analog signal we can use a digital-to-analog (D/A) converter. As stated previously, the task of a D/A converter is to interpolate between samples.

The sampling theorem specifies the optimum interpolation for a bandlimited signal. However, this type of interpolation is too complicated and, hence, impractical, as indicated previously. From a practical viewpoint, the simplest D/A converter is the zero-order hold shown in Fig. 1.4.2, which simply holds constant the value of one sample until the next one is received. Additional improvement can be obtained by using linear

interpolation as shown in Fig. 1.4.10 to connect successive samples with straight-line segments. Better interpolation can be achieved by using more sophisticated higher-order interpolation techniques.

In general, suboptimum interpolation techniques result in passing frequencies above the folding frequency. Such frequency components are undesirable and are usually removed by passing the output of the interpolator through a proper analog filter, which is called a *postfilter* or *smoothing filter.*

Thus D/A conversion usually involves a suboptimum interpolator followed by a postfilter. D/A converters are treated in more detail in Chapter 6.

1.4.7 Analysis of Digital Signals and Systems Versus Discrete-Time Signals and Systems

We have seen that a digital signal is defined as a function of an integer independent variable and its values are taken from a finite set of possible values. The usefulness of such signals is a consequence of the possibilities offered by digital computers. Computers operate on numbers, which are represented by a stringof 0's and 1's. The length of this string (*word length*) is fixed and finite and usually is 8, 12, 16, or 32 bits. The effects of finite word length in computations cause complications in the analysis of digital signal processing systems. To avoid these complications, we neglect the quantized nature of digital signals and systems in much of our analysis and consider them as discrete-time signals and systems.

In Chapters 6, 9, and 10 we investigate the consequences of using a finite word length. This is an important topic, since many digital signal processing problems are solved with small computers or microprocessors that employ fixed-point arithmetic.

Consequently, one must look carefully at the problem of finite-precision arithmetic and account for it in the design of software and hardware that performs the desired signal processing tasks.

1.5 Summary and References

In this introductory chapter we have attempted to provide the motivation for digital signal processing as an alternative to analog signal processing. We presented the basic elements of a digital signal processing system and defined the operations needed to convert an analog signal into a digital signal ready for processing. Of particular importance is the sampling theorem, which was introduced by Nyquist (1928) and later popularized in the classic paper by Shannon (1949). The sampling theorem as described in Section 1.4.2 is derived in Chapter 6. Sinusoidal signals were introduced primarily for the purpose of illustrating the aliasing phenomenon and for the subsequent development of the sampling theorem.

Quantization effects that are inherent in the A/D conversion of a signal were also introduced in this chapter. Signal quantization is best treated in statistical terms, as described in Chapters 6, 9, and 10.

Finally, the topic of signal reconstruction, or D/A conversion, was described briefly. Signal reconstruction based on staircase interpolation is treated in Section 6.3.

There are numerous practical applications of digital signal processing. The book edited by Oppenheim (1978) treats applications to speech processing, image processing, radar signal processing, sonar signal processing, and geophysical signal processing.

Problems

1.1 Classify the following signals according to whether they are (1) one- or multi-dimensional; (2) single or multichannel, (3) continuous time or discrete time, and (4) analog or digital (in amplitude). Give a brief explanation.

(a) Closing prices of utility stocks on the New York Stock Exchange.

(b) A color movie.

(c) Position of the steering wheel of a car in motion relative to car's reference frame.

(d) Position of the steering wheel of a car in motion relative to ground reference frame.

(e) Weight and height measurements of a child taken every month.

1.2 Determine which of the following sinusoids are periodic and compute their fundamental period.

(a) $\cos 0.01\pi n$ **(b)** $\cos\left(\pi \frac{30n}{105}\right)$ **(c)** $\cos 3\pi n$ **(d)** $\sin 3n$ **(e)** $\sin\left(\pi \frac{62n}{10}\right)$

1.3 Determine whether or not each of the following signals is periodic. In case a signal is periodic, specify its fundamental period.

(a) $x_a(t) = 3\cos(5t + \pi/6)$

(b) $x(n) = 3\cos(5n + \pi/6)$

(c) $x(n) = 2\exp[j(n/6 - \pi)]$

(d) $x(n) = \cos(n/8)\cos(\pi n/8)$

(e) $x(n) = \cos(\pi n/2) - \sin(\pi n/8) + 3\cos(\pi n/4 + \pi/3)$

1.4 **(a)** Show that the fundamental period N_p of the signals

$$s_k(n) = e^{j2\pi kn/N}, \qquad k = 0, 1, 2, \ldots$$

is given by $N_p = N/\text{GCD}(k, N)$, where GCD is the greatest common divisor of k and N.

(b) What is the fundamental period of this set for $N = 7$?

(c) What is it for $N = 16$?

1.5 Consider the following analog sinusoidal signal:

$$x_a(t) = 3\sin(100\pi t)$$

(a) Sketch the signal $x_a(t)$ for $0 \leq t \leq 30$ ms.

(b) The signal $x_a(t)$ is sampled with a sampling rate $F_s = 300$ samples/s. Determine the frequency of the discrete-time signal $x(n) = x_a(nT)$, $T = 1/F_s$, and show that it is periodic.

(c) Compute the sample values in one period of $x(n)$. Sketch $x(n)$ on the same diagram with $x_a(t)$. What is the period of the discrete-time signal in milliseconds?

(d) Can you find a sampling rate F_s such that the signal $x(n)$ reaches its peak value of 3? What is the minimum F_s suitable for this task?

1.6 A continuous-time sinusoid $x_a(t)$ with fundamental period $T_p = 1/F_0$ is sampled at a rate $F_s = 1/T$ to produce a discrete-time sinusoid $x(n) = x_a(nT)$.

(a) Show that $x(n)$ is periodic if $T/T_p = k/N$ (i.e., T/T_p is a rational number).

(b) If $x(n)$ is periodic, what is its fundamental period T_p in seconds?

(c) Explain the statement: $x(n)$ is periodic if its fundamental period T_p, in seconds, is equal to an integer number of periods of $x_a(t)$.

1.7 An analog signal contains frequencies up to 10 kHz.

(a) What range of sampling frequencies allows exact reconstruction of this signal from its samples?

(b) Suppose that we sample this signal with a sampling frequency $F_s = 8$ kHz. Examine what happens to the frequency $F_1 = 5$ kHz.

(c) Repeat part (b) for a frequency $F_2 = 9$ kHz.

1.8 An analog electrocardiogram (ECG) signal contains useful frequencies up to 100 Hz.

(a) What is the Nyquist rate for this signal?

(b) Suppose that we sample this signal at a rate of 250 samples/s. What is the highest frequency that can be represented uniquely at this sampling rate?

1.9 An analog signal $x_a(t) = \sin(480\pi t) + 3\sin(720\pi t)$ is sampled 600 times per second.

(a) Determine the Nyquist sampling rate for $x_a(t)$.

(b) Determine the folding frequency.

(c) What are the frequencies, in radians, in the resulting discrete time signal $x(n)$?

(d) If $x(n)$ is passed through an ideal D/A converter, what is the reconstructed signal $y_a(t)$?

1.10 A digital communication link carries binary-coded words representing samples of an input signal

$$x_a(t) = 3\cos 600\pi t + 2\cos 1800\pi t$$

The link is operated at 10,000 bits/s and each input sample is quantized into 1024 different voltage levels.

(a) What are the sampling frequency and the folding frequency?

(b) What is the Nyquist rate for the signal $x_a(t)$?

(c) What are the frequencies in the resulting discrete-time signal $x(n)$?

(d) What is the resolution Δ?

1.11 Consider the simple signal processing system shown in Fig. P1.11. The sampling periods of the A/D and D/A converters are $T = 5$ ms and $T' = 1$ ms, respectively. Determine the output $y_a(t)$ of the system, if the input is

$$x_a(t) = 3\cos 100\pi t + 2\sin 250\pi t \qquad (t \text{ in seconds})$$

The postfilter removes any frequency component above $F_s/2$.

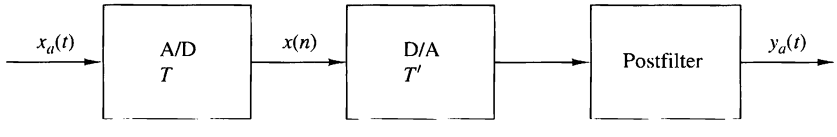


Figure P1.11

1.12 **(a)** Derive the expression for the discrete-time signal $x(n)$ in Example 1.4.2 using the periodicity properties of sinusoidal functions.

(b) What is the analog signal we can obtain from $x(n)$ if in the reconstruction process we assume that $F_s = 10$ kHz?

1.13 The discrete-time signal $x(n) = 6.35\cos(\pi/10)n$ is quantized with a resolution (a) $\Delta = 0.1$ or (b) $\Delta = 0.02$. How many bits are required in the A/D converter in each case?

1.14 Determine the bit rate and the resolution in the sampling of a seismic signal with dynamic range of 1 volt if the sampling rate is $F_s = 20$ samples/s and we use an 8-bit A/D converter. What is the maximum frequency that can be present in the resulting digital seismic signal?

1.15 *Sampling of sinusoidal signals: aliasing* Consider the following continuous-time sinusoidal signal

$$x_a(t) = \sin 2\pi F_0 t, \qquad -\infty < t < \infty$$

Since $x_a(t)$ is described mathematically, its sampled version can be described by values every T seconds. The sampled signal is described by the formula

$$x(n) = x_a(nT) = \sin 2\pi \frac{F_0}{F_s} n, \qquad -\infty < n < \infty$$

where $F_s = 1/T$ is the sampling frequency.

(a) Plot the signal $x(n)$, $0 \le n \le 99$ for $F_s = 5$ kHz and $F_0 = 0.5, 2, 3$, and 4.5 kHz. Explain the similarities and differences among the various plots.

(b) Suppose that $F_0 = 2$ kHz and $F_s = 50$ kHz.

1. Plot the signal $x(n)$. What is the frequency f_0 of the signal $x(n)$?
2. Plot the signal $y(n)$ created by taking the even-numbered samples of $x(n)$. Is this a sinusoidal signal? Why? If so, what is its frequency?

1.16 *Quantization error in A/D conversion of a sinusoidal signal* Let $x_q(n)$ be the signal obtained by quantizing the signal $x(n) = \sin 2\pi f_0 n$. The quantization error power P_q is defined by

$$P_q = \frac{1}{N} \sum_{n=0}^{N-1} e^2(n) = \frac{1}{N} \sum_{n=0}^{N-1} [x_q(n) - x(n)]^2$$

The "quality" of the quantized signal can be measured by the signal-to-quantization noise ratio (SQNR) defined by

$$\text{SQNR} = 10 \log_{10} \frac{P_x}{P_q}$$

where P_x is the power of the unquantized signal $x(n)$.

(a) For $f_0 = 1/50$ and $N = 200$, write a program to quantize the signal $x(n)$, using truncation, to 64, 128, and 256 quantization levels. In each case plot the signals $x(n)$, $x_q(n)$, and $e(n)$ and compute the corresponding SQNR.

(b) Repeat part (a) by using rounding instead of truncation.

(c) Comment on the results obtained in parts (a) and (b).

(d) Compare the experimentally measured SQNR with the theoretical SQNR predicted by formula (1.4.32) and comment on the differences and similarities.

CHAPTER

2

Discrete-Time Signals and Systems

In Chapter 1 we introduced the reader to a number of important types of signals and described the sampling process by which an analog signal is converted to a discrete-time signal. In addition, we presented in some detail the characteristics of discrete-time sinusoidal signals. The sinusoid is an important elementary signal that serves as a basic building block in more complex signals. However, there are other elementary signals that are important in our treatment of signal processing. These discrete-time signals are introduced in this chapter and are used as basis functions or building blocks to describe more complex signals.

The major emphasis in this chapter is the characterization of discrete-time systems in general and the class of linear time-invariant (LTI) systems in particular. A number of important time-domain properties of LTI systems are defined and developed, and an important formula, called the convolution formula, is derived which allows us to determine the output of an LTI system to any given arbitrary input signal. In addition to the convolution formula, difference equations are introduced as an alternative method for describing the input–output relationship of an LTI system, and in addition, recursive and nonrecursive realizations of LTI systems are treated.

Our motivation for the emphasis on the study of LTI systems is twofold. First, there is a large collection of mathematical techniques that can be applied to the analysis of LTI systems. Second, many practical systems are either LTI systems or can be approximated by LTI systems. Because of its importance in digital signal processing applications and its close resemblance to the convolution formula, we also introduce the correlation between two signals. The autocorrelation and crosscorrelation of signals are defined and their properties are presented.

2.1 Discrete-Time Signals

As we discussed in Chapter 1, a discrete-time signal $x(n)$ is a function of an independent variable that is an integer. It is graphically represented as in Fig. 2.1.1. It is important to note that a discrete-time signal is *not defined* at instants between two successive samples. Also, it is incorrect to think that $x(n)$ is equal to zero if n is not an integer. Simply, the signal $x(n)$ is not defined for noninteger values of n.

In the sequel we will assume that a discrete-time signal is defined for every integer value n for $-\infty < n < \infty$. By tradition, we refer to $x(n)$ as the "nth sample" of the signal even if the signal $x(n)$ is inherently discrete time (i.e., not obtained by sampling an analog signal). If, indeed, $x(n)$ was obtained from sampling an analog signal $x_a(t)$, then $x(n) \equiv x_a(nT)$, where T is the sampling period (i.e., the time between successive samples).

Besides the graphical representation of a discrete-time signal or sequence as illustrated in Fig. 2.1.1, there are some alternative representations that are often more convenient to use. These are:

1. Functional representation, such as

$$x(n) = \begin{cases} 1, & \text{for } n = 1, 3 \\ 4, & \text{for } n = 2 \\ 0, & \text{elsewhere} \end{cases} \tag{2.1.1}$$

2. Tabular representation, such as

n	$\cdots$	-2	-1	0	1	2	3	4	5	$\cdots$
$x(n)$	$\cdots$	0	0	0	1	4	1	0	0	$\cdots$

3. Sequence representation

An infinite-duration signal or sequence with the time origin ($n = 0$) indicated by the symbol $\uparrow$ is represented as

$$x(n) = \{\ldots 0, \underset{\uparrow}{0}, 1, 4, 1, 0, 0, \ldots\} \tag{2.1.2}$$

A sequence $x(n)$, which is zero for $n < 0$, can be represented as

$$x(n) = \{\underset{\uparrow}{0}, 1, 4, 1, 0, 0, \ldots\} \tag{2.1.3}$$

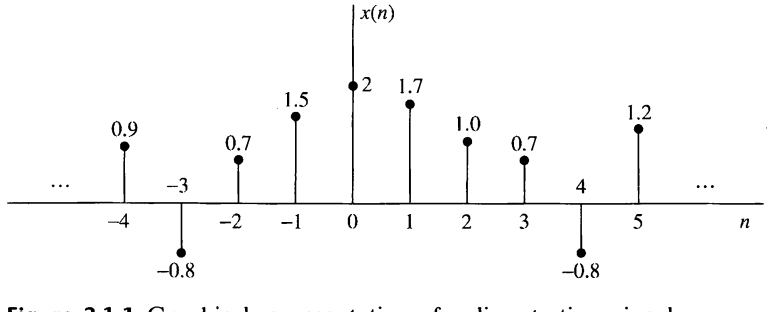


Figure 2.1.1 Graphical representation of a discrete-time signal.

The time origin for a sequence $x(n)$, which is zero for $n < 0$, is understood to be the first (leftmost) point in the sequence.

A finite-duration sequence can be represented as

$$x(n) = \{3, -1, \underset{\uparrow}{-2}, 5, 0, 4, -1\} \tag{2.1.4}$$

whereas a finite-duration sequence that satisfies the condition $x(n) = 0$ for $n < 0$ can be represented as

$$x(n) = \{\underset{\uparrow}{0}, 1, 4, 1\} \tag{2.1.5}$$

The signal in (2.1.4) consists of seven samples or points (in time), so it is called or identified as a seven-point sequence. Similarly, the sequence given by (2.1.5) is a four-point sequence.

2.1.1 Some Elementary Discrete-Time Signals

In our study of discrete-time signals and systems there are a number of basic signals that appear often and play an important role. These signals are defined below.

1. The *unit sample sequence* is denoted as $\delta(n)$ and is defined as

$$\delta(n) \equiv \begin{cases} 1, & \text{for } n = 0 \\ 0, & \text{for } n \neq 0 \end{cases} \tag{2.1.6}$$

   In words, the unit sample sequence is a signal that is zero everywhere, except at $n = 0$ where its value is unity. This signal is sometimes referred to as a *unit impulse*. In contrast to the analog signal $\delta(t)$, which is also called a unit impulse and is defined to be zero everywhere except at $t = 0$, and has unit area, the unit sample sequence is much less mathematically complicated. The graphical representation of $\delta(n)$ is shown in Fig. 2.1.2.

2. The *unit step signal* is denoted as $u(n)$ and is defined as

$$u(n) \equiv \begin{cases} 1, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases} \tag{2.1.7}$$

   Figure 2.1.3 illustrates the unit step signal.

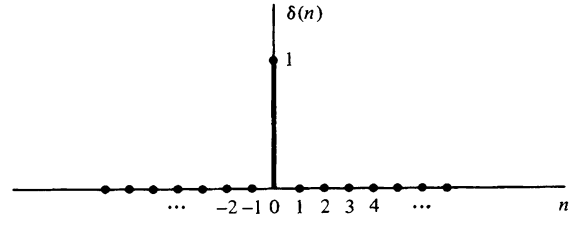


Figure 2.1.2
Graphical representation of the unit sample signal.

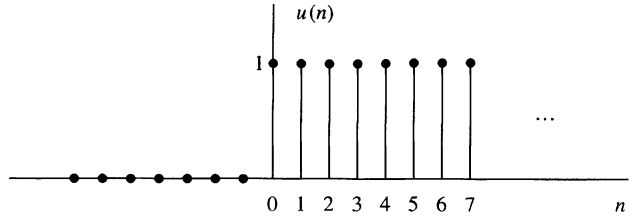


Figure 2.1.3
Graphical representation of the unit step signal.

3. The *unit ramp signal* is denoted as $u_r(n)$ and is defined as

$$u_r(n) \equiv \begin{cases} n, & \text{for } n \geq 0 \\ 0, & \text{for } n < 0 \end{cases} \tag{2.1.8}$$

This signal is illustrated in Fig. 2.1.4.

4. The *exponential signal* is a sequence of the form

$$x(n) = a^n \qquad \text{for all } n \tag{2.1.9}$$

If the parameter a is real, then $x(n)$ is a real signal. Figure 2.1.5 illustrates $x(n)$ for various values of the parameter a.

When the parameter a is complex valued, it can be expressed as

$$a \equiv re^{j\theta}$$

where r and θ are now the parameters. Hence we can express $x(n)$ as

$$\begin{aligned} x(n) &= r^n e^{j\theta n} \\ &= r^n(\cos\theta n + j \sin\theta n) \end{aligned} \tag{2.1.10}$$

Since $x(n)$ is now complex valued, it can be represented graphically by plotting the real part

$$x_R(n) \equiv r^n \cos\theta n \tag{2.1.11}$$

as a function of n, and separately plotting the imaginary part

$$x_I(n) \equiv r^n \sin\theta n \tag{2.1.12}$$

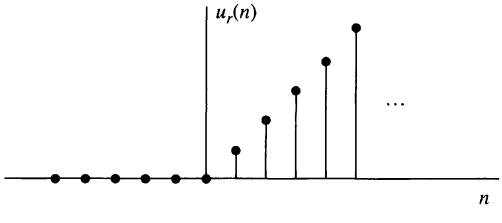


Figure 2.1.4
Graphical representation of the unit ramp signal.

as a function of n. Figure 2.1.6 illustrates the graphs of $x_R(n)$ and $x_I(n)$ for $r = 0.9$ and $\theta = \pi/10$. We observe that the signals $x_R(n)$ and $x_I(n)$ are a damped (decaying exponential) cosine function and a damped sine function. The angle variable θ is simply the frequency of the sinusoid, previously denoted by the (normalized) frequency variable ω. Clearly, if $r = 1$, the damping disappears and $x_R(n)$, $x_I(n)$, and $x(n)$ have a fixed amplitude, which is unity.

Alternatively, the signal $x(n)$ given by (2.1.10) can be represented graphically by the amplitude function

$$|x(n)| = A(n) \equiv r^n \tag{2.1.13}$$

and the phase function

$$\angle x(n) = \phi(n) \equiv \theta n \tag{2.1.14}$$

Figure 2.1.7 illustrates $A(n)$ and $\phi(n)$ for $r = 0.9$ and $\theta = \pi/10$. We observe that the phase function is linear with n. However, the phase is defined only over the interval $-\pi < \theta \leq \pi$ or, equivalently, over the interval $0 \leq \theta < 2\pi$. Consequently, by convention $\phi(n)$ is plotted over the finite interval $-\pi < \theta \leq \pi$ or $0 \leq \theta < 2\pi$. In other words, we subtract multiples of 2π from $\phi(n)$ before plotting. The subtraction of multiples of 2π from $\phi(n)$ is equivalent to interpreting the function $\phi(n)$ as $\phi(n)$, modulo 2π.

2.1.2 Classification of Discrete-Time Signals

The mathematical methods employed in the analysis of discrete-time signals and systems depend on the characteristics of the signals. In this section we classify discrete-time signals according to a number of different characteristics.

Energy signals and power signals. The energy E of a signal $x(n)$ is defined as

$$E \equiv \sum_{n=-\infty}^{\infty} |x(n)|^2 \tag{2.1.15}$$

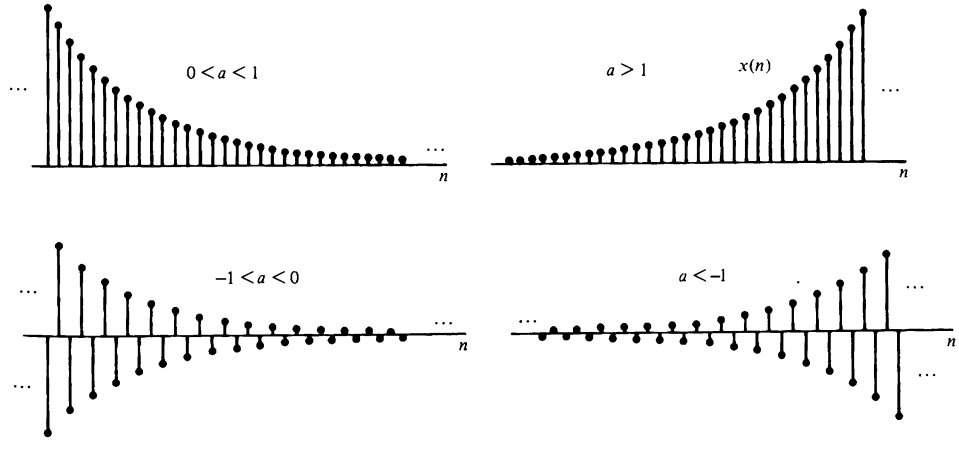


Figure 2.1.5 Graphical representation of exponential signals.

We have used the magnitude-squared values of $x(n)$, so that our definition applies to complex-valued signals as well as real-valued signals. The energy of a signal can be finite or infinite. If E is finite (i.e., $0 < E < \infty$), then $x(n)$ is called an *energy signal*. Sometimes we add a subscript x to E and write E_x to emphasize that E_x is the energy of the signal $x(n)$.

Many signals that possess infinite energy have a finite average power. The average power of a discrete-time signal $x(n)$ is defined as

$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2 \tag{2.1.16}$$

If we define the signal energy of $x(n)$ over the finite interval $-N \le n \le N$ as

$$E_N \equiv \sum_{n=-N}^{N} |x(n)|^2 \tag{2.1.17}$$

then we can express the signal energy E as

$$E \equiv \lim_{N\to\infty} E_N \tag{2.1.18}$$

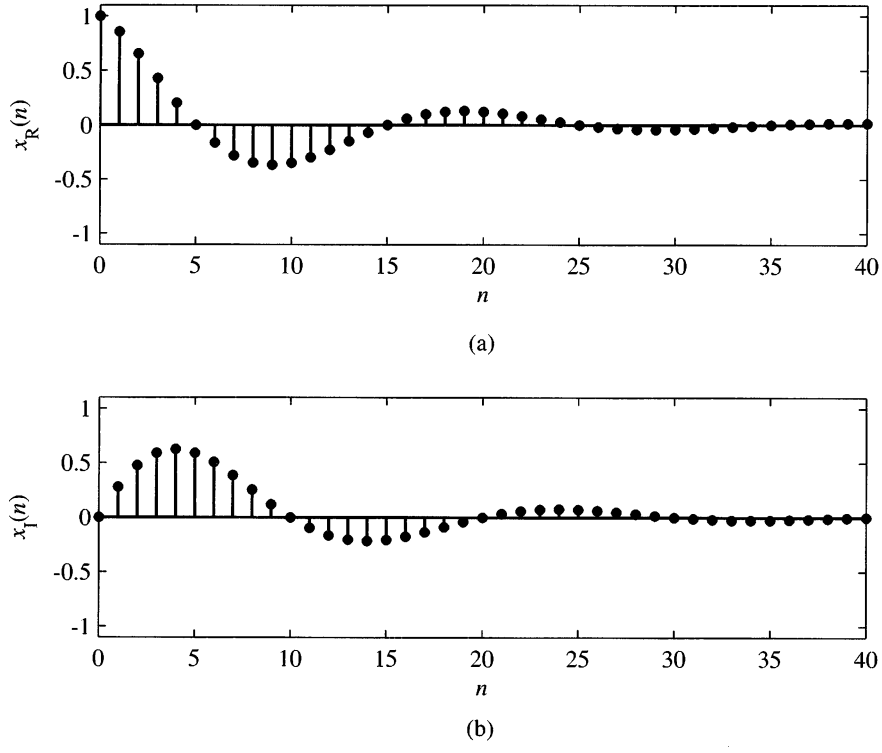


Figure 2.1.6 Graph of the real and imaginary components of a complex-valued exponential signal.

and the average power of the signal $x(n)$ as

$$P \equiv \lim_{N\to\infty} \frac{1}{2N+1} E_N \tag{2.1.19}$$

Clearly, if E is finite, $P = 0$. On the other hand, if E is infinite, the average power P may be either finite or infinite. If P is finite (and nonzero), the signal is called a *power signal.* The following example illustrates such a signal.

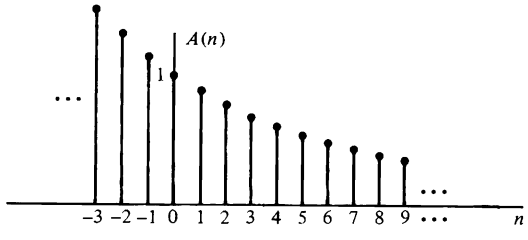
EXAMPLE 2.1.1

Determine the power and energy of the unit step sequence. The average power of the unit step signal is

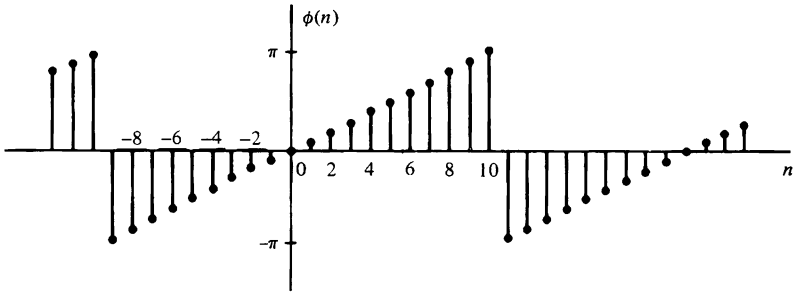
$$P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=0}^{N} u^2(n)$$

$$= \lim_{N\to\infty} \frac{N+1}{2N+1} = \lim_{N\to\infty} \frac{1+1/N}{2+1/N} = \frac{1}{2}$$

Consequently, the unit step sequence is a power signal. Its energy is infinite.



(a) Graph of $A(n) = r^n$, $r = 0.9$



(b) Graph of $\phi(n) = \frac{\pi}{10} n$, modulo 2π plotted in the range $(-\pi, \pi)$

Figure 2.1.7 Graph of amplitude and phase function of a complex-valued exponential signal: (a) graph of $A(n) = r^n$, $r = 0.9$; (b) graph of $\phi(n) = (\pi/10)n$, modulo 2π plotted in the range $(-\pi, \pi]$.

Similarly, it can be shown that the complex exponential sequence $x(n) = Ae^{j\omega_0 n}$ has average power A^2, so it is a power signal. On the other hand, the unit ramp sequence is neither a power signal nor an energy signal.

Periodic signals and aperiodic signals. As defined in Section 1.3, a signal $x(n)$ is periodic with period $N (N > 0)$ if and only if

$$x(n + N) = x(n) \text{ for all } n \tag{2.1.20}$$

The smallest value of N for which (2.1.20) holds is called the (fundamental) period. If there is no value of N that satisfies (2.1.20), the signal is called *nonperiodic* or *aperiodic*.

We have already observed that the sinusoidal signal of the form

$$x(n) = A \sin 2\pi f_0 n \tag{2.1.21}$$

is periodic when f_0 is a rational number, that is, if f_0 can be expressed as

$$f_0 = \frac{k}{N} \tag{2.1.22}$$

where k and N are integers.

The energy of a periodic signal $x(n)$ over a single period, say, over the interval $0 \leq n \leq N - 1$, is finite if $x(n)$ takes on finite values over the period. However, the energy of the periodic signal for $-\infty \leq n \leq \infty$ is infinite. On the other hand, the average power of the periodic signal is finite and it is equal to the average power over a single period. Thus if $x(n)$ is a periodic signal with fundamental period N and takes on finite values, its power is given by

$$P = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 \tag{2.1.23}$$

Consequently, periodic signals are power signals.

Symmetric (even) and antisymmetric (odd) signals. A real-valued signal $x(n)$ is called symmetric (even) if

$$x(-n) = x(n) \tag{2.1.24}$$

On the other hand, a signal $x(n)$ is called antisymmetric (odd) if

$$x(-n) = -x(n) \tag{2.1.25}$$

We note that if $x(n)$ is odd, then $x(0) = 0$. Examples of signals with even and odd symmetry are illustrated in Fig. 2.1.8.

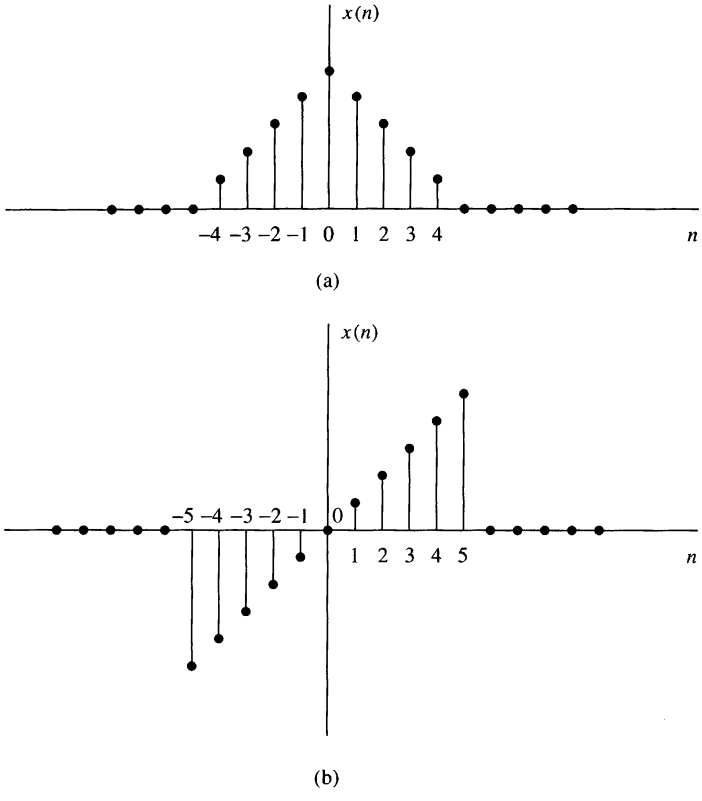


Figure 2.1.8 Example of even (a) and odd (b) signals.

We wish to illustrate that any arbitrary signal can be expressed as the sum of two signal components, one of which is even and the other odd. The even signal component is formed by adding $x(n)$ to $x(-n)$ and dividing by 2, that is,

$$x_e(n) = \frac{1}{2}[x(n) + x(-n)] \tag{2.1.26}$$

Clearly, $x_e(n)$ satisfies the symmetry condition (2.1.24). Similarly, we form an odd signal component $x_o(n)$ according to the relation

$$x_o(n) = \frac{1}{2}[x(n) - x(-n)] \tag{2.1.27}$$

Again, it is clear that $x_o(n)$ satisfies (2.1.25); hence it is indeed odd. Now, if we add the two signal components, defined by (2.1.26) and (2.1.27), we obtain $x(n)$, that is,

$$x(n) = x_e(n) + x_o(n) \tag{2.1.28}$$

Thus any arbitrary signal can be expressed as in (2.1.28).

2.1.3 Simple Manipulations of Discrete-Time Signals

In this section we consider some simple modifications or manipulations involving the independent variable and the signal amplitude (dependent variable).

Transformation of the independent variable (time). A signal $x(n)$ may be shifted in time by replacing the independent variable n by $n - k$, where k is an integer. If k is a positive integer, the time shift results in a delay of the signal by k units of time. If k is a negative integer, the time shift results in an advance of the signal by $|k|$ units in time.

EXAMPLE 2.1.2

A signal $x(n)$ is graphically illustrated in Fig. 2.1.9(a). Show a graphical representation of the signals $x(n - 3)$ and $x(n + 2)$.

Solution. The signal $x(n - 3)$ is obtained by delaying $x(n)$ by three units in time. The result is illustrated in Fig. 2.1.9(b). On the other hand, the signal $x(n + 2)$ is obtained by advancing $x(n)$ by two units in time. The result is illustrated in Fig. 2.1.9(c). Note that delay corresponds to shifting a signal to the right, whereas advance implies shifting the signal to the left on the time axis.

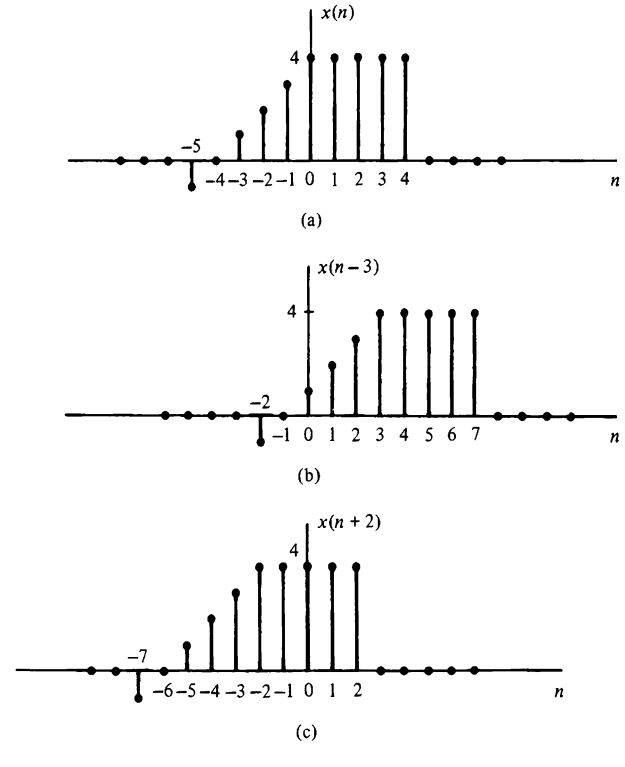


Figure 2.1.9
Graphical representation of a signal, and its delayed and advanced versions.

If the signal $x(n)$ is stored on magnetic tape or on a disk or, perhaps, in the memory of a computer, it is a relatively simple operation to modify the base by introducing a delay or an advance. On the other hand, if the signal is not stored but is being generated by some physical phenomenon in real time, it is not possible to advance the signal in time, since such an operation involves signal samples that have not yet been generated. Whereas it is always possible to insert a delay into signal samples that have already been generated, it is physically impossible to view the future signal samples. Consequently, in real-time signal processing applications, the operation of advancing the time base of the signal is physically unrealizable.

Another useful modification of the time base is to replace the independent variable n by $-n$. The result of this operation is a *folding* or a *reflection* of the signal about the time origin $n = 0$.

EXAMPLE 2.1.3

Show the graphical representation of the signals $x(-n)$ and $x(-n+2)$, where $x(n)$ is the signal illustrated in Fig. 2.1.10(a).

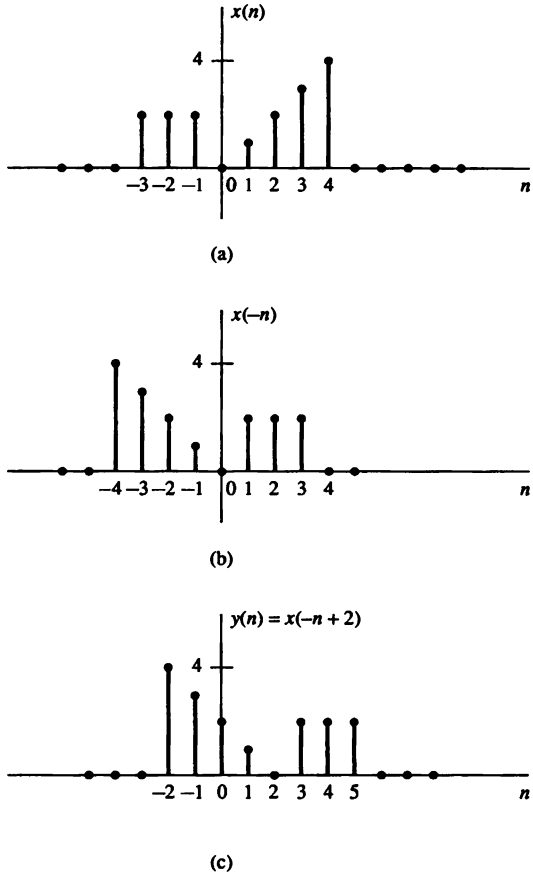


Figure 2.1.10
Graphical illustration of the folding and shifting operations.

Solution. The new signal $y(n) = x(-n)$ is shown in Fig. 2.1.10(b). Note that $y(0) = x(0)$, $y(1) = x(-1)$, $y(2) = x(-2)$, and so on. Also, $y(-1) = x(1)$, $y(-2) = x(2)$, and so on. Therefore, $y(n)$ is simply $x(n)$ reflected or folded about the time origin $n = 0$. The signal $y(n) = x(-n + 2)$ is simply $x(-n)$ delayed by two units in time. The resulting signal is illustrated in Fig. 2.1.10(c). A simple way to verify that the result in Fig. 2.1.10(c) is correct is to compute samples, such as $y(0) = x(2)$, $y(1) = x(1)$, $y(2) = x(0)$, $y(-1) = x(3)$, and so on.

It is important to note that the operations of folding and time delaying (or advancing) a signal are not commutative. If we denote the time-delay operation by TD and the folding operation by FD, we can write

$$\begin{aligned} \mathrm{TD}_k[x(n)] &= x(n-k), \qquad k > 0 \\ \mathrm{FD}[x(n)] &= x(-n) \end{aligned} \tag{2.1.29}$$

Now

$$\mathrm{TD}_k\{\mathrm{FD}[x(n)]\} = \mathrm{TD}_k[x(-n)] = x(-n+k) \tag{2.1.30}$$

whereas

$$\mathrm{FD}\{\mathrm{TD}_k[x(n)]\} = \mathrm{FD}[x(n-k)] = x(-n-k) \tag{2.1.31}$$

Note that because the signs of n and k in $x(n-k)$ and $x(-n+k)$ are different, the result is a shift of the signals $x(n)$ and $x(-n)$ to the right by k samples, corresponding to a time delay.

A third modification of the independent variable involves replacing n by μn, where μ is an integer. We refer to this time-base modification as *time scaling* or *down-sampling*.

EXAMPLE 2.1.4

Show the graphical representation of the signal $y(n) = x(2n)$, where $x(n)$ is the signal illustrated in Fig. 2.1.11(a).

Solution. We note that the signal $y(n)$ is obtained from $x(n)$ by taking every other sample from $x(n)$, starting with $x(0)$. Thus $y(0) = x(0)$, $y(1) = x(2)$, $y(2) = x(4), \ldots$ and $y(-1) = x(-2)$, $y(-2) = x(-4)$, and so on. In other words, we have skipped the odd-numbered samples in $x(n)$ and retained the even-numbered samples. The resulting signal is illustrated in Fig. 2.1.11(b).

If the signal $x(n)$ was originally obtained by sampling an analog signal $x_a(t)$, then $x(n) = x_a(nT)$, where T is the sampling interval. Now, $y(n) = x(2n) = x_a(2Tn)$. Hence the time-scaling operation described in Example 2.1.4 is equivalent to changing the sampling rate from $1/T$ to $1/2T$, that is, to decreasing the rate by a factor of 2. This is a *down-sampling* operation.

Addition, multiplication, and scaling of sequences. Amplitude modifications include *addition, multiplication,* and *scaling* of discrete-time signals.

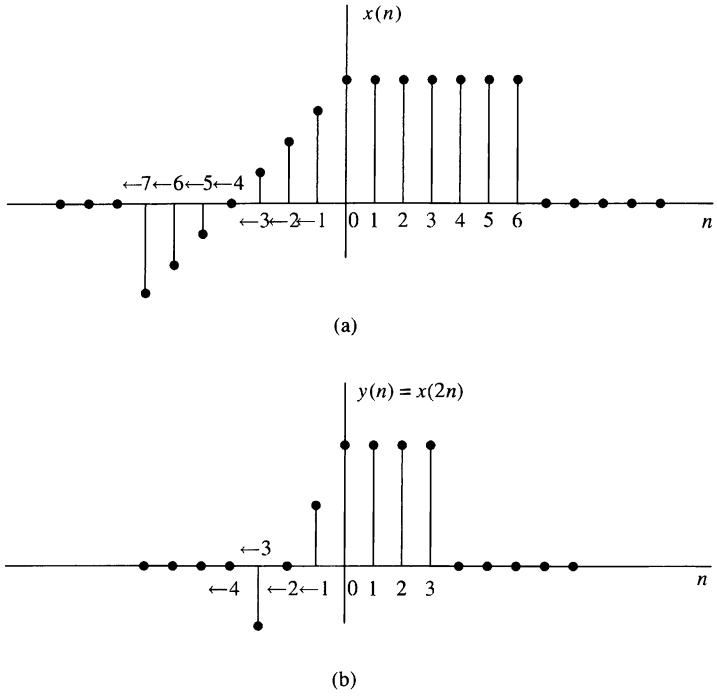


Figure 2.1.11 Graphical illustration of down-sampling operation.

Amplitude scaling of a signal by a constant A is accomplished by multiplying the value of every signal sample by A. Consequently, we obtain

$$y(n) = Ax(n), \qquad -\infty < n < \infty$$

The *sum* of two signals $x_1(n)$ and $x_2(n)$ is a signal $y(n)$, whose value at any instant is equal to the sum of the values of these two signals at that instant, that is,

$$y(n) = x_1(n) + x_2(n), \qquad -\infty < n < \infty$$

The *product* of two signals is similarly defined on a sample-to-sample basis as

$$y(n) = x_1(n)x_2(n), \qquad -\infty < n < \infty$$

2.2 Discrete-Time Systems

In many applications of digital signal processing we wish to design a device or an algorithm that performs some prescribed operation on a discrete-time signal. Such a device or algorithm is called a discrete-time system. More specifically, a *discrete-time system* is a device or algorithm that operates on a discrete-time signal, called the *input* or *excitation*, according to some well-defined rule, to produce another discrete-time

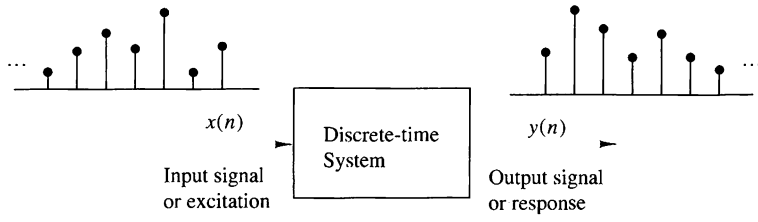


Figure 2.2.1 Block diagram representation of a discrete-time system.

signal called the *output* or *response* of the system. In general, we view a system as an operation or a set of operations performed on the input signal $x(n)$ to produce the output signal $y(n)$. We say that the input signal $x(n)$ is *transformed* by the system into a signal $y(n)$, and express the general relationship between $x(n)$ and $y(n)$ as

$$y(n) \equiv \mathcal{T}[x(n)] \tag{2.2.1}$$

where the symbol $\mathcal{T}$ denotes the transformation (also called an operator) or processing performed by the system on $x(n)$ to produce $y(n)$. The mathematical relationship in (2.2.1) is depicted graphically in Fig. 2.2.1.

There are various ways to describe the characteristics of the system and the operation it performs on $x(n)$ to produce $y(n)$. In this chapter we shall be concerned with the time-domain characterization of systems. We shall begin with an input–output description of the system. The input–output description focuses on the behavior at the terminals of the system and ignores the detailed internal construction or realization of the system. Later, in Chapter 9, we consider the implementation of discrete-time systems and describe the different structures for their realization.

2.2.1 Input–Output Description of Systems

The input–output description of a discrete-time system consists of a mathematical expression or a rule, which explicitly defines the relation between the input and output signals (*input–output relationship*). The exact internal structure of the system is either unknown or ignored. Thus the only way to interact with the system is by using its input and output terminals (i.e., the system is assumed to be a "black box" to the user). To reflect this philosophy, we use the graphical representation depicted in Fig. 2.2.1, and the general input–output relationship in (2.2.1) or, alternatively, the notation

$$x(n) \xrightarrow{\mathcal{T}} y(n) \tag{2.2.2}$$

which simply means that $y(n)$ is the response of the system $\mathcal{T}$ to the excitation $x(n)$. The following examples illustrate several different systems.

EXAMPLE 2.2.1

Determine the response of the following sytems to the input signal

$$x(n) = \begin{cases} |n|, & -3 \le n \le 3 \\ 0, & \text{otherwise} \end{cases}$$

(a) $y(n) = x(n)$ (identity system)

(b) $y(n) = x(n-1)$ (unit delay system)

(c) $y(n) = x(n+1)$ (unit advance system)

(d) $y(n) = \frac{1}{3}[x(n+1) + x(n) + x(n-1)]$ (moving average filter)

(e) $y(n) = \text{median}\{x(n+1), x(n), x(n-1)\}$ (median filter)

(f) $y(n) = \sum_{k=-\infty}^{n} x(k) = x(n) + x(n-1) + x(n-2) + \cdots$ (accumulator) (2.2.3)

Solution. First, we determine explicitly the sample values of the input signal

$$x(n) = \{\ldots, 0, 3, 2, 1, \underset{\uparrow}{0}, 1, 2, 3, 0, \ldots\}$$

Next, we determine the output of each system using its input–output relationship.

(a) In this case the output is exactly the same as the input signal. Such a system is known as the *identity* system.

(b) This system simply delays the input by one sample. Thus its output is given by

$$x(n) = \{\ldots, 0, 3, 2, \underset{\uparrow}{1}, 0, 1, 2, 3, 0, \ldots\}$$

(c) In this case the system "advances" the input one sample into the future. For example, the value of the output at time $n = 0$ is $y(0) = x(1)$. The response of this system to the given input is

$$x(n) = \{\ldots, 0, 3, 2, 1, 0, \underset{\uparrow}{1}, 2, 3, 0, \ldots\}$$

(d) The output of this system at any time is the mean value of the present, the immediate past, and the immediate future samples. For example, the output at time $n = 0$ is

$$y(0) = \frac{1}{3}[x(-1) + x(0) + x(1)] = \frac{1}{3}[1 + 0 + 1] = \frac{2}{3}$$

Repeating this computation for every value of n, we obtain the output signal

$$y(n) = \{\ldots, 0, 1, \frac{5}{3}, 2, 1, \underset{\uparrow}{\frac{2}{3}}, 1, 2, \frac{5}{3}, 1, 0, \ldots\}$$

(e) This system selects as its output at time n the median value of the three input samples $x(n-1)$, $x(n)$, and $x(n+1)$. Thus the response of this system to the input signal $x(n)$ is

$$y(n) = \{0, 2, 2, 1, 1, \underset{\uparrow}{1}, 2, 2, 0, 0, 0, \ldots\}$$

(f) This system is basically an *accumulator* that computes the running sum of all the past input values up to present time. The response of this system to the given input is

$$y(n) = \{\ldots, 0, 3, 5, 6, \underset{\uparrow}{6}, 7, 9, 12, 0, \ldots\}$$

We observe that for several of the systems considered in Example 2.2.1 the output at time $n = n_0$ depends not only on the value of the input at $n = n_0$ [i.e., $x(n_0)$], but also on the values of the input applied to the system before and after $n = n_0$. Consider, for instance, the accumulator in the example. We see that the output at time $n = n_0$ depends not only on the input at time $n = n_0$, but also on $x(n)$ at times $n = n_0 - 1$, $n_0 - 2$, and so on. By a simple algebraic manipulation the input–output relation of the accumulator can be written as

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{n} x(k) = \sum_{k=-\infty}^{n-1} x(k) + x(n) \\ &= y(n-1) + x(n) \end{aligned} \tag{2.2.4}$$

which justifies the term *accumulator*. Indeed, the system computes the current value of the output by adding (accumulating) the current value of the input to the previous output value.

There are some interesting conclusions that can be drawn by taking a close look into this apparently simple system. Suppose that we are given the input signal $x(n)$ for $n \geq n_0$, and we wish to determine the output $y(n)$ of this system for $n \geq n_0$. For $n = n_0, n_0 + 1, \ldots$, (2.2.4) gives

$$y(n_0) = y(n_0 - 1) + x(n_0)$$

$$y(n_0 + 1) = y(n_0) + x(n_0 + 1)$$

and so on. Note that we have a problem in computing $y(n_0)$, since it depends on $y(n_0 - 1)$. However,

$$y(n_0 - 1) = \sum_{k=-\infty}^{n_0-1} x(k)$$

that is, $y(n_0 - 1)$ "summarizes" the effect on the system from all the inputs which had been applied to the system before time n_0. Thus the response of the system for $n \geq n_0$ to the input $x(n)$ that is applied at time n_0 is the combined result of this input and all inputs that had been applied previously to the system. Consequently, $y(n)$, $n \geq n_0$ is not uniquely determined by the input $x(n)$ for $n \geq n_0$.

The additional information required to determine $y(n)$ for $n \geq n_0$ is the *initial condition* $y(n_0 - 1)$. This value summarizes the effect of all previous inputs to the system. Thus the initial condition $y(n_0 - 1)$ together with the input sequence $x(n)$ for $n \geq n_0$ uniquely determine the output sequence $y(n)$ for $n \geq n_0$.

If the accumulator had no excitation prior to n_0, the initial condition is $y(n_0 - 1) = 0$. In such a case we say that the system is *initially relaxed*. Since $y(n_0 - 1) = 0$, the output sequence $y(n)$ depends only on the input sequence $x(n)$ for $n \geq n_0$.

It is customary to assume that every system is relaxed at $n = -\infty$. In this case, if an input $x(n)$ is applied at $n = -\infty$, the corresponding output $y(n)$ is *solely* and *uniquely* determined by the given input.

EXAMPLE 2.2.2

The accumulator described by (2.2.30) is excited by the sequence $x(n) = nu(n)$. Determine its output under the condition that:

(a) It is initially relaxed [i.e., $y(-1) = 0$].

(b) Initially, $y(-1) = 1$.

Solution. The output of the system is defined as

$$y(n) = \sum_{k=-\infty}^{n} x(k) = \sum_{k=-\infty}^{-1} x(k) + \sum_{k=0}^{n} x(k)$$

$$= y(-1) + \sum_{k=0}^{n} x(k)$$

$$= y(-1) + \frac{n(n+1)}{2}$$

(a) If the system is initially relaxed, $y(-1) = 0$ and hence

$$y(n) = \frac{n(n+1)}{2}, \qquad n \geq 0$$

(b) On the other hand, if the initial condition is $y(-1) = 1$, then

$$y(n) = 1 + \frac{n(n+1)}{2} = \frac{n^2 + n + 2}{2}, \qquad n \geq 0$$

2.2.2 Block Diagram Representation of Discrete-Time Systems

It is useful at this point to introduce a block diagram representation of discrete-time systems. For this purpose we need to define some basic building blocks that can be interconnected to form complex systems.

An adder. Figure 2.2.2 illustrates a system (adder) that performs the addition of two signal sequences to form another (the sum) sequence, which we denote as $y(n)$. Note that it is not necessary to store either one of the sequences in order to perform the addition. In other words, the addition operation is *memoryless*.

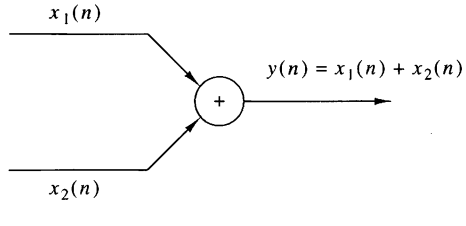


Figure 2.2.2
Graphical representation of an adder.

A constant multiplier. This operation is depicted by Fig. 2.2.3, and simply represents applying a scale factor on the input $x(n)$. Note that this operation is also memoryless.

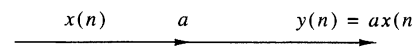


Figure 2.2.3
Graphical representation of a constant multiplier.

A signal multiplier. Figure 2.2.4 illustrates the multiplication of two signal sequences to form another (the product) sequence, denoted in the figure as $y(n)$. As in the preceding two cases, we can view the multiplication operation as memoryless.

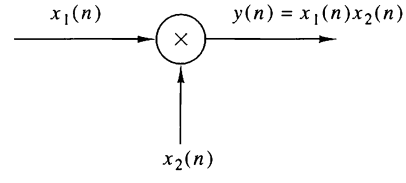


Figure 2.2.4
Graphical representation of a signal multiplier.

A unit delay element. The unit delay is a special system that simply delays the signal passing through it by one sample. Figure 2.2.5 illustrates such a system. If the input signal is $x(n)$, the output is $x(n-1)$. In fact, the sample $x(n-1)$ is stored in memory at time $n-1$ and it is recalled from memory at time n to form

$$y(n) = x(n-1)$$

Thus this basic building block requires memory. The use of the symbol z^{-1} to denote the unit of delay will become apparent when we discuss the z-transform in Chapter 3.

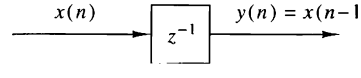


Figure 2.2.5
Graphical representation of the unit delay element.

A unit advance element. In contrast to the unit delay, a unit advance moves the input $x(n)$ ahead by one sample in time to yield $x(n+1)$. Figure 2.2.6 illustrates this operation, with the operator z being used to denote the unit advance. We observe that any such advance is physically impossible in real time, since, in fact, it involves looking into the future of the signal. On the other hand, if we store the signal in the memory of the computer, we can recall any sample at any time. In such a non-real-time application, it is possible to advance the signal $x(n)$ in time.

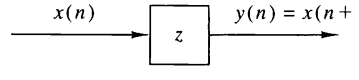


Figure 2.2.6
Graphical representation of the unit advance element.

EXAMPLE 2.2.3

Using basic building blocks introduced above, sketch the block diagram representation of the discrete-time system described by the input–output relation

$$y(n) = \frac{1}{4}y(n-1) + \frac{1}{2}x(n) + \frac{1}{2}x(n-1) \tag{2.2.5}$$

where $x(n)$ is the input and $y(n)$ is the output of the system.

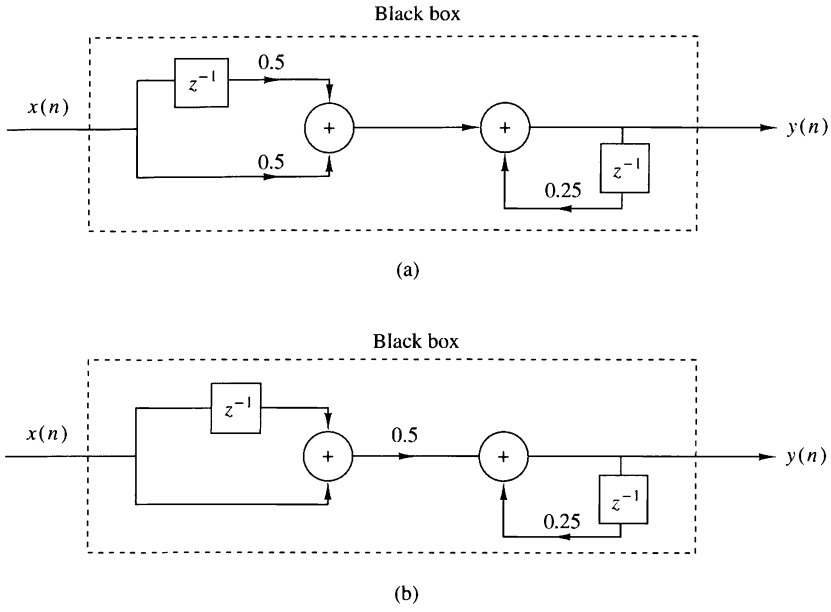


Figure 2.2.7 Block diagram realizations of the system $y(n) = 0.25y(n-1) + 0.5x(n) + 0.5x(n-1)$.

Solution. According to (2.2.5), the output $y(n)$ is obtained by multiplying the input $x(n)$ by 0.5, multiplying the previous input $x(n-1)$ by 0.5, adding the two products, and then adding the previous output $y(n-1)$ multiplied by $\frac{1}{4}$. Figure 2.2.7(a) illustrates this block diagram realization of the system. A simple rearrangement of (2.2.5), namely,

$$y(n) = \frac{1}{4}y(n-1) + \frac{1}{2}[x(n) + x(n-1)] \tag{2.2.6}$$

leads to the block diagram realization shown in Fig. 2.2.7(b). Note that if we treat "the system" from the "viewpoint" of an input–output or an external description, we are not concerned about how the system is realized. On the other hand, if we adopt an internal description of the system, we know exactly how the system building blocks are configured. In terms of such a realization, we can see that a system is *relaxed* at time $n = n_0$ if the outputs of all the *delays* existing in the system are zero at $n = n_0$ (i.e., all memory is *filled* with zeros).

2.2.3 Classification of Discrete-Time Systems

In the analysis as well as in the design of systems, it is desirable to classify the systems according to the general properties that they satisfy. In fact, the mathematical techniques that we develop in this and in subsequent chapters for analyzing and designing discrete-time systems depend heavily on the general characteristics of the systems that are being considered. For this reason it is necessary for us to develop a number of properties or categories that can be used to describe the general characteristics of systems.

We stress the point that for a system to possess a given property, the property must hold for every possible input signal to the system. If a property holds for some

input signals but not for others, the system does not possess that property. Thus a counterexample is sufficient to prove that a system does not possess a property. However, to prove that the system has some property, we must prove that this property holds for every possible input signal.

Static versus dynamic systems. A discrete-time system is called *static* or memoryless if its output at any instant n depends at most on the input sample at the same time, but not on past or future samples of the input. In any other case, the system is said to be *dynamic* or to have memory. If the output of a system at time n is completely determined by the input samples in the interval from $n - N$ to $n (N \geq 0)$, the system is said to have *memory* of duration N. If $N = 0$, the system is static. If $0 < N < \infty$, the system is said to have *finite memory*, whereas if $N = \infty$, the system is said to have *infinite memory*.

The systems described by the following input–output equations

$$y(n) = ax(n) \tag{2.2.7}$$

$$y(n) = nx(n) + bx^3(n) \tag{2.2.8}$$

are both static or memoryless. Note that there is no need to store any of the past inputs or outputs in order to compute the present output. On the other hand, the systems described by the following input–output relations

$$y(n) = x(n) + 3x(n-1) \tag{2.2.9}$$

$$y(n) = \sum_{k=0}^{n} x(n-k) \tag{2.2.10}$$

$$y(n) = \sum_{k=0}^{\infty} x(n-k) \tag{2.2.11}$$

are dynamic systems or systems with memory. The systems described by (2.2.9) and (2.2.10) have finite memory, whereas the system described by (2.2.11) has infinite memory.

We observe that static or memoryless systems are described in general by input–output equations of the form

$$y(n) = \mathcal{T}[x(n), n] \tag{2.2.12}$$

and they do not include delay elements (memory).

Time-invariant versus time-variant systems. We can subdivide the general class of systems into the two broad categories, time-invariant systems and time-variant systems. A system is called time-invariant if its input–output characteristics do not change with time. To elaborate, suppose that we have a system $\mathcal{T}$ in a relaxed state

which, when excited by an input signal $x(n)$, produces an output signal $y(n)$. Thus we write

$$y(n) = \mathcal{T}[x(n)] \tag{2.2.13}$$

Now suppose that the same input signal is delayed by k units of time to yield $x(n-k)$, and again applied to the same system. If the characteristics of the system do not change with time, the output of the relaxed system will be $y(n-k)$. That is, the output will be the same as the response to $x(n)$, except that it will be delayed by the same k units in time that the input was delayed. This leads us to define a time-invariant or shift-invariant system as follows.

Definition. A relaxed system $\mathcal{T}$ is *time invariant* or *shift invariant* if and only if

$$x(n) \xrightarrow{\mathcal{T}} y(n)$$

implies that

$$x(n-k) \xrightarrow{\mathcal{T}} y(n-k) \tag{2.2.14}$$

for every input signal $x(n)$ and every time shift k.

To determine if any given system is time invariant, we need to perform the test specified by the preceding definition. Basically, we excite the system with an arbitrary input sequence $x(n)$, which produces an output denoted as $y(n)$. Next we delay the input sequence by some amount k and recompute the output. In general, we can write the output as

$$y(n,k) = \mathcal{T}[x(n-k)]$$

Now if this output $y(n,k) = y(n-k)$, for all possible values of k, the system is time invariant. On the other hand, if the output $y(n,k) \neq y(n-k)$, even for one value of k, the system is time variant.

EXAMPLE 2.2.4

Determine if the systems shown in Fig. 2.2.8 are time invariant or time variant.

Solution.

(a) This system is described by the input–output equations

$$y(n) = \mathcal{T}[x(n)] = x(n) - x(n-1) \tag{2.2.15}$$

Now if the input is delayed by k units in time and applied to the system, it is clear from the block diagram that the output will be

$$y(n,k) = x(n-k) - x(n-k-1) \tag{2.2.16}$$

On the other hand, from (2.2.14) we note that if we delay $y(n)$ by k units in time, we obtain

$$y(n-k) = x(n-k) - x(n-k-1) \tag{2.2.17}$$

Since the right-hand sides of (2.2.16) and (2.2.17) are identical, it follows that $y(n,k) = y(n-k)$. Therefore, the system is time invariant.

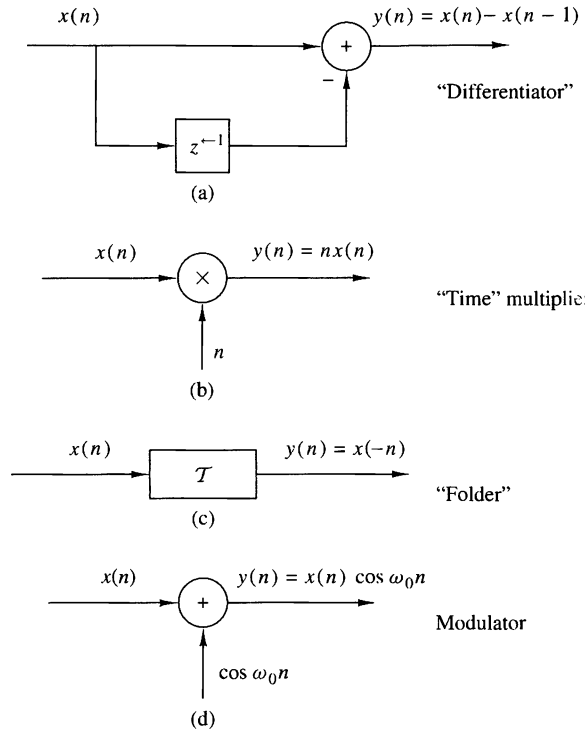


Figure 2.2.8
Examples of a time-invariant (a) and some time-variant systems (b)–(d).

(b) The input–output equation for this system is

$$y(n) = \mathcal{T}[x(n)] = nx(n) \tag{2.2.18}$$

The response of this system to $x(n-k)$ is

$$y(n,k) = nx(n-k) \tag{2.2.19}$$

Now if we delay $y(n)$ in (2.2.18) by k units in time, we obtain

$$\begin{aligned} y(n-k) &= (n-k)x(n-k) \\ &= nx(n-k) - kx(n-k) \end{aligned} \tag{2.2.20}$$

This system is time variant, since $y(n,k) \neq y(n-k)$.

(c) This system is described by the input–output relation

$$y(n) = \mathcal{T}[x(n)] = x(-n) \tag{2.2.21}$$

The response of this system to $x(n-k)$ is

$$y(n,k) = \mathcal{T}[x(n-k)] = x(-n-k) \tag{2.2.22}$$

Now, if we delay the output $y(n)$, as given by (2.2.21), by k units in time, the result will be

$$y(n-k) = x(-n+k) \tag{2.2.23}$$

Since $y(n,k) \neq y(n-k)$, the system is time variant.

(d) The input–output equation for this system is

$$y(n) = x(n)\cos\omega_0 n \tag{2.2.24}$$

The response of this system to $x(n-k)$ is

$$y(n,k) = x(n-k)\cos\omega_0 n \tag{2.2.25}$$

If the expression in (2.2.24) is delayed by k units and the result is compared to (2.2.25), it is evident that the system is time variant.

Linear versus nonlinear systems. The general class of systems can also be subdivided into linear systems and nonlinear systems. A linear system is one that satisfies the *superposition principle*. Simply stated, the principle of superposition requires that the response of the system to a weighted sum of signals be equal to the corresponding weighted sum of the responses (outputs) of the system to each of the individual input signals. Hence we have the following definition of linearity.

Definition. A system is linear if and only if

$$\mathcal{T}[a_1x_1(n) + a_2x_2(n)] = a_1\mathcal{T}[x_1(n)] + a_2\mathcal{T}[x_2(n)] \tag{2.2.26}$$

for any arbitrary input sequences $x_1(n)$ and $x_2(n)$, and any arbitrary constants a_1 and a_2. Figure 2.2.9 gives a pictorial illustration of the superposition principle.

The superposition principle embodied in the relation (2.2.26) can be separated into two parts. First, suppose that $a_2 = 0$. Then (2.2.26) reduces to

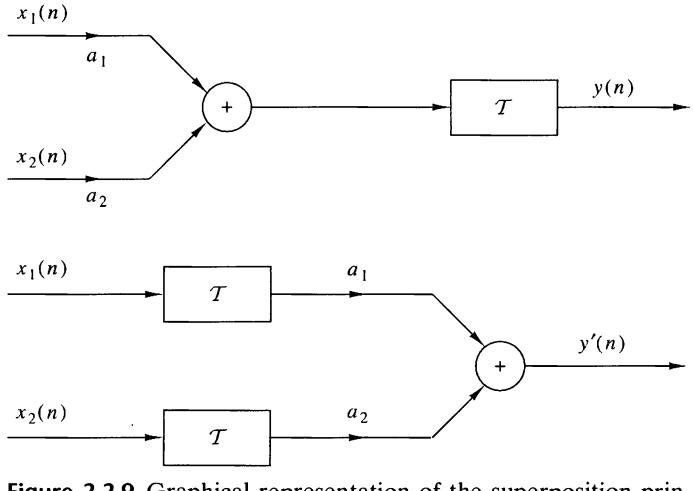


Figure 2.2.9 Graphical representation of the superposition principle. $\mathcal{T}$ is linear if and only if $y(n) = y'(n)$.

$$\mathcal{T}[a_1x_1(n)] = a_1\mathcal{T}[x_1(n)] = a_1y_1(n) \tag{2.2.27}$$

where

$$y_1(n) = \mathcal{T}[x_1(n)]$$

The relation (2.2.27) demonstrates the *multiplicative* or *scaling property* of a linear system. That is, if the response of the system to the input $x_1(n)$ is $y_1(n)$, the response to $a_1x_1(n)$ is simply $a_1y_1(n)$. Thus any scaling of the input results in an identical scaling of the corresponding output.

Second, suppose that $a_1 = a_2 = 1$ in (2.2.26). Then

$$\begin{aligned}\mathcal{T}[x_1(n) + x_2(n)] &= \mathcal{T}[x_1(n)] + \mathcal{T}[x_1(n)] \\ &= y_1(n) + y_2(n)\end{aligned} \tag{2.2.28}$$

This relation demonstrates the *additivity property* of a linear system. The additivity and multiplicative properties constitute the superposition principle as it applies to linear systems.

The linearity condition embodied in (2.2.26) can be extended arbitrarily to any weighted linear combination of signals by induction. In general, we have

$$x(n) = \sum_{k=1}^{M-1} a_kx_k(n) \xrightarrow{\mathcal{T}} y(n) = \sum_{k=1}^{M-1} a_ky_k(n) \tag{2.2.29}$$

where

$$y_k(n) = \mathcal{T}[x_k(n)], \qquad k = 1, 2, \ldots, M-1 \tag{2.2.30}$$

We observe from (2.2.27) that if $a_1 = 0$, then $y(n) = 0$. In other words, a relaxed, linear system with zero input produces a zero output. If a system produces a nonzero output with a zero input, the system may be either nonrelaxed or nonlinear. If a relaxed system does not satisfy the superposition principle as given by the definition above, it is called *nonlinear.*

EXAMPLE 2.2.5

Determine if the systems described by the following input–output equations are linear or nonlinear.

(a) $y(n) = nx(n)$ **(b)** $y(n) = x(n^2)$ **(c)** $y(n) = x^2(n)$
(d) $y(n) = Ax(n) + B$ **(e)** $y(n) = e^{x(n)}$

Solution.

(a) For two input sequences $x_1(n)$ and $x_2(n)$, the corresponding outputs are

$$\begin{aligned}y_1(n) &= nx_1(n) \\ y_2(n) &= nx_2(n)\end{aligned} \tag{2.2.31}$$

A linear combination of the two input sequences results in the output

$$\begin{aligned}y_3(n) &= \mathcal{T}[a_1x_1(n) + a_2x_2(n)] = n[a_1x_1(n) + a_2x_2(n)] \\ &= a_1nx_1(n) + a_2nx_2(n)\end{aligned} \tag{2.2.32}$$

On the other hand, a linear combination of the two outputs in (2.2.31) results in the output

$$a_1 y_1(n) + a_2 y_2(n) = a_1 n x_1(n) + a_2 n x_2(n) \tag{2.2.33}$$

Since the right-hand sides of (2.2.32) and (2.2.33) are identical, the system is linear.

(b) As in part (a), we find the response of the system to two separate input signals $x_1(n)$ and $x_2(n)$. The result is

$$\begin{aligned} y_1(n) &= x_1(n^2) \\ y_2(n) &= x_2(n^2) \end{aligned} \tag{2.2.34}$$

The output of the system to a linear combination of $x_1(n)$ and $x_2(n)$ is

$$y_3(n) = \mathcal{T}[a_1 x_1(n) + a_2 x_2(n)] = a_1 x_1(n^2) + a_2 x_2(n^2) \tag{2.2.35}$$

Finally, a linear combination of the two outputs in (2.2.34) yields

$$a_1 y_1(n) + a_2 y_2(n) = a_1 x_1(n^2) + a_2 x_2(n^2) \tag{2.2.36}$$

By comparing (2.2.35) with (2.2.36), we conclude that the system is linear.

(c) The output of the system is the square of the input. (Electronic devices that have such an input–output characteristic are called square-law devices.) From our previous discussion it is clear that such a system is memoryless. We now illustrate that this system is nonlinear.

The responses of the system to two separate input signals are

$$\begin{aligned} y_1(n) &= x_1^2(n) \\ y_2(n) &= x_2^2(n) \end{aligned} \tag{2.2.37}$$

The response of the system to a linear combination of these two input signals is

$$\begin{aligned} y_3(n) &= \mathcal{T}[a_1 x_1(n) + a_2 x_2(n)] \\ &= [a_1 x_1(n) + a_2 x_2(n)]^2 \\ &= a_1^2 x_1^2(n) + 2a_1 a_2 x_1(n) x_2(n) + a_2^2 x_2^2(n) \end{aligned} \tag{2.2.38}$$

On the other hand, if the system is linear, it will produce a linear combination of the two outputs in (2.2.37), namely,

$$a_1 y_1(n) + a_2 y_2(n) = a_1 x_1^2(n) + a_2 x_2^2(n) \tag{2.2.39}$$

Since the actual output of the system, as given by (2.2.38), is not equal to (2.2.39), the system is nonlinear.

(d) Assuming that the system is excited by $x_1(n)$ and $x_2(n)$ separately, we obtain the corresponding outputs

$$\begin{aligned} y_1(n) &= Ax_1(n) + B \\ y_2(n) &= Ax_2(n) + B \end{aligned} \tag{2.2.40}$$

A linear combination of $x_1(n)$ and $x_2(n)$ produces the output

$$\begin{aligned} y_3(n) &= \mathcal{T}[a_1x_1(n) + a_2x_2(n)] \\ &= A[a_1x_1(n) + a_2x_2(n)] + B \\ &= Aa_1x_1(n) + a_2Ax_2(n) + B \end{aligned} \tag{2.2.41}$$

On the other hand, if the system were linear, its output to the linear combination of $x_1(n)$ and $x_2(n)$ would be a linear combination of $y_1(n)$ and $y_2(n)$, that is,

$$a_1y_1(n) + a_2y_2(n) = a_1Ax_1(n) + a_1B + a_2Ax_2(n) + a_2B \tag{2.2.42}$$

Clearly, (2.2.41) and (2.2.42) are different and hence the system fails to satisfy the linearity test.

The reason that this system fails to satisfy the linearity test is not that the system is nonlinear (in fact, the system is described by a linear equation) but the presence of the constant B. Consequently, the output depends on both the input excitation and on the parameter $B \neq 0$. Hence, for $B \neq 0$, the system is not relaxed. If we set $B = 0$, the system is now relaxed and the linearity test is satisfied.

(e) Note that the system described by the input–output equation

$$y(n) = e^{x(n)} \tag{2.2.43}$$

is non-relaxed. If $x(n) = 0$, we find that $y(n) = 1$. This is an indication that the system is nonlinear. This, in fact, is the conclusion reached when the linearity test is applied.

Causal versus noncausal systems. We begin with the definition of causal discrete-time systems.

Definition. A system is said to be *causal* if the output of the system at any time n [i.e., $y(n)$] depends only on present and past inputs [i.e., $x(n)$, $x(n-1)$, $x(n-2)$, ...], but does not depend on future inputs [i.e., $x(n+1)$, $x(n+2)$, ...]. In mathematical terms, the output of a causal system satisfies an equation of the form

$$y(n) = F[x(n), x(n-1), x(n-2), \ldots] \tag{2.2.44}$$

where $F[\cdot]$ is some arbitrary function.

If a system does not satisfy this definition, it is called *noncausal.* Such a system has an output that depends not only on present and past inputs but also on future inputs.

It is apparent that in real-time signal processing applications we cannot observe future values of the signal, and hence a noncausal system is physically unrealizable (i.e., it cannot be implemented). On the other hand, if the signal is recorded so that the processing is done off-line (nonreal time), it is possible to implement a noncausal system, since all values of the signal are available at the time of processing. This is often the case in the processing of geophysical signals and images.

EXAMPLE 2.2.6

Determine if the systems described by the following input–output equations are causal or noncausal.
(a) $y(n) = x(n) - x(n-1)$ **(b)** $y(n) = \sum_{k=-\infty}^{n} x(k)$ **(c)** $y(n) = ax(n)$ **(d)** $y(n) = x(n) + 3x(n+4)$
(e) $y(n) = x(n^2)$ **(f)** $y(n) = x(2n)$ **(g)** $y(n) = x(-n)$

Solution. The systems described in parts (a), (b), and (c) are clearly causal, since the output depends only on the present and past inputs. On the other hand, the systems in parts (d), (e), and (f) are clearly noncausal, since the output depends on future values of the input. The system in (g) is also noncausal, as we note by selecting, for example, $n = -1$, which yields $y(-1) = x(1)$. Thus the output at $n = -1$ depends on the input at $n = 1$, which is two units of time into the future.

Stable versus unstable systems. Stability is an important property that must be considered in any practical application of a system. Unstable systems usually exhibit erratic and extreme behavior and cause overflow in any practical implementation. Here, we define mathematically what we mean by a stable system, and later, in Section 2.3.6, we explore the implications of this definition for linear, time-invariant systems.

Definition. An arbitrary relaxed system is said to be bounded input–bounded output (BIBO) stable if and only if every bounded input produces a bounded output.

The condition that the input sequence $x(n)$ and the output sequence $y(n)$ are bounded is translated mathematically to mean that there exist some finite numbers, say M_x and M_y, such that

$$|x(n)| \leq M_x < \infty, \qquad |y(n)| \leq M_y < \infty \tag{2.2.45}$$

for all n. If, for some bounded input sequence $x(n)$, the output is unbounded (infinite), the system is classified as unstable.

EXAMPLE 2.2.7

Consider the nonlinear system described by the input–output equation

$$y(n) = y^2(n-1) + x(n)$$

As an input sequence we select the bounded signal

$$x(n) = C\delta(n)$$

where C is a constant. We also assume that $y(-1) = 0$. Then the output sequence is

$$y(0) = C, \quad y(1) = C^2, \quad y(2) = C^4, \quad \ldots, \quad y(n) = C^{2^n}$$

Clearly, the output is unbounded when $1 < |C| < \infty$. Therefore, the system is BIBO unstable, since a bounded input sequence has resulted in an unbounded output.

2.2.4 Interconnection of Discrete-Time Systems

Discrete-time systems can be interconnected to form larger systems. There are two basic ways in which systems can be interconnected: in cascade (series) or in parallel. These interconnections are illustrated in Fig. 2.2.10. Note that the two interconnected systems are different.

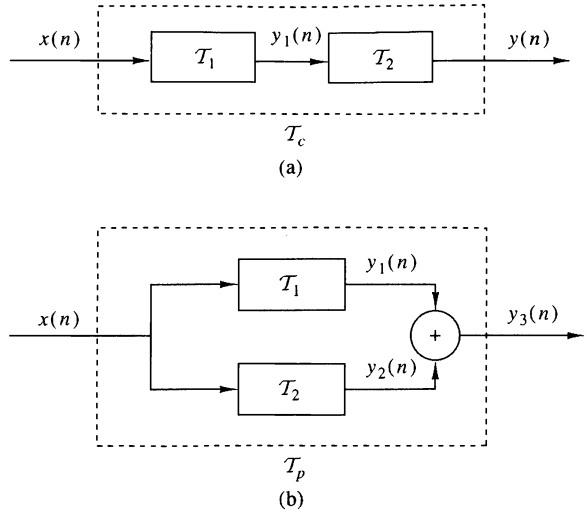


Figure 2.2.10
Cascade (a) and parallel (b) interconnections of systems.

In the cascade interconnection the output of the first system is

$$y_1(n) = \mathcal{T}_1[x(n)] \tag{2.2.46}$$

and the output of the second system is

$$\begin{aligned} y(n) &= \mathcal{T}_2[y_1(n)] \\ &= \mathcal{T}_2\{\mathcal{T}_1[x(n)]\} \end{aligned} \tag{2.2.47}$$

We observe that systems $\mathcal{T}_1$ and $\mathcal{T}_2$ can be combined or consolidated into a single overall system

$$\mathcal{T}_c \equiv \mathcal{T}_2\mathcal{T}_1 \tag{2.2.48}$$

Consequently, we can express the output of the combined system as

$$y(n) = \mathcal{T}_c[x(n)]$$

In general, the order in which the operations $\mathcal{T}_1$ and $\mathcal{T}_2$ are performed is important. That is,

$$\mathcal{T}_2\mathcal{T}_1 \neq \mathcal{T}_1\mathcal{T}_2$$

for arbitrary systems. However, if the systems $\mathcal{T}_1$ and $\mathcal{T}_2$ are linear and time invariant, then (a) $\mathcal{T}_c$ is time invariant and (b) $\mathcal{T}_2\mathcal{T}_1 = \mathcal{T}_1\mathcal{T}_2$, that is, the order in which the systems process the signal is not important. $\mathcal{T}_2\mathcal{T}_1$ and $\mathcal{T}_1\mathcal{T}_2$ yield identical output sequences.

The proof of (a) follows. The proof of (b) is given in Section 2.3.4. To prove time invariance, suppose that $\mathcal{T}_1$ and $\mathcal{T}_2$ are time invariant; then

$$x(n-k) \xrightarrow{\mathcal{T}_1} y_1(n-k)$$

and

$$y_1(n-k) \xrightarrow{\mathcal{T}_2} y(n-k)$$

Thus

$$x(n-k) \xrightarrow{\mathcal{T}_c=\mathcal{T}_2\mathcal{T}_1} y(n-k)$$

and therefore, $\mathcal{T}_c$ is time invariant.

In the parallel interconnection, the output of the system $\mathcal{T}_1$ is $y_1(n)$ and the output of the system $\mathcal{T}_2$ is $y_2(n)$. Hence the output of the parallel interconnection is

$$\begin{aligned} y_3(n) &= y_1(n) + y_2(n) \\ &= \mathcal{T}_1[x(n)] + \mathcal{T}_2[x(n)] \\ &= (\mathcal{T}_1 + \mathcal{T}_2)[x(n)] \\ &= \mathcal{T}_p[x(n)] \end{aligned}$$

where $\mathcal{T}_p = \mathcal{T}_1 + \mathcal{T}_2$.

In general, we can use parallel and cascade interconnection of systems to construct larger, more complex systems. Conversely, we can take a larger system and break it down into smaller subsystems for purposes of analysis and implementation. We shall use these notions later, in the design and implementation of digital filters.

2.3 Analysis of Discrete-Time Linear Time-Invariant Systems

In Section 2.2 we classified systems in accordance with a number of characteristic properties or categories, namely: linearity, causality, stability, and time invariance. Having done so, we now turn our attention to the analysis of the important class of linear, time-invariant (LTI) systems. In particular, we shall demonstrate that such systems are characterized in the time domain simply by their response to a unit sample sequence. We shall also demonstrate that any arbitrary input signal can be decomposed and represented as a weighted sum of unit sample sequences. As a consequence of the linearity and time-invariance properties of the system, the response of the system to any arbitrary input signal can be expressed in terms of the unit sample response of the system. The general form of the expression that relates the unit sample response of the system and the arbitrary input signal to the output signal, called the convolution sum or the convolution formula, is also derived. Thus we are able to determine the output of any linear, time-invariant system to any arbitrary input signal.

2.3.1 Techniques for the Analysis of Linear Systems

There are two basic methods for analyzing the behavior or response of a linear system to a given input signal. One method is based on the direct solution of the input–output equation for the system, which, in general, has the form

$$y(n) = F[y(n-1), y(n-2), \ldots, y(n-N), x(n), x(n-1), \ldots, x(n-M)]$$

where $F[\cdot]$ denotes some function of the quantities in brackets. Specifically, for an LTI system, we shall see later that the general form of the input–output relationship is

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{2.3.1}$$

where $\{a_k\}$ and $\{b_k\}$ are constant parameters that specify the system and are independent of $x(n)$ and $y(n)$. The input–output relationship in (2.3.1) is called a difference equation and represents one way to characterize the behavior of a discrete-time LTI system. The solution of (2.3.1) is the subject of Section 2.4.

The second method for analyzing the behavior of a linear system to a given input signal is first to decompose or resolve the input signal into a sum of elementary signals. The elementary signals are selected so that the response of the system to each signal component is easily determined. Then, using the linearity property of the system, the responses of the system to the elementary signals are added to obtain the total response of the system to the given input signal. This second method is the one described in this section.

To elaborate, suppose that the input signal $x(n)$ is resolved into a weighted sum of elementary signal components $\{x_k(n)\}$ so that

$$x(n) = \sum_{k} c_k x_k(n) \tag{2.3.2}$$

where the $\{c_k\}$ are the set of amplitudes (weighting coefficients) in the decomposition of the signal $x(n)$. Now suppose that the response of the system to the elementary signal component $x_k(n)$ is $y_k(n)$. Thus,

$$y_k(n) \equiv \mathcal{T}[x_k(n)] \tag{2.3.3}$$

assuming that the system is relaxed and that the response to $c_k x_k(n)$ is $c_k y_k(n)$, as a consequence of the scaling property of the linear system.

Finally, the total response to the input $x(n)$ is

$$\begin{aligned} y(n) = \mathcal{T}[x(n)] &= \mathcal{T}\left[\sum_{k} c_k x_k(n)\right] \\ &= \sum_{k} c_k \mathcal{T}[x_k(n)] \\ &= \sum_{k} c_k y_k(n) \end{aligned} \tag{2.3.4}$$

In (2.3.4) we used the additivity property of the linear system.

Although to a large extent, the choice of the elementary signals appears to be arbitrary, our selection is heavily dependent on the class of input signals that we wish to consider. If we place no restriction on the characteristics of the input signals,

their resolution into a weighted sum of unit sample (impulse) sequences proves to be mathematically convenient and completely general. On the other hand, if we restrict our attention to a subclass of input signals, there may be another set of elementary signals that is more convenient mathematically in the determination of the output. For example, if the input signal $x(n)$ is periodic with period N, we have already observed in Section 1.3.3 that a mathematically convenient set of elementary signals is the set of exponentials

$$x_k(n) = e^{j\omega_k n}, \qquad k = 0, 1, \ldots, N-1 \tag{2.3.5}$$

where the frequencies $\{\omega_k\}$ are harmonically related, that is,

$$\omega_k = \left(\frac{2\pi}{N}\right) k, \qquad k = 0, 1, \ldots, N-1 \tag{2.3.6}$$

The frequency $2\pi/N$ is called the fundamental frequency, and all higher-frequency components are multiples of the fundamental frequency component. This subclass of input signals is considered in more detail later.

For the resolution of the input signal into a weighted sum of unit sample sequences, we must first determine the response of the system to a unit sample sequence and then use the scaling and multiplicative properties of the linear system to determine the formula for the output given any arbitrary input. This development is described in detail as follows.

2.3.2 Resolution of a Discrete-Time Signal into Impulses

Suppose we have an arbitrary signal $x(n)$ that we wish to resolve into a sum of unit sample sequences. To utilize the notation established in the preceding section, we select the elementary signals $x_k(n)$ to be

$$x_k(n) = \delta(n-k) \tag{2.3.7}$$

where k represents the delay of the unit sample sequence. To handle an arbitrary signal $x(n)$ that may have nonzero values over an infinite duration, the set of unit impulses must also be infinite, to encompass the infinite number of delays.

Now suppose that we multiply the two sequences $x(n)$ and $\delta(n-k)$. Since $\delta(n-k)$ is zero everywhere except at $n = k$, where its value is unity, the result of this multiplication is another sequence that is zero everywhere except at $n = k$, where its value is $x(k)$, as illustrated in Fig. 2.3.1. Thus

$$x(n)\delta(n-k) = x(k)\delta(n-k) \tag{2.3.8}$$

is a sequence that is zero everywhere except at $n = k$, where its value is $x(k)$. If we repeat the multiplication of $x(n)$ with $\delta(n-m)$, where m is another delay ($m \neq k$), the result will be a sequence that is zero everywhere except at $n = m$, where its value is $x(m)$. Hence

$$x(n)\delta(n-m) = x(m)\delta(n-m) \tag{2.3.9}$$

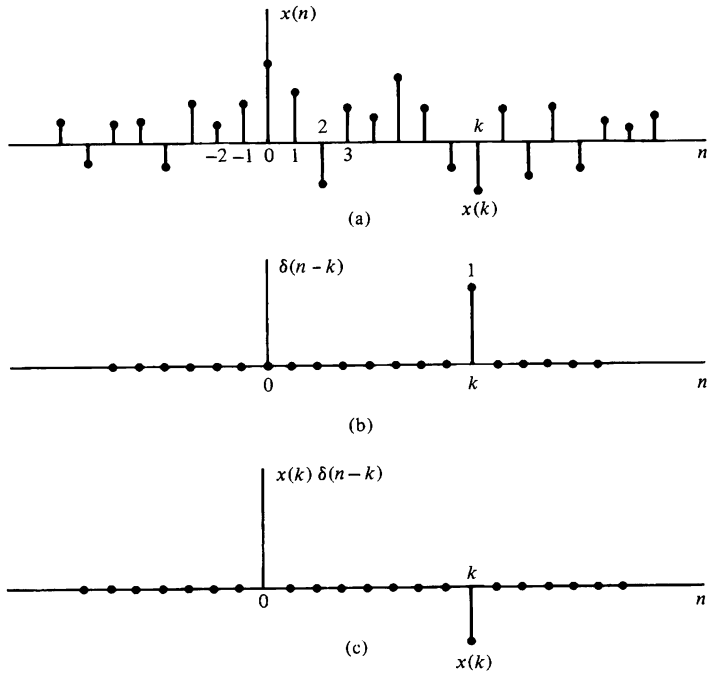


Figure 2.3.1 Multiplication of a signal $x(n)$ with a shifted unit sample sequence.

In other words, each multiplication of the signal $x(n)$ by a unit impulse at some delay k, [i.e., $\delta(n-k)$], in essence picks out the single value $x(k)$ of the signal $x(n)$ at the delay where the unit impulse is nonzero. Consequently, if we repeat this multiplication over all possible delays, $-\infty < k < \infty$, and sum all the product sequences, the result will be a sequence equal to the sequence $x(n)$, that is,

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \tag{2.3.10}$$

We emphasize that the right-hand side of (2.3.10) is the summation of an infinite number of scaled unit sample sequences where the unit sample sequence $\delta(n-k)$ has an amplitude value of $x(k)$. Thus the right-hand side of (2.3.10) gives the resolution or decomposition of any arbitrary signal $x(n)$ into a weighted (scaled) sum of shifted unit sample sequences.

EXAMPLE 2.3.1

Consider the special case of a finite-duration sequence given as

$$x(n) = \{2, \underset{\uparrow}{4}, 0, 3\}$$

Resolve the sequence $x(n)$ into a sum of weighted impulse sequences.

Solution. Since the sequence $x(n)$ is nonzero for the time instants $n = -1, 0, 2$, we need three impulses at delays $k = -1, 0,$ Following (2.3.10) we find that

$$x(n) = 2\delta(n+1) + 4\delta(n) + 3\delta(n-2)$$

2.3.3 Response of LTI Systems to Arbitrary Inputs: The Convolution Sum

Having resolved an arbitrary input signal $x(n)$ into a weighted sum of impulses, we are now ready to determine the response of any relaxed linear system to any input signal. First, we denote the response $y(n, k)$ of the system to the input unit sample sequence at $n = k$ by the special symbol $h(n, k)$, $-\infty < k < \infty$. That is,

$$y(n, k) \equiv h(n, k) = \mathcal{T}[\delta(n-k)] \tag{2.3.11}$$

In (2.3.11) we note that n is the time index and k is a parameter showing the location of the input impulse. If the impulse at the input is scaled by an amount $c_k \equiv x(k)$, the response of the system is the correspondingly scaled output, that is,

$$c_k h(n, k) = x(k)h(n, k) \tag{2.3.12}$$

Finally, if the input is the arbitrary signal $x(n)$ that is expressed as a sum of weighted impulses, that is,

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \tag{2.3.13}$$

then the response of the system to $x(n)$ is the corresponding sum of weighted outputs, that is,

$$\begin{aligned} y(n) = \mathcal{T}[x(n)] &= \mathcal{T}\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right] \\ &= \sum_{k=-\infty}^{\infty} x(k)\mathcal{T}[\delta(n-k)] \\ &= \sum_{k=-\infty}^{\infty} x(k)h(n, k) \end{aligned} \tag{2.3.14}$$

Clearly, (2.3.14) follows from the superposition property of linear systems, and is known as the *superposition summation.*

We note that (2.3.14) is an expression for the response of a linear system to any arbitrary input sequence $x(n)$. This expression is a function of both $x(n)$ and the responses $h(n, k)$ of the system to the unit impulses $\delta(n-k)$ for $-\infty < k < \infty$. In deriving (2.3.14) we used the linearity property of the system but not its time-invariance property. Thus the expression in (2.3.14) applies to any relaxed linear (time-variant) system.

If, in addition, the system is time invariant, the formula in (2.3.14) simplifies considerably. In fact, if the response of the LTI system to the unit sample sequence $\delta(n)$ is denoted as $h(n)$, that is,

$$h(n) \equiv \mathcal{T}[\delta(n)] \tag{2.3.15}$$

then by the time-invariance property, the response of the system to the delayed unit sample sequence $\delta(n-k)$ is

$$h(n-k) = \mathcal{T}[\delta(n-k)] \tag{2.3.16}$$

Consequently, the formula in (2.3.14) reduces to

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.3.17}$$

Now we observe that the relaxed LTI system is completely characterized by a single function $h(n)$, namely, its response to the unit sample sequence $\delta(n)$. In contrast, the general characterization of the output of a time-variant, linear system requires an infinite number of unit sample response functions, $h(n, k)$, one for each possible delay.

The formula in (2.3.17) that gives the response $y(n)$ of the LTI system as a function of the input signal $x(n)$ and the unit sample (impulse) response $h(n)$ is called a *convolution sum*. We say that the input $x(n)$ is convolved with the impulse response $h(n)$ to yield the output $y(n)$. We shall now explain the procedure for computing the response $y(n)$, both mathematically and graphically, given the input $x(n)$ and the impulse response $h(n)$ of the system.

Suppose that we wish to compute the output of the system at some time instant, say $n = n_0$. According to (2.3.17), the response at $n = n_0$ is given as

$$y(n_0) = \sum_{k=-\infty}^{\infty} x(k)h(n_0-k) \tag{2.3.18}$$

Our first observation is that the index in the summation is k, and hence both the input signal $x(k)$ and the impulse response $h(n_0-k)$ are functions of k. Second, we observe that the sequences $x(k)$ and $h(n_0-k)$ are multiplied together to form a product sequence. The output $y(n_0)$ is simply the sum over all values of the product sequence. The sequence $h(n_0-k)$ is obtained from $h(k)$ by, first, folding $h(k)$ about $k=0$ (the time origin), which results in the sequence $h(-k)$. The folded sequence is then shifted by n_0 to yield $h(n_0-k)$. To summarize, the process of computing the convolution between $x(k)$ and $h(k)$ involves the following four steps.

1. *Folding.* Fold $h(k)$ about $k=0$ to obtain $h(-k)$.
2. *Shifting.* Shift $h(-k)$ by n_0 to the right (left) if n_0 is positive (negative), to obtain $h(n_0-k)$.
3. *Multiplication.* Multiply $x(k)$ by $h(n_0-k)$ to obtain the product sequence $v_{n_0}(k) \equiv x(k)h(n_0-k)$.
4. *Summation.* Sum all the values of the product sequence $v_{n_0}(k)$ to obtain the value of the output at time $n = n_0$.

We note that this procedure results in the response of the system at a single time instant, say $n = n_0$. In general, we are interested in evaluating the response of the system over all time instants $-\infty < n < \infty$. Consequently, steps 2 through 4 in the summary must be repeated, for all possible time shifts $-\infty < n < \infty$.

In order to gain a better understanding of the procedure for evaluating the convolution sum, we shall demonstrate the process graphically. The graphs will aid us in explaining the four steps involved in the computation of the convolution sum.

EXAMPLE 2.3.2

The impulse response of a linear time-invariant system is

$$h(n) = \{1, \underset{\uparrow}{2}, 1, -1\} \tag{2.3.19}$$

Determine the response of the system to the input signal

$$x(n) = \{\underset{\uparrow}{1}, 2, 3, 1\} \tag{2.3.20}$$

Solution. We shall compute the convolution according to the formula (2.3.17), but we shall use graphs of the sequences to aid us in the computation. In Fig. 2.3.2(a) we illustrate the input signal sequence $x(k)$ and the impulse response $h(k)$ of the system, using k as the time index in order to be consistent with (2.3.17).

The first step in the computation of the convolution sum is to fold $h(k)$. The folded sequence $h(-k)$ is illustrated in Fig. 2.3.2(b). Now we can compute the output at $n = 0$, according to (2.3.17), which is

$$y(0) = \sum_{k=-\infty}^{\infty} x(k)h(-k) \tag{2.3.21}$$

Since the shift $n = 0$, we use $h(-k)$ directly without shifting it. The product sequence

$$v_0(k) \equiv x(k)h(-k) \tag{2.3.22}$$

is also shown in Fig. 2.3.2(b). Finally, the sum of all the terms in the product sequence yields

$$y(0) = \sum^{\infty} h = -\infty v_0(k) = 4$$

We continue the computation by evaluating the response of the system at $n = 1$. According to (2.3.17),

$$y(1) = \sum_{h=-\infty}^{\infty} x(k)h(1-k) \tag{2.3.23}$$

The sequence $h(1-k)$ is simply the folded sequence $h(-k)$ shifted to the right by one unit in time. This sequence is illustrated in Fig. 2.3.2(c). The product sequence

$$v_1(k) = x(k)h(1-k) \tag{2.3.24}$$

is also illustrated in Fig. 2.3.2(c). Finally, the sum of all the values in the product sequence yields

$$y(1) = \sum_{k=-\infty}^{\infty} v_1(k) = 8$$

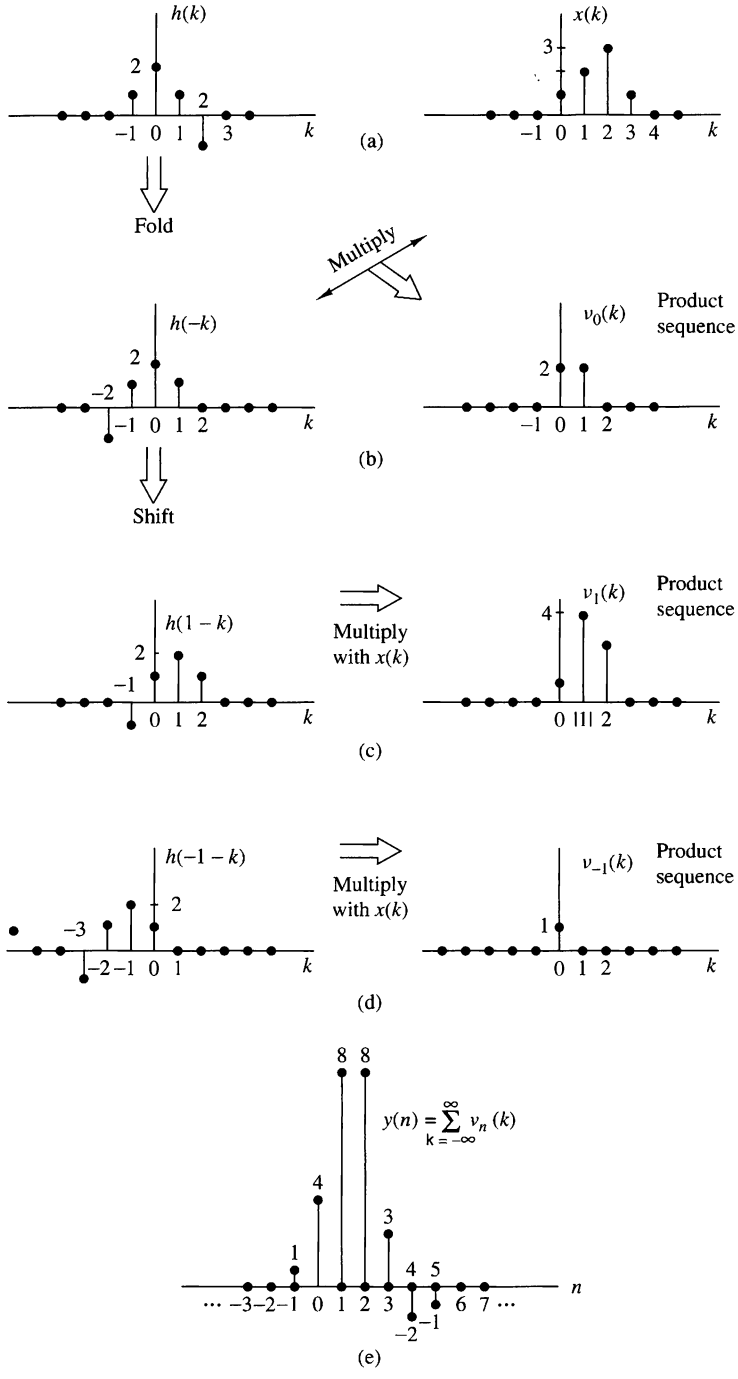


Figure 2.3.2 Graphical computation of convolution.

In a similar manner, we obtain $y(2)$ by shifting $h(-k)$ two units to the right, forming the product sequence $v_2(k) = x(k)h(2-k)$ and then summing all the terms in the product sequence obtaining $y(2) = 8$. By shifting $h(-k)$ farther to the right, multiplying the corresponding sequence, and summing over all the values of the resulting product sequences, we obtain $y(3) = 3$, $y(4) = -2$, $y(5) = -1$. For $n > 5$, we find that $y(n) = 0$ because the product sequences contain all zeros. Thus we have obtained the response $y(n)$ for $n > 0$.

Next we wish to evaluate $y(n)$ for $n < 0$. We begin with $n = -1$. Then

$$y(-1) = \sum_{k=-\infty}^{\infty} x(k)h(-1-k) \tag{2.3.25}$$

Now the sequence $h(-1-k)$ is simply the folded sequence $h(-k)$ shifted one time unit to the left. The resulting sequence is illustrated in Fig. 2.3.2(d). The corresponding product sequence is also shown in Fig. 2.3.2(d). Finally, summing over the values of the product sequence, we obtain

$$y(-1) = 1$$

From observation of the graphs of Fig. 2.3.2, it is clear that any further shifts of $h(-1-k)$ to the left always result in an all-zero product sequence, and hence

$$y(n) = 0 \qquad \text{for } n \leq -2$$

Now we have the entire response of the system for $-\infty < n < \infty$, which we summarize below as

$$y(n) = \{\ldots, 0, 0, 1, \underset{\uparrow}{4}, 8, 8, 3, -2, -1, 0, 0, \ldots\} \tag{2.3.26}$$

In Example 2.3.2 we illustrated the computation of the convolution sum, using graphs of the sequences to aid us in visualizing the steps involved in the computation procedure.

Before working out another example, we wish to show that the convolution operation is commutative in the sense that it is irrelevant which of the two sequences is folded and shifted. Indeed, if we begin with (2.3.17) and make a change in the variable of the summation, from k to m, by defining a new index $m = n - k$, then $k = n - m$ and (2.3.17) becomes

$$y(n) = \sum_{m=-\infty}^{\infty} x(n-m)h(m) \tag{2.3.27}$$

Since m is a dummy index, we may simply replace m by k so that

$$y(n) = \sum_{k=-\infty}^{\infty} x(n-k)h(k) \tag{2.3.28}$$

The expression in (2.3.28) involves leaving the impulse response $h(k)$ unaltered, while the input sequence is folded and shifted. Although the output $y(n)$ in (2.3.28)

is identical to (2.3.17), the product sequences in the two forms of the convolution formula are not identical. In fact, if we define the two product sequences as

$$v_n(k) = x(k)h(n-k)$$

$$w_n(k) = x(n-k)h(k)$$

it can be easily shown that

$$v_n(k) = w_n(n-k)$$

and therefore,

$$y(n) = \sum_{k=-\infty}^{\infty} v_n(k) = \sum_{k=-\infty}^{\infty} w_n(n-k)$$

since both sequences contain the same sample values in a different arrangement. The reader is encouraged to rework Example 2.3.2 using the convolution sum in (2.3.28).

EXAMPLE 2.3.3

Determine the output $y(n)$ of a relaxed linear time-invariant system with impulse response

$$h(n) = a^{nu}(n), \; |a|1$$

when the input is a unit step sequence, that is,

$$x(n) = u(n)$$

Solution. In this case both $h(n)$ and $x(n)$ are infinite-duration sequences. We use the form of the convolution formula given by (2.3.28) in which $x(k)$ is folded. The sequences $h(k)$, $x(k)$, and $x(-k)$ are shown in Fig. 2.3.3. The product sequences $v_0(k)$, $v_1(k)$, and $v_2(k)$ corresponding to $x(-k)h(k)$, $x(1-k)h(k)$, and $x(2-k)h(k)$ are illustrated in Fig. 2.3.3(c), (d), and (e), respectively. Thus we obtain the outputs

$$y(0) = 1$$

$$y(1) = 1 + a$$

$$y(2) = 1 + a + a^2$$

Clearly, for $n > 0$, the output is

$$\begin{aligned} y(n) &= 1 + a + a^2 + \cdots + a^n \\ &= \frac{1 - a^{n+1}}{1 - a} \end{aligned} \tag{2.3.29}$$

On the other hand, for $n < 0$, the product sequences consist of all zeros. Hence

$$y(n) = 0, \qquad n < 0$$

A graph of the output $y(n)$ is illustrated in Fig. 2.3.3(f) for the case $0 < a < 1$. Note the exponential rise in the output as a function of n. Since $|a| < 1$, the final value of the output as n approaches infinity is

$$y(\infty) = \lim_{n \to \infty} y(n) = \frac{1}{1 - a} \tag{2.3.30}$$

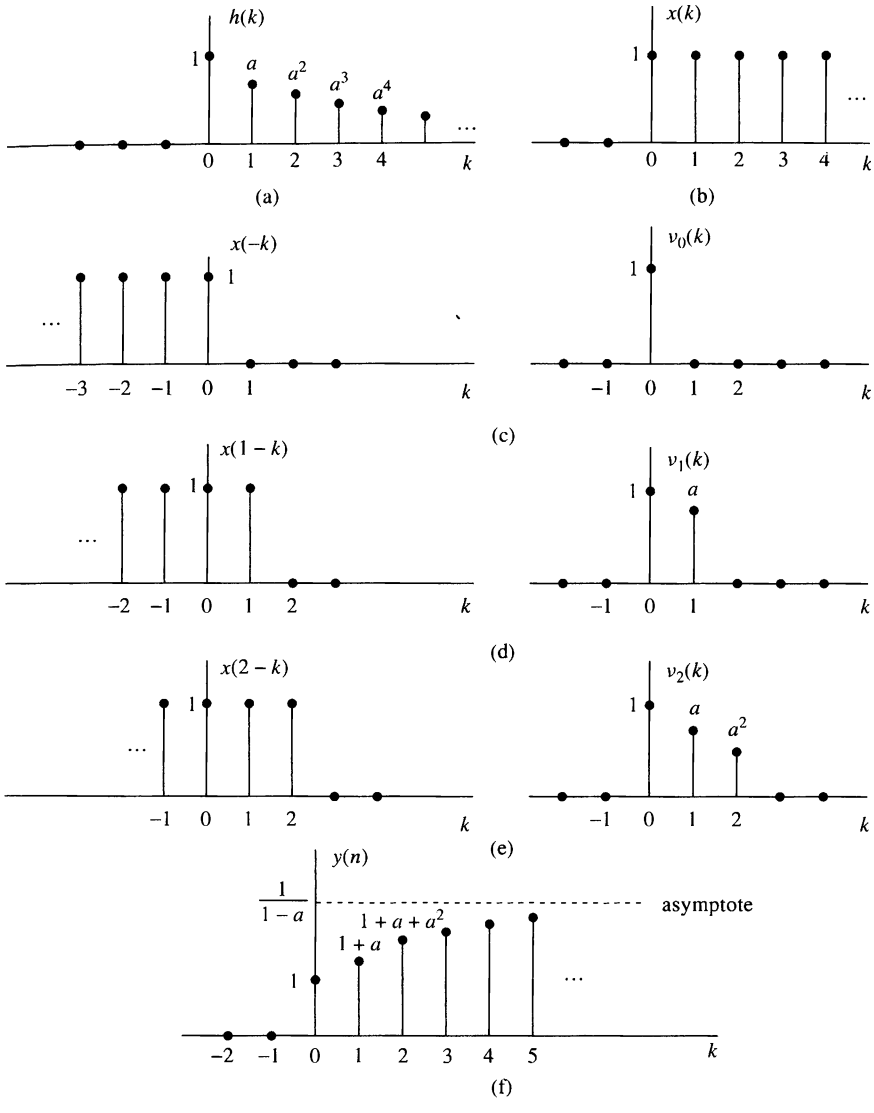


Figure 2.3.3 Graphical computation of convolution in Example 2.3.3.

To summarize, the convolution formula provides us with a means for computing the response of a relaxed, linear time-invariant system to any arbitrary input signal $x(n)$. It takes one of two equivalent forms, either (2.3.17) or (2.3.28), where $x(n)$ is the input signal to the system, $h(n)$ is the impulse response of the system, and $y(n)$ is the *output* of the system in response to the input signal $x(n)$. The evaluation of the convolution formula involves four operations, namely: *folding* either the impulse response as specified by (2.3.17) or the input sequence as specified by (2.3.28) to yield either $h(-k)$ or $x(-k)$, respectively, *shifting* the folded sequence by n units in time to yield either $h(n-k)$ or $x(n-k)$, *multiplying* the two sequences to yield the product

sequence, either $x(k)h(n-k)$ or $x(n-k)h(k)$, and finally *summing* all the values in the product sequence to yield the output $y(n)$ of the system at time n. The folding operation is done only once. However, the other three operations are repeated for all possible shifts $-\infty < n < \infty$ in order to obtain $y(n)$ for $-\infty < n < \infty$.

2.3.4 Properties of Convolution and the Interconnection of LTI Systems

In this section we investigate some important properties of convolution and interpret these properties in terms of interconnecting linear time-invariant systems. We should stress that these properties hold for every input signal.

It is convenient to simplify the notation by using an asterisk to denote the convolution operation. Thus

$$y(n) = x(n) * h(n) \equiv \sum_{k=-\infty}^{\infty} x(k)h(n-k) \tag{2.3.31}$$

In this notation the sequence following the asterisk [i.e., the impulse response $h(n)$] is folded and shifted. The input to the system is $x(n)$. On the other hand, we also showed that

$$y(n) = h(n) * x(n) \equiv \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{2.3.32}$$

In this form of the convolution formula, it is the input signal that is folded. Alternatively, we may interpret this form of the convolution formula as resulting from an interchange of the roles of $x(n)$ and $h(n)$. In other words, we may regard $x(n)$ as the impulse response of the system and $h(n)$ as the excitation or input signal. Figure 2.3.4 illustrates this interpretation.

Identity and Shifting Properties. We also note that the unit sample sequence $\delta(n)$ is the identity element for convolution, that is

$$y(n) = x(n) * \delta(n) = x(n)$$

If we shift $\delta(n)$ by k, the convolution sequence is shifted also by k, that is

$$x(n) * \delta(n-k) = y(n-k) = x(n-k)$$

We can view convolution more abstractly as a mathematical operation between two signal sequences, say $x(n)$ and $h(n)$, that satisfies a number of properties. The property embodied in (2.3.31) and (2.3.32) is called the commutative law.

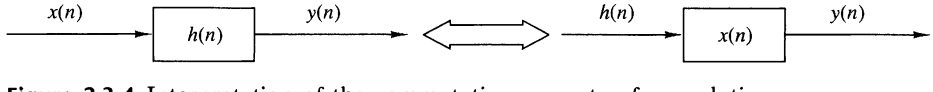


Figure 2.3.4 Interpretation of the commutative property of convolution.

Commutative law

$$x(n) * h(n) = h(n) * x(n) \tag{2.3.33}$$

Viewed mathematically, the convolution operation also satisfies the associative law, which can be stated as follows.

Associative law

$$[x(n) * h_1(n)] * h_2(n) = x(n) * [h_1(n) * h_2(n)] \tag{2.3.34}$$

From a physical point of view, we can interpret $x(n)$ as the input signal to a linear time-invariant system with impulse response $h_1(n)$. The output of this system, denoted as $y_1(n)$, becomes the input to a second linear time-invariant system with impulse response $h_2(n)$. Then the output is

$$\begin{aligned} y(n) &= y_1(n) * h_2(n) \\ &= [x(n) * h_1(n)] * h_2(n) \end{aligned}$$

which is precisely the left-hand side of (2.3.34). Thus the left-hand side of (2.3.34) corresponds to having two linear time-invariant systems in cascade. Now the right-hand side of (2.3.34) indicates that the input $x(n)$ is applied to an equivalent system having an impulse response, say $h(n)$, which is equal to the convolution of the two impulse responses. That is,

$$h(n) = h_1(n) * h_2(n)$$

and

$$y(n) = x(n) * h(n)$$

Furthermore, since the convolution operation satisfies the commutative property, one can interchange the order of the two systems with responses $h_1(n)$ and $h_2(n)$ without altering the overall input–output relationship. Figure 2.3.5 graphically illustrates the associative property.

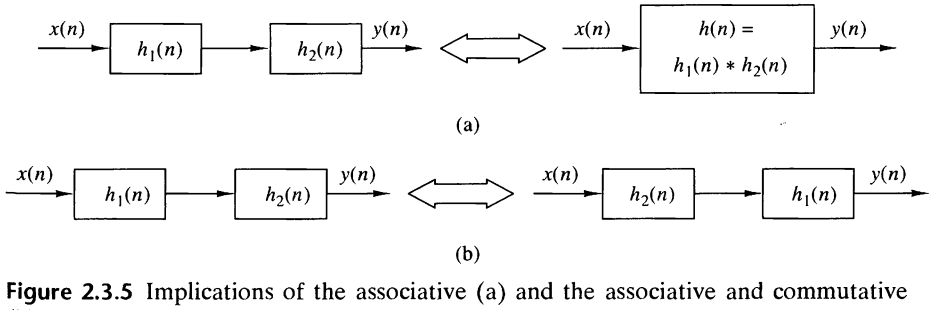


Figure 2.3.5 Implications of the associative (a) and the associative and commutative (b) properties of convolution.

EXAMPLE 2.3.4

Determine the impulse response for the cascade of two linear time-invariant systems having impulse responses

$$h_1(n) = (\frac{1}{2})^n u(n)$$

and

$$h_2(n) = (\frac{1}{4})^n u(n)$$

Solution. To determine the overall impulse response of the two systems in cascade, we simply convolve $h_1(n)$ with $h_2(n)$. Hence

$$h(n) = \sum_{k=-\infty}^{\infty} h_1(k)h_2(n-k)$$

where $h_2(n)$ is folded and shifted. We define the product sequence

$$v_n(k) = h_1(k)h_2(n-k)$$

$$= (\frac{1}{2})^k(\frac{1}{4})^{n-k}$$

which is nonzero for $k \geq 0$ and $n - k \geq 0$ or $n \geq k \geq 0$. On the other hand, for $n < 0$, we have $v_n(k) = 0$ for all k, and hence

$$h(n) = 0, n < 0$$

For $n \geq k \geq 0$, the sum of the values of the product sequence $v_n(k)$ over all k yields

$$h(n) = \sum_{k=0}^{n}(\frac{1}{2})^k(\frac{1}{4})^{n-k}$$

$$= (\frac{1}{4})^n \sum_{k=0}^{n} 2^k$$

$$= (\frac{1}{4})^n(2^{n+1} - 1)$$

$$= (\frac{1}{2})^n[2 - (\frac{1}{2})^n], \quad n \geq 0$$

The generalization of the associative law to more than two systems in cascade follows easily from the discussion given above. Thus if we have L linear time-invariant systems in cascade with impulse responses $h_1(n), h_2(n), \ldots, h_L(n)$, there is an equivalent linear time-invariant system having an impulse response that is equal to the $(L-1)$-fold convolution of the impulse responses. That is,

$$h(n) = h_1(n) * h_2(n) * \cdots * h_L(n) \tag{2.3.35}$$

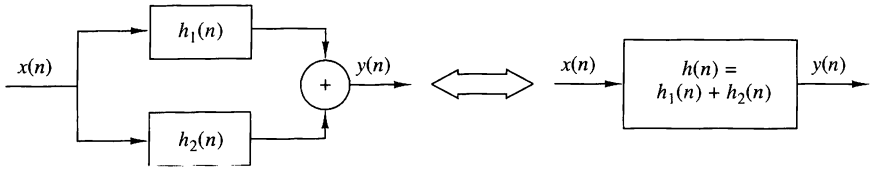


Figure 2.3.6 Interpretation of the distributive property of convolution: two LTI systems connected in parallel can be replaced by a single system with $h(n) = h_1(n) + h_2(n)$.

The commutative law implies that the order in which the convolutions are performed is immaterial. Conversely, any linear time-invariant system can be decomposed into a cascade interconnection of subsystems. A method for accomplishing the decomposition will be described later.

Another property that is satisfied by the convolution operation is the distributive law, which may be stated as follows.

Distributive law

$$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n) \tag{2.3.36}$$

Interpreted physically, this law implies that if we have two linear time-invariant systems with impulse responses $h_1(n)$ and $h_2(n)$ excited by the same input signal $x(n)$, the sum of the two responses is identical to the response of an overall system with impulse response

$$h(n) = h_1(n) + h_2(n)$$

Thus the overall system is viewed as a parallel combination of the two linear time-invariant systems as illustrated in Fig. 2.3.6.

The generalization of (2.3.36) to more than two linear time-invariant systems in parallel follows easily by mathematical induction. Thus the interconnection of L linear time-invariant systems in parallel with impulse responses $h_1(n), h_2(n), \ldots, h_L(n)$ and excited by the same input $x(n)$ is equivalent to one overall system with impulse response

$$h(n) = \sum_{j=1}^{L} h_j(n) \tag{2.3.37}$$

Conversely, any linear time-invariant system can be decomposed into a parallel interconnection of subsystems.

2.3.5 Causal Linear Time-Invariant Systems

In Section 2.2.3 we defined a causal system as one whose output at time n depends only on present and past inputs but does not depend on future inputs. In other words, the output of the system at some time instant n, say $n = n_0$, depends only on values of $x(n)$ for $n \leq n_0$.

In the case of a linear time-invariant system, causality can be translated to a condition on the impulse response. To determine this relationship, let us consider a

linear time-invariant system having an output at time $n = n_0$ given by the convolution formula

$$y(n_0) = \sum_{k=-\infty}^{\infty} h(k)x(n_0 - k)$$

Suppose that we subdivide the sum into two sets of terms, one set involving present and past values of the input [i.e., $x(n)$ for $n \leq n_0$] and one set involving future values of the input [i.e., $x(n)$, $n > n_0$]. Thus we obtain

$$\begin{aligned} y(n_0) &= \sum_{k=0}^{\infty} h(k)x(n_0 - k) + \sum_{k=-\infty}^{-1} h(k)x(n_0 - k) \\ &= [h(0)x(n_0) + h(1)x(n_0 - 1) + h(2)x(n_0 - 2) + \cdots] \\ &\quad + [h(-1)x(n_0 + 1) + h(-2)x(n_0 + 2) + \cdots] \end{aligned}$$

We observe that the terms in the first sum involve $x(n_0)$, $x(n_0 - 1), \ldots$, which are the present and past values of the input signal. On the other hand, the terms in the second sum involve the input signal components $x(n_0+1), x(n_0+2), \ldots$. Now, if the output at time $n = n_0$ is to depend only on the present and past inputs, then, clearly, the impulse response of the system must satisfy the condition

$$h(n) = 0, \qquad n < 0 \tag{2.3.38}$$

Since $h(n)$ is the response of the relaxed linear time-invariant system to a unit impulse applied at $n = 0$, it follows that $h(n) = 0$ for $n < 0$ is both a necessary and a sufficient condition for causality. Hence an *LTI system is causal if and only if its impulse response is zero for negative values of n*.

Since for a causal system, $h(n) = 0$ for $n < 0$, the limits on the summation of the convolution formula may be modified to reflect this restriction. Thus we have the two equivalent forms

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n - k) \tag{2.3.39}$$

$$= \sum_{k=-\infty}^{n} x(k)h(n - k) \tag{2.3.40}$$

As indicated previously, causality is required in any real-time signal processing application, since at any given time n we have no access to future values of the input signal. Only the present and past values of the input signal are available in computing the present output.

It is sometimes convenient to call a sequence that is zero for $n < 0$, a *causal sequence*, and one that is nonzero for $n < 0$ and $n > 0$, a *noncausal sequence*. This terminology means that such a sequence could be the unit sample response of a causal or a noncausal system, respectively.

If the input to a causal linear time-invariant system is a causal sequence [i.e., if $x(n) = 0$ for $n < 0$], the limits on the convolution formula are further restricted. In this case the two equivalent forms of the convolution formula become

$$y(n) = \sum_{k=0}^{n} h(k)x(n-k) \tag{2.3.41}$$

$$= \sum_{k=0}^{n} x(k)h(n-k) \tag{2.3.42}$$

We observe that in this case, the limits on the summations for the two alternative forms are identical, and the upper limit is growing with time. Clearly, the response of a causal system to a causal input sequence is causal, since $y(n) = 0$ for $n < 0$.

EXAMPLE 2.3.5

Determine the unit step response of the linear time-invariant system with impulse response

$$h(n) = a^n u(n), \qquad |a| < 1$$

Solution. Since the input signal is a unit step, which is a causal signal, and the system is also causal, we can use one of the special forms of the convolution formula, either (2.3.41) or (2.3.42). Since $x(n) = 1$ for $n \geq 0$, (2.3.41) is simpler to use. Because of the simplicity of this problem, one can skip the steps involved with sketching the folded and shifted sequences. Instead, we use direct substitution of the signals sequences in (2.3.41) and obtain

$$y(n) = \sum_{k=0}^{n} a^k$$

$$= \frac{1 - a^{n+1}}{1 - a}$$

and $y(n) = 0$ for $n < 0$. We note that this result is identical to that obtained in Example 2.3.3. In this simple case, however, we computed the convolution algebraically without resorting to the detailed procedure outlined previously.

2.3.6 Stability of Linear Time-Invariant Systems

As indicated previously, stability is an important property that must be considered in any practical implementation of a system. We defined an arbitrary relaxed system as BIBO stable if and only if its output sequence $y(n)$ is bounded for every bounded input $x(n)$.

If $x(n)$ is bounded, there exists a constant M_x such that

$$|x(n)| \leq M_x < \infty$$

Similarly, if the output is bounded, there exists a constant M_y such that

$$|y(n)| < M_y < \infty$$

for all n.

Now, given such a bounded input sequence $x(n)$ to a linear time-invariant system, let us investigate the implications of the definition of stability on the characteristics of the system. Toward this end, we work again with the convolution formula

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

If we take the absolute value of both sides of this equation, we obtain

$$|y(n)| = \left| \sum_{k=-\infty}^{\infty} h(k)x(n-k) \right|$$

Now, the absolute value of the sum of terms is always less than or equal to the sum of the absolute values of the terms. Hence

$$|y(n)| \leq \sum_{k=-\infty}^{\infty} |h(k)||x(n-k)|$$

If the input is bounded, there exists a finite number M_x such that $|x(n)| \leq M_x$. By substituting this upper bound for $x(n)$ in the equation above, we obtain

$$|y(n)| \leq M_x \sum_{k=-\infty}^{\infty} |h(k)|$$

From this expression we observe that the output is bounded if the impulse response of the system satisfies the condition

$$S_h \equiv \sum_{k=-\infty}^{\infty} |h(k)| < \infty \tag{2.3.43}$$

That is, *a linear time-invariant system is stable if its impulse response is absolutely summable.* This condition is not only sufficient but it is also necessary to ensure the stability of the system. Indeed, we shall show that if $S_h = \infty$, there is a bounded input for which the output is not bounded. We choose the bounded input

$$x(n) = \begin{cases} \dfrac{h^*(-n)}{|h(-n)|}, & h(n) \neq 0 \\ 0, & h(n) = 0 \end{cases}$$

where $h^*(n)$ is the complex conjugate of $h(n)$. It is sufficient to show that there is one value of n for which $y(n)$ is unbounded. For $n = 0$ we have

$$y(0) = \sum_{k=-\infty}^{\infty} x(-k)h(k) = \sum_{k=-\infty}^{\infty} \frac{|h(k)|^2}{|h(k)|} = S_h$$

Thus, if $S_h = \infty$, a bounded input produces an unbounded output since $y(0) = \infty$.

The condition in (2.3.43) implies that the impulse response $h(n)$ goes to zero as n approaches infinity. As a consequence, the output of the system goes to zero as n approaches infinity if the input is set to zero beyond $n > n_0$. To prove this, suppose that $|x(n)| < M_x$ for $n < n_0$ and $x(n) = 0$ for $n \geq n_0$. Then, at $n = n_0 + N$, the system output is

$$y(n_0 + N) = \sum_{k=-\infty}^{N-1} h(k)x(n_0 + N - k) + \sum_{k=N}^{\infty} h(k)x(n_0 + N - k)$$

But the first sum is zero since $x(n) = 0$ for $n \geq n_0$. For the remaining part, we take the absolute value of the output, which is

$$|y(n_0 + N)| = \left| \sum_{k=N}^{\infty} h(k)x(n_0 + N - k) \right| \leq \sum_{k=N}^{\infty} |h(k)||x(n_0 + N - k)|$$

$$\leq M_x \sum_{k=N}^{\infty} |h(k)|$$

Now, as N approaches infinity,

$$\lim_{N\to\infty} \sum_{k=N}^{\infty} |h(n)| = 0$$

and hence

$$\lim_{N\to\infty} |y(n_0 + N)| = 0$$

This result implies that any excitation at the input to the system, which is of a finite duration, produces an output that is "transient" in nature; that is, its amplitude decays with time and dies out eventually, when the system is stable.

EXAMPLE 2.3.6

Determine the range of values of the parameter a for which the linear time-invariant system with impulse response

$$h(n) = a^n u(n)$$

is stable.

Solution. First, we note that the system is causal. Consequently, the lower index on the summation in (2.3.43) begins with $k = 0$. Hence

$$\sum_{k=0}^{\infty} |a^k| = \sum_{k=0}^{\infty} |a|^k = 1 + |a| + |a|^2 + \cdots$$

Clearly, this geometric series converges to

$$\sum_{k=0}^{\infty} |a|^k = \frac{1}{1 - |a|}$$

provided that $|a| < 1$. Otherwise, it diverges. Therefore, the system is stable if $|a| < 1$. Otherwise, it is unstable. In effect, $h(n)$ must decay exponentially toward zero as n approaches infinity for the system to be stable.

EXAMPLE 2.3.7

Determine the range of values of a and b for which the linear time-invariant system with impulse response

$$h(n) = \begin{cases} a^n, & n \geq 0 \\ b^n, & n < 0 \end{cases}$$

is stable.

Solution. This system is noncasual. The condition on stability given by (2.3.43) yields

$$\sum_{n=-\infty}^{\infty} |h(n)| = \sum_{n=0}^{\infty} |a|^n + \sum_{n=-\infty}^{-1} |b|^n$$

From Example 2.3.6 we have already determined that the first sum converges for $|a| < 1$. The second sum can be manipulated as follows:

$$\sum_{n=-\infty}^{-1} |b|^n = \sum_{n=1}^{\infty} \frac{1}{|b|^n} = \frac{1}{|b|}\left(1 + \frac{1}{|b|} + \frac{1}{|b|^2} + \cdots\right)$$

$$= \beta(1 + \beta + \beta^2 + \cdots) = \frac{\beta}{1-\beta}$$

where $\beta = 1/|b|$ must be less than unity for the geometric series to converge. Consequently, the system is stable if both $|a| < 1$ and $|b| > 1$ are satisfied.

2.3.7 Systems with Finite-Duration and Infinite-Duration Impulse Response

Up to this point we have characterized a linear time-invariant system in terms of its impulse response $h(n)$. It is also convenient, however, to subdivide the class of linear time-invariant systems into two types, those that have a finite-duration impulse response (FIR) and those that have an infinite-duration impulse response (IIR). Thus an FIR system has an impulse response that is zero outside of some finite time interval. Without loss of generality, we focus our attention on causal FIR systems, so that

$$h(n) = 0, \qquad n < 0 \text{ and } n \geq M$$

The convolution formula for such a system reduces to

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

A useful interpretation of this expression is obtained by observing that the output at any time n is simply a weighted linear combination of the input signal samples $x(n)$, $x(n-1), \ldots, x(n-M+1)$. In other words, the system simply weights, by the values of the impulse response $h(k)$, $k = 0, 1, \ldots, M-1$, the most recent M signal samples

and sums the resulting M products. In effect, the system acts as a *window* that views only the most recent M input signal samples in forming the output. It neglects or simply "forgets" all prior input samples [i.e., $x(n-M)$, $x(n-M-1), \ldots$]. Thus we say that an FIR system has a finite memory of length-M samples.

In contrast, an IIR linear time-invariant system has an infinite-duration impulse response. Its output, based on the convolution formula, is

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k)$$

where causality has been assumed, although this assumption is not necessary. Now, the system output is a weighted [by the impulse response $h(k)$] linear combination of the input signal samples $x(n)$, $x(n-1)$, $x(n-2), \ldots$. Since this weighted sum involves the present and all the past input samples, we say that the system has an infinite memory.

We investigate the characteristics of FIR and IIR systems in more detail in subsequent chapters.

2.4 Discrete-Time Systems Described by Difference Equations

Up to this point we have treated linear and time-invariant systems that are characterized by their unit sample response $h(n)$. In turn, $h(n)$ allows us to determine the output $y(n)$ of the system for any given input sequence $x(n)$ by means of the convolution summation,

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{2.4.1}$$

In general, then, we have shown that any linear time-invariant system is characterized by the input–output relationship in (2.4.1). Moreover, the convolution summation formula in (2.4.1) suggests a means for the realization of the system. In the case of FIR systems, such a realization involves additions, multiplications, and a finite number of memory locations. Consequently, an FIR system is readily implemented directly, as implied by the convolution summation.

If the system is IIR, however, its practical implementation as implied by convolution is clearly impossible, since it requires an infinite number of memory locations, multiplications, and additions. A question that naturally arises, then, is whether or not it is possible to realize IIR systems other than in the form suggested by the convolution summation. Fortunately, the answer is yes, there is a practical and computationally efficient means for implementing a family of IIR systems, as will be demonstrated in this section. Within the general class of IIR systems, this family of discrete-time systems is more conveniently described by difference equations. This family or subclass of IIR systems is very useful in a variety of practical applications, including the implementation of digital filters, and the modeling of physical phenomena and physical systems.

2.4.1 Recursive and Nonrecursive Discrete-Time Systems

As indicated above, the convolution summation formula expresses the output of the linear time-invariant system explicitly and only in terms of the input signal. However, this need not be the case, as is shown here. There are many systems where it is either necessary or desirable to express the output of the system not only in terms of the present and past values of the input, but also in terms of the already available past output values. The following problem illustrates this point.

Suppose that we wish to compute the *cumulative average* of a signal $x(n)$ in the interval $0 \leq k \leq n$, defined as

$$y(n) = \frac{1}{n+1} \sum_{k=0}^{n} x(k), \qquad n = 0, 1, \ldots \tag{2.4.2}$$

As implied by (2.4.2), the computation of $y(n)$ requires the storage of all the input samples $x(k)$ for $0 \leq k \leq n$. Since n is increasing, our memory requirements grow linearly with time.

Our intuition suggests, however, that $y(n)$ can be computed more efficiently by utilizing the previous output value $y(n-1)$. Indeed, by a simple algebraic rearrangement of (2.4.2), we obtain

$$\begin{aligned}(n+1)y(n) &= \sum_{k=0}^{n-1} x(k) + x(n) \\ &= ny(n-1) + x(n)\end{aligned}$$

and hence

$$y(n) = \frac{n}{n+1} y(n-1) + \frac{1}{n+1} x(n) \tag{2.4.3}$$

Now, the cumulative average $y(n)$ can be computed recursively by multiplying the previous output value $y(n-1)$ by $n/(n+1)$, multiplying the present input $x(n)$ by $1/(n+1)$, and adding the two products. Thus the computation of $y(n)$ by means of (2.4.3) requires two multiplications, one addition, and one memory location, as illustrated in Fig. 2.4.1. This is an example of a *recursive system*. In general, a system whose output $y(n)$ at time n depends on any number of past output values $y(n-1)$, $y(n-2), \ldots$ is called a recursive system.

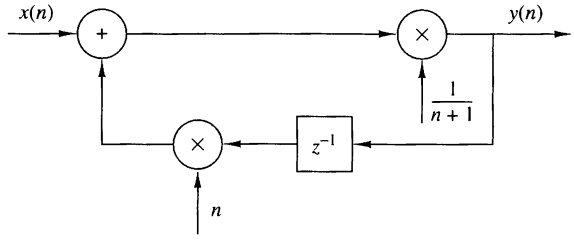


Figure 2.4.1
Realization of a recursive cumulative averaging system.

To determine the computation of the recursive system in (2.4.3) in more detail, suppose that we begin the process with $n = 0$ and proceed forward in time. Thus, according to (2.4.3), we obtain

$$y(0) = x(0)$$

$$y(1) = \frac{1}{2}y(0) + \frac{1}{2}x(1)$$

$$y(2) = \frac{2}{3}y(1) + \frac{1}{3}x(2)$$

and so on. If one grows fatigued with this computation and wishes to pass the problem to someone else at some time, say $n = n_0$, the only information that one needs to provide his or her successor is the past value $y(n_0 - 1)$ and the new input samples $x(n), x(n+1), \ldots$. Thus the successor begins with

$$y(n_0) = \frac{n_0}{n_0 + 1}y(n_0 - 1) + \frac{1}{n_0 + 1}x(n_0)$$

and proceeds forward in time until some time, say $n = n_1$, when he or she becomes fatigued and passes the computational burden to someone else with the information on the value $y(n_1 - 1)$, and so on.

The point we wish to make in this discussion is that if one wishes to compute the response (in this case, the cumulative average) of the system (2.4.3) to an input signal $x(n)$ applied at $n = n_0$, we need the value $y(n_0 - 1)$ and the input samples $x(n)$ for $n \geq n_0$. The term $y(n_0 - 1)$ is called the *initial condition* for the system in (2.4.3) and contains all the information needed to determine the response of the system for $n \geq n_0$ to the input signal $x(n)$, independent of what has occurred in the past.

The following example illustrates the use of a (nonlinear) recursive system to compute the square root of a number.

EXAMPLE 2.4.1 Square-Root Algorithm

Many computers and calculators compute the square root of a positive number A, using the iterative algorithm

$$s_n = \frac{1}{2}\left(s_{n-1} + \frac{A}{s_{n-1}}\right), \qquad n = 0, 1, \ldots$$

where s_{n-1} is an initial guess (estimate) of $\sqrt{A}$. As the iteration converges we have $s_n \approx s_{n-1}$. Then it easily follows that $s_n \approx \sqrt{A}$.

Consider now the recursive system

$$y(n) = \frac{1}{2}\left[y(n-1) + \frac{x(n)}{y(n-1)}\right] \tag{2.4.4}$$

which is realized as in Fig. 2.4.2. If we excite this system with a step of amplitude A [i.e., $x(n) = Au(n)$] and use as an initial condition $y(-1)$ an estimate of $\sqrt{A}$, the response $y(n)$ of the system will tend toward $\sqrt{A}$ as n increases. Note that in contrast to the system (2.4.3), we do not need to specify exactly the initial condition. A rough estimate is sufficient for the proper performance of the system. For example, if we let $A = 2$ and $y(-1) = 1$, we obtain $y(0) = \frac{3}{2}$, $y(1) = 1.4166667$, $y(2) = 1.4142157$. Similarly, for $y(-1) = 1.5$, we have $y(0) = 1.416667$, $y(1) = 1.4142157$. Compare these values with the $\sqrt{2}$, which is approximately 1.4142136.

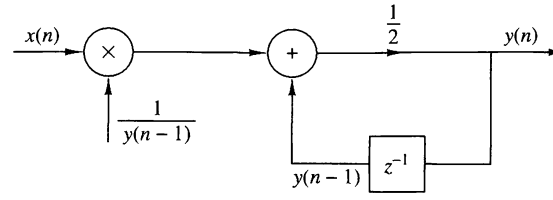


Figure 2.4.2
Realization of the square-root system.

We have now introduced two simple recursive systems, where the output $y(n)$ depends on the previous output value $y(n-1)$ and the current input $x(n)$. Both systems are causal. In general, we can formulate more complex causal recursive systems, in which the output $y(n)$ is a function of several past output values and present and past inputs. The system should have a finite number of delays or, equivalently, should require a finite number of storage locations to be practically implemented. Thus the output of a causal and practically realizable recursive system can be expressed in general as

$$y(n) = F[y(n-1), y(n-2), \ldots, y(n-N), x(n), x(n-1), \ldots, x(n-M)] \quad (2.4.5)$$

where $F[\cdot]$ denotes some function of its arguments. This is a recursive equation specifying a procedure for computing the system output in terms of previous values of the output and present and past inputs.

In contrast, if $y(n)$ depends only on the present and past inputs, then

$$y(n) = F[x(n), x(n-1), \ldots, x(n-M)] \quad (2.4.6)$$

Such a system is called *nonrecursive*. We hasten to add that the causal FIR systems described in Section 2.3.7 in terms of the convolution sum formula have the form of (2.4.6). Indeed, the convolution summation for a causal FIR system is

$$\begin{aligned} y(n) &= \sum_{k=0}^{M} h(k)x(n-k) \\ &= h(0)x(n) + h(1)x(n-1) + \cdots + h(M)x(n-M) \\ &= F[x(n), x(n-1), \ldots, x(n-M)] \end{aligned}$$

where the function $F[\cdot]$ is simply a linear weighted sum of present and past inputs and the impulse response values $h(n)$, $0 \le n \le M$, constitute the weighting coefficients. Consequently, the causal linear time-invariant FIR systems described by the convolution formula in Section 2.3.7 are nonrecursive. The basic differences between nonrecursive and recursive systems are illustrated in Fig. 2.4.3. A simple inspection of this figure reveals that the fundamental difference between these two systems is the feedback loop in the recursive system, which feeds back the output of the system into the input. This feedback loop contains a delay element. The presence of this delay is crucial for the realizability of the system, since the absence of this delay would force the system to compute $y(n)$ in terms of $y(n)$, which is not possible for discrete-time systems.

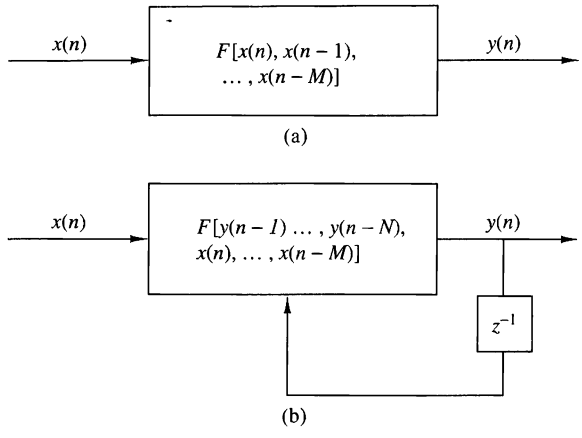


Figure 2.4.3
Basic form for a causal and realizable (a) nonrecursive and (b) recursive system.

The presence of the feedback loop or, equivalently, the recursive nature of (2.4.5) creates another important difference between recursive and nonrecursive systems. For example, suppose that we wish to compute the output $y(n_0)$ of a system when it is excited by an input applied at time $n = 0$. If the system is recursive, to compute $y(n_0)$, we first need to compute all the previous values $y(0), y(1), \ldots, y(n_0 - 1)$. In contrast, if the system is nonrecursive, we can compute the output $y(n_0)$ immediately without having $y(n_0 - 1), y(n_0 - 2), \ldots$. In conclusion, the output of a recursive system should be computed in order [i.e., $y(0), y(1), y(2), \ldots$], whereas for a nonrecursive system, the output can be computed in any order [i.e., $y(200), y(15), y(3), y(300)$, etc.]. This feature is desirable in some practical applications.

2.4.2 Linear Time-Invariant Systems Characterized by Constant-Coefficient Difference Equations

In Section 2.3 we treated linear time-invariant systems and characterized them in terms of their impulse responses. In this subsection we focus our attention on a family of linear time-invariant systems described by an input–output relation called a difference equation with constant coeffficients. Systems described by constant-coefficient linear difference equations are a subclass of the recursive and nonrecursive systems introduced in the preceding subsection. To bring out the important ideas, we begin by treating a simple recursive system described by a first-order difference equation.

Suppose that we have a recursive system with an input–output equation

$$y(n) = ay(n - 1) + x(n) \tag{2.4.7}$$

where a is a constant. Figure 2.4.4 shows a block diagram realization of the system. In comparing this system with the cumulative averaging system described by the input–output equation (2.4.3), we observe that the system in (2.4.7) has a constant coefficient (independent of time), whereas the system described in (2.4.3) has time-variant coefficients. As we will show, (2.4.7) is an input–output equation for a linear time-invariant system, whereas (2.4.3) describes a linear time-variant system.

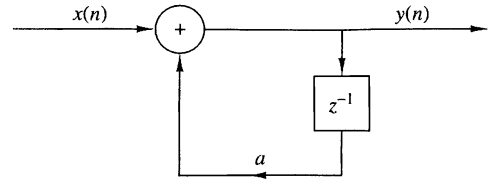


Figure 2.4.4
Block diagram realization of a simple recursive system.

Now, suppose that we apply an input signal $x(n)$ to the system for $n \geq 0$. We make no assumptions about the input signal for $n < 0$, but we do assume the existence of the initial condition $y(-1)$. Since (2.4.7) describes the system output implicitly, we must solve this equation to obtain an explicit expression for the system output. Suppose that we compute successive values of $y(n)$ for $n \geq 0$, beginning with $y(0)$. Thus

$$\begin{aligned}
y(0) &= ay(-1) + x(0) \\
y(1) &= ay(0) + x(1) = a^2y(-1) + ax(0) + x(1) \\
y(2) &= ay(1) + x(2) = a^3y(-1) + a^2x(0) + ax(1) + x(2) \\
&\vdots \quad \vdots \\
y(n) &= ay(n-1) + x(n) \\
&= a^{n+1}y(-1) + a^nx(0) + a^{n-1}x(1) + \cdots + ax(n-1) + x(n)
\end{aligned}$$

or, more compactly,

$$y(n) = a^{n+1}y(-1) + \sum_{k=0}^{n} a^k x(n-k), \qquad n \geq 0 \tag{2.4.8}$$

The response $y(n)$ of the system as given by the right-hand side of (2.4.8) consists of two parts. The first part, which contains the term $y(-1)$, is a result of the initial condition $y(-1)$ of the system. The second part is the response of the system to the input signal $x(n)$.

If the system is initially relaxed at time $n = 0$, then its memory (i.e., the output of the delay) should be zero. Hence $y(-1) = 0$. Thus a recursive system is relaxed if it starts with zero initial conditions. Because the memory of the system describes, in some sense, its "state," we say that the system is at zero state and its corresponding output is called the *zero-state response*, and is denoted by $y_{zs}(n)$. Obviously, the zero-state response of the system (2.4.7) is given by

$$y_{zs}(n) = \sum_{k=0}^{n} a^k x(n-k), \qquad n \geq 0 \tag{2.4.9}$$

It is interesting to note that (2.4.9) is a convolution summation involving the input signal convolved with the impulse response

$$h(n) = a^n u(n) \tag{2.4.10}$$

We also observe that the system described by the first-order difference equation in (2.4.7) is causal. As a result, the lower limit on the convolution summation in (2.4.9) is $k = 0$. Furthermore, the condition $y(-1) = 0$ implies that the input signal can be assumed causal and hence the upper limit on the convolution summation in (2.4.9) is n, since $x(n - k) = 0$ for $k > n$. In effect, we have obtained the result that the relaxed recursive system described by the first-order difference equation in (2.4.7) is a linear time-invariant IIR system with impulse response given by (2.4.10).

Now, suppose that the system described by (2.4.7) is initially nonrelaxed [i.e., $y(-1) \neq 0$] and the input $x(n) = 0$ for all n. Then the output of the system with zero input is called the *zero-input response* or *natural response* and is denoted by $y_{zi}(n)$. From (2.4.7), with $x(n) = 0$ for $-\infty < n < \infty$, we obtain

$$y_{zi}(n) = a^{n+1}y(-1), \qquad n \geq 0 \tag{2.4.11}$$

We observe that a recursive system with nonzero initial condition is nonrelaxed in the sense that it can produce an output without being excited. Note that the zero-input response is due to the memory of the system.

To summarize, the zero-input response is obtained by setting the input signal to zero, making it independent of the input. It depends only on the nature of the system and the initial condition. Thus the zero-input response is a characteristic of the system itself, and it is also known as the *natural* or *free response* of the system. On the other hand, the zero-state response depends on the nature of the system and the input signal. Since this output is a response forced upon it by the input signal, it is usually called the *forced response* of the system. In general, the total response of the system can be expressed as $y(n) = y_{zi}(n) + y_{zs}(n)$.

The system described by the first-order difference equation in (2.4.7) is the simplest possible recursive system in the general class of recursive systems described by linear constant-coefficient difference equations. The general form for such an equation is

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{2.4.12}$$

or, equivalently,

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k), \qquad a_0 \equiv 1 \tag{2.4.13}$$

The integer N is called the *order* of the difference equation or the order of the system. The negative sign on the right-hand side of (2.4.12) is introduced as a matter of convenience to allow us to express the difference equation in (2.4.13) without any negative signs.

Equation (2.4.12) expresses the output of the system at time n directly as a weighted sum of past outputs $y(n-1), y(n-2), \ldots, y(n-N)$ as well as past and present input signals samples. We observe that in order to determine $y(n)$ for $n \geq 0$, we need the input $x(n)$ for all $n \geq 0$, and the initial conditions $y(-1)$,

$y(-2), \ldots, y(-N)$. In other words, the initial conditions summarize all that we need to know about the past history of the response of the system to compute the present and future outputs. The general solution of the N-order constant-coefficient difference equation is considered in the following subsection.

At this point we restate the properties of linearity, time invariance, and stability in the context of recursive systems described by linear constant-coefficient difference equations. As we have observed, a recursive system may be relaxed or nonrelaxed, depending on the initial conditions. Hence the definitions of these properties must take into account the presence of the initial conditions.

We begin with the definition of linearity. A system is linear if it satisfies the following three requirements:

1. The total response is equal to the sum of the zero-input and zero-state responses [i.e., $y(n) = y_{zi}(n) + y_{zs}(n)$].
2. The principle of superposition applies to the zero-state response (*zero-state linear*).
3. The principle of superposition applies to the zero-input response (*zero-input linear*).

A system that does not satisfy *all three* separate requirements is by definition nonlinear. Obviously, for a relaxed system, $y_{zi}(n) = 0$, and thus requirement 2, which is the definition of linearity given in Section 2.2.4, is sufficient.

We illustrate the application of these requirements by a simple example.

EXAMPLE 2.4.2

Determine if the recursive system defined by the difference equation

$$y(n) = ay(n-1) + x(n)$$

is linear.

Solution. By combining (2.4.9) and (2.4.11), we obtain (2.4.8), which can be expressed as

$$y(n) = y_{zi}(n) + y_{zs}(n)$$

Thus the first requirement for linearity is satisfied.

To check for the second requirement, let us assume that $x(n) = c_1x_1(n) + c_2x_2(n)$. Then (2.4.9) gives

$$\begin{aligned} y_{zs}(n) &= \sum_{k=0}^{n} a^k[c_1x_1(n-k) + c_2x_2(n-k)] \\ &= c_1 \sum_{k=0}^{n} a^k x_1(n-k) + c_2 \sum_{k=0}^{n} a^k x_2(n-k) \\ &= c_1 y_{zs}^{(1)}(n) + c_2 y_{zs}^{(2)}(n) \end{aligned}$$

Hence $y_{zs}(n)$ satisfies the principle of superposition, and thus the system is zero-state linear.

Now let us assume that $y(-1) = c_1 y_1(-1) + c_2 y_2(-1)$. From (2.4.11) we obtain

$$\begin{aligned} y_{zi}(n) &= a^{n+1}[c_1 y_1(-1) + c_2 y_2(-1)] \\ &= c_1 a^{n+1} y_1(-1) + c_2 a^{n+1} y_2(-1) \\ &= c_1 y_{zi}^{(1)}(n) + c_2 y_{zi}^{(2)}(n) \end{aligned}$$

Hence the system is zero-input linear.

Since the system satisfies all three conditions for linearity, it is linear.

Although it is somewhat tedious, the procedure used in Example 2.4.2 to demonstrate linearity for the system described by the first-order difference equation carries over directly to the general recursive systems described by the constant-coefficient difference equation given in (2.4.13). Hence, a recursive system described by the linear difference equation in (2.4.13) also satisfies all three conditions in the definition of linearity, and therefore it is linear.

The next question that arises is whether or not the causal linear system described by the linear constant-coefficient difference equation in (2.4.13) is time invariant. This is fairly easy, when dealing with systems described by explicit input–output mathematical relationships. Clearly, the system described by (2.4.13) is time invariant because the coefficients a_k and b_k are constants. On the other hand, if one or more of these coefficients depends on time, the system is time variant, since its properties change as a function of time. Thus we conclude that *the recursive system described by a linear constant-coefficient difference equation is linear and time invariant.*

The final issue is the stability of the recursive system described by the linear, constant-coefficient difference equation in (2.4.13). In Section 2.3.6 we introduced the concept of bounded input–bounded output (BIBO) stability for relaxed systems. For nonrelaxed systems that may be nonlinear, BIBO stability should be viewed with some care. However, in the case of a linear time-invariant recursive system described by the linear constant-coefficient difference equation in (2.4.13), it suffices to state that such a system is BIBO stable if and only if for every bounded input and every bounded initial condition, the total system response is bounded.

EXAMPLE 2.4.3

Determine if the linear time-invariant recursive system described by the difference equation given in (2.4.7) is stable.

Solution. Let us assume that the input signal $x(n)$ is bounded in amplitude, that is, $|x(n)| \le M_x < \infty$ for all $n \ge 0$. From (2.4.8) we have

$$|y(n)| \le |a^{n+1} y(-1)| + \left| \sum_{k=0}^{n} a^k x(n-k) \right|, \qquad n \ge 0$$

$$\le |a|^{n+1} |y(-1)| + M_x \sum_{k=0}^{n} |a|^k, \qquad n \ge 0$$

$$\le |a|^{n+1} |y(-1)| + M_x \frac{1 - |a|^{n+1}}{1 - |a|} = M_y, \qquad n \ge 0$$

If n is finite, the bound M_y is finite and the output is bounded independently of the value of a. However, as $n \to \infty$, the bound M_y remains finite only if $|a| < 1$ because $|a|^n \to 0$ as $n \to \infty$. Then $M_y = M_x/(1 - |a|)$.

Thus the system is stable only if $|a| < 1$.

For the simple first-order system in Example 2.4.3, we were able to express the condition for BIBO stability in terms of the system parameter a, namely $|a| < 1$. We should stress, however, that this task becomes more difficult for higher-order systems. Fortunately, as we shall see in subsequent chapters, other simple and more efficient techniques exist for investigating the stability of recursive systems.

2.4.3 Solution of Linear Constant-Coefficient Difference Equations

Given a linear constant-coefficient difference equation as the input–output relationship describing a linear time-invariant system, our objective in this subsection is to determine an explicit expression for the output $y(n)$. The method that is developed is termed the *direct method.* An alternative method based on the z-transform is described in Chapter 3. For reasons that will become apparent later, the z-transform approach is called the *indirect method.*

Basically, the goal is to determine the output $y(n)$, $n \geq 0$, of the system given a specific input $x(n)$, $n \geq 0$, and a set of initial conditions. The direct solution method assumes that the total solution is the sum of two parts:

$$y(n) = y_h(n) + y_p(n)$$

The part $y_h(n)$ is known as the *homogeneous* or *complementary* solution, whereas $y_p(n)$ is called the *particular* solution.

The homogeneous solution of a difference equation. We begin the problem of solving the linear constant-coefficient difference equation given by (2.4.13) by obtaining first the solution to the *homogeneous difference equation*

$$\sum_{k=0}^{N} a_k y(n-k) = 0 \tag{2.4.14}$$

The procedure for solving a linear constant-coefficient difference equation directly is very similar to the procedure for solving a linear constant-coefficient differential equation. Basically, we assume that the solution is in the form of an exponential, that is,

$$y_h(n) = \lambda^n \tag{2.4.15}$$

where the subscript h on $y(n)$ is used to denote the solution to the homogeneous difference equation. If we substitute this assumed solution in (2.4.14), we obtain the polynomial equation

$$\sum_{k=0}^{N} a_k \lambda^{n-k} = 0$$

or

$$\lambda^{n-N}(\lambda^N + a_1\lambda^{N-1} + a_2\lambda^{N-2} + \cdots + a_{N-1}\lambda + a_N) = 0 \tag{2.4.16}$$

The polynomial in parentheses is called the *characteristic polynomial* of the system. In general, it has N roots, which we denote as $\lambda_1, \lambda_2, \ldots, \lambda_N$. The roots can be real or complex valued. In practice the coefficients $a_1, a_2, \ldots, a_N$ are usually real. Complex-valued roots occur as complex-conjugate pairs. Some of the N roots may be identical, in which case we have multiple-order roots.

For the moment, let us assume that the roots are distinct, that is, there are no multiple-order roots. Then the most general solution to the homogeneous difference equation in (2.4.14) is

$$y_h(n) = C_1\lambda_1^n + C_2\lambda_2^n + \cdots + C_N\lambda_N^n \tag{2.4.17}$$

where $C_1, C_2, \ldots, C_N$ are weighting coefficients.

These coefficients are determined from the initial conditions specified for the system. Since the input $x(n) = 0$, (2.4.17) can be used to obtain the *zero-input response* of the system. The following examples illustrate the procedure.

EXAMPLE 2.4.4

Determine the homogeneous solution of the system described by the first-order difference equation

$$y(n) + a_1y(n-1) = x(n) \tag{2.4.18}$$

Solution. The assumed solution obtained by setting $x(n) = 0$ is

$$y_h(n) = \lambda^n$$

When we substitute this solution in (2.4.18), we obtain [with $x(n) = 0$]

$$\lambda^n + a_1\lambda^{n-1} = 0$$

$$\lambda^{n-1}(\lambda + a_1) = 0$$

$$\lambda = -a_1$$

Therefore, the solution to the homogeneous difference equation is

$$y_h(n) = C\lambda^n = C(-a_1)^n \tag{2.4.19}$$

The zero-input response of the system can be determined from (2.4.18) and (2.4.19). With $x(n) = 0$, (2.4.18) yields

$$y(0) = -a_1y(-1)$$

On the other hand, from (2.4.19) we have

$$y_h(0) = C$$

and hence the zero-input response of the system is

$$y_{zi}(n) = (-a_1)^{n+1}y(-1), \qquad n \geq 0 \tag{2.4.20}$$

With $a = -a_1$, this result is consistent with (2.4.11) for the first-order system, which was obtained earlier by iteration of the difference equation.

EXAMPLE 2.4.5

Determine the zero-input response of the system described by the homogeneous second-order difference equation

$$y(n) - 3y(n-1) - 4y(n-2) = 0 \tag{2.4.21}$$

Solution. First we determine the solution to the homogeneous equation. We assume the solution to be the exponential

$$y_h(n) = \lambda^n$$

Upon substitution of this solution into (2.4.21), we obtain the characteristic equation

$$\lambda^n - 3\lambda^{n-1} - 4\lambda^{n-2} = 0$$

$$\lambda^{n-2}(\lambda^2 - 3\lambda - 4) = 0$$

Therefore, the roots are $\lambda = -1, 4$, and the general form of the solution to the homogeneous equation is

$$\begin{aligned} y_h(n) &= C_1\lambda_1^n + C_2\lambda_2^n \\ &= C_1(-1)^n + C_2(4)^n \end{aligned} \tag{2.4.22}$$

The zero-input response of the system can be obtained from the homogenous solution by evaluating the constants in (2.4.22), given the initial conditions $y(-1)$ and $y(-2)$. From the difference equation in (2.4.21) we have

$$y(0) = 3y(-1) + 4y(-2)$$

$$\begin{aligned} y(1) &= 3y(0) + 4y(-1) \\ &= 3[3y(-1) + 4y(-2)] + 4y(-1) \\ &= 13y(-1) + 12y(-2) \end{aligned}$$

On the other hand, from (2.4.22) we obtain

$$y(0) = C_1 + C_2$$

$$y(1) = -C_1 + 4C_2$$

By equating these two sets of relations, we have

$$C_1 + C_2 = 3y(-1) + 4y(-2)$$

$$-C_1 + 4C_2 = 13y(-1) + 12y(-2)$$

The solution of these two equations is

$$C_1 = -\frac{1}{5}y(-1) + \frac{4}{5}y(-2)$$

$$C_2 = \frac{16}{5}y(-1) + \frac{16}{5}y(-2)$$

Therefore, the zero-input response of the system is

$$\begin{aligned} y_{zi}(n) &= [-\frac{1}{5}y(-1) + \frac{4}{5}y(-2)](-1)^n \\ &\quad + [\frac{16}{5}y(-1) + \frac{16}{5}y(-2)](4)^n, \qquad n \geq 0 \end{aligned} \tag{2.4.23}$$

For example, if $y(-2) = 0$ and $y(-1) = 5$, then $C_1 = -1$, $C_2 = 16$, and hence

$$y_{zi}(n) = (-1)^{n+1} + (4)^{n+2}, \qquad n \geq 0$$

These examples illustrate the method for obtaining the homogeneous solution and the zero-input response of the system when the characteristic equation contains distinct roots. On the other hand, if the characteristic equation contains multiple roots, the form of the solution given in (2.4.17) must be modified. For example, if λ_1 is a root of multiplicity m, then (2.4.17) becomes

$$\begin{aligned} y_h(n) &= C_1\lambda_1^n + C_2 n\lambda_1^n + C_3 n^2\lambda_1^n + \cdots + C_m n^{m-1}\lambda_1^n \\ &\quad + C_{m+1}\lambda_{m+1}^n + \cdots + C_N\lambda_N^n \end{aligned} \tag{2.4.24}$$

The particular solution of the difference equation. The particular solution $y_p(n)$ is required to satisfy the difference equation (2.4.13) for the specific input signal $x(n)$, $n \geq 0$. In other words, $y_p(n)$ is any solution satisfying

$$\sum_{k=0}^{N} a_k y_p(n-k) = \sum_{k=0}^{M} b_k x(n-k), \qquad a_0 = 1 \tag{2.4.25}$$

To solve (2.4.25), we assume for $y_p(n)$, a form that depends on the form of the input $x(n)$. The following example illustrates the procedure.

EXAMPLE 2.4.6

Determine the particular solution of the first-order difference equation

$$y(n) + a_1 y(n-1) = x(n), \qquad |a_1| < 1 \tag{2.4.26}$$

when the input $x(n)$ is a unit step sequence, that is,

$$x(n) = u(n)$$

Solution. Since the input sequence $x(n)$ is a constant for $n \geq 0$, the form of the solution that we assume is also a constant. Hence the assumed solution of the difference equation to the forcing function $x(n)$, called the *particular solution* of the difference equation, is

$$y_p(n) = Ku(n)$$

where K is a scale factor determined so that (2.4.26) is satisfied. Upon substitution of this assumed solution into (2.4.26), we obtain

$$Ku(n) + a_1 Ku(n-1) = u(n)$$

To determine K, we must evaluate this equation for any $n \geq 1$, where none of the terms vanish. Thus

$$K + a_1 K = 1$$

$$K = \frac{1}{1 + a_1}$$

Therefore, the particular solution to the difference equation is

$$y_p(n) = \frac{1}{1 + a_1} u(n) \tag{2.4.27}$$

In this example, the input $x(n)$, $n \geq 0$, is a constant and the form assumed for the particular solution is also a constant. If $x(n)$ is an exponential, we would assume that the particular solution is also an exponential. If $x(n)$ were a sinusoid, then $y_p(n)$ would also be a sinusoid. Thus our assumed form for the particular solution takes the basic form of the signal $x(n)$. Table 2.1 provides the general form of the particular solution for several types of excitation.

EXAMPLE 2.4.7

Determine the particular solution of the difference equation

$$y(n) = \frac{5}{6} y(n-1) - \frac{1}{6} y(n-2) + x(n)$$

when the forcing function $x(n) = 2^n$, $n \geq 0$ and zero elsewhere.

TABLE 2.1 General Form of the Particular Solution for Several Types of Input Signals

Input Signal, $x(n)$	Particular Solution, $y_p(n)$
A (constant)	K
AM^n	KM^n
An^M	$K_0 n^M + K_1 n^{M-1} + \cdots + K_M$
$A^n n^M$	$A^n(K_0 n^M + K_1 n^{M-1} + \cdots + K_M)$
$\left\{ \begin{matrix} A\cos\omega_0 n \\ A\sin\omega_0 n \end{matrix} \right\}$	$K_1 \cos\omega_0 n + K_2 \sin\omega_0 n$

Solution. The form of the particular solution is

$$y_p(n) = K2^n, \qquad n \geq 0$$

Upon substitution of $y_p(n)$ into the difference equation, we obtain

$$K2^n u(n) = \frac{5}{6}K2^{n-1}u(n-1) - \frac{1}{6}K2^{n-2}u(n-2) + 2^n u(n)$$

To determine the value of K, we can evaluate this equation for any $n \geq 2$, where none of the terms vanish. Thus we obtain

$$4K = \frac{5}{6}(2K) - \frac{1}{6}K + 4$$

and hence $K = \frac{8}{5}$. Therefore, the particular solution is

$$y_p(n) = \frac{8}{5}2^n, \qquad n \geq 0$$

We have now demonstrated how to determine the two components of the solution to a difference equation with constant coefficients. These two components are the homogeneous solution and the particular solution. From these two components, we construct the total solution from which we can obtain the zero-state response.

The total solution of the difference equation. The linearity property of the linear constant-coefficient difference equation allows us to add the homogeneous solution and the particular solution in order to obtain the *total solution*. Thus

$$y(n) = y_h(n) + y_p(n)$$

The resultant sum $y(n)$ contains the constant parameters $\{C_i\}$ embodied in the homogeneous solution component $y_h(n)$. These constants can be determined to satisfy the initial conditions. The following example illustrates the procedure.

EXAMPLE 2.4.8

Determine the total solution $y(n)$, $n \geq 0$, to the difference equation

$$y(n) + a_1 y(n-1) = x(n) \tag{2.4.28}$$

when $x(n)$ is a unit step sequence [i.e., $x(n) = u(n)$] and $y(-1)$ is the initial condition.

Solution. From (2.4.19) of Example 2.4.4, the homogeneous solution is

$$y_h(n) = C(-a_1)^n$$

and from (2.4.26) of Example 2.4.6, the particular solution is

$$y_p(n) = \frac{1}{1+a_1}u(n)$$

Consequently, the total solution is

$$y(n) = C(-a_1)^n + \frac{1}{1+a_1}, \qquad n \geq 0 \tag{2.4.29}$$

where the constant C is determined to satisfy the initial condition $y(-1)$.

In particular, suppose that we wish to obtain the zero-state response of the system described by the first-order difference equation in (2.4.28). Then we set $y(-1) = 0$. To evaluate C, we evaluate (2.4.28) at $n = 0$, obtaining

$$y(0) + a_1 y(-1) = 1$$

Hence,

$$y(0) = 1 - a_1 y(-1)$$

On the other hand, (2.4.29) evaluated at $n = 0$ yields

$$y(0) = C + \frac{1}{1+a_1}$$

By equating these two relations, we obtain

$$C + \frac{1}{1+a_1} = -a_1 y(-1) + 1$$

$$C = -a_1 y(-1) + \frac{a_1}{1+a_1}$$

Finally, if we substitute this value of C into (2.4.29), we obtain

$$\begin{aligned} y(n) &= (-a_1)^{n+1} y(-1) + \frac{1-(-a_1)^{n+1}}{1+a_1}, \qquad n \geq 0 \\ &= y_{zi}(n) + y_{zs}(n) \end{aligned} \tag{2.4.30}$$

We observe that the system response as given by (2.4.30) is consistent with the response $y(n)$ given in (2.4.8) for the first-order system (with $a = -a_1$), which was obtained by solving the difference equation iteratively. Furthermore, we note that the value of the constant C depends both on the initial condition $y(-1)$ and on the excitation function. Consequently, the value of C influences both the zero-input response and the zero-state response.

We further observe that the particular solution to the difference equation can be obtained from the zero-state response of the system. Indeed, if $|a_1| < 1$, which is the condition for stability of the system, as will be shown in Section 2.4.4, the limiting value of $y_{zs}(n)$ as n approaches infinity is the particular solution, that is,

$$y_p(n) = \lim_{n\to\infty} y_{zs}(n) = \frac{1}{1+a_1}$$

Since this component of the system response does not go to zero as n approaches infinity, it is usually called the *steady-state response* of the system. This response persists as long as the input persists. The component that dies out as n approaches infinity is called the *transient response* of the system.

The following example illustrates the evaluation of the total solution for a second-order recursive system.

EXAMPLE 2.4.9

Determine the response $y(n)$, $n \geq 0$, of the system described by the second-order difference equation

$$y(n) - 3y(n-1) - 4y(n-2) = x(n) + 2x(n-1) \tag{2.4.31}$$

when the input sequence is

$$x(n) = 4^n u(n)$$

Solution. We have already determined the solution to the homogeneous difference equation for this system in Example 2.4.5. From (2.4.22) we have

$$y_h(n) = C_1(-1)^n + C_2(4)^n \tag{2.4.32}$$

The particular solution to (2.4.31) is assumed to be an exponential sequence of the same form as $x(n)$. Normally, we could assume a solution of the form

$$y_p(n) = K(4)^n u(n)$$

However, we observe that $y_p(n)$ is already contained in the homogeneous solution, so that this particular solution is redundant. Instead, we select the particular solution to be linearly independent of the terms contained in the homogeneous solution. In fact, we treat this situation in the same manner as we have already treated multiple roots in the characteristic equation. Thus we assume that

$$y_p(n) = Kn(4)^n u(n) \tag{2.4.33}$$

Upon substitution of (2.4.33) into (2.4.31), we obtain

$$Kn(4)^n u(n) - 3K(n-1)(4)^{n-1}u(n-1) - 4K(n-2)(4)^{n-2}u(n-2) = (4)^n u(n) + 2(4)^{n-1}u(n-1)$$

To determine K, we evaluate this equation for any $n \geq 2$, where none of the unit step terms vanish. To simplify the arithmetic, we select $n = 2$, from which we obtain $K = \frac{6}{5}$. Therefore,

$$y_p(n) = \frac{6}{5}n(4)^n u(n) \tag{2.4.34}$$

The total solution to the difference equation is obtained by adding (2.4.32) to (2.4.34). Thus

$$y(n) = C_1(-1)^n + C_2(4)^n + \frac{6}{5}n(4)^n, \qquad n \geq 0 \tag{2.4.35}$$

where the constants C_1 and C_2 are determined such that the initial conditions are satisfied. To accomplish this, we return to (2.4.31), from which we obtain

$$\begin{aligned} y(0) &= 3y(-1) + 4y(-2) + 1 \\ y(1) &= 3y(0) + 4y(-1) + 6 \\ &= 13y(-1) + 12y(-2) + 9 \end{aligned}$$

On the other hand, (2.4.35) evaluated at $n = 0$ and $n = 1$ yields

$$\begin{aligned} y(0) &= C_1 + C_2 \\ y(1) &= -C_1 + 4C_2 + \frac{24}{5} \end{aligned}$$

We can now equate these two sets of relations to obtain C_1 and C_2. In so doing, we have the response due to initial conditions $y(-1)$ and $y(-2)$ (the zero-input response), and the zero-state response.

Since we have already solved for the zero-input response in Example 2.4.5, we can simplify the computations above by setting $y(-1) = y(-2) = 0$. Then we have

$$C_1 + C_2 = 1$$

$$-C_1 + 4C_2 + \frac{24}{5} = 9$$

Hence $C_1 = -\frac{1}{25}$ and $C_2 = \frac{26}{25}$. Finally, we have the zero-state response to the forcing function $x(n) = (4)^n u(n)$ in the form

$$y_{zs}(n) = -\frac{1}{25}(-1)^n + \frac{26}{25}(4)^n + \frac{6}{5}n(4)^n, \qquad n \geq 0 \tag{2.4.36}$$

The total response of the system, which includes the response to arbitrary initial conditions, is the sum of (2.4.23) and (2.4.36).

2.4.4 The Impulse Response of a Linear Time-Invariant Recursive System

The impulse response of a linear time-invariant system was previously defined as the response of the system to a unit sample excitation [i.e., $x(n) = \delta(n)$]. In the case of a recursive system, $h(n)$ is simply equal to the zero-state response of the system when the input $x(n) = \delta(n)$ and the system is initially relaxed.

For example, in the simple first-order recursive system given in (2.4.7), the zero-state response given in (2.4.8), is

$$y_{zs}(n) = \sum_{k=0}^{n} a^k x(n-k) \tag{2.4.37}$$

When $x(n) = \delta(n)$ is substituted into (2.4.37), we obtain

$$\begin{aligned} y_{zs}(n) &= \sum_{k=0}^{n} a^k \delta(n-k) \\ &= a^n, \qquad n \geq 0 \end{aligned}$$

Hence the impulse response of the first-order recursive system described by (2.4.7) is

$$h(n) = a^n u(n) \tag{2.4.38}$$

as indicated in Section 2.4.2.

In the general case of an arbitrary, linear time-invariant recursive system, the zero-state response expressed in terms of the convolution summation is

$$y_{zs}(n) = \sum_{k=0}^{n} h(k)x(n-k), \qquad n \geq 0 \tag{2.4.39}$$

When the input is an impulse [i.e., $x(n) = \delta(n)$], (2.4.39) reduces to

$$y_{zs}(n) = h(n) \tag{2.4.40}$$

Now, let us consider the problem of determining the impulse response $h(n)$ given a linear constant-coefficient difference equation description of the system. In terms of our discussion in the preceding subsection, we have established the fact that the total response of the system to any excitation function consists of the sum of two solutions of the difference equation: the solution to the homogeneous equation plus the particular solution to the excitation function. In the case where the excitation is an impulse, the particular solution is zero, since $x(n) = 0$ for $n > 0$, that is,

$$y_p(n) = 0$$

Consequently, the response of the system to an impulse consists only of the solution to the homogeneous equation, with the $\{C_k\}$ parameters evaluated to satisfy the initial conditions dictated by the impulse. The following example illustrates the procedure for obtaining $h(n)$ given the difference equation for the system.

EXAMPLE 2.4.10

Determine the impulse response $h(n)$ for the system described by the second-order difference equation

$$y(n) - 3y(n-1) - 4y(n-2) = x(n) + 2x(n-1) \tag{2.4.41}$$

Solution. We have already determined in Example 2.4.5 that the solution to the homogeneous difference equation for this system is

$$y_h(n) = C_1(-1)^n + C_2(4)^n, \qquad n \geq 0 \tag{2.4.42}$$

Since the particular solution is zero when $x(n) = \delta(n)$, the impulse response of the system is simply given by (2.4.42), where C_1 and C_2 must be evaluated to satisfy (2.4.41).

For $n = 0$ and $n = 1$, (2.4.41) yields

$$y(0) = 1$$

$$y(1) = 3y(0) + 2 = 5$$

where we have imposed the conditions $y(-1) = y(-2) = 0$, since the system must be relaxed. On the other hand, (2.4.42) evaluated at $n = 0$ and $n = 1$ yields

$$y(0) = C_1 + C_2$$

$$y(1) = -C_1 + 4C_2$$

By solving these two sets of equations for C_1 and C_2, we obtain

$$C_1 = -\frac{1}{5}, \qquad C_2 = \frac{6}{5}$$

Therefore, the impulse response of the system is

$$h(n) = \left[-\frac{1}{5}(-1)^n + \frac{6}{5}(4)^n\right] u(n)$$

When the system is described by an Nth-order linear difference equation of the type given in (2.4.13), the solution of the homogeneous equation is

$$y_h(n) = \sum_{k=1}^{N} C_k \lambda^n k$$

when the roots $\{\lambda_k\}$ of the characteristic polynomial are distinct. Hence the impulse response of the system is identical in form, that is,

$$h(n) = \sum_{k=1}^{N} C_k \lambda_k^n \tag{2.4.43}$$

where the parameters $\{C_k\}$ are determined by setting the initial conditions $y(-1) = \cdots = y(-N) = 0$.

This form of $h(n)$ allows us to easily relate the stability of a system, described by an Nth-order difference equation, to the values of the roots of the characteristic polynomial. Indeed, since BIBO stability requires that the impulse response be absolutely summable, then, for a causal system, we have

$$\sum_{n=0}^{\infty} |h(n)| = \sum_{n=0}^{\infty} \left| \sum_{k=1}^{N} C_k \lambda^n k \right| \leq \sum_{k=1}^{N} |C_k| \sum_{n=0}^{\infty} |\lambda_k|^n$$

Now if $|\lambda_k| < 1$ for all k, then

$$\sum_{n=0}^{\infty} |\lambda_k|^n < \infty$$

and hence

$$\sum_{n=0}^{\infty} |h(n)| < \infty$$

On the other hand, if one or more of the $|\lambda_k| \geq 1$, $h(n)$ is no longer absolutely summable, and consequently, the system is unstable. Therefore, a necessary and sufficient condition for the stability of a causal IIR system described by a linear constant-coefficient difference equation is that all roots of the characteristic polynomial be less than unity in magnitude. The reader may verify that this condition carries over to the case where the system has roots of multiplicity m.

Finally we note that any recursive system described by a linear constant-coefficient difference equation is an IIR system. The converse is not true, however. That is, not every linear time-invariant IIR system can be described by a linear constant-coefficient difference equation. In other words, recursive systems described by linear constant-coefficient difference equations are a subclass of linear time-invariant IIR systems.

2.5 Implementation of Discrete-Time Systems

Our treatment of discrete-time systems has been focused on the time-domain characterization and analysis of linear time-invariant systems described by constant-coefficient linear difference equations. Additional analytical methods are developed in the next two chapters, where we characterize and analyze LTI systems in the frequency domain. Two other important topics that will be treated later are the design and implementation of these systems.

In practice, system design and implementation are usually treated jointly rather than separately. Often, the system design is driven by the method of implementation and by implementation constraints, such as cost, hardware limitations, size limitations, and power requirements. At this point, we have not as yet developed the necessary analysis and design tools to treat such complex issues. However, we have developed sufficient background to consider some basic implementation methods for realizations of LTI systems described by linear constant-coefficient difference equations.

2.5.1 Structures for the Realization of Linear Time-Invariant Systems

In this subsection we describe structures for the realization of systems described by linear constant-coefficient difference equations. Additional structures for these systems are introduced in Chapter 9.

As a beginning, let us consider the first-order system

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1) \tag{2.5.1}$$

which is realized as in Fig. 2.5.1(a). This realization uses separate delays (memory) for both the input and output signal samples and it is called a *direct form I structure.* Note that this system can be viewed as two linear time-invariant systems in cascade. The first is a nonrecursive system described by the equation

$$v(n) = b_0 x(n) + b_1 x(n-1) \tag{2.5.2}$$

whereas the second is a recursive system described by the equation

$$y(n) = -a_1 y(n-1) + v(n) \tag{2.5.3}$$

However, as we have seen in Section 2.3.4, if we interchange the order of the cascaded linear time-invariant systems, the overall system response remains the same. Thus if we interchange the order of the recursive and nonrecursive systems, we obtain an alternative structure for the realization of the system described by (2.5.1). The resulting system is shown in Fig. 2.5.1(b). From this figure we obtain the two difference equations

$$w(n) = -a_1 w(n-1) + x(n) \tag{2.5.4}$$

$$y(n) = b_0 w(n) + b_1 w(n-1) \tag{2.5.5}$$

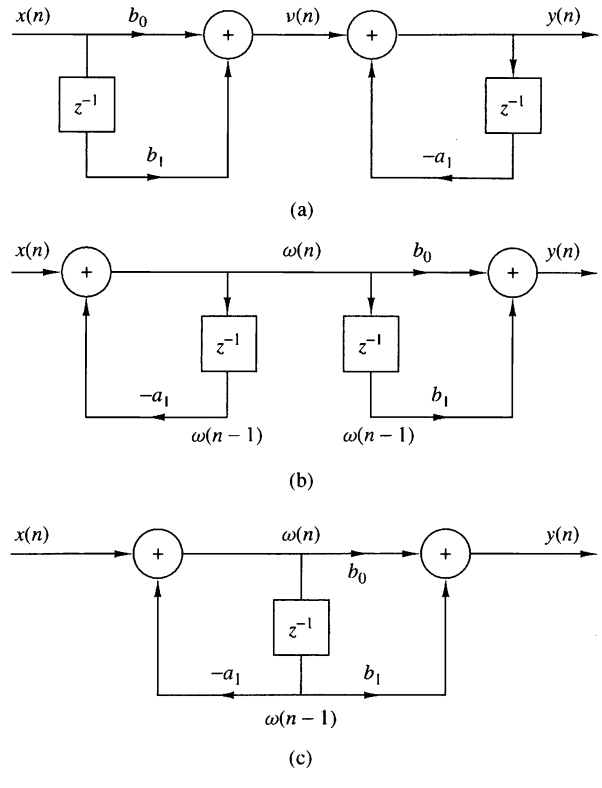


Figure 2.5.1
Steps in converting from the direct form I realization in (a) to the direct form II realization in (c).

which provide an alternative algorithm for computing the output of the system described by the single difference equation given in (2.5.1). In other words, the two difference equations (2.5.4) and (2.5.5) are equivalent to the single difference equation (2.5.1).

A close observation of Fig. 2.5.1 reveals that the two delay elements contain the same input $w(n)$ and hence the same output $w(n-1)$. Consequently, these two elements can be merged into one delay, as shown in Fig. 2.5.1(c). In contrast to the direct form I structure, this new realization requires only one delay for the auxiliary quantity $w(n)$, and hence it is more efficient in terms of memory requirements. It is called the *direct form II structure* and it is used extensively in practical applications.

These structures can readily be generalized for the general linear time-invariant recursive system described by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.6}$$

Figure 2.5.2 illustrates the direct form I structure for this system. This structure requires $M + N$ delays and $N + M + 1$ multiplications. It can be viewed as the

cascade of a nonrecursive system

$$v(n) = \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.7}$$

and a recursive system

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + v(n) \tag{2.5.8}$$

By reversing the order of these two systems, as was previously done for the first-order system, we obtain the direct form II structure shown in Fig. 2.5.3 for $N > M$. This structure is the cascade of a recursive system

$$w(n) = -\sum_{k=1}^{N} a_k w(n-k) + x(n) \tag{2.5.9}$$

followed by a nonrecursive system

$$y(n) = \sum_{k=0}^{M} b_k w(n-k) \tag{2.5.10}$$

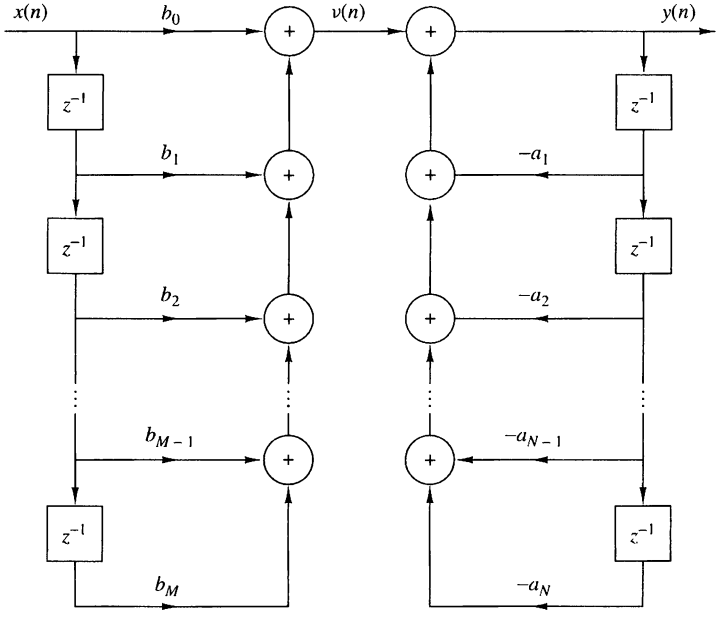


Figure 2.5.2 Direct form I structure of the system described by (2.5.6).

We observe that if $N \geq M$, this structure requires a number of delays equal to the order N of the system. However, if $M > N$, the required memory is specified by M. Figure 2.5.3 can easily be modified to handle this case. Thus the direct form II structure requires $M + N + 1$ multiplications and $\max\{M, N\}$ delays. Because it requires the minimum number of delays for the realization of the system described by (2.5.6), it is sometimes called a *canonic form.*

A special case of (2.5.6) occurs if we set the system parameters $a_k = 0$, $k = 1, \ldots, N$. Then the input–output relationship for the system reduces to

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.11}$$

which is a nonrecursive linear time-invariant system. This system views only the most recent $M + 1$ input signal samples and, prior to addition, weights each sample by the appropriate coefficient b_k from the set $\{b_k\}$. In other words, the system output is basically a *weighted moving average* of the input signal. For this reason it is sometimes called a *moving average (MA) system.* Such a system is an FIR system

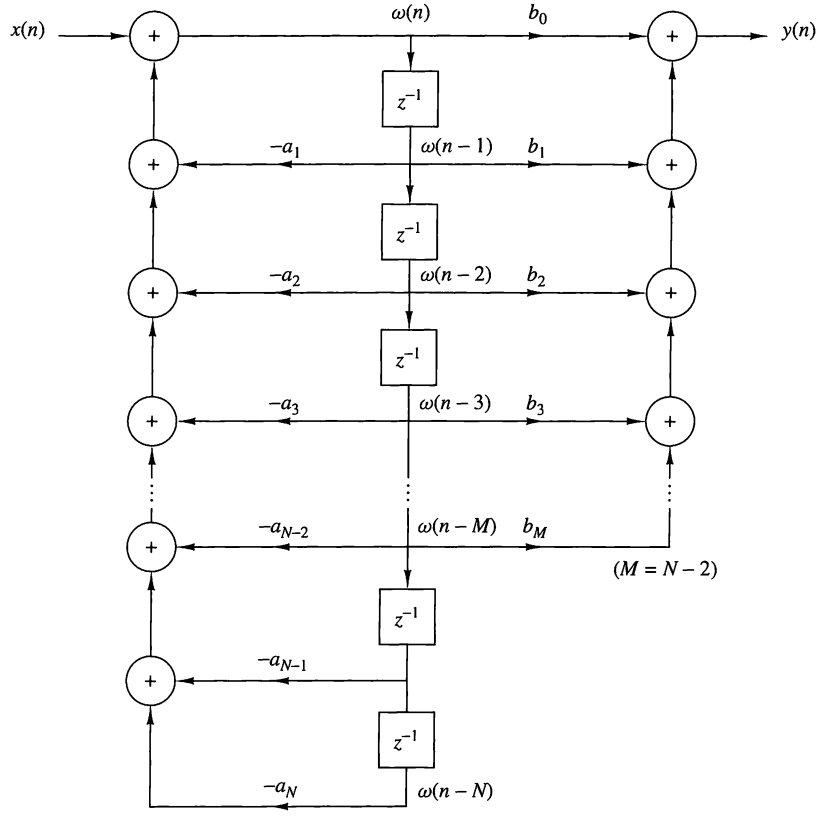


Figure 2.5.3 Direct form II structure for the system described by (2.5.6).

with an impulse response $h(k)$ equal to the coefficients b_k, that is,

$$h(k) = \begin{cases} b_k, & 0 \le k \le M \\ 0, & \text{otherwise} \end{cases} \tag{2.5.12}$$

If we return to (2.5.6) and set $M = 0$, the general linear time-invariant system reduces to a "purely recursive" system described by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + b_0 x(n) \tag{2.5.13}$$

In this case the system output is a weighted linear combination of N past outputs and the present input.

Linear time-invariant systems described by a second-order difference equation are an important subclass of the more general systems described by (2.5.6) or (2.5.10) or (2.5.13). The reason for their importance will be explained later when we discuss quantization effects. Suffice to say at this point that second-order systems are usually used as basic building blocks for realizing higher-order systems.

The most general second-order system is described by the difference equation

$$\begin{aligned} y(n) = &- a_1 y(n-1) - a_2 y(n-2) + b_0 x(n) \\ &+ b_1 x(n-1) + b_2 x(n-2) \end{aligned} \tag{2.5.14}$$

which is obtained from (2.5.6) by setting $N = 2$ and $M = 2$. The direct form II structure for realizing this system is shown in Fig. 2.5.4(a). If we set $a_1 = a_2 = 0$, then (2.5.14) reduces to

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2) \tag{2.5.15}$$

which is a special case of the FIR system described by (2.5.11). The structure for realizing this system is shown in Fig. 2.5.4(b). Finally, if we set $b_1 = b_2 = 0$ in (2.5.14), we obtain the purely recursive second-order system described by the difference equation

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 x(n) \tag{2.5.16}$$

which is a special case of (2.5.13). The structure for realizing this system is shown in Fig. 2.5.4(c).

2.5.2 Recursive and Nonrecursive Realizations of FIR Systems

We have already made the distinction between FIR and IIR systems, based on whether the impulse response $h(n)$ of the system has a finite duration, or an infinite duration. We have also made the distinction between recursive and nonrecursive systems. Basically, a causal recursive system is described by an input–output equation of the form

$$y(n) = F[y(n-1), \ldots, y(n-N), x(n), \ldots, x(n-M)] \tag{2.5.17}$$

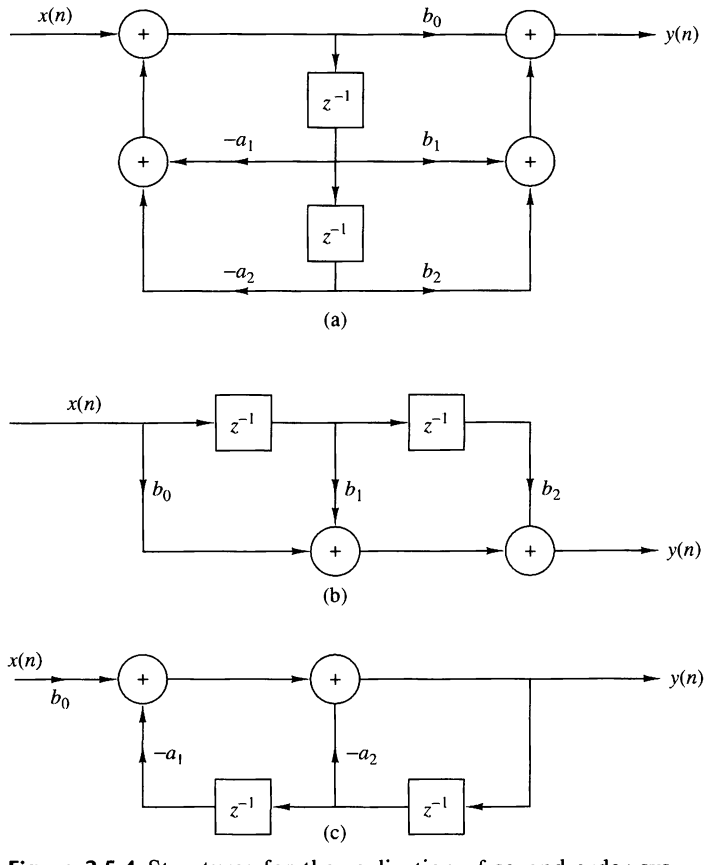


Figure 2.5.4 Structures for the realization of second-order systems: (a) general second-order system; (b) FIR system; (c) "purely recursive system."

and for a linear time-invariant system specifically, by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.18}$$

On the other hand, causal nonrecursive systems do not depend on past values of the output and hence are described by an input–output equation of the form

$$y(n) = F[x(n), x(n-1), \ldots, x(n-M)] \tag{2.5.19}$$

and for linear time-invariant systems specifically, by the difference equation in (2.5.18) with $a_k = 0$ for $k = 1, 2, \ldots, N$.

In the case of FIR systems, we have already observed that it is always possible to realize such systems nonrecursively. In fact, with $a_k = 0, k = 1, 2, \ldots, N$, in (2.5.18),

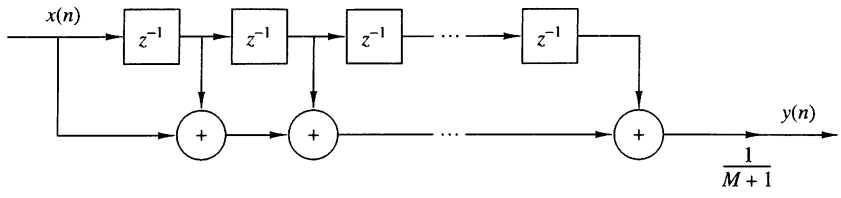


Figure 2.5.5 Nonrecursive realization of an FIR moving average system.

we have a system with an input–output equation

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) \tag{2.5.20}$$

This is a nonrecursive and FIR system. As indicated in (2.5.12), the impulse response of the system is simply equal to the coefficients $\{b_k\}$. Hence every FIR system can be realized nonrecursively. On the other hand, any FIR system can also be realized recursively. Although the general proof of this statement is given later, we shall give a simple example to illustrate the point.

Suppose that we have an FIR system of the form

$$y(n) = \frac{1}{M+1} \sum_{k=0}^{M} x(n-k) \tag{2.5.21}$$

for computing the *moving average* of a signal $x(n)$. Clearly, this system is FIR with impulse response

$$h(n) = \frac{1}{M+1}, \qquad 0 \le n \le M$$

Figure 2.5.5 illustrates the structure of the nonrecursive realization of the system. Now, suppose that we express (2.5.21) as

$$\begin{aligned} y(n) &= \frac{1}{M+1} \sum_{k=0}^{M} x(n-1-k) \\ &\quad + \frac{1}{M+1}[x(n) - x(n-1-M)] \\ &= y(n-1) + \frac{1}{M+1}[x(n) - x(n-1-M)] \end{aligned} \tag{2.5.22}$$

Now, (2.5.22) represents a recursive realization of the FIR system. The structure of this recursive realization of the moving average system is illustrated in Fig. 2.5.6.

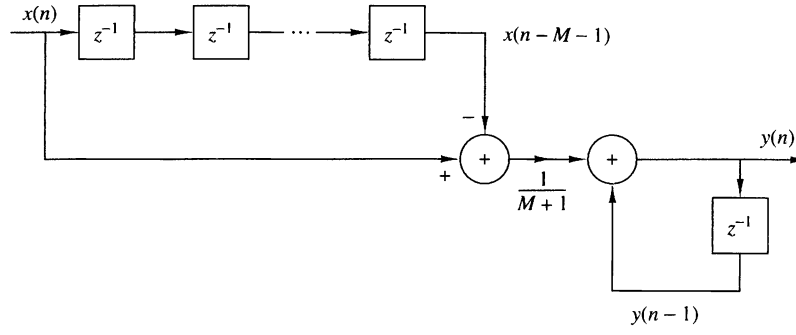


Figure 2.5.6 Recursive realization of an FIR moving average system.

In summary, we can think of the terms FIR and IIR as general characteristics that distinguish a type of linear time-invariant system, and of the terms *recursive* and *nonrecursive* as descriptions of the structures for realizing or implementing the system.

2.6 Correlation of Discrete-Time Signals

A mathematical operation that closely resembles convolution is correlation. Just as in the case of convolution, two signal sequences are involved in correlation. In contrast to convolution, however, our objective in computing the correlation between the two signals is to measure the degree to which the two signals are similar and thus to extract some information that depends to a large extent on the application. Correlation of signals is often encountered in radar, sonar, digital communications, geology, and other areas in science and engineering.

To be specific, let us suppose that we have two signal sequences $x(n)$ and $y(n)$ that we wish to compare. In radar and active sonar applications, $x(n)$ can represent the sampled version of the transmitted signal and $y(n)$ can represent the sampled version of the received signal at the output of the analog-to-digital (A/D) converter. If a target is present in the space being searched by the radar or sonar, the received signal $y(n)$ consists of a delayed version of the transmitted signal, reflected from the target, and corrupted by additive noise. Figure 2.6.1 depicts the radar signal reception problem.

We can represent the received signal sequence as

$$y(n) = \alpha x(n - D) + w(n) \tag{2.6.1}$$

where α is some attenuation factor representing the signal loss involved in the round-trip transmission of the signal $x(n)$, D is the round-trip delay, which is assumed to be an integer multiple of the sampling interval, and $w(n)$ represents the additive noise that is picked up by the antenna and any noise generated by the electronic components and amplifiers contained in the front end of the receiver. On the other hand, if there is no target in the space searched by the radar and sonar, the received signal $y(n)$ consists of noise alone.

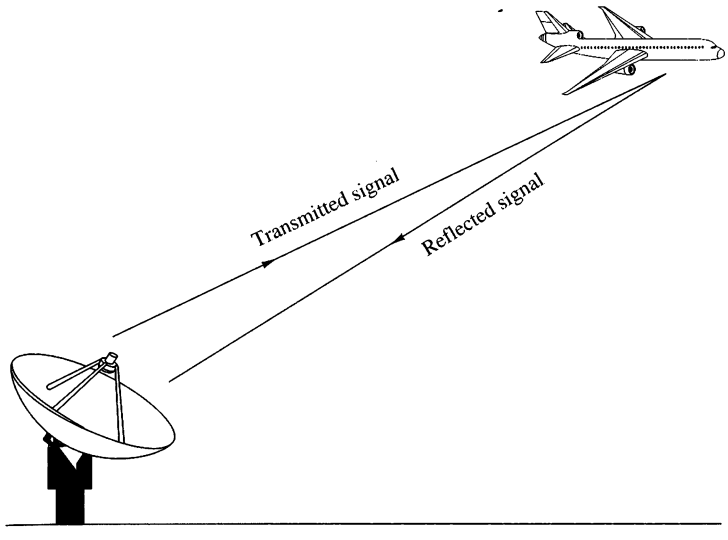


Figure 2.6.1 Radar target detection.

Having the two signal sequences, $x(n)$, which is called the reference signal or transmitted signal, and $y(n)$, the received signal, the problem in radar and sonar detection is to compare $y(n)$ and $x(n)$ to determine if a target is present and, if so, to determine the time delay D and compute the distance to the target. In practice, the signal $x(n - D)$ is heavily corrupted by the additive noise to the point where a visual inspection of $y(n)$ does not reveal the presence or absence of the desired signal reflected from the target. Correlation provides us with a means for extracting this important information from $y(n)$.

Digital communications is another area where correlation is often used. In digital communications the information to be transmitted from one point to another is usually converted to binary form, that is, a sequence of zeros and ones, which are then transmitted to the intended receiver. To transmit a 0 we can transmit the signal sequence $x_0(n)$ for $0 \leq n \leq L - 1$, and to transmit a 1 we can transmit the signal sequence $x_1(n)$ for $0 \leq n \leq L-1$, where L is some integer that denotes the number of samples in each of the two sequences. Very often, $x_1(n)$ is selected to be the negative of $x_0(n)$. The signal received by the intended receiver may be represented as

$$y(n) = x_i(n) + w(n), \qquad i = 0, 1, \qquad 0 \leq n \leq L - 1 \tag{2.6.2}$$

where now the uncertainty is whether $x_0(n)$ or $x_1(n)$ is the signal component in $y(n)$, and $w(n)$ represents the additive noise and other interference inherent in any communication system. Again, such noise has its origin in the electronic components contained in the front end of the receiver. In any case, the receiver knows the possible transmitted sequences $x_0(n)$ and $x_1(n)$ and is faced with the task of comparing the received signal $y(n)$ with both $x_0(n)$ and $x_1(n)$ to determine which of the two signals better matches $y(n)$. This comparison process is performed by means of the correlation operation described in the following subsection.

2.6.1 Crosscorrelation and Autocorrelation Sequences

Suppose that we have two real signal sequences $x(n)$ and $y(n)$ each of which has finite energy. The *crosscorrelation* of $x(n)$ and $y(n)$ is a sequence $r_{xy}(l)$, which is defined as

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n)y(n-l), \qquad l = 0, \pm 1, \pm 2, \ldots \tag{2.6.3}$$

or, equivalently, as

$$r_{xy}(l) = \sum_{n=-\infty}^{\infty} x(n+l)y(n), \qquad l = 0, \pm 1, \pm 2, \ldots \tag{2.6.4}$$

The index l is the (time) shift (or *lag*) parameter and the subscripts xy on the crosscorrelation sequence $r_{xy}(l)$ indicate the sequences being correlated. The order of the subscripts, with x preceding y, indicates the direction in which one sequence is shifted, relative to the other. To elaborate, in (2.6.3), the sequence $x(n)$ is left unshifted and $y(n)$ is shifted by l units in time, to the right for l positive and to the left for l negative. Equivalently, in (2.6.4), the sequence $y(n)$ is left unshifted and $x(n)$ is shifted by l units in time, to the left for l positive and to the right for l negative. But shifting $x(n)$ to the left by l units relative to $y(n)$ is equivalent to shifting $y(n)$ to the right by l units relative to $x(n)$. Hence the computations (2.6.3) and (2.6.4) yield identical crosscorrelation sequences.

If we reverse the roles of $x(n)$ and $y(n)$ in (2.6.3) and (2.6.4) and therefore reverse the order of the indices xy, we obtain the crosscorrelation sequence

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n)x(n-l) \tag{2.6.5}$$

or, equivalently,

$$r_{yx}(l) = \sum_{n=-\infty}^{\infty} y(n+l)x(n) \tag{2.6.6}$$

By comparing (2.6.3) with (2.6.6) or (2.6.4) with (2.6.5), we conclude that

$$r_{xy}(l) = r_{yx}(-l) \tag{2.6.7}$$

Therefore, $r_{yx}(l)$ is simply the folded version of $r_{xy}(l)$, where the folding is done with respect to $l = 0$. Hence, $r_{yx}(l)$ provides exactly the same information as $r_{xy}(l)$, with respect to the similarity of $x(n)$ to $y(n)$.

EXAMPLE 2.6.1

Determine the crosscorrelation sequence $r_{xy}(l)$ of the sequences

$$x(n) = \{\ldots, 0, 0, 2, -1, 3, 7, \underset{\uparrow}{1}, 2, -3, 0, 0, \ldots\}$$

$$y(n) = \{\ldots, 0, 0, 1, -1, 2, -2, \underset{\uparrow}{4}, 1, -2, 5, 0, 0, \ldots\}$$

Solution. Let us use the definition in (2.6.3) to compute $r_{xy}(l)$. For $l = 0$ we have

$$r_{xy}(0) = \sum_{n=-\infty}^{\infty} x(n)y(n)$$

The product sequence $v_0(n) = x(n)y(n)$ is

$$v_0(n) = \{\ldots, 0, 0, 2, 1, 6, -14, \underset{\uparrow}{4}, 2, 6, 0, 0, \ldots\}$$

and hence the sum over all values of n is

$$r_{xy}(0) = 7$$

For $l > 0$, we simply shift $y(n)$ to the right relative to $x(n)$ by l units, compute the product sequence $v_l(n) = x(n)y(n-l)$, and finally, sum over all values of the product sequence. Thus we obtain

$$r_{xy}(1) = 13, \quad r_{xy}(2) = -18, \quad r_{xy}(3) = 16, \quad r_{xy}(4) = -7$$
$$r_{xy}(5) = 5, \quad r_{xy}(6) = -3, \quad r_{xy}(l) = 0, \quad l \geq 7$$

For $l < 0$, we shift $y(n)$ to the left relative to $x(n)$ by l units, compute the product sequence $v_l(n) = x(n)y(n-l)$, and sum over all values of the product sequence. Thus we obtain the values of the crosscorrelation sequence

$$r_{xy}(-1) = 0, \quad r_{xy}(-2) = 33, \quad r_{xy}(-3) = -14, \quad r_{xy}(-4) = 36$$
$$r_{xy}(-5) = 19, \quad r_{xy}(-6) = -9, \quad r_{xy}(-7) = 10, \quad r_{xy}(l) = 0, \ l \leq -8$$

Therefore, the crosscorrelation sequence of $x(n)$ and $y(n)$ is

$$r_{xy}(l) = \{10, -9, 19, 36, -14, 33, 0, \underset{\uparrow}{7}, 13, -18, 16, -7, 5, -3\}$$

The similarities between the computation of the crosscorrelation of two sequences and the convolution of two sequences is apparent. In the computation of convolution, one of the sequences is folded, then shifted, then multiplied by the other sequence to form the product sequence for that shift, and finally, the values of the product sequence are summed. Except for the folding operation, the computation of the crosscorrelation sequence involves the same operations: shifting one of the sequences, multiplying the two sequences, and summing over all values of the product sequence. Consequently, if we have a computer program that performs convolution, we can use it to perform crosscorrelation by providing as inputs to the program the sequence $x(n)$ and the folded sequence $y(-n)$. Then the convolution of $x(n)$ with $y(-n)$ yields the crosscorrelation $r_{xy}(l)$, that is,

$$r_{xy}(l) = x(l) * y(-l) \tag{2.6.8}$$

We note that the absence of folding makes crosscorrelation a noncommutative operation. In the special case where $y(n) = x(n)$, we have the *autocorrelation* of $x(n)$, which is defined as the sequence

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) \tag{2.6.9}$$

or, equivalently, as

$$r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n+l)x(n) \tag{2.6.10}$$

In dealing with finite-duration sequences, it is customary to express the autocorrelation and crosscorrelation in terms of the finite limits on the summation. In particular, if $x(n)$ and $y(n)$ are causal sequences of length N [i.e., $x(n) = y(n) = 0$ for $n < 0$ and $n \geq N$], the crosscorrelation and autocorrelation sequences may be expressed as

$$r_{xy}(l) = \sum_{n=l}^{N-|k|-1} x(n)y(n-l) \tag{2.6.11}$$

and

$$r_{xx}(l) = \sum_{n=i}^{N-|k|-1} x(n)x(n-l) \tag{2.6.12}$$

where $i = l$, $k = 0$ for $l \geq 0$, and $i = 0$, $k = l$ for $l < 0$.

2.6.2 Properties of the Autocorrelation and Crosscorrelation Sequences

The autocorrelation and crosscorrelation sequences have a number of important properties that we now present. To develop these properties, let us assume that we have two sequences $x(n)$ and $y(n)$ with finite energy from which we form the linear combination,

$$ax(n) + by(n-l)$$

where a and b are arbitrary constants and l is some time shift. The energy in this signal is

$$\begin{aligned}\sum_{n=-\infty}^{\infty} [ax(n) + by(n-l)]^2 &= a^2 \sum_{n=-\infty}^{\infty} x^2(n) + b^2 \sum_{n=-\infty}^{\infty} y^2(n-l) \\ &\quad + 2ab \sum_{n=-\infty}^{\infty} x(n)y(n-l) \\ &= a^2 r_{xx}(0) + b^2 r_{yy}(0) + 2ab r_{xy}(l)\end{aligned} \tag{2.6.13}$$

First, we note that $r_{xx}(0) = E_x$ and $r_{yy}(0) = E_y$, which are the energies of $x(n)$ and $y(n)$, respectively. It is obvious that

$$a^2 r_{xx}(0) + b^2 r_{yy}(0) + 2ab r_{xy}(l) \geq 0 \tag{2.6.14}$$

Now, assuming that $b \neq 0$, we can divide (2.6.14) by b^2 to obtain

$$r_{xx}(0)\left(\frac{a}{b}\right)2 + 2r_{xy}(l)\left(\frac{a}{b}\right) + r_{yy}(0) \geq 0$$

We view this equation as a quadratic with coefficients $r_{xx}(0)$, $2r_{xy}(l)$, and $r_{yy}(0)$. Since the quadratic is nonnegative, it follows that the discriminant of this quadratic must be nonpositive, that is,

$$4[r_{xy}^2(l) - r_{xx}(0)r_{yy}(0)] \leq 0$$

Therefore, the crosscorrelation sequence satisfies the condition that

$$|r_{xy}(l)| \leq \sqrt{r_{xx}(0)r_{yy}(0)} = \sqrt{E_x E_y} \tag{2.6.15}$$

In the special case where $y(n) = x(n)$, (2.6.15) reduces to

$$|r_{xx}(l)| \leq r_{xx}(0) = E_x \tag{2.6.16}$$

This means that the autocorrelation sequence of a signal attains its maximum value at zero lag. This result is consistent with the notion that a signal matches perfectly with itself at zero shift. In the case of the crosscorrelation sequence, the upper bound on its values is given in (2.6.15).

Note that if any one or both of the signals involved in the crosscorrelation are scaled, the shape of the crosscorrelation sequence does not change; only the amplitudes of the crosscorrelation sequence are scaled accordingly. Since scaling is unimportant, it is often desirable, in practice, to normalize the autocorrelation and crosscorrelation sequences to the range from -1 to 1. In the case of the autocorrelation sequence, we can simply divide by $r_{xx}(0)$. Thus the normalized autocorrelation sequence is defined as

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)} \tag{2.6.17}$$

Similarly, we define the normalized crosscorrelation sequence

$$\rho_{xy}(l) = \frac{r_{xy}(l)}{\sqrt{r_{xx}(0)r_{yy}(0)}} \tag{2.6.18}$$

Now $|\rho_{xx}(l)| \leq 1$ and $|\rho_{xy}(l)| \leq 1$, and hence these sequences are independent of signal scaling.

Finally, as we have already demonstrated, the crosscorrelation sequence satisfies the property

$$r_{xy}(l) = r_{yx}(-l)$$

With $y(n) = x(n)$, this relation results in the following important property for the autocorrelation sequence

$$r_{xx}(l) = r_{xx}(-l) \tag{2.6.19}$$

Hence the autocorrelation function is an even function. Consequently, it suffices to compute $r_{xx}(l)$ for $l \geq 0$.

EXAMPLE 2.6.2

Compute the autocorrelation of the signal

$$x(n) = a^n u(n), 0 < a < 1$$

Solution. Since $x(n)$ is an infinite-duration signal, its autocorrelation also has infinite duration. We distinguish two cases.

If $l \geq 0$, from Fig. 2.6.2 we observe that

$$r_{xx}(l) = \sum_{n=1}^{\infty} x(n)x(n-l) = \sum_{n=1}^{\infty} a^n a^{n-l} = a^{-l} \sum_{n=1}^{\infty} (a^2)^n$$

Since $a < 1$, the infinite series *converges* and we obtain

$$r_{xx}(l) = \frac{1}{1-a^2} a^{|l|}, \qquad l \geq 0$$

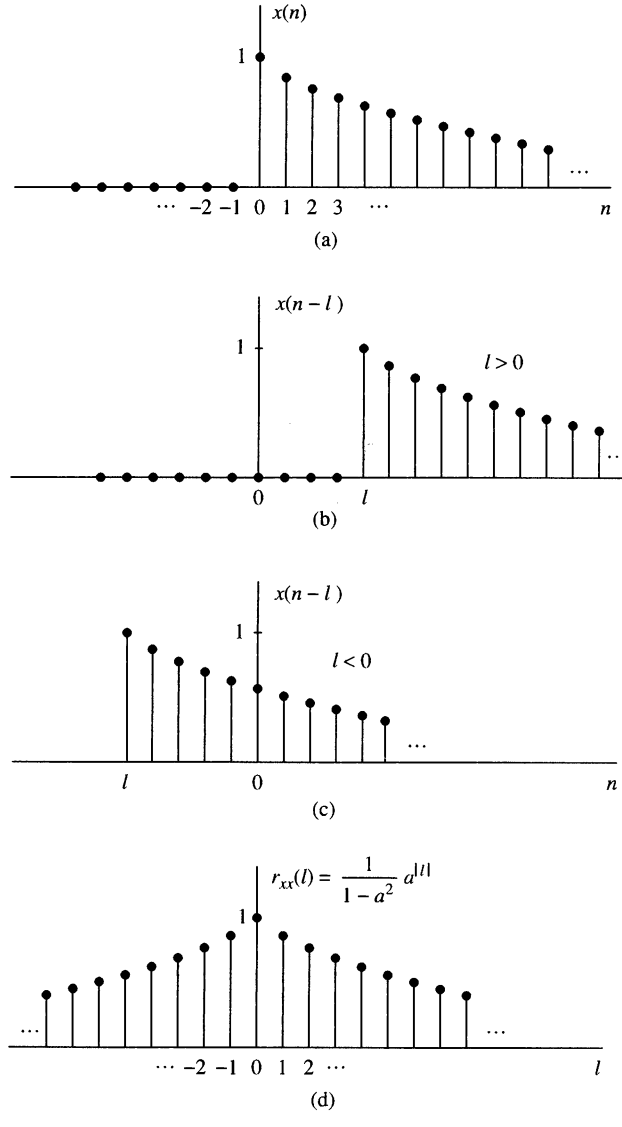


Figure 2.6.2 Computation of the autocorrelation of the signal $x(n) = a^n$, $0 < a < 1$.

For $l < 0$ we have

$$r_{xx}(l) = \sum_{n=0}^{\infty} x(n)x(n-l) = a^{-l}\sum_{n=0}^{\infty}(a^2)^n = \frac{1}{1-a^2}a^{-l}, \qquad l < 0$$

But when l is negative, $a^{-l} = a^{|l|}$. Thus the two relations for $r_{xx}(l)$ can be combined into the following expression:

$$r_{xx}(l) = \frac{1}{1-a^2}a^{|l|}, \qquad -\infty < l < \infty \tag{2.6.20}$$

The sequence $r_{xx}(l)$ is shown in Fig. 2.6.2(d). We observe that

$$r_{xx}(-l) = r_{xx}(l)$$

and

$$r_{xx}(0) = \frac{1}{1-a^2}$$

Therefore, the normalized autocorrelation sequence is

$$\rho_{xx}(l) = \frac{r_{xx}(l)}{r_{xx}(0)} = a^{|l|}, \qquad -\infty < l < \infty \tag{2.6.21}$$

2.6.3 Correlation of Periodic Sequences

In Section 2.6.1 we defined the crosscorrelation and autocorrelation sequences of energy signals. In this section we consider the correlation sequences of power signals and, in particular, periodic signals.

Let $x(n)$ and $y(n)$ be two power signals. Their crosscorrelation sequence is defined as

$$r_{xy}(l) = \lim_{M\to\infty}\frac{1}{2M+1}\sum_{n=-M}^{M} x(n)y(n-l) \tag{2.6.22}$$

If $x(n) = y(n)$, we have the definition of the autocorrelation sequence of a power signal as

$$r_{xx}(l) = \lim_{M\to\infty}\frac{1}{2M+1}\sum_{n=-M}^{M} x(n)x(n-l) \tag{2.6.23}$$

In particular, if $x(n)$ and $y(n)$ are two periodic sequences, each with period N, the averages indicated in (2.6.22) and (2.6.23) over the infinite interval are identical to the averages over a single period, so that (2.6.22) and (2.6.23) reduce to

$$r_{xy}(l) = \frac{1}{N}\sum_{n=0}^{N-1} x(n)y(n-l) \tag{2.6.24}$$

and

$$r_{xx}(l) = \frac{1}{N} \sum_{n=0}^{N-1} x(n)x(n-l) \tag{2.6.25}$$

It is clear that $r_{xy}(l)$ and $r_{xx}(l)$ are periodic correlation sequences with period N. The factor $1/N$ can be viewed as a normalization scale factor.

In some practical applications, correlation is used to identify periodicities in an observed physical signal which may be corrupted by random interference. For example, consider a signal sequence $y(n)$ of the form

$$y(n) = x(n) + w(n) \tag{2.6.26}$$

where $x(n)$ is a periodic sequence of some unknown period N and $w(n)$ represents an additive random interference. Suppose that we observe M samples of $y(n)$, say $0 \leq n \leq M-1$, where $M >> N$. For all practical purposes, we can assume that $y(n) = 0$ for $n < 0$ and $n \geq M$. Now the autocorrelation sequence of $y(n)$, using the normalization factor of $1/M$, is

$$r_{yy}(l) = \frac{1}{M} \sum_{n=0}^{M-1} y(n)y(n-l) \tag{2.6.27}$$

If we substitute for $y(n)$ from (2.6.26) into (2.6.27) we obtain

$$\begin{aligned} r_{yy}(l) &= \frac{1}{M} \sum_{n=0}^{M-1} [x(n) + w(n)][x(n-l) + w(n-l)] \\ &= \frac{1}{M} \sum_{n=0}^{M-1} x(n)x(n-l) \\ &\quad + \frac{1}{M} \sum_{n=0}^{M-1} [x(n)w(n-l) + w(n)x(n-l)] \\ &\quad + \frac{1}{M} \sum_{n=0}^{M-1} w(n)w(n-l) \\ &= r_{xx}(l) + r_{xw}(l) + r_{wx}(l) + r_{ww}(l) \end{aligned} \tag{2.6.28}$$

The first factor on the right-hand side of (2.6.28) is the autocorrelation sequence of $x(n)$. Since $x(n)$ is periodic, its autocorrelation sequence exhibits the same periodicity, thus containing relatively large peaks at $l = 0, N, 2N$, and so on. However, as the shift l approaches M, the peaks are reduced in amplitude due to the fact that we have a finite data record of M samples so that many of the products $x(n)x(n-l)$ are zero. Consequently, we should avoid computing $r_{yy}(l)$ for large lags, say, $l > M/2$.

The crosscorrelations $r_{xw}(l)$ and $r_{wx}(l)$ between the signal $x(n)$ and the additive random interference are expected to be relatively small as a result of the expectation that $x(n)$ and $w(n)$ will be totally unrelated. Finally, the last term on the right-hand side of (2.6.28) is the autocorrelation sequence of the random sequence $w(n)$. This correlation sequence will certainly contain a peak at $l = 0$, but because of its random characteristics, $r_{ww}(l)$ is expected to decay rapidly toward zero. Consequently, only $r_{xx}(l)$ is expected to have large peaks for $l > 0$. This behavior allows us to detect the presence of the periodic signal $x(n)$ buried in the interference $w(n)$ and to identify its period.

An example that illustrates the use of autocorrelation to identify a hidden periodicity in an observed physical signal is shown in Fig. 2.6.3. This figure illustrates the autocorrelation (normalized) sequence for the Wölfer sunspot numbers in the 100-year period 1770–1869 for $0 \leq l \leq 20$, where any value of l corresponds to one year. There is clear evidence in this figure that a periodic trend exists, with a period of 10 to 11 years.

EXAMPLE 2.6.3

Suppose that a signal sequence $x(n) = \sin(\pi/5)n$, for $0 \leq n \leq 99$ is corrupted by an additive noise sequence $w(n)$, where the values of the additive noise are selected independently from sample to sample, from a uniform distribution over the range $(-\Delta/2, \Delta/2)$, where Δ is a parameter of the distribution. The observed sequence is $y(n) = x(n) + w(n)$. Determine the autocorrelation sequence $r_{yy}(l)$ and thus determine the period of the signal $x(n)$.

Solution. The assumption is that the signal sequence $x(n)$ has some unknown period that we are attempting to determine from the noise-corrupted observations $\{y(n)\}$. Although $x(n)$ is periodic with period 10, we have only a finite-duration sequence of length $M = 100$ [i.e., 10 periods of $x(n)$]. The noise power level P_w in the sequence $w(n)$ is determined by the parameter Δ. We simply state that $P_w = \Delta^2/12$. The signal power level is $P_x = \frac{1}{2}$. Therefore, the signal-to-noise ratio (SNR) is defined as

$$\frac{P_x}{P_w} = \frac{\frac{1}{2}}{\Delta^2/12} = \frac{6}{\Delta^2}$$

Usually, the SNR is expressed on a logarithmic scale in decibels (dB) as $10 \log_{10} (P_x/P_w)$.

Figure 2.6.4 illustrates a sample of a noise sequence $w(n)$, and the observed sequence $y(n) = x(n) + w(n)$ when the SNR = 1 dB. The autocorrelation sequence $r_{yy}(l)$ is illustrated in Fig. 2.6.4(c). We observe that the periodic signal $x(n)$, embedded in $y(n)$, results in a periodic autocorrelation function $r_{xx}(l)$ with period $N = 10$. The effect of the additive noise is to add to the peak value at $l = 0$, but for $l \neq 0$, the correlation sequence $r_{ww}(l) \approx 0$ as a result of the fact that values of $w(n)$ were generated independently. Such noise is usually called *white noise*. The presence of this noise explains the reason for the large peak at $l = 0$. The smaller, nearly equal peaks at $l = \pm 10, \pm 20, \ldots$ are due to the periodic characteristics of $x(n)$.

2.6.4 Input–Output Correlation Sequences

In this section we derive two input–output relationships for LTI systems in the "correlation domain." Let us assume that a signal $x(n)$ with known autocorrelation $r_{xx}(l)$

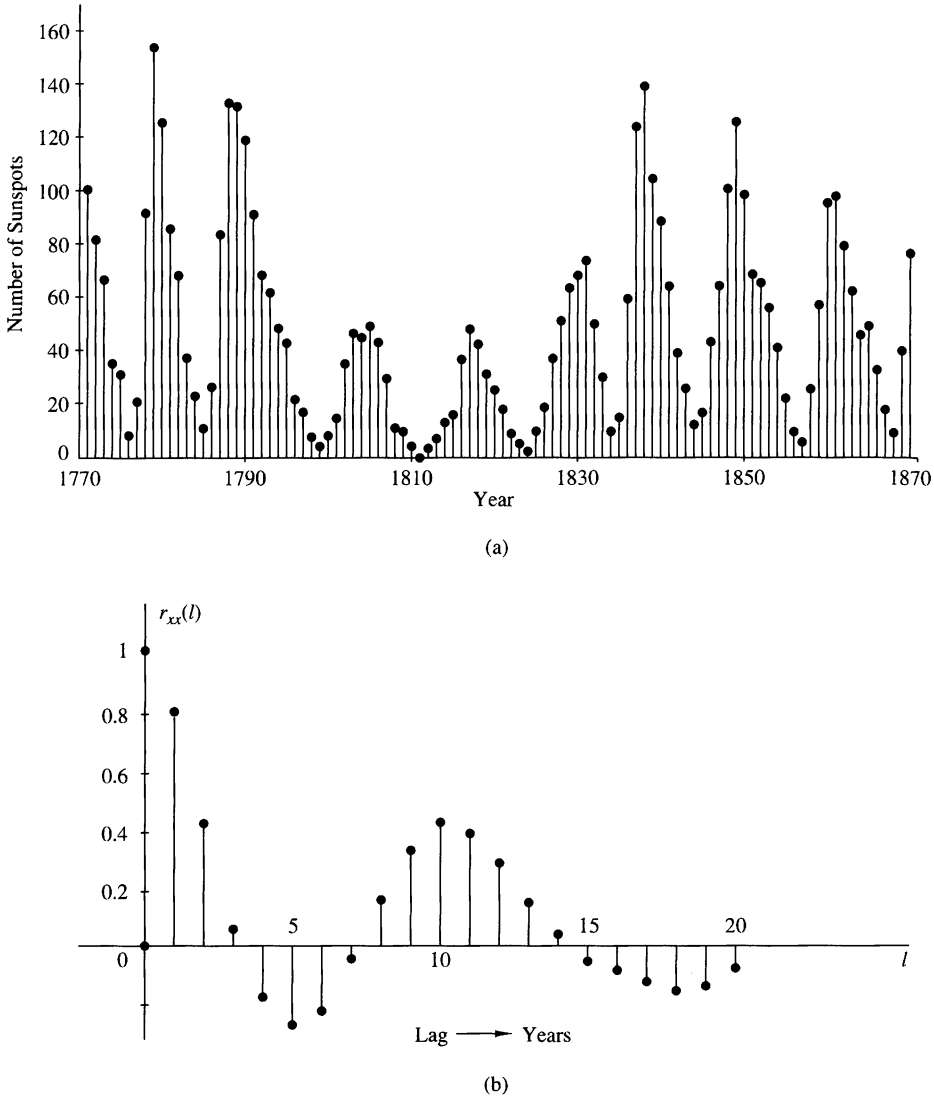


Figure 2.6.3 Identification of periodicity in the Wölfer sunspot numbers: (a) annual Wölfer sunspot numbers; (b) normalized autocorrelation sequence.

is applied to an LTI system with impulse response $h(n)$, producing the output signal

$$y(n) = h(n) * x(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k)$$

The crosscorrelation between the output and the input signal is

$$r_{yx}(l) = y(l) * x(-l) = h(l) * [x(l) * x(-l)]$$

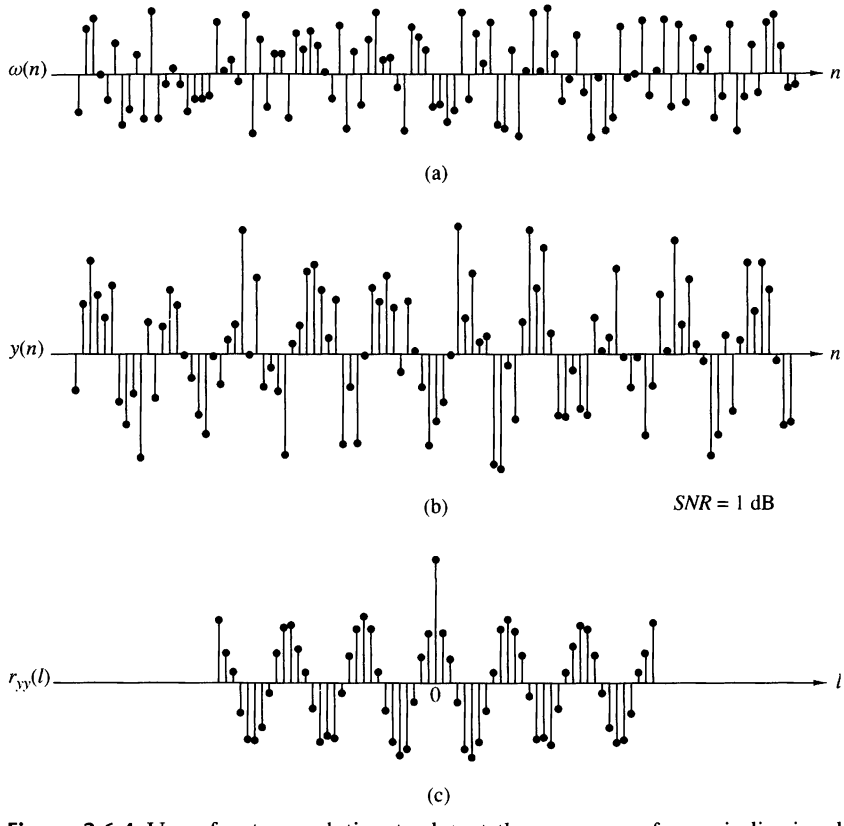


Figure 2.6.4 Use of autocorrelation to detect the presence of a periodic signal corrupted by noise.

or

$$r_{yx}(l) = h(l) * r_{xx}(l) \tag{2.6.29}$$

where we have used (2.6.8) and the properties of convolution. Hence the cross-correlation between the input and the output of the system is the convolution of the impulse response with the autocorrelation of the input sequence. Alternatively, $r_{yx}(l)$ may be viewed as the output of the LTI system when the input sequence is $r_{xx}(l)$. This is illustrated in Fig. 2.6.5. If we replace l by $-l$ in (2.6.29), we obtain

$$r_{xy}(l) = h(-l) * r_{xx}(l)$$

The autocorrelation of the output signal can be obtained by using (2.6.8) with $x(n) = y(n)$ and the properties of convolution. Thus we have

$$\begin{aligned} r_{yy}(l) &= y(l) * y(-l) \\ &= [h(l) * x(l)] * [h(-l) * x(-l)] \\ &= [h(l) * h(-l)] * [x(l) * x(-l)] \\ &= r_{hh}(l) * r_{xx}(l) \end{aligned} \tag{2.6.30}$$

The autocorrelation $r_{hh}(l)$ of the impulse response $h(n)$ exists if the system is stable. Furthermore, the stability insures that the system does not change the type (energy or power) of the input signal. By evaluating (2.6.30) for $l = 0$ we obtain

$$r_{yy}(0) = \sum_{k=-\infty}^{\infty} r_{hh}(k) r_{xx}(k) \tag{2.6.31}$$

which provides the energy (or power) of the output signal in terms of autocorrelations. These relationships hold for both energy and power signals. The direct derivation of these relationships for energy and power signals, and their extensions to complex signals, are left as exercises for the student.

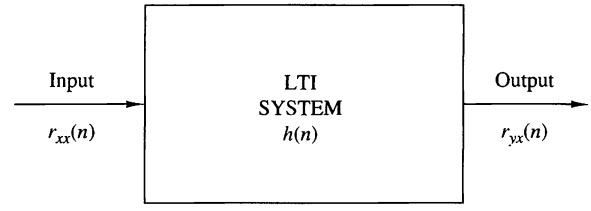


Figure 2.6.5
Input–output relation for crosscorrelation $r_{yx}(n)$.

2.7 Summary and References

The major theme of this chapter is the characterization of discrete-time signals and systems in the time domain. Of particular importance is the class of linear time-invariant (LTI) systems which are widely used in the design and implementation of digital signal processing systems. We characterized LTI systems by their unit sample response $h(n)$ and derived the convolution summation, which is a formula for determining the response $y(n)$ of the system characterized by $h(n)$ to any given input sequence $x(n)$.

The class of LTI systems characterized by linear difference equations with constant coefficients is by far the most important of the LTI systems in the theory and application of digital signal processing. The general solution of a linear difference equation with constant coefficients was derived in this chapter and shown to consist of two components: the solution of the homogeneous equation, which represents the natural response of the system when the input is zero, and the particular solution, which represents the response of the system to the input signal. From the difference equation, we also demonstrated how to derive the unit sample response of the LTI system.

Linear time-invariant systems were generally subdivided into FIR (finite-duration impulse response) and IIR (infinite-duration impulse response) depending on whether $h(n)$ has finite duration or infinite duration, respectively. The realizations of such systems were briefly described. Furthermore, in the realization of FIR systems, we made the distinction between recursive and nonrecursive realizations. On the other hand, we observed that IIR systems can be implemented recursively, only.

There are a number of texts on discrete-time signals and systems. We mention as examples the books by McGillem and Cooper (1984), Oppenheim and Willsky (1983), and Siebert (1986). Linear constant-coefficient difference equations are treated in depth in the books by Hildebrand (1952) and Levy and Lessman (1961).

The last topic in this chapter, on correlation of discrete-time signals, plays an important role in digital signal processing, especially in applications dealing with digital communications, radar detection and estimation, sonar, and geophysics. In our treatment of correlation sequences, we avoided the use of statistical concepts. Correlation is simply defined as a mathematical operation between two sequences, which produces another sequence, called either the *crosscorrelation sequence* when the two sequences are different, or the *autocorrelation sequence* when the two sequences are identical.

In practical applications in which correlation is used, one (or both) of the sequences is (are) contaminated by noise and, perhaps, by other forms of interference. In such a case, the noisy sequence is called a *random sequence* and is characterized in statistical terms. The corresponding correlation sequence becomes a function of the statistical characteristics of the noise and any other interference.

The statistical characterization of sequences and their correlation is treated in Chapter 12. Supplementary reading on probabilistic and statistical concepts dealing with correlation can be found in the books by Davenport (1970), Helstrom (1990), Peebles (1987), and Stark and Woods (1994).

Problems

2.1 A discrete-time signal $x(n)$ is defined as

$$x(n) = \begin{cases} 1 + \frac{n}{3}, & -3 \leq n \leq -1 \\ 1, & 0 \leq n \leq 3 \\ 0, & \text{elsewhere} \end{cases}$$

(a) Determine its values and sketch the signal $x(n)$.

(b) Sketch the signals that result if we:

1. First fold $x(n)$ and then delay the resulting signal by four samples.
2. First delay $x(n)$ by four samples and then fold the resulting signal.

(c) Sketch the signal $x(-n + 4)$.

(d) Compare the results in parts (b) and (c) and derive a rule for obtaining the signal $x(-n + k)$ from $x(n)$.

(e) Can you express the signal $x(n)$ in terms of signals $\delta(n)$ and $u(n)$?

2.2 A discrete-time signal $x(n)$ is shown in Fig. P2.2. Sketch and label carefully each of the following signals.

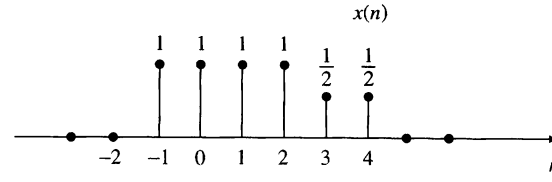


Figure P2.2

(a) $x(n-2)$ **(b)** $x(4-n)$ **(c)** $x(n+2)$ **(d)** $x(n)u(2-n)$ **(e)** $x(n-1)\delta(n-3)$
(f) $x(n^2)$ **(g)** even part of $x(n)$ **(h)** odd part of $x(n)$

2.3 Show that

(a) $\delta(n) = u(n) - u(n-1)$

(b) $u(n) = \sum_{k=-\infty}^{n} \delta(k) = \sum_{k=0}^{\infty} \delta(n-k)$

2.4 Show that any signal can be decomposed into an even and an odd component. Is the decomposition unique? Illustrate your arguments using the signal

$$x(n) = \{2, 3, \underset{\uparrow}{4}, 5, 6\}$$

2.5 Show that the energy (power) of a real-valued energy (power) signal is equal to the sum of the energies (powers) of its even and odd components.

2.6 Consider the system

$$y(n) = \mathcal{T}[x(n)] = x(n^2)$$

(a) Determine if the system is time invariant.

(b) To clarify the result in part (a) assume that the signal

$$x(n) = \begin{cases} 1, & 0 \le n \le 3 \\ 0, & \text{elsewhere} \end{cases}$$

is applied into the system.

(1) Sketch the signal $x(n)$.

(2) Determine and sketch the signal $y(n) = \mathcal{T}[x(n)]$.

(3) Sketch the signal $y_2'(n) = y(n-2)$.

(4) Determine and sketch the signal $x_2(n) = x(n-2)$.

(5) Determine and sketch the signal $y_2(n) = \mathcal{T}[x_2(n)]$.

(6) Compare the signals $y_2(n)$ and $y(n-2)$. What is your conclusion?

(c) Repeat part (b) for the system

$$y(n) = x(n) - x(n-1)$$

Can you use this result to make any statement about the time invariance of this system? Why?

(d) Repeat parts (b) and (c) for the system

$$y(n) = \mathcal{T}[x(n)] = nx(n)$$

2.7 A discrete-time system can be

(1) Static or dynamic
(2) Linear or nonlinear
(3) Time invariant or time varying
(4) Causal or noncausal
(5) Stable or unstable

Examine the following systems with respect to the properties above.

(a) $y(n) = \cos[x(n)]$

(b) $y(n) = \sum_{k=-\infty}^{n+1} x(k)$

(c) $y(n) = x(n)\cos(\omega_0 n)$

(d) $y(n) = x(-n+2)$

(e) $y(n) = \text{Trun}[x(n)]$, where $\text{Trun}[x(n)]$ denotes the integer part of $x(n)$, obtained by truncation

(f) $y(n) = \text{Round}[x(n)]$, where $\text{Round}[x(n)]$ denotes the integer part of $x(n)$ obtained by rounding

Remark: The systems in parts (e) and (f) are quantizers that perform truncation and rounding, respectively.

(g) $y(n) = |x(n)|$

(h) $y(n) = x(n)u(n)$

(i) $y(n) = x(n) + nx(n+1)$

(j) $y(n) = x(2n)$

(k) $y(n) = \begin{cases} x(n), & \text{if } x(n) \geq 0 \\ 0, & \text{if } x(n) < 0 \end{cases}$

(l) $y(n) = x(-n)$

(m) $y(n) = \text{sign}[x(n)]$

(n) The ideal sampling system with input $x_a(t)$ and output $x(n) = x_a(nT)$, $-\infty < n < \infty$

2.8 Two discrete-time systems $\mathcal{T}_1$ and $\mathcal{T}_2$ are connected in cascade to form a new system $\mathcal{T}$ as shown in Fig. P2.8. Prove or disprove the following statements.

(a) If $\mathcal{T}_1$ and $\mathcal{T}_2$ are linear, then $\mathcal{T}$ is linear (i.e., the cascade connection of two linear systems is linear).

(b) If $\mathcal{T}_1$ and $\mathcal{T}_2$ are time invariant, then $\mathcal{T}$ is time invariant.

(c) If $\mathcal{T}_1$ and $\mathcal{T}_2$ are causal, then $\mathcal{T}$ is causal.

(d) If $\mathcal{T}_1$ and $\mathcal{T}_2$ are linear and time invariant, the same holds for $\mathcal{T}$.

(e) If $\mathcal{T}_1$ and $\mathcal{T}_2$ are linear and time invariant, then interchanging their order does not change the system $\mathcal{T}$.

(f) As in part (e) except that $\mathcal{T}_1$, $\mathcal{T}_2$ are now time varying. (*Hint:* Use an example.)

(g) If $\mathcal{T}_1$ and $\mathcal{T}_2$ are nonlinear, then $\mathcal{T}$ is nonlinear.

(h) If $\mathcal{T}_1$ and $\mathcal{T}_2$ are stable, then $\mathcal{T}$ is stable.

(i) Show by an example that the inverses of parts (c) and (h) do not hold in general.

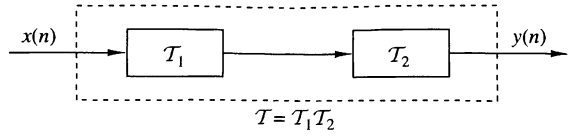


Figure P2.8

2.9 Let $\mathcal{T}$ be an LTI, relaxed, and BIBO stable system with input $x(n)$ and output $y(n)$. Show that:

(a) If $x(n)$ is periodic with period N [i.e., $x(n) = x(n+N)$ for all $n \geq 0$], the output $y(n)$ tends to a periodic signal with the same period.

(b) If $x(n)$ is bounded and tends to a constant, the output will also tend to a constant.

(c) If $x(n)$ is an energy signal, the output $y(n)$ will also be an energy signal.

2.10 The following input–output pairs have been observed during the operation of a time-invariant system:

$$x_1(n) = \{\underset{\uparrow}{1}, 0, 2\} \xleftrightarrow{\mathcal{T}} y_1(n) = \{\underset{\uparrow}{0}, 1, 2\}$$

$$x_2(n) = \{\underset{\uparrow}{0}, 0, 3\} \xleftrightarrow{\mathcal{T}} y_2(n) = \{\underset{\uparrow}{0}, 1, 0, 2\}$$

$$x_3(n) = \{\underset{\uparrow}{0}, 0, 0, 1\} \xleftrightarrow{\mathcal{T}} y_3(n) = \{1, \underset{\uparrow}{2}, 1\}$$

Can you draw any conclusions regarding the linearity of the system. What is the impulse response of the system?

2.11 The following input–output pairs have been observed during the operation of a linear system:

$$x_1(n) = \{-1, \underset{\uparrow}{2}, 1\} \xleftrightarrow{\mathcal{T}} y_1(n) = \{1, \underset{\uparrow}{2}, -1, 0, 1\}$$

$$x_2(n) = \{1, \underset{\uparrow}{-1}, -1\} \xleftrightarrow{\mathcal{T}} y_2(n) = \{-1, \underset{\uparrow}{1}, 0, 2\}$$

$$x_3(n) = \{0, \underset{\uparrow}{1}, 1\} \xleftrightarrow{\mathcal{T}} y_3(n) = \{\underset{\uparrow}{1}, 2, 1\}$$

Can you draw any conclusions about the time invariance of this system?

2.12 The only available information about a system consists of N input–output pairs, of signals $y_i(n) = \mathcal{T}[x_i(n)]$, $i = 1, 2, \ldots, N$.

(a) What is the class of input signals for which we can determine the output, using the information above, if the system is known to be linear?

(b) The same as above, if the system is known to be time invariant.

2.13 Show that the necessary and sufficient condition for a relaxed LTI system to be BIBO stable is

$$\sum_{n=-\infty}^{\infty} |h(n)| \leq M_h < \infty$$

for some constant M_h.

2.14 Show that:

(a) A relaxed linear system is causal if and only if for any input $x(n)$ such that

$$x(n) = 0 \text{ for } n < n_0 \Rightarrow y(n) = 0 \qquad \text{for } n < n_0$$

(b) A relaxed LTI system is causal if and only if

$$h(n) = 0 \qquad \text{for } n < 0$$

2.15

(a) Show that for any real or complex constant a, and any finite integer numbers M and N, we have

$$\sum_{n}^{N} n = Ma^n = \begin{cases} \dfrac{a^M - a^{N+1}}{1-a}, & \text{if } a \neq 1 \\ N - M + 1, & \text{if } a = 1 \end{cases}$$

(b) Show that if $|a| < 1$, then

$$\sum_{n=0}^{\infty} a^n = \frac{1}{1-a}$$

2.16 **(a)** If $y(n) = x(n) * h(n)$, show that $\sum_y = \sum_x \sum_h$, where $\sum_x = \sum_{n=-\infty}^{\infty} x(n)$.

(b) Compute the convolution $y(n) = x(n) * h(n)$ of the following signals and check the correctness of the results by using the test in (a).

(1) $x(n) = \{1, 2, 4\}, h(n) = \{1, 1, 1, 1, 1\}$

(2) $x(n) = \{1, 2, -1\}, h(n) = x(n)$

(3) $x(n) = \{0, 1, -2, 3, -4\}, h(n) = \{\frac{1}{2}, \frac{1}{2}, 1, \frac{1}{2}\}$

(4) $x(n) = \{1, 2, 3, 4, 5\}, h(n) = \{1\}$

(5) $x(n) = \{\underset{\uparrow}{1}, -2, 3\}, h(n) = \{\underset{\uparrow}{0}, 0, 1, 1, 1, 1\}$

(6) $x(n) = \{\underset{\uparrow}{0}, 0, 1, 1, 1, 1\}, h(n) = \{1, \underset{\uparrow}{-2}, 3\}$

(7) $x(n) = \{\underset{\uparrow}{0}, 1, 4, -3\}, h(n) = \{\underset{\uparrow}{1}, 0, -1, -1\}$

(8) $x(n) = \{\underset{\uparrow}{1}, 1, 2\}, h(n) = u(n)$

(9) $x(n) = \{1, 1, \underset{\uparrow}{0}, 1, 1\}, h(n) = \{1, -2, -3, \underset{\uparrow}{4}\}$

(10) $x(n) = \{1, 2, \underset{\uparrow}{0}, 2, 1\} h(n) = x(n)$

(11) $x(n) = (\frac{1}{2})^n u(n), h(n) = (\frac{1}{4})^n u(n)$

2.17 Compute and plot the convolutions $x(n) * h(n)$ and $h(n) * x(n)$ for the pairs of signals shown in Fig. P2.17.

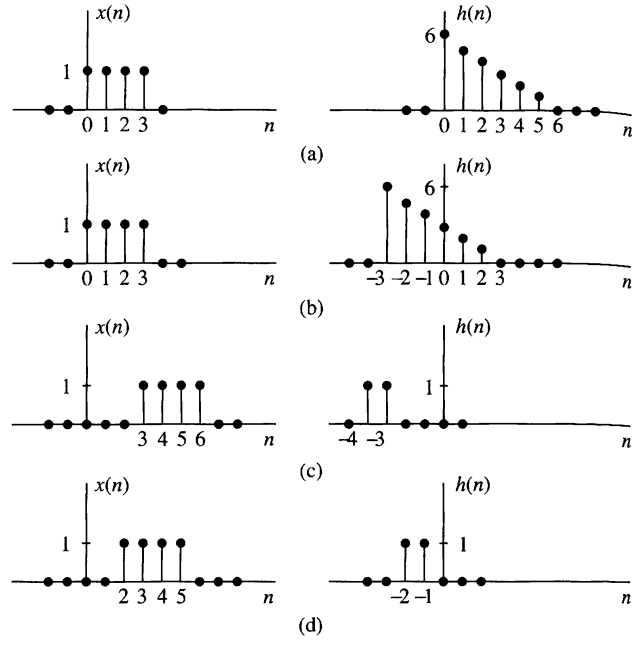


Figure P2.17

2.18 Determine and sketch the convolution $y(n)$ of the signals

$$x(n) = \begin{cases} \frac{1}{3}n, & 0 \le n \le 6 \\ 0, & \text{elsewhere} \end{cases}$$

$$h(n) = \begin{cases} 1, & -2 \le n \le 2 \\ 0, & \text{elsewhere} \end{cases}$$

(a) Graphically

(b) Analytically

2.19 Compute the convolution $y(n)$ of the signals

$$x(n) = \begin{cases} \alpha^n, & -3 \le n \le 5 \\ 0, & \text{elsewhere} \end{cases}$$

$$h(n) = \begin{cases} 1, & 0 \le n \le 4 \\ 0, & \text{elsewhere} \end{cases}$$

2.20 Consider the following three operations.

(a) Multiply the integer numbers: 131 and 122.

(b) Compute the convolution of signals: $\{1, 3, 1\} * \{1, 2, 2\}$.

(c) Multiply the polynomials: $1 + 3z + z^2$ and $1 + 2z + 2z^2$.

(d) Repeat part (a) for the numbers 1.31 and 12.2.

(e) Comment on your results.

Compute the convolution $y(n) = x(n) * h(n)$ of the following pairs of signals.

(a) $x(n) = a^n u(n)$, $h(n) = b^n u(n)$ when $a \neq b$ and when $a = b$

(b) $x(n) = \begin{cases} 1, & n = -2, 0, 1 \\ 2, & n = -1 \\ 0, & \text{elsewhere} \end{cases}$ $h(n) = \delta(n) - \delta(n-1) + \delta(n-4) + \delta(n-5)$

(c) $x(n) = u(n+1) - u(n-4) - \delta(n-5)$; $h(n) = [u(n+2) - u(n-3)] \cdot (3 - |n|)$

(d) $x(n) = u(n) - u(n-5)$; $h(n) = u(n-2) - u(n-8) + u(n-11) - u(n-17)$

Let $x(n)$ be the input signal to a discrete-time filter with impulse response $h_i(n)$ and let $y_i(n)$ be the corresponding output.

(a) Compute and sketch $x(n)$ and $y_i(n)$ in the following cases, using the same scale in all figures.

$$x(n) = \{1, 4, 2, 3, 5, 3, 3, 4, 5, 7, 6, 9\}$$

$$h_1(n) = \{1, 1\}$$

$$h_2(n) = \{1, 2, 1\}$$

$$h_3(n) = \{\frac{1}{2}, \frac{1}{2}\}$$

$$h_4(n) = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$$

$$h_5(n) = \{\frac{1}{4}, -\frac{1}{2}, \frac{1}{4}\}$$

Sketch $x(n)$, $y_1(n)$, $y_2(n)$ on one graph and $x(n)$, $y_3(n)$, $y_4(n)$, $y_5(n)$ on another graph

(b) What is the difference between $y_1(n)$ and $y_2(n)$, and between $y_3(n)$ and $y_4(n)$?

(c) Comment on the smoothness of $y_2(n)$ and $y_4(n)$. Which factors affect the smoothness?

(d) Compare $y_4(n)$ with $y_5(n)$. What is the difference? Can you explain it?

(e) Let $h_6(n) = \{\frac{1}{2}, -\frac{1}{2}\}$. Compute $y_6(n)$. Sketch $x(n)$, $y_2(n)$, and $y_6(n)$ on the same figure and comment on the results.

Express the output $y(n)$ of a linear time-invariant system with impulse response $h(n)$ in terms of its step response $s(n) = h(n)^* u(n)$ and the input $x(n)$.

The discrete-time system

$$y(n) = ny(n-1) + x(n), \qquad n \geq 0$$

is at rest [i.e., $y(-1) = 0$]. Check if the system is linear time invariant and BIBO stable.

Consider the signal $\gamma(n) = a^n u(n)$, $0 < a < 1$.

(a) Show that any sequence $x(n)$ can be decomposed as

$$x(n) = \sum_{n=-\infty}^{\infty} c_k \gamma(n-k)$$

and express c_k in terms of $x(n)$.

(b) Use the properties of linearity and time invariance to express the output $y(n) = \mathcal{T}[x(n)]$ in terms of the input $x(n)$ and the signal $g(n) = \mathcal{T}[\gamma(n)]$, where $\mathcal{T}[\cdot]$ is an LTI system.

(c) Express the impulse response $h(n) = \mathcal{T}[\delta(n)]$ in terms of $g(n)$.

2.26 Determine the zero-input response of the system described by the second-order difference equation

$$x(n) - 3y(n-1) - 4y(n-2) = 0$$

2.27 Determine the particular solution of the difference equation

$$y(n) = \frac{5}{6}y(n-1) - \frac{1}{6}y(n-2) + x(n)$$

when the forcing function is $x(n) = 2^n u(n)$.

2.28 In Example 2.4.8, equation (2.4.30), separate the output sequence $y(n)$ into the transient response and the steady-state response. Plot these two responses for $a_1 = -0.9$.

2.29 Determine the impulse response for the cascade of two linear time-invariant systems having impulse responses.

$$h_1(n) = a^n[u(n) - u(n-N)] \text{ and } h_2(n) = [u(n) - u(n-M)]$$

2.30 Determine the response $y(n)$, $n \geq 0$, of the system described by the second-order difference equation

$$y(n) - 3y(n-1) - 4y(n-2) = x(n) + 2x(n-1)$$

to the input $x(n) = 4^n u(n)$.

2.31 Determine the impulse response of the following causal system:

$$y(n) - 3y(n-1) - 4y(n-2) = x(n) + 2x(n-1)$$

2.32 Let $x(n)$, $N_1 \leq n \leq N_2$ and $h(n)$, $M_1 \leq n \leq M_2$ be two finite-duration signals.

(a) Determine the range $L_1 \leq n \leq L_2$ of their convolution, in terms of N_1, N_2, M_1 and M_2.

(b) Determine the limits of the cases of partial overlap from the left, full overlap, and partial overlap from the right. For convenience, assume that $h(n)$ has shorter duration than $x(n)$.

(c) Illustrate the validity of your results by computing the convolution of the signals

$$x(n) = \begin{cases} 1, & -2 \leq n \leq 4 \\ 0, & \text{elsewhere} \end{cases}$$

$$h(n) = \begin{cases} 2, & -1 \leq n \leq 2 \\ 0, & \text{elsewhere} \end{cases}$$

2.33 Determine the impulse response and the unit step response of the systems described by the difference equation

(a) $y(n) = 0.6y(n-1) - 0.08y(n-2) + x(n)$

(b) $y(n) = 0.7y(n-1) - 0.1y(n-2) + 2x(n) - x(n-2)$

2.34 Consider a system with impulse response

$$h(n) = \begin{cases} (\frac{1}{2})^n, & 0 \le n \le 4 \\ 0, & \text{elsewhere} \end{cases}$$

Determine the input $x(n)$ for $0 \le n \le 8$ that will generate the output sequence

$$y(n) = \{\underset{\uparrow}{1}, 2, 2.5, 3, 3, 3, 2, 1, 0, \ldots\}$$

2.35 Consider the interconnection of LTI systems as shown in Fig. P2.35.

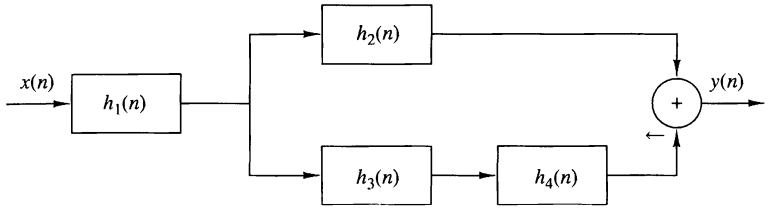


Figure P2.35

(a) Express the overall impulse response in terms of $h_1(n)$, $h_2(n)$, $h_3(n)$, and $h_4(n)$.

(b) Determine $h(n)$ when

$$h_1(n) = \{\frac{1}{2}, \frac{1}{4}, \frac{1}{2}\}$$

$$h_2(n) = h_3(n) = (n+1)u(n)$$

$$h_4(n) = \delta(n-2)$$

(c) Determine the response of the system in part (b) if

$$x(n) = \delta(n+2) + 3\delta(n-1) - 4\delta(n-3)$$

2.36 Consider the system in Fig. P2.36 with $h(n) = a^n u(n)$, $-1 < a < 1$. Determine the response $y(n)$ of the system to the excitation

$$x(n) = u(n+5) - u(n-10)$$

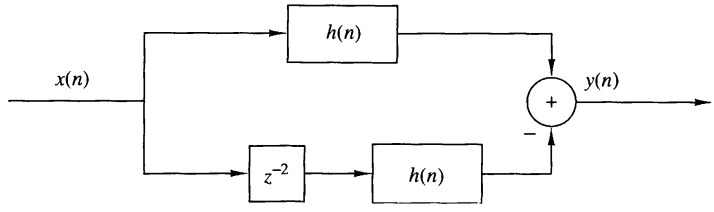


Figure P2.36

2.37 Compute and sketch the step response of the system

$$y(n) = \frac{1}{M} \sum_{k=0}^{M-1} x(n-k)$$

2.38 Determine the range of values of the parameter a for which the linear time-invariant system with impulse response

$$h(n) = \begin{cases} a^n, & n \geq 0, n \text{ even} \\ 0, & \text{otherwise} \end{cases}$$

is stable.

2.39 Determine the response of the system with impulse response

$$h(n) = a^n u(n)$$

to the input signal

$$x(n) = u(n) - u(n-10)$$

(*Hint:* The solution can be obtained easily and quickly by applying the linearity and time-invariance properties to the result in Example 2.3.5.)

2.40 Determine the response of the (relaxed) system characterized by the impulse response

$$h(n) = (\frac{1}{2})^n u(n)$$

to the input signal

$$x(n) = \begin{cases} 1, & 0 \leq n < 10 \\ 0, & \text{otherwise} \end{cases}$$

2.41 Determine the response of the (relaxed) system characterized by the impulse response

$$h(n) = (\frac{1}{2})^n u(n)$$

to the input signals

(a) $x(n) = 2^n u(n)$

(b) $x(n) = u(-n)$

Three systems with impulse responses $h_1(n) = \delta(n) - \delta(n-1)$, $h_2(n) = h(n)$, and $h_3(n) = u(n)$, are connected in cascade.

(a) What is the impulse response, $h_c(n)$, of the overall system?

(b) Does the order of the interconnection affect the overall system?

(a) Prove and explain graphically the difference between the relations

$$x(n)\delta(n-n_0) = x(n_0)\delta(n-n_0) \qquad \text{and} \qquad x(n) * \delta(n-n_0) = x(n-n_0)$$

(b) Show that a discrete-time system, which is described by a convolution summation, is LTI and relaxed,

(c) What is the impulse response of the system described by $y(n) = x(n-n_0)$?

Two signals $s(n)$ and $v(n)$ are related through the following difference equations.

$$s(n) + a_1 s(n-1) + \cdots + a_N s(n-N) = b_0 v(n)$$

Design the block diagram realization of:

(a) The system that generates $s(n)$ when excited by $v(n)$.

(b) The system that generates $v(n)$ when excited by $s(n)$.

(c) What is the impulse response of the cascade interconnection of systems in parts (a) and (b)?

Compute the zero-state response of the system described by the difference equation

$$y(n) + \frac{1}{2}y(n-1) = x(n) + 2x(n-2)$$

to the input

$$x(n) = \{1, 2, \underset{\uparrow}{3}, 4, 2, 1\}$$

by solving the difference equation recursively.

Determine the direct form II realization for each of the following LTI systems:

(a) $2y(n) + y(n-1) - 4y(n-3) = x(n) + 3x(n-5)$

(b) $y(n) = x(n) - x(n-1) + 2x(n-2) - 3x(n-4)$

Consider the discrete-time system shown in Fig. P2.47.

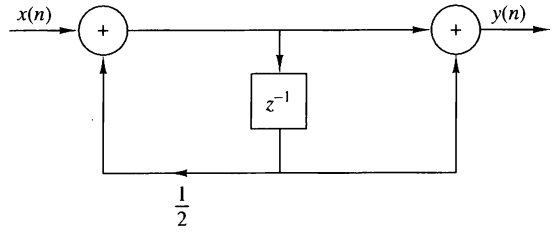


Figure P2.47

(a) Compute the 10 first samples of its impulse response.

(b) Find the input–output relation.

(c) Apply the input $x(n) = \{\underset{\uparrow}{1}, 1, 1, \ldots\}$ and compute the first 10 samples of the output.

(d) Compute the first 10 samples of the output for the input given in part (c) by using convolution.

(e) Is the system causal? Is it stable?

2.48 Consider the system described by the difference equation

$$y(n) = ay(n-1) + bx(n)$$

(a) Determine b in terms of a so that

$$\sum_{n=-\infty}^{\infty} h(n) = 1$$

(b) Compute the zero-state step response $s(n)$ of the system and choose b so that $s(\infty) = 1$.

(c) Compare the values of b obtained in parts (a) and (b). What did you notice?

2.49 A discrete-time system is realized by the structure shown in Fig. P2.49.

(a) Determine the impulse response.

(b) Determine a realization for its inverse system, that is, the system which produces $x(n)$ as an output when $y(n)$ is used as an input.

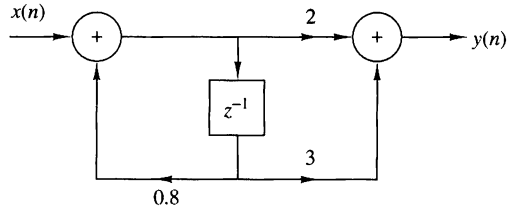


Figure P2.49

2.50 Consider the discrete-time system shown in Fig. P2.50.

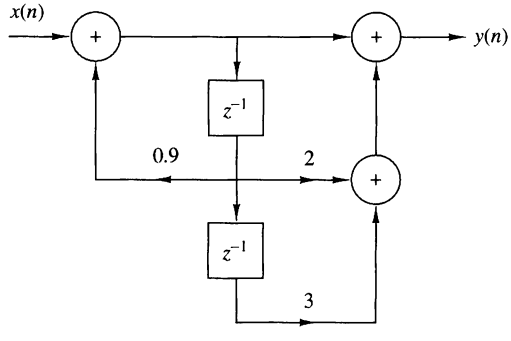


Figure P2.50

(a) Compute the first six values of the impulse response of the system.

(b) Compute the first six values of the zero-state step response of the system.

(c) Determine an analytical expression for the impulse response of the system.

2.51 Determine and sketch the impulse response of the following systems for $n = 0, 1, \ldots, 9$.

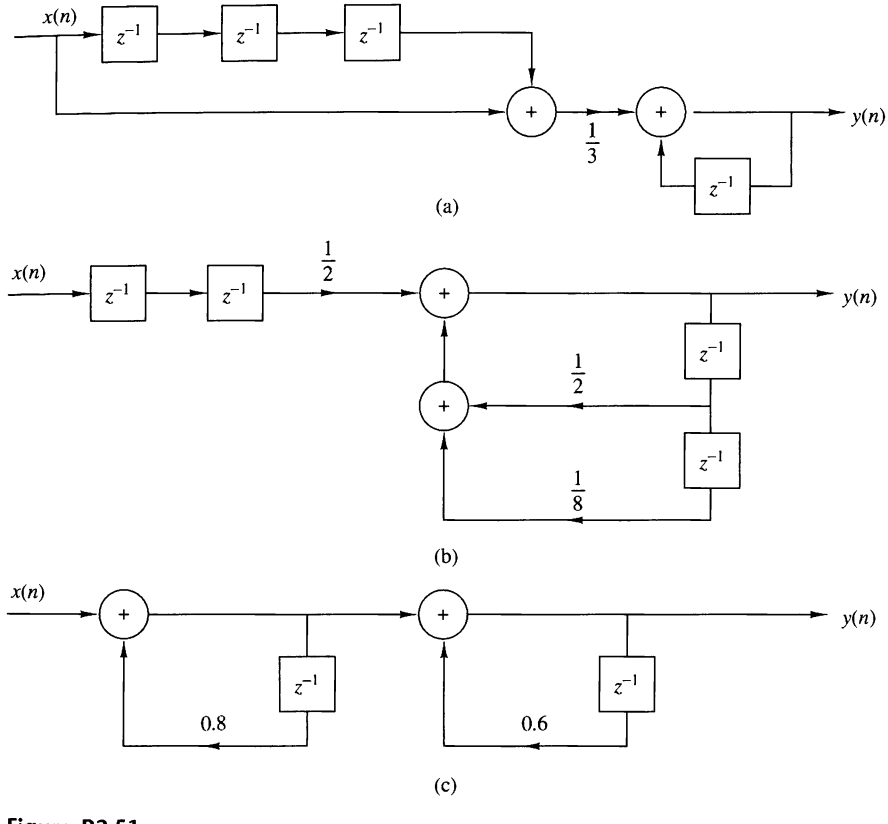


Figure P2.51

(a) Fig. P2.51(a).

(b) Fig. P2.51(b).

(c) Fig. P2.51(c).

(d) Classify the systems above as FIR or IIR.

(e) Find an explicit expression for the impulse response of the system in part (c).

2.52 Consider the systems shown in Fig. P2.52.

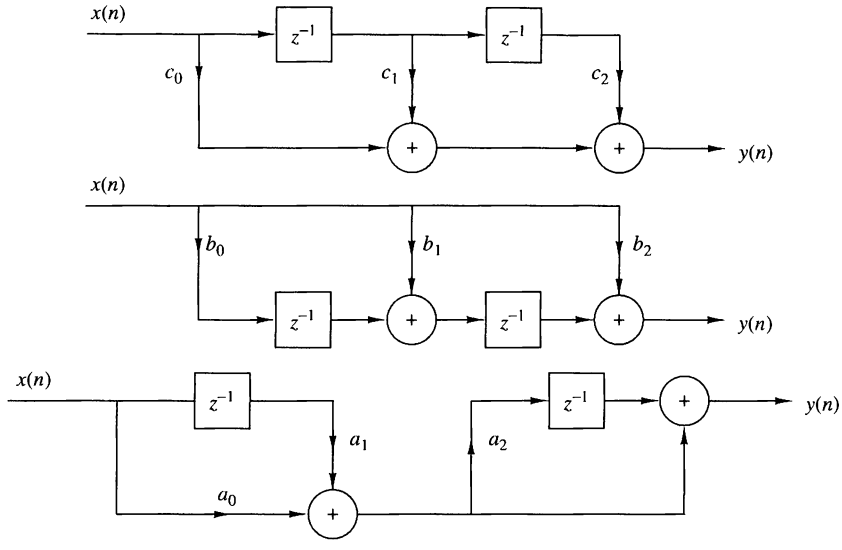


Figure P2.52

(a) Determine and sketch their impulse responses $h_1(n)$, $h_2(n)$, and $h_3(n)$.

(b) Is it possible to choose the coefficients of these systems in such a way that

$$h_1(n) = h_2(n) = h_3(n)$$

2.53 Consider the system shown in Fig. P2.53.

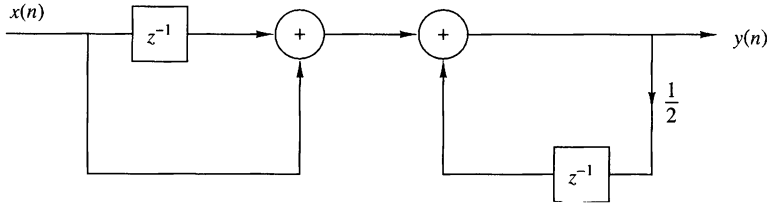


Figure P2.53

(a) Determine its impulse response $h(n)$.

(b) Show that $h(n)$ is equal to the convolution of the following signals:

$$h_1(n) = \delta(n) + \delta(n-1)$$

$$h_2(n) = (\frac{1}{2})^n u(n)$$

2.54 Compute and sketch the convolution $y_i(n)$ and correlation $r_i(n)$ sequences for the following pair of signals and comment on the results obtained.

(a) $x_1(n) = \{\underset{\uparrow}{1}, 2, 4\}$ $\quad h_1(n) = \{\underset{\uparrow}{1}, 1, 1, 1, 1\}$

(b) $x_2(n) = \{\underset{\uparrow}{0}, 1, -2, 3, -4\}$ $\quad h_2(n) = \{\frac{1}{2}, \underset{\uparrow}{1}, 2, 1, \frac{1}{2}\}$

(c) $x_3(n) = \{\underset{\uparrow}{1}, 2, 3, 4\}$ $\quad h_3(n) = \{\underset{\uparrow}{4}, 3, 2, 1\}$

(d) $x_4(n) = \{\underset{\uparrow}{1}, 2, 3, 4\}$ $\quad h_4(n) = \{\underset{\uparrow}{1}, 2, 3, 4\}$

2.55 The zero-state response of a causal LTI system to the input $x(n) = \{\underset{\uparrow}{1}, 3, 3, 1\}$ is $y(n) = \{\underset{\uparrow}{1}, 4, 6, 4, 1\}$. Determine its impulse response.

2.56 Prove by direct substitution the equivalence of equations (2.5.9) and (2.5.10), which describe the direct form II structure, to the relation (2.5.6), which describes the direct form I structure.

2.57 Determine the response $y(n)$, $n \geq 0$ of the system described by the second-order difference equation

$$y(n) - 4y(n-1) + 4y(n-2) = x(n) - x(n-1)$$

when the input is

$$x(n) = (-1)^n u(n)$$

and the initial conditions are $y(-1) = y(-2) = 0$.

2.58 Determine the impulse response $h(n)$ for the system described by the second-order difference equation

$$y(n) - 4y(n-1) + 4y(n-2) = x(n) - x(n-1)$$

2.59 Show that any discrete-time signal $x(n)$ can be expressed as

$$x(n) = \sum_{k=-\infty}^{\infty} [x(k) - x(k-1)]u(n-k)$$

where $u(n-k)$ is a unit step delayed by k units in time, that is,

$$u(n-k) = \begin{cases} 1, & n \geq k \\ 0, & \text{otherwise} \end{cases}$$

2.60 Show that the output of an LTI system can be expressed in terms of its unit step response $s(n)$ as follows.

$$y(n) = \sum_{k=-\infty}^{\infty} [s(k) - s(k-1)]x(n-k)$$

$$= \sum_{k=-\infty}^{\infty} [x(k) - x(k-1)]s(n-k)$$

2.61 Compute the correlation sequences $r_{xx}(l)$ and $r_{xy}(l)$ for the following signal sequences.

$$x(n) = \begin{cases} 1, & n_0 - N \leq n \leq n_0 + N \\ 0, & \text{otherwise} \end{cases}$$

$$y(n) = \begin{cases} 1, & -N \leq n \leq N \\ 0, & \text{otherwise} \end{cases}$$

2.62 Determine the autocorrelation sequences of the following signals.

(a) $x(n) = \{\underset{\uparrow}{1}, 2, 1, 1\}$

(b) $y(n) = \{\underset{\uparrow}{1}, 1, 2, 1\}$

What is your conclusion?

2.63 What is the normalized autocorrelation sequence of the signal $x(n)$ given by

$$x(n) = \begin{cases} 1, & -N \le n \le N \\ 0, & \text{otherwise} \end{cases}$$

2.64 An audio signal $s(t)$ generated by a loudspeaker is reflected at two different walls with reflection coefficients r_1 and r_2. The signal $x(t)$ recorded by a microphone close to the loudspeaker, after sampling, is

$$x(n) = s(n) + r_1 s(n - k_1) + r_2 s(n - k_2)$$

where k_1 and k_2 are the delays of the two echoes.

(a) Determine the autocorrelation $r_{xx}(l)$ of the signal $x(n)$.

(b) Can we obtain r_1, r_2, k_1, and k_2 by observing $r_{xx}(l)$?

(c) What happens if $r_2 = 0$?

2.65 *Time-delay estimation in radar* Let $x_a(t)$ be the transmitted signal and $y_a(t)$ be the received signal in a radar system, where

$$y_a(t) = a x_a(t - t_d) + v_a(t)$$

and $v_a(t)$ is additive random noise. The signals $x_a(t)$ and $y_a(t)$ are sampled in the receiver, according to the sampling theorem, and are processed digitally to determine the time delay and hence the distance of the object. The resulting discrete-time signals are

$$\begin{aligned} x(n) &= x_a(nT) \\ y(n) &= y_a(nT) = a x_a(nT - DT) + v_a(nT) \\ &\triangleq a x(n - D) + v(n) \end{aligned}$$

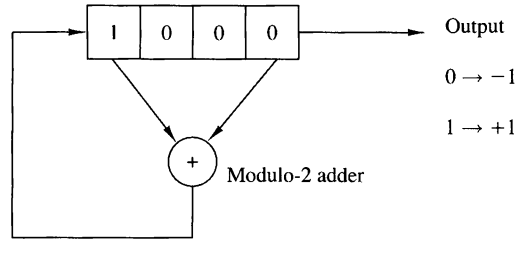


Figure P2.65 Linear feedback shift register.

(a) Explain how we can measure the delay D by computing the crosscorrelation $r_{xy}(l)$.

(b) Let $x(n)$ be the 13-point *Barker sequence*

$$x(n) = \{+1, +1, +1, +1, +1, -1, -1, +1, +1, -1, +1, -1, +1\}$$

and $v(n)$ be a Gaussian random sequence with zero mean and variance $\sigma^2 = 0.01$. Write a program that generates the sequence $y(n)$, $0 \leq n \leq 199$ for $a = 0.9$ and $D = 20$. Plot the signals $x(n)$, $y(n)$, $0 \leq n \leq 199$.

(c) Compute and plot the crosscorrelation $r_{xy}(l)$, $0 \leq l \leq 59$. Use the plot to estimate the value of the delay D.

(d) Repeat parts (b) and (c) for $\sigma^2 = 0.1$ and $\sigma^2 = 1$.

(e) Repeat parts (b) and (c) for the signal sequence

$$x(n) = \{-1, -1, -1, +1, +1, +1, +1, -1, +1, -1, +1, +1, -1, -1, +1\}$$

which is obtained from the four-stage feedback shift register shown in Fig. P2.65. Note that $x(n)$ is just one period of the periodic sequence obtained from the feedback shift register.

(f) Repeat parts (b) and (c) for a sequence of period $N = 2^7 - 1$, which is obtained from a seven-stage feedback shift register. Table 2.2 gives the stages connected to the modulo-2 adder for (maximal-length) shift-register sequences of length $N = 2^m - 1$.

TABLE 2.2 Shift-Register Connections for Generating Maximal-Length Sequences

m	Stages Connected to Modulo-2 Adder
1	1
2	1, 2
3	1, 3
4	1, 4
5	1, 4
6	1, 6
7	1, 7
8	1, 5, 6, 7
9	1, 6
10	1, 8
11	1, 10
12	1, 7, 9, 12
13	1, 10, 11, 13
14	1, 5, 9, 14
15	1, 15
16	1, 5, 14, 16
17	1, 15

2.66 *Implementation of LTI systems* Consider the recursive discrete-time system described by the difference equation

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 x(n)$$

where $a_1 = -0.8$, $a_2 = 0.64$, and $b_0 = 0.866$.

(a) Write a program to compute and plot the impulse response $h(n)$ of the system for $0 \leq n \leq 49$.

(b) Write a program to compute and plot the zero-state step response $s(n)$ of the system for $0 \leq n \leq 100$.

(c) Define an FIR system with impulse response $h_{\text{FIR}}(n)$ given by

$$h_{\text{FIR}}(n) = \begin{cases} h(n), & 0 \leq n \leq 19 \\ 0, & \text{elsewhere} \end{cases}$$

where $h(n)$ is the impulse response computed in part (a). Write a program to compute and plot its step response.

(d) Compare the results obtained in parts (b) and (c) and explain their similarities and differences.

2.67 Write a computer program that computes the overall impulse response $h(n)$ of the system shown in Fig. P2.67 for $0 \leq n \leq 99$. The systems $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$, and $\mathcal{T}_4$ are specified by

$$\mathcal{T}_1 : h_1(n) = \{\underset{\uparrow}{1}, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}\}$$

$$\mathcal{T}_2 : h_2(n) = \{\underset{\uparrow}{1}, 1, 1, 1, 1\}$$

$$\mathcal{T}_3 : y_3(n) = \frac{1}{4}x(n) + \frac{1}{2}x(n-1) + \frac{1}{4}x(n-2)$$

$$\mathcal{T}_4 : y(n) = 0.9y(n-1) - 0.81y(n-2) + v(n) + v(n-1)$$

Plot $h(n)$ for $0 \leq n \leq 99$.

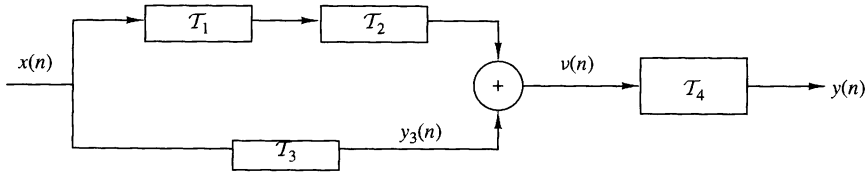


Figure P2.67

CHAPTER

3

The z-Transform and Its Application to the Analysis of LTI Systems

Transform techniques are an important tool in the analysis of signals and linear time-invariant (LTI) systems. In this chapter we introduce the z-transform, develop its properties, and demonstrate its importance in the analysis and characterization of linear time-invariant systems.

The z-transform plays the same role in the analysis of discrete-time signals and LTI systems as the Laplace transform does in the analysis of continuous-time signals and LTI systems. For example, we shall see that in the z-domain (complex z-plane) the convolution of two time-domain signals is equivalent to multiplication of their corresponding z-transforms. This property greatly simplifies the analysis of the response of an LTI system to various signals. In addition, the z-transform provides us with a means of characterizing an LTI system, and its response to various signals, by its pole–zero locations.

We begin this chapter by defining the z-transform. Its important properties are presented in Section 3.2. In Section 3.3 the transform is used to characterize signals in terms of their pole–zero patterns. Section 3.4 describes methods for inverting the z-transform of a signal so as to obtain the time-domain representation of the signal. Section 3.5 is focused on the use of the z-transform in the analysis of LTI systems. Finally, in Section 3.6, we treat the one-sided z-transform and use it to solve linear difference equations with nonzero initial conditions.

3.1 The z-Transform

In this section we introduce the z-transform of a discrete-time signal, investigate its convergence properties, and briefly discuss the inverse z-transform.

3.1.1 The Direct z-Transform

The z-transform of a discrete-time signal $x(n)$ is defined as the power series

$$X(z) \equiv \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{3.1.1}$$

where z is a complex variable. The relation (3.1.1) is sometimes called the *direct z-transform* because it transforms the time-domain signal $x(n)$ into its complex-plane representation $X(z)$. The inverse procedure [i.e., obtaining $x(n)$ from $X(z)$] is called the *inverse z-transform* and is examined briefly in Section 3.1.2 and in more detail in Section 3.4.

For convenience, the z-transform of a signal $x(n)$ is denoted by

$$X(z) \equiv Z\{x(n)\} \tag{3.1.2}$$

whereas the relationship between $x(n)$ and $X(z)$ is indicated by

$$x(n) \overset{z}{\longleftrightarrow} X(z) \tag{3.1.3}$$

Since the z-transform is an infinite power series, it exists only for those values of z for which this series converges. The *region of convergence* (ROC) of $X(z)$ is the set of all values of z for which $X(z)$ attains a finite value. Thus any time we cite a z-transform we should also indicate its ROC.

We illustrate these concepts by some simple examples.

EXAMPLE 3.1.1

Determine the z-transforms of the following *finite-duration* signals.

(a) $x_1(n) = \{\underset{\uparrow}{1}, 2, 5, 7, 0, 1\}$

(b) $x_2(n) = \{1, 2, \underset{\uparrow}{5}, 7, 0, 1\}$

(c) $x_3(n) = \{\underset{\uparrow}{0}, 0, 1, 2, 5, 7, 0, 1\}$

(d) $x_4(n) = \{2, 4, \underset{\uparrow}{5}, 7, 0, 1\}$

(e) $x_5(n) = \delta(n)$

(f) $x_6(n) = \delta(n-k), k > 0$

(g) $x_7(n) = \delta(n+k), k > 0$

Solution. From definition (3.1.1), we have

(a) $X_1(z) = 1 + 2z^{-1} + 5z^{-2} + 7z^{-3} + z^{-5}$, ROC: entire z-plane except $z = 0$

(b) $X_2(z) = z^2 + 2z + 5 + 7z^{-1} + z^{-3}$, ROC: entire z-plane except $z = 0$ and $z = \infty$

(c) $X_3(z) = z^{-2} + 2z^{-3} + 5z^{-4} + 7z^{-5} + z^{-7}$, ROC: entire z-plane except $z = 0$

(d) $X_4(z) = 2z^2 + 4z + 5 + 7z^{-1} + z^{-3}$, ROC: entire z-plane except $z = 0$ and $z = \infty$

(e) $X_5(z) = 1$ [i.e., $\delta(n) \overset{z}{\longleftrightarrow} 1$], ROC: entire z-plane

(f) $X_6(z) = z^{-k}$ [i.e., $\delta(n-k) \overset{z}{\longleftrightarrow} z^{-k}$], $k > 0$, ROC: entire z-plane except $z = 0$

(g) $X_7(z) = z^k$ [i.e., $\delta(n+k) \overset{z}{\longleftrightarrow} z^k$], $k > 0$, ROC: entire z-plane except $z = \infty$

From this example it is easily seen that the ROC of a *finite-duration signal* is the entire z-plane, except possibly the points $z = 0$ and/or $z = \infty$. These points are excluded, because z^k $(k > 0)$ becomes unbounded for $z = \infty$ and z^{-k} $(k > 0)$ becomes unbounded for $z = 0$.

From a mathematical point of view the z-transform is simply an alternative representation of a signal. This is nicely illustrated in Example 3.1.1, where we see that the coefficient of z^{-n}, in a given transform, is the value of the signal at time n. In other words, the exponent of z contains the time information we need to identify the samples of the signal.

In many cases we can express the sum of the finite or infinite series for the z-transform in a closed-form expression. In such cases the z-transform offers a compact alternative representation of the signal.

EXAMPLE 3.1.2

Determine the z-transform of the signal

$$x(n) = (\frac{1}{2})^n u(n)$$

Solution. The signal $x(n)$ consists of an infinite number of nonzero values

$$x(n) = \{1, (\frac{1}{2}), (\frac{1}{2})^2, (\frac{1}{2})^3, \ldots, (\frac{1}{2})^n, \ldots\}$$

The z-transform of $x(n)$ is the infinite power series

$$\begin{aligned} X(z) &= 1 + \frac{1}{2}z^{-1} + (\frac{1}{2})^2 z^{-2} + (\frac{1}{2})^n z^{-n} + \cdots \\ &= \sum_{n=0}^{\infty} (\frac{1}{2})^n z^{-n} = \sum_{n=0}^{\infty} (\frac{1}{2}z^{-1})^n \end{aligned}$$

This is an infinite geometric series. We recall that

$$1 + A + A^2 + A^3 + \cdots = \frac{1}{1 - A} \qquad \text{if } |A| < 1$$

Consequently, for $|\frac{1}{2}z^{-1}| < 1$, or equivalently, for $|z| > \frac{1}{2}$, $X(z)$ converges to

$$X(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}, \qquad \text{ROC: } |z| > \frac{1}{2}$$

We see that in this case, the z-transform provides a compact alternative representation of the signal $x(n)$.

Let us express the complex variable z in polar form as

$$z = re^{j\theta} \tag{3.1.4}$$

where $r = |z|$ and $\theta = \measuredangle z$. Then $X(z)$ can be expressed as

$$X(z)|_{z=re^{j\theta}} = \sum_{n=-\infty}^{\infty} x(n)r^{-n}e^{-j\theta n}$$

In the ROC of $X(z)$, $|X(z)| < \infty$. But

$$\begin{aligned} |X(z)| &= \left| \sum_{n=-\infty}^{\infty} x(n)r^{-n}e^{-j\theta n} \right| \\ &\leq \sum_{n=-\infty}^{\infty} |x(n)r^{-n}e^{-j\theta n}| = \sum_{n=-\infty}^{\infty} |x(n)r^{-n}| \end{aligned} \tag{3.1.5}$$

Hence $|X(z)|$ is finite if the sequence $x(n)r^{-n}$ is absolutely summable.

The problem of finding the ROC for $X(z)$ is equivalent to determining the range of values of r for which the sequence $x(n)r^{-n}$ is absolutely summable. To elaborate, let us express (3.1.5) as

$$\begin{aligned} |X(z)| &\leq \sum_{n=-\infty}^{-1} |x(n)r^{-n}| + \sum_{n=0}^{\infty} \left| \frac{x(n)}{r^n} \right| \\ &\leq \sum_{n=1}^{\infty} |x(-n)r^{n}| + \sum_{n=0}^{\infty} \left| \frac{x(n)}{r^n} \right| \end{aligned} \tag{3.1.6}$$

If $X(z)$ converges in some region of the complex plane, both summations in (3.1.6) must be finite in that region. If the first sum in (3.1.6) converges, there must exist values of r small enough such that the product sequence $x(-n)r^n$, $1 \leq n < \infty$, is absolutely summable. Therefore, the ROC for the first sum consists of all points in a circle of some radius r_1, where $r_1 < \infty$, as illustrated in Fig. 3.1.1(a). On the other hand, if the second sum in (3.1.6) converges, there must exist values of r large enough such that the product sequence $x(n)/r^n$, $0 \leq n < \infty$, is absolutely summable. Hence the ROC for the second sum in (3.1.6) consists of all points outside a circle of radius $r > r_2$, as illustrated in Fig. 3.1.1(b).

Since the convergence of $X(z)$ requires that both sums in (3.1.6) be finite, it follows that the ROC of $X(z)$ is generally specified as the annular region in the z-plane, $r_2 < r < r_1$, which is the common region where both sums are finite. This region is illustrated in Fig. 3.1.1(c). On the other hand, if $r_2 > r_1$, there is no common region of convergence for the two sums and hence $X(z)$ does not exist.

The following examples illustrate these important concepts.

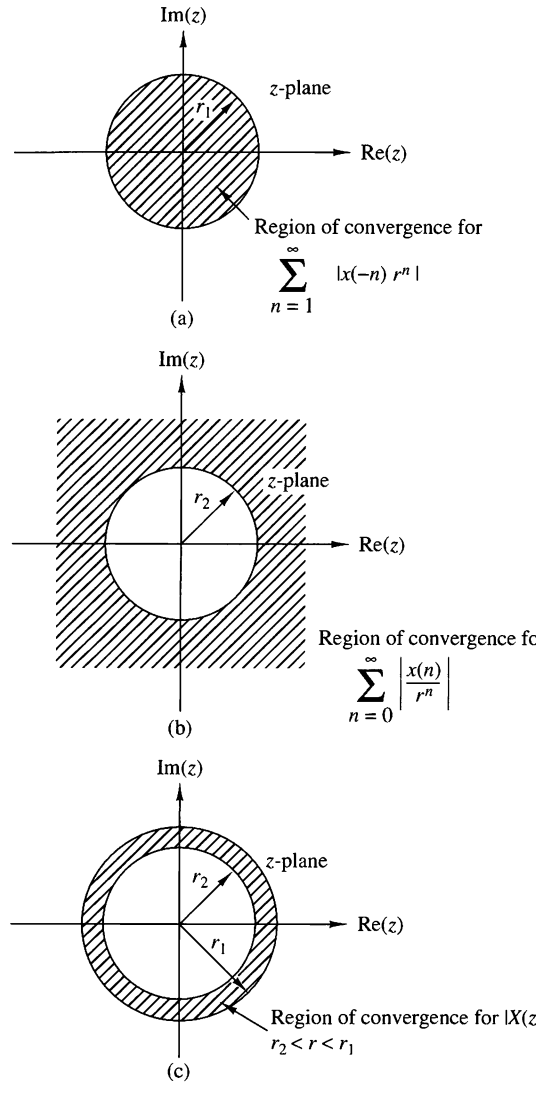


Figure 3.1.1
Region of convergence for $X(z)$ and its corresponding causal and anticausal components.

EXAMPLE 3.1.3

Determine the z-transform of the signal

$$x(n) = \alpha^n u(n) = \begin{cases} \alpha^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

Solution. From the definition (3.1.1) we have

$$X(z) = \sum_{n=0}^{\infty} \alpha^n z^{-n} = \sum_{n=0}^{\infty} (\alpha z^{-1})^n$$

If $|\alpha z^{-1}| < 1$ or equivalently, $|z| > |\alpha|$, this power series converges to $1/(1 - \alpha z^{-1})$. Thus we have the z-transform pair

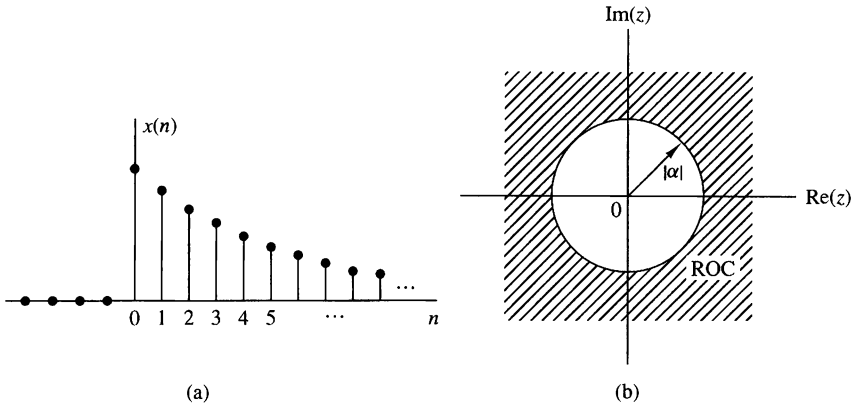


Figure 3.1.2 The exponential signal $x(n) = \alpha^n u(n)$ (a), and the ROC of its z-transform (b).

$$x(n) = \alpha^n u(n) \overset{z}{\longleftrightarrow} X(z) = \frac{1}{1 - \alpha z^{-1}}, \qquad \text{ROC: } |z| > |\alpha| \tag{3.1.7}$$

The ROC is the exterior of a circle having radius $|\alpha|$. Figure 3.1.2 shows a graph of the signal $x(n)$ and its corresponding ROC. Note that, in general, α need not be real.

If we set $\alpha = 1$ in (3.1.7), we obtain the z-transform of the unit step signal

$$x(n) = u(n) \overset{z}{\longleftrightarrow} X(z) = \frac{1}{1 - z^{-1}}, \qquad \text{ROC: } |z| > 1 \tag{3.1.8}$$

EXAMPLE 3.1.4

Determine the z-transform of the signal

$$x(n) = -\alpha^n u(-n-1) = \begin{cases} 0, & n \geq 0 \\ -\alpha^n, & n \leq -1 \end{cases}$$

Solution. From the definition (3.1.1) we have

$$X(z) = \sum_{n=-\infty}^{-1} (-\alpha^n) z^{-n} = -\sum_{l=1}^{\infty} (\alpha^{-1} z)^l$$

where $l = -n$. Using the formula

$$A + A^2 + A^3 + \cdots = A(1 + A + A^2 + \cdots) = \frac{A}{1 - A}$$

when $|A| < 1$ gives

$$X(z) = -\frac{\alpha^{-1} z}{1 - \alpha^{-1} z} = \frac{1}{1 - \alpha z^{-1}}$$

provided that $|\alpha^{-1} z| < 1$ or, equivalently, $|z| < |\alpha|$. Thus

$$x(n) = -\alpha^n u(-n-1) \overset{z}{\longleftrightarrow} X(z) = -\frac{1}{1 - \alpha z^{-1}}, \qquad \text{ROC: } |z| < |\alpha| \tag{3.1.9}$$

The ROC is now the interior of a circle having radius $|\alpha|$. This is shown in Fig. 3.1.3.

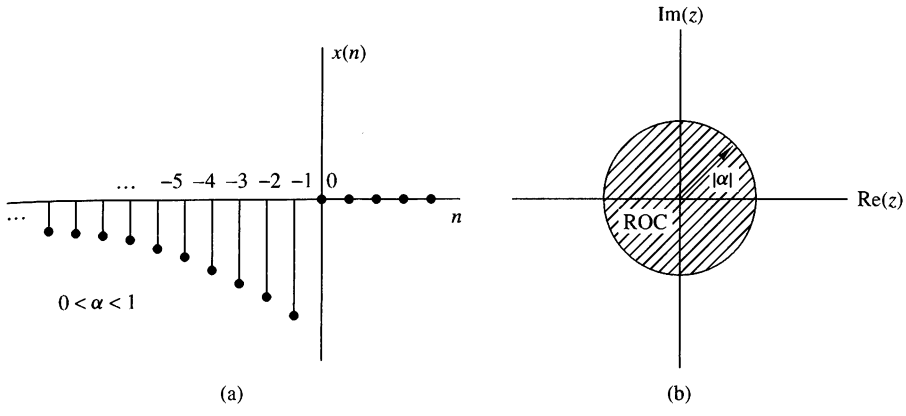


Figure 3.1.3 Anticausal signal $x(n) = -\alpha^n u(-n-1)$ (a), and the ROC of its z-transform (b).

Examples 3.1.3 and 3.1.4 illustrate two very important issues. The first concerns the uniqueness of the z-transform. From (3.1.7) and (3.1.9) we see that the causal signal $\alpha^n u(n)$ and the anticausal signal $-\alpha^n u(-n-1)$ have identical closed-form expressions for the z-transform, that is,

$$Z\{\alpha^n u(n)\} = Z\{-\alpha n u(-n-1)\} = \frac{1}{1-\alpha z^{-1}}$$

This implies that a closed-form expression for the z-transform does not uniquely specify the signal in the time domain. The ambiguity can be resolved only if in addition to the closed-form expression, the ROC is specified. In summary, *a discrete-time signal $x(n)$ is uniquely determined by its z-transform $X(z)$ and the region of convergence of $X(z)$*. In this text the term "z-transform" is used to refer to both the closed-form expression and the corresponding ROC. Example 3.1.3 also illustrates the point that *the ROC of a causal signal is the exterior of a circle of some radius r_2 while the ROC of an anticausal signal is the interior of a circle of some radius r_1*. The following example considers a sequence that is nonzero for $-\infty < n < \infty$.

EXAMPLE 3.1.5

Determine the z-transform of the signal

$$x(n) = \alpha^n u(n) + b^n u(-n-1)$$

Solution. From definition (3.1.1) we have

$$X(z) = \sum_{n=0}^{\infty} \alpha^n z^{-n} + \sum_{n=-\infty}^{-1} b^n z^{-n} = \sum_{n=0}^{\infty} (\alpha z^{-1})^n + \sum_{l=1}^{\infty} (b^{-1} z)^l$$

The first power series converges if $|\alpha z^{-1}| < 1$ or $|z| > |\alpha|$. The second power series converges if $|b^{-1} z| < 1$ or $|z| < |b|$.

In determining the convergence of $X(z)$, we consider two different cases.

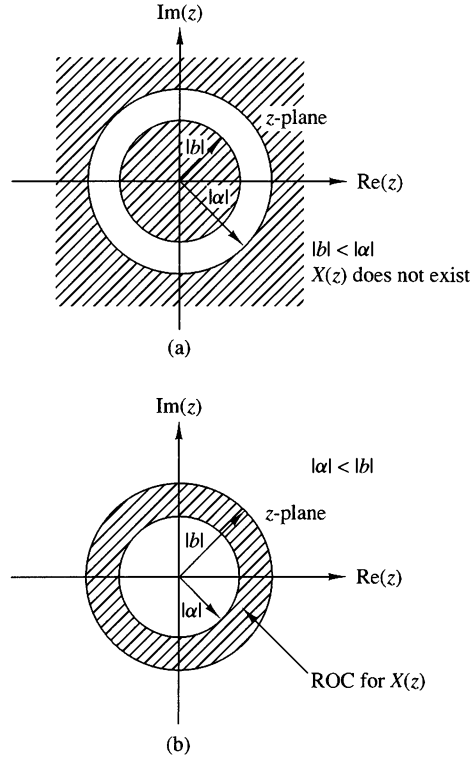


Figure 3.1.4
ROC for z-transform in Example 3.1.5.

Case 1 $|b| < |\alpha|$: In this case the two ROC above do not overlap, as shown in Fig. 3.1.4(a). Consequently, we cannot find values of z for which both power series converge simultaneously. Clearly, in this case, $X(z)$ does not exist.

Case 2 $|b| > |\alpha|$: In this case there is a ring in the z-plane where both power series converge simultaneously, as shown in Fig. 3.1.4(b). Then we obtain

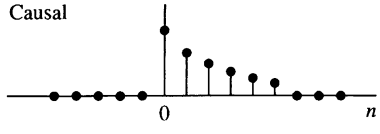
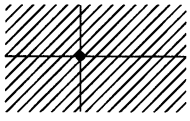
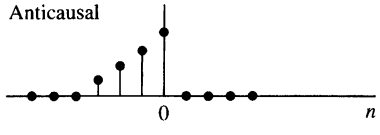
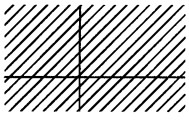
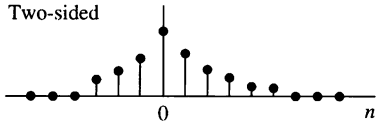
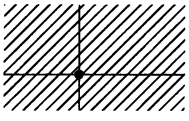
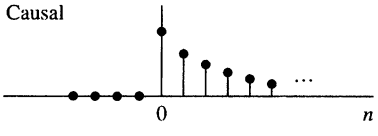
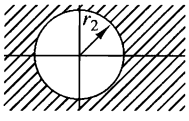
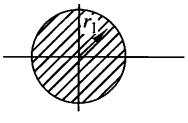
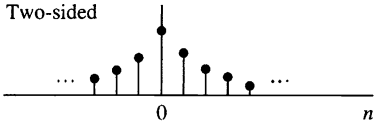
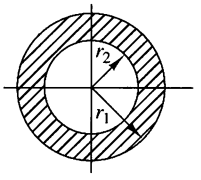
$$\begin{aligned} X(z) &= \frac{1}{1-\alpha z^{-1}} - \frac{1}{1-bz^{-1}} \\ &= \frac{b-\alpha}{\alpha + b - z - \alpha b z^{-1}} \end{aligned} \tag{3.1.10}$$

The ROC of $X(z)$ is $|\alpha| < |z| < |b|$.

This example shows that *if there is a ROC for an infinite-duration two-sided signal, it is a ring (annular region) in the z-plane.* From Examples 3.1.1, 3.1.3, 3.1.4, and 3.1.5, we see that the ROC of a signal depends both on its duration (finite or infinite) and on whether it is causal, anticausal, or two-sided. These facts are summarized in Table 3.1.

One special case of a two-sided signal is a signal that has infinite duration on the right side but not on the left [i.e., $x(n) = 0$ for $n < n_0 < 0$]. A second case is

TABLE 3.1 Characteristic Families of Signals with Their Corresponding ROCs

Signal	ROC
Finite-Duration Signals	
Causal 	 Entire z-plane except $z = 0$
Anticausal 	 Entire z-plane except $z = \infty$
Two-sided 	 Entire z-plane except $z = 0$ and $z = \infty$
Infinite-Duration Signals	
Causal 	 $\lvert z\rvert > r_2$
Anticausal 	 $\lvert z\rvert < r_1$
Two-sided 	 $r_2 < \lvert z\rvert < r_1$

a signal that has infinite duration on the left side but not on the right [i.e., $x(n) = 0$ for $n > n_1 > 0$]. A third special case is a signal that has finite duration on both the left and right sides [i.e., $x(n) = 0$ for $n < n_0 < 0$ and $n > n_1 > 0$]. These types of signals are sometimes called *right-sided, left-sided,* and *finite-duration two-sided* signals, respectively. The determination of the ROC for these three types of signals is left as an exercise for the reader (Problem 3.5).

Finally, we note that the z-transform defined by (3.1.1) is sometimes referred to as the *two-sided* or *bilateral z-transform*, to distinguish it from the *one-sided* or

unilateral z-transform given by

$$X^+(z) = \sum_{n=0}^{\infty} x(n)z^{-n} \tag{3.1.11}$$

The one-sided z-transform is examined in Section 3.6. In this text we use the expression z-transform exclusively to mean the two-sided z-transform defined by (3.1.1). The term "two-sided" will be used only in cases where we want to resolve any ambiguities. Clearly, if $x(n)$ is causal [i.e., $x(n) = 0$ for $n < 0$], the one-sided and two-sided z-transforms are identical. In any other case, they are different.

3.1.2 The Inverse z-Transform

Often, we have the z-transform $X(z)$ of a signal and we must determine the signal sequence. The procedure for transforming from the z-domain to the time domain is called the *inverse z-transform*. An inversion formula for obtaining $x(n)$ from $X(z)$ can be derived by using the *Cauchy integral theorem*, which is an important theorem in the theory of complex variables.

To begin, we have the z-transform defined by (3.1.1) as

$$X(z) = \sum_{k=-\infty}^{\infty} x(k)z^{-k} \tag{3.1.12}$$

Suppose that we multiply both sides of (3.1.12) by z^{n-1} and integrate both sides over a closed contour within the ROC of $X(z)$ which encloses the origin. Such a contour is illustrated in Fig. 3.1.5. Thus we have

$$\oint_C X(z)z^{n-1}\,dz = \oint_C \sum_{k=-\infty}^{\infty} x(k)z^{n-1-k}\,dz \tag{3.1.13}$$

where C denotes the closed contour in the ROC of $X(z)$, taken in a counterclockwise direction. Since the series converges on this contour, we can interchange the order of

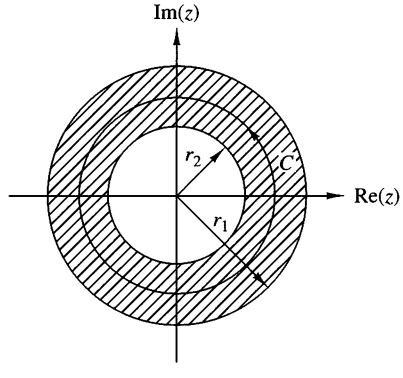


Figure 3.1.5
Contour C for integral in (3.1.13).

integration and summation on the right-hand side of (3.1.13). Thus (3.1.13) becomes

$$\oint_C X(z)z^{n-1}dz = \sum_{k=-\infty}^{\infty} x(k) \oint_C z^{n-1-k}dz \tag{3.1.14}$$

Now we can invoke the Cauchy integral theorem, which states that

$$\frac{1}{2\pi j} \oint_C z^{n-1-k}\, dz = \begin{cases} 1, & k = n \\ 0, & k \neq n \end{cases} \tag{3.1.15}$$

where C is any contour that encloses the origin. By applying (3.1.15), the right-hand side of (3.1.14) reduces to $2\pi j x(n)$ and hence the desired inversion formula

$$x(n) = \frac{1}{2\pi j} \oint_C X(z)z^{n-1}\, dz \tag{3.1.16}$$

Although the contour integral in (3.1.16) provides the desired inversion formula for determining the sequence $x(n)$ from the z-transform, we shall not use (3.1.16) directly in our evaluation of inverse z-transforms. In our treatment we deal with signals and systems in the z-domain which have rational z-transforms (i.e., z-transforms that are a ratio of two polynomials). For such z-transforms we develop a simpler method for inversion that stems from (3.1.16) and employs a table lookup.

3.2 Properties of the z-Transform

The z-transform is a very powerful tool for the study of discrete-time signals and systems. The power of this transform is a consequence of some very important properties that the transform possesses. In this section we examine some of these properties.

In the treatment that follows, it should be remembered that when we combine several z-transforms, the ROC of the overall transform is, at least, the intersection of the ROC of the individual transforms. This will become more apparent later, when we discuss specific examples.

Linearity. If

$$x_1(n) \overset{z}{\longleftrightarrow} X_1(z)$$

and

$$x_2(n) \overset{z}{\longleftrightarrow} X_2(z)$$

then

$$x(n) = a_1x_1(n) + a_2x_2(n) \overset{z}{\longleftrightarrow} X(z) = a_1X_1(z) + a_2X_2(z) \tag{3.2.1}$$

for any constants a_1 and a_2. The proof of this property follows immediately from the definition of linearity and is left as an exercise for the reader.

The linearity property can easily be generalized for an arbitrary number of signals. Basically, it implies that the z-transform of a linear combination of signals is the same linear combination of their z-transforms. Thus the linearity property helps us to find the z-transform of a signal by expressing the signal as a sum of elementary signals, for each of which, the z-transform is already known.

EXAMPLE 3.2.1

Determine the z-transform and the ROC of the signal

$$x(n) = [3(2^n) - 4(3^n)]u(n)$$

Solution. If we define the signals

$$x_1(n) = 2^n u(n)$$

and

$$x_2(n) = 3^n u(n)$$

then $x(n)$ can be written as

$$x(n) = 3x_1(n) - 4x_2(n)$$

According to (3.2.1), its z-transform is

$$X(z) = 3X_1(z) - 4X_2(z)$$

From (3.1.7) we recall that

$$\alpha^n u(n) \overset{z}{\longleftrightarrow} \frac{1}{1 - \alpha z^{-1}}, \qquad \text{ROC: } |z| > |\alpha| \tag{3.2.2}$$

By setting $\alpha = 2$ and $\alpha = 3$ in (3.2.2), we obtain

$$x_1(n) = 2^n u(n) \overset{z}{\longleftrightarrow} X_1(z) = \frac{1}{1 - 2z^{-1}}, \qquad \text{ROC: } |z| > 2$$

$$x_2(n) = 3^n u(n) \overset{z}{\longleftrightarrow} X_2(z) = \frac{1}{1 - 3z^{-1}}, \qquad \text{ROC: } |z| > 3$$

The intersection of the ROC of $X_1(z)$ and $X_2(z)$ is $|z| > 3$. Thus the overall transform $X(z)$ is

$$X(z) = \frac{3}{1 - 2z^{-1}} - \frac{4}{1 - 3z^{-1}}, \qquad \text{ROC: } |z| > 3$$

EXAMPLE 3.2.2

Determine the z-transform of the signals

(a) $x(n) = (\cos \omega_0 n)u(n)$

(b) $x(n) = (\sin \omega_0 n)u(n)$

Solution.

(a) By using Euler's identity, the signal $x(n)$ can be expressed as

$$x(n) = (\cos \omega_0 n)u(n) = \frac{1}{2}e^{j\omega_0 n}u(n) + \frac{1}{2}e^{-j\omega_0 n}u(n)$$

Thus (3.2.1) implies that

$$X(z) = \frac{1}{2}Z\{e^{j\omega_0 n}u(n)\} + \frac{1}{2}Z\{e^{-j\omega_0 n}u(n)\}$$

If we set $\alpha = e^{\pm j\omega_0}$ ($|\alpha| = |e^{\pm j\omega_0}| = 1$) in (3.2.2), we obtain

$$e^{j\omega_0 n}u(n) \overset{z}{\longleftrightarrow} \frac{1}{1 - e^{j\omega_0}z^{-1}}, \qquad \text{ROC: } |z| > 1$$

and

$$e^{-j\omega_0 n}u(n) \overset{z}{\longleftrightarrow} \frac{1}{1 - e^{-j\omega_0}z^{-1}}, \qquad \text{ROC: } |z| > 1$$

Thus

$$X(z) = \frac{1}{2}\frac{1}{1 - e^{j\omega_0}z^{-1}} + \frac{1}{2}\frac{1}{1 - e^{-j\omega_0}z^{-1}}, \qquad \text{ROC: } |z| > 1$$

After some simple algebraic manipulations we obtain the desired result, namely,

$$(\cos\omega_0 n)u(n) \overset{z}{\longleftrightarrow} \frac{1 - z^{-1}\cos\omega_0}{1 - 2z^{-1}\cos\omega_0 + z^{-2}}, \qquad \text{ROC: } |z| > 1 \tag{3.2.3}$$

(b) From Euler's identity,

$$x(n) = (\sin\omega 0n)u(n) = \frac{1}{2j}[e^{j\omega_0 n}u(n) - e^{-j\omega_0 n}u(n)]$$

Thus

$$X(z) = \frac{1}{2j}\left(\frac{1}{1 - e^{j\omega_0}z^{-1}} - \frac{1}{1 - e^{-j\omega_0}z^{-1}}\right), \qquad \text{ROC: } |z| > 1$$

and finally,

$$(\sin\omega_0 n)u(n) \overset{z}{\longleftrightarrow} \frac{z^{-1}\sin\omega_0}{1 - 2z^{-1}\cos\omega_0 + z^{-2}}, \qquad \text{ROC: } |z| > 1 \tag{3.2.4}$$

Time shifting. If

$$x(n) \overset{z}{\longleftrightarrow} X(z)$$

then

$$x(n - k) \overset{z}{\longleftrightarrow} z^{-k}X(z) \tag{3.2.5}$$

The ROC of $z^{-k}X(z)$ is the same as that of $X(z)$ except for $z = 0$ if $k > 0$ and $z = \infty$ if $k < 0$. The proof of this property follows immediately from the definition of the z-transform given in (3.1.1)

The properties of linearity and time shifting are the key features that make the z-transform extremely useful for the analysis of discrete-time LTI systems.

EXAMPLE 3.2.3

By applying the time-shifting property, determine the z-transform of the signals $x_2(n)$ and $x_3(n)$ in Example 3.1.1 from the z-transform of $x_1(n)$.

Solution. It can easily be seen that

$$x_2(n) = x_1(n+2)$$

and

$$x_3(n) = x_1(n-2)$$

Thus from (3.2.5) we obtain

$$X_2(z) = z^2 X_1(z) = z^2 + 2z + 5 + 7z^{-1} + z^{-3}$$

and

$$X_3(z) = z^{-2} X_1(z) = z^{-2} + 2z^{-3} + 5z^{-4} + 7z^{-5} + z^{-7}$$

Note that because of the multiplication by z^2, the ROC of $X_2(z)$ does not include the point $z = \infty$, even if it is contained in the ROC of $X_1(z)$.

Example 3.2.3 provides additional insight in understanding the meaning of the shifting property. Indeed, if we recall that the coefficient of z^{-n} is the sample value at time n, it is immediately seen that delaying a signal by $k (k > 0)$ samples [i.e., $x(n) \rightarrow x(n-k)$] corresponds to multiplying all terms of the z-transform by z^{-k}. The coefficient of z^{-n} becomes the coefficient of $z^{-(n+k)}$.

EXAMPLE 3.2.4

Determine the transform of the signal

$$x(n) = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & \text{elsewhere} \end{cases} \tag{3.2.6}$$

Solution. We can determine the z-transform of this signal by using the definition (3.1.1). Indeed,

$$X(z) = \sum_{n=0}^{N-1} 1 \cdot z^{-n} = 1 + z^{-1} + \cdots + z^{-(N-1)} = \begin{cases} N, & \text{if } z = 1 \\ \dfrac{1 - z^{-N}}{1 - z^{-1}}, & \text{if } z \neq 1 \end{cases} \tag{3.2.7}$$

Since $x(n)$ has finite duration, its ROC is the entire z-plane, except $z = 0$.

Let us also derive this transform by using the linearity and time-shifting properties. Note that $x(n)$ can be expressed in terms of two unit step signals

$$x(n) = u(n) - u(n-N)$$

By using (3.2.1) and (3.2.5) we have

$$X(z) = Z\{u(n)\} - Z\{u(n-N)\} = (1 - z^{-N}) Z\{u(n)\} \tag{3.2.8}$$

However, from (3.1.8) we have

$$Z\{u(n)\} = \frac{1}{1 - z^{-1}}, \qquad \text{ROC: } |z| > 1$$

which, when combined with (3.2.8), leads to (3.2.7).

Example 3.2.4 helps to clarify a very important issue regarding the ROC of the combination of several z-transforms. If the linear combination of several signals has finite duration, the ROC of its z-transform is exclusively dictated by the finite-duration nature of this signal, not by the ROC of the individual transforms.

Scaling in the z-domain. If

$$x(n) \overset{z}{\longleftrightarrow} X(z), \qquad \text{ROC: } r_1 < |z| < r_2$$

then

$$a^n x(n) \overset{z}{\longleftrightarrow} X(a^{-1}z), \qquad \text{ROC: } |a|r_1 < |z| < |a|r_2 \tag{3.2.9}$$

for any constant a, real or complex.

Proof From the definition (3.1.1)

$$Z\{a^n x(n)\} = \sum_{n=-\infty}^{\infty} a^n x(n) z^{-n} = \sum_{n=-\infty}^{\infty} x(n)(a^{-1}z)^{-n}$$

$$= X(a^{-1}z)$$

Since the ROC of $X(z)$ is $r_1 < |z| < r_2$, the ROC of $X(a^{-1}z)$ is

$$r_1 < |a^{-1}z| < r_2$$

or

$$|a|r_1 < |z| < |a|r_2$$

To better understand the meaning and implications of the scaling property, we express a and z in polar form as $a = r_0 e^{j\omega_0}$, $z = re^{j\omega}$, and we introduce a new complex variable $w = a^{-1}z$. Thus $Z\{x(n)\} = X(z)$ and $Z\{a^n x(n)\} = X(w)$. It can easily be seen that

$$w = a^{-1}z = \left(\frac{1}{r_0} r\right) e^{j(\omega - \omega_0)}$$

This change of variables results in either shrinking (if $r_0 > 1$) or expanding (if $r_0 < 1$) the z-plane in combination with a rotation (if $\omega_0 \neq 2k\pi$) of the z-plane (see Fig. 3.2.1). This explains why we have a change in the ROC of the new transform where $|a| < 1$. The case $|a| = 1$, that is, $a = e^{j\omega_0}$ is of special interest because it corresponds only to rotation of the z-plane.

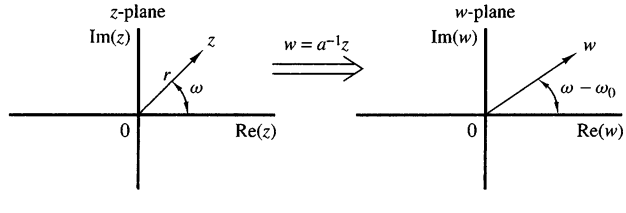


Figure 3.2.1
Mapping of the z-plane to the w-plane via the transformation $w = a^{-1}z$, $a = r_0 e^{j\omega_0}$.

EXAMPLE 3.2.5

Determine the z-transforms of the signals

(a) $x(n) = a^n(\cos \omega_0 n)u(n)$

(b) $x(n) = a^n(\sin \omega_0 n)u(n)$

Solution.

(a) From (3.2.3) and (3.2.9) we easily obtain

$$a^n(\cos \omega_0 n)u(n) \overset{z}{\longleftrightarrow} \frac{1 - az^{-1}\cos \omega_0}{1 - 2az^{-1}\cos \omega_0 + a^2 z^{-2}}, \qquad |z| > |a| \tag{3.2.10}$$

(b) Similarly, (3.2.4) and (3.2.9) yield

$$a^n(\sin \omega_0 n)u(n) \overset{z}{\longleftrightarrow} \frac{az^{-1}\sin \omega_0}{1 - 2az^{-1}\cos \omega_0 + a^2 z^{-2}}, \qquad |z| > |a| \tag{3.2.11}$$

Time reversal. If

$$x(n) \overset{z}{\longleftrightarrow} X(z), \qquad \text{ROC: } r_1 < |z| < r_2$$

then

$$x(-n) \overset{z}{\longleftrightarrow} X(z^{-1}), \qquad \text{ROC: } \frac{1}{r_2} < |z| < \frac{1}{r_1} \tag{3.2.12}$$

Proof From the definition (3.1.1), we have

$$Z\{x(-n)\} = \sum_{n=-\infty}^{\infty} x(-n)z^{-n} = \sum_{l=-\infty}^{\infty} x(l)(z^{-1})^{-l} = X(z^{-1})$$

where the change of variable $l = -n$ is made. The ROC of $X(z^{-1})$ is

$$r_1 < |z^{-1}| < r_2 \quad \text{or equivalently} \quad \frac{1}{r_2} < |z| < \frac{1}{r_1}$$

Note that the ROC for $x(n)$ is the inverse of that for $x(-n)$. This means that if z_0 belongs to the ROC of $x(n)$, then $1/z_0$ is in the ROC for $x(-n)$.

An intuitive proof of (3.2.12) is the following. When we fold a signal, the coefficient of z^{-n} becomes the coefficient of z^n. Thus, folding a signal is equivalent to replacing z by z^{-1} in the z-transform formula. In other words, reflection in the time domain corresponds to inversion in the z-domain.

EXAMPLE 3.2.6

Determine the z-transform of the signal

$$x(n) = u(-n)$$

Solution. It is known from (3.1.8) that

$$u(n) \overset{z}{\longleftrightarrow} \frac{1}{1-z^{-1}}, \qquad \text{ROC: } |z| > 1$$

By using (3.2.12), we easily obtain

$$u(-n) \overset{z}{\longleftrightarrow} \frac{1}{1-z}, \qquad \text{ROC: } |z| < 1 \tag{3.2.13}$$

Differentiation in the z-domain. If

$$x(n) \overset{z}{\longleftrightarrow} X(z)$$

then

$$nx(n) \overset{z}{\longleftrightarrow} -z\frac{dX(z)}{dz} \tag{3.2.14}$$

Proof By differentiating both sides of (3.1.1), we have

$$\frac{dX(z)}{dz} = \sum_{n=-\infty}^{\infty} x(n)(-n)z^{-n-1} = -z^{-1}\sum_{n=-\infty}^{\infty}[nx(n)]z^{-n}$$

$$= -z^{-1}Z\{nx(n)\}$$

Note that both transforms have the same ROC.

EXAMPLE 3.2.7

Determine the z-transform of the signal

$$x(n) = na^n u(n)$$

Solution. The signal $x(n)$ can be expressed as $nx_1(n)$, where $x_1(n) = a^n u(n)$. From (3.2.2) we have that

$$x_1(n) = a^n u(n) \overset{z}{\longleftrightarrow} X_1(z) = \frac{1}{1-az^{-1}}, \qquad \text{ROC: } |z| > |a|$$

Thus, by using (3.2.14), we obtain

$$na^n u(n) \overset{z}{\longleftrightarrow} X(z) = -z\frac{dX_1(z)}{dz} = \frac{az^{-1}}{(1-az^{-1})^2}, \qquad \text{ROC: } |z| > |a| \tag{3.2.15}$$

If we set $a = 1$ in (3.2.15), we find the z-transform of the unit ramp signal

$$nu(n) \overset{z}{\longleftrightarrow} \frac{z^{-1}}{(1-z^{-1})^2}, \qquad \text{ROC: } |z| > 1 \tag{3.2.16}$$

EXAMPLE 3.2.8

Determine the signal $x(n)$ whose z-transform is given by

$$X(z) = \log(1 + az^{-1}), \qquad |z| > |a|$$

Solution. By taking the first derivative of $X(z)$, we obtain

$$\frac{dX(z)}{dz} = \frac{-az^{-2}}{1 + az^{-1}}$$

Thus

$$-z\frac{dX(z)}{dz} = az^{-1}\left[\frac{1}{1 - (-a)z^{-1}}\right], \qquad |z| > |a|$$

The inverse z-transform of the term in brackets is $(-a)^n$. The multiplication by z^{-1} implies a time delay by one sample (time-shifting property), which results in $(-a)^{n-1}u(n-1)$. Finally, from the differentiation property we have

$$nx(n) = a(-a)^{n-1}u(n-1)$$

or

$$x(n) = (-1)^{n+1}\frac{a^n}{n}u(n-1)$$

Convolution of two sequences. If

$$x_1(n) \overset{z}{\longleftrightarrow} X_1(z)$$

$$x_2(n) \overset{z}{\longleftrightarrow} X_2(z)$$

then

$$x(n) = x_1(n) * x_2(n) \overset{z}{\longleftrightarrow} X(z) = X_1(z)X_2(z) \tag{3.2.17}$$

The ROC of $X(z)$ is, at least, the intersection of that for $X_1(z)$ and $X_2(z)$.

Proof The convolution of $x_1(n)$ and $x_2(n)$ is defined as

$$x(n) = \sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k)$$

The z-transform of $x(n)$ is

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} = \sum_{n=-\infty}^{\infty}\left[\sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k)\right]z^{-n}$$

Upon interchanging the order of the summations and applying the time-shifting property in (3.2.5), we obtain

$$X(z) = \sum_{k=-\infty}^{\infty} x_1(k)\left[\sum_{n=-\infty}^{\infty} x_2(n-k)z^{-n}\right]$$

$$= X_2(z)\sum_{k=-\infty}^{\infty} x_1(k)z^{-k} = X_2(z)X_1(z)$$

EXAMPLE 3.2.9

Compute the convolution $x(n)$ of the signals

$$x_1(n) = \{1, -2, 1\}$$

$$x_2(n) = \begin{cases} 1, & 0 \le n \le 5 \\ 0, & \text{elsewhere} \end{cases}$$

Solution. From (3.1.1), we have

$$X_1(z) = 1 - 2z^{-1} + z^{-2}$$

$$X_2(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5}$$

According to (3.2.17), we carry out the multiplication of $X_1(z)$ and $X_2(z)$. Thus

$$X(z) = X_1(z)X_2(z) = 1 - z^{-1} - z^{-6} + z^{-7}$$

Hence

$$x(n) = \{\underset{\uparrow}{1}, -1, 0, 0, 0, 0, -1, 1\}$$

The same result can also be obtained by noting that

$$X_1(z) = (1 - z^{-1})^2$$

$$X_2(z) = \frac{1 - z^{-6}}{1 - z^{-1}}$$

Then

$$X(z) = (1 - z^{-1})(1 - z^{-6}) = 1 - z^{-1} - z^{-6} + z^{-7}$$

The reader is encouraged to obtain the same result explicitly by using the convolution summation formula (time-domain approach).

The convolution property is one of the most powerful properties of the z-transform because it converts the convolution of two signals (time domain) to multiplication of their transforms. Computation of the convolution of two signals, using the z-transform, requires the following steps:

1. Compute the z-transforms of the signals to be convolved.

$$X_1(z) = Z\{x_1(n)\}$$

(time domain $\longrightarrow$ z-domain)

$$X_2(z) = Z\{x_2(n)\}$$

2. Multiply the two z-transforms.

$$X(z) = X_1(z)X_2(z), \qquad (z\text{-domain})$$

3. Find the inverse z-transform of $X(z)$.

$$x(n) = Z^{-1}\{X(z)\}, \qquad (z\text{-domain} \longrightarrow \text{time domain})$$

This procedure is, in many cases, computationally easier than the direct evaluation of the convolution summation.

Correlation of two sequences. If

$$x_1(n) \overset{z}{\longleftrightarrow} X_1(z)$$

$$x_2(n) \overset{z}{\longleftrightarrow} X_2(z)$$

then

$$r_{x_1x_2}(l) = \sum_{n=-\infty}^{\infty} x_1(n)x_2(n-l) \overset{z}{\longleftrightarrow} R_{x_1x_2}(z) = X_1(z)X_2(z^{-1}) \tag{3.2.18}$$

Proof We recall that

$$r_{x_1x_2}(l) = x_1(l) * x_2(-l)$$

Using the convolution and time-reversal properties, we easily obtain

$$R_{x_1x_2}(z) = Z\{x_1(l)\}Z\{x_2(-l)\} = X_1(z)X_2(z^{-1})$$

The ROC of $R_{x_1x_2}(z)$ is at least the intersection of that for $X_1(z)$ and $X_2(z^{-1})$.

As in the case of convolution, the crosscorrelation of two signals is more easily done via polynomial multiplication according to (3.2.18) and then inverse transforming the result.

EXAMPLE 3.2.10

Determine the autocorrelation sequence of the signal

$$x(n) = a^n u(n), \qquad -1 < a < 1$$

Solution. Since the autocorrelation sequence of a signal is its correlation with itself, (3.2.18) gives

$$R_{xx}(z) = Z\{r_{xx}(l)\} = X(z)X(z^{-1})$$

From (3.2.2) we have

$$X(z) = \frac{1}{1-az^{-1}}, \qquad \text{ROC: } |z| > |a| \qquad \text{(causal signal)}$$

and by using (3.2.15), we obtain

$$X(z^{-1}) = \frac{1}{1-az}, \qquad \text{ROC: } |z| < \frac{1}{|a|} \qquad \text{(anticausal signal)}$$

Thus

$$R_{xx}(z) = \frac{1}{1-az^{-1}}\frac{1}{1-az} = \frac{1}{1-a(z+z^{-1})+a^2}, \qquad \text{ROC: } |a| < |z| < \frac{1}{|a|}$$

Since the ROC of $R_{xx}(z)$ is a ring, $r_{xx}(l)$ is a two-sided signal, even if $x(n)$ is causal.

To obtain $r_{xx}(l)$, we observe that the z-transform of the sequence in Example 3.1.5 with $b = 1/a$ is simply $(1 - a^2)R_{xx}(z)$. Hence it follows that

$$r_{xx}(l) = \frac{1}{1 - a^2} a^{|l|}, \qquad -\infty < l < \infty$$

The reader is encouraged to compare this approach with the time-domain solution of the same problem given in Section 2.6.

Multiplication of two sequences. If

$$x_1(n) \overset{z}{\longleftrightarrow} X_1(z)$$

$$x_2(n) \overset{z}{\longleftrightarrow} X_2(z)$$

then

$$x(n) = x_1(n)x_2(n) \overset{z}{\longleftrightarrow} X(z) = \frac{1}{2\pi j} \oint_C X_1(v) X_2\left(\frac{z}{v}\right) v^{-1}\, dv \tag{3.2.19}$$

where C is a closed contour that encloses the origin and lies within the region of convergence common to both $X_1(v)$ and $X_2(1/v)$.

Proof The z-transform of $x_3(n)$ is

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} = \sum_{n=-\infty}^{\infty} x_1(n)x_2(n)z^{-n}$$

Let us substitute the inverse transform

$$x_1(n) = \frac{1}{2\pi j} \oint_C X_1(v)v^{n-1}dv$$

for $x_1(n)$ in the z-transform $X(z)$ and interchange the order of summation and integration. Thus we obtain

$$X(z) = \frac{1}{2\pi j} \oint_C X_1(v) \left[\sum_{n=-\infty}^{\infty} x_2(n) \left(\frac{z}{v}\right)^{-n} \right] v^{-1}dv$$

The sum in the brackets is simply the transform $X_2(z)$ evaluated at z/v. Therefore,

$$X(z) = \frac{1}{2\pi j} \oint_C X_1(v) X_2\left(\frac{z}{v}\right) v^{-1}\, dv$$

which is the desired result.

To obtain the ROC of $X(z)$ we note that if $X_1(v)$ converges for $r_{1l} < |v| < r_{1u}$ and $X_2(z)$ converges for $r_{2l} < |z| < r_{2u}$, then the ROC of $X_2(z/v)$ is

$$r_{2l} < \left|\frac{z}{v}\right| < r_{2u}$$

Hence the ROC for $X(z)$ is at least

$$r_{1l}r_{2l} < |z| < r_{1u}r_{2u} \tag{3.2.20}$$

Although this property will not be used immediately, it will prove useful later, especially in our treatment of filter design based on the window technique, where we multiply the impulse response of an IIR system by a finite-duration "window" which serves to truncate the impulse response of the IIR system.

For complex-valued sequences $x_1(n)$ and $x_2(n)$ we can define the product sequence as $x(n) = x_1(n)x_2^*(n)$. Then the corresponding complex convolution integral becomes

$$x(n) = x_1(n)x_2^*(n) \overset{z}{\longleftrightarrow} X(z) = \frac{1}{2\pi j}\oint_C X_1(v)X_2^*\left(\frac{z^*}{v^*}\right)v^{-1}\,dv \tag{3.2.21}$$

The proof of (3.2.21) is left as an exercise for the reader.

Parseval's relation. If $x_1(n)$ and $x_2(n)$ are complex-valued sequences, then

$$\sum_{n=-\infty}^{\infty} x_1(n)x_2^*(n) = \frac{1}{2\pi j}\oint_C X_1(v)X_2^*\left(\frac{1}{v^*}\right)v^{-1}\,dv \tag{3.2.22}$$

provided that $r_{1l}r_{2l} < 1 < r_{1u}r_{2u}$, where $r_{1l} < |z| < r_{1u}$ and $r_{2l} < |z| < r_{2u}$ are the ROC of $X_1(z)$ and $X_2(z)$. The proof of (3.2.22) follows immediately by evaluating $X(z)$ in (3.2.21) at $z = 1$.

The Initial Value Theorem. If $x(n)$ is *causal* [i.e., $x(n) = 0$ for $n < 0$], then

$$x(0) = \lim_{z\to\infty} X(z) \tag{3.2.23}$$

Proof Since $x(n)$ is causal, (3.1.1) gives

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n} = x(0) + x(1)z^{-1} + x(2)z^{-2} + \cdots$$

Obviously, as $z \to \infty$, $z^{-n} \to 0$ since $n > 0$, and (3.2.23) follows.

TABLE 3.2 Properties of the z-Transform

Property	Time Domain	z-Domain	ROC
Notation	$x(n)$	$X(z)$	ROC: $r_2 < \|z\| < r_1$
	$x_1(n)$	$X_1(z)$	ROC_1
	$x_2(n)$	$X_2(z)$	ROC_2
Linearity	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(z) + a_2X_2(z)$	At least the intersection of ROC_1 and ROC_2
Time shifting	$x(n-k)$	$z^{-k}X(z)$	That of $X(z)$, except $z = 0$ if $k > 0$ and $z = \infty$ if $k < 0$
Scaling in the z-domain	$a^n x(n)$	$X(a^{-1}z)$	$\|a\|r_2 < \|z\| < \|a\|r_1$
Time reversal	$x(-n)$	$X(z^{-1})$	$\frac{1}{r_1} < \|z\| < \frac{1}{r_2}$
Conjugation	$x^*(n)$	$X^*(z^*)$	ROC
Real part	$\text{Re}\{x(n)\}$	$\frac{1}{2}[X(z) + X^*(z^*)]$	Includes ROC
Imaginary part	$\text{Im}\{x(n)\}$	$\frac{1}{2}j[X(z) - X^*(z^*)]$	Includes ROC
Differentiation in the z-domain	$nx(n)$	$-z\frac{dX(z)}{dz}$	$r_2 < \|z\| < r_1$
Convolution	$x_1(n) * x_2(n)$	$X_1(z)X_2(z)$	At least, the intersection of ROC_1 and ROC_2
Correlation	$r_{x_1x_2}(l) = x_1(l) * x_2(-l)$	$R_{x_1x_2}(z) = X_1(z)X_2(z^{-1})$	At least, the intersection of ROC of $X_1(z)$ and $X_2(z^{-1})$
Initial value theorem	If $x(n)$ causal	$x(0) = \lim_{z\to\infty} X(z)$	
Multiplication	$x_1(n)x_2(n)$	$\frac{1}{2\pi j}\oint_C X_1(v)X_2\left(\frac{z}{v}\right)v^{-1}\,dv$	At least, $r_{1l}r_{2l} < \|z\| < r_{1u}r_{2u}$
Parseval's relation	$\sum_{n=-\infty}^{\infty} x_1(n)x_2^*(n) = \frac{1}{2\pi j}\oint_C X_1(v)X_2^*(1/v^*)v^{-1}dv$		

All the properties of the z-transform presented in this section are summarized in Table 3.2 for easy reference. They are listed in the same order as they have been introduced in the text. The conjugation properties and Parseval's relation are left as exercises for the reader.

We have now derived most of the z-transforms that are encountered in many practical applications. These z-transform pairs are summarized in Table 3.3 for easy reference. A simple inspection of this table shows that these z-transforms are all *rational functions* (i.e., ratios of polynomials in z^{-1}). As will soon become apparent, rational z-transforms are encountered not only as the z-transforms of various important signals but also in the characterization of discrete-time linear time-invariant systems described by constant-coefficient difference equations.

TABLE 3.3 Some Common z-Transform Pairs

	Signal, $x(n)$	z-Transform, $X(z)$	ROC
1	$\delta(n)$	1	All z
2	$u(n)$	$\frac{1}{1-z^{-1}}$	$\|z\| > 1$
3	$a^n u(n)$	$\frac{1}{1-az^{-1}}$	$\|z\| > \|a\|$
4	$na^n u(n)$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| > \|a\|$
5	$-a^n u(-n-1)$	$\frac{1}{1-az^{-1}}$	$\|z\| < \|a\|$
6	$-na^n u(-n-1)$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| < \|a\|$
7	$(\cos \omega_0 n)u(n)$	$\frac{1-z^{-1}\cos\omega_0}{1-2z^{-1}\cos\omega_0+z^{-2}}$	$\|z\| > 1$
8	$(\sin \omega_0 n)u(n)$	$\frac{z^{-1}\sin\omega_0}{1-2z^{-1}\cos\omega_0+z^{-2}}$	$\|z\| > 1$
9	$(a^n \cos \omega_0 n)u(n)$	$\frac{1-az^{-1}\cos\omega_0}{1-2az^{-1}\cos\omega_0+a^2z^{-2}}$	$\|z\| > \|a\|$
10	$(a^n \sin \omega_0 n)u(n)$	$\frac{az^{-1}\sin\omega_0}{1-2az^{-1}\cos\omega_0+a^2z^{-2}}$	$\|z\| > \|a\|$

3.3 Rational z-Transforms

As indicated in Section 3.2, an important family of z-transforms are those for which $X(z)$ is a rational function, that is, a ratio of two polynomials in z^{-1} (or z). In this section we discuss some very important issues regarding the class of rational z-transforms.

3.3.1 Poles and Zeros

The *zeros* of a z-transform $X(z)$ are the values of z for which $X(z) = 0$. The *poles* of a z-transform are the values of z for which $X(z) = \infty$. If $X(z)$ is a rational function, then

$$X(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \tag{3.3.1}$$

If $a_0 \neq 0$ and $b_0 \neq 0$, we can avoid the negative powers of z by factoring out the terms $b_0 z^{-M}$ and $a_0 z^{-N}$ as follows:

$$X(z) = \frac{B(z)}{A(z)} = \frac{b_0 z^{-M}}{a_0 z^{-N}} \frac{z^M + (b_1/b_0) z^{M-1} + \cdots + b_M/b_0}{z^N + (a_1/a_0) z^{N-1} + \cdots + a_N/a_0}$$

Since $B(z)$ and $A(z)$ are polynomials in z, they can be expressed in factored form as

$$X(z) = \frac{B(z)}{A(z)} = \frac{b_0}{a_0} z^{-M+N} \frac{(z - z_1)(z - z_2) \cdots (z - z_M)}{(z - p_1)(z - p_2) \cdots (z - p_N)}$$

$$X(z) = G z^{N-M} \frac{\prod_{k=1}^{M} (z - z_k)}{\prod_{k=1}^{N} (z - p_k)} \tag{3.3.2}$$

where $G \equiv b_0/a_0$. Thus $X(z)$ has M finite zeros at $z = z_1, z_2, \ldots, z_M$ (the roots of the numerator polynomial), N finite poles at $z = p_1, p_2, \ldots, p_N$ (the roots of the denominator polynomial), and $|N - M|$ zeros (if $N > M$) or poles (if $N < M$) at the origin $z = 0$. Poles or zeros may also occur at $z = \infty$. A zero exists at $z = \infty$ if $X(\infty) = 0$ and a pole exists at $z = \infty$ if $X(\infty) = \infty$. If we count the poles and zeros at zero and infinity, we find that $X(z)$ has exactly the same number of poles as zeros.

We can represent $X(z)$ graphically by a *pole–zero plot* (or *pattern*) in the complex plane, which shows the location of poles by crosses ($\times$) and the location of zeros by circles ($\circ$). The multiplicity of multiple-order poles or zeros is indicated by a number close to the corresponding cross or circle. Obviously, by definition, the ROC of a z-transform should not contain any poles.

EXAMPLE 3.3.1

Determine the pole–zero plot for the signal

$$x(n) = a^n u(n), \qquad a > 0$$

Solution. From Table 3.3 we find that

$$X(z) = \frac{1}{1 - az^{-1}} = \frac{z}{z - a}, \qquad \text{ROC: } |z| > a$$

Thus $X(z)$ has one zero at $z_1 = 0$ and one pole at $p_1 = a$. The pole–zero plot is shown in Fig. 3.3.1. Note that the pole $p_1 = a$ is not included in the ROC since the z-transform does not converge at a pole.

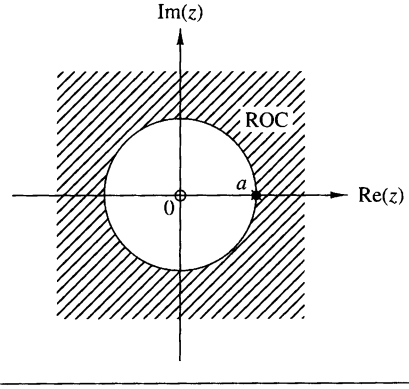


Figure 3.3.1
Pole–zero plot for the causal exponential signal $x(n) = a^n u(n)$.

EXAMPLE 3.3.2

Determine the pole–zero plot for the signal

$$x(n) = \begin{cases} a^n, & 0 \le n \le M-1 \\ 0, & \text{elsewhere} \end{cases}$$

where $a > 0$.

Solution. From the definition (3.1.1) we obtain

$$X(z) = \sum_{n=0}^{M-1} (az^{-1})^n = \frac{1-(az^{-1})^M}{1-az^{-1}} = \frac{z^M - a^M}{z^{M-1}(z-a)}$$

Since $a > 0$, the equation $z^M = a^M$ has M roots at

$$z_k = ae^{j2\pi k/M} \qquad k = 0, 1, \ldots, M-1$$

The zero $z_0 = a$ cancels the pole at $z = a$. Thus

$$X(z) = \frac{(z-z_1)(z-z_2)\cdots(z-z_{M-1})}{z^{M-1}}$$

which has $M-1$ zeros and $M-1$ poles, located as shown in Fig. 3.3.2 for $M = 8$. Note that the ROC is the entire z-plane except $z = 0$ because of the $M-1$ poles located at the origin.

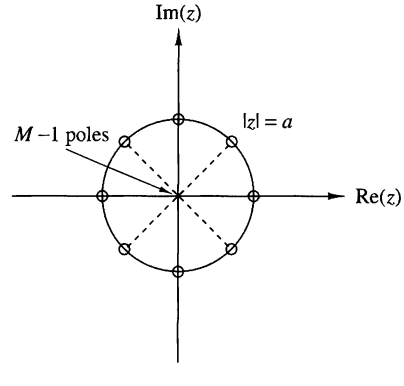


Figure 3.3.2
Pole–zero pattern for the finite-duration signal $x(n) = a^n$, $0 \le n \le M-1 (a > 0)$, for $M = 8$.

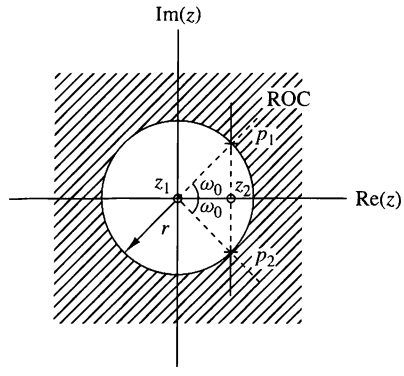


Figure 3.3.3
Pole-zero pattern for Example 3.3.3.

Clearly, if we are given a pole–zero plot, we can determine $X(z)$, by using (3.3.2), to within a scaling factor G. This is illustrated in the following example.

EXAMPLE 3.3.3

Determine the z-transform and the signal that corresponds to the pole–zero plot of Fig. 3.3.3.

Solution. There are two zeros ($M = 2$) at $z_1 = 0$, $z_2 = r\cos\omega_0$ and two poles ($N = 2$) at $p_1 = re^{j\omega_0}$, $p_2 = re^{-j\omega_0}$. By substitution of these relations into (3.3.2), we obtain

$$X(z) = G\frac{(z - z_1)(z - z_2)}{(z - p_1)(z - p_2)} = G\frac{z(z - r\cos\omega 0)}{(z - re^{j\omega_0})(z - re^{-j\omega_0})}, \qquad \text{ROC: } |z| > r$$

After some simple algebraic manipulations, we obtain

$$X(z) = G\frac{1 - rz^{-1}\cos\omega 0}{1 - 2rz^{-1}\cos\omega_0 + r^2z^{-2}}, \qquad \text{ROC: } |z| > r$$

From Table 3.3 we find that

$$x(n) = G(r^n\cos\omega_0 n)u(n)$$

From Example 3.3.3, we see that the product $(z - p_1)(z - p_2)$ results in a polynomial with real coefficients, when p_1 and p_2 are complex conjugates. In general, if a polynomial has real coefficients, its roots are either real or occur in complex-conjugate pairs.

As we have seen, the z-transform $X(z)$ is a complex function of the complex variable $z = \Re(z) + j\Im(z)$. Obviously, $|X(z)|$, the magnitude of $X(z)$, is a real and positive function of z. Since z represents a point in the complex plane, $|X(z)|$ is a two-dimensional function and describes a "surface." This is illustrated in Fig. 3.3.4 for the z-transform

$$X(z) = \frac{z^{-1} - z^{-2}}{1 - 1.2732z^{-1} + 0.81z^{-2}} \tag{3.3.3}$$

which has one zero at $z_1 = 1$ and two poles at p_1, $p_2 = 0.9e^{\pm j\pi/4}$. Note the high peaks near the singularities (poles) and the deep valley close to the zero.

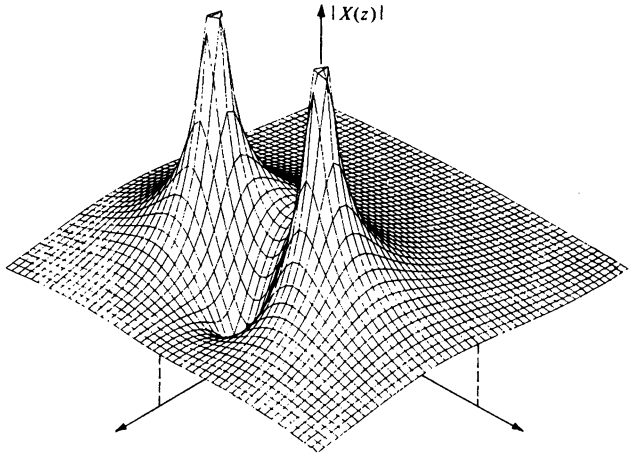


Figure 3.3.4 Graph of $|X(z)|$ for the z-transform in (3.3.3).

3.3.2 Pole Location and Time-Domain Behavior for Causal Signals

In this subsection we consider the relation between the z-plane location of a pole pair and the form (shape) of the corresponding signal in the time domain. The discussion is based generally on the collection of z-transform pairs given in Table 3.3 and the results in the preceding subsection. We deal exclusively with real, causal signals. In particular, we see that the characteristic behavior of causal signals depends on whether the poles of the transform are contained in the region $|z| < 1$, or in the region $|z| > 1$, or on the circle $|z| = 1$. Since the circle $|z| = 1$ has a radius of 1, it is called the *unit circle.*

If a real signal has a z-transform with one pole, this pole has to be real. The only such signal is the real exponential

$$x(n) = a^n u(n) \overset{z}{\longleftrightarrow} X(z) = \frac{1}{1 - az^{-1}}, \qquad \text{ROC: } |z| > |a|$$

having one zero at $z_1 = 0$ and one pole at $p_1 = a$ on the real axis. Figure 3.3.5 illustrates the behavior of the signal with respect to the location of the pole relative to the unit circle. The signal is decaying if the pole is inside the unit circle, fixed if the pole is on the unit circle, and growing if the pole is outside the unit circle. In addition, a negative pole results in a signal that alternates in sign. Obviously, causal signals with poles outside the unit circle become unbounded, cause overflow in digital systems, and in general, should be avoided.

A causal real signal with a double real pole has the form

$$x(n) = na^n u(n)$$

(see Table 3.3) and its behavior is illustrated in Fig. 3.3.6. Note that in contrast to the single-pole signal, a double real pole on the unit circle results in an unbounded signal.

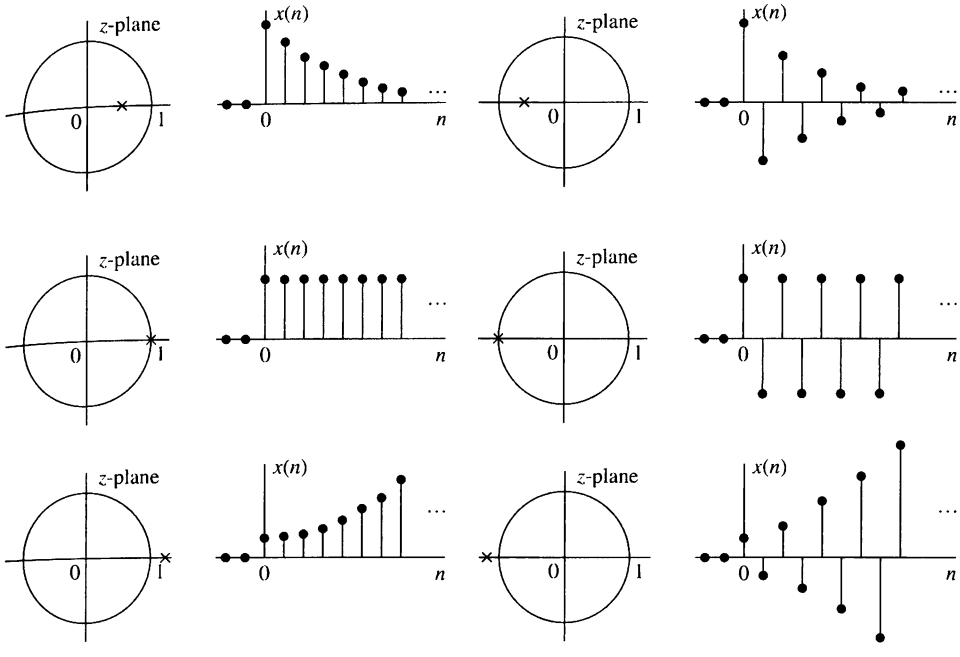


Figure 3.3.5 Time-domain behavior of a single-real-pole causal signal as a function of the location of the pole with respect to the unit circle.

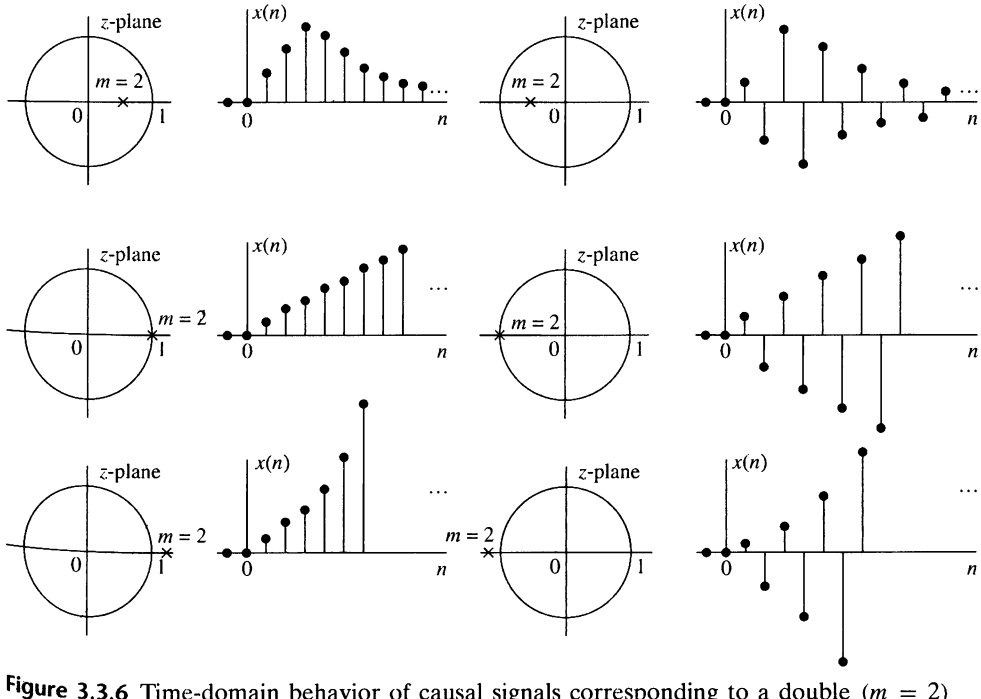


Figure 3.3.6 Time-domain behavior of causal signals corresponding to a double ($m = 2$) real pole, as a function of the pole location.

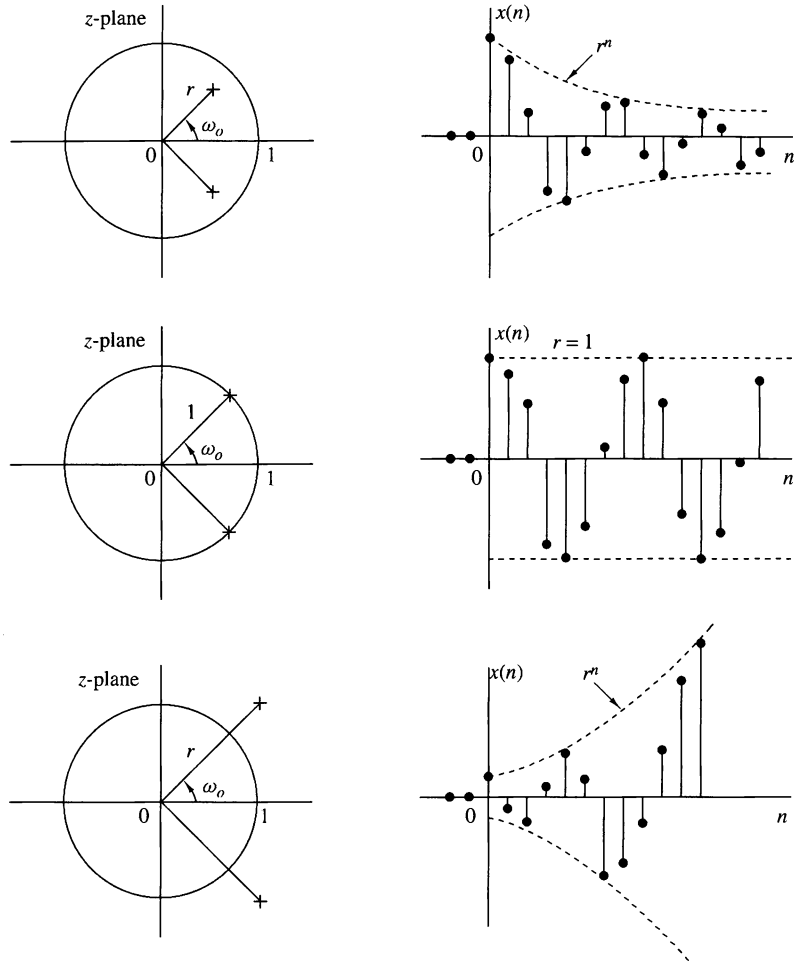


Figure 3.3.7 A pair of complex-conjugate poles corresponds to causal signals with oscillatory behavior.

Figure 3.3.7 illustrates the case of a pair of complex-conjugate poles. According to Table 3.3, this configuration of poles results in an exponentially weighted sinusoidal signal. The distance r of the poles from the origin determines the envelope of the sinusoidal signal and their angle with the real positive axis, its relative frequency. Note that the amplitude of the signal is growing if $r > 1$, constant if $r = 1$ (sinusoidal signals), and decaying if $r < 1$.

Finally, Fig. 3.3.8 shows the behavior of a causal signal with a double pair of poles on the unit circle. This reinforces the corresponding results in Fig. 3.3.6 and illustrates that multiple poles on the unit circle should be treated with great care.

To summarize, causal real signals with simple real poles or simple complex-conjugate pairs of poles, which are inside or on the unit circle, are always bounded in amplitude. Furthermore, a signal with a pole (or a complex-conjugate pair of poles)

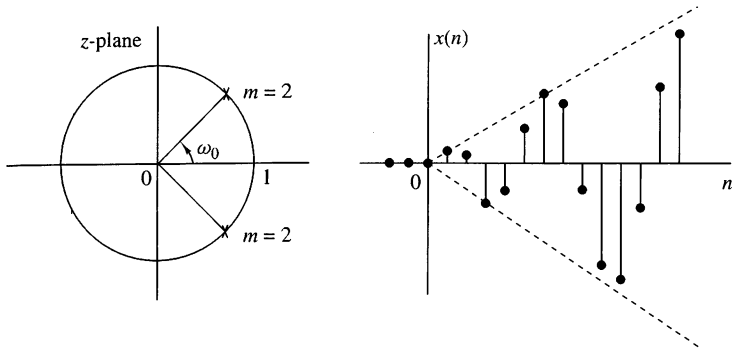


Figure 3.3.8 Causal signal corresponding to a double pair of complex-conjugate poles on the unit circle.

near the origin decays more rapidly than one associated with a pole near (but inside) the unit circle. Thus the time behavior of a signal depends strongly on the location of its poles relative to the unit circle. Zeros also affect the behavior of a signal but not as strongly as poles. For example, in the case of sinusoidal signals, the presence and location of zeros affects only their phase.

At this point, it should be stressed that everything we have said about causal signals applies as well to causal LTI systems, since their impulse response is a causal signal. Hence if a pole of a system is outside the unit circle, the impulse response of the system becomes unbounded and, consequently, the system is unstable.

3.3.3 The System Function of a Linear Time-Invariant System

In Chapter 2 we demonstrated that the output of a (relaxed) linear time-invariant system to an input sequence $x(n)$ can be obtained by computing the convolution of $x(n)$ with the unit sample response of the system. The convolution property, derived in Section 3.2, allows us to express this relationship in the z-domain as

$$Y(z) = H(z)X(z) \tag{3.3.4}$$

where $Y(z)$ is the z-transform of the output sequence $y(n)$, $X(z)$ is the z-transform of the input sequence $x(n)$ and $H(z)$ is the z-transform of the unit sample response $h(n)$.

If we know $h(n)$ and $x(n)$, we can determine their corresponding z-transforms $H(z)$ and $X(z)$, multiply them to obtain $Y(z)$, and therefore determine $y(n)$ by evaluating the inverse z-transform of $Y(z)$. Alternatively, if we know $x(n)$ and we observe the output $y(n)$ of the system, we can determine the unit sample response by first solving for $H(z)$ from the relation

$$H(z) = \frac{Y(z)}{X(z)} \tag{3.3.5}$$

and then evaluating the inverse z-transform of $H(z)$.

Since

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} \tag{3.3.6}$$

it is clear that $H(z)$ represents the z-domain characterization of a system, whereas $h(n)$ is the corresponding time-domain characterization of the system. In other words, $H(z)$ and $h(n)$ are equivalent descriptions of a system in the two domains. The transform $H(z)$ is called the *system function.*

The relation in (3.3.5) is particularly useful in obtaining $H(z)$ when the system is described by a linear constant-coefficient difference equation of the form

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{3.3.7}$$

In this case the system function can be determined directly from (3.3.7) by computing the z-transform of both sides of (3.3.7). Thus, by applying the time-shifting property, we obtain

$$\begin{aligned} Y(z) &= -\sum_{k=1}^{N} a_k Y(z)z^{-k} + \sum_{k=0}^{M} b_k X(z)z^{-k} \\ Y(z)\left(1 + \sum_{k=1}^{N} a_k z^{-k}\right) &= X(z)\left(\sum_{k=0}^{M} b_k z^{-k}\right) \\ \frac{Y(z)}{X(z)} = H(z) &= \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \end{aligned} \tag{3.3.8}$$

Therefore, a linear time-invariant system described by a constant-coefficient difference equation has a rational system function.

This is the general form for the system function of a system described by a linear constant-coefficient difference equation. From this general form we obtain two important special forms. First, if $a_k = 0$ for $1 \le k \le N$, (3.3.8) reduces to

$$H(z) = \sum_{k=0}^{M} b_k z^{-k} = \frac{1}{z^M} \sum_{k=0}^{M} b_k z^{M-k} \tag{3.3.9}$$

In this case, $H(z)$ contains M zeros, whose values are determined by the system parameters $\{b_k\}$, and an Mth-order pole at the origin $z = 0$. Since the system contains only trivial poles (at $z = 0$) and M nontrivial zeros, it is called an *all-zero system.* Clearly, such a system has a finite-duration impulse response (FIR), and it is called an FIR system or a moving average (MA) system.

On the other hand, if $b_k = 0$ for $1 \leq k \leq M$, the system function reduces to

$$H(z) = \frac{b_0}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \frac{b_0 z^N}{\sum_{k=0}^{N} a_k z^{N-k}}, \qquad a_0 \equiv 1 \tag{3.3.10}$$

In this case $H(z)$ consists of N poles, whose values are determined by the system parameters $\{a_k\}$ and an Nth-order zero at the origin $z = 0$. We usually do not make reference to these trivial zeros. Consequently, the system function in (3.3.10) contains only nontrivial poles and the corresponding system is called an *all-pole system*. Due to the presence of poles, the impulse response of such a system is infinite in duration, and hence it is an IIR system.

The general form of the system function given by (3.3.8) contains both poles and zeros, and hence the corresponding system is called a *pole–zero system*, with N poles and M zeros. Poles and/or zeros at $z = 0$ and $z = \infty$ are implied but are not counted explicitly. Due to the presence of poles, a pole–zero system is an IIR system.

The following example illustrates the procedure for determining the system function and the unit sample response from the difference equation.

EXAMPLE 3.3.4

Determine the system function and the unit sample response of the system described by the difference equation

$$y(n) = \frac{1}{2} y(n-1) + 2x(n)$$

Solution. By computing the z-transform of the difference equation, we obtain

$$Y(z) = \frac{1}{2} z^{-1} Y(z) + 2X(z)$$

Hence the system function is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{2}{1 - \frac{1}{2} z^{-1}}$$

This system has a pole at $z = \frac{1}{2}$ and a zero at the origin. Using Table 3.3 we obtain the inverse transform

$$h(n) = 2\left(\frac{1}{2}\right)^n u(n)$$

This is the unit sample response of the system.

We have now demonstrated that rational z-transforms are encountered in commonly used systems and in the characterization of linear time-invariant systems. In Section 3.4 we describe several methods for determining the inverse z-transform of rational functions.

3.4 Inversion of the z-Transform

As we saw in Section 3.1.2, the inverse z-transform is formally given by

$$x(n) = \frac{1}{2\pi j} \oint_C X(z) z^{n-1} dz \tag{3.4.1}$$

where the integral is a contour integral over a closed path C that encloses the origin and lies within the region of convergence of $X(z)$. For simplicity, C can be taken as a circle in the ROC of $X(z)$ in the z-plane.

There are three methods that are often used for the evaluation of the inverse z-transform in practice:

1. Direct evaluation of (3.4.1), by contour integration.
2. Expansion into a series of terms, in the variables z, and z^{-1}.
3. Partial-fraction expansion and table lookup.

3.4.1 The Inverse z-Transform by Contour Integration

In this section we demonstrate the use of the Cauchy's integral theorem to determine the inverse z-transform directly from the contour integral.

Cauchy's integral theorem. Let $f(z)$ be a function of the complex variable z and C be a closed path in the z-plane. If the derivative $df(z)/dz$ exists on and inside the contour C and if $f(z)$ has no poles at $z = z_0$, then

$$\frac{1}{2\pi j} \oint_C \frac{f(z)}{z - z_0} dz = \begin{cases} f(z_0), & \text{if } z_0 \text{ is inside } C \\ 0, & \text{if } z_0 \text{ is outside } C \end{cases} \tag{3.4.2}$$

More generally, if the $(k+1)$-order derivative of $f(z)$ exists and $f(z)$ has no poles at $z = z_0$, then

$$\frac{1}{2\pi j} \oint_C \frac{f(z)}{(z - z_0)^k} dz = \begin{cases} \dfrac{1}{(k-1)!} \dfrac{d^{k-1} f(z)}{dz^{k-1}}\Big|_{z=z_0}, & \text{if } z_0 \text{ is inside } C \\ 0, & \text{if } z_0 \text{ is outside } C \end{cases} \tag{3.4.3}$$

The values on the right-hand side of (3.4.2) and (3.4.3) are called the residues of the pole at $z = z_0$. The results in (3.4.2) and (3.4.3) are two forms of the *Cauchy's integral theorem.*

We can apply (3.4.2) and (3.4.3) to obtain the values of more general contour integrals. To be specific, suppose that the integrand of the contour integral is a

proper fraction $f(z)/g(z)$, where $f(z)$ has no poles inside the contour C and $g(z)$ is a polynomial with distinct (simple) roots $z_1, z_2, \ldots, z_n$ inside C. Then

$$\begin{aligned}\frac{1}{2\pi j}\oint_C \frac{f(z)}{g(z)}\,dz &= \frac{1}{2\pi j}\oint_C \left[\sum_{i=1}^{n}\frac{A_i}{z-z_i}\right] dz \\ &= \sum_{i=1}^{n}\frac{1}{2\pi j}\oint_C \frac{A_i}{z-z_i}\,dz \\ &= \sum_{i=1}^{n} A_i\end{aligned} \tag{3.4.4}$$

where

$$A_i = (z - z_i)\left.\frac{f(z)}{g(z)}\right|_{z=z_i} \tag{3.4.5}$$

The values $\{A_i\}$ are residues of the corresponding poles at $z = z_i$, $i = 1, 2, \ldots, n$. Hence the value of the contour integral is equal to the sum of the residues of all the poles inside the contour C.

We observe that (3.4.4) was obtained by performing a partial-fraction expansion of the integrand and applying (3.4.2). When $g(z)$ has multiple-order roots as well as simple roots inside the contour, the partial-fraction expansion, with appropriate modifications, and (3.4.3) can be used to evaluate the residues at the corresponding poles.

In the case of the inverse z-transform, we have

$$\begin{aligned}x(n) &= \frac{1}{2\pi j}\oint_C X(z)z^{n-1}\,dz \\ &= \sum_{\text{all poles } \{z_i\} \text{ inside } C} [\text{residue of } X(z)z^{n-1} \text{ at } z = z_i] \\ &= \sum_i (z - z_i)X(z)z^{n-1}|_{z=z_i}\end{aligned} \tag{3.4.6}$$

provided that the poles $\{z_i\}$ are simple. If $X(z)z^{n-1}$ has no poles inside the contour C for one or more values of n, then $x(n) = 0$ for these values.

The following example illustrates the evaluation of the inverse z-transform by use of the Cauchy's integral theorem.

EXAMPLE 3.4.1

Evaluate the inverse z-transform of

$$X(z) = \frac{1}{1 - az^{-1}}, \qquad |z| > |a|$$

using the complex inversion integral.

Solution. We have

$$x(n) = \frac{1}{2\pi j} \oint_C \frac{z^{n-1}}{1 - az^{-1}}\, dz = \frac{1}{2\pi j} \oint_C \frac{z^n\, dz}{z - a}$$

where C is a circle at radius greater than $|a|$. We shall evaluate this integral using (3.4.2) with $f(z) = z^n$. We distinguish two cases.

1. If $n \geq 0$, $f(z)$ has only zeros and hence no poles inside C. The only pole inside C is $z = a$. Hence
$$x(n) = f(z_0) = a^n, \qquad n \geq 0$$
2. If $n < 0$, $f(z) = z^n$ has an nth-order pole at $z = 0$, which is also inside C. Thus there are contributions from both poles. For $n = -1$ we have
$$x(-1) = \frac{1}{2\pi j} \oint_C \frac{1}{z(z - a)}\, dz = \frac{1}{z - a}\bigg|_{z=0} + \frac{1}{z}\bigg|_{z=a} = 0$$
If $n = -2$, we have
$$x(-2) = \frac{1}{2\pi j} \oint_C \frac{1}{z^2(z - a)}\, dz = \frac{d}{dz}\left(\frac{1}{z - a}\right)\bigg|_{z=0} + \frac{1}{z^2}\bigg|_{z=a} = 0$$
By continuing in the same way we can show that $x(n) = 0$ for $n < 0$. Thus
$$x(n) = a^n u(n)$$

3.4.2 The Inverse z-Transform by Power Series Expansion

The basic idea in this method is the following: Given a z-transform $X(z)$ with its corresponding ROC, we can expand $X(z)$ into a power series of the form

$$X(z) = \sum_{n=-\infty}^{\infty} c_n z^{-n} \tag{3.4.7}$$

which converges in the given ROC. Then, by the uniqueness of the z-transform, $x(n) = c_n$ for all n. When $X(z)$ is rational, the expansion can be performed by long division.

To illustrate this technique, we will invert some z-transforms involving the same expression for $X(z)$, but different ROC. This will also serve to emphasize again the importance of the ROC in dealing with z-transforms.

EXAMPLE 3.4.2

Determine the inverse z-transform of

$$X(z) = \frac{1}{1 - 1.5z^{-1} + 0.5z^{-2}}$$

when

(a) ROC: $|z| > 1$

(b) ROC: $|z| < 0.5$

Solution.

(a) Since the ROC is the exterior of a circle, we expect $x(n)$ to be a causal signal. Thus we seek a power series expansion in negative powers of z. By dividing the numerator of $X(z)$ by its denominator, we obtain the power series

$$X(z) = \frac{1}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}} = 1 + \frac{3}{2}z^{-1} + \frac{7}{4}z^{-2} + \frac{15}{8}z^{-3} + \frac{31}{16}z^{-4} + \cdots$$

By comparing this relation with (3.1.1), we conclude that

$$x(n) = \{1, \underset{\uparrow}{\frac{3}{2}}, \frac{7}{4}, \frac{15}{8}, \frac{31}{16}, \ldots\}$$

Note that in each step of the long-division process, we eliminate the lowest-power term of z^{-1}.

(b) In this case the ROC is the interior of a circle. Consequently, the signal $x(n)$ is anticausal. To obtain a power series expansion in positive powers of z, we perform the long division in the following way:

$$\begin{array}{r} 2z^2 + 6z^3 + 14z^4 + 30z^5 + 62z^6 + \cdots \\ \hline \frac{1}{2}z^{-2} - \frac{3}{2}z^{-1} + 1 \;\big)\, 1 \\ \underline{1 - 3z + 2z^2} \\ 3z - 2z^2 \\ \underline{3z - 9z^2 + 6z^3} \\ 7z^2 - 6z^3 \\ \underline{7z^2 - 21z^3 + 14z^4} \\ 15z^3 - 14z^4 \\ \underline{15z^3 - 45z^4 + 30z^5} \\ 31z^4 - 30z^5 \end{array}$$

Thus

$$X(z) = \frac{1}{1 - \frac{3}{2}z^{-1} + \frac{1}{2}z^{-2}} = 2z^2 + 6z^3 + 14z^4 + 30z^5 + 62z^6 + \cdots$$

In this case $x(n) = 0$ for $n \geq 0$. By comparing this result to (3.1.1), we conclude that

$$x(n) = \{\cdots 62, 30, 14, 6, 2, 0, \underset{\uparrow}{0}\}$$

We observe that in each step of the long-division process, the lowest-power term of z is eliminated. We emphasize that in the case of anticausal signals we simply carry out the long division by writing down the two polynomials in "reverse" order (i.e., starting with the most negative term on the left).

From this example we note that, in general, the method of long division will not provide answers for $x(n)$ when n is large because the long division becomes tedious. Although the method provides a direct evaluation of $x(n)$, a closed-form solution is not possible, except if the resulting pattern is simple enough to infer the general term $x(n)$. Hence this method is used only if one wishes to determine the values of the first few samples of the signal.

EXAMPLE 3.4.3

Determine the inverse z-transform of

$$X(z) = \log(1 + az^{-1}), \qquad |z| > |a|$$

Solution. Using the power series expansion for $\log(1 + x)$, with $|x| < 1$, we have

$$X(z) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} a^n z^{-n}}{n}$$

Thus

$$x(n) = \begin{cases} (-1)^{n+1} \dfrac{a^n}{n}, & n \geq 1 \\ 0, & n \leq 0 \end{cases}$$

Expansion of irrational functions into power series can be obtained from tables.

3.4.3 The Inverse z-Transform by Partial-Fraction Expansion

In the table lookup method, we attempt to express the function $X(z)$ as a linear combination

$$X(z) = \alpha_1 X_1(z) + \alpha_2 X_2(z) + \cdots + \alpha_K X_K(z) \tag{3.4.8}$$

where $X_1(z), \ldots, X_K(z)$ are expressions with inverse transforms $x_1(n), \ldots, x_K(n)$ available in a table of z-transform pairs. If such a decomposition is possible, then $x(n)$, the inverse z-transform of $X(z)$, can easily be found using the linearity property as

$$x(n) = \alpha_1 x_1(n) + \alpha_2 x_2(n) + \cdots + \alpha_K x_K(n) \tag{3.4.9}$$

This approach is particularly useful if $X(z)$ is a rational function, as in (3.3.1). Without loss of generality, we assume that $a_0 = 1$, so that (3.3.1) can be expressed as

$$X(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \tag{3.4.10}$$

Note that if $a_0 \neq 1$, we can obtain (3.4.10) from (3.3.1) by dividing both numerator and denominator by a_0.

A rational function of the form (3.4.10) is called *proper* if $a_N \neq 0$ and $M < N$. From (3.3.2) it follows that this is equivalent to saying that the number of finite zeros is less than the number of finite poles.

An improper rational function ($M \geq N$) can always be written as the sum of a polynomial and a proper rational function. This procedure is illustrated by the following example.

EXAMPLE 3.4.4

Express the improper rational transform

$$X(z) = \frac{1 + 3z^{-1} + \frac{11}{6} z^{-2} + \frac{1}{3} z^{-3}}{1 + \frac{5}{6} z^{-1} + \frac{1}{6} z^{-2}}$$

in terms of a polynomial and a proper function.

Solution. First, we note that we should reduce the numerator so that the terms z^{-2} and z^{-3} are eliminated. Thus we should carry out the long division with these two polynomials written in *reverse* order. We stop the division when the order of the remainder becomes z^{-1}. Then we obtain

$$X(z) = 1 + 2z^{-1} + \frac{\frac{1}{6}z^{-1}}{1 + \frac{5}{6}z^{-1} + \frac{1}{6}z^{-2}}$$

In general, any improper rational function ($M \geq N$) can be expressed as

$$X(z) = \frac{B(z)}{A(z)} = c_0 + c_1 z^{-1} + \cdots + c_{M-N} z^{-(M-N)} + \frac{B_1(z)}{A(z)} \tag{3.4.11}$$

The inverse z-transform of the polynomial can easily be found by inspection. We focus our attention on the inversion of proper rational transforms, since any improper function can be transformed into a proper function by using (3.4.11). We carry out the development in two steps. First, we perform a partial fraction expansion of the proper rational function and then we invert each of the terms.

Let $X(z)$ be a proper rational function, that is,

$$X(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \tag{3.4.12}$$

where

$$a_N \neq 0 \qquad \text{and} \qquad M < N$$

To simplify our discussion we eliminate negative powers of z by multiplying both the numerator and denominator of (3.4.12) by z^N. This results in

$$X(z) = \frac{b_0 z^N + b_1 z^{N-1} + \cdots + b_M z^{N-M}}{z^N + a_1 z^{N-1} + \cdots + a_N} \tag{3.4.13}$$

which contains only positive powers of z. Since $N > M$, the function

$$\frac{X(z)}{z} = \frac{b_0 z^{N-1} + b_1 z^{N-2} + \cdots + b_M z^{N-M-1}}{z^N + a_1 z^{N-1} + \cdots + a_N} \tag{3.4.14}$$

is also always proper.

Our task in performing a partial-fraction expansion is to express (3.4.14) or, equivalently, (3.4.12) as a sum of simple fractions. For this purpose we first factor the denominator polynomial in (3.4.14) into factors that contain the poles $p_1, p_2, \ldots, p_N$ of $X(z)$. We distinguish two cases.

Distinct poles. Suppose that the poles $p_1, p_2, \ldots, p_N$ are all different (distinct). Then we seek an expansion of the form

$$\frac{X(z)}{z} = \frac{A_1}{z - p_1} + \frac{A_2}{z - p_2} + \cdots + \frac{A_N}{z - p_N} \tag{3.4.15}$$

The problem is to determine the coefficients $A_1, A_2, \ldots, A_N$. There are two ways to solve this problem, as illustrated in the following example.

EXAMPLE 3.4.5

Determine the partial-fraction expansion of the proper function

$$X(z) = \frac{1}{1 - 1.5z^{-1} + 0.5z^{-2}} \tag{3.4.16}$$

Solution. First we eliminate the negative powers, by multiplying both numerator and denominator by z^2. Thus

$$X(z) = \frac{z^2}{z^2 - 1.5z + 0.5}$$

The poles of $X(z)$ are $p_1 = 1$ and $p_2 = 0.5$. Consequently, the expansion of the form (3.4.15) is

$$\frac{X(z)}{z} = \frac{z}{(z-1)(z-0.5)} = \frac{A_1}{z-1} + \frac{A_2}{z-0.5} \tag{3.4.17}$$

A very simple method to determine A_1 and A_2 is to multiply the equation by the denominator term $(z-1)(z-0.5)$. Thus we obtain

$$z = (z - 0.5)A_1 + (z-1)A_2 \tag{3.4.18}$$

Now if we set $z = p_1 = 1$ in (3.4.18), we eliminate the term involving A_2. Hence

$$1 = (1 - 0.5)A_1$$

Thus we obtain the result $A_1 = 2$. Next we return to (3.4.18) and set $z = p_2 = 0.5$, thus eliminating the term involving A_1, so we have

$$0.5 = (0.5 - 1)A_2$$

and hence $A_2 = -1$. Therefore, the result of the partial-fraction expansion is

$$\frac{X(z)}{z} = \frac{2}{z-1} - \frac{1}{z-0.5} \tag{3.4.19}$$

The example given above suggests that we can determine the coefficients $A_1, A_2, \ldots, A_N$, by multiplying both sides of (3.4.15) by each of the terms $(z - p_k)$, $k = 1, 2, \ldots, N$, and evaluating the resulting expressions at the corresponding pole positions, $p_1, p_2, \ldots, p_N$. Thus we have, in general,

$$\frac{(z-p_k)X(z)}{z} = \frac{(z-p_k)A_1}{z-p_1} + \cdots + A_k + \cdots + \frac{(z-p_k)A_N}{z-p_N} \tag{3.4.20}$$

Consequently, with $z = p_k$, (3.4.20) yields the kth coefficient as

$$A_k = \left.\frac{(z-p_k)X(z)}{z}\right|_{z=p_k}, \qquad k = 1, 2, \ldots, N \tag{3.4.21}$$

EXAMPLE 3.4.6

Determine the partial-fraction expansion of

$$X(z) = \frac{1 + z^{-1}}{1 - z^{-1} + 0.5z^{-2}} \tag{3.4.22}$$

Solution. To eliminate negative powers of z in (3.4.22), we multiply both numerator and denominator by z^2. Thus

$$\frac{X(z)}{z} = \frac{z + 1}{z^2 - z + 0.5}$$

The poles of $X(z)$ are complex conjugates

$$p_1 = \frac{1}{2} + j\frac{1}{2}$$

and

$$p_2 = \frac{1}{2} - j\frac{1}{2}$$

Since $p_1 \neq p_2$, we seek an expansion of the form (3.4.15). Thus

$$\frac{X(z)}{z} = \frac{z + 1}{(z - p_1)(z - p_2)} = \frac{A_1}{z - p_1} + \frac{A_2}{z - p_2}$$

To obtain A_1 and A_2, we use the formula (3.4.21). Thus we obtain

$$A_1 = \left.\frac{(z - p_1)X(z)}{z}\right|_{z=p_1} = \left.\frac{z + 1}{z - p_2}\right|_{z=p_1} = \frac{\frac{1}{2} + j\frac{1}{2} + 1}{\frac{1}{2} + j\frac{1}{2} - \frac{1}{2} + j\frac{1}{2}} = \frac{1}{2} - j\frac{3}{2}$$

$$A_2 = \left.\frac{(z - p_2)X(z)}{z}\right|_{z=p_2} = \left.\frac{z + 1}{z - p_1}\right|_{z=p_2} = \frac{\frac{1}{2} - j\frac{1}{2} + 1}{\frac{1}{2} - j\frac{1}{2} - \frac{1}{2} - j\frac{1}{2}} = \frac{1}{2} + j\frac{3}{2}$$

The expansion (3.4.15) and the formula (3.4.21) hold for both real and complex poles. The only constraint is that all poles be distinct. We also note that $A_2 = A_1^*$. It can be easily seen that this is a consequence of the fact that $p_2 = p_1^*$. In other words, *complex-conjugate poles result in complex-conjugate coefficients in the partial-fraction expansion.* This simple result will prove very useful later in our discussion.

Multiple-order poles. If $X(z)$ has a pole of multiplicity l, that is, it contains in its denominator the factor $(z - p_k)^l$, then the expansion (3.4.15) is no longer true. In this case a different expansion is needed. First, we investigate the case of a double pole (i.e., $l = 2$).

EXAMPLE 3.4.7

Determine the partial-fraction expansion of

$$X(z) = \frac{1}{(1+z^{-1})(1-z^{-1})^2} \tag{3.4.23}$$

Solution. First, we express (3.4.23) in terms of positive powers of z, in the form

$$\frac{X(z)}{z} = \frac{z^2}{(z+1)(z-1)^2}$$

$X(z)$ has a simple pole at $p_1 = -1$ and a double pole $p_2 = p_3 = 1$. In such a case the appropriate partial-fraction expansion is

$$\frac{X(z)}{z} = \frac{z^2}{(z+1)(z-1)^2} = \frac{A_1}{z+1} + \frac{A_2}{z-1} + \frac{A_3}{(z-1)^2} \tag{3.4.24}$$

The problem is to determine the coefficients A_1, A_2, and A_3.

We proceed as in the case of distinct poles. To determine A_1, we multiply both sides of (3.4.24) by $(z+1)$ and evaluate the result at $z = -1$. Thus (3.4.24) becomes

$$\frac{(z+1)X(z)}{z} = A_1 + \frac{z+1}{z-1}A_2 + \frac{z+1}{(z-1)^2}A_3$$

which, when evaluated at $z = -1$, yields

$$A_1 = \left.\frac{(z+1)X(z)}{z}\right|_{z=-1} = \frac{1}{4}$$

Next, if we multiply both sides of (3.4.24) by $(z-1)^2$, we obtain

$$\frac{(z-1)^2X(z)}{z} = \frac{(z-1)^2}{z+1}A_1 + (z-1)A_2 + A_3 \tag{3.4.25}$$

Now, if we evaluate (3.4.25) at $z = 1$, we obtain A_3. Thus

$$A_3 = \left.\frac{(z-1)2X(z)}{z}\right|_{z=1} = \frac{1}{2}$$

The remaining coefficient A_2 can be obtained by differentiating both sides of (3.4.25) with respect to z and evaluating the result at $z = 1$. Note that it is not necessary formally to carry out the differentiation of the right-hand side of (3.4.25), since all terms except A_2 vanish when we set $z = 1$. Thus

$$A_2 = \frac{d}{dz}\left[\frac{(z-1)^2X(z)}{z}\right]_{z=1} = \frac{3}{4} \tag{3.4.26}$$

The generalization of the procedure in the example above to the case of an mth-order pole $(z - p_k)^m$ is straightforward. The partial-fraction expansion must contain the terms

$$\frac{A_{1k}}{z-p_k} + \frac{A_{2k}}{(z-p_k)^2} + \cdots + \frac{A_{mk}}{(z-p_k)^m}$$

The coefficients $\{A_{ik}\}$ can be evaluated through differentiation as illustrated in Example 3.4.7 for $m = 2$.

Now that we have performed the partial-fraction expansion, we are ready to take the final step in the inversion of $X(z)$. First, let us consider the case in which $X(z)$ contains distinct poles. From the partial-fraction expansion (3.4.15), it easily follows that

$$X(z) = A_1 \frac{1}{1 - p_1 z^{-1}} + A_2 \frac{1}{1 - p_2 z^{-1}} + \cdots + A_N \frac{1}{1 - p_N z^{-1}} \tag{3.4.27}$$

The inverse z-transform, $x(n) = Z^{-1}\{X(z)\}$, can be obtained by inverting each term in (3.4.27) and taking the corresponding linear combination. From Table 3.3 it follows that these terms can be inverted using the formula

$$Z^{-1}\left\{\frac{1}{1 - p_k z^{-1}}\right\} = \begin{cases} (p_k)^n u(n), & \text{if ROC: } |z| > |p_k| \\ & \text{(causal signals)} \\ -(p_k)^n u(-n-1), & \text{if ROC: } |z| < |p_k| \\ & \text{(anticausal signals)} \end{cases} \tag{3.4.28}$$

If the signal $x(n)$ is causal, the ROC is $|z| > p_{\max}$, where $p_{\max} = \max\{|p_1|, |p_2|, \ldots, |p_N|\}$. In this case all terms in (3.4.27) result in causal signal components and the signal $x(n)$ is given by

$$x(n) = (A_1 p_1^n + A_2 p_2^n + \cdots + A_N p_N^n) u(n) \tag{3.4.29}$$

If all poles are real, (3.4.29) is the desired expression for the signal $x(n)$. Thus a causal signal, having a z-transform that contains real and distinct poles, is a linear combination of real exponential signals.

Suppose now that all poles are distinct but some of them are complex. In this case some of the terms in (3.4.27) result in complex exponential components. However, if the signal $x(n)$ is real, we should be able to reduce these terms into real components. If $x(n)$ is real, the polynomials appearing in $X(z)$ have real coefficients. In this case, as we have seen in Section 3.3, if p_j is a pole, its complex conjugate p_j^* is also a pole. As was demonstrated in Example 3.4.6, the corresponding coefficients in the partial-fraction expansion are also complex conjugates. Thus the contribution of two complex-conjugate poles is of the form

$$x_k(n) = [A_k (p_k)^n + A_k^* (p_k^*)^n] u(n) \tag{3.4.30}$$

These two terms can be combined to form a real signal component. First, we express A_j and p_j in polar form (i.e., amplitude and phase) as

$$A_k = |A_k| e^{j\alpha_k} \tag{3.4.31}$$

$$p_k = r_k e^{j\beta_k} \tag{3.4.32}$$

where α_k and β_k are the phase components of A_k and p_k. Substitution of these relations into (3.4.30) gives

$$x_k(n) = |A_k| r_k^n [e^{j(\beta_k n + \alpha_k)} + e^{-j(\beta_k n + \alpha_k)}] u(n)$$

or, equivalently,

$$x_k(n) = 2|A_k|r_k^n \cos(\beta_k n + \alpha_k)u(n) \tag{3.4.33}$$

Thus we conclude that

$$Z^{-1}\left(\frac{A_k}{1 - p_k z^{-1}} + \frac{A_k^*}{1 - p_k^* z^{-1}}\right) = 2|A_k|r_k^n \cos(\beta_k n + \alpha_k)u(n) \tag{3.4.34}$$

if the ROC is $|z| > |p_k| = r_k$.

From (3.4.34) we observe that each pair of complex-conjugate poles in the z-domain results in a causal sinusoidal signal component with an exponential envelope. The distance r_k of the pole from the origin determines the exponential weighting (growing if $r_k > 1$, decaying if $r_k < 1$, constant if $r_k = 1$). The angle of the poles with respect to the positive real axis provides the frequency of the sinusoidal signal. The zeros, or equivalently the numerator of the rational transform, affect only indirectly the amplitude and the phase of $x_k(n)$ through A_k.

In the case of *multiple* poles, either real or complex, the inverse transform of terms of the form $A/(z - p_k)^n$ is required. In the case of a double pole the following transform pair (see Table 3.3) is quite useful:

$$Z^{-1}\left\{\frac{pz^{-1}}{(1 - pz^{-1})^2}\right\} = np^n u(n) \tag{3.4.35}$$

provided that the ROC is $|z| > |p|$. The generalization to the case of poles with higher multiplicity is obtained by using multiple differentiation.

EXAMPLE 3.4.8

Determine the inverse z-transform of

$$X(z) = \frac{1}{1 - 1.5z^{-1} + 0.5z^{-2}}$$

if

(a) ROC: $|z| > 1$

(b) ROC: $|z| < 0.5$

(c) ROC: $0.5 < |z| < 1$

Solution. This is the same problem that we treated in Example 3.4.2. The partial-fraction expansion for $X(z)$ was determined in Example 3.4.5. The partial-fraction expansion of $X(z)$ yields

$$X(z) = \frac{2}{1 - z^{-1}} - \frac{1}{1 - 0.5z^{-1}} \tag{3.4.36}$$

To invert $X(z)$ we should apply (3.4.28) for $p_1 = 1$ and $p_2 = 0.5$. However, this requires the specification of the corresponding ROC.

(a) In the case when the ROC is $|z| > 1$, the signal $x(n)$ is causal and both terms in (3.4.36) are causal terms. According to (3.4.28), we obtain

$$x(n) = 2(1)^n u(n) - (0.5)^n u(n) = (2 - 0.5^n)u(n) \tag{3.4.37}$$

which agrees with the result in Example 3.4.2(a).

(b) When the ROC is $|z| < 0.5$, the signal $x(n)$ is anticausal. Thus both terms in (3.4.36) result in anticausal components. From (3.4.28) we obtain

$$x(n) = [-2 + (0.5)^n]u(-n-1) \tag{3.4.38}$$

(c) In this case the ROC $0.5 < |z| < 1$ is a ring, which implies that the signal $x(n)$ is two-sided. Thus one of the terms corresponds to a causal signal and the other to an anticausal signal. Obviously, the given ROC is the overlapping of the regions $|z| > 0.5$ and $|z| < 1$. Hence the pole $p_2 = 0.5$ provides the causal part and the pole $p_1 = 1$ the anticausal. Thus

$$x(n) = -2(1)^n u(-n-1) - (0.5)^n u(n) \tag{3.4.39}$$

EXAMPLE 3.4.9

Determine the causal signal $x(n)$ whose z-transform is given by

$$X(z) = \frac{1 + z^{-1}}{1 - z^{-1} + 0.5z^{-2}}$$

Solution. In Example 3.4.6 we have obtained the partial-fraction expansion as

$$X(z) = \frac{A_1}{1 - p_1 z^{-1}} + \frac{A_2}{1 - p_2 z^{-1}}$$

where

$$A_1 = A_2^* = \frac{1}{2} - j\frac{3}{2}$$

and

$$p_1 = p_2^* = \frac{1}{2} + j\frac{1}{2}$$

Since we have a pair of complex-conjugate poles, we should use (3.4.34). The polar forms of A_1 and p_1 are

$$A_1 = \frac{\sqrt{10}}{2} e^{-j71.565}$$

$$p_1 = \frac{1}{\sqrt{2}} e^{j\pi/4}$$

Hence

$$x(n) = \sqrt{10}\left(\frac{1}{\sqrt{2}}\right)^n \cos\left(\frac{\pi n}{4} - 71.565^\circ\right) u(n)$$

EXAMPLE 3.4.10

Determine the causal signal $x(n)$ having the z-transform

$$X(z) = \frac{1}{(1 + z^{-1})(1 - z^{-1})^2}$$

Solution. From Example 3.4.7 we have

$$X(z) = \frac{1}{4}\frac{1}{1 + z^{-1}} + \frac{3}{4}\frac{1}{1 - z^{-1}} + \frac{1}{2}\frac{z^{-1}}{(1 - z^{-1})2}$$

By applying the inverse transform relations in (3.4.28) and (3.4.35), we obtain

$$x(n) = \frac{1}{4}(-1)^n u(n) + \frac{3}{4}u(n) + \frac{1}{2}nu(n) = \left[\frac{1}{4}(-1)^n + \frac{3}{4} + \frac{n}{2}\right]u(n)$$

3.4.4 Decomposition of Rational z-Transforms

At this point it is appropriate to discuss some additional issues concerning the decomposition of rational z-transforms, which will prove very useful in the implementation of discrete-time systems.

Suppose that we have a rational z-transform $X(z)$ expressed as

$$X(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = b_0 \frac{\prod_{k=1}^{M}(1 - z_k z^{-1})}{\prod_{k=1}^{N}(1 - p_k z^{-1})} \tag{3.4.40}$$

where, for simplicity, we have assumed that $a_0 \equiv 1$. If $M \geq N$ [i.e., $X(z)$ is improper], we convert $X(z)$ to a sum of a polynomial and a proper function

$$X(z) = \sum_{k=0}^{M-N} c_k z^{-k} + X_{\text{pr}}(z) \tag{3.4.41}$$

If the poles of $X_{\text{pr}}(z)$ are distinct, it can be expanded in partial fractions as

$$X_{\text{pr}}(z) = A_1 \frac{1}{1 - p_1 z^{-1}} + A_2 \frac{1}{1 - p_2 z^{-1}} + \cdots + A_N \frac{1}{1 - p_N z^{-1}} \tag{3.4.42}$$

As we have already observed, there may be some complex-conjugate pairs of poles in (3.4.42). Since we usually deal with real signals, we should avoid complex coefficients in our decomposition. This can be achieved by grouping and combining terms containing complex-conjugate poles, in the following way:

$$\begin{aligned} \frac{A}{1 - pz^{-1}} + \frac{A^*}{1 - p^* z^{-1}} &= \frac{A - Ap^* z^{-1} + A^* - A^* p z^{-1}}{1 - pz^{-1} - p^* z^{-1} + pp^* z^{-2}} \\ &= \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} \end{aligned} \tag{3.4.43}$$

where

$$\begin{aligned} b_0 &= 2\,\text{Re}(A), & a_1 &= -2\,\text{Re}(p) \\ b_1 &= 2\,\text{Re}(Ap*), & a_2 &= |p|^2 \end{aligned} \tag{3.4.44}$$

are the desired coefficients. Obviously, any rational transform of the form (3.4.43) with coefficients given by (3.4.44), which is the case when $a_1^2 - 4a_2 < 0$, can be inverted using (3.4.34). By combining (3.4.41), (3.4.42), and (3.4.43) we obtain a

partial-fraction expansion for the z-transform with *distinct* poles that contains real coefficients. The general result is

$$X(z) = \sum_{k=0}^{M-N} c_k z^{-k} + \sum_{k=1}^{K_1} \frac{b_k}{1 + a_k z^{-1}} + \sum_{k=1}^{K_2} \frac{b_{0k} + b_{1k} z^{-1}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}} \tag{3.4.45}$$

where $K_1 + 2K_2 = N$. Obviously, if $M = N$, the first term is just a constant, and when $M < N$, this term vanishes. When there are also multiple poles, some additional higher-order terms should be included in (3.4.45).

An alternative form is obtained by expressing $X(z)$ as a product of simple terms as in (3.4.40). However, the complex-conjugate poles and zeros should be combined to avoid complex coefficients in the decomposition. Such combinations result in second-order rational terms of the following form:

$$\frac{(1 - z_k z^{-1})(1 - z_k^* z^{-1})}{(1 - p_k z^{-1})(1 - p_k^* z^{-1})} = \frac{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}} \tag{3.4.46}$$

where

$$\begin{aligned} b_{1k} &= -2\,\mathrm{Re}(z_k), \qquad & a_{1k} &= -2\,\mathrm{Re}(p_k) \\ b_{2k} &= |z_k|^2, & a_{2k} &= |p_k|^2 \end{aligned} \tag{3.4.47}$$

Assuming for simplicity that $M = N$, we see that $X(z)$ can be decomposed in the following way:

$$X(z) = b_0 \prod_{k=1}^{K_1} \frac{1 + b_k z^{-1}}{1 + a_k z^{-1}} \prod_{k=1}^{K_2} \frac{1 + b_{1k} z^{-1} + b_{2k} z^{-2}}{1 + a_{1k} z^{-1} + a_{2k} z^{-2}} \tag{3.4.48}$$

where $N = K_1 + 2K_2$. We will return to these important forms in Chapters 9 and 10.

3.5 Analysis of Linear Time-Invariant Systems in the z-Domain

In Section 3.3.3 we introduced the system function of a linear time-invariant system and related it to the unit sample response and to the difference equation description of systems. In this section we describe the use of the system function in the determination of the response of the system to some excitation signal. In Section 3.6.3, we extend this method of analysis to nonrelaxed systems. Our attention is focused on the important class of pole–zero systems represented by linear constant-coefficient difference equations with arbitrary initial conditions.

We also consider the topic of stability of linear time-invariant systems and describe a test for determining the stability of a system based on the coefficients of the denominator polynomial in the system function. Finally, we provide a detailed analysis of second-order systems, which form the basic building blocks in the realization of higher-order systems.

3.5.1 Response of Systems with Rational System Functions

Let us consider a pole–zero system described by the general linear constant-coefficient difference equation in (3.3.7) and the corresponding system function in (3.3.8). We represent $H(z)$ as a ratio of two polynomials $B(z)/A(z)$, where $B(z)$ is the numerator polynomial that contains the zeros of $H(z)$, and $A(z)$ is the denominator polynomial that determines the poles of $H(z)$. Furthermore, let us assume that the input signal $x(n)$ has a rational z-transform $X(z)$ of the form

$$X(z) = \frac{N(z)}{Q(z)} \tag{3.5.1}$$

This assumption is not overly restrictive, since, as indicated previously, most signals of practical interest have rational z-transforms.

If the system is initially relaxed, that is, the initial conditions for the difference equation are zero, $y(-1) = y(-2) = \cdots = y(-N) = 0$, the z-transform of the output of the system has the form

$$Y(z) = H(z)X(z) = \frac{B(z)N(z)}{A(z)Q(z)} \tag{3.5.2}$$

Now suppose that the system contains simple poles $p_1, p_2, \ldots, p_N$ and the z-transform of the input signal contains poles $q_1, q_2, \ldots, q_L$, where $p_k \neq q_m$ for all $k = 1, 2, \ldots, N$ and $m = 1, 2, \ldots, L$. In addition, we assume that the zeros of the numerator polynomials $B(z)$ and $N(z)$ do not coincide with the poles $\{p_k\}$ and $\{q_k\}$, so that there is no pole–zero cancellation. Then a partial-fraction expansion of $Y(z)$ yields

$$Y(z) = \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}} + \sum_{k=1}^{L} \frac{Q_k}{1 - q_k z^{-1}} \tag{3.5.3}$$

The inverse transform of $Y(z)$ yields the output signal from the system in the form

$$y(n) = \sum_{k=1}^{N} A_k (p_k)^n u(n) + \sum_{k=1}^{L} Q_k (q_k)^n u(n) \tag{3.5.4}$$

We observe that the output sequence $y(n)$ can be subdivided into two parts. The first part is a function of the poles $\{p_k\}$ of the system and is called the *natural response* of the system. The influence of the input signal on this part of the response is through the scale factors $\{A_k\}$. The second part of the response is a function of the poles $\{q_k\}$ of the input signal and is called the *forced response* of the system. The influence of the system on this response is exerted through the scale factors $\{Q_k\}$.

We should emphasize that the scale factors $\{A_k\}$ and $\{Q_k\}$ are functions of both sets of poles $\{p_k\}$ and $\{q_k\}$. For example, if $X(z) = 0$ so that the input is zero, then $Y(z) = 0$, and consequently, the output is zero. Clearly, then, the natural response of the system is zero. This implies that the natural response of the system is different from the zero-input response.

When $X(z)$ and $H(z)$ have one or more poles in common or when $X(z)$ and/or $H(z)$ contain multiple-order poles, then $Y(z)$ will have multiple-order poles. Consequently, the partial-fraction expansion of $Y(z)$ will contain factors of the form $1/(1 - p_l z^{-1})^k$, $k = 1, 2, \ldots, m$, where m is the pole order. The inversion of these factors will produce terms of the form $n^{k-1} p_l^n$ in the output $y(n)$ of the system, as indicated in Section 3.4.3.

3.5.2 Transient and Steady-State Responses

As we have seen from our previous discussion, the zero-state response of a system to a given input can be separated into two components, the natural response and the forced response. The natural response of a causal system has the form

$$y_{\text{nr}}(n) = \sum_{k=1}^{N} A_k (p_k)^n u(n) \tag{3.5.5}$$

where $\{p_k\}$, $k = 1, 2, \ldots, N$ are the poles of the system and $\{A_k\}$ are scale factors that depend on the initial conditions and on the characteristics of the input sequence.

If $|p_k| < 1$ for all k, then, $y_{\text{nr}}(n)$ decays to zero as n approaches infinity. In such a case we refer to the natural response of the system as the *transient response*. The rate at which $y_{\text{nr}}(n)$ decays toward zero depends on the magnitude of the pole positions. If all the poles have small magnitudes, the decay is very rapid. On the other hand, if one or more poles are located near the unit circle, the corresponding terms in $y_{\text{nr}}(n)$ will decay slowly toward zero and the transient will persist for a relatively long time.

The forced response of the system has the form

$$y_{\text{fr}}(n) = \sum_{k=1}^{L} Q_k (q_k)^n u(n) \tag{3.5.6}$$

where $\{q_k\}$, $k = 1, 2, \ldots, L$ are the poles in the forcing function and $\{Q_k\}$ are scale factors that depend on the input sequence and on the characteristics of the system. If all the poles of the input signal fall inside the unit circle, $y_{\text{fr}}(n)$ will decay toward zero as n approaches infinity, just as in the case of the natural response. This should not be surprising since the input signal is also a transient signal. On the other hand, when the causal input signal is a sinusoid, the poles fall on the unit circle and consequently, the forced response is also a sinusoid that persists for all $n \geq 0$. In this case, the forced response is called the *steady-state response* of the system. Thus, for the system to sustain a steady-state output for $n \geq 0$, the input signal must persist for all $n \geq 0$.

The following example illustrates the presence of the steady-state response.

EXAMPLE 3.5.1

Determine the transient and steady-state responses of the system characterized by the difference equation

$$y(n) = 0.5y(n-1) + x(n)$$

when the input signal is $x(n) = 10\cos(\pi n/4)u(n)$. The system is initially at rest (i.e., it is relaxed).

Solution. The system function for this system is

$$H(z) = \frac{1}{1 - 0.5z^{-1}}$$

and therefore the system has a pole at $z = 0.5$. The z-transform of the input signal is (from Table 3.3)

$$X(z) = \frac{10(1 - (1/\sqrt{2})z^{-1})}{1 - \sqrt{2}z^{-1} + z^{-2}}$$

Consequently,

$$\begin{aligned} Y(z) &= H(z)X(z) \\ &= \frac{10(1 - (1/\sqrt{2})z^{-1})}{(1 - 0.5z^{-1})(1 - e^{j\pi/4}z^{-1})(1 - e^{-j\pi/4}z^{-1})} \\ &= \frac{6.3}{1 - 0.5z^{-1}} + \frac{6.78e^{-j28.7^\circ}}{1 - e^{j\pi/4}z^{-1}} + \frac{6.78e^{j28.7^\circ}}{1 - e^{-j\pi/4}z^{-1}} \end{aligned}$$

The natural or transient response is

$$y_{\text{nr}}(n) = 6.3(0.5)^n u(n)$$

and the forced or steady-state response is

$$\begin{aligned} y_{\text{fr}}(n) &= [6.78e^{-j28.7}(e^{j\pi n/4}) + 6.78e^{j28.7}e^{-j\pi n/4}]u(n) \\ &= 13.56\cos\left(\frac{\pi}{4}n - 28.7^\circ\right)u(n) \end{aligned}$$

Thus we see that the steady-state response persists for all $n \geq 0$, just as the input signal persists for all $n \geq 0$.

3.5.3 Causality and Stability

As defined previously, a causal linear time-invariant system is one whose unit sample response $h(n)$ satisfies the condition

$$h(n) = 0, \qquad n < 0$$

We have also shown that the ROC of the z-transform of a causal sequence is the exterior of a circle. Consequently, *a linear time-invariant system is causal if and only if the ROC of the system function is the exterior of a circle of radius $r < \infty$, including the point $z = \infty$.*

The stability of a linear time-invariant system can also be expressed in terms of the characteristics of the system function. As we recall from our previous discussion, a necessary and sufficient condition for a linear time-invariant system to be BIBO stable is

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

In turn, this condition implies that $H(z)$ must contain the unit circle within its ROC.

Indeed, since

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$$

it follows that

$$|H(z)| \leq \sum_{n=-\infty}^{\infty} |h(n)z^{-n}| = \sum_{n=-\infty}^{\infty} |h(n)||z^{-n}|$$

When evaluated on the unit circle (i.e., $|z| = 1$),

$$|H(z)| \leq \sum_{n=-\infty}^{\infty} |h(n)|$$

Hence, if the system is BIBO stable, the unit circle is contained in the ROC of $H(z)$. The converse is also true. Therefore, a *linear time-invariant system* is BIBO stable if and only if the ROC of the system function includes the unit circle.

We should stress, however, that the conditions for causality and stability are different and that one does not imply the other. For example, a causal system may be stable or unstable, just as a noncausal system may be stable or unstable. Similarly, an unstable system may be either causal or noncausal, just as a stable system may be causal or noncausal.

For a causal system, however, the condition on stability can be narrowed to some extent. Indeed, a causal system is characterized by a system function $H(z)$ having as a ROC the exterior of some circle of radius r. For a stable system, the ROC must include the unit circle. Consequently, a causal and stable system must have a system function that converges for $|z| > r < 1$. Since the ROC cannot contain any poles of $H(z)$, it follows that a *causal linear time-invariant system* is BIBO stable if and only if all the poles of $H(z)$ are inside the unit circle.

EXAMPLE 3.5.2

A linear time-invariant system is characterized by the system function

$$H(z) = \frac{3 - 4z^{-1}}{1 - 3.5z^{-1} + 1.5z^{-2}}$$
$$= \frac{1}{1 - \frac{1}{2}z^{-1}} + \frac{2}{1 - 3z^{-1}}$$

Specify the ROC of $H(z)$ and determine $h(n)$ for the following conditions:

(a) The system is stable.

(b) The system is causal.

(c) The system is anticausal.

Solution. The system has poles at $z = \frac{1}{2}$ and $z = 3$.

(a) Since the system is stable, its ROC must include the unit circle and hence it is $\frac{1}{2} < |z| < 3$. Consequently, $h(n)$ is noncausal and is given as

$$h(n) = (\frac{1}{2})^n u(n) - 2(3)^n u(-n-1)$$

(b) Since the system is causal, its ROC is $|z| > 3$. In this case

$$h(n) = (\frac{1}{2})^n u(n) + 2(3)^n u(n)$$

This system is unstable.

(c) If the system is anticausal, its ROC is $|z| < 0.5$. Hence

$$h(n) = -[(\frac{1}{2})^n + 2(3)^n]u(-n-1)$$

In this case the system is unstable.

3.5.4 Pole–Zero Cancellations

When a z-transform has a pole that is at the same location as a zero, the pole is canceled by the zero and, consequently, the term containing that pole in the inverse z-transform vanishes. Such pole–zero cancellations are very important in the analysis of pole–zero systems.

Pole–zero cancellations can occur either in the system function itself or in the product of the system function with the z-transform of the input signal. In the first case we say that the order of the system is reduced by one. In the latter case we say that the pole of the system is suppressed by the zero in the input signal, or vice versa. Thus, by properly selecting the position of the zeros of the input signal, it is possible to suppress one or more system modes (pole factors) in the response of the system. Similarly, by proper selection of the zeros of the system function, it is possible to suppress one or more modes of the input signal from the response of the system.

When the zero is located very near the pole but not exactly at the same location, the term in the response has a very small amplitude. For example, nonexact pole–zero cancellations can occur in practice as a result of insufficiant numerical precision used in representing the coefficients of the system. Consequently, one should not attempt to stabilize an inherently unstable system by placing a zero in the input signal at the location of the pole.

EXAMPLE 3.5.3

Determine the unit sample response of the system characterized by the difference equation

$$y(n) = 2.5y(n-1) - y(n-2) + x(n) - 5x(n-1) + 6x(n-2)$$

Solution. The system function is

$$H(z) = \frac{1 - 5z^{-1} + 6z^{-2}}{1 - 2.5z^{-1} + z^{-2}}$$

$$= \frac{1 - 5z^{-1} + 6z^{-2}}{(1 - \frac{1}{2}z^{-1})(1 - 2z^{-1})}$$

This system has poles at $p_1 = 2$ and $p_1 = \frac{1}{2}$. Consequently, at first glance it appears that the unit sample response is

$$Y(z) = H(z)X(z) = \frac{1 - 5z^{-1} + 6z^{-2}}{(1 - \frac{1}{2}z^{-1})(1 - 2z^{-1})}$$

$$= z\left(\frac{A}{z - \frac{1}{2}} + \frac{B}{z - 2}\right)$$

By evaluating the constants at $z = \frac{1}{2}$ and $z = 2$, we find that

$$A = \frac{5}{2}, \qquad B = 0$$

The fact that $B = 0$ indicates that there exists a zero at $z = 2$ which cancels the pole at $z = 2$. In fact, the zeros occur at $z = 2$ and $z = 3$. Consequently, $H(z)$ reduces to

$$H(z) = \frac{1 - 3z^{-1}}{1 - \frac{1}{2}z^{-1}} = \frac{z - 3}{z - \frac{1}{2}}$$

$$= 1 - \frac{2.5z^{-1}}{1 - \frac{1}{2}z^{-1}}$$

and therefore

$$h(n) = \delta(n) - 2.5(\frac{1}{2})^{n-1}u(n - 1)$$

The reduced-order system obtained by canceling the common pole and zero is characterized by the difference equation

$$y(n) = \frac{1}{2}y(n - 1) + x(n) - 3x(n - 1)$$

Although the original system is also BIBO stable due to the pole–zero cancellation, in a practical implementation of this second-order system, we may encounter an instability due to imperfect cancellation of the pole and the zero.

EXAMPLE 3.5.4

Determine the response of the system

$$y(n) = \frac{5}{6}y(n - 1) - \frac{1}{6}y(n - 2) + x(n)$$

to the input signal $x(n) = \delta(n) - \frac{1}{3}\delta(n - 1)$.

Solution. The system function is

$$H(z) = \frac{1}{1 - \frac{5}{6}z^{-1} + \frac{1}{6}z^{-2}}$$

$$= \frac{1}{(1 - \frac{1}{2}z^{-1})(1 - \frac{1}{3}z^{-1})}$$

This system has two poles, one at $z = \frac{1}{2}$ and the other at $z = \frac{1}{3}$. The z-transform of the input signal is

$$X(z) = 1 - \frac{1}{3}z^{-1}$$

In this case the input signal contains a zero at $z = \frac{1}{3}$ which cancels the pole at $z = \frac{1}{3}$. Consequently,

$$Y(z) = H(z)X(z)$$

$$Y(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

and hence the response of the system is

$$y(n) = (\frac{1}{2})^n u(n)$$

Clearly, the mode $(\frac{1}{3})^n$ is suppressed from the output as a result of the pole–zero cancellation.

3.5.5 Multiple-Order Poles and Stability

As we have observed, a necessary and sufficient condition for a causal linear time-invariant system to be BIBO stable is that all its poles lie inside the unit circle. The input signal is bounded if its z-transform contains poles $\{q_k\}$, $k = 1, 2, \ldots, L$, which satisfy the condition $|q_k| \leq 1$ for all k. We note that the forced response of the system, given in (3.5.6), is also bounded, even when the input signal contains one or more distinct poles on the unit circle.

In view of the fact that a bounded input signal may have poles on the unit circle, it might appear that a stable system may also have poles on the unit circle. This is not the case, however, since such a system produces an unbounded response when excited by an input signal that also has a pole at the same position on the unit circle. The following example illustrates this point.

EXAMPLE 3.5.5

Determine the step response of the causal system described by the difference equation

$$y(n) = y(n - 1) + x(n)$$

Solution. The system function for the system is

$$H(z) = \frac{1}{1 - z^{-1}}$$

We note that the system contains a pole on the unit circle at $z = 1$. The z-transform of the input signal $x(n) = u(n)$ is

$$X(z) = \frac{1}{1 - z^{-1}}$$

which also contains a pole at $z = 1$. Hence the output signal has the transform

$$\begin{aligned} Y(z) &= H(z)X(z) \\ &= \frac{1}{(1 - z^{-1})^2} \end{aligned}$$

which contains a double pole at $z = 1$.

The inverse z-transform of $Y(z)$ is

$$y(n) = (n + 1)u(n)$$

which is a ramp sequence. Thus $y(n)$ is unbounded, even when the input is bounded. Consequently, the system is unstable.

Example 3.5.5 demonstrates clearly that BIBO stability requires that the system poles be strictly inside the unit circle. If the system poles are all inside the unit circle and the excitation sequence $x(n)$ contains one or more poles that coincide with the poles of the system, the output $Y(z)$ will contain multiple-order poles. As indicated previously, such multiple-order poles result in an output sequence that contains terms of the form

$$A_k n^b (p_k)^n u(n)$$

where $0 \leq b \leq m - 1$ and m is the order of the pole. If $|p_k| < 1$, these terms decay to zero as n approaches infinity because the exponential factor $(p_k)^n$ dominates the term n^b. Consequently, no bounded input signal can produce an unbounded output signal if the system poles are all inside the unit circle.

Finally, we should state that the only useful systems which contain poles on the unit circle are the digital oscillators discussed in Chapter 5. We call such systems *marginally stable.*

3.5.6 Stability of Second-Order Systems

In this section we provide a detailed analysis of a system having two poles. As we shall see in Chapter 9, two-pole systems form the basic building blocks for the realization of higher-order systems.

Let us consider a causal two-pole system described by the second-order difference equation

$$y(n) = -a_1 y(n - 1) - a_2 y(n - 2) + b_0 x(n) \tag{3.5.7}$$

The system function is

$$\begin{aligned} H(z) = \frac{Y(z)}{X(z)} &= \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-1}} \\ &= \frac{b_0 z^2}{z^2 + a_1 z + a_2} \end{aligned} \tag{3.5.8}$$

This system has two zeros at the origin and poles at

$$p_1, p_2 = -\frac{a_1}{2} \pm \sqrt{\frac{a_1^2 - 4a_2}{4}} \tag{3.5.9}$$

The system is BIBO stable if the poles lie inside the unit circle, that is, if $|p_1| < 1$ and $|p_2| < 1$. These conditions can be related to the values of the coefficients a_1 and a_2. In particular, the roots of a quadratic equation satisfy the relations

$$a_1 = -(p_1 + p_2) \tag{3.5.10}$$

$$a_2 = p_1 p_2 \tag{3.5.11}$$

From (3.5.10) and (3.5.11) we easily obtain the conditions that a_1 and a_2 must satisfy for stability. First, a_2 must satisfy the condition

$$|a_2| = |p_1 p_2| = |p_1||p_2| < 1 \tag{3.5.12}$$

The condition for a_1 can be expressed as

$$|a_1| < 1 + a_2 \tag{3.5.13}$$

Therefore, a two-pole system is stable if and only if the coefficients a_1 and a_2 satisfy the conditions in (3.5.12) and (3.5.13).

The stability conditions given in (3.5.12) and (3.5.13) define a region in the coefficient plane (a_1, a_2), which is in the form of a triangle, as shown in Fig. 3.5.1. The system is stable if and only if the point (a_1, a_2) lies inside the triangle, which we call the *stability triangle*.

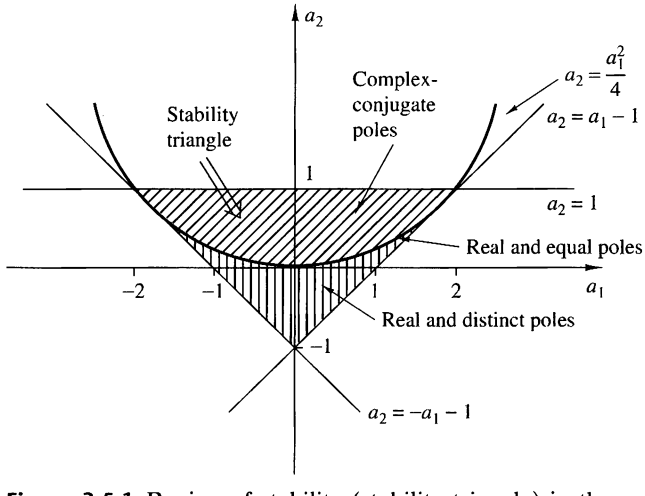


Figure 3.5.1 Region of stability (stability triangle) in the (a_1, a_2) coefficient plane for a second-order system.

The characteristics of the two-pole system depend on the location of the poles or, equivalently, on the location of the point (a_1, a_2) in the stability triangle. The poles of the system may be real or complex conjugate, depending on the value of the discriminant $\Delta = a_1^2 - 4a_2$. The parabola $a_2 = a_1^2/4$ splits the stability triangle into two regions, as illustrated in Fig. 3.5.1. The region below the parabola ($a_1^2 > 4a_2$) corresponds to real and distinct poles. The points on the parabola ($a_1^2 = 4a_2$) result in real and equal (double) poles. Finally, the points above the parabola correspond to complex-conjugate poles.

Additional insight into the behavior of the system can be obtained from the unit sample responses for these three cases.

Real and distinct poles ($a_1^2 > 4a_2$). Since p_1, p_2 are real and $p_1 \neq p_2$, the system function can be expressed in the form

$$H(z) = \frac{A_1}{1 - p_1 z^{-1}} + \frac{A_2}{1 - p_2 z^{-1}} \tag{3.5.14}$$

where

$$A_1 = \frac{b_0 p_1}{p_1 - p_2}, \qquad A_2 = \frac{-b_0 p_2}{p_1 - p_2} \tag{3.5.15}$$

Consequently, the unit sample response is

$$h(n) = \frac{b_0}{p_1 - p_2}(p_1^{n+1} - p_2^{n+1})u(n) \tag{3.5.16}$$

Therefore, the unit sample response is the difference of two decaying exponential sequences. Figure 3.5.2 illustrates a typical graph for $h(n)$ when the poles are distinct.

Real and equal poles ($a_1^2 = 4a_2$). In this case $p_1 = p_2 = p = -a_1/2$. The system function is

$$H(z) = \frac{b_0}{(1 - pz^{-1})^2} \tag{3.5.17}$$

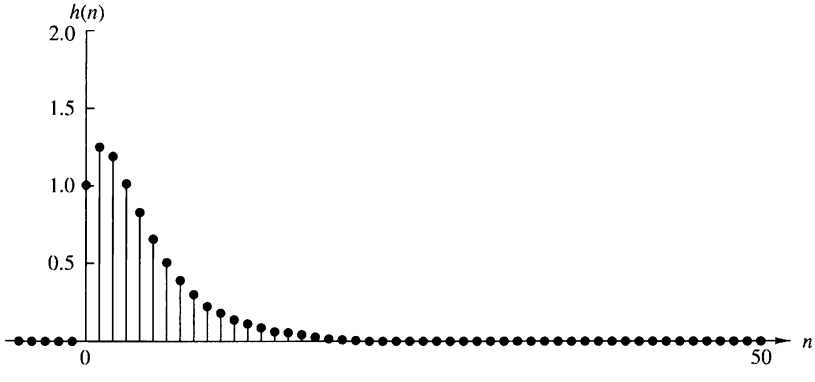


Figure 3.5.2 Plot of $h(n)$ given by (3.5.16) with $p_1 = 0.5$, $p_2 = 0.75$; $h(n) = [1/(p_1 - p_2)](p_1^{n+1} - p_2^{n+1})u(n)$.

and hence the unit sample response of the system is

$$h(n) = b_0(n+1)p^n u(n) \tag{3.5.18}$$

We observe that $h(n)$ is the product of a ramp sequence and a real decaying exponential sequence. The graph of $h(n)$ is shown in Fig. 3.5.3.

Complex-conjugate poles ($a_1^2 < 4a_2$). Since the poles are complex conjugate, the system function can be factored and expressed as

$$\begin{aligned} H(z) &= \frac{A}{1 - pz^{-1}} + \frac{A^*}{1 - p^*z^{-1}} \\ &= \frac{A}{1 - re^{j\omega_0}z^{-1}} + \frac{A^*}{1 - re^{-j\omega_0}z^{-1}} \end{aligned} \tag{3.5.19}$$

where $p = re^{j\omega}$ and $0 < \omega_0 < \pi$. Note that when the poles are complex conjugates, the parameters a_1 and a_2 are related to r and ω_0 according to

$$\begin{aligned} a_1 &= -2r\cos\omega_0 \\ a_2 &= r^2 \end{aligned} \tag{3.5.20}$$

The constant A in the partial-fraction expansion of $H(z)$ is easily shown to be

$$\begin{aligned} A &= \frac{b_0 p}{p - p^*} = \frac{b_0 r e^{j\omega_0}}{r(e^{j\omega_0} - e^{-j\omega_0})} \\ &= \frac{b_0 e^{j\omega_0}}{j2\sin\omega_0} \end{aligned} \tag{3.5.21}$$

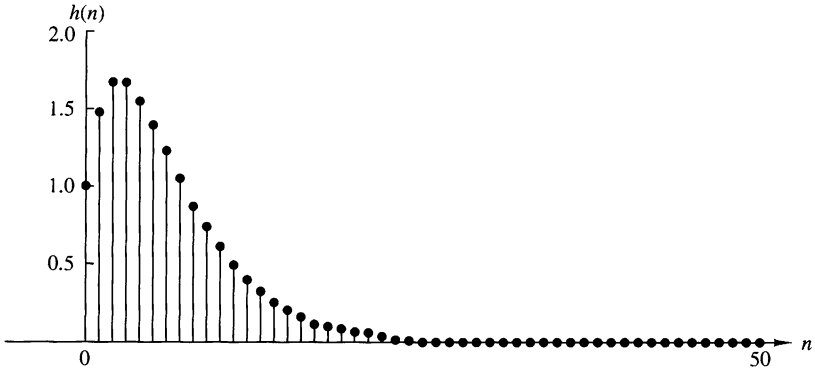


Figure 3.5.3 Plot of $h(n)$ given by (3.5.18) with $p = \frac{3}{4}$; $h(n) = (n+1)p^n u(n)$.

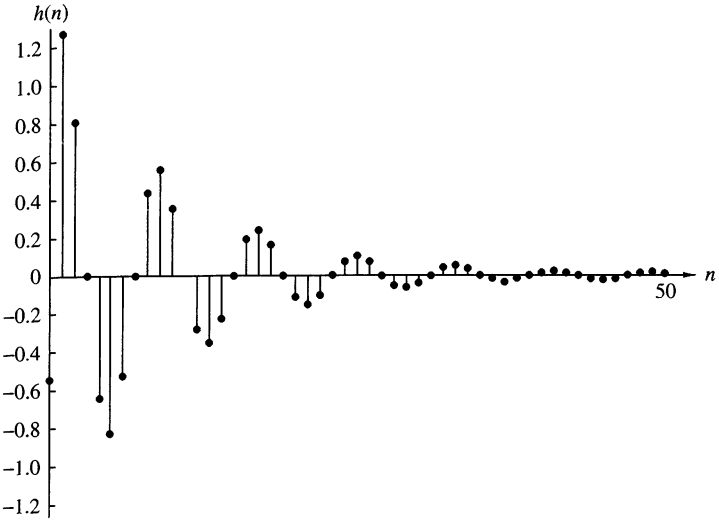


Figure 3.5.4 Plot of $h(n)$ given by (3.5.22) with $b_0 = 1$, $\omega_0 = \pi/4$, $r = 0.9$; $h(n) = [b_0 r^n/(\sin\omega_0)]\sin[(n+1)\omega_0]u(n)$.

Consequently, the unit sample response of a system with complex-conjugate poles is

$$\begin{aligned} h(n) &= \frac{b_0 r^n}{\sin\omega_0}\frac{e^{j(n+1)\omega_0} - e^{-j(n+1)\omega_0}}{2j}u(n) \\ &= \frac{b_0 r^n}{\sin\omega_0}\sin(n+1)\omega_0 u(n) \end{aligned} \tag{3.5.22}$$

In this case $h(n)$ has an oscillatory behavior with an exponentially decaying envelope when $r < 1$. The angle ω_0 of the poles determines the frequency of oscillation and the distance r of the poles from the origin determines the rate of decay. When r is close to unity, the decay is slow. When r is close to the origin, the decay is fast. A typical graph of $h(n)$ is illustrated in Fig. 3.5.4.

3.6 The One-sided z-Transform

The two-sided z-transform requires that the corresponding signals be specified for the entire time range $-\infty < n < \infty$. This requirement prevents its use for a very useful family of practical problems, namely the evaluation of the output of nonrelaxed systems. As we recall, these systems are described by difference equations with nonzero initial conditions. Since the input is applied at a finite time, say n_0, both input and output signals are specified for $n \geq n_0$, but by no means are zero for $n < n_0$. Thus the two-sided z-transform cannot be used. In this section we develop the one-sided z-transform which can be used to solve difference equations with initial conditions.

3.6.1 Definition and Properties

The *one-sided* or *unilateral* z-transform of a signal $x(n)$ is defined by

$$X^+(z) \equiv \sum_{n=0}^{\infty} x(n)z^{-n} \tag{3.6.1}$$

We also use the notations $Z^+\{x(n)\}$ and

$$x(n) \overset{z^+}{\longleftrightarrow} X^+(z)$$

The one-sided z-transform differs from the two-sided transform in the lower limit of the summation, which is always zero, whether or not the signal $x(n)$ is zero for $n < 0$ (i.e., causal). Due to this choice of lower limit, the one-sided z-transform has the following characteristics:

1. It does not contain information about the signal $x(n)$ for negative values of time (i.e., for $n < 0$).
2. It is *unique* only for causal signals, because only these signals are zero for $n < 0$.
3. The one-sided z-transform $X^+(z)$ of $x(n)$ is identical to the two-sided z-transform of the signal $x(n)u(n)$. Since $x(n)u(n)$ is causal, the ROC of its transform, and hence the ROC of $X^+(z)$, is always the exterior of a circle. Thus when we deal with one-sided z-transforms, it is not necessary to refer to their ROC.

EXAMPLE 3.6.1

Determine the one-sided z-transform of the signals in Example 3.1.1.

Solution. From the definition (3.6.1), we obtain

$$x_1(n) = \{\underset{\uparrow}{1}, 2, 5, 7, 0, 1\} \overset{z^+}{\longleftrightarrow} X_1^+(z) = 1 + 2z^{-1} + 5z^{-2} + 7z^{-3} + z^{-5}$$

$$x_2(n) = \{1, 2, \underset{\uparrow}{5}, 7, 0, 1\} \overset{z^+}{\longleftrightarrow} X_2^+(z) = 5 + 7z^{-1} + z^{-3}$$

$$x_3(n) = \{\underset{\uparrow}{0}, 0, 1, 2, 5, 7, 0, 1\} \overset{z^+}{\longleftrightarrow} X_3^+(z) = z^{-2} + 2z^{-3} + 5z^{-4} + 7z^{-5} + z^{-7}$$

$$x_4(n) = \{2, 4, \underset{\uparrow}{5}, 7, 0, 1\} \overset{z^+}{\longleftrightarrow} X_4^+(z) = 5 + 7z^{-1} + z^{-3}$$

$$x_5(n) = \delta(n) \overset{z^+}{\longleftrightarrow} X_5^+(z) = 1$$

$$x_6(n) = \delta(n-k), \qquad k > 0 \overset{z^+}{\longleftrightarrow} X_6^+(z) = z^{-k}$$

$$x_7(n) = \delta(n+k), \qquad k > 0 \overset{z^+}{\longleftrightarrow} X_7^+(z) = 0$$

Note that for a noncausal signal, the one-sided z-transform is not unique. Indeed, $X_2^+(z) = X_4^+(z)$ but $x_2(n) \neq x_4(n)$. Also for anticausal signals, $X^+(z)$ is always zero.

Almost all properties we have studied for the two-sided z-transform carry over to the one-sided z-transform with the exception of the *shifting* property.

Shfiting Property

Case 1: Time delay If

$$x(n) \overset{z^+}{\longleftrightarrow} X^+(z)$$

then

$$x(n-k) \overset{z^+}{\longleftrightarrow} z^{-k}[X^+(z) + \sum_{n=1}^{k} x(-n)z^n], \qquad k > 0 \tag{3.6.2}$$

In case $x(n)$ is causal, then

$$x(n-k) \overset{z^+}{\longleftrightarrow} z^{-k}X^+(z) \tag{3.6.3}$$

Proof From the definition (3.6.1) we have

$$Z^+\{x(n-k)\} = z^{-k}\left[\sum_{l=-k}^{-1} x(l)z^{-l} + \sum_{l=0}^{\infty} x(l)z^{-l}\right]$$

$$= z^{-k}\left[\sum_{l=-1}^{-k} x(l)z^{-l} + X^+(z)\right]$$

By changing the index from l to $n = -l$, the result in (3.6.2) is easily obtained.

EXAMPLE 3.6.2

Determine the one-sided z-transform of the signals

(a) $x(n) = a^n u(n)$

(b) $x_1(n) = x(n-2)$ where $x(n) = a^n$

Solution.

(a) From (3.6.1) we easily obtain

$$X^+(z) = \frac{1}{1 - az^{-1}}$$

(b) We will apply the shifting property for $k = 2$. Indeed, we have

$$Z^+\{x(n-2)\} = z^{-2}[X^+(z) + x(-1)z + x(-2)z^2]$$

$$= z^{-2}X^+(z) + x(-1)z^{-1} + x(-2)$$

Since $x(-1) = a^{-1}$, $x(-2) = a^{-2}$, we obtain

$$X_1^+(z) = \frac{z^{-2}}{1 - az^{-1}} + a^{-1}z^{-1} + a^{-2}$$

The meaning of the shifting property can be intuitively explained if we write (3.6.2) as follows:

$$Z^+\{x(n-k)\} = [x(-k) + x(-k+1)z^{-1} + \cdots + x(-1)z^{-k+1}] + z^{-k}X^+(z), \qquad k > 0 \tag{3.6.4}$$

To obtain $x(n-k)(k>0)$ from $x(n)$, we should shift $x(n)$ by k samples to the right. Then k "new" samples, $x(-k)$, $x(-k+1), \ldots, x(-1)$, enter the positive time axis with $x(-k)$ located at time zero. The first term in (3.6.4) stands for the z-transform of these samples. The "old" samples of $x(n-k)$ are the same as those of $x(n)$ simply shifted by k samples to the right. Their z-transform is obviously $z^{-k}X^+(z)$, which is the second term in (3.6.4).

Case 2: Time advance If

$$x(n) \overset{z^+}{\longleftrightarrow} X^+(z)$$

then

$$x(n+k) \overset{z^+}{\longleftrightarrow} z^k \left[X^+(z) - \sum_{n=0}^{k-1} x(n)z^{-n} \right], \qquad k > 0 \tag{3.6.5}$$

Proof From (3.6.1) we have

$$Z^+\{x(n+k)\} = \sum_{n=0}^{\infty} x(n+k)z^{-n} = z^k \sum_{l=k}^{\infty} x(l)z^{-l}$$

where we have changed the index of summation from n to $l = n + k$. Now, from (3.6.1) we obtain

$$X^+(z) = \sum_{l=0}^{\infty} x(l)z^{-l} = \sum_{l=0}^{k-1} x(l)z^{-l} + \sum_{l=k}^{\infty} x(l)z^{-l}$$

By combining the last two relations, we easily obtain (3.6.5).

EXAMPLE 3.6.3

With $x(n)$, as given in Example 3.6.2, determine the one-sided z-transform of the signal

$$x_2(n) = x(n+2)$$

Solution. We will apply the shifting theorem for $k = 2$. From (3.6.5), with $k = 2$, we obtain

$$Z^+\{x(n+2)\} = z^2X^+(z) - x(0)z^2 - x(1)z$$

But $x(0) = 1$, $x(1) = a$, and $X^+(z) = 1/(1 - az^{-1})$. Thus

$$Z^+\{x(n+2)\} = \frac{z^2}{1 - az^{-1}} - z^2 - az$$

The case of a time advance can be intuitively explained as follows. To obtain $x(n+k)$, $k > 0$, we should shift $x(n)$ by k samples to the left. As a result, the samples $x(0), x(1), \ldots, x(k-1)$ "leave" the positive time axis. Thus we first remove their contribution to the $X^+(z)$, and then multiply what remains by z^k to compensate for the shifting of the signal by k samples.

The importance of the shifting property lies in its application to the solution of difference equations with constant coefficients and nonzero initial conditions. This makes the one-sided z-transform a very useful tool for the analysis of recursive linear time-invariant discrete-time systems.

An important theorem useful in the analysis of signals and systems is the final value theorem.

Final Value Theorem. If

$$x(n) \overset{z^+}{\longleftrightarrow} X^+(z)$$

then

$$\lim_{n\to\infty} x(n) = \lim_{z\to 1}(z-1)X^+(z) \tag{3.6.6}$$

The limit in (3.6.6) exists if the ROC of $(z-1)X^+(z)$ includes the unit circle.

The proof of this theorem is left as an exercise for the reader.

This theorem is useful when we are interested in the asymptotic behavior of a signal $x(n)$ and we know its z-transform, but not the signal itself. In such cases, especially if it is complicated to invert $X^+(z)$, we can use the final value theorem to determine the limit of $x(n)$ as n goes to infinity.

EXAMPLE 3.6.4

The impulse response of a relaxed linear time-invariant system is $h(n) = \alpha^n u(n)$, $|\alpha| < 1$. Determine the value of the step response of the system as $n \to \infty$.

Solution. The step response of the system is

$$y(n) = h(n) * x(n)$$

where

$$x(n) = u(n)$$

Obviously, if we excite a causal system with a causal input the output will be causal. Since $h(n)$, $x(n)$, $y(n)$ are causal signals, the one-sided and two-sided z-transforms are identical. From the convolution property (3.2.17) we know that the z-transforms of $h(n)$ and $x(n)$ must be multiplied to yield the z-transform of the output. Thus

$$Y(z) = \frac{1}{1-\alpha z^{-1}}\frac{1}{1-z^{-1}} = \frac{z^2}{(z-1)(z-\alpha)}, \qquad \text{ROC: } |z| > |\alpha|$$

Now

$$(z-1)Y(z) = \frac{z^2}{z-\alpha}, \qquad \text{ROC: } |z| < |\alpha|$$

Since $|\alpha| < 1$, the ROC of $(z-1)Y(z)$ includes the unit circle. Consequently, we can apply (3.6.6) and obtain

$$\lim_{n\to\infty} y(n) = \lim_{z\to 1}\frac{z^2}{z-\alpha} = \frac{1}{1-\alpha}$$

3.6.2 Solution of Difference Equations

The one-sided z-transform is a very efficient tool for the solution of difference equations with nonzero initial conditions. It achieves that by reducing the difference equation relating the two time-domain signals to an equivalent algebraic equation relating their one-sided z-transforms. This equation can be easily solved to obtain the transform of the desired signal. The signal in the time domain is obtained by inverting the resulting z-transform. We will illustrate this approach with two examples.

EXAMPLE 3.6.5

The well-known Fibonacci sequence of integer numbers is obtained by computing each term as the sum of the two previous ones. The first few terms of the sequence are

$$1, 1, 2, 3, 5, 8, \ldots$$

Determine a closed-form expression for the nth term of the Fibonacci sequence.

Solution. Let $y(n)$ be the nth term of the Fibonacci sequence. Clearly, $y(n)$ satisfies the difference equation

$$y(n) = y(n-1) + y(n-2) \tag{3.6.7}$$

with initial conditions

$$y(0) = y(-1) + y(-2) = 1 \tag{3.6.8a}$$

$$y(1) = y(0) + y(-1) = 1 \tag{3.6.8b}$$

From (3.6.8b) we have $y(-1) = 0$. Then (3.6.8a) gives $y(-2) = 1$. Thus we have to determine $y(n)$, $n \geq 0$, which satisfies (3.6.7), with initial conditions $y(-1) = 0$ and $y(-2) = 1$.

By taking the one-sided z-transform of (3.6.7) and using the shifting property (3.6.2), we obtain

$$Y^+(z) = [z^{-1}Y^+(z) + y(-1)] + [z^{-2}Y^+(z) + y(-2) + y(-1)z^{-1}]$$

or

$$Y^+(z) = \frac{1}{1 - z^{-1} - z^2} = \frac{z^2}{z^2 - z - 1} \tag{3.6.9}$$

where we have used the fact that $y(-1) = 0$ and $y(-2) = 1$.

We can invert $Y^+(z)$ by the partial-fraction expansion method. The poles of $Y^+(z)$ are

$$p_1 = \frac{1 + \sqrt{5}}{2}, \qquad p_2 = \frac{1 - \sqrt{5}}{2}$$

and the corresponding coefficients are $A_1 = p_1/\sqrt{5}$ and $A_2 = -p_2/\sqrt{5}$. Therefore,

$$y(n) = \left[\frac{1 + \sqrt{5}}{2\sqrt{5}}\left(\frac{1 + \sqrt{5}}{2}\right)^n - \frac{1 - \sqrt{5}}{2\sqrt{5}}\left(\frac{1 - \sqrt{5}}{2}\right)^n\right]u(n)$$

or, equivalently,

$$y(n) = \frac{1}{\sqrt{5}}\left(\frac{1}{2}\right)^{n+1}\left[\left(1 + \sqrt{5}\right)^{n+1} - \left(1 - \sqrt{5}\right)^{n+1}\right]u(n) \tag{3.6.10}$$

EXAMPLE 3.6.6

Determine the step response of the system

$$y(n) = \alpha y(n-1) + x(n), \qquad -1 < \alpha < 1 \tag{3.6.11}$$

when the initial condition is $y(-1) = 1$.

Solution. By taking the one-sided z-transform of both sides of (3.6.11), we obtain

$$Y^+(z) = \alpha[z^{-1}Y^+(z) + y(-1)] + X^+(z)$$

Upon substitution for $y(-1)$ and $X^+(z)$ and solving for $Y^+(z)$, we obtain the result

$$Y^+(z) = \frac{\alpha}{1-\alpha z^{-1}} + \frac{1}{(1-\alpha z^{-1})(1-z^{-1})} \tag{3.6.12}$$

By performing a partial-fraction expansion and inverse transforming the result, we have

$$\begin{aligned} y(n) &= \alpha^{n+1}u(n) + \frac{1-\alpha^{n+1}}{1-\alpha}u(n) \\ &= \frac{1}{1-\alpha}(1-\alpha^{n+2})u(n) \end{aligned} \tag{3.6.13}$$

3.6.3 Response of Pole–Zero Systems with Nonzero Initial Conditions

Suppose that the signal $x(n)$ is applied to the pole–zero system at $n = 0$. Thus the signal $x(n)$ is assumed to be causal. The effects of all previous input signals to the system are reflected in the initial conditions $y(-1), y(-2), \ldots, y(-N)$. Since the input $x(n)$ is causal and since we are interested in determining the output $y(n)$ for $n \geq 0$, we can use the one-sided z-transform, which allows us to deal with the initial conditions. Thus the one-sided z-transform of (3.3.7) becomes

$$Y^+(z) = -\sum_{k=1}^{N} a_k z^{-k}\left[Y^+(z) + \sum_{n=1}^{k} y(-n)z^n\right] + \sum_{k=0}^{M} b_k z^{-k} X^+(z) \tag{3.6.14}$$

Since $x(n)$ is causal, we can set $X^+(z) = X(z)$. In any case (3.6.14) may be expressed as

$$\begin{aligned} Y^+(z) &= \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} X(z) - \frac{\sum_{k=1}^{N} a_k z^{-k} \sum_{n=1}^{k} y(-n)z^n}{1 + \sum_{k=1}^{N} a_k z^{-k}} \\ &= H(z)X(z) + \frac{N_0(z)}{A(z)} \end{aligned} \tag{3.6.15}$$

where

$$N_0(z) = -\sum_{k=1}^{N} a_k z^{-k} \sum_{n=1}^{k} y(-n) z^n \tag{3.6.16}$$

From (3.6.15) it is apparent that the output of the system with nonzero initial conditions can be subdivided into two parts. The first is the zero-state response of the system, defined in the z-domain as

$$Y_{zs}(z) = H(z)X(z) \tag{3.6.17}$$

The second component corresponds to the output resulting from the nonzero initial conditions. This output is the zero-input response of the system, which is defined in the z-domain as

$$Y_{zi}^{+}(z) = \frac{N_0(z)}{A(z)} \tag{3.6.18}$$

Hence the total response is the sum of these two output components, which can be expressed in the time domain by determining the inverse z-transforms of $Y_{zs}(z)$ and $Y_{zi}(z)$ separately, and then adding the results. Thus

$$y(n) = y_{zs}(n) + y_{zi}(n) \tag{3.6.19}$$

Since the denominator of $Y_{zi}^{+}(z)$, is $A(z)$, its poles are $p_1, p_2, \ldots, p_N$. Consequently, the zero-input response has the form

$$y_{zi}(n) = \sum_{k=1}^{N} D_k (p_k)^n u(n) \tag{3.6.20}$$

This can be added to (3.6.4) and the terms involving the poles $\{p_k\}$ can be combined to yield the total response in the form

$$y(n) = \sum_{k=1}^{N} A_k' (p_k)^n u(n) + \sum_{k=1}^{L} Q_k (q_k)^n u(n) \tag{3.6.21}$$

where, by definition,

$$A_k' = A_k + D_k \tag{3.6.22}$$

This development indicates clearly that the effect of the initial conditions is to alter the natural response of the system through modification of the scale factors $\{A_k\}$. There are no new poles introduced by the nonzero initial conditions. Furthermore, there is no effect on the forced response of the system. These important points are reinforced in the following example.

EXAMPLE 3.6.7

Determine the unit step response of the system described by the difference equation

$$y(n) = 0.9y(n-1) - 0.81y(n-2) + x(n)$$

under the following initial conditions $y(-1) = y(-2) = 1$.

Solution. The system function is

$$H(z) = \frac{1}{1 - 0.9z^{-1} + 0.81z^{-2}}$$

This system has two complex-conjugate poles at

$$p_1 = 0.9e^{j\pi/3}, \qquad p_2 = 0.9e^{-j\pi/3}$$

The z-transform of the unit step sequence is

$$X(z) = \frac{1}{1 - z^{-1}}$$

Therefore,

$$\begin{aligned} Y_{zs}(z) &= \frac{1}{(1 - 0.9e^{j\pi/3}z^{-1})(1 - 0.9e^{-j\pi/3}z^{-1})(1 - z^{-1})} \\ &= \frac{0.0496 - j0.542}{1 - 0.9e^{j\pi/3}z^{-1}} + \frac{0.0496 + j0.542}{1 - 0.9e^{-j\pi/3}z^{-1}} + \frac{1.099}{1 - z^{-1}} \end{aligned}$$

and hence the zero-state response is

$$y_{zs}(n) = \left[1.099 + 1.088(0.9)^n \cos\left(\frac{\pi}{3}n - 5.2^\circ\right)\right] u(n)$$

For the initial conditions $y(-1) = y(-2) = 1$, the additional component in the z-transform is

$$\begin{aligned} Y_{zi}(z) &= \frac{N_0(z)}{A(z)} = \frac{0.09 - 0.81z^{-1}}{1 - 0.9z^{-1} + 0.81z^{-2}} \\ &= \frac{0.045 + j0.4936}{1 - 0.9e^{j\pi/3}z^{-1}} + \frac{0.045 - j0.4936}{1 - 0.9e^{-j\pi/3}z^{-1}} \end{aligned}$$

Consequently, the zero-input response is

$$y_{zi}(n) = 0.988(0.9)^n \cos\left(\frac{\pi}{3}n + 87^\circ\right) u(n)$$

In this case the total response has the z-transform

$$\begin{aligned} Y(z) &= Y_{zs}(z) + Y_{zi}(z) \\ &= \frac{1.099}{1 - z^{-1}} + \frac{0.568 + j0.445}{1 - 0.9e^{j\pi/3}z^{-1}} + \frac{0.568 - j0.445}{1 - 0.9e^{-j\pi/3}z^{-1}} \end{aligned}$$

The inverse transform yields the total response in the form

$$y(n) = 1.099u(n) + 1.44(0.9)^n \cos\left(\frac{\pi}{3}n + 38^\circ\right) u(n)$$

3.7 Summary and References

The z-transform plays the same role in discrete-time signals and systems as the Laplace transform does in continuous-time signals and systems. In this chapter we derived the important properties of the z-transform, which are extremely useful in the analysis of discrete-time systems. Of particular importance is the convolution property, which transforms the convolution of two sequences into a product of their z-transforms.

In the context of LTI systems, the convolution property results in the product of the z-transform $X(z)$ of the input signal with the system function $H(z)$, where the latter is the z-transform of the unit sample response of the system. This relationship allows us to determine the output of an LTI system in response to an input with transform $X(z)$ by computing the product $Y(z) = H(z)X(z)$ and then determining the inverse z-transform of $Y(z)$ to obtain the output sequence $y(n)$.

We observed that many signals of practical interest have rational z-transforms. Moreover, LTI systems characterized by constant-coefficient linear difference equations also possess rational system functions. Consequently, in determining the inverse z-transform, we naturally emphasized the inversion of rational transforms. For such transforms, the partial-fraction expansion method is relatively easy to apply, in conjunction with the ROC, to determine the corresponding sequence in the time domain.

We considered the characterization of LTI systems in the z-transform domain. In particular, we related the pole–zero locations of a system to its time-domain characteristics and restated the requirements for stability and causality of LTI systems in terms of the pole locations. We demonstrated that a causal system has a system function $H(z)$ with a ROC $|z| > r_1$, where $0 < r_1 \leq \infty$. In a stable and causal system, the poles of $H(z)$ lie inside the unit circle. On the other hand, if the system is noncausal, the condition for stability requires that the unit circle be contained in the ROC of $H(z)$. Hence a noncausal stable LTI system has a system function with poles both inside and outside the unit circle with an annular ROC that includes the unit circle. Finally, the one-sided z-transform was introduced to solve for the response of causal systems excited by causal input signals with nonzero initial conditions.

Problems

3.1 Determine the z-transform of the following signals.

(a) $x(n) = \{3, 0, 0, 0, 0, \underset{\uparrow}{6}, 1, -4\}$

(b) $x(n) = \begin{cases} (\frac{1}{2})^n, & n \geq 5 \\ 0, & n \leq 4 \end{cases}$

3.2 Determine the z-transforms of the following signals and sketch the corresponding pole–zero patterns.

(a) $x(n) = (1 + n)u(n)$

(b) $x(n) = (a^n + a^{-n})u(n)$, a real

(c) $x(n) = (-1)^n 2^{-n} u(n)$

(d) $x(n) = (na^n \sin \omega_0 n)u(n)$

(e) $x(n) = (na^n \cos \omega_0 n)u(n)$

(f) $x(n) = Ar^n \cos(\omega_0 n + \phi)u(n), 0 < r < 1$

(g) $x(n) = \frac{1}{2}(n^2 + n)(\frac{1}{3})^{n-1}u(n-1)$

(h) $x(n) = (\frac{1}{2})^n[u(n) - u(n-10)]$

3.3 Determine the z-transforms and sketch the ROC of the following signals.

(a) $x_1(n) = \begin{cases} (\frac{1}{3})^n, & n \geq 0 \\ (\frac{1}{2})^{-n}, & n < 0 \end{cases}$

(b) $x_2(n) = \begin{cases} (\frac{1}{3})^n - 2^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$

(c) $x_3(n) = x_1(n+4)$

(d) $x_4(n) = x_1(-n)$

3.4 Determine the z-transform of the following signals.

(a) $x(n) = n(-1)^n u(n)$

(b) $x(n) = n^2 u(n)$

(c) $x(n) = -na^n u(-n-1)$

(d) $x(n) = (-1)^n \left(\cos \frac{\pi}{3} n\right) u(n)$

(f) $x(n) = \{\underset{\uparrow}{1}, 0, -1, 0, 1, -1, \ldots\}$

3.5 Determine the regions of convergence of right-sided, left-sided, and finite-duration two-sided sequences.

3.6 Express the z-transform of

$$y(n) = \sum_{k=-\infty}^{n} x(k)$$

in terms of $X(z)$. [*Hint*: Find the difference $y(n) - y(n-1)$.]

3.7 Compute the convolution of the following signals by means of the z-transform.

$$x_1(n) = \begin{cases} (\frac{1}{3})^n, & n \geq 0 \\ (\frac{1}{2})^{-n}, & n < 0 \end{cases}$$

$$x_2(n) = (\frac{1}{2})^n u(n)$$

3.8 Use the convolution property to:

(a) Express the z-transform of

$$y(n) = \sum_{k=-\infty}^{n} x(k)$$

in terms of $X(z)$.

(b) Determine the z-transform of $x(n) = (n+1)u(n)$. [*Hint*: Show first that $x(n) = u(n) * u(n)$.]

3.9 The z-transform $X(z)$ of a real signal $x(n)$ includes a pair of complex-conjugate zeros and a pair of complex-conjugate poles. What happens to these pairs if we multiply $x(n)$ by $e^{j\omega_0 n}$? (*Hint*: Use the scaling theorem in the z-domain.)

3.10 Apply the final value theorem to determine $x(\infty)$ for the signal

$$x(n) = \begin{cases} 1, & \text{if } n \text{ is even} \\ 0, & \text{otherwise} \end{cases}$$

3.11 Using long division, determine the inverse z-transform of

$$X(z) = \frac{1 + 2z^{-1}}{1 - 2z^{-1} + z^{-2}}$$

if **(a)** $x(n)$ is causal and **(b)** $x(n)$ is anticausal.

3.12 Determine the causal signal $x(n)$ having the z-transform

$$X(z) = \frac{1}{(1 - 2z^{-1})(1 - z^{-1})2}$$

3.13 Let $x(n)$ be a sequence with z-transform $X(z)$. Determine, in terms of $X(z)$, the z-transforms of the following signals.

(a) $x_1(n) = \begin{cases} x\left(\frac{n}{2}\right), & \text{if } n \text{ even} \\ 0, & \text{if } n \text{ odd} \end{cases}$

(b) $x_2(n) = x(2n)$

3.14 Determine the causal signal $x(n)$ if its z-transform $X(z)$ is given by:

(a) $X(z) = \frac{1+3z^{-1}}{1+3z^{-1}+2z^{-2}}$

(b) $X(z) = \frac{1}{1-z^{-1}+\frac{1}{2}z^{-2}}$

(c) $X(z) = \frac{z^{-6}+z^{-7}}{1-z^{-1}}$

(d) $X(z) = \frac{1+2z^{-2}}{1+z^{-2}}$

(e) $X(z) = \frac{1}{4}\frac{1+6z^{-1}+z^{-2}}{(1-2z^{-1}+2z^{-2})(1-0.5z^{-1})}$

(f) $X(z) = \frac{2-1.5z^{-1}}{1-1.5z^{-1}+0.5z^{-2}}$

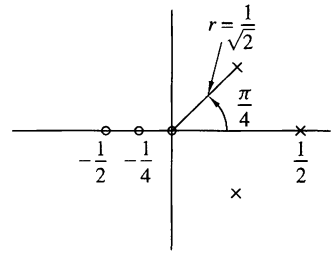


Figure P3.14

(g) $X(z) = \frac{1+2z^{-1}+z^{-2}}{1+4z^{-1}+4z^{-2}}$

(h) $X(z)$ is specified by a pole–zero pattern in Fig. P3.14. The constant $G = \frac{1}{4}$.

(i) $X(z) = \frac{1-\frac{1}{2}z^{-1}}{1+\frac{1}{2}z^{-1}}$

(j) $X(z) = \frac{1-az^{-1}}{z^{-1}-a}$

3.15 Determine all possible signals $x(n)$ associated with the z-transform

$$X(z) = \frac{5z^{-1}}{(1-2z^{-1})(3-z^{-1})}$$

3.16 Determine the convolution of the following pairs of signals by means of the z-transform.

(a) $x_1(n) = (\frac{1}{4})^n u(n-1), \quad x_2(n) = [1 + (\frac{1}{2})^n]u(n)$

(b) $x_1(n) = u(n), \quad x_2(n) = \delta(n) + (\frac{1}{2})^n u(n)$

(c) $x_1(n) = (\frac{1}{2})^n u(n), \quad x_2(n) = \cos \pi n u(n)$

(d) $x_1(n) = nu(n), \quad x_2(n) = 2^n u(n-1)$

3.17 Prove the final value theorem for the one-sided z-transform.

3.18 If $X(z)$ is the z-transform of $x(n)$, show that:

(a) $Z\{x^*(n)\} = X^*(z^*)$

(b) $Z\{\text{Re}[x(n)]\} = \frac{1}{2}[X(z) + X^*(z^*)]$

(c) $Z\{\text{Im}[x(n)]\} = \frac{1}{2}[X(z) - X^*(z^*)]$

(d) If

$$x_k(n) = \begin{cases} x\left(\frac{n}{k}\right), & \text{if } n/k \text{ integer} \\ 0, & \text{otherwise} \end{cases}$$

then

$$X_k(z) = X(z^k)$$

(e) $Z\{e^{j\omega_0 n}x(n)\} = X(ze^{-j\omega_0})$

3.19 By first differentiating $X(z)$ and then using appropriate properties of the z-transform, determine $x(n)$ for the following transforms.

(a) $X(z) = \log(1-2z), \quad |z| < \frac{1}{2}$

(b) $X(z) = \log(1-z^{-1}), \quad |z| > \frac{1}{2}$

3.20

(a) Draw the pole–zero pattern for the signal

$$x_1(n) = (r^n \sin \omega_0 n)u(n), \qquad 0 < r < 1$$

(b) Compute the z-transform $X_2(z)$, which corresponds to the pole–zero pattern in part (a).

(c) Compare $X_1(z)$ with $X_2(z)$. Are they indentical? If not, indicate a method to derive $X_1(z)$ from the pole–zero pattern.

3.21 Show that the roots of a polynomial with real coefficients are real or form complex-conjugate pairs. The inverse is not true, in general.

3.22 Prove the convolution and correlation properties of the z-transform using only its definition.

3.23 Determine the signal $x(n)$ with z-transform

$$X(z) = e^{z} + e^{1/z}, \qquad |z| \neq 0$$

3.24 Determine, in closed form, the causal signals $x(n)$ whose z-transforms are given by:

(a) $X(z) = \dfrac{1}{1+1.5z^{-1}-0.5z^{-2}}$

(b) $X(z) = \dfrac{1}{1-0.5z^{-1}+0.6z^{-2}}$

Partially check your results by computing $x(0), x(1), x(2)$, and $x(\infty)$ by an alternative method.

3.25 Determine all possible signals that can have the following z-transforms.

(a) $X(z) = \dfrac{1}{1-1.5z^{-1}+0.5z^{-2}}$

(b) $X(z) = \dfrac{1}{1-\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}}$

3.26 Determine the signal $x(n)$ with z-transform

$$X(z) = \frac{3}{1-\frac{10}{3}z^{-1}+z^{-2}}$$

if $X(z)$ converges on the unit circle.

3.27 Prove the complex convolution relation given by (3.2.22).

3.28 Prove the conjugation properties and Parseval's relation for the z-transform given in Table 3.2.

3.29 In Example 3.4.1 we solved for $x(n)$, $n < 0$, by performing contour integrations for each value of n. In general, this procedure proves to be tedious. It can be avoided by making a transformation in the contour integral from z-plane to the $w = 1/z$ plane. Thus a circle of radius R in the z-plane is mapped into a circle of radius $1/R$ in the w-plane. As a consequence, a pole inside the unit circle in the z-plane is mapped into a pole outside the unit circle in the w-plane. By making the change of variable $w = 1/z$ in the contour integral, determine the sequence $x(n)$ for $n < 0$ in Example 3.4.1.

3.30 Let $x(n)$, $0 \leq n \leq N-1$ be a finite-duration sequence, which is also real-valued and even. Show that the zeros of the polynomial $X(z)$ occur in mirror-image pairs about the unit circle. That is, if $z = re^{j\theta}$ is a zero of $X(z)$, then $z = (1/r)e^{j\theta}$ is also a zero.

3.31 Prove that the Fibonacci sequence can be thought of as the impulse response of the system described by the difference equation $y(n) = y(n-1) + y(n-2) + x(n)$. Then determine $h(n)$ using z-transform techniques.

3.32 Show that the following systems are equivalent.

(a) $y(n) = 0.2y(n-1) + x(n) - 0.3x(n-1) + 0.02x(n-2)$

(b) $y(n) = x(n) - 0.1x(n-1)$

3.33 Consider the sequence $x(n) = a^n u(n)$, $-1 < a < 1$. Determine at least two sequences that are not equal to $x(n)$ but have the same autocorrelation.

3.34 Compute the unit step response of the system with impulse response

$$h(n) = \begin{cases} 3^n, & n < 0 \\ (\frac{2}{5})^n, & n \geq 0 \end{cases}$$

3.35 Compute the zero-state response for the following pairs of systems and input signals.

(a) $h(n) = (\frac{1}{3})^n u(n)$, $x(n) = (\frac{1}{2})^n (\cos \frac{\pi}{3} n) u(n)$

(b) $h(n) = (\frac{1}{2})^n u(n)$, $x(n) = (\frac{1}{3})^n u(n) + (\frac{1}{2})^{-n} u(-n-1)$

(c) $y(n) = -0.1y(n-1) + 0.2y(n-2) + x(n) + x(n-1) x(n) = (\frac{1}{3})^n u(n)$

(d) $y(n) = \frac{1}{2} x(n) - \frac{1}{2} x(n-1) x(n) = 10(\cos \frac{\pi}{2} n) u(n)$

(e) $y(n) = -y(n-2) + 10x(n) x(n) = 10(\cos \frac{\pi}{2} n) u(n)$

(f) $h(n) = (\frac{2}{5})^n u(n)$, $x(n) = u(n) - u(n-7)$

(g) $h(n) = (\frac{1}{2})^n u(n)$, $x(n) = (-1)^n$, $-\infty < n < \infty$

(h) $h(n) = (\frac{1}{2})^n u(n)$, $x(n) = (n+1)(\frac{1}{4})^n u(n)$

3.36 Consider the system

$$H(z) = \frac{1 - 2z^{-1} + 2z^{-2} - z^{-3}}{(1 - z^{-1})(1 - 0.5z^{-1})(1 - 0.2z^{-1})}, \qquad \text{ROC: } 0.5|z| > 1$$

(a) Sketch the pole–zero pattern. Is the system stable?

(b) Determine the impulse response of the system.

3.37 Compute the response of the system

$$y(n) = 0.7y(n-1) - 0.12y(n-2) + x(n-1) + x(n-2)$$

to the input $x(n) = nu(n)$. Is the system stable?

3.38 Determine the impulse response and the step response of the following causal systems. Plot the pole–zero patterns and determine which of the systems are stable.

(a) $y(n) = \frac{3}{4} y(n-1) - \frac{1}{8} y(n-2) + x(n)$

(b) $y(n) = y(n-1) - 0.5y(n-2) + x(n) + x(n-1)$

(c) $H(z) = \frac{z^{-1}(1 + z^{-1})}{(1 - z^{-1})^3}$

(d) $y(n) = 0.6y(n-1) - 0.08y(n-2) + x(n)$

(e) $y(n) = 0.7y(n-1) - 0.1y(n-2) + 2x(n) - x(n-2)$

3.39 Let $x(n)$ be a causal sequence with z-transform $X(z)$ whose pole–zero plot is shown in Fig. P3.39. Sketch the pole–zero plots and the ROC of the following sequence:

(a) $x_1(n) = x(-n+2)$

(b) $x_2(n) = e^{j(\pi/3)n} x(n)$

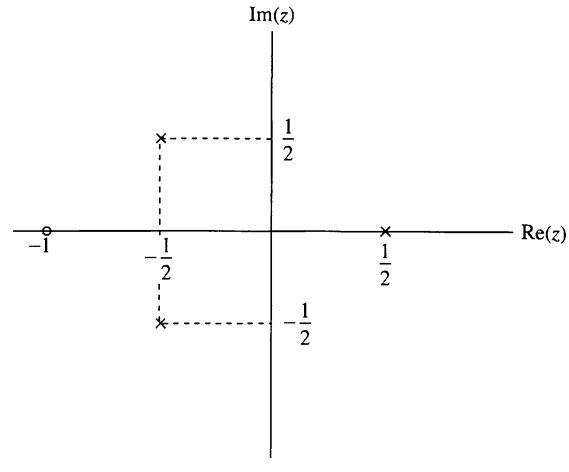


Figure P3.39

3.40 We want to design a causal discrete-time LTI system with the property that if the input is

$$x(n) = (\frac{1}{2})^n u(n) - \frac{1}{4}(\frac{1}{2})^{n-1} u(n-1)$$

then the output is

$$y(n) = (\frac{1}{3})^n u(n)$$

(a) Determine the impulse response $h(n)$ and the system function $H(z)$ of a system that satisfies the foregoing conditions.

(b) Find the difference equation that characterizes this system.

(c) Determine a realization of the system that requires the minimum possible amount of memory.

(d) Determine if the system is stable.

3.41 Determine the stability region for the causal system

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

by computing its poles and restricting them to be inside the unit circle.

3.42 Consider the system

$$H(z) = \frac{z^{-1} + \frac{1}{2} z^{-2}}{1 - \frac{3}{5} z^{-1} + \frac{2}{25} z^{-2}}$$

Determine:

(a) The impulse response

(b) The zero-state step response

(c) The step response if $y(-1) = 1$ and $y(-2) = 2$

Determine the system function, impulse response, and zero-state step response of the system shown in Fig P3.43.

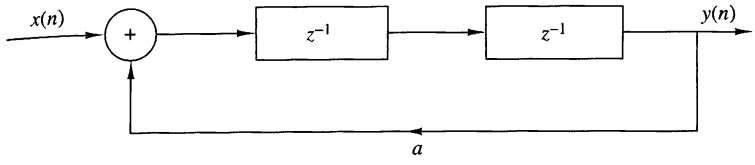


Figure P3.43

Consider the causal system

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$$

Determine:

(a) The impulse response

(b) The zero-state step response

(c) The step response if $y(-1) = A \neq 0$

(d) The response to the input

$$x(n) = \cos \omega_0 n, \qquad 0 \leq n < \infty$$

Determine the zero-state response of the system

$$y(n) = \frac{1}{2} y(n-1) + 4x(n) + 3x(n-1)$$

to the input

$$x(n) = e^{j\omega_0 n} u(n)$$

What is the steady-state response of the system?

Consider the causal system defined by the pole–zero pattern shown in Fig. P3.46.

(a) Determine the system function and the impulse response of the system given that $H(z)|_{z=1} = 1$.

(b) Is the system stable?

(c) Sketch a possible implementation of the system and determine the corresponding difference equations.

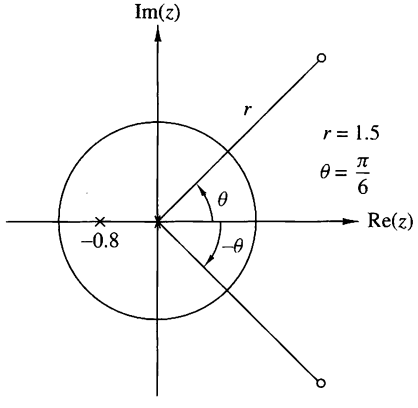


Figure P3.46

3.47 Compute the convolution of the following pair of signals in the time domain and by using the one-sided z-transform.

(a) $x_1(n) = \{1, 1, \underset{\uparrow}{1}, 1, 1\}$, $x_2(n) = \{\underset{\uparrow}{1}, 1, 1\}$

(b) $x_1(n) = (\frac{1}{2})^n u(n)$, $x_2(n) = (\frac{1}{3})^n u(n)$

(c) $x_1(n) = \{1, \underset{\uparrow}{2}, 3, 4\}$, $x_2(n) = \{4, 3, \underset{\uparrow}{2}, 1\}$

(d) $x_1(n) = \{\underset{\uparrow}{1}, 1, 1, 1, 1\}$, $x_2(n) = \{\underset{\uparrow}{1}, 1, 1\}$

Did you obtain the same results by both methods? Explain.

3.48 Determine the one-sided z-transform of the constant signal $x(n) = 1, -\infty < n < \infty$.

3.49 Use the one-sided z-transform to determine $y(n)$, $n \geq 0$ in the following cases.

(a) $y(n) + \frac{1}{2}y(n-1) - \frac{1}{4}y(n-2) = 0$; $y(-1) = y(-2) = 1$

(b) $y(n) - 1.5y(n-1) + 0.5y(n-2) = 0$; $y(-1) = 1, y(-2) = 0$

(c) $y(n) = \frac{1}{2}y(n-1) + x(n)x(n) = (\frac{1}{3})^n u(n)$, $y(-1) = 1$

(d) $y(n) = \frac{1}{4}y(n-2) + x(n)x(n) = u(n)y(-1) = 0$; $y(-2) = 1$

3.50 An FIR LTI system has an impulse response $h(n)$, which is real valued, even, and has finite duration of $2N + 1$. Show that if $z_1 = re^{j\omega_0}$ is a zero of the system, then $z_1 = (1/r)e^{j\omega_0}$ is also a zero.

3.51 Consider an LTI discrete-time system whose pole–zero pattern is shown in Fig. P3.51.

(a) Determine the ROC of the system function $H(z)$ if the system is known to be stable.

(b) It is possible for the given pole–zero plot to correspond to a causal and stable system? If so, what is the appropriate ROC?

(c) How many possible systems can be associated with this pole–zero pattern?

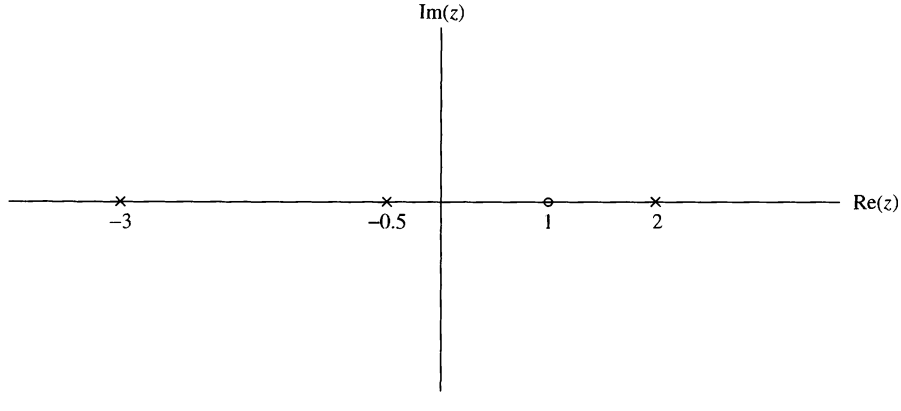


Figure P3.51

3.52 Let $x(n)$ be a causal sequence.

(a) What conclusion can you draw about the value of its z-transform $X(z)$ at $z = \infty$?

(b) Use the result in part (a) to check which of the following transforms cannot be associated with a causal sequence.

$$\text{(i)}X(z) = \frac{(z-\frac{1}{2})^4}{(z-\frac{1}{3})^3} \qquad \text{(ii)}X(z) = \frac{(1-\frac{1}{2}z^{-1})^2}{(1-\frac{1}{3}z^{-1})} \qquad \text{(iii)}X(z) = \frac{(z-\frac{1}{3})^2}{(z-\frac{1}{2})^3}$$

3.53 A causal pole–zero system is BIBO stable if its poles are inside the unit circle. Consider now a pole–zero system that is BIBO stable and has its poles inside the unit circle. Is the system always causal? [*Hint:* Consider the systems $h_1(n) = a^n u(n)$ and $h_2(n) = a^n u(n+3)$, $|a| < 1$.]

3.54 Let $x(n)$ be an anticausal signal [i.e., $x(n) = 0$ for $n > 0$]. Formulate and prove an initial value theorem for anticausal signals.

3.55 The step response of an LTI system is

$$s(n) = (\frac{1}{3})^{n-2}u(n+2)$$

(a) Find the system function $H(z)$ and sketch the pole–zero plot.

(b) Determine the impulse response $h(n)$.

(c) Check if the system is causal and stable.

3.56 Use contour integration to determine the sequence $x(n)$ whose z-transform is given by

(a) $X(z) = \frac{1}{1-\frac{1}{2}z^{-1}}, \qquad |z| > \frac{1}{2}$

(b) $X(z) = \frac{1}{1-\frac{1}{2}z^{-1}}, \qquad |z| < \frac{1}{2}$

(c) $X(z) = \frac{z-a}{1-az}, \qquad |z| > |1/a|$

(d) $X(z) = \frac{1-\frac{1}{4}z^{-1}}{1-\frac{1}{6}z^{-1}-\frac{1}{6}z^{-2}}, \qquad |z| > \frac{1}{2}$

3.57 Let $x(n)$ be a sequence with z-transform

$$X(z) = \frac{1-a^2}{(1-az)(1-az^{-1})}, \qquad \text{ROC: } a > |z| > 1/a$$

with $0 < a < 1$. Determine $x(n)$ by using contour integration.

3.58 The z-transform of a sequence $x(n)$ is given by

$$X(z) = \frac{z^{20}}{(z-\frac{1}{2})(z-2)^5(z+\frac{5}{2})^2(z+3)}$$

Furthermore it is known that $X(z)$ converges for $|z| = 1$.

(a) Determine the ROC of $X(z)$.

(b) Determine $x(n)$ at $n = -18$. (*Hint:* Use contour integration.)

CHAPTER

4

Frequency Analysis of Signals

The Fourier transform is one of several mathematical tools that is useful in the analysis and design of LTI systems. Another is the Fourier series. These signal representations basically involve the decomposition of the signals in terms of sinusoidal (or complex exponential) components. With such a decomposition, a signal is said to be represented in the *frequency domain.*

As we shall demonstrate, most signals of practical interest can be decomposed into a sum of sinusoidal signal components. For the class of periodic signals, such a decomposition is called a *Fourier series.* For the class of finite energy signals, the decomposition is called the *Fourier transform.* These decompositions are extremely important in the analysis of LTI systems because the response of an LTI system to a sinusoidal input signal is a sinusoid of the same frequency but of different amplitude and phase. Furthermore, the linearity property of the LTI system implies that a linear sum of sinusoidal components at the input produces a similar linear sum of sinusoidal components at the output, which differ only in the amplitudes and phases from the input sinusoids. This characteristic behavior of LTI systems renders the sinusoidal decomposition of signals very important. Although many other decompositions of signals are possible, only the class of sinusoidal (or complex exponential) signals possess this desirable property in passing through an LTI system.

We begin our study of frequency analysis of signals with the representation of continuous-time periodic and aperiodic signals by means of the Fourier series and the Fourier transform, respectively. This is followed by a parallel treatment of discrete-time periodic and aperiodic signals. The properties of the Fourier transform are described in detail and a number of time-frequency dualities are presented.

4.1 Frequency Analysis of Continuous-Time Signals

It is well known that a prism can be used to break up white light (sunlight) into the colors of the rainbow (see Fig. 4.1.1(a)). In a paper submitted in 1672 to the Royal Society, Isaac Newton used the term *spectrum* to describe the *continuous* bands of colors produced by this apparatus. To understand this phenomenon, Newton placed another prism upside-down with respect to the first, and showed that the colors blended back into white light, as in Fig. 4.1.1(b). By inserting a slit between the two prisms and blocking one or more colors from hitting the second prism, he showed that the remixed light is no longer white. Hence the light passing through the first prism is simply analyzed into its component colors without any other change. However, only if we mix again all of these colors do we obtain the original white light.

Later, Joseph Fraunhofer (1787–1826), in making measurements of light emitted by the sun and stars, discovered that the spectrum of the observed light consists of distinct color lines. A few years later (mid-1800s) Gustav Kirchhoff and Robert Bunsen found that each chemical element, when heated to incandescence, radiated its own distinct color of light. As a consequence, each chemical element can be identified by its own *line spectrum.*

From physics we know that each color corresponds to a specific frequency of the visible spectrum. Hence the analysis of light into colors is actually a form of *frequency analysis.*

Frequency analysis of a signal involves the resolution of the signal into its frequency (sinusoidal) components. Instead of light, our signal waveforms are basically functions of time. The role of the prism is played by the Fourier analysis tools that we will develop: the Fourier series and the Fourier transform. The recombination of the sinusoidal components to reconstruct the original signal is basically a Fourier synthesis problem. The problem of signal analysis is basically the same for the case of a signal waveform and for the case of the light from heated chemical composi-

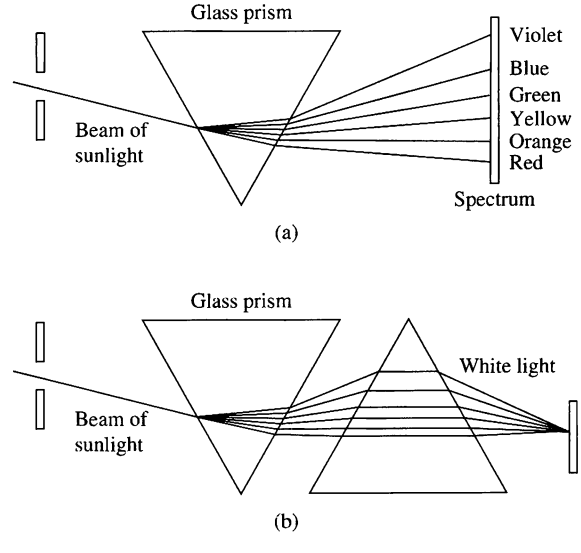


Figure 4.1.1
(a) Analysis and (b) synthesis of the white light (sunlight) using glass prisms.

tions. Just as in the case of chemical compositions, different signal waveforms have different spectra. Thus the spectrum provides an "identity" or a signature for the signal in the sense that no other signal has the same spectrum. As we will see, this attribute is related to the mathematical treatment of frequency-domain techniques.

If we decompose a waveform into sinusoidal components, in much the same way that a prism separates white light into different colors, the sum of these sinusoidal components results in the original waveform. On the other hand, if any of these components is missing, the result is a different signal.

In our treatment of frequency analysis, we will develop the proper mathematical tools ("prisms") for the decomposition of signals ("light") into sinusoidal frequency components (colors). Furthermore, the tools ("inverse prisms") for synthesis of a given signal from its frequency components will also be developed.

The basic motivation for developing the frequency analysis tools is to provide a mathematical and pictorial representation for the frequency components that are contained in any given signal. As in physics, the term *spectrum* is used when referring to the frequency content of a signal. The process of obtaining the spectrum of a given signal using the basic mathematical tools described in this chapter is known as *frequency* or *spectral analysis*. In contrast, the process of determining the spectrum of a signal in practice, based on actual measurements of the signal, is called *spectrum estimation*. This distinction is very important. In a practical problem the signal to be analyzed does not lend itself to an exact mathematical description. The signal is usually some information-bearing signal from which we are attempting to extract the relevant information. If the information that we wish to extract can be obtained either directly or indirectly from the spectral content of the signal, we can perform *spectrum estimation*on the information-bearing signal, and thus obtain an estimate of the signal spectrum. In fact, we can view spectral estimation as a type of spectral analysis performed on signals obtained from physical sources (e.g., speech, EEG, ECG, etc.). The instruments or software programs used to obtain spectral estimates of such signals are known as *spectrum analyzers*.

Here, we will deal with spectral analysis. However, in Chapter 14 we shall treat the subject of power spectrum estimation.

4.1.1 The Fourier Series for Continuous-Time Periodic Signals

In this section we present the frequency analysis tools for continuous-time periodic signals. Examples of periodic signals encountered in practice are square waves, rectangular waves, triangular waves, and of course, sinusoids and complex exponentials.

The basic mathematical representation of periodic signals is the Fourier series, which is a linear weighted sum of harmonically related sinusoids or complex exponentials. Jean Baptiste Joseph Fourier (1768–1830), a French mathematician, used such trigonometric series expansions in describing the phenomenon of heat conduction and temperature distribution through bodies. Although his work was motivated by the problem of heat conduction, the mathematical techniques that he developed during the early part of the nineteenth century now find application in a variety of problems encompassing many different fields, including optics, vibrations in mechanical systems, system theory, and electromagnetics.

From Chapter 1 we recall that a linear combination of harmonically related complex exponentials of the form

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t} \tag{4.1.1}$$

is a periodic signal with fundamental period $T_p = 1/F_0$. Hence we can think of the exponential signals

$$\{e^{j2\pi k F_0 t}, \qquad k = 0, \pm 1, \pm 2, \ldots\}$$

as the basic "building blocks" from which we can construct periodic signals of various types by proper choice of the fundamental frequency and the coefficients $\{c_k\}$. F_0 determines the fundamental period of $x(t)$ and the coefficients $\{c_k\}$ specify the shape of the waveform.

Suppose that we are given a periodic signal $x(t)$ with period T_p. We can represent the periodic signal by the series (4.1.1), called a *Fourier series,* where the fundamental frequency F_0 is selected to be the reciprocal of the given period T_p. To determine the expression for the coefficients $\{c_k\}$, we first multiply both sides of (4.1.1) by the complex exponential

$$e^{-j2\pi F_0 l t}$$

where l is an integer and then integrate both sides of the resulting equation over a single period, say from 0 to T_p, or more generally, from t_0 to $t_0 + T_p$, where t_0 is an arbitrary but mathematically convenient starting value. Thus we obtain

$$\int_{t_0}^{t_0+T_p} x(t) e^{-j2\pi l F_0 t}\, dt = \int_{t_0}^{t_0+T_p} e^{-j2\pi l F_0 t} \left(\sum_{k=-\infty}^{\infty} c_k e^{+j2\pi k F_0 t} \right) dt \tag{4.1.2}$$

To evaluate the integral on the right-hand side of (4.1.2), we interchange the order of the summation and integration and combine the two exponentials. Hence

$$\sum_{k=-\infty}^{\infty} c_k \int_{t_0}^{t_0+T_p} e^{j2\pi F_0 (k-l) t}\, dt = \sum_{k=-\infty}^{\infty} c_k \left[\frac{e^{j2\pi F_0 (k-l) t}}{j2\pi F_0 (k-l)} \right]_{t_0}^{t_0+T_p} \tag{4.1.3}$$

For $k \neq l$, the right-hand side of (4.1.3) evaluated at the lower and upper limits, t_0 and $t_0 + T_p$, respectively, yields zero. On the other hand, if $k = l$, we have

$$\int_{t_0}^{t_0+T_p} dt = t \Big|_{t_0}^{t_0+T_p} = T_p$$

Consequently, (4.1.2) reduces to

$$\int_{t_0}^{t_0+T_p} x(t) e^{-j2\pi l F_0 t}\, dt = c_l T_p$$

and therefore the expression for the Fourier coefficients in terms of the given periodic signal becomes

$$c_l = \frac{1}{T_p} \int_{t_0}^{t_0+T_p} x(t) e^{-j2\pi l F_0 t}\, dt$$

Since t_0 is arbitrary, this integral can be evaluated over any interval of length T_p, that is, over any interval equal to the period of the signal $x(t)$. Consequently, the integral for the Fourier series coefficients will be written as

$$c_l = \frac{1}{T_p} \int_{T_p} x(t) e^{-j2\pi l F_0 t}\, dt \tag{4.1.4}$$

An important issue that arises in the representation of the periodic signal $x(t)$ by the Fourier series is whether or not the series converges to $x(t)$ for every value of t, that is, whether the signal $x(t)$ and its Fourier series representation

$$\sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t} \tag{4.1.5}$$

are equal at every value of t. The so-called *Dirichlet conditions* guarantee that the series (4.1.5) will be equal to $x(t)$, except at the values of t for which $x(t)$ is discontinuous. At these values of t, (4.1.5) converges to the midpoint (average value) of the discontinuity. The Dirichlet conditions are:

1. The signal $x(t)$ has a finite number of discontinuities in any period.
2. The signal $x(t)$ contains a finite number of maxima and minima during any period.
3. The signal $x(t)$ is absolutely integrable in any period, that is,

$$\int_{T_p} |x(t)|\, dt < \infty \tag{4.1.6}$$

All periodic signals of practical interest satisfy these conditions.

The weaker condition, that the signal has finite energy in one period,

$$\int_{T_p} |x(t)|^2\, dt < \infty \tag{4.1.7}$$

guarantees that the energy in the difference signal

$$e(t) = x(t) - \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t}$$

is zero, although $x(t)$ and its Fourier series may not be equal for all values of t. Note that (4.1.6) implies (4.1.7), but not vice versa. Also, both (4.1.7) and the Dirichlet

conditions are sufficient but not necessary conditions (i.e., there are signals that have a Fourier series representation but do not satisfy these conditions).

In summary, if $x(t)$ is periodic and satisfies the Dirichlet conditions, it can be represented in a Fourier series as in (4.1.1), where the coefficients are specified by (4.1.4). These relations are summarized below.

Frequency Analysis of Continuous-Time Periodic Signals

Synthesis equation	$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t}$	(4.1.8)
Analysis equation	$c_k = \frac{1}{T_p} \int_{T_p} x(t) e^{-j2\pi k F_0 t} dt$	(4.1.9)

In general, the Fourier coefficients c_k are complex valued. Moreover, it is easily shown that if the periodic signal is real, c_k and c_{-k} are complex conjugates. As a result, if

$$c_k = |c_k| e^{j\theta_k}$$

then

$$c_{-k} = |c_k|^{-j\theta_k}$$

Consequently, the Fourier series may also be represented in the form

$$x(t) = c_0 + 2 \sum_{k=1}^{\infty} |c_k| \cos(2\pi k F_0 t + \theta_k) \tag{4.1.10}$$

where c_0 is real valued when $x(t)$ is real.

Finally, we should indicate that yet another form for the Fourier series can be obtained by expanding the cosine function in (4.1.10) as

$$\cos(2\pi k F_0 t + \theta_k) = \cos 2\pi k F_0 t \cos\theta_k - \sin 2\pi k F_0 t \sin\theta_k$$

Consequently, we can rewrite (4.1.10) in the form

$$x(t) = a_0 + \sum_{k=1}^{\infty} (a_k \cos 2\pi k F_0 t - b_k \sin 2\pi k F_0 t) \tag{4.1.11}$$

where

$$a_0 = c_0$$

$$a_k = 2|c_k| \cos\theta_k$$

$$b_k = 2|c_k| \sin\theta_k$$

The expressions in (4.1.8), (4.1.10), and (4.1.11) constitute three equivalent forms for the Fourier series representation of a real periodic signal.

4.1.2 Power Density Spectrum of Periodic Signals

A periodic signal has infinite energy and a finite average power, which is given as

$$P_x = \frac{1}{T_p} \int_{T_p} |x(t)|^2 \, dt \tag{4.1.12}$$

If we take the complex conjugate of (4.1.8) and substitute for $x^*(t)$ in (4.1.12), we obtain

$$\begin{aligned} P_x &= \frac{1}{T_p} \int_{T_p} x(t) \sum_{k=-\infty}^{\infty} c_k^* e^{-j2\pi k F_0 t} \, dt \\ &= \sum_{k=-\infty}^{\infty} c_k^* \left[\frac{1}{T_p} \int_{T_p} x(t) e^{-j2\pi k F_0 t} \, dt \right] \\ &= \sum_{k=-\infty}^{\infty} |c_k|^2 \end{aligned} \tag{4.1.13}$$

Therefore, we have established the relation

$$P_x = \frac{1}{T_p} \int_{T_p} |x(t)|^2 \, dt = \sum_{k=-\infty}^{\infty} |c_k|^2 \tag{4.1.14}$$

which is called *Parseval's relation*for power signals.

To illustrate the physical meaning of (4.1.14), suppose that $x(t)$ consists of a single complex exponential

$$x(t) = c_k e^{j2\pi k F_0 t}$$

In this case, all the Fourier series coefficients except c_k are zero. Consequently, the average power in the signal is

$$P_x = |c_k|^2$$

It is obvious that $|c_k|^2$ represents the power in the kth harmonic component of the signal. Hence the total average power in the periodic signal is simply the sum of the average powers in all the harmonics.

If we plot the $|c_k|^2$ as a function of the frequencies kF_0, $k = 0, \pm 1, \pm 2, \ldots$, the diagram that we obtain shows how the power of the periodic signal is distributed among the various frequency components. This diagram, which is illustrated in Fig. 4.1.2, is called the *power density spectrum*[1] of the periodic signal $x(t)$. Since the power in a periodic signal exists only at discrete values of frequencies (i.e., $F = 0, \pm F_0, \pm 2F_0, \ldots$), the signal is said to have a *line spectrum.* The spacing between two consecutive spectral lines is equal to the reciprocal of the fundamental period T_p, whereas the shape of the spectrum (i.e., the power distribution of the signal), depends on the time-domain characteristics of the signal.

[1] This function is also called the *power spectral density* or, simply, the *power spectrum.*

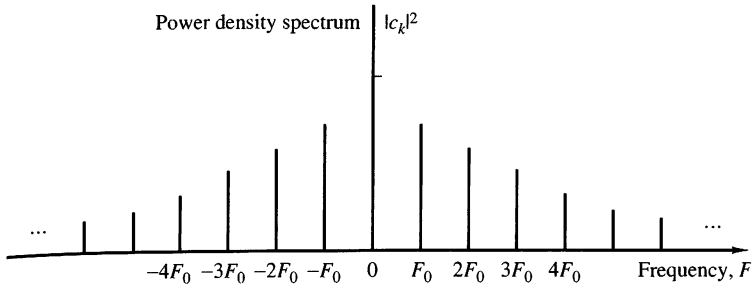


Figure 4.1.2 Power density spectrum of a continuous-time periodic signal.

As indicated in the preceding section, the Fourier series coefficients $\{c_k\}$ are complex valued, that is, they can be represented as

$$c_k = |c_k|e^{j\theta_k}$$

where

$$\theta_k = \measuredangle c_k$$

Instead of plotting the power density spectrum, we can plot the magnitude voltage spectrum $\{|c_k|\}$ and the phase spectrum $\{\theta_k\}$ as a function of frequency. Clearly, the power spectral density in the periodic signal is simply the square of the magnitude spectrum. The phase information is totally destroyed (or does not appear) in the power spectral density.

If the periodic signal is real valued, the Fourier series coefficients $\{c_k\}$ satisfy the condition

$$c_{-k} = c_k^*$$

Consequently, $|c_k|^2 = |c_k^*|^2$. Hence the power spectrum is a symmetric function of frequency. This condition also means that the magnitude spectrum is symmetric (even function) about the origin and the phase spectrum is an odd function. As a consequence of the symmetry, it is sufficient to specify the spectrum of a real periodic signal for positive frequencies only. Furthermore, the total average power can be expressed as

$$P_x = c_0^2 + 2\sum_{k=1}^{\infty} |c_k|^2 \tag{4.1.15}$$

$$= a_0^2 + \frac{1}{2}\sum_{k=1}^{\infty}(a_k^2 + b_k^2) \tag{4.1.16}$$

which follows directly from the relationships given in Section 4.1.1 among $\{a_k\}$, $\{b_k\}$, and $\{c_k\}$ coefficients in the Fourier series expressions.

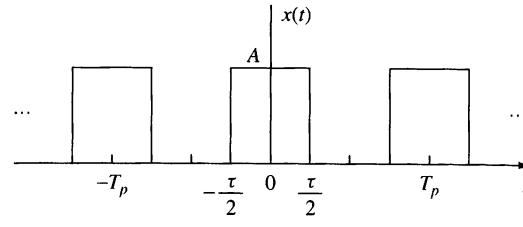


Figure 4.1.3
Continuous-time periodic train of rectangular pulses.

EXAMPLE 4.1.1

Determine the Fourier series and the power density spectrum of the rectangular pulse train signal illustrated in Fig 4.1.3.

Solution. The signal is periodic with fundamental period T_p and, clearly, satisfies the Dirichlet conditions. Consequently, we can represent the signal in the Fourier series given by (4.1.8) with the Fourier coefficients specified by (4.1.9).

Since $x(t)$ is an even signal [i.e., $x(t) = x(-t)$], it is convenient to select the integration interval from $-T_p/2$ to $T_p/2$. Thus (4.1.9) evaluated for $k = 0$ yields

$$c_0 = \frac{1}{T_p}\int_{-T_p/2}^{T_p/2} x(t)\,dt = \frac{1}{T_p}\int_{-\tau/2}^{\tau/2} A\,dt = \frac{A\tau}{T_p} \tag{4.1.17}$$

The term c_0 represents the average value (dc component) of the signal $x(t)$. For $k \neq 0$ we have

$$\begin{aligned} c_k &= \frac{1}{T_p}\int_{-\tau/2}^{\tau/2} Ae^{-j2\pi kF_0 t}\,dt = \frac{A}{T_p}\left[\frac{e^{-j2\pi F_0 kt}}{-j2\pi kF_0}\right]_{-\tau/2}^{\tau/2} \\ &= \frac{A}{\pi F_0 kT_p}\frac{e^{j\pi kF_0\tau} - e^{-j\pi kF_0\tau}}{j2} \\ &= \frac{A\tau}{T_p}\frac{\sin \pi kF_0\tau}{\pi kF_0\tau}, \qquad k = \pm 1, \pm 2, \ldots \end{aligned} \tag{4.1.18}$$

It is interesting to note that the right-hand side of (4.1.18) has the form $(\sin\phi)/\phi$, where $\phi = \pi kF_0\tau$. In this case ϕ takes on discrete values since F_0 and τ are fixed and the index k varies. However, if we plot $(\sin\phi)/\phi$ with ϕ as a continuous parameter over the range $-\infty < \phi < \infty$, we obtain the graph shown in Fig 4.1.4. We observe that this function decays to zero as $\phi \to \pm\infty$, has a maximum value of unity at $\phi = 0$, and is zero at multiples of π (i.e., at $\phi = m\pi$, $m = \pm 1, \pm 2, \ldots$). It is clear that the Fourier coefficients given by (4.1.18) are the sample values of the $(\sin\phi)/\phi$ function for $\phi = \pi kF_0\tau$ and scaled in amplitude by $A\tau/T_p$.

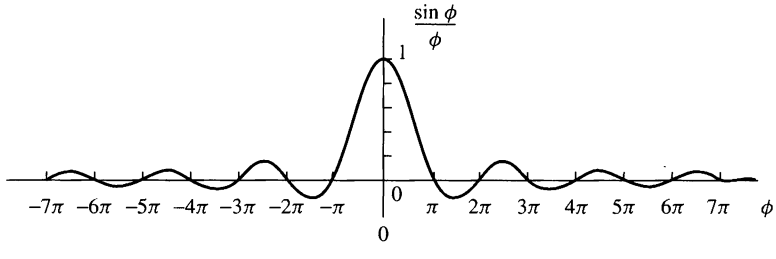


Figure 4.1.4 The function $(\sin\phi)/\phi$.

Since the periodic function $x(t)$ is even, the Fourier coefficients c_k are real. Consequently, the phase spectrum is either zero, when c_k is positive, or π when c_k is negative. Instead of plotting the magnitude and phase spectra separately, we may simply plot $\{c_k\}$ on a single graph, showing both the positive and negative values c_k on the graph. This is commonly done in practice when the Fourier coefficients $\{c_k\}$ are real.

Figure 4.1.5 illustrates the Fourier coefficients of the rectangular pulse train when T_p is fixed and the pulse width τ is allowed to vary. In this case $T_p = 0.25$ second, so that $F_0 = 1/T_p = 4$ Hz and $\tau = 0.05T_p$, $\tau = 0.1T_p$, and $\tau = 0.2T_p$. We observe that the effect of decreasing τ while keeping T_p fixed is to spread out the signal power over the frequency range. The spacing between adjacent spectral lines is $F_0 = 4$ Hz, independent of the value of the pulse width τ.

On the other hand, it is also instructive to fix τ and vary the period T_p when $T_p > \tau$. Figure 4.1.6 illustrates this condition when $T_p = 5\tau$, $T_p = 10\tau$, and $T_p = 20\tau$. In this case, the spacing between adjacent spectral lines decreases as T_p increases. In the limit as $T_p \to \infty$, the Fourier coefficients c_k approach zero due to the factor of T_p in the denominator of (4.1.18). This behavior is consistent with the fact that as $T_p \to \infty$ and τ remains fixed, the resulting signal is no longer a power signal. Instead, it becomes an energy signal and its average power is zero. The spectra of finite energy signals are described in the next section.

We also note that if $k \neq 0$ and $\sin(\pi k F_0 \tau) = 0$, then $c_k = 0$. The harmonics with zero power occur at frequencies kF_0 such that $\pi(kF_0)\tau = m\pi$, $m = \pm 1, \pm 2, \ldots$, or at $kF_0 = m/\tau$. For example, if $F_0 = 4$ Hz and $\tau = 0.2T_p$, it follows that the spectral components at ± 20 Hz, ± 40 Hz, ... have zero power. These frequencies correspond to the Fourier coefficients c_k, $k = \pm 5, \pm 10, \pm 15, \ldots$. On the other hand, if $\tau = 0.1T_p$, the spectral components with zero power are $k = \pm 10, \pm 20, \pm 30, \ldots$.

The power density spectrum for the rectangular pulse train is

$$|c_k|^2 = \begin{cases} \left(\dfrac{A\tau}{T_p}\right)^2, & k = 0 \\ \left(\dfrac{A\tau}{T_p}\right)^2 \left(\dfrac{\sin \pi k F_0 \tau}{\pi k F_0 \tau}\right)^2, & k = \pm 1, \pm 2, \ldots \end{cases} \tag{4.1.19}$$

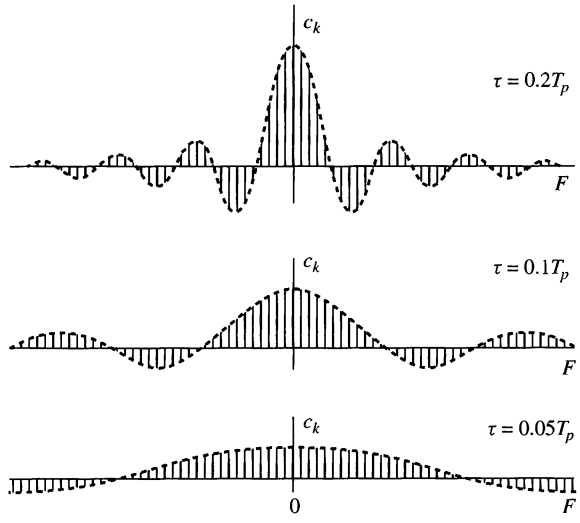


Figure 4.1.5
Fourier coefficients of the rectangular pulse train when T_p is fixed and the pulse width τ varies.

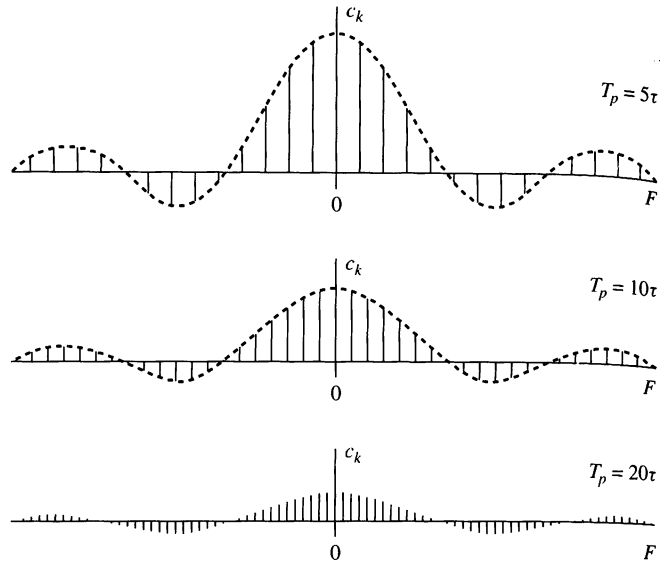


Figure 4.1.6
Fourier coefficient of a rectangular pulse train with fixed pulse width τ and varying period T_p.

4.1.3 The Fourier Transform for Continuous-Time Aperiodic Signals

In Section 4.1.1 we developed the Fourier series to represent a periodic signal as a linear combination of harmonically related complex exponentials. As a consequence of the periodicity, we saw that these signals possess line spectra with equidistant lines. The line spacing is equal to the fundamental frequency, which in turn is the inverse of the fundamental period of the signal. We can view the fundamental period as providing the number of lines per unit of frequency (line density), as illustrated in Fig 4.1.6.

With this interpretation in mind, it is apparent that if we allow the period to increase without limit, the line spacing tends toward zero. In the limit, when the period becomes infinite, the signal becomes aperiodic and its spectrum becomes continuous. This argument suggests that the spectrum of an aperiodic signal will be the envelope of the line spectrum in the corresponding periodic signal obtained by repeating the aperiodic signal with some period T_p.

Let us consider an aperiodic signal $x(t)$ with finite duration as shown in Fig 4.1.7(a). From this aperiodic signal, we can create a periodic signal $x_p(t)$ with period T_p, as shown in Fig 4.1.7(b). Clearly, $x_p(t) = x(t)$ in the limit as $T_p \to \infty$, that is,

$$x(t) = \lim_{T_p \to \infty} x_p(t)$$

This interpretation implies that we should be able to obtain the spectrum of $x(t)$ from the spectrum of $x_p(t)$ simply by taking the limit as $T_p \to \infty$.

We begin with the Fourier series representation of $x_p(t)$,

$$x_p(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 t}, \qquad F_0 = \frac{1}{T_p} \tag{4.1.20}$$

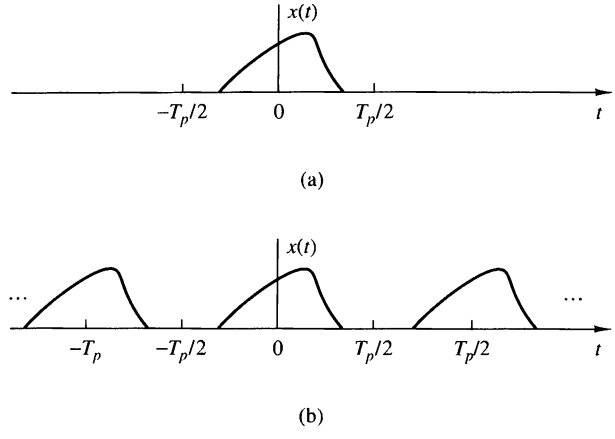


Figure 4.1.7
(a) Aperiodic signal $x(t)$ and (b) periodic signal $x_p(t)$ constructed by repeating $x(t)$ with a period T_p.

where

$$c_k = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} x_p(t) e^{-j2\pi k F_0 t}\, dt \tag{4.1.21}$$

Since $x_p(t) = x(t)$ for $-T_p/2 \le t \le T_p/2$, (4.1.21) can be expressed as

$$c_k = \frac{1}{T_p} \int_{-T_p/2}^{T_p/2} x(t) e^{-j2\pi k F_0 t}\, dt \tag{4.1.22}$$

It is also true that $x(t) = 0$ for $|t| > T_p/2$. Consequently, the limits on the integral in (4.1.22) can be replaced by $-\infty$ and ∞. Hence

$$c_k = \frac{1}{T_p} \int_{-\infty}^{\infty} x(t) e^{-j2\pi k F_0 t}\, dt \tag{4.1.23}$$

Let us now define a function $X(F)$, called the *Fourier transform* of $x(t)$, as

$$X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi F t}\, dt \tag{4.1.24}$$

$X(F)$ is a function of the continuous variable F. It does not depend on T_p or F_0. However, if we compare (4.1.23) and (4.1.24), it is clear that the Fourier coefficients c_k can be expressed in terms of $X(F)$ as

$$c_k = \frac{1}{T_p} X(kF_0)$$

or equivalently,

$$T_p c_k = X(kF_0) = X\left(\frac{k}{T_p}\right) \tag{4.1.25}$$

Thus the Fourier coefficients are samples of $X(F)$ taken at multiples of F_0 and scaled by F_0 (multiplied by $1/T_p$). Substitution for c_k from (4.1.25) into (4.1.20) yields

$$x_p(t) = \frac{1}{T_p} \sum_{k=-\infty}^{\infty} X\left(\frac{k}{T_p}\right) e^{j2\pi k F_0 t} \tag{4.1.26}$$

We wish to take the limit of (4.1.26) as T_p approaches infinity. First, we define $\Delta F = 1/T_p$. With this substitution, (4.1.26) becomes

$$x_p(t) = \sum_{k=-\infty}^{\infty} X(k\Delta F) e^{j2\pi k \Delta F t} \Delta F \tag{4.1.27}$$

It is clear that in the limit as T_p approaches infinity, $x_p(t)$ reduces to $x(t)$. Also, ΔF becomes the differential dF and $k\,\Delta F$ becomes the continuous frequency variable F. In turn, the summation in (4.1.27) becomes an integral over the frequency variable F. Thus

$$\begin{aligned} \lim_{T_p \to \infty} x_p(t) = x(t) &= \lim_{\Delta F \to 0} \sum_{k=-\infty}^{\infty} X(k\Delta F) e^{-j2\pi k \Delta F t} \Delta F \\ x(t) &= \int_{-\infty}^{\infty} X(F) e^{j2\pi F t}\, dF \end{aligned} \tag{4.1.28}$$

This integral relationship yields $x(t)$ when $X(F)$ is known, and it is called the *inverse Fourier transform.*

This concludes our heuristic derivation of the Fourier transform pair given by (4.1.24) and (4.1.28) for an aperiodic signal $x(t)$. Although the derivation is not mathematically rigorous, it led to the desired Fourier transform relationships with relatively simple intuitive arguments. In summary, the frequency analysis of continuous-time aperiodic signals involves the following Fourier transform pair.

Frequency Analysis of Continuous-Time Aperiodic Signals

Synthesis equation (inverse transform)	$x(t) = \int_{-\infty}^{\infty} X(F) e^{j2\pi F t}\, dF$ (4.1.29)
Analysis equation (direct transform)	$X(F) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi F t}\, dt$ (4.1.30)

It is apparent that the essential difference between the Fourier series and the Fourier transform is that the spectrum in the latter case is continuous and hence the synthesis of an aperiodic signal from its spectrum is accomplished by means of integration instead of summation.

Finally, we wish to indicate that the Fourier transform pair in (4.1.29) and (4.1.30) can be expressed in terms of the radial frequency variable $\Omega = 2\pi F$. Since $dF = d\Omega/2\pi$, (4.1.29) and (4.1.30) become

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\Omega) e^{j\Omega t}\, d\Omega \tag{4.1.31}$$

$$X(\Omega) = \int_{-\infty}^{\infty} x(t) e^{-j\Omega t}\, dt \tag{4.1.32}$$

The set of conditions that guarantee the existence of the Fourier transform is the *Dirichlet conditions*, which may be expressed as:

1. The signal $x(t)$ has a finite number of finite discontinuities.
2. The signal $x(t)$ has a finite number of maxima and minima.
3. The signal $x(t)$ is absolutely integrable, that is,

$$\int_{-\infty}^{\infty} |x(t)|\, dt < \infty \tag{4.1.33}$$

The third condition follows easily from the definition of the Fourier transform, given in (4.1.30). Indeed,

$$|X(F)| = \left| \int_{-\infty}^{\infty} x(t) e^{-j2\pi Ft}\, dt \right| \leq \int_{-\infty}^{\infty} |x(t)|\, dt$$

Hence $|X(F)| < \infty$ if (4.1.33) is satisfied.

A weaker condition for the existence of the Fourier transform is that $x(t)$ has finite energy; that is,

$$\int_{-\infty}^{\infty} |x(t)|^2\, dt < \infty \tag{4.1.34}$$

Note that if a signal $x(t)$ is absolutely integrable, it will also have finite energy. That is, if

$$\int_{-\infty}^{\infty} |x(t)|\, dt < \infty$$

then

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2\, dt < \infty \tag{4.1.35}$$

However, the converse is not true. That is, a signal may have finite energy but may not be absolutely integrable. For example, the signal

$$x(t) = \frac{\sin 2\pi F_0 t}{\pi t} \tag{4.1.36}$$

is square integrable but is not absolutely integrable. This signal has the Fourier transform

$$X(F) = \begin{cases} 1, & |F| \leq F_0 \\ 0, & |F| > F_0 \end{cases} \tag{4.1.37}$$

Since this signal violates (4.1.33), it is apparent that the Dirichlet conditions are sufficient but not necessary for the existence of the Fourier transform. In any case, nearly all finite energy signals have a Fourier transform, so that we need not worry about the pathological signals, which are seldom encountered in practice.

4.1.4 Energy Density Spectrum of Aperiodic Signals

Let $x(t)$ be any finite energy signal with Fourier transform $X(F)$. Its energy is

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 \, dt$$

which, in turn, may be expressed in terms of $X(F)$ as follows:

$$\begin{aligned} E_x &= \int_{-\infty}^{\infty} x(t)x^*(t) \, dt \\ &= \int_{-\infty}^{\infty} x(t) \, dt \left[\int_{-\infty}^{\infty} X^*(F) e^{-j2\pi Ft} \, dF \right] \\ &= \int_{-\infty}^{\infty} X^*(F) \, dF \left[\int_{-\infty}^{\infty} x(t) e^{-j2\pi Ft} \, dt \right] \\ &= \int_{-\infty}^{\infty} |X(F)|^2 \, dF \end{aligned}$$

Therefore, we conclude that

$$E_x = \int_{-\infty}^{\infty} |x(t)|^2 \, dt = \int_{-\infty}^{\infty} |X(F)|^2 \, dF \tag{4.1.38}$$

This is *Parseval's relation* for aperiodic, finite energy signals and expresses the principle of conservation of energy in the time and frequency domains.

The spectrum $X(F)$ of a signal is, in general, complex valued. Consequently, it is usually expressed in polar form as

$$X(F) = |X(F)|e^{j\Theta(F)}$$

where $|X(F)|$ is the magnitude spectrum and $\Theta(F)$ is the phase spectrum,

$$\Theta(F) = \measuredangle X(F)$$

On the other hand, the quantity

$$S_{xx}(F) = |X(F)|^2 \tag{4.1.39}$$

which is the integrand in (4.1.38), represents the distribution of energy in the signal as a function of frequency. Hence $S_{xx}(F)$ is called the *energy density spectrum* of $x(t)$. The integral of $S_{xx}(F)$ over all frequencies gives the total energy in the signal. Viewed in another way, the energy in the signal $x(t)$ over a band of frequencies $F_1 \leq F \leq F_1 + \Delta F$ is

$$\int_{F_1}^{F_1+\Delta F} S_{xx}(F)\, dF \geq 0$$

which implies that $S_{xx}(f) \geq 0$ for all F.

From (4.1.39) we observe that $S_{xx}(F)$ does not contain any phase information [i.e., $S_{xx}(F)$ is purely real and nonnegative]. Since the phase spectrum of $x(t)$ is not contained in $S_{xx}(F)$, it is impossible to reconstruct the signal given $S_{xx}(F)$.

Finally, as in the case of Fourier series, it is easily shown that if the signal $x(t)$ is real, then

$$|X(-F)| = |X(F)| \tag{4.1.40}$$

$$\measuredangle X(-F) = -\measuredangle X(F) \tag{4.1.41}$$

By combining (4.1.40) and (4.1.39), we obtain

$$S_{xx}(-F) = S_{xx}(F) \tag{4.1.42}$$

In other words, the energy density spectrum of a real signal has even symmetry.

EXAMPLE 4.1.2

Determine the Fourier transform and the energy density spectrum of a rectangular pulse signal defined as

$$x(t) = \begin{cases} A, & |t| \leq \tau/2 \\ 0, & |t| > \tau/2 \end{cases} \tag{4.1.43}$$

and illustrated in Fig 4.1.8(a).

Solution. Clearly, this signal is aperiodic and satisfies the Dirichlet conditions. Hence its Fourier transform exists. By applying (4.1.30), we find that

$$X(F) = \int_{-\tau/2}^{\tau/2} Ae^{-j2\pi Ft}\, dt = A\tau \frac{\sin \pi F\tau}{\pi F\tau} \tag{4.1.44}$$

We observe that $X(F)$ is real and hence it can be depicted graphically using only one diagram, as shown in Fig 4.1.8(b). Obviously, $X(F)$ has the shape of the $(\sin\phi)/\phi$ function shown in Fig 4.1.4. Hence the spectrum of the rectangular pulse is the envelope of the line spectrum (Fourier coefficients) of the periodic signal obtained by periodically repeating the pulse with period T_p as in Fig 4.1.3. In other words, the Fourier coefficients c_k in the corresponding periodic signal $x_p(t)$ are simply samples of $X(F)$ at frequencies $kF_0 = k/T_p$. Specifically,

$$c_k = \frac{1}{T_p} X(kF_0) = \frac{1}{T_p} X\left(\frac{k}{T_p}\right) \tag{4.1.45}$$

From (4.1.44) we note that the zero crossings of $X(F)$ occur at multiples of $1/\tau$. Furthermore, the width of the main lobe, which contains most of the signal energy, is equal to $2/\tau$. As the

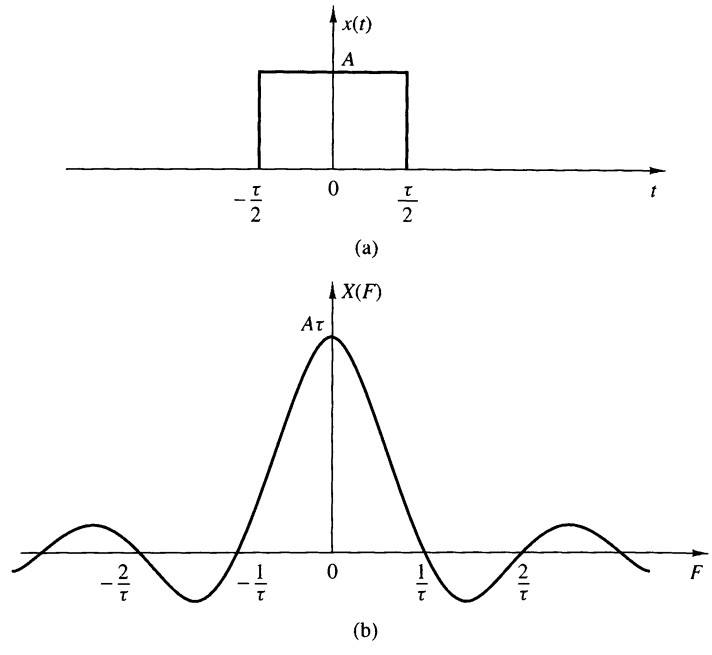


Figure 4.1.8
(a) Rectangular pulse and (b) its Fourier transform.

pulse duration τ decreases (increases), the main lobe becomes broader (narrower) and more energy is moved to the higher (lower) frequencies, as illustrated in Fig 4.1.9. Thus as the signal

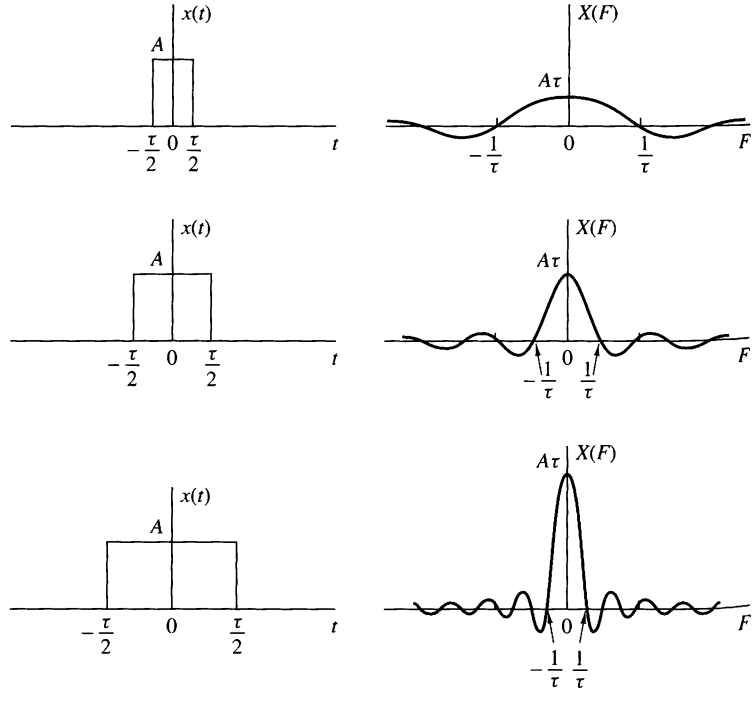


Figure 4.1.9
Fourier transform of a rectangular pulse for various width values.

pulse is expanded (compressed) in time, its transform is compressed (expanded) in frequency. This behavior, between the time function and its spectrum, is a type of uncertainty principle that appears in different forms in various branches of science and engineering.

Finally, the energy density spectrum of the rectangular pulse is

$$S_{xx}(F) = (A\tau)^2 \left(\frac{\sin \pi F\tau}{\pi F\tau} \right)^2 \tag{4.1.46}$$

4.2 Frequency Analysis of Discrete-Time Signals

In Section 4.1 we developed the Fourier series representation for continuous-time periodic (power) signals and the Fourier transform for finite energy aperiodic signals. In this section we repeat the development for the class of discrete-time signals.

As we have observed from the discussion of Section 4.1, the Fourier series representation of a continuous-time periodic signal can consist of an infinite number of frequency components, where the frequency spacing between two successive harmonically related frequencies is $1/T_p$, and where T_p is the fundamental period. Since the frequency range for continuous-time signals extends from $-\infty$ to ∞, it is possible to have signals that contain an infinite number of frequency components. In contrast, the frequency range for discrete-time signals is unique over the interval $(-\pi, \pi)$ or $(0, 2\pi)$. A discrete-time signal of fundamental period N can consist of frequency components separated by $2\pi/N$ radians or $f = 1/N$ cycles. Consequently, the Fourier series representation of the discrete-time periodic signal will contain at most N frequency components. This is the basic difference between the Fourier series representations for continuous-time and discrete-time periodic signals.

4.2.1 The Fourier Series for Discrete-Time Periodic Signals

Suppose that we are given a periodic sequence $x(n)$ with period N, that is, $x(n) = x(n+N)$ for all n. The Fourier series representation for $x(n)$ consists of N harmonically related exponential functions

$$e^{j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1$$

and is expressed as

$$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N} \tag{4.2.1}$$

where the $\{c_k\}$ are the coefficients in the series representation.

To derive the expression for the Fourier coefficients, we use the following formula:

$$\sum_{n=0}^{N-1} e^{j2\pi kn/N} = \begin{cases} N, & k = 0, \pm N, \pm 2N, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{4.2.2}$$

Note the similarity of (4.2.2) with the continuous-time counterpart in (4.1.3). The proof of (4.2.2) follows immediately from the application of the geometric summation formula

$$\sum_{n=0}^{N-1} a^n = \begin{cases} N, & a = 1 \\ \dfrac{1-a^N}{1-a}, & a \neq 1 \end{cases} \tag{4.2.3}$$

The expression for the Fourier coefficients c_k can be obtained by multiplying both sides of (4.2.1) by the exponential $e^{-j2\pi ln/N}$ and summing the product from $n = 0$ to $n = N - 1$. Thus

$$\sum_{n=0}^{N-1} x(n)e^{-j2\pi ln/N} = \sum_{n=0}^{N-1}\sum_{k=0}^{N-1} c_k e^{j2\pi(k-l)n/N} \tag{4.2.4}$$

If we perform the summation over n first, in the right-hand side of (4.2.4), we obtain

$$\sum_{n=0}^{N-1} e^{j2\pi(k-l)n/N} = \begin{cases} N, & k-l = 0, \pm N, \pm 2N, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{4.2.5}$$

where we have made use of (4.2.2). Therefore, the right-hand side of (4.2.4) reduces to Nc_l and hence

$$c_l = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi ln/N}, \qquad l = 0, 1, \ldots, N-1 \tag{4.2.6}$$

Thus we have the desired expression for the Fourier coefficients in terms of the signal $x(n)$.

The relationships (4.2.1) and (4.2.6) for the frequency analysis of discrete-time signals are summarized below.

Frequency Analysis of Discrete-Time Periodic Signals

Synthesis equation	$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}$	(4.2.7)
Analysis equation	$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$	(4.2.8)

Equation (4.2.7) is often called the *discrete-time Fourier series* (DTFS). The Fourier coefficients $\{c_k\}$, $k = 0, 1, \ldots, N-1$ provide the description of $x(n)$ in the frequency domain, in the sense that c_k represents the amplitude and phase associated with the frequency component

$$s_k(n) = e^{j2\pi kn/N} = e^{j\omega_k n}$$

where $\omega_k = 2\pi k/N$.

We recall from Section 1.3.3 that the functions $s_k(n)$ are periodic with period N. Hence $s_k(n) = s_k(n + N)$. In view of this periodicity, it follows that the Fourier coefficients c_k, when viewed beyond the range $k = 0, 1, \ldots, N - 1$, also satisfy a periodicity condition. Indeed, from (4.2.8), which holds for every value of k, we have

$$c_{k+N} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi(k+N)n/N} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} = c_k \tag{4.2.9}$$

Therefore, the Fourier series coefficients $\{c_k\}$ form a periodic sequence when extended outside of the range $k = 0, 1, \ldots, N - 1$. Hence

$$c_{k+N} = c_k$$

that is, $\{c_k\}$ is a periodic sequence with fundamental period N. *Thus the spectrum of a signal $x(n)$, which is periodic with period N, is a periodic sequence with period N.* Consequently, any N consecutive samples of the signal or its spectrum provide a complete description of the signal in the time or frequency domains.

Although the Fourier coefficients form a periodic sequence, we will focus our attention on the single period with range $k = 0, 1, \ldots, N - 1$. This is convenient, since in the frequency domain, this amounts to covering the fundamental range $0 \leq \omega_k = 2\pi k/N < 2\pi$, for $0 \leq k \leq N - 1$. In contrast, the frequency range $-\pi < \omega_k = 2\pi k/N \leq \pi$ corresponds to $-N/2 < k \leq N/2$, which creates an inconvenience when N is odd. Clearly, if we use a sampling frequency F_s, the range $0 \leq k \leq N - 1$ corresponds to the frequency range $0 \leq F < F_s$.

EXAMPLE 4.2.1

Determine the spectra of the signals

(a) $x(n) = \cos \sqrt{2}\pi n$

(b) $x(n) = \cos \pi n/3$

(c) $x(n)$ is periodic with period $N = 4$ and $x(n) = \{\underset{\uparrow}{1}, 1, 0, 0\}$

Solution.

(a) For $\omega_0 = \sqrt{2}\pi$, we have $f_0 = 1/\sqrt{2}$. Since f_0 is not a rational number, the signal is not periodic. Consequently, this signal cannot be expanded in a Fourier series. Nevertheless, the signal does possess a spectrum. Its spectral content consists of the single frequency component at $\omega = \omega_0 = \sqrt{2}\pi$.

(b) In this case $f_0 = \frac{1}{6}$ and hence $x(n)$ is periodic with fundamental period $N = 6$. From (4.2.8) we have

$$c_k = \frac{1}{6}\sum_{n=0}^{5} x(n)e^{-j2\pi kn/6}, \qquad k = 0, 1, \ldots, 5$$

However, $x(n)$ can be expressed as

$$x(n) = \cos\frac{2\pi n}{6} = \frac{1}{2}e^{j2\pi n/6} + \frac{1}{2}e^{-j2\pi n/6}$$

which is already in the form of the exponential Fourier series in (4.2.7). In comparing the two exponential terms in $x(n)$ with (4.2.7), it is apparent that $c_1 = \frac{1}{2}$. The second exponential in $x(n)$ corresponds to the term $k = -1$ in (4.2.7). However, this term can also be written as

$$e^{-j2\pi n/6} = e^{j2\pi(5-6)n/6} = e^{j2\pi(5n)/6}$$

which means that $c_{-1} = c_5$. But this is consistent with (4.2.9), and with our previous observation that the Fourier series coefficients form a periodic sequence of period N. Consequently, we conclude that

$$c_0 = c_2 = c_3 = c_4 = 0$$
$$c_1 = \frac{1}{2}, \qquad c_5 = \frac{1}{2}$$

(c) From (4.2.8), we have

$$c_k = \frac{1}{4}\sum_{n=0}^{3} x(n)e^{-j2\pi kn/4}, \qquad k = 0, 1, 2, 3$$

or

$$c_k = \frac{1}{4}(1 + e^{-j\pi k/2}), \qquad k = 0, 1, 2, 3$$

For $k = 0, 1, 2, 3$ we obtain

$$c_0 = \frac{1}{2}, \qquad c_1 = \frac{1}{4}(1 - j), \qquad c_2 = 0, \qquad c_3 = \frac{1}{4}(1 + j)$$

The magnitude and phase spectra are

$$|c_0| = \frac{1}{2}, \qquad |c_1| = \frac{\sqrt{2}}{4}, \qquad |c_2| = 0, \qquad |c_3| = \frac{\sqrt{2}}{4}$$
$$\measuredangle c_0 = 0, \qquad \measuredangle c_1 = -\frac{\pi}{4}, \qquad \measuredangle c_2 = \text{undefined}, \qquad \measuredangle c_3 = \frac{\pi}{4}$$

Figure 4.2.1 illustrates the spectral content of the signals in (b) and (c).

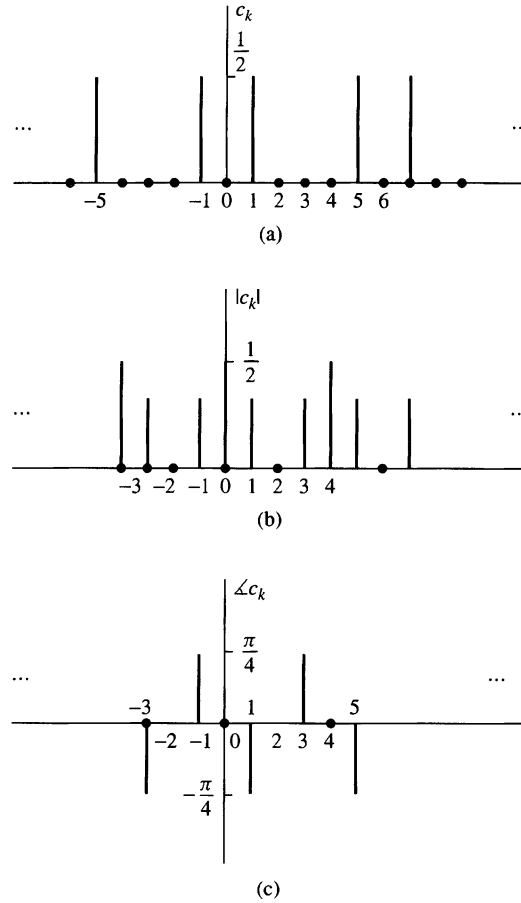


Figure 4.2.1
Spectra of the periodic signals discussed in Example 4.2.1 (b) and (c).

4.2.2 Power Density Spectrum of Periodic Signals

The average power of a discrete-time periodic signal with period N was defined in (2.1.23) as

$$P_x = \frac{1}{N}\sum_{n=0}^{N-1} |x(n)|^2 \tag{4.2.10}$$

We shall now derive an expression for P_x in terms of the Fourier coefficient $\{c_k\}$.

If we use the relation (4.2.7) in (4.2.10), we have

$$\begin{aligned} P_x &= \frac{1}{N}\sum_{n=0}^{N-1} x(n)x^*(n) \\ &= \frac{1}{N}\sum_{n=0}^{N-1} x(n)\left(\sum_{k=0}^{N-1} c_k^* e^{-j2\pi kn/N}\right) \end{aligned}$$

Now, we can interchange the order of the two summations and make use of (4.2.8), obtaining

$$\begin{aligned} P_x &= \sum_{k=0}^{N-1} c_k^* \left[\frac{1}{N} \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \right] \\ &= \sum_{k=0}^{N-1} |c_k|^2 = \frac{1}{N} \sum_{n=0}^{N-1} |x(n)|^2 \end{aligned} \tag{4.2.11}$$

which is the desired expression for the average power in the periodic signal. In other words, the average power in the signal is the sum of the powers of the individual frequency components. We view (4.2.11) as a Parseval's relation for discrete-time periodic signals. The sequence $|c_k|^2$ for $k = 0, 1, \ldots, N-1$ is the distribution of power as a function of frequency and is called the *power density spectrum* of the periodic signal.

If we are interested in the energy of the sequence $x(n)$ over a single period, (4.2.11) implies that

$$E_N = \sum_{n=0}^{N-1} |x(n)|^2 = N \sum_{k=0}^{N-1} |c_k|^2 \tag{4.2.12}$$

which is consistent with our previous results for continuous-time periodic signals. If the signal $x(n)$ is real [i.e., $x^*(n) = x(n)$], then, proceeding as in Section 4.2.1, we can easily show that

$$c_k^* = c_{-k} \tag{4.2.13}$$

or equivalently,

$$|c_{-k}| = |c_k| \qquad \text{(even symmetry)} \tag{4.2.14}$$

$$-\measuredangle c_{-k} = \measuredangle c_k \qquad \text{(odd symmetry)} \tag{4.2.15}$$

These symmetry properties for the magnitude and phase spectra of a periodic signal, in conjunction with the periodicity property, have very important implications on the frequency range of discrete-time signals.

Indeed, by combining (4.2.9) with (4.2.14) and (4.2.15), we obtain

$$|c_k| = |c_{N-k}| \tag{4.2.16}$$

and

$$\measuredangle c_k = -\measuredangle c_{N-k} \tag{4.2.17}$$

More specifically, we have

$$\begin{array}{lll} |c_0| = |c_N|, & \measuredangle c_0 = -\measuredangle c_N = 0 & \\ |c_1| = |c_{N-1}|, & \measuredangle c_1 = -\measuredangle c_{N-1} & \\ |c_{N/2}| = |c_{N/2}|, & \measuredangle c_{N/2} = 0 & \text{if } N \text{ is even} \\ |c_{(N-1)/2}| = |c_{(N+1)/2}|, & \measuredangle c_{(N-1)/2} = -\measuredangle c_{(N+1)/2} & \text{if } N \text{ is odd} \end{array} \tag{4.2.18}$$

Thus, for a real signal, the spectrum c_k, $k = 0, 1, \ldots, N/2$ for N even, or $k = 0, 1, \ldots, (N-1)/2$ for N odd, completely specifies the signal in the frequency domain. Clearly, this is consistent with the fact that the highest relative frequency that can be represented by a discrete-time signal is equal to π. Indeed, if $0 \leq \omega_k = 2\pi k/N \leq \pi$, then $0 \leq k \leq N/2$.

By making use of these symmetry properties of the Fourier series coefficients of a real signal, the Fourier series in (4.2.7) can also be expressed in the alternative forms

$$x(n) = c_0 + 2\sum_{k=1}^{L} |c_k| \cos\left(\frac{2\pi}{N}kn + \theta_k\right) \tag{4.2.19}$$

$$= a_0 + \sum_{k=1}^{L}\left(a_k \cos\frac{2\pi}{N}kn - b_k \sin\frac{2\pi}{N}kn\right) \tag{4.2.20}$$

where $a_0 = c_0$, $a_k = 2|c_k| \cos\theta_k$, $b_k = 2|c_k| \sin\theta_k$, and $L = N/2$ if N is even and $L = (N-1)/2$ if N is odd.

Finally, we note that as in the case of continuous-time signals, the power density spectrum $|c_k|^2$ does not contain any phase information. Furthermore, the spectrum is discrete and periodic with a fundamental period equal to that of the signal itself.

EXAMPLE 4.2.2 Periodic "Square-Wave" Signal

Determine the Fourier series coefficients and the power density spectrum of the periodic signal shown in Fig 4.2.2.

Solution. By applying the analysis equation (4.2.8) to the signal shown in Fig 4.2.2, we obtain

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} = \frac{1}{N}\sum_{n=0}^{L-1} Ae^{-j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1$$

which is a geometric summation. Now we can use (4.2.3) to simplify the summation above. Thus we obtain

$$c_k = \frac{A}{N}\sum_{n=0}^{L-1}(e^{-j2\pi k/N})^n = \begin{cases} \dfrac{AL}{N}, & k = 0 \\ \dfrac{A}{N}\dfrac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}}, & k = 1, 2, \ldots, N-1 \end{cases}$$

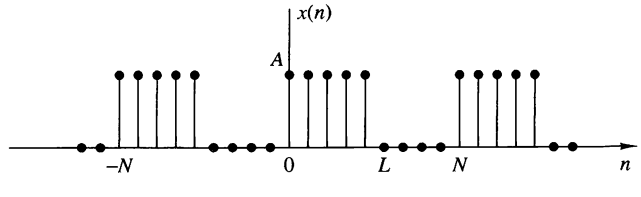


Figure 4.2.2
Discrete-time periodic square-wave signal.

The last expression can be simplified further if we note that

$$\frac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}} = \frac{e^{-j\pi kL/N}}{e^{-j\pi k/N}} \frac{e^{j\pi kL/N} - e^{-j\pi kL/N}}{e^{j\pi k/N} - e^{-j\pi k/N}}$$

$$= e^{-j\pi k(L-1)/N} \frac{\sin(\pi kL/N)}{\sin(\pi k/N)}$$

Therefore,

$$c_k = \begin{cases} \dfrac{AL}{N}, & k = 0, +N, \pm 2N, \ldots \\ \dfrac{A}{N} e^{-j\pi k(L-1)/N} \dfrac{\sin(\pi kL/N)}{\sin(\pi k/N)}, & \text{otherwise} \end{cases} \tag{4.2.21}$$

The power density spectrum of this periodic signal is

$$|c_k|^2 = \begin{cases} \left(\dfrac{AL}{N}\right)^2, & k = 0, +N, \pm 2N, \ldots \\ \left(\dfrac{A}{N}\right)^2 \left(\dfrac{\sin \pi kL/N}{\sin \pi k/N}\right)^2, & \text{otherwise} \end{cases} \tag{4.2.22}$$

Figure 4.2.3 illustrates the plots of $|c_k|^2$ for $L = 2$, $N = 10$ and 40, and $A = 1$.

4.2.3 The Fourier Transform of Discrete-Time Aperiodic Signals

Just as in the case of continuous-time aperiodic energy signals, the frequency analysis of discrete-time aperiodic finite-energy signals involves a Fourier transform of the time-domain signal. Consequently, the development in this section parallels, to a large extent, that given in Section 4.1.3.

The Fourier transform of a finite-energy discrete-time signal $x(n)$ is defined as

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n) e^{-j\omega n} \tag{4.2.23}$$

Physically, $X(\omega)$ represents the frequency content of the signal $x(n)$. In other words, $X(\omega)$ is a decomposition of $x(n)$ into its frequency components.

We observe two basic differences between the Fourier transform of a discrete-time finite-energy signal and the Fourier transform of a finite-energy analog signal. First, for continuous-time signals, the Fourier transform, and hence the spectrum of the signal, have a frequency range of $(-\infty, \infty)$. In contrast, the frequency range for a discrete-time signal is unique over the frequency interval of $(-\pi, \pi)$ or, equivalently, $(0, 2\pi)$. This property is reflected in the Fourier transform of the signal. Indeed, $X(\omega)$

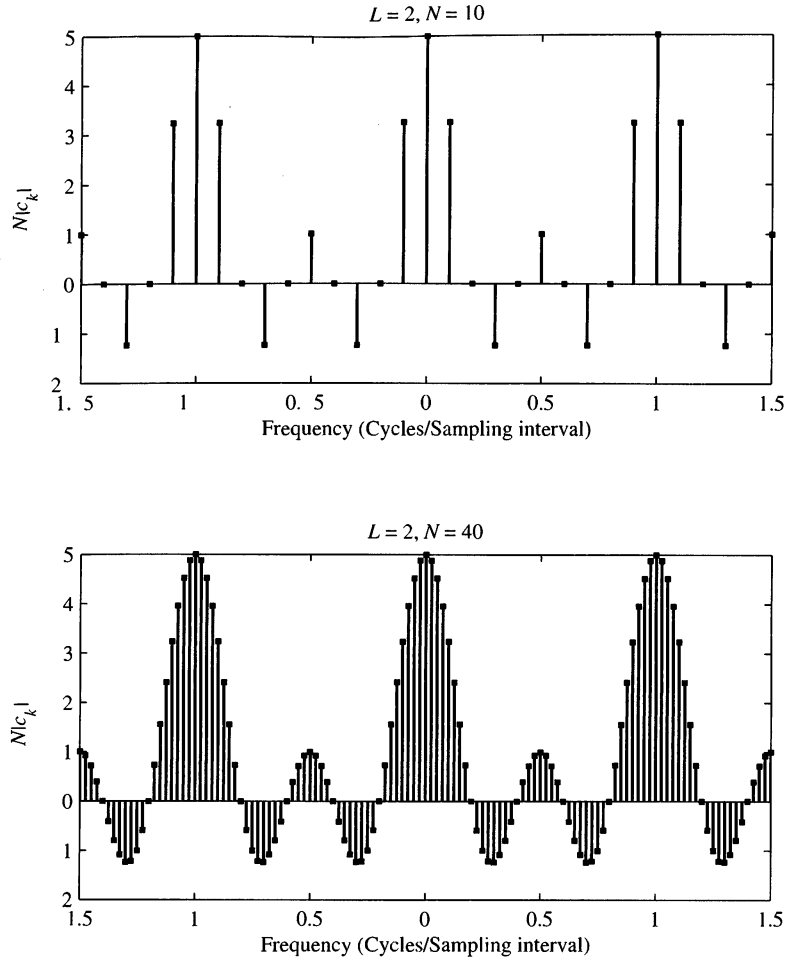


Figure 4.2.3 Plot of the power density spectrum given by (4.2.22).

is periodic with period 2π, that is,

$$\begin{aligned} X(\omega + 2\pi k) &= \sum_{n=-\infty}^{\infty} x(n)e^{-j(\omega+2\pi k)n} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}e^{-j2\pi kn} \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} = X(\omega) \end{aligned} \tag{4.2.24}$$

Hence $X(\omega)$ is periodic with period 2π. But this property is just a consequence of the fact that the frequency range for any discrete-time signal is limited to $(-\pi, \pi)$ or

$(0, 2\pi)$, and any frequency outside this interval is equivalent to a frequency within the interval.

The second basic difference is also a consequence of the discrete-time nature of the signal. Since the signal is discrete in time, the Fourier transform of the signal involves a summation of terms instead of an integral, as in the case of continuous-time signals.

Since $X(\omega)$ is a periodic function of the frequency variable ω, it has a Fourier series expansion, provided that the conditions for the existence of the Fourier series, described previously, are satisfied. In fact, from the definition of the Fourier transform $X(\omega)$ of the sequence $x(n)$, given by (4.2.23), we observe that $X(\omega)$ has the form of a Fourier series. The Fourier coefficients in this series expansion are the values of the sequence $x(n)$.

To demonstrate this point, let us evaluate the sequence $x(n)$ from $X(\omega)$. First, we multiply both sides (4.2.23) by $e^{j\omega m}$ and integrate over the interval $(-\pi, \pi)$. Thus we have

$$\int_{-\pi}^{\pi} X(\omega)e^{j\omega m}\,d\omega = \int_{-\pi}^{\pi}\left[\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}\right]e^{j\omega m}\,d\omega \tag{4.2.25}$$

The integral on the right-hand side of (4.2.25) can be evaluated if we can interchange the order of summation and integration. This interchange can be made if the series

$$X_N(\omega) = \sum_{n=-N}^{N} x(n)e^{-j\omega n}$$

converges uniformly to $X(\omega)$ as $N \to \infty$. Uniform convergence means that, for every ω, $X_N(\omega) \to X(\omega)$, as $N \to \infty$. The convergence of the Fourier transform is discussed in more detail in the following section. For the moment, let us assume that the series converges uniformly, so that we can interchange the order of summation and integration in (4.2.25). Then

$$\int_{-\pi}^{\pi} e^{j\omega(m-n)}\,d\omega = \begin{cases} 2\pi, & m = n \\ 0, & m \neq n \end{cases}$$

Consequently,

$$\sum_{n=-\infty}^{\infty} x(n)\int_{-\pi}^{\pi} e^{j\omega(m-n)}\,d\omega = \begin{cases} 2\pi x(m), & m = n \\ 0, & m \neq n \end{cases} \tag{4.2.26}$$

By combining (4.2.25) and (4.2.26), we obtain the desired result that

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}\,d\omega \tag{4.2.27}$$

If we compare the integral in (4.2.27) with (4.1.9), we note that this is just the expression for the Fourier series coefficient for a function that is periodic with period

2π. The only difference between (4.1.9) and (4.2.27) is the sign on the exponent in the integrand, which is a consequence of our definition of the Fourier transform as given by (4.2.23). Therefore, the Fourier transform of the sequence $x(n)$, defined by (4.2.23), has the form of a Fourier series expansion.

In summary, the *Fourier transform pair for discrete-time signals* is as follows.

Frequency Analysis of Discrete-Time Aperiodic Signals

Synthesis equation (inverse transform)	$x(n) = \frac{1}{2\pi}\int_{2\pi} X(\omega)e^{j\omega n}\,d\omega$ (4.2.28)
Analysis equation (direct transform)	$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$ (4.2.29)

4.2.4 Convergence of the Fourier Transform

In the derivation of the inverse transform given by (4.2.28), we assumed that the series

$$X_N(\omega) = \sum_{n=-N}^{N} x(n)e^{-j\omega n} \tag{4.2.30}$$

converges uniformly to $X(\omega)$, given in the integral of (4.2.25), as $N \to \infty$. By uniform convergence we mean that for each ω,

$$\lim_{N\to\infty} \{\sup_{\omega}|X(\omega) - X_N(\omega)|\} = 0 \tag{4.2.31}$$

Uniform convergence is guaranteed if $x(n)$ is absolutely summable. Indeed, if

$$\sum_{n=-\infty}^{\infty} |x(n)| < \infty \tag{4.2.32}$$

then

$$|X(\omega)| = \left|\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}\right| \le \sum_{n=-\infty}^{\infty} |x(n)| < \infty$$

Hence (4.2.32) is a sufficient condition for the existence of the discrete-time Fourier transform. We note that this is the discrete-time counterpart of the third Dirichlet condition for the Fourier transform of continuous-time signals. The first two conditions do not apply due to the discrete-time nature of $\{x(n)\}$.

Some sequences are not absolutely summable, but they are square summable. That is, they have finite energy

$$E_x = \sum_{n=-\infty}^{\infty} |x(n)|^2 < \infty \tag{4.2.33}$$

which is a weaker condition than (4.2.32). We would like to define the Fourier transform of finite-energy sequences, but we must relax the condition of uniform convergence. For such sequences we can impose a mean-square convergence condition:

$$\lim_{N\to\infty} \int_{-\pi}^{\pi} |X(\omega) - X_N(\omega)|^2 \, d\omega = 0 \tag{4.2.34}$$

Thus the energy in the error $X(\omega) - X_N(\omega)$ tends toward zero, but the error $|X(\omega) - X_N(\omega)|$ does not necessarily tend to zero. In this way we can include finite-energy signals in the class of signals for which the Fourier transform exists.

Let us consider an example from the class of finite-energy signals. Suppose that

$$X(\omega) = \begin{cases} 1, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases} \tag{4.2.35}$$

The reader should remember that $X(\omega)$ is periodic with period 2π. Hence (4.2.35) represents only one period of $X(\omega)$. The inverse transform of $X(\omega)$ results in the sequence

$$\begin{aligned} x(n) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\omega) e^{j\omega n} \, d\omega \\ &= \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} \, d\omega = \frac{\sin \omega_c n}{\pi n}, \qquad n \neq 0 \end{aligned}$$

For $n = 0$, we have

$$x(0) = \frac{1}{2\pi} \int^{\omega_c} -\omega_c \, d\omega = \frac{\omega_c}{\pi}$$

Hence

$$x(n) = \begin{cases} \dfrac{\omega_c}{\pi}, & n = 0 \\ \dfrac{\omega_c}{\pi} \dfrac{\sin \omega_c n}{\omega_c n}, & n \neq 0 \end{cases} \tag{4.2.36}$$

This transform pair is illustrated in Fig 4.2.4.

Sometimes, the sequence $\{x(n)\}$ in (4.2.36) is expressed as

$$x(n) = \frac{\sin \omega_c n}{\pi n}, \qquad -\infty < n < \infty \tag{4.2.37}$$

with the understanding that at $n = 0$, $x(n) = \omega_c/\pi$. We should emphasize, however, that $(\sin \omega_c n)/\pi n$ is not a continuous function, and hence L'Hospital's rule cannot be used to determine $x(0)$.

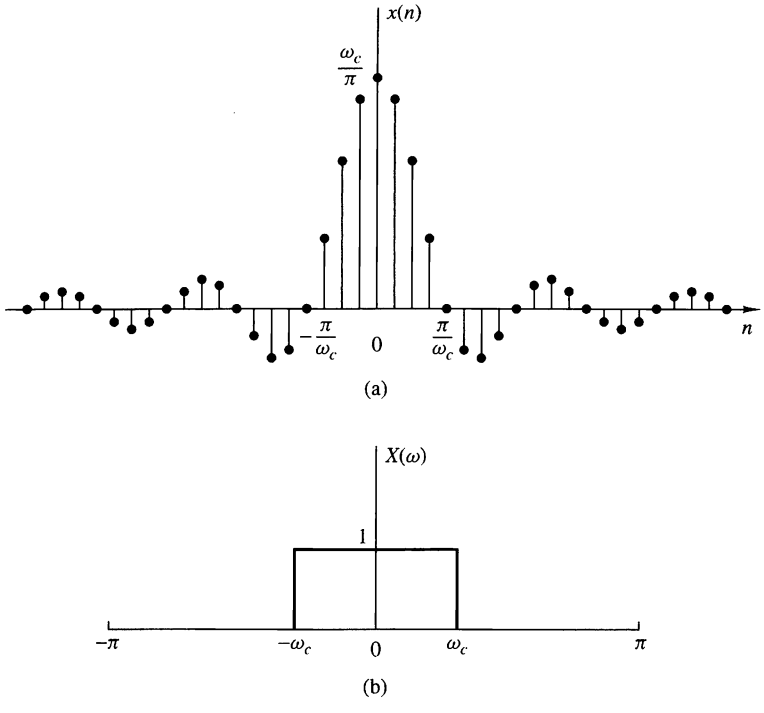


Figure 4.2.4 Fourier transform pair in (4.2.35) and (4.2.36).

Now let us consider the determination of the Fourier transform of the sequence given by (4.2.37). The sequence $\{x(n)\}$ is not absolutely summable. Hence the infinite series

$$\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \frac{\sin \omega_c n}{\pi n} e^{-j\omega n} \tag{4.2.38}$$

does not converge uniformly for all ω. However, the sequence $\{x(n)\}$ has a finite energy $E_x = \omega_c/\pi$ as will be shown in Section 4.3. Hence the sum in (4.2.38) is guaranteed to converge to the $X(\omega)$ given by (4.2.35) in the mean-square sense.

To elaborate on this point, let us consider the finite sum

$$X_N(\omega) = \sum_{n=-N}^{N} \frac{\sin \omega_c n}{\pi n} e^{-j\omega n} \tag{4.2.39}$$

Figure 4.2.5 shows the function $X_N(\omega)$ for several values of N. We note that there is a significant oscillatory overshoot at $\omega = \omega_c$, independent of the value of N. As N increases, the oscillations become more rapid, but the size of the ripple remains the same. One can show that as $N \to \infty$, the oscillations converge to the point of the discontinuity at $\omega = \omega_c$, but their amplitude does not go to zero. However, (4.2.34) is satisfied, and therefore $X_N(\omega)$ converges to $X(\omega)$ in the mean-square sense.

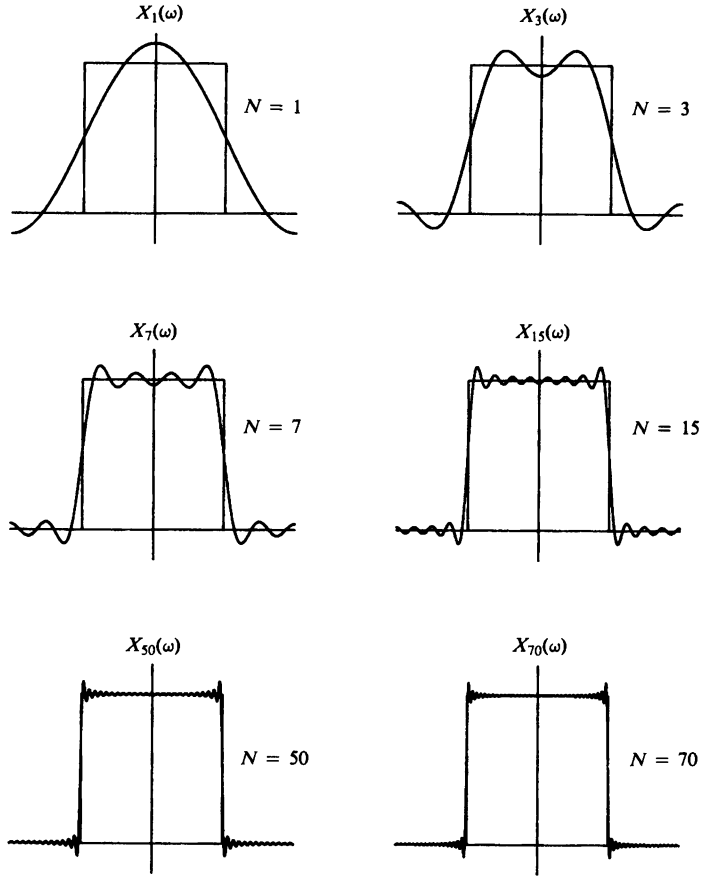


Figure 4.2.5 Illustration of convergence of the Fourier transform and the Gibbs phenomenon at the point of discontinuity.

The oscillatory behavior of the approximation $X_N(\omega)$ to the function $X(\omega)$ at a point of discontinuity of $X(\omega)$ is called the *Gibbs phenomenon.* A similar effect is observed in the truncation of the Fourier series of a continuous-time periodic signal, given by the synthesis equation (4.1.8). For example, the truncation of the Fourier series for the periodic square-wave signal in Example 4.1.1 gives rise to the same oscillatory behavior in the finite-sum approximation of $x(t)$. The Gibbs phenomenon will be encountered again in the design of practical, discrete-time FIR systems considered in Chapter 10.

4.2.5 Energy Density Spectrum of Aperiodic Signals

Recall that the energy of a discrete-time signal $x(n)$ is defined as

$$E_x = \sum_{n=-\infty}^{\infty} |x(n)|^2 \tag{4.2.40}$$

Let us now express the energy E_x in terms of the spectral characteristic $X(\omega)$. First we have

$$E_x = \sum_{n=-\infty}^{\infty} x^*(n)x(n) = \sum_{n=-\infty}^{\infty} x(n)\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} X^*(\omega)e^{-j\omega n}\,d\omega\right]$$

If we interchange the order of integration and summation in the equation above, we obtain

$$\begin{aligned} E_x &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X^*(\omega)\left[\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}\right]d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2\,d\omega \end{aligned}$$

Therefore, the energy relation between $x(n)$ and $X(\omega)$ is

$$E_x = \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2\,d\omega \tag{4.2.41}$$

This is Parseval's relation for discrete-time aperiodic signals with finite energy.

The spectrum $X(\omega)$ is, in general, a complex-valued function of frequency. It may be expressed as

$$X(\omega) = |X(\omega)|e^{j\Theta(\omega)} \tag{4.2.42}$$

where

$$\Theta(\omega) = \measuredangle X(\omega)$$

is the phase spectrum and $|X(\omega)|$ is the magnitude spectrum.

As in the case of continuous-time signals, the quantity

$$S_{xx}(\omega) = |X(\omega)|^2 \tag{4.2.43}$$

represents the distribution of energy as a function of frequency, and it is called the *energy density spectrum* of $x(n)$. Clearly, $S_{xx}(\omega)$ does not contain any phase information.

Suppose now that the signal $x(n)$ is real. Then it easily follows that

$$X^*(\omega) = X(-\omega) \tag{4.2.44}$$

or equivalently,

$$|X(-\omega)| = |X(\omega)|, \qquad \text{(even symmetry)} \tag{4.2.45}$$

and

$$\measuredangle X(-\omega) = -\measuredangle X(\omega), \qquad \text{(odd symmetry)} \tag{4.2.46}$$

From (4.2.43) it also follows that

$$S_{xx}(-\omega) = S_{xx}(\omega), \qquad \text{(even symmetry)} \tag{4.2.47}$$

From these symmetry properties we conclude that the frequency range of real discrete-time signals can be limited further to the range $0 \leq \omega \leq \pi$ (i.e., one-half of the period). Indeed, if we know $X(\omega)$ in the range $0 \leq \omega \leq \pi$, we can determine it for the range $-\pi \leq \omega < 0$ using the symmetry properties given above. As we have already observed, similar results hold for discrete-time periodic signals. Therefore, the frequency-domain description of a real discrete-time signal is completely specified by its spectrum in the frequency range $0 \leq \omega \leq \pi$.

Usually, we work with the fundamental interval $0 \leq \omega \leq \pi$ or $0 \leq F \leq F_s/2$, expressed in hertz. We sketch more than half a period only when required by the specific application.

EXAMPLE 4.2.3

Determine and sketch the energy density spectrum $S_{xx}(\omega)$ of the signal

$$x(n) = a^n u(n), \qquad -1 < a < 1$$

Solution. Since $|a| < 1$, the sequence $x(n)$ is absolutely summable, as can be verified by applying the geometric summation formula,

$$\sum_{n=-\infty}^{\infty} |x(n)| = \sum_{n=0}^{\infty} |a|^n = \frac{1}{1-|a|} < \infty$$

Hence the Fourier transform of $x(n)$ exists and is obtained by applying (4.2.29). Thus

$$X(\omega) = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \sum_{n=0}^{\infty} (ae^{-j\omega})^n$$

Since $|ae^{-j\omega}| = |a| < 1$, use of the geometric summation formula again yields

$$X(\omega) = \frac{1}{1-ae^{-j\omega}}$$

The energy density spectrum is given by

$$S_{xx}(\omega) = |X(\omega)|2 = X(\omega)X(\omega) = \frac{1}{(1-ae^{-j\omega})(1-ae^{j\omega})}$$

or, equivalently, as

$$S_{xx}(\omega) = \frac{1}{1-2a\cos\omega + a^2}$$

Note that $S_{xx}(-\omega) = S_{xx}(\omega)$ in accordance with (4.2.47).

Figure 4.2.6 shows the signal $x(n)$ and its corresponding spectrum for $a = 0.5$ and $a = -0.5$. Note that for $a = -0.5$ the signal has more rapid variations and as a result its spectrum has stronger high frequencies.

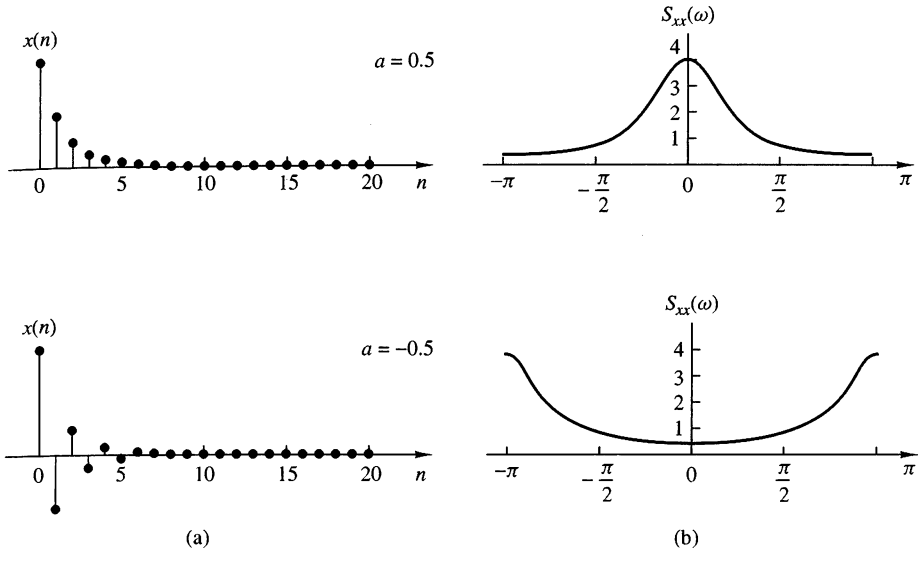


Figure 4.2.6 (a) Sequence $x(n) = (\frac{1}{2})^n u(n)$ and $x(n) = (-\frac{1}{2})^n u(n)$; (b) their energy density spectra.

EXAMPLE 4.2.4

Determine the Fourier transform and the energy density spectrum of the sequence

$$x(n) = \begin{cases} A, & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases} \tag{4.2.48}$$

which is illustrated in Fig 4.2.7.

Solution. Before computing the Fourier transform, we observe that

$$\sum_{n=-\infty}^{\infty} |x(n)| = \sum_{n=0}^{L-1} |A| = L|A| < \infty$$

Hence $x(n)$ is absolutely summable and its Fourier transform exists. Furthermore, we note that $x(n)$ is a finite-energy signal with $E_x = |A|^2 L$.

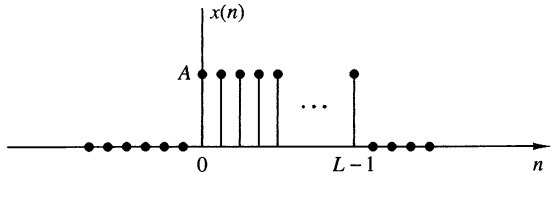


Figure 4.2.7
Discrete-time rectangular pulse.

The Fourier transform of this signal is

$$\begin{aligned} X(\omega) &= \sum_{n=0}^{L-1} Ae^{-j\omega n} \\ &= A\frac{1-e^{-j\omega L}}{1-e^{-j\omega}} \\ &= Ae^{-j(\omega/2)(L-1)}\frac{\sin(\omega L/2)}{\sin(\omega/2)} \end{aligned} \tag{4.2.49}$$

For $\omega = 0$ the transform in (4.2.49) yields $X(0) = AL$, which is easily established by setting $\omega = 0$ in the defining equation for $X(\omega)$, or by using L'Hospital's rule in (4.2.49) to resolve the indeterminate form when $\omega = 0$.

The magnitude and phase spectra of $x(n)$ are

$$|X(\omega)| = \begin{cases} |A|L, & \omega = 0 \\ |A|\left|\dfrac{\sin(\omega L/2)}{\sin(\omega/2)}\right|, & \text{otherwise} \end{cases} \tag{4.2.50}$$

and

$$\measuredangle X(\omega) = \measuredangle A - \frac{\omega}{2}(L-1) + \measuredangle \frac{\sin(\omega L/2)}{\sin(\omega/2)} \tag{4.2.51}$$

where we should remember that the phase of a real quantity is zero if the quantity is positive and π if it is negative.

The spectra $|X(\omega)|$ and $\measuredangle X(\omega)$ are shown in Fig 4.2.8 for the case $A = 1$ and $L = 5$. The energy density spectrum is simply the square of the expression given in (4.2.50).

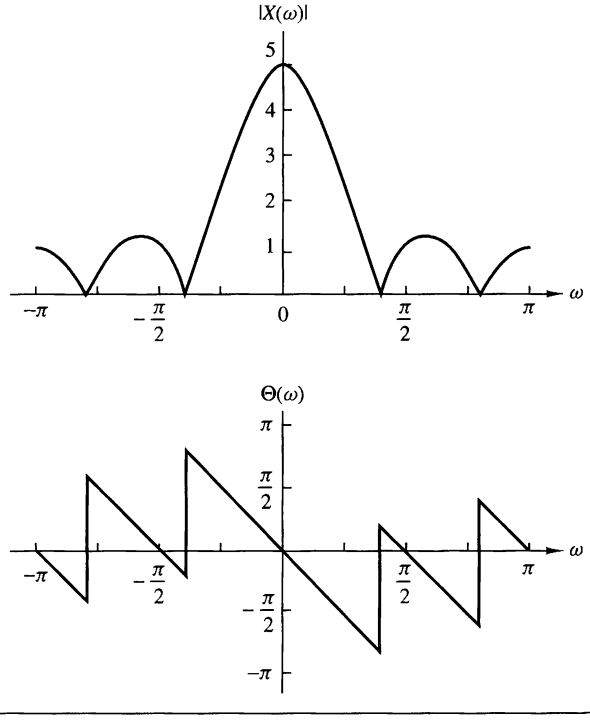


Figure 4.2.8
Magnitude and phase of Fourier transform of the discrete-time rectangular pulse in Fig 4.2.7.

There is an interesting relationship between the Fourier transform of the constant amplitude pulse in Example 4.2.4 and the periodic rectangular wave considered in Example 4.2.2. If we evaluate the Fourier transform as given in (4.2.49) at a set of equally spaced (harmonically related) frequencies

$$\omega_k = \frac{2\pi}{N}k, \qquad k = 0, 1, \ldots, N-1$$

we obtain

$$X\left(\frac{2\pi}{N}k\right) = Ae^{-j(\pi/N)k(L-1)}\frac{\sin[(\pi/N)kL]}{\sin[(\pi/N)k]} \tag{4.2.52}$$

If we compare this result with the expression for the Fourier series coefficients given in (4.2.21) for the periodic rectangular wave, we find that

$$X\left(\frac{2\pi}{N}k\right) = Nc_k, \qquad k = 0, 1, \ldots, N-1 \tag{4.2.53}$$

To elaborate, we have established that the Fourier transform of the rectangular pulse, which is identical with a single period of the periodic rectangular pulse train, evaluated at the frequencies $\omega = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, which are identical to the harmonically related frequency components used in the Fourier series representation of the periodic signal, is simply a multiple of the Fourier coefficients $\{c_k\}$ at the corresponding frequencies.

The relationship given in (4.2.53) for the Fourier transform of the rectangular pulse evaluated at $\omega = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, and the Fourier coefficients of the corresponding periodic signal, is not only true for these two signals but, in fact, holds in general. This relationship is developed further in Chapter 7.

4.2.6 Relationship of the Fourier Transform to the z-Transform

The z-transform of a sequence $x(n)$ is defined as

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n}, \qquad \text{ROC: } r_2 < |z| < r_1 \tag{4.2.54}$$

where $r_2 < |z| < r_1$ is the region of convergence of $X(z)$. Let us express the complex variable z in polar form as

$$z = re^{j\omega} \tag{4.2.55}$$

where $r = |z|$ and $\omega = \measuredangle z$. Then, within the region of convergence of $X(z)$, we can substitute $z = re^{j\omega}$ into (4.2.54). This yields

$$X(z)|_{z=re^{j\omega}} = \sum_{n=-\infty}^{\infty} [x(n)r^{-n}]e^{-j\omega n} \tag{4.2.56}$$

From the relationship in (4.2.56) we note that $X(z)$ can be interpreted as the Fourier transform of the signal sequence $x(n)r^{-n}$. The weighting factor r^{-n} is growing with n if $r < 1$ and decaying if $r > 1$. Alternatively, if $X(z)$ converges for $|z| = 1$, then

$$X(z)|_{z=e^{j\omega}} \equiv X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{4.2.57}$$

Therefore, the Fourier transform can be viewed as the z-transform of the sequence evaluated on the unit circle. If $X(z)$ does not converge in the region $|z| = 1$ [i.e., if the unit circle is not contained in the region of convergence of $X(z)$], the Fourier transform $X(\omega)$ does not exist. Figure 4.2.9 illustrates the relationship between $X(z)$ and $X(\omega)$ for the rectangular sequence in Example 4.2.4, where $A = 1$ and $L = 10$.

We should note that the existence of the z-transform requires that the sequence $\{x(n)r^{-n}\}$ be absolutely summable for some value of r, that is,

$$\sum_{n=-\infty}^{\infty} |x(n)r^{-n}| < \infty \tag{4.2.58}$$

Hence if (4.2.58) converges only for values of $r > r_0 > 1$, the z-transform exists, but the Fourier transform does not exist. This is the case, for example, for causal sequences of the form $x(n) = a^n u(n)$, where $|a| > 1$.

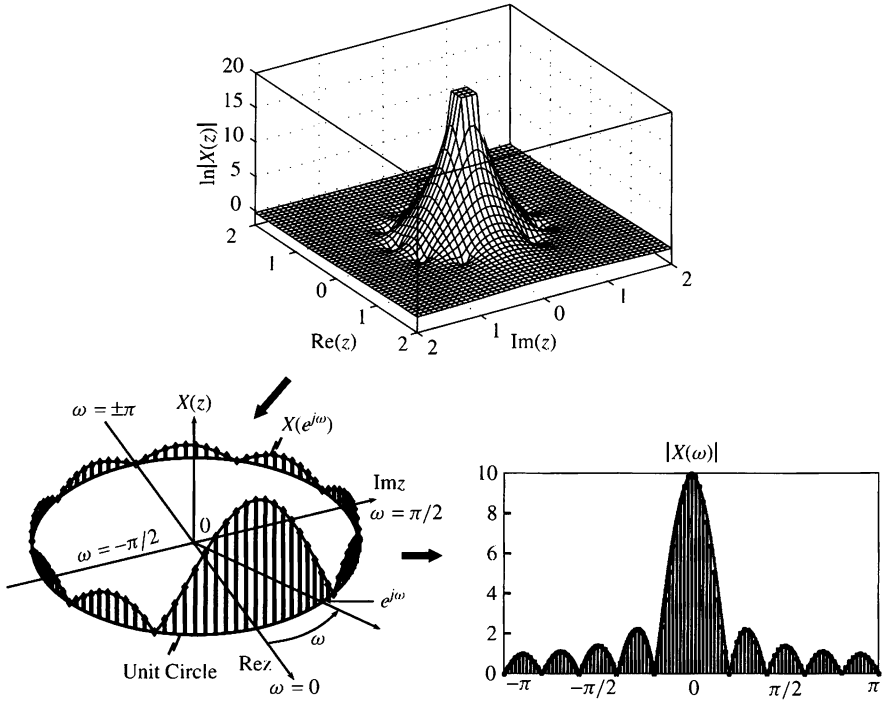


Figure 4.2.9 relationship between $X(z)$ and $X(\omega)$ for the sequence in Example 4.2.4, with $A = 1$ and $L = 10$

There are sequences, however, that do not satisfy the requirement in (4.2.58), for example, the sequence

$$x(n) = \frac{\sin \omega_c n}{\pi n}, \qquad -\infty < n < \infty \tag{4.2.59}$$

This sequence does not have a z-transform. Since it has a finite energy, its Fourier transform converges in the mean-square sense to the discontinuous function $X(\omega)$, defined as

$$X(\omega) = \begin{cases} 1, & |\omega| < \omega_c \\ 0, & \omega_c < |\omega| \leq \pi \end{cases} \tag{4.2.60}$$

In conclusion, the existence of the z-transform requires that (4.2.58) be satisfied for some region in the z-plane. If this region contains the unit circle, the Fourier transform $X(\omega)$ exists. However, the existence of the Fourier transform, which is defined for finite energy signals, does not necessarily ensure the existence of the z-transform.

4.2.7 The Cepstrum

Let us consider a sequence $\{x(n)\}$ having a z-transform $X(z)$. We assume that $\{x(n)\}$ is a stable sequence so that $X(z)$ converges on the unit circle. The *complex cepstrum* of the sequence $\{x(n)\}$ is defined as the sequence $\{c_x(n)\}$, which is the inverse z-transform of $C_x(z)$, where

$$C_x(z) = \ln X(z) \tag{4.2.61}$$

The complex cepstrum exists if $C_x(z)$ converges in the annular region $r_1 < |z| < r_2$, where $0 < r_1 < 1$ and $r_2 > 1$. Within this region of convergence, $C_x(z)$ can be represented by the Laurent series

$$C_x(z) = \ln X(z) = \sum_{n=-\infty}^{\infty} c_x(n) z^{-n} \tag{4.2.62}$$

where

$$c_x(n) = \frac{1}{2\pi j} \int_C \ln X(z) z^{n-1}\, dz \tag{4.2.63}$$

C is a closed contour about the origin and lies within the region of convergence. Clearly, if $C_x(z)$ can be represented as in (4.2.62), the complex cepstrum sequence $\{c_x(n)\}$ is stable. Furthermore, if the complex cepstrum exists, $C_x(z)$ converges on the unit circle and hence we have

$$C_x(\omega) = \ln X(\omega) = \sum_{n=-\infty}^{\infty} c_x(n) e^{-j\omega n} \tag{4.2.64}$$

where $\{c_x(n)\}$ is the sequence obtained from the inverse Fourier transform of $\ln X(\omega)$, that is,

$$c_x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln X(\omega) e^{j\omega n}\, d\omega \tag{4.2.65}$$

If we express $X(\omega)$ in terms of its magnitude and phase, say

$$X(\omega) = |X(\omega)|e^{j\theta(\omega)} \tag{4.2.66}$$

then

$$\ln X(\omega) = \ln |X(\omega)| + j\theta(\omega) \tag{4.2.67}$$

By substituting (4.2.67) into (4.2.65), we obtain the complex cepstrum in the form

$$c_x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} [\ln |X(\omega)| + j\theta(\omega)]e^{j\omega n}\, d\omega \tag{4.2.68}$$

We can separate the inverse Fourier transform in (4.2.68) into the inverse Fourier transforms of $\ln |X(\omega)|$ and $\theta(\omega)$:

$$c_m(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln |X(\omega)|e^{j\omega n}\, d\omega \tag{4.2.69}$$

$$c_\theta(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \theta(\omega)e^{j\omega n}\, d\omega \tag{4.2.70}$$

In some applications, such as speech signal processing, only the component $c_m(n)$ is computed. In such a case the phase of $X(\omega)$ is ignored. Therefore, the sequence $\{x(n)\}$ cannot be recovered from $\{c_m(n)\}$. That is, the transformation from $\{x(n)\}$ to $\{c_m(n)\}$ is not invertible.

In speech signal processing, the (real) cepstrum has been used to separate and thus to estimate the spectral content of the speech from the pitch frequency of the speech. The complex cepstrum is used in practice to separate signals that are convolved. The process of separating two convolved signals is called *deconvolution* and the use of the complex cepstrum to perform the separation is called *homomorphic deconvolution*. This topic is discussed in Section 5.5.4.

4.2.8 The Fourier Transform of Signals with Poles on the Unit Circle

As was shown in Section 4.2.6, the Fourier transform of a sequence $x(n)$ can be determined by evaluating its z-transform $X(z)$ on the unit circle, provided that the unit circle lies within the region of convergence of $X(z)$. Otherwise, the Fourier transform does not exist.

There are some aperiodic sequences that are neither absolutely summable nor square summable. Hence their Fourier transforms do not exist. One such sequence is the unit step sequence, which has the z-transform

$$X(z) = \frac{1}{1 - z^{-1}}$$

Another such sequence is the causal sinusoidal signal sequence $x(n) = (\cos \omega_0 n)u(n)$. This sequence has the z-transform

$$X(z) = \frac{1 - z^{-1} \cos \omega_0}{1 - 2z^{-1} \cos \omega_0 + z^{-2}}$$

Note that both of these sequences have poles on the unit circle.

For sequences such as these two examples, it is sometimes useful to extend the Fourier transform representation. This can be accomplished, in a mathematically rigorous way, by allowing the Fourier transform to contain impulses at certain frequencies corresponding to the location of the poles of $X(z)$ that lie on the unit circle. The impulses are functions of the continuous frequency variable ω and have infinite amplitude, zero width, and unit area. An impulse can be viewed as the limiting form of a rectangular pulse of height $1/a$ and width a, in the limit as $a \to 0$. Thus, by allowing impulses in the spectrum of a signal, it is possible to extend the Fourier transform representation to some signal sequences that are neither absolutely summable nor square summable.

The following example illustrates the extension of the Fourier transform representation for three sequences.

EXAMPLE 4.2.5

Determine the Fourier transform of the following signals.

(a) $x_1(n) = u(n)$

(b) $x_2(n) = (-1)^n u(n)$

(c) $x_3(n) = (\cos \omega_0 n) u(n)$

by evaluating their z-transforms on the unit circle.

Solution.

(a) From Table 3.3 we find that

$$X_1(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}, \qquad \text{ROC: } |z| > 1$$

$X_1(z)$ has a pole, $p_1 = 1$, on the unit circle, but converges for $|z| > 1$.

If we evaluate $X_1(z)$ on the unit circle, except at $z = 1$, we obtain

$$X_1(\omega) = \frac{e^{j\omega/2}}{2j \sin(\omega/2)} = \frac{1}{2 \sin(\omega/2)} e^{j(\omega - \pi/2)}, \qquad \omega \neq 2\pi k, \quad k = 0, 1, \ldots$$

At $\omega = 0$ and multiples of 2π, $X_1(\omega)$ contains impulses of area π.

Hence the presence of a pole at $z = 1$ (i.e., at $\omega = 0$) creates a problem only when we want to compute $|X_1(\omega)|$ at $\omega = 0$, because $|X_1(\omega)| \to \infty$ as $\omega \to 0$. For any other value of ω, $X_1(\omega)$ is finite (i.e., well behaved). Although at first glance one might expect the signal to have zero-frequency components at all frequencies except at $\omega = 0$, this is not the case. This happens because the signal $x_1(n)$ is not a constant for all $-\infty < n < \infty$. Instead, it is *turned on* at $n = 0$. This abrupt jump creates all frequency components existing in the range $0 < \omega \leq \pi$. Generally, all signals which start at a finite time have nonzero-frequency components everywhere in the frequency axis from zero up to the folding frequency.

(b) From Table 3.3 we find that the z-transform of $a^n u(n)$ with $a = -1$ reduces to

$$X_2(z) = \frac{1}{1 + z^{-1}} = \frac{z}{z + 1}, \qquad \text{ROC: } |z| > 1$$

which has a pole at $z = -1 = e^{j\pi}$. The Fourier transform evaluated at frequencies other than $\omega = \pi$ and multiples of 2π is

$$X_2(\omega) = \frac{e^{j\omega/2}}{2\cos(\omega/2)}, \qquad \omega \neq 2\pi(k + \frac{1}{2}), \quad k = 0, 1, \ldots$$

In this case the impulse occurs at $\omega = \pi + 2\pi k$.

Hence the magnitude is

$$|X_2(\omega)| = \frac{1}{2|\cos(\omega/2)|}, \qquad \omega \neq 2\pi k + \pi, \quad k = 0, 1, \ldots$$

and the phase is

$$\measuredangle X_2(\omega) = \begin{cases} \frac{\omega}{2}, & \text{if } \cos\frac{\omega}{2} \geq 0 \\ \frac{\omega}{2} + \pi, & \text{if } \cos\frac{\omega}{2} < 0 \end{cases}$$

Note that due to the presence of the pole at $a = -1$ (i.e., at frequency $\omega = \pi$), the magnitude of the Fourier transform becomes infinite. Now $|X(\omega)| \to \infty$ as $\omega \to \pi$. We observe that $(-1)^n u(n) = (\cos \pi n)u(n)$, which is the fastest possible oscillating signal in discrete time.

(c) From the discussion above, it follows that $X_3(\omega)$ is infinite at the frequency component $\omega = \omega_0$. Indeed, from Table 3.3, we find that

$$x_3(n) = (\cos \omega_0 n)u(n) \overset{z}{\longleftrightarrow} X_3(z) = \frac{1 - z^{-1}\cos\omega_0}{1 - 2z^{-1}\cos\omega_0 + z^{-2}}, \qquad \text{ROC: } |z| > 1$$

The Fourier transform is

$$X_3(\omega) = \frac{1 - e^{-j\omega}\cos\omega_0}{(1 - e^{-j(\omega-\omega_0)})(1 - e^{j(\omega+\omega_0)})}, \qquad \omega \neq \pm\omega_0 + 2\pi k, \quad k = 0, 1, \ldots$$

The magnitude of $X_3(\omega)$ is given by

$$|X_3(\omega)| = \frac{|1 - e^{-j\omega}\cos\omega_0|}{|1 - e^{-j(\omega-\omega_0)}||1 - e^{-j(\omega+\omega_0)}|}, \qquad \omega \neq \pm\omega_0 + 2\pi k, \quad k = 0, 1, \ldots$$

Now if $\omega = -\omega_0$ or $\omega = \omega_0$, $|X_3(\omega)|$ becomes infinite. For all other frequencies, the Fourier transform is well behaved.

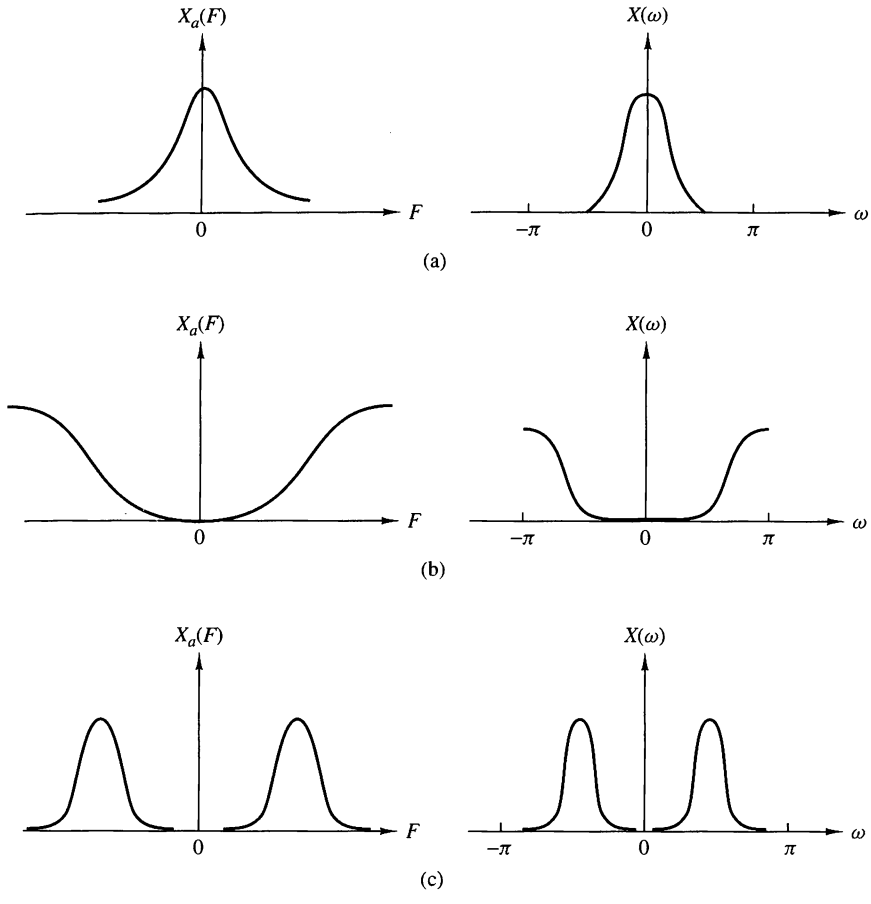


Figure 4.2.10 (a) Low-frequency, (b) high-frequency, and (c) medium-frequency signals.

4.2.9 Frequency-Domain Classification of Signals: The Concept of Bandwidth

Just as we have classified signals according to their time-domain characteristics, it is also desirable to classify signals according to their frequency-domain characteristics. It is common practice to classify signals in rather broad terms according to their frequency content.

In particular, if a power signal (or energy signal) has its power density spectrum (or its energy density spectrum) concentrated about zero frequency, such a signal is called a *low-frequency signal.* Figure 4.2.10(a) illustrates the spectral characteristics of such a signal. On the other hand, if the signal power density spectrum (or the energy density spectrum) is concentrated at high frequencies, the signal is called a *high-frequency signal.* Such a signal spectrum is illustrated in Fig 4.2.10(b). A signal having a power density spectrum (or an energy density spectrum) concentrated somewhere in the broad frequency range between low frequencies and high frequencies

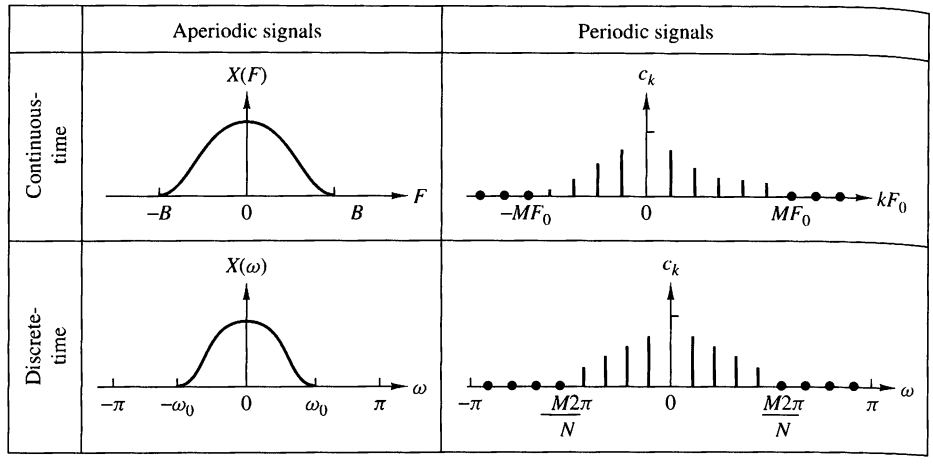


Figure 4.2.11 Some examples of bandlimited signals.

is called a *medium-frequency signal* or a *bandpass signal.*Figure 4.2.10(c) illustrates such a signal spectrum.

In addition to this relatively broad frequency-domain classification of signals, it is often desirable to express quantitatively the range of frequencies over which the power or energy density spectrum is concentrated. This quantitative measure is called the *bandwidth* of a signal. For example, suppose that a continuous-time signal has 95% of its power (or energy) density spectrum concentrated in the frequency range $F_1 \leq F \leq F_2$. Then the 95% bandwidth of the signal is $F_2 - F_1$. In a similar manner, we may define the 75% or 90% or 99% bandwidth of the signal.

In the case of a bandpass signal, the term *narrowband* is used to describe the signal if its bandwidth $F_2 - F_1$ is much smaller (say, by a factor of 10 or more) than the median frequency $(F_2 + F_1)/2$. Otherwise, the signal is called *wideband.*

We shall say that a signal is *bandlimited* if its spectrum is zero outside the frequency range $|F| \geq B$. For example, a continuous-time finite-energy signal $x(t)$ is bandlimited if its Fourier transform $X(F) = 0$ for $|F| > B$. A discrete-time finite-energy signal $x(n)$ is said to be *(periodically) bandlimited* if

$$|X(\omega)| = 0, \qquad \text{for } \omega_0 < |\omega| < \pi$$

Similarly, a periodic continuous-time signal $x_p(t)$ is periodically bandlimited if its Fourier coefficients $c_k = 0$ for $|k| > M$, where M is some positive integer. A periodic discrete-time signal with fundamental period N is periodically bandlimited if the Fourier coefficients $c_k = 0$ for $k_0 < |k| < N$. Figure 4.2.11 illustrates the four types of bandlimited signals.

By exploiting the duality between the frequency domain and the time domain, we can provide similar means for characterizing signals in the time domain. In particular, a signal $x(t)$ will be called *time-limited* if

$$x(t) = 0, \qquad |t| > \tau$$

If the signal is periodic with period T_p, it will be called *periodically time-limited* if

$$x_p(t) = 0, \qquad \tau < |t| < T_p/2$$

If we have a discrete-time signal $x(n)$ of finite duration, that is,

$$x(n) = 0, \qquad |n| > N$$

it is also called time-limited. When the signal is periodic with fundamental period N, it is said to be periodically time-limited if

$$x(n) = 0, \qquad n_0 < |n| < N$$

We state, without proof, that no *signal can be time-limited and bandlimited simultaneously*. Furthermore, a reciprocal relationship exists between the time duration and the frequency duration of a signal. To elaborate, if we have a short-duration rectangular pulse in the time domain, its spectrum has a width that is inversely proportional to the duration of the time-domain pulse. The narrower the pulse becomes in the time domain, the larger the bandwidth of the signal becomes. Consequently, the product of the time duration and the bandwidth of a signal cannot be made arbitrarily small. A short-duration signal has a large bandwidth and a small bandwidth signal has a long duration. Thus, for any signal, the time–bandwidth product is fixed and cannot be made arbitrarily small.

Finally, we note that we have discussed frequency analysis methods for periodic and aperiodic signals with finite energy. However, there is a family of deterministic aperiodic signals with finite power. These signals consist of a linear superposition of complex exponentials with nonharmonically related frequencies, that is,

$$x(n) = \sum_{k=1}^{M} A_k e^{j\omega_k n}$$

where $\omega_1, \omega_2, \ldots, \omega_M$ are nonharmonically related. These signals have discrete spectra but the distances among the lines are nonharmonically related. Signals with discrete nonharmonic spectra are sometimes called quasi-periodic.

4.2.10 The Frequency Ranges of Some Natural Signals

The frequency analysis tools that we have developed in this chapter are usually applied to a variety of signals that are encountered in practice (e.g., seismic, biological, and electromagnetic signals). In general, the frequency analysis is performed for the purpose of extracting information from the observed signal. For example, in the case of biological signals, such as an ECG signal, the analytical tools are used to extract information relevant for diagnostic purposes. In the case of seismic signals, we may be interested in detecting the presence of a nuclear explosion or in determining the characteristics and location of an earthquake. An electromagnetic signal, such as a radar signal reflected from an airplane, contains information on the position of the

plane and its radial velocity. These parameters can be estimated from observation of the received radar signal.

In processing any signal for the purpose of measuring parameters or extracting other types of information, one must know approximately the range of frequencies contained by the signal. For reference, Tables 4.1, 4.2, and 4.3 give approximate limits in the frequency domain for biological, seismic, and electromagnetic signals.

4.3 Frequency-Domain and Time-Domain Signal Properties

In the previous sections of the chapter we have introduced several methods for the frequency analysis of signals. Several methods were necessary to accommodate the different types of signals. To summarize, the following frequency analysis tools have been introduced:

1. The Fourier series for continuous-time periodic signals.
2. The Fourier transform for continuous-time aperiodic signals.
3. The Fourier series for discrete-time periodic signals.
4. The Fourier transform for discrete-time aperiodic signals.

TABLE 4.1 Frequency Ranges of Some Biological Signals

Type of Signal	Frequency Range (Hz)
Electroretinogram[a]	0–20
Electronystagmogram[b]	0–20
Pneumogram[c]	0–40
Electrocardiogram (ECG)	0–100
Electroencephalogram (EEG)	0–100
Electromyogram[d]	10–200
Sphygmomanogram[e]	0–200
Speech	100–4000

[a] A graphic recording of retina characteristics.
[b] A graphic recording of involuntary movement of the eyes.
[c] A graphic recording of respiratory activity.
[d] A graphic recording of muscular action, such as muscular contraction.
[e] A recording of blood pressure.

TABLE 4.2 Frequency Ranges of Some Seismic Signals

Type of Signal	Frequency Range (Hz)
Wind noise	100–1000
Seismic exploration signals	10–100
Earthquake and nuclear explosion signals	0.01–10
Seismic noise	0.1–1

TABLE 4.3 Frequency Ranges of Electromagnetic Signals

Type of Signal	Wavelength (m)	Frequency Range (Hz)
Radio broadcast	10^4–10^2	3×10^4–3×10^6
Shortwave radio signals	10^2–10^{-2}	3×10^6–3×10^{10}
Radar, satellite communications, space communications, common-carrier microwave	1–10^{-2}	3×10^8–3×10^{10}
Infrared	10^{-3}–10^{-6}	3×10^{11}–3×10^{14}
Visible light	3.9×10^{-7}–8.1×10^{-7}	3.7×10^{14}–7.7×10^{14}
Ultraviolet	10^{-7}–10^{-8}	3×10^{15}–3×10^{16}
Gamma rays and X rays	10^{-9}–10^{-10}	3×10^{17}–3×10^{18}

Figure 4.3.1 summarizes the analysis and synthesis formulas for these types of signals.

As we have already indicated several times, there are two time-domain characteristics that determine the type of signal spectrum we obtain. These are whether the time variable is continuous or discrete, and whether the signal is periodic or aperiodic. Let us briefly summarize the results of the previous sections.

Continuous-time signals have aperiodic spectra A close inspection of the Fourier series and Fourier transform analysis formulas for continuous-time signals does not reveal any kind of periodicity in the spectral domain. This lack of periodicity is a consequence of the fact that the complex exponential $\exp(j2\pi Ft)$ is a function of the continuous variable t, and hence it is not periodic in F. Thus the frequency range of continuous-time signals extends from $F = 0$ to $F = \infty$.

Discrete-time signals have periodic spectra Indeed, both the Fourier series and the Fourier transform for discrete-time signals are periodic with period $\omega = 2\pi$. As a result of this periodicity, the frequency range of discrete-time signals is finite and extends from $\omega = -\pi$ to $\omega = \pi$ radians, where $\omega = \pi$ corresponds to the highest possible rate of oscillation.

Periodic signals have discrete spectra As we have observed, periodic signals are described by means of Fourier series. The Fourier series coefficients provide the "lines" that constitute the discrete spectrum. The line spacing ΔF or Δf is equal to the inverse of the period T_p or N, respectively, in the time domain. That is, $\Delta F = 1/T_p$ for continuous-time periodic signals and $\Delta f = 1/N$ for discrete-time signals.

Aperiodic finite energy signals have continuous spectra This property is a direct consequence of the fact that both $X(F)$ and $X(\omega)$ are functions of $\exp(j2\pi Ft)$ and $\exp(j\omega n)$, respectively, which are continuous functions of the variables F and ω. The continuity in frequency is necessary to break the harmony and thus create aperiodic signals.

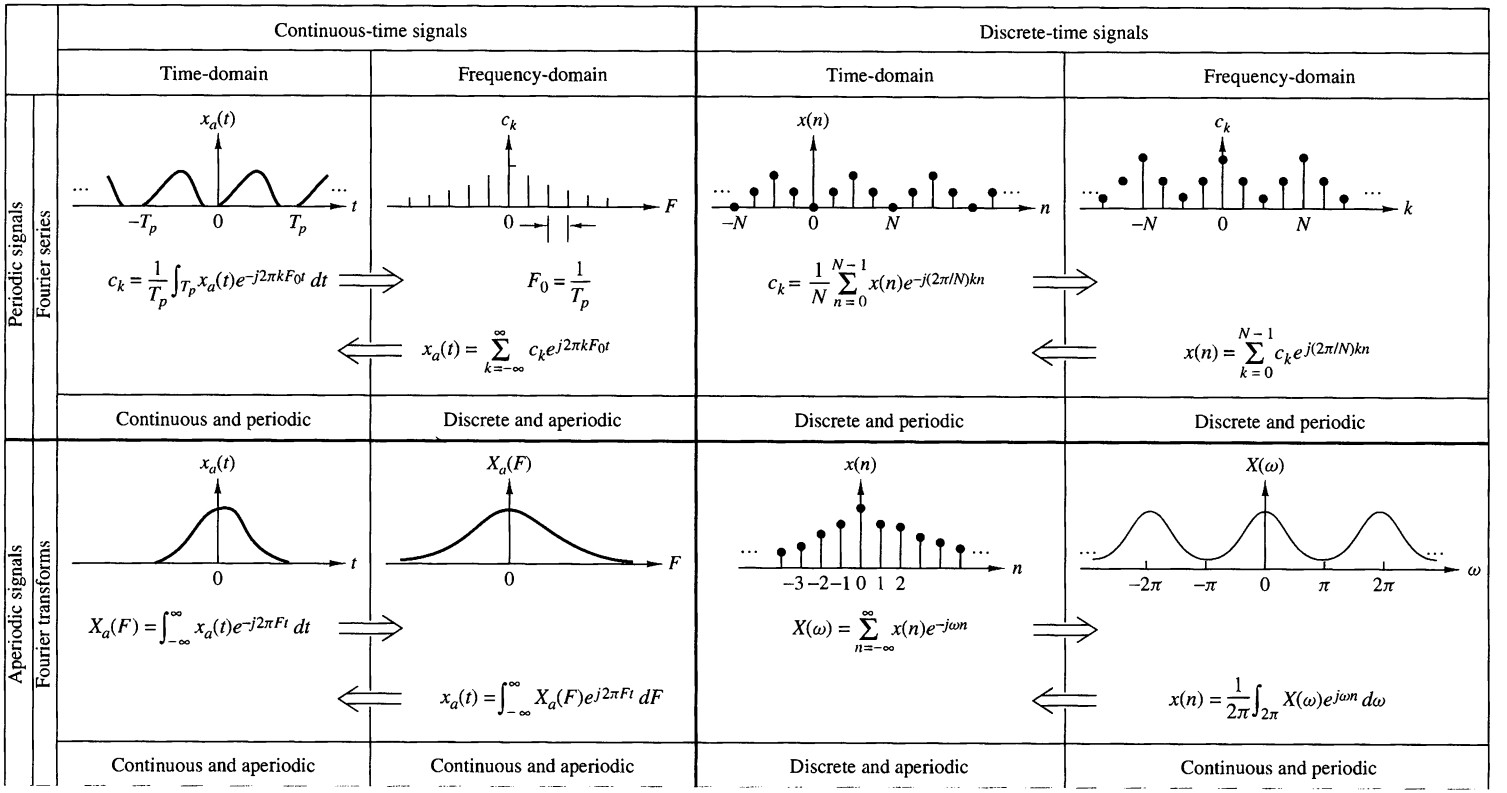


Figure 4.3.1 Summary of analysis and synthesis formulas.

In summary, we can conclude that *periodicity with "period" α in one domain automatically implies discretization with "spacing" of $1/\alpha$ in the other domain, and vice versa.*

If we keep in mind that "period" in the frequency domain means the frequency range, "spacing" in the time domain is the sampling period T, line spacing in the frequency domain is ΔF, then $\alpha = T_p$ implies that $1/\alpha = 1/T_p = \Delta F$, $\alpha = N$ implies that $\Delta f = 1/N$, and $\alpha = F_s$ implies that $T = 1/F_s$.

These time-frequency dualities are apparent from observation of Fig 4.3.1. We stress, however, that the illustrations used in this figure do not correspond to any actual transform pairs. Thus any comparison among them should be avoided.

A careful inspection of Fig 4.3.1 also reveals some mathematical symmetries and dualities among the several frequency analysis relationships. In particular, we observe that there are dualities between the following analysis and synthesis equations:

1. The analysis and synthesis equations of the continuous-time Fourier transform.
2. The analysis and synthesis equations of the discrete-time Fourier series.
3. The analysis equation of the continuous-time Fourier series and the synthesis equation of the discrete-time Fourier transform.
4. The analysis equation of the discrete-time Fourier transform and the synthesis equation of the continuous-time Fourier series.

Note that all dual relations differ only in the sign of the exponent of the corresponding complex exponential. It is interesting to note that this change in sign can be thought of either as a folding of the signal or a folding of the spectrum, since

$$e^{-j2\pi Ft} = e^{j2\pi(-F)t} = e^{j2\pi F(-t)}$$

If we turn our attention now to the spectral density of signals, we recall that we have used the term *energy density spectrum* for characterizing finite-energy aperiodic signals and the term *power density spectrum* for periodic signals. This terminology is consistent with the fact that periodic signals are power signals and aperiodic signals with finite energy are energy signals.

4.4 Properties of the Fourier Transform for Discrete-Time Signals

The Fourier transform for aperiodic finite-energy discrete-time signals described in the preceding section possesses a number of properties that are very useful in reducing the complexity of frequency analysis problems in many practical applications. In this section we develop the important properties of the Fourier transform. Similar properties hold for the Fourier transform of aperiodic finite-energy continuous-time signals.

For convenience, we adopt the notation

$$X(\omega) \equiv F\{x(n)\} = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{4.4.1}$$

for the direct transform (analysis equation) and

$$x(n) \equiv F^{-1}\{X(\omega)\} = \frac{1}{2\pi}\int_{2\pi} X(\omega)e^{j\omega n}\, d\omega \tag{4.4.2}$$

for the inverse transform (synthesis equation). We also refer to $x(n)$ and $X(\omega)$ as a *Fourier transform pair* and denote this relationship with the notation

$$x(n) \overset{F}{\longleftrightarrow} X(\omega) \tag{4.4.3}$$

Recall that $X(\omega)$ is periodic with period 2π. Consequently, any interval of length 2π is sufficient for the specification of the spectrum. Usually, we plot the spectrum in the fundamental interval $[-\pi, \pi]$. We emphasize that all the spectral information contained in the fundamental interval is necessary for the complete description or characterization of the signal. For this reason, the range of integration in (4.4.2) is always 2π, independent of the specific characteristics of the signal within the fundamental interval.

4.4.1 Symmetry Properties of the Fourier Transform

When a signal satisfies some symmetry properties in the time domain, these properties impose some symmetry conditions on its Fourier transform. Exploitation of any symmetry characteristics leads to simpler formulas for both the direct and inverse Fourier transform. A discussion of various symmetry properties and the implications of these properties in the frequency domain is given here.

Suppose that both the signal $x(n)$ and its transform $X(\omega)$ are complex-valued functions. Then they can be expressed in rectangular form as

$$x(n) = x_R(n) + jx_I(n) \tag{4.4.4}$$

$$X(\omega) = X_R(\omega) + jX_I(\omega) \tag{4.4.5}$$

By substituting (4.4.4) and $e^{-j\omega} = \cos\omega - j\sin\omega$ into (4.4.1) and separating the real and imaginary parts, we obtain

$$X_R(\omega) = \sum_{n=-\infty}^{\infty} [x_R(n)\cos\omega n + x_I(n)\sin\omega n] \tag{4.4.6}$$

$$X_I(\omega) = -\sum_{n=-\infty}^{\infty} [x_R(n)\sin\omega n - x_I(n)\cos\omega n] \tag{4.4.7}$$

In a similar manner, by substituting (4.4.5) and $e^{j\omega} = \cos\omega + j\sin\omega$ into (4.4.2), we obtain

$$x_R(n) = \frac{1}{2\pi}\int_{2\pi}[X_R(\omega)\cos\omega n - X_I(\omega)\sin\omega n]\,d\omega \tag{4.4.8}$$

$$x_I(n) = \frac{1}{2\pi}\int_{2\pi}[X_R(\omega)\sin\omega n + X_I(\omega)\cos\omega n]\,d\omega \tag{4.4.9}$$

Now, let us investigate some special cases.

Real signals. If $x(n)$ is real, then $x_R(n) = x(n)$ and $x_I(n) = 0$. Hence (4.4.6) and (4.4.7) reduce to

$$X_R(\omega) = \sum_{n=-\infty}^{\infty} x(n)\cos\omega n \tag{4.4.10}$$

and

$$X_I(\omega) = -\sum_{n=-\infty}^{\infty} x(n)\sin\omega n \tag{4.4.11}$$

Since $\cos(-\omega n) = \cos\omega n$ and $\sin(-\omega n) = -\sin\omega n$, it follows from (4.4.10) and (4.4.11) that

$$X_R(-\omega) = X_R(\omega)\ , \qquad \text{(even)} \tag{4.4.12}$$

$$X_I(-\omega) = -X_I(\omega), \qquad \text{(odd)} \tag{4.4.13}$$

If we combine (4.4.12) and (4.4.13) into a single equation, we have

$$X^*(\omega) = X(-\omega) \tag{4.4.14}$$

In this case we say that the spectrum of a real signal has *Hermitian symmetry.*

With the aid of Fig 4.4.1, we observe that the magnitude and phase spectra for real signals are

$$|X(\omega)| = \sqrt{X_R^2(\omega) + X_I^2(\omega)} \tag{4.4.15}$$

$$\measuredangle X|\omega| = \tan^{-1}\frac{X_I(\omega)}{X_R(\omega)} \tag{4.4.16}$$

As a consequence of (4.4.12) and (4.4.13), the magnitude and phase spectra also possess the symmetry properties

$$|X(\omega)| = |X(-\omega)|\ , \qquad \text{(even)} \tag{4.4.17}$$

$$\measuredangle X(-\omega) = -\measuredangle X(\omega), \qquad \text{(odd)} \tag{4.4.18}$$

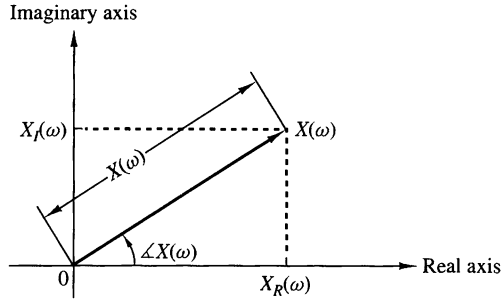


Figure 4.4.1
Magnitude and phase functions.

In the case of the inverse transform of a real-valued signal [i.e., $x(n) = x_R(n)$], (4.4.8) implies that

$$x(n) = \frac{1}{2\pi} \int_{2\pi} [X_R(\omega) \cos \omega n - X_I(\omega) \sin \omega n] \, d\omega \tag{4.4.19}$$

Since both products $X_R(\omega) \cos \omega n$ and $X_I(\omega) \sin \omega n$ are even functions of ω, we have

$$x(n) = \frac{1}{\pi} \int_0^{\pi} [X_R(\omega) \cos \omega n - X_I(\omega) \sin \omega n] \, d\omega \tag{4.4.20}$$

Real and even signals. If $x(n)$ is real and even [i.e., $x(-n) = x(n)$], then $x(n) \cos \omega n$ is even and $x(n) \sin \omega n$ is odd. Hence, from (4.4.10), (4.4.11), and (4.4.20) we obtain

$$X_R(\omega) = x(0) + 2 \sum_{n=1}^{\infty} x(n) \cos \omega n, \qquad \text{(even)} \tag{4.4.21}$$

$$X_I(\omega) = 0 \tag{4.4.22}$$

$$x(n) = \frac{1}{\pi} \int_0^{\pi} X_R(\omega) \cos \omega n \, d\omega \tag{4.4.23}$$

Thus real and even signals possess real-valued spectra, which, in addition, are even functions of the frequency variable ω.

Real and odd signals. If $x(n)$ is real and odd [i.e., $x(-n) = -x(n)$], then $x(n) \cos \omega n$ is odd and $x(n) \sin \omega n$ is even. Consequently, (4.4.10), (4.4.11) and (4.4.20) imply that

$$X_R(\omega) = 0 \tag{4.4.24}$$

$$X_I(\omega) = -2 \sum_{n=1}^{\infty} x(n) \sin \omega n, \qquad \text{(odd)} \tag{4.4.25}$$

$$x(n) = -\frac{1}{\pi} \int_0^{\pi} X_I(\omega) \sin \omega n \, d\omega \tag{4.4.26}$$

Thus real-valued odd signals possess purely imaginary-valued spectral characteristics, which, in addition, are odd functions of the frequency variable ω.

Purely imaginary signals. In this case $x_R(n) = 0$ and $x(n) = jx_I(n)$. Thus (4.4.6), (4.4.7), and (4.4.9) reduce to

$$X_R(\omega) = \sum_{n=-\infty}^{\infty} x_I(n) \sin \omega n, \qquad \text{(odd)} \tag{4.4.27}$$

$$X_I(\omega) = \sum_{n=-\infty}^{\infty} x_I(n) \cos \omega n, \qquad \text{(even)} \tag{4.4.28}$$

$$x_I(n) = \frac{1}{\pi} \int_0^{\pi} [X_R(\omega) \sin \omega n + X_I(\omega) \cos \omega n] \, d\omega \tag{4.4.29}$$

If $x_I(n)$ is odd [i.e., $x_I(-n) = -x_I(n)$], then

$$X_R(\omega) = 2 \sum_{n=1}^{\infty} x_I(n) \sin \omega n, \qquad \text{(odd)} \tag{4.4.30}$$

$$X_I(\omega) = 0 \tag{4.4.31}$$

$$x_I(n) = \frac{1}{\pi} \int_0^{\pi} X_R(\omega) \sin \omega n \, d\omega \tag{4.4.32}$$

Similarly, if $x_I(n)$ is even [i.e., $x_I(-n) = x_I(n)$], we have

$$X_R(\omega) = 0 \tag{4.4.33}$$

$$X_I(\omega) = x_I(0) + 2 \sum_{n=1}^{\infty} x_I(n) \cos \omega n, \qquad \text{(even)} \tag{4.4.34}$$

$$x_I(n) = \frac{1}{\pi} \int_0^{\pi} X_I(\omega) \cos \omega n \, d\omega \tag{4.4.35}$$

An arbitrary, possibly complex-valued signal $x(n)$ can be decomposed as

$$x(n) = x_R(n) + jx_I(n) = x_R^e(n) + x_R^o(n) + j[x_I^e(n) + x_I^o(n)] = x_e(n) + x_o(n) \tag{4.4.36}$$

where, by definition,

$$x_e(n) = x_R^e(n) + jx_I^e(n) = \frac{1}{2}[x(n) + x^*(-n)]$$

$$x_o(n) = x_R^o(n) + jx_I^o(n) = \frac{1}{2}[x(n) - x^*(-n)]$$

The superscripts e and o denote the even and odd signal components, respectively. We note that $x_e(n) = x_e(-n)$ and $x_o(-n) = -x_o(n)$. From (4.4.36) and the Fourier transform properties established above, we obtain the following relationships:

$$\begin{array}{ccccccc} x(n) = [x_R^e(n) + jx_I^e(n)] & + & [x_R^o(n) + jx_I^o(n)] & = & x_e(n) & + & x_o(n) \\ \updownarrow \quad \updownarrow & & \searrow \quad \swarrow & & \updownarrow & & \updownarrow \\ X(\omega) = [X_R^e(\omega) + jX_I^e(\omega)] & + & [X_R^o(\omega) - jX_I^o(\omega)] & = & X_e(\omega) & + & X_o(\omega) \end{array} \tag{4.4.37}$$

These symmetry properties of the Fourier transform are summarized in Table 4.4 and in Fig 4.4.2. They are often used to simplify Fourier transform calculations in practice.

TABLE 4.4 Symmetry Properties of the Discrete-Time Fourier Transform

Sequence	DTFT
$x(n)$	$X(\omega)$
$x^*(n)$	$X^*(-\omega)$
$x^*(-n)$	$X^*(\omega)$
$x_R(n)$	$X_e(\omega) = \frac{1}{2}[X(\omega) + X^*(-\omega)]$
$jx_I(n)$	$X_o(\omega) = \frac{1}{2}[X(\omega) - X^*(-\omega)]$
$x_e(n) = \frac{1}{2}[x(n) + x^*(-n)]$	$X_R(\omega)$
$x_o(n) = \frac{1}{2}[x(n) - x^*(-n)]$	$jX_I(\omega)$
Real Signals	
	$X(\omega) = X^*(-\omega)$
Any real signal	$X_R(\omega) = X_R(-\omega)$
$x(n)$	$X_I(\omega) = -X_I(-\omega)$
	$\lvert X(\omega)\rvert = \lvert X(-\omega)\rvert$
	$\measuredangle X(\omega) = -\measuredangle X(-\omega)$
$x_e(n) = \frac{1}{2}[x(n) + x(-n)]$	$X_R(\omega)$
(real and even)	(real and even)
$x_o(n) = \frac{1}{2}[x(n) - x(-n)]$	$jX_I(\omega)$
(real and odd)	(imaginary and odd)

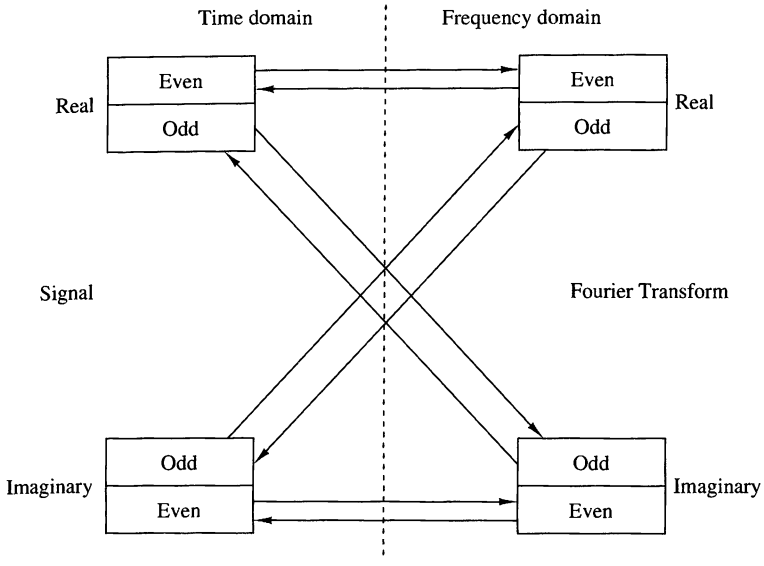


Figure 4.4.2 Summary of symmetry properties for the Fourier transform.

EXAMPLE 4.4.1

Determine and sketch $X_R(\omega)$, $X_I(\omega)$, $|X(\omega)|$, and $\measuredangle X(\omega)$ for the Fourier transform

$$X(\omega) = \frac{1}{1 - ae^{-j\omega}}, \qquad -1 < a < 1 \tag{4.4.38}$$

Solution. By multiplying both the numerator and denominator of (4.4.38) by the complex conjugate of the denominator, we obtain

$$X(\omega) = \frac{1 - ae^{j\omega}}{(1 - ae^{-j\omega})(1 - ae^{j\omega})} = \frac{1 - a\cos\omega - ja\sin\omega}{1 - 2a\cos\omega + a^2}$$

This expression can be subdivided into real and imaginary parts. Thus we obtain

$$X_R(\omega) = \frac{1 - a\cos\omega}{1 - 2a\cos\omega + a^2}$$

$$X_I(\omega) = -\frac{a\sin\omega}{1 - 2a\cos\omega + a^2}$$

Substitution of the last two equations into (4.4.15) and (4.4.16) yields the magnitude and phase spectra as

$$|X(\omega)| = \frac{1}{\sqrt{1 - 2a\cos\omega + a^2}} \tag{4.4.39}$$

and

$$\measuredangle X(\omega) = -\tan^{-1}\frac{a\sin\omega}{1 - a\cos\omega} \tag{4.4.40}$$

Figures 4.4.3 and 4.4.4 show the graphical representation of these spectra for $a = 0.8$. The reader can easily verify that as expected, all symmetry properties for the spectra of real signals apply to this case.

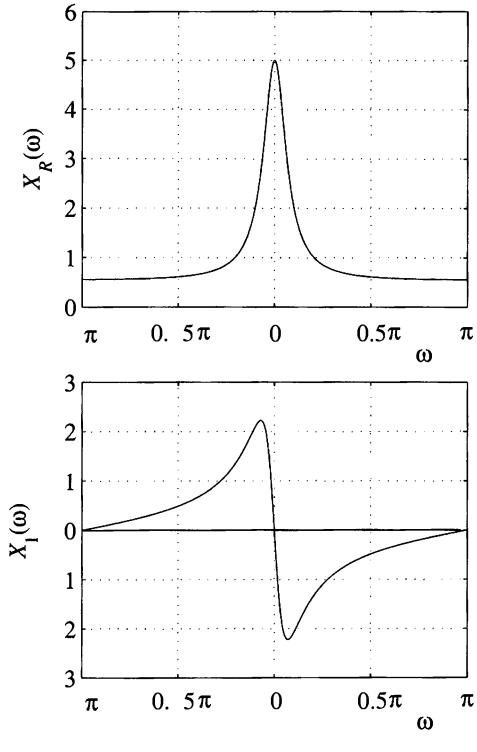


Figure 4.4.3
Graph of $X_R(\omega)$ and $X_I(\omega)$ for the transform in Example 4.4.1.

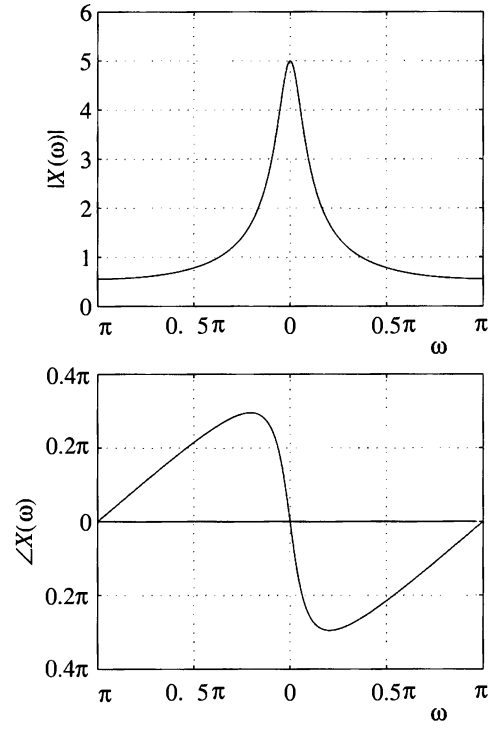


Figure 4.4.4
Magnitude and phase spectra of the transform in Example 4.4.1.

EXAMPLE 4.4.2

Determine the Fourier transform of the signal

$$x(n) = \begin{cases} A, & -M \le n \le M \\ 0, & \text{elsewhere} \end{cases} \tag{4.4.41}$$

Solution. Clearly, $x(-n) = x(n)$. Thus $x(n)$ is a real and even signal. From (4.4.21) we obtain

$$X(\omega) = X_R(\omega) = A\left(1 + 2\sum_{n=1}^{M} \cos \omega n\right)$$

If we use the identity given in Problem 4.13, we obtain the simpler form

$$X(\omega) = A\frac{\sin(M + \frac{1}{2})\omega}{\sin(\omega/2)}$$

Since $X(\omega)$ is real, the magnitude and phase spectra are given by

$$|X(\omega)| = \left|A\frac{\sin(M + \frac{1}{2})\omega}{\sin(\omega/2)}\right| \tag{4.4.42}$$

and

$$\measuredangle X(\omega) = \begin{cases} 0, & \text{if } X(\omega) > 0 \\ \pi, & \text{if } X(\omega) < 0 \end{cases} \tag{4.4.43}$$

Figure 4.4.5 shows the graphs for $X(\omega)$.

4.4.2 Fourier Transform Theorems and Properties

In this section we introduce several Fourier transform theorems and illustrate their use in practice by examples.

Linearity. If

$$x_1(n) \overset{F}{\longleftrightarrow} X_1(\omega)$$

and

$$x_2(n) \overset{F}{\longleftrightarrow} X_2(\omega)$$

then

$$a_1x_1(n) + a_2x_2(n) \overset{F}{\longleftrightarrow} a_1X_1(\omega) + a_2X_2(\omega) \tag{4.4.44}$$

Simply stated, the Fourier transformation, viewed as an operation on a signal $x(n)$, is a linear transformation. Thus the Fourier transform of a linear combination of two or more signals is equal to the same linear combination of the Fourier transforms of the individual signals. This property is easily proved by using (4.4.1). The linearity property makes the Fourier transform suitable for the study of linear systems.

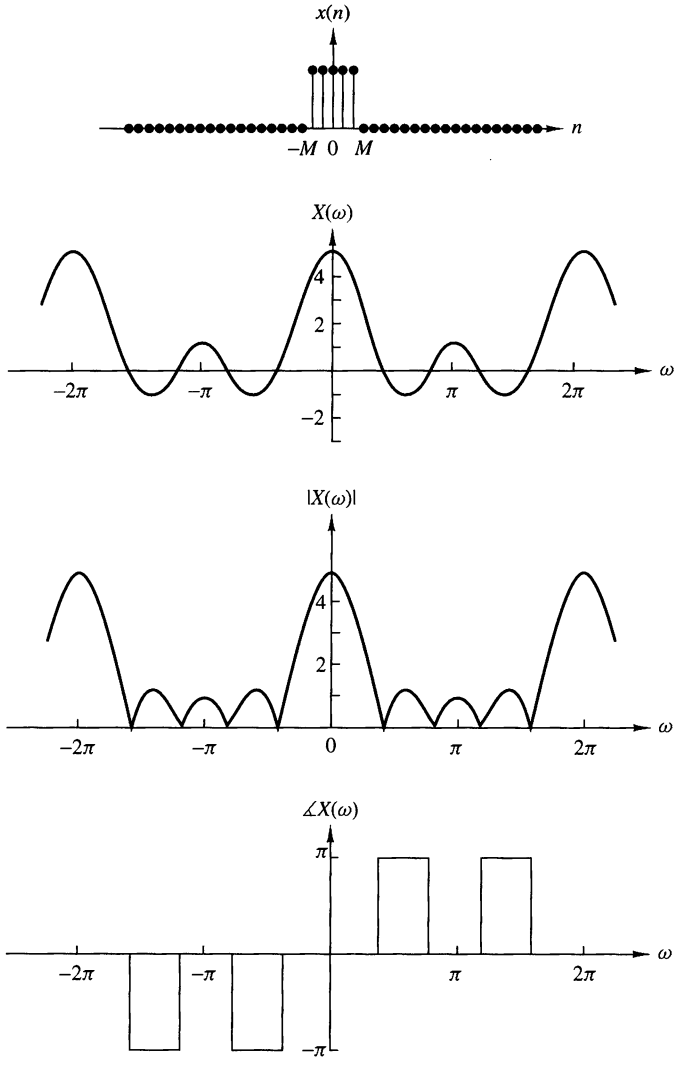


Figure 4.4.5 Spectral characteristics of rectangular pulse in Example 4.4.2.

EXAMPLE 4.4.3

Determine the Fourier transform of the signal

$$x(n) = a^{|n|}, \qquad -1 < a < 1 \tag{4.4.45}$$

Solution. First, we observe that $x(n)$ can be expressed as

$$x(n) = x_1(n) + x_2(n)$$

where

$$x_1(n) = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

and

$$x_2(n) = \begin{cases} a^{-n}, & n < 0 \\ 0, & n \geq 0 \end{cases}$$

Beginning with the definition of the Fourier transform in (4.4.1), we have

$$X_1(\omega) = \sum_{n=-\infty}^{\infty} x_1(n)e^{-j\omega n} = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \sum_{n=0}^{\infty} (ae^{-j\omega})^n$$

The summation is a geometric series that converges to

$$X_1(\omega) = \frac{1}{1 - ae^{-j\omega}}$$

provided that

$$|ae^{-j\omega}| = |a| \cdot |e^{-j\omega}| = |a| < 1$$

which is a condition that is satisfied in this problem. Similarly, the Fourier transform of $x_2(n)$ is

$$\begin{aligned} X_2(\omega) &= \sum_{n=-\infty}^{\infty} x_2(n)e^{-j\omega n} = \sum_{n=-\infty}^{-1} a^{-n}e^{-j\omega n} \\ &= \sum_{n=-\infty}^{-1} (ae^{j\omega})^{-n} = \sum_{k=1}^{\infty} (ae^{j\omega})^k \\ &= \frac{ae^{j\omega}}{1 - ae^{j\omega}} \end{aligned}$$

By combining these two transforms, we obtain the Fourier transform of $x(n)$ in the form

$$\begin{aligned} X(\omega) &= X_1(\omega) + X_2(\omega) \\ &= \frac{1 - a^2}{1 - 2a\cos\omega + a^2} \end{aligned} \tag{4.4.46}$$

Figure 4.4.6 illustrates $x(n)$ and $X(\omega)$ for the case in which $a = 0.8$.

Time shifting. If

$$x(n) \overset{F}{\longleftrightarrow} X(\omega) \tag{4.4.47}$$

then

$$x(n-k) \overset{F}{\longleftrightarrow} e^{-j\omega k}X(\omega)$$

The proof of this property follows immediately from the Fourier transform of $x(n-k)$ by making a change in the summation index. Thus

$$\begin{aligned} F\{x(n-k)\} &= X(\omega)e^{-j\omega k} \\ &= |X(\omega)|e^{j[\measuredangle X(\omega) - \omega k]} \end{aligned}$$

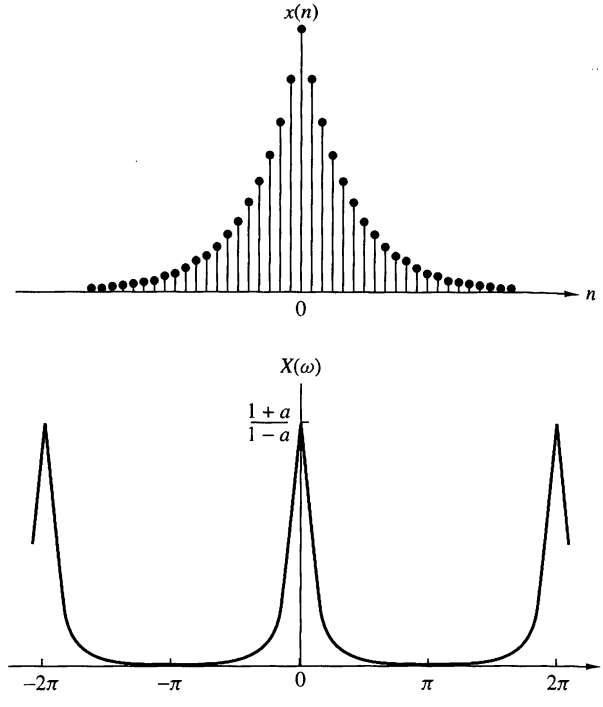


Figure 4.4.6
Sequence $x(n)$ and its Fourier transform in Example 4.4.3 with $a = 0.8$.

This relation means that if a signal is shifted in the time domain by k samples, its magnitude spectrum remains unchanged. However, the phase spectrum is changed by an amount $-\omega k$. This result can easily be explained if we recall that the frequency content of a signal depends only on its shape. From a mathematical point of view, we can say that shifting by k in the time domain is equivalent to multiplying the spectrum by $e^{-j\omega k}$ in the frequency domain.

Time reversal. If

$$x(n) \overset{F}{\longleftrightarrow} X(\omega)$$

then

$$x(-n) \overset{F}{\longleftrightarrow} X(-\omega) \tag{4.4.48}$$

This property can be established by performing the Fourier transformation of $x(-n)$ and making a simple change in the summation index. Thus

$$F\{x(-n)\} = \sum_{l=-\infty}^{\infty} x(l)e^{j\omega l} = X(-\omega)$$

If $x(n)$ is real, then from (4.4.17) and (4.4.18) we obtain

$$\begin{aligned} F\{x(-n)\} = X(-\omega) &= |X(-\omega)|e^{j\measuredangle X(-\omega)} \\ &= |X(\omega)|e^{-j\measuredangle X(\omega)} \end{aligned}$$

This means that if a signal is folded about the origin in time, its magnitude spectrum remains unchanged, and the phase spectrum undergoes a change in sign (phase reversal).

Convolution theorem. If

$$x_1(n) \overset{F}{\longleftrightarrow} X_1(\omega)$$

and

$$x_2(n) \overset{F}{\longleftrightarrow} X_2(\omega)$$

then

$$x(n) = x_1(n) * x_2(n) \overset{F}{\longleftrightarrow} X(\omega) = X_1(\omega)X_2(\omega) \tag{4.4.49}$$

To prove (4.4.49), we recall the convolution formula

$$x(n) = x_1(n) * x_2(n) = \sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k)$$

By multiplying both sides of this equation by the exponential $\exp(-j\omega n)$ and summing over all n, we obtain

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k) \right] e^{-j\omega n}$$

After interchanging the order of the summations and making a simple change in the summation index, the right-hand side of this equation reduces to the product $X_1(\omega)X_2(\omega)$. Thus (4.4.49) is established.

The convolution theorem is one of the most powerful tools in linear systems analysis. That is, if we convolve two signals in the time domain, then this is equivalent to multiplying their spectra in the frequency domain. In later chapters we will see that the convolution theorem provides an important computational tool for many digital signal processing applications.

EXAMPLE 4.4.4

By use of (4.4.49), determine the convolution of the sequences

$$x_1(n) = x_2(n) = \{1, \underset{\uparrow}{1}, 1\}$$

Solution. By using (4.4.21), we obtain

$$X_1(\omega) = X_2(\omega) = 1 + 2\cos\omega$$

Then

$$\begin{aligned} X(\omega) = X_1(\omega)X_2(\omega) &= (1 + 2\cos\omega)^2 \\ &= 3 + 4\cos\omega + 2\cos 2\omega \\ &= 3 + 2(e^{j\omega} + e^{-j\omega}) + (e^{j2\omega} + e^{-j2\omega}) \end{aligned}$$

Hence the convolution of $x_1(n)$ with $x_2(n)$ is

$$x(n) = \{1\ 2\ \underset{\uparrow}{3}\ 2\ 1\}$$

Figure 4.4.7 illustrates the foregoing relationships.

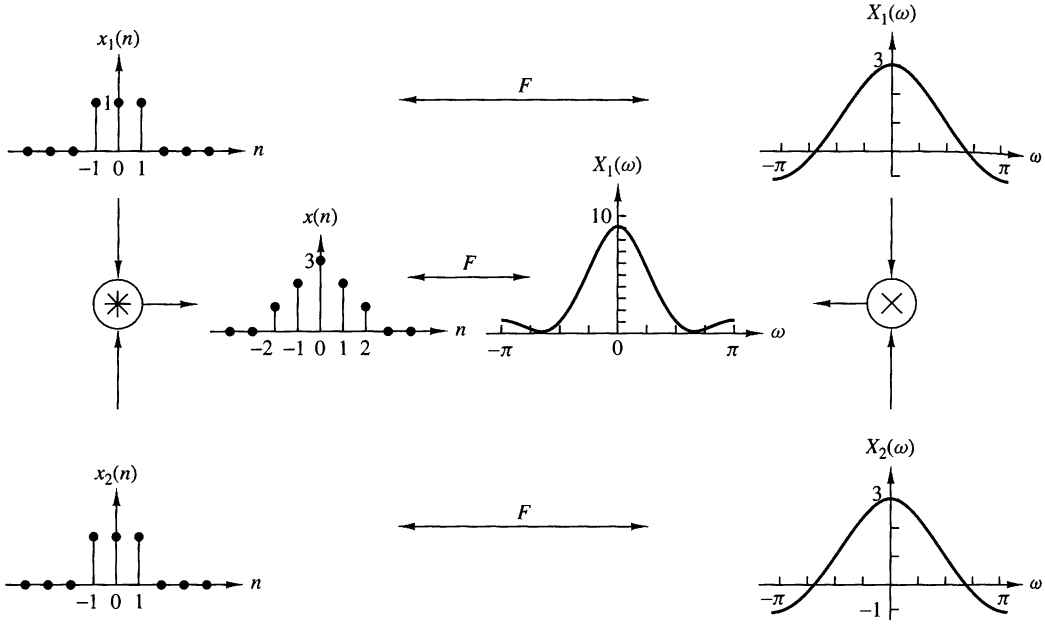


Figure 4.4.7 Graphical representation of the convolution property.

The correlation theorem. If

$$x_1(n) \overset{F}{\longleftrightarrow} X_1(\omega)$$

and

$$x_2(n) \overset{F}{\longleftrightarrow} X_2(\omega)$$

then

$$r_{x_1x_2}(m) \overset{F}{\longleftrightarrow} S_{x_1x_2}(\omega) = X_1(\omega)X_2(-\omega) \tag{4.4.50}$$

The proof of (4.4.50) is similar to the proof of (4.4.49). In this case, we have

$$r_{x_1x_2}(n) = \sum_{k=-\infty}^{\infty} x_1(k)x_2(k-n)$$

By multiplying both sides of this equation by the exponential $\exp(-j\omega n)$ and summing over all n, we obtain

$$S_{x_1x_2}(\omega) = \sum_{n=-\infty}^{\infty} r_{x1x2}(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} x_1(k)x_2(k-n) \right] e^{-j\omega n}$$

Finally, we interchange the order of the summations and make a change in the summation index. Thus we find that the right-hand side of the equation above reduces to $X_1(\omega)X_2(-\omega)$. The function $S_{x_1x_2}(\omega)$ is called the *cross-energy density spectrum* of the signals $x_1(n)$ and $x_2(n)$.

The Wiener–Khintchine theorem. Let $x(n)$ be a real signal. Then

$$r_{xx}(l) \overset{F}{\longleftrightarrow} S_{xx}(\omega) \tag{4.4.51}$$

That is, the energy spectral density of an energy signal is the Fourier transform of its autocorrelation sequence. This is a special case of (4.4.50).

This is a very important result. It means that the autocorrelation sequence of a signal and its energy spectral density contain the same information about the signal. Since neither of these contains any phase information, it is impossible to uniquely reconstruct the signal from the autocorrelation function or the energy density spectrum.

EXAMPLE 4.4.5

Determine the energy density spectrum of the signal

$$x(n) = a^n u(n), \qquad -1 < a < 1$$

Solution. From Example 2.6.2 we found that the autocorrelation function for this signal is

$$r_{xx}(l) = \frac{1}{1-a2}a^{|l|}, \qquad \infty < l < \infty$$

By using the result in (4.4.46) for the Fourier transform of $a^{|l|}$, derived in Example 4.4.3, we have

$$F\{r_{xx}(l)\} = \frac{1}{1-a^2}F\{a^{|l|}\} = \frac{1}{1-2a\cos\omega + a^2}$$

Thus, according to the Wiener–Khintchine theorem,

$$S_{xx}(\omega) = \frac{1}{1-2a\cos\omega + a^2}$$

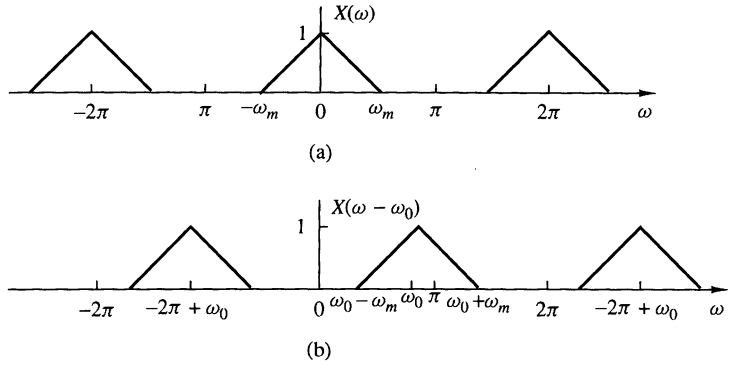


Figure 4.4.8 Illustration of the frequency-shifting property of the Fourier transform ($\omega_0 \leq 2\pi - \omega_m$).

Frequency shifting. If

$$x(n) \overset{F}{\longleftrightarrow} X(\omega)$$

then

$$e^{j\omega_0 n}x(n) \overset{F}{\longleftrightarrow} X(\omega - \omega_0) \tag{4.4.52}$$

This property is easily proved by direct substitution into the analysis equation (4.4.1). According to this property, multiplication of a sequence $x(n)$ by $e^{j\omega_0 n}$ is equivalent to a frequency translation of the spectrum $X(\omega)$ by ω_0. This frequency translation is illustrated in Fig 4.4.8. Since the spectrum $X(\omega)$ is periodic, the shift ω_0 applies to the spectrum of the signal in every period.

The modulation theorem. If

$$x(n) \overset{F}{\longleftrightarrow} X(\omega)$$

then

$$x(n)\cos\omega_0 n \overset{F}{\longleftrightarrow} \frac{1}{2}[X(\omega + \omega_0) + X(\omega - \omega_0)] \tag{4.4.53}$$

To prove the modulation theorem, we first express the signal $\cos\omega_0 n$ as

$$\cos\omega_0 n = \frac{1}{2}(e^{j\omega_0 n} + e^{-j\omega_0 n})$$

Upon multiplying $x(n)$ by these two exponentials and using the frequency-shifting property described in the preceding section, we obtain the desired result in (4.4.53).

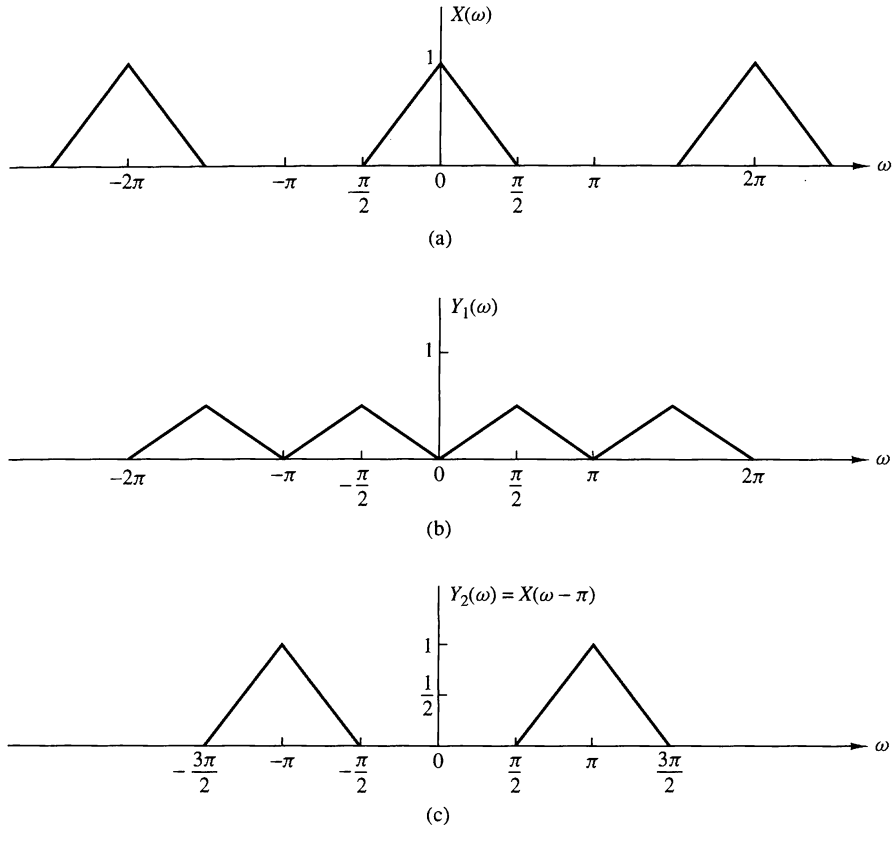


Figure 4.4.9 Graphical representation of the modulation theorem.

Although the property given in (4.4.52) can also be viewed as (complex) modulation, in practice we prefer to use (4.4.53) because the signal $x(n)\cos\omega_0 n$ is real. Clearly, in this case the symmetry properties (4.4.12) and (4.4.13) are preserved.

The modulation theorem is illustrated in Fig 4.4.9, which contains a plot of the spectra of the signals $x(n)$, $y_1(n) = x(n)\cos 0.5\pi n$ and $y_2(n) = x(n)\cos\pi n$.

Parseval's theorem. If

$$x_1(n) \overset{F}{\longleftrightarrow} X_1(\omega)$$

and

$$x_2(n) \overset{F}{\longleftrightarrow} X_2(\omega)$$

then

$$\sum_{n=-\infty}^{\infty} x_1(n)x_2^*(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\omega)X_2^*(\omega)\,d\omega \tag{4.4.54}$$

To prove this theorem, we use (4.4.1) to eliminate $X_1(\omega)$ on the right-hand side of (4.4.54). Thus we have

$$\frac{1}{2\pi}\int_{2\pi}\left[\sum_{n=-\infty}^{\infty} x_1(n)e^{-j\omega n}\right]X_2^*(\omega)\,d\omega$$

$$=\sum_{n=-\infty}^{\infty} x_1(n)\frac{1}{2\pi}\int_{2\pi} X_2^*(\omega)e^{-j\omega n}\,d\omega=\sum_{n=-\infty}^{\infty} x_1(n)x_2^*(n)$$

In the special case where $x_2(n) = x_1(n) = x(n)$, Parseval's relation (4.4.54) reduces to

$$\sum_{n=-\infty}^{\infty}|x(n)|^2=\frac{1}{2\pi}\int_{2\pi}|X(\omega)|^2\,d\omega \tag{4.4.55}$$

We observe that the left-hand side of (4.4.55) is simply the energy E_x of the signal $x(n)$. It is also equal to the autocorrelation of $x(n)$, $r_{xx}(l)$, evaluated at $l = 0$. The integrand in the right-hand side of (4.4.55) is equal to the energy density spectrum, so the integral over the interval $-\pi \le \omega \le \pi$ yields the total signal energy. Therefore, we conclude that

$$E_x = r_{xx}(0)=\sum_{n=-\infty}^{\infty}|x(n)|^2=\frac{1}{2\pi}\int_{2\pi}|X(\omega)|^2\,d\omega=\frac{1}{2\pi}\int_{-\pi}^{\pi}S_{xx}(\omega)\,d\omega \tag{4.4.56}$$

Multiplication of two sequences (Windowing theorem). If

$$x_1(n)\overset{F}{\longleftrightarrow}X_1(\omega)$$

and

$$x_2(n)\overset{F}{\longleftrightarrow}X_2(\omega)$$

then

$$x_3(n)\equiv x_1(n)x_2(n)\overset{F}{\longleftrightarrow}X_3(\omega)=\frac{1}{2\pi}\int_{-\pi}^{\pi}X_1(\lambda)X_2(\omega-\lambda)d\lambda \tag{4.4.57}$$

The integral on the right-hand side of (4.4.57) represents the convolution of the Fourier transforms $X_1(\omega)$ and $X_2(\omega)$. This relation is the dual of the time-domain convolution. In other words, the multiplication of two time-domain sequences is equivalent to the convolution of their Fourier transforms. On the other hand, the convolution of two time-domain sequences is equivalent to the multiplication of their Fourier transforms.

To prove (4.4.57) we begin with the Fourier transform of $x_3(n) = x_1(n)x_2(n)$ and use the formula for the inverse transform, namely,

$$x_1(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\lambda)e^{j\lambda n}d\lambda$$

Thus, we have

$$\begin{aligned} X_3(\omega) &= \sum_{n=-\infty}^{\infty} x_3(n)e^{-j\omega n} = \sum_{n=-\infty}^{\infty} x_1(n)x_2(n)e^{-j\omega n} \\ &= \sum_{n=-\infty}^{\infty}\left[\frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\lambda)e^{j\lambda n}d\lambda\right]x_2(n)e^{-j\omega n} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\lambda)d\lambda\left[\sum_{n=-\infty}^{\infty} x_2(n)e^{-j(\omega-\lambda)n}\right] \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\lambda)X_2(\omega-\lambda)d\lambda \end{aligned}$$

The convolution integral in (4.4.57) is known as the *periodic convolution* of $X_1(\omega)$ and $X_2(\omega)$ because it is the convolution of two periodic functions having the same period. We note that the limits of integration extend over a single period. Furthermore, we note that due to the periodicity of the Fourier transform for discrete-time signals, there is no "perfect" duality between the time and frequency domains with respect to the convolution operation, as in the case of continuous-time signals. Indeed, convolution in the time domain (aperiodic summation) is equivalent to multiplication of continuous periodic Fourier transforms. However, multiplication of aperiodic sequences is equivalent to periodic convolution of their Fourier transforms.

The Fourier transform pair in (4.4.57) will prove useful in our treatment of FIR filter design based on the window technique.

Differentiation in the frequency domain. If

$$x(n) \overset{F}{\longleftrightarrow} X(\omega)$$

then

$$nx(n) \overset{F}{\longleftrightarrow} j\frac{dX(\omega)}{d\omega} \tag{4.4.58}$$

To prove this property, we use the definition of the Fourier transform in (4.4.1) and differentiate the series term by term with respect to ω. Thus we obtain

$$\begin{aligned}\frac{dX(\omega)}{d\omega} &= \frac{d}{d\omega}\left[\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}\right] \\ &= \sum_{n=-\infty}^{\infty} x(n)\frac{d}{d\omega}e^{-j\omega n} \\ &= -j\sum_{n=-\infty}^{\infty} nx(n)e^{-j\omega n}\end{aligned}$$

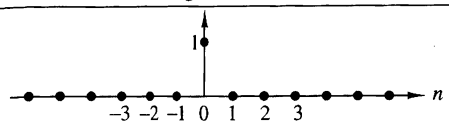
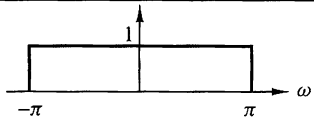
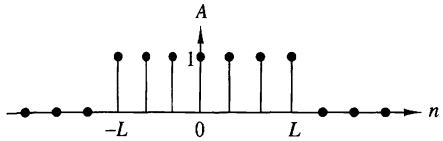
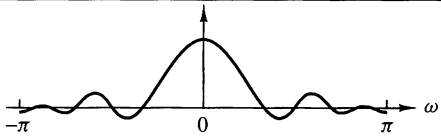
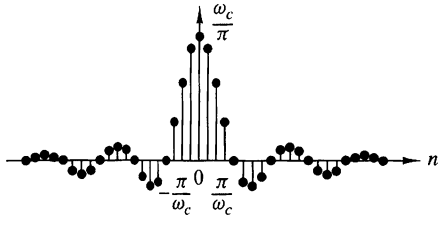
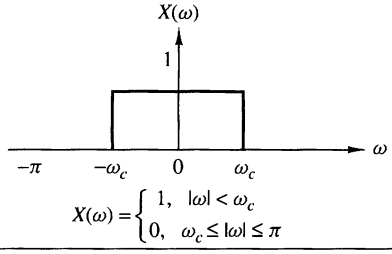
Now we multiply both sides of the equation by j to obtain the desired result in (4.4.58).

The properties derived in this section are summarized in Table 4.5, which serves as a convenient reference. Table 4.6 illustrates some useful Fourier transform pairs that will be encountered in later chapters.

TABLE 4.5 Properties of the Fourier Transform for Discrete-Time Signals

Property	Time Domain	Frequency Domain
Notation	$x(n)$	$X(\omega)$
	$x_1(n)$	$X_1(\omega)$
	$x_2(n)$	$X_2(\omega)$
Linearity	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(\omega) + a_2X_2(\omega)$
Time shifting	$x(n-k)$	$e^{-j\omega k}X(\omega)$
Time reversal	$x(-n)$	$X(-\omega)$
Convolution	$x_1(n) * x_2(n)$	$X_1(\omega)X_2(\omega)$
Correlation	$r_{x_1x_2}(l) = x_1(l) * x_2(-l)$	$S_{x_1x_2}(\omega) = X_1(\omega)X_2(-\omega)$ $= X_1(\omega)X_2^*(\omega)$ [if $x_2(n)$ is real]
Wiener–Khintchine theorem	$r_{xx}(l)$	$S_{xx}(\omega)$
Frequency shifting	$e^{j\omega_0 n}x(n)$	$X(\omega - \omega_0)$
Modulation	$x(n)\cos\omega_0 n$	$\frac{1}{2}X(\omega + \omega_0) + \frac{1}{2}X(\omega - \omega_0)$
Multiplication	$x_1(n)x_2(n)$	$\frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\lambda)X_2(\omega - \lambda)d\lambda$
Differentiation in the frequency domain	$nx(n)$	$j\frac{dX(\omega)}{d\omega}$
Conjugation	$x^*(n)$	$X^*(-\omega)$
Parseval's theorem	$\sum_{n=-\infty}^{\infty} x_1(n)x_2^*(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_1(\omega)X_2^*(\omega)d\omega$	

TABLE 4.6 Some Useful Fourier Transform Pairs for Discrete-Time Aperiodic Signals

Signal $x(n)$	Spectrum $X(\omega)$
 $x(n) = \delta(n)$	 $X(\pi) = 1$
 $x(n) = \begin{cases} A, & \lvert n\rvert \le L \\ 0, & \lvert n\rvert > L \end{cases}$	 $X(\omega) = A \dfrac{\sin\left(L+\frac{1}{2}\right)\omega}{\sin\frac{\omega}{2}}$
 $x(n) = \begin{cases} \frac{\omega_c}{\pi}, & n = 0 \\ \frac{\sin \omega_c n}{\pi n}, & n \neq 0 \end{cases}$	 $X(\omega) = \begin{cases} 1, & \lvert\omega\rvert < \omega_c \\ 0, & \omega_c \le \lvert\omega\rvert \le \pi \end{cases}$
$x(n) = \begin{cases} a^n, & n \ge 0 \\ 0, & n > 0 \end{cases}$	$X(\omega) = \frac{1}{1-ae^{-j\omega}}$

4.5 Summary and References

The Fourier series and the Fourier transform are the mathematical tools for analyzing the characteristics of signals in the frequency domain. The Fourier series is appropriate for representing a periodic signal as a weighted sum of harmonically related sinusoidal components, where the weighting coefficients represent the strengths of each of the harmonics, and the magnitude squared of each weighting coefficient represents the power of the corresponding harmonic. As we have indicated, the Fourier series is one of many possible orthogonal series expansions for a periodic signal. Its importance stems from the characteristic behavior of LTI systems, as we shall see in Chapter 5.

The Fourier transform is appropriate for representing the spectral characteristics of aperiodic signals with finite energy. The important properties of the Fourier transform were also presented in this chapter.

There are many excellent texts on Fourier series and Fourier transforms. For reference, we include the texts by Bracewell (1978), Davis (1963), Dym and McKean (1972), and Papoulis (1962).

Problems

4.1 Consider the full-wave rectified sinusoid in Fig. P4.1.

(a) Determine its spectrum $X_a(F)$.

(b) Compute the power of the signal.

(c) Plot the power spectral density.

(d) Check the validity of Parseval's relation for this signal.

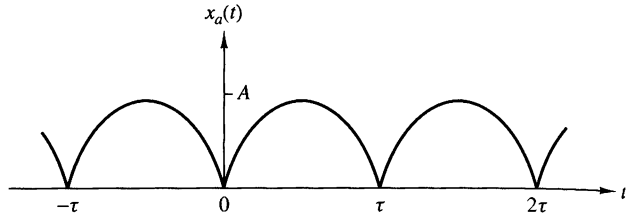


Figure P4.1

4.2 Compute and sketch the magnitude and phase spectra for the following signals ($a > 0$).

(a) $x_a(t) = \begin{cases} Ae^{-at}, & t \geq 0 \\ 0, & t < 0 \end{cases}$

(b) $x_a(t) = Ae^{-a|t|}$

4.3 Consider the signal

$$x(t) = \begin{cases} 1 - |t|/\tau, & |t| \leq \tau \\ 0, & \text{elsewhere} \end{cases}$$

(a) Determine and sketch its magnitude and phase spectra, $|X_a(F)|$ and $\measuredangle X_a(F)$, respectively.

(b) Create a periodic signal $x_p(t)$ with fundamental period $T_p \geq 2\tau$, so that $x(t) = x_p(t)$ for $|t| < T_p/2$. What are the Fourier coefficients c_k for the signal $x_p(t)$?

(c) Using the results in parts (a) and (b), show that $c_k = (1/T_p)X_a(k/T_p)$.

4.4 Consider the following periodic signal:

$$x(n) = \{\ldots, 1, 0, 1, 2, \underset{\uparrow}{3}, 2, 1, 0, 1, \ldots\}$$

(a) Sketch the signal $x(n)$ and its magnitude and phase spectra.

(b) Using the results in part (a), verify Parseval's relation by computing the power in the time and frequency domains.

4.5 Consider the signal

$$x(n) = 2 + 2\cos\frac{\pi n}{4} + \cos\frac{\pi n}{2} + \frac{1}{2}\cos\frac{3\pi n}{4}$$

(a) Determine and sketch its power density spectrum.

(b) Evaluate the power of the signal.

Determine and sketch the magnitude and phase spectra of the following periodic signals.

(a) $x(n) = 4 \sin \frac{\pi(n-2)}{3}$

(b) $x(n) = \cos \frac{2\pi}{3} n + \sin \frac{2\pi}{5} n$

(c) $x(n) = \cos \frac{2\pi}{3} n \sin \frac{2\pi}{5} n$

(d) $x(n) = \{\ldots, -2, -1, \underset{\uparrow}{0}, 1, 2, -2, -1, 0, 1, 2, \ldots\}$

(e) $x(n) = \{\ldots, -1, 2, \underset{\uparrow}{1}, 2, -1, 0, -1, 2, 1, 2, \ldots\}$

(f) $x(n) = \{\ldots, 0, 0, \underset{\uparrow}{1}, 1, 0, 0, 0, 1, 1, 0, 0, \ldots\}$

(g) $x(n) = 1, -\infty < n < \infty$

(h) $x(n) = (-1)^n, -\infty < n < \infty$

Determine the periodic signals $x(n)$, with fundamental period $N = 8$, if their Fourier coefficients are given by:

(a) $c_k = \cos \frac{k\pi}{4} + \sin \frac{3k\pi}{4}$

(b) $c_k = \begin{cases} \sin \frac{k\pi}{3}, & 0 \le k \le 6 \\ 0, & k = 7 \end{cases}$

(c) $\{c_k\} = \{\ldots, 0, \frac{1}{4}, \frac{1}{2}, 1, \underset{\uparrow}{2}, 1, \frac{1}{2}, \frac{1}{4}, 0 \ldots\}$

Two DT signals, $s_k(n)$ and $s_l(n)$, are said to be orthogonal over an interval $[N_1, N_2]$ if

$$\sum_{n=N_1}^{N_2} s_k(n) s_l^*(n) = \begin{cases} A_k, & k = l \\ 0, & k \neq l \end{cases}$$

If $A_k = 1$, the signals are called orthonormal.

(a) Prove the relation

$$\sum_{n=0}^{N-1} e^{j2\pi kn/N} = \begin{cases} N, & k = 0, \pm N, \pm 2N, \ldots \\ 0, & \text{otherwise} \end{cases}$$

(b) Illustrate the validity of the relation in part (a) by plotting for every value of $k = 1, 2, \ldots, 6$, the signals $s_k(n) = e^{j(2\pi/6)kn}$, $n = 0, 1, \ldots, 5$. [*Note*: For a given k, n the signal $s_k(n)$ can be represented as a vector in the complex plane.]

(c) Show that the harmonically related signals

$$s_k(n) = e^{j(2\pi/N)kn}$$

are orthogonal over any interval of length N.

4.9 Compute the Fourier transform of the following signals.

(a) $x(n) = u(n) - u(n-6)$

(b) $x(n) = 2^n u(-n)$

(c) $x(n) = (\frac{1}{4})^n u(n+4)$

(d) $x(n) = (\alpha^n \sin \omega_0 n)u(n), \quad |\alpha| < 1$

(e) $x(n) = |\alpha|^n \sin \omega_0 n, \quad |\alpha| < 1$

(f) $x(n) = \begin{cases} 2 - (\frac{1}{2})n, & |n| \leq 4 \\ 0, & \text{elsewhere} \end{cases}$

(g) $x(n) = \{-2, -1, \underset{\uparrow}{0}, 1, 2\}$

(h) $x(n) = \begin{cases} A(2M + 1 - |n|), & |n| \leq M \\ 0, & |n| > M \end{cases}$

Sketch the magnitude and phase spectra for parts (a), (f), and (g).

4.10 Determine the signals having the following Fourier transforms.

(a) $X(\omega) = \begin{cases} 0, & 0 \leq |\omega| \leq \omega_0 \\ 1, & \omega_0 < |\omega| \leq \pi \end{cases}$

(b) $X(\omega) = \cos^2 \omega$

(c) $X(\omega) = \begin{cases} 1, & \omega_0 - \delta\omega/2 \leq |\omega| \leq \omega_0 + \delta\omega/2 \\ 0, & \text{elsewhere} \end{cases}$

(d) The signal shown in Fig. P4.10.

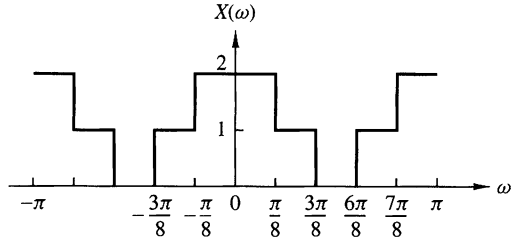


Figure P4.10

4.11 Consider the signal

$$x(n) = \{1, 0, -1, \underset{\uparrow}{2}, 3\}$$

with Fourier transform $X(\omega) = X_R(\omega) + j(X_I(\omega))$. Determine and sketch the signal $y(n)$ with Fourier transform

$$Y(\omega) = X_I(\omega) + X_R(\omega)e^{j2\omega}$$

4.12 Determine the signal $x(n)$ if its Fourier transform is as given in Fig. P4.12.

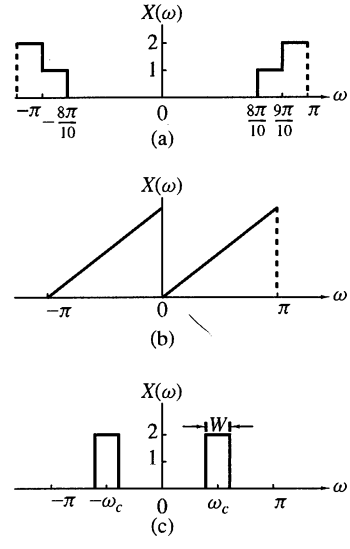


Figure P4.12

4.13 In Example 4.4.2, the Fourier transform of the signal

$$x(n) = \begin{cases} 1, & -M \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

was shown to be

$$X(\omega) = 1 + 2\sum_{n=1}^{M} \cos \omega n$$

Show that the Fourier transform of

$$x_1(n) = \begin{cases} 1, & 0 \le n \le M \\ 0, & \text{otherwise} \end{cases}$$

and

$$x_2(n) = \begin{cases} 1, & -M \le n \le -1 \\ 0, & \text{otherwise} \end{cases}$$

are, respectively,

$$X_1(\omega) = \frac{1 - e^{-j\omega(M+1)}}{1 - e^{-j\omega}}$$

$$X_2(\omega) = \frac{e^{j\omega} - e^{j\omega(M+1)}}{1 - e^{j\omega}}$$

Thus prove that

$$X(\omega) = X_1(\omega) + X_2(\omega)$$
$$= \frac{\sin(M + \frac{1}{2})\omega}{\sin(\omega/2)}$$

and therefore,

$$1 + 2\sum_{n=1}^{M} \cos \omega n = \frac{\sin(M + \frac{1}{2})\omega}{\sin(\omega/2)}$$

4.14 Consider the signal

$$x(n) = \{-1, 2, \underset{\uparrow}{-3}, 2, -1\}$$

with Fourier transform $X(\omega)$. Compute the following quantities, without explicitly computing $X(\omega)$:

(a) $X(0)$

(b) $\angle X(\omega)$

(c) $\int_{-\pi}^{\pi} X(\omega)\, d\omega$

(d) $X(\pi)$

(e) $\int_{-\pi}^{\pi} |X(\omega)|^2\, d\omega$

4.15 The center of gravity of a signal $x(n)$ is defined as

$$c = \frac{\sum_{n=-\infty}^{\infty} nx(n)}{\sum_{n=-\infty}^{\infty} x(n)}$$

and provides a measure of the "time delay" of the signal.

(a) Express c in terms of $X(\omega)$.

(b) Compute c for the signal $x(n)$ whose Fourier transform is shown in Fig. P4.15.

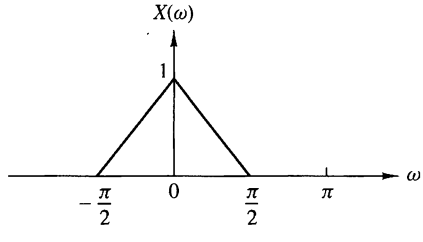


Figure P4.15

4.16 Consider the Fourier transform pair

$$a^n u(n) \overset{F}{\longleftrightarrow} \frac{1}{1 - ae^{-j\omega}}, \qquad |a| < 1$$

Use the differentiation in frequency theorem and induction to show that

$$x(n) = \frac{(n + l - 1)!}{n!(l - 1)!} a^n u(n) \overset{F}{\longleftrightarrow} X(\omega) = \frac{1}{(1 - ae^{-j\omega})^l}$$

4.17 Let $x(n)$ be an arbitrary signal, not necessarily real valued, with Fourier transform $X(\omega)$. Express the Fourier transforms of the following signals in terms of $X(\omega)$.

(a) $x^*(n)$

(b) $x^*(-n)$

(c) $y(n) = x(n) - x(n-1)$

(d) $y(n) = \sum_{k=-\infty}^{n} x(k)$

(e) $y(n) = x(2n)$

(f) $y(n) = \begin{cases} x(n/2), & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$

4.18 Determine and sketch the Fourier transforms $X_1(\omega)$, $X_2(\omega)$, and $X_3(\omega)$ of the following signals.

(a) $x_1(n) = \{1, 1, \underset{\uparrow}{1}, 1, 1\}$

(b) $x_2(n) = \{1, 0, 1, 0, \underset{\uparrow}{1}, 0, 1, 0, 1\}$

(c) $x_3(n) = \{1, 0, 0, 1, 0, 0, \underset{\uparrow}{1}, 0, 0, 1, 0, 0, 1\}$

(d) Is there any relation between $X_1(\omega)$, $X_2(\omega)$, and $X_3(\omega)$? What is its physical meaning?

(e) Show that if

$$x_k(n) = \begin{cases} x\left(\frac{n}{k}\right), & \text{if } n/k \text{ integer} \\ 0, & \text{otherwise} \end{cases}$$

then

$$X_k(\omega) = X(k\omega)$$

4.19 Let $x(n)$ be a signal with Fourier transform as shown in Fig. P4.19. Determine and sketch the Fourier transforms of the following signals.

(a) $x_1(n) = x(n)\cos(\pi n/4)$

(b) $x_2(n) = x(n)\sin(\pi n/2)$

(c) $x_3(n) = x(n)\cos(\pi n/2)$

(d) $x_4(n) = x(n)\cos \pi n$

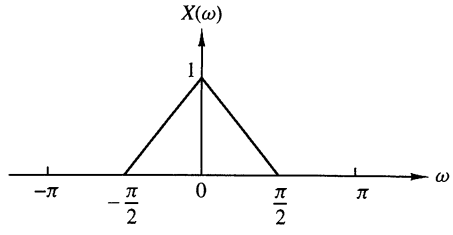


Figure P4.19

Note that these signal sequences are obtained by *amplitude modulation* of a carrier $\cos \omega_c n$ or $\sin \omega_c n$ by the sequence $x(n)$.

4.20 Consider an aperiodic signal $x(n)$ with Fourier transform $X(\omega)$. Show that the Fourier series coefficients C_k^y of the periodic signal

$$y(n) = \sum_{l=-\infty}^{\infty} x(n - lN)$$

are given by

$$C_k^y = \frac{1}{N} X\left(\frac{2\pi}{N} k\right), \qquad k = 0, 1, \ldots, N-1$$

4.21 Prove that

$$X_N(\omega) = \sum_{n=-N}^{N} \frac{\sin \omega_c n}{\pi n} e^{-j\omega n}$$

may be expressed as

$$X_N(\omega) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} \frac{\sin[(2N+1)(\omega - \theta/2)]}{sin[(\omega - \theta)/2]} d\theta$$

4.22 A signal $x(n)$ has the following Fourier transform:

$$X(\omega) = \frac{1}{1 - ae^{-j\omega}}$$

Determine the Fourier transforms of the following signals:

(a) $x(2n+1)$

(b) $e^{\pi n/2} x(n+2)$

(c) $x(-2n)$

(d) $x(n) \cos(0.3\pi n)$

(e) $x(n) * x(n-1)$

(f) $x(n) * x(-n)$

4.23 From a discrete-time signal $x(n)$ with Fourier transform $X(\omega)$, shown in Fig. P4.23, determine and sketch the Fourier transform of the following signals:

(a) $y_1(n) = \begin{cases} x(n), & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$

(b) $y_2(n) = x(2n)$

(c) $y_3(n) = \begin{cases} x(n/2), & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$

Note that $y_1(n) = x(n)s(n)$, where $s(n) = \{\ldots 0, 1, 0, \underset{\uparrow}{1}, 0, 1, 0, 1, \ldots\}$

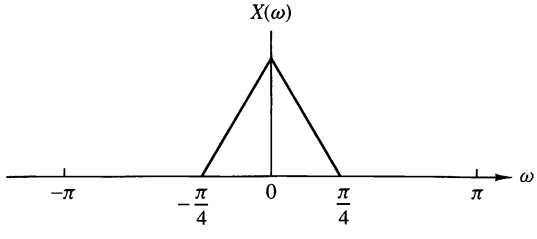


Figure P4.23

CHAPTER

5

Frequency-Domain Analysis of LTI Systems

In this chapter we treat the characterization of linear time-invariant systems in the frequency domain. The basic excitation signals in this development are the complex exponentials and sinusoidal functions. We will observe that an LTI system performs a discrimination or filtering on the various frequency components at its input. This observation leads us to characterize and classify some simple LTI systems according to the type of filtering they perform on any input signal. The design of these simple filters is described and some applications are given.

We also develop frequency-domain relationships between the spectra of the input and output sequences of an LTI system. The final section of this chapter is focused on the application of LTI systems for performing inverse filtering and deconvolution.

5.1 Frequency-Domain Characteristics of Linear Time-Invariant Systems

In this section we develop the characterization of linear time-invariant systems in the frequency domain. The basic excitation signals in this development are the complex exponentials and sinusoidal functions. The characteristics of the system are described by a function of the frequency variable ω called the frequency response, which is the Fourier transform of the impulse response $h(n)$ of the system.

The frequency response function completely characterizes a linear time-invariant system in the frequency domain. This allows us to determine the steady-state response of the system to any arbitrary weighted linear combination of sinusoids or complex exponentials. Since periodic sequences, in particular, lend themselves to a Fourier series decomposition as a weighted sum of harmonically related complex

exponentials, it becomes a simple matter to determine the response of a linear time-invariant system to this class of signals. This methodology is also applied to aperiodic signals since such signals can be viewed as a superposition of infinitesimal size complex exponentials.

5.1.1 Response to Complex Exponential and Sinusoidal Signals: The Frequency Response Function

In Chapter 2, it was demonstrated that the response of any relaxed linear time-invariant system to an arbitrary input signal $x(n)$ is given by the convolution sum formula

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{5.1.1}$$

In this input–output relationship, the system is characterized in the time domain by its unit sample response $\{h(n), -\infty < n < \infty\}$.

To develop a frequency-domain characterization of the system, let us excite the system with the complex exponential

$$x(n) = Ae^{j\omega n}, \qquad -\infty < n < \infty \tag{5.1.2}$$

where A is the amplitude and ω is any arbitrary frequency confined to the frequency interval $[-\pi, \pi]$. By substituting (5.1.2) into (5.1.1), we obtain the response

$$\begin{aligned} y(n) &= \sum_{k=-\infty}^{\infty} h(k)[Ae^{j\omega(n-k)}] \\ &= A\left[\sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k}\right] e^{j\omega n} \end{aligned} \tag{5.1.3}$$

We observe that the term in brackets in (5.1.3) is a function of the frequency variable ω. In fact, this term is the Fourier transform of the unit sample response $h(k)$ of the system. Hence we denote this function as

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k} \tag{5.1.4}$$

Clearly, the function $H(\omega)$ exists if the system is BIBO stable, that is, if

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

With the definition in (5.1.4), the response of the system to the complex exponential given in (5.1.2) is

$$y(n) = AH(\omega)e^{j\omega n} \tag{5.1.5}$$

We note that the response is also in the form of a complex exponential with the same frequency as the input, but altered by the multiplicative factor $H(\omega)$.

As a result of this characteristic behavior, the exponential signal in (5.1.2) is called an *eigenfunction* of the system. In other words, an eigenfunction of a system is an input signal that produces an output that differs from the input by a constant multiplicative factor. The multiplicative factor is called an *eigenvalue* of the system. In this case, a complex exponential signal of the form (5.1.2) is an eigenfunction of a linear time-invariant system, and $H(\omega)$ evaluated at the frequency of the input signal is the corresponding eigenvalue.

EXAMPLE 5.1.1

Determine the output sequence of the system with impulse response

$$h(n) = (\frac{1}{2})^n u(n) \tag{5.1.6}$$

when the input is the complex exponential sequence

$$x(n) = Ae^{j\pi n/2}, \qquad -\infty < n < \infty$$

Solution. First we evaluate the Fourier transform of the impulse response $h(n)$, and then we use (5.1.5) to determine $y(n)$. From Example 4.2.3 we recall that

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n} = \frac{1}{1 - \frac{1}{2}e^{-j\omega}} \tag{5.1.7}$$

At $\omega = \pi/2$, (5.1.7) yields

$$H\left(\frac{\pi}{2}\right) = \frac{1}{1 + j\frac{1}{2}} = \frac{2}{\sqrt{5}}e^{-j26.6^\circ}$$

and therefore the output is

$$\begin{aligned} y(n) &= A\left(\frac{2}{\sqrt{5}}e^{-j26.6^\circ}\right)e^{j\pi n/2} \\ y(n) &= \frac{2}{\sqrt{5}}Ae^{j(\pi n/2 - 26.6^\circ)}, \qquad -\infty < n < \infty \end{aligned} \tag{5.1.8}$$

This example clearly illustrates that the only effect of the system on the input signal is to scale the amplitude by $2/\sqrt{5}$ and shift the phase by -26.6°. Thus the output is also a complex exponential of frequency $\pi/2$, amplitude $2A/\sqrt{5}$, and phase -26.6°.

If we alter the frequency of the input signal, the effect of the system on the input also changes and hence the output changes. In particular, if the input sequence is a complex exponential of frequency π, that is,

$$x(n) = Ae^{j\pi n}, \qquad -\infty < n < \infty \tag{5.1.9}$$

then, at $\omega = \pi$,

$$H(\pi) = \frac{1}{1 - \frac{1}{2}e^{-j\pi}} = \frac{1}{\frac{3}{2}} = \frac{2}{3}$$

and the output of the system is

$$y(n) = \frac{2}{3}Ae^{j\pi n}, \qquad -\infty < n < \infty \tag{5.1.10}$$

We note that $H(\pi)$ is purely real [i.e., the phase associated with $H(\omega)$ is zero at $\omega = \pi$]. Hence, the input is scaled in amplitude by the factor $H(\pi) = \frac{2}{3}$, but the phase shift is zero.

In general, $H(\omega)$ is a complex-valued function of the frequency variable ω. Hence it can be expressed in polar form as

$$H(\omega) = |H(\omega)|e^{j\Theta(\omega)} \tag{5.1.11}$$

where $|H(\omega)|$ is the magnitude of $H(\omega)$ and

$$\Theta(\omega) = \measuredangle H(\omega)$$

which is the phase shift imparted on the input signal by the system at the frequency ω.

Since $H(\omega)$ is the Fourier transform of $\{h(k)\}$, it follows that $H(\omega)$ is a periodic function with period 2π. Furthermore, we can view (5.1.4) as the exponential Fourier series expansion for $H(\omega)$, with $h(k)$ as the Fourier series coefficients. Consequently, the unit impulse $h(k)$ is related to $H(\omega)$ through the integral expression

$$h(k) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega k}d\omega \tag{5.1.12}$$

For a linear time-invariant system with a real-valued impulse response, the magnitude and phase functions possess symmetry properties which are developed as follows. From the definition of $H(\omega)$, we have

$$\begin{aligned} H(\omega) &= \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k} \\ &= \sum_{k=-\infty}^{\infty} h(k)\cos\omega k - j\sum_{k=-\infty}^{\infty} h(k)\sin\omega k \\ &= H_R(\omega) + jH_I(\omega) \\ &= \sqrt{H_R^2(\omega) + H_I^2(\omega)}e^{j\tan^{-1}[H_I(\omega)/H_R(\omega)]} \end{aligned} \tag{5.1.13}$$

where $H_R(\omega)$ and $H_I(\omega)$ denote the real and imaginary components of $H(\omega)$, defined as

$$\begin{aligned} H_R(\omega) &= \sum_{k=-\infty}^{\infty} h(k)\cos\omega k \\ H_I(\omega) &= -\sum_{k=-\infty}^{\infty} h(k)\sin\omega k \end{aligned} \tag{5.1.14}$$

It is clear from (5.1.12) that the magnitude and phase of $H(\omega)$, expressed in terms of $H_R(\omega)$ and $H_I(\omega)$, are

$$\begin{aligned} |H(\omega)| &= \sqrt{H_R^2(\omega) + H_I^2(\omega)} \\ \Theta(\omega) &= \tan^{-1}\frac{H_I(\omega)}{H_R(\omega)} \end{aligned} \tag{5.1.15}$$

We note that $H_R(\omega) = H_R(-\omega)$ and $H_I(\omega) = -H_I(-\omega)$, so that $H_R(\omega)$ is an even function of ω and $H_I(\omega)$ is an odd function of ω. As a consequence, it follows that $|H(\omega)|$ is an even function of ω and $\Theta(\omega)$ is an odd function of ω. Hence, if we know $|H(\omega)|$ and $\Theta(\omega)$ for $0 \le \omega \le \pi$, we also know these functions for $-\pi \le \omega \le 0$.

EXAMPLE 5.1.2 Moving Average Filter

Determine the magnitude and phase of $H(\omega)$ for the three-point moving average (MA) system

$$y(n) = \frac{1}{3}[x(n+1) + x(n) + x(n-1)]$$

and plot these two functions for $0 \le \omega \le \pi$.

Solution. Since

$$h(n) = \{\frac{1}{3}, \underset{\uparrow}{\frac{1}{3}}, \frac{1}{3}\}$$

it follows that

$$H(\omega) = \frac{1}{3}(e^{j\omega} + 1 + e^{-j\omega}) = \frac{1}{3}(1 + 2\cos\omega)$$

Hence

$$\begin{aligned} |H(\omega)| &= \frac{1}{3}|1 + 2\cos\omega| \\ \Theta(\omega) &= \begin{cases} 0, & 0 \le \omega \le 2\pi/3 \\ \pi, & 2\pi/3 \le \omega < \pi \end{cases} \end{aligned} \tag{5.1.16}$$

Figure 5.1.1 illustrates the graphs of the magnitude and phase of $H(\omega)$. As indicated previously, $|H(\omega)|$ is an even function of frequency and $\Theta(\omega)$ is an odd function of frequency. It is apparent from the frequency response characteristic $H(\omega)$ that this moving average filter smooths the input data, as we would expect from the input–output equation.

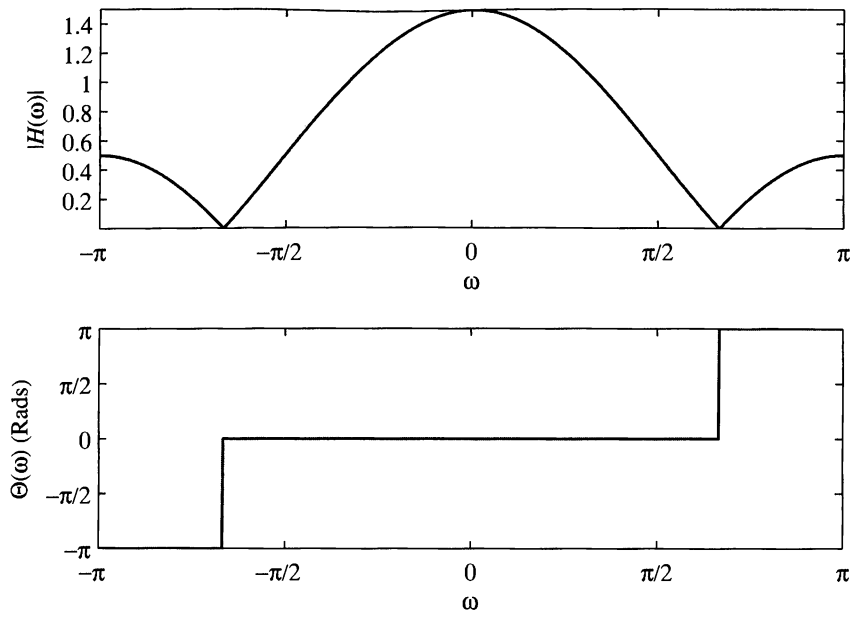


Figure 5.1.1 Magnitude and phase responses for the MA system in Example 5.1.2.

The symmetry properties satisfied by the magnitude and phase functions of $H(\omega)$, and the fact that a sinusoid can be expressed as a sum or difference of two complex-conjugate exponential functions, imply that the response of a linear time-invariant system to a sinusoid is similar in form to the response when the input is a complex exponential. Indeed, if the input is

$$x_1(n) = Ae^{j\omega n}$$

the output is

$$y_1(n) = A|H(\omega)|e^{j\Theta(\omega)}e^{j\omega n}$$

On the other hand, if the input is

$$x_2(n) = Ae^{-j\omega n}$$

the response of the system is

$$\begin{aligned} y_2(n) &= A|H(-\omega)|e^{j\Theta(-\omega)}e^{-j\omega n} \\ &= A|H(\omega)|e^{-j\Theta(\omega)}e^{-j\omega n} \end{aligned}$$

where, in the last expression, we have made use of the symmetry properties $|H(\omega)| = |H(-\omega)|$ and $\Theta(\omega) = -\Theta(-\omega)$. Now, by applying the superposition property of the linear time-invariant system, we find that the response of the system to the input

$$x(n) = \frac{1}{2}[x_1(n) + x_2(n)] = A\cos\omega n$$

is

$$\begin{aligned} y(n) &= \frac{1}{2}[y_1(n) + y_2(n)] \\ y(n) &= A|H(\omega)|\cos[\omega n + \Theta(\omega)] \end{aligned} \tag{5.1.17}$$

Similarly, if the input is

$$x(n) = \frac{1}{j2}[x_1(n) - x_2(n)] = A\sin\omega n$$

the response of the system is

$$\begin{aligned} y(n) &= \frac{1}{j2}[y_1(n) - y_2(n)] \\ y(n) &= A|H(\omega)|\sin[\omega n + \Theta(\omega)] \end{aligned} \tag{5.1.18}$$

It is apparent from this discussion that $H(\omega)$, or equivalently, $|H(\omega)|$ and $\Theta(\omega)$, completely characterize the effect of the system on a sinusoidal input signal of any arbitrary frequency. Indeed, we note that $|H(\omega)|$ determines the amplification ($|H(\omega)| > 1$) or attenuation ($|H(\omega)| < 1$) imparted by the system on the input sinusoid. The phase $\Theta(\omega)$ determines the amount of phase shift imparted by the system on the input sinusoid. Consequently, by knowing $H(\omega)$, we are able to determine the response of the system to any sinusoidal input signal. Since $H(\omega)$ specifies the response of the system in the frequency domain, it is called the *frequency response* of the system. Correspondingly, $|H(\omega)|$ is called the *magnitude response* and $\Theta(\omega)$ is called the *phase response* of the system.

If the input to the system consists of more than one sinusoid, the superposition property of the linear system can be used to determine the response. The following examples illustrate the use of the superposition property.

EXAMPLE 5.1.3

Determine the response of the system in Example 5.1.1 to the input signal

$$x(n) = 10 - 5\sin\frac{\pi}{2}n + 20\cos\pi n, \qquad -\infty < n < \infty$$

Solution. The frequency response of the system is given in (5.1.7) as

$$H(\omega) = \frac{1}{1 - \frac{1}{2}e^{-j\omega}}$$

The first term in the input signal is a fixed signal component corresponding to $\omega = 0$. Thus

$$H(0) = \frac{1}{1 - \frac{1}{2}} = 2$$

The second term in $x(n)$ has a frequency $\pi/2$. At this frequency the frequency response of the system is

$$H\left(\frac{\pi}{2}\right) = \frac{2}{\sqrt{5}}e^{-j26.6^\circ}$$

Finally, the third term in $x(n)$ has a frequency $\omega = \pi$. At this frequency

$$H(\pi) = \frac{2}{3}$$

Hence the response of the system to $x(n)$ is

$$y(n) = 20 - \frac{10}{\sqrt{5}}\sin\left(\frac{\pi}{2}n - 26.6^\circ\right) + \frac{40}{3}\cos \pi n, \qquad -\infty < n < \infty$$

EXAMPLE 5.1.4

A linear time-invariant system is described by the following difference equation:

$$y(n) = ay(n-1) + bx(n), \qquad 0 < a < 1$$

(a) Determine the magnitude and phase of the frequency response $H(\omega)$ of the system.

(b) Choose the parameter b so that the maximum value of $|H(\omega)|$ is unity, and sketch $|H(\omega)|$ and $\measuredangle H(\omega)$ for $a = 0.9$.

(c) Determine the output of the system to the input signal

$$x(n) = 5 + 12\sin\frac{\pi}{2}n - 20\cos\left(\pi n + \frac{\pi}{4}\right)$$

Solution. The impulse response of the system is

$$h(n) = ba^n u(n)$$

Since $|a| < 1$, the system is BIBO stable and hence $H(\omega)$ exists.

(a) The frequency response is

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n}$$

$$= \frac{b}{1 - ae^{-j\omega}}$$

Since

$$1 - ae^{-j\omega} = (1 - a\cos\omega) + ja\sin\omega$$

it follows that

$$|1 - ae^{-j\omega}| = \sqrt{(1 - a\cos\omega)^2 + (a\sin\omega)^2}$$

$$= \sqrt{1 + a^2 - 2a\cos\omega}$$

and

$$\measuredangle(1 - ae^{-j\omega}) = \tan^{-1}\frac{a\sin\omega}{1 - a\cos\omega}$$

Therefore,

$$|H(\omega)| = \frac{|b|}{\sqrt{1 + a^2 - 2a\cos\omega}}$$

$$\measuredangle H(\omega) = \Theta(\omega) = \measuredangle b - \tan^{-1}\frac{a\sin\omega}{1 - a\cos\omega}$$

(b) Since the parameter a is positive, the denominator of $|H(\omega)|$ attains a minimum at $\omega = 0$. Therefore, $|H(\omega)|$ attains its maximum value at $\omega = 0$. At this frequency we have

$$|H(0)| = \frac{|b|}{1 - a} = 1$$

which implies that $b = \pm(1 - a)$. We choose $b = 1 - a$, so that

$$|H(\omega)| = \frac{1 - a}{\sqrt{1 + a^2 - 2a\cos\omega}}$$

and

$$\Theta(\omega) = -\tan^{-1}\frac{a\sin\omega}{1 - a\cos\omega}$$

The frequency response plots for $|H(\omega)|$ and $\Theta(\omega)$ are illustrated in Fig. 5.1.2. We observe that this system attenuates high-frequency signals.

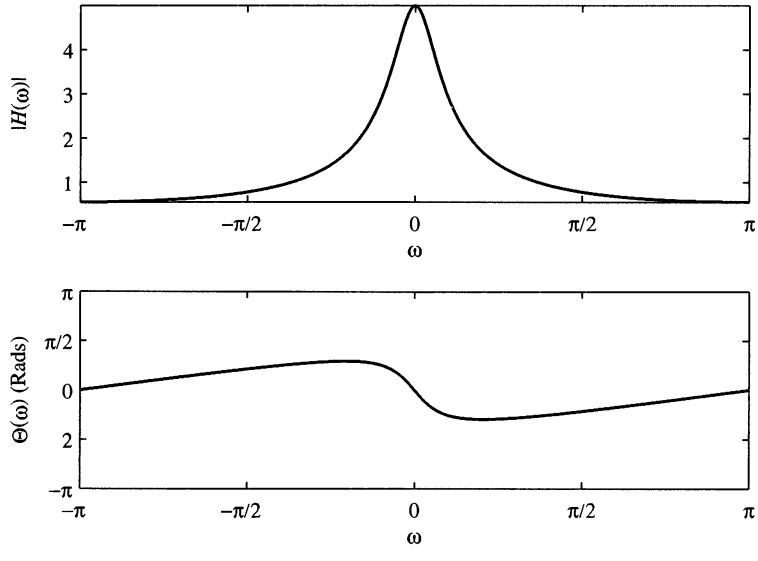


Figure 5.1.2 Magnitude and phase responses for the system in Example 5.1.4 with $a = 0.9$.

(c) The input signal consists of components at frequencies $\omega = 0$, $\pi/2$, and π. For $\omega = 0$, $|H(0)| = 1$ and $\Theta(0) = 0$. For $\omega = \pi/2$,

$$\left|H\left(\frac{\pi}{2}\right)\right| = \frac{1-a}{\sqrt{1+a^2}} = \frac{0.1}{\sqrt{1.81}} = 0.074$$

$$\Theta\left(\frac{\pi}{2}\right) = -\tan^{-1} a = -42^\circ$$

For $\omega = \pi$,

$$|H(\pi)| = \frac{1-a}{1+a} = \frac{0.1}{1.9} = 0.053$$

$$\Theta(\pi) = 0$$

Therefore, the output of the system is

$$\begin{aligned} y(n) &= 5|H(0)| + 12\left|H\left(\frac{\pi}{2}\right)\right| \sin\left[\frac{\pi}{2}n + \Theta\left(\frac{\pi}{2}\right)\right] \\ &\quad - 20|H(\pi)| \cos\left[\pi n + \frac{\pi}{4} + \Theta(\pi)\right] \\ &= 5 + 0.888 \sin\left(\frac{\pi}{2}n - 42^\circ\right) - 1.06 \cos\left(\pi n + \frac{\pi}{4}\right), \qquad -\infty < n < \infty \end{aligned}$$

In the most general case, if the input to the system consists of an arbitrary linear combination of sinusoids of the form

$$x(n) = \sum_{i=1}^{L} A_i \cos(\omega_i n + \phi_i), \qquad -\infty < n < \infty$$

where $\{A_i\}$ and $\{\phi_i\}$ are the amplitudes and phases of the corresponding sinusoidal components, then the response of the system is simply

$$y(n) = \sum_{i=1}^{L} A_i |H(\omega_i)| \cos[\omega_i n + \phi_i + \Theta(\omega_i)] \tag{5.1.19}$$

where $|H(\omega_i)|$ and $\Theta(\omega_i)$ are the magnitude and phase, respectively, imparted by the system to the individual frequency components of the input signal.

It is clear that depending on the frequency response $H(\omega)$ of the system, input sinusoids of different frequencies will be affected differently by the system. For example, some sinusoids may be completely suppressed by the system if $H(\omega) = 0$ at the frequencies of these sinusoids. Other sinusoids may receive no attenuation (or perhaps, some amplification) by the system. In effect, we can view the linear time-invariant system functioning as a *filter* to sinusoids of different frequencies, passing some of the frequency components to the output and suppressing or preventing other frequency components from reaching the output. In fact, as discussed in Chapter 10, the basic digital filter design problem involves determining the parameters of a linear time-invariant system to achieve a desired frequency response $H(\omega)$.

5.1.2 Steady-State and Transient Response to Sinusoidal Input Signals

In the discussion in the preceding section, we determined the response of a linear time-invariant system to exponential and sinusoidal input signals applied to the system at $n = -\infty$. We usually call such signals eternal exponentials or eternal sinusoids, because they were applied at $n = -\infty$. In such a case, the response that we observe at the output of the system is the steady-state response. There is no transient response in this case.

On the other hand, if the exponential or sinusoidal signal is applied at some finite time instant, say at $n = 0$, the response of the system consists of two terms, the transient response and the steady-state response. To demonstrate this behavior, let us consider, as an example, the system described by the first-order difference equation

$$y(n) = ay(n-1) + x(n) \tag{5.1.20}$$

This system was considered in Section 2.4.2. Its response to any input $x(n)$ applied at $n = 0$ is given by (2.4.8) as

$$y(n) = a^{n+1}y(-1) + \sum_{k=0}^{n} a^k x(n-k), \qquad n \geq 0 \tag{5.1.21}$$

where $y(-1)$ is the initial condition.

Now, let us assume that the input to the system is the complex exponential

$$x(n) = Ae^{j\omega n}, \qquad n \geq 0 \tag{5.1.22}$$

applied at $n = 0$. When we substitute (5.1.22) into (5.1.21), we obtain

$$\begin{aligned} y(n) &= a^{n+1}y(-1) + A\sum_{k=0}^{n} a^k e^{j\omega(n-k)} \\ &= a^{n+1}y(-1) + A\left[\sum_{k=0}^{n}(ae^{-j\omega})^k\right]e^{j\omega n} \\ &= a^{n+1}y(-1) + A\frac{1 - a^{n+1}e^{-j\omega(n+1)}}{1 - ae^{-j\omega}}e^{j\omega n}, \qquad n \geq 0 \\ &= a^{n+1}y(-1) - \frac{Aa^{n+1}e^{-j\omega(n+1)}}{1 - ae^{-j\omega}}e^{j\omega n} + \frac{A}{1 - ae^{-j\omega}}e^{j\omega n}, \qquad n \geq 0 \end{aligned} \tag{5.1.23}$$

We recall that the system in (5.1.20) is BIBO stable if $|a| < 1$. In this case the two terms involving a^{n+1} in (5.1.23) decay toward zero as n approaches infinity. Consequently, we are left with the steady-state response

$$\begin{aligned} y_{ss}(n) &= \lim_{n\to\infty} y(n) = \frac{A}{1 - ae^{-j\omega}}e^{j\omega n} \\ &= AH(\omega)e^{j\omega n} \end{aligned} \tag{5.1.24}$$

The first two terms in (5.1.23) constitute the transient response of the system, that is,

$$y_{\text{tr}}(n) = a^{n+1}y(-1) - \frac{Aa^{n+1}e^{-j\omega(n+1)}}{1 - ae^{-j\omega}}e^{j\omega n}, \qquad n \geq 0 \tag{5.1.25}$$

which decay toward zero as n approaches infinity. The first term in the transient response is the zero-input response of the system and the second term is the transient produced by the exponential input signal.

In general, all linear time-invariant BIBO systems behave in a similar fashion when excited by a complex exponential, or by a sinusoid at $n = 0$ or at some other finite time instant. That is, the transient response decays toward zero as $n \to \infty$, leaving only the steady-state response that we determined in the preceding section. In many practical applications, the transient response of the system is unimportant, and therefore it is usually ignored in dealing with the response of the system to sinusoidal inputs.

5.1.3 Steady-State Response to Periodic Input Signals

Suppose that the input to a stable linear time-invariant system is a periodic signal $x(n)$ with fundamental period N. Since such a signal exists from $-\infty < n < \infty$, the total response of the system at any time instant n is simply equal to the steady-state response.

To determine the response $y(n)$ of the system, we make use of the Fourier series representation of the periodic signal, which is

$$x(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1 \tag{5.1.26}$$

where the $\{c_k\}$ are the Fourier series coefficients. Now the response of the system to the complex exponential signal

$$x_k(n) = c_k e^{j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1$$

is

$$y_k(n) = c_k H\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1 \tag{5.1.27}$$

where

$$H\left(\frac{2\pi k}{N}\right) = H(\omega)|_{\omega=2\pi k/N}, \qquad k = 0, 1, \ldots, N-1$$

By using the superposition principle for linear systems, we obtain the response of the system to the periodic signal $x(n)$ in (5.1.26) as

$$y(n) = \sum_{k=0}^{N-1} c_k H\left(\frac{2\pi k}{N}\right) e^{j2\pi kn/N}, \qquad -\infty < n < \infty \tag{5.1.28}$$

This result implies that the response of the system to the periodic input signal $x(n)$ is also periodic with the same period N. The Fourier series coefficients for $y(n)$ are

$$d_k \equiv c_k H\left(\frac{2\pi k}{N}\right), \qquad k = 0, 1, \ldots, N-1 \tag{5.1.29}$$

Hence, the linear system can change the shape of the periodic input signal by scaling the amplitude and shifting the phase of the Fourier series components, but it does not affect the period of the periodic input signal.

5.1.4 Response to Aperiodic Input Signals

The convolution theorem, given in (4.4.49), provides the desired frequency-domain relationship for determining the output of an LTI system to an aperiodic finite-energy signal. If $\{x(n)\}$ denotes the input sequence, $\{y(n)\}$ denotes the output sequence, and $\{h(n)\}$ denotes the unit sample response of the system, then from the convolution theorem, we have

$$Y(\omega) = H(\omega)X(\omega) \tag{5.1.30}$$

where $Y(\omega)$, $X(\omega)$, and $H(\omega)$ are the corresponding Fourier transforms of $\{y(n)\}$, $\{x(n)\}$, and $\{h(n)\}$, respectively. From this relationship we observe that the spectrum of the output signal is equal to the spectrum of the input signal multiplied by the frequency response of the system.

If we express $Y(\omega)$, $H(\omega)$, and $X(\omega)$ in polar form, the magnitude and phase of the output signal can be expressed as

$$|Y(\omega)| = |H(\omega)||X(\omega)| \tag{5.1.31}$$

$$\measuredangle Y(\omega) = \measuredangle X(\omega) + \measuredangle H(\omega) \tag{5.1.32}$$

where $|H(\omega)|$ and $\measuredangle H(\omega)$ are the magnitude and phase responses of the system.

By its very nature, a finite-energy aperiodic signal contains a continuum of frequency components. The linear time-invariant system, through its frequency response function, attenuates some frequency components of the input signal and amplifies other frequency components. Thus the system acts as a *filter* to the input signal. Observation of the graph of $|H(\omega)|$ shows which frequency components are amplified and which are attenuated. On the other hand, the angle of $H(\omega)$ determines the phase shift imparted in the continuum of frequency components of the input signal as a function of frequency. If the input signal spectrum is changed by the system in an undesirable way, we say that the system has caused *magnitude and phase distortion.*

We also observe that *the output of a linear time-invariant system cannot contain frequency components that are not contained in the input signal.* It takes either a linear time-variant system or a nonlinear system to create frequency components that are not necessarily contained in the input signal.

Figure 5.1.3 illustrates the time-domain and frequency-domain relationships that can be used in the analysis of BIBO-stable LTI systems. We observe that in time-domain analysis, we deal with the convolution of the input signal with the impulse

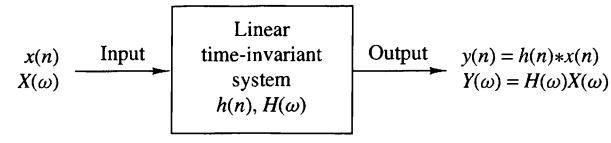


Figure 5.1.3
Time- and frequency-domain input–output relationships in LTI systems.

response of the system to obtain the output sequence of the system. On the other hand, in frequency-domain analysis, we deal with the input signal spectrum $X(\omega)$ and the frequency response $H(\omega)$ of the system, which are related through multiplication, to yield the spectrum of the signal at the output of the system.

We can use the relation in (5.1.30) to determine the spectrum $Y(\omega)$ of the output signal. Then the output sequence $\{y(n)\}$ can be determined from the inverse Fourier transform

$$y(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} Y(\omega)e^{j\omega n}d\omega \tag{5.1.33}$$

However, this method is seldom used. Instead, the z-transform introduced in Chapter 3 is a simpler method for solving the problem of determining the output sequence $\{y(n)\}$.

Let us return to the basic input–output relation in (5.1.30) and compute the squared magnitude of both sides. Thus we obtain

$$\begin{aligned} |Y(\omega)|^2 &= |H(\omega)|^2|X(\omega)|^2 \\ S_{yy}(\omega) &= |H(\omega)|^2 S_{xx}(\omega) \end{aligned} \tag{5.1.34}$$

where $S_{xx}(\omega)$ and $S_{yy}(\omega)$ are the energy density spectra of the input and output signals, respectively. By integrating (5.1.34) over the frequency range $(-\pi, \pi)$, we obtain the energy of the output signal as

$$\begin{aligned} E_y &= \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{yy}(\omega)d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2 S_{xx}(\omega)d\omega \end{aligned} \tag{5.1.35}$$

EXAMPLE 5.1.5

A linear time-invariant system is characterized by its impulse response

$$h(n) = (\frac{1}{2})^n u(n)$$

Determine the spectrum and the energy density spectrum of the output signal when the system is excited by the signal

$$x(n) = (\frac{1}{4})^n u(n)$$

Solution. The frequency response function of the system

$$H(\omega) = \sum_{n=0}^{\infty} (\frac{1}{2})^n e^{-j\omega n}$$

$$= \frac{1}{1 - \frac{1}{2}e^{-j\omega}}$$

Similarly, the input sequence $\{x(n)\}$ has a Fourier transform

$$X(\omega) = \frac{1}{1 - \frac{1}{4}e^{-j\omega}}$$

Hence the spectrum of the signal at the output of the system is

$$Y(\omega) = H(\omega)X(\omega)$$

$$= \frac{1}{(1 - \frac{1}{2}e^{-j\omega})(1 - \frac{1}{4}e^{-j\omega})}$$

The corresponding energy density spectrum is

$$S_{yy}(\omega) = |Y(\omega)|^2 = |H(\omega)|^2 |X(\omega)|^2$$

$$= \frac{1}{(\frac{5}{4} - \cos\omega)(\frac{17}{16} - \frac{1}{2}\cos\omega)}$$

5.2 Frequency Response of LTI Systems

In this section we focus on determining the frequency response of LTI systems that have rational system functions. Recall that this class of LTI systems is described in the time domain by constant-coefficient difference equations.

5.2.1 Frequency Response of a System with a Rational System Function

From the discussion in Section 4.2.6 we know that if the system function $H(z)$ converges on the unit circle, we can obtain the frequency response of the system by evaluating $H(z)$ on the unit circle. Thus

$$H(\omega) = H(z)|_{z=e^{j\omega}} = \sum_{n=-\infty}^{\infty} h(n)e^{-j\omega n} \tag{5.2.1}$$

In the case where $H(z)$ is a rational function of the form $H(z) = B(z)/A(z)$, we have

$$H(\omega) = \frac{B(\omega)}{A(\omega)} = \frac{\sum_{k=0}^{M} b_k e^{-j\omega k}}{1 + \sum_{k=1}^{N} a_k e^{-j\omega k}} \tag{5.2.2}$$

$$= b_0 \frac{\prod_{k=1}^{M}(1 - z_k e^{-j\omega})}{\prod_{k=1}^{N}(1 - p_k e^{-j\omega})} \tag{5.2.3}$$

where the $\{a_k\}$ and $\{b_k\}$ are real, but $\{z_k\}$ and $\{p_k\}$ may be complex valued.

It is sometimes desirable to express the magnitude squared of $H(\omega)$ in terms of $H(z)$. First, we note that

$$|H(\omega)|^2 = H(\omega)H^*(\omega)$$

For the rational system function given by (5.2.3), we have

$$H^*(\omega) = b_0 \frac{\prod_{k=1}^{M}(1 - z_k^* e^{j\omega})}{\prod_{k=1}^{N}(1 - p_k^* e^{j\omega})} \tag{5.2.4}$$

It follows that $H^*(\omega)$ is obtained by evaluating $H^*(1/z^*)$ on the unit circle, where for a rational system function,

$$H^*(1/z^*) = b_0 \frac{\prod_{k=1}^{M}(1 - z_k^* z)}{\prod_{k=1}^{N}(1 - p_k^* z)} \tag{5.2.5}$$

However, when $\{h(n)\}$ is real or, equivalently, the coefficients $\{a_k\}$ and $\{b_k\}$ are real, complex-valued poles and zeros occur in complex-conjugate pairs. In this case, $H^*(1/z^*) = H(z^{-1})$. Consequently, $H^*(\omega) = H(-\omega)$, and

$$|H(\omega)|^2 = H(\omega)H^*(\omega) = H(\omega)H(-\omega) = H(z)H(z^{-1})|_{z=e^{j\omega}} \tag{5.2.6}$$

According to the correlation theorem for the z-transform (see Table 3.2), the function $H(z)H(z^{-1})$ is the z-transform of the autocorrelation sequence $\{r_{hh}(m)\}$

of the unit sample response $\{h(n)\}$. Then it follows from the Wiener–Khintchine theorem that $|H(\omega)|^2$ is the Fourier transform of $\{r_{hh}(m)\}$.

Similarly, if $H(z) = B(z)/A(z)$, the transforms $D(z) = B(z)B(z^{-1})$ and $C(z) = A(z)A(z^{-1})$ are the z-transforms of the autocorrelation sequences $\{c_l\}$ and $\{d_l\}$, where

$$c_l = \sum_{k=0}^{N-|l|} a_k a_{k+l}, \qquad -N \le l \le N \tag{5.2.7}$$

$$d_l = \sum_{k=0}^{M-|l|} b_k b_{k+l}, \qquad -M \le l \le M \tag{5.2.8}$$

Since the system parameters $\{a_k\}$ and $\{b_k\}$ are real valued, it follows that $c_l = c_{-l}$ and $d_l = d_{-l}$. By using this symmetry property, $|H(\omega)|^2$ may be expressed as

$$|H(\omega)|^2 = \frac{d_0 + 2\sum_{k=1}^{M} d_k \cos k\omega}{c_0 + 2\sum_{k=1}^{N} c_k \cos k\omega} \tag{5.2.9}$$

Finally, we note that $\cos k\omega$ can be expressed as a polynomial function of $\cos\omega$. That is,

$$\cos k\omega = \sum_{m=0}^{k} \beta_m (\cos\omega)^m \tag{5.2.10}$$

where $\{\beta_m\}$ are the coefficients in the expansion. Consequently, the numerator and denominator of $|H(\omega)|^2$ can be viewed as polynomial functions of $\cos\omega$. The following example illustrates the foregoing relationships.

EXAMPLE 5.2.1

Determine $|H(\omega)|^2$ for the system

$$y(n) = -0.1y(n-1) + 0.2y(n-2) + x(n) + x(n-1)$$

Solution. The system function is

$$H(z) = \frac{1 + z^{-1}}{1 + 0.1z^{-1} - 0.2z^{-2}}$$

and its ROC is $|z| > 0.5$. Hence $H(\omega)$ exists. Now

$$H(z)H(z^{-1}) = \frac{1 + z^{-1}}{1 + 0.1z^{-1} - 0.2z^{-2}} \cdot \frac{1 + z}{1 + 0.1z - 0.2z^2}$$

$$= \frac{2 + z + z^{-1}}{1.05 + 0.08(z + z^{-1}) - 0.2(z^{-2} + z^{-2})}$$

By evaluating $H(z)H(z^{-1})$ on the unit circle, we obtain

$$|H(\omega)|^2 = \frac{2 + 2\cos\omega}{1.05 + 0.16\cos\omega - 0.4\cos 2\omega}$$

However, $\cos 2\omega = 2\cos^2\omega - 1$. Consequently, $|H(\omega)|^2$ may be expressed as

$$|H(\omega)|^2 = \frac{2(1 + \cos\omega)}{1.45 + 0.16\cos\omega - 0.8\cos^2\omega}$$

We note that given $H(z)$, it is straightforward to determine $H(z^{-1})$ and then $|H(\omega)|^2$. However, the inverse problem of determining $H(z)$, given $|H(\omega)|^2$ or the corresponding impulse response $\{h(n)\}$, is not straightforward. Since $|H(\omega)|^2$ does not contain the phase information in $H(\omega)$, it is not possible to uniquely determine $H(z)$.

To elaborate on the point, let us assume that the N poles and M zeros of $H(z)$ are $\{p_k\}$ and $\{z_k\}$, respectively. The corresponding poles and zeros of $H(z^{-1})$ are $\{1/p_k\}$ and $\{1/z_k\}$, respectively. Given $|H(\omega)|^2$ or, equivalently, $H(z)H(z^{-1})$, we can determine different system functions $H(z)$ by assigning to $H(z)$, a pole p_k or its reciprocal $1/p_k$, and a zero z_k or its reciprocal $1/z_k$. For example, if $N = 2$ and $M = 1$, the poles and zeros of $H(z)H(z^{-1})$ are $\{p_1, p_2, 1/p_1, 1/p_2\}$ and $\{z_1, 1/z_1\}$. If p_1 and p_2 are real, the possible poles for $H(z)$ are $\{p_1, p_2\}$, $\{1/p_1, 1/p_2\}$, $\{p_1, 1/p_2\}$, and $\{p_2, 1/p_1\}$ and the possible zeros are $\{z_1\}$ or $\{1/z_1\}$. Therefore, there are eight possible choices of system functions, all of which result in the same $|H(\omega)|^2$. Even if we restrict the poles of $H(z)$ to be inside the unit circle, there are still two different choices for $H(z)$, depending on whether we pick the zero $\{z_1\}$ or $\{1/z_1\}$. Therefore, we cannot determine $H(z)$ uniquely given only the magnitude response $|H(\omega)|$.

5.2.2 Computation of the Frequency Response Function

In evaluating the magnitude response and the phase response as functions of frequency, it is convenient to express $H(\omega)$ in terms of its poles and zeros. Hence we write $H(\omega)$ in factored form as

$$H(\omega) = b_0 \frac{\prod_{k=1}^{M}(1 - z_k e^{-j\omega k})}{\prod_{k=1}^{N}(1 - p_k e^{-j\omega k})} \tag{5.2.11}$$

or, equivalently, as

$$H(\omega) = b_0 e^{j\omega(N-M)} \frac{\prod_{k=1}^{M}(e^{j\omega} - z_k)}{\prod_{k=1}^{N}(e^{j\omega} - p_k)} \tag{5.2.12}$$

Let us express the complex-valued factors in (5.2.12) in polar form as

$$e^{j\omega} - z_k = V_k(\omega)e^{j\Theta_k(\omega)} \tag{5.2.13}$$

and

$$e^{j\omega} - p_k = U_k(\omega)e^{j\Phi_k(\omega)} \tag{5.2.14}$$

where

$$V_k(\omega) \equiv |e^{j\omega} - z_k|, \quad \Theta_k(\omega) \equiv \measuredangle(e^{j\omega} - z_k) \tag{5.2.15}$$

and

$$U_k(\omega) \equiv |e^{j\omega} - p_k|, \quad \Phi_k(\omega) = \measuredangle(e^{j\omega} - p_k) \tag{5.2.16}$$

The magnitude of $H(\omega)$ is equal to the product of magnitudes of all terms in (5.2.12). Thus, using (5.2.13) through (5.2.16), we obtain

$$|H(\omega)| = |b_0|\frac{V_1(\omega)\cdots V_M(\omega)}{U_1(\omega)U_2(\omega)\cdots U_N(\omega)} \tag{5.2.17}$$

since the magnitude of $e^{j\omega(N-M)}$ is 1.

The phase of $H(\omega)$ is the sum of the phases of the numerator factors, minus the phases of the denominator factors. Thus, by combining (5.2.13) through (5.2.16), we have

$$\begin{aligned}\measuredangle H(\omega) = \measuredangle b_0 + \omega(N-M) + \Theta_1(\omega) + \Theta_2(\omega) + \cdots + \Theta_M(\omega) \\ - [\Phi_1(\omega) + \Phi_2(\omega) + \cdots + \Phi_N(\omega)]\end{aligned} \tag{5.2.18}$$

The phase of the gain term b_0 is zero or π, depending on whether b_0 is positive or negative. Clearly, if we know the zeros and the poles of the system function $H(z)$, we can evaluate the frequency response from (5.2.17) and (5.2.18).

There is a geometric interpretation of the quantities appearing in (5.2.17) and (5.2.18). Let us consider a pole p_k and a zero z_k located at points A and B of the z-plane, as shown in Fig. 5.2.1(a). Assume that we wish to compute $H(\omega)$ at a specific value of frequency ω. The given value of ω determines the angle of $e^{j\omega}$ with the positive real axis. The tip of the vector $e^{j\omega}$ specifies a point L on the unit circle. The evaluation of the Fourier transform for the given value of ω is equivalent to evaluating the z-transform at the point L of the complex plane. Let us draw the vectors **AL** and **BL** from the pole and zero locations to the point L, at which we wish to compute the Fourier transform. From Fig. 5.2.1(a) it follows that

$$\mathbf{CL} = \mathbf{CA} + \mathbf{AL}$$

and

$$\mathbf{CL} = \mathbf{CB} + \mathbf{BL}$$

However, $\mathbf{CL} = e^{j\omega}$, $\mathbf{CA} = p_k$ and $\mathbf{CB} = z_k$. Thus

$$\mathbf{AL} = e^{j\omega} - p_k \tag{5.2.19}$$

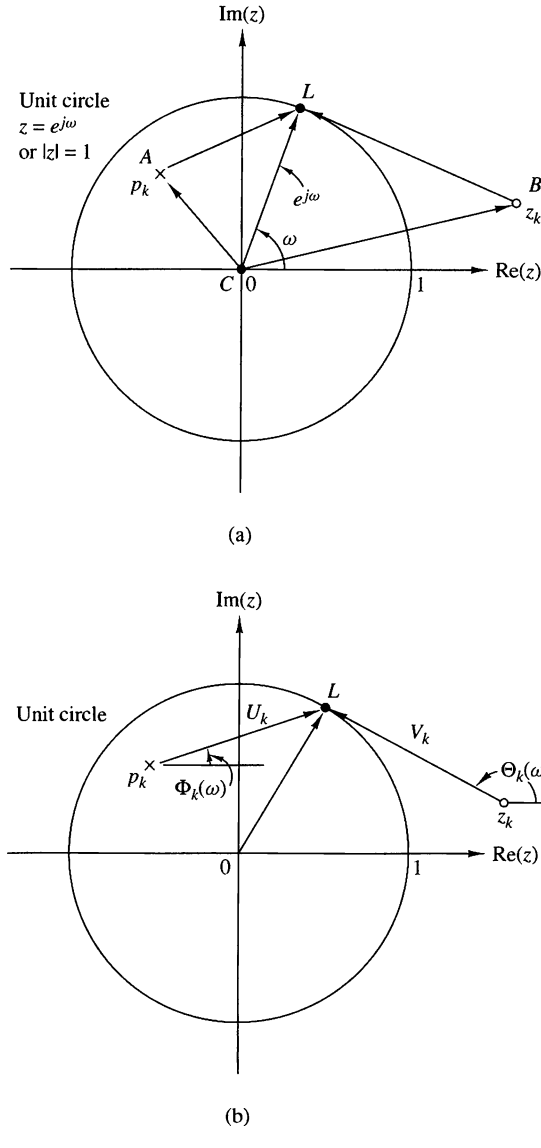


Figure 5.2.1
Geometric interpretation of the contribution of a pole and a zero to the Fourier transform (1) magnitude: the factor V_k/U_k, (2) phase: the factor $\Theta_k - \Phi_k$.

and

$$\mathbf{BL} = e^{j\omega} - z_k \tag{5.2.20}$$

By combining these relations with (5.2.13) and (5.2.14), we obtain

$$\mathbf{AL} = e^{j\omega} - p_k = U_k(\omega)e^{j\Phi_k(\omega)} \tag{5.2.21}$$

$$\mathbf{BL} = e^{j\omega} - z_k = V_k(\omega)e^{j\Theta_k(\omega)} \tag{5.2.22}$$

Thus $U_k(\omega)$ is the length of **AL**, that is, the distance of the pole p_k from the point L corresponding to $e^{j\omega}$, whereas $V_k(\omega)$ is the distance of the zero z_k from the same

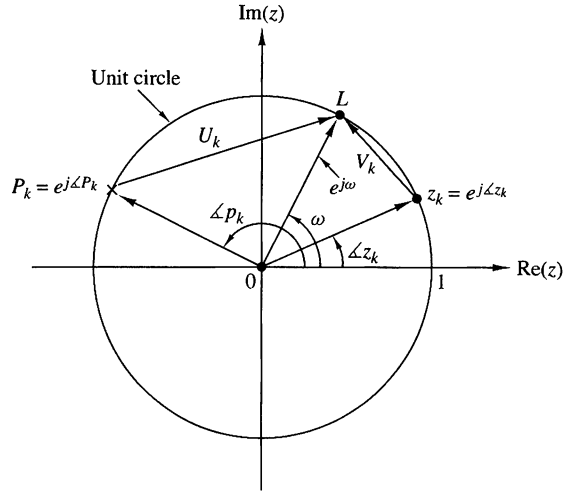


Figure 5.2.2
A zero on the unit circle causes $|H(\omega)| = 0$ and $\omega = \measuredangle z_k$. In contrast, a pole on the unit circle results in $|H(\omega)| = \infty$ at $\omega = \measuredangle p_k$.

point L. The phases $\Phi_k(\omega)$ and $\Theta_k(\omega)$ are the angles of the vectors **AL** and **BL** with the positive real axis, respectively. These geometric interpretations are shown in Fig. 5.2.1(b).

Geometric interpretations are very useful in understanding how the location of poles and zeros affects the magnitude and phase of the Fourier transform. Suppose that a zero, say z_k, and a pole, say p_k, are on the unit circle as shown in Fig. 5.2.2. We note that at $\omega = \measuredangle z_k$, $V_k(\omega)$ and consequently $|H(\omega)|$ become zero. Similarly, at $\omega = \measuredangle p_k$ the length $U_k(\omega)$ becomes zero and hence $|H(\omega)|$ becomes infinite. Clearly, the evaluation of phase in these cases has no meaning.

From this discussion we can easily see that the presence of a zero close to the unit circle causes the magnitude of the frequency response, at frequencies that correspond to points of the unit circle close to that point, to be small. In contrast, the presence of a pole close to the unit circle causes the magnitude of the frequency response to be large at frequencies close to that point. Thus poles have the opposite effect of zeros. Also, placing a zero close to a pole cancels the effect of the pole, and vice versa. This can be also seen from (5.2.12), since if $z_k = p_k$, the terms $e^{j\omega} - z_k$ and $e^{j\omega} - p_k$ cancel. Obviously, the presence of both poles and zeros in a transform results in a greater variety of shapes for $|H(\omega)|$ and $\measuredangle H(\omega)$. This observation is very important in the design of digital filters. We conclude our discussion with the following example illustrating these concepts.

EXAMPLE 5.2.2

Evaluate the frequency response of the system described by the system function

$$H(z) = \frac{1}{1 - 0.8z^{-1}} = \frac{z}{z - 0.8}$$

Solution. Clearly, $H(z)$ has a zero at $z = 0$ and a pole at $p = 0.8$. Hence the frequency response of the system is

$$H(\omega) = \frac{e^{j\omega}}{e^{j\omega} - 0.8}$$

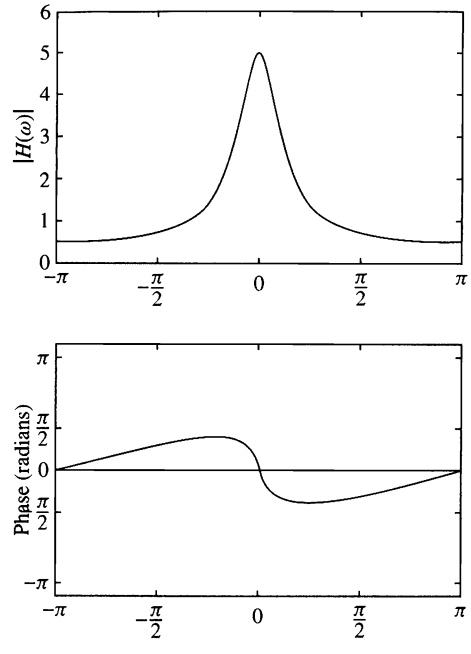


Figure 5.2.3
Magnitude and phase of system with $H(z) = 1/(1 - 0.8z^{-1})$.

The magnitude response is

$$|H(\omega)| = \frac{|e^{j\omega}|}{|e^{j\omega} - 0.8|} = \frac{1}{\sqrt{1.64 - 1.6\cos\omega}}$$

and the phase response is

$$\theta(\omega) = \omega - \tan^{-1}\frac{\sin\omega}{\cos\omega - 0.8}$$

The magnitude and phase responses are illustrated in Fig. 5.2.3. Note that the peak of the magnitude response occurs at $\omega = 0$, the point on the unit circle closest to the pole located at 0.8.

If the magnitude response in (5.2.17) is expressed in decibels,

$$|H(\omega)|_{dB} = 20\log_{10}|b_0| + 20\sum_{k=1}^{M}\log_{10}V_k(\omega) - 20\sum_{k=1}^{N}\log_{10}U_k(\omega) \qquad (5.2.23)$$

Thus the magnitude response is expressed as a sum of the magnitude factors in $|H(\omega)|$.

Correlation Functions and Spectra at the Output of LTI Systems

In this section, we derive the spectral relationships between the input and output signals of LTI systems. Section 5.3.1 describes the relationships for the energy density spectra of deterministic input and output signals. Section 5.3.2 is focused on the relationships for the power density spectra of random input and output signals.

5.3.1 Input–Output Correlation Functions and Spectra

In Section 2.6.4 we developed several correlation relationships between the input and the output sequences of an LTI system. Specifically, we derived the following equations:

$$r_{yy}(m) = r_{hh}(m) * r_{xx}(m) \tag{5.3.1}$$

$$r_{yx}(m) = h(m) * r_{xx}(m) \tag{5.3.2}$$

where $r_{xx}(m)$ is the autocorrelation sequence of the input signal $\{x(n)\}$, $r_{yy}(m)$ is the autocorrelation sequence of the output $\{y(n)\}$, $r_{hh}(m)$ is the autocorrelation sequence of the impulse response $\{h(n)\}$, and $r_{yx}(m)$ is the crosscorrelation sequence between the output and the input signals. Since (5.3.1) and (5.3.2) involve the convolution operation, the z-transform of these equations yields

$$\begin{aligned} S_{yy}(z) &= S_{hh}(z)S_{xx}(z) \\ &= H(z)H(z^{-1})S_{xx}(z) \end{aligned} \tag{5.3.3}$$

$$S_{yx}(z) = H(z)S_{xx}(z) \tag{5.3.4}$$

If we substitute $z = e^{j\omega}$ in (5.3.4), we obtain

$$\begin{aligned} S_{yx}(\omega) &= H(\omega)S_{xx}(\omega) \\ &= H(\omega)|X(\omega)|^2 \end{aligned} \tag{5.3.5}$$

where $S_{yx}(\omega)$ is the cross-energy density spectrum of $\{y(n)\}$ and $\{x(n)\}$. Similarly, evaluating $S_{yy}(z)$ on the unit circle yields the energy density spectrum of the output signal as

$$S_{yy}(\omega) = |H(\omega)|^2 S_{xx}(\omega) \tag{5.3.6}$$

where $S_{xx}(\omega)$ is the energy density spectrum of the input signal.

Since $r_{yy}(m)$ and $S_{yy}(\omega)$ are a Fourier transform pair, it follows that

$$r_{yy}(m) = \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{yy}(\omega)e^{j\omega m}d\omega \tag{5.3.7}$$

The total energy in the output signal is simply

$$\begin{aligned} E_y &= \frac{1}{2\pi}\int_{-\pi}^{\pi} S_{yy}(\omega)d\omega = r_{yy}(0) \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} |H(\omega)|^2\, S_{xx}(\omega)d\omega \end{aligned} \tag{5.3.8}$$

The result in (5.3.8) may be used to easily prove that $E_y \geq 0$.

Finally, we note that if the input signal has a flat spectrum [i.e., $S_{xx}(\omega) = S_x =$ constant for $\pi \leq \omega \leq -\pi$], (5.3.5) reduces to

$$S_{yx}(\omega) = H(\omega)S_x \tag{5.3.9}$$

where S_x is the constant value of the spectrum. Hence

$$H(\omega) = \frac{1}{S_x} S_{yx}(\omega) \tag{5.3.10}$$

or, equivalently,

$$h(n) = \frac{1}{S_x} r_{yx}(m) \tag{5.3.11}$$

The relation in (5.3.11) implies that $h(n)$ can be determined by exciting the input to the system by a spectrally flat signal $\{x(n)\}$, and crosscorrelating the input with the output of the system. This method is useful in measuring the impulse response of an unknown system.

5.3.2 Correlation Functions and Power Spectra for Random Input Signals

This development parallels the derivations in Section 5.3.1, with the exception that we now deal with the statistical mean and autocorrelation of the input and output signals of an LTI system.

Let us consider a discrete-time linear time-invariant system with unit sample response $\{h(n)\}$ and frequency response $H(f)$. For this development we assume that $\{h(n)\}$ is real. Let $x(n)$ be a sample function of a stationary random process $X(n)$ that excites the system and let $y(n)$ denote the response of the system to $x(n)$.

From the convolution summation that relates the output to the input we have

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{5.3.12}$$

Since $x(n)$ is a random input signal, the output is also a random sequence. In other words, for each sample sequence $x(n)$ of the process $X(n)$, there is a corresponding sample sequence $y(n)$ of the output random process $Y(n)$. We wish to relate the statistical characteristics of the output random process $Y(n)$ to the statistical characterization of the input process and the characteristics of the system.

The expected value of the output $y(n)$ is

$$\begin{aligned} m_y \equiv E[y(n)] &= E\Big[\sum_{k=-\infty}^{\infty} h(k)x(n-k)\Big] \\ &= \sum_{k=-\infty}^{\infty} h(k)E[x(n-k)] \\ m_y &= m_x \sum_{k=-\infty}^{\infty} h(k) \end{aligned} \tag{5.3.13}$$

From the Fourier transform relationship

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)e^{-j\omega k} \tag{5.3.14}$$

we have

$$H(0) = \sum_{k=-\infty}^{\infty} h(k) \tag{5.3.15}$$

which is the dc gain of the system. The relationship in (5.3.15) allows us to express the mean value in (5.3.13) as

$$m_y = m_x H(0) \tag{5.3.16}$$

The autocorrelation sequence for the output random process is defined as

$$\begin{aligned}
\gamma_{yy}(m) &= E[y^*(n)y(n+m)] \\
&= E\left[\sum_{k=-\infty}^{\infty} h(k)x^*(n-k) \sum_{j=-\infty}^{\infty} h(j)x(n+m-j)\right] \\
&= \sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} h(k)h(j)E[x^*(n-k)x(n+m-j)] \\
&= \sum_{k=-\infty}^{\infty}\sum_{j=-\infty}^{\infty} h(k)h(j)\gamma_{xx}(k-j+m)
\end{aligned} \tag{5.3.17}$$

This is the general form for the autocorrelation of the output in terms of the autocorrelation of the input and the impulse response of the system.

A special form of (5.3.17) is obtained when the input random process is white, that is, when $m_x = 0$ and

$$\gamma_{xx}(m) = \sigma_x^2 \delta(m) \tag{5.3.18}$$

where $\sigma_x^2 \equiv \gamma_{xx}(0)$ is the input signal power. Then (5.3.17) reduces to

$$\gamma_{yy}(m) = \sigma_x^2 \sum_{k=-\infty}^{\infty} h(k)h(k+m) \tag{5.3.19}$$

Under this condition the output process has the average power

$$\gamma_{yy}(0) = \sigma_x^2 \sum_{n=-\infty}^{\infty} h^2(n) = \sigma_x^2 \int_{-1/2}^{1/2} |H(f)|^2 df \tag{5.3.20}$$

where we have applied Parseval's theorem.

The relationship in (5.3.17) can be transformed into the frequency domain by determining the power density spectrum of $\gamma_{yy}(m)$. We have

$$\begin{aligned}
\Gamma_{yy}(\omega) &= \sum_{m=-\infty}^{\infty} \gamma_{yy}(m) e^{-j\omega m} \\
&= \sum_{m=-\infty}^{\infty} \left[\sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h(k)h(l)\gamma_{xx}(k-l+m) \right] e^{-j\omega m} \\
&= \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h(k)h(l) \left[\sum_{m=-\infty}^{\infty} \gamma_{xx}(k-l+m) e^{-j\omega m} \right] \\
&= \Gamma_{xx}(f) \left[\sum_{k=-\infty}^{\infty} h(k) e^{j\omega k} \right] \left[\sum_{l=-\infty}^{\infty} h(l) e^{-j\omega l} \right] \\
&= |H(\omega)|^2 \Gamma_{xx}(\omega)
\end{aligned} \tag{5.3.21}$$

This is the desired relationship for the power density spectrum of the output process, in terms of the power density spectrum of the input process and the frequency response of the system.

The equivalent expression for continuous-time systems with random inputs is

$$\Gamma_{yy}(F) = |H(F)|^2 \Gamma_{xx}(F) \tag{5.3.22}$$

where the power density spectra $\Gamma_{yy}(F)$ and $\Gamma_{xx}(F)$ are the Fourier transforms of the autocorrelation functions $\gamma_{yy}(\tau)$ and $\gamma_{xx}(\tau)$, respectively, and where $H(F)$ is the frequency response of the system, which is related to the impulse response by the Fourier transform, that is,

$$H(F) = \int_{-\infty}^{\infty} h(t) e^{-j2\pi Ft}\, dt \tag{5.3.23}$$

As a final exercise, we determine the crosscorrelation of the output $y(n)$ with the input signal $x(n)$. If we multiply both sides of (5.3.12) by $x^*(n-m)$ and take the expected value, we obtain

$$\begin{aligned}
E[y(n)x^*(n-m)] &= E\left[\sum_{k=-\infty}^{\infty} h(k)x^*(n-m)x(n-k) \right] \\
\gamma_{yx}(m) &= \sum_{k=-\infty}^{\infty} h(k) E[x^*(n-m)x(n-k)] \\
&= \sum_{k=-\infty}^{\infty} h(k)\gamma_{xx}(m-k)
\end{aligned} \tag{5.3.24}$$

Since (5.3.24) has the form of a convolution, the frequency-domain equivalent expression is

$$\Gamma_{yx}(\omega) = H(\omega)\Gamma_{xx}(\omega) \tag{5.3.25}$$

In the special case where $x(n)$ is white noise, (5.3.25) reduces to

$$\Gamma_{yx}(\omega) = \sigma_x^2 H(\omega) \tag{5.3.26}$$

where σ_x^2 is the input noise power. This result means that an unknown system with frequency response $H(\omega)$ can be identified by exciting the input with white noise, crosscorrelating the input sequence with the output sequence to obtain $\gamma_{yx}(m)$, and finally, computing the Fourier transform of $\gamma_{yx}(m)$. The result of these computations is proportional to $H(\omega)$.

5.4 Linear Time-Invariant Systems as Frequency-Selective Filters

The term *filter* is commonly used to describe a device that discriminates, according to some attribute of the objects applied at its input, what passes through it. For example, an air filter allows air to pass through it but prevents dust particles that are present in the air from passing through. An oil filter performs a similar function, with the exception that oil is the substance allowed to pass through the filter, while particles of dirt are collected at the input to the filter and prevented from passing through. In photography, an ultraviolet filter is often used to prevent ultraviolet light, which is present in sunlight and which is not a part of visible light, from passing through and affecting the chemicals on the film.

As we have observed in the preceding section, a linear time-invariant system also performs a type of discrimination or filtering among the various frequency components at its input. The nature of this filtering action is determined by the frequency response characteristics $H(\omega)$, which in turn depends on the choice of the system parameters (e.g., the coefficients $\{a_k\}$ and $\{b_k\}$ in the difference equation characterization of the system). Thus, by proper selection of the coefficients, we can design frequency-selective filters that pass signals with frequency components in some bands while they attenuate signals containing frequency components in other frequency bands.

In general, a linear time-invariant system modifies the input signal spectrum $X(\omega)$ according to its frequency response $H(\omega)$ to yield an output signal with spectrum $Y(\omega) = H(\omega)X(\omega)$. In a sense, $H(\omega)$ acts as a *weighting function* or a *spectral shaping function* to the different frequency components in the input signal. When viewed in this context, any linear time-invariant system can be considered to be a frequency-shaping filter, even though it may not necessarily completely block any or all frequency components. Consequently, the terms "linear time-invariant system" and "filter" are synonymous and are often used interchangeably.

We use the term *filter* to describe a linear time-invariant system used to perform spectral shaping or frequency-selective filtering. Filtering is used in digital signal processing in a variety of ways, such as removal of undesirable noise from desired signals, spectral shaping such as equalization of communication channels, signal detection in radar, sonar, and communications, and for performing spectral analysis of signals, and so on.

5.4.1 Ideal Filter Characteristics

Filters are usually classified according to their frequency-domain characteristics as lowpass, highpass, bandpass, and bandstop or band-elimination filters. The ideal magnitude response characteristics of these types of filters are illustrated in Fig. 5.4.1. As shown, these ideal filters have a constant-gain (usually taken as unity-gain) passband characteristic and zero gain in their stopbass.

Another characteristic of an ideal filter is a linear phase response. To demonstrate this point, let us assume that a signal sequence $\{x(n)\}$ with frequency components confined to the frequency range $\omega_1 < \omega < \omega_2$ is passed through a filter with

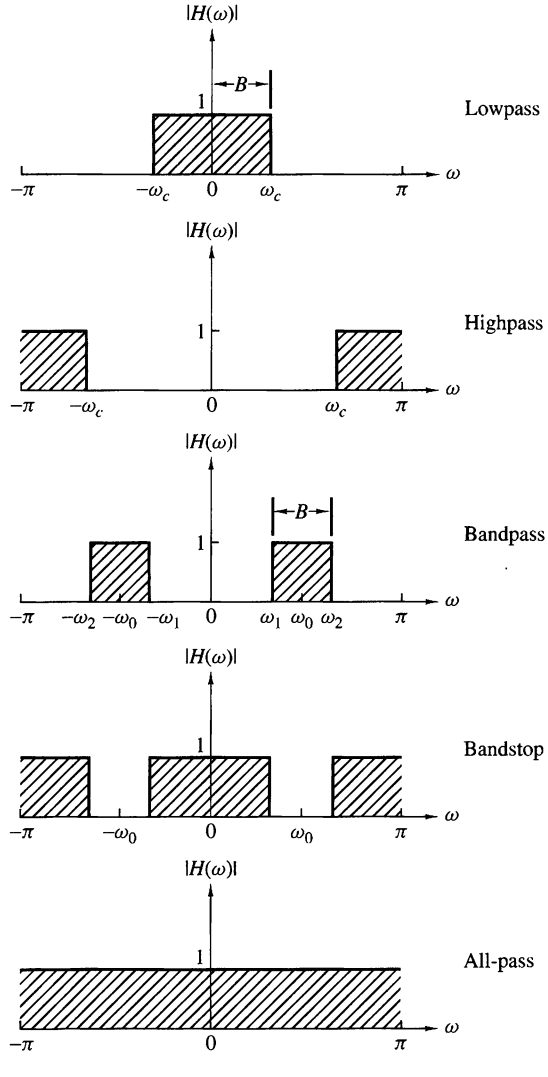


Figure 5.4.1
Magnitude responses for some ideal frequency-selective discrete-time filters.

frequency response

$$H(\omega) = \begin{cases} Ce^{-j\omega n_0}, & \omega_1 < \omega < \omega_2 \\ 0, & \text{otherwise} \end{cases} \tag{5.4.1}$$

where C and n_0 are constants. The signal at the output of the filter has a spectrum

$$\begin{aligned} Y(\omega) &= X(\omega)H(\omega) \\ &= CX(\omega)e^{-j\omega n_0}, \qquad \omega_1 < \omega < \omega_2 \end{aligned} \tag{5.4.2}$$

By applying the scaling and time-shifting properties of the Fourier transform, we obtain the time-domain output

$$y(n) = Cx(n - n_0) \tag{5.4.3}$$

Consequently, the filter output is simply a delayed and amplitude-scaled version of the input signal. A pure delay is usually tolerable and is not considered a distortionof the signal. Neither is amplitude scaling. Therefore, ideal filters have a linear phase characteristic within their passband, that is,

$$\Theta(\omega) = -\omega n_0 \tag{5.4.4}$$

The derivative of the phase with respect to frequency has the units of delay. Hence we can define the signal delay as a function of frequency as

$$\tau_g(\omega) = -\frac{d\Theta(\omega)}{d\omega} \tag{5.4.5}$$

$\tau_g(\omega)$ is usually called the *envelope delay*or the *group delay* of the filter. We interpret $\tau_g(\omega)$ as the time delay that a signal component of frequency ω undergoes as it passes from the input to the output of the system. Note that when $\Theta(\omega)$ is linear as in (5.4.4), $\tau_g(\omega) = n_0 =$ constant. In this case all frequency components of the input signal undergo the same time delay.

In conclusion, ideal filters have a constant magnitude characteristic and a linear phase characteristic within their passband. In all cases, such filters are not physically realizable but serve as a mathematical idealization of practical filters. For example, the ideal lowpass filter has an impulse response

$$h_{lp}(n) = \frac{\sin \omega_c \pi n}{\pi n}, \qquad -\infty < n < \infty \tag{5.4.6}$$

We note that this filter is not causal and it is not absolutely summable and therefore it is also unstable. Consequently, this ideal filter is physically unrealizable. Nevertheless, its frequency response characteristics can be approximated very closely by practical, physically realizable filters, as will be demonstrated in Chapter 10.

In the following discussion, we treat the design of some simple digital filters by the placement of poles and zeros in the z-plane. We have already described how

the location of poles and zeros affects the frequency response characteristics of the system. In particular, in Section 5.2.2 we presented a graphical method for computing the frequency response characteristics from the pole–zero plot. This same approach can be used to design a number of simple but important digital filters with desirable frequency response characteristics.

The basic principle underlying the pole–zero placement method is to locate poles near points of the unit circle corresponding to frequencies to be emphasized, and to place zeros near the frequencies to be deemphasized. Furthermore, the following constraints must be imposed:

1. All poles should be placed inside the unit circle in order for the filter to be stable. However, zeros can be placed anywhere in the z-plane.
2. All complex zeros and poles must occur in complex-conjugate pairs in order for the filter coefficients to be real.

From our previous discussion we recall that for a given pole–zero pattern, the system function $H(z)$ can be expressed as

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = b_0 \frac{\prod_{k=1}^{M}(1 - z_k z^{-1})}{\prod_{k=1}^{N}(1 - p_k z^{-1})} \tag{5.4.7}$$

where b_0 is a gain constant selected to normalize the frequency response at some specified frequency. That is, b_0 is selected such that

$$|H(\omega_0)| = 1 \tag{5.4.8}$$

where ω_0 is a frequency in the passband of the filter. Usually, N is selected to equal or exceed M, so that the filter has more nontrivial poles than zeros.

In the next section, we illustrate the method of pole–zero placement in the design of some simple lowpass, highpass, and bandpass filters, digital resonators, and comb filters. The design procedure is facilitated when carried out interactively on a digital computer with a graphics terminal.

5.4.2 Lowpass, Highpass, and Bandpass Filters

In the design of lowpass digital filters, the poles should be placed near the unit circle at points corresponding to low frequencies (near $\omega = 0$) and zeros should be placed near or on the unit circle at points corresponding to high frequencies (near $\omega = \pi$). The opposite holds true for highpass filters.

Figure 5.4.2 illustrates the pole–zero placement of three lowpass and three highpass filters. The magnitude and phase responses for the single-pole filter with system function

$$H_1(z) = \frac{1 - a}{1 - az^{-1}} \tag{5.4.9}$$

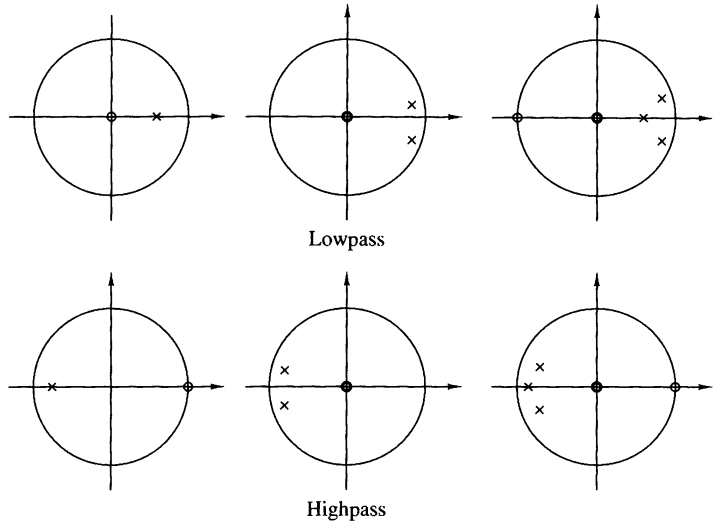


Figure 5.4.2 Pole–zero patterns for several lowpass and highpass filters.

are illustrated in Fig. 5.4.3 for $a = 0.9$. The gain G was selected as $1 - a$, so that the filter has unity gain at $\omega = 0$. The gain of this filter at high frequencies is relatively small.

The addition of a zero at $z = -1$ further attenuates the response of the filter at high frequencies. This leads to a filter with a system function

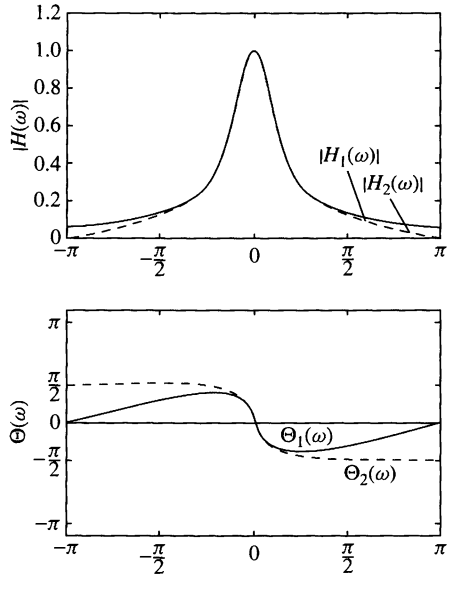


Figure 5.4.3 Magnitude and phase response of (1) a single-pole filter and (2) a one-pole, one-zero filter; $H_1(z) = (1 - a)/(1 - az^{-1})$, $H_2(z) = [(1 - a)/2][(1 + z^{-1})/(1 - az^{-1})]$ and $a = 0.9$.

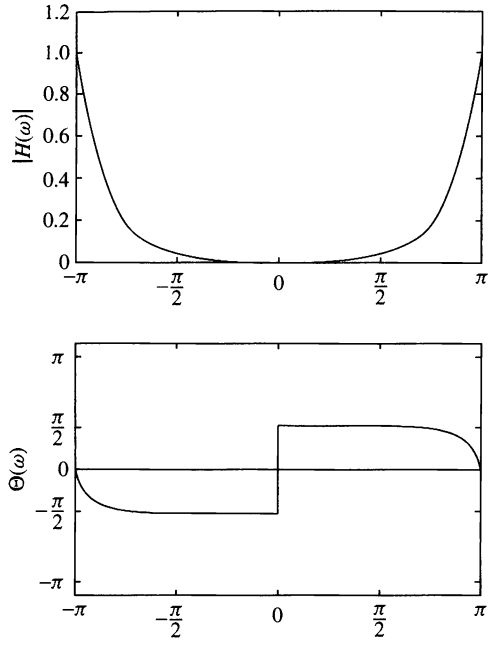


Figure 5.4.4
Magnitude and phase response of a simple highpass filter; $H(z) = [(1 - a)/2][(1 - z^{-1})/(1 + az^{-1})]$ with $a = 0.9$.

$$H_2(z) = \frac{1-a}{2}\frac{1+z^{-1}}{1-az^{-1}} \tag{5.4.10}$$

and a frequency response characteristic that is also illustrated in Fig. 5.4.3. In this case the magnitude of $H_2(\omega)$ goes to zero at $\omega = \pi$.

Similarly, we can obtain simple highpass filters by reflecting (folding) the pole–zero locations of the lowpass filters about the imaginary axis in the z-plane. Thus we obtain the system function

$$H_3(z) = \frac{1-a}{2}\frac{1-z^{-1}}{1+az^{-1}} \tag{5.4.11}$$

which has the frequency response characteristics illustrated in Fig. 5.4.4 for $a = 0.9$.

EXAMPLE 5.4.1

A two-pole lowpass filter has the system function

$$H(z) = \frac{b_0}{(1 - pz^{-1})^2}$$

Determine the values of b_0 and p such that the frequency response $H(\omega)$ satisfies the conditions

$$H(0) = 1$$

and

$$\left|H\left(\frac{\pi}{4}\right)\right|^2 = \frac{1}{2}$$

Solution. At $\omega = 0$ we have

$$H(0) = \frac{b_0}{(1-p)^2} = 1$$

Hence

$$b_0 = (1-p)^2$$

At $\omega = \pi/4$,

$$H\left(\frac{\pi}{4}\right) = \frac{(1-p)^2}{(1-pe^{-j\pi/4})^2}$$

$$= \frac{(1-p)^2}{(1-p\cos(\pi/4) + jp\sin(\pi/4))^2}$$

$$= \frac{(1-p)^2}{(1-p/\sqrt{2}+jp/\sqrt{2})^2}$$

Hence

$$\frac{(1-p)^4}{[(1-p/\sqrt{2})^2 + p^2/2]^2} = \frac{1}{2}$$

or, equivalently,

$$\sqrt{2}(1-p)^2 = 1 + p^2 - \sqrt{2}p$$

The value of $p = 0.32$ satisfies this equation. Consequently, the system function for the desired filter is

$$H(z) = \frac{0.46}{(1-0.32z^{-1})^2}$$

The same principles can be applied for the design of bandpass filters. Basically, the bandpass filter should contain one or more pairs of complex-conjugate poles near the unit circle, in the vicinity of the frequency band that constitutes the passband of the filter. The following example serves to illustrate the basic ideas.

EXAMPLE 5.4.2

Design a two-pole bandpass filter that has the center of its passband at $\omega = \pi/2$, zero in its frequency response characteristic at $\omega = 0$ and $\omega = \pi$, and a magnitude response of $1/\sqrt{2}$ at $\omega = 4\pi/9$.

Solution. Clearly, the filter must have poles at

$$p_{1,2} = re^{\pm j\pi/2}$$

and zeros at $z = 1$ and $z = -1$. Consequently, the system function is

$$H(z) = G\frac{(z-1)(z+1)}{(z-jr)(z+jr)}$$

$$= G\frac{z^2-1}{z^2+r^2}$$

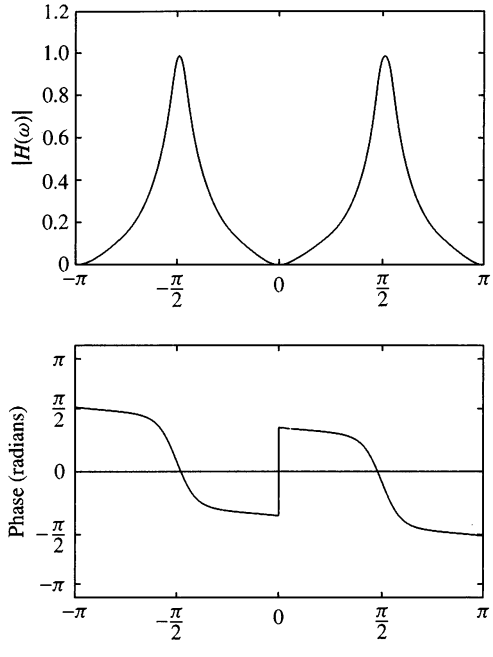


Figure 5.4.5
Magnitude and phase response of a simple bandpass filter in Example 5.4.2; $H(z) = 0.15[(1 - z^{-2})/(1 + 0.7z^{-2})]$.

The gain factor is determined by evaluating the frequency response $H(\omega)$ of the filter at $\omega = \pi/2$. Thus we have

$$H\left(\frac{\pi}{2}\right) = G\frac{2}{1 - r^2} = 1$$

$$G = \frac{1 - r^2}{2}$$

The value of r is determined by evaluating $H(\omega)$ at $\omega = 4\pi/9$. Thus we have

$$\left|H\left(\frac{4\pi}{9}\right)\right|^2 = \frac{(1 - r^2)^2}{4}\frac{2 - 2\cos(8\pi/9)}{1 + r^4 + 2r^2\cos(8\pi/9)} = \frac{1}{2}$$

or, equivalently,

$$1.94(1 - r^2)^2 = 1 - 1.88r^2 + r^4$$

The value of $r^2 = 0.7$ satisfies this equation. Therefore, the system function for the desired filter is

$$H(z) = 0.15\frac{1 - z^{-2}}{1 + 0.7z^{-2}}$$

Its frequency response is illustrated in Fig. 5.4.5.

It should be emphasized that the main purpose of the foregoing methodology for designing simple digital filters by pole–zero placement is to provide insight into the effect that poles and zeros have on the frequency response characteristic of

systems. The methodology is not intended as a good method for designing digital filters with well-specified passband and stopband characteristics. Systematic methods for the design of sophisticated digital filters for practical applications are discussed in Chapter 10.

A simple lowpass-to-highpass filter transformation. Suppose that we have designed a prototype lowpass filter with impulse response $h_{lp}(n)$. By using the frequency translation property of the Fourier transform, it is possible to convert the prototype filter to either a bandpass or a highpass filter. Frequency transformations for converting a prototype lowpass filter into a filter of another type are described in detail in Section 10.3. In this section we present a simple frequency transformation for converting a lowpass filter into a highpass filter, and vice versa.

If $h_{lp}(n)$ denotes the impulse response of a lowpass filter with frequency response $H_{lp}(\omega)$, a highpass filter can be obtained by translating $H_{lp}(\omega)$ by π radians (i.e., replacing ω by $\omega - \pi$). Thus

$$H_{hp}(\omega) = H_{lp}(\omega - \pi) \tag{5.4.12}$$

where $H_{hp}(\omega)$ is the frequency response of the highpass filter. Since a frequency translation of π radians is equivalent to multiplication of the impulse response $h_{lp}(n)$ by $e^{j\pi n}$, the impulse response of the highpass filter is

$$h_{hp}(n) = (e^{j\pi})^n h_{lp}(n) = (-1)^n h_{lp}(n) \tag{5.4.13}$$

Therefore, the impulse response of the highpass filter is simply obtained from the impulse response of the lowpass filter by changing the signs of the odd-numbered samples in $h_{lp}(n)$. Conversely,

$$h_{lp}(n) = (-1)^n h_{hp}(n) \tag{5.4.14}$$

If the lowpass filter is described by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{5.4.15}$$

its frequency response is

$$H_{lp}(\omega) = \frac{\sum_{k=0}^{M} b_k e^{-j\omega k}}{1 + \sum_{k=1}^{N} a_k e^{-j\omega k}} \tag{5.4.16}$$

Now, if we replace ω by $\omega - \pi$, in (5.4.16), then

$$H_{hp}(\omega) = \frac{\sum_{k=0}^{M} (-1)^k b_k e^{-j\omega k}}{1 + \sum_{k=1}^{N} (-1)^k a_k e^{-j\omega k}} \tag{5.4.17}$$

which corresponds to the difference equation

$$y(n) = -\sum_{k=1}^{N}(-1)^k a_k y(n-k) + \sum_{k=0}^{M}(-1)^k b_k x(n-k) \tag{5.4.18}$$

EXAMPLE 5.4.3

Convert the lowpass filter described by the difference equation

$$y(n) = 0.9y(n-1) + 0.1x(n)$$

into a highpass filter.

Solution. The difference equation for the highpass filter, according to (5.4.18), is

$$y(n) = -0.9y(n-1) + 0.1x(n)$$

and its frequency response is

$$H_{\text{hp}}(\omega) = \frac{0.1}{1 + 0.9e^{-j\omega}}$$

The reader may verify that $H_{\text{hp}}(\omega)$ is indeed highpass.

5.4.3 Digital Resonators

A *digital resonator* is a special two-pole bandpass filter with the pair of complex-conjugate poles located near the unit circle as shown in Fig. 5.4.6(a). The magnitude of the frequency response of the filter is shown in Fig. 5.4.6(b). The name resonator refers to the fact that the filter has a large magnitude response (i.e., it resonates) in the vicinity of the pole location. The angular position of the pole determines the resonant frequency of the filter. Digital resonators are useful in many applications, including simple bandpass filtering and speech generation.

In the design of a digital resonator with a resonant peak at or near $\omega = \omega_0$, we select the complex-conjugate poles at

$$p_{1,2} = re^{\pm j\omega_0}, \qquad 0 < r < 1$$

In addition, we can select up to two zeros. Although there are many possible choices, two cases are of special interest. One choice is to locate the zeros at the origin. The other choice is to locate a zero at $z = 1$ and a zero at $z = -1$. This choice completely eliminates the response of the filter at frequencies $\omega = 0$ and $\omega = \pi$, and it is useful in many practical applications.

The system function of the digital resonator with zeros at the origin is

$$H(z) = \frac{b_0}{(1 - re^{j\omega_0}z^{-1})(1 - re^{-j\omega_0}z^{-1})} \tag{5.4.19}$$

$$H(z) = \frac{b_0}{1 - (2r\cos\omega_0)z^{-1} + r^2z^{-2}} \tag{5.4.20}$$

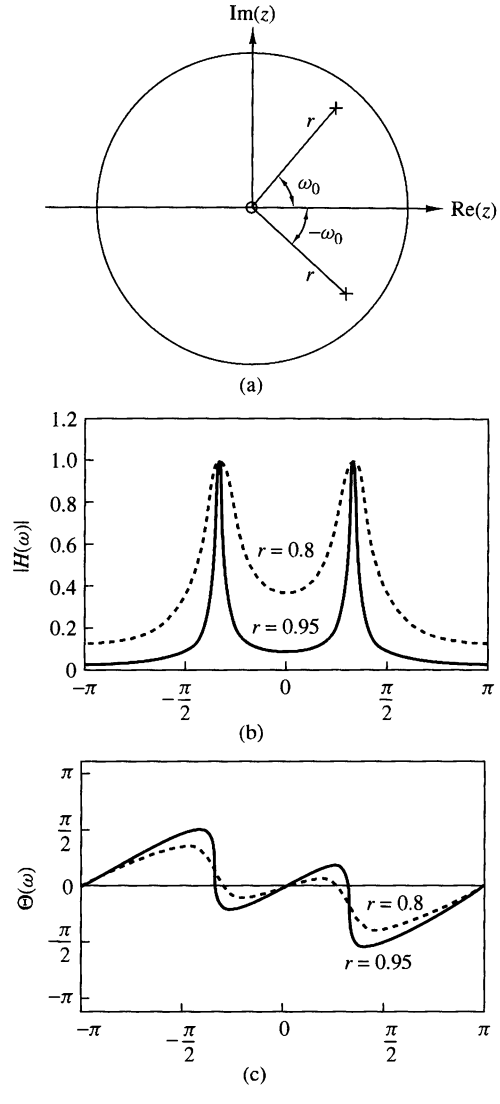


Figure 5.4.6
(a) Pole–zero pattern and (b) the corresponding magnitude and phase response of a digital resonator with (1) $r = 0.8$ and (2) $r = 0.95$.

Since $|H(\omega)|$ has its peak at or near $\omega = \omega_0$, we select the gain b_0 so that $|H(\omega_0)| = 1$. From (5.4.19) we obtain

$$\begin{aligned} H(\omega_0) &= \frac{b_0}{(1 - re^{j\omega_0}e^{-j\omega_0})(1 - re^{-j\omega_0}e^{-j\omega_0})} \\ &= \frac{b_0}{(1 - r)(1 - re^{-j2\omega_0})} \end{aligned} \tag{5.4.21}$$

and hence

$$|H(\omega_0)| = \frac{b_0}{(1 - r)\sqrt{1 + r^2 - 2r\cos 2\omega_0}} = 1$$

Thus the desired normalization factor is

$$b_0 = (1 - r)\sqrt{1 + r^2 - 2r\cos 2\omega_0} \tag{5.4.22}$$

The frequency response of the resonator in (5.4.19) can be expressed as

$$\begin{aligned} |H(\omega)| &= \frac{b_0}{U_1(\omega)U_2(\omega)} \\ \Theta(\omega) &= 2\omega - \Phi_1(\omega) - \Phi_2(\omega) \end{aligned} \tag{5.4.23}$$

where $U_1(\omega)$ and $U_2(\omega)$ are the magnitudes of the vectors from p_1 and p_2 to the point ω in the unit circle and $\Phi_1(\omega)$ and $\Phi_2(\omega)$ are the corresponding angles of these two vectors. The magnitudes $U_1(\omega)$ and $U_2(\omega)$ may be expressed as

$$\begin{aligned} U_1(\omega) &= \sqrt{1 + r^2 - 2r\cos(\omega_0 - \omega)} \\ U_2(\omega) &= \sqrt{1 + r^2 - 2r\cos(\omega_0 + \omega)} \end{aligned} \tag{5.4.24}$$

For any value of r, $U_1(\omega)$ takes its minimum value $(1 - r)$ at $\omega = \omega_0$. The product $U_1(\omega)U_2(\omega)$ reaches a minimum value at the frequency

$$\omega_r = \cos^{-1}\left(\frac{1 + r^2}{2r}\cos\omega_0\right) \tag{5.4.25}$$

which defines precisely the resonant frequency of the filter. We observe that when r is very close to unity, $\omega_r \approx \omega_0$, which is the angular position of the pole. We also observe that as r approaches unity, the resonance peak becomes sharper because $U_1(\omega)$ changes more rapidly in relative size in the vicinity of ω_0. A quantitative measure of the sharpness of the resonance is provided by the 3-dB bandwidth $\Delta\omega$ of the filter. For values of r close to unity,

$$\Delta\omega \approx 2(1 - r) \tag{5.4.26}$$

Figure 5.4.6 illustrates the magnitude and phase of digital resonators with $\omega_0 = \pi/3$, $r = 0.8$ and $\omega_0 = \pi/3$, $r = 0.95$. We note that the phase response undergoes its greatest rate of change near the resonant frequency.

If the zeros of the digital resonator are placed at $z = 1$ and $z = -1$, the resonator has the system function

$$\begin{aligned} H(z) &= G\frac{(1 - z^{-1})(1 + z^{-1})}{(1 - re^{j\omega_0}z^{-1})(1 - re^{-j\omega_0}z^{-1})} \\ &= G\frac{1 - z^{-2}}{1 - (2r\cos\omega_0)z^{-1} + r^2z^{-2}} \end{aligned} \tag{5.4.27}$$

and a frequency response characteristic

$$H(\omega) = b_0 \frac{1 - e^{-j2\omega}}{[1 - re^{j(\omega_0 - \omega)}][1 - re^{-j(\omega_0 + \omega)}]} \tag{5.4.28}$$

We observe that the zeros at $z = \pm 1$ affect both the magnitude and phase response of the resonator. For example, the magnitude response is

$$|H(\omega)| = b_0 \frac{N(\omega)}{U_1(\omega)U_2(\omega)} \tag{5.4.29}$$

where $N(\omega)$ is defined as

$$N(\omega) = \sqrt{2(1 - \cos 2\omega)}$$

Due to the presence of the zero factor, the resonant frequency is altered from that given by the expression in (5.4.25). The bandwidth of the filter is also altered. Although exact values for these two parameters are rather tedious to derive, we can easily compute the frequency response in (5.4.28) and compare the result with the previous case in which the zeros are located at the origin.

Figure 5.4.7 illustrates the magnitude and phase characteristics for $\omega_0 = \pi/3$, $r = 0.8$ and $\omega_0 = \pi/3$, $r = 0.95$. We observe that this filter has a slightly smaller bandwidth than the resonator, which has zeros at the origin. In addition, there appears to be a very small shift in the resonant frequency due to the presence of the zeros.

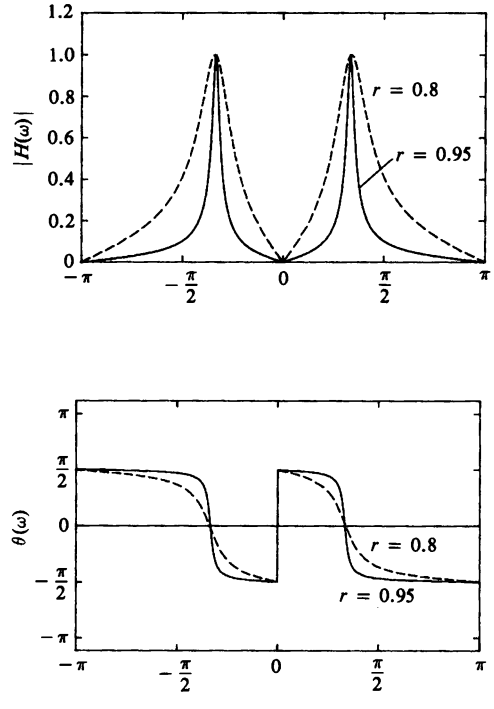


Figure 5.4.7 Magnitude and phase response of digital resonator with zeros at $\omega = 0$ and $\omega = \pi$ and (1) $r = 0.8$ and (2) $r = 0.95$.

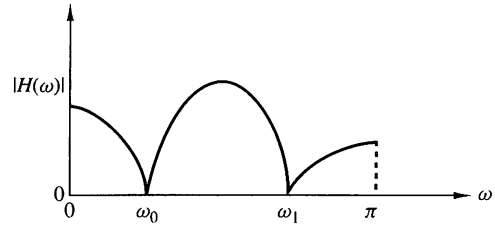


Figure 5.4.8
Frequency response characteristic of a notch filter.

5.4.4 Notch Filters

A notch filter is a filter that contains one or more deep notches or, ideally, perfect nulls in its frequency response characteristic. Figure 5.4.8 illustrates the frequency response characteristic of a notch filter with nulls at frequencies ω_0 and ω_1. Notch filters are useful in many applications where specific frequency components must be eliminated. For example, instrumentation and recording systems require that the power-line frequency of 60 Hz and its harmonics be eliminated.

To create a null in the frequency response of a filter at a frequency ω_0, we simply introduce a pair of complex-conjugate zeros on the unit circle at an angle ω_0. That is,

$$z_{1,2} = e^{\pm j\omega_0}$$

Thus the system function for an FIR notch filter is simply

$$\begin{aligned} H(z) &= b_0(1 - e^{j\omega_0}z^{-1})(1 - e^{-j\omega_0}z^{-1}) \\ &= b_0(1 - 2\cos\omega_0 z^{-1} + z^{-2}) \end{aligned} \tag{5.4.30}$$

As an illustration, Fig. 5.4.9 shows the magnitude response for a notch filter having a null at $\omega = \pi/4$.

The problem with the FIR notch filter is that the notch has a relatively large bandwidth, which means that other frequency components around the desired null are severely attenuated. To reduce the bandwidth of the null, we can resort to a more sophisticated, longer FIR filter designed according to criteria described in Chapter 10. Alternatively, we could, in an ad hoc manner, attempt to improve on the frequency response characteristics by introducing poles in the system function.

Suppose that we place a pair of complex-conjugate poles at

$$p_{1,2} = re^{\pm j\omega_0}$$

The effect of the poles is to introduce a resonance in the vicinity of the null and thus to reduce the bandwidth of the notch. The system function for the resulting filter is

$$H(z) = b_0\frac{1 - 2\cos\omega_0 z^{-1} + z^{-2}}{1 - 2r\cos\omega_0 z^{-1} + r^2 z^{-2}} \tag{5.4.31}$$

The magnitude response $|H(\omega)|$ of the filter in (5.4.31) is plotted in Fig. 5.4.10 for $\omega_0 = \pi/4$, $r = 0.85$, and for $\omega_0 = \pi/4$, $r = 0.95$. When compared with the frequency response of the FIR filter in Fig. 5.4.9, we note that the effect of the poles is to reduce the bandwidth of the notch.

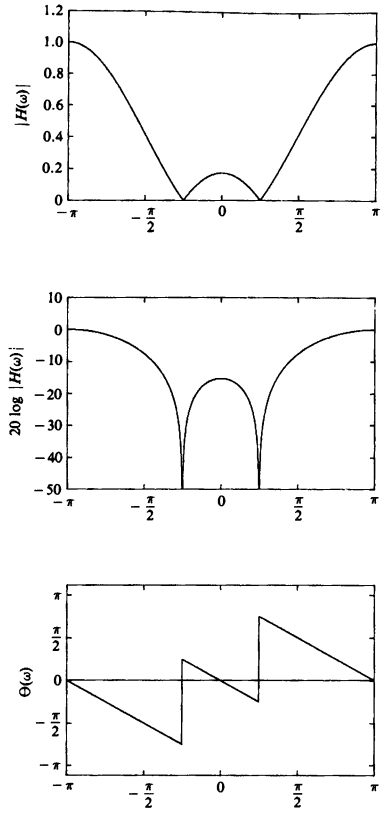


Figure 5.4.9
Frequency response characteristics of a notch filter with a notch at $\omega = \pi/4$ or $f = 1/8$; $H(z) = G[1 - 2\cos\omega_0 z^{-1} + z^{-2}]$.

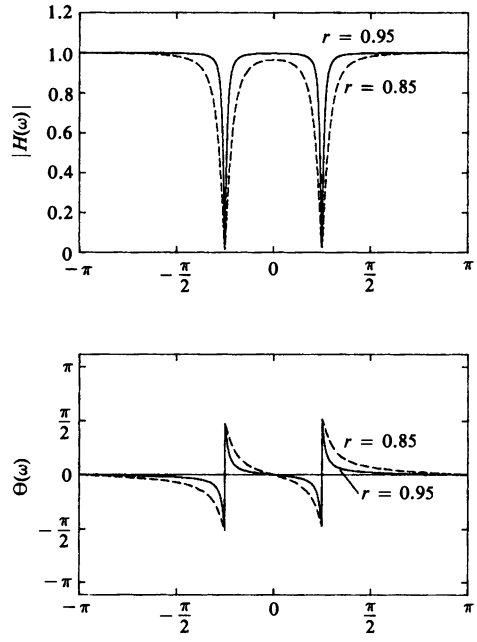


Figure 5.4.10
Frequency response characteristics of two notch filters with poles at (1) $r = 0.85$ and (2) $r = 0.95$; $H(z) = b_0[(1-2\cos\omega_0 z^{-1}+z^{-2})/(1-2r\cos\omega_0 z^{-1}+r^2 z^{-2})]$.

In addition to reducing the bandwidth of the notch, the introduction of a pole in the vicinity of the null may result in a small ripple in the passband of the filter due to the resonance created by the pole. The effect of the ripple can be reduced by introducing additional poles and/or zeros in the system function of the notch filter. The major problem with this approach is that it is basically an ad hoc, trial-and-error method.

5.4.5 Comb Filters

In its simplest form, a comb filter can be viewed as a notch filter in which the nulls occur periodically across the frequency band, hence the analogy to an ordinary comb that has periodically spaced teeth. Comb filters find applications in a wide range of practical systems such as in the rejection of power-line harmonics, in the separation of solar and lunar components from ionospheric measurements of electron concentration, and in the suppression of clutter from fixed objects in moving-target-indicator (MTI) radars.

To illustrate a simple form of a comb filter, consider a moving average (FIR) filter described by the difference equation

$$y(n) = \frac{1}{M+1} \sum_{k=0}^{M} x(n-k) \tag{5.4.32}$$

The system function of this FIR filter is

$$\begin{aligned} H(z) &= \frac{1}{M+1} \sum_{k=0}^{M} z^{-k} \\ &= \frac{1}{M+1} \frac{[1 - z^{-(M+1)}]}{(1 - z^{-1})} \end{aligned} \tag{5.4.33}$$

and its frequency response is

$$H(\omega) = \frac{e^{-j\omega M/2}}{M+1} \frac{\sin \omega(\frac{M+1}{2})}{\sin(\omega/2)} \tag{5.4.34}$$

From (5.4.33) we observe that the filter has zeros on the unit circle at

$$z = e^{j2\pi k/(M+1)}, \qquad k = 1, 2, 3, \ldots, M \tag{5.4.35}$$

Note that the pole at $z = 1$ is actually canceled by the zero at $z = 1$, so that in effect the FIR filter does not contain poles outside $z = 0$.

A plot of the magnitude characteristic of (5.4.34) clearly illustrates the existence of the periodically spaced zeros in frequency at $\omega_k = 2\pi k/(M + 1)$ for $k = 1, 2, \ldots, M$. Figure 5.4.11 shows $|H(\omega)|$ for $M = 10$.

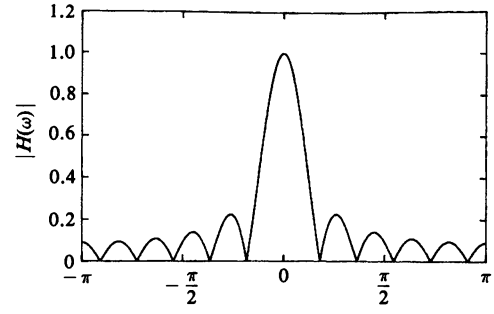


Figure 5.4.11
Magnitude response characteristic for the comb filter given by (5.4.34) with $M = 10$.

In more general terms, we can create a comb filter by taking an FIR filter with system function

$$H(z) = \sum_{k=0}^{M} h(k)z^{-k} \tag{5.4.36}$$

and replacing z by z^L, where L is a positive integer. Thus the new FIR filter has a system function

$$H_L(z) = \sum_{k=0}^{M} h(k)z^{-kL} \tag{5.4.37}$$

If the frequency response of the original FIR filter is $H(\omega)$, the frequency response of the FIR in (5.4.37) is

$$\begin{aligned} H_L(\omega) &= \sum_{k=0}^{M} h(k)e^{-jkL\omega} \\ &= H(L\omega) \end{aligned} \tag{5.4.38}$$

Consequently, the frequency response characteristic $H_L(\omega)$ is simply an L-order repetition of $H(\omega)$ in the range $0 \le \omega \le 2\pi$. Figure 5.4.12 illustrates the relationship between $H_L(\omega)$ and $H(\omega)$ for $L = 5$.

Now, suppose that the original FIR filter with system function $H(z)$ has a spectral null (i.e., a zero), at some frequency ω_0. Then the filter with system function $H_L(z)$ has periodically spaced nulls at $\omega_k = \omega_0 + 2\pi k/L$, $k = 0, 1, 2, \ldots, L-1$. As an illustration, Fig. 5.4.13 shows an FIR comb filter with $M = 3$ and $L = 3$. This FIR filter can be viewed as an FIR filter of length 10, but only four of the 10 filter coefficients are nonzero.

Let us now return to the moving average filter with system function given by (5.4.33). Suppose that we replace z by z^L. Then the resulting comb filter has the system function

$$H_L(z) = \frac{1}{M+1}\,\frac{1 - z^{-L(M+1)}}{1 - z^{-L}} \tag{5.4.39}$$

and a frequency response

$$H_L(\omega) = \frac{1}{M+1}\,\frac{\sin[\omega L(M+1)/2]}{\sin(\omega L/2)}\,e^{-j\omega LM/2} \tag{5.4.40}$$

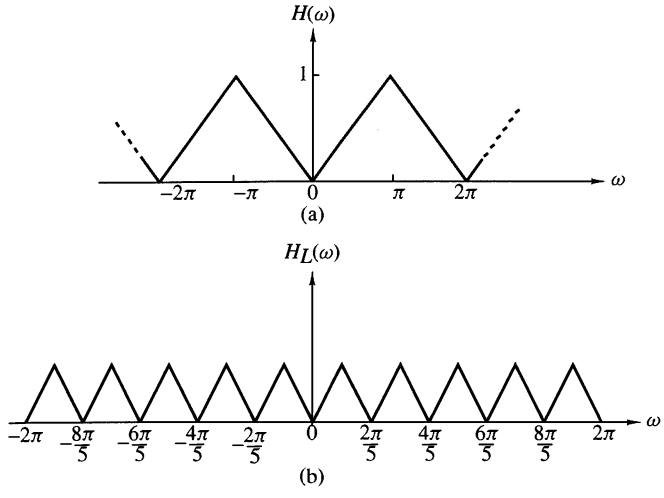


Figure 5.4.12 Comb filter with frequency response $H_L(\omega)$ obtained from $H(\omega)$.

This filter has zeros on the unit circle at

$$z_k = e^{j2\pi k/L(M+1)} \tag{5.4.41}$$

for all integer values of k except $k = 0, L, 2L, \ldots, ML$. Figure 5.4.14 illustrates $|H_L(\omega)|$ for $L = 3$ and $M = 10$.

The comb filter described by (5.4.39) finds application in the separation of solar and lunar spectral components in ionospheric measurements of electron concentration as described in the paper by Bernhardt et al. (1976). The solar period is $T_s = 24$ hours and results in a solar component of one cycle per day and its harmonics. The lunar period is $T_L = 24.84$ hours and provides spectral lines at 0.96618 cycle per day and its harmonics. Figure 5.4.15(a) shows a plot of the power density spectrum of the unfiltered ionospheric measurements of the electron concentration. Note that the weak lunar spectral components are almost hidden by the strong solar spectral components.

The two sets of spectral components can be separated by the use of comb filters. If we wish to obtain the solar components, we can use a comb filter with a narrow passband at multiples of one cycle per day. This can be achieved by selecting L such that $F_s/L = 1$ cycle per day, where F_s is the corresponding sampling frequency. The

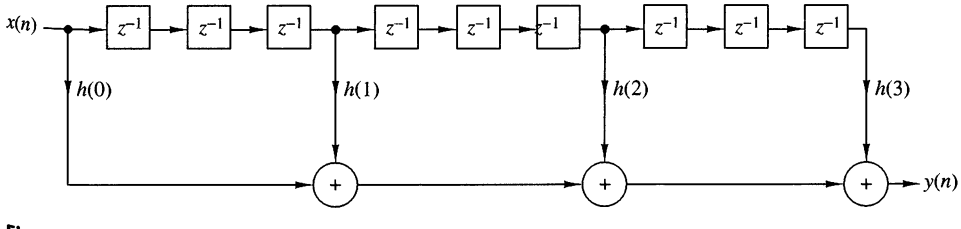


Figure 5.4.13 Realization of an FIR comb filter having $M = 3$ and $L = 3$.

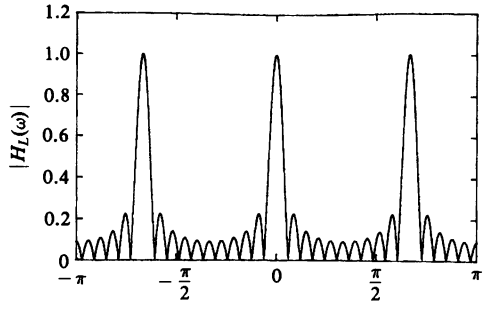


Figure 5.4.14
Magnitude response characteristic for a comb filter given by (5.4.40), with $L = 3$ and $M = 10$.

result is a filter that has peaks in its frequency response at multiples of one cycle per day. By selecting $M = 58$, the filter will have nulls at multiples of $(F_s/L)/(M+1) = 1/59$ cycle per day. These nulls are very close to the lunar components and result in good rejection. Figure 5.4.15(b) illustrates the power spectral density of the output of the comb filter that isolates the solar components. A comb filter that rejects the solar components and passes the lunar components can be designed in a similar manner. Figure 5.4.15(c) illustrates the power spectral density at the output of such a lunar filter.

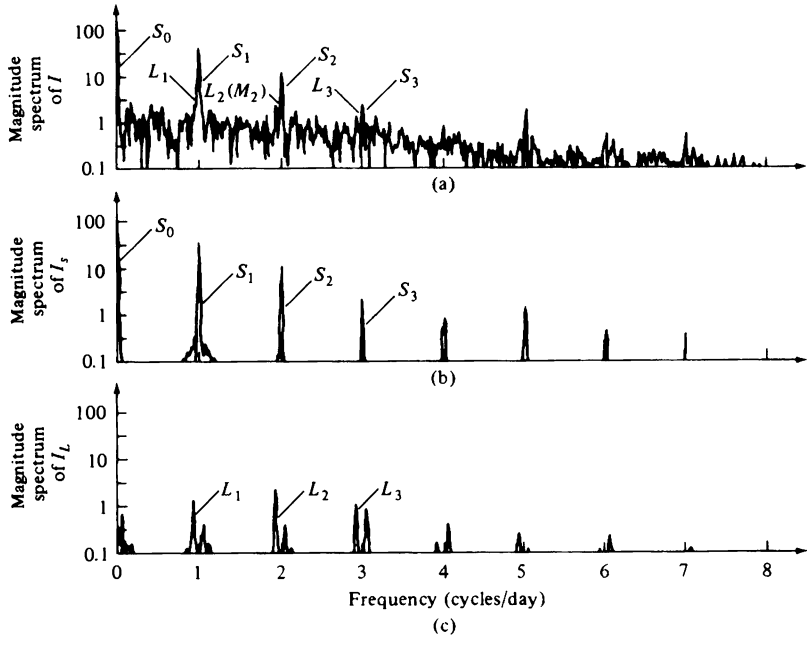


Figure 5.4.15 (a) Spectrum of unfiltered electron content data; (b) spectrum of output of solar filter; (c) spectrum of output of lunar filter. [From paper by Bernhardt et al. (1976). Reprinted with permission of the American Geophysical Union.]

5.4.6 All-Pass Filters

An all-pass filter is defined as a system that has a constant magnitude response for all frequencies, that is,

$$|H(\omega)| = 1, \qquad 0 \le \omega \le \pi \tag{5.4.42}$$

The simplest example of an all-pass filter is a pure delay system with system function

$$H(z) = z^{-k}$$

This system passes all signals without modification except for a delay of k samples. This is a trivial all-pass system that has a linear phase response characteristic.

A more interesting all-pass filter is described by the system function

$$\begin{aligned} H(z) &= \frac{a_N + a_{N-1}z^{-1} + \cdots + a_1 z^{-N+1} + z^{-N}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}} \\ &= \frac{\sum_{k=0}^{N} a_k z^{-N+k}}{\sum_{k=0}^{N} a_k z^{-k}}, \qquad a_0 = 1 \end{aligned} \tag{5.4.43}$$

where all the filter coefficients $\{a_k\}$ are real. If we define the polynomial $A(z)$ as

$$A(z) = \sum_{k=0}^{N} a_k z^{-k}, \qquad a_0 = 1$$

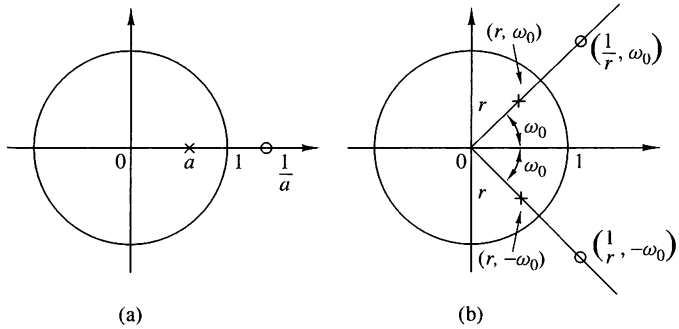
then (5.4.43) can be expressed as

$$H(z) = z^{-N}\frac{A(z^{-1})}{A(z)} \tag{5.4.44}$$

Since

$$|H(\omega)|^2 = H(z)H(z^{-1})|_{z=e^{j\omega}} = 1$$

the system given by (5.4.44) is an all-pass system. Furthermore, if z_0 is a pole of $H(z)$, then $1/z_0$ is a zero of $H(z)$ (i.e., the poles and zeros are reciprocals of one another). Figure 5.4.16 illustrates typical pole–zero patterns for a single-pole, single-zero filter and a two-pole, two-zero filter. A plot of the phase characteristics of these filters is shown in Fig. 5.4.17 for $a = 0.6$ and $r = 0.9$, $\omega_0 = \pi/4$.



(a) (b)

Figure 5.4.16 Pole–zero patterns of (a) a first-order and (b) a second-order all-pass filter.

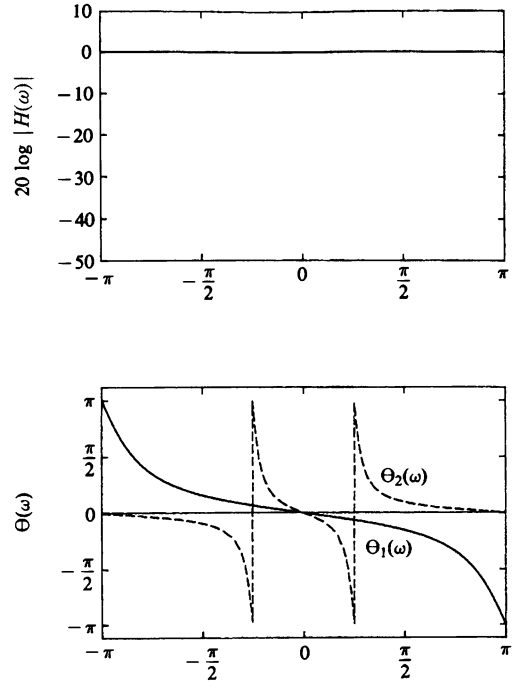


Figure 5.4.17
Frequency response characteristics of an all-pass filter with system functions (1) $H(z) = (0.6 + z^{-1})/(1 + 0.6z^{-1})$, (2) $H(z) = (r^2 - 2r\cos\omega_0 z^{-1} + z^{-2})/(1 - 2r\cos\omega_0 z^{-1} + r^2 z^{-2})$, $r = 0.9$, $\omega_0 = \pi/4$.

The most general form for the system function of an all-pass system with real coefficients, expressed in factored form in terms of poles and zeros, is

$$H_{\text{ap}}(z) = \prod_{k=1}^{N_R} \frac{z^{-1} - \alpha_k}{1 - \alpha_k z^{-1}} \prod_{k=1}^{N_c} \frac{(z^{-1} - \beta_k)(z^{-1} - \beta_k^*)}{(1 - \beta_k z^{-1})(1 - \beta_k^* z^{-1})} \tag{5.4.45}$$

where there are N_R real poles and zeros and N_c complex-conjugate pairs of poles and zeros. For causal and stable systems we require that $-1 < \alpha_k < 1$ and $|\beta_k| < 1$.

Expressions for the phase response and group delay of all-pass systems can easily be obtained using the method described in Section 5.2.1. For a single pole–single zero all-pass system we have

$$H_{\text{ap}}(\omega) = \frac{e^{j\omega} - re^{-j\theta}}{1 - re^{j\theta}e^{-j\omega}}$$

Hence

$$\Theta_{\text{ap}}(\omega) = -\omega - 2\tan^{-1}\frac{r\sin(\omega - \theta)}{1 - r\cos(\omega - \theta)}$$

and

$$\tau_g(\omega) = -\frac{d\Theta_{\text{ap}}(\omega)}{d\omega} = \frac{1 - r^2}{1 + r^2 - 2r\cos(\omega - \theta)} \tag{5.4.46}$$

We note that for a causal and stable system, $r < 1$ and hence $\tau_g(\omega) \geq 0$. Since the group delay of a higher-order pole–zero system consists of a sum of positive terms as in (5.4.46), the group delay will always be positive.

All-pass filters find application as phase equalizers. When placed in cascade with a system that has an undesired phase response, a phase equalizer is designed to compensate for the poor phase characteristics of the system and therefore to produce an overall linear-phase response.

5.4.7 Digital Sinusoidal Oscillators

A *digital sinusoidal oscillator*can be viewed as a limiting form of a two-pole resonator for which the complex-conjugate poles lie on the unit circle. From our previous discussion of second-order systems, we recall that a system with system function

$$H(z) = \frac{b_0}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{5.4.47}$$

and parameters

$$a_1 = -2r\cos\omega_0 \quad \text{and} \quad a_2 = r^2 \tag{5.4.48}$$

has complex-conjugate poles at $p = re^{\pm j\omega_0}$, and a unit sample response

$$h(n) = \frac{b_0 r^n}{\sin\omega_0} \sin(n+1)\omega_0 u(n) \tag{5.4.49}$$

If the poles are placed on the unit circle ($r = 1$) and b_0 is set to $A\sin\omega_0$, then

$$h(n) = A\sin(n+1)\omega_0 u(n) \tag{5.4.50}$$

Thus the impulse response of the second-order system with complex-conjugate poles on the unit circle is a sinusoid and the system is called a digital sinusoidal oscillator or a *digital sinusoidal generator.*

A digital sinusoidal generator is a basic component of a digital frequency synthesizer.

The block diagram representation of the system function given by (5.4.47) is illustrated in Fig. 5.4.18. The corresponding difference equation for this system is

$$y(n) = -a_1 y(n-1) - y(n-2) + b_0\delta(n) \tag{5.4.51}$$

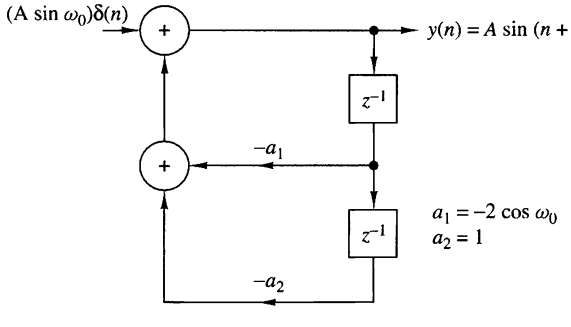


Figure 5.4.18
Digital sinusoidal generator.

where the parameters are $a_1 = -2\cos\omega_0$ and $b_0 = A\sin\omega_0$, and the initial conditions are $y(-1) = y(-2) = 0$. By iterating the difference equation in (5.4.51), we obtain

$$\begin{aligned} y(0) &= A\sin\omega_0 \\ y(1) &= 2\cos\omega_0 y(0) = 2A\sin\omega_0\cos\omega_0 = A\sin 2\omega_0 \\ y(2) &= 2\cos\omega_0 y(1) - y(0) \\ &= 2A\cos\omega_0\sin 2\omega_0 - A\sin\omega_0 \\ &= A(4\cos^2\omega_0 - 1)\sin\omega_0 \\ &= 3A\sin\omega_0 - 4\sin^3\omega_0 = A\sin 3\omega_0 \end{aligned}$$

and so forth. We note that the application of the impulse at $n = 0$ serves the purpose of beginning the sinusoidal oscillation. Thereafter, the oscillation is self-sustaining because the system has no damping (i.e., $r = 1$).

It is interesting to note that the sinusoidal oscillation obtained from the system in (5.4.51) can also be obtained by setting the input to zero and setting the initial conditions to $y(-1) = 0$, $y(-2) = -A\sin\omega_0$. Thus the zero-input response to the second-order system described by the homogeneous difference equation

$$y(n) = -a_1 y(n-1) - y(n-2) \tag{5.4.52}$$

with initial conditions $y(-1) = 0$ and $y(-2) = -A\sin\omega_0$, is exactly the same as the response of (5.4.51) to an impulse excitation. In fact, the difference equation in (5.4.52) can be obtained directly from the trigonometric identity

$$\sin\alpha + \sin\beta = 2\sin\frac{\alpha+\beta}{2}\cos\frac{\alpha-\beta}{2} \tag{5.4.53}$$

where, by definition, $\alpha = (n+1)\omega_0$, $\beta = (n-1)\omega_0$, and $y(n) = \sin(n+1)\omega_0$.

In some practical applications involving modulation of two sinusoidal carrier signals in phase quadrature, there is a need to generate the sinusoids $A\sin\omega_0 n$ and $A\cos\omega_0 n$. These signals can be generated from the so-called *coupled-form oscillator,* which can be obtained from the trigonometric formulas

$$\begin{aligned} \cos(\alpha+\beta) &= \cos\alpha\cos\beta - \sin\alpha\sin\beta \\ \sin(\alpha+\beta) &= \sin\alpha\cos\beta + \cos\alpha\sin\beta \end{aligned}$$

where, by definition, $\alpha = n\omega_0$, $\beta = \omega_0$, and

$$y_c(n) = \cos n\omega_0 u(n) \tag{5.4.54}$$

$$y_s(n) = \sin n\omega_0 u(n) \tag{5.4.55}$$

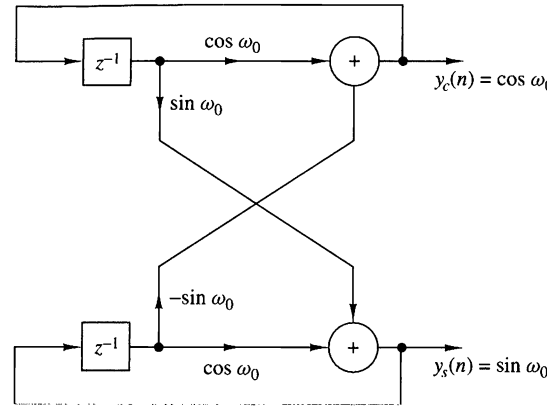


Figure 5.4.19
Realization of the coupled-form oscillator.

Thus we obtain the two coupled difference equations

$$y_c(n) = (\cos \omega_0) y_c(n-1) - (\sin \omega_0) y_s(n-1) \tag{5.4.56}$$

$$y_s(n) = (\sin \omega_0) y_c(n-1) + (\cos \omega_0) y_s(n-1) \tag{5.4.57}$$

which can also be expressed in matrix form as

$$\begin{bmatrix} y_c(n) \\ y_s(n) \end{bmatrix} = \begin{bmatrix} \cos \omega_0 & -\sin \omega_0 \\ \sin \omega_0 & \cos \omega_0 \end{bmatrix} \begin{bmatrix} y_c(n-1) \\ y_s(n-1) \end{bmatrix} \tag{5.4.58}$$

The structure for the realization of the coupled-form oscillator is illustrated in Fig. 5.4.19. We note that this is a two-output system which is not driven by any input, but which requires the initial conditions $y_c(-1) = A \cos \omega_0$ and $y_s(-1) = -A \sin \omega_0$ in order to begin its self-sustaining oscillations.

Finally, it is interesting to note that (5.4.58) corresponds to vector rotation in the two-dimensional coordinate system with coordinates $y_c(n)$ and $y_s(n)$. As a consequence, the coupled-form oscillator can also be implemented by use of the so-called CORDIC algorithm [see the book by Kung et al. (1985)].

5.5 Inverse Systems and Deconvolution

As we have seen, a linear time-invariant system takes an input signal $x(n)$ and produces an output signal $y(n)$, which is the convolution of $x(n)$ with the unit sample response $h(n)$ of the system. In many practical applications we are given an output signal from a system whose characteristics are unknown and we are asked to determine the input signal. For example, in the transmission of digital information at high data rates over telephone channels, it is well known that the channel distorts the signal and causes intersymbol interference among the data symbols. The intersymbol interference may cause errors when we attempt to recover the data. In such a case the problem is to design a corrective system which, when cascaded with the channel, produces an output that, in some sense, corrects for the distortion caused

by the channel, and thus yields a replica of the desired transmitted signal. In digital communications such a corrective system is called an *equalizer*. In the general context of linear systems theory, however, we call the corrective system an *inverse system*, because the corrective system has a frequency response which is basically the reciprocal of the frequency response of the system that caused the distortion. Furthermore, since the distortive system yields an output $y(n)$ that is the convolution of the input $x(n)$ with the impulse response $h(n)$, the inverse system operation that takes $y(n)$ and produces $x(n)$ is called *deconvolution*.

If the characteristics of the distortive system are unknown, it is often necessary, when possible, to excite the system with a known signal, observe the output, compare it with the input, and in some manner, determine the characteristics of the system. For example, in the digital communication problem just described, where the frequency response of the channel is unknown, the measurement of the channel frequency response can be accomplished by transmitting a set of equal-amplitude sinusoids, at different frequencies with a specified set of phases, within the frequency band of the channel. The channel will attenuate and phase shift each of the sinusoids. By comparing the received signal with the transmitted signal, the receiver obtains a measurement of the channel frequency response which can be used to design the inverse system. The process of determining the characteristics of the unknown system, either $h(n)$ or $H(\omega)$, by a set of measurements performed on the system is called *system identification*.

The term "deconvolution" is often used in seismic signal processing, and more generally, in geophysics to describe the operation of separating the input signal from the characteristics of the system which we wish to measure. The deconvolution operation is actually intended to identify the characteristics of the system, which in this case, is the earth, and can also be viewed as a system identification problem. The "inverse system," in this case, has a frequency response that is the reciprocal of the input signal spectrum that has been used to excite the system.

5.5.1 Invertibility of Linear Time-Invariant Systems

A system is said to be *invertible* if there is a one-to-one correspondence between its input and output signals. This definition implies that if we know the output sequence $y(n)$, $-\infty < n < \infty$, of an invertible system $\mathcal{T}$, we can uniquely determine its input $x(n)$, $-\infty < n < \infty$. The *inverse system* with input $y(n)$ and output $x(n)$ is denoted by $\mathcal{T}^{-1}$. Clearly, the cascade connection of a system and its inverse is equivalent to the identity system, since

$$w(n) = \mathcal{T}^{-1}[y(n)] = \mathcal{T}^{-1}\{\mathcal{T}[x(n)]\} = x(n) \tag{5.5.1}$$

as illustrated in Fig. 5.5.1. For example, the systems defined by the input–output relations $y(n) = ax(n)$ and $y(n) = x(n-5)$ are invertible, whereas the input–output relations $y(n) = x^2(n)$ and $y(n) = 0$ represent noninvertible systems.

As indicated above, inverse systems are important in many practical applications, including geophysics and digital communications. Let us begin by considering the problem of determining the inverse of a given system. We limit our discussion to the class of linear time-invariant discrete-time systems.

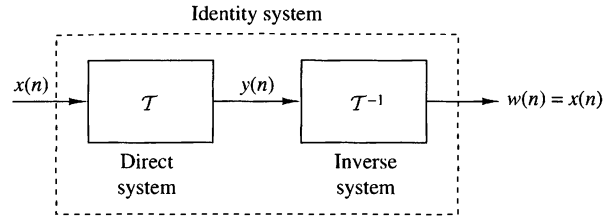


Figure 5.5.1
System $\mathcal{T}$ in cascade with its inverse $\mathcal{T}^{-1}$.

Now, suppose that the linear time-invariant system $\mathcal{T}$ has an impulse response $h(n)$ and let $h_I(n)$ denote the impulse response of the inverse system $\mathcal{T}^{-1}$. Then (5.5.1) is equivalent to the convolution equation

$$w(n) = h_I(n) * h(n) * x(n) = x(n) \tag{5.5.2}$$

But (5.5.2) implies that

$$h(n) * h_I(n) = \delta(n) \tag{5.5.3}$$

The convolution equation in (5.5.3) can be used to solve for $h_I(n)$ for a given $h(n)$. However, the solution of (5.5.3) in the time domain is usually difficult. A simpler approach is to transform (5.5.3) into the z-domain and solve for $\mathcal{T}^{-1}$. Thus in the z-transform domain, (5.5.3) becomes

$$H(z)H_I(z) = 1$$

and therefore the system function for the inverse system is

$$H_I(z) = \frac{1}{H(z)} \tag{5.5.4}$$

If $H(z)$ has a rational system function

$$H(z) = \frac{B(z)}{A(z)} \tag{5.5.5}$$

then

$$H_I(z) = \frac{A(z)}{B(z)} \tag{5.5.6}$$

Thus the zeros of $H(z)$ become the poles of the inverse system, and vice versa. Furthermore, if $H(z)$ is an FIR system, then $H_I(z)$ is an all-pole system, or if $H(z)$ is an all-pole system, then $H_I(z)$ is an FIR system.

EXAMPLE 5.5.1

Determine the inverse of the system with impulse response

$$h(n) = (\frac{1}{2})^n u(n)$$

Solution. The system function corresponding to $h(n)$ is

$$H(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}, \qquad \text{ROC: } |z| > \frac{1}{2}$$

This system is both causal and stable. Since $H(z)$ is an all-pole system, its inverse is FIR and is given by the system function

$$H_I(z) = 1 - \frac{1}{2}z^{-1}$$

Hence its impulse response is

$$h_I(n) = \delta(n) - \frac{1}{2}\delta(n-1)$$

EXAMPLE 5.5.2

Determine the inverse of the system with impulse response

$$h(n) = \delta(n) - \frac{1}{2}\delta(n-1)$$

This is an FIR system and its system function is

$$H(z) = 1 - \frac{1}{2}z^{-1}, \qquad \text{ROC: } |z| > 0$$

The inverse system has the system function

$$H_I(z) = \frac{1}{H(z)} = \frac{1}{1 - \frac{1}{2}z^{-1}} = \frac{z}{z - \frac{1}{2}}$$

Thus $H_I(z)$ has a zero at the origin and a pole at $z = \frac{1}{2}$. In this case there are two possible regions of convergence and hence two possible inverse systems, as illustrated in Fig. 5.5.2. If we take the ROC of $H_I(z)$ as $|z| > \frac{1}{2}$, the inverse transform yields

$$h_I(n) = (\frac{1}{2})^n u(n)$$

which is the impulse response of a causal and stable system. On the other hand, if the ROC is assumed to be $|z| < \frac{1}{2}$, the inverse system has an impulse response

$$h_I(n) = -\left(\frac{1}{2}\right)^n u(-n-1)$$

In this case the inverse system is anticausal and unstable.

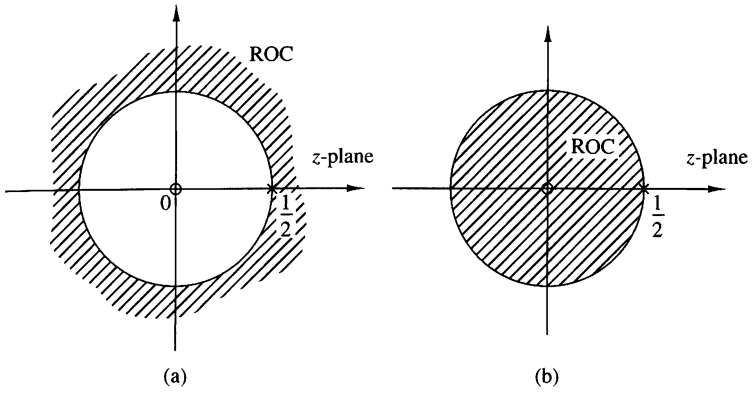


Figure 5.5.2 Two possible regions of convergence for $H(z) = z/(z - \frac{1}{2})$.

We observe that (5.5.3) cannot be solved uniquely by using (5.5.6) unless we specify the region of convergence for the system function of the inverse system.

In some practical applications the impulse response $h(n)$ does not possess a z-transform that can be expressed in closed form. As an alternative we may solve (5.5.3) directly using a digital computer. Since (5.5.3) does not, in general, possess a unique solution, we assume that the system and its inverse are causal. Then (5.5.3) simplifies to the equation

$$\sum_{k=0}^{n} h(k)h_I(n-k) = \delta(n) \tag{5.5.7}$$

By assumption, $h_I(n) = 0$ for $n < 0$. For $n = 0$ we obtain

$$h_I(0) = 1/h(0) \tag{5.5.8}$$

The values of $h_I(n)$ for $n \geq 1$ can be obtained recursively from the equation

$$h_I(n) = \sum_{k=1}^{n} \frac{h(n)h_I(n-k)}{h(0)}, \qquad n \geq 1 \tag{5.5.9}$$

This recursive relation can easily be programmed on a digital computer.

There are two problems associated with (5.5.9). First, the method does not work if $h(0) = 0$. However, this problem can easily be remedied by introducing an appropriate delay in the right-hand side of (5.5.7), that is, by replacing $\delta(n)$ by $\delta(n-m)$, where $m = 1$ if $h(0) = 0$ and $h(1) \neq 0$, and so on. Second, the recursion in (5.5.9) gives rise to round-off errors which grow with n and, as a result, the numerical accuracy of $h(n)$ deteriorates for large n.

EXAMPLE 5.5.3

Determine the causal inverse of the FIR system with impulse response

$$h(n) = \delta(n) - \alpha\delta(n-1)$$

Since $h(0) = 1$, $h(1) = -\alpha$, and $h(n) = 0$ for $n \geq \alpha$, we have

$$h_I(0) = 1/h(0) = 1$$

and

$$h_I(n) = \alpha h_I(n-1), \qquad n \geq 1$$

Consequently,

$$h_I(1) = \alpha, \quad h_I(2) = \alpha^2, \quad \ldots, \quad h_I(n) = \alpha^n$$

which corresponds to a causal IIR system as expected.

5.5.2 Minimum-Phase, Maximum-Phase, and Mixed-Phase Systems

The invertibility of a linear time-invariant system is intimately related to the characteristics of the phase spectral function of the system. To illustrate this point, let us consider two FIR systems, characterized by the system functions

$$H_1(z) = 1 + \frac{1}{2}z^{-1} = z^{-1}(z + \frac{1}{2}) \tag{5.5.10}$$

$$H_2(z) = \frac{1}{2} + z^{-1} = z^{-1}(\frac{1}{2}z + 1) \tag{5.5.11}$$

The system in (5.5.10) has a zero at $z = -\frac{1}{2}$ and an impulse response $h(0) = 1$, $h(1) = 1/2$. The system in (5.5.11) has a zero at $z = -2$ and an impulse response $h(0) = 1/2$, $h(1) = 1$, which is the reverse of the system in (5.5.10). This is due to the reciprocal relationship between the zeros of $H_1(z)$ and $H_2(z)$.

In the frequency domain, the two systems are characterized by their frequency response functions, which can be expressed as

$$|H_1(\omega)| = |H_2(\omega)| = \sqrt{\frac{5}{4} + \cos\omega} \tag{5.5.12}$$

and

$$\Theta_1(\omega) = -\omega + \tan^{-1}\frac{\sin\omega}{\frac{1}{2} + \cos\omega} \tag{5.5.13}$$

$$\Theta_2(\omega) = -\omega + \tan^{-1}\frac{\sin\omega}{2 + \cos\omega} \tag{5.5.14}$$

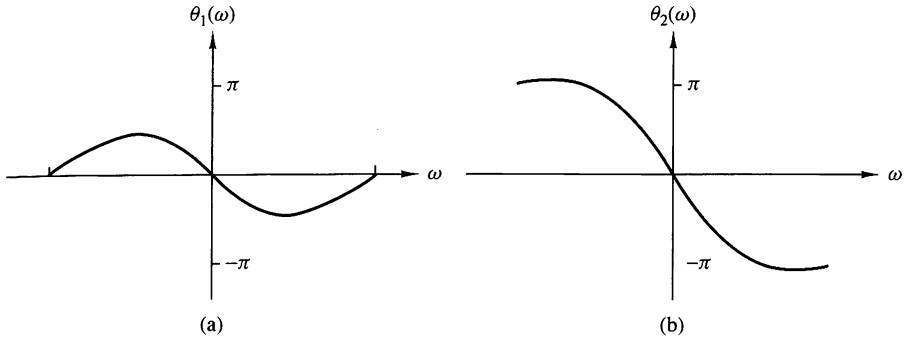


Figure 5.5.3 Phase response characteristics for the systems in (5.5.10). and (5.5.11).

The magnitude characteristics for the two systems are identical because the zeros of $H_1(z)$ and $H_2(z)$ are reciprocals.

The graphs of $\Theta_1(\omega)$ and $\Theta_2(\omega)$ are illustrated in Fig. 5.5.3. We observe that the phase characteristic $\Theta_1(\omega)$ for the first system begins at zero phase at the frequency $\omega = 0$ and terminates at zero phase at the frequency $\omega = \pi$. Hence the net phase change, $\Theta_1(\pi) - \Theta_1(0)$, is zero. On the other hand, the phase characteristic for the system with the zero outside the unit circle undergoes a net phase change $\Theta_2(\pi) - \Theta_2(0) = \pi$ radians. As a consequence of these different phase characteristics, we call the first system a *minimum-phase system* and the second system a *maximum-phase system.*

These definitions are easily extended to an FIR system of arbitrary length. To be specific, an FIR system of length $M + 1$ has M zeros. Its frequency response can be expressed as

$$H(\omega) = b_0(1 - z_1 e^{-j\omega})(1 - z_2 e^{-j\omega}) \cdots (1 - z_M e^{-j\omega}) \tag{5.5.15}$$

where $\{z_i\}$ denote the zeros and b_0 is an arbitrary constant. When all the zeros are inside the unit circle, each term in the product of (5.5.15), corresponding to a real-valued zero, will undergo a net phase change of zero between $\omega = 0$ and $\omega = \pi$. Also, each pair of complex-conjugate factors in $H(\omega)$ will undergo a net phase change of zero. Therefore,

$$\measuredangle H(\pi) - \measuredangle H(0) = 0 \tag{5.5.16}$$

and hence the system is called a minimum-phase system. On the other hand, when all the zeros are outside the unit circle, a real-valued zero will contribute a net phase change of π radians as the frequency varies from $\omega = 0$ to $\omega = \pi$, and each pair of complex-conjugate zeros will contribute a net phase change of 2π radians over the same range of ω. Therefore,

$$\measuredangle H(\pi) - \measuredangle H(0) = M\pi \tag{5.5.17}$$

which is the largest possible phase change for an FIR system with M zeros. Hence the system is called maximum phase. It follows from the discussion above that

$$\measuredangle H_{\max}(\pi) \geq \measuredangle H_{\min}(\pi) \tag{5.5.18}$$

If the FIR system with M zeros has some of its zeros inside the unit circle and the remaining zeros outside the unit circle, it is called a *mixed-phase system* or a *nonminimum-phase system.*

Since the derivative of the phase characteristic of the system is a measure of the time delay that signal frequency components undergo in passing through the system, a minimum-phase characteristic implies a minimum delay function, while a maximum-phase characteristic implies that the delay characteristic is also maximum.

Now suppose that we have an FIR system with real coefficients. Then the magnitude square value of its frequency response is

$$|H(\omega)|^2 = H(z)H(z^{-1})|_{z=e^{j\omega}} \tag{5.5.19}$$

This relationship implies that if we replace a zero z_k of the system by its inverse $1/z_k$, the magnitude characteristic of the system does not change. Thus if we reflect a zero z_k that is inside the unit circle into a zero $1/z_k$ outside the unit circle, we see that the magnitude characteristic of the frequency response is invariant to such a change.

It is apparent from this discussion that if $|H(\omega)|^2$ is the magnitude square frequency response of an FIR system having M zeros, there are 2^M possible configurations for the M zeros, of which some are inside the unit circle and the remaining are outside the unit circle. Clearly, one configuration has all the zeros inside the unit circle, which corresponds to the minimum-phase system. A second configuration has all the zeros outside the unit circle, which corresponds to the maximum-phase system. The remaining $2^M - 2$ configurations correspond to mixed-phase systems. However, not all $2^M - 2$ mixed-phase configurations necessarily correspond to FIR systems with real-valued coefficients. Specifically, any pair of complex-conjugate zeros results in only two possible configurations, whereas a pair of real-valued zeros yields four possible configurations.

EXAMPLE 5.5.4

Determine the zeros for the following FIR systems and indicate whether the system is minimum phase, maximum phase, or mixed phase.

$$H_1(z) = 6 + z^{-1} - z^{-2}$$

$$H_2(z) = 1 - z^{-1} - 6z^{-2}$$

$$H_3(z) = 1 - \frac{5}{2}z^{-1} - \frac{3}{2}z^{-2}$$

$$H_4(z) = 1 + \frac{5}{3}z^{-1} - \frac{2}{3}z^{-2}$$

Solution. By factoring the system functions we find the zeros for the four systems are

$$H_1(z) \longrightarrow z_{1,2} = -\frac{1}{2}, \frac{1}{3} \longrightarrow \text{minimum phase}$$

$$H_2(z) \longrightarrow z_{1,2} = -2, 3 \longrightarrow \text{maximum phase}$$

$$H_3(z) \longrightarrow z_{1,2} = -\frac{1}{2}, 3 \longrightarrow \text{mixed phase}$$

$$H_4(z) \longrightarrow z_{1,2} = -2, \frac{1}{3} \longrightarrow \text{mixed phase}$$

Since the zeros of the four systems are reciprocals of one another, it follows that all four systems have identical magnitude frequency response characteristics but different phase characteristics.

The minimum-phase property of FIR systems carries over to IIR systems that have rational system functions. Specifically, an IIR system with system function

$$H(z) = \frac{B(z)}{A(z)} \tag{5.5.20}$$

is called *minimum phase* if all its poles and zeros are inside the unit circle. For a stable and causal system [all roots of $A(z)$ fall inside the unit circle] the system is called *maximum phase* if all the zeros are outside the unit circle, and *mixed phase* if some, but not all, of the zeros are outside the unit circle.

This discussion brings us to an important point that should be emphasized. That is, a *stable* pole–zero system that is minimum phase has a stable inverse which is also minimum phase. The inverse system has the system function

$$H^{-1}(z) = \frac{A(z)}{B(z)} \tag{5.5.21}$$

Hence the minimum-phase property of $H(z)$ ensures the stability of the inverse system $H^{-1}(z)$ and the stability of $H(z)$ implies the minimum-phase property of $H^{-1}(z)$. Mixed-phase systems and maximum-phase systems result in unstable inverse systems.

Decomposition of nonminimum-phase pole–zero systems. Any nonminimum-phase pole–zero system can be expressed as

$$H(z) = H_{\min}(z)H_{\text{ap}}(z) \tag{5.5.22}$$

where $H_{\min}(z)$ is a minimum-phase system and $H_{\text{ap}}(z)$ is an all-pass system. We demonstrate the validity of this assertion for the class of causal and stable systems with a rational system function $H(z) = B(z)/A(z)$. In general, if $B(z)$ has one or more roots outside the unit circle, we factor $B(z)$ into the product $B_1(z)B_2(z)$, where $B_1(z)$ has all its roots inside the unit circle and $B_2(z)$ has all its roots outside the unit circle. Then $B_2(z^{-1})$ has all its roots inside the unit circle. We define the minimum-phase system

$$H_{\min}(z) = \frac{B_1(z)B_2(z^{-1})}{A(z)}$$

and the all-pass system

$$H_{\text{ap}}(z) = \frac{B_2(z)}{B_2(z^{-1})}$$

Thus $H(z) = H_{\min}(z)H_{\text{ap}}(z)$. Note that $H_{\text{ap}}(z)$ is a stable, all-pass, maximum-phase system.

Group delay of nonminimum-phase system. Based on the decomposition of a nonminimum-phase system given by (5.5.22), we can express the group delay of $H(z)$ as

$$\tau_g(\omega) = \tau_g^{\min}(\omega) + \tau_g^{\text{ap}}(\omega) \tag{5.5.23}$$

Since $\tau_g^{ap}(\omega) \geq 0$ for $0 \leq \omega \leq \pi$, it follows that $\tau_g(\omega) \geq \tau_g^{\min}(\omega)$, $0 \leq \omega \leq \pi$. From (5.5.23) we conclude that among all pole–zero systems having the same magnitude response, the minimum-phase system has the smallest group delay.

Partial energy of nonminimum-phase system. The *partial energy* of a causal system with impulse response $h(n)$ is defined as

$$E(n) = \sum_{k=0}^{n} |h(k)|^2 \tag{5.5.24}$$

It can be shown that among all systems having the same magnitude response and the same total energy $E(\infty)$, the minimum-phase system has the largest partial energy [i.e., $E_{\min}(n) \geq E(n)$, where $E_{\min}(n)$ is the partial energy of the minimum-phase system].

5.5.3 System Identification and Deconvolution

Suppose that we excite an unknown linear time-invariant system with an input sequence $x(n)$ and we observe the output sequence $y(n)$. From the output sequence we wish to determine the impulse response of the unknown system. This is a problem in *system identification,* which can be solved by *deconvolution.* Thus we have

$$\begin{aligned} y(n) &= h(n) * x(n) \\ &= \sum_{k=-\infty}^{\infty} h(k)x(n-k) \end{aligned} \tag{5.5.25}$$

An analytical solution of the deconvolution problem can be obtained by working with the z-transform of (5.5.25). In the z-transform domain we have

$$Y(z) = H(z)X(z)$$

and hence

$$H(z) = \frac{Y(z)}{X(z)} \tag{5.5.26}$$

$X(z)$ and $Y(z)$ are the z-transforms of the available input signal $x(n)$ and the observed output signal $y(n)$, respectively. This approach is appropriate only when there are closed-form expressions for $X(z)$ and $Y(z)$.

EXAMPLE 5.5.5

A causal system produces the output sequence

$$y(n) = \begin{cases} 1, & n = 0 \\ \frac{7}{10}, & n = 1 \\ 0, & \text{otherwise} \end{cases}$$

when excited by the input sequence

$$x(n) = \begin{cases} 1, & n = 0 \\ -\frac{7}{10}, & n = 1 \\ \frac{1}{10}, & n = 2 \\ 0, & \text{otherwise} \end{cases}$$

Determine its impulse response and its input–output equation.

Solution. The system function is easily determined by taking the z-transforms of $x(n)$ and $y(n)$. Thus we have

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + \frac{7}{10}z^{-1}}{1 - \frac{7}{10}z^{-1} + \frac{1}{10}z^{-2}}$$

$$= \frac{1 + \frac{7}{10}z^{-1}}{(1 - \frac{1}{2}z^{-1})(1 - \frac{1}{5}z^{-1})}$$

Since the system is causal, its ROC is $|z| > \frac{1}{2}$. The system is also stable since its poles lie inside the unit circle.

The input–output difference equation for the system is

$$y(n) = \frac{7}{10}y(n-1) - \frac{1}{10}y(n-2) + x(n) + \frac{7}{10}x(n-1)$$

Its impulse response is determined by performing a partial-fraction expansion of $H(z)$ and inverse transforming the result. This computation yields

$$h(n) = [4(\frac{1}{2})^n - 3(\frac{1}{5})^n]u(n)$$

We observe that (5.5.26) determines the unknown system uniquely if it is known that the system is causal. However, the example above is artificial, since the system response $\{y(n)\}$ is very likely to be infinite in duration. Consequently, this approach is usually impractical.

As an alternative, we can deal directly with the time-domain expression given by (5.5.25). If the system is causal, we have

$$y(n) = \sum_{k=0}^{n} h(k)x(n-k), \qquad n \geq 0$$

and hence

$$h(0) = \frac{y(0)}{x(0)}$$

$$h(n) = \frac{y(n) - \sum_{k=0}^{n-1} h(k)x(n-k)}{x(0)}, \qquad n \geq 1 \tag{5.5.27}$$

This recursive solution requires that $x(0) \neq 0$. However, we note again that when $\{h(n)\}$ has infinite duration, this approach may not be practical unless we truncate the recursive solution at same stage [i.e., truncate $\{h(n)\}$].

Another method for identifying an unknown system is based on a crosscorrelation technique. Recall that the input–output crosscorrelation function derived in Section 2.6.4 is given as

$$r_{yx}(m) = \sum_{k=0}^{\infty} h(k)r_{xx}(m-k) = h(n) * r_{xx}(m) \tag{5.5.28}$$

where $r_{yx}(m)$ is the crosscorrelation sequence of the input $\{x(n)\}$ to the system with the output $\{y(n)\}$ of the system, and $r_{xx}(m)$ is the autocorrelation sequence of the input signal. In the frequency domain, the corresponding relationship is

$$S_{yx}(\omega) = H(\omega)Sxx(\omega) = H(\omega)|X(\omega)|^2$$

Hence

$$H(\omega) = \frac{S_{yx}(\omega)}{S_{xx}(\omega)} = \frac{S_{yx}(\omega)}{|X(\omega)|^2} \tag{5.5.29}$$

These relations suggest that the impulse response $\{h(n)\}$ or the frequency response of an unknown system can be determined (measured) by crosscorrelating the input sequence $\{x(n)\}$ with the output sequence $\{y(n)\}$, and then solving the deconvolution problem in (5.5.28) by means of the recursive equation in (5.5.27). Alternatively, we could simply compute the Fourier transform of (5.5.28) and determine the frequency response given by (5.5.29). Furthermore, if we select the input sequence $\{x(n)\}$ such that its autocorrelation sequence $\{r_{xx}(n)\}$, is a unit sample sequence, or equivalently, that its spectrum is flat (constant) over the passband of $H(\omega)$, the values of the impulse response $\{h(n)\}$ are simply equal to the values of the crosscorrelation sequence $\{r_{yx}(n)\}$.

In general, the crosscorrelation method described above is an effective and practical method for system identification. Another practical approach based on least-squares optimization is described in Chapter 13.

5.5.4 Homomorphic Deconvolution

The complex cepstrum, introduced in Section 4.2.7, is a useful tool for performing deconvolution in some applications such as seismic signal processing. To describe this

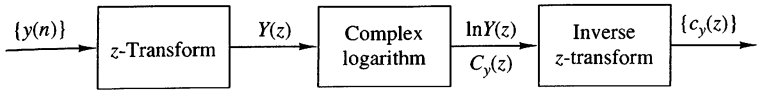


Figure 5.5.4 Homomorphic system for obtaining the cepstrum $\{c_y(n)\}$ of the sequence $\{y(n)\}$.

method, let us suppose that $\{y(n)\}$ is the output sequence of a linear time-invariant system which is excited by the input sequence $\{x(n)\}$. Then

$$Y(z) = X(z)H(z) \tag{5.5.30}$$

where $H(z)$ is the system function. The logarithm of $Y(z)$ is

$$\begin{aligned} C_y(z) &= \ln Y(z) \\ &= \ln X(z) + \ln H(z) \\ &= C_x(z) + C_h(z) \end{aligned} \tag{5.5.31}$$

Consequently, the complex cepstrum of the output sequence $\{y(n)\}$ is expressed as the sum of the cepstrum of $\{x(n)\}$ and $\{h(n)\}$, that is,

$$c_y(n) = c_x(n) + c_h(n) \tag{5.5.32}$$

Thus we observe that convolution of the two sequences in the time domain corresponds to the summation of the cepstrum sequences in the cepstral domain. The system for performing these transformations is called a *homormorphic system*and is illustrated in Fig. 5.5.4.

In some applications, such as seismic signal processing and speech signal processing, the characteristics of the cepstral sequences $\{c_x(n)\}$ and $\{c_h(n)\}$ are sufficiently different so that they can be separated in the cepstral domain. Specifically, suppose that $\{c_h(n)\}$ has its main components (main energy) in the vicinity of small values of n, whereas $\{c_x(n)\}$ has its components concentrated at large values of n. We may say that $\{c_h(n)\}$ is "lowpass" and $\{c_x(n)\}$ is "highpass." We can then separate $\{c_h(n)\}$ from $\{c_x(n)\}$ using appropriate "lowpass" and "highpass" windows, as illustrated in Fig. 5.5.5. Thus

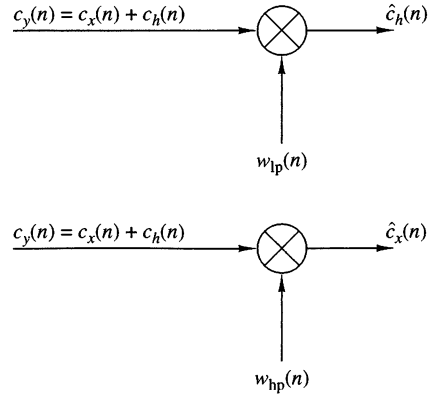


Figure 5.5.5
Separating the two cepstral components by "lowpass" and "highpass" windows.

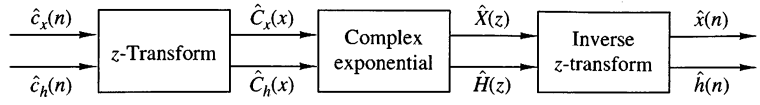


Figure 5.5.6 Inverse homomorphic system for recovering the sequences $\{x(n)\}$ and $\{h(n)\}$ from the corresponding cepstra.

$$\hat{c}_h(n) = c_y(n)w_{\text{lp}}(n) \tag{5.5.33}$$

and

$$\hat{c}_x(n) = c_y(n)w_{\text{hp}}(n) \tag{5.5.34}$$

where

$$w_{\text{lp}}(n) = \begin{cases} 1, & |n| \leq N_1 \\ 0, & \text{otherwise} \end{cases} \tag{5.5.35}$$

$$w_{\text{hp}}(n) = \begin{cases} 0, & |n| \leq N_1 \\ 1, & |n| > N_1 \end{cases} \tag{5.5.36}$$

Once we have separated the cepstrum sequences $\{\hat{c}_h(n)\}$ and $\{\hat{c}_x(n)\}$ by windowing, the sequences $\{\hat{x}(n)\}$ and $\{\hat{h}(n)\}$ are obtained by passing $\{\hat{c}_h(n)\}$ and $\{\hat{c}_x(n)\}$ through the inverse homomorphic system, shown in Fig. 5.5.6.

In practice, a digital computer would be used to compute the cepstrum of the sequence $\{y(n)\}$, to perform the windowing functions, and to implement the inverse homomorphic system shown in Fig. 5.5.6. In place of the z-transform and inverse z-transform, we would substitute a special form of the Fourier transform and its inverse. This special form, called the discrete Fourier transform, is described in Chapter 7.

5.6 Summary and References

In this chapter we considered the frequency-domain characteristics of LTI systems. We showed that an LTI system is characterized in the frequency domain by its frequency response function $H(\omega)$, which is the Fourier transform of the impulse response of the system. We also observed that the frequency response function determines the effect of the system on any input signal. In fact, by transforming the input signal into the frequency domain, we observed that it is a simple matter to determine the effect of the system on the signal and to determine the system output. When viewed in the frequency domain, an LTI system performs spectral shaping or spectral filtering on the input signal.

The design of some simple IIR filters was also considered in this chapter from the viewpoint of pole–zero placement. By means of this method, we were able to design simple digital resonators, notch filters, comb filters, all-pass filters, and digital sinusoidal generators. The design of more complex IIR filters is treated in detail in Chapter 10, which also includes several references. Digital sinusoidal generators find use in frequency synthesis applications. A comprehensive treatment of frequency synthesis techniques is given in the text edited by Gorski-Popiel (1975).

Finally, we characterized LTI systems as either minimum-phase, maximum-phase, or mixed-phase, depending on the position of their poles and zeros in the frequency domain. Using these basic characteristics of LTI systems, we considered practical problems in inverse filtering, deconvolution, and system identification. We concluded with the description of a deconvolution method based on cepstral analysis of the output signal from a linear system.

A vast amount of technical literature exists on the topics of inverse filtering, deconvolution, and system identification. In the context of communications, system identification and inverse filtering as they relate to channel equalization are treated in the book by Proakis (2001). Deconvolution techniques are widely used in seismic signal processing. For reference, we suggest the papers by Wood and Treitel (1975), Peacock and Treitel (1969), and the books by Robinson and Treitel (1978, 1980). Homomorphic deconvolution and its applications to speech processing aretreated in the book by Oppenheim and Schafer (1989).

Problems

5.1 The following input–output pairs have been observed during the operation of various systems:

(a) $x(n) = (\frac{1}{2})^n \xrightarrow{\mathcal{T}_1} y(n) = (\frac{1}{8})^n$

(b) $x(n) = (\frac{1}{2})^n u(n) \xrightarrow{\mathcal{T}_2} y(n) = (\frac{1}{8})^n u(n)$

(c) $x(n) = e^{j\pi/5} \xrightarrow{\mathcal{T}_3} y(n) = 3e^{j\pi/5}$

(d) $x(n) = e^{j\pi/5}u(n) \xrightarrow{\mathcal{T}_4} y(n) = 3e^{j\pi/5u(n)}$

(e) $x(n) = x(n + N_1) \xrightarrow{\mathcal{T}_5} y(n) = y(n + N_2), \qquad N_1 \neq N_2, \quad N_1, N_2$ prime

Determine their frequency response if each of the above systems is LTI.

5.2 **(a)** Determine and sketch the Fourier transform $W_R(\omega)$ of the rectangular sequence

$$w_R(n) = \begin{cases} 1, & 0 \leq n \leq M \\ 0, & \text{otherwise} \end{cases}$$

(b) Consider the triangular sequence

$$w_T(n) = \begin{cases} n, & 0 \leq n \leq M/2 \\ M - n, & M/2 < 2 \leq M \\ 0, & \text{otherwise} \end{cases}$$

Determine and sketch the Fourier transform $W_T(\omega)$ of $w_T(n)$ by expressing it as the convolution of a rectangular sequence with itself.

(c) Consider the sequence

$$w_c(n) = \frac{1}{2}\left(1 + \cos\frac{2\pi n}{M}\right) w_R(n)$$

Determine and sketch $W_c(\omega)$ by using $W_R(\omega)$.

5.3 Consider an LTI system with impulse response $h(n) = (\frac{1}{2})^n u(n)$.

(a) Determine and sketch the magnitude and phase response $|H(\omega)|$ and $\angle H(\omega)$, respectively.

(b) Determine and sketch the magnitude and phase spectra for the input and output signals for the following inputs:

1. $x(n) = \cos \dfrac{3\pi n}{10}, \quad -\infty < n < \infty$

2. $x(n) = \{\ldots, 1, 0, 0, \underset{\uparrow}{1}, 1, 1, 0, 1, 1, 1, 0, 1, \ldots\}$

5.4 Determine and sketch the magnitude and phase response of the following systems:

(a) $y(n) = \frac{1}{2}[x(n) + x(n-1)]$

(b) $y(n) = \frac{1}{2}[x(n) - x(n-1)]$

(c) $y(n) = \frac{1}{2}[x(n+1) - x(n-1)]$

(d) $y(n) = \frac{1}{2}[x(n+1) + x(n-1)]$

(e) $y(n) = \frac{1}{2}[x(n) + x(n-2)]$

(f) $y(n) = \frac{1}{2}[x(n) - x(n-2)]$

(g) $y(n) = \frac{1}{3}[x(n) + x(n-1) + x(n-2)]$

(h) $y(n) = x(n) - x(n-8)$

(i) $y(n) = 2x(n-1) - x(n-2)$

(j) $y(n) = \frac{1}{4}[x(n) + x(n-1) + x(n-2) + x(n-3)]$

(k) $y(n) = \frac{1}{8}[x(n) + 3x(n-1) + 3x(n-2) + x(n-3)]$

(l) $y(n) = x(n-4)$

(m) $y(n) = x(n+4)$

(n) $y(n) = \frac{1}{4}[x(n) - 2x(n-1) + x(n-2)]$

5.5 An FIR filter is described by the difference equation

$$y(n) = x(n) + x(n-10)$$

(a) Compute and sketch its magnitude and phase response.

(b) Determine its response to the inputs

1. $x(n) = \cos \dfrac{\pi}{10} n + 3 \sin\left(\dfrac{\pi}{3} n + \dfrac{\pi}{10}\right), \qquad -\infty < n < \infty$

2. $x(n) = 10 + 5 \cos\left(\dfrac{2\pi}{5} n + \dfrac{\pi}{2}\right), \qquad -\infty < n < \infty$

Determine the transient and steady-state responses of the FIR *filter* shown in Fig. P5.6 to the input signal $x(n) = 10e^{j\pi n/2}u(n)$. Let $b = 2$ and $y(-1) = y(-2) = y(-3) = y(-4) = 0$.

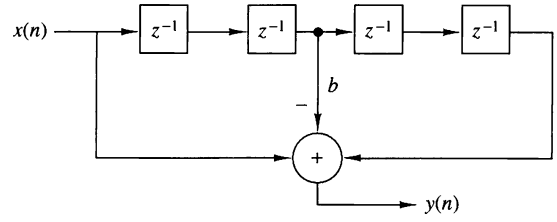


Figure P5.6

Consider the FIR filter

$$y(n) = x(n) + x(n-4)$$

(a) Compute and sketch its magnitude and phase response.

(b) Compute its response to the input

$$x(n) = \cos\frac{\pi}{2}n + \cos\frac{\pi}{4}n, \qquad -\infty < n < \infty$$

(c) Explain the results obtained in part (b) in terms of the magnitude and phase responses obtained in part (a).

Determine the steady-state and transient responses of the system

$$y(n) = \frac{1}{2}[x(n) - x(n-2)]$$

to the input signal

$$x(n) = 5 + 3\cos\left(\frac{\pi}{2}n + 60^\circ\right), \qquad -\infty < n < \infty$$

From our discussions it is apparent that an LTI system cannot produce frequencies at its output that are different from those applied in its input. Thus, if a system creates "new" frequencies, it must be nonlinear and/or time varying. Determine the frequency content of the outputs of the following systems to the input signal

$$x(n) = A\cos\frac{\pi}{4}n$$

(a) $y(n) = x(2n)$

(b) $y(n) = x^2(n)$

(c) $y(n) = (\cos\pi n)x(n)$

5.10 Determine and sketch the magnitude and phase response of the systems shown in Fig. P5.10(a) through (c).

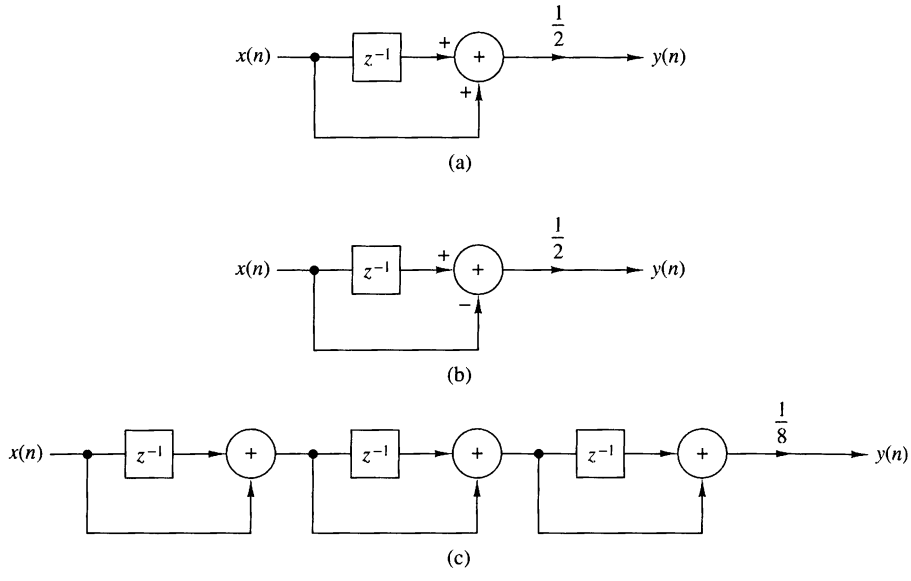


Figure P5.10

5.11 Determine the magnitude and phase response of the multipath channel

$$y(n) = x(n) + x(n - M)$$

At what frequencies does $H(\omega) = 0$?

5.12 Consider the filter

$$y(n) = 0.9y(n - 1) + bx(n)$$

(a) Determine b so that $|H(0)| = 1$.

(b) Determine the frequency at which $|H(\omega)| = 1/\sqrt{2}$.

(c) Is this filter lowpass, bandpass, or highpass?

(d) Repeat parts (b) and (c) for the filter $y(n) = -0.9y(n - 1) + 0.1x(n)$.

5.13 *Harmonic distortion in digital sinusoidal generators* An ideal sinusoidal generator produces the signal

$$x(n) = \cos 2\pi f_0 n, \qquad -\infty < n < \infty$$

which is periodic with fundamental period N if $f_0 = k_0/N$ and k_0, N are relatively prime numbers. The spectrum of such a "pure" sinusoid consist of two lines at $k = k_0$ and $k = N - k_0$ (we limit ourselves in the fundamental interval $0 \leq k \leq N - 1$). In practice, the approximations made in computing the samples of a sinusoid of relative frequency f_0 result in a certain amount of power falling into other frequencies. This spurious power results in distortion, which is referred to as *harmonic distortion*.

Harmonic distortion is usually measured in terms of the *total harmonic distortion* (THD), which is defined as the ratio

$$\text{THD} = \frac{\text{spurious harmonic power}}{\text{total power}}$$

(a) Show that

$$\text{THD} = 1 - 2\frac{|c_{k_0}|^2}{P_x}$$

where

$$c_{k_0} = \frac{1}{N}\sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)k_0 n}$$

$$P_x = \frac{1}{N}\sum_{n=0}^{N-1} |x(n)|^2$$

(b) By using the Taylor approximation

$$\cos\phi = 1 - \frac{\phi^2}{2!} + \frac{\phi^4}{4!} - \frac{\phi^6}{6!} + \cdots$$

compute one period of $x(n)$ for $f_0 = 1/96, 1/32, 1/256$ by increasing the number of terms in the Taylor expansion from 2 to 8.

(c) Compute the THD and plot the power density spectrum for each sinusoid in part (b) as well as for the sinusoids obtained using the computer cosine function. Comment on the results.

Measurement of the total harmonic distortion in quantized sinusoids Let $x(n)$ be a periodic sinusoidal signal with frequency $f_0 = k/N$, that is,

$$x(n) = \sin 2\pi f_0 n$$

(a) Write a computer program that quantizes the signal $x(n)$ into b bits or equivalently into $L = 2^b$ levels by using rounding. The resulting signal is denoted by $x_q(n)$.

(b) For $f_0 = 1/50$ compute the THD of the quantized signals $x_q(n)$ obtained by using $b = 4, 6, 8$, and 16 bits.

(c) Repeat part (b) for $f_0 = 1/100$.

(d) Comment on the results obtained in parts (b) and (c).

Consider the discrete-time system

$$y(n) = ay(n-1) + (1-a)x(n), \qquad n \geq 0$$

where $a = 0.9$ and $y(-1) = 0$.

(a) Compute and sketch the output $y_i(n)$ of the system to the input signals

$$x_i(n) = \sin 2\pi f_i n, \qquad 0 \le n \le 100$$

where $f_1 = \frac{1}{4}$, $f_2 = \frac{1}{5}$, $f_3 = \frac{1}{10}$, $f_4 = \frac{1}{20}$.

(b) Compute and sketch the magnitude and phase response of the system and use these results to explain the response of the system to the signals given in part (a).

5.16 Consider an LTI system with impulse response $h(n) = (\frac{1}{3})^{|n|}$.

(a) Determine and sketch the magnitude and phase response $H(\omega)|$ and $\measuredangle H(\omega)$, respectively.

(b) Determine and sketch the magnitude and phase spectra for the input and output signals for the following inputs:

1. $x(n) = \cos \dfrac{3\pi n}{8}, \quad -\infty < n < \infty$
2. $x(n) = \{\ldots, -1, 1, -1, \underset{\uparrow}{1}, -1, 1, -1, 1, -1, 1, -1, 1, \ldots\}$

5.17 Consider the digital filter shown in Fig. P5.17.

(a) Determine the input–output relation and the impulse response $h(n)$.

(b) Determine and sketch the magnitude $|H(\omega)|$ and the phase response $\measuredangle H(\omega)$ of the filter and find which frequencies are completely blocked by the filter.

(c) When $\omega_0 = \pi/2$, determine the output $y(n)$ to the input

$$x(n) = 3\cos\left(\frac{\pi}{3}n + 30^\circ\right), \qquad -\infty < n < \infty$$

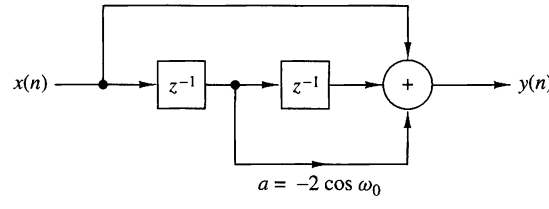


Figure P5.17

5.18 Consider the FIR filter

$$y(n) = x(n) - x(n-4)$$

(a) Compute and sketch its magnitude and phase response.

(b) Compute its response to the input

$$x(n) = \cos\frac{\pi}{2}n + \cos\frac{\pi}{4}n, \qquad -\infty < n < \infty$$

(c) Explain the results obtained in part (b) in terms of the answer given in part (a).

Determine the steady-state response of the system

$$y(n) = \frac{1}{2}[x(n) - x(n-2)]$$

to the input signal

$$x(n) = 5 + 3\cos\left(\frac{\pi}{2}n + 60°\right) + 4\sin(\pi n + 45°), \qquad -\infty < n < \infty$$

Recall from Problem 5.9 that an LTI system cannot produce frequencies at its output that are different from those applied in its input. Thus if a system creates "new" frequencies, it must be nonlinear and/or time varying. Indicate whether the following systems are nonlinear and/or time varying and determine the output spectra when the input spectrum is

$$X(\omega) = \begin{cases} 1, & |\omega| \le \pi/4 \\ 0, & \pi/4 \le |\omega| \le \pi \end{cases}$$

(a) $y(n) = x(2n)$

(b) $y(n) = x^2(n)$

(c) $y(n) = (\cos \pi n)x(n)$

Consider an LTI system with impulse response

$$h(n) = \left[\left(\frac{1}{4}\right)^n \cos\left(\frac{\pi}{4}n\right)\right]u(n)$$

(a) Determine its system function $H(z)$.

(b) Is it possible to implement this system using a finite number of adders, multipliers, and unit delays? If yes, how?

(c) Provide a rough sketch of $|H(\omega)|$ using the pole–zero plot.

(d) Determine the response of the system to the input

$$x(n) = (\frac{1}{4})^n u(n)$$

An FIR filter is described by the difference equation

$$y(n) = x(n) - x(n-6)$$

(a) Compute and sketch its magnitude and phase response.

(b) Determine its response to the inputs

1. $x(n) = \cos\frac{\pi}{10}n + 3\sin\left(\frac{\pi}{3}n + \frac{\pi}{10}\right), \qquad -\infty < n < \infty$

2. $x(n) = 5 + 6\cos\left(\frac{2\pi}{5}n + \frac{\pi}{2}\right), \qquad -\infty < n < \infty$

5.23 The frequency response of an ideal bandpass filter is given by

$$H(\omega) = \begin{cases} 0, & |\omega| \le \dfrac{\pi}{8} \\ 1, & \dfrac{\pi}{8} < |\omega| < \dfrac{3\pi}{8} \\ 0, & \dfrac{3\pi}{8} \le |\omega| \le \pi \end{cases}$$

(a) Determine its impulse response

(b) Show that this impulse response can be expressed as the product of $\cos(n\pi/4)$ and the impulse response of a lowpass filter.

5.24 Consider the system described by the difference equation

$$y(n) = \frac{1}{2}y(n-1) + x(n) + \frac{1}{2}x(n-1)$$

(a) Determine its impulse response.

(b) Determine its frequency response:

1. From the impulse response
2. From the difference equation

(c) Determine its response to the input

$$x(n) = \cos\left(\frac{\pi}{2}n + \frac{\pi}{4}\right), \qquad -\infty < n < \infty$$

5.25 Sketch roughly the magnitude $|H(\omega)|$ of the Fourier transforms corresponding to the pole–zero patterns of systems given in Fig. P5.25.

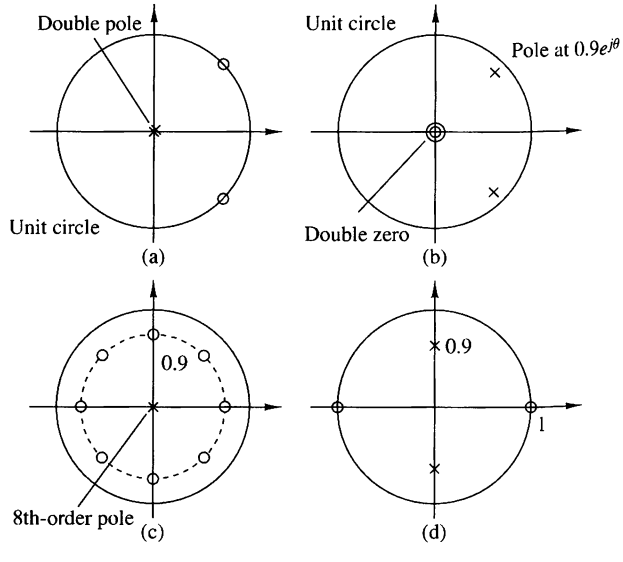


Figure P5.25

Design an FIR filter that completely blocks the frequency $\omega_0 = \pi/4$ and then compute its output if the input is

$$x(n) = \left(\sin \frac{\pi}{4} n\right) u(n)$$

for $n = 0, 1, 2, 3, 4$. Does the filter fulfill your expectations? Explain.

A digital filter is characterized by the following properties:

1. It is highpass and has one pole and one zero.
2. The pole is at a distance $r = 0.9$ from the origin of the z-plane.
3. Constant signals do not pass through the system.

(a) Plot the pole–zero pattern of the filter and determine its system function $H(z)$.

(b) Compute the magnitude response $|H(\omega)|$ and the phase response $\measuredangle H(\omega)$ of the filter.

(c) Normalize the frequency response $H(\omega)$ so that $|H(\pi)| = 1$.

(d) Determine the input–output relation (difference equation) of the filter in the time domain.

(e) Compute the output of the system if the input is

$$x(n) = 2\cos\left(\frac{\pi}{6} n + 45^\circ\right), \qquad -\infty < n < \infty$$

(You can use either algebraic or geometrical arguments.)

A causal first-order digital filter is described by the system function

$$H(z) = b_0 \frac{1 + bz^{-1}}{1 + az^{-1}}$$

(a) Sketch the direct form I and direct form II realizations of this filter and find the corresponding difference equations.

(b) For $a = 0.5$ and $b = -0.6$, sketch the pole–zero pattern. Is the system stable? Why?

(c) For $a = -0.5$ and $b = 0.5$, determine b_0, so that the maximum value of $|H(\omega)|$ is equal to 1.

(d) Sketch the magnitude response $|H(\omega)|$ and the phase response $\measuredangle H(\omega)$ of the filter obtained in part (c).

(e) In a specific application it is known that $a = 0.8$. Does the resulting filter amplify high frequencies or low frequencies in the input? Choose the value of b so as to improve the characteristics of this filter (i.e., make it a better lowpass or a better highpass filter).

Derive the expression for the resonant frequency of a two-pole filter with poles at $p_1 = re^{j\theta}$ and $p_2 = p_1^*$, given by (5.4.25).

5.30 Determine and sketch the magnitude and phase responses of the Hanning filter characterized by the (moving average) difference equation

$$y(n) = \frac{1}{4}x(n) + \frac{1}{2}x(n-1) + \frac{1}{4}x(n-2)$$

5.31 A causal LTI system excited by the input

$$x(n) = (\frac{1}{4})^n u(n) + u(-n-1)$$

produces an output $y(n)$ with z-transform

$$Y(z) = \frac{-\frac{3}{4}z^{-1}}{(1-\frac{1}{4}z^{-1})(1+z^{-1})}$$

(a) Determine the system function $H(z)$ and its ROC.

(b) Determine the output $y(n)$ of the system.
(*Hint:* Pole cancellation increases the original ROC.)

5.32 Determine the coefficients of a linear-phase FIR filter

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$$

such that:

(a) It rejects completely a frequency component at $\omega_0 = 2\pi/3$.

(b) Its frequency response is normalized so that $H(0) = 1$.

(c) Compute and sketch the magnitude and phase response of the filter to check if it satisfies the requirements.

5.33 Determine the frequency response $H(\omega)$ of the following moving average filters.

(a) $y(n) = \dfrac{1}{2M+1}\displaystyle\sum_{k=-M}^{M} x(n-k)$

(b) $y(n) = \dfrac{1}{4M}x(n+M) + \dfrac{1}{2M}\displaystyle\sum_{k=-M+1}^{M-1} x(n-k) + \dfrac{1}{4M}x(n-M)$

Which filter provides better smoothing? Why?

5.34 Compute the magnitude and phase response of a filter with system function

$$H(z) = 1 + z^{-1} + z^{-2} + \cdots + z^{-8}$$

If the sampling frequency is $F_s = 1$ kHz, determine the frequencies of the analog sinusoids that cannot pass through the filter.

5.35 A second-order system has a double pole at $p_{1,2} = 0.5$ and two zeros at

$$z_{1,2} = e^{\pm j3\pi/4}$$

Using geometric arguments, choose the gain G of the filter so that $|H(0)| = 1$.

In this problem we consider the effect of a single zero on the frequency response of a system. Let $z = re^{j\theta}$ be a zero inside the unit circle ($r < 1$). Then

$$\begin{aligned} H_z(\omega) &= 1 - re^{j\theta}e^{-j\omega} \\ &= 1 - r\cos(\omega - \theta) + jr\sin(\omega - \theta) \end{aligned}$$

(a) Show that the magnitude response is

$$|H_z(\omega)| = [1 - 2r\cos(\omega - \theta) + r^2]^{1/2}$$

or, equivalently,

$$20\log_{10}|H_z(\omega)| = 10\log_{10}[1 - 2r\cos(\omega - \theta) + r^2]$$

(b) Show that the phase response is given as

$$\Theta_z(\omega) = \tan^{-1}\frac{r\sin(\omega - \theta)}{1 - r\cos(\omega - \theta)}$$

(c) Show that the group delay is given as

$$\tau_g(\omega) = \frac{r^2 - r\cos(\omega - \theta)}{1 + r^2 - 2r\cos(\omega - \theta)}$$

(d) Plot the magnitude $|H(\omega)|_{\text{dB}}$, the phase $\Theta(\omega)$ and the group delay $\tau_g(\omega)$ for $r = 0.7$ and $\theta = 0, \pi/2$, and π.

In this problem we consider the effect of a single pole on the frequency response of a system. Hence, we let

$$H_p(\omega) = \frac{1}{1 - re^{j\theta}e^{-j\omega}}, \qquad r < 1$$

Show that

$$\begin{aligned} |H_p(\omega)|_{\text{dB}} &= -|H_z(\omega)|_{\text{dB}} \\ \measuredangle H_p(\omega) &= -\measuredangle H_z(\omega) \\ \tau_g^p(\omega) &= -\tau_g^z(\omega) \end{aligned}$$

where $H_z(\omega)$ and $\tau_g^z(\omega)$ are defined in Problem 5.36.

In this problem we consider the effect of complex-conjugate pairs of poles and zeros on the frequency response of a system. Let

$$H_z(\omega) = (1 - re^{j\theta}e^{-j\omega})(1 - re^{-j\theta}e^{-j\omega})$$

(a) Show that the magnitude response in decibels is

$$|H_z(\omega)|_{\text{dB}} = 10\log_{10}[1 + r^2 - 2r\cos(\omega - \theta)] + 10\log_{10}[1 + r^2 - 2r\cos(\omega + \theta)]$$

(b) Show that the phase response is given as

$$\Theta_z(\omega) = \tan^{-1}\frac{r\sin(\omega - \theta)}{1 - r\cos(\omega - \theta)} + \tan^{-1}\frac{r\sin(\omega + \theta)}{1 - r\cos(\omega + \theta)}$$

(c) Show that the group delay is given as

$$\tau_g^z(\omega) = \frac{r^2 - r\cos(\omega - \theta)}{1 + r^2 - 2r\cos(\omega - \theta)} + \frac{r^2 - r\cos(\omega + \theta)}{1 + r^2 - 2r\cos(\omega + \theta)}$$

(d) If $H_p(\omega) = 1/H_z(\omega)$, show that

$$|H_p(\omega)|_{\text{dB}} = -|H_z(\omega)|_{\text{dB}}$$

$$\Theta_p(\omega) = -\Theta_z(\omega)$$

$$\tau_g^p(\omega) = -\tau_g^z(\omega)$$

(e) Plot $|H_p(\omega)|$, $\Theta_p(\omega)$ and $\tau_g^p(\omega)$ for $\tau = 0.9$, and $\theta = 0, \pi/2$.

5.39 Determine the 3-dB bandwidth of the filters $(0 < a < 1)$

$$H_1(z) = \frac{1 - a}{1 - az^{-1}}$$

$$H_2(z) = \frac{1 - a}{2}\frac{1 + z^{-1}}{1 - az^{-1}}$$

Which is a better lowpass filter?

5.40 Design a digital oscillator with adjustable phase, that is, a digital filter which produces the signal

$$y(n) = \cos(\omega_0 n + \theta)u(n)$$

5.41 This problem provides another derivation of the structure for the coupled-form oscillator by considering the system

$$y(n) = ay(n - 1) + x(n)$$

for $a = e^{j\omega_0}$.

Let $x(n)$ be real. Then $y(n)$ is complex. Thus

$$y(n) = y_R(n) + jy_I(n)$$

(a) Determine the equations describing a system with one input $x(n)$ and the two outputs $y_R(n)$ and $y_I(n)$.

(b) Determine a block diagram realization

(c) Show that if $x(n) = \delta(n)$, then

$$y_R(n) = (\cos \omega_0 n)u(n)$$

$$y_I(n) = (\sin \omega_0 n)u(n)$$

(d) Compute $y_R(n)$, $y_I(n)$, $n = 0, 1, \ldots, 9$ for $\omega_0 = \pi/6$. Compare these with the true values of the sine and cosine.

Consider a filter with system function

$$H(z) = b_0 \frac{(1 - e^{j\omega_0} z^{-1})(1 - e^{-j\omega_0} z^{-1})}{(1 - re^{j\omega_0} z^{-1})(1 - re^{-j\omega_0} z^{-1})}$$

(a) Sketch the pole–zero pattern.

(b) Using geometric arguments, show that for $r \simeq 1$, the system is a notch filter and provide a rough sketch of its magnitude response if $\omega_0 = 60°$.

(c) For $\omega_0 = 60°$, choose b_0 so that the maximum value of $|H(\omega)|$ is 1.

(d) Draw a direct form II realization of the system.

(e) Determine the approximate 3-dB bandwidth of the system.

Design an FIR digital filter that will reject a very strong 60-Hz sinusoidal interference contaminating a 200-Hz useful sinusoidal signal. Determine the gain of the filter so that the useful signal does not change amplitude. The filter works at a sampling frequency $F_s = 500$ samples/s. Compute the output of the filter if the input is a 60-Hz sinusoid or a 200-Hz sinusoid with unit amplitude. How does the performance of the filter compare with your requirements?

Determine the gain b_0 for the digital resonator described by (5.4.28) so that $|H(\omega_0)| = 1$.

Demonstrate that the difference equation given in (5.4.52) can be obtained by applying the trigonometric identity

$$\cos \alpha + \cos \beta = 2 \cos \frac{\alpha + \beta}{2} \cos \frac{\alpha - \beta}{2}$$

where $\alpha = (n + 1)\omega_0$, $\beta = (n - 1)\omega_0$, and $y(n) = \cos \omega_0 n$. Thus show that the sinusoidal signal $y(n) = A \cos \omega_0 n$ can be generated from (5.4.52) by use of the initial conditions $y(-1) = A \cos \omega_0$ and $y(-2) = A \cos 2\omega_0$.

Use the trigonometric identity in (5.4.53) with $\alpha = n\omega_0$ and $\beta = (n - 2)\omega_0$ to derive the difference equation for generating the sinusoidal signal $y(n) = A \sin n\omega_0$. Determine the corresponding initial conditions.

Using the z-transform pairs 8 and 9 in Table 3.3, determine the difference equations for the digital oscillators that have impulse responses $h(n) = A \cos n\omega_0 u(n)$ and $h(n) = A \sin n\omega_0 u(n)$, respectively.

Determine the structure for the coupled-form oscillator by combining the structure for the digital oscillators obtained in Problem 5.47.

5.49 Convert the highpass filter with system function

$$H(z) = \frac{1 - z^{-1}}{1 - az^{-1}}, \qquad a < 1$$

into a notch filter that rejects the frequency $\omega_0 = \pi/4$ and its harmonics.

(a) Determine the difference equation.

(b) Sketch the pole–zero pattern.

(c) Sketch the magnitude response for both filters.

5.50 Choose L and M for a lunar filter that must have narrow passbands at $(k \pm \Delta F)$ cycles/day, where $k = 1, 2, 3, \ldots$ and $\Delta F = 0.067726$.

5.51 **(a)** Show that the systems corresponding to the pole–zero patterns of Fig. 5.4.16 are all-pass.

(b) What is the number of delays and multipliers required for the efficient implementation of a second-order all-pass system?

5.52 A digital notch filter is required to remove an undesirable 60-Hz hum associated with a power supply in an ECG recording application. The sampling frequency used is $F_s = 500$ samples/s. **(a)** Design a second-order FIR notch filter and **(b)** a second-order pole–zero notch filter for this purpose. In both cases choose the gain b_0 so that $|H(\omega)| = 1$ for $\omega = 0$.

5.53 Determine the coefficients $\{h(n)\}$ of a highpass linear phase FIR filter of length $M = 4$ which has an antisymmetric unit sample response $h(n) = -h(M - 1 - n)$ and a frequency response that satisfies the condition

$$\left|H\left(\frac{\pi}{4}\right)\right| = \frac{1}{2}, \qquad \left|H\left(\frac{3\pi}{4}\right)\right| = 1$$

5.54 In an attempt to design a four-pole bandpass digital filter with desired magnitude response

$$|H_d(\omega)| = \begin{cases} 1, & \dfrac{\pi}{6} \le \omega \le \dfrac{\pi}{2} \\ 0, & \text{elsewhere} \end{cases}$$

we select the four poles at

$$p_{1,2} = 0.8e^{\pm j^2\pi/9}$$

$$p_{3,4} = 0.8e^{\pm j^4\pi/9}$$

and four zeros at

$$z_1 = 1, \qquad z_2 = -1, \qquad z_{3,4} = e^{\pm 3\pi/4}$$

(a) Determine the value of the gain so that

$$\left|H\left(\frac{5\pi}{12}\right)\right| = 1$$

(b) Determine the system function $H(z)$.

(c) Determine the magnitude of the frequency response $H(\omega)$ for $0 \le \omega \le \pi$ and compare it with the desired response $|H_d(\omega)|$.

5.55 A discrete-time system with input $x(n)$ and output $y(n)$ is described in the frequency domain by the relation

$$Y(\omega) = e^{-j2\pi\omega}X(\omega) + \frac{dX(\omega)}{d\omega}$$

(a) Compute the response of the system to the input $x(n) = \delta(n)$.

(b) Check if the system is LTI and stable.

5.56 Consider an ideal lowpass filter with impulse response $h(n)$ and frequency response

$$H(\omega) = \begin{cases} 1, & |\omega| \leq \omega_c \\ 0, & \omega_c < |\omega| < \pi \end{cases}$$

What is the frequency response of the filter defined by

$$g(n) = \begin{cases} h\left(\frac{n}{2}\right), & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$$

5.57 Consider the system shown in Fig. P5.57. Determine its impulse response and its frequency response if the system $H(\omega)$ is:

(a) Lowpass with cutoff frequency ω_c.

(b) Highpass with cutoff frequency ω_c.

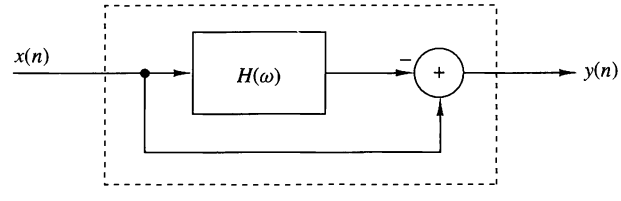


Figure P5.57

5.58 Frequency inverters have been used for many years for speech scrambling. Indeed, a voice signal $x(n)$ becomes unintelligible if we invert its spectrum as shown in Fig. P5.58.

(a) Determine how frequency inversion can be performed in the time domain.

(b) Design an unscrambler. (*Hint:* The required operations are very simple and can easily be done in real time.)

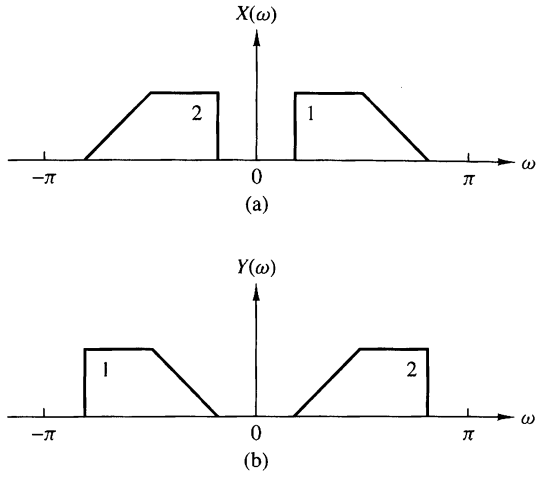


Figure P5.58
(a) Original spectrum;
(b) frequency-inverted spectrum.

5.59 A lowpass filter is described by the difference equation

$$y(n) = 0.9y(n-1) + 0.1x(n)$$

(a) By performing a frequency translation of $\pi/2$, transform the filter into a bandpass filter.

(b) What is the impulse response of the bandpass filter?

(c) What is the major problem with the frequency translation method for transforming a prototype lowpass filter into a bandpass filter?

5.60 Consider a system with a real-valued impulse response $h(n)$ and frequency response

$$H(\omega) = |H(\omega)|e^{j\theta(\omega)}$$

The quantity

$$D = \sum_{n=-\infty}^{\infty} n^2 h^2(n)$$

provides a measure of the "effective duration" of $h(n)$.

(a) Express D in terms of $H(\omega)$.

(b) Show that D is minimized for $\theta(\omega) = 0$.

5.61 Consider the lowpass filter

$$y(n) = ay(n-1) + bx(n), \qquad 0 < a < 1$$

(a) Determine b so that $|H(0)| = 1$.

(b) Determine the 3-dB bandwidth ω_3 for the normalized filter in part (a).

(c) How does the choice of the parameter a affect ω_3?

(d) Repeat parts (a) through (c) for the highpass filter obtained by choosing $-1 < a < 0$.

Sketch the magnitude and phase response of the multipath channel

$$y(n) = x(n) + \alpha x(n - M), \qquad \alpha > 0$$

for $\alpha << 1$.

Determine the system functions and the pole-zero locations for the systems shown in Fig. P5.63(a) through (c), and indicate whether or not the systems are stable.

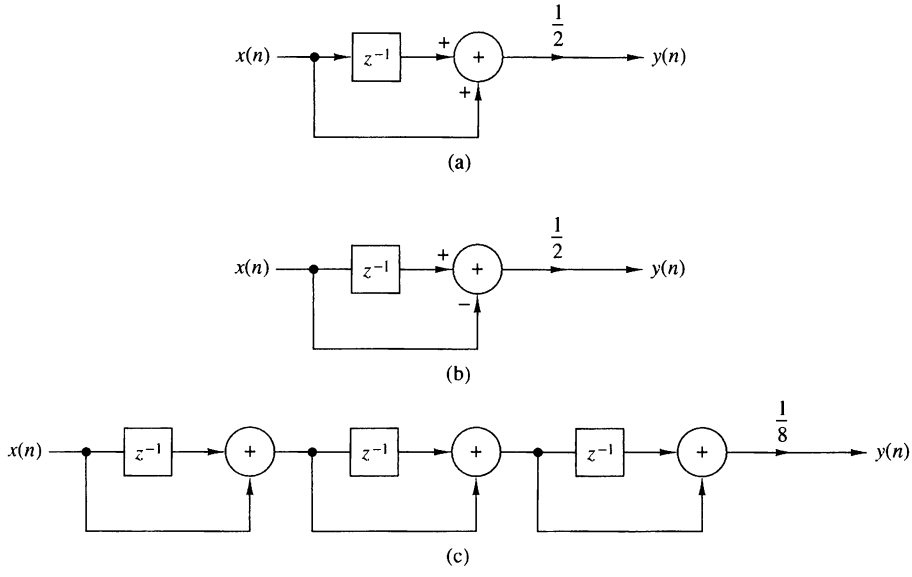


Figure P5.63

Determine and sketch the impulse response and the magnitude and phase responses of the FIR filter shown in Fig. P5.64 for $b = 1$ and $b = -1$.

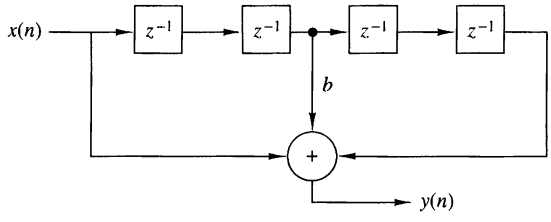


Figure P5.64

Consider the system

$$y(n) = x(n) - 0.95x(n - 6)$$

(a) Sketch its pole–zero pattern.

(b) Sketch its magnitude response using the pole–zero plot.

(c) Determine the system function of its causal inverse system.

(d) Sketch the magnitude response of the inverse system using the pole–zero plot.

5.66 Determine the impulse response and the difference equation for all possible systems specified by the system functions

(a) $H(z) = \dfrac{z^{-1}}{1 - z^{-1} - z^{-2}}$

(b) $H(z) = \dfrac{1}{1 - e^{-4a}z^{-4}}, \qquad 0 < a < 1$

5.67 Determine the impulse response of a causal LTI system which produces the response

$$y(n) = \{\underset{\uparrow}{1}, -1, 3, -1, 6\}$$

when excited by the input signal

$$x(n) = \{\underset{\uparrow}{1}, 1, 2\}$$

5.68 The system

$$y(n) = \frac{1}{2}y(n-1) + x(n)$$

is excited with the input

$$x(n) = (\frac{1}{4})^n u(n)$$

Determine the sequences $r_{xx}(l)$, $r_{hh}(l)$, $r_{xy}(l)$, and $r_{yy}(l)$.

5.69 Determine if the following FIR systems are minimum phase.

(a) $h(n) = \{\underset{\uparrow}{10}, 9, -7, -8, 0, 5, 3\}$

(b) $h(n) = \{\underset{\uparrow}{5}, 4, -3, -4, 0, 2, 1\}$

5.70 Can you determine the coefficients of the all-pole system

$$H(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

if you know its order N and the values $h(0), h(1), \ldots, h(L-1)$ of its impulse response? How? What happens if you do not know N?

5.71 Consider a system with impulse response

$$h(n) = b_0\delta(n) + b_1\delta(n - D) + b_2\delta(n - 2D)$$

(a) Explain why the system generates echoes spaced D samples apart.

(b) Determine the magnitude and phase response of the system.

(c) Show that for $|b_0 + b_2| << |b_1|$, the locations of maxima and minima of $|H(\omega)|^2$ are at

$$\omega = \pm\frac{k}{D}\pi, \qquad k = 0, 1, 2, \ldots$$

(d) Plot $|H(\omega)|$ and $\measuredangle H(\omega)$ for $b_0 = 0.1$, $b_1 = 1$, and $b_2 = 0.05$ and discuss the results.

5.72 Consider the pole–zero system

$$H(z) = \frac{B(z)}{A(z)} = \frac{1 + bz^{-1}}{1 + az^{-1}} = \sum_{n=0}^{\infty} h(n)z^{-n}$$

(a) Determine $h(0)$, $h(1)$, $h(2)$, and $h(3)$ in terms of a and b.

(b) Let $r_{hh}(l)$ be the autocorrelation sequence of $h(n)$. Determine $r_{hh}(0)$, $r_{hh}(1)$, $r_{hh}(2)$, and $r_{hh}(3)$ in terms of a and b.

5.73 Let $x(n)$ be a real-valued minimum-phase sequence. Modify $x(n)$ to obtain another real-valued minimum-phase sequence $y(n)$ such that $y(0) = x(0)$ and $y(n) = |x(n)|$.

5.74 The frequency response of a stable LTI system is known to be real and even. Is the inverse system stable?

5.75 Let $h(n)$ be a real filter with nonzero linear or nonlinear phase response. Show that the following operations are equivalent to filtering the signal $x(n)$ with a zero-phase filter.

(a) $g(n) = h(n) * x(n)$

$f(n) = h(n) * g(-n)$

$y(n) = f(-n)$

(b) $g(n) = h(n) * x(n)$

$f(n) = h(n) * x(-n)$

$y(n) = g(n) + f(-n)$

(*Hint:* Determine the frequency response of the composite system $y(n) = H[x(n)]$.)

5.76 Check the validity of the following statements:

(a) The convolution of two minimum-phase sequences is always a minimum-phase sequence.

(b) The sum of two minimum-phase sequences is always minimum phase.

5.77 Determine the minimum-phase system whose squared magnitude response is given by:

(a) $|H(\omega)|^2 = \dfrac{\frac{5}{4} - \cos\omega}{\frac{10}{9} - \frac{2}{3}\cos\omega}$

(b) $|H(\omega)|^2 = \dfrac{2(1 - a^2)}{(1 + a^2) - 2a\cos\omega}, \qquad |a| < 1$

5.78 Consider an FIR system with the following system function:

$$H(z) = (1 - 0.8e^{j\pi/2}z^{-1})(1 - 0.8e^{-j\pi/2}z^{-1})(1 - 1.5e^{j\pi/4}z^{-1})(1 - 1.5e^{-j\pi/4}z^{-1})$$

(a) Determine all systems that have the same magnitude response. Which is the minimum-phase system?

(b) Determine the impulse response of all systems in part (a).

(c) Plot the partial energy

$$E(n) = \sum_{k=0}^{n} h^2(n)$$

for every system and use it to identify the minimum- and maximum-phase systems.

5.79 The causal system

$$H(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

is known to be unstable.

We modify this system by changing its impulse response $h(n)$ to

$$h'(n) = \lambda^n h(n) u(n)$$

(a) Show that by properly choosing λ we can obtain a new stable system.

(b) What is the difference equation describing the new system?

5.80 Given a signal $x(n)$, we can create echoes and reverberations by delaying and scaling the signal as follows

$$y(n) = \sum_{k=0}^{\infty} g_k x(n - kD)$$

where D is positive integer and $g_k > g_{k-1} > 0$.

(a) Explain why the comb filter

$$H(z) = \frac{1}{1 - az^{-D}}$$

can be used as a reverberator (i.e., as a device to produce artificial reverberations). (*Hint:* Determine and sketch its impulse response.)

(b) The all-pass comb filter

$$H(z) = \frac{z^{-D} - a}{1 - az^{-D}}$$

is used in practice to build digital reverberators by cascading three to five such filters and properly choosing the parameters a and D. Compute and plot the impulse response of two such reverberators each obtained by cascading three sections with the following parameters.

UNIT 1

Section	D	a
1	50	0.7
2	40	0.665
3	32	0.63175

UNIT 2

Section	D	a
1	50	0.7
2	17	0.77
3	6	0.847

(c) The difference between echo and reverberation is that with pure echo there are clear repetitions of the signal, but with reverberations, there are not. How is this reflected in the shape of the impulse response of the reverberator? Which unit in part (b) is a better reverberator?

(d) If the delays D_1, D_2, D_3 in a certain unit are prime numbers, the impulse response of the unit is more "dense." Explain why.

(e) Plot the phase response of units 1 and 2 and comment on them.

(f) Plot $h(n)$ for D_1, D_2, and D_3 being nonprime. What do you notice?

More details about this application can be found in a paper by J. A. Moorer, "Signal Processing Aspects of Computer Music: A Survey," *Proc. IEEE,* Vol. 65, No. 8, Aug. 1977, pp. 1108–1137.

5.81 By trial-and-error design a third-order lowpass filter with cutoff frequency at $\omega_c = \pi/9$ radians/sample interval. Start your search with

(a) $z_1 = z_2 = z_3 = 0$, $p_1 = r$, $p_{2,3} = re^{\pm j\omega_c}$, $r = 0.8$

(b) $r = 0.9$, $z_1 = z_2 = z_3 = -1$

5.82 A speech signal with bandwidth $B = 10$ kHz is sampled at $F_2 = 20$ kHz. Suppose that the signal is corrupted by four sinusoids with frequencies

$$F_1 = 10{,}000 \text{ Hz}, \quad F_3 = 7778 \text{ Hz}$$
$$F_2 = 8889 \text{ Hz}, \quad F_4 = 6667 \text{ Hz}$$

(a) Design a FIR filter that eliminates these frequency components.

(b) Choose the gain of the filter so that $|H(0)| = 1$ and then plot the log magnitude response and the phase response of the filter.

(c) Does the filter fulfill your objectives? Do you recommend the use of this filter in a practical application?

5.83 Compute and sketch the frequency response of a digital resonator with $\omega = \pi/6$ and $r = 0.6, 0.9, 0.99$. In each case, compute the bandwidth and the resonance frequency from the graph, and check if they are in agreement with the theoretical results.

5.84 The system function of a communication channel is given by

$$H(z) = (1 - 0.9e^{j0.4\pi}z^{-1})(1 - 0.9e^{-j0.4\pi}z^{-1})(1 - 1.5e^{j0.6\pi}z^{-1})(1 - 1.5e^{-j0.6\pi}z^{-1})$$

Determine the system function $H_c(z)$ of a causal and stable compensating system so that the cascade interconnection of the two systems has a flat magnitude response. Sketch the pole–zero plots and the magnitude and phase responses of all systems involved into the analysis process. [*Hint:* Use the decomposition $H(z) = H_{\text{ap}}(z)H_{\text{min}}(z)$.]

CHAPTER

6

Sampling and Reconstruction of Signals

In Chapter 1 we treated the sampling of continuous-time signals and demonstrated that if the signals are bandlimited, it is possible to reconstruct the original signal from the samples, provided that the sampling rate is at least twice the highest frequency contained in the signal. We also briefly described the subsequent operations of quantization and coding that are necessary to convert an analog signal to a digital signal appropriate for digital processing.

In this chapter we consider time-domain sampling, analog-to-digital (A/D) conversion (quantization and coding), and digital-to-analog (D/A) conversion (signal reconstruction) in greater depth. We also consider the sampling of signals that are characterized as bandpass signals. The final topic deals with the use of oversampling and sigma-delta modulation in the design of high precision A/D converters.

6.1 Ideal Sampling and Reconstruction of Continuous-Time Signals

To process a continuous-time signal using digital signal processing techniques, it is necessary to convert the signal into a sequence of numbers. As was discussed in Section 1.4, this is usually done by sampling the analog signal, say $x_a(t)$, periodically every T seconds to produce a discrete-time signal $x(n)$ given by

$$x(n) = x_a(nT), \qquad -\infty < n < \infty \tag{6.1.1}$$

The relationship (6.1.1) describes the sampling process in the time domain. As discussed in Chapter 1, the sampling frequency $F_s = 1/T$ must be selected large

enough such that the sampling does not cause any loss of spectral information (no aliasing). Indeed, if the spectrum of the analog signal can be recovered from the spectrum of the discrete-time signal, there is no loss of information. Consequently, we investigate the sampling process by finding the relationship between the spectra of signals $x_a(t)$ and $x(n)$.

If $x_a(t)$ is an aperiodic signal with finite energy, its (voltage) spectrum is given by the Fourier transform relation

$$X_a(F) = \int_{-\infty}^{\infty} x_a(t)e^{-j2\pi Ft}\,dt \tag{6.1.2}$$

whereas the signal $x_a(t)$ can be recovered from its spectrum by the inverse Fourier transform

$$x_a(t) = \int_{-\infty}^{\infty} X_a(F)e^{j2\pi Ft}\,dF \tag{6.1.3}$$

Note that utilization of all frequency components in the infinite frequency range $-\infty < F < \infty$ is necessary to recover the signal $x_a(t)$ if the signal $x_a(t)$ is not bandlimited.

The spectrum of a discrete-time signal $x(n)$, obtained by sampling $x_a(t)$, is given by the Fourier transform relation

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{6.1.4}$$

or, equivalently,

$$X(f) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fn} \tag{6.1.5}$$

The sequence $x(n)$ can be recovered from its spectrum $X(\omega)$ or $X(f)$ by the inverse transform

$$\begin{aligned} x(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)e^{j\omega n}d\omega \\ &= \int_{-1/2}^{1/2} X(f)e^{j2\pi fn}\,df \end{aligned} \tag{6.1.6}$$

In order to determine the relationship between the spectra of the discrete-time signal and the analog signal, we note that periodic sampling imposes a relationship between the independent variables t and n in the signals $x_a(t)$ and $x(n)$, respectively. That is,

$$t = nT = \frac{n}{F_s} \tag{6.1.7}$$

This relationship in the time domain implies a corresponding relationship between the frequency variables F and f in $X_a(F)$ and $X(f)$, respectively.

Indeed, substitution of (6.1.7) into (6.1.3) yields

$$x(n) \equiv x_a(nT) = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi nF/F_s}\, dF \tag{6.1.8}$$

If we compare (6.1.6) with (6.1.8), we conclude that

$$\int_{-1/2}^{1/2} X(f) e^{j2\pi fn}\, df = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi nF/F_s}\, dF \tag{6.1.9}$$

From the development in Chapter 1 we know that periodic sampling imposes a relationship between the frequency variables F and f of the corresponding analog and discrete-time signals, respectively. That is,

$$f = \frac{F}{F_s} \tag{6.1.10}$$

With the aid of (6.1.10), we can make a simple change in variable in (6.1.9), and obtain the result

$$\frac{1}{F_s} \int_{-F_s/2}^{F_s/2} X(F) e^{j2\pi nF/F_s}\, dF = \int_{-\infty}^{\infty} X_a(F) e^{j2\pi nF/F_s}\, dF \tag{6.1.11}$$

We now turn our attention to the integral on the right-hand side of (6.1.11). The integration range of this integral can be divided into an infinite number of intervals of width F_s. Thus the integral over the infinite range can be expressed as a sum of integrals, that is,

$$\int_{-\infty}^{\infty} X_a(F) e^{j2\pi nF/F_s}\, dF = \sum_{k=-\infty}^{\infty} \int_{(k-1/2)F_s}^{(k+1/2)F_s} X_a(F) e^{j2\pi nF/F_s}\, dF \tag{6.1.12}$$

We observe that $X_a(F)$ in the frequency interval $(k - \frac{1}{2})F_s$ to $(k + \frac{1}{2})F_s$ is identical to $X_a(F - kF_s)$ in the interval $-F_s/2$ to $F_s/2$. Consequently,

$$\begin{aligned} \sum_{k=-\infty}^{\infty} \int_{(k-1/2)F_s}^{(k+1/2)F_s} X_a(F) e^{j2\pi nF/F_s}\, dF &= \sum_{k=-\infty}^{\infty} \int_{-F_s/2}^{F_s/2} X_a(F - kF_s) e^{j2\pi nF/F_s}\, dF \\ &= \int_{-F_s/2}^{F_s/2} \left[\sum_{k=-\infty}^{\infty} X_a(F - kF_s) \right] e^{j2\pi nF/F_s}\, dF \end{aligned} \tag{6.1.13}$$

where we have used the periodicity of the complex exponential, namely,

$$e^{j2\pi n(F+kF_s)/F_s} = e^{j2\pi nF/F_s}$$

Comparing (6.1.13), (6.1.12), and (6.1.11), we conclude that

$$X(F) = F_s \sum_{k=-\infty}^{\infty} X_a(F - kF_s) \tag{6.1.14}$$

or, equivalently,

$$X(f) = F_s \sum_{k=-\infty}^{\infty} X_a[(f - k)F_s] \tag{6.1.15}$$

This is the desired relationship between the spectrum $X(F)$ or $X(f)$ of the discrete-time signal and the spectrum $X_a(F)$ of the analog signal. The right-hand side of (6.1.14) or (6.1.15) consists of a periodic repetition of the scaled spectrum $F_s X_a(F)$ with period F_s. This periodicity is necessary because the spectrum $X(f)$ of the discrete-time signal is periodic with period $f_p = 1$ or $F_p = F_s$.

For example, suppose that the spectrum of a band-limited analog signal is as shown in Fig. 6.1.1(a). The spectrum is zero for $|F| \geq B$. Now, if the sampling frequency F_s is selected to be greater than $2B$, the spectrum $X(F)$ of the discrete-time signal will appear as shown in Fig. 6.1.1(b). Thus, if the sampling frequency F_s is selected such that $F_s \geq 2B$, where $2B$ is the Nyquist rate, then

$$X(F) = F_s X_a(F), \qquad |F| \leq F_s/2 \tag{6.1.16}$$

In this case there is no aliasing and therefore the spectrum of the discrete-time signal is identical (within the scale factor F_s) to the spectrum of the analog signal, within the fundamental frequency range $|F| \leq F_s/2$ or $|f| \leq \frac{1}{2}$.

On the other hand, if the sampling frequency F_s is selected such that $F_s < 2B$, the periodic continuation of $X_a(F)$ results in spectral overlap, as illustrated in Fig. 6.1.1(c) and (d). Thus the spectrum $X(F)$ of the discrete-time signal contains aliased frequency components of the analog signal spectrum $X_a(F)$. The end result is that the aliasing which occurs prevents us from recovering the original signal $x_a(t)$ from the samples.

Given the discrete-time signal $x(n)$ with the spectrum $X(F)$, as illustrated in Fig. 6.1.1(b), with no aliasing, it is now possible to reconstruct the original analog signal from the samples $x(n)$. Since in the absence of aliasing

$$X_a(F) = \begin{cases} \dfrac{1}{F_s} X(F), & |F| \leq F_s/2 \\ 0, & |F| > F_s/2 \end{cases} \tag{6.1.17}$$

and by the Fourier transform relationship (6.1.5),

$$X(F) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi Fn/F_s} \tag{6.1.18}$$

the inverse Fourier transform of $X_a(F)$ is

$$x_a(t) = \int_{-F_s/2}^{F_s/2} X_a(F) e^{j2\pi Ft}\, dF \tag{6.1.19}$$

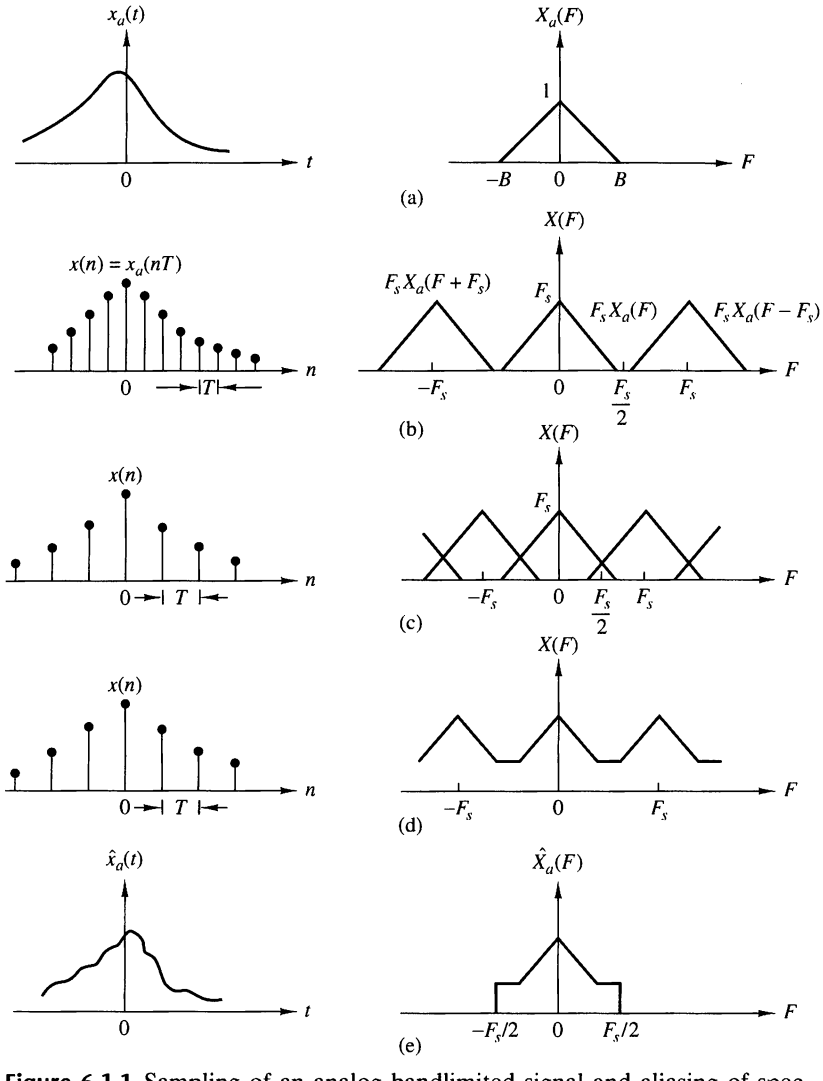


Figure 6.1.1 Sampling of an analog bandlimited signal and aliasing of spectral components.

Let us assume that $F_s \geq 2B$. With the substitution of (6.1.17) into (6.1.19), we have

$$\begin{aligned} x_a(t) &= \frac{1}{F_s} \int_{-F_s/2}^{F_s/2} \left[\sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi Fn/F_s} \right] e^{j2\pi Ft}\, dF \\ &= \frac{1}{F_s} \sum_{n=-\infty}^{\infty} x(n) \int_{-F_s/2}^{F_s/2} e^{j2\pi F(t-n/F_s)}\, dF \\ &= \sum_{n=-\infty}^{\infty} x_a(nT) \frac{\sin(\pi/T)(t-nT)}{(\pi/T)(t-nT)} \end{aligned} \tag{6.1.20}$$

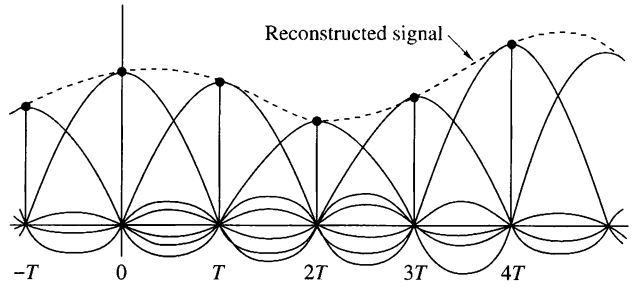


Figure 6.1.2
Reconstruction of a continuous-time signal using ideal interpolation.

where $x(n) = x_a(nT)$ and where $T = 1/F_s$ is the sampling interval. This is the reconstruction formula given by (1.4.24) in our discussion of the sampling theorem.

The reconstruction formula in (6.1.20) involves the function

$$g(t) = \frac{\sin(\pi/T)t}{(\pi/T)t} \tag{6.1.21}$$

appropriately shifted by nT, $n = 0, \pm 1, \pm 2, \ldots$, and multiplied or weighted by the corresponding samples $x_a(nT)$ of the signal. We call (6.1.20) an interpolation formula for reconstructing $x_a(t)$ from its samples, and $g(t)$, given in (6.1.21), is the interpolation function. We note that at $t = kT$, the interpolation function $g(t - nT)$ is zero except at $k = n$. Consequently, $x_a(t)$ evaluated at $t = kT$ is simply the sample $x_a(kT)$. At all other times the weighted sums of the time-shifted versions of the interpolation function combine to yield exactly $x_a(t)$. This combination is illustrated in Fig. 6.1.2.

The formula in (6.1.20) for reconstructing the analog signal $x_a(t)$ from its samples is called the *ideal interpolation formula.* It forms the basis for the *sampling theorem,* which can be stated as follows.

Sampling Theorem. A bandlimited continuous-time signal, with highest frequency (bandwidth) B hertz, can be uniquely recovered from its samples provided that the sampling rate $F_s \geq 2B$ samples per second.

According to the sampling theorem and the reconstruction formula in (6.1.20), the recovery of $x_a(t)$ from its samples $x(n)$ requires an infinite number of samples. However, in practice we use a finite number of samples of the signal and deal with finite-duration signals. As a consequence, we are concerned only with reconstructing a finite-duration signal from a finite number of samples.

When aliasing occurs due to too low a sampling rate, the effect can be described by a multiple folding of the frequency axis of the frequency variable F for the analog signal. Figure 6.1.3(a) shows the spectrum $X_a(F)$ of an analog signal. According to (6.1.14), sampling of the signal with a sampling frequency F_s results in a periodic repetition of $X_a(F)$ with period F_s. If $F_s < 2B$, the shifted replicas of $X_a(F)$ overlap. The overlap that occurs within the fundamental frequency range $-F_s/2 \leq F \leq F_s/2$ is illustrated in Fig. 6.1.3(b). The corresponding spectrum of the discrete-time signal within the fundamental frequency range is obtained by adding all the shifted portions within the range $|f| \leq \frac{1}{2}$, to yield the spectrum shown in Fig. 6.1.3(c).

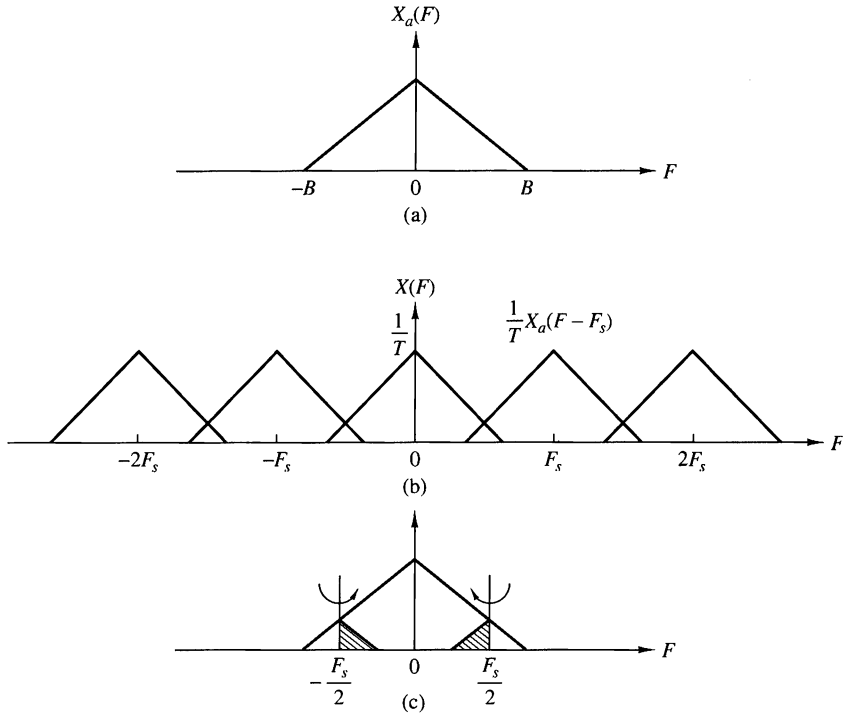


Figure 6.1.3 Illustration of aliasing around the folding frequency.

A careful inspection of Fig. 6.1.3(a) and (b) reveals that the aliased spectrum in Fig. 6.1.3(c) can be obtained by folding the original spectrum like an accordion with pleats at every odd multiple of $F_s/2$. Consequently, the frequency $F_s/2$ is called the *folding frequency*, as indicated in Chapter 1. Clearly, then, periodic sampling automatically forces a folding of the frequency axis of an analog signal at odd multiples of $F_s/2$, and this results in the relationship $F = fF_s$ between the frequencies for continuous-time signals and discrete-time signals. Due to the folding of the frequency axis, the relationship $F = fF_s$ is not truly linear, but piecewise linear, to accommodate for the aliasing effect. This relationship is illustrated in Fig. 6.1.4.

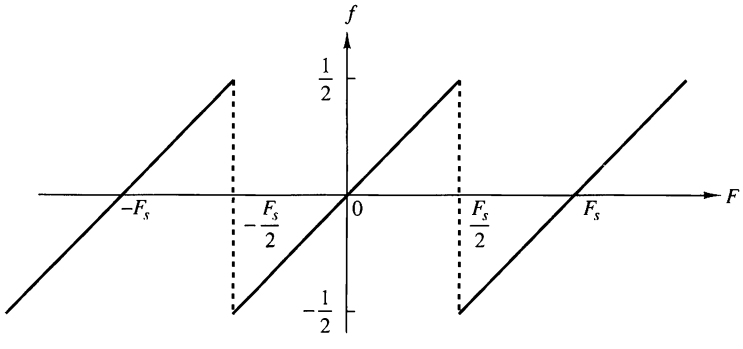


Figure 6.1.4 Relationship between frequency variables F and f.

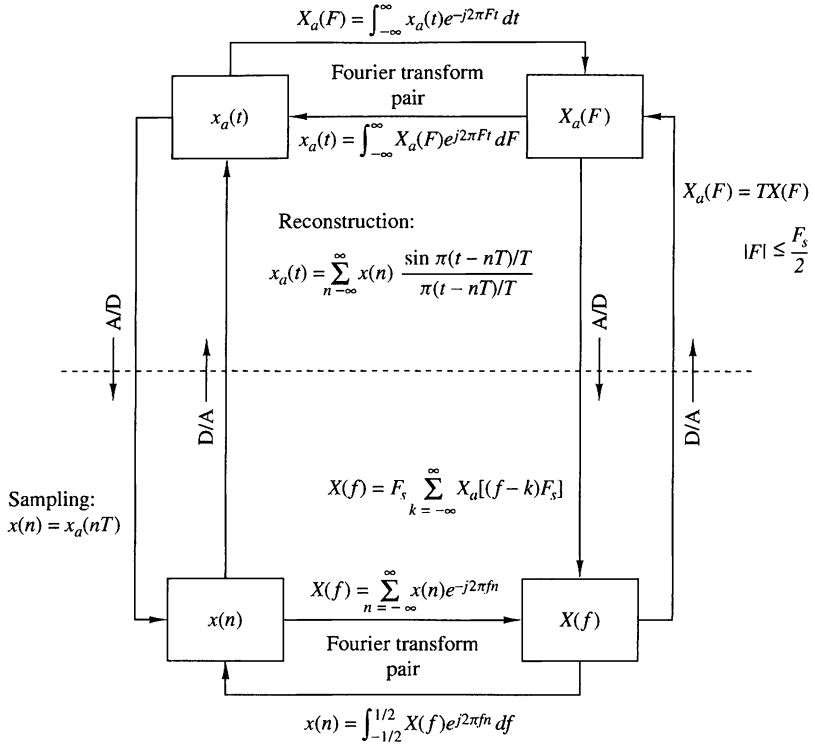


Figure 6.1.5 Time-domain and frequency-domain relationships for sampled signals.

If the analog signal is bandlimited to $B \le F_s/2$, the relationship between f and F is linear and one-to-one. In other words, there is no aliasing. In practice, prefiltering with an antialiasing filter is usually employed prior to sampling. This ensures that frequency components of the signal above $F \ge B$ are sufficiently attenuated so that, if aliased, they cause negligible distortion on the desired signal.

The relationships among the time-domain and frequency-domain functions $x_a(t)$, $x(n)$, $X_a(F)$, and $X(f)$ are summarized in Fig. 6.1.5. The relationships for recovering the continuous-time functions, $x_a(t)$ and $X_a(F)$, from the discrete-time quantities $x(n)$ and $X(f)$, assume that the analog signal is bandlimited and that it is sampled at the Nyquist rate (or faster).

The following examples serve to illustrate the problem of the aliasing of frequency components.

EXAMPLE 6.1.1 Aliasing in Sinusoidal Signals

The continuous-time signal

$$x_a(t) = \cos 2\pi F_0 t = \frac{1}{2}e^{j2\pi F_0 t} + \frac{1}{2}e^{-j2\pi F_0 t}$$

has a discrete spectrum with spectral lines at $F = \pm F_0$, as shown in Fig. 6.1.6(a). The process

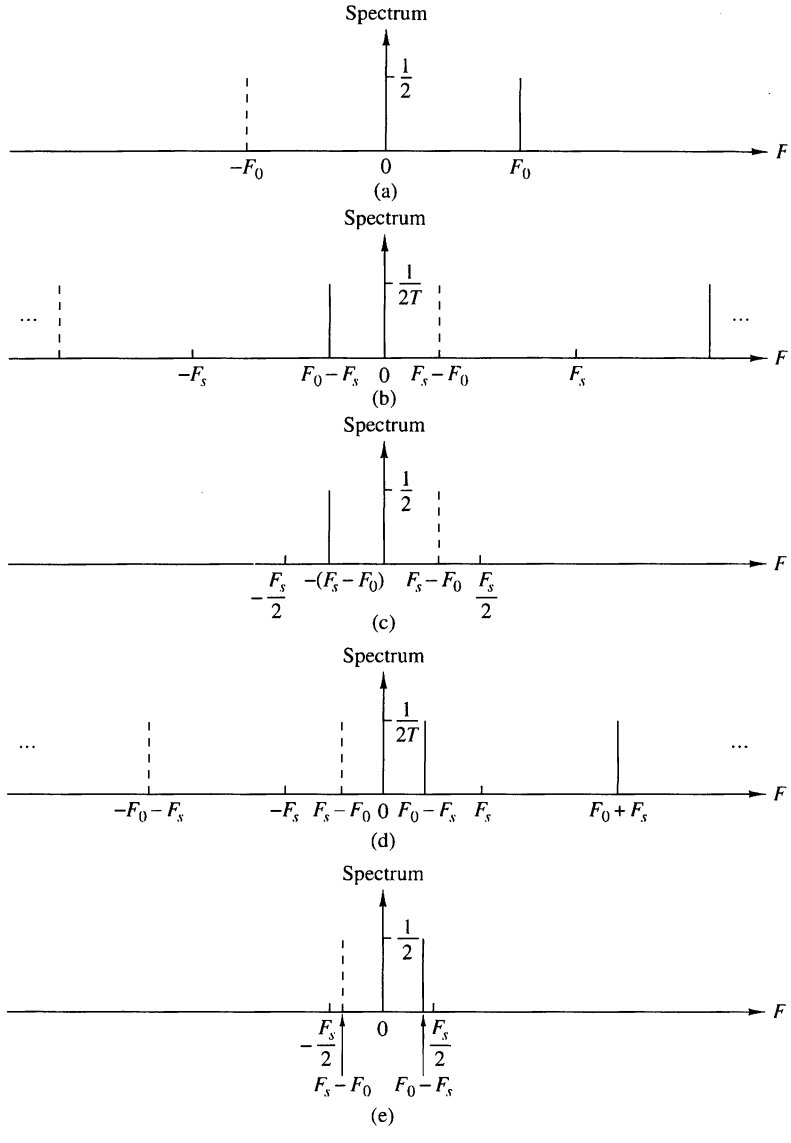


Figure 6.1.6 Aliasing of sinusoidal signals.

of sampling this signal with a sampling frequency F_s introduces replicas of the spectrum about multiples of F_s. This is illustrated in Fig. 6.1.6(b) for $F_s/2 < F_0 < F_s$.

To reconstruct the continuous-time signal, we should select the frequency components inside the fundamental frequency range $|F| \leq F_s/2$. The resulting spectrum is shown in Fig. 6.1.6(c). The reconstructed signal is

$$\hat{x}_a(t) = \cos 2\pi(F_s - F_0)t$$

Now, if F_s is selected such that $F_s < F_0 < 3F_s/2$, the spectrum of the sampled signal is shown in Fig. 6.1.6(d). The reconstructed signal, shown in Fig. 6.1.6(e), is

$$x_a(t) = \cos 2\pi(F_0 - F_s)t$$

In both cases, aliasing has occurred, so that the frequency of the reconstructed signal is an aliased version of the frequency of the original signal.

EXAMPLE 6.1.2 Sampling and Reconstruction of a Nonbandlimited Signal

Consider the following continuous-time two-sided exponential signal:

$$x_a(t) = e^{-A|t|} \overset{\mathcal{F}}{\longleftrightarrow} X_a(F) = \frac{2A}{A^2 + (2\pi F)^2}, \quad A > 0$$

(a) Determine the spectrum of the sampled signal $x(n) = x_a(nT)$. **(b)** Plot the signals $x_a(t)$ and $x(n) = x_a(nT)$, for $T = 1/3$ sec and $T = 1$ sec, and their spectra. **(c)** Plot the continuous-time signal $\hat{x}_a(t)$ after reconstruction with ideal bandlimited interpolation.

Solution.

(a) If we sample $x_a(nT)$ with a sampling frequency $F_s = 1/T$, we have

$$x(n) = x_a(nT) = e^{-AT|n|} = (e^{-AT})^{|n|}, \quad -\infty < n < \infty$$

The spectrum of $x(n)$ can be found easily if we use a direct computation of the discrete-time Fourier transform. We find that

$$X(F) = \frac{1 - a^2}{1 - 2a\cos 2\pi(F/F_s) + a^2}, \quad a = e^{-AT}$$

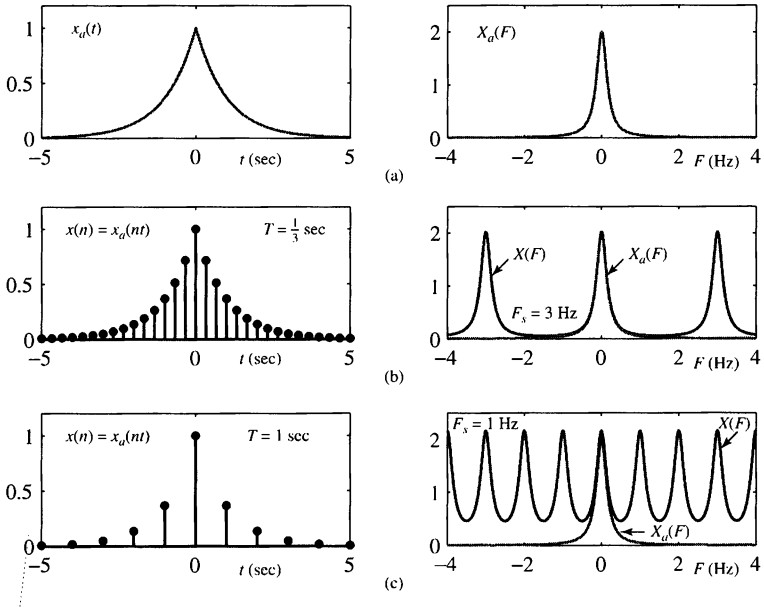


Figure 6.1.7 (a) Analog signal $x_a(t)$ and its spectrum $X_a(F)$; (b) $x(n) = x_a(nT)$ and its spectrum for $F_s = 3$ Hz; and (c) $x(n) = x_a(nT)$ and its spectrum for $F_s = 1$ Hz.

Clearly, since $\cos 2\pi(F/F_s)$ is periodic with period F_s, so is the spectrum $X(F)$.

(b) Since $X_a(F)$ is not bandlimited, aliasing cannot be avoided. Figure 6.1.7(a) shows the original signal $x_a(t)$ and its spectrum $X_a(F)$ for $A = 1$. The sampled signal $x(n)$ and its spectrum $X(F)$ are shown for $F_s = 3$ Hz and $F_s = 1$ Hz in Figures 6.1.7(b) and 6.1.7(c). The aliasing distortion is clearly noticeable in the frequency domain when $F_s = 1$ Hz and almost unnoticeable when $F_s = 3$ Hz.

(c) The spectrum $\hat{X}_a(F)$ of the reconstructed signal $\hat{x}_a(t)$ is given by

$$\hat{X}_a(F) = \begin{cases} TX(F), & |F| \le F_s/2 \\ 0, & \text{otherwise} \end{cases}$$

The values of $\hat{x}_a(t)$ can be evaluated for plotting purposes using the ideal bandlimited interpolation formula (6.1.20) for all significant values of $x(n)$ and $\sin(\pi t/T)/(\pi t/T)$. Figure 6.1.8 illustrates the reconstructed signal and its spectrum for $F_s = 3$ Hz and $F_s = 1$ Hz. It is interesting to note that in every case $\hat{x}_a(nT) = x_a(nt)$, but $\hat{x}_a(nt) \neq x_a(nT)$ for $t \neq nT$. The results of aliasing are clearly evident in the spectrum of $\hat{x}_a(t)$ for $F_s = 1$ Hz, where we note how the folding of the spectrum about $F = \pm 0.5$ Hz increases the high frequency content of $\hat{X}_a(F)$.

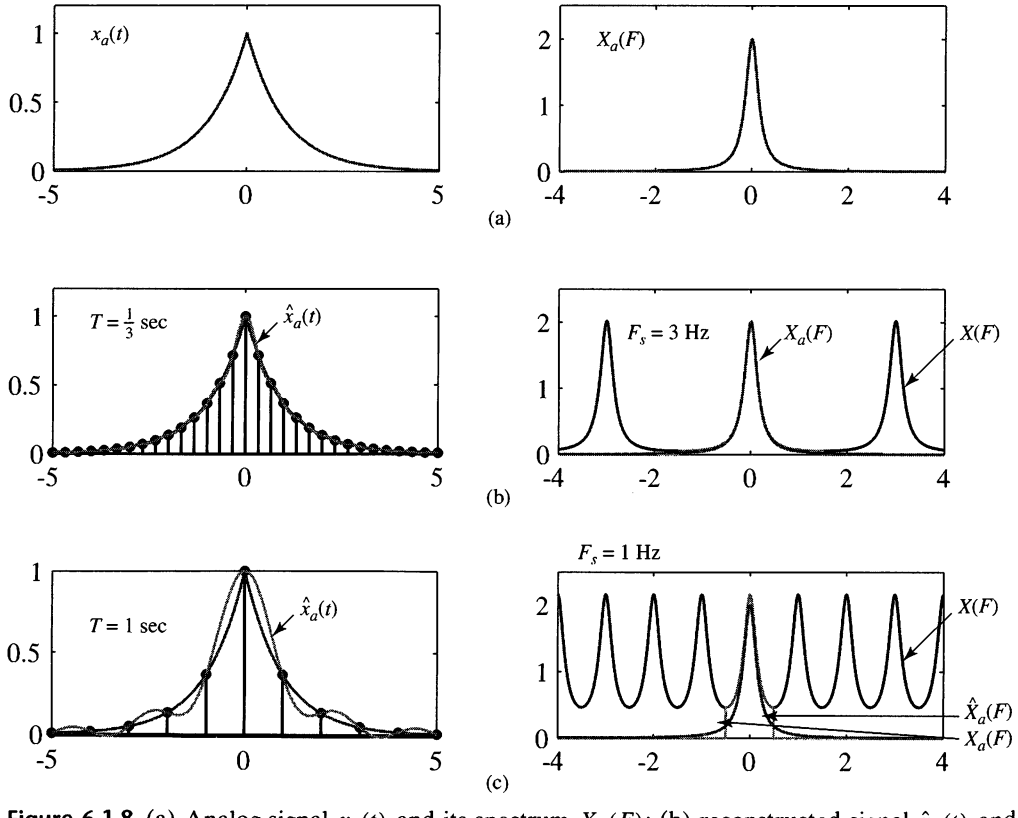


Figure 6.1.8 (a) Analog signal $x_a(t)$ and its spectrum $X_a(F)$; (b) reconstructed signal $\hat{x}_a(t)$ and its spectrum for $F_s = 3$ Hz; and (c) reconstructed signal $\hat{x}_a(t)$ and its spectrum for $F_s = 1$ Hz

Discrete-Time Processing of Continuous-Time Signals

Many practical applications require the discrete-time processing of continuous-time signals. Figure 6.2.1 shows the configuration of a general system used to achieve this objective. In designing the processing to be performed, we must first select the bandwidth of the signal to be processed, since the signal bandwidth determines the minimum sampling rate. For example, a speech signal, which is to be transmitted digitally, can contain frequency components above 3000 Hz, but for purposes of speech intelligibility and speaker identification, the preservation of frequency components bellow 3000 Hz is sufficient. Consequently, it would be inefficient from a processing viewpoint to preserve the higher-frequency components and wasteful of channel bandwidth to transmit the extra bits needed to represent these higher-frequency components in the speech signal. Once the desired frequency band is selected we can specify the sampling rate and the characteristics of the prefilter, which is also called an antialiasing filter.

The prefilter is an analog filter which has a twofold purpose. First, it ensures that the bandwidth of the signal to be sampled is limited to the desired frequency range. Thus any frequency components of the signal above $F_s/2$ are sufficiently attenuated so that the amount of signal distortion due to aliasing is negligible. For example, the speech signal to be transmitted digitally over a telephone channel would be filtered by a lowpass filter having a passband extending to 3000 Hz, a transition band of approximately 400 to 500 Hz, and a stopband above 3400 to 3500 Hz. The speech signal may be sampled at 8000 Hz and hence the folding frequency would be 4000 Hz. Thus aliasing would be negligible. Another reason for using a prefilter is to limit the additive noise spectrum and other interference, which often corrupts the desired signal. Usually, additive noise is wideband and exceeds the bandwidth of the desired signal. By prefiltering we reduce the additive noise power to that which falls within the bandwidth of the desired signal and we reject the out-of-band noise.

Once we have specified the prefilter requirements and have selected the desired sampling rate, we can proceed with the design of the digital signal processing operations to be performed on the discrete-time signal. The selection of the sampling rate $F_s = 1/T$, where T is the sampling interval, not only determines the highest frequency $(F_s/2)$ that is preserved in the analog signal, but also serves as a scale factor that influences the design specifications for digital filters and any other discrete-time systems through which the signal is processed.

The ideal A/D converter and the ideal D/A converter provide the interface between the continuous-time and discrete-time domains. The overall system is equivalent to a continuous-time system, which may or may not be linear and time-invariant

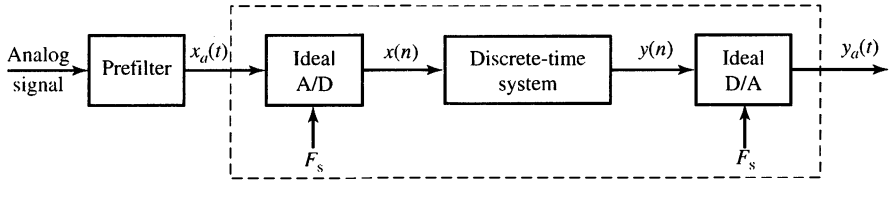


Figure 6.2.1 System for the discrete-time processing of continuous-time signals.

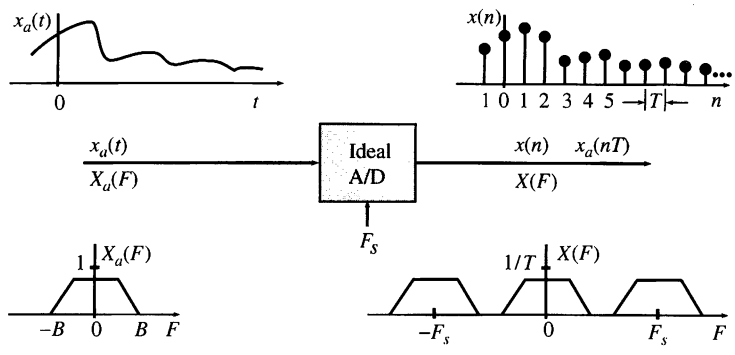


Figure 6.2.2 Characteristics of an ideal A/D converter in the time and frequency domains.

(even if the discrete-time system is linear and time invariant) because ideal A/D and D/A converters are time-varying operations.

Figure 6.2.2 summarizes the input-output characteristics of an ideal A/D converter in the time and frequency domains. We recall that if $x_a(t)$ is the input signal and $x(n)$ is the output signal, we have

$$x(n) = x_a(t)|_{t=nT} = x_a(nT) \qquad \text{(Time domain)} \tag{6.2.1}$$

$$X(F) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a(F - kF_s) \qquad \text{(Frequency domain)} \tag{6.2.2}$$

Basically, the ideal A/D converter is a linear time-varying system that (a) scales the analog spectrum by a factor $F_s = 1/T$ and (b) creates a periodic repetition of the scaled spectrum with period F_s.

The input–output characteristics of the ideal D/A converter are illustrated in Figure 6.2.3. In the time domain the input and output signals are related by

$$y_a(t) = \sum_{n=-\infty}^{\infty} y(n)g_a(t - nT), \qquad \text{(Time domain)} \tag{6.2.3}$$

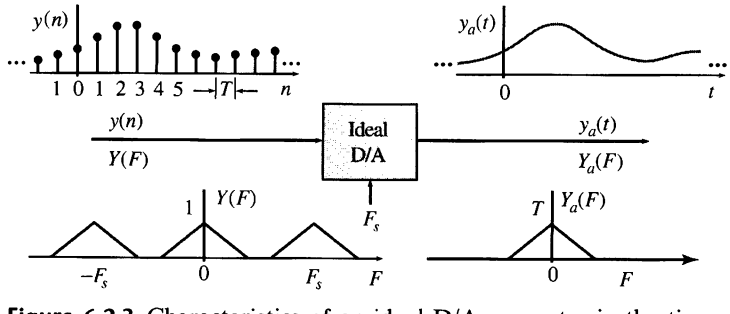


Figure 6.2.3 Characteristics of an ideal D/A converter in the time and frequency domains.

where

$$g_a(t) = \frac{\sin(\pi t/T)}{\pi t/T} \overset{\mathcal{F}}{\longleftrightarrow} G_a(F) = \begin{cases} T, & |F| \le F_s/2 \\ 0, & \text{otherwise} \end{cases} \tag{6.2.4}$$

is the interpolation function of the ideal D/A. We emphasize that (6.2.3) looks like, but it is *not*, a convolution because the D/A is a linear time-varying system with a discrete-time input and a continuous-time output. To obtain a frequency-domain description, we evaluate the Fourier transform of the output signal

$$\begin{aligned} Y_a(F) &= \sum_{n=-\infty}^{\infty} y(n) \int_{-\infty}^{\infty} g_a(t - nT) e^{-j2\pi Ft}\, dt \\ &= \sum_{n=-\infty}^{\infty} y(n) G_a(F) e^{-j2\pi FnT} \end{aligned}$$

After taking $G_a(F)$ outside the summation, we obtain

$$Y_a(F) = G_a(F)Y(F) \qquad \text{(Frequency domain)} \tag{6.2.5}$$

where $Y(F)$ is the discrete-time Fourier transform of $y(n)$. We note that the ideal D/A converter (a) scales the input spectrum by a factor $T = 1/F_s$ and (b) removes the frequency components for $|F| > F_s/2$. Basically, the ideal D/A acts as a frequency window that "removes" the discrete-time spectral periodicity to generate an aperiodic continuous-time signal spectrum. We stress once again that the ideal D/A is *not* a continuous-time ideal lowpass filter because (6.2.3) is not a continuous-time convolution integral.

Suppose now that we are given a continuous-time LTI system defined by

$$\tilde{y}_a(t) = \int_{-\infty}^{\infty} h_a(\tau) x_a(t - \tau)\, dt \tag{6.2.6}$$

$$\tilde{Y}_a(F) = H_a(F)X_a(F) \tag{6.2.7}$$

and we wish to determine whether there is a discrete-time system $H(F)$ such that the entire system in Figure 6.2.1 is equivalent to the continuous-time system $H_a(F)$. If $x_a(t)$ is not bandlimited or it is bandlimited but $F_s < 2B$, it is impossible to find such a system due to the presence of aliasing. However, if $x_a(t)$ is bandlimited and $F_s > 2B$, we have $X(F) = X_a(F)/T$ for $|F| \le F_s/2$. Therefore, the output of the system in Figure 6.2.1 is given by

$$Y_a(F) = H(F)X(F)G_a(F) = \begin{cases} H(F)X_a(F), & |F| \le F_s/2 \\ 0, & |F| > F_s/2 \end{cases} \tag{6.2.8}$$

To assure that $y_a(t) = \hat{y}_a(t)$, we should choose the discrete-time system so that

$$H(F) = \begin{cases} H_a(F), & |F| \le F_s/2 \\ 0, & |F| > F_s/2 \end{cases} \tag{6.2.9}$$

We note that, in this special case, the cascade connection of the A/D converter (linear time-varying system), an LTI system, and the D/A converter (linear time-varying system) is equivalent to a continuous-time LTI system. This important result provides the theoretical basis for the discrete-time filtering of continuous-time signals. These concepts are illustrated in the following examples.

EXAMPLE 6.2.1 Simulation of an analog integrator

Consider the analog integrator circuit shown in Figure 6.2.4(a). Its input–output relation is given by

$$RC\frac{dy_a(t)}{dt} + y_a(t) = x_a(t)$$

Taking the Fourier transform of both sides, we can show that the frequency response of the integrator is

$$H_a(F) = \frac{Y_a(F)}{X_a(F)} = \frac{1}{1 + jF/F_c}, \quad F_c = \frac{1}{2\pi RC}$$

Evaluating the inverse Fourier transform yields the impulse response

$$h_a(t) = Ae^{-At}u(t), \quad A = \frac{1}{RC}$$

Clearly the impulse response $h_a(t)$ is a nonbandlimited signal. We now define a discrete-time system by sampling the continuous-time impulse response as follows:

$$h(n) = h_a(nT) = A(e^{-AT})^n u(n)$$

We say that the discrete-time system is obtained from the continuous-time system through an *impulse-invariance* transformation (see Section 10.3.2). The system function and the difference equation of the discrete-time system are

$$H(z) = \sum_{n=0}^{\infty} A(e^{-AT})^n z^{-n} = \frac{1}{1 - e^{-AT}z^{-1}}$$

$$y(n) = e^{-AT}y(n-1) + Ax(n)$$

The system is causal and has a pole $p = e^{-AT}$. Since $A > 0$, $|p| < 1$ and the system is always stable. The frequency response of the system is obtained by evaluating $H(z)$ for $z = e^{j2\pi F/F_s}$. Figure 6.2.4(b) shows the magnitude frequency responses of the analog integrator and the discrete-time simulator for F_s =50, 100, 200, and 1000 Hz. We note that the effects of aliasing, caused by the sampling of $h_a(t)$, become negligible only for sampling frequencies larger than 1 kHz. The discrete-time implementation is accurate for input signals with bandwidth much less than the sampling frequency.

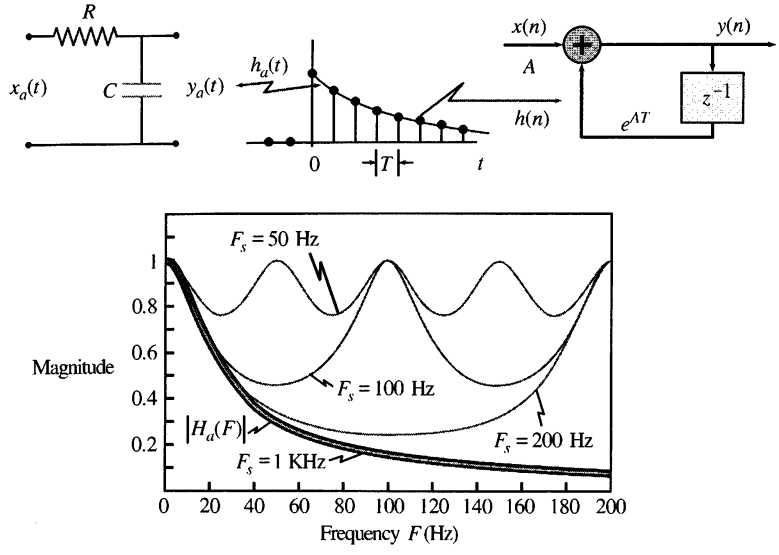


Figure 6.2.4 Discrete-time implementation of an analog integrator using impulse response sampling. The approximation is satisfactory when the bandwidth of the input signal is much less than the sampling frequency.

EXAMPLE 6.2.2 **Ideal bandlimited differentiator**

The ideal continuous-time differentiator is defined by

$$y_a(t) = \frac{dx_a(t)}{dt} \tag{6.2.10}$$

and has frequency response function

$$H_a(F) = \frac{Y_a(F)}{X_a(F)} = j2\pi F \tag{6.2.11}$$

For processing bandlimited signals, it is sufficient to use the ideal bandlimited differentiator defined by

$$H_a(F) = \begin{cases} j2\pi F, & |F| \le F_c \\ 0, & |F| > F_c \end{cases} \tag{6.2.12}$$

If we choose $F_s = 2F_c$, we can define an ideal discrete time differentiator by

$$H(F) = H_a(F) = j2\pi F, \quad |F| \le F_s/2 \tag{6.2.13}$$

Since by definition $H(F) = \sum_k H_a(F - kF_s)$, we have $h(n) = h_a(nT)$. In terms of $\omega = 2\pi F/F_s$, $H(\omega)$ is periodic with period 2π. Therefore, the discrete-time impulse response is given by

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega n}d\omega = \frac{\pi n\cos\pi n - \sin\pi n}{\pi n^2 T} \tag{6.2.14}$$

or in a more compact form

$$h(n) = \begin{cases} 0, & n = 0 \\ \frac{\cos\pi n}{nT}, & n \neq 0 \end{cases} \tag{6.2.15}$$

The magnitude and phase responses of the continuous-time and discrete-time ideal differentiators are shown in Figure 6.2.5.

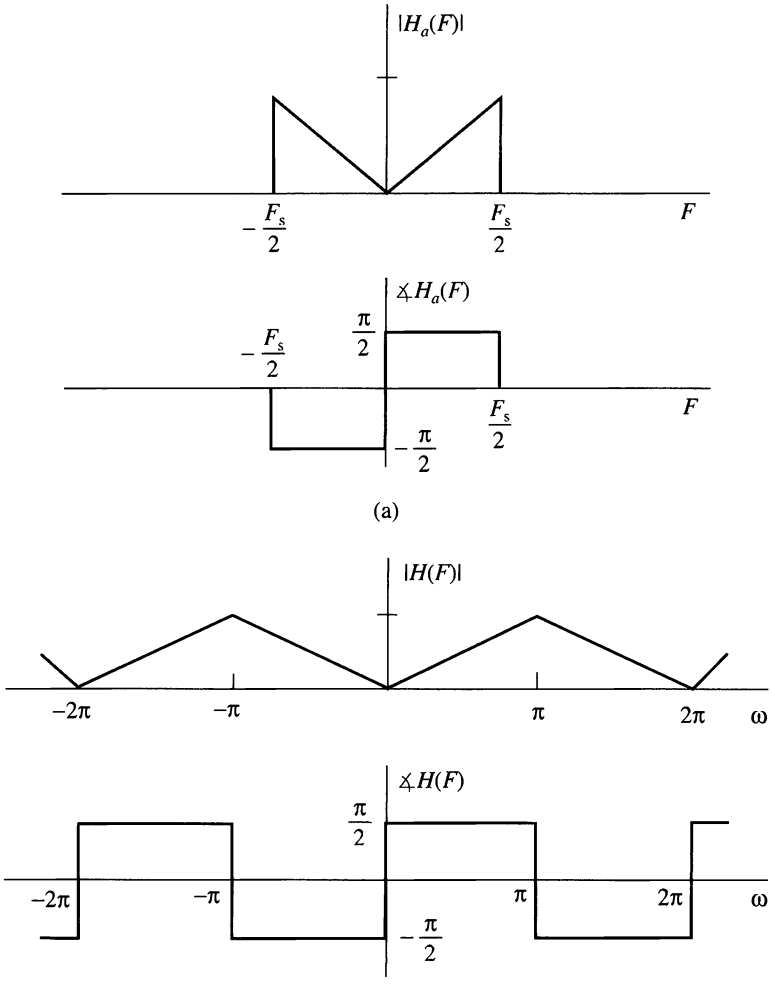


Figure 6.2.5 Frequency responses of the ideal bandlimited continuous-time differentiator (a) and its discrete-time counterpart (b).

EXAMPLE 6.2.3 Fractional delay

A continuous-time delay system is defined by

$$y_a(t) = x_a(t - t_d) \tag{6.2.16}$$

for any $t_d > 0$. Although the concept is simple, its practical implementation is quite complicated. If $x_a(t)$ is bandlimited and sampled at the Nyquist rate, we obtain

$$y(n) = y_a(nT) = x_a(nT - t_d) = x_a[(n - \Delta)T] = x(n - \Delta) \tag{6.2.17}$$

where $\Delta = t_d/T$. If Δ is an integer, delaying the sequence $x(n)$ is a simple process. For noninteger values of Δ, the delayed value of $x(n)$ would lie somewhere between two samples. However, this value is unavailable and the only way to generate an appropriate value is by ideal bandlimited interpolation. One way to approach this problem is by considering the frequency response

$$H_{\text{id}}(\omega) = e^{-j\omega\Delta} \tag{6.2.18}$$

of the delay system in (6.2.17) and its impulse response

$$h_{\text{id}}(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega n}d\omega = \frac{\sin\pi(n - \Delta)}{\pi(n - \Delta)} \tag{6.2.19}$$

When the delay Δ assumes integer values, $h_{\text{id}}(n)$ reduces to $\delta(n - \Delta)$ because the sin function is sampled at the zero crossings. When Δ is noninteger, $h_{\text{id}}(n)$ is infinitely long because the sampling times fall between the zero crossings. Unfortunately, the ideal impulse response for fractional delay systems is noncausal and has infinite duration. Therefore, the frequency response (6.2.18) has to be approximated with realizable FIR or IIR filters. More details about fractional delay filter design can be found in Laakso et al. (1996). Implementation of fractional delay using sampling rate conversion techniques is discussed in Section 11.8.

Analog-to-Digital and Digital-to-Analog Converters

In the previous section we assumed that the A/D and D/A converters in the processing of continuous-time signals are ideal. The one implicit assumption that we have made in the discussion on the equivalence of continuous-time and discrete-time signal processing is that the quantization error in analog-to-digital conversion and round-off errors in digital signal processing are negligible. These issues are further discussed in this section. However, we should emphasize that analog signal processing operations cannot be done very precisely either, since electronic components in analog systems have tolerances and they introduce noise during their operation. In general, a digital system designer has better control of tolerances in a digital signal processing system than an analog system designer who is designing an equivalent analog system.

The discussion in Section 6.1 focused on the conversion of continuous-time signals to discrete-time signals using an ideal sampler and ideal interpolation. In this section we deal with the devices for performing these conversions from analog to digital.

6.3.1 Analog-to-Digital Converters

Recall that the process of converting a continuous-time (analog) signal to a digital sequence that can be processed by a digital system requires that we quantize the sampled values to a finite number of levels and represent each level by a number of bits.

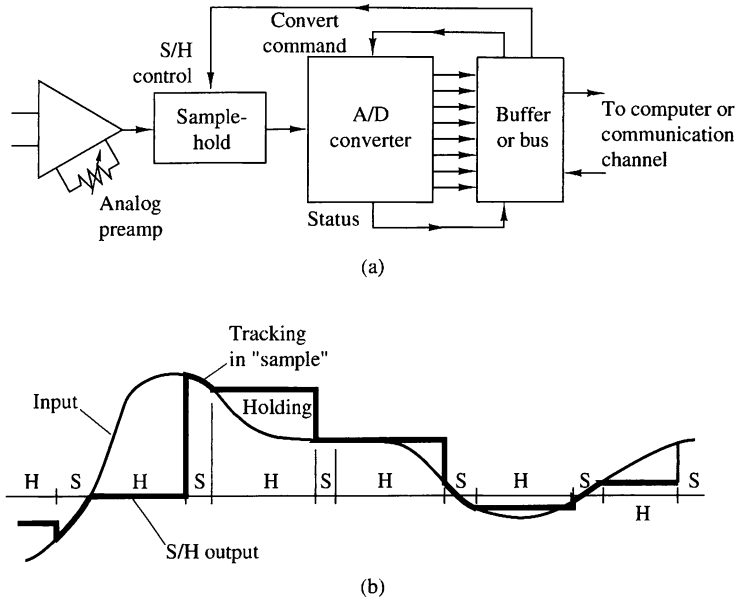


Figure 6.3.1 (a) Block diagram of basic elements of an A/D converter; (b) time-domain response of an ideal S/H circuit.

Figure 6.3.1(a) shows a block diagram of the basic elements of an A/D converter. In this section we consider the performance requirements for these elements. Although we focus mainly on ideal system characteristics, we shall also mention some key imperfections encountered in practical devices and indicate how they affect the performance of the converter. We concentrate on those aspects that are more relevant to signal processing applications. The practical aspects of A/D converters and related circuitry can be found in the manufacturers' specifications and data sheets.

In practice, the sampling of an analog signal is performed by a sample-and-hold (S/H) circuit. The sampled signal is then quantized and converted to digital form. Usually, the S/H is integrated into the A/D converter. The S/H is a digitally controlled analog circuit that tracks the analog input signal during the sample mode, and then holds it fixed during the hold mode to the instantaneous value of the signal at the time the system is switched from the sample mode to the hold mode. Figure 6.3.1(b) shows the time-domain response of an ideal S/H circuit (i.e., an S/H that responds instantaneously and accurately).

The goal of the S/H is to continuously sample the input signal and then to hold that value constant as long as it takes for the A/D converter to obtain its digital representation. The use of an S/H allows the A/D converter to operate more slowly compared to the time actually used to acquire the sample. In the absence of an S/H, the input signal must not change by more than one-half of the quantization step during the conversion, which may be an impractical constraint. Consequently, the S/H is crucial in high-resolution (12 bits per sample or higher) digital conversion of signals that have large bandwidths (i.e., they change very rapidly).

An ideal S/H introduces no distortion in the conversion process and is accurately modeled as an ideal sampler. However, time-related degradations such as errors in the periodicity of the sampling process ("jitter"), nonlinear variations in the duration of the sampling aperture, and changes in the voltage held during conversion ("droop") do occur in practical devices.

The A/D converter begins the conversion after it receives a convert command. The time required to complete the conversion should be less than the duration of the hold mode of the S/H. Furthermore, the sampling period T should be larger than the duration of the sample mode and the hold mode.

In the following sections we assume that the S/H introduces negligible errors and we focus on the digital conversion of the analog samples.

6.3.2 Quantization and Coding

The basic task of the A/D converter is to convert a continuous range of input amplitudes into a discrete set of digital code words. This conversion involves the processes of *quantization* and *coding*. Quantization is a nonlinear and noninvertible process that maps a given amplitude $x(n) \equiv x(nT)$ at time $t = nT$ into an amplitude x_k, taken from a finite set of values. The procedure is illustrated in Fig. 6.3.2(a), where the signal amplitude range is divided into L intervals

$$I_k = \{x_k < x(n) \leq x_{k+1}\}, \qquad k = 1, 2, \ldots, L \tag{6.3.1}$$

by the $L + 1$ *decision levels* $x_1, x_2, \ldots, x_{L+1}$. The possible outputs of the quantizer (i.e., the quantization levels) are denoted as $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_L$. The operation of the quantizer is defined by the relation

$$x_q(n) \equiv Q[x(n)] = \hat{x}_k, \qquad \text{if } x(n) \in I_k \tag{6.3.2}$$

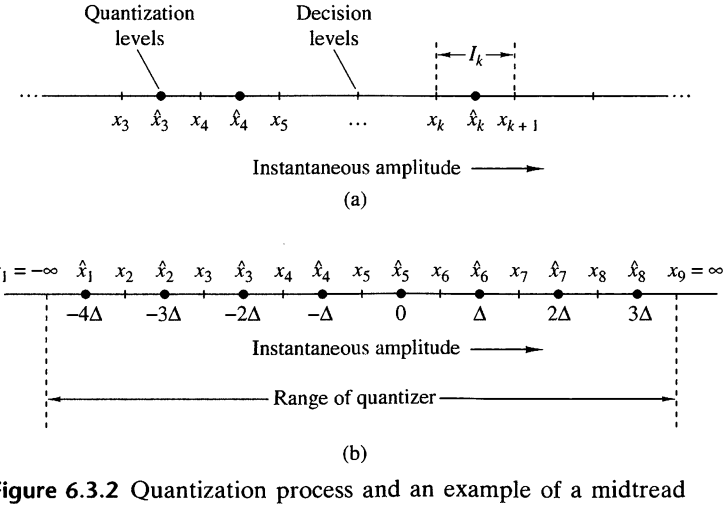


Figure 6.3.2 Quantization process and an example of a midtread quantizer.

In most digital signal processing operations the mapping in (6.3.2) is independent of n (i.e., the quantization is memoryless and is simply denoted as $x_q = Q[x]$). Furthermore, in signal processing we often use *uniform* or *linear quantizers* defined by

$$\begin{aligned} \hat{x}_{k+1} - \hat{x}_k &= \Delta, \qquad k = 1, 2, \ldots, L-1 \\ x_{k+1} - x_k &= \Delta, \qquad \text{for finite } x_k, x_{k+1} \end{aligned} \tag{6.3.3}$$

where Δ is the *quantizer step size*. Uniform quantization is usually a requirement if the resulting digital signal is to be processed by a digital system. However, in transmission and storage applications of signals such as speech, nonlinear and time-variant quantizers are frequently used.

If zero is assigned a quantization level, the quantizer is of the *midtread* type. If zero is assigned a decision level, the quantizer is called a *midrise* type. Figure 6.3.2(b) illustrates a midtread quantizer with $L = 8$ levels. In theory, the extreme decision levels are taken as $x_1 = -\infty$ and $x_{L+1} = \infty$, to cover the total *dynamic range* of the input signal. However, practical A/D converters can handle only a finite range. Hence we define the *range* R of the quantizer by assuming that $I_1 = I_L = \Delta$. For example, the range of the quantizer shown in Fig. 6.3.2(b) is equal to 8Δ. In practice, the term *full-scale range* (FSR) is used to describe the range of an A/D converter for bipolar signals (i.e., signals with both positive and negative amplitudes). The term *full scale* (FS) is used for unipolar signals.

It can be easily seen that the quantization error $e_q(n)$ is always in the range $-\Delta/2$ to $\Delta/2$:

$$-\frac{\Delta}{2} < e_q(n) \leq \frac{\Delta}{2} \tag{6.3.4}$$

In other words, the instantaneous quantization error cannot exceed half of the quantization step. If the dynamic range of the signal, defined as $x_{\max} - x_{\min}$, is larger than the range of the quantizer, the samples that exceed the quantizer range are clipped, resulting in a large (greater than $\Delta/2$) quantization error.

The operation of the quantizer is better described by the quantization characteristic function, illustrated in Fig. 6.3.3 for a midtread quantizer with eight quantization levels. This characteristic is preferred in practice over the midriser because it provides an output that is insensitive to infinitesimal changes of the input signal about zero. Note that the input amplitudes of a midtread quantizer are rounded to the nearest quantization levels.

The *coding* process in an A/D converter assigns a unique binary number to each quantization level. If we have L levels, we need at least L different binary numbers. With a word length of $b + 1$ bits we can represent 2^{b+1} distinct binary numbers. Hence we should have $2^{b+1} \geq L$ or, equivalently, $b + 1 \geq \log_2 L$. Then the step size or the *resolution* of the A/D converter is given by

$$\Delta = \frac{R}{2^{b+1}} \tag{6.3.5}$$

where R is the range of the quantizer.

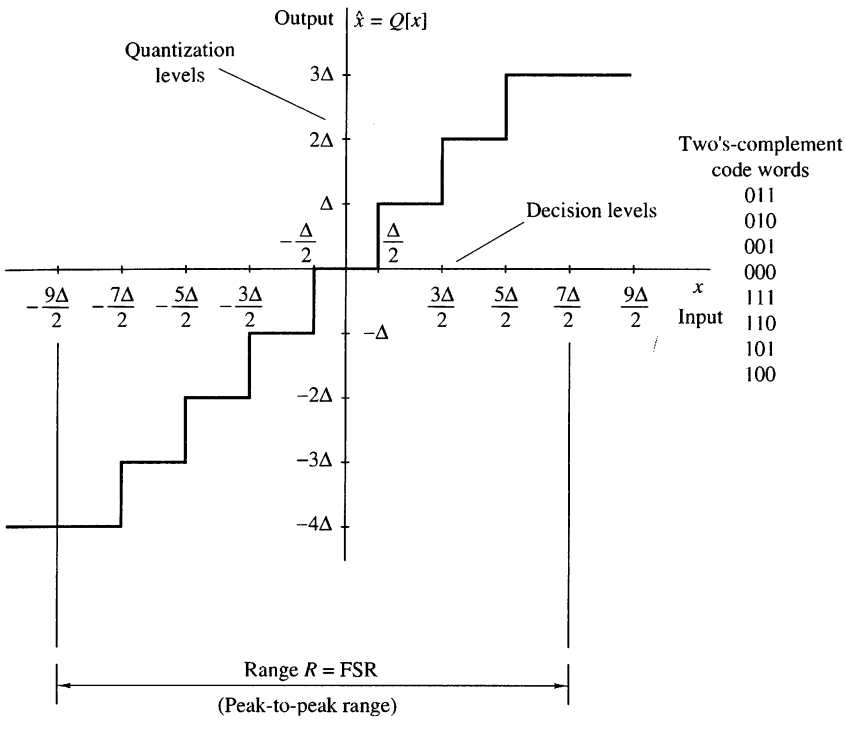


Figure 6.3.3 Example of a midtread quantizer.

There are various binary coding schemes, each with its advantages and disadvantages. Table 6.1 illustrates some existing schemes for 3-bit binary coding. These number representation schemes are described in more detail in Section 9.4.

The two's-complement representation is used in most digital signal processors. Thus it is convenient to use the same system to represent digital signals because we can operate on them directly without any extra format conversion. In general, a $(b+1)$-bit binary fraction of the form $\beta_0\beta_1\beta_2\cdots\beta_b$ has the value

$$-\beta_0 \cdot 2^0 + \beta_1 \cdot 2^{-1} + \beta_2 \cdot 2^{-2} + \cdots + \beta_b \cdot 2^{-b}$$

if we use the two's-complement representation. Note that β_0 is the most significant bit (MSB) and β_b is the least significant bit (LSB). Although the binary code used to represent the quantization levels is important for the design of the A/D converter and the subsequent numerical computations, it does not have any effect in the performance of the quantization process. Thus in our subsequent discussions we ignore the process of coding when we analyze the performance of A/D converters.

The only degradation introduced by an ideal converter is the quantization error, which can be reduced by increasing the number of bits. This error, which dominates the performance of practical A/D converters, is analyzed in the next section.

Practical A/D converters differ from ideal converters in several ways. Various degradations are usually encountered in practice. Specifically, practical A/D converters may have an *offset* error (the first transition may not occur at exactly $+\frac{1}{2}$LSB),

TABLE 6.1 Commonly Used Bipolar Codes

	Decimal Fraction					
Number	Positive Reference	Negative Reference	Sign + Magnitude	Two's Complement	Offset Binary	One's Complement
+7	$+\frac{7}{8}$	$-\frac{7}{8}$	0111	0111	1111	0111
+6	$+\frac{6}{8}$	$-\frac{6}{8}$	0110	0110	1110	0110
+5	$+\frac{5}{8}$	$-\frac{5}{8}$	0101	0101	1101	0101
+4	$+\frac{4}{8}$	$-\frac{4}{8}$	0100	0100	1100	0100
+3	$+\frac{3}{8}$	$-\frac{3}{8}$	0011	0011	1011	0011
+2	$+\frac{2}{8}$	$-\frac{2}{8}$	0010	0010	1010	0010
+1	$+\frac{1}{8}$	$-\frac{1}{8}$	0001	0001	1001	0001
0	0+	0−	0000	0000	1000	0000
0	0−	0+	1000	(0000)	(1000)	1111
−1	$-\frac{1}{8}$	$+\frac{1}{8}$	1001	1111	0111	1110
−2	$-\frac{2}{8}$	$+\frac{2}{8}$	1010	1110	0110	1101
−3	$-\frac{3}{8}$	$+\frac{3}{8}$	1011	1101	0101	1100
−4	$-\frac{4}{8}$	$+\frac{4}{8}$	1100	1100	0100	1011
−5	$-\frac{5}{8}$	$+\frac{5}{8}$	1101	1011	0011	1010
−6	$-\frac{6}{8}$	$+\frac{6}{8}$	1110	1010	0010	1001
−7	$-\frac{7}{8}$	$+\frac{7}{8}$	1111	1001	0001	1000
−8	$-\frac{8}{8}$	$+\frac{8}{8}$		(1000)	(0000)	

scale-factor (or gain) error (the difference between the values at which the first transition and the last transition occur is not equal to FS − 2LSB), and a *linearity* error (the differences between transition values are not all equal or uniformly changing). If the *differential linearity* error is large enough, it is possible for one or more code words to be missed. Performance data on commercially available A/D converters are specified in manufacturers' data sheets.

6.3.3 Analysis of Quantization Errors

To determine the effects of quantization on the performance of an A/D converter, we adopt a statistical approach. The dependence of the quantization error on the characteristics of the input signal and the nonlinear nature of the quantizer make a deterministic analysis intractable, except in very simple cases.

In the statistical approach, we assume that the quantization error is random in nature. We model this error as noise that is added to the original (unquantized) signal. If the input analog signal is within the range of the quantizer, the quantization error

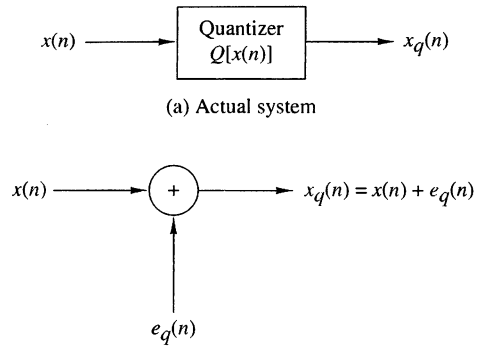


Figure 6.3.4
Mathematical model of quantization noise.

$e_q(n)$ is bounded in magnitude [i.e., $|e_q(n)| < \Delta/2$], and the resulting error is called *granular noise*. When the input falls outside the range of the quantizer (clipping), $e_q(n)$ becomes unbounded and results in *overload noise*. This type of noise can result in severe signal distortion when it occurs. Our only remedy is to scale the input signal so that its dynamic range falls within the range of the quantizer. The following analysis is based on the assumption that there is no overload noise.

The mathematical model for the quantization error $e_q(n)$ is shown in Fig. 6.3.4. To carry out the analysis, we make the following assumptions about the statistical properties of $e_q(n)$:

1. The error $e_q(n)$ is uniformly distributed over the range $-\Delta/2 < e_q(n) < \Delta/2$.
2. The error sequence $\{e_q(n)\}$ is a stationary white noise sequence. In other words, the error $e_q(n)$ and the error $e_q(m)$ for $m \neq n$ are uncorrelated.
3. The error sequence $\{e_q(n)\}$ is uncorrelated with the signal sequence $x(n)$.
4. The signal sequence $x(n)$ is zero mean and stationary.

These assumptions do not hold, in general. However, they do hold when the quantization step size is small and the signal sequence $x(n)$ traverses several quantization levels between two successive samples.

Under these assumptions, the effect of the additive noise $e_q(n)$ on the desired signal can be quantified by evaluating the signal-to-quantization-noise (power) ratio (SQNR), which can be expressed on a logarithmic scale (in decibels or dB) as

$$\text{SQNR} = 10 \log_{10} \frac{P_x}{P_n} \tag{6.3.6}$$

where $P_x = \sigma_x^2 = E[x^2(n)]$ is the signal power and $P_n = \sigma_e^2 = E[e_q^2(n)]$ is the power of the quantization noise.

If the quantization error is uniformly distributed in the range $(-\Delta/2, \Delta/2)$ as shown in Fig. 6.3.5, the mean value of the error is zero and the variance (the quantization noise power) is

$$P_n = \sigma_e^2 = \int_{-\Delta/2}^{\Delta/2} e^2 p(e)\, de = \frac{1}{\Delta} \int_{-\Delta/2}^{\Delta/2} e^2\, de = \frac{\Delta^2}{12} \tag{6.3.7}$$

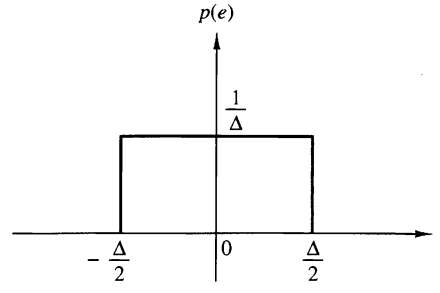


Figure 6.3.5
Probability density function for the quantization error.

By combining (6.3.5) with (6.3.7) and substituting the result into (6.3.6), the expression for the SQNR becomes

$$\begin{aligned} \text{SQNR} &= 10 \log \frac{P_x}{P_n} = 20 \log \frac{\sigma_x}{\sigma_e} \\ &= 6.02b + 16.81 - 20 \log \frac{R}{\sigma_x} \text{ dB} \end{aligned} \tag{6.3.8}$$

The last term in (6.3.8) depends on the range R of the A/D converter and the statistics of the input signal. For example, if we assume that $x(n)$ is Gaussian distributed and the range of the quantizer extends from $-3\sigma_x$ to $3\sigma_x$ (i.e., $R = 6\sigma_x$), then less than 3 out of every 1000 input signal amplitudes would result in an overload on the average. For $R = 6\sigma_x$, (6.3.8) becomes

$$\text{SQNR} = 6.02b + 1.25 \text{ dB} \tag{6.3.9}$$

The formula in (6.3.8) is frequently used to specify the precision needed in an A/D converter. It simply means that each additional bit in the quantizer increases the signal-to-quantization-noise ratio by 6 dB. (It is interesting to note that the same result was derived in Section 1.4 for a sinusoidal signal using a deterministic approach.) However, we should bear in mind the conditions under which this result has been derived.

Due to limitations in the fabrication of A/D converters, their performance falls short of the theoretical value given by (6.3.8). As a result, the effective number of bits may be somewhat less than the number of bits in the A/D converter. For instance, a 16-bit converter may have only an effective 14 bits of accuracy.

6.3.4 Digital-to-Analog Converters

In practice, D/A conversion is usually performed by combining a D/A converter with a sample-and-hold (S/H) followed by a lowpass (smoothing) filter, as shown in Fig. 6.3.6. The D/A converter accepts, at its input, electrical signals that correspond

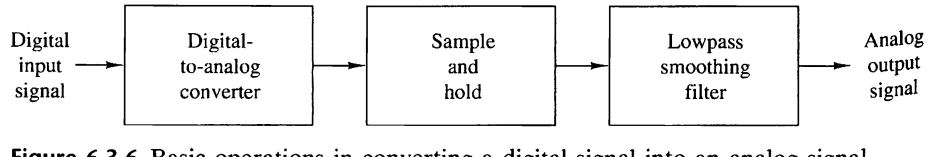


Figure 6.3.6 Basic operations in converting a digital signal into an analog signal.

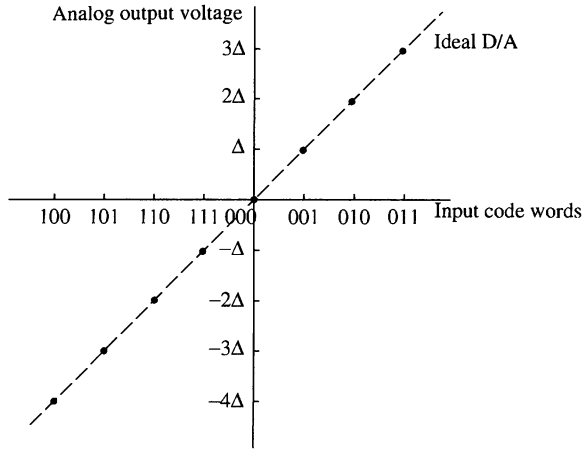


Figure 6.3.7
Ideal D/A converter characteristic.

to a binary word, and produces an output voltage or current that is proportional to the value of the binary word. Ideally, its input–output characteristic is as shown in Fig. 6.3.7 for a 3-bit bipolar signal. The line connecting the dots is a straight line through the origin. In practical D/A converters, the line connecting the dots may deviate from the ideal. Some of the typical deviations from ideal are offset errors, gain errors, and nonlinearities in the input–output characteristic.

An important parameter of a D/A converter is its *settling time*, which is defined as the time required for the output of the D/A converter to reach and remain within a given fraction (usually, $\pm\frac{1}{2}$LSB) of the final value, after application of the input code word. Often, the application of the input code word results in a high-amplitude transient, called a "glitch." This is especially the case when two consecutive code words to the A/D differ by several bits. The usual way to remedy this problem is to use an S/H circuit designed to serve as a "deglitcher." Hence the basic task of the S/H is to hold the output of the D/A converter constant at the previous output value until the new sample at the output of the D/A reaches steady state. Then it samples and holds the new value in the next sampling interval. Thus the S/H approximates the analog signal by a series of rectangular pulses whose height is equal to the corresponding value of the signal pulse. Figure 6.3.8 illustrates the response of an S/H to a discrete-time sinusoidal signal. As shown, the approximation, is basically a staircase function which takes the signal sample from the D/A converter and holds it for T seconds. When the next sample arrives, it jumps to the next value and holds it for T seconds, and so on.

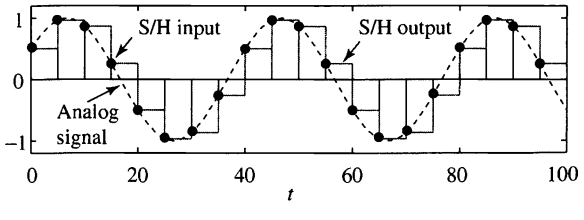


Figure 6.3.8
Response of an S/H interpolator to a discrete-time sinusoidal signal.

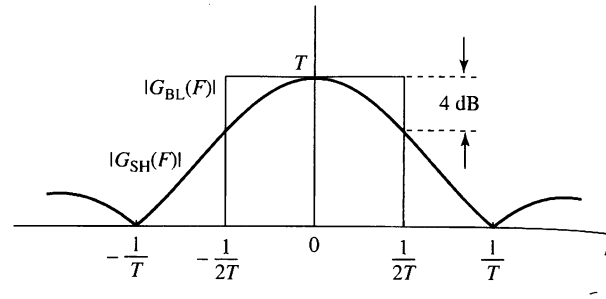


Figure 6.3.9
Frequency responses of sample-and-hold and the ideal bandlimited interpolator.

The interpolation function of the S/H system is a square pulse defined by

$$g_{SH}(t) = \begin{cases} 1, & 0 \le t \le T \\ 0, & \text{otherwise} \end{cases} \tag{6.3.10}$$

The frequency-domain characteristics are obtained by evaluating its Fourier transform

$$G_{SH}(F) = \int_{-\infty}^{\infty} g_{SH}(t) e^{-j2\pi Ft}\, dt = T \frac{\sin \pi FT}{\pi FT} e^{-2\pi F(T/2)} \tag{6.3.11}$$

The magnitude of $G_{SH}(F)$ is shown in Figure 6.3.9, where we superimpose the magnitude response of the ideal bandlimited interpolator for comparison purposes. It is apparent that the S/H does not possess a sharp cutoff frequency characteristic. This is due to a large extent to the sharp transitions of its interpolation function $g_{SH}(t)$. As a consequence, the S/H passes undesirable aliased frequency components (frequencies above $F_s/2$) to its output. This effect is sometimes referred to as *post-aliasing*. To remedy this problem, it is common practice to filter the output of the S/H by passing it through a lowpass filter, which highly attenuates frequency components above $F_s/2$. In effect, the lowpass filter following the S/H smooths its output by removing sharp discontinuities. Sometimes, the frequency response of the lowpass filter is defined by

$$H_a(F) = \begin{cases} \dfrac{\pi FT}{\sin \pi FT} e^{2\pi F(T/2)}, & |F| \le F_s/2 \\ 0, & |F| > F_s/2 \end{cases} \tag{6.3.12}$$

to compensate for the $\sin x/x$ distortion of the S/H (aperture effect). The aperture effect attenuation compensation, which reaches a maximum of $2/\pi$ or 4 dB at $F = F_s/2$, is usually neglected. However, it may be introduced using a digital filter before the sequence is applied to the D/A converter. The half-sample delay introduced by the S/H cannot be compensated because we cannot design analog filters that can introduce a time advance.

6.4 Sampling and Reconstruction of Continuous-Time Bandpass Signals

A continuous-time bandpass signal with bandwidth B and center frequency F_c has its frequency content in the two frequency bands defined by $0 < F_L < |F| < F_H$, where $F_c = (F_L + F_H)/2$ (see Figure 6.4.1(a)). A naive application of the sampling theorem

would suggest a sampling rate $F_s \geq 2F_H$; however, as we show in this section, there are sampling techniques that allow sampling rates consistent with the bandwidth B, rather than the highest frequency, F_H, of the signal spectrum. Sampling of bandpass signals is of great interest in the areas of digital communications, radar, and sonar systems.

6.4.1 Uniform or First-Order Sampling

Uniform or first-order sampling is the typical periodic sampling introduced in Section 6.1. Sampling the bandpass signal in Figure 6.4.1(a) at a rate $F_s = 1/T$ produces a sequence $x(n) = x_a(nT)$ with spectrum

$$X(F) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a(F - kF_s) \tag{6.4.1}$$

The positioning of the shifted replicas $X(F - kF_s)$ is controlled by a single parameter, the sampling frequency F_s. Since bandpass signals have two spectral bands, in general, it is more complicated to control their positioning, in order to avoid aliasing, with the single parameter F_s.

Integer Band Positioning. We initially restrict the higher frequency of the band to be an integer multiple of the bandwidth, that is, $F_H = mB$ (*integer band positioning*). The number $m = F_H/B$, which is in general fractional, is known as the *band position*. Figures 6.4.1(a) and 6.4.1(d) show two bandpass signals with even ($m = 4$) and odd ($m = 3$) band positioning. It can be easily seen from Figure 6.4.1(b) that, for integer-positioned bandpass signals, choosing $F_s = 2B$ results in a sequence with a spectrum without aliasing. From Figure 6.4.1(c), we see that the original bandpass signal can be reconstructed using the reconstruction formula

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a(nT) g_a(t - nT) \tag{6.4.2}$$

where

$$g_a(t) = \frac{\sin \pi Bt}{\pi Bt} \cos 2\pi F_c t \tag{6.4.3}$$

is the inverse Fourier transform of the bandpass frequency gating function shown in Figure 6.4.1(c). We note that $g_a(t)$ is equal to the ideal interpolation function for lowpass signals [see (6.1.21)], modulated by a carrier with frequency F_c.

It is worth noticing that, by properly choosing the center frequency F_c of $G_a(F)$, we can reconstruct a continuous-time bandpass signal with spectral bands centered at $F_c = \pm(kB + B/2)$, $k = 0, 1, \ldots$. For $k = 0$ we obtain the equivalent baseband signal, a process known as *down-conversion*. A simple inspection of Figure 6.4.1 demonstrates that the baseband spectrum for $m = 3$ has the same spectral structure as the original spectrum; however, the baseband spectrum for $m = 4$ has been "inverted." In general, when the band position is an *even* integer the baseband spectral images are inverted versions of the original ones. Distinguishing between these two cases is important in communications applications.

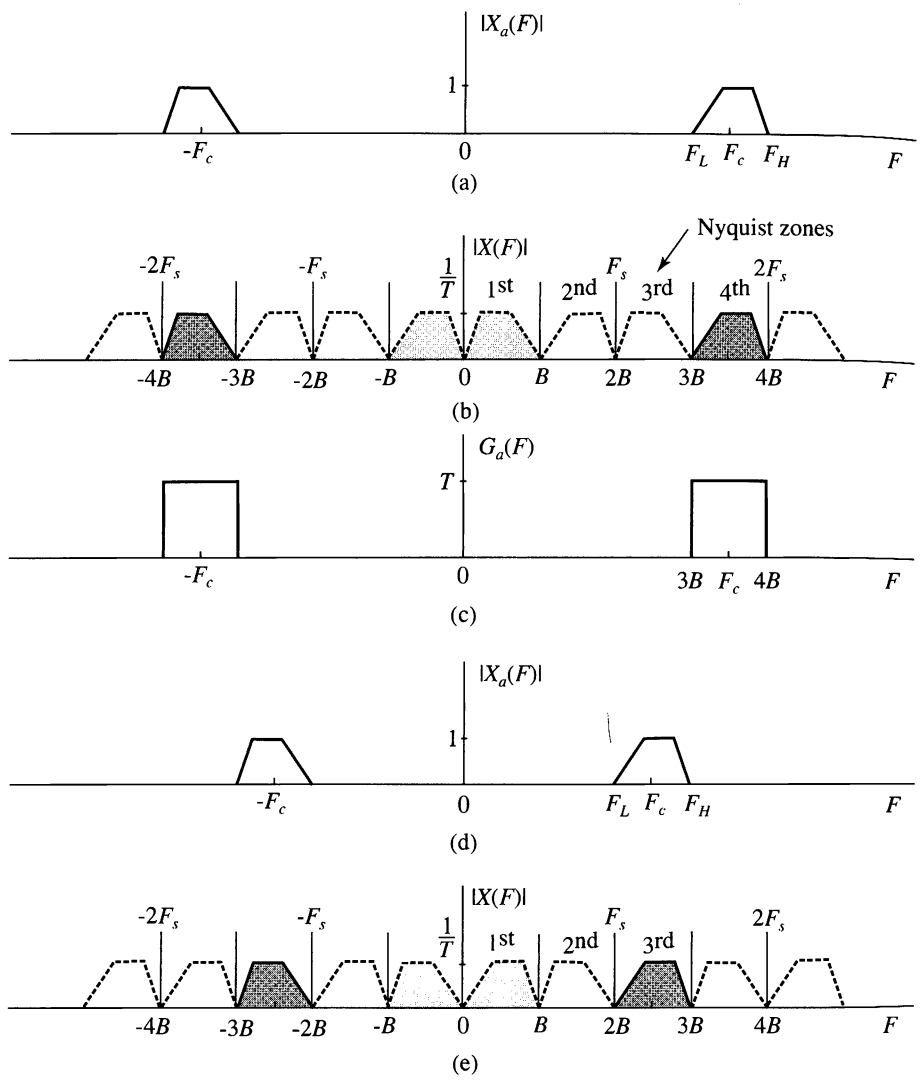


Figure 6.4.1 Illustration of bandpass signal sampling for integer band positioning.

Arbitrary Band Positioning. Consider now a bandpass signal with arbitrarily positioned spectral bands, as shown in Figure 6.4.2. To avoid aliasing, the sampling frequency should be such that the $(k-1)$th and kth shifted replicas of the "negative" spectral band do not overlap with the "positive" spectral band. From Figure 6.4.2(b) we see that this is possible if there is an integer k and a sampling frequency F_s that satisfy the following conditions:

$$2F_H \leq kF_s \tag{6.4.4}$$

$$(k-1)F_s \leq 2F_L \tag{6.4.5}$$

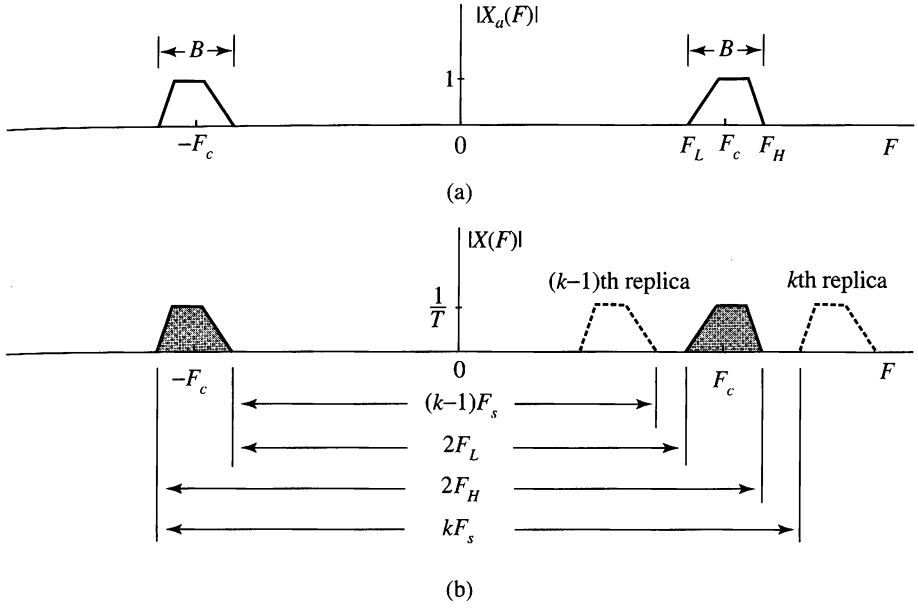


Figure 6.4.2 Illustration of bandpass signal sampling for arbitrary band positioning.

which is a system of two inequalities with two unknowns, k and F_s. From (6.4.4) and (6.4.5) we can easily see that F_s should be in the range

$$\frac{2F_H}{k} \leq F_s \leq \frac{2F_L}{k-1} \tag{6.4.6}$$

To determine the integer k we rewrite (6.4.4) and (6.4.5) as follows:

$$\frac{1}{F_s} \leq \frac{k}{2F_H} \tag{6.4.7}$$

$$(k-1)F_s \leq 2F_H - 2B \tag{6.4.8}$$

By multiplying (6.4.7) and (6.4.8) by sides and solving the resulting inequality for k we obtain

$$k_{\max} \leq \frac{F_H}{B} \tag{6.4.9}$$

The maximum value of integer k is the number of bands that we can fit in the range from 0 to F_H, that is

$$k_{\max} = \left\lfloor \frac{F_H}{B} \right\rfloor \tag{6.4.10}$$

where $\lfloor b \rfloor$ denotes the integer part of b. The minimum sampling rate required to avoid aliasing is $F_{s\max} = 2F_H/k_{\max}$. Therefore, the range of acceptable uniform sampling rates is determined by

$$\frac{2F_H}{k} \leq F_s \leq \frac{2F_L}{k-1} \tag{6.4.11}$$

where k is an integer number given by

$$1 \leq k \leq \left\lfloor \frac{F_H}{B} \right\rfloor \tag{6.4.12}$$

As long as there is no aliasing, reconstruction is done using (6.4.2) and (6.4.3), which are valid for both integer and arbitrary band positioning.

Choosing a Sampling Frequency. To appreciate the implications of conditions (6.4.11) and (6.4.12), we depict them graphically in Figure 6.4.3, as suggested by Vaughan et al. (1991). The plot shows the sampling frequency, normalized by B, as a function of the band position, F_H/B. This is facilitated by rewriting (6.4.11) as follows:

$$\frac{2}{k}\frac{F_H}{B} \leq \frac{F_s}{B} \leq \frac{2}{k-1}\left(\frac{F_H}{B} - 1\right) \tag{6.4.13}$$

The shaded areas represent sampling rates that result in aliasing. The allowed range of sampling frequencies is inside the white wedges. For $k = 1$, we obtain $2F_H \leq F_s \leq \infty$, which is the sampling theorem for lowpass signals. Each wedge in the plot corresponds to a different value of k.

To determine the allowed sampling frequencies, for a given F_H and B, we draw a vertical line at the point determined by F_H/B. The segments of the line within the allowed areas represent permissible sampling rates. We note that the theoretically minimum sampling frequency $F_s = 2B$, corresponding to integer band positioning,

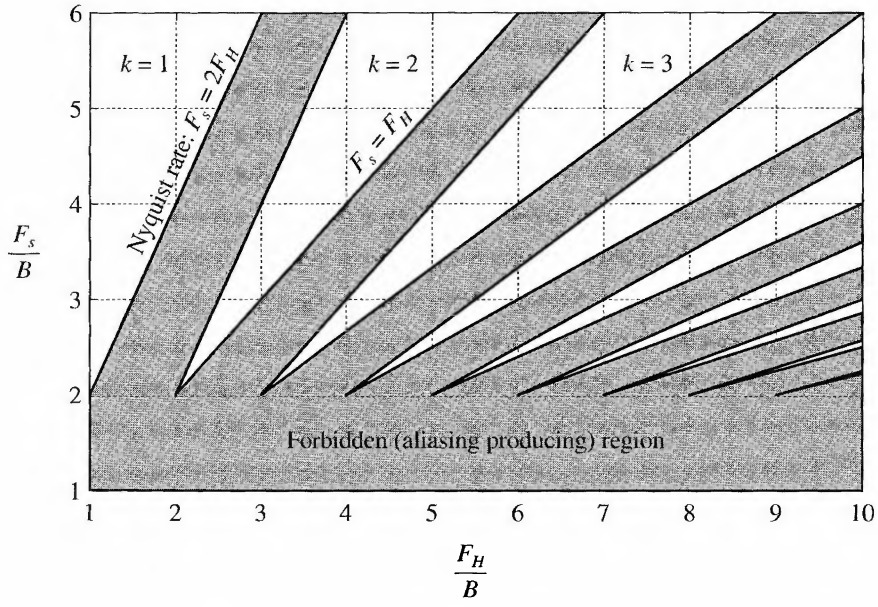


Figure 6.4.3 Allowed (white) and forbidden (shaded) sampling frequency regions for bandpass signals. The minimum sampling frequency $F_s = 2B$, which corresponds to the corners of the alias-free wedges, is possible for integer-positioned bands only.

occurs at the tips of the wedges. Therefore, any small variation of the sampling rate or the carrier frequency of the signal will move the sampling frequency into the forbidden area. A practical solution is to sample at a higher sampling rate, which is equivalent to augmenting the signal band with a guard band $\Delta B = \Delta B_L + \Delta B_H$. The augmented band locations and bandwidth are given by

$$F'_L = F_L - \Delta B_L \tag{6.4.14}$$

$$F'_H = F_H + \Delta B_H \tag{6.4.15}$$

$$B' = B + \Delta B \tag{6.4.16}$$

The lower-order wedge and the corresponding range of allowed sampling are given by

$$\frac{2F'_H}{k'} \le F_s \le \frac{2F'_L}{k'-1} \quad \text{where} \quad k' = \left\lfloor \frac{F'_H}{B'} \right\rfloor \tag{6.4.17}$$

The k'th wedge with the guard bands and the sampling frequency tolerances are illustrated in Figure 6.4.4. The allowable range of sampling rates is divided into values above and below the practical operating points as

$$\Delta F_s = \frac{2F'_L}{k'-1} - \frac{2F'_H}{k'} = \Delta F_{sL} + \Delta F_{sH} \tag{6.4.18}$$

From the shaded orthogonal triangles in Figure 6.4.4, we obtain

$$\Delta B_L = \frac{k'-1}{2}\Delta F_{sH} \tag{6.4.19}$$

$$\Delta B_H = \frac{k'}{2}\Delta F_{sL} \tag{6.4.20}$$

which shows that symmetric guard bands lead to asymmetric sampling rate tolerance.

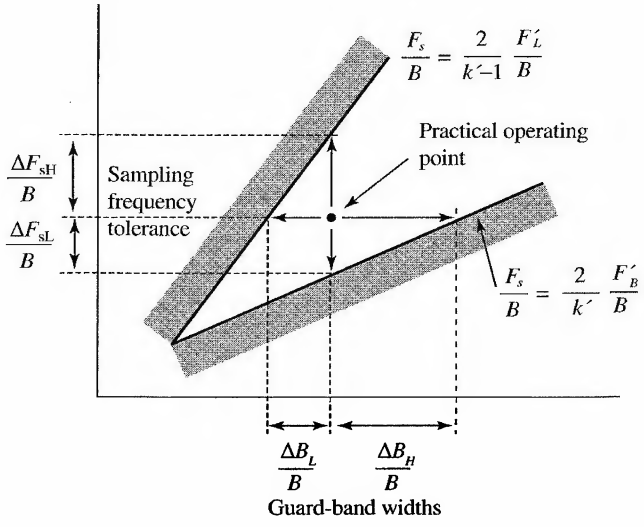


Figure 6.4.4
Illustration of the relationship between the size of guard bands and allowed sampling frequency deviations from its nominal value for the kth wedge.

If we choose the practical operating point at the vertical midpoint of the wedge, the sampling rate is

$$F_s = \frac{1}{2}\left(\frac{2F'_H}{k'} + \frac{2F'_L}{k'-1}\right) \tag{6.4.21}$$

Since, by construction, $\Delta F_{sL} = \Delta F_{sH} = \Delta F_s/2$, the guard bands become

$$\Delta B_L = \frac{k'-1}{4}\Delta F_s \tag{6.4.22}$$

$$\Delta B_H = \frac{k'}{4}\Delta F_s \tag{6.4.23}$$

We next provide an example that illustrates the use of this approach.

EXAMPLE 6.4.1

Suppose we are given a bandpass signal with $B = 25$ kHz and $F_L = 10{,}702.5$ kHz. From (6.4.10) the maximum wedge index is

$$k_{\max} = \lfloor F_H/B \rfloor = 429$$

This yields the theoretically minimum sampling frequency

$$F_s = \frac{2F_H}{k_{\max}} = 50.0117 \text{ kHz}$$

To avoid potential aliasing due to hardware imperfections, we wish to use two guard bands of $\Delta B_L = 2.5$ kHz and $\Delta B_H = 2.5$ kHz on each side of the signal band. The effective bandwidth of the signal becomes $B' = B + \Delta B_L + \Delta B_H = 30$ kHz. In addition, $F'_L = F_L - \Delta B_L = 10{,}700$ kHz and $F'_H = F_H + \Delta B_H = 10{,}730$ kHz. From (6.4.17), the maximum wedge index is

$$k'_{\max} = \lfloor F'_H/B' \rfloor = 357$$

Substitution of $k_{\max}$ into the inequality in (6.4.17) provides the range of acceptable sampling frequencies

$$60.1120 \text{ kHz} \leq F_s \leq 60.1124 \text{ kHz}$$

A detailed analysis on how to choose in practice the sampling rate for bandpass signals is provided by Vaughan et al. (1991) and Qi et al. (1996).

6.4.2 Interleaved or Nonuniform Second-Order Sampling

Suppose that we sample a continuous-time signal $x_a(t)$ with sampling rate $F_i = 1/T_i$ at time instants $t = nT_i + \Delta_i$, where Δ_i is a fixed time offset. Using the sequence

$$x_i(nT_i) = x_a(nT_i + \Delta_i), \quad -\infty < n < \infty \tag{6.4.24}$$

and a reconstruction function $g_a^{(i)}(t)$ we generate a continuous-time signal

$$y_a^{(i)}(t) = \sum_{n=-\infty}^{\infty} x_i(nT_i) g_a^{(i)}(t - nT_i - \Delta_i) \tag{6.4.25}$$

The Fourier transform of $y_a^{(i)}(t)$ is given by

$$Y_a^{(i)}(F) = \sum_{n=-\infty}^{\infty} x_i(nT_i) G_a^{(i)}(F) e^{-j2\pi F(nT_i + \Delta_i)} \tag{6.4.26}$$

$$= G_a^{(i)}(F) X_i(F) e^{-j2\pi F \Delta_i} \tag{6.4.27}$$

where $X_i(F)$ is the Fourier transform of $x_i(nT_i)$. From the sampling theorem (6.1.14), the Fourier transform of $x_i(nT_i)$ can be expressed in terms of the Fourier transform $X_a(F)e^{j2\pi F\Delta_i}$ of $x_a(t + \Delta_i)$ as

$$X_i(F) = \frac{1}{T_i} \sum_{k=-\infty}^{\infty} X_a\left(F - \frac{k}{T_i}\right) e^{j2\pi(F - \frac{k}{T_i})\Delta_i} \tag{6.4.28}$$

Substitution of (6.4.28) into (6.4.27) yields

$$Y_a^{(i)}(F) = G_a^{(i)}(F) \frac{1}{T_i} \sum_{k=-\infty}^{\infty} X_a\left(F - \frac{k}{T_i}\right) e^{-j2\pi \frac{k}{T_i} \Delta_i} \tag{6.4.29}$$

If we repeat the sampling process (6.4.24) for $i = 1, 2, \ldots, p$, we obtain p interleaved uniformly sampled sequences $x_i(nT_i)$, $-\infty < n < \infty$. The sum of the p reconstructed signals is given by

$$y_a(t) = \sum_{i=1}^{p} y_a^{(i)}(t) \tag{6.4.30}$$

Using (6.4.29) and (6.4.30), the Fourier transform of $y_a(t)$ can be expressed as

$$Y_a(F) = \sum_{i=1}^{p} G_a^{(i)}(F) V^{(i)}(F) \tag{6.4.31}$$

where

$$V^{(i)}(F) = \frac{1}{T_i} \sum_{k=-\infty}^{\infty} X_a\left(F - \frac{k}{T_i}\right) e^{-j2\pi \frac{k}{T_i} \Delta_i} \tag{6.4.32}$$

We will focus on the most commonly used second-order sampling, defined by

$$p = 2, \Delta_1 = 0, \Delta_2 = \Delta, T_1 = T_2 = \frac{1}{B} = T \tag{6.4.33}$$

In this case, which is illustrated in Figure 6.4.5, relations (6.4.31) and (6.4.32) yield

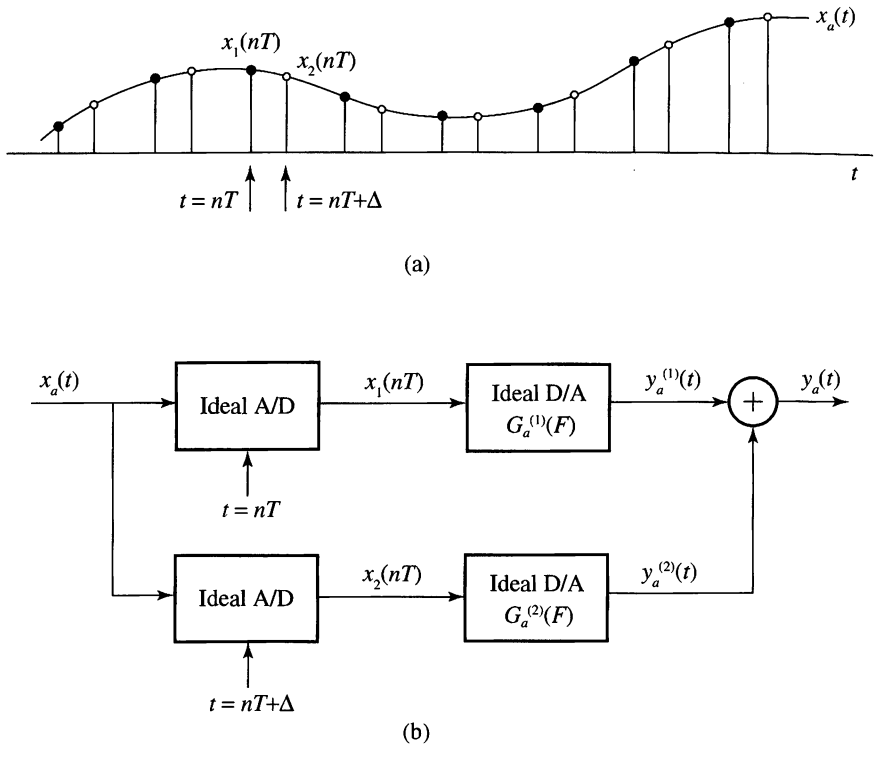


Figure 6.4.5 Illustration of second-order bandpass sampling: (a) interleaved sampled sequences (b) second-order sampling and reconstruction system.

$$Y_a(F) = BG_a^{(1)}(F) \sum_{k=-\infty}^{\infty} X_a(F - kB) + BG_a^{(2)}(F) \sum_{k=-\infty}^{\infty} \gamma^k X_a(F - kB) \tag{6.4.34}$$

where

$$\gamma = e^{-j2\pi B\Delta} \tag{6.4.35}$$

To understand the nature of (6.4.34) we first split the spectrum $X_a(F)$ into a "positive" band and a "negative" band as follows:

$$X_a^+(F) = \begin{cases} X_a(F), & F \geq 0 \\ 0, & F < 0 \end{cases}, \quad X_a^-(F) = \begin{cases} X_a(F), & F \leq 0 \\ 0, & F > 0 \end{cases} \tag{6.4.36}$$

Then, we plot the repeated replicas of $X_a(F-kB)$ and $\gamma^k X_a(F-kB)$ as four separate components, as illustrated in Figure 6.4.6. We note that because each individual component has bandwidth B and sampling rate $F_s = 1/B$, its repeated copies fill the entire frequency axis without overlapping, that is, without aliasing. However, when we combine them, the negative bands cause aliasing to the positive bands, and vice versa.

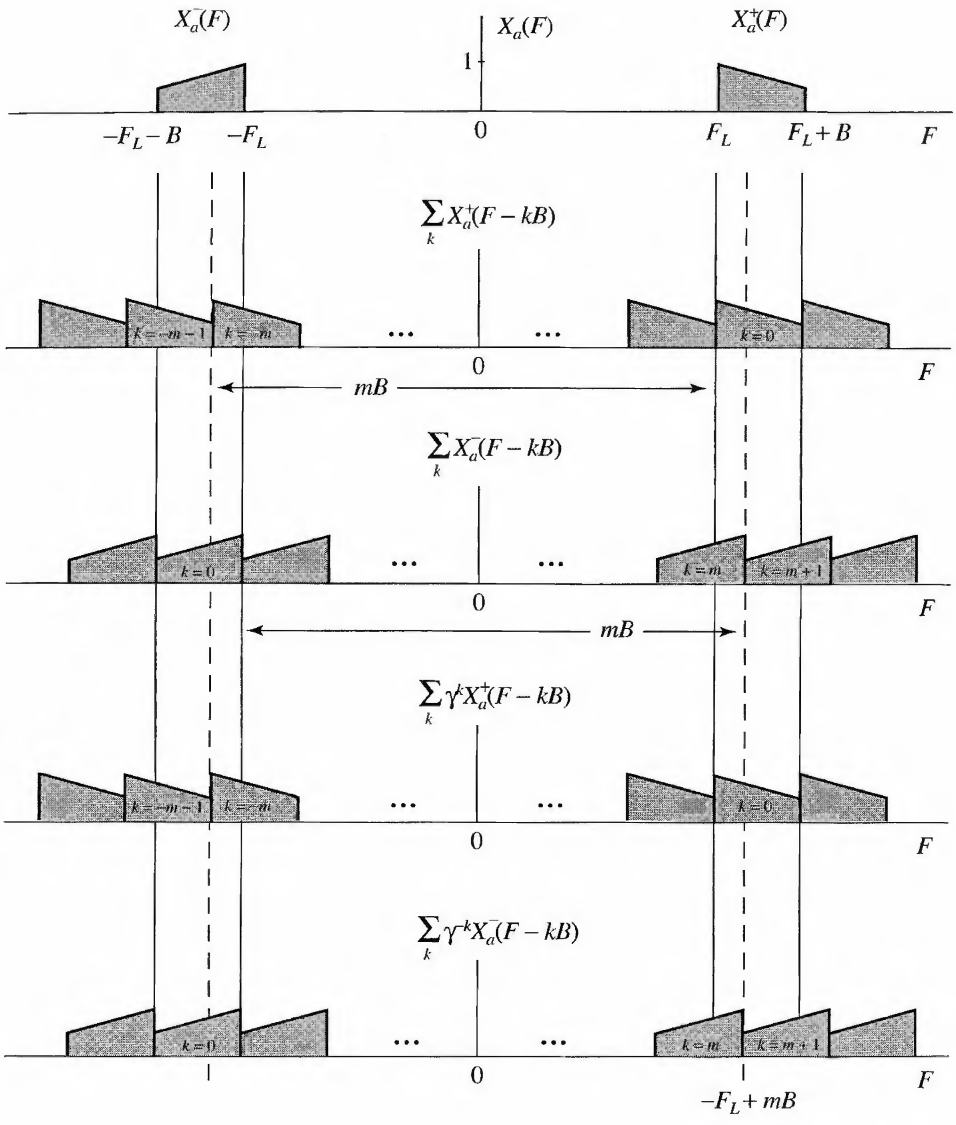


Figure 6.4.6 Illustration of aliasing in second-order bandpass sampling.

We want to determine the interpolation functions $G_a^{(1)}(F)$, $G_a^{(2)}(F)$, and the time offset Δ so that $Y_a(F) = X_a(F)$. From Figure 6.4.6 we see that the first requirement is

$$G_a^{(1)}(F) = G_a^{(2)}(F) = 0, \quad \text{for } |F| < F_L \text{ and } |F| > F_L + B \tag{6.4.37}$$

To determine $G_a^{(1)}(F)$ and $G_a^{(2)}(F)$ for $F_L \le |F| \le F_L + B$, we see from Figure 6.4.6 that only the components with $k = \pm m$ and $k = \pm(m+1)$, where

$$m = \left\lceil \frac{2F_L}{B} \right\rceil \tag{6.4.38}$$

is the smallest integer greater or equal to $2F_L/B$, overlap with the original spectrum.

In the region $F_L \le F \le -F_L + mB$, equation (6.4.34) becomes

$$Y_a^+(F) = \left[BG_a^{(1)}(F) + BG_a^{(2)}(F)\right] X_a^+(F) \qquad \text{(Signal component)}$$
$$+ \left[BG_a^{(1)}(F) + B\gamma^m G_a^{(2)}(F)\right] X_a^+(F - mB) \qquad \text{(Aliasing component)}$$

The conditions that assure perfect reconstruction $Y_a^+(F) = X_a^+(F)$ are given by

$$BG_a^{(1)}(F) + BG_a^{(2)}(F) = 1 \tag{6.4.39}$$
$$BG_a^{(1)}(F) + B\gamma^m G_a^{(2)}(F) = 0 \tag{6.4.40}$$

Solving this system of equations yields the solution

$$G_a^{(1)}(F) = \frac{1}{B}\frac{1}{1-\gamma^{-m}}, \quad G_a^{(2)}(F) = \frac{1}{B}\frac{1}{1-\gamma^m} \tag{6.4.41}$$

which exists for all Δ such that $\gamma^{\pm m} = e^{\mp j2\pi mB\Delta} \neq 1$.

In the region $-F_L + mB \le F \le F_L + B$, equation (6.4.34) becomes

$$Y_a^+(F) = \left[BG_a^{(1)}(F) + BG_a^{(2)}(F)\right] X_a^+(F)$$
$$+ \left[BG_a^{(1)}(F) + B\gamma^{m+1} G_a^{(2)}(F)\right] X_a^+(F - (m+1)B)$$

The conditions that assure perfect reconstruction $Y_a^+(F) = X_a^+(F)$ are given by

$$BG_a^{(1)}(F) + BG_a^{(2)}(F) = 1 \tag{6.4.42}$$
$$BG_a^{(1)}(F) + B\gamma^{m+1} G_a^{(2)}(F) = 0 \tag{6.4.43}$$

Solving this system of equations yields the solution

$$G_a^{(1)}(F) = \frac{1}{B}\frac{1}{1-\gamma^{-(m+1)}}, \quad G_a^{(2)}(F) = \frac{1}{B}\frac{1}{1-\gamma^{m+1}} \tag{6.4.44}$$

which exists for all Δ such that $\gamma^{\pm(m+1)} = e^{\mp j2\pi(m+1)B\Delta} \neq 1$.

The reconstruction functions in the frequency range $-(F_L + B) \leq F \leq -F_L$ can be obtained in a similar manner. The formulas are given by (6.4.41) and (6.4.44) if we replace m by $-m$ and $m+1$ by $-(m+1)$. The function $G_a^{(1)}(F)$ has the bandpass response shown in Figure 6.4.7. A similar plot for $G_a^{(2)}(F)$ reveals that

$$G_a^{(2)}(F) = G_a^{(1)}(-F) \tag{6.4.45}$$

which implies that $g_a^{(2)}(t) = g_a^{(1)}(-t)$. Therefore, for simplicity, we adopt the notation $g_a(t) = g_a^{(1)}(t)$ and express the reconstruction formula (6.4.30) as follows

$$x_a(t) = \sum_{n=-\infty}^{\infty} x_a\left(\frac{n}{B}\right) g_a\left(t - \frac{n}{B}\right) + x_a\left(\frac{n}{B} + \Delta\right) g_a\left(-t + \frac{n}{B} + \Delta\right) \tag{6.4.46}$$

Taking the inverse Fourier transform of the function shown in Figure 6.4.7, we can show (see Problem 6.7) that the interpolation function is given by

$$g_a(t) = a(t) + b(t) \tag{6.4.47}$$

$$a(t) = \frac{\cos[2\pi(mB - F_L)t - \pi m B\Delta] - \cos(2\pi F_L t - \pi m B\Delta)}{2\pi Bt \sin(\pi m B\Delta)} \tag{6.4.48}$$

$$b(t) = \frac{\cos[2\pi(F_L + B)t - \pi(m+1)B\Delta] - \cos[2\pi(mB - F_L)t - \pi(m+1)B\Delta]}{2\pi Bt \sin[\pi(m+1)B\Delta]} \tag{6.4.49}$$

We can see that $g_a(0) = 1$, $g_a(n/B) = 0$ for $n \neq 0$, and $g_a(n/B \pm \Delta) = 0$ for $n = 0, \pm 1, \pm 2, \ldots$, as expected for any interpolation function.

We have shown that a bandpass signal $x_a(t)$ with frequencies in the range $F_L \leq |F| \leq F_L + B$ can be perfectly reconstructed from two interleaved uniformly sampled sequences $x_a(n/B)$ and $x_a(n/B + \Delta)$, $-\infty < n < \infty$, using the interpolation formula (6.4.46) with an average rate $F_s = 2B$ samples/second without any restrictions on the band location. The time offset Δ cannot take values that may cause the interpolation function to take infinite values. This second-order sampling theorem was introduced by Kohlenberg (1953). The general pth-order sampling case ($p > 2$) is discussed by Coulson (1995).

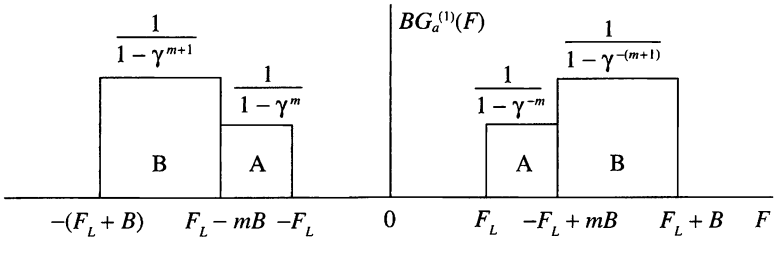


Figure 6.4.7 Frequency domain characterization of the bandpass interpolation function for second-order sampling.

Some useful simplifications occur when $m = 2F_L/B$, that is, for integer band positioning (Linden 1959, Vaughan et al. 1991). In this case, the region A becomes zero, which implies that $a(t) = 0$. Therefore, we have $g_a(t) = b(t)$. There are two cases of special interest.

For low pass signals, $F_L = 0$ and $m = 0$, and the interpolation function becomes

$$g_{\text{LP}}(t) = \frac{\cos(2\pi Bt - \pi B\Delta) - \cos(\pi B\Delta)}{2\pi Bt \sin(\pi B\Delta)} \tag{6.4.50}$$

The additional constraint $\Delta = 1/2B$, which results in uniform sampling rate, yields the well-known sine interpolation function $g_{\text{LP}}(t) = \sin(2\pi Bt)/2\pi Bt$.

For bandpass signals with $F_L = mB/2$ we can choose the time offset Δ such that $\gamma^{\pm(m+1)} = -1$. This requirement is satisfied if

$$\Delta = \frac{2k+1}{2B(m+1)} = \frac{1}{4F_c} + \frac{k}{2F_c}, \quad k = 0, \pm 1, \pm 2, \ldots \tag{6.4.51}$$

where $F_c = F_L + B/2 = B(m+1)/2$ is the center frequency of the band. In this case, the interpolation function is specified by $G_Q(F) = 1/2$ in the range $mB/2 \leq |F| \leq (m+1)B/2$ and $G_Q(F) = 0$ elsewhere. Taking the inverse Fourier transform, we obtain

$$g_Q(t) = \frac{\sin \pi Bt}{\pi Bt} \cos 2\pi F_c t \tag{6.4.52}$$

which is a special case of (6.4.47)–(6.4.49). This special case is known as direct quadrature sampling because the in-phase and quadrature components are obtained explicitly from the bandpass signal (see Section 6.4.3).

Finally, we note that it is possible to sample a bandpass signal, and then to reconstruct the discrete-time signal at a band position other than the original. This spectral relocation or frequency shifting of the bandpass signal is most commonly done using direct quadrature sampling [Coulson et al. (1994)]. The significance of this approach is that it can be implemented using digital signal processing.

6.4.3 Bandpass Signal Representations

The main cause of complications in the sampling of a real bandpass signal $x_a(t)$ is the presence of two separate spectral bands in the frequency regions $-(F_L + B) \leq F \leq -F_L$ and $F_L \leq F \leq F_L + B$. Since $x_a(t)$ is real, the negative and positive frequencies in its spectrum are related by

$$X_a(-F) = X_a^*(F) \tag{6.4.53}$$

Therefore, the signal can be completely specified by one half of the spectrum. We next exploit this idea to introduce simplified representations for bandpass signals. We start with the identity

$$\cos 2\pi F_c t = \frac{1}{2} e^{j2\pi F_c t} + \frac{1}{2} e^{-j2\pi F_c t} \tag{6.4.54}$$

which represents the real signal $\cos 2\pi F_c t$ by two spectral lines of magnitude 1/2, one at $F = F_c$ and the other at $F = -F_c$. Equivalently, we have the identity

$$\cos 2\pi F_c t = 2\Re\left\{\frac{1}{2}e^{j2\pi F_c t}\right\} \tag{6.4.55}$$

which represents the real signal as the real part of a complex signal. In terms of the spectrum, we now specify the real signal $\cos 2\pi F_c t$ by the positive part of its spectrum, that is, the spectral line at $F = F_c$. The amplitude of the positive frequencies is doubled to compensate for the omission of the negative frequencies.

The extension to signals with continuous spectra is straightforward. Indeed, the integral of the inverse Fourier transform of $x_a(t)$ can be split into two parts as

$$x_a(t) = \int_0^\infty X_a(F)e^{j2\pi Ft}dF + \int_{-\infty}^0 X_a(F)e^{j2\pi Ft}\,dF \tag{6.4.56}$$

Changing the variable in the second integral from F to $-F$ and using (6.4.53) yields

$$x_a(t) = \int_0^\infty X_a(F)e^{j2\pi Ft}\,dF + \int_0^\infty X_a^*(F)e^{-j2\pi Ft}\,dF \tag{6.4.57}$$

The last equation can be equivalently written as

$$x_a(t) = \Re\left\{\int_0^\infty 2X_a(F)e^{j2\pi Ft}\,dF\right\} = \Re\{\psi_a(t)\} \tag{6.4.58}$$

where the complex signal

$$\psi_a(t) = \int_0^\infty 2X_a(F)e^{j2\pi Ft}\,dF \tag{6.4.59}$$

is known as the *analytic signal* or the *pre-envelope* of $x_a(t)$. The spectrum of the analytic signal can be expressed in terms of the unit step function $V_a(F)$ as follows:

$$\Psi_a(F) = 2X_a(F)V_a(F) = \begin{cases} 2X_a(F), & F > 0 \\ 0, & F < 0 \end{cases} \tag{6.4.60}$$

In case $X_a(0) \neq 0$, we define $\Psi_a(0) = X_a(0)$. To express the analytic signal $\psi_a(t)$ in terms of $x_a(t)$, we recall that the inverse Fourier transform of $V_a(F)$ is given by

$$v_a(t) = \frac{1}{2}\delta(t) + \frac{j}{2\pi t} \tag{6.4.61}$$

From (6.4.60), (6.4.61), and the frequency-domain convolution theorem, we obtain

$$\psi_a(t) = 2x_a(t) * v_a(t) = x_a(t) + j\frac{1}{\pi t} * x_a(t) \tag{6.4.62}$$

The signal obtained from the convolution of the impulse response

$$h_Q(t) = \frac{1}{\pi t} \tag{6.4.63}$$

and the input signal $x_a(t)$ is given by

$$\hat{x}_a(t) = \frac{1}{\pi t} * x_a(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x_a(\tau)}{t - \tau} \, d\tau \tag{6.4.64}$$

and it is called the *Hilbert transform* of $x_a(t)$, denoted by $\hat{x}_a(t)$. We emphasize that the Hilbert transform is a convolution and does not change the domain, so both its input $x_a(t)$ and its output $\hat{x}_a(t)$ are functions of time.

The filter defined by (6.4.63) has the frequency response function given by

$$H_Q(F) = \int_{-\infty}^{\infty} h_Q(t) e^{-j2\pi Ft} \, dt = \begin{cases} -j, & F > 0 \\ j, & F < 0 \end{cases} \tag{6.4.65}$$

or in terms of magnitude and phase

$$|H_Q(F)| = 1, \quad \measuredangle H_Q(F) = \begin{cases} -\pi/2, & F > 0 \\ \pi/2, & F < 0 \end{cases} \tag{6.4.66}$$

The Hilbert transformer $H_Q(F)$ is an allpass quadrature filter that simply shifts the phase of positive frequency components by $-\pi/2$ and the phase of negative frequency components by $\pi/2$. From (6.4.63) we see that $h_Q(t)$ is noncausal, which means that the Hilbert transformer is physically unrealizable.

We can now express the analytic signal using the Hilbert transform as

$$\psi_a(t) = x_a(t) + j\hat{x}_a(t) \tag{6.4.67}$$

We see that the Hilbert transform of $x_a(t)$ provides the imaginary part of its analytic signal representation.

The analytic signal $\psi_a(t)$ of $x_a(t)$ is bandpass in the region $F_L \leq F \leq F_L + B$. Therefore, it can be shifted to the baseband region $-B/2 \leq F \leq B/2$ using the modulation property of the Fourier transform

$$x_{\text{LP}}(t) = e^{-j2\pi F_c t} \psi_a(t) \overset{\mathcal{F}}{\longleftrightarrow} X_{\text{LP}}(F) = \Psi_a(F + F_c) \tag{6.4.68}$$

The complex lowpass signal $x_{\text{LP}}(t)$ is known as the *complex envelope* of $x_a(t)$.

The complex envelope can be expressed in rectangular coordinates as

$$x_{\text{LP}}(t) = x_I(t) + jx_Q(t) \tag{6.4.69}$$

where $x_I(t)$ and $x_Q(t)$ are both real-valued lowpass signals in the same frequency region with $x_{\text{LP}}(t)$. From (6.4.58), (6.4.68), and (6.4.69) we can easily deduce the so-called *quadrature representation* of bandpass signals

$$x_a(t) = x_I(t) \cos 2\pi F_c t - x_Q(t) \sin 2\pi F_c t \tag{6.4.70}$$

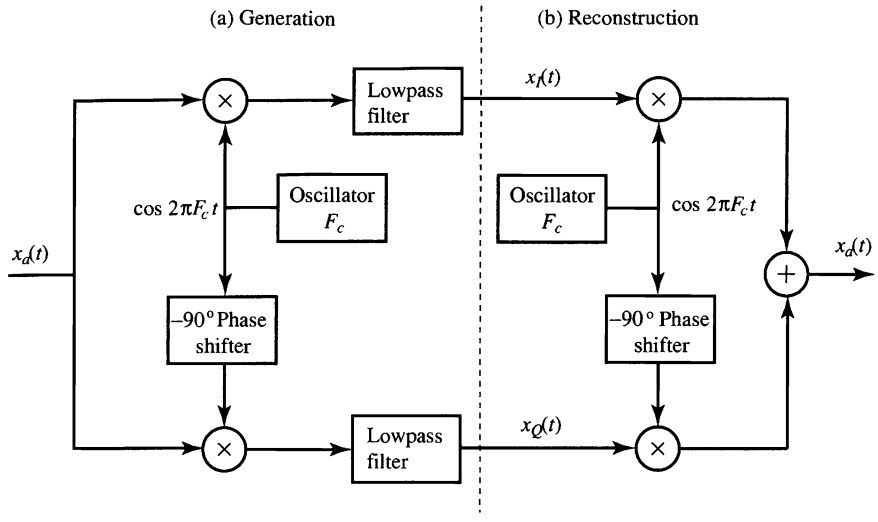


Figure 6.4.8 (a) Scheme for generating the in-phase and quadrature components of a bandpass signal. (b) Scheme for reconstructing a bandpass signal from its in-phase and quadrature components.

We refer to $x_I(t)$ as the *in-phase component* of the bandpass signal and $x_Q(t)$ as the *quadrature component* because the carriers $\cos 2\pi F_c t$ and $\sin 2\pi F_c t$ are in-phase quadrature with respect to each other. The in-phase and quadrature components can be obtained from the signal $x_a(t)$ using quadrature demodulation as shown in Figure 6.4.8(a). The bandpass signal can be reconstructed using the scheme in 6.4.8(b).

Alternatively, we can express the complex envelope in polar coordinates as

$$x_{\text{LP}}(t) = A(t)e^{j\phi(t)} \tag{6.4.71}$$

where $A(t)$ and $\phi(t)$ are both real-valued lowpass signals. In terms of this polar representation, the bandpass signal $x_a(t)$ can be written as

$$x_a(t) = A(t)\cos[2\pi F_c t + \phi(t)] \tag{6.4.72}$$

where $A(t)$ is known as the *envelope* and $\phi(t)$ as the *phase* of the bandpass signal. Equation (6.4.72) represents a bandpass signal using a combination of amplitude and angle modulation. We can easily see that $x_I(t)$ and $x_Q(t)$ are related to $A(t)$ and $\phi(t)$ as follows:

$$x_I(t) = A(t)\cos 2\pi F_c t, \qquad x_Q(t) = A(t)\sin 2\pi F_c t \tag{6.4.73}$$

$$A(t) = \sqrt{x_I^2(t) + x_Q^2(t)}, \qquad \phi(t) = \tan^{-1}\left[\frac{x_Q(t)}{x_I(t)}\right] \tag{6.4.74}$$

The phase $\phi(t)$ is uniquely defined in terms of $x_I(t)$ and $x_Q(t)$, modulo 2π.

6.4.4 Sampling Using Bandpass Signal Representations

The complex envelope (6.4.68) and quadrature representations (6.4.70) allow the sampling of a bandpass signal at a rate $F_s = 2B$ independently of the band location.

Since the analytic signal $\psi_a(t)$ has a single band at $F_L \leq F \leq F_L + B$, it can be sampled at a rate of B complex samples per second or $2B$ real samples per second without aliasing (see second graph in Figure 6.4.6). According to (6.4.67) these samples can be obtained by sampling $x_a(t)$ and its Hilbert transform $\hat{x}_a(t)$ at a rate of B samples per second. Reconstruction requires a complex bandpass interpolation function defined by

$$g_a(t) = \frac{\sin \pi Bt}{\pi Bt} e^{j2\pi F_c t} \overset{\mathcal{F}}{\longleftrightarrow} G_a(F) = \begin{cases} 1, & F_L \leq F \leq F_L + B \\ 0, & \text{otherwise} \end{cases} \tag{6.4.75}$$

where $F_c = F_L + B/2$. The major problem with this approach is the design of practical analog Hilbert transformers.

Similarly, since the in-phase $x_I(t)$ and quadrature $x_Q(t)$ components of the bandpass signal $x_a(t)$ are lowpass signals with one-sided bandwidth $B/2$, they can be uniquely represented by the sequences $x_I(nT)$ and $x_Q(nT)$, where $T = 1/B$. This results in a total sampling rate of $F_s = 2B$ real samples per second. The original bandpass signal can be reconstructed by first reconstructing the in-phase and quadrature components using ideal interpolation and then recombining them using formula (6.4.70).

The in-phase and quadrature components can be obtained by directly sampling the signal $x_a(t)$, using second-order sampling with a proper choice of Δ. This leads to a major simplification because we can avoid the complex demodulation process required to generate the in-phase and quadrature signals. To extract directly $x_I(t)$ from $x_a(t)$, that is

$$x_a(t_n) = x_I(t_n) \tag{6.4.76}$$

requires sampling at time instants

$$2\pi F_c t_n = \pi n, \text{ or } t_n = \frac{n}{2F_c}, n = 0, \pm 1, \pm 2, \ldots \tag{6.4.77}$$

Similarly, to obtain $x_Q(t)$, we should sample at time instants

$$2\pi F_c t_n = \frac{\pi}{2}(2n+1), \text{ or } t_n = \frac{2n+1}{4F_c}, n = 0, \pm 1, \pm 2, \ldots \tag{6.4.78}$$

which yields

$$x_a(t_n) = -x_Q(t_n) \tag{6.4.79}$$

which is equivalent to the special case second-order sampling defined by (6.4.51). Several variations of this approach are described by Grace and Pitt (1969), Rice and Wu (1982), Waters and Jarret (1982), and Jackson and Matthewson (1986).

Finally, we note that the quadrature approach to bandpass sampling has been widely used in radar and communications systems to generate in-phase and quadrature sequences for further processing. However, with the development of faster A/D

converters and digital signal processors it is more convenient and economic to sample directly the bandpass signal, as described in Section 6.4.1, and then obtain $x_I(n)$ and $x_Q(n)$ using the discrete-time approach developed in Section 6.5.

5 Sampling of Discrete-Time Signals

In this section we use the techniques developed for the sampling and representation of continuous-time signals to discuss the sampling and reconstruction of lowpass and bandpass discrete-time signals. Our approach is to conceptually reconstruct the underlying continuous-time signal and then resample at the desired sampling rate. However, the final implementations involve only discrete-time operations. The more general area of sampling rate conversion is the subject of Chapter 11.

6.5.1 Sampling and Interpolation of Discrete-Time Signals

Suppose that a sequence $x(n)$ is sampled periodically by keeping every Dth sample of $x(n)$ and deleting the $(D-1)$ samples in between. This operation, which is also known as decimation or down-sampling, yields a new sequence defined by

$$x_d(n) = x(nD), \quad -\infty < n < \infty \tag{6.5.1}$$

Without loss of generality we assume that $x(n)$ has been obtained by sampling a signal $x_a(t)$ with spectrum $X_a(F) = 0$, $|F| > B$ at a sampling rate $F_s = 1/T \geq 2B$, that is, $x(n) = x_a(nT)$. Therefore, the spectrum $X(F)$ of $x(n)$ is given by

$$X(F) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X_a(F - kF_s) \tag{6.5.2}$$

We next sample $x_a(t)$ at time instants $t = nDT$, that is, with a sampling rate F_s/D. The spectrum of the sequence $x_d(n) = x_a(nDT)$ is provided by

$$X_d(F) = \frac{1}{DT} \sum_{k=-\infty}^{\infty} X_a\left(F - k\frac{F_s}{D}\right) \tag{6.5.3}$$

This process is illustrated in Figure 6.5.1 for $D = 2$ and $D = 4$. We can easily see from Figure 6.5.1(c) that the spectrum $X_d(F)$ can be expressed in terms of the periodic spectrum $X(F)$ as

$$X_d(F) = \frac{1}{D} \sum_{k=0}^{D-1} X\left(F - k\frac{F_s}{D}\right) \tag{6.5.4}$$

To avoid aliasing, the sampling rate should satisfy the condition $F_s/D \geq 2B$. If the sampling frequency F_s is fixed, we can avoid aliasing by reducing the bandwidth of $x(n)$ to $(F_s/2)/D$. In terms of the normalized frequency variables, we can avoid aliasing if the highest frequency $f_{\max}$ or $\omega_{\max}$ in $x(n)$ satisfies the conditions

$$f_{\max} \leq \frac{1}{2D} = \frac{f_s}{2} \quad \text{or} \quad \omega_{\max} \leq \frac{\pi}{D} = \frac{\omega_s}{2} \tag{6.5.5}$$

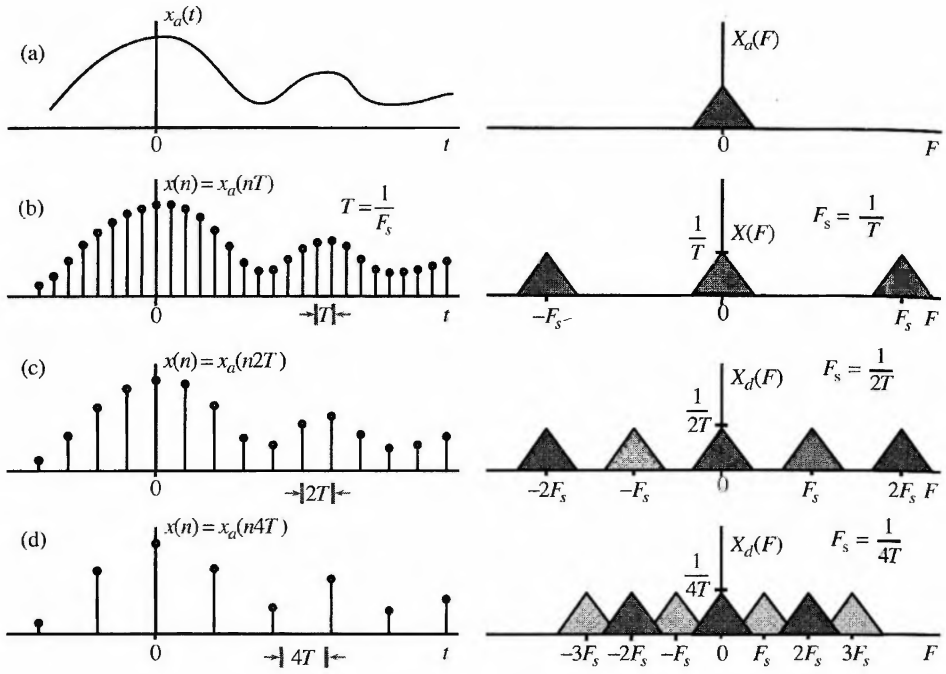


Figure 6.5.1 Illustration of discrete-time signal sampling in the frequency domain.

In continuous-time sampling the continuous-time spectrum $X_a(F)$ is repeated an infinite number of times to create a periodic spectrum covering the infinite frequency range. In discrete-time sampling the periodic spectrum $X(F)$ is repeated D times to cover one period of the periodic frequency domain.

To reconstruct the original sequence $x(n)$ from the sampled sequence $x_d(n)$, we start with the ideal interpolation formula

$$x_a(t) = \sum_{m=-\infty}^{\infty} x_d(m) \frac{\sin \frac{\pi}{DT}(t - mDT)}{\frac{\pi}{DT}(t - mDT)} \tag{6.5.6}$$

which reconstructs $x_a(t)$ assuming that $F_s/D \geq 2B$. Since $x(n) = x_a(nT)$, substitution into (6.5.6) yields

$$x(n) = \sum_{m=-\infty}^{\infty} x_d(m) \frac{\sin \frac{\pi}{D}(n - mD)}{\frac{\pi}{D}(n - mD)} \tag{6.5.7}$$

This is not a practical interpolator, since the $\sin(x)/x$ function is infinite in extent. In practice, we use a finite summation from $m = -L$ to $m = L$. The quality of this approximation improves with increasing L. The Fourier transform of the ideal bandlimited interpolation sequence in (6.5.7) is

$$g_{\text{BL}}(n) = D\frac{\sin(\pi/D)n}{\pi n} \overset{\mathcal{F}}{\longleftrightarrow} G_{\text{BL}}(\omega) = \begin{cases} D, & |\omega| \leq \pi/D \\ 0, & \pi/D < |\omega| \leq \pi \end{cases} \tag{6.5.8}$$

Therefore, the ideal discrete-time interpolator has an ideal lowpass frequency characteristic.

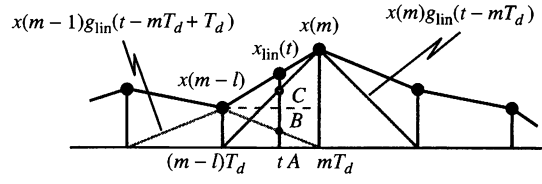


Figure 6.5.2 Illustration of continuous-time linear interpolation.

To understand the process of discrete-time interpolation, we will analyze the widely used linear interpolation. For simplicity we use the notation $T_d = DT$ for the sampling period of $x_d(m) = x_a(mT_d)$. The value of $x_a(t)$ at a time instant between mT_d and $(m+1)T_d$ is obtained by raising a vertical line from t to the line segment connecting the samples $x_d(mT_d)$ and $x_d(mT_d + T_d)$, as shown in Figure 6.5.2. The interpolated value is given by

$$x_{\text{lin}}(t) = x(m-1) + \frac{x(m) - x(m-1)}{T_d}[t - (m-1)T_d], \; (m-1)T_d \le t \le mT_d \quad (6.5.9)$$

which can be rearranged as follows:

$$x_{\text{lin}}(t) = \left[1 - \frac{t - (m-1)T_d}{T_d}\right] x(m-1) + \left[1 - \frac{mT_d - t)}{T_d}\right] x(m) \quad (6.5.10)$$

To put (6.5.10) in the form of the general reconstruction formula

$$x_{\text{lin}}(t) = \sum_{m=-\infty}^{\infty} x(m) g_{\text{lin}}(t - mT_d) \quad (6.5.11)$$

we note that we always have $t - (m-1)T_d = |t - (m-1)T_d|$ and $mT_d - t = |t - mT_d|$ because $(m-1)T_d \le t \le mT_d$. Therefore, we can express (6.5.10) in the form (6.5.11) if we define

$$g_{\text{lin}}(t) = \begin{cases} 1 - \frac{|t|}{T_d}, & |t| \le T_d \\ 0, & |t| > T_d \end{cases} \quad (6.5.12)$$

The discrete-time interpolation formulas are obtained by replacing t by nT in (6.5.11) and (6.5.12). Since $T_d = DT$, we obtain

$$x_{\text{lin}}(n) = \sum_{m=-\infty}^{\infty} x(m) g_{\text{lin}}(n - mD) \quad (6.5.13)$$

where

$$g_{\text{lin}}(n) = \begin{cases} 1 - \frac{|n|}{D}, & |n| \le D \\ 0, & |n| > D \end{cases} \quad (6.5.14)$$

As expected from any interpolation function, $g_{\text{lin}}(0) = 1$ and $g_{\text{lin}}(n) = 0$ for $n = \pm D, \pm 2D, \ldots$. The performance of the linear interpolator can be assessed by comparing its Fourier transform

$$G_{\text{lin}}(\omega) = \frac{1}{D}\left[\frac{\sin(\omega D/2)}{\sin(\omega/2)}\right]^2 \quad (6.5.15)$$

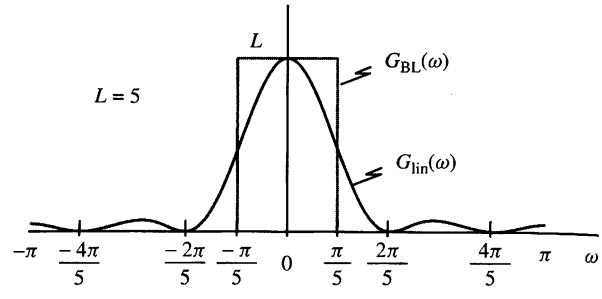


Figure 6.5.3
Frequency response of ideal and linear discrete-time interpolators.

to that of the ideal interpolator (6.5.8). This is illustrated in Figure 6.5.3 which shows that the linear interpolator has good performance only when the spectrum of the interpolated signal is negligible for $|\omega| > \pi/D$, that is, when the original continuous-time signal has been oversampled.

Equations (6.5.11) and (6.5.13) resemble a convolution summation; however, they are *not* convolutions. This is illustrated in Figure 6.5.4 which shows the computation of interpolated samples $x(nT)$ and $x((n+1)T)$ for $D = 5$. We note that only a subset of the coefficients of the linear interpolator is used in each case. Basically, we decompose $g_{\text{lin}}(n)$ into D components and we use one at a time periodically to compute the interpolated values. This is essentially the idea behind the polyphase filter structures discussed in Chapter 11. However, if we create a new sequence $\tilde{x}(n)$ by inserting $(D-1)$ zero samples between successive samples of $x_d(m)$, we can compute $x(n)$ using the convolution

$$x(n) = \sum_{k=-\infty}^{\infty} \tilde{x}(k) g_{\text{lin}}(n-k) \tag{6.5.16}$$

at the expense of unnecessary computations involving zero values. A more efficient implementation can be obtained using equation (6.5.13).

Sampling and interpolation of a discrete-time signal essentially corresponds to a change of its sampling rate by an integer factor. The subject of sampling rate conversion, which is very important in practical applications, it is extensively discussed in Chapter 11.

6.5.2 Representation and Sampling of Bandpass Discrete-Time Signals

The bandpass representations of continuous-time signals, discussed in Section 6.4.3, can be adapted for discrete-time signals with some simple modifications that take into consideration the periodic nature of discrete-time spectra. Since we cannot require that the discrete-time Fourier transform is zero for $\omega < 0$ without violating its periodicity, we define the analytic signal $\psi(n)$ of a bandpass sequence $x(n)$ by

$$\Psi(\omega) = \begin{cases} 2X(\omega), & 0 \le \omega < \pi \\ 0, & -\pi \le \omega < 0 \end{cases} \tag{6.5.17}$$

where $X(\omega)$ and $\Psi(\omega)$ are the Fourier transforms of $x(n)$ and $\psi(n)$, respectively.

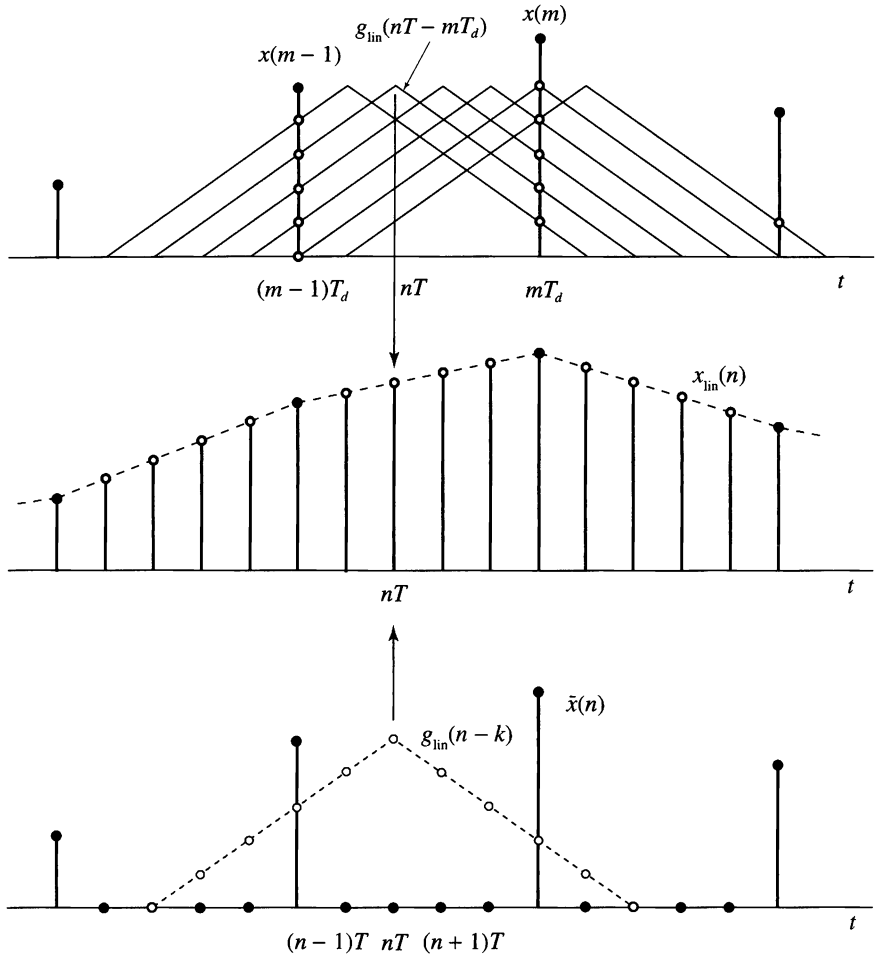


Figure 6.5.4 Illustration of linear interpolation as a linear filtering process.

The ideal discrete-time Hilbert transformer, defined by

$$H(\omega) = \begin{cases} -j, & 0 < \omega < \pi \\ j, & -\pi < \omega < 0 \end{cases} \tag{6.5.18}$$

is a 90-degree phase shifter as in the continuous-time case. We can easily show that

$$\Psi(\omega) = X(\omega) + j\hat{X}(\omega) \tag{6.5.19}$$

where

$$\hat{X}(\omega) = H(\omega)X(\omega) \tag{6.5.20}$$

To compute the analytic signal in the time domain, we need the impulse response of the Hilbert transformer. It is obtained by

$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{0} je^{j\omega n}d\omega - \frac{1}{2\pi}\int_{0}^{\pi} je^{j\omega n}d\omega \tag{6.5.21}$$

which yields

$$h(n) = \begin{cases} \frac{2}{\pi}\frac{\sin^2(\pi n/2)}{n}, & n \neq 0 \\ 0, & n = 0 \end{cases} = \begin{cases} 0, & n = \text{even} \\ \frac{2}{\pi n}, & n = \text{odd} \end{cases} \tag{6.5.22}$$

The sequence $h(n)$ is nonzero for $n < 0$ and not absolutely summable; thus, the ideal Hilbert transformer is noncausal and unstable. The impulse response and the frequency response of the ideal Hilbert transformer are illustrated in Figure 6.5.5.

As in the continuous-time case the Hilbert transform $\hat{x}(n)$ of a sequence $x(n)$ provides the imaginary part of its analytic signal representation, that is,

$$\psi(n) = x(n) + j\hat{x}(n) \tag{6.5.23}$$

The complex envelope, quadrature, and envelope/phase representations are obtained by the corresponding formulas for continuous-time signals by replacing t by nT in all relevant equations.

Given a bandpass sequence $x(n)$, $0 < \omega_L \leq |\omega| \leq \omega_L + w$ with normalized bandwidth $w = 2\pi B/F_s$, we can derive equivalent complex envelope or in-phase and quadrature lowpass representations that can be sampled at a rate $f_s = 1/D$

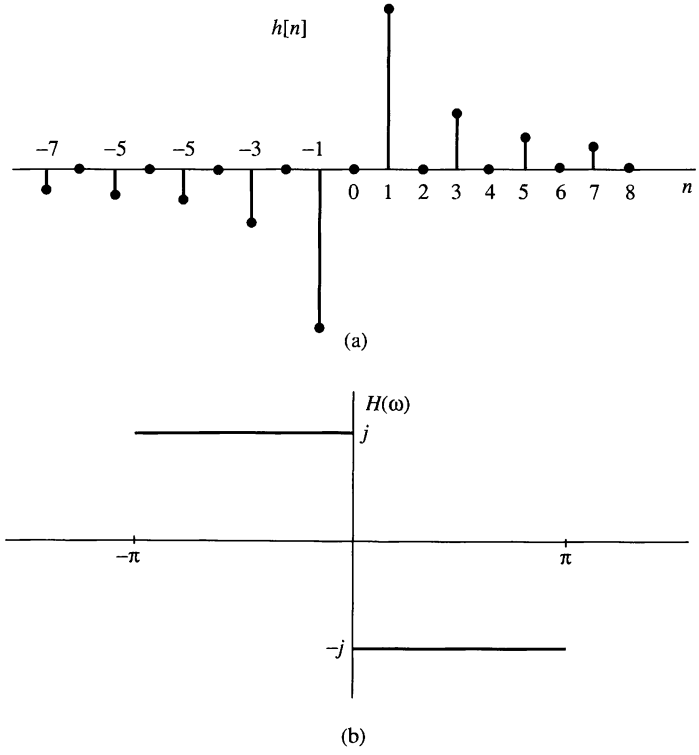


Figure 6.5.5 Impulse response (a) and frequency response (b) of the discrete-time Hilbert transformer.

compatible with the bandwidth w. If $\omega_L = (k-1)\pi/D$ and $w = \pi/D$ the sequence $x(n)$ can be sampled directly without any aliasing as described in Section 11.7.

In many radar and communication systems applications it is necessary to process a bandpass signal $x_a(t)$, $F_L \leq |F| \leq F_L + B$ in lowpass form. Conventional techniques employ two quadrature analog channels and two A/D converters following the two lowpass filters in Figure 6.4.8. A more up-to-date approach is to uniformly sample the analog signal and then obtain the quadrature representation using digital quadrature demodulation, that is, a discrete-time implementation of the first part of Figure 6.4.8. A similar approach can be used to digitally generate single sideband signals for communications applications (Frerking 1994).

Oversampling A/D and D/A Converters

In this section we treat oversampling A/D and D/A converters.

6.6.1 Oversampling A/D Converters

The basic idea in oversampling A/D converters is to increase the sampling rate of the signal to the point where a low-resolution quantizer suffices. By oversampling, we can reduce the dynamic range of the signal values between successive samples and thus reduce the resolution requirements on the quantizer. As we have observed in the preceding section, the variance of the quantization error in A/D conversion is $\sigma_e^2 = \Delta^2/12$, where $\Delta = R/2^{b+1}$. Since the dynamic range of the signal, which is proportional to its standard deviation σ_x, should match the range R of the quantizer, it follows that Δ is proportional to σ_x. Hence for a given number of bits, the power of the quantization noise is proportional to the variance of the signal to be quantized. Consequently, for a given fixed SQNR, a reduction in the variance of the signal to be quantized allows us to reduce the number of bits in the quantizer.

The basic idea for reducing the dynamic range leads us to consider *differential quantization*. To illustrate this point, let us evaluate the variance of the difference between two successive signal samples. Thus we have

$$d(n) = x(n) - x(n-1) \tag{6.6.1}$$

The variance of $d(n)$ is

$$\begin{aligned} \sigma_d^2 &= E[d^2(n)] = E\{[x(n) - x(n-1)]^2\} \\ &= E[x^2(n)] - 2E[x(n)x(n-1)] + E[x^2(n-1)] \\ &= 2\sigma_x^2[1 - \gamma_{xx}(1)] \end{aligned} \tag{6.6.2}$$

where $\gamma_{xx}(1)$ is the value of the autocorrelation sequence $\gamma_{xx}(m)$ of $x(n)$ evaluated at $m = 1$. If $\gamma_{xx}(1) > 0.5$, we observe that $\sigma_d^2 < \sigma_x^2$. Under this condition, it is better to quantize the difference $d(n)$ and to recover $x(n)$ from the quantized values $\{d_q(n)\}$. To obtain a high correlation between successive samples of the signal, we require that the sampling rate be significantly higher than the Nyquist rate.

An even better approach is to quantize the difference

$$d(n) = x(n) - ax(n-1) \tag{6.6.3}$$

where a is a parameter selected to minimize the variance in $d(n)$. This leads to the result (see Problem 6.16) that the optimum choice of a is

$$a = \frac{\gamma_{xx}(1)}{\gamma_{xx}(0)} = \frac{\gamma_{xx}(1)}{\sigma_x^2}$$

and

$$\sigma_d^2 = \sigma_x^2[1 - a^2] \tag{6.6.4}$$

In this case, $\sigma_d^2 < \sigma_x^2$, since $0 \leq a \leq 1$. The quantity $ax(n-1)$ is called a first-order predictor of $x(n)$.

Figure 6.6.1 shows a more general *differential predictive* signal quantizer system. This system is used in speech encoding and transmission over telephone channels and is known as differential pulse code modulation (DPCM). The goal of the predictor is to provide an estimate $\hat{x}(n)$ of $x(n)$ from a linear combination of past values of $x(n)$, so as to reduce the dynamic range of the difference signal $d(n) = x(n) - \hat{x}(n)$. Thus a predictor of order p has the form

$$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{6.6.5}$$

The use of the feedback loop around the quantizer as shown in Fig. 6.6.1 is necessary to avoid the accumulation of quantization errors at the decoder. In this configuration, the error $e(n) = d(n) - d_q(n)$ is

$$e(n) = d(n) - d_q(n) = x(n) - \hat{x}(n) - d_q(n) = x(n) - x_q(n)$$

Thus the error in the reconstructed quantized signal $x_q(n)$ is equal to the quantization error for the sample $d(n)$. The decoder for DPCM that reconstructs the signal from the quantized values is also shown in Fig. 6.6.1.

The simplest form of differential predictive quantization is called *delta modulation* (DM). In DM, the quantizer is a simple 1-bit (two-level) quantizer and the

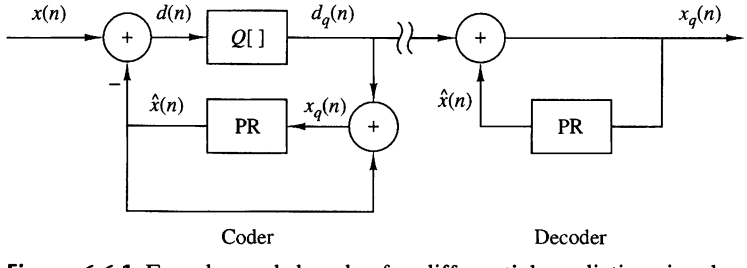


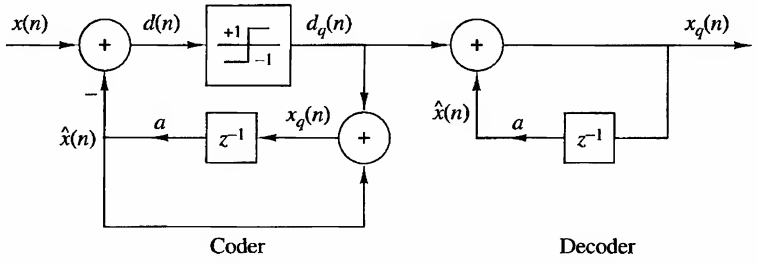
Figure 6.6.1 Encoder and decoder for differential predictive signal quantizer system.

predictor is a first-order predictor, as shown in Fig. 6.6.2(a). Basically, DM provides a staircase approximation of the input signal. At every sampling instant, the sign of the difference between the input sample $x(n)$ and its most recent staircase approximation $\hat{x}(n) = ax_q(n-1)$ is determined, and then the staircase signal is updated by a step Δ in the direction of the difference.

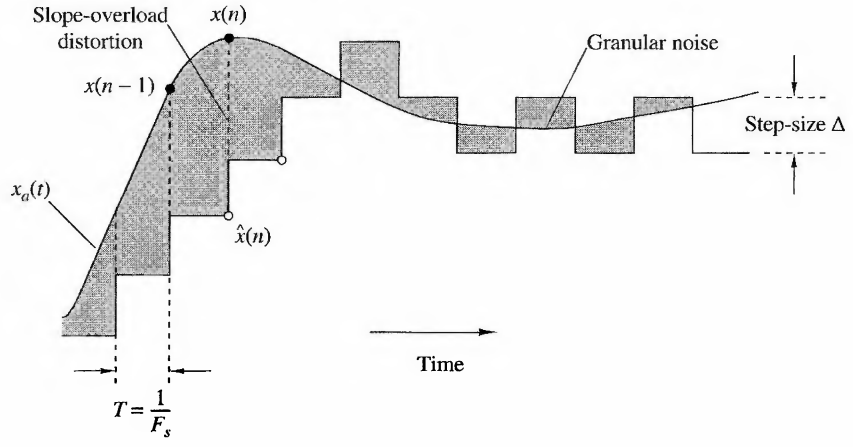
From Fig. 6.6.2(a) we observe that

$$x_q(n) = ax_q(n-1) + d_q(n) \tag{6.6.6}$$

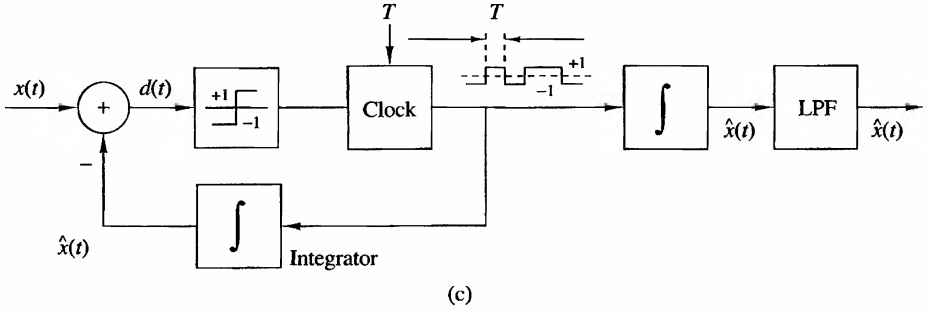
which is the discrete-time equivalent of an analog integrator. If $a = 1$, we have an ideal accumulator (integrator) whereas the choice $a < 1$ results in a "leaky integra-



(a)



(b)



(c)

Figure 6.6.2 Delta modulation system and two types of quantization errors.

tor." Figure 6.6.2(c) shows an analog model that illustrates the basic principle for the practical implementation of a DM system. The analog lowpass filter is necessary for the rejection of out-of-band components in the frequency range between B and $F_s/2$, since $F_s >> B$ due to oversampling.

The crosshatched areas in Fig. 6.6.2(b) illustrate two types of quantization error in DM, slope-overload distortion and granular noise. Since the maximum slope Δ/T in $x(n)$ is limited by the step size, slope-overload distortion can be avoided if $\max |dx(t)/dt| \leq \Delta/T$. The granular noise occurs when the DM tracks a relatively flat (slowly changing) input signal. We note that increasing Δ reduces overload distortion but increases the granular noise, and vice versa.

One way to reduce these two types of distortion is to use an integrator in front of the DM, as shown in Fig. 6.6.3(a). This has two effects. First, it emphasizes the low frequencies of $x(t)$ and increases the correlation of the signal into the DM input. Second, it simplifies the DM decoder because the differentiator (inverse system) required at the decoder is canceled by the DM integrator. Hence the decoder is simply a lowpass filter, as shown in Fig. 6.6.3(a). Furthermore, the two integrators at the encoder can be replaced by a single integrator placed before the comparator, as shown in Fig. 6.6.3(b). This system is known as *sigma-delta modulation* (SDM).

SDM is an ideal candidate for A/D conversion. Such a converter takes advantage of the high sampling rate and spreads the quantization noise across the band up to $F_s/2$. Since $F_s >> B$, the noise in the signal-free band $B \leq F \leq F_s/2$ can be

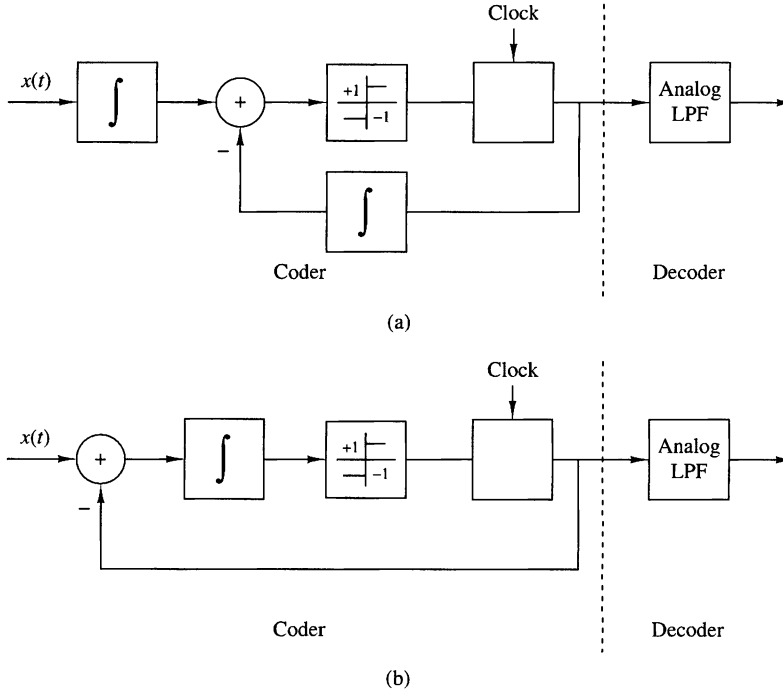


Figure 6.6.3 Sigma-delta modulation system.

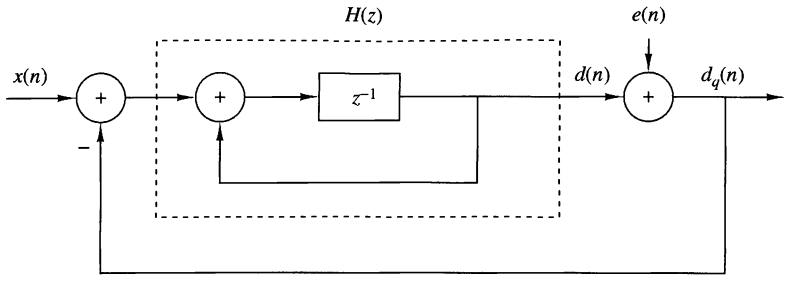


Figure 6.6.4 Discrete-time model of sigma-delta modulation.

removed by appropriate digital filtering. To illustrate this principle, let us consider the discrete-time model of SDM, shown in Fig. 6.6.4, where we have assumed that the comparator (1-bit quantizer) is modeled by an additive white noise source with variance $\sigma_e^2 = \Delta^2/12$. The integrator is modeled by the discrete-time system with system function

$$H(z) = \frac{z^{-1}}{1 - z^{-1}} \tag{6.6.7}$$

The z-transform of the sequence $\{d_q(n)\}$ is

$$\begin{aligned} D_q(z) &= \frac{H(z)}{1 + H(z)} X(z) + \frac{1}{1 + H(z)} E(z) \\ &= H_s(z)X(z) + H_n(z)E(z) \end{aligned} \tag{6.6.8}$$

where $H_s(z)$ and $H_n(z)$ are the signal and noise system functions, respectively. A good SDM system has a flat frequency response $H_s(\omega)$ in the signal frequency band $0 \le F \le B$. On the other hand, $H_n(z)$ should have high attenuation in the frequency band $0 \le F \le B$ and low attenuation in the band $B \le F \le F_s/2$.

For the first-order SDM system with the integrator specified by (6.6.7), we have

$$H_s(z) = z^{-1}, \qquad H_n(z) = 1 - z^{-1} \tag{6.6.9}$$

Thus $H_s(z)$ does not distort the signal. The performance of the SDM system is therefore determined by the noise system function $H_n(z)$, which has a magnitude frequency response

$$|H_n(F)| = 2\left|\sin\frac{\pi F}{F_s}\right| \tag{6.6.10}$$

as shown in Fig. 6.6.5. The in-band quantization noise variance is given as

$$\sigma_n^2 = \int_{-B}^{B} |H_n(F)|^2 S_e(F)\, dF \tag{6.6.11}$$

where $S_e(F) = \sigma_e^2/F_s$ is the power spectral density of the quantization noise. From this relationship we note that doubling F_s (increasing the sampling rate by a factor of 2), while keeping B fixed, reduces the power of the quantization noise by 3 dB. This result is true for any quantizer. However, additional reduction may be possible by properly choosing the filter $H(z)$.

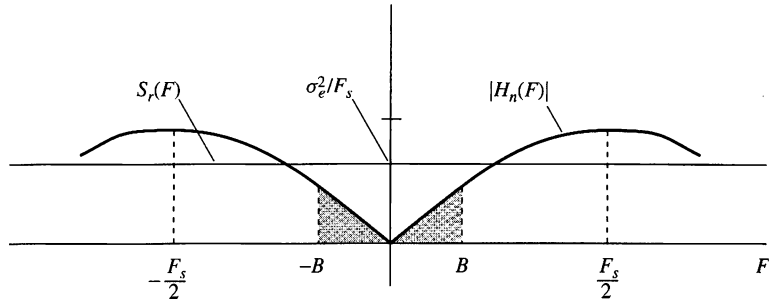


Figure 6.6.5 Frequency (magnitude) response of noise system function.

For the first-order SDM, it can be shown (see Problem 6.19) that for $F_s >> 2B$, the in-band quantization noise power is

$$\sigma_n^2 \approx \frac{1}{3}\pi^2\sigma_e^2\left(\frac{2B}{F_s}\right)^3 \tag{6.6.12}$$

Note that doubling the sampling frequency reduces the noise power by 9 dB, of which 3 dB is due to the reduction in $S_e(F)$ and 6 dB is due to the filter characteristic $H_n(F)$. An additional 6-dB reduction can be achieved by using a double integrator (see Problem 6.20).

In summary, the noise power σ_n^2 can be reduced by increasing the sampling rate to spread the quantization noise power over a larger frequency band $(-F_s/2, F_s/2)$, and then shaping the noise power spectral density by means of an appropriate filter. Thus, SDM provides a 1-bit quantized signal at a sampling frequency $F_s = 2IB$, where the oversampling (interpolation) factor I determines the SNR of the SDM quantizer.

Next, we explain how to convert this signal into a b-bit quantized signal at the Nyquist rate. First, we recall that the SDM decoder is an analog lowpass filter with a cutoff frequency B. The output of this filter is an approximation to the input signal $x(t)$. Given the 1-bit signal $d_q(n)$ at sampling frequency F_s, we can obtain a signal $x_q(n)$ at a lower sampling frequency, say the Nyquist rate of $2B$ or somewhat faster, by resampling the output of the lowpass filter at the $2B$ rate. To avoid aliasing, we first filter out the out-of-band $(B, F_s/2)$ noise by processing the wideband signal. The signal is then passed through the lowpass filter and resampled (down-sampled) at the lower rate. The down-sampling process is called *decimation* and is treated in great detail in Chapter 11.

For example, if the interpolation factor is $I = 256$, the A/D converter output can be obtained by averaging successive nonoverlapping blocks of 128 bits. This averaging would result in a digital signal with a range of values from zero to 256($b \approx 8$ bits) at the Nyquist rate. The averaging process also provides the required antialiasing filtering.

Figure 6.6.6 illustrates the basic elements of an oversampling A/D converter. Oversampling A/D converters for voiceband (3-kHz) signals are currently fabricated

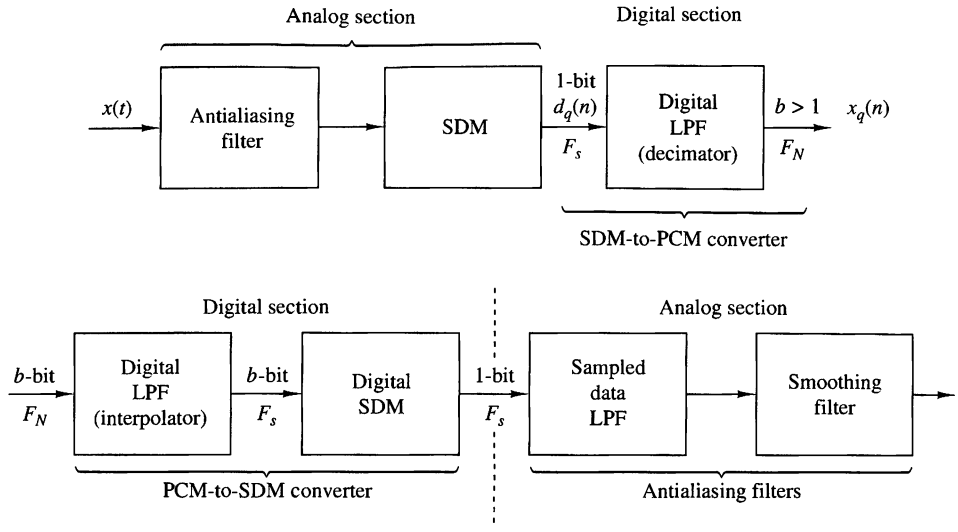


Figure 6.6.6 Basic elements of an oversampling A/D converter.

as integrated circuits. Typically, they operate at a 2-MHz sampling rate, down-sample to 8 kHz, and provide 16-bit accuracy.

6.6.2 Oversampling D/A Converters

The elements of an oversampling D/A converter are shown in Fig. 6.6.7. As we observe, it is subdivided into a digital front end followed by an analog section. The digital section consists of an interpolator whose function is to increase the sampling rate by some factor I, which is followed by an SDM. The interpolator simply increases the digital sampling rate by inserting $I-1$ zeros between successive low rate samples. The resulting signal is then processed by a digital filter with cutoff frequency $F_c = B/F_s$ in order to reject the images (replicas) of the input signal spectrum. This higher rate signal is fed to the SDM, which creates a noise-shaped 1-bit sample. Each 1-bit sample is fed to the 1-bit D/A, which provides the analog interface to the antialiasing and smoothing filters. The output analog filters have a passband of $0 \leq F \leq B$ hertz and serve to smooth the signal and to remove the quantization noise in the frequency band $B \leq F \leq F_s/2$. In effect, the oversampling D/A converter uses SDM with the roles of the analog and digital sections reversed compared to the A/D converter.

In practice, oversampling D/A (and A/D) converters have many advantages over the more conventional D/A (and A/D) converters. First, the high sampling rate and

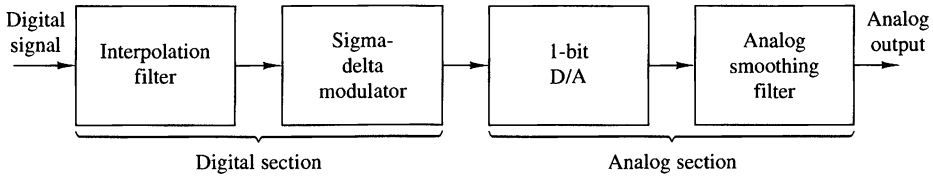


Figure 6.6.7 Elements of an oversampling D/A converter.

the subsequent digital filtering minimize or remove the need for complex and expensive analog antialiasing filters. Furthermore, any analog noise introduced during the conversion phase is filtered out. Also, there is no need for S/H circuits. Oversampling SDM A/D and D/A converters are very robust with respect to variations in the analog circuit parameters, are inherently linear, and have low cost.

This concludes our discussion of signal reconstruction based on simple interpolation techniques. The techniques that we have described are easily incorporated into the design of practical D/A converters for the reconstruction of analog signals from digital signals. We shall consider interpolation again in Chapter 11 in the context of changing the sampling rate in a digital signal processing system.

6.7 Summary and References

The major focus of this chapter was on the sampling and reconstruction of signals. In particular, we treated the sampling of continuous-time signals and the subsequent operation of A/D conversion. These are necessary operations in the digital processing of analog signals, either on a general-purpose computer or on a custom-designed digital signal processor. The related issue of D/A conversion was also treated. In addition to the conventional A/D and D/A conversion techniques, we also described another type of A/D and D/A conversion, based on the principle of oversampling and a type of waveform encoding called sigma-delta modulation. Sigma-delta conversion technology is especially suitable for audio band signals due to their relatively small bandwidth (less than 20 kHz) and in some applications, the requirements for high fidelity.

The sampling theorem was introduced by Nyquist (1928) and later popularized in the classic paper by Shannon (1949). D/A and A/D conversion techniques are treated in a book by Sheingold (1986). Oversampling A/D and D/A conversion has been treated in the technical literature. Specifically, we cite the work of Candy (1986), Candy et al. (1981) and Gray (1990).

Problems

6.1 Given a continuous-time signal $x_a(t)$ with $X_a(F) = 0$ for $|F| > B$ determine the minimum sampling rate F_s for a signal $y_a(t)$ defined by **(a)** $dx_a(t)/dt$ **(b)** $x_a^2(t)$ **(c)** $x_a(2t)$ **(d)** $x_a(t)\cos 6\pi Bt$ and **(e)** $x_a(t)\cos 7\pi Bt$

6.2 The sampled sequence $x_a(nT)$ is reconstructed using an ideal D/A with interpolation function $g_a(t) = A$ for $|F| < F_c$ and zero otherwise to produce a continuous-time signal $\hat{x}_a(t)$.

(a) If the spectrum of the original signal $x_a(t)$ satisfies $X_a(F) = 0$ for $|F| > B$, find the maximum value of T, and the values of F_c, and A such that $\hat{x}_a(t) = x_a(t)$.

(b) If $X_1(F) = 0$ for $|F| > B$, $X_2(F) = 0$ for $|F| > 2B$, and $x_a(t) = x_1(t)x_2(t)$, find the maximum value of T, and the values of F_c, and A such that $\hat{x}_a(t) = x_a(t)$.

(c) Repeat part (b) for $x_a(t) = x_1(t)x_2(t/2)$.

6.3 A continuous-time periodic signal with Fourier series coefficients $c_k = (1/2)^{|k|}$ and period $T_p = 0.1$ sec passes through an ideal lowpass filter with cutoff frequency $F_c = 102.5$ Hz. The resulting signal $y_a(t)$ is sampled periodically with $T = 0.005$ sec. Determine the spectrum of the sequence $y(n) = y_a(nT)$.

6.4 Repeat Example 6.1.2 for the signal $x_a(t) = te^{-t}u_a(t)$.

6.5 Consider the system in Figure 6.2.1. If $X_a(F) = 0$ for $|F| > F_s/2$, determine the frequency response $H(\omega)$ of the discrete-time system such that $y_a(t) = \int_{-\infty}^{t} x_a(\tau)d\tau$.

6.6 Consider a signal $x_a(t)$ with spectrum $X_a(F) \neq 0$ for $0 < F_1 \leq |F| \leq F_2 < \infty$ and $X_a(F) = 0$ otherwise.

(a) Determine the minimum sampling frequency required to sample $x_a(t)$ without aliasing.

(b) Find the formula needed to reconstruct $x_a(t)$ from the samples $x_a(nT)$, $-\infty < n < \infty$.

6.7 Prove the nonuniform second-order sampling interpolation formula described by equations (6.4.47)–(6.4.49).

6.8 A discrete-time sample-and-hold interpolator, by a factor I, repeats the last input sample $(I - 1)$ times.

(a) Determine the interpolation function $g_{SH}(n)$.

(b) Determine the Fourier transform $G_{SH}(\omega)$ of $g_{SH}(n)$.

(c) Plot the magnitude and phase responses of the ideal interpolator, the linear interpolator, and the sample-and-hold interpolator for $I = 5$.

6.9 *Time-domain sampling* Consider the continuous-time signal

$$x_a(t) = \begin{cases} e^{-j2\pi F_0 t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

(a) Compute analytically the spectrum $X_a(F)$ of $x_a(t)$.

(b) Compute analytically the spectrum of the signal $x(n) = x_a(nT)$, $T = 1/F_s$.

(c) Plot the magnitude spectrum $|X_a(F)|$ for $F_0 = 10$ Hz.

(d) Plot the magnitude spectrum $|X(F)|$ for $F_s = 10, 20, 40$, and 100 Hz.

(e) Explain the results obtained in part (d) in terms of the aliasing effect.

6.10 Consider the sampling of the bandpass signal whose spectrum is illustrated in Fig. P6.10. Determine the minimum sampling rate F_s to avoid aliasing.

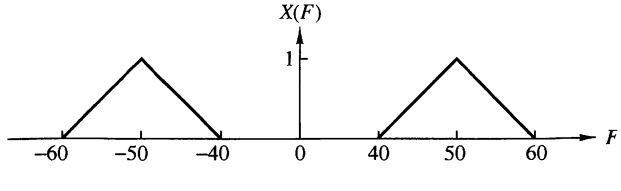


Figure P6.10

6.11 Consider the sampling of the bandpass signal whose spectrum is illustrated in Fig. P6.11. Determine the minimum sampling rate F_s to avoid aliasing.

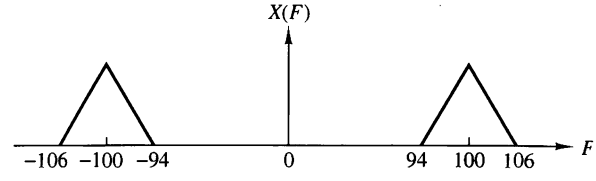


Figure P6.11

6.12 Consider the two systems shown in Fig. P6.12.

(a) Sketch the spectra of the various signals if $x_a(t)$ has the Fourier transform shown in Fig. P6.12(b) and $F_s = 2B$. How are $y_1(t)$ and $y_2(t)$ related to $x_a(t)$?

(b) Determine $y_1(t)$ and $y_2(t)$ if $x_a(t) = \cos 2\pi F_0 t$, $F_0 = 20$ Hz, and $F_s = 50$ Hz or $F_s = 30$ Hz.

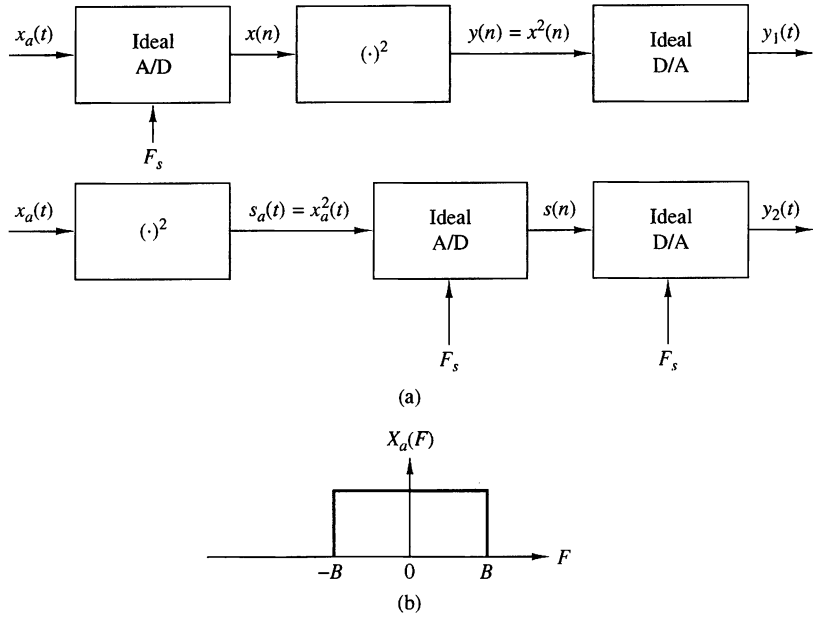


Figure P6.12

6.13 A continuous-time signal $x_a(t)$ with bandwidth B and its echo $x_a(t - \tau)$ arrive simultaneously at a TV receiver. The received analog signal

$$s_a(t) = x_a(t) + \alpha x_a(t - \tau), \qquad |\alpha| < 1$$

is processed by the system shown in Fig. P6.13. Is it possible to specify F_s and $H(z)$ so that $y_a(t) = x_a(t)$ [i.e., remove the "ghost" $x_a(t - \tau)$ from the received signal]?

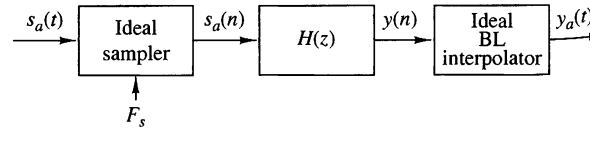


Figure P6.13

6.14 A bandlimited continuous-time signal $x_a(t)$ is sampled at a sampling frequency $F_s \geq 2B$. Determine the energy E_d of the resulting discrete-time signal $x(n)$ as a function of the energy of the analog signal, E_a, and the sampling period $T = 1/F_s$.

6.15 In a linear interpolator successive sample points are connected by straight-line segments. Thus the resulting interpolated signal $\hat{x}(t)$ can be repressed as

$$\hat{x}(t) = x(nT - T) + \frac{x(nT) - x(nT - T)}{T}(t - nT), \quad nT \leq t \leq (n + 1)T$$

We observe that at $t = nT$, $\hat{x}(nT) = x(nT-T)$ and at $t = nT+T$, $\hat{x}(nT+T) = x(nT)$. Therefore, $x(t)$ has an inherent delay of T seconds in interpolating the actual signal $x(t)$. Figure P6.15 illustrates this linear interpolation technique.

(a) Viewed as a linear filter, show that the linear interpolation with a T-second delay has an impulse response

$$h(t) = \begin{cases} t/T, & 0 \leq t < T \\ 2 - t/T, & T \leq t < 2T \\ 0, & \text{otherwise} \end{cases}$$

Derive the corresponding frequency response $H(F)$.

(b) Plot $|H(F)|$ and compare this frequency response with that of the ideal reconstruction filter for a lowpass bandlimited signal.

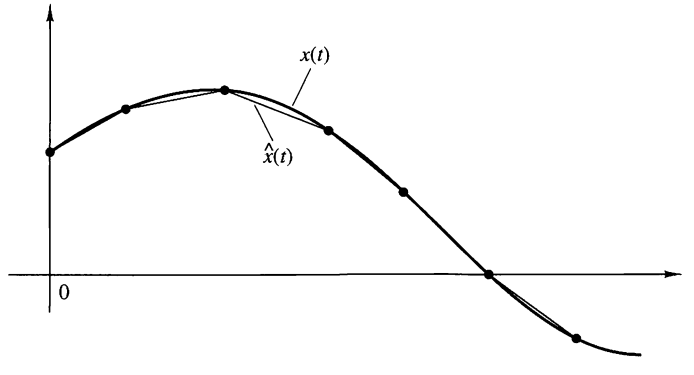


Figure P6.15

6.16 Let $x(n)$ be a zero-mean stationary process with variance σ_x^2 and autocorrelation $\gamma_x(l)$.

(a) Show that the variance σ_d^2 of the first-order prediction error

$$d(n) = x(n) - ax(n-1)$$

is given by

$$\sigma_d^2 = \sigma_x^2[1 + a^2 - 2a\rho_x(1)]$$

where $\rho_x(1) = \gamma_x(1)/\gamma_x(0)$ is the normalized autocorrelation sequence.

(b) Show that σ_d^2 attains its minimum value

$$\sigma_d^2 = \sigma_x^2[1 - \rho_x^2(1)]$$

for $a = \gamma_x(1)/\gamma_x(0) = \rho_x(1)$.

(c) Under what conditions is $\sigma_d^2 < \sigma_x^2$?

(d) Repeat steps (a) to (c) for the second-order prediction error

$$d(n) = x(n) - a_1x(n-1) - a_2x(n-2)$$

6.17 Consider a DM coder with input $x(n) = A\cos(2\pi nF/F_s)$. What is the condition for avoiding slope overload? Illustrate this condition graphically.

6.18 Let $x_a(t)$ be a bandlimited signal with fixed bandwidth B and variance σ_x^2.

(a) Show that the signal-to-quantization noise ratio, SQNR $= 10\log_{10}(\sigma_x^2/\sigma_e^2)$, increases by 3 dB each time we double the sampling frequency F_s. Assume that the quantization noise model discussed in Section 6.3.3 is valid.

(b) If we wish to increase the SQNR of a quantizer by doubling its sampling frequency, what is the most efficient way to do it? Should we choose a linear multibit A/D converter or an oversampling one?

6.19 Consider the first-order SDM model shown in Fig. 6.6.4.

(a) Show that the quantization noise power in the signal band $(-B, B)$ is given by

$$\sigma_n^2 = \frac{2\sigma_e^2}{\pi}\left[\frac{2\pi B}{Fs} - \sin\left(2\pi\frac{B}{F_s}\right)\right]$$

(b) Using a two-term Taylor series expansion of the sine function and assuming that $F_s >> B$, show that

$$\sigma_n^2 \approx \frac{1}{3}\pi 2\sigma_e^2\left(\frac{2B}{F_s}\right)^3$$

6.20 Consider the second-order SDM model shown in Fig. P6.20.

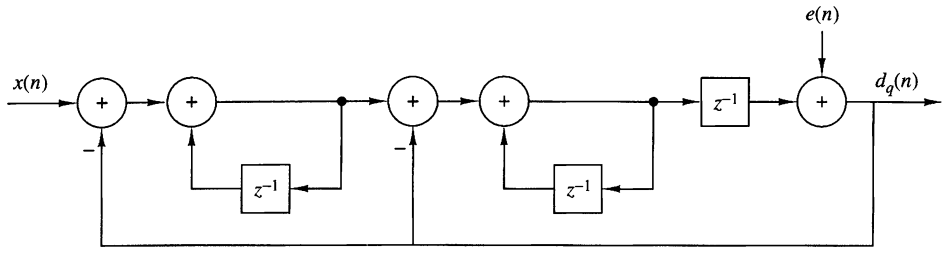


Figure P6.20

(a) Determine the signal and noise system functions $H_s(z)$ and $H_n(z)$, respectively.

(b) Plot the magnitude response for the noise system function and compare it with the one for the first-order SDM. Can you explain the 6-dB difference from these curves?

(c) Show that the in-band quantization noise power σ_n^2 is given approximately by

$$\sigma_n^2 \approx \frac{\pi\sigma_e^2}{5}\left(\frac{2B}{F_s}\right)^5$$

which implies a 15-dB increase for every doubling of the sampling frequency.

6.21 Figure P6.21 illustrates the basic idea for a lookup-table-based sinusoidal signal generator. The samples of one period of the signal

$$x(n) = \cos\left(\frac{2\pi}{N}n\right), \qquad n = 0, 1, \ldots, N-1$$

are stored in memory. A digital sinusoidal signal is generated by stepping through the table and wrapping around at the end when the angle exceeds 2π. This can be done by using modulo-N addressing (i.e., using a "circular" buffer). Samples of $x(n)$ are feeding the ideal D/A converter every T seconds.

(a) Show that by changing F_s we can adjust the frequency F_0 of the resulting analog sinusoid.

(b) Suppose now that $F_s = 1/T$ is fixed. How many distinct analog sinusoids can be generated using the given lookup table? Explain.

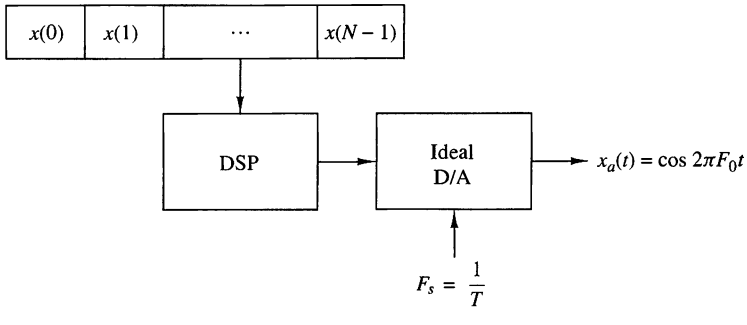


Figure P6.21

6.22 Suppose that we represent an analog bandpass filter by the frequency response

$$H(F) = C(F - F_c) + C^*(-F - F_c)$$

where $C(f)$ is the frequency response of an equivalent lowpass filter, as shown in Fig. P6.22.

(a) Show that the impulse response $c(t)$ of the equivalent lowpass filter is related to the impulse response $h(t)$ of the bandpass filter as follows:

$$h(t) = 2\Re[c(t)e^{j2\pi F_c t}]$$

(b) Suppose that the bandpass system with frequency response $H(F)$ is excited by a bandpass signal of the form

$$x(t) = \Re[u(t)e^{j2\pi F_c t}]$$

where $u(t)$ is the equivalent lowpass signal. Show that the filter output may be expressed as

$$y(t) = \Re[v(t)e^{j2\pi F_c t}]$$

where

$$v(t) = \int^{\omega} c(\tau)u(t - \tau)d\tau$$

(*Hint:* Use the frequency domain to prove this result.)

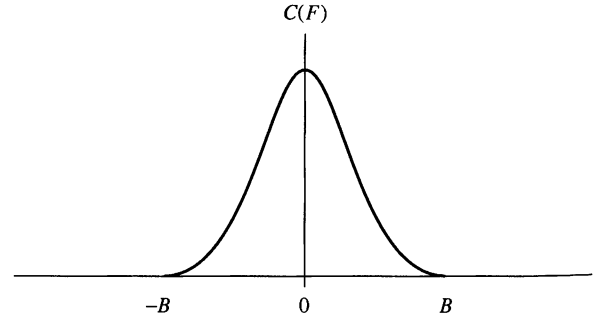


Figure P6.22

6.23 Consider the sinusoidal signal generator in Fig. P6.23, where both the stored sinusoidal data

$$x(n) = \cos\left(\frac{2A}{N}n\right), \qquad 0 \le n \le N - 1$$

and the sampling frequency $F_s = 1/T$ are fixed. An engineer wishing to produce a sinusoid with period $2N$ suggests that we use either zero-order or first-order (linear) interpolation to double the number of samples per period in the original sinusoid as illustrated in Fig. P6.23(a).

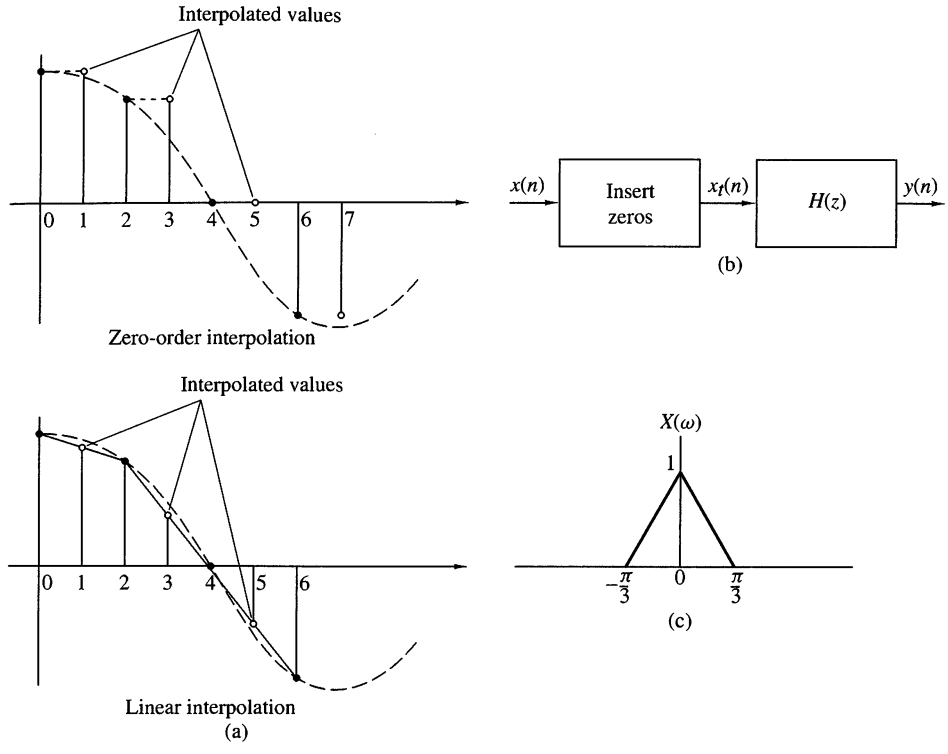


Figure P6.23

(a) Determine the signal sequences $y(n)$ generated using zero-order interpolation and linear interpolation and then compute the total harmonic distortion (THD) in each case for $N = 32, 64, 128$.

(b) Repeat part (a) assuming that all sample values are quantized to 8 bits.

(c) Show that the interpolated signal sequences $y(n)$ can be obtained by the system shown in Fig. P6.23(b). The first module inserts one zero sample between successive samples of $x(n)$. Determine the system $H(z)$ and sketch its magnitude response for the zero-order interpolation and for the linear interpolation cases. Can you explain the difference in performance in terms of the frequency response functions?

(d) Determine and sketch the spectra of the resulting sinusoids in each case both analytically [using the results in part (c)] and evaluating the DFT of the resulting signals.

(e) Sketch the spectra of $x_i(n)$ and $y(n)$, if $x(n)$ has the spectrum shown in Fig. P6.23(c) for both zero-order and linear interpolation. Can you suggest a better choice for $H(z)$?

6.24 Let $x_a(t)$ be a time-limited signal; that is, $x_a(t) = 0$ for $|t| > \tau$, with Fourier transform $X_a(F)$. The function $X_a(F)$ is sampled with sampling interval $\delta F = 1/T_s$.

(a) Show that the function

$$x_p(t) = \sum_{n=-\infty}^{\infty} x_a(t - nT_s)$$

can be expressed as a Fourier series with coefficients

$$c_k = \frac{1}{T_s} X_a(k\delta F)$$

(b) Show that $X_a(F)$ can be recovered from the samples $X_a(k\delta F)$, $-\infty < k < \infty$ if $T_s \geq 2\tau$.

(c) Show that if $T_s < 2\tau$, there is "time-domain aliasing" that prevents exact reconstruction of $X_a(F)$.

(d) Show that if $T_s \geq 2\tau$, perfect reconstruction of $X_a(F)$ from the samples $X(k\delta F)$ is possible using the interpolation formula

$$X_a(F) = \sum_{k=-\infty}^{\infty} X_a(k\delta F) \frac{\sin[(\pi/\delta F)(F - k\delta F)]}{(\pi/\delta F)(F - k\delta F)}$$

CHAPTER

7

The Discrete Fourier Transform: Its Properties and Applications

Frequency analysis of discrete-time signals is usually and most conveniently performed on a digital signal processor, which may be a general-purpose digital computer or specially designed digital hardware. To perform frequency analysis on a discrete-time signal $\{x(n)\}$, we convert the time-domain sequence to an equivalent frequency-domain representation. We know that such a representation is given by the Fourier transform $X(\omega)$ of the sequence $\{x(n)\}$. However, $X(\omega)$ is a continuous function of frequency and therefore it is not a computationally convenient representation of the sequence $\{x(n)\}$.

In this chapter we consider the representation of a sequence $\{x(n)\}$ by samples of its spectrum $X(\omega)$. Such a frequency-domain representation leads to the discrete Fourier transform (DFT), which is a powerful computational tool for performing frequency analysis of discrete-time signals.

Frequency-Domain Sampling: The Discrete Fourier Transform

Before we introduce the DFT, we consider the sampling of the Fourier transform of an aperiodic discrete-time sequence. Thus, we establish the relationship between the sampled Fourier transform and the DFT.

7.1.1 Frequency-Domain Sampling and Reconstruction of Discrete-Time Signals

We recall that aperiodic finite-energy signals have continuous spectra. Let us consider such an aperiodic discrete-time signal $x(n)$ with Fourier transform

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} \tag{7.1.1}$$

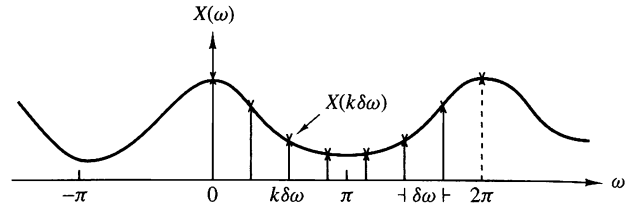


Figure 7.1.1
Frequency-domain sampling of the Fourier transform.

Suppose that we sample $X(\omega)$ periodically in frequency at a spacing of $\delta\omega$ radians between successive samples. Since $X(\omega)$ is periodic with period 2π, only samples in the fundamental frequency range are necessary. For convenience, we take N equidistant samples in the interval $0 \leq \omega < 2\pi$ with spacing $\delta\omega = 2\pi/N$, as shown in Fig. 7.1.1. First, we consider the selection of N, the number of samples in the frequency domain.

If we evaluate (7.1.1) at $\omega = 2\pi k/N$, we obtain

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1 \tag{7.1.2}$$

The summation in (7.1.2) can be subdivided into an infinite number of summations, where each sum contains N terms. Thus

$$\begin{aligned} X\left(\frac{2\pi}{N}k\right) = & \cdots + \sum_{n=-N}^{-1} x(n)e^{-j2\pi kn/N} + \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \\ & + \sum_{n=N}^{2N-1} x(n)e^{-j2\pi kn/N} + \cdots \\ = & \sum_{l=-\infty}^{\infty} \sum_{n=lN}^{lN+N-1} x(n)e^{-j2\pi kn/N} \end{aligned}$$

If we change the index in the inner summation from n to $n - lN$ and interchange the order of the summation, we obtain the result

$$X\left(\frac{2\pi}{N}k\right) = \sum_{n=0}^{N-1} \left[\sum_{l=-\infty}^{\infty} x(n - lN)\right] e^{-j2\pi kn/N} \tag{7.1.3}$$

for $k = 0, 1, 2, \ldots, N-1$.

The signal

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \tag{7.1.4}$$

obtained by the periodic repetition of $x(n)$ every N samples, is clearly periodic with fundamental period N. Consequently, it can be expanded in a Fourier series as

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi kn/N}, \qquad n = 0, 1, \ldots, N-1 \tag{7.1.5}$$

with Fourier coefficients

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j2\pi kn/N}, \qquad k = 0, 1, \ldots, N-1 \tag{7.1.6}$$

Upon comparing (7.1.3) with (7.1.6), we conclude that

$$c_k = \frac{1}{N} X\left(\frac{2\pi}{N}k\right), \qquad k = 0, 1, \ldots, N-1 \tag{7.1.7}$$

Therefore,

$$x_p(n) = \frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N}, \qquad n = 0, 1, \ldots, N-1 \tag{7.1.8}$$

The relationship in (7.1.8) provides the reconstruction of the periodic signal $x_p(n)$ from the samples of the spectrum $X(\omega)$. However, it does not imply that we can recover $X(\omega)$ or $x(n)$ from the samples. To accomplish this, we need to consider the relationship between $x_p(n)$ and $x(n)$.

Since $x_p(n)$ is the periodic extension of $x(n)$ as given by (7.1.4), it is clear that $x(n)$ can be recovered from $x_p(n)$ if there is no aliasing in the time domain, that is, if $x(n)$ is time-limited to less than the period N of $x_p(n)$. This situation is illustrated in Fig. 7.1.2, where without loss of generality, we consider a finite-duration sequence $x(n)$, which is nonzero in the interval $0 \le n \le L-1$. We observe that when $N \ge L$,

$$x(n) = x_p(n), \qquad 0 \le n \le N-1$$

so that $x(n)$ can be recovered from $x_p(n)$ without ambiguity. On the other hand, if $N < L$, it is not possible to recover $x(n)$ from its periodic extension due to *time-domain aliasing*. Thus, we conclude that the spectrum of an aperiodic discrete-time signal with finite duration L can be exactly recovered from its samples at frequencies $\omega_k = 2\pi k/N$, if $N \ge L$. The procedure is to compute $x_p(n)$, $n = 0, 1, \ldots, N-1$ from (7.1.8); then

$$x(n) = \begin{cases} x_p(n), & 0 \le n \le N-1 \\ 0, & \text{elsewhere} \end{cases} \tag{7.1.9}$$

and finally, $X(\omega)$ can be computed from (7.1.1).

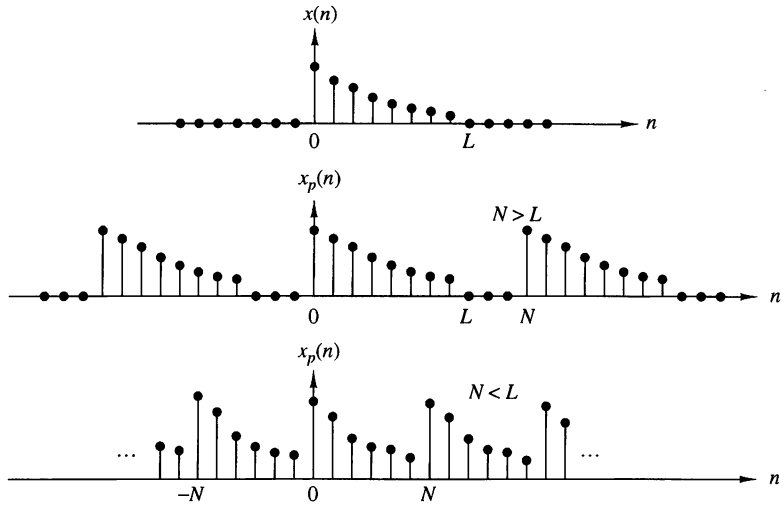


Figure 7.1.2 Aperiodic sequence $x(n)$ of length L and its periodic extension for $N \geq L$ (no aliasing) and $N < L$ (aliasing).

As in the case of continuous-time signals, it is possible to express the spectrum $X(\omega)$ directly in terms of its samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$. To derive such an interpolation formula for $X(\omega)$, we assume that $N \geq L$ and begin with (7.1.8). Since $x(n) = x_p(n)$ for $0 \leq n \leq N-1$,

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N}, \qquad 0 \leq n \leq N-1 \tag{7.1.10}$$

If we use (7.1.1) and substitute for $x(n)$, we obtain

$$\begin{aligned} X(\omega) &= \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) e^{j2\pi kn/N} \right] e^{-j\omega n} \\ &= \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) \left[\frac{1}{N} \sum_{n=0}^{N-1} e^{-j(\omega - 2\pi k/N)n} \right] \end{aligned} \tag{7.1.11}$$

The inner summation term in the brackets of (7.1.11) represents the basic interpolation function shifted by $2\pi k/N$ in frequency. Indeed, if we define

$$\begin{aligned} P(\omega) &= \frac{1}{N} \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1}{N} \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}} \\ &= \frac{\sin(\omega N/2)}{N \sin(\omega/2)} e^{-j\omega(N-1)/2} \end{aligned} \tag{7.1.12}$$

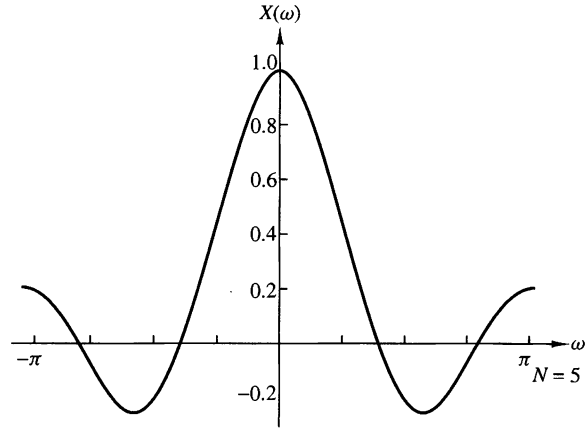


Figure 7.1.3
Plot of the function $[\sin(\omega N/2)]/[N \sin(\omega/2)]$.

then (7.1.11) can be expressed as

$$X(\omega) = \sum_{k=0}^{N-1} X\left(\frac{2\pi}{N}k\right) P\left(\omega - \frac{2\pi}{N}k\right), \qquad N \geq L \tag{7.1.13}$$

The interpolation function $P(\omega)$ is not the familiar $(\sin\theta)/\theta$ but instead, it is a periodic counterpart of it, and it is due to the periodic nature of $X(\omega)$. The phase shift in (7.1.12) reflects the fact that the signal $x(n)$ is a causal, finite-duration sequence of length N. The function $\sin(\omega N/2)/(N \sin(\omega/2))$ is plotted in Fig. 7.1.3 for $N = 5$. We observe that the function $P(\omega)$ has the property

$$P\left(\frac{2\pi}{N}k\right) = \begin{cases} 1, & k = 0 \\ 0, & k = 1, 2, \ldots, N-1 \end{cases} \tag{7.1.14}$$

Consequently, the interpolation formula in (7.1.13) gives exactly the sample values $X(2\pi k/N)$ for $\omega = 2\pi k/N$. At all other frequencies, the formula provides a properly weighted linear combination of the original spectral samples.

The following example illustrates the frequency-domain sampling of a discrete-time signal and the time-domain aliasing that results.

EXAMPLE 7.1.1

Consider the signal

$$x(n) = a^n u(n), \qquad 0 < a < 1$$

The spectrum of this signal is sampled at frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$. Determine the reconstructed spectra for $a = 0.8$ when $N = 5$ and $N = 50$.

Solution. The Fourier transform of the sequence $x(n)$ is

$$X(\omega) = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \frac{1}{1 - ae^{-j\omega}}$$

Suppose that we sample $X(\omega)$ at N equidistant frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$. Thus we obtain the spectral samples

$$X(\omega k) \equiv X\left(\frac{2\pi k}{N}\right) = \frac{1}{1 - ae^{-j2\pi k/N}}, \qquad k = 0, 1, \ldots, N-1$$

The periodic sequence $x_p(n)$, corresponding to the frequency samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$, can be obtained from either (7.1.4) or (7.1.8). Hence

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) = \sum_{l=-\infty}^{0} a^{n-lN}$$

$$= a^n \sum_{l=0}^{\infty} a^{lN} = \frac{a^n}{1 - a^N}, \qquad 0 \le n \le N-1$$

where the factor $1/(1 - a^N)$ represents the effect of aliasing. Since $0 < a < 1$, the aliasing error tends toward zero as $N \to \infty$.

For $a = 0.8$, the sequence $x(n)$ and its spectrum $X(\omega)$ are shown in Fig. 7.1.4(a) and 7.1.4(b), respectively. The aliased sequences $x_p(n)$ for $N = 5$ and $N = 50$ and the corresponding spectral samples are shown in Fig. 7.1.4(c) and 7.1.4(d), respectively. We note that the aliasing effects are negligible for $N = 50$.

If we define the aliased finite-duration sequence $x(n)$ as

$$\hat{x}(n) = \begin{cases} x_p(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

then its Fourier transform is

$$\hat{X}(\omega) = \sum_{n=0}^{N-1} \hat{x}(n)e^{-j\omega n} = \sum_{n=0}^{N-1} x_p(n)e^{-j\omega N}$$

$$= \frac{1}{1 - a^N} \cdot \frac{1 - a^N e^{-j\omega n}}{1 - ae^{-j\omega}}$$

Note that although $\hat{X}(\omega) \neq X(\omega)$, the sample values at $\omega_k = 2\pi k/N$ are identical. That is,

$$\hat{X}\left(\frac{2\pi}{N}k\right) = \frac{1}{1 - a^N} \cdot \frac{1 - a^N}{1 - ae^{-j2\pi k/N}} = X\left(\frac{2\pi}{N}k\right)$$

7.1.2 The Discrete Fourier Transform (DFT)

The development in the preceding section is concerned with the frequency-domain sampling of an aperiodic finite-energy sequence $x(n)$. In general, the equally spaced frequency samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$, do not uniquely represent the original sequence $x(n)$ when $x(n)$ has infinite duration. Instead, the frequency samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$, correspond to a periodic sequence $x_p(n)$ of

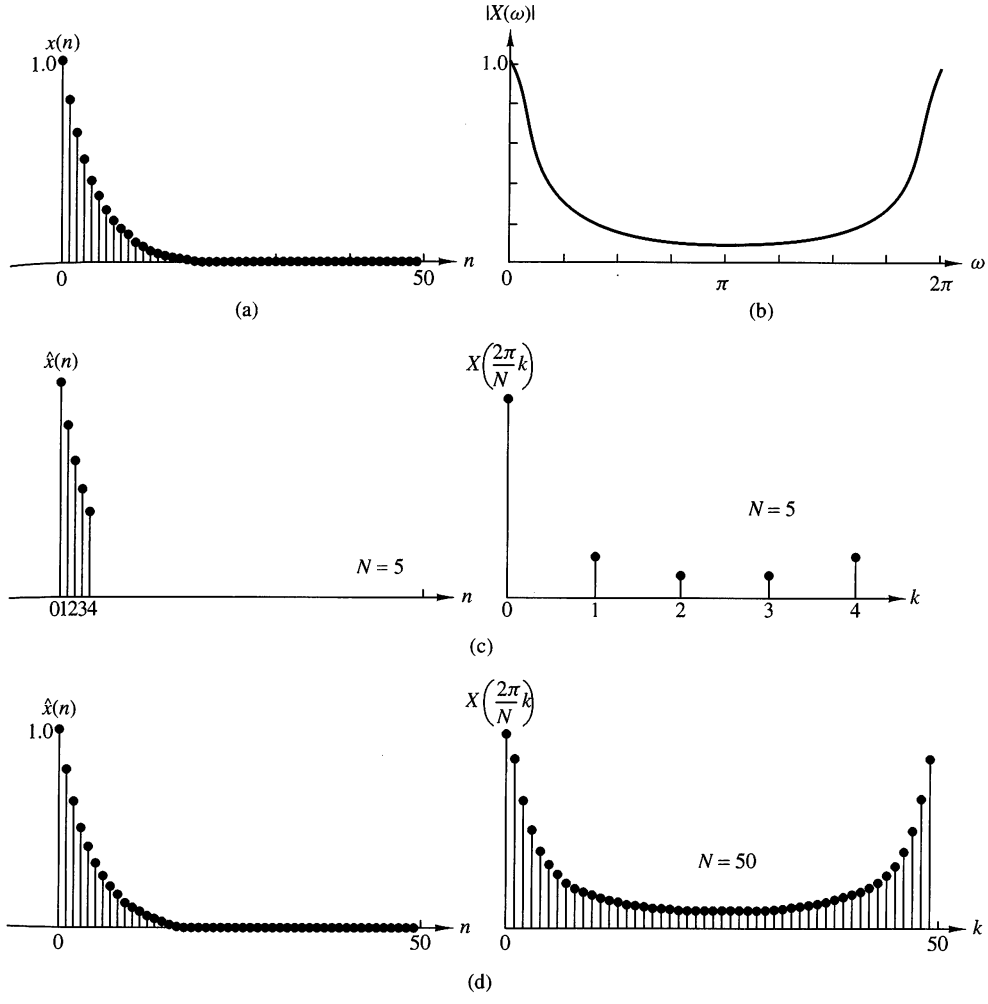


Figure 7.1.4 (a) Plot of sequence $x(n) = (0.8)^n u(n)$; (b) its Fourier transform (magnitude only); (c) effect of aliasing with $N = 5$; (d) reduced effect of aliasing with $N = 50$.

period N, where $x_p(n)$ is an aliased version of $x(n)$, as indicated by the relation in (7.1.4), that is,

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \tag{7.1.15}$$

When the sequence $x(n)$ has a finite duration of length $L \leq N$, then $x_p(n)$ is simply a periodic repetition of $x(n)$, where $x_p(n)$ over a single period is given as

$$x_p(n) = \begin{cases} x(n), & 0 \leq n \leq L-1 \\ 0, & L \leq n \leq N-1 \end{cases} \tag{7.1.16}$$

Consequently, the frequency samples $X(2\pi k/N)$, $k = 0, 1, \ldots, N-1$, uniquely represent the finite-duration sequence $x(n)$. Since $x(n) \equiv x_p(n)$ over a single period

(padded by $N - L$ zeros), the original finite-duration sequence $x(n)$ can be obtained from the frequency samples $\{X(2\pi k/N\}$ by means of the formula (7.1.8).

It is important to note that *zero padding* does not provide any additional information about the spectrum $X(\omega)$ of the sequence $\{x(n)\}$. The L equidistant samples of $X(\omega)$ are sufficient to reconstruct $X(\omega)$ using the reconstruction formula (7.1.13). However, padding the sequence $\{x(n)\}$ with $N - L$ zeros and computing an N-point DFT results in a "better display" of the Fourier transform $X(\omega)$.

In summary, a finite-duration sequence $x(n)$ of length L [i.e., $x(n) = 0$ for $n < 0$ and $n \geq L$] has a Fourier transform

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n}, \qquad 0 \leq \omega \leq 2\pi \tag{7.1.17}$$

where the upper and lower indices in the summation reflect the fact that $x(n) = 0$ outside the range $0 \leq n \leq L-1$. When we sample $X(\omega)$ at equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, 2, \ldots, N - 1$, where $N \geq L$, the resultant samples are

$$\begin{aligned} X(k) \equiv X\left(\frac{2\pi k}{N}\right) &= \sum_{n=0}^{L-1} x(n)e^{-j2\pi kn/N} \\ X(k) &= \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \qquad k = 0, 1, 2, \ldots, N - 1 \end{aligned} \tag{7.1.18}$$

where for convenience, the upper index in the sum has been increased from $L - 1$ to $N - 1$ since $x(n) = 0$ for $n \geq L$.

The relation in (7.1.18) is a formula for transforming a sequence $\{x(n)\}$ of length $L \leq N$ into a sequence of frequency samples $\{X(k)\}$ of length N. Since the frequency samples are obtained by evaluating the Fourier transform $X(\omega)$ at a set of N (equally spaced) discrete frequencies, the relation in (7.1.18) is called the *discrete Fourier transform* (DFT) of $x(n)$. In turn, the relation given by (7.1.10), which allows us to recover the sequence $x(n)$ from the frequency samples

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}, \qquad n = 0, 1, \ldots, N - 1 \tag{7.1.19}$$

is called the *inverse DFT* (IDFT). Clearly, when $x(n)$ has length $L < N$, the N-point IDFT yields $x(n) = 0$ for $L \leq n \leq N - 1$. To summarize, the formulas for the DFT and IDFT are

$$\text{DFT:} \quad X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \qquad k = 0, 1, 2, \ldots, N - 1 \tag{7.1.20}$$

$$\text{IDFT:} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}, \qquad n = 0, 1, 2, \ldots, N - 1 \tag{7.1.21}$$

EXAMPLE 7.1.2

A finite-duration sequence of length L is given as

$$x(n) = \begin{cases} 1, & 0 \le n \le L-1 \\ 0, & \text{otherwise} \end{cases}$$

Determine the N-point DFT of this sequence for $N \ge L$.

Solution. The Fourier transform of this sequence is

$$X(\omega) = \sum_{n=0}^{L-1} x(n)e^{-j\omega n}$$

$$= \sum_{n=0}^{L-1} e^{-j\omega n} = \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}} = \frac{\sin(\omega L/2)}{\sin(\omega/2)} e^{-j\omega(L-1)/2}$$

The magnitude and phase of $X(\omega)$ are illustrated in Fig. 7.1.5 for $L = 10$. The N-point DFT of $x(n)$ is simply $X(\omega)$ evaluated at the set of N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$. Hence

$$X(k) = \frac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}}, \qquad k = 0, 1, \ldots, N-1$$

$$= \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} e^{-j\pi k(L-1)/N}$$

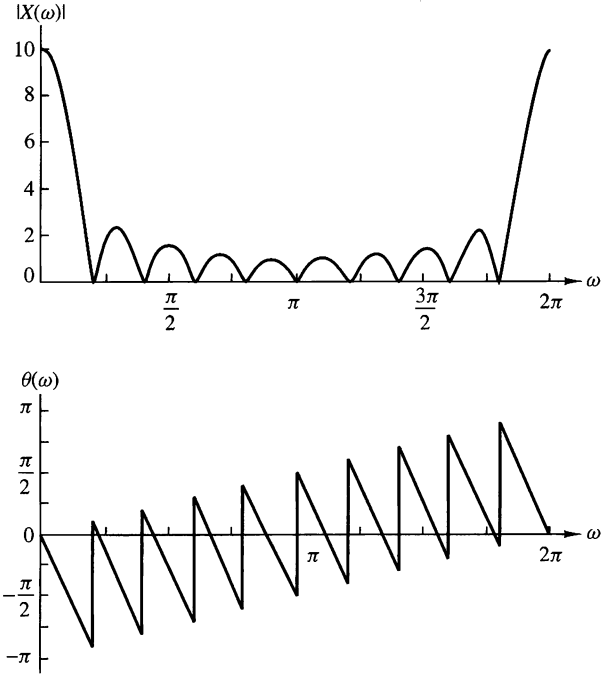


Figure 7.1.5 Magnitude and phase characteristics of the Fourier transform for signal in Example 7.1.2.

If N is selected such that $N = L$, then the DFT becomes

$$X(k) = \begin{cases} L, & k = 0 \\ 0, & k = 1, 2, \ldots, L-1 \end{cases}$$

Thus there is only one nonzero value in the DFT. This is apparent from observation of $X(\omega)$, since $X(\omega) = 0$ at the frequencies $\omega_k = 2\pi k/L$, $k \neq 0$. The reader should verify that $x(n)$ can be recovered from $X(k)$ by performing an L-point IDFT.

Although the L-point DFT is sufficient to uniquely represent the sequence $x(n)$ in the frequency domain, it is apparent that it does not provide sufficient detail to yield a good picture of the spectral characteristics of $x(n)$. If we wish to have a better picture, we must evaluate (interpolate) $X(\omega)$ at more closely spaced frequencies, say $\omega_k = 2\pi k/N$, where $N > L$. In effect, we can view this computation as expanding the size of the sequence from L points to N points by appending $N - L$ zeros to the sequence $x(n)$, that is, zero padding. Then the N-point DFT provides finer interpolation than the L-point DFT.

Figure 7.1.6 provides a plot of the N-point DFT, in magnitude and phase, for $L = 10$, $N = 50$, and $N = 100$. Now the spectral characteristics of the sequence are more clearly evident, as one will conclude by comparing these spectra with the continuous spectrum $X(\omega)$.

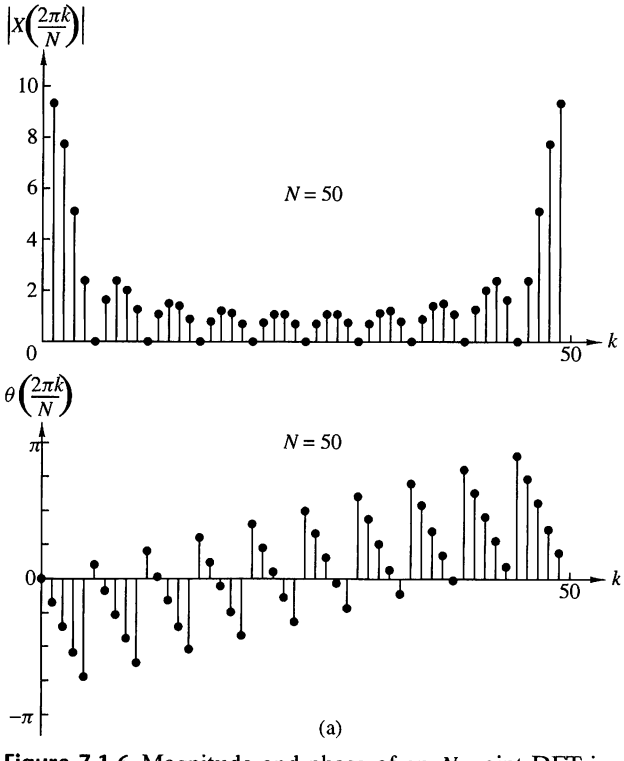


Figure 7.1.6 Magnitude and phase of an N-point DFT in Example 7.1.2; (a) $L = 10$, $N = 50$; (b) $L = 10$, $N = 100$.

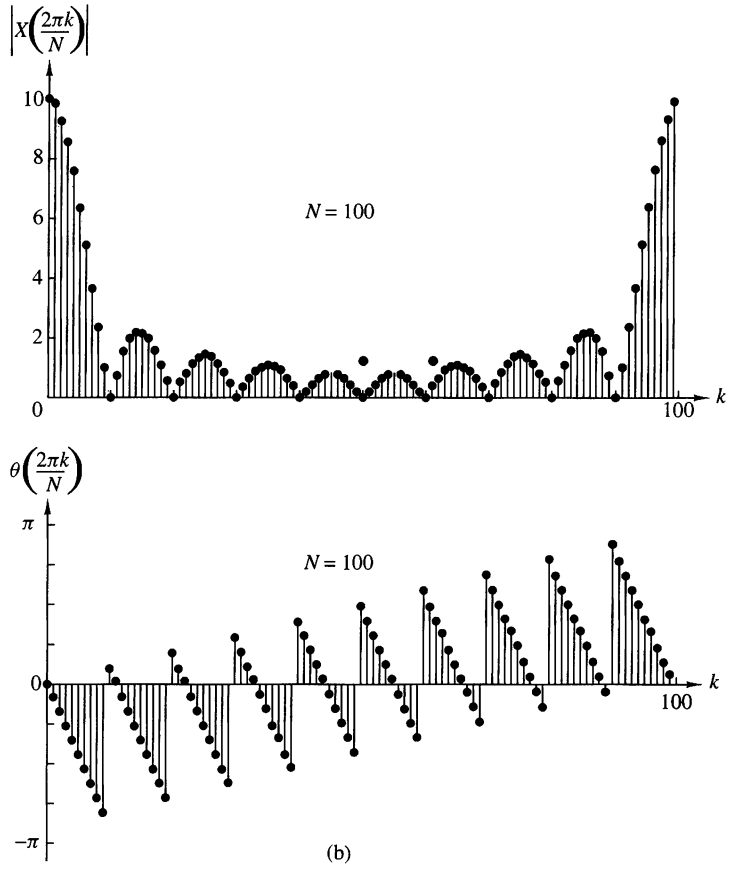


Figure 7.1.6 *continued*

7.1.3 The DFT as a Linear Transformation

The formulas for the DFT and IDFT given by (7.1.18) and (7.1.19) may be expressed as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1 \tag{7.1.22}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \qquad n = 0, 1, \ldots, N-1 \tag{7.1.23}$$

where, by definition,

$$W_N = e^{-j2\pi/N} \tag{7.1.24}$$

which is an Nth root of unity.

We note that the computation of each point of the DFT can be accomplished by N complex multiplications and $(N-1)$ complex additions. Hence the N-point DFT values can be computed in a total of N^2 complex multiplications and $N(N-1)$ complex additions.

It is instructive to view the DFT and IDFT as linear transformations on sequences $\{x(n)\}$ and $\{X(k)\}$, respectively. Let us define an N-point vector $\mathbf{x}_N$ of the signal sequence $x(n)$, $n = 0, 1, \ldots, N-1$, an N-point vector $\mathbf{X}_N$ of frequency samples, and an $N \times N$ matrix $\mathbf{W}_N$ as

$$\mathbf{x}_N = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}, \qquad \mathbf{X}_N = \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

$$\mathbf{W}_N = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{N-1} \\ & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \vdots\vdots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \tag{7.1.25}$$

With these definitions, the N-point DFT may be expressed in matrix form as

$$\mathbf{X}_N = \mathbf{W}_N \mathbf{x}_N \tag{7.1.26}$$

where $\mathbf{W}_N$ is the matrix of the linear transformation. We observe that $\mathbf{W}_N$ is a symmetric matrix. If we assume that the inverse of $\mathbf{W}_N$ exists, then (7.1.26) can be inverted by premultiplying both sides by $\mathbf{W}_N^{-1}$. Thus we obtain

$$\mathbf{x}_N = \mathbf{W}_N^{-1} \mathbf{X}_N \tag{7.1.27}$$

But this is just an expression for the IDFT.

In fact, the IDFT as given by (7.1.23) can be expressed in matrix form as

$$\mathbf{x}_N = \frac{1}{N} \mathbf{W}_N^* \mathbf{X}_N \tag{7.1.28}$$

where $\mathbf{W}_N^*$ denotes the complex conjugate of the matrix $\mathbf{W}_N$. Comparison of (7.1.27) with (7.1.28) leads us to conclude that

$$\mathbf{W}_N^{-1} = \frac{1}{N} \mathbf{W}_N^* \tag{7.1.29}$$

which, in turn, implies that

$$\mathbf{W}_N \mathbf{W}_N^* = N \mathbf{I}_N \tag{7.1.30}$$

where $\mathbf{I}_N$ is an $N \times N$ identity matrix. Therefore, the matrix $\mathbf{W}_N$ in the transformation is an orthogonal (unitary) matrix. Furthermore, its inverse exists and is given as $\mathbf{W}_N^*/N$. Of course, the existence of the inverse of $\mathbf{W}_N$ was established previously from our derivation of the IDFT.

EXAMPLE 7.1.3

Compute the DFT of the four-point sequence

$$x(n) = (0 \quad 1 \quad 2 \quad 3)$$

Solution. The first step is to determine the matrix $\mathbf{W}_4$. By exploiting the periodicity property of $\mathbf{W}_4$ and the symmetry property

$$W_N^{k+N/2} = -W_N^k$$

the matrix $\mathbf{W}_4$ may be expressed as

$$\mathbf{W}_4 = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & W_4^0 & W_4^2 \\ 1 & W_4^3 & W_4^2 & W_4^1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

Then

$$\mathbf{X}_4 = \mathbf{W}_4\mathbf{x}_4 = \begin{bmatrix} 6 \\ -2 + 2j \\ -2 \\ -2 - 2j \end{bmatrix}$$

The IDFT of $\mathbf{X}_4$ may be determined by conjugating the elements in $\mathbf{W}_4$ to obtain $\mathbf{W}_4^*$ and then applying the formula (7.1.28).

The DFT and IDFT are computational tools that play a very important role in many digital signal processing applications, such as frequency analysis (spectrum analysis) of signals, power spectrum estimation, and linear filtering. The importance of the DFT and IDFT in such practical applications is due to a large extent to the existence of computationally efficient algorithms, known collectively as fast Fourier transform (FFT) algorithms, for computing the DFT and IDFT. This class of algorithms is described in Chapter 8.

7.1.4 Relationship of the DFT to Other Transforms

In this discussion we have indicated that the DFT is an important computational tool for performing frequency analysis of signals on digital signal processors. In view of the other frequency analysis tools and transforms that we have developed, it is important to establish the relationships of the DFT to these other transforms.

Relationship to the Fourier series coefficients of a periodic sequence. A periodic sequence $\{x_p(n)\}$ with fundamental period N can be represented in a Fourier series of the form

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j2\pi nk/N}, \qquad -\infty < n < \infty \tag{7.1.31}$$

where the Fourier series coefficients are given by the expression

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1 \tag{7.1.32}$$

If we compare (7.1.31) and (7.1.32) with (7.1.18) and (7.1.19), we observe that the formula for the Fourier series coefficients has the form of a DFT. In fact, if we define a sequence $x(n) = x_p(n)$, $0 \le n \le N-1$, the DFT of this sequence is simply

$$X(k) = Nc_k \tag{7.1.33}$$

Furthermore, (7.1.31) has the form of an IDFT. Thus the N-point DFT provides the exact line spectrum of a periodic sequence with fundamental period N.

Relationship to the Fourier transform of an aperiodic sequence. We have already shown that if $x(n)$ is an aperiodic finite energy sequence with Fourier transform $X(\omega)$, which is sampled at N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, the spectral components

$$X(k) = X(\omega)|_{\omega=2\pi k/N} = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1 \tag{7.1.34}$$

are the DFT coefficients of the periodic sequence of period N, given by

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \tag{7.1.35}$$

Thus $x_p(n)$ is determined by aliasing $\{x(n)\}$ over the interval $0 \le n \le N-1$. The finite-duration sequence

$$\hat{x}(n) = \begin{cases} x_p(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{7.1.36}$$

bears no resemblance to the original sequence $\{x(n)\}$, unless $x(n)$ is of finite duration and length $L \le N$, in which case

$$x(n) = \hat{x}(n), \qquad 0 \le n \le N-1 \tag{7.1.37}$$

Only in this case will the IDFT of $\{X(k)\}$ yield the original sequence $\{x(n)\}$.

Relationship to the z-transform. Let us consider a sequence $x(n)$ having the z-transform

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \tag{7.1.38}$$

with an ROC that includes the unit circle. If $X(z)$ is sampled at the N equally spaced points on the unit circle $z_k = e^{j2\pi k/N}$, $0, 1, 2, \ldots, N-1$, we obtain

$$\begin{aligned} X(k) &\equiv X(z)|_{z=e^{j2\pi nk/N}}, \qquad k = 0, 1, \ldots, N-1 \\ &= \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi nk/N} \end{aligned} \tag{7.1.39}$$

The expression in (7.1.39) is identical to the Fourier transform $X(\omega)$ evaluated at the N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, which is the topic treated in Section 7.1.1.

If the sequence $x(n)$ has a finite duration of length N or less, the sequence can be recovered from its N-point DFT. Hence its z-transform is uniquely determined by its N-point DFT. Consequently, $X(z)$ can be expressed as a function of the DFT {X(k)} as follows:

$$\begin{aligned} X(z) &= \sum_{n=0}^{N-1} x(n)z^{-n} \\ X(z) &= \sum_{n=0}^{N-1} \left[\frac{1}{N} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N} \right] z^{-n} \\ X(z) &= \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} \left(e^{j2\pi k/N} z^{-1} \right)^n \\ X(z) &= \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{j2\pi k/N} z^{-1}} \end{aligned} \tag{7.1.40}$$

When evaluated on the unit circle, (7.1.40) yields the Fourier transform of the finite-duration sequence in terms of its DFT, in the form

$$X(\omega) = \frac{1 - e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{-j(\omega - 2\pi k/N)}} \tag{7.1.41}$$

This expression for the Fourier transform is a polynomial (Lagrange) interpolation formula for $X(\omega)$ expressed in terms of the values $\{X(k)\}$ of the polynomial at a set of equally spaced discrete frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$. With some algebraic manipulations, it is possible to reduce (7.1.41) to the interpolation formula given previously in (7.1.13).

Relationship to the Fourier series coefficients of a continuous-time signal. Suppose that $x_a(t)$ is a continuous-time periodic signal with fundamental period $T_p = 1/F_0$. The signal can be expressed in a Fourier series

$$x_a(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k t F_0} \tag{7.1.42}$$

where $\{c_k\}$ are the Fourier coefficients. If we sample $x_a(t)$ at a uniform rate $F_s = N/T_p = 1/T$, we obtain the discrete-time sequence

$$\begin{aligned} x(n) \equiv x_a(nT) &= \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 nT} = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi kn/N} \\ &= \sum_{k=0}^{N-1} \left[\sum_{l=-\infty}^{\infty} c_{k-lN} \right] e^{j2\pi kn/N} \end{aligned} \tag{7.1.43}$$

It is clear that (7.1.43) is in the form of an IDFT formula, where

$$X(k) = N \sum_{l=-\infty}^{\infty} c_{k-lN} \equiv N\tilde{c}_k \tag{7.1.44}$$

and

$$\tilde{c}_k = \sum_{l=-\infty}^{\infty} c_{k-lN} \tag{7.1.45}$$

Thus the $\{\tilde{c}_k\}$ sequence is an aliased version of the sequence $\{c_k\}$.

7.2 Properties of the DFT

In Section 7.1.2 we introduced the DFT as a set of N samples $\{X(k)\}$ of the Fourier transform $X(\omega)$ for a finite-duration sequence $\{x(n)\}$ of length $L \leq N$. The sampling of $X(\omega)$ occurs at the N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, 2, \ldots, N-1$. We demonstrated that the N samples $\{X(k)\}$ uniquely represent the sequence $\{x(n)\}$ in the frequency domain. Recall that the DFT and inverse DFT (IDFT) for an N-point sequence $\{x(n)\}$ are given as

$$\text{DFT:} \quad X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1 \tag{7.2.1}$$

$$\text{IDFT:} \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \qquad n = 0, 1, \ldots, N-1 \tag{7.2.2}$$

where W_N is defined as

$$W_N = e^{-j2\pi/N} \tag{7.2.3}$$

In this section we present the important properties of the DFT. In view of the relationships established in Section 7.1.4 between the DFT and Fourier series, and Fourier transforms and z-transforms of discrete-time signals, we expect the properties of the DFT to resemble the properties of these other transforms and series. However, some important differences exist, one of which is the circular convolution property derived in the following section. A good understanding of these properties is extremely helpful in the application of the DFT to practical problems.

The notation used below to denote the N-point DFT pair $x(n)$ and $X(k)$ is

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)$$

7.2.1 Periodicity, Linearity, and Symmetry Properties

Periodicity. If $x(n)$ and X(k) are an N-point DFT pair, then

$$x(n+N) = x(n) \qquad \text{for all } n \tag{7.2.4}$$

$$X(k+N) = X(k) \qquad \text{for all } k \tag{7.2.5}$$

These periodicities in $x(n)$ and $X(k)$ follow immediately from formulas (7.2.1) and (7.2.2) for the DFT and IDFT, respectively.

We previously illustrated the periodicity property in the sequence $x(n)$ for a given DFT. However, we had not previously viewed the DFT $X(k)$ as a periodic sequence. In some applications it is advantageous to do this.

Linearity. If

$$x_1(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_1(k)$$

and

$$x_2(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_2(k)$$

then for any real-valued or complex-valued constants a_1 and a_2,

$$a_1x_1(n) + a_2x_2(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} a_1X_1(k) + a_2X_2(k) \tag{7.2.6}$$

This property follows immediately from the definition of the DFT given by (7.2.1).

Circular Symmetries of a Sequence. As we have seen, the N-point DFT of a finite duration sequence $x(n)$, of length $L \leq N$, is equivalent to the N-point DFT of a periodic sequence $x_p(n)$, of period N, which is obtained by periodically extending $x(n)$, that is,

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN) \tag{7.2.7}$$

Now suppose that we shift the periodic sequence $x_p(n)$ by k units to the right. Thus we obtain another periodic sequence

$$x_p'(n) = x_p(n-k) = \sum_{l=-\infty}^{\infty} x(n-k-lN) \tag{7.2.8}$$

The finite-duration sequence

$$x'(n) = \begin{cases} x_p'(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{7.2.9}$$

is related to the original sequence $x(n)$ by a circular shift. This relationship is illustrated in Fig. 7.2.1 for $N = 4$.

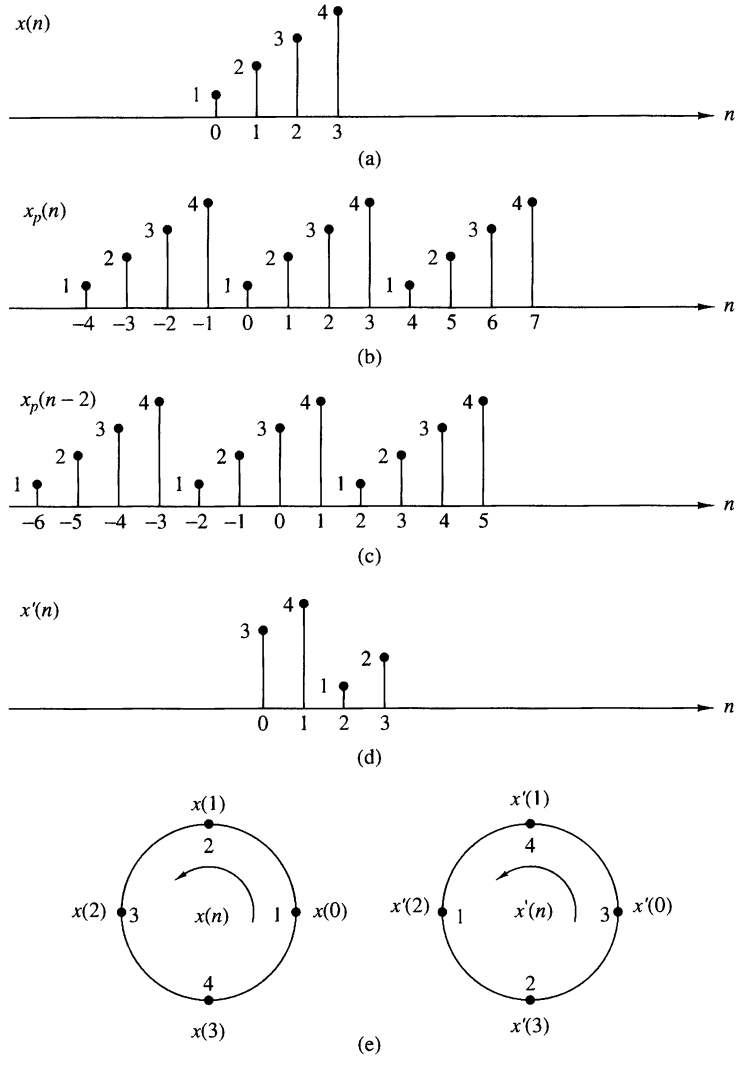


Figure 7.2.1 Circular shift of a sequence.

In general, the circular shift of the sequence can be represented as the index modulo N. Thus we can write

$$\begin{aligned} x'(n) &= x(n-k, \text{modulo } N) \\ &\equiv x((n-k))_N \end{aligned} \tag{7.2.10}$$

For example, if $k = 2$ and $N = 4$, we have

$$x'(n) = x((n-2))_4$$

which implies that

$$\begin{aligned} x'(0) &= x((-2))_4 = x(2) \\ x'(1) &= x((-1))_4 = x(3) \\ x'(2) &= x((0))_4 = x(0) \\ x'(3) &= x((1))_4 = x(1) \end{aligned}$$

Hence $x'(n)$ is simply $x(n)$ shifted circularly by two units in time, where the counterclockwise direction has been arbitrarily selected as the positive direction. Thus we conclude that a circular shift of an N-point sequence is equivalent to a linear shift of its periodic extension, and vice versa.

The inherent periodicity resulting from the arrangement of the N-point sequence on the circumference of a circle dictates a different definition of even and odd symmetry, and time reversal of a sequence.

An N-point sequence is called circularly *even* if it is symmetric about the point zero on the circle. This implies that

$$x(N-n) = x(n), \qquad 1 \le n \le N-1 \tag{7.2.11}$$

An N-point sequence is called circularly *odd* if it is antisymmetric about the point zero on the circle. This implies that

$$x(N-n) = -x(n), \qquad 1 \le n \le N-1 \tag{7.2.12}$$

The time reversal of an N-point sequence is attained by reversing its samples about the point zero on the circle. Thus the sequence $x((-n))_N$ is simply given as

$$x((-n))_N = x(N-n), \qquad 0 \le n \le N-1 \tag{7.2.13}$$

This time reversal is equivalent to plotting $x(n)$ in a clockwise direction on a circle.

An equivalent definition of even and odd sequences for the associated periodic sequence $x_p(n)$ is given as follows

$$\begin{aligned} \text{even:} \quad & x_p(n) = x_p(-n) = x_p(N-n) \\ \text{odd:} \quad & x_p(n) = -x_p(-n) = -x_p(N-n) \end{aligned} \tag{7.2.14}$$

If the periodic sequence is complex valued, we have

$$\begin{aligned} &\text{conjugate even:} \quad x_p(n) = x_p^*(N-n) \\ &\text{conjugate odd:} \quad x_p(n) = -x_p^*(N-n) \end{aligned} \tag{7.2.15}$$

These relationships suggest that we decompose the sequence $x_p(n)$ as

$$x_p(n) = x_{pe}(n) + x_{po}(n) \tag{7.2.16}$$

where

$$\begin{aligned} x_{pe}(n) &= \frac{1}{2}[x_p(n) + x_p^*(N-n)] \\ x_{po}(n) &= \frac{1}{2}[x_p(n) - x_p^*(N-n)] \end{aligned} \tag{7.2.17}$$

Symmetry properties of the DFT. The symmetry properties for the DFT can be obtained by applying the methodology previously used for the Fourier transform. Let us assume that the N-point sequence $x(n)$ and its DFT are both complex valued. Then the sequences can be expressed as

$$x(n) = x_R(n) + jx_I(n), \qquad 0 \le n \le N-1 \tag{7.2.18}$$

$$X(k) = X_R(k) + jX_I(k), \qquad 0 \le k \le N-1 \tag{7.2.19}$$

By substituting (7.2.18) into the expression for the DFT given by (7.2.1), we obtain

$$X_R(k) = \sum_{n=0}^{N-1} \left[x_R(n) \cos \frac{2\pi kn}{N} + x_I(n) \sin \frac{2\pi kn}{N} \right] \tag{7.2.20}$$

$$X_I(k) = -\sum_{n=0}^{N-1} \left[x_R(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N} \right] \tag{7.2.21}$$

Similarly, by substituting (7.2.19) into the expression for the IDFT given by (7.2.2), we obtain

$$x_R(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left[X_R(k) \cos \frac{2\pi kn}{N} - X_I(k) \sin \frac{2\pi kn}{N} \right] \tag{7.2.22}$$

$$x_I(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left[X_R(k) \sin \frac{2\pi kn}{N} + X_I(k) \cos \frac{2\pi kn}{N} \right] \tag{7.2.23}$$

Real-valued sequences. If the sequence $x(n)$ is real, it follows directly from (7.2.1) that

$$X(N-k) = X^*(k) = X(-k) \tag{7.2.24}$$

Consequently, $|X(N-k)| = |X(k)|$ and $\angle X(N-k) = -\angle X(k)$. Furthermore, $x_I(n) = 0$ and therefore $x(n)$ can be determined from (7.2.22), which is another form for the IDFT.

Real and even sequences. If $x(n)$ is real and even, that is,

$$x(n) = x(N-n), \qquad 0 \le n \le N-1$$

then (7.2.21) yields $X_I(k) = 0$. Hence the DFT reduces to

$$X(k) = \sum_{n=0}^{N-1} x(n) \cos \frac{2\pi kn}{N}, \qquad 0 \le k \le N-1 \tag{7.2.25}$$

which is itself real valued and even. Furthermore, since $X_I(k) = 0$, the IDFT reduces to

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cos \frac{2\pi kn}{N}, \qquad 0 \le n \le N-1 \tag{7.2.26}$$

Real and odd sequences. If $x(n)$ is real and odd, that is,

$$x(n) = -x(N-n), \qquad 0 \le n \le N-1$$

then (7.2.20) yields $X_R(k) = 0$. Hence

$$X(k) = -j \sum_{n=0}^{N-1} x(n) \sin \frac{2\pi kn}{N}, \qquad 0 \le k \le N-1 \tag{7.2.27}$$

which is purely imaginary and odd. Since $X_R(k) = 0$, the IDFT reduces to

$$x(n) = j \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sin \frac{2\pi kn}{N}, \qquad 0 \le n \le N-1 \tag{7.2.28}$$

Purely imaginary sequences. In this case, $x(n) = jx_I(n)$. Consequently, (7.2.20) and (7.2.21) reduce to

$$X_R(k) = \sum_{n=0}^{N-1} x_I(n) \sin \frac{2\pi kn}{N} \tag{7.2.29}$$

$$X_I(k) = \sum_{n=0}^{N-1} x_I(n) \cos \frac{2\pi kn}{N} \tag{7.2.30}$$

We observe that $X_R(k)$ is odd and $X_I(k)$ is even.

If $x_I(n)$ is odd, then $X_I(k) = 0$ and hence $X(k)$ is purely real. On the other hand, if $x_I(n)$ is even, then $X_R(k) = 0$ and hence $X(k)$ is purely imaginary.

The symmetry properties given above may be summarized as follows:

$$\begin{array}{c} x(n) = x_R^e(n) + x_R^o(n) + jx_I^e(n) + jx_I^o(n) \\ \updownarrow \qquad\qquad \updownarrow \\ X(k) = X_R^e(k) + X_R^o(k) + jX_I^e(k) + jX_I^o(k) \end{array} \tag{7.2.31}$$

All the symmetry properties of the DFT can easily be deduced from (7.2.31). For example, the DFT of the sequence

$$x_{pe}(n) = \tfrac{1}{2}[x_p(n) + x_p^*(N-n)]$$

is

$$X_R(k) = X_R^e(k) + X_R^o(k)$$

The symmetry properties of the DFT are summarized in Table 7.1. Exploitation of these properties for the efficient computation of the DFT of special sequences is considered in some of the problems at the end of the chapter.

TABLE 7.1 Symmetry Properties of the DFT

N-Point Sequence $x(n)$, $0 \le n \le N-1$	N-Point DFT
$x(n)$	$X(k)$
$x^*(n)$	$X^*(N-k)$
$x^*(N-n)$	$X^*(k)$
$x_R(n)$	$X_{ce}(k) = \frac{1}{2}[X(k) + X^*(N-k)]$
$jX_I(n)$	$X_{co}(k) = \frac{1}{2}[X(k) - X^*(N-k)]$
$x_{ce}(n) = \frac{1}{2}[x(n) + x^*(N-n)]$	$X_R(k)$
$x_{co}(n) = \frac{1}{2}[x(n) - x^*(N-n)]$	$jX_I(k)$
Real Signals	
Any real signal	$X(k) = X^*(N-k)$
$x(n)$	$X_R(k) = X_R(N-k)$
	$X_I(k) = -X_I(N-k)$
	$\lvert X(k)\rvert = \lvert X(N-k)\rvert$
	$\angle X(k) = -\angle X(N-k)$
$x_{ce}(n) = \frac{1}{2}[x(n) + x(N-n)]$	$X_R(k)$
$x_{co}(n) = \frac{1}{2}[x(n) - x(N-n)]$	$jX_I(k)$

7.2.2 Multiplication of Two DFTs and Circular Convolution

Suppose that we have two finite-duration sequences of length N, $x_1(n)$ and $x_2(n)$. Their respective N-point DFTs are

$$X_1(k) = \sum_{n=0}^{N-1} x_1(n)e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1 \tag{7.2.32}$$

$$X_2(k) = \sum_{n=0}^{N-1} x_2(n)e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1 \tag{7.2.33}$$

If we multiply the two DFTs together, the result is a DFT, say $X_3(k)$, of a sequence $x_3(n)$ of length N. Let us determine the relationship between $x_3(n)$ and the sequences $x_1(n)$ and $x_2(n)$.

We have

$$X_3(k) = X_1(k)X_2(k), \qquad k = 0, 1, \ldots, N-1 \tag{7.2.34}$$

The IDFT of $\{X_3(k)\}$ is

$$\begin{aligned} x_3(m) &= \frac{1}{N}\sum_{k=0}^{N-1} X_3(k)e^{j2\pi km/N} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} X_1(k)X_2(k)e^{j2\pi km/N} \end{aligned} \tag{7.2.35}$$

Suppose that we substitute for $X_1(k)$ and $X_2(k)$ in (7.2.35) using the DFTs given in (7.2.32) and (7.2.33). Thus we obtain

$$\begin{aligned} x_3(m) &= \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{n=0}^{N-1} x_1(n)e^{-j2\pi kn/N}\right]\left[\sum_{l=0}^{N-1} x_2(l)e^{-j2\pi kl/N}\right]e^{j2\pi km/N} \\ &= \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)\sum_{l=0}^{N-1} x_2(l)\left[\sum_{k=0}^{N-1} e^{j2\pi k(m-n-l)/N}\right] \end{aligned} \tag{7.2.36}$$

The inner sum in the brackets in (7.2.36) has the form

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N, & a = 1 \\ \dfrac{1-a^N}{1-a}, & a \neq 1 \end{cases} \tag{7.2.37}$$

where a is defined as

$$a = e^{j2\pi(m-n-l)/N}$$

We observe that $a = 1$ when $m - n - l$ is a multiple of N. On the other hand, $a^N = 1$ for any value of $a \neq 0$. Consequently, (7.2.37) reduces to

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N, & l = m - n + pN = ((m-n))_N, \quad p \text{ an integer} \\ 0, & \text{otherwise} \end{cases} \tag{7.2.38}$$

If we substitute the result in (7.2.38) into (7.2.36), we obtain the desired expression for $x_3(m)$ in the form

$$x_3(m) = \sum_{n=0}^{N-1} x_1(n) x_2((m-n))_N, \qquad m = 0, 1, \ldots, N-1 \tag{7.2.39}$$

The expression in (7.2.39) has the form of a convolution sum. However, it is not the ordinary linear convolution that was introduced in Chapter 2, which relates the output sequence $y(n)$ of a linear system to the input sequence $x(n)$ and the impulse response $h(n)$. Instead, the convolution sum in (7.2.39) involves the index $((m-n))_N$ and is called *circular convolution.* Thus we conclude that multiplication of the DFTs of two sequences is equivalent to the circular convolution of the two sequences in the time domain.

The following example illustrates the operations involved in circular convolution.

EXAMPLE 7.2.1

Perform the circular convolution of the following two sequences:

$$x_1(n) = \{\underset{\uparrow}{2}, 1, 2, 1\}$$

$$x_2(n) = \{\underset{\uparrow}{1}, 2, 3, 4\}$$

Solution. Each sequence consists of four nonzero points. For the purposes of illustrating the operations involved in circular convolution, it is desirable to graph each sequence as points on a circle. Thus the sequences $x_1(n)$ and $x_2(n)$ are graphed as illustrated in Fig. 7.2.2(a). We note that the sequences are graphed in a counterclockwise direction on a circle. This establishes the reference direction in rotating one of the sequences relative to the other.

Now, $x_3(m)$ is obtained by circularly convolving $x_1(n)$ with $x_2(n)$ as specified by (7.2.39). Beginning with $m = 0$ we have

$$x_3(0) = \sum_{n=0}^{3} x_1(n) x_2((-n))N$$

$x_2((-n))_4$ is simply the sequence $x_2(n)$ folded and graphed on a circle as illustrated in Fig. 7.2.2(b). In other words, the folded sequence is simply $x_2(n)$ graphed in a clockwise direction.

The product sequence is obtained by multiplying $x_1(n)$ with $x_2((-n))_4$, point by point. This sequence is also illustrated in Fig. 7.2.2(b). Finally, we sum the values in the product sequence to obtain

$$x_3(0) = 14$$

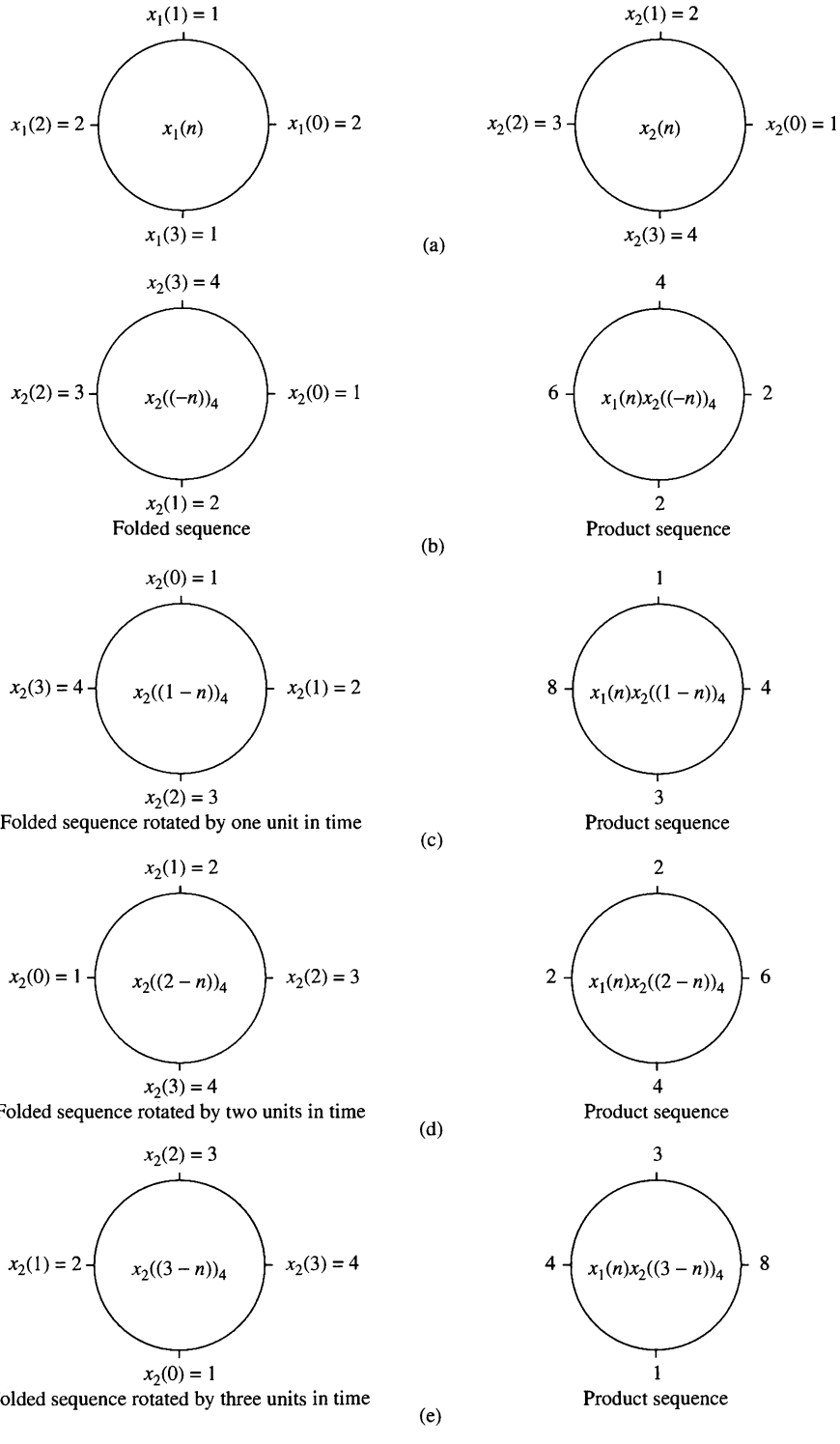


Figure 7.2.2 Circular convolution of two sequences.

For $m = 1$ we have

$$x_3(1) = \sum_{n=0}^{3} x_1(n)x_2((1-n))_4$$

It is easily verified that $x_2((1-n))_4$ is simply the sequence $x_2((-n))_4$ rotated counterclockwise by one unit in time as illustrated in Fig. 7.2.2(c). This rotated sequence multiplies $x_1(n)$ to yield the product sequence, also illustrated in Fig. 7.2.2(c). Finally, we sum the values in the product sequence to obtain $x_3(1)$. Thus

$$x_3(1) = 16$$

For $m = 2$ we have

$$x_3(2) = \sum_{n=0}^{3} x_1(n)x_2((2-n))_4$$

Now $x_2((2-n))_4$ is the folded sequence in Fig. 7.2.2(b) rotated two units of time in the counterclockwise direction. The resultant sequence is illustrated in Fig. 7.2.2(d) along with the product sequence $x_1(n)x_2((2-n))_4$. By summing the four terms in the product sequence, we obtain

$$x_3(2) = 14$$

For $m = 3$ we have

$$x_3(3) = \sum_{n=0}^{3} x_1(n)x_2((3-n))_4$$

The folded sequence $x_2((-n))_4$ is now rotated by three units in time to yield $x_2((3-n))_4$ and the resultant sequence is multiplied by $x_1(n)$ to yield the product sequence as illustrated in Fig. 7.2.2(e). The sum of the values in the product sequence is

$$x_3(3) = 16$$

We observe that if the computation above is continued beyond $m = 3$, we simply repeat the sequence of four values obtained above. Therefore, the circular convolution of the two sequences $x_1(n)$ and $x_2(n)$ yields the sequence

$$x_3(n) = \{\underset{\uparrow}{14}, 16, 14, 16\}$$

From this example, we observe that circular convolution involves basically the same four steps as the ordinary *linear convolution* introduced in Chapter 2: *folding* (time reversing) one sequence, *shifting* the folded sequence, *multiplying* the two sequences to obtain a product sequence, and finally, *summing* the values of the product sequence. The basic difference between these two types of convolution is that, in circular convolution, the folding and shifting (rotating) operations are performed in a circular fashion by computing the index of one of the sequences modulo N. In linear convolution, there is no modulo N operation.

The reader can easily show from our previous development that either one of the two sequences may be folded and rotated without changing the result of the circular convolution. Thus

$$x_3(m) = \sum_{n=0}^{N-1} x_2(n)x_1((m-n))_N, \qquad m = 0, 1, \ldots, N-1 \tag{7.2.40}$$

The following example serves to illustrate the computation of $x_3(n)$ by means of the DFT and IDFT.

EXAMPLE 7.2.2

By means of the DFT and IDFT, determine the sequence $x_3(n)$ corresponding to the circular convolution of the sequences $x_1(n)$ and $x_2(n)$ given in Example 7.2.1.

Solution. First we compute the DFTs of $x_1(n)$ and $x_2(n)$. The four-point DFT of $x_1(n)$ is

$$X_1(k) = \sum_{n=0}^{3} x_1(n)e^{-j2\pi nk/4}, \qquad k = 0, 1, 2, 3$$

$$= 2 + e^{-j\pi k/2} + 2e^{-j\pi k} + e^{-j3\pi k/2}$$

Thus

$$X_1(0) = 6, \qquad X_1(1) = 0, \qquad X_1(2) = 2, \qquad X_1(3) = 0$$

The DFT of $x_2(n)$ is

$$X_2(k) = \sum_{n=0}^{3} x_2(n)e^{-j2\pi nk/4}, \qquad k = 0, 1, 2, 3$$

$$= 1 + 2e^{-j\pi k/2} + 3e^{-j\pi k} + 4e^{-j3\pi k/2}$$

Thus

$$X_2(0) = 10, \qquad X_2(1) = -2 + j2, \qquad X_2(2) = -2, \qquad X_2(3) = -2 - j2$$

When we multiply the two DFTs, we obtain the product

$$X_3(k) = X_1(k)X_2(k)$$

or, equivalently,

$$X_3(0) = 60, \qquad X_3(1) = 0, \qquad X_3(2) = -4, \qquad X_3(3) = 0$$

Now, the IDFT of $X_3(k)$ is

$$x_3(n) = \sum_{k=0}^{3} X_3(k)e^{j2\pi nk/4}, \qquad n = 0, 1, 2, 3$$

$$= \frac{1}{4}(60 - 4e^{j\pi n})$$

Thus

$$x_3(0) = 14, \qquad x_3(1) = 16, \qquad x_3(2) = 14, \qquad x_3(3) = 16$$

which is the result obtained in Example 7.2.1 from circular convolution.

We conclude this section by formally stating this important property of the DFT.

Circular convolution. If

$$x_1(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_1(k)$$

and

$$x_2(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_2(k)$$

then

$$x_1(n) \textcircled{N} x_2(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_1(k)X_2(k) \tag{7.2.41}$$

where $x_1(n) \textcircled{N} x_2(n)$ denotes the circular convolution of the sequence $x_1(n)$ and $x_2(n)$.

7.2.3 Additional DFT Properties

Time reversal of a sequence. If

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)$$

then

$$x((-n))_N = x(N-n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X((-k))_N = X(N-k) \tag{7.2.42}$$

Hence reversing the N-point sequence in time is equivalent to reversing the DFT values. Time reversal of a sequence $x(n)$ is illustrated in Fig. 7.2.3.

Proof From the definition of the DFT in (7.2.1) we have

$$\text{DFT}\{x(N-n)\} = \sum_{n=0}^{N-1} x(N-n)e^{-j2\pi kn/N}$$

If we change the index from n to $m = N - n$, then

$$\text{DFT}\{x(N-n)\} = \sum_{m=0}^{N-1} x(m)e^{-j2\pi k(N-m)/N}$$

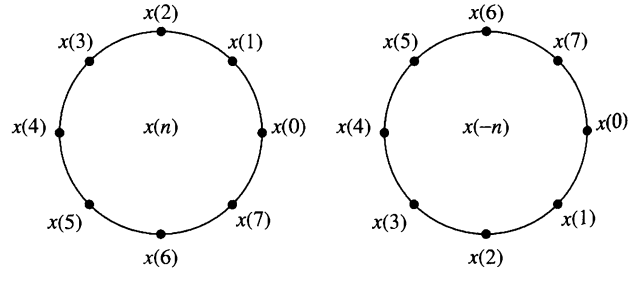


Figure 7.2.3
Time reversal of a sequence.

$$= \sum_{m=0}^{N-1} x(m)e^{j2\pi km/N}$$

$$= \sum_{m=0}^{N-1} x(m)e^{-j2\pi m(N-k)/N} = X(N-k)$$

We note that $X(N-k) = X((-k))_N$, $0 \le k \le N-1$.

Circular time shift of a sequence. If

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)$$

then

$$x((n-l))_N \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)e^{-j2\pi kl/N} \tag{7.2.43}$$

Proof From the definition of the DFT we have

$$\text{DFT}\{x((n-l))_N\} = \sum_{n=0}^{N-1} x((n-l))_N e^{-j2\pi kn/N}$$

$$= \sum_{n=0}^{l-1} x((n-l))_N e^{-j2\pi kn/N}$$

$$+ \sum_{n=l}^{N-1} x(n-l)e^{-j\pi kn/N}$$

But $x((n-l))_N = x(N-l+n)$. Consequently,

$$\sum_{n=0}^{l-1} x((n-l))_N e^{-j2\pi kn/N} = \sum_{n=0}^{l-1} x(N-l+n)e^{-j2\pi kn/N}$$

$$= \sum_{m=N-l}^{N-1} x(m)e^{-j2\pi k(m+l)/N}$$

Furthermore,

$$\sum_{n=l}^{N-1} x(n-l)e^{-j2\pi kn/N} = \sum_{m=0}^{N-1-l} x(m)e^{-j2\pi k(m+l)/N}$$

Therefore,

$$\text{DFT}\{x((n-l))\} = \sum_{m=0}^{N-1} x(m)e^{-j2\pi k(m+l)/N}$$

$$= X(k)e^{-j2\pi kl/N}$$

Circular frequency shift. If

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)$$

then

$$x(n)e^{j2\pi ln/N} \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X((k-l))_N \tag{7.2.44}$$

Hence, the multiplication of the sequence $x(n)$ with the complex exponential sequence $e^{j2\pi kn/N}$ is equivalent to the circular shift of the DFT by l units in frequency. This is the dual to the circular time-shifting property and its proof is similar to that of the latter.

Complex-conjugate properties. If

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)$$

then

$$x^*(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X^*((-k))_N = X^*(N-k) \tag{7.2.45}$$

The proof of this property is left as an exercise for the reader. The IDFT of $X^*(k)$ is

$$\frac{1}{N}\sum_{k=0}^{N-1} X^*(k)e^{j2\pi kn/N} = \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k)e^{j2\pi k(N-n)/N}\right]$$

Therefore,

$$x^*((-n))_N = x^*(N-n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X^*(k) \tag{7.2.46}$$

Circular correlation. In general, for complex-valued sequences $x(n)$ and $y(n)$, if

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)$$

and

$$y(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} Y(k)$$

then

$$\tilde{r}_{xy}(l) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} \tilde{R}_{xy}(k) = X(k)Y^*(k) \tag{7.2.47}$$

where $\tilde{r}_{xy}(l)$ is the (unnormalized) circular crosscorrelation sequence, defined as

$$\tilde{r}xy(l) = \sum_{n=0}^{N-1} x(n)y^*((n-l))N$$

Proof We can write $\tilde{r}_{xy}(l)$ as the circular convolution of $x(n)$ with $y^*(-n)$, that is,

$$\tilde{r}_{xy}(l) = x(l) \textcircled{N} y^*(-l)$$

Then, with the aid of the properties in (7.2.41) and (7.2.46), the N-point DFT of $\tilde{r}_{xy}(l)$ is

$$\tilde{R}_{xy}(k) = X(k)Y^*(k)$$

In the special case where $y(n) = x(n)$, we have the corresponding expression for the circular autocorrelation of $x(n)$,

$$\tilde{r}_{xx}(l) \overset{\underset{N}{\text{DFT}}}{\longleftrightarrow} \tilde{R}_{xx}(k) = |X(k)|^2 \tag{7.2.48}$$

Multiplication of two sequences. If

$$x_1(n) \overset{\underset{N}{\text{DFT}}}{\longleftrightarrow} X_1(k)$$

and

$$x_2(n) \overset{\underset{N}{\text{DFT}}}{\longleftrightarrow} X_2(k)$$

then

$$x_1(n)x_2(n) \overset{\underset{N}{\text{DFT}}}{\longleftrightarrow} \frac{1}{N} X_1(k) \textcircled{N} X_2(k) \tag{7.2.49}$$

This property is the dual of (7.2.41). Its proof follows simply by interchanging the roles of time and frequency in the expression for the circular convolution of two sequences.

Parseval's Theorem. For complex-valued sequences $x(n)$ and $y(n)$, in general, if

$$x(n) \overset{\underset{N}{\text{DFT}}}{\longleftrightarrow} X(k)$$

and

$$y(n) \overset{\underset{N}{\text{DFT}}}{\longleftrightarrow} Y(k)$$

then

$$\sum_{n=0}^{N-1} x(n)y^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)Y^*(k) \tag{7.2.50}$$

Proof The property follows immediately from the circular correlation property in (7.2.47). We have

$$\sum_{n=0}^{N-1} x(n)y^*(n) = \tilde{r}_{xy}(0)$$

TABLE 7.2 Properties of the DFT

Property	Time Domain	Frequency Domain
Notation	$x(n), y(n)$	$X(k), Y(k)$
Periodicity	$x(n) = x(n+N)$	$X(k) = X(k+N)$
Linearity	$a_1x_1(n) + a_2x_2(n)$	$a_1X_1(k) + a_2X_2(k)$
Time reversal	$x(N-n)$	$X(N-k)$
Circular time shift	$x((n-l))_N$	$X(k)e^{-j2\pi kl/N}$
Circular frequency shift	$x(n)e^{j2\pi ln/N}$	$X((k-l))_N$
Complex conjugate	$x^*(n)$	$X^*(N-k)$
Circular convolution	$x_1(n) \textcircled{N} x_2(n)$	$X_1(k)X_2(k)$
Circular correlation	$x(n) \textcircled{N} y^*(-n)$	$X(k)Y^*(k)$
Multiplication of two sequences	$x_1(n)x_2(n)$	$\frac{1}{N}X_1(k) \textcircled{N} X_2(k)$
Parseval's theorem	$\sum_{n=0}^{N-1} x(n)y^*(n)$	$\frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)$

and

$$\tilde{r}_{xy}(l) = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{R}_{xy}(k)e^{j2\pi kl/N}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X(k)Y^*(k)e^{j2\pi kl/N}$$

Hence (7.2.50) follows by evaluating the IDFT at $l = 0$.

The expression in (7.2.50) is the general form of Parseval's theorem. In the special case where $y(n) = x(n)$, (7.2.50) reduces to

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2 \tag{7.2.51}$$

which expresses the energy in the finite-duration sequence $x(n)$ in terms of the frequency components $\{X(k)\}$.

The properties of the DFT given above are summarized in Table 7.2.

7.3 Linear Filtering Methods Based on the DFT

Since the DFT provides a discrete frequency representation of a finite-duration sequence in the frequency domain, it is interesting to explore its use as a computational tool for linear system analysis and, especially, for linear filtering. We have already established that a system with frequency response $H(\omega)$, when excited with an input

signal that has a spectrum $X(\omega)$, possesses an output spectrum $Y(\omega) = X(\omega)H(\omega)$. The output sequence $y(n)$ is determined from its spectrum via the inverse Fourier transform. Computationally, the problem with this frequency-domain approach is that $X(\omega)$, $H(\omega)$, and $Y(\omega)$ are functions of the continuous variable ω. As a consequence, the computations cannot be done on a digital computer, since the computer can only store and perform computations on quantities at discrete frequencies.

On the other hand, the DFT does lend itself to computation on a digital computer. In the discussion that follows, we describe how the DFT can be used to perform linear filtering in the frequency domain. In particular, we present a computational procedure that serves as an alternative to time-domain convolution. In fact, the frequency-domain approach based on the DFT is computationally more efficient than time-domain convolution due to the existence of efficient algorithms for computing the DFT. These algorithms, which are described in Chapter 8, are collectively called fast Fourier transform (FFT) algorithms.

7.3.1 Use of the DFT in Linear Filtering

In the preceding section it was demonstrated that the product of two DFTs is equivalent to the circular convolution of the corresponding time-domain sequences. Unfortunately, circular convolution is of no use to us if our objective is to determine the output of a linear filter to a given input sequence. In this case we seek a frequency-domain methodology equivalent to linear convolution.

Suppose that we have a finite-duration sequence $x(n)$ of length L which excites an FIR filter of length M. Without loss of generality, let

$$x(n) = 0, \qquad n < 0 \text{ and } n \geq L$$

$$h(n) = 0, \qquad n < 0 \text{ and } n \geq M$$

where $h(n)$ is the impulse response of the FIR filter.

The output sequence $y(n)$ of the FIR filter can be expressed in the time domain as the convolution of $x(n)$ and $h(n)$, that is

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{7.3.1}$$

Since $h(n)$ and $x(n)$ are finite-duration sequences, their convolution is also finite in duration. In fact, the duration of $y(n)$ is $L + M - 1$.

The frequency-domain equivalent to (7.3.1) is

$$Y(\omega) = X(\omega)H(\omega) \tag{7.3.2}$$

If the sequence $y(n)$ is to be represented uniquely in the frequency domain by samples of its spectrum $Y(\omega)$ at a set of discrete frequencies, the number of distinct samples must equal or exceed $L + M - 1$. Therefore, a DFT of size $N \geq L + M - 1$ is required to represent $\{y(n)\}$ in the frequency domain.

Now if

$$\begin{aligned} Y(k) &\equiv Y(\omega)|_{\omega=2\pi k/N}, \qquad k = 0, 1, \ldots, N-1 \\ &= X(\omega)H(\omega)|_{\omega=2\pi k/N}, \qquad k = 0, 1, \ldots, N-1 \end{aligned}$$

then

$$Y(k) = X(k)H(k), \qquad k = 0, 1, \ldots, N-1 \tag{7.3.3}$$

where $\{X(k)\}$ and $\{H(k)\}$ are the N-point DFTs of the corresponding sequences $x(n)$ and $h(n)$, respectively. Since the sequences $x(n)$ and $h(n)$ have a duration less than N, we simply pad these sequences with zeros to increase their length to N. This increase in the size of the sequences does not alter their spectra $X(\omega)$ and $H(\omega)$, which are continuous spectra, since the sequences are aperiodic. However, by sampling their spectra at N equally spaced points in frequency (computing the N-point DFTs), we have increased the number of samples that represent these sequences in the frequency domain beyond the minimum number (L or M, respectively).

Since the ($N = L + M - 1$)-point DFT of the output sequence $y(n)$ is sufficient to represent $y(n)$ in the frequency domain, it follows that the multiplication of the N-point DFTs $X(k)$ and $H(k)$, according to (7.3.3), followed by the computation of the N-point IDFT, must yield the sequence $\{y(n)\}$. In turn, this implies that the N-point circular convolution of $x(n)$ with $h(n)$ must be equivalent to the linear convolution of $x(n)$ with $h(n)$. In other words, by increasing the length of the sequences $x(n)$ and $h(n)$ to N points (by appending zeros), and then circularly convolving the resulting sequences, we obtain the same result as would have been obtained with linear convolution. Thus with zero padding, the DFT can be used to perform linear filtering.

The following example illustrates the methodology in the use of the DFT in linear filtering.

EXAMPLE 7.3.1

By means of the DFT and IDFT, determine the response of the FIR filter with impulse response

$$h(n) = \{\underset{\uparrow}{1}, 2, 3\}$$

to the input sequence

$$x(n) = \{\underset{\uparrow}{1}, 2, 2, 1\}$$

Solution. The input sequence has length $L = 4$ and the impulse response has length $M = 3$. Linear convolution of these two sequences produces a sequence of length $N = 6$. Consequently, the size of the DFTs must be at least six.

For simplicity we compute eight-point DFTs. We should also mention that the efficient computation of the DFT via the fast Fourier transform (FFT) algorithm is usually performed for a length N that is a power of 2. Hence the eight-point DFT of $x(n)$ is

$$\begin{aligned} X(k) &= \sum_{n=0}^{7} x(n)e^{-j2\pi kn/8} \\ &= 1 + 2e^{-j\pi k/4} + 2e^{-j\pi k/2} + e^{-j3\pi k/4}, \qquad k = 0, 1, \ldots, 7 \end{aligned}$$

This computation yields

$$X(0) = 6, \qquad X(1) = \frac{2+\sqrt{2}}{2} - j\left(\frac{4+3\sqrt{2}}{2}\right)$$

$$X(2) = -1 - j, \qquad X(3) = \frac{2-\sqrt{2}}{2} + j\left(\frac{4-3\sqrt{2}}{2}\right)$$

$$X(4) = 0, \qquad X(5) = \frac{2-\sqrt{2}}{2} - j\left(\frac{4-3\sqrt{2}}{2}\right)$$

$$X(6) = -1 + j, \qquad X(7) = \frac{2+\sqrt{2}}{2} + j\left(\frac{4+3\sqrt{2}}{2}\right)$$

The eight-point DFT of $h(n)$ is

$$\begin{aligned} H(k) &= \sum_{n=0}^{7} h(n)e^{-j2\pi kn/8} \\ &= 1 + 2e^{-j\pi k/4} + 3e^{-j\pi k/2} \end{aligned}$$

Hence

$$H(0) = 6, \qquad H(1) = 1 + \sqrt{2} - j\left(3 + \sqrt{2}\right), \qquad H(2) = -2 - j2$$

$$H(3) = 1 - \sqrt{2} + j\left(3 - \sqrt{2}\right), \qquad H(4) = 2$$

$$H(5) = 1 - \sqrt{2} - j\left(3 - \sqrt{2}\right), \qquad H(6) = -2 + j2$$

$$H(7) = 1 + \sqrt{2} + j\left(3 + \sqrt{2}\right)$$

The product of these two DFTs yields $Y(k)$, which is

$$Y(0) = 36, \qquad Y(1) = -14.07 - j17.48, \qquad Y(2) = j4, \qquad Y(3) = 0.07 + j0.515$$

$$Y(4) = 0, \qquad Y(5) = 0.07 - j0.515, \qquad Y(6) = -j4, \qquad Y(7) = -14.07 + j17.48$$

Finally, the eight-point IDFT is

$$y(n) = \sum_{k=0}^{7} Y(k)e^{j2\pi kn/8}, \qquad n = 0, 1, \ldots, 7$$

This computation yields the result

$$y(n) = \{\underset{\uparrow}{1}, 4, 9, 11, 8, 3, 0, 0\}$$

We observe that the first six values of $y(n)$ constitute the set of desired output values. The last two values are zero because we used an eight-point DFT and IDFT, when, in fact, the minimum number of points required is six.

Although the multiplication of two DFTs corresponds to circular convolution in the time domain, we have observed that padding the sequences $x(n)$ and $h(n)$ with a sufficient number of zeros forces the circular convolution to yield the same output sequence as linear convolution. In the case of the FIR filtering problem in Example 7.3.1, it is a simple matter to demonstrate that the six-point circular convolution of the sequences

$$h(n) = \{\underset{\uparrow}{1}, 2, 3, 0, 0, 0\} \tag{7.3.4}$$

$$x(n) = \{\underset{\uparrow}{1}, 2, 2, 1, 0, 0\} \tag{7.3.5}$$

results in the output sequence

$$y(n) = \{\underset{\uparrow}{1}, 4, 9, 11, 8, 3\} \tag{7.3.6}$$

which is the same sequence obtained from linear convolution.

It is important for us to understand the aliasing that results in the time domain when the size of the DFTs is smaller than $L + M - 1$. The following example focuses on the aliasing problem.

EXAMPLE 7.3.2

Determine the sequence $y(n)$ that results from the use of four-point DFTs in Example 7.3.1.

Solution. The four-point DFT of $h(n)$ is

$$H(k) = \sum_{n=0}^{3} h(n)e^{-j2\pi kn/4}$$

$$H(k) = 1 + 2e^{-j\pi k/2} + 3e^{-jk\pi}, \qquad k = 0, 1, 2, 3$$

Hence

$$H(0) = 6, \qquad H(1) = -2 - j2, \qquad H(2) = 2, \qquad H(3) = -2 + j2$$

The four-point DFT of $x(n)$ is

$$X(k) = 1 + 2e^{-j\pi k/2} + 2e^{-j\pi k} + 1e^{-j3\pi k/2}, \qquad k = 0, 1, 2, 3$$

Hence

$$X(0) = 6, \qquad X(1) = -1 - j, \qquad X(2) = 0, \qquad X(3) = -1 + j$$

The product of these two four-point DFTs is

$$\hat{Y}(0) = 36, \qquad \hat{Y}(1) = j4, \qquad \hat{Y}(2) = 0, \qquad \hat{Y}(3) = -j4$$

The four-point IDFT yields

$$\hat{y}(n) = \frac{1}{4}\sum_{k=0}^{3} \hat{Y}(k)e^{j2\pi kn/4}, \qquad n = 0, 1, 2, 3$$

$$= \frac{1}{4}(36 + j4e^{j\pi n/2} - j4e^{j3\pi n/2})$$

Therefore,

$$\hat{y}(n) = \{\underset{\uparrow}{9}, 7, 9, 11\}$$

The reader can verify that the four-point circular convolution of $h(n)$ with $x(n)$ yields the same sequence $\hat{y}(n)$.

If we compare the result $\hat{y}(n)$, obtained from four-point DFTs, with the sequence $y(n)$ obtained from the use of eight-point (or six-point) DFTs, the time-domain aliasing effects derived in Section 7.2.2 are clearly evident. In particular, $y(4)$ is aliased into $y(0)$ to yield

$$\hat{y}(0) = y(0) + y(4) = 9$$

Similarly, $y(5)$ is aliased into $y(1)$ to yield

$$\hat{y}(1) = y(1) + y(5) = 7$$

All other aliasing has no effect, since $y(n) = 0$ for $n \geq 6$. Consequently, we have

$$\hat{y}(2) = y(2) = 9$$

$$\hat{y}(3) = y(3) = 11$$

Therefore, only the first two points of $\hat{y}(n)$ are corrupted by the effect of aliasing [i.e., $\hat{y}(0) \neq y(0)$ and $\hat{y}(1) \neq y(1)$]. This observation has important ramifications in the discussion of the following section, in which we treat the filtering of long sequences.

7.3.2 Filtering of Long Data Sequences

In practical applications involving linear filtering of signals, the input sequence $x(n)$ is often a very long sequence. This is especially true in some real-time signal processing applications concerned with signal monitoring and analysis.

Since linear filtering performed via the DFT involves operations on a block of data, which by necessity must be limited in size due to limited memory of a digital computer, a long input signal sequence must be segmented to fixed-size blocks prior to processing. Since the filtering is linear, successive blocks can be processed one at a time via the DFT, and the output blocks are fitted together to form the overall output signal sequence.

We now describe two methods for linear FIR filtering a long sequence on a block-by-block basis using the DFT. The input sequence is segmented into blocks and each block is processed via the DFT and IDFT to produce a block of output data. The output blocks are fitted together to form an overall output sequence which is identical to the sequence obtained if the long block had been processed via time-domain convolution.

The two methods are called the *overlap-save method* and the *overlap-add method.* For both methods we assume that the FIR filter has duration M. The input data sequence is segmented into blocks of L points, where, by assumption, $L >> M$ without loss of generality.

Overlap-save method. In this method the size of the input data blocks is $N = L + M - 1$ and the DFTs and IDFT are of length N. Each data block consists of the last $M-1$ data points of the previous data block followed by L new data points to form a data sequence of length $N = L+M-1$. An N-point DFT is computed for each data block. The impulse response of the FIR filter is increased in length by appending $L-1$ zeros and an N-point DFT of the sequence is computed once and stored. The multiplication of the two N-point DFTs $\{H(k)\}$ and $\{X_m(k)\}$ for the mth block of data yields

$$\hat{Y}_m(k) = H(k)X_m(k), \qquad k = 0, 1, \ldots, N-1 \tag{7.3.7}$$

Then the N-point IDFT yields the result

$$\hat{Y}_m(n) = \{\hat{y}_m(0)\hat{y}_m(1)\cdots\hat{y}_m(M-1)\hat{y}_m(M)\cdots\hat{y}_m(N-1)\} \tag{7.3.8}$$

Since the data record is of length N, the first $M-1$ points of $y_m(n)$ are corrupted by aliasing and must be discarded. The last L points of $y_m(n)$ are exactly the same as the result from linear convolution and, as a consequence,

$$\hat{y}_m(n) = y_m(n), n = M, M+1, \ldots, N-1 \tag{7.3.9}$$

To avoid loss of data due to aliasing, the last $M-1$ points of each data record are saved and these points become the first $M-1$ data points of the subsequent record, as indicated above. To begin the processing, the first $M-1$ points of the first record are set to zero. Thus the blocks of data sequences are

$$x_1(n) = \{\underbrace{0, 0, \ldots, 0}_{M-1 \text{ points}}, x(0), x(1), \ldots, x(L-1)\} \tag{7.3.10}$$

$$x_2(n) = \{\underbrace{x(L-M+1), \ldots, x(L-1)}_{M-1 \text{data points from } x_1(n)}, \underbrace{x(L), \ldots, x(2L-1)}_{L \text{ new data points}}\} \tag{7.3.11}$$

$$x_3(n) = \{\underbrace{x(2L-M+1), \ldots, x(2L-1)}_{M-1 \text{ data points from } x_2(n)}, \underbrace{x(2L), \ldots, x(3L-1)}_{L \text{ new data points}}\} \tag{7.3.12}$$

and so forth. The resulting data sequences from the IDFT are given by (7.3.8), where the first $M-1$ points are discarded due to aliasing and the remaining L points constitute the desired result from linear convolution. This segmentation of the input data and the fitting of the output data blocks together to form the output sequence are graphically illustrated in Fig. 7.3.1.

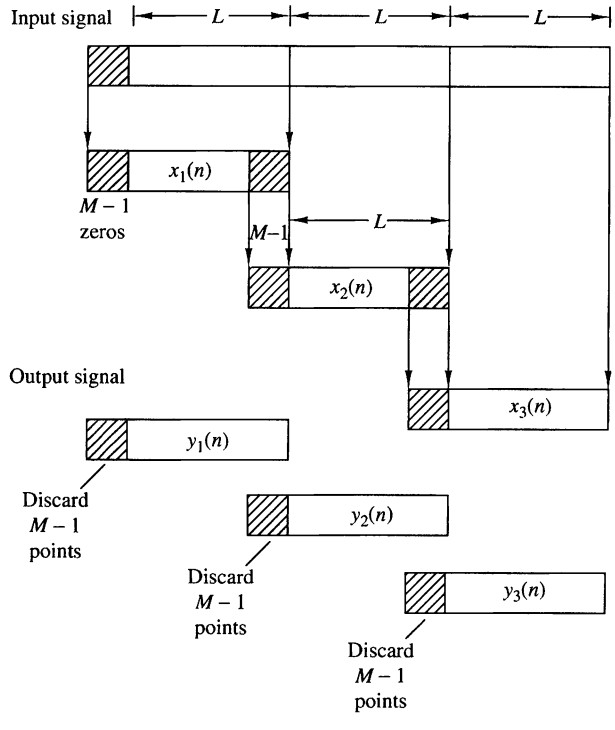


Figure 7.3.1
Linear FIR filtering by the overlap-save method.

Overlap-add method. In this method the size of the input data block is L points and the size of the DFTs and IDFT is $N = L+M-1$. To each data block we append $M-1$ zeros and compute the N-point DFT. Thus the data blocks may be represented as

$$x_1(n) = \{x(0), x(1), \ldots, x(L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\} \tag{7.3.13}$$

$$x_2(n) = \{x(L), x(L+1), \ldots, x(2L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\} \tag{7.3.14}$$

$$x_3(n) = \{x(2L), \ldots, x(3L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\} \tag{7.3.15}$$

and so on. The two N-point DFTs are multiplied together to form

$$Y_m(k) = H(k)X_m(k), \qquad k = 0, 1, \ldots, N-1 \tag{7.3.16}$$

The IDFT yields data blocks of length N that are free of aliasing, since the size of the DFTs and IDFT is $N = L + M - 1$ and the sequences are increased to N-points by appending zeros to each block.

Since each data block is terminated with $M - 1$ zeros, the last $M - 1$ points from each output block must be overlapped and added to the first $M - 1$ points of

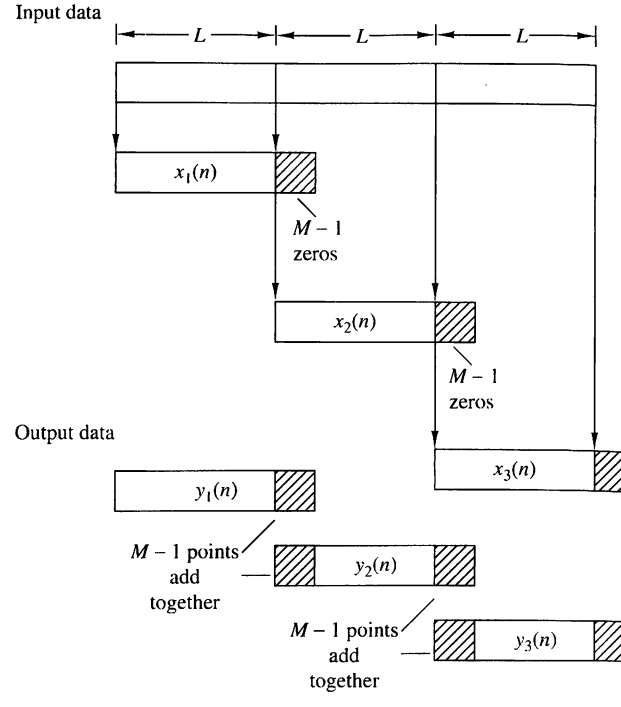


Figure 7.3.2
Linear FIR filtering by the overlap-add method.

the succeeding block. Hence this method is called the overlap-add method. This overlapping and adding yields the output sequence

$$\begin{aligned} y(n) = \{&y_1(0), y_1(1), \ldots, y_1(L-1), y_1(L) + y_2(0), y_1(L+1) \\ &+ y_2(1), \ldots, y_1(N-1) + y_2(M-1), y_2(M), \ldots\} \end{aligned} \tag{7.3.17}$$

The segmentation of the input data into blocks and the fitting of the output data blocks to form the output sequence are graphically illustrated in Fig. 7.3.2.

At this point, it may appear to the reader that the use of the DFT in linear FIR filtering not only is an indirect method of computing the output of an FIR filter, but also may be more expensive computationally, since the input data must first be converted to the frequency domain via the DFT, multiplied by the DFT of the FIR filter, and finally, converted back to the time domain via the IDFT. On the contrary, however, by using the fast Fourier transform algorithm, as will be shown in Chapter 8, the DFTs and IDFT require fewer computations to compute the output sequence than the direct realization of the FIR filter in the time domain. This computational efficiency is the basic advantage of using the DFT to compute the output of an FIR filter.

7.4 Frequency Analysis of Signals Using the DFT

To compute the spectrum of either a continuous-time or discrete-time signal, the values of the signal for all time are required. However, in practice, we observe signals for only a finite duration. Consequently, the spectrum of a signal can only be

approximated from a finite data record. In this section we examine the implications of a finite data record in frequency analysis using the DFT.

If the signal to be analyzed is an analog signal, we would first pass it through an antialiasing filter and then sample it at a rate $F_s \geq 2B$, where B is the bandwidth of the filtered signal. Thus the highest frequency that is contained in the sampled signal is $F_s/2$. Finally, for practical purposes, we limit the duration of the signal to the time interval $T_0 = LT$, where L is the number of samples and T is the sample interval. As we shall observe in the following discussion, the finite observation interval for the signal places a limit on the frequency resolution; that is, it limits our ability to distinguish two frequency components that are separated by less than $1/T_0 = 1/LT$ in frequency.

Let $\{x(n)\}$ denote the sequence to be analyzed. Limiting the duration of the sequence to L samples, in the interval $0 \leq n \leq L-1$, is equivalent to multiplying $\{x(n)\}$ by a rectangular window $w(n)$ of length L. That is,

$$\hat{x}(n) = x(n)w(n) \tag{7.4.1}$$

where

$$w(n) = \begin{cases} 1, & 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases} \tag{7.4.2}$$

Now suppose that the sequence $x(n)$ consists of a single sinusoid, that is,

$$x(n) = \cos \omega_0 n \tag{7.4.3}$$

Then the Fourier transform of the finite-duration sequence $x(n)$ can be expressed as

$$\hat{X}(\omega) = \frac{1}{2}[W(\omega - \omega_0) + W(\omega + \omega_0)] \tag{7.4.4}$$

where $W(\omega)$ is the Fourier transform of the window sequence, which is (for the rectangular window)

$$W(\omega) = \frac{\sin(\omega L/2)}{\sin(\omega/2)} e^{-j\omega(L-1)/2} \tag{7.4.5}$$

To compute $\hat{X}(\omega)$ we use the DFT. By padding the sequence $\hat{x}(n)$ with $N-L$ zeros, we can compute the N-point DFT of the truncated (L points) sequence $\{\hat{x}(n)\}$. The magnitude spectrum $|\hat{X}(k)| = |\hat{X}(\omega_k)|$ for $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N$, is illustrated in Fig. 7.4.1 for $L = 25$ and $N = 2048$. We note that the windowed spectrum $\hat{X}(\omega)$ is not localized to a single frequency, but instead it is spread out over the whole frequency range. Thus the power of the original signal sequence $\{x(n)\}$ that was concentrated at a single frequency has been spread by the window into the entire frequency range. We say that the power has "leaked out" into the entire frequency range. Consequently, this phenomenon, which is a characteristic of windowing the signal, is called *leakage*.

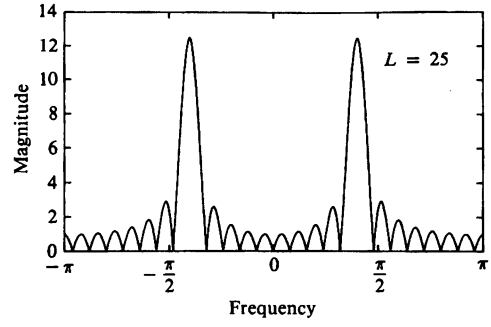


Figure 7.4.1
Magnitude spectrum for $L = 25$ and $N = 2048$, illustrating the occurrence of leakage.

Windowing not only distorts the spectral estimate due to the leakage effects, it also reduces spectral resolution. To illustrate this problem, let us consider a signal sequence consisting of two frequency components,

$$x(n) = \cos \omega_1 n + \cos \omega_2 n \tag{7.4.6}$$

When this sequence is truncated to L samples in the range $0 \leq n \leq L - 1$, the windowed spectrum is

$$\hat{X}(\omega) = \frac{1}{2}[W(\omega - \omega_1) + W(\omega - \omega_2) + W(\omega + \omega_1) + W(\omega + \omega_2)] \tag{7.4.7}$$

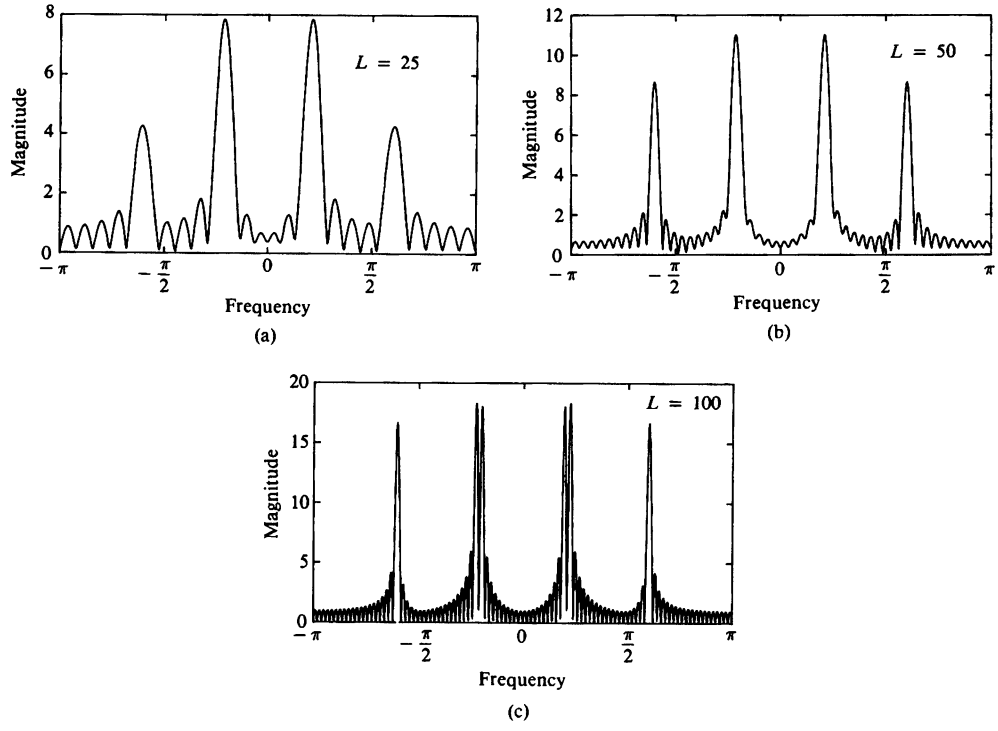


Figure 7.4.2 Magnitude spectrum for the signal given by (7.4.8), as observed through a rectangular window.

The spectrum $W(\omega)$ of the rectangular window sequence has its first zero crossing at $\omega = 2\pi/L$. Now if $|\omega_1 - \omega_2| < 2\pi/L$, the two window functions $W(\omega - \omega_1)$ and $W(\omega - \omega_2)$ overlap and, as a consequence, the two spectral lines in $x(n)$ are not distinguishable. Only if $(\omega_1 - \omega_2) \geq 2\pi/L$ will we see two separate lobes in the spectrum $\hat{X}(\omega)$. Thus our ability to resolve spectral lines of different frequencies is limited by the window main lobe width. Figure 7.4.2 illustrates the magnitude spectrum $|\hat{X}(\omega)|$, computed via the DFT, for the sequence

$$x(n) = \cos \omega_0 n + \cos \omega_1 n + \cos \omega_2 n \tag{7.4.8}$$

where $\omega_0 = 0.2\pi$, $\omega_1 = 0.22\pi$, and $\omega_2 = 0.6\pi$. The window lengths selected are $L = 25, 50$, and 100. Note that ω_0 and ω_1 are not resolvable for $L = 25$ and 50, but they are resolvable for $L = 100$.

To reduce leakage, we can select a data window $w(n)$ that has lower sidelobes in the frequency domain compared with the rectangular window. However, as we describe in more detail in Chapter 10, a reduction of the sidelobes in a window $W(\omega)$ is obtained at the expense of an increase in the width of the main lobe of $W(\omega)$ and hence a loss in resolution. To illustrate this point, let us consider the Hanning window, which is specified as

$$w(n) = \begin{cases} \frac{1}{2}(1 - \cos \frac{2\pi}{L-1} n), & 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases} \tag{7.4.9}$$

Figure 7.4.3 shows $|\hat{X}(\omega)|$ given by (7.4.4) for the window of (7.4.9). Its sidelobes are significantly smaller than those of the rectangular window, but its main lobe is approximately twice as wide. Figure 7.4.4 shows the spectrum of the signal in (7.4.8), after it is windowed by the Hanning window, for $L = 50, 75$, and 100. The reduction of the sidelobes and the decrease in the resolution, compared with the rectangular window, is clearly evident.

For a general signal sequence $\{x(n)\}$, the frequency-domain relationship between the windowed sequence $\hat{x}(n)$ and the original sequence $x(n)$ is given by the convolution formula

$$\hat{X}(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(\theta) W(\omega - \theta) d\theta \tag{7.4.10}$$

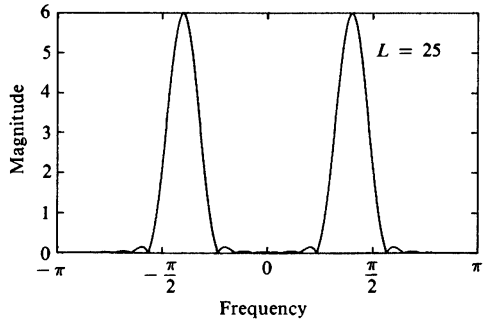


Figure 7.4.3
Magnitude spectrum of the Hanning window.

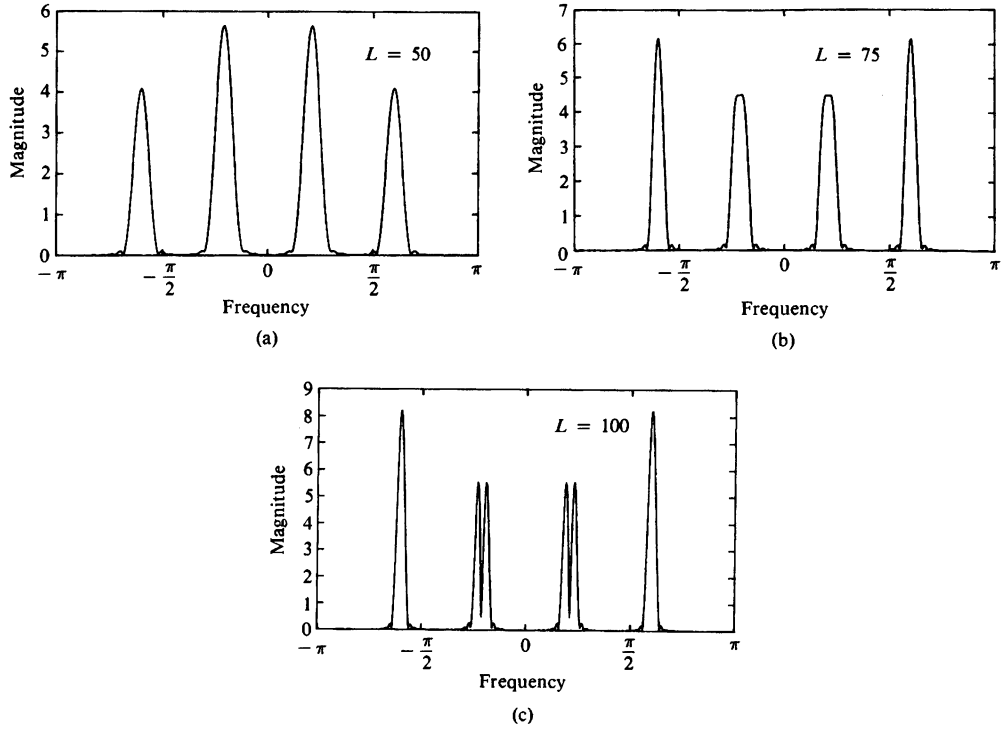


Figure 7.4.4 Magnitude spectrum of the signal in (7.4.8) as observed through a Hanning window.

The DFT of the windowed sequence $\hat{x}(n)$ is the sampled version of the spectrum $\hat{X}(\omega)$. Thus we have

$$\begin{aligned}\hat{X}(k) &\equiv \hat{X}(\omega)|_{\omega=2\pi k/N} \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\theta)W\left(\frac{2\pi k}{N}-\theta\right)d\theta, \qquad k=0,1,\ldots,N-1\end{aligned} \tag{7.4.11}$$

Just as in the case of the sinusoidal sequence, if the spectrum of the window is relatively narrow in width compared to the spectrum $X(\omega)$ of the signal, the window function has only a small (smoothing) effect on the spectrum $X(\omega)$. On the other hand, if the window function has a wide spectrum compared to the width of $X(\omega)$, as would be the case when the number of samples L is small, the window spectrum masks the signal spectrum and, consequently, the DFT of the data reflects the spectral characteristics of the window function. Of course, this situation should be avoided.

EXAMPLE 7.4.1

The exponential signal

$$x_a(t) = \begin{cases} e^{-t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$$

is sampled at the rate $F_s = 20$ samples per second, and a block of 100 samples is used to

estimate its spectrum. Determine the spectral characteristics of the signal $x_a(t)$ by computing the DFT of the finite-duration sequence. Compare the spectrum of the truncated discrete-time signal to the spectrum of the analog signal.

Solution. The spectrum of the analog signal is

$$X_a(F) = \frac{1}{1 + j2\pi F}$$

The exponential analog signal sampled at the rate of 20 samples per second yields the sequence

$$\begin{aligned} x(n) &= e^{-nT} = e^{-n/20}, \qquad n \geq 0 \\ &= (e^{-1/20})^n = (0.95)^n, \quad n \geq 0 \end{aligned}$$

Now, let

$$x(n) = \begin{cases} (0.95)^n, & 0 \leq n \leq 99 \\ 0, & \text{otherwise} \end{cases}$$

The N-point DFT of the $L = 100$ point sequence is

$$\hat{X}(k) = \sum_{k=0}^{99} \hat{x}(n) e^{-j2\pi k/N}, \qquad k = 0, 1, \ldots, N-1$$

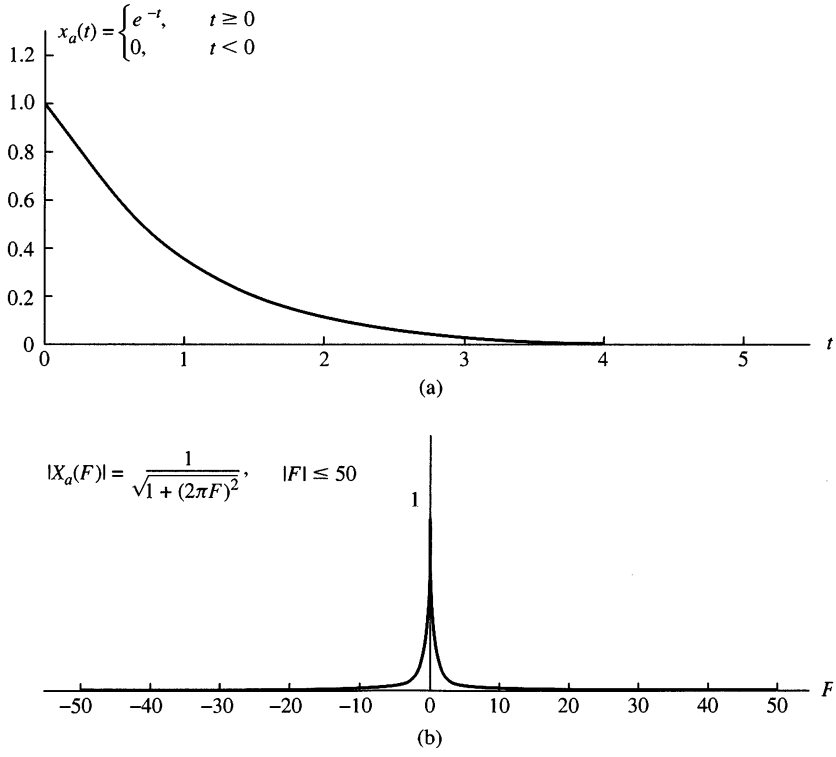
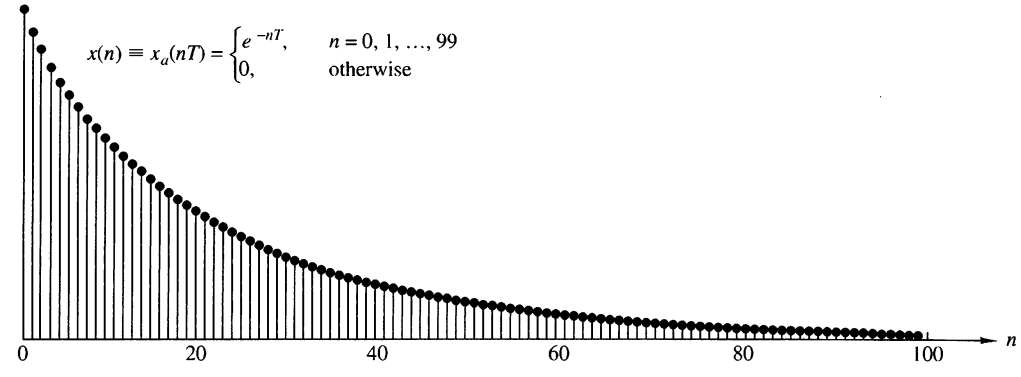
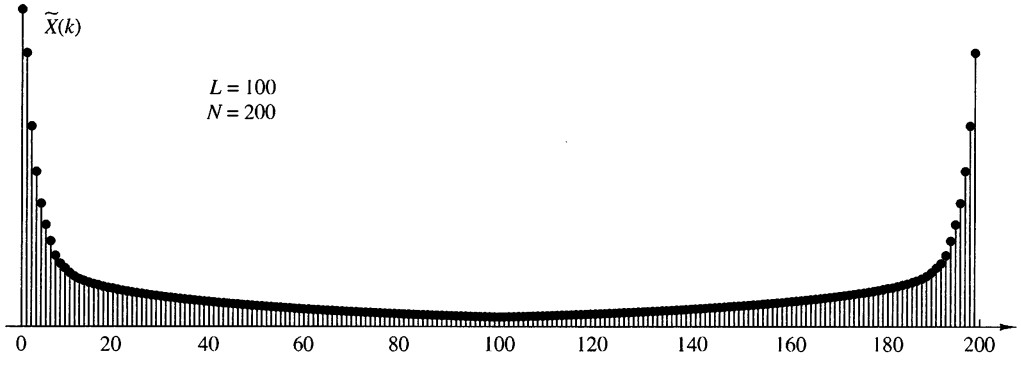


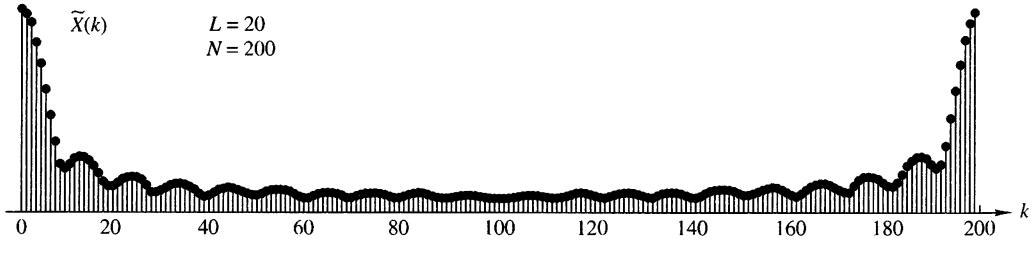
Figure 7.4.5 Effect of windowing (truncating) the sampled version of the analog signal in Example 7.4.1



(c)



(d)



(e)

Figure 7.4.5 *Continued*

To obtain sufficient detail in the spectrum we choose $N = 200$. This is equivalent to padding the sequence $x(n)$ with 100 zeros.

The graph of the analog signal $x_a(t)$ and its magnitude spectrum $|X_a(F)|$ are illustrated in Fig. 7.4.5(a) and 7.4.5(b), respectively. The truncated sequence $x(n)$ and its $N = 200$ point DFT (magnitude) are illustrated in Fig. 7.4.5(c) and 7.4.5(d), respectively. In this case the DFT $\{X(k)\}$ bears a close resemblance to the spectrum of the analog signal. The effect of the window function is relatively small.

On the other hand, suppose that a window function of length $L = 20$ is selected. Then the truncated sequence $x(n)$ is given as

$$\hat{x}(n) = \begin{cases} (0.95)^n, & 0 \le n \le 19 \\ 0, & \text{otherwise} \end{cases}$$

Its $N = 200$-point DFT is illustrated in Fig. 7.4.5(e). Now the effect of the wider spectral window function is clearly evident. First, the main peak is very wide as a result of the wide spectral window. Second, the sinusoidal envelope variations in the spectrum away from the main peak are due to the large sidelobes of the rectangular window spectrum. Consequently, the DFT is no longer a good approximation of the analog signal spectrum.

7.5 The Discrete Cosine Transform

The DFT represents an N-point sequence $x(n)$, $0 \le n \le N-1$, as a linear combination of complex exponentials. As a result, the DFT coefficients are, in general, complex even if $x(n)$ is real. Suppose that we wish to find an $N \times N$ orthogonal transform that expresses a real sequence $x(n)$ as a linear combination of cosine sequences. From (7.2.25) and (7.2.26), we see that this is possible if the N-point sequence $x(n)$ is real and even, that is, $x(n) = x(N-n)$, $0 \le n \le N-1$. The resulting DFT, $X(k)$, is itself real and even. This observation suggests that we could possibly derive a discrete cosine transform for any N-point real sequence by taking the $2N$-point DFT of an "even extension" of the sequence. Because there are eight ways to do this even extension, there are as many definitions of the DCT (Wang 1984, Martucci 1994). We discuss a version known as DCT-II, which is widely used in practice for speech and image compression applications as part of various standards (Rao and Huang, 1996). For simplicity, we will use the term DCT to refer to DCT-II.

7.5.1 Forward DCT

Let $s(n)$ be a $2N$-point even symmetric extension of $x(n)$ defined by

$$s(n) = \begin{cases} x(n), & 0 \le n \le N-1 \\ x(2N-n-1), & N \le n \le 2N-1 \end{cases} \tag{7.5.1}$$

The sequence $s(n)$ has even symmetry about the "half-sample" point $n = N + (1/2)$ (see Figure 7.5.1). The $2N$-point DFT of $s(n)$ is given by

$$S(k) = \sum_{n=0}^{2N-1} s(n) W_{2N}^{nk}, \quad 0 \le k \le 2N-1 \tag{7.5.2}$$

Substitution of (7.5.1) in (7.5.2) yields

$$S(k) = \sum_{n=0}^{N-1} x(n) W_{2N}^{nk} + \sum_{n=N}^{2N-1} x(2N-n-1) W_{2N}^{nk} \tag{7.5.3}$$

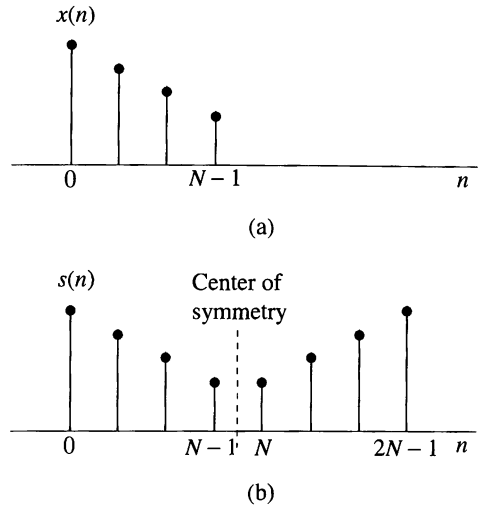


Figure 7.5.1
Original sequence $x(n)$, $0 \le n \le N-1$ and its $2N$-point even extension $s(n)$, $0 \le n \le 2N-1$.

If we change the second index of summation using $n = 2N - 1 - m$, we recall that $W_{2N}^{2mN} = 1$ for integer m, and we factor out $W_{2N}^{-k/2}$, we obtain

$$S(k) = W_{2N}^{-k/2} \sum_{n=0}^{N-1} x(n) \left[W_{2N}^{nk} W_{2N}^{k/2} + W_{2N}^{-nk} W_{2N}^{-k/2} \right], \quad 0 \le k \le 2N-1 \tag{7.5.4}$$

The last expression may be written as

$$S(k) = W_{2N}^{-k/2} 2 \sum_{n=0}^{N-1} x(n) \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right], \quad 0 \le k \le 2N-1 \tag{7.5.5}$$

or equivalently

$$S(k) = W_{2N}^{-k/2} 2\Re \left[W_{2N}^{k/2} \sum_{n=0}^{N-1} x(n) W_{2N}^{kn} \right], \quad 0 \le k \le 2N-1 \tag{7.5.6}$$

If we define the forward DCT by

$$V(k) = 2 \sum_{n=0}^{N-1} x(n) \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right], \quad 0 \le k \le N-1 \tag{7.5.7}$$

we can easily show that

$$V(k) = W_{2N}^{k/2} S(k) \text{ or } S(k) = W_{2N}^{-k/2} V(k), \quad 0 \le k \le N-1 \tag{7.5.8}$$

and

$$V(k) = 2\Re \left[W_{2N}^{k/2} \sum_{n=0}^{N-1} x(n) W_{2N}^{kn} \right], \quad 0 \le k \le N-1 \tag{7.5.9}$$

We note that $V(k)$ is real and $S(k)$ is complex. $S(k)$ is complex because the real sequence $s(n)$ satisfies the symmetry relation $s(2N-1-n) = s(n)$ instead of $s(2N-n) = s(n)$.

The DCT of $x(n)$ can be computed by taking the $2N$-point DFT of $s(n)$, as in (7.5.2), and multiplying the result by $W_{2N}^{k/2}$, as in (7.5.8). Another approach, suggested by (7.5.9), is to take the $2N$-point DFT of the original sequence $x(n)$ with N zeros appended to it, multiply the result by $W_{2N}^{k/2}$, and then take twice the real part.

7.5.2 Inverse DCT

We shall derive the inverse DCT from the inverse DFT of the even extended sequence $s(n)$. The inverse DFT of $S(k)$ is given by

$$s(n) = \frac{1}{2N} \sum_{k=0}^{2N-1} S(k) W_{2N}^{-nk} \tag{7.5.10}$$

Since $s(n)$ is real, $S(k)$ is Hermitian symmetric, that is,

$$S(2N-k) = S^*(k) \tag{7.5.11}$$

Furthermore, from (7.5.7), it is easy to show that

$$S(N) = 0 \tag{7.5.12}$$

With the help of (7.5.11) and (7.5.12), (7.5.10) yields

$$\begin{aligned} s(n) &= \frac{1}{2N} \sum_{k=0}^{N-1} S(k) W_{2N}^{-kn} + \frac{1}{2N} \sum_{k=N}^{2N-1} S(k) W_{2N}^{-kn} \\ &= \frac{1}{2N} \sum_{k=0}^{N-1} S(k) W_{2N}^{-kn} + \frac{1}{2N} \sum_{m=1}^{N} S(2N-m) W_{2N}^{-(2N-m)n} \\ &= \frac{1}{2N} S(0) + \frac{1}{2N} \sum_{k=1}^{N-1} S(k) W_{2N}^{-kn} + \frac{1}{2N} \sum_{k=1}^{N-1} S^*(k) W_{2N}^{kn} \end{aligned}$$

or since $S(0)$ is real

$$s(n) = \frac{1}{N} \Re \left[\frac{S(0)}{2} + \sum_{k=1}^{N-1} S(k) W_{2N}^{-kn} \right], \quad 0 \le n \le 2N-1 \tag{7.5.13}$$

Substituting (7.5.8) in (7.5.13) and using (7.5.1) yields the desired inverse DCT

$$x(n) = \frac{1}{N} \left\{ \frac{V(0)}{2} + \sum_{k=1}^{N-1} V(k) \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right] \right\}, \quad 0 \le n \le N-1 \tag{7.5.14}$$

Given $V(k)$, we first compute $S(k)$ by (7.5.8). In the next step, we take the $2N$-point inverse DFT implied by (7.5.13). The real part of this inverse DFT yields $s(n)$ and, hence $x(n)$.

An approach to compute the DCT and inverse DCT using an N-point DFT is discussed in Makhoul (1980). Many special algorithms for the hardware and software implementation of the DCT are discussed in Rao and Yip (1990).

7.5.3 DCT as an Orthogonal Transform

Equations (7.5.7) and (7.5.14) form a DCT pair. However, for reasons to be seen later, we usually redistribute symmetrically the normalization factors between the forward and inverse transform. Thus, the DCT of the sequence $x(n)$, $0 \le n \le N-1$ and its inverse are defined by

$$C(k) = \alpha(k) \sum_{n=0}^{N-1} x(n) \cos\left[\frac{\pi(2n+1)k}{2N}\right], \qquad 0 \le k \le N-1 \tag{7.5.15}$$

$$x(n) = \sum_{k=0}^{N-1} \alpha(k) C(k) \cos\left[\frac{\pi(2n+1)k}{2N}\right], \qquad 0 \le n \le N-1 \tag{7.5.16}$$

where

$$\alpha(0) = \sqrt{\frac{1}{N}}, \quad \alpha(k) = \sqrt{\frac{2}{N}} \text{ for } 1 \le k \le N-1 \tag{7.5.17}$$

Like the DFT in Section 7.1.3, the DCT formulas (7.5.15) and (7.5.16) can be expressed in matrix form using the $N \times N$ DCT matrix $\boldsymbol{C}_N$ with elements given by

$$c_{kn} = \begin{cases} \dfrac{1}{\sqrt{N}}, & k = 0, 0 \le n \le N-1 \\ \sqrt{\dfrac{2}{N}} \cos\dfrac{\pi(2n+1)k}{2N}, & 1 \le k \le N-1, 0 \le n \le N-1 \end{cases} \tag{7.5.18}$$

If we define the following signal and coefficient vectors

$$\mathbf{x}_N = [\, x(0) \quad x(1) \quad \ldots \quad x(N-1) \,]^T \tag{7.5.19}$$

$$\mathbf{c}_N = [\, C(0) \quad C(1) \quad \ldots \quad C(N-1) \,]^T \tag{7.5.20}$$

the forward DCT (7.5.15) and the inverse DCT (7.5.16) can be written in matrix form as

$$\mathbf{c}_N = \mathbf{C}_N \mathbf{x}_N \tag{7.5.21}$$

$$\mathbf{x}_N = \mathbf{C}_N^T \mathbf{c}_N \tag{7.5.22}$$

It follows from (7.5.19) and (7.5.20) that $\mathbf{C}_N$ is a real orthogonal matrix, that is, it satisfies

$$\mathbf{C}_N^{-1} = \mathbf{C}_N^T \tag{7.5.23}$$

Orthogonality simplifies the computation of the inverse transform because it replaces matrix inversion by matrix transposition.

If we denote by $\mathbf{c}_N(k)$ the columns of $\mathbf{C}_N^T$, the inverse DCT can be written as

$$\mathbf{x}_N = \sum_{k=0}^{N-1} C(k)\mathbf{c}_N(k) \tag{7.5.24}$$

which represents the signal as a linear combination of the DCT cosine basis sequences. The value of the coefficient $C(k)$ measures the similarity of the signal with the kth basis vector.

EXAMPLE 7.5.1

Consider the discrete-time sinusoidal signal

$$x(n) = \cos(2\pi k_0 n/N), \quad 0 \le n \le N-1$$

In Figure 7.5.2 we show plots of the sequence $x(n)$, the absolute values of the N-point DFT $X(k)$ coefficients, and the N-point DCT coefficients for $k_0 = 5$ and $N = 32$. We note that, in contrast to the DFT, the DCT, although it shows a distinct peak at $2k_0$, also exhibits a significant amount of ripples at other frequencies. For this reason, the DCT is not useful for frequency analysis of signals and systems.

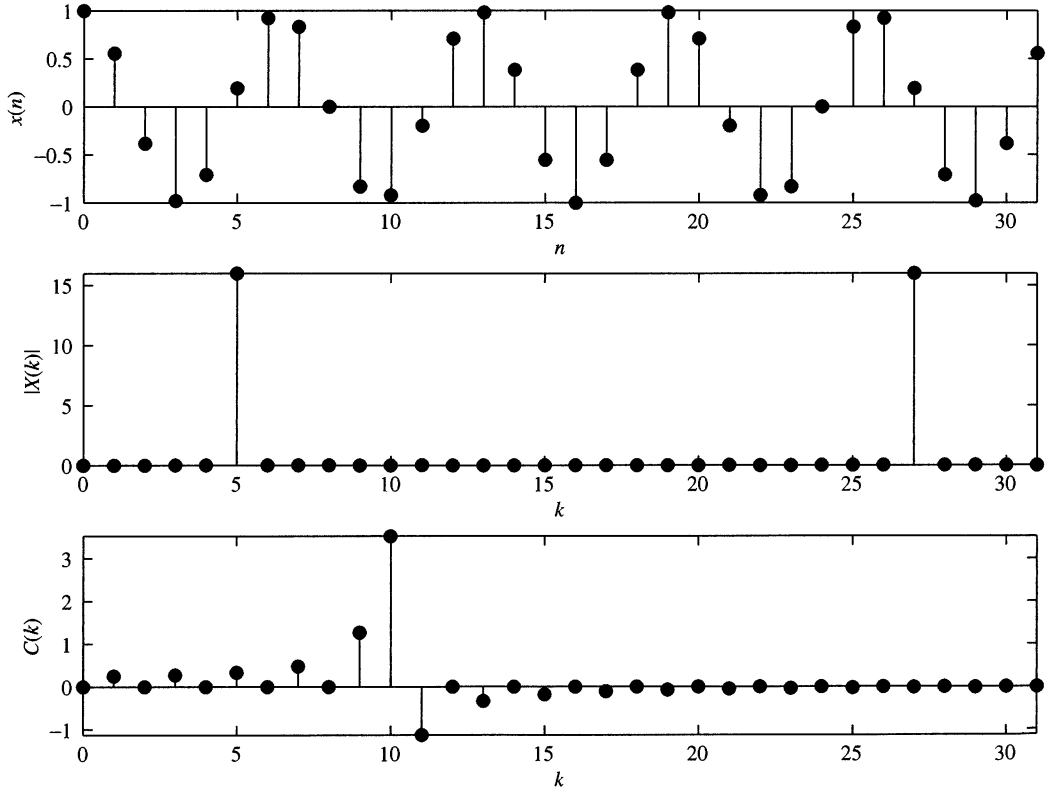


Figure 7.5.2 A discrete-time sinusoidal signal and its DFT and DCT representations.

Using the orthogonality property (7.5.23), we can easily show that

$$\sum_{k=0}^{N-1} |C(k)|^2 = \mathbf{c}_N^T \mathbf{c}_N = \mathbf{x}_N^T \mathbf{C}_N^T \mathbf{C}_N \mathbf{x}_N = \mathbf{x}_N^T \mathbf{x}_N = \sum_{n=0}^{N-1} |x(n)|^2 = E_x \tag{7.5.25}$$

Thus, an orthogonal transformation preserves the signal energy, or equivalently, the length of the vector $\mathbf{x}$ in the N-dimensional vector space (generalized Parseval's theorem). This means that every orthogonal transformation is simply a rotation of the vector $\mathbf{x}$ in the N-dimensional vector space.

Most orthogonal transforms tend to pack a large fraction of the average energy of the signal into a relatively few components of the transform coefficients (energy compaction property). Since the total energy is preserved, many of the transform coefficients will contain very little energy. However, as we illustrate in the next example, different transforms have different energy compaction capabilities.

EXAMPLE 7.5.2

We will compare the energy compaction capabilities of the DFT and DCT using the ramp sequence $x(n) = n$, $0 \leq n \leq N-1$, shown in Figure 7.5.3(a) for $N = 32$. Figures 7.5.3(d) and 7.5.3(f) show the absolute values of the DFT coefficients and the values of the DCT coefficients, respectively. Clearly, the DCT coefficients demonstrate better "energy packing" than the DFT ones. This implies that we should be able to represent the sequence $x(n)$ using a smaller number of DCT coefficients.

With the DCT we set the last k_0 coefficients to zero and take the inverse DCT to obtain an approximation $x_{\text{DCT}}(n)$ of the original sequence. However, since the DFT of a real sequence is complex, the information is carried in the first $N/2$ values. (For simplicity, we assume that N is an even number.) Therefore, we should remove DFT coefficients in a way that preserves the complex-conjugate symmetry. This is done by first removing the coefficient $X(N/2)$, then the coefficients $X(N/2-1)$ and $X(N/2+1)$, and so forth. Clearly, we can remove only an odd number of DFT coefficients, that is, $k_0 = 1, 3, \ldots, N-1$. The reconstructed sequence using the DFT is denoted by $x_{\text{DFT}}(n)$.

The reconstruction error for the DCT, which is a function of k_0, is defined by

$$E_{\text{DCT}}(k_0) = \frac{1}{N} \sum_{n=0}^{N-1} |x(n) - x_{\text{DCT}}(n)|^2$$

A similar definition is used for the DFT. Figure 7.5.3(b) shows the reconstruction errors for the DFT and DCT as a function of the number k_0 of omitted coefficients. Figures 7.5.3(c) and 7.5.3(e) show the reconstructed signals when we retain $N - k_0 = 5$ coefficients. We see that we need fewer DCT coefficients than DFT coefficients to get a good approximation of the original signal. In this example, the DFT (due to its inherent periodicity) is trying to model a sawtooth wave. Therefore, it has to devote many high-frequency coefficients to approximate the discontinuities at the ends. In contrast, the DCT operates on the "even extension" of $x(n)$, which is a triangular wave with no discontinuities. As a result, the DCT can approximate better small blocks of signals that have fairly different values in the first and last samples.

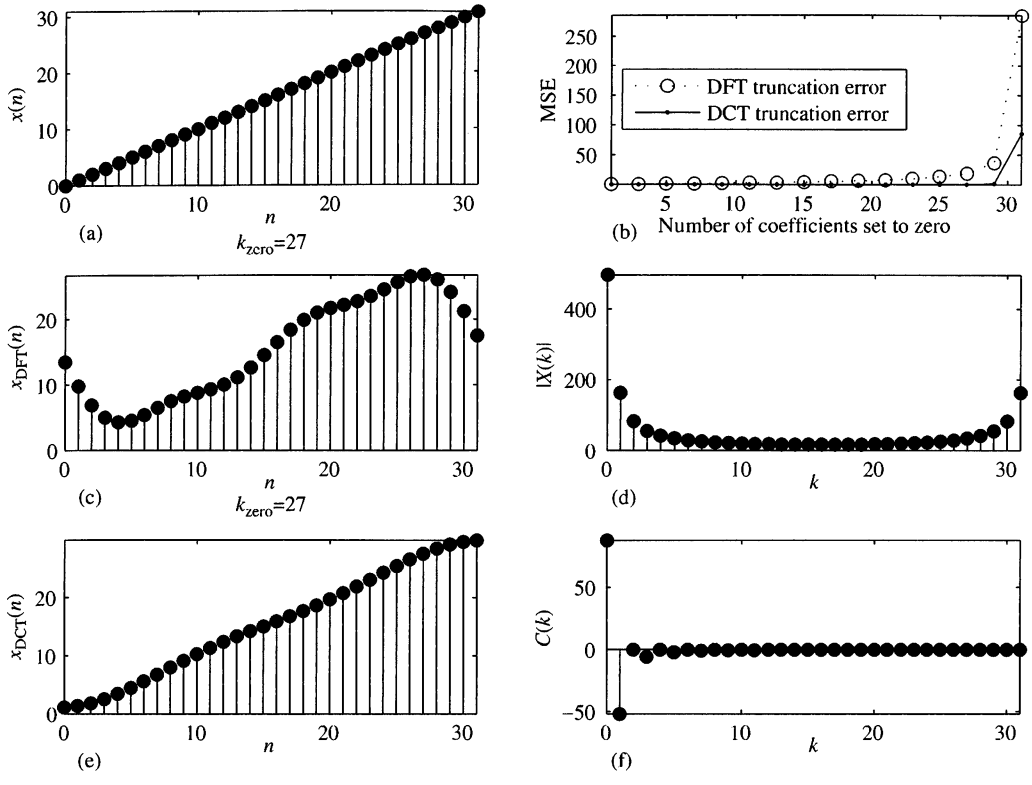


Figure 7.5.3 A discrete time sinusoidal signal and its DFT and DCT representations.

From a statistical viewpoint, the optimum orthogonal transform for signal compression is the Karhunen–Loeve (KL) or Hotelling transform (Jayant and Noll, 1984). The KL transform has two optimality properties: (a) it minimizes the reconstruction error for any number of retained coefficients, and (b) it generates a set of uncorrelated transform coefficients. The KL transform is defined by the eigenvectors of the covariance matrix of the input sequence. The DCT provides a good approximation to the KL transform for signals that follow the difference equation $x(n) = ax(n-1)+w(n)$, where $w(n)$ is a white noise sequence, and a $(0 < a < 1)$ is a constant coefficient with values close to one. Many signals, including natural images, have this characteristic. More details about orthogonal transforms and their applications can be found in Jayant and Noll (1984), Clarke (1985), Rao and Yip (1990), and Goyal (2001).

7.6 Summary and References

The major focus of this chapter was on the discrete Fourier transform, its properties and its applications. We developed the DFT by sampling the spectrum $X(\omega)$ of the sequence $x(n)$.

Frequency-domain sampling of the spectrum of a discrete-time signal is particularly important in the processing of digital signals. Of particular significance is the DFT, which was shown to uniquely represent a finite-duration sequence in the frequency domain. The existence of computationally efficient algorithms for the

DFT, which are described in Chapter 8, make it possible to digitally process signals in the frequency domain much faster than in the time domain. The processing methods in which the DFT is especially suitable include linear filtering as described in this chapter and correlation and spectrum analysis, which are treated in Chapters 8 and 4. A particularly lucid and concise treatment of the DFT and its application to frequency analysis is given in the book by Brigham (1988).

We also described the discrete cosine transform (DCT) in this chapter. An interesting treatment of the DCT from a linear algebra perspective is given in a paper by Strang (1999).

Problems

7.1 The first five points of the eight-point DFT of a real-valued sequence are $\{0.25, 0.125 - j0.3018, 0, 0.125 - j0.0518, 0\}$. Determine the remaining three points.

7.2 Compute the eight-point circular convolution for the following sequences.

(a) $x_1(n) = \{1, 1, 1, 1, 0, 0, 0, 0\}$

$$x_2(n) = \sin \frac{3\pi}{8} n, \qquad 0 \le n \le 7$$

(b) $x_1(n) = (\frac{1}{4})^n, \qquad 0 \le n \le 7$

$$x_2(n) = \cos \frac{3\pi}{8} n, \qquad 0 \le n \le 7$$

(c) Compute the DFT of the two circular convolution sequences using the DFTs of $x_1(n)$ and $x_2(n)$.

7.3 Let $X(k), 0 \le k \le N-1$, be the N-point DFT of the sequence $x(n), 0 \le n \le N-1$. We define

$$\hat{X}(k) = \begin{cases} X(k), & 0 \le k \le k_c, N - k_c \le k \le N-1 \\ 0, & k_c < k < N - k_c \end{cases}$$

and we compute the inverse N-point DFT of $\hat{X}(k)$, $0 \le k \le N-1$. What is the effect of this process on the sequence $x(n)$? Explain.

7.4 For the sequences

$$x_1(n) = \cos \frac{2\pi}{N} n, \qquad x_2(n) = \sin \frac{2\pi}{N} n, \qquad 0 \le n \le N-1$$

determine the N-point:

(a) Circular convolution $x_1(n) \textcircled{N} x_2(n)$

(b) Circular correlation of $x_1(n)$ and $x_2(n)$

(c) Circular autocorrelation of $x_1(n)$

(d) Circular autocorrelation of $x_2(n)$

7.5 Compute the quantity

$$\sum_{n=0}^{N-1} x_1(n) x_2(n)$$

for the following pairs of sequences.

(a) $x_1(n) = x_2(n) = \cos \dfrac{2\pi}{N} n, \qquad 0 \le n \le N-1$

(b) $x_1(n) = \cos \dfrac{2\pi}{N} n, \qquad x_2(n) = \sin \dfrac{2\pi}{N} n, \qquad 0 \le n \le N-1$

(c) $x_1(n) = \delta(n) + \delta(n-8), \qquad x_2(n) = u(n) - u(n-N)$

7.6 Determine the N-point DFT of the Blackman window

$$w(n) = 0.42 - 0.5 \cos \frac{2\pi n}{N-1} + 0.08 \cos \frac{4\pi n}{N-1}, \qquad 0 \le n \le N-1$$

7.7 If $X(k)$ is the DFT of the sequence $x(n)$, determine the N-point DFTs of the sequences

$$x_c(n) = x(n) \cos \frac{2\pi k_0 n}{N}, \qquad 0 \le n \le N-1$$

and

$$x_s(n) = x(n) \sin \frac{2\pi k_0 n}{N}, \qquad 0 \le n \le N-1$$

in terms of $X(k)$.

7.8 Determine the circular convolution of the sequences

$$x_1(n) = \{\underset{\uparrow}{1}, 2, 3, 1\}$$

$$x_2(n) = \{\underset{\uparrow}{4}, 3, 2, 2\}$$

using the time-domain formula in (7.2.39).

7.9 Use the four-point DFT and IDFT to determine the sequence

$$x_3(n) = x_1(n) \textcircled{N} x_2(n)$$

where $x_1(n)$ and $x_2(n)$ are the sequence given in Problem 7.8.

7.10 Compute the energy of the N-point sequence

$$x(n) = \cos \frac{2\pi k_0 n}{N}, \qquad 0 \le n \le N-1$$

7.11 Given the eight-point DFT of the sequence

$$x(n) = \begin{cases} 1, & 0 \le n \le 3 \\ 0, & 4 \le n \le 7 \end{cases}$$

compute the DFT of the sequences

(a) $x_1(n) = \begin{cases} 1, & n = 0 \\ 0, & 1 \le n \le 4 \\ 1, & 5 \le n \le 7 \end{cases}$

(b) $x_2(n) = \begin{cases} 0, & 0 \le n \le 1 \\ 1, & 2 \le n \le 5 \\ 0, & 6 \le n \le 7 \end{cases}$

7.12 Consider a finite-duration sequence

$$x(n) = \{\underset{\uparrow}{0}, 1, 2, 3, 4\}$$

(a) Sketch the sequence $s(n)$ with six-point DFT

$$S(k) = W_2^* X(k), \qquad k = 0, 1, \ldots, 6$$

(b) Determine the sequence $y(n)$ with six-point DFT $Y(k) = \Re|X(k)|$.

(c) Determine the sequence $v(n)$ with six-point DFT $V(k) = \Im|X(k)|$.

7.13 Let $x_p(n)$ be a periodic sequence with fundamental period N. Consider the following DFTs:

$$x_p(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_1(k)$$

$$x_p(n) \overset{\text{DFT}}{\underset{3N}{\longleftrightarrow}} X_3(k)$$

(a) What is the relationship between $X_1(k)$ and $X_3(k)$?

(b) Verify the result in part (a) using the sequence

$$x_p(n) = \{\cdots 1, 2, 1, \underset{\uparrow}{2}, 1, 2, 1, 2 \cdots\}$$

7.14 Consider the sequences

$$x_1(n) = \{\underset{\uparrow}{0}, 1, 2, 3, 4\}, \qquad x_2(n) = \{\underset{\uparrow}{0}, 1, 0, 0, 0\}, \qquad s(n) = \{\underset{\uparrow}{1}, 0, 0, 0, 0\}$$

and their five-point DFTs.

(a) Determine a sequence $y(n)$ so that $Y(k) = X_1(k)X_2(k)$.

(b) Is there a sequence $x_3(n)$ such that $S(k) = X_1(k)X_3(k)$?

7.15 Consider a causal LTI system with system function

$$H(z) = \frac{1}{1 - 0.5z^{-1}}$$

The output $y(n)$ of the system is known for $0 \leq n \leq 63$. Assuming that $H(z)$ is available, can you develop a 64-point DFT method to recover the sequence $x(n)$, $0 \leq n \leq 63$? Can you recover all values of $x(n)$ in this interval?

7.16 The impulse response of an LTI system is given by $h(n) = \delta(n) - \frac{1}{4}\delta(n - k_0)$. To determine the impulse response $g(n)$ of the inverse system, an engineer computes the N-point DFT $H(k)$, $N = 4k_0$, of $h(n)$ and then defines $g(n)$ as the inverse DFT of $G(k) = 1/H(k)$, $k = 0, 1, 2, \ldots, N - 1$. Determine $g(n)$ and the convolution $h(n) \times g(n)$, and comment on whether the system with impulse response $g(n)$ is the inverse of the system with impulse response $h(n)$.

7.17 Determine the eight-point DFT of the signal

$$x(n) = \{1, 1, 1, 1, 1, 1, 0, 0\}$$

and sketch its magnitude and phase.

7.18 A linear time-invariant system with frequency response $H(\omega)$ is excited with the periodic input

$$x(n) = \sum_{k=-\infty}^{\infty} \delta(n - kN)$$

Suppose that we compute the N-point DFT $Y(k)$ of the samples $y(n)$, $0 \leq n \leq N-1$ of the output sequence. How is $Y(k)$ related to $H(\omega)$?

7.19 *DFT of real sequences with special symmetries*

(a) Using the symmetry properties of Section 7.2 (especially the decomposition properties), explain how we can compute the DFT of two real symmetric (even) and two real antisymmetric (odd) sequences simultaneously using an N-point DFT only.

(b) Suppose now that we are given four real sequences $x_i(n)$, $i = 1, 2, 3, 4$, that are all symmetric [i.e., $x_i(n) = x_i(N - n)$, $0 \leq n \leq N - 1$]. Show that the sequences

$$s_i(n) = x_i(n + 1) - x_i(n - 1)$$

are antisymmetric [i.e., $s_i(n) = -s_i(N - n)$ and $s_i(0) = 0$].

(c) Form a sequence $x(n)$ using $x_1(n)$, $x_2(n)$, $s_3(n)$, and $s_4(n)$ and show how to compute the DFT $X_i(k)$ of $x_i(n)$, $i = 1, 2, 3, 4$ from the N-point DFT $X(k)$ of $x(n)$.

(d) Are there any frequency samples of $X_i(k)$ that cannot be recovered from $X(k)$? Explain.

7.20 *DFT of real sequences with odd harmonics only* Let $x(n)$ be an N-point real sequence with N-point DFT $X(k)$ (N even). In addition, $x(n)$ satisfies the following symmetry property:

$$x\left(n + \frac{N}{2}\right) = -x(n), \qquad n = 0, 1, \ldots, \frac{N}{2} - 1$$

that is, the upper half of the sequence is the negative of the lower half.

(a) Show that

$$X(k) = 0, \qquad k \text{ even}$$

that is, the sequence has a spectrum with odd harmonics.

(b) Show that the values of this odd-harmonic spectrum can be computed by evaluating the $N/2$-point DFT of a complex modulated version of the original sequence $x(n)$.

7.21 Let $x_a(t)$ be an analog signal with bandwidth $B = 3$ kHz. We wish to use an $N = 2^m$-point DFT to compute the spectrum of the signal with a resolution less than or equal to 50 Hz. Determine **(a)** the minimum sampling rate, **(b)** the minimum number of required samples, and **(c)** the minimum length of the analog signal record.

7.22 Consider the periodic sequence

$$x_p(n) = \cos \frac{2\pi}{10} n, \qquad -\infty < n < \infty$$

with frequency $f_0 = \frac{1}{10}$ and fundamental period $N = 10$. Determine the 10-point DFT of the sequence $x(n) = x_p(n)$, $0 \le n \le N - 1$.

7.23 Compute the N-point DFTs of the signals

(a) $x(n) = \delta(n)$

(b) $x(n) = \delta(n - n_0), \quad 0 < n_0 < N$

(c) $x(n) = a^n, \quad 0 \le n \le N - 1$

(d) $x(n) = \begin{cases} 1, & 0 \le n \le N/2 - 1 \quad (N \text{ even}) \\ 0, & N/2 \le n \le N - 1 \end{cases}$

(e) $x(n) = e^{j(2\pi/N)k_0 n}, \quad 0 \le n \le N - 1$

(f) $x(n) = \cos \frac{2\pi}{N} k_0 n, \quad 0 \le n \le N - 1$

(g) $x(n) = \sin \frac{2n}{N} k_0 n, \quad 0 \le n \le N - 1$

(h) $x(n) = \begin{cases} 1, & n \text{ even} \\ 0, & n \text{ odd}, \quad 0 \le n \le N - 1 \end{cases}$

7.24 Consider the finite-duration signal

$$x(n) = \{1, 2, 3, 1\}$$

(a) Compute its four-point DFT by solving explicitly the 4-by-4 system of linear equations defined by the inverse DFT formula.

(b) Check the answer in part (a) by computing the four-point DFT, using its definition.

7.25 **(a)** Determine the Fourier transform $X(\omega)$ of the signal

$$x(n) = \{1, 2, \underset{\uparrow}{3}, 2, 1, 0\}$$

(b) Compute the six-point DFT $V(k)$ of the signal

$$v(n) = \{3, 2, 1, 0, 1, 2\}$$

(c) Is there any relation between $X(\omega)$ and $V(k)$? Explain.

7.26 Prove the identity

$$\sum_{l=-\infty}^{\infty} \delta(n + lN) = \frac{1}{N} \sum_{k=0}^{N-1} e^{j(2\pi/N)kn}$$

(*Hint:* Find the DFT of the periodic signal in the left-hand side.)

7.27 *Computation of the even and odd harmonics using the DFT.* Let $x(n)$ be an N-point sequence with an N-point DFT $X(k)$ (N even).

(a) Consider the time-aliased sequence

$$y(n) = \begin{cases} \sum_{l=-\infty}^{\infty} x(n+lM), & 0 \le n \le M-1 \\ 0, & \text{elsewhere} \end{cases}$$

What is the relationship between the M-point DFT $Y(k)$ of $y(n)$ and the Fourier transform $X(\omega)$ of $x(n)$?

(b) Let

$$y(n) = \begin{cases} x(n) + x\left(n + \frac{N}{2}\right), & 0 \le n \le N-1 \\ 0, & \text{elsewhere} \end{cases}$$

and

$$y(n) \underset{N/2}{\overset{\text{DFT}}{\longleftrightarrow}} Y(k)$$

Show that $X(k) = Y(k/2)$, $k = 2, 4, \ldots, N-2$.

(c) Use the results in parts (a) and (b) to develop a procedure that computes the odd harmonics of $X(k)$ using an $N/2$-point DFT.

7.28 *Frequency-domain sampling.* Consider the following discrete-time signal

$$x(n) = \begin{cases} a^{|n|}, & |n| \le L \\ 0, & |n| > L \end{cases}$$

where $a = 0.95$ and $L = 10$.

(a) Compute and plot the signal $x(n)$.

(b) Show that

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} = x(0) + 2\sum_{n-1}^{L} x(n)\cos \omega n$$

Plot $X(\omega)$ by computing it at $\omega = \pi k/100$, $k = 0, 1, \ldots, 100$.

(c) Compute

$$c_k = \frac{1}{N} X\left(\frac{2\pi}{N} K\right), \qquad k = 0, 1, \ldots, N-1$$

for $N = 30$.

(d) Determine and plot the signal

$$\tilde{x}(n) = \sum_{k=0}^{N-1} ck e^{j(2\pi/N)kn}$$

What is the relation between the signals $x(n)$ and $\tilde{x}(n)$? Explain.

(e) Compute and plot the signal $\tilde{x}_1(n) = \sum_{l=-\infty}^{\infty} x(n-lN)$, $-L \le n \le L$ for $N = 30$. Compare the signals $\tilde{x}(n)$ and $\tilde{x}_1(n)$.

(f) Repeat parts (c) to (e) for $N = 15$.

7.29 *Frequency-domain sampling* The signal $x(n) = a^{|n|}$, $-1 < a < 1$ has a Fourier transform

$$X(\omega) = \frac{1 - a^2}{1 - 2a\cos\omega + a^2}$$

(a) Plot $X(\omega)$ for $0 \leq \omega \leq 2\pi$, $a = 0.8$. Reconstruct and plot $X(\omega)$ from its samples $X(2\pi k/N)$, $0 \leq k \leq N - 1$ for

(b) $N = 20$

(c) $N = 100$

(d) Compare the spectra obtained in parts (b) and (c) with the original spectrum $X(\omega)$ and explain the differences.

(e) Illustrate the time-domain aliasing when $N = 20$.

7.30 *Frequency analysis of amplitude-modulated discrete-time signal* The discrete-time signal

$$x(n) = \cos 2\pi f_1 n + \cos 2\pi f_2 n$$

where $f_1 = \frac{1}{18}$ and $f_2 = \frac{5}{128}$, modulates the amplitude of the carrier

$$x_c(n) = \cos 2\pi f_c n$$

where $f_c = \frac{50}{128}$. The resulting amplitude-modulated signal is

$$x_{\text{am}}(n) = x(n)\cos 2\pi f_c n$$

(a) Sketch the signals $x(n)$, $x_c(n)$, and $x_{\text{am}}(n)$, $0 \leq n \leq 255$.

(b) Compute and sketch the 128-point DFT of the signal $x_{\text{am}}(n)$, $0 \leq n \leq 127$.

(c) Compute and sketch the 128-point DFT of the signal $x_{\text{am}}(n)$, $0 \leq n \leq 99$.

(d) Compute and sketch the 256-point DFT of the signal $x_{\text{am}}(n)$, $0 \leq n \leq 179$.

(e) Explain the results obtained in parts (b) through (d), by deriving the spectrum of the amplitude-modulated signal and comparing it with the experimental results.

7.31 The sawtooth waveform in Fig. P7.31 can be expressed in the form of a Fourier series as

$$x(t) = \frac{2}{\pi}\left(\sin \pi t - \frac{1}{2}\sin 2\pi t + \frac{1}{3}\sin 3\pi t - \frac{1}{4}\sin 4\pi t \cdots\right)$$

(a) Determine the Fourier series coefficients c_k.

(b) Use an N-point subroutine to generate samples of this signal in the time domain using the first six terms of the expansion for $N = 64$ and $N = 128$. Plot the signal $x(t)$ and the samples generated, and comment on the results.

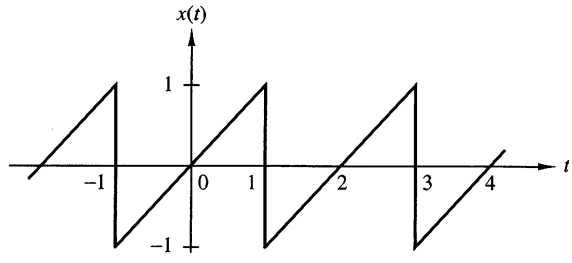


Figure P7.31

7.32 Recall that the Fourier transform of $x(t) = e^{j\Omega_0 t}$ is $X(j\Omega) = 2\pi\delta(\Omega - \Omega_0)$ and the Fourier transform of

$$p(t) = \begin{cases} 1, & 0 \le t \le T_0 \\ 0, & \text{otherwise} \end{cases}$$

is

$$P(j\Omega) = T_0 \frac{\sin \Omega T_0/2}{\Omega T_0/2} e^{-j\Omega T_0/2}$$

(a) Determine the Fourier transform $Y(j\Omega)$ of

$$y(t) = p(t)e^{j\Omega_0 t}$$

and roughly sketch $|Y(j\Omega)|$ versus Ω.

(b) Now consider the exponential sequence

$$x(n) = e^{j\omega_0 n}$$

where ω_0 is some arbitrary frequency in the range $0 < \omega_0 < \pi$ radians. Give the most general condition that ω_0 must satisfy in order for $x(n)$ to be periodic with period P (P is a positive integer).

(c) Let $y(n)$ be the finite-duration sequence

$$y(n) = x(n)w_N(n) = e^{j\omega_0 n} w_N(n)$$

where $w_N(n)$ is a finite-duration rectangular sequence of length N and where $x(n)$ is not necessarily periodic. Determine $Y(\omega)$ and roughly sketch $|Y(\omega)|$ for $0 \le \omega \le 2\pi$. What effect does N have in $|Y(\omega)|$? Briefly comment on the similarities and differences between $|Y(\omega)|$ and $|Y(j\Omega)|$.

(d) Suppose that

$$x(n) = e^{j(2\pi/P)n}, \qquad P \text{ a positive integer}$$

and

$$y(n) = w_N(n)x(n)$$

where $N = lP$, l a positive integer. Determine and sketch the N-point DFT of $y(n)$. Relate your answer to the characteristics of $|Y(\omega)|$.

(e) Is the frequency sampling for the DFT in part (d) adequate for obtaining a rough approximation of $|Y(\omega)|$ directly from the magnitude of the DFT sequence $|Y(k)|$? If not, explain briefly how the sampling can be increased so that it will be possible to obtain a rough sketch of $|Y(\omega)|$ from an appropriate sequence $|Y(k)|$.

7.33 Develop an algorithm that computes the DCT using the DFT as described in Sections 7.5.1 and 7.5.2.

7.34 Use the algorithm developed in Problem 7.33 to reproduce the results in Example 7.5.2.

7.35 Repeat Example 7.5.2 using the signal $x(n) = a^n \cos(2\pi f_0 n + \phi)$ with $a = 0.8$, $f_0 = 0.05$, and $N = 32$.

CHAPTER

8

Efficient Computation of the DFT: Fast Fourier Transform Algorithms

As we have observed in the preceding chapter, the discrete fourier transform (DFT) plays an important role in many applications of digital signal processing, including linear filtering, correlation analysis, and spectrum analysis. A major reason for its importance is the existence of efficient algorithms for computing the DFT.

The main topic of this chapter is the description of computationally efficient algorithms for evaluating the DFT. Two different approaches are described. One is a divide-and-conquer approach in which a DFT of size N, where N is a composite number, is reduced to the computation of smaller DFTs from which the larger DFT is computed. In particular, we present important computational algorithms, called fast Fourier transform (FFT) algorithms, for computing the DFT when the size N is a power of 2 and when it is a power of 4.

The second approach is based on the formulation of the DFT as a linear filtering operation on the data. This approach leads to two algorithms, the Goertzel algorithm and the chirp-z transform algorithm, for computing the DFT via linear filtering of the data sequence.

8.1 Efficient Computation of the DFT: FFT Algorithms

In this section we present several methods for computing the DFT efficiently. In view of the importance of the DFT in various digital signal processing applications, such as linear filtering, correlation analysis, and spectrum analysis, its efficient computation is a topic that has received considerable attention by many mathematicians, engineers, and applied scientists.

Basically, the computational problem for the DFT is to compute the sequence $\{X(k)\}$ of N complex-valued numbers given another sequence of data $\{x(n)\}$ of length

N, according to the formula

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad 0 \le k \le N-1 \tag{8.1.1}$$

where

$$W_N = e^{-j2\pi/N} \tag{8.1.2}$$

In general, the data sequence $x(n)$ is also assumed to be complex valued.

Similarly, the IDFT becomes

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk}, \qquad 0 \le n \le N-1 \tag{8.1.3}$$

Since the DFT and IDFT involve basically the same type of computations, our discussion of efficient computational algorithms for the DFT applies as well to the efficient computation of the IDFT.

We observe that for each value of k, direct computation of $X(k)$ involves N complex multiplications ($4N$ real multiplications) and $N-1$ complex additions ($4N-2$ real additions). Consequently, to compute all N values of the DFT requires N^2 complex multiplications and $N^2 - N$ complex additions.

Direct computation of the DFT is basically inefficient, primarily because it does not exploit the symmetry and periodicity properties of the phase factor W_N. In particular, these two properties are:

$$\text{Symmetry property:} \qquad W_N^{k+N/2} = -W_N^k \tag{8.1.4}$$

$$\text{Periodicity property:} \qquad W_N^{k+N} = W_N^k \tag{8.1.5}$$

The computationally efficient algorithms described in this section, known collectively as fast Fourier transform (FFT) algorithms, exploit these two basic properties of the phase factor.

8.1.1 Direct Computation of the DFT

For a complex-valued sequence $x(n)$ of N points, the DFT may be expressed as

$$X_R(k) = \sum_{n=0}^{N-1} \left[x_R(n) \cos \frac{2\pi kn}{N} + x_I(n) \sin \frac{2\pi kn}{N} \right] \tag{8.1.6}$$

$$X_I(k) = -\sum_{n=0}^{N-1} \left[x_R(n) \sin \frac{2\pi kn}{N} - x_I(n) \cos \frac{2\pi kn}{N} \right] \tag{8.1.7}$$

The direct computation of (8.1.6) and (8.1.7) requires:

1. $2N^2$ evaluations of trigonometric functions.
2. $4N^2$ real multiplications.
3. $4N(N-1)$ real additions.
4. A number of indexing and addressing operations.

These operations are typical of DFT computational algorithms. The operations in items 2 and 3 result in the DFT values $X_R(k)$ and $X_I(k)$. The indexing and addressing operations are necessary to fetch the data $x(n), 0 \leq n \leq N-1$, and the phase factors and to store the results. The variety of DFT algorithms optimize each of these computational processes in a different way.

8.1.2 Divide-and-Conquer Approach to Computation of the DFT

The development of computationally efficient algorithms for the DFT is made possible if we adopt a divide-and-conquer approach. This approach is based on the decomposition of an N-point DFT into successively smaller DFTs. This basic approach leads to a family of computationally efficient algorithms known collectively as FFT algorithms.

To illustrate the basic notions, let us consider the computation of an N-point DFT, where N can be factored as a product of two integers, that is,

$$N = LM \tag{8.1.8}$$

The assumption that N is not a prime number is not restrictive, since we can pad any sequence with zeros to ensure a factorization of the form (8.1.8).

Now the sequence $x(n), 0 \leq n \leq N-1$, can be stored either in a one-dimensional array indexed by n or as a two-dimensional array indexed by l and m, where $0 \leq l \leq L-1$ and $0 \leq m \leq M-1$ as illustrated in Fig. 8.1.1. Note that l is the row index and m is the column index. Thus, the sequence $x(n)$ can be stored in a rectangular

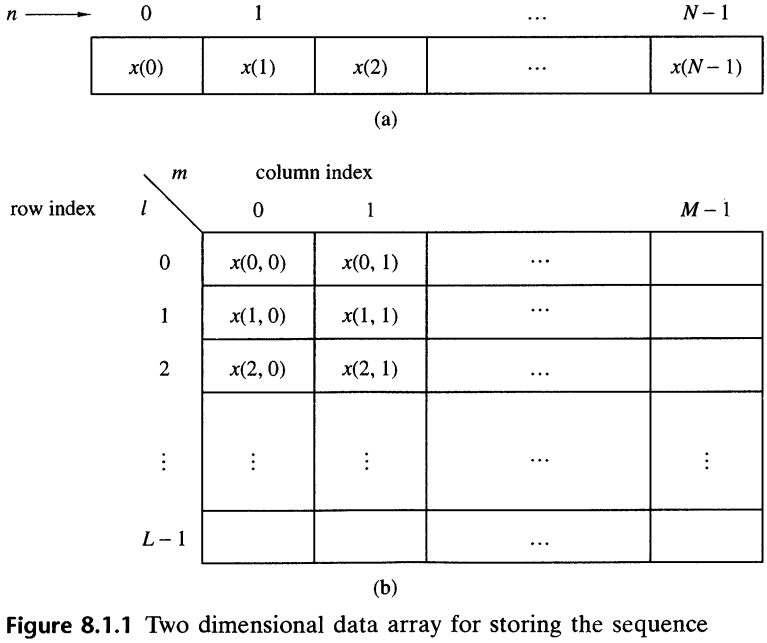


Figure 8.1.1 Two dimensional data array for storing the sequence $x(n), 0 \leq n \leq N-1$.

array in a variety of ways, each of which depends on the mapping of index n to the indexes (l, m).

For example, suppose that we select the mapping

$$n = Ml + m \tag{8.1.9}$$

This leads to an arrangement in which the first row consists of the first M elements of $x(n)$, the second row consists of the next M elements of $x(n)$, and so on, as illustrated in Fig. 8.1.2(a). On the other hand, the mapping

$$n = l + mL \tag{8.1.10}$$

stores the first L elements of $x(n)$ in the first column, the next L elements in the second column, and so on, as illustrated in Fig. 8.1.2(b).

Row-wise $n = Ml + m$

l \ m	0	1	2	...	$M-1$
0	$x(0)$	$x(1)$	$x(2)$	...	$x(M-1)$
1	$x(M)$	$x(M+1)$	$x(M+2)$	...	$x(2M-1)$
2	$x(2M)$	$x(2M+1)$	$x(2M+2)$	...	$x(3M-1)$
	⋮	⋮	⋮	...	⋮
$L-1$	$x((L-1)M)$	$x((L-1)M+1)$	$x((L-1)M+2)$	...	$x(LM-1)$

(a)

Column-wise $n = l + mL$

l \ m	0	1	2	...	$M-1$
0	$x(0)$	$x(L)$	$x(2L)$	...	$x((M-1)L)$
1	$x(1)$	$x(L+1)$	$x(2L+1)$	...	$x((M-1)L+1)$
2	$x(2)$	$x(L+2)$	$x(2L+2)$	...	$x((M-1)L+2)$
	⋮	⋮	⋮	...	⋮
$L-1$	$x(L-1)$	$x(2L-1)$	$x(3L-1)$	...	$x(LM-1)$

(b)

Figure 8.1.2 Two arrangements for the data arrays.

A similar arrangement can be used to store the computed DFT values. In particular, the mapping is from the index k to a pair of indices (p, q), where $0 \le p \le L-1$ and $0 \le q \le M-1$. If we select the mapping

$$k = Mp + q \tag{8.1.11}$$

the DFT is stored on a row-wise basis, where the first row contains the first M elements of the DFT $X(k)$, the second row contains the next set of M elements, and so on. On the other hand, the mapping

$$k = qL + p \tag{8.1.12}$$

results in a column-wise storage of $X(k)$, where the first L elements are stored in the first column, the second set of L elements are stored in the second column, and so on.

Now suppose that $x(n)$ is mapped into the rectangular array $x(l, m)$ and $X(k)$ is mapped into a corresponding rectangular array $X(p, q)$. Then the DFT can be expressed as a double sum over the elements of the rectangular array multiplied by the corresponding phase factors. To be specific, let us adopt a column-wise mapping for $x(n)$ given by (8.1.10) and the row-wise mapping for the DFT given by (8.1.11). Then

$$X(p, q) = \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l, m) W_N^{(Mp+q)(mL+l)} \tag{8.1.13}$$

But

$$W_N^{(Mp+q)(mL+l)} = W_N^{MLmp} W_N^{mLq} W_N^{Mpl} W_N^{lq} \tag{8.1.14}$$

However, $W_N^{Nmp} = 1$, $W_N^{mqL} = W_{N/L}^{mq} = W_M^{mq}$, and $W_N^{Mpl} = W_{N/M}^{pl} = W_L^{pl}$.

With these simplications, (8.1.13) can be expressed as

$$X(p, q) = \sum_{l=0}^{L-1} \left\{ W_N^{lq} \left[\sum_{m=0}^{M-1} x(l, m) W_M^{mq} \right] \right\} W_L^{lp} \tag{8.1.15}$$

The expression in (8.1.15) involves the computation of DFTs of length M and length L. To elaborate, let us subdivide the computation into three steps:

1. First, we compute the M-point DFTs

$$F(l, q) \equiv \sum_{m=0}^{M-1} x(l, m) W_M^{mq}, \qquad 0 \le q \le M-1 \tag{8.1.16}$$

   for each of the rows $l = 0, 1, \ldots, L-1$.

2. Second, we compute a new rectangular array $G(l, q)$ defined as

$$G(l, q) = W_N^{lq} F(l, q), \qquad \begin{matrix} 0 \le l \le L-1 \\ 0 \le q \le M-1 \end{matrix} \tag{8.1.17}$$

3. Finally, we compute the L-point DFTs

$$X(p, q) = \sum_{l=0}^{L-1} G(l, q) W_L^{lp} \tag{8.1.18}$$

for each column $q = 0, 1, \ldots, M-1$, of the array $G(l, q)$.

On the surface it may appear that the computational procedure outlined above is more complex than the direct computation of the DFT. However, let us evaluate the computational complexity of (8.1.15). The first step involves the computation of L DFTs, each of M points. Hence this step requires LM^2 complex multiplications and $LM(M-1)$ complex additions. The second step requires LM complex multiplications. Finally, the third step in the computation requires ML^2 complex multiplications and $ML(L-1)$ complex additions. Therefore, the computational complexity is

$$\begin{array}{ll} \text{Complex multiplications:} & N(M+L+1) \\ \text{Complex additions:} & N(M+L-2) \end{array} \tag{8.1.19}$$

where $N = ML$. Thus the number of multiplications has been reduced from N^2 to $N(M+L+1)$ and the number of additions has been reduced from $N(N-1)$ to $N(M+L-2)$.

For example, suppose that $N = 1000$ and we select $L = 2$ and $M = 500$. Then, instead of having to perform 10^6 complex multiplications via direct computation of the DFT, this approach leads to 503,000 complex multiplications. This represents a reduction by approximately a factor of 2. The number of additions is also reduced by about a factor of 2.

When N is a highly composite number, that is, N can be factored into a product of prime numbers of the form

$$N = r_1 r_2 \cdots r_v \tag{8.1.20}$$

then the decomposition above can be repeated $(v-1)$ more times. This procedure results in smaller DFTs, which, in turn, leads to a more efficient computational algorithm.

In effect, the first segmentation of the sequence $x(n)$ into a rectangular array of M columns with L elements in each column resulted in DFTs of sizes L and M. Further decomposition of the data in effect involves the segmentation of each row (or column) into smaller rectangular arrays which result in smaller DFTs. This procedure terminates when N is factored into its prime factors.

EXAMPLE 8.1.1

To illustrate this computational procedure, let us consider the computation of an $N = 15$ point DFT. Since $N = 5 \times 3 = 15$, we select $L = 5$ and $M = 3$. In other words, we store the 15-point sequence $x(n)$ column-wise as follows:

$$\begin{array}{llll} \text{Row 1:} & x(0,0) = x(0) & x(0,1) = x(5) & x(0,2) = x(10) \\ \text{Row 2:} & x(1,0) = x(1) & x(1,1) = x(6) & x(1,2) = x(11) \\ \text{Row 3:} & x(2,0) = x(2) & x(2,1) = x(7) & x(2,2) = x(12) \\ \text{Row 4:} & x(3,0) = x(3) & x(3,1) = x(8) & x(3,2) = x(13) \\ \text{Row 5:} & x(4,0) = x(4) & x(4,1) = x(9) & x(4,2) = x(14) \end{array}$$

Now, we compute the three-point DFTs for each of the five rows. This leads to the following 5×3 array:

$$\begin{array}{lll} F(0,0) & F(0,1) & F(0,2) \\ F(1,0) & F(1,1) & F(1,2) \\ F(2,0) & F(2,1) & F(2,2) \\ F(3,0) & F(3,1) & F(3,2) \\ F(4,0) & F(4,1) & F(4,2) \end{array}$$

The next step is to multiply each of the terms $F(l,q)$ by the phase factors $W_N^{lq} = W_{15}^{lq}$, $0 \le l \le 4$ and $0 \le q \le 2$. This computation results in the 5×3 array:

Column 1	Column 2	Column 3
$G(0,0)$	$G(0,1)$	$G(0,2)$
$G(1,0)$	$G(1,1)$	$G(1,2)$
$G(2,0)$	$G(2,1)$	$G(2,2)$
$G(3,0)$	$G(3,1)$	$G(3,2)$
$G(4,0)$	$G(4,1)$	$G(4,2)$

The final step is to compute the five-point DFTs for each of the three columns. This computation yields the desired values of the DFT in the form

$$\begin{array}{lll} X(0,0) = X(0) & X(0,1) = X(1) & X(0,2) = X(2) \\ X(1,0) = X(3) & X(1,1) = X(4) & X(1,2) = X(5) \\ X(2,0) = X(6) & X(2,1) = X(7) & X(2,2) = X(8) \\ X(3,0) = X(9) & X(3,1) = X(10) & X(3,2) = X(11) \\ X(4,0) = X(12) & X(4,1) = X(13) & X(4,2) = X(14) \end{array}$$

Figure 8.1.3 illustrates the steps in the computation.

It is interesting to view the segmented data sequence and the resulting DFT in terms of one-dimensional arrays. When the input sequence $x(n)$ and the output DFT $X(k)$ in the two-dimensional arrays are read across from row 1 through row 5, we obtain the following sequences:

INPUT ARRAY

$x(0)\ x(5)\ x(10)\ x(1)\ x(6)\ x(11)\ x(2)\ x(7)\ x(12)\ x(3)\ x(8)\ x(13)\ x(4)\ x(9)\ x(14)$

OUTPUT ARRAY

$X(0)\ X(1)\ X(2)\ X(3)\ X(4)\ X(5)\ X(6)\ X(7)\ X(8)\ X(9)\ X(10)\ X(11)\ X(12)\ X(13)\ X(14)$

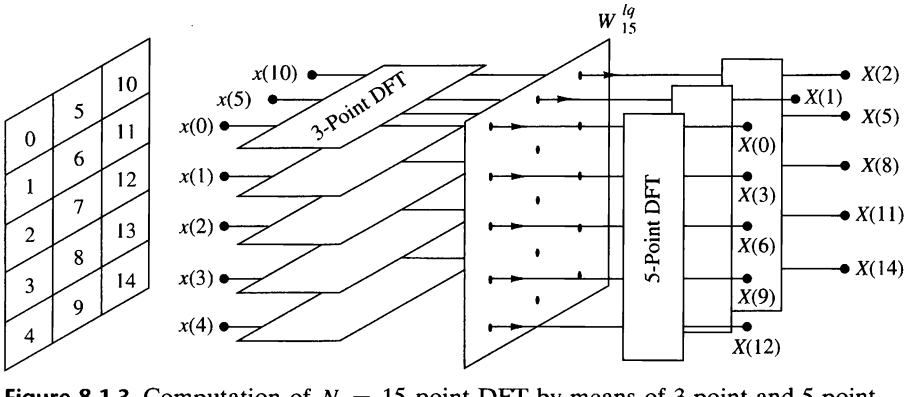


Figure 8.1.3 Computation of $N = 15$-point DFT by means of 3-point and 5-point DFTs.

We observe that the input data sequence is shuffled from the normal order in the computation of the DFT. On the other hand, the output sequence occurs in normal order. In this case the rearrangement of the input data array is due to the segmentation of the one-dimensional array into a rectangular array and the order in which the DFTs are computed. This shuffling of either the input data sequence or the output DFT sequence is a characteristic of most FFT algorithms.

To summarize, the algorithm that we have introduced involves the following computations:

Algorithm 1

1. Store the signal column-wise.
2. Compute the M-point DFT of each row.
3. Multiply the resulting array by the phase factors W_N^{lq}.
4. Compute the L-point DFT of each column
5. Read the resulting array row-wise.

An additional algorithm with a similar computational structure can be obtained if the input signal is stored row-wise and the resulting transformation is column-wise. In this case we select

$$\begin{aligned} n &= Ml + m \\ k &= qL + p \end{aligned} \tag{8.1.21}$$

This choice of indices leads to the formula for the DFT in the form

$$\begin{aligned} X(p,q) &= \sum_{m=0}^{M-1} \sum_{l=0}^{L-1} x(l,m) W_N^{pm} W_L^{pl} W_M^{qm} \\ &= \sum_{m=0}^{M-1} W_M^{mq} \left[\sum_{l=0}^{L-1} x(l,m) W_L^{lp} \right] W_N^{mp} \end{aligned} \tag{8.1.22}$$

Thus we obtain a second algorithm.

Algorithm 2

1. Store the signal row-wise.
2. Compute the L-point DFT at each column.
3. Multiply the resulting array by the factors W_N^{pm}.
4. Compute the M-point DFT of each row.
5. Read the resulting array column-wise.

The two algorithms given above have the same complexity. However, they differ in the arrangement of the computations. In the following sections we exploit the divide-and-conquer approach to derive fast algorithms when the size of the DFT is restricted to be a power of 2 or a power of 4.

8.1.3 Radix-2 FFT Algorithms

In the preceding section we described four algorithms for efficient computation of the DFT based on the divide-and-conquer approach. Such an approach is applicable when the number N of data points is not a prime. In particular, the approach is very efficient when N is highly composite, that is, when N can be factored as $N = r_1 r_2 r_3 \cdots r_\nu$, where the $\{r_j\}$ are prime.

Of particular importance is the case in which $r_1 = r_2 = \cdots = r_\nu \equiv r$, so that $N = r^\nu$. In such a case the DFTs are of size r, so that the computation of the N-point DFT has a regular pattern. The number r is called the radix of the FFT algorithm.

In this section we describe radix-2 algorithms, which are by far the most widely used FFT algorithms. Radix-4 algorithms are described in the following section.

Let us consider the computation of the $N = 2^\nu$ point DFT by the divide-and-conquer approach specified by (8.1.16) through (8.1.18). We select $M = N/2$ and $L = 2$. This selection results in a split of the N-point data sequence into two $N/2$-point data sequences $f_1(n)$ and $f_2(n)$, corresponding to the even-numbered and odd-numbered samples of $x(n)$, respectively, that is,

$$\begin{aligned} f_1(n) &= x(2n) \\ f_2(n) &= x(2n+1), \qquad n = 0, 1, \ldots, \frac{N}{2} - 1 \end{aligned} \tag{8.1.23}$$

Thus $f_1(n)$ and $f_2(n)$ are obtained by decimating $x(n)$ by a factor of 2, and hence the resulting FFT algorithm is called a decimation-in-time algorithm.

Now the N-point DFT can be expressed in terms of the DFTs of the decimated sequences as follows:

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1 \\ &= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn} \\ &= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)} \end{aligned} \tag{8.1.24}$$

But $W_N^2 = W_{N/2}$. With this substitution, (8.1.24) can be expressed as

$$\begin{aligned} X(k) &= \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km} \\ &= F_1(k) + W_N^k F_2(k), \qquad k = 0, 1, \ldots, N-1 \end{aligned} \tag{8.1.25}$$

where $F_1(k)$ and $F_2(k)$ are the $N/2$-point DFTs of the sequences $f_1(m)$ and $f_2(m)$, respectively.

Since $F_1(k)$ and $F_2(k)$ are periodic, with period $N/2$, we have $F_1(k + N/2) = F_1(k)$ and $F_2(k + N/2) = F_2(k)$. In addition, the factor $W_N^{k+N/2} = -W_N^k$. Hence (8.1.25) can be expressed as

$$X(k) = F_1(k) + W_N^k F_2(k), \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{8.1.26}$$

$$X\left(k + \frac{N}{2}\right) = F_1(k) - W_N^k F_2(k), \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{8.1.27}$$

We observe that the direct computation of $F_1(k)$ requires $(N/2)^2$ complex multiplications. The same applies to the computation of $F_2(k)$. Furthermore, there are $N/2$ additional complex multiplications required to compute $W_N^k F_2(k)$. Hence the computation of $X(k)$ requires $2(N/2)^2 + N/2 = N^2/2 + N/2$ complex multiplications. This first step results in a reduction of the number of multiplications from N^2 to $N^2/2 + N/2$, which is about a factor of 2 for N large.

To be consistent with our previous notation, we may define

$$G_1(k) = F_1(k) \quad , \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

$$G_2(k) = W_N^k F_2(k), \quad k = 0, 1, \ldots, \frac{N}{2} - 1$$

Then the DFT $X(k)$ may be expressed as

$$\begin{aligned} X(k) &= G_1(k) + G_2(k), \quad k = 0, 1, \ldots, \frac{N}{2} - 1 \\ X(k + \frac{N}{2}) &= G_1(k) - G_2(k), \quad k = 0, 1, \ldots, \frac{N}{2} - 1 \end{aligned} \tag{8.1.28}$$

This computation is illustrated in Fig. 8.1.4.

Having performed the decimation-in-time once, we can repeat the process for each of the sequences $f_1(n)$ and $f_2(n)$. Thus $f_1(n)$ would result in the two $N/4$-point sequences

$$\begin{aligned} v_{11}(n) &= f_1(2n), \qquad n = 0, 1, \ldots, \frac{N}{4} - 1 \\ v_{12}(n) &= f_1(2n + 1), \quad n = 0, 1, \ldots, \frac{N}{4} - 1 \end{aligned} \tag{8.1.29}$$

and $f_2(n)$ would yield

$$\begin{aligned} v_{21}(n) &= f_2(2n), \qquad n = 0, 1, \ldots, \frac{N}{4} - 1 \\ v_{22}(n) &= f_2(2n + 1), \quad n = 0, 1, \ldots, \frac{N}{4} - 1 \end{aligned} \tag{8.1.30}$$

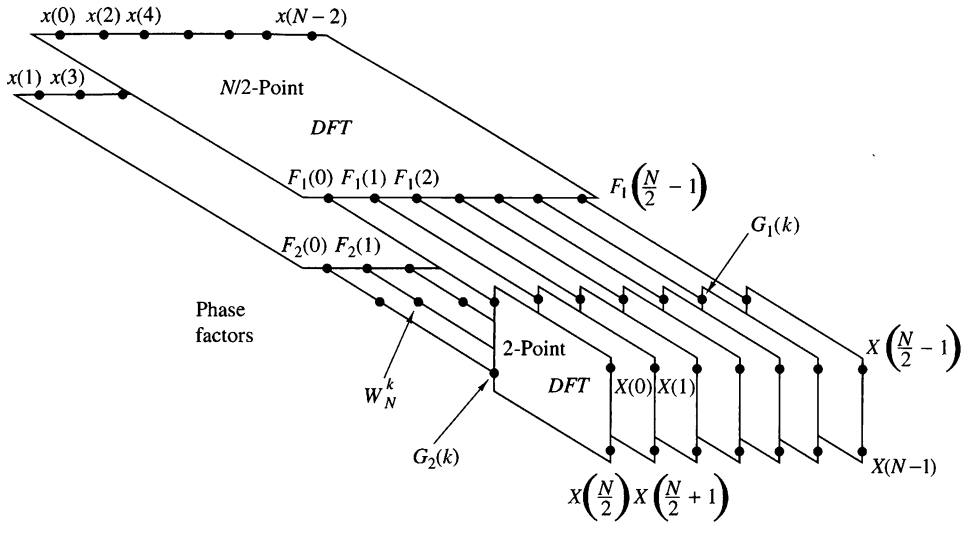


Figure 8.1.4 First step in the decimation-in-time algorithm.

By computing $N/4$-point DFTs, we would obtain the $N/2$-point DFTs $F_1(k)$ and $F_2(k)$ from the relations

$$F_1(k) = V_{11}(k) + W_{N/2}^k V_{12}(k), \quad k = 0, 1, \ldots, \frac{N}{4} - 1$$

$$F_1\left(k + \frac{N}{4}\right) = V_{11}(k) - W_{N/2}^k V_{12}(k), \quad k = 0, 1, \ldots, \frac{N}{4} - 1 \tag{8.1.31}$$

$$F_2(k) = V_{21}(k) + W_{N/2}^k V_{22}(k), \quad k = 0, 1, \ldots, \frac{N}{4} - 1$$

$$F_2\left(k + \frac{N}{4}\right) = V_{21}(k) - W_{N/2}^k V_{22}(k), \quad k = 0, \ldots, \frac{N}{4} - 1 \tag{8.1.32}$$

where the $\{V_{ij}(k)\}$ are the $N/4$-point DFTs of the sequences $\{v_{ij}(n)\}$.

We observe that the computation of $\{V_{ij}(k)\}$ requires $4(N/4)^2$ multiplications and hence the computation of $F_1(k)$ and $F_2(k)$ can be accomplished with $N^2/4 + N/2$ complex multiplications. An additional $N/2$ complex multiplications are required to compute $X(k)$ from $F_1(k)$ and $F_2(k)$. Consequently, the total number of multiplications is reduced approximately by a factor of 2 again to $N^2/4 + N$.

The decimation of the data sequence can be repeated again and again until the resulting sequences are reduced to one-point sequences. For $N = 2^\nu$, this decimation can be performed $\nu = \log_2 N$ times. Thus the total number of complex multiplications is reduced to $(N/2)\log_2 N$. The number of complex additions is $N \log_2 N$. Table 8.1 presents a comparison of the number of complex multiplications in the FFT and in the direct computation of the DFT.

For illustrative purposes, Fig. 8.1.5 depicts the computation of an $N = 8$-point DFT. We observe that the computation is performed in three stages, beginning with the computations of four two-point DFTs, then two four-point DFTs, and finally, one

TABLE 8.1 Comparison of Computational Complexity for the Direct Computation of the DFT Versus the FFT Algorithm

Number of Points, N	Complex Multiplications in Direct Computation, N^2	Complex Multiplications in FFT Algorithm, $(N/2)\log_2 N$	Speed Improvement Factor
4	16	4	4.0
8	64	12	5.3
16	256	32	8.0
32	1,024	80	12.8
64	4,096	192	21.3
128	16,384	448	36.6
256	65,536	1,024	64.0
512	262,144	2,304	113.8
1,024	1,048,576	5,120	204.8

eight-point DFT. The combination of the smaller DFTs to form the larger DFT is illustrated in Fig. 8.1.6 for $N = 8$.

Observe that the basic computation performed at every stage, as illustrated in Fig. 8.1.6, is to take two complex numbers, say the pair (a, b), multiply b by W_N^r, and then add and subtract the product from a to form two new complex numbers (A, B). This basic computation, which is shown in Fig. 8.1.7, is called a *butterfly* because the flow graph resembles a butterfly.

In general, each butterfly involves one complex multiplication and two complex additions. For $N = 2^\nu$, there are $N/2$ butterflies per stage of the computation process and $\log_2 N$ stages. Therefore, as previously indicated, the total number of complex multiplications is $(N/2)\log_2 N$ and complex additions is $N \log_2 N$.

Once a butterfly operation is performed on a pair of complex numbers (a, b) to produce (A, B), there is no need to save the input pair (a, b). Hence we can store the result (A, B) in the same locations as (a, b). Consequently, we require a fixed amount of storage, namely, $2N$ storage registers, in order to store the results

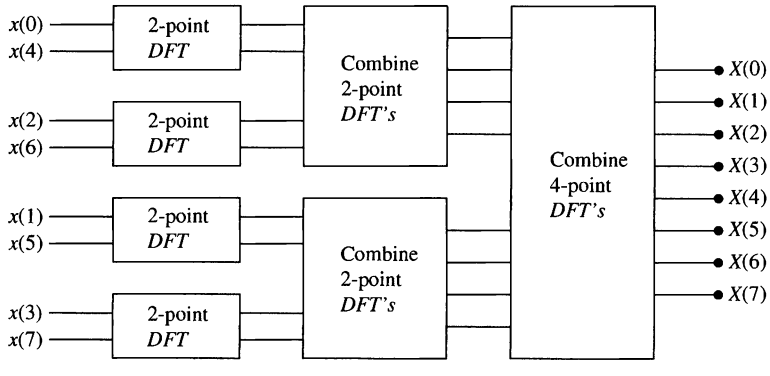


Figure 8.1.5 Three stages in the computation of an $N = 8$-point DFT.

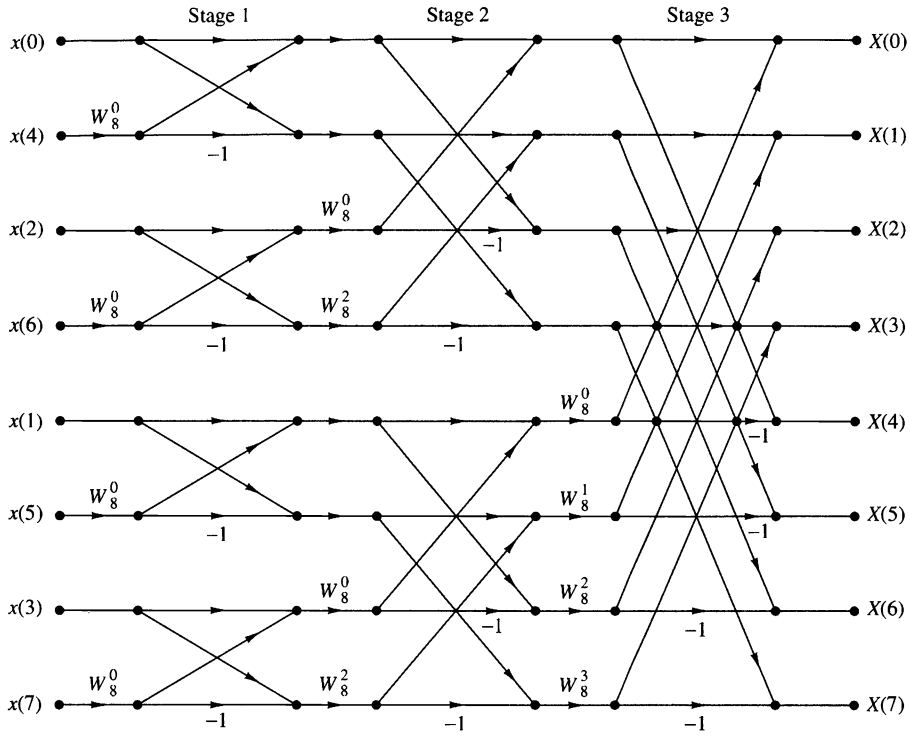


Figure 8.1.6 Eight-point decimation-in-time FFT algorithm.

(N complex numbers) of the computations at each stage. Since the same $2N$ storage locations are used throughout the computation of the N-point DFT, we say that *the computations are done in place.*

A second important observation is concerned with the order of the input data sequence after it is decimated $(\nu - 1)$ times. For example, if we consider the case where $N = 8$, we know that the first decimation yields the sequence $x(0), x(2), x(4), x(6), x(1), x(3), x(5), x(7)$, and the second decimation results in the sequence $x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7)$. This *shuffling* of the input data sequence has a well-defined order as can be ascertained from observing Fig. 8.1.8, which illustrates the decimation of the eight-point sequence. By expressing the index n, in the sequence $x(n)$, in binary form, we note that the order of the decimated data sequence is easily obtained by reading the binary representation of the index n in reverse order. Thus the data point $x(3) \equiv x(011)$ is placed in position $m = 110$ or $m = 6$ in the decimated array. Thus we say that the data $x(n)$ after decimation is stored in bit-reversed order.

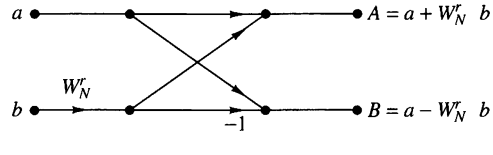
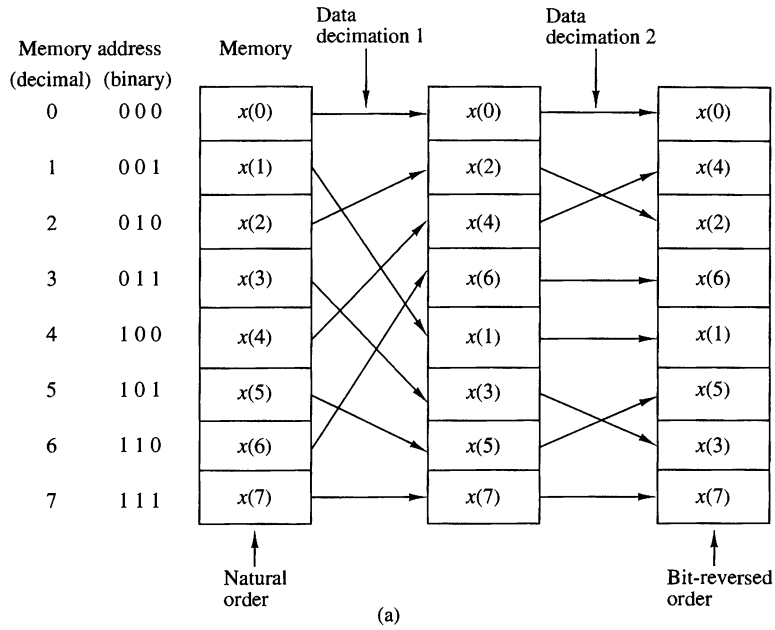


Figure 8.1.7
Basic butterfly computation in the decimation-in-time FFT algorithm.



$(n_2n_1n_0)$	→	$(n_1n_0n_2)$	→	$(n_0n_1n_2)$
(0 0 0)	→	(0 0 0)	→	(0 0 0)
(0 0 1)	→	(0 1 0)	→	(1 0 0)
(0 1 0)	→	(1 0 0)	→	(0 1 0)
(0 1 1)	→	(1 1 0)	→	(1 1 0)
(1 0 0)	→	(0 0 1)	→	(0 0 1)
(1 0 1)	→	(0 1 1)	→	(1 0 1)
(1 1 0)	→	(1 0 1)	→	(0 1 1)
(1 1 1)	→	(1 1 1)	→	(1 1 1)

(b)

Figure 8.1.8 Shuffling of the data and bit reversal.

With the input data sequence stored in bit-reversed order and the butterfly computations performed in place, the resulting DFT sequence $X(k)$ is obtained in natural order (i.e., $k = 0, 1, \ldots, N-1$). On the other hand, we should indicate that it is possible to arrange the FFT algorithm such that the input is left in natural order and the resulting output DFT will occur in bit-reversed order. Furthermore, we can impose the restriction that both the input data $x(n)$ and the output DFT $X(k)$ be in natural order, and derive an FFT algorithm in which the computations are not done in place. Hence such an algorithm requires additional storage.

Another important radix-2 FFT algorithm, called the decimation-in-frequency algorithm, is obtained by using the divide-and-conquer approach described in Section 8.1.2 with the choice of $M = 2$ and $L = N/2$. This choice of parameters implies a column-wise storage of the input data sequence. To derive the algorithm, we begin by splitting the DFT formula into two summations, of which one involves the sum over the first $N/2$ data points and the second the sum over the last $N/2$ data points.

Thus we obtain

$$\begin{aligned} X(k) &= \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + \sum_{n=N/2}^{N-1} x(n)W_N^{kn} \\ &= \sum_{n=0}^{(N/2)-1} x(n)W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{(N/2)-1} x\left(n+\frac{N}{2}\right) W_N^{kn} \end{aligned} \tag{8.1.33}$$

Since $W_N^{kN/2} = (-1)^k$, the expression (8.1.33) can be rewritten as

$$X(k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + (-1)^k x\left(n+\frac{N}{2}\right)\right] W_N^{kn} \tag{8.1.34}$$

Now, let us split (decimate) $X(k)$ into the even- and odd-numbered samples. Thus we obtain

$$X(2k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + x\left(n+\frac{N}{2}\right)\right] W_{N/2}^{kn}, \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{8.1.35}$$

and

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} \left\{\left[x(n) - x\left(n+\frac{N}{2}\right)\right] W_N^n\right\} W_{N/2}^{kn}, \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{8.1.36}$$

where we have used the fact that $W_N^2 = W_{N/2}$.

If we define the $N/2$-point sequences $g_1(n)$ and $g_2(n)$ as

$$\begin{aligned} g_1(n) &= x(n) + x\left(n+\frac{N}{2}\right) \\ g_2(n) &= \left[x(n) - x\left(n+\frac{N}{2}\right)\right] W_N^n, \qquad n = 0, 1, 2, \ldots, \frac{N}{2} - 1 \end{aligned} \tag{8.1.37}$$

then

$$\begin{aligned} X(2k) &= \sum_{n=0}^{(N/2)-1} g_1(n)W_{N/2}^{kn} \\ X(2k+1) &= \sum_{n=0}^{(N/2)-1} g_2(n)W_{N/2}^{kn} \end{aligned} \tag{8.1.38}$$

The computation of the sequences $g_1(n)$ and $g_2(n)$ according to (8.1.37) and the subsequent use of these sequences to compute the $N/2$-point DFTs are depicted in Fig. 8.1.9. We observe that the basic computation in this figure involves the butterfly operation illustrated in Fig. 8.1.10.

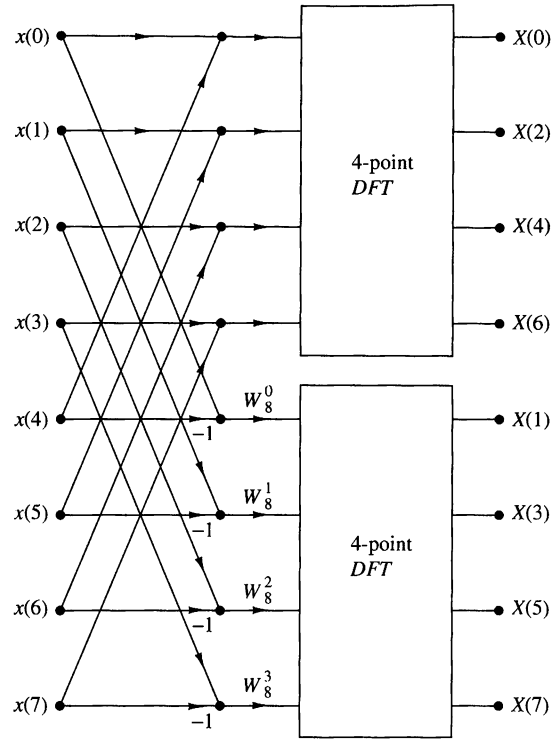


Figure 8.1.9
First stage of the decimation-in-frequency FFT algorithm.

This computational procedure can be repeated through decimation of the $N/2$-point DFTs, $X(2k)$ and $X(2k+1)$. The entire process involves $\nu = \log_2 N$ stages of decimation, where each stage involves $N/2$ butterflies of the type shown in Fig. 8.1.10. Consequently, the computation of the N-point DFT via the decimation-in-frequency FFT algorithm requires $(N/2)\log_2 N$ complex multiplications and $N \log_2 N$ complex additions, just as in the decimation-in-time algorithm. For illustrative purposes, the eight-point decimation-in-frequency algorithm is given in Fig. 8.1.11.

We observe from Fig. 8.1.11 that the input data $x(n)$ occurs in natural order, but the output DFT occurs in bit-reversed order. We also note that the computations are performed in place. However, it is possible to reconfigure the decimation-in-frequency algorithm so that the input sequence occurs in bit-reversed order while the output DFT occurs in normal order. Furthermore, if we abandon the requirement that the computations be done in place, it is also possible to have both the input data and the output DFT in normal order.

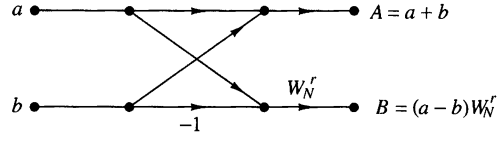


Figure 8.1.10
Basic butterfly computation in the decimation-in-frequency FFT algorithm.

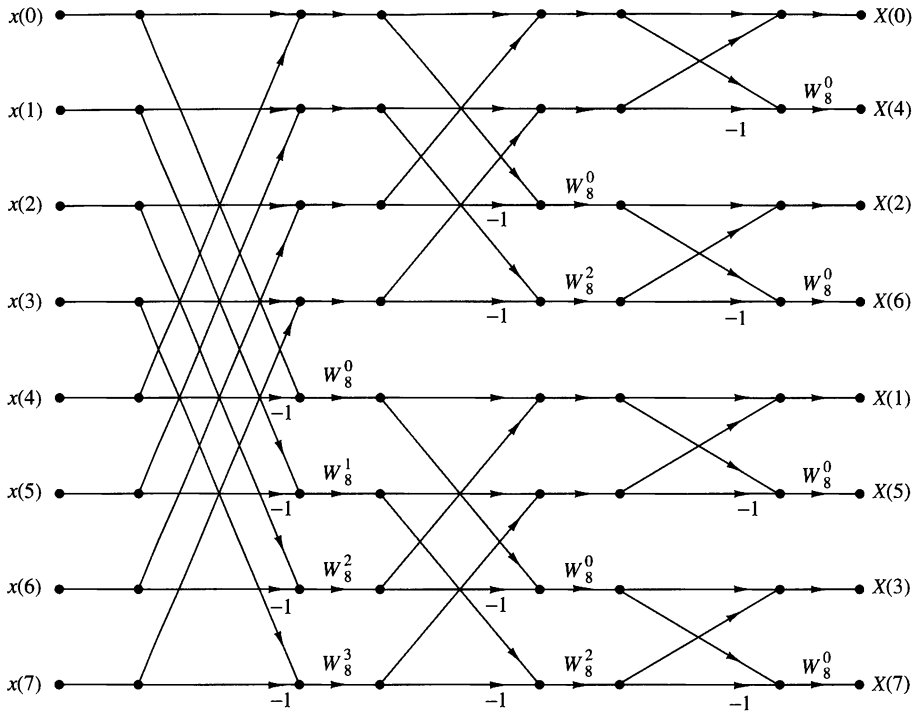


Figure 8.1.11 $N = 8$-point decimation-in-frequency FFT algorithm.

8.1.4 Radix-4 FFT Algorithms

When the number of data points N in the DFT is a power of 4 (i.e., $N = 4^\nu$), we can, of course, always use a radix-2 algorithm for the computation. However, for this case, it is more efficient computationally to employ a radix-4 FFT algorithm.

Let us begin by describing a radix-4 decimation-in-time FFT algorithm, which is obtained by selecting $L = 4$ and $M = N/4$ in the divide-and-conquer approach described in Section 8.1.2. For this choice of L and M, we have l, $p = 0, 1, 2, 3$; m, $q = 0, 1, \ldots, N/4 - 1$; $n = 4m + l$; and $k = (N/4)p + q$. Thus we split or decimate the N-point input sequence into four subsequences, $x(4n)$, $x(4n + 1)$, $x(4n + 2)$, $x(4n + 3)$, $n = 0, 1, \ldots, N/4 - 1$.

By applying (8.1.15) we obtain

$$X(p, q) = \sum_{l=0}^{3} \left[W_N^{lq} F(l, q) \right] W_4^{lp}, \qquad p = 0, 1, 2, 3 \tag{8.1.39}$$

where $F(l, q)$ is given by (8.1.16), that is,

$$F(l, q) = \sum_{m=0}^{(N/4)-1} x(l, m) W_{N/4}^{mq}, \qquad \begin{matrix} l = 0, 1, 2, 3, \\ q = 0, 1, 2, \ldots, \dfrac{N}{4} - 1 \end{matrix} \tag{8.1.40}$$

and

$$x(l, m) = x(4m + l) \tag{8.1.41}$$

$$X(p, q) = X\left(\frac{N}{4}p + q\right) \tag{8.1.42}$$

Thus, the four $N/4$-point DFTs obtained from (8.1.40) are combined according to (8.1.39) to yield the N-point DFT. The expression in (8.1.39) for combining the $N/4$-point DFTs defines a radix-4 decimation-in-time butterfly, which can be expressed in matrix form as

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^q F(1,q) \\ W_N^{2q} F(2,q) \\ W_N^{3q} F(3,q) \end{bmatrix} \tag{8.1.43}$$

The radix-4 butterfly is depicted in Fig. 8.1.12(a) and in a more compact form in Fig. 8.1.12(b). Note that since $W_N^0 = 1$, each butterfly involves three complex multiplications, and 12 complex additions.

This decimation-in-time procedure can be repeated recursively ν times. Hence the resulting FFT algorithm consists of ν stages, where each stage contains $N/4$ butterflies. Consequently, the computational burden for the algorithm is $3\nu N/4 = (3N/8)\log_2 N$ complex multiplications and $(3N/2)\log_2 N$ complex additions. We note that the number of multiplications is reduced by 25%, but the number of additions has increased by 50% from $N\log_2 N$ to $(3N/2)\log_2 N$.

It is interesting to note, however, that by performing the additions in two steps, it is possible to reduce the number of additions per butterfly from 12 to 8. This can

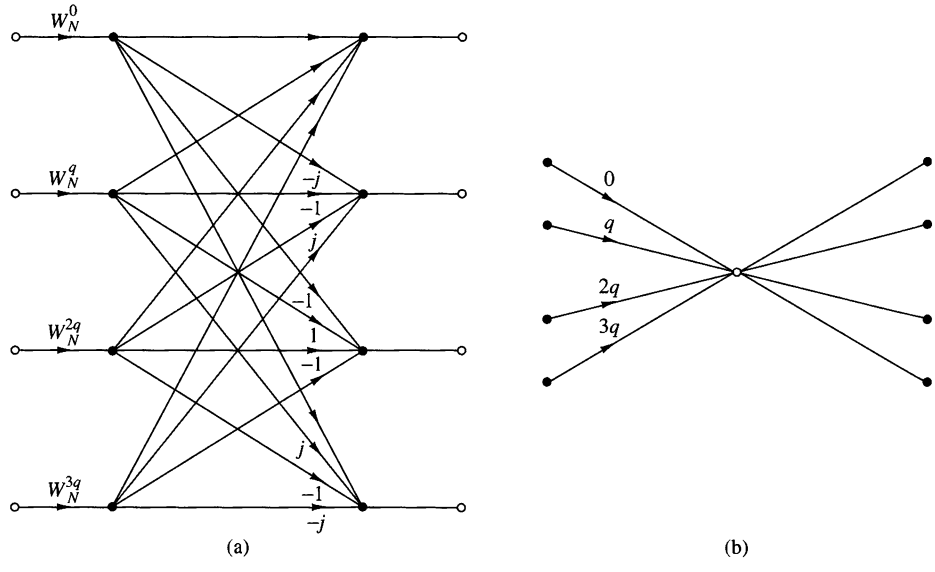


Figure 8.1.12 Basic butterfly computation in a radix-4 FFT algorithm.

be accomplished by expressing the matrix of the linear transformation in (8.1.43) as a product of two matrices as follows:

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -j \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & j \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^q F(1,q) \\ W_N^{2q} F(2,q) \\ W_N^{3q} F(3,q) \end{bmatrix} \tag{8.1.44}$$

Now each matrix multiplication involves four additions for a total of eight additions. Thus the total number of complex additions is reduced to $N \log_2 N$, which is identical to the radix-2 FFT algorithm. The computational savings results from the 25% reduction in the number of complex multiplications.

An illustration of a radix-4 decimation-in-time FFT algorithm is shown in Fig. 8.1.13 for $N = 16$. Note that in this algorithm, the input sequence is in normal order while the output DFT is shuffled. In the radix-4 FFT algorithm, where the decimation is by a factor of 4, the order of the decimated sequence can be determined by reversing the order of the number that represents the index n in a quaternary number system (i.e., the number system based on the digits 0, 1, 2, 3).

A radix-4 decimation-in-frequency FFT algorithm can be obtained by selecting $L = N/4$, $M = 4$; l, $p = 0, 1, \ldots, N/4 - 1$; m, $q = 0, 1, 2, 3$; $n = (N/4)m + l$; and

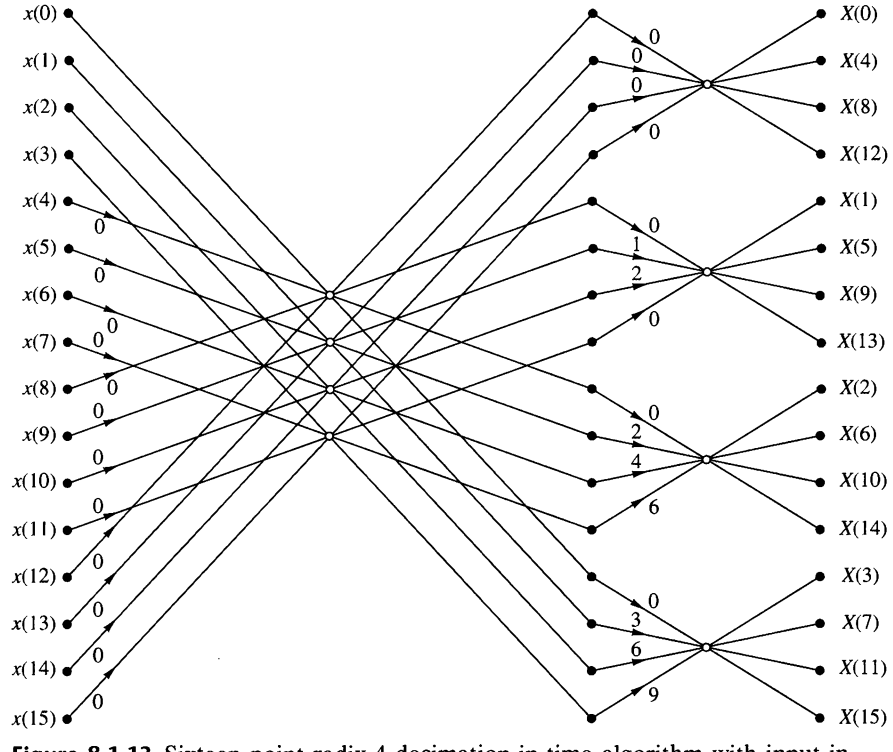


Figure 8.1.13 Sixteen-point radix-4 decimation-in-time algorithm with input in normal order and output in digit-reversed order. The integer multipliers shown on the graph represent the exponents on W_{16}.

$k = 4p + q$. With this choice of parameters, the general equation given by (8.1.15) can be expressed as

$$X(p, q) = \sum_{l=0}^{(N/4)-1} G(l, q) W_{N/4}^{lp} \tag{8.1.45}$$

where

$$G(l, q) = W_N^{lq} F(l, q), \qquad \begin{matrix} q = 0, 1, 2, 3 \\ l = 0, 1, \ldots, \dfrac{N}{4} - 1 \end{matrix} \tag{8.1.46}$$

and

$$F(l, q) = \sum_{m=0}^{3} x(l, m) W_4^{mq}, \qquad \begin{matrix} q = 0, 1, 2, 3 \\ l = 0, 1, 2, 3, \ldots, \dfrac{N}{4} - 1 \end{matrix} \tag{8.1.47}$$

We note that $X(p, q) = X(4p + q)$, $q = 0, 1, 2, 3$. Consequently, the N-point DFT is decimated into four $N/4$-point DFTs and hence we have a decimation-in-frequency FFT algorithm. The computations in (8.1.46) and (8.1.47) define the basic radix-4 butterfly for the decimation-in-frequency algorithm. Note that the multiplications by the factors W_N^{lq} occur after the combination of the data points $x(l, m)$, just as in the case of the radix-2 decimation-in-frequency algorithm.

A 16-point radix-4 decimation-in-frequency FFT algorithm is shown in Fig. 8.1.14. Its input is in normal order and its output is in digit-reversed order. It has exactly the same computational complexity as the decimation-in-time radix-4 FFT algorithm.

For illustrative purposes, let us rederive the radix-4 decimation-in-frequency algorithm by breaking the N-point DFT formula into four smaller DFTs. We have

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{N/4-1} x(n) W_N^{kn} + \sum_{n=N/4}^{N/2-1} x(n) W_N^{kn} + \sum_{n=N/2}^{3N/4-1} x(n) W_N^{kn} + \sum_{n=3N/4}^{N-1} x(n) W_N^{kn} \\ &= \sum_{n=0}^{N/4-1} x(n) W_N^{kn} + W_N^{Nk/4} \sum_{n=0}^{N/4-1} x\left(n + \frac{N}{4}\right) W_N^{kn} \\ &\quad + W_N^{kN/2} \sum_{n=0}^{N/4-1} x\left(n + \frac{N}{2}\right) W_N^{nk} + W_N^{3kN/4} \sum_{n=0}^{N/4-1} x\left(n + \frac{3N}{4}\right) W_N^{kn} \end{aligned} \tag{8.1.48}$$

From the definition of the phase factors, we have

$$W_N^{kN/4} = (-j)^k, \qquad W_N^{Nk/2} = (-1)^k, \qquad W_N^{3Nk/4} = (j)^k \tag{8.1.49}$$

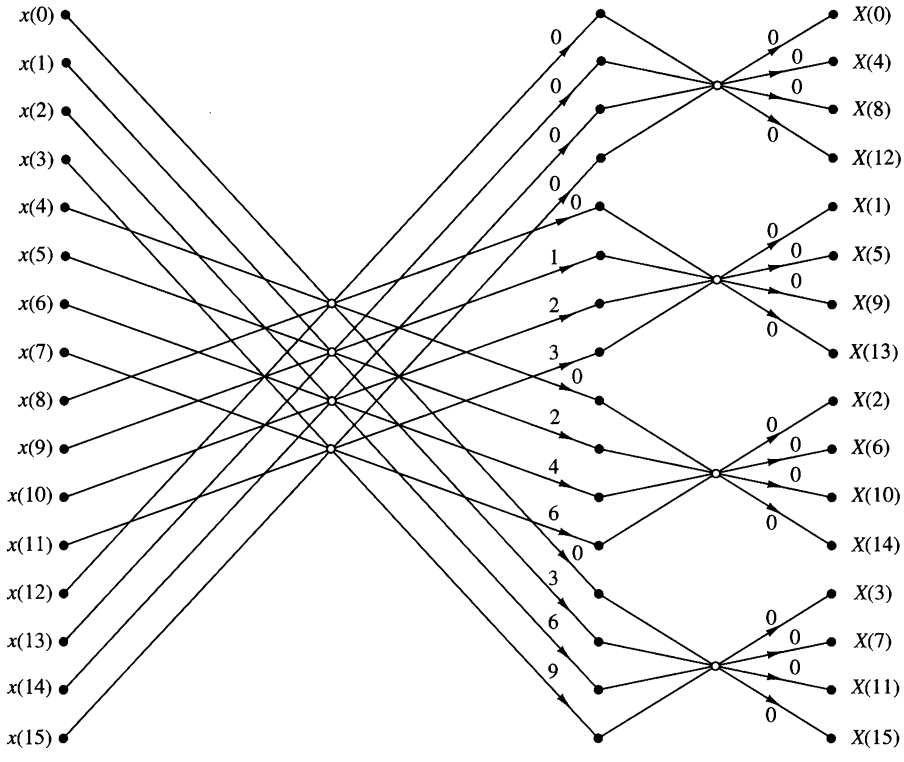


Figure 8.1.14 Sixteen-point, radix-4 decimation-in-frequency algorithm with input in normal order and output in digit-reversed order.

After substitution of (8.1.49) into (8.1.48), we obtain

$$X(k) = \sum_{n=0}^{N/4-1} \left[x(n) + (-j)^k x\left(n + \frac{N}{4}\right) + (-1)^k x\left(n + \frac{N}{4}\right) + (j)^k x\left(n + \frac{3N}{4}\right) \right] W_N^{nk} \tag{8.1.50}$$

The relation in (8.1.50) is not an $N/4$-point DFT because the phase factor depends on N and not on $N/4$. To convert it into an $N/4$-point DFT, we subdivide the DFT sequence into four $N/4$-point subsequences, $X(4k)$, $X(4k+1)$, $X(4k+2)$, and $X(4k+3)$, $k = 0, 1, \ldots, N/4-1$. Thus we obtain the radix-4 decimation-in-frequency DFT as

$$X(4k) = \sum_{n=0}^{N/4-1} \left[x(n) + x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) + x\left(n + \frac{3N}{4}\right) \right] W_N^0 W_{N/4}^{kn} \tag{8.1.51}$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} \left[x(n) - jx\left(n + \frac{N}{4}\right) - x\left(n + \frac{N}{2}\right) + jx\left(n + \frac{3N}{4}\right) \right] W_N^n W_{N/4}^{kn} \tag{8.1.52}$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} \left[x(n) - x\left(n + \frac{N}{4}\right) + x\left(n + \frac{N}{2}\right) - x\left(n + \frac{3N}{4}\right) \right] W_N^{2n} W_{N/4}^{kn} \tag{8.1.53}$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} \left[x(n) + jx\left(n + \frac{N}{4}\right) - x\left(n + \frac{N}{2}\right) - jx\left(n + \frac{3N}{4}\right) \right] W_N^{3n} W_{N/4}^{kn} \tag{8.1.54}$$

where we have used the property $W_N^{4kn} = W_{N/4}^{kn}$. Note that the input to each $N/4$-point DFT is a linear combination of four signal samples scaled by a phase factor. This procedure is repeated ν times, where $\nu = \log_4 N$.

8.1.5 Split-Radix FFT Algorithms

An inspection of the radix-2 decimation-in-frequency flowgraph shown in Fig. 8.1.11 indicates that the even-numbered points of the DFT can be computed independently of the odd-numbered points. This suggests the possibility of using different computational methods for independent parts of the algorithm with the objective of reducing the number of computations. The split-radix FFT (SRFFT) algorithms exploit this idea by using both a radix-2 and a radix-4 decomposition in the same FFT algorithm.

We illustrate this approach with a decimation-in-frequency SRFFT algorithm due to Duhamel (1986). First, we recall that in the radix-2 decimation-in-frequency FFT algorithm, the even-numbered samples of the N-point DFT are given as

$$X(2k) = \sum_{n=0}^{N/2-1} \left[x(n) + x\left(n + \frac{N}{2}\right) \right] W_{N/2}^{nk}, \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \tag{8.1.55}$$

Note that these DFT points can be obtained from an $N/2$-point DFT without any additional multiplications. Consequently, a radix-2 suffices for this computation.

The odd-numbered samples $\{X(2k+1)\}$ of the DFT require the premultiplication of the input sequence with the phase factors W_N^n. For these samples a radix-4 decomposition produces some computational efficiency because the four-point DFT has the largest multiplication-free butterfly. Indeed, it can be shown that using a radix greater than 4 does not result in a significant reduction in computational complexity.

If we use a radix-4 decimation-in-frequency FFT algorithm for the odd-numbered samples of the N-point DFT, we obtain the following $N/4$-point DFTs:

$$X(4k+1) = \sum_{n=0}^{N/4-1} \{[x(n) - x(n+N/2)] - j[x(n+N/4) - x(n+3N/4)]\} W_N^n W_{N/4}^{kn} \tag{8.1.56}$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} \{[x(n) - x(n+N/2)] + j[x(n+N/4) - x(n+3N/4)]\} W_N^{3n} W_{N/4}^{kn} \tag{8.1.57}$$

Thus the N-point DFT is decomposed into one $N/2$-point DFT without additional phase factors and two $N/4$-point DFTs with phase factors. The N-point DFT is obtained by successive use of these decompositions up to the last stage. Thus we obtain a decimation-in-frequency SRFFT algorithm.

Figure 8.1.15 shows the flow graph for an in-place 32-point decimation-in-frequency SRFFT algorithm. At stage A of the computation for $N = 32$, the top 16 points constitute the sequence

$$g_0(n) = x(n) + x(n+N/2), \qquad 0 \le n \le 15 \tag{8.1.58}$$

This is the sequence required for the computation of $X(2k)$. The next 8 points constitute the sequence

$$g_1(n) = x(n) - x(n+N/2), \qquad 0 \le n \le 7 \tag{8.1.59}$$

The bottom eight points constitute the sequence $jg_2(n)$, where

$$g_2(n) = x(n+N/4) - x(n+3N/4), \qquad 0 \le n \le 7 \tag{8.1.60}$$

The sequences $g_1(n)$ and $g_2(n)$ are used in the computation of $X(4k+1)$ and $X(4k+3)$. Thus, at stage A we have completed the first decimation for the radix-2 component of the algorithm. At stage B, the bottom eight points constitute the computation of $[g_1(n) + jg_2(n)]W_{32}^{3n}$, $0 \le n \le 7$, which is used to compute $X(4k+3)$, $0 \le k \le 7$. The next eight points from the bottom constitute the computation of $[g_1(n) - jg_2(n)]$ W_{32}^n, $0 \le n \le 7$, which is used to compute $X(4k+1)$, $0 \le k \le 7$. Thus at stage B, we have completed the first decimation for the radix-4 algorithm, which results in two 8-point sequences. Hence the basic butterfly computation for the SRFFT algorithm has the "L-shaped" form illustrated in Fig. 8.1.16.

Now we repeat the steps in the computation above. Beginning with the top 16 points at stage A, we repeat the decomposition for the 16-point DFT. In other words,

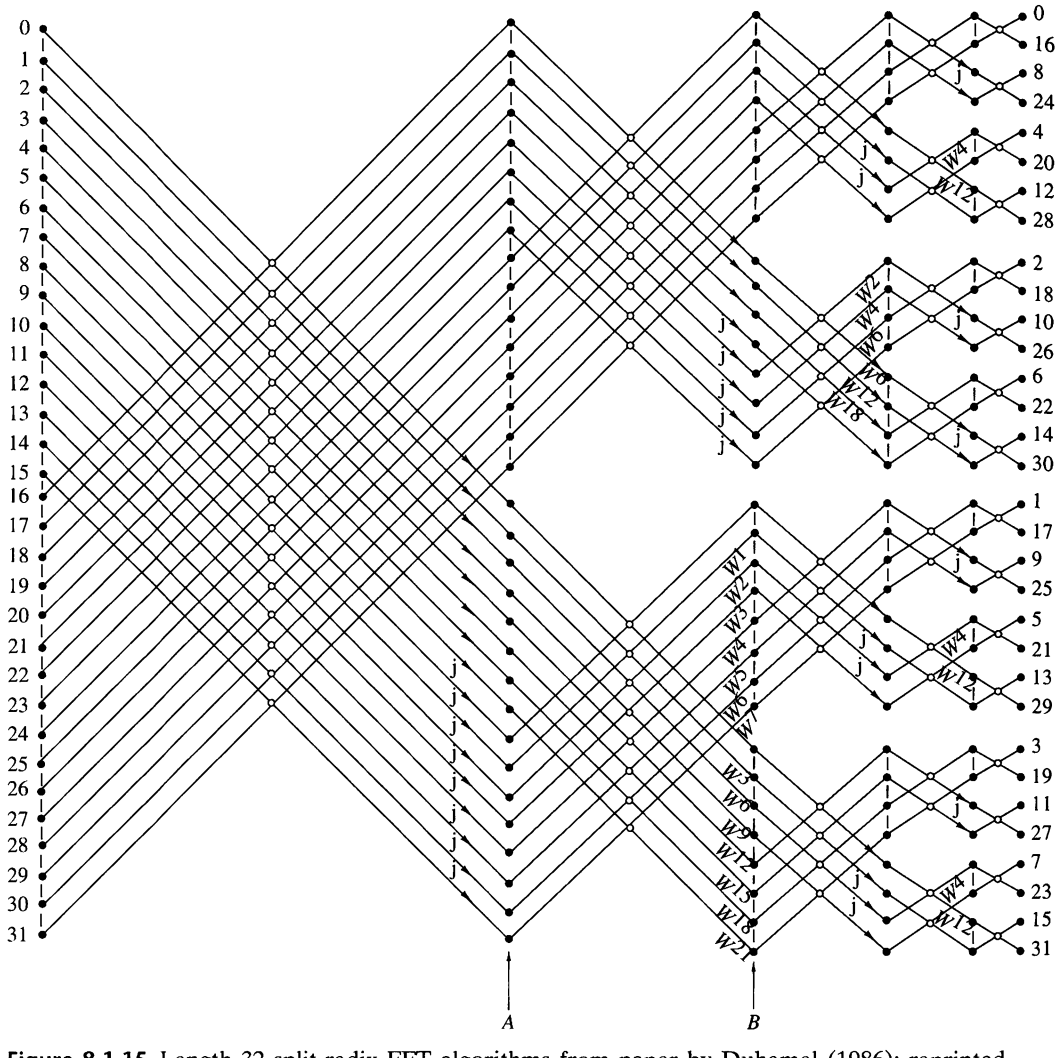


Figure 8.1.15 Length 32 split-radix FFT algorithms from paper by Duhamel (1986); reprinted with permission from the IEEE.

we decompose the computation into an eight-point, radix-2 DFT and two four-point, radix-4 DFTs. Thus at stage B, the top eight points constitute the sequence (with $N = 16$)

$$g_0'(n) = g_0(n) + g_0(n + N/2), \qquad 0 \le n \le 7 \tag{8.1.61}$$

and the next eight points constitute the two four-point sequences $g_1'(n)$ and $jg_2'(n)$, where

$$\begin{aligned} g_1'(n) &= g_0(n) - g_0(n + N/2), \qquad 0 \le n \le 3 \\ g_2'(n) &= g_0(n + N/4) - g_0(n + 3N/4), \qquad 0 \le n \le 3 \end{aligned} \tag{8.1.62}$$

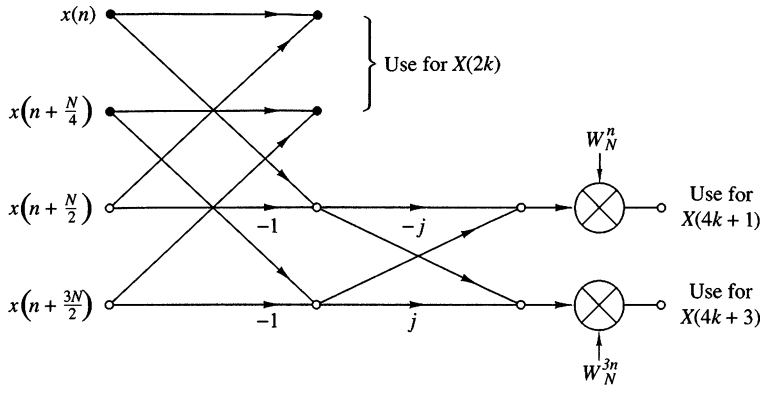


Figure 8.1.16 Butterfly for SRFFT algorithm.

The bottom 16 points of stage B are in the form of two eight-point DFTs. Hence each eight-point DFT is decomposed into a four-point, radix-2 DFT and a four-point, radix-4 DFT. In the final stage, the computations involve the combination of two-point sequences.

Table 8.2 presents a comparison of the number of *nontrivial* real multiplications and additions required to perform an N-point DFT with complex-valued data, using a radix-2, radix-4, radix-8, and a split-radix FFT. Note that the SRFFT algorithm requires the lowest number of multiplication and additions. For this reason, it is preferable in many practical applications.

Another type of SRFFT algorithm has been developed by Price (1990). Its relation to Duhamel's algorithm described previously can be seen by noting that the radix-4 DFT terms $X(4k + 1)$ and $X(4k + 3)$ involve the $N/4$-point DFTs of the sequences $[g_1(n) - jg_2(n)]W_N^n$ and $[g_1(n) + jg_2(n)]W_N^{3n}$, respectively. In effect, the sequences $g_1(n)$ and $g_2(n)$ are multiplied by the factor (vector) $(1, -j) = (1, W_{32}^8)$

TABLE 8.2 Number of Nontrivial Real Multiplications and Additions to Compute an N-point Complex DFT

	Real Multiplications				Real Additions			
N	Radix 2	Radix 4	Radix 8	Split Radix	Radix 2	Radix 4	Radix 8	Split Radix
16	24	20		20	152	148		148
32	88			68	408			388
64	264	208	204	196	1,032	976	972	964
128	712			516	2,504			2,308
256	1,800	1,392		1,284	5,896	5,488		5,380
512	4,360		3,204	3,076	13,566		12,420	12,292
1,024	10,248	7,856		7,172	30,728	28,336		27,652

Source: Extracted from Duhamel (1986).

and by W_N^n for the computation of $X(4k+1)$, while the computation of $X(4k+3)$ involves the factor $(1, j) = (1, W_{32}^{-8})$ and W_N^{3n}. Instead, one can rearrange the computation so that the factor for $X(4k+3)$ is $(-j, -1) = -(W_{32}^{-8}, 1)$. As a result of this phase rotation, the phase factors in the computation of $X(4k+3)$ become exactly the same as those for $X(4k+1)$, except that they occur in mirror image order. For example, at stage B of Fig. 8.1.15, the phase factors $W^{21}, W^{18}, \ldots, W^3$ are replaced by $W^1, W^2, \ldots, W^7$, respectively. This mirror-image symmetry occurs at every subsequent stage of the algorithm. As a consequence, the number of phase factors that must be computed and stored is reduced by a factor of 2 in comparison to Duhamel's algorithm. The resulting algorithm is called the "mirror" FFT (MFFT) algorithm.

An additional factor-of-2 savings in storage of phase factors can be obtained by introducing a 90° phase offset at the midpoint of each factor array, which can be removed if necessary at the output of the SRFFT computation. The incorporation of this improvement into the SRFFT (or the MFFT) results in another algorithm, also due to Price (1990), called the "phase" FFT (PFFT) algorithm.

8.1.6 Implementation of FFT Algorithms

Now that we have described the basic radix-2 and radix-4 FFT algorithms, let us consider some of the implementation issues. Our remarks apply directly to radix-2 algorithms, although similar comments may be made about radix-4 and higher-radix algorithms.

Basically, the radix-2 FFT algorithm consists of taking two data points at a time from memory, performing the butterfly computations and returning the resulting numbers to memory. This procedure is repeated many times $((N \log_2 N)/2$ times) in the computation of an N-point DFT.

The butterfly computations require the phase factors $\{W_N^k\}$ at various stages in either natural or bit-reversed order. In an efficient implementation of the algorithm, the phase factors are computed once and stored in a table, either in normal order or in bit-reversed order, depending on the specific implementation of the algorithm.

Memory requirement is another factor that must be considered. If the computations are performed in place, the number of memory locations required is $2N$ since the numbers are complex. However, we can instead double the memory to 4N, thus simplifying the indexing and control operations in the FFT algorithms. In this case we simply alternate in the use of the two sets of memory locations from one stage of the FFT algorithm to the other. Doubling of the memory also allows us to have both the input sequence and the output sequence in normal order.

There are a number of other implementation issues regarding indexing, bit reversal, and the degree of parallelism in the computations. To a large extent, these issues are a function of the specific algorithm and the type of implementation, namely, a hardware or software implementation. In implementations based on a fixed-point arithmetic, or floating-point arithmetic on small machines, there is also the issue of round-off errors in the computation. This topic is considered in Section 8.4.

Although the FFT algorithms described previously were presented in the context of computing the DFT efficiently, they can also be used to compute the IDFT, which is

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} \tag{8.1.63}$$

The only difference between the two transforms is the normalization factor $1/N$ and the sign of the phase factor W_N. Consequently, an FFT algorithm for computing the DFT can be converted to an FFT algorithm for computing the IDFT by changing the sign on all the phase factors and dividing the final output of the algorithm by N.

In fact, if we take the decimation-in-time algorithm that we described in Section 8.1.3, reverse the direction of the flow graph, change the sign on the phase factors, interchange the output and input, and finally, divide the output by N, we obtain a decimation-in-frequency FFT algorithm for computing the IDFT. On the other hand, if we begin with the decimation-in-frequency FFT algorithm described in Section 8.1.3 and repeat the changes described above, we obtain a decimation-in-time FFT algorithm for computing the IDFT. Thus it is a simple matter to devise FFT algorithms for computing the IDFT.

Finally, we note that the emphasis in our discussion of FFT algorithms was on radix-2, radix-4, and split-radix algorithms. These are by far the most widely used in practice. When the number of data points is not a power of 2 or 4, it is a simple matter to pad the sequence $x(n)$ with zeros such that $N = 2^\nu$ or $N = 4^\nu$.

The measure of complexity for FFT algorithms that we have emphasized is the required number of arithmetic operations (multiplications and additions). Although this is a very important benchmark for computational complexity, there are other issues to be considered in practical implementation of FFT algorithms. These include the architecture of the processor, the available instruction set, the data structures for storing phase factors, and other considerations.

For general-purpose computers, where the cost of the numerical operations dominates, radix-2, radix-4, and split-radix FFT algorithms are good candidates. However, in the case of special-purpose digital signal processors, featuring single-cycle multiply-and-accumulate operation, bit-reversed addressing, and a high degree of instruction parallelism, the structural regularity of the algorithm is equally as important as arithmetic complexity. Hence for DSP processors, radix-2 or radix-4 decimation-in-frequency FFT algorithms are preferable in terms of speed and accuracy. The irregular structure of the SRFFT may render it less suitable for implementation on digital signal processors. Structural regularity is also important in the implementation of FFT algorithms on vector processors, multiprocessors, and in VLSI. Interprocessor communication is an important consideration in such implementations on parallel processors.

In conclusion, we have presented several important considerations in the implementation of FFT algorithms. Advances in digital signal processing technology, in hardware and software, will continue to influence the choice among FFT algorithms for various practical applications.

8.2 Applications of FFT Algorithms

The FFT algorithms described in the preceding section find application in a variety of areas, including linear filtering, correlation, and spectrum analysis. Basically, the FFT algorithm is used as an efficient means to compute the DFT and the IDFT.

In this section we consider the use of the FFT algorithm in linear filtering and in the computation of the crosscorrelation of two sequences. The use of the FFT in spectrum estimation is considered in Chapter 14. In addition we illustrate how to enhance the efficiency of the FFT algorithm by forming complex-valued sequences from real-valued sequences prior to the computation of the DFT.

8.2.1 Efficient Computation of the DFT of Two Real Sequences

The FFT algorithm is designed to perform complex multiplications and additions, even though the input data may be real valued. The basic reason for this situation is that the phase factors are complex and hence, after the first stage of the algorithm, all variables are basically complex valued.

In view of the fact that the algorithm can handle complex-valued input sequences, we can exploit this capability in the computation of the DFT of two real-valued sequences.

Suppose that $x_1(n)$ and $x_2(n)$ are two real-valued sequences of length N, and let $x(n)$ be a complex-valued sequence defined as

$$x(n) = x_1(n) + jx_2(n), \qquad 0 \le n \le N-1 \tag{8.2.1}$$

The DFT operation is linear and hence the DFT of $x(n)$ can be expressed as

$$X(k) = X_1(k) + jX_2(k) \tag{8.2.2}$$

The sequences $x_1(n)$ and $x_2(n)$ can be expressed in terms of $x(n)$ as follows:

$$x_1(n) = \frac{x(n) + x^*(n)}{2} \tag{8.2.3}$$

$$x_2(n) = \frac{x(n) - x^*(n)}{2j} \tag{8.2.4}$$

Hence the DFTs of $x_1(n)$ and $x_2(n)$ are

$$X_1(k) = \frac{1}{2}\{\text{DFT}[x(n)] + \text{DFT}[x^*(n)]\} \tag{8.2.5}$$

$$X_2(k) = \frac{1}{2j}\{\text{DFT}[x(n)] - \text{DFT}[x^*(n)]\} \tag{8.2.6}$$

Recall that the DFT of $x^*(n)$ is $X^*(N-k)$. Therefore,

$$X_1(k) = \frac{1}{2}[X(k) + X^*(N-k)] \tag{8.2.7}$$

$$X_2(k) = \frac{1}{j2}[X(k) - X^*(N-k)] \tag{8.2.8}$$

Thus, by performing a single DFT on the complex-valued sequence $x(n)$, we have obtained the DFT of the two real sequences with only a small amount of additional computation that is involved in computing $X_1(k)$ and $X_2(k)$ from $X(k)$ by use of (8.2.7) and (8.2.8).

8.2.2 Efficient Computation of the DFT of a $2N$-Point Real Sequence

Suppose that $g(n)$ is a real-valued sequence of $2N$ points. We now demonstrate how to obtain the $2N$-point DFT of $g(n)$ from computation of one N-point DFT involving complex-valued data. First, we define

$$\begin{aligned} x_1(n) &= g(2n) \\ x_2(n) &= g(2n+1) \end{aligned} \tag{8.2.9}$$

Thus we have subdivided the $2N$-point real sequence into two N-point real sequences. Now we can apply the method described in the preceding section.

Let $x(n)$ be the N-point complex-valued sequence

$$x(n) = x_1(n) + jx_2(n) \tag{8.2.10}$$

From the results of the preceding section, we have

$$\begin{aligned} X_1(k) &= \frac{1}{2}[X(k) + X^*(N-k)] \\ X_2(k) &= \frac{1}{2j}[X(k) - X^*(N-k)] \end{aligned} \tag{8.2.11}$$

Finally, we must express the $2N$-point DFT in terms of the two N-point DFTs, $X_1(k)$ and $X_2(k)$. To accomplish this, we proceed as in the decimation-in-time FFT algorithm, namely,

$$\begin{aligned} G(k) &= \sum_{n=0}^{N-1} g(2n) W_{2N}^{2nk} + \sum_{n=0}^{N-1} g(2n+1) W_{2N}^{(2n+1)k} \\ &= \sum_{n=0}^{N-1} x_1(n) W_N^{nk} + W_{2N}^k \sum_{n=0}^{N-1} x_2(n) W_N^{nk} \end{aligned}$$

Consequently,

$$\begin{aligned} G(k) &= X_1(k) + W_2^k N X_2(k), \qquad k = 0, 1, \ldots, N-1 \\ G(k+N) &= X_1(k) - W_2^k N X_2(k), \qquad k = 0, 1, \ldots, N-1 \end{aligned} \tag{8.2.12}$$

Thus we have computed the DFT of a $2N$-point real sequence from one N-point DFT and some additional computation as indicated by (8.2.11) and (8.2.12).

8.2.3 Use of the FFT Algorithm in Linear Filtering and Correlation

An important application of the FFT algorithm is in FIR linear filtering of long data sequences. In Chapter 7 we described two methods, the overlap-add and the overlap-save methods for filtering a long data sequence with an FIR filter, based on the use of the DFT. In this section we consider the use of these two methods in conjunction with the FFT algorithm for computing the DFT and the IDFT.

Let $h(n)$, $0 \leq n \leq M-1$, be the unit sample response of the FIR filter and let $x(n)$ denote the input data sequence. The block size of the FFT algorithm is N, where $N = L + M - 1$ and L is the number of new data samples being processed by the filter. We assume that for any given value of M, the number L of data samples is selected so that N is a power of 2. For purposes of this discussion, we consider only radix-2 FFT algorithms.

The N-point DFT of $h(n)$, which is padded by $L-1$ zeros, is denoted as $H(k)$. This computation is performed once via the FFT and the resulting N complex numbers are stored. To be specific we assume that the decimation-in-frequency FFT algorithm is used to compute $H(k)$. This yields $H(k)$ in bit-reversed order, which is the way it is stored in memory.

In the overlap-save method, the first $M-1$ data points of each data block are the last $M-1$ data points of the previous data block. Each data block contains L new data points, such that $N = L + M - 1$. The N-point DFT of each data block is performed by the FFT algorithm. If the decimation-in-frequency algorithm is employed, the input data block requires no shuffling and the values of the DFT occur in bit-reversed order. Since this is exactly the order of $H(k)$, we can multiply the DFT of the data, say $X_m(k)$, with $H(k)$, and thus the result

$$Y_m(k) = H(k)X_m(k)$$

is also in bit-reversed order.

The inverse DFT (IDFT) can be computed by use of an FFT algorithm that takes the input in bit-reversed order and produces an output in normal order. Thus there is no need to shuffle any block of data in computing either the DFT or the IDFT.

If the overlap-add method is used to perform the linear filtering, the computational method using the FFT algorithm is basically the same. The only difference is that the N-point data blocks consist of L new data points and $M-1$ additional zeros. After the IDFT is computed for each data block, the N-point filtered blocks are overlapped as indicated in Section 7.3.2, and the $M-1$ overlapping data points between successive output records are added together.

Let us assess the computational complexity of the FFT method for linear filtering. For this purpose, the one-time computation of $H(k)$ is insignificant and can be ignored. Each FFT requires $(N/2)\log_2 N$ complex multiplications and $N \log_2 N$ additions. Since the FFT is performed twice, once for the DFT and once for the IDFT, the computational burden is $N \log_2 N$ complex multiplications and $2N \log_2 N$ additions. There are also N complex multiplications and $N-1$ additions required to compute $Y_m(k)$. Therefore, we have $(N \log_2 2N)/L$ complex multiplications per output data point and approximately $(2N \log_2 2N)/L$ additions per output data point.

The overlap-add method requires an incremental increase of $(M-1)/L$ in the number of additions.

By way of comparison, a direct-form realization of the FIR filter involves M real multiplications per output point if the filter is not linear phase, and $M/2$ if it is linear phase (symmetric). Also, the number of additions is $M-1$ per output point (see Sec. 10.2).

It is interesting to compare the efficiency of the FFT algorithm with the direct form realization of the FIR filter. Let us focus on the number of multiplications, which are more time consuming than additions. Suppose that $M = 128 = 2^7$ and $N = 2^\nu$. Then the number of complex multiplications per output point for an FFT size of $N = 2^\nu$ is

$$c(\nu) = \frac{N \log_2 2N}{L} = \frac{2^\nu(\nu+1)}{N-M+1}$$
$$\approx \frac{2^\nu(\nu+1)}{2^\nu - 2^7}$$

The values of $c(\nu)$ for different values of ν are given in Table 8.3. We observe that there is an optimum value of ν which minimizes $c(\nu)$. For the FIR filter of size $M = 128$, the optimum occurs at $\nu = 10$.

We should emphasize that $c(\nu)$ represents the number of complex multiplications for the FFT-based method. The number of real multiplications is four times this number. However, even if the FIR filter has linear phase (see Sec. 10.2), the number of computations per output point is still less with the FFT-based method. Furthermore, the efficiency of the FFT method can be improved by computing the DFT of two successive data blocks simultaneously, according to the method just described. Consequently, the FFT-based method is indeed superior from a computational point of view when the filter length is relatively large.

The computation of the cross correlation between two sequences by means of the FFT algorithm is similar to the linear FIR filtering problem just described. In practical applications involving crosscorrelation, at least one of the sequences has finite duration and is akin to the impulse response of the FIR filter. The second sequence may be a long sequence which contains the desired sequence corrupted by additive noise. Hence the second sequence is akin to the input to the FIR filter.

TABLE 8.3 Computational Complexity

Size of FFT $\nu = \log_2 N$	$c(\nu)$ Number of Complex Multiplications per Output Point
9	13.3
10	12.6
11	12.8
12	13.4
14	15.1

By time reversing the first sequence and computing its DFT, we have reduced the cross correlation to an equivalent convolution problem (i.e., a linear FIR filtering problem). Therefore, the methodology we developed for linear FIR filtering by use of the FFT applies directly.

8.3 A Linear Filtering Approach to Computation of the DFT

The FFT algorithm takes N points of input data and produces an output sequence of N points corresponding to the DFT of the input data. As we have shown, the radix-2 FFT algorithm performs the computation of the DFT in $(N/2)\log_2 N$ multiplications and $N \log_2 N$ additions for an N-point sequence.

There are some applications where only a selected number of values of the DFT are desired, but the entire DFT is not required. In such a case, the FFT algorithm may no longer be more efficient than a direct computation of the desired values of the DFT. In fact, when the desired number of values of the DFT is less than $\log_2 N$, a direct computation of the desired values is more efficient.

The direct computation of the DFT can be formulated as a linear filtering operation on the input data sequence. As we will demonstrate, the linear filter takes the form of a parallel bank of resonators where each resonator selects one of the frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, corresponding to the N frequencies in the DFT.

There are other applications in which we require the evaluation of the z-transform of a finite-duration sequence at points other than the unit circle. If the set of desired points in the z-plane possesses some regularity, it is possible to also express the computation of the z-transform as a linear filtering operation. In this connection, we introduce another algorithm, called the chirp-z transform algorithm, which is suitable for evaluating the z-transform of a set of data on a variety of contours in the z-plane. This algorithm is also formulated as a linear filtering of a set of input data. As a consequence, the FFT algorithm can be used to compute the chirp-z transform and thus to evaluate the z-transform at various contours in the z-plane, including the unit circle.

8.3.1 The Goertzel Algorithm

The Goertzel algorithm exploits the periodicity of the phase factors $\{ W_N^k \}$ and allows us to express the computation of the DFT as a linear filtering operation. Since $W_N^{-kN} = 1$, we can multiply the DFT by this factor. Thus

$$X(k) = W_N^{-kN} \sum_{m=0}^{N-1} x(m) W_N^{km} = \sum_{m=0}^{N-1} x(m) W_N^{-k(N-m)} \tag{8.3.1}$$

We note that (8.3.1) is in the form of a convolution. Indeed, if we define the sequence $y_k(n)$ as

$$y_k(n) = \sum_{m=0}^{N-1} x(m) W_N^{-k(n-m)} \tag{8.3.2}$$

then it is clear that $y_k(n)$ is the convolution of the finite-duration input sequence $x(n)$ of length N with a filter that has an impulse response

$$h_k(n) = W_N^{-kn} u(n) \tag{8.3.3}$$

The output of this filter at $n = N$ yields the value of the DFT at the frequency $\omega_k = 2\pi k/N$. That is,

$$X(k) = y_k(n)|_{n=N} \tag{8.3.4}$$

as can be verified by comparing (8.3.1) with (8.3.2).

The filter with impulse response $h_k(n)$ has the system function

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \tag{8.3.5}$$

This filter has a pole on the unit circle at the frequency $\omega_k = 2\pi k/N$. Thus, the entire DFT can be computed by passing the block of input data into a parallel bank of N single-pole filters (resonators), where each filter has a pole at the corresponding frequency of the DFT.

Instead of performing the computation of the DFT as in (8.3.2), via convolution, we can use the difference equation corresponding to the filter given by (8.3.5) to compute $y_k(n)$ recursively. Thus we have

$$y_k(n) = W_N^{-k} y_k(n-1) + x(n), \qquad y_k(-1) = 0 \tag{8.3.6}$$

The desired output is $X(k) = y_k(N)$, for $k = 0, 1, \ldots, N-1$. To perform this computation, we can compute once and store the phase factors W_N^{-k}.

The complex multiplications and additions inherent in (8.3.6) can be avoided by combining the pairs of resonators possessing complex-conjugate poles. This leads to two-pole filters with system functions of the form

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos(2\pi k/N) z^{-1} + z^{-2}} \tag{8.3.7}$$

The direct form II realization of the system illustrated in Fig. 8.3.1 is described by the difference equations

$$v_k(n) = 2\cos\frac{2\pi k}{N} v_k(n-1) - v_k(n-2) + x(n) \tag{8.3.8}$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \tag{8.3.9}$$

with initial conditions $v_k(-1) = v_k(-2) = 0$.

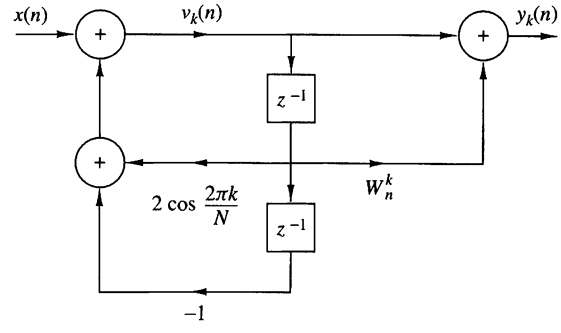


Figure 8.3.1
Direct form II realization of two-pole resonator for computing the DFT.

The recursive relation in (8.3.8) is iterated for $n = 0, 1, \ldots, N$, but the equation in (8.3.9) is computed only once at time $n = N$. Each iteration requires one real multiplication and two additions. Consequently, for a real input sequence $x(n)$, this algorithm requires $N + 1$ real multiplications to yield not only $X(k)$ but also, due to symmetry, the value of $X(N - k)$.

The Goertzel algorithm is particularly attractive when the DFT is to be computed at a relatively small number M of values, where $M \leq \log_2 N$. Otherwise, the FFT algorithm is a more efficient method.

8.3.2 The Chirp-z Transform Algorithm

The DFT of an N-point data sequence $x(n)$ has been viewed as the z-transform of $x(n)$ evaluated at N equally spaced points on the unit circle. It has also been viewed as N equally spaced samples of the Fourier transform of the data sequence $x(n)$. In this section we consider the evaluation of $X(z)$ on other contours in the z-plane, including the unit circle.

Suppose that we wish to compute the values of the z-transform of $x(n)$ at a set of points $\{z_k\}$. Then,

$$X(z_k) = \sum_{n=0}^{N-1} x(n) z_k^{-n}, \qquad k = 0, 1, \ldots, L - 1 \tag{8.3.10}$$

For example, if the contour is a circle of radius r and the z_k are N equally spaced points, then

$$\begin{aligned} z_k &= re^{j2\pi kn/N}, \qquad k = 0, 1, 2, \ldots, N - 1 \\ X(z_k) &= \sum_{n=0}^{N-1} [x(n)r^{-n}]e^{-j2\pi kn/N}, \qquad k = 0, 1, 2, \ldots, N - 1 \end{aligned} \tag{8.3.11}$$

In this case the FFT algorithm can be applied on the modified sequence $x(n)r^{-n}$.

More generally, suppose that the points z_k in the z-plane fall on an arc which begins at some point

$$z_0 = r_0 e^{j\theta_0}$$

and spirals either in toward the origin or out away from the origin such that the points $\{z_k\}$ are defined as

$$z_k = r_0 e^{j\theta_0}(R_0 e^{j\phi_0})^k, \qquad k = 0, 1, \ldots, L-1 \tag{8.3.12}$$

Note that if $R_0 < 1$, the points fall on a contour that spirals toward the origin, and if $R_0 > 1$, the contour spirals away from the origin. If $R_0 = 1$, the contour is a circular arc of radius r_0. If $r_0 = 1$ and $R_0 = 1$, the contour is an arc of the unit circle. The latter contour would allow us to compute the frequency content of the sequence $x(n)$ at a dense set of L frequencies in the range covered by the arc without having to compute a large DFT, that is, a DFT of the sequence $x(n)$ padded with many zeros to obtain the desired resolution in frequency. Finally, if $r_0 = R_0 = 1$, $\theta_0 = 0$, $\phi_0 = 2\pi/N$, and $L = N$, the contour is the entire unit circle and the frequencies are those of the DFT. The various contours are illustrated in Fig. 8.3.2.

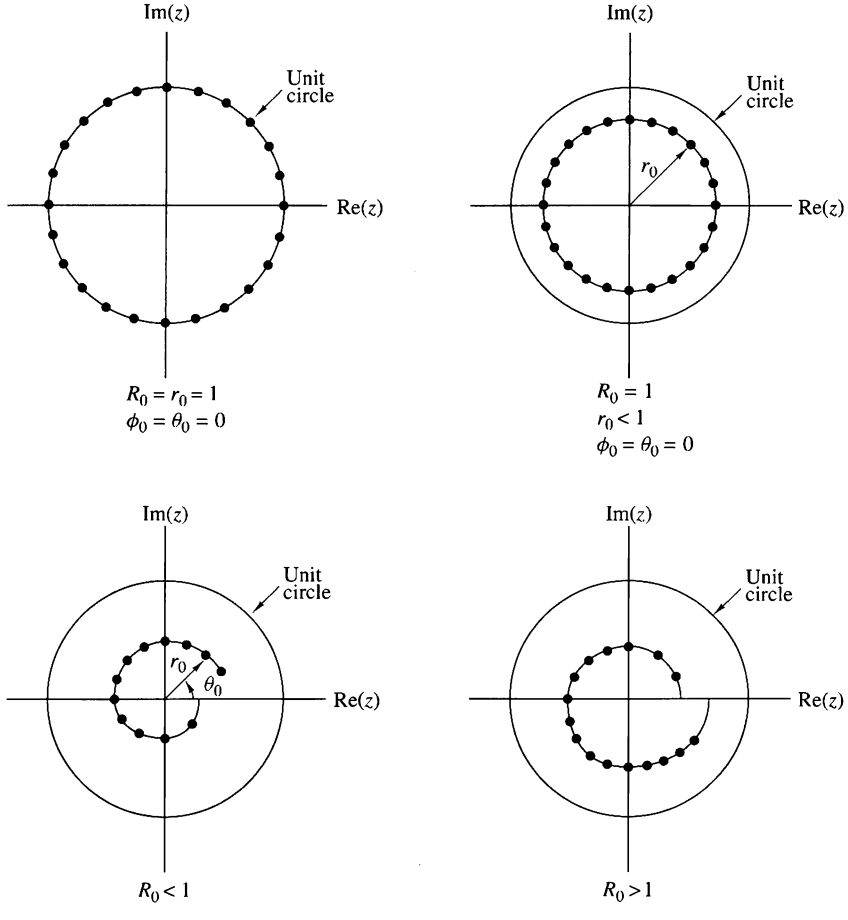


Figure 8.3.2 Some examples of contours on which we may evaluate the z-transform.

When points $\{z_k\}$ in (8.3.12) are substituted into the expression for the z-transform, we obtain

$$\begin{aligned} X(z_k) &= \sum_{n=0}^{N-1} x(n) z_k^{-n} \\ &= \sum_{n=0}^{N-1} x(n)(r_0 e^{j\theta_0})^{-n} V^{-nk} \end{aligned} \tag{8.3.13}$$

where, by definition,

$$V = R_0 e^{j\phi_0} \tag{8.3.14}$$

We can express (8.3.13) in the form of a convolution, by noting that

$$nk = \tfrac{1}{2}[n^2 + k^2 - (k-n)^2] \tag{8.3.15}$$

Substitution of (8.3.15) into (8.3.13) yields

$$X(z_k) = V^{-k^2/2} \sum_{n=0}^{N-1} [x(n)(r_0 e^{j\theta_0})^{-n} V^{-n^2/2}] V^{(k-n)^2/2} \tag{8.3.16}$$

Let us define a new sequence $g(n)$ as

$$g(n) = x(n)(r_0 e^{j\theta_0})^{-n} V^{-n^2/2} \tag{8.3.17}$$

Then (8.3.16) can be expressed as

$$X(z_k) = V^{-k^2/2} \sum_{n=0}^{N-1} g(n) V^{(k-n)^2/2} \tag{8.3.18}$$

The summation in (8.3.18) can be interpreted as the convolution of the sequence $g(n)$ with the impulse response $h(n)$ of a filter, where

$$h(n) = V^{n^2/2} \tag{8.3.19}$$

Consequently, (8.3.18) may be expressed as

$$\begin{aligned} X(z_k) &= V^{-k^2/2} y(k) \\ &= \frac{y(k)}{h(k)}, \qquad k = 0, 1, \ldots, L-1 \end{aligned} \tag{8.3.20}$$

where $y(k)$ is the output of the filter

$$y(k) = \sum_{n=0}^{N-1} g(n) h(k-n), \qquad k = 0, 1, \ldots, L-1 \tag{8.3.21}$$

We observe that both $h(n)$ and $g(n)$ are complex-valued sequences.

The sequence $h(n)$ with $R_0 = 1$ has the form of a complex exponential with argument $\omega n = n^2\phi_0/2 = (n\phi_0/2)n$. The quantity $n\phi_0/2$ represents the frequency of the complex exponential signal, which increases linearly with time. Such signals are used in radar systems and are called *chirp signals*. Hence the z-transform evaluated as in (8.3.18) is called the *chirp-z transform*.

The linear convolution in (8.3.21) is most efficiently done by use of the FFT algorithm. The sequence $g(n)$ is of length N. However, $h(n)$ has infinite duration. Fortunately, only a portion $h(n)$ is required to compute the L values of $X(z)$.

Since we will compute the convolution in (8.3.21) via the FFT, let us consider the circular convolution of the N-point sequence $g(n)$ with an M-point section of $h(n)$, where $M > N$. In such a case, we know that the first $N-1$ points contain aliasing and that the remaining $M-N+1$ points are identical to the result that would be obtained from a linear convolution of $h(n)$ with $g(n)$. In view of this, we should select a DFT of size

$$M = L + N - 1$$

which would yield L valid points and $N-1$ points corrupted by aliasing.

The section of $h(n)$ that is needed for this computation corresponds to the values of $h(n)$ for $-(N-1) \le n \le (L-1)$, which is of length $M = L+N-1$, as observed from (8.3.21). Let us define the sequence $h_1(n)$ of length M as

$$h_1(n) = h(n-N+1), \qquad n = 0, 1, \ldots, M-1 \tag{8.3.22}$$

and compute its M-point DFT via the FFT algorithm to obtain $H_1(k)$. From $x(n)$ we compute $g(n)$ as specified by (8.3.17), pad $g(n)$ with $L-1$ zeros, and compute its M-point DFT to yield $G(k)$. The IDFT of the product $Y_1(k) = G(k)H_1(k)$ yields the M-point sequence $y_1(n)$, $n = 0, 1, \ldots, M-1$. The first $N-1$ points of $y_1(n)$ are corrupted by aliasing and are discarded. The desired values are $y_1(n)$ for $N-1 \le n \le M-1$, which correspond to the range $0 \le n \le L-1$ in (8.3.21), that is,

$$y(n) = y_1(n+N-1), \qquad n = 0, 1, \ldots, L-1 \tag{8.3.23}$$

Alternatively, we can define a sequence $h_2(n)$ as

$$h_2(n) = \begin{cases} h(n), & 0 \le n \le L-1 \\ h(n-N-L+1), & L \le n \le M-1 \end{cases} \tag{8.3.24}$$

The M-point DFT of $h_2(n)$ yields $H_2(k)$, which when multiplied by $G(k)$ yields $Y_2(k) = G(k)H_2(k)$. The IDFT of $Y_2(k)$ yields the sequence $y_2(n)$ for $0 \le n \le M-1$. Now the desired values of $y_2(n)$ are in the range $0 \le n \le L-1$, that is,

$$y(n) = y_2(n), \qquad n = 0, 1, \ldots, L-1 \tag{8.3.25}$$

Finally, the complex values $X(z_k)$ are computed by dividing $y(k)$ by $h(k)$, $k = 0, 1, \ldots, L-1$, as specified by (8.3.20).

In general, the computational complexity of the chirp-z transform algorithm described above is of the order of $M \log_2 M$ complex multiplications, where $M = N + L - 1$. This number should be compared with the product, $N \cdot L$, the number of computations required by direct evaluation of the z-transform. Clearly, if L is small, direct computation is more efficient. However, if L is large, then the chirp-z transform algorithm is more efficient.

The chirp-z transform method has been implemented in hardware to compute the DFT of signals. For the computation of the DFT, we select $r_0 = R_0 = 1$, $\theta_0 = 0$, $\phi_0 = 2\pi/N$, and $L = N$. In this case

$$\begin{aligned} V^{-n^2/2} &= e^{-j\pi n^2/N} \\ &= \cos\frac{\pi n^2}{N} - j\sin\frac{\pi n^2}{N} \end{aligned} \tag{8.3.26}$$

The chirp filter with impulse response

$$\begin{aligned} h(n) &= V^{n^2/2} \\ &= \cos\frac{\pi n^2}{N} + j\sin\frac{\pi n^2}{N} \\ &= h_r(n) + jh_i(n) \end{aligned} \tag{8.3.27}$$

has been implemented as a pair of FIR filters with coefficients $h_r(n)$ and $h_i(n)$, respectively. Both *surface acoustic wave* (SAW) devices and *charge coupled devices*

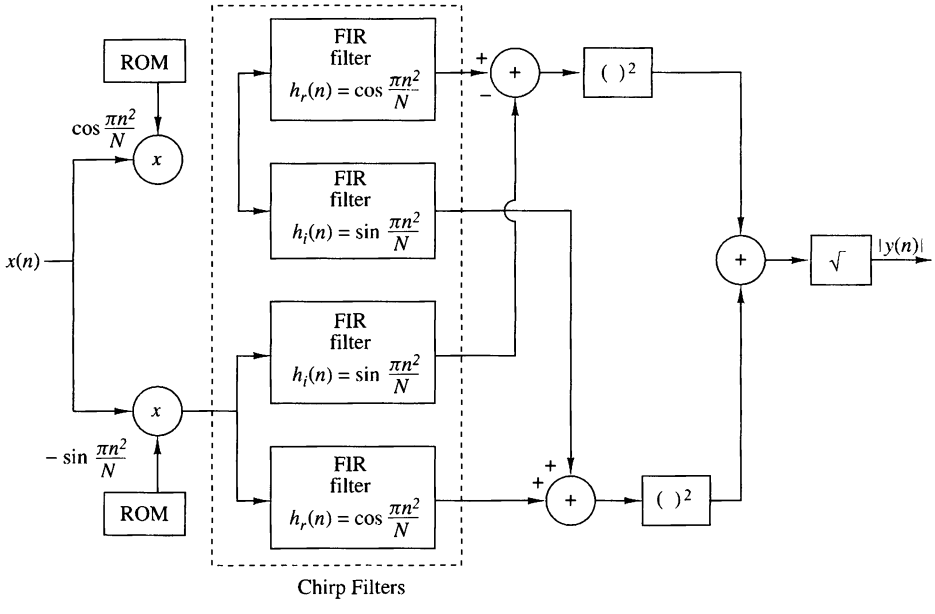


Figure 8.3.3 Block diagram illustrating the implementation of the chirp-z transform for computing the DFT (magnitude only).

(CCD) have been used in practice for the FIR filters. The cosine and sine sequences given in (8.3.26) needed for the premultiplications and postmultiplications are usually stored in a read-only memory (ROM). Furthermore, we note that if only the magnitude of the DFT is desired, the postmultiplications are unnecessary. In this case,

$$|X(z_k)| = |y(k)|, \qquad k = 0, 1, \ldots, N-1 \tag{8.3.28}$$

as illustrated in Fig. 8.3.3. Thus the linear FIR filtering approach using the chirp-z transform has been implemented for the computation of the DFT.

8.4 Quantization Effects in the Computation of the DFT[1]

As we have observed in our previous discussions, the DFT plays an important role in many digital signal processing applications, including FIR filtering, the computation of the correlation between signals, and spectral analysis. For this reason it is important for us to know the effect of quantization errors in its computation. In particular, we shall consider the effect of round-off errors due to the multiplications performed in the DFT with fixed-point arithmetic.

The model that we shall adopt for characterizing round-off errors in multiplication is the additive white noise model that we use in the statistical analysis of round-off errors in IIR and FIR filters (see Fig. 9.6.8). Although the statistical analysis is performed for rounding, the analysis can be easily modified to apply to truncation in two's-complement arithmetic (see Sec. 9.4.3).

Of particular interest is the analysis of round-off errors in the computation of the DFT via the FFT algorithm. However, we shall first establish a benchmark by determining the round-off errors in the direct computation of the DFT.

8.4.1 Quantization Errors in the Direct Computation of the DFT

Given a finite-duration sequence $\{x(n)\}$, $0 \le n \le N-1$, the DFT of $\{x(n)\}$ is defined as

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0, 1, \ldots, N-1 \tag{8.4.1}$$

where $W_N = e^{-j2\pi/N}$. We assume that in general, $\{x(n)\}$ is a complex-valued sequence. We also assume that the real and imaginary components of $\{x(n)\}$ and $\{W_N^{kn}\}$ are represented by b bits. Consequently, the computation of the product $x(n)W_N^{kn}$ requires four real multiplications. Each real multiplication is rounded from $2b$ bits to b bits, and hence there are four quantization errors for each complex-valued multiplication.

In the direct computation of the DFT, there are N complex-valued multiplications for each point in the DFT. Therefore, the total number of real multiplications in the computation of a single point in the DFT is $4N$. Consequently, there are $4N$ quantization errors.

[1] It is recommended that the reader review Section 9.5 prior to reading this section.

Let us evaluate the variance of the quantization errors in a fixed-point computation of the DFT. First, we make the following assumptions about the statistical properties of the quantization errors.

1. The quantization errors due to rounding are uniformly distributed random variables in the range $(-\Delta/2, \Delta/2)$ where $\Delta = 2^{-b}$.
2. The $4N$ quantization errors are mutually uncorrelated.
3. The $4N$ quantization errors are uncorrelated with the sequence $\{x(n)\}$.

Since each of the quantization errors has a variance

$$\sigma_e^2 = \frac{\Delta^2}{12} = \frac{2^{-2b}}{12} \tag{8.4.2}$$

the variance of the quantization errors from the $4N$ multiplications is

$$\begin{aligned} \sigma_q^2 &= 4N\sigma_e^2 \\ &= \frac{N}{3} \cdot 2^{-2b} \end{aligned} \tag{8.4.3}$$

Hence the variance of the quantization error is proportional to the size of DFT. Note that when N is a power of 2 (i.e., $N = 2^\nu$), the variance can be expressed as

$$\sigma_q^2 = \frac{2^{-2(b-\nu/2)}}{3} \tag{8.4.4}$$

This expression implies that every fourfold increase in the size N of the DFT requires an additional bit in computational precision to offset the additional quantization errors.

To prevent overflow, the input sequence to the DFT requires scaling. Clearly, an upper bound on $|X(k)|$ is

$$|X(k)| \le \sum_{n=0}^{N-1} |x(n)| \tag{8.4.5}$$

If the dynamic range in addition is $(-1, 1)$, then $|X(k)| < 1$ requires that

$$\sum_{n=0}^{N-1} |x(n)| < 1 \tag{8.4.6}$$

If $|x(n)|$ is initially scaled such that $|x(n)| < 1$ for all n, then each point in the sequence can be divided by N to ensure that (8.4.6) is satisfied.

The scaling implied by (8.4.6) is extremely severe. For example, suppose that the signal sequence $\{x(n)\}$ is white and, after scaling, each value $|x(n)|$ of the sequence is uniformly distributed in the range $(-1/N, 1/N)$. Then the variance of the signal sequence is

$$\sigma_x^2 = \frac{(2/N)^2}{12} = \frac{1}{3N^2} \tag{8.4.7}$$

and the variance of the output DFT coefficients $|X(k)|$ is

$$\begin{aligned}\sigma_X^2 &= N\sigma_x^2 \\ &= \frac{1}{3N}\end{aligned} \tag{8.4.8}$$

Thus the signal-to-noise power ratio is

$$\frac{\sigma_X^2}{\sigma_q^2} = \frac{2^{2b}}{N^2} \tag{8.4.9}$$

We observe that the scaling is responsible for reducing the SNR by N and the combination of scaling and quantization errors results in a total reduction that is proportional to N^2. Hence scaling the input sequence $\{x(n)\}$ to satisfy (8.4.6) imposes a severe penalty on the signal-to-noise ratio in the DFT.

EXAMPLE 8.4.1

Use (8.4.9) to determine the number of bits required to compute the DFT of a 1024-point sequence with an SNR of 30 dB.

Solution. The size of the sequence is $N = 2^{10}$. Hence the SNR is

$$10\log_{10}\frac{\sigma_X^2}{\sigma_q^2} = 10\log_{10}2^{2b-20}$$

For an SNR of 30 dB, we have

$$3(2b - 20) = 30$$

$$b = 15 \text{ bits}$$

Note that the 15 bits is the precision for both multiplication and addition.

Instead of scaling the input sequence $\{x(n)\}$, suppose we simply require that $|x(n)| < 1$. Then we must provide a sufficiently large dynamic range for addition such that $|X(k)| < N$. In such a case, the variance of the sequence $\{|x(n)|\}$ is $\sigma_x^2 = \frac{1}{3}$, and hence the variance of $|X(k)|$ is

$$\sigma_X^2 = N\sigma_x^2 = \frac{N}{3} \tag{8.4.10}$$

Consequently, the SNR is

$$\frac{\sigma_X^2}{\sigma_q^2} = 2^{2b} \tag{8.4.11}$$

If we repeat the computation in Example 8.4.1, we find that the number of bits required to achieve an SNR of 30 dB is $b = 5$ bits. However, we need an additional 10 bits for the accumulator (the adder) to accommodate the increase in the dynamic range for addition. Although we did not achieve any reduction in the dynamic range for addition, we have managed to reduce the precision in multiplication from 15 bits to 5 bits, which is highly significant.

8.4.2 Quantization Errors in FFT Algorithms

As we have shown, the FFT algorithms require significantly fewer multiplications than the direct computation of the DFT. In view of this we might conclude that the computation of the DFT via an FFT algorithm will result in smaller quantization errors. Unfortunately, that is not the case, as we will demonstrate.

Let us consider the use of fixed-point arithmetic in the computation of a radix-2 FFT algorithm. To be specific, we select the radix-2, decimation-in-time algorithm illustrated in Fig. 8.4.1 for the case $N = 8$. The results on quantization errors that we obtain for this radix-2 FFT algorithm are typical of the results obtained with other radix-2 and higher radix algorithms.

We observe that each butterfly computation involves one complex-valued multiplication or, equivalently, four real multiplications. We ignore the fact that some butterflies contain a trivial multiplication by ± 1. If we consider the butterflies that affect the computation of any one value of the DFT, we find that, in general, there are $N/2$ in the first stage of the FFT, $N/4$ in the second stage, $N/8$ in the third state, and so on, until the last stage, where there is only one. Consequently, the number of

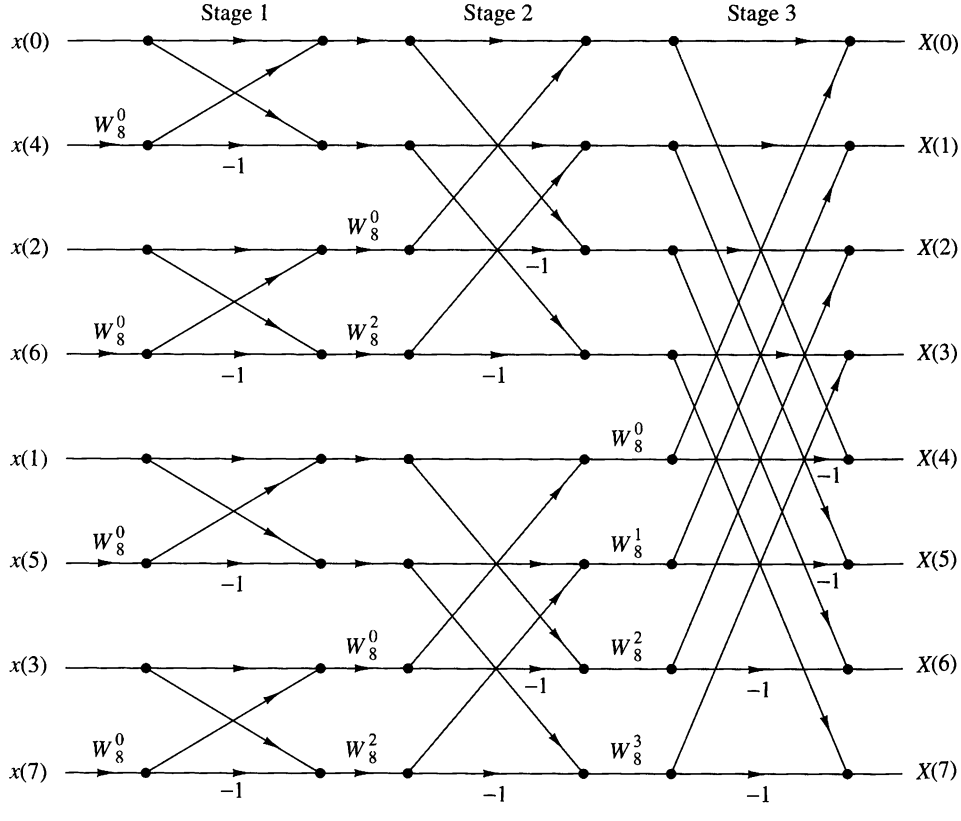


Figure 8.4.1 Decimation-in-time FFT algorithm.

butterflies per output point is

$$2^{\nu-1}+2^{\nu-2}+\cdots+2+1=2^{\nu-1}\left[1+\left(\tfrac{1}{2}\right)+\cdots+\left(\tfrac{1}{2}\right)^{\nu-1}\right]$$
$$=2^{\nu}[1-\left(\tfrac{1}{2}\right)^{\nu}]=N-1 \tag{8.4.12}$$

For example, the butterflies that affect the computation of $X(3)$ in the eight-point FFT algorithm of Fig. 8.4.1 are illustrated in Fig. 8.4.2.

The quantization errors introduced in each butterfly propagate to the output. Note that the quantization errors introduced in the first stage propagate through $(\nu-1)$ stages, those introduced in the second stage propagate through $(\nu-2)$ stages, and so on. As these quantization errors propagate through a number of subsequent stages, they are phase shifted (phase rotated) by the phase factors W_N^{kn}. These phase rotations do not change the statistical properties of the quantization errors and, in particular, the variance of each quantization error remains invariant.

If we assume that the quantization errors in each butterfly are uncorrelated with the errors in other butterflies, then there are $4(N-1)$ errors that affect the output of each point of the FFT. Consequently, the variance of the total quantization error at the output is

$$\sigma_q^2=4(N-1)\frac{\Delta^2}{12}\approx\frac{N\Delta^2}{3} \tag{8.4.13}$$

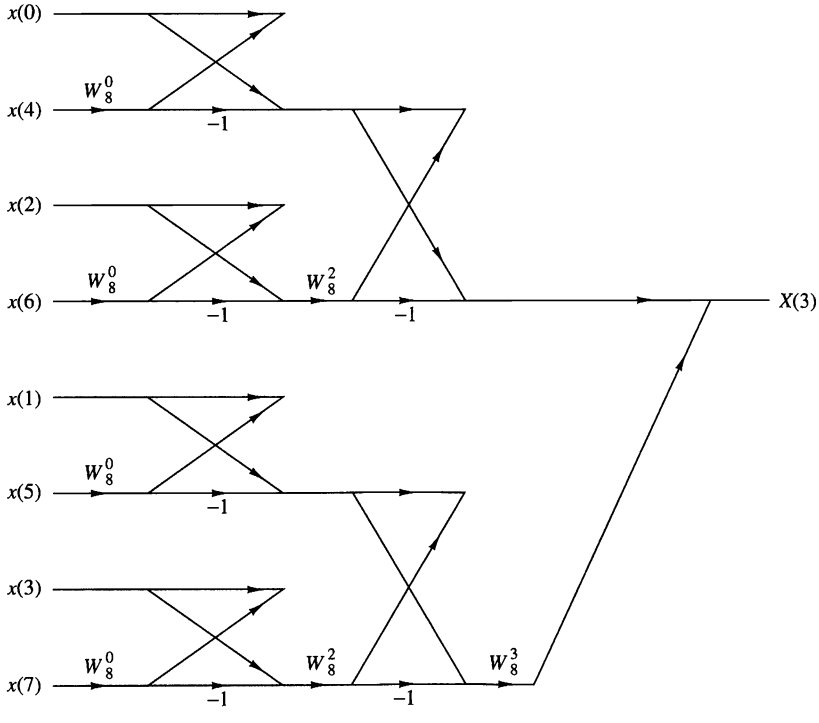


Figure 8.4.2 Butterflies that affect the computation of $X(3)$.

where $\Delta = 2^{-b}$. Hence

$$\sigma_q^2 = \frac{N}{3} \cdot 2^{-2b} \tag{8.4.14}$$

This is exactly the same result that we obtained for the direct computation of the DFT.

The result in (8.4.14) should not be surprising. In fact, the FFT algorithm does not reduce the number of multiplications required to compute a single point of the DFT. It does, however, exploit the periodicities in W_N^{kn} and thus reduces the number of multiplications in the computation of the entire block of N points in the DFT.

As in the case of the direct computation of the DFT, we must scale the input sequence to prevent overflow. Recall that if $|x(n)| < 1/N$, $0 \le n \le N-1$, then $|X(k)| < 1$ for $0 \le k \le N-1$. Thus overflow is avoided. With this scaling, the relations in (8.4.7), (8.4.8), and (8.4.9), obtained previously for the direct computation of the DFT, apply to the FFT algorithm as well. Consequently, the same SNR is obtained for the FFT.

Since the FFT algorithm consists of a sequence of stages, where each stage contains butterflies that involve pairs of points, it is possible to devise a different scaling strategy that is not as severe as dividing each input point by N. This alternative scaling strategy is motivated by the observation that the intermediate values $|X_n(k)|$ in the $n = 1, 2, \ldots, \nu$ stages of the FFT algorithm satisfy the conditions (see Problem 8.35)

$$\begin{aligned} \max[|X_{n+1}(k)|, |X_{n+1}(l)|] &\ge \max[|X_n(k)|, |X_n(l)|] \\ \max[|X_{n+1}(k)|, |X_{n+1}(l)|] &\le 2\max[|X_n(k)|, |X_n(l)|] \end{aligned} \tag{8.4.15}$$

In view of these relations, we can distribute the total scaling of $1/N$ into each of the stages of the FFT algorithm. In particular, if $|x(n)| < 1$, we apply a scale factor of $\frac{1}{2}$ in the first stage so that $|x(n)| < \frac{1}{2}$. Then the output of each subsequent stage in the FFT algorithm is scaled by $\frac{1}{2}$, so that after ν stages we have achieved an overall scale factor of $(\frac{1}{2})^\nu = 1/N$. Thus overflow in the computation of the DFT is avoided.

This scaling procedure does not affect the signal level at the output of the FFT algorithm, but it significantly reduces the variance of the quantization errors at the output. Specifically, each factor of $\frac{1}{2}$ reduces the variance of a quantization error term by a factor of $\frac{1}{4}$. Thus the $4(N/2)$ quantization errors introduced in the first stage are reduced in variance by $(\frac{1}{4})^{\nu-1}$, the $4(N/4)$ quantization errors introduced in the second stage are reduced in variance by $(\frac{1}{4})^{\nu-2}$, and so on. Consequently, the total variance of the quantization errors at the output of the FFT algorithm is

$$\begin{aligned} \sigma_q^2 &= \frac{\Delta^2}{12}\left\{4\left(\frac{N}{2}\right)\left(\frac{1}{4}\right)^{\nu-1} + 4\left(\frac{N}{4}\right)\left(\frac{1}{4}\right)^{\nu-2} + 4\left(\frac{N}{8}\right)\left(\frac{1}{4}\right)^{\nu-3} + \cdots + 4\right\} \\ &= \frac{\Delta^2}{3}\left\{\left(\frac{1}{2}\right)^{\nu-1} + \left(\frac{1}{2}\right)^{\nu-2} + \cdots + \frac{1}{2} + 1\right\} \\ &= \frac{2\Delta^2}{3}\left[1 - \left(\frac{1}{2}\right)^{\nu}\right] \approx \frac{2}{3} \cdot 2^{-2b} \end{aligned} \tag{8.4.16}$$

where the factor $(\frac{1}{2})^\nu$ is negligible.

We now observe that (8.4.16) is no longer proportional to N. On the other hand, the signal has the variance $\sigma_X^2 = 1/3N$, as given in (8.4.8). Hence the SNR is

$$\begin{aligned} \frac{\sigma_X^2}{\sigma_q^2} &= \frac{1}{2N} \cdot 2^{2b} \\ &= 2^{2b-\nu-1} \end{aligned} \tag{8.4.17}$$

Thus, by distributing the scaling of $1/N$ uniformly throughout the FFT algorithm, we have achieved an SNR that is inversely proportional to N instead of N^2.

EXAMPLE 8.4.2

Determine the number of bits required to compute an FFT of 1024 points with an SNR of 30 dB when the scaling is distributed as described above.

Solution. The size of the FFT is $N = 2^{10}$. Hence the SNR according to (8.4.17) is

$$\begin{aligned} 10 \log_{10} 2^{2b-\nu-1} &= 30 \\ 3(2b - 11) &= 30 \\ b &= \frac{21}{2} \text{(11 bits)} \end{aligned}$$

This can be compared with the 15 bits required if all the scaling is performed in the first stage of the FFT algorithm.

8.5 Summary and References

The focus of this chapter was on the efficient computation of the DFT. We demonstrated that by taking advantage of the symmetry and periodicity properties of the exponential factors W_N^{kn}, we can reduce the number of complex multiplications needed to compute the DFT from N^2 to $N \log_2 N$ when N is a power of 2. As we indicated, any sequence can be augmented with zeros, such that $N = 2^\nu$.

For decades, FFT-type algorithms were of interest to mathematicians who were concerned with computing values of Fourier series by hand. However, it was not until Cooley and Tukey (1965) published their well-known paper that the impact and significance of the efficient computation of the DFT was recognized. Since then the Cooley–Tukey FFT algorithm and its various forms, for example, the algorithms of Singleton (1967, 1969), have had a tremendous influence on the use of the DFT in convolution, correlation, and spectrum analysis. For a historical perspective on the FFT algorithm, the reader is referred to the paper by Cooley et al. (1967).

The split-radix FFT (SRFFT) algorithm described in Section 8.1.5 is due to Duhamel and Hollmann (1984, 1986). The "mirror" FFT (MFFT) and "phase" FFT (PFFT) algorithms were described to the authors by R. Price. The exploitation of symmetry properties in the data to reduce the computation time is described in a paper by Swarztrauber (1986).

Over the years, a number of tutorial papers have been published on FFT algorithms. We cite the early papers by Brigham and Morrow (1967), Cochran et al. (1967), Bergland (1969), and Cooley et al. (1967, 1969).

The recognition that the DFT can be arranged and computed as a linear convolution is also highly significant. Goertzel (1968) indicated that the DFT can be computed via linear filtering, although the computational savings of this approach is rather modest, as we have observed. More significant is the work of Bluestein (1970), who demonstrated that the computation of the DFT can be formulated as a chirp linear filtering operation. This work led to the development of the chirp-z transform algorithm by Rabiner et al. (1969).

In addition to the FFT algorithms described in this chapter, there are other efficient algorithms for computing the DFT, some of which further reduce the number of multiplications, but usually require more additions. Of particular importance is an algorithm due to Rader and Brenner (1976), the class of prime factor algorithms, such as the Good algorithm (1971), and the Winograd algorithm (1976, 1978). For a description of these and related algorithms, the reader may refer to the text by Blahut (1985).

Problems

8.1 Show that each of the numbers

$$e^{j(2\pi/N)k}, \qquad 0 \le k \le N-1$$

corresponds to an Nth root of unity. Plot these numbers as phasors in the complex plane and illustrate, by means of this figure, the orthogonality property

$$\sum_{n=0}^{N-1} e^{j(2\pi/N)kn} e^{-j(2\pi/N)ln} = \begin{cases} N, & \text{if } k = l \\ 0, & \text{if } k \neq l \end{cases}$$

8.2 **(a)** Show that the phase factors can be computed recursively by

$$W_N^{ql} = W_N^q W_N^{q(l-1)}$$

(b) Perform this computation once using single-precision floating-point arithmetic and once using only four significant digits. Note the deterioration due to the accumulation of round-off errors in the latter case.

(c) Show how the results in part (b) can be improved by resetting the result to the correct value $-j$, each time $ql = N/4$.

8.3 Let $x(n)$ be a real-valued N-point ($N = 2^\nu$) sequence. Develop a method to compute an N-point DFT $X'(k)$, which contains only the odd harmonics [i.e., $X'(k) = 0$ if k is even] by using only a real $N/2$-point DFT.

8.4 A designer has available a number of eight-point FFT chips. Show explicitly how he should interconnect three such chips in order to compute a 24-point DFT.

8.5 The z-transform of the sequence $x(n) = u(n) - u(n-7)$ is sampled at five points on the unit circle as follows:

$$x(k) = X(z)|_z = e^{j2\pi k/5}, \qquad k = 0, 1, 2, 3, 4$$

Determine the inverse DFT $x'(n)$ of $X(k)$. Compare it with $x(n)$ and explain the results.

8.6 Consider a finite-duration sequence $x(n)$, $0 \leq n \leq 7$, with z-transform $X(z)$. We wish to compute $X(z)$ at the following set of values:

$$z_k = 0.8e^{j[(2\pi k/8)+(\pi/8)]}, \qquad 0 \leq k \leq 7$$

(a) Sketch the points $\{z_k\}$ in the complex plane.

(b) Determine a sequence $s(n)$ such that its DFT provides the desired samples of $X(z)$.

8.7 Derive the radix-2 decimation-in-time FFT algorithm given by (8.1.26) and (8.1.27) as a special case of the more general algorithmic procedure given by (8.1.16) through (8.1.18).

8.8 Compute the eight-point DFT of the sequence

$$x(n) = \begin{cases} 1, & 0 \leq n \leq 7 \\ 0, & \text{otherwise} \end{cases}$$

by using the decimation-in-frequency FFT algorithm described in the text.

8.9 Derive the signal flow graph for the $N = 16$-point, radix-4 decimation-in-time FFT algorithm in which the input sequence is in normal order and the computations are done in place.

8.10 Derive the signal flow graph for the $N = 16$-point, radix-4 decimation-in-frequency FFT algorithm in which the input sequence is in digit-reversed order and the output DFT is in normal order.

8.11 Compute the eight-point DFT of the sequence

$$x(n) = \left\{\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0\right\}$$

using the in-place radix-2 decimation-in-time and radix-2 decimation-in-frequency algorithms. Follow exactly the corresponding signal flow graphs and keep track of all the intermediate quantities by putting them on the diagrams.

8.12 Compute the 16-point DFT of the sequence

$$x(n) = \cos\frac{\pi}{2}n, \qquad 0 \leq n \leq 15$$

using the radix-4 decimation-in-time algorithm.

8.13 Consider the eight-point decimation-in-time (DIT) flow graph in Fig. 8.1.6.

(a) What is the gain of the "signal path" that goes from $x(7)$ to $X(2)$?

(b) How many paths lead from the input to a given output sample? Is this true for every output sample?

(c) Compute $X(3)$ using the operations dictated by this flow graph.

8.14 Draw the flow graph for the decimation-in-frequency (DIF) SRFFT algorithm for $N = 16$. What is the number of nontrivial multiplications?

8.15 Derive the algorithm and draw the $N = 8$ flow graph for the DIT SRFFT algorithm. Compare your flow graph with the DIF radix-2 FFT flow graph shown in Fig. 8.1.11.

8.16 Show that the product of two complex numbers $(a+jb)$ and $(c+jd)$ can be performed with three real multiplications and five additions using the algorithm

$$x_R = (a - b)d + (c - d)a$$

$$x_I = (a - b)d + (c + d)b$$

where

$$x = x_R + jx_I = (a + jb)(c + jd)$$

8.17 Explain how the DFT can be used to compute N equispaced samples of the z-transform of an N-point sequence, on a circle of radius r.

8.18 A real-valued N-point sequence $x(n)$ is called DFT bandlimited if its DFT $X(k) = 0$ for $k_0 \leq k \leq N - k_0$. We insert $(L - 1)N$ zeros in the middle of $X(k)$ to obtain the following LN-point DFT:

$$X'(k) = \begin{cases} X(k), & 0 \leq k \leq k_0 - 1 \\ 0, & k_0 \leq k \leq LN - k_0 \\ X(k + N - LN), & LN - k_0 + 1 \leq k \leq LN - 1 \end{cases}$$

Show that

$$Lx'(Ln) = x(n), \qquad 0 \leq n \leq N - 1$$

where

$$x'(n) \overset{DFT}{\underset{LN}{\longleftrightarrow}} X'(k)$$

Explain the meaning of this type of processing by working out an example with $N = 4$, $L = 1$, and $X(k) = \{1, 0, 0, 1\}$.

8.19 Let $X(k)$ be the N-point DFT of the sequence $x(n)$, $0 \leq n \leq N - 1$. What is the N-point DFT of the sequence $s(n) = X(n)$, $0 \leq n \leq N - 1$?

8.20 Let $X(k)$ be the N-point DFT of the sequence $x(n)$, $0 \leq n \leq N - 1$. We define a $2N$-point sequence $y(n)$ as

$$y(n) = \begin{cases} x\left(\frac{n}{2}\right), & n \text{ even} \\ 0, & n \text{ odd} \end{cases}$$

Express the $2N$-point DFT of $y(n)$ in terms of $X(k)$.

8.21 **(a)** Determine the z-transform $W(z)$ of the Hanning window $w(n) = \left(1 - \cos \frac{2\pi n}{N-1}\right)/2$.

(b) Determine a formula to compute the N-point DFT $X_w(k)$ of the signal $x_w(n) = w(n)x(n)$, $0 \leq n \leq N - 1$, from the N-point DFT $X(k)$ of the signal $x(n)$.

8.22 Create a DFT coefficient table that uses only $N/4$ memory locations to store the first quadrant of the sine sequence (assume N even).

8.23 Determine the computational burden of the algorithm given by (8.2.12) and compare it with the computational burden required in the $2N$-point DFT of $g(n)$. Assume that the FFT algorithm is a radix-2 algorithm.

8.24 Consider an IIR system described by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

Describe a procedure that computes the frequency response $H\left(\frac{2\pi}{N}k\right), k = 0, 1, \ldots, N-1$ using the FFT algorithm ($N = 2^\nu$).

8.25 Develop a radix-3 decimation-in-time FFT algorithm for $N = 3^\nu$ and draw the corresponding flow graph for $N = 9$. What is the number of required complex multiplications? Can the operations be performed in place?

8.26 Repeat Problem 8.25 for the DIF case.

8.27 *FFT input and output pruning* In many applications we wish to compute only a few points M of the N-point DFT of a finite-duration sequence of length L (i.e., $M << N$ and $L << N$).

(a) Draw the flow graph of the radix-2 DIF FFT algorithm for $N = 16$ and eliminate [i.e., prune] all signal paths that originate from zero inputs assuming that only $x(0)$ and $x(1)$ are nonzero.

(b) Repeat part (a) for the radix-2 DIT algorithm.

(c) Which algorithm is better if we wish to compute all points of the DFT? What happens if we want to compute only the points $X(0)$, $X(1)$, $X(2)$, and $X(3)$? Establish a rule to choose between DIT and DIF pruning depending on the values of M and L.

(d) Give an estimate of saving in computations in terms of M, L, and N.

8.28 *Parallel computation of the DFT* Suppose that we wish to compute an $N = 2^p 2^\nu$-point DFT using 2^p digital signal processors (DSPs). For simplicity we assume that $p = \nu = 2$. In this case each DSP carries out all the computations that are necessary to compute 2^ν DFT points.

(a) Using the radix-2 DIF flow graph, show that to avoid data shuffling, the entire sequence $x(n)$ should be loaded to the memory of each DSP.

(b) Identify and redraw the portion of the flow graph that is executed by the DSP that computes the DFT samples $X(2)$, $X(10)$, $X(6)$, and $X(14)$.

(c) Show that, if we use $M = 2^p$ DSPs, the computation speed-up S is given by

$$S = M \frac{\log_2 N}{\log_2 N - \log_2 M + 2(M-1)}$$

8.29 Develop an inverse radix-2 DIT FFT algorithm starting with the definition. Draw the flow graph for computation and compare with the corresponding flow graph for the direct FFT. Can the IFFT flow graph be obtained from the one for the direct FFT?

8.30 Repeat Problem 8.29 for the DIF case.

8.31 Show that an FFT on data with Hermitian symmetry can be derived by reversing the flow graph of an FFT for real data.

8.32 Determine the system function $H(z)$ and the difference equation for the system that uses the Goertzel algorithm to compute the DFT value $X(N-k)$.

8.33 **(a)** Suppose that $x(n)$ is a finite-duration sequence of $N = 1024$ points. It is desired to evaluate the z-transform $X(z)$ of the sequence at the points

$$z_k = e^{j(2\pi/1024)k}, \qquad k = 0, 100, 200, \ldots, 1000$$

by using the most efficient method or algorithm possible. Describe an algorithm for performing this computation efficiently. Explain how you arrived at your answer by giving the various options or algorithms that can be used.

(b) Repeat part (a) if $X(z)$ is to be evaluated at

$$z_k = 2(0.9)^k e^{j[(2\pi/5000)k+\pi/2]}, \qquad k = 0, 1, 2, \ldots, 999$$

8.34 Repeat the analysis for the variance of the quantization error, carried out in Section 8.4.2, for the decimation-in-frequency radix-2 FFT algorithm.

8.35 The basic butterfly in the radix-2 decimation-in-time FFT algorithm is

$$X_{n+1}(k) = X_n(k) + W_N^m X_n(l)$$
$$X_{n+1}(l) = X_n(k) - W_N^m X_n(l)$$

(a) If we require that $|X_n(k)| < \frac{1}{2}$ and $|X_n(l)| < \frac{1}{2}$, show that

$$|\text{Re}[X_{[}X_{n+1}(k)]| < 1, \qquad |\text{Re}[X_{n+1}(l)]| < 1$$
$$|\text{Im}[X_{[}X_{n+1}(k)]| < 1, \qquad |\text{Im}[X_{n+1}(l)]| < 1$$

Thus overflow does not occur.

(b) Prove that

$$\max[|X_{n+1}(k)|, |X_{n+1}(l)|] \geq \max[|X_n(k)|, |X_n(l)|]$$
$$\max[|X_{n+1}(k)|, |X_{n+1}(l)|] \leq 2\max[|X_n(k)|, |X_n(l)|]$$

8.36 *Computation of the DFT* Use an FFT subroutine to compute the following DFTs and plot the magnitudes $|X(k)|$ of the DFTs.

(a) The 64-point DFT of the sequence

$$x(n) = \begin{cases} 1, & n = 0, 1, \ldots, 15 \quad (N_1 = 16) \\ 0, & \text{otherwise} \end{cases}$$

(b) The 64-point DFT of the sequence

$$x(n) = \begin{cases} 1, & n = 0, 1, \ldots, 7 \quad (N_1 = 8) \\ 0, & \text{otherwise} \end{cases}$$

(c) The 128-point DFT of the sequence in part (a).

(d) The 64-point DFT of the sequence

$$x(n) = \begin{cases} 10e^{j(\pi/8)n}, & n = 0, 1, \ldots, 63 \quad (N_1 = 64) \\ 0, & \text{otherwise} \end{cases}$$

Answer the following questions.

1. What is the frequency interval between successive samples for the plots in parts (a), (b), (c), and (d)?
2. What is the value of the spectrum at zero frequency (dc value) obtained from the plots in parts (a), (b), (c), (d)?
   From the formula

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j(2\pi/N)nk}$$

   compute the theoretical values for the dc value and check these with the computer results.
3. In plots (a), (b), and (c), what is the *frequency interval* between successive nulls in the spectrum? What is the relationship between N_1 of the sequence $x(n)$ and the frequency interval between successive nulls?
4. Explain the difference between the plots obtained from parts (a) and (c).

8.37 *Identification of pole positions in a system* Consider the system described by the difference equation

$$y(n) = -r^2 y(n-2) + x(n)$$

(a) Let $r = 0.9$ and $x(n) = \delta(n)$. Generate the output sequence $y(n)$ for $0 \leq n \leq 127$. Compute the $N = 128$-point DFT $\{Y(k)\}$ and plot $\{|Y(k)|\}$.

(b) Compute the $N = 128$-point DFT of the sequence

$$w(n) = (0.92)^{-n} y(n)$$

where $y(n)$ is the sequence generated in part (a). Plot the DFT values $|W(k)|$. What can you conclude from the plots in parts (a) and (b)?

(c) Let $r = 0.5$ and repeat part (a).

(d) Repeat part (b) for the sequence

$$w(n) = (0.55)^{-n} y(n)$$

where $y(n)$ is the sequence generated in part (c). What can you conclude from the plots in parts (c) and (d)?

(e) Now let the sequence generated in part (c) be corrupted by a sequence of "measurement" noise which is Gaussian with zero mean and variance $\sigma^2 = 0.1$. Repeat parts (c) and (d) for the noise-corrupted signal.

CHAPTER

9

Implementation of Discrete-Time Systems

The focus of this chapter is on the realization of linear time-invariant discrete-time systems either in software or hardware. As we noted in Chapter 2, there are various configurations or structures for the realization of any FIR and IIR discrete-time system. In Chapter 2 we described the simplest of these structures, namely, the direct-form realizations. However, there are other more practical structures that offer some distinct advantages, especially when quantization effects are taken into consideration.

Of particular importance are the cascade, parallel, and lattice structures, which exhibit robustness in finite-word-length implementations. Also described in this chapter is the frequency-sampling realization for an FIR system, which often has the advantage of being computationally efficient when compared with alternative FIR realizations. Other filter structures are obtained by employing a state-space formulation for linear time-invariant systems. Due to space limitations, state-space structures are not covered.

In addition to describing the various structures for the realization of discrete-time systems, we also treat problems associated with quantization effects in the implementation of digital filters using finite-precision arithmetic. This treatment includes the effects on the filter frequency response characteristics resulting from coefficient quantization and the round-off noise effects inherent in the digital implementation of discrete-time systems.

9.1 Structures for the Realization of Discrete-Time Systems

Let us consider the important class of linear time-invariant discrete-time systems characterized by the general linear constant-coefficient difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{9.1.1}$$

As we have shown by means of the z-transform, linear time-invariant discrete-time systems of this class are also characterized by the rational system function

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{9.1.2}$$

which is a ratio of two polynomials in z^{-1}. From the latter characterization, we obtain the zeros and poles of the system function, which depend on the choice of the system parameters $\{b_k\}$ and $\{a_k\}$ and which determine the frequency response characteristics of the system.

Our focus in this chapter is on the various methods of implementing (9.1.1) or (9.1.2) either in hardware, or in software on a programmable digital computer. We shall show that (9.1.1) or (9.1.2) can be implemented in a variety of ways depending on the form in which these two characterizations are arranged.

In general, we can view (9.1.1) as a computational procedure (an algorithm) for determining the output sequence $y(n)$ of the system from the input sequence $x(n)$. However, in various ways, the computations in (9.1.1) can be arranged into equivalent sets of difference equations. Each set of equations defines a computational procedure or an algorithm for implementing the system. From each set of equations we can construct a block diagram consisting of an interconnection of delay elements, multipliers, and adders. In Section 2.5 we referred to such a block diagram as a *realization* of the system or, equivalently, as a *structure* for realizing the system.

If the system is to be implemented in software, the block diagram or, equivalently, the set of equations that are obtained by rearranging (9.1.1), can be converted into a program that runs on a digital computer. Alternatively, the structure in block diagram form implies a hardware configuration for implementing the system.

Perhaps, the one issue that may not be clear to the reader at this point is why we are considering any rearrangements of (9.1.1) or (9.1.2). Why not just implement (9.1.1) or (9.1.2) directly without any rearrangement? If either (9.1.1) or (9.1.2) is rearranged in some manner, what are the benefits gained in the corresponding implementation?

These are the important questions which are answered in this chapter. At this point in our development, we simply state that the major factors that influence our choice of a specific realization are computational complexity, memory requirements, and finite-word-length effects in the computations.

Computational complexity refers to the number of arithmetic operations (multiplications, divisions, and additions) required to compute an output value $y(n)$ for the system. In the past, these were the only items used to measure computational complexity. However, with recent developments in the design and fabrication of rather

sophisticated programmable digital signal processing chips, other factors, such as the number of times a fetch from memory is performed or the number of times a comparison between two numbers is performed per output sample, have become important in assessing the computational complexity of a given realization of a system.

Memory requirements refers to the number of memory locations required to store the system parameters, past inputs, past outputs, and any intermediate computed values.

Finite-word-length effects or finite-precision effects refer to the quantization effects that are inherent in any digital implementation of the system, either in hardware or in software. The parameters of the system must necessarily be represented with finite precision. The computations that are performed in the process of computing an output from the system must be rounded off or truncated to fit within the limited precision constraints of the computer or the hardware used in the implementation. Whether the computations are performed in fixed-point or floating-point arithmetic is another consideration. All these problems are usually called finite-word-length effects and are extremely important in influencing our choice of a system realization. We shall see that different structures of a system, which are equivalent for infinite precision, exhibit different behavior when finite-precision arithmetic is used in the implementation. Therefore, it is very important in practice to select a realization that is not very sensitive to finite-word-length effects.

Although these three factors are the major ones in influencing our choice of the realization of a system of the type described by either (9.1.1) or (9.1.2), other factors, such as whether the structure or the realization lends itself to parallel processing, or whether the computations can be pipelined, may play a role in our selection of the specific implementation. These additional factors are usually important in the realization of more complex digital signal processing algorithms.

In our discussion of alternative realizations, we concentrate on the three major factors just outlined. Occasionally, we will include some additional factors that may be important in some implementations.

9.2 Structures for FIR Systems

In general, an FIR system is described by the difference equation

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \tag{9.2.1}$$

or, equivalently, by the system function

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k} \tag{9.2.2}$$

Furthermore, the unit sample response of the FIR system is identical to the coefficients $\{b_k\}$, that is,

$$h(n) = \begin{cases} b_n, & 0 \le n \le M-1 \\ 0, & \text{otherwise} \end{cases} \tag{9.2.3}$$

The length of the FIR filter is selected as M to conform with the established notation in the technical literature. The reader should note this change in notation in the treatment of FIR filters in this and subsequent chapters.

We shall present several methods for implementing an FIR system, beginning with the simplest structure, called the direct form. A second structure is the cascade-form realization. The third structure that we shall describe is the frequency-sampling realization. Finally, we present a lattice realization of an FIR system. In this discussion we follow the convention often used in the technical literature, which is to use $\{h(n)\}$ for the parameters of an FIR system.

In addition to the four realizations indicated above, an FIR system can be realized by means of the DFT, as described in Section 8.2. From one point of view, the DFT can be considered as a computational procedure rather than a structure for an FIR system. However, when the computational procedure is implemented in hardware, there is a corresponding structure for the FIR system. In practice, hardware implementations of the DFT are based on the use of the fast Fourier transform (FFT) algorithms described in Chapter 8.

9.2.1 Direct-Form Structure

The direct-form realization follows immediately from the nonrecursive difference equation given by (9.2.1) or, equivalently, by the convolution summation

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{9.2.4}$$

The structure is illustrated in Fig. 9.2.1.

We observe that this structure requires $M-1$ memory locations for storing the $M-1$ previous inputs, and has a complexity of M multiplications and $M-1$ additions per output point. Since the output consists of a weighted linear combination of $M-1$ past values of the input and the weighted current value of the input, the structure in Fig. 9.2.1 resembles a tapped delay line or a transversal system. Consequently, the direct-form realization is often called a transversal or tapped-delay-line filter.

When the FIR system has linear phase, as described in Section 10.2, the unit sample response of the system satisfies either the symmetry or asymmetry condition

$$h(n) = \pm h(M-1-n) \tag{9.2.5}$$

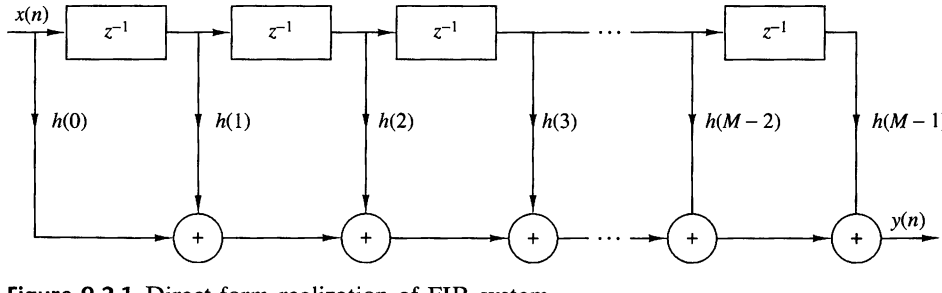


Figure 9.2.1 Direct-form realization of FIR system.

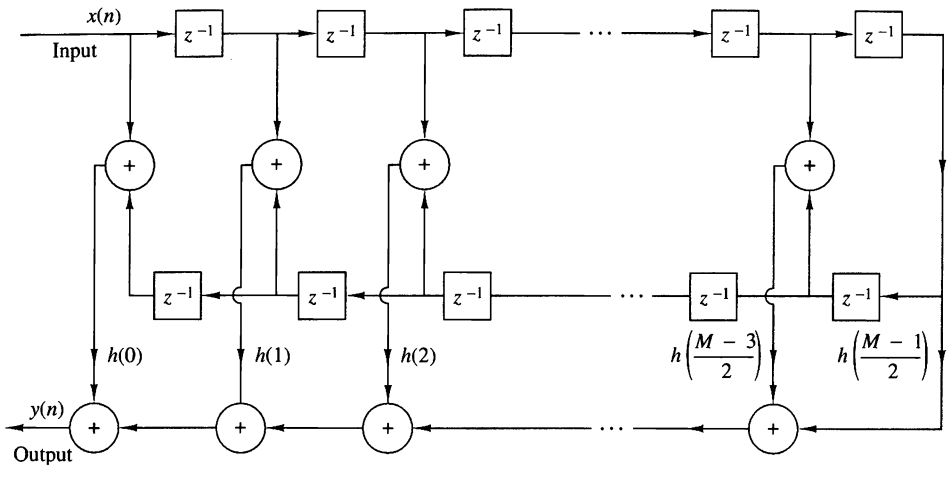


Figure 9.2.2 Direct-form realization of linear-phase FIR system (M odd).

For such a system the number of multiplications is reduced from M to $M/2$ for M even and to $(M-1)/2$ for M odd. For example, the structure that takes advantage of this symmetry is illustrated in Fig. 9.2.2 for the case in which M is odd.

9.2.2 Cascade-Form Structures

The cascade realization follows naturally from the system function given by (9.2.2). It is a simple matter to factor $H(z)$ into second-order FIR systems so that

$$H(z) = \prod_{k=1}^{K} H_k(z) \tag{9.2.6}$$

where

$$H_k(z) = b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}, \qquad k = 1, 2, \ldots, K \tag{9.2.7}$$

and K is the integer part of $(M+1)/2$. The filter parameter b_0 may be equally distributed among the K filter sections, such that $b_0 = b_{10}b_{20}\cdots b_{K0}$ or it may be assigned to a single filter section. The zeros of $H(z)$ are grouped in pairs to produce the second-order FIR systems of the form (9.2.7). It is always desirable to form pairs of complex-conjugate roots so that the coefficients $\{b_{ki}\}$ in (9.2.7) are real valued. On the other hand, real-valued roots can be paired in any arbitrary manner. The cascade-form realization along with the basic second-order section are shown in Fig. 9.2.3.

In the case of linear-phase FIR filters, the symmetry in $h(n)$ implies that the zeros of $H(z)$ also exhibit a form of symmetry. In particular, if z_k and z_k^* are a pair of complex-conjugate zeros then $1/z_k$ and $1/z_k^*$ are also a pair of complex-conjugate zeros (see Sec. 10.2). Consequently, we gain some simplification by forming fourth-

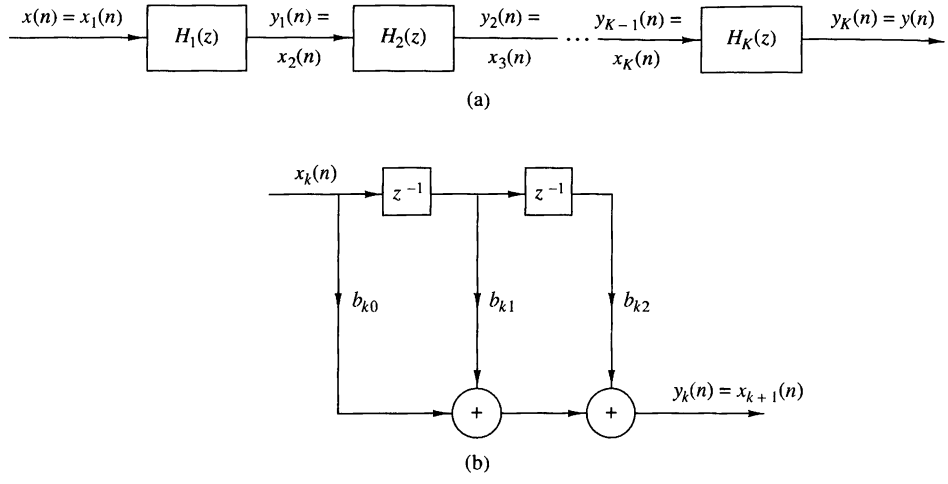


Figure 9.2.3 Cascade realization of an FIR system.

order sections of the FIR system as follows:

$$\begin{aligned} H_k(z) &= c_{k0}(1 - z_k z^{-1})(1 - z_k^* z^{-1})(1 - z^{-1}/z_k)(1 - z^{-1}/z_k^*) \\ &= c_{k0} + c_{k1} z^{-1} + c_{k2} z^{-2} + c_{k1} z^{-3} + c_{k0} z^{-4} \end{aligned} \tag{9.2.8}$$

where the coefficients $\{c_{k1}\}$ and $\{c_{k2}\}$ are functions of z_k. Thus, by combining the two pairs of poles to form a fourth-order filter section, we have reduced the number of multiplications from six to three (i.e., by a factor of 50%). Figure 9.2.4 illustrates the basic fourth-order FIR filter structure.

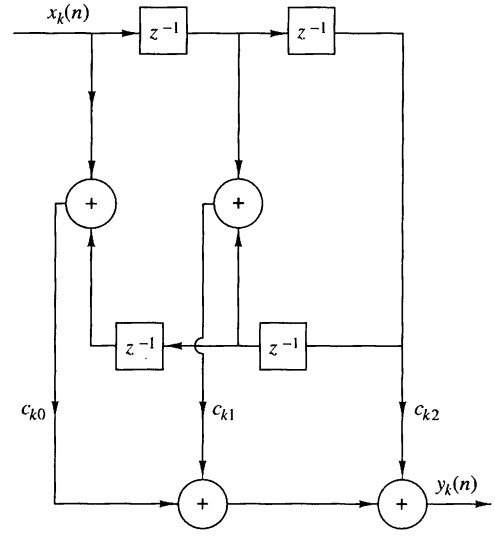


Figure 9.2.4
Fourth-order section in a cascade realization of an FIR system.

9.2.3 Frequency-Sampling Structures[1]

The frequency-sampling realization is an alternative structure for an FIR filter in which the parameters that characterize the filter are the values of the desired frequency response instead of the impulse response $h(n)$. To derive the frequency-sampling structure, we specify the desired frequency response at a set of equally spaced frequencies, namely

$$\omega_k = \frac{2\pi}{M}(k+\alpha), \qquad k = 0, 1, \ldots, \frac{M-1}{2}, \quad M \text{ odd}$$

$$k = 0, 1, \ldots, \frac{M}{2} - 1, \quad M \text{ even}$$

$$\alpha = 0 \text{ or } \tfrac{1}{2}$$

and solve for the unit sample response $h(n)$ from these equally spaced frequency specifications. Thus we can write the frequency response as

$$H(\omega) = \sum_{n=0}^{M-1} h(n)e^{-j\omega n}$$

and the values of $H(\omega)$ at frequencies $\omega_k = (2\pi/M)(k+\alpha)$ are simply

$$\begin{aligned} H(k+\alpha) &= H\left(\frac{2\pi}{M}(k+\alpha)\right) \\ &= \sum_{n=0}^{M-1} h(n)e^{-j2\pi(k+\alpha)n/M}, \qquad k = 0, 1, \ldots, M-1 \end{aligned} \tag{9.2.9}$$

The set of values $\{H(k+\alpha)\}$ are called the frequency samples of $H(\omega)$. In the case where $\alpha = 0$, $\{H(k)\}$ corresponds to the M-point DFT of $\{h(n)\}$.

It is a simple matter to invert (9.2.9) and express $h(n)$ in terms of the frequency samples. The result is

$$h(n) = \frac{1}{M}\sum_{k=0}^{M-1} H(k+\alpha)e^{j2\pi(k+\alpha)n/M}, \qquad n = 0, 1, \ldots, M-1 \tag{9.2.10}$$

When $\alpha = 0$, (9.2.10) is simply the IDFT of $\{H(k)\}$. Now if we use (9.2.10) to substitute for $h(n)$ in the z-transform $H(z)$, we have

$$\begin{aligned} H(z) &= \sum_{n=0}^{M-1} h(n)z^{-n} \\ &= \sum_{n=0}^{M-1}\left[\frac{1}{M}\sum_{k=0}^{M-1} H(k+\alpha)e^{j2\pi(k+\alpha)n/M}\right]z^{-n} \end{aligned} \tag{9.2.11}$$

[1] The reader may also refer to Section 10.2.3 for additional discussion of frequency-sampling FIR filters.

By interchanging the order of the two summations in (9.2.11) and performing the summation over the index n we obtain

$$\begin{aligned} H(z) &= \sum_{k=0}^{M-1} H(k+\alpha)\left[\frac{1}{M}\sum_{n=0}^{M-1}(e^{j2\pi(k+\alpha)/M}z^{-1})^n\right] \\ &= \frac{1-z^{-M}e^{j2\pi\alpha}}{M}\sum_{k=0}^{M-1}\frac{H(k+\alpha)}{1-e^{j2\pi(k+\alpha)/M}z^{-1}} \end{aligned} \tag{9.2.12}$$

Thus the system function $H(z)$ is characterized by the set of frequency samples $\{H(k+\alpha)\}$ instead of $\{h(n)\}$.

We view this FIR filter realization as a cascade of two filters [i.e., $H(z) = H_1(z)H_2(z)$]. One is an all-zero filter, or a comb filter, with system function

$$H_1(z) = \frac{1}{M}(1 - z^{-M}e^{j2\pi\alpha}) \tag{9.2.13}$$

Its zeros are located at equally spaced points on the unit circle at

$$z_k = e^{j2\pi(k+\alpha)/M}, \qquad k = 0, 1, \ldots, M-1$$

The second filter with system function

$$H_2(z) = \sum_{k=0}^{M-1}\frac{H(k+\alpha)}{1-e^{j2\pi(k+\alpha)/M}z^{-1}} \tag{9.2.14}$$

consists of a parallel bank of single-pole filters with resonant frequencies

$$p_k = e^{j2\pi(k+\alpha)/M}, \qquad k = 0, 1, \ldots, M-1$$

Note that the pole locations are identical to the zero locations and that both occur at $\omega_k = 2\pi(k+\alpha)/M$, which are the frequencies at which the desired frequency response is specified. The gains of the parallel bank of resonant filters are simply the complex-valued parameters $\{H(k+\alpha)\}$. This cascade realization is illustrated in Fig. 9.2.5.

When the desired frequency response characteristic of the FIR filter is narrowband, most of the gain parameters $\{H(k+\alpha)\}$ are zero. Consequently, the corresponding resonant filters can be eliminated and only the filters with nonzero gains need be retained. The net result is a filter that requires fewer computations (mul-

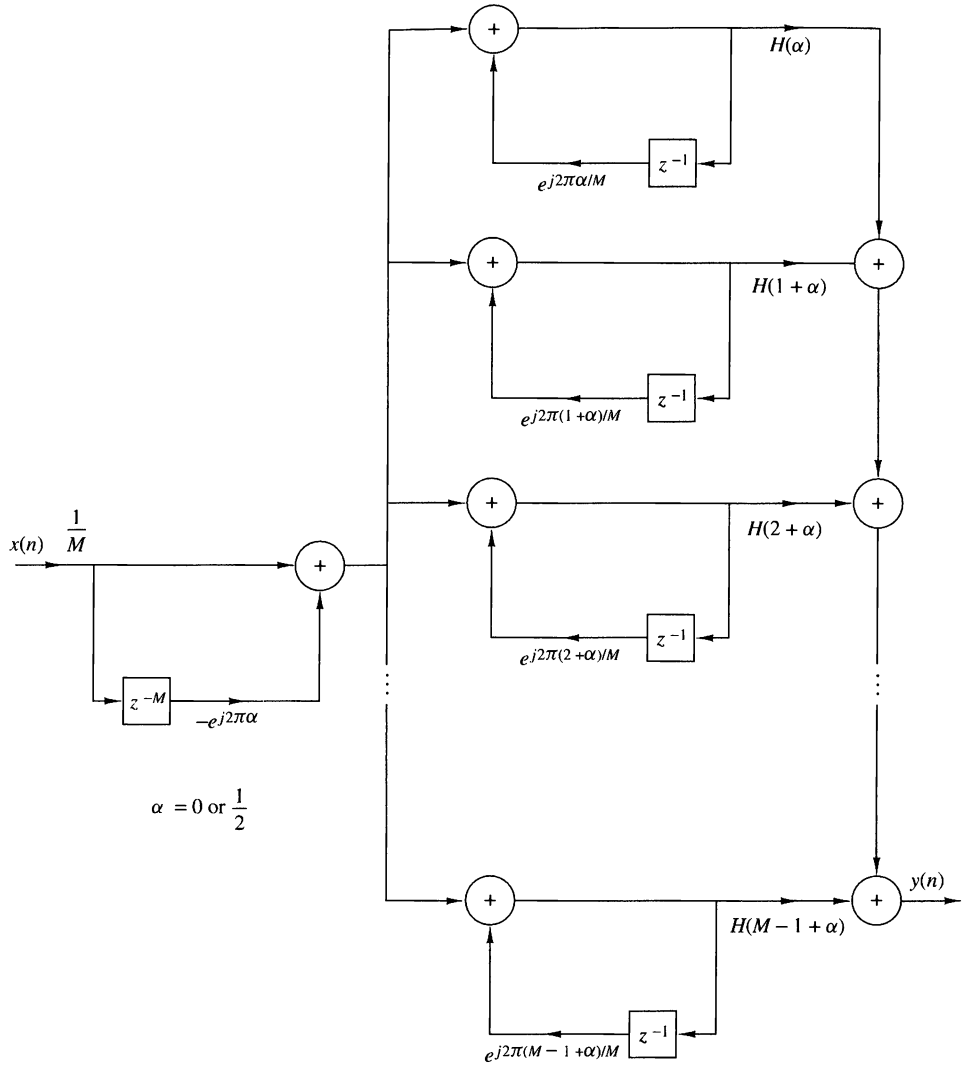


Figure 9.2.5 Frequency-sampling realization of FIR filter.

tiplications and additions) than the corresponding direct-form realization. Thus we obtain a more efficient realization.

The frequency-sampling filter structure can be simplified further by exploiting the symmetry in $H(k+\alpha)$, namely, $H(k) = H^*(M-k)$ for $\alpha = 0$ and

$$H\left(k+\tfrac{1}{2}\right) = H\left(M-k-\tfrac{1}{2}\right), \qquad \text{for } \alpha = \tfrac{1}{2}$$

These relations are easily deduced from (9.2.9). As a result of this symmetry, a pair of single-pole filters can be combined to form a single two-pole filter with real-valued

parameters. Thus for $\alpha = 0$ the system function $H_2(z)$ reduces to

$$H_2(z) = \frac{H(0)}{1 - z^{-1}} + \sum_{k=1}^{(M-1)/2} \frac{A(k) + B(k)z^{-1}}{1 - 2\cos(2\pi k/M)z^{-1} + z^{-2}}, \qquad M \text{ odd}$$

$$H_2(z) = \frac{H(0)}{1 - z^{-1}} + \frac{H(M/2)}{1 + z^{-1}} + \sum_{k=1}^{(M/2)-1} \frac{A(k) + B(k)z^{-1}}{1 - 2\cos(2\pi k/M)z^{-1} + z^{-2}}, \qquad M \text{ even} \tag{9.2.15}$$

where, by definition,

$$\begin{aligned} A(k) &= H(k) + H(M - k) \\ B(k) &= H(k)e^{-j2\pi k/M} + H(M - k)e^{j2\pi k/M} \end{aligned} \tag{9.2.16}$$

Similar expressions can be obtained for $\alpha = \frac{1}{2}$.

EXAMPLE 9.2.1

Sketch the block diagram for the direct-form realization and the frequency-sampling realization of the $M = 32$, $\alpha = 0$, linear-phase (symmetric) FIR filter which has frequency samples

$$H\left(\frac{2\pi k}{32}\right) = \begin{cases} 1, & k = 0, 1, 2 \\ \frac{1}{2}, & k = 3 \\ 0, & k = 4, 5, \ldots, 15 \end{cases}$$

Compare the computational complexity of these two structures.

Solution. Since the filter is symmetric, we exploit this symmetry and thus reduce the number of multiplications per output point by a factor of 2, from 32 to 16 in the direct-form realization. The number of additions per output point is 31. The block diagram of the direct realization is illustrated in Fig. 9.2.6.

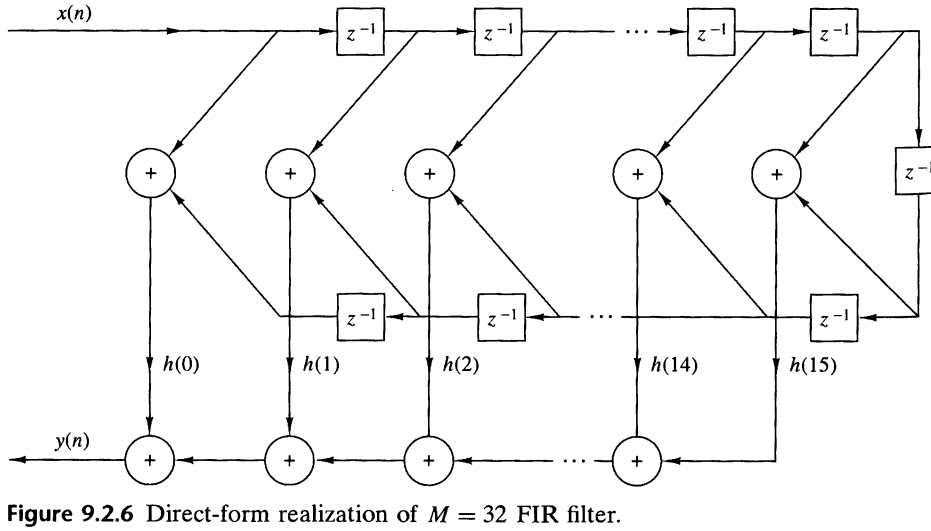


Figure 9.2.6 Direct-form realization of $M = 32$ FIR filter.

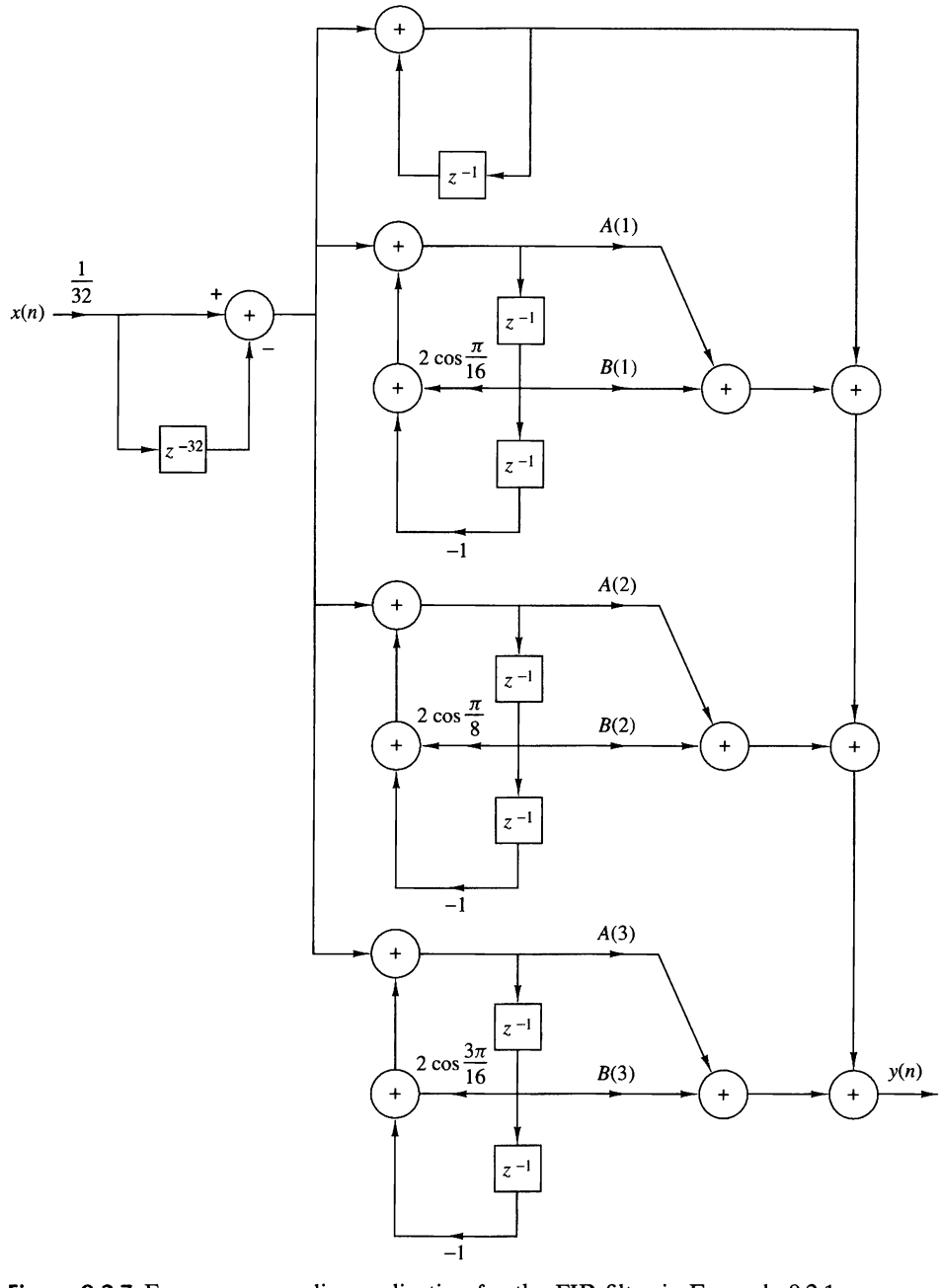


Figure 9.2.7 Frequency-sampling realization for the FIR filter in Example 9.2.1.

We use the form in (9.2.13) and (9.2.15) for the frequency-sampling realization and drop all terms that have zero-gain coefficients {H(k)}. The nonzero coefficients are $H(k)$ and the corresponding pairs are $H(M-k)$, for $k = 0, 1, 2, 3$. The block diagram of the resulting realization is shown in Fig. 9.2.7. Since $H(0) = 1$, the single-pole filter requires no multipli-

cation. The three double-pole filter sections require three multiplications each for a total of nine multiplications. The total number of additions is 13. Therefore, the frequency-sampling realization of this FIR filter is computationally more efficient than the direct-form realization.

9.2.4 Lattice Structure

In this section we introduce another FIR filter structure, called the lattice filter or lattice realization. Lattice filters are used extensively in digital speech processing and in the implementation of adaptive filters.

Let us begin the development by considering a sequence of FIR filters with system functions

$$H_m(z) = A_m(z), \qquad m = 0, 1, 2, \ldots, M-1 \tag{9.2.17}$$

where, by definition, $A_m(z)$ is the polynomial

$$A_m(z) = 1 + \sum_{k=1}^{m} \alpha_m(k) z^{-k}, \qquad m \geq 1 \tag{9.2.18}$$

and $A_0(z) = 1$. The unit sample response of the mth filter is $h_m(0) = 1$ and $h_m(k) = \alpha_m(k)$, $k = 1, 2, \ldots, m$. The subscript m on the polynomial $A_m(z)$ denotes the degree of the polynomial. For mathematical convenience, we define $\alpha_m(0) = 1$.

If $\{x(n)\}$ is the input sequence to the filter $A_m(z)$ and $\{y(n)\}$ is the output sequence, we have

$$y(n) = x(n) + \sum_{k=1}^{m} \alpha_m(k) x(n-k) \tag{9.2.19}$$

Two direct-form structures of the FIR filter are illustrated in Fig. 9.2.8.

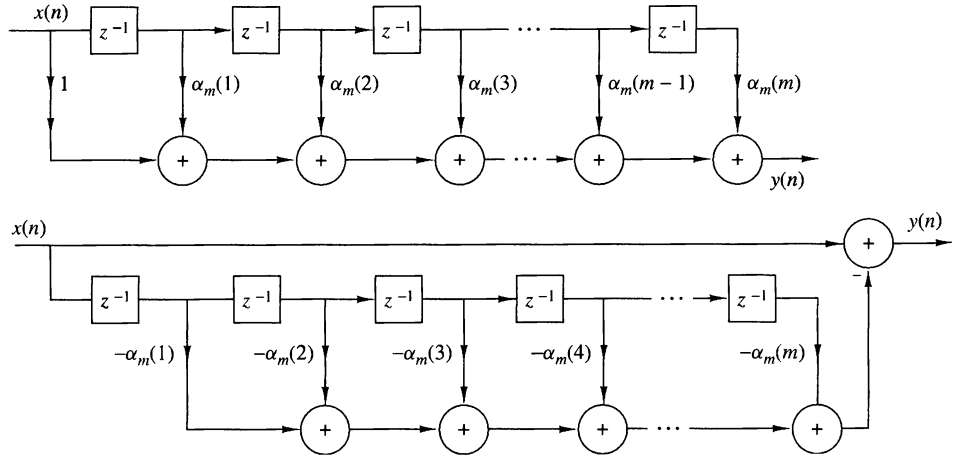


Figure 9.2.8 Direct-form realization of the FIR prediction filter.

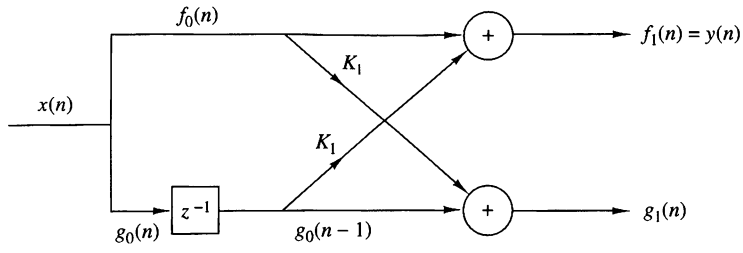


Figure 9.2.9 Single-stage lattice filter.

In Chapter 12, we show that the FIR structures shown in Fig. 9.2.8 are intimately related with the topic of linear prediction, where

$$\hat{x}(n) = -\sum_{k=1}^{m} \alpha_m(k) x(n-k) \tag{9.2.20}$$

is the one-step forward predicted value of $x(n)$, based on m past inputs, and $y(n) = x(n) - \hat{x}(n)$, given by (9.2.19), represents the prediction error sequence. In this context, the top filter structure in Fig. 9.2.8 is called a *prediction error filter.*

Now suppose that we have a filter of order $m = 1$. The output of such a filter is

$$y(n) = x(n) + \alpha_1(1) x(n-1) \tag{9.2.21}$$

This output can also be obtained from a first-order or single-stage lattice filter, illustrated in Fig. 9.2.9, by exciting both of the inputs by $x(n)$ and selecting the output from the top branch. Thus the output is exactly (9.2.21), if we select $K_1 = \alpha_1(1)$. The parameter K_1 in the lattice is called a reflection coefficient.

Next, let us consider an FIR filter for which $m = 2$. In this case the output from a direct-form structure is

$$y(n) = x(n) + \alpha_2(1) x(n-1) + \alpha_2(2) x(n-2) \tag{9.2.22}$$

By cascading two lattice stages as shown in Fig. 9.2.10, it is possible to obtain the same output as (9.2.22). Indeed, the output from the first stage is

$$\begin{aligned} f_1(n) &= x(n) + K_1 x(n-1) \\ g_1(n) &= K_1 x(n) + x(n-1) \end{aligned} \tag{9.2.23}$$

The output from the second stage is

$$\begin{aligned} f_2(n) &= f_1(n) + K_2 g_1(n-1) \\ g_2(n) &= K_2 f_1(n) + g_1(n-1) \end{aligned} \tag{9.2.24}$$

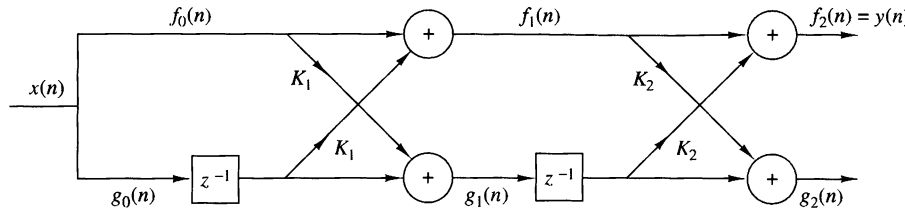


Figure 9.2.10 Two-stage lattice filter.

If we focus our attention on $f_2(n)$ and substitute for $f_1(n)$ and $g_1(n-1)$ from (9.2.23) into (9.2.24), we obtain

$$\begin{aligned} f_2(n) &= x(n) + K_1 x(n-1) + K_2[K_1 x(n-1) + x(n-2)] \\ &= x(n) + K_1(1 + K_2)x(n-1) + K_2 x(n-2) \end{aligned} \tag{9.2.25}$$

Now (9.2.25) is identical to the output of the direct-form FIR filter as given by (9.2.22), if we equate the coefficients, that is,

$$\alpha_2(2) = K_2, \qquad \alpha_2(1) = K_1(1 + K_2) \tag{9.2.26}$$

or, equivalently,

$$K_2 = \alpha_2(2), \qquad K_1 = \frac{\alpha_2(1)}{1 + \alpha_2(2)} \tag{9.2.27}$$

Thus the reflection coefficients K_1 and K_2 of the lattice filter can be obtained from the coefficients $\{\alpha_m(k)\}$ of the direct-form realization.

By continuing this process, one can easily demonstrate, by induction, the equivalence between an mth-order direct-form FIR filter and an m-order or m-stage lattice filter. The lattice filter is generally described by the following set of order-recursive equations:

$$f_0(n) = g_0(n) = x(n) \tag{9.2.28}$$

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.29}$$

$$g_m(n) = K_m f_{m-1}(n) + g_{m-1}(n-1), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.30}$$

Then the output of the $(M-1)$-stage filter corresponds to the output of an $(M-1)$-order FIR filter, that is,

$$y(n) = f_{M-1}(n)$$

Figure 9.2.11 illustrates an $(M-1)$-stage lattice filter in block diagram form along with a typical stage that shows the computations specified by (9.2.29) and (9.2.30).

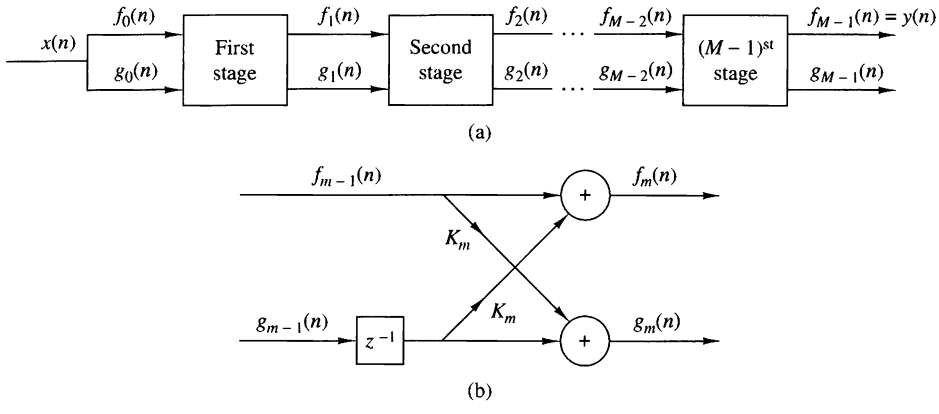


Figure 9.2.11 $(M-1)$-stage lattice filter.

As a consequence of the equivalence between an FIR filter and a lattice filter, the output $f_m(n)$ of an m-stage lattice filter can be expressed as

$$f_m(n) = \sum_{k=0}^{m} \alpha_m(k)x(n-k), \qquad \alpha_m(0) = 1 \tag{9.2.31}$$

Since (9.2.31) is a convolution sum, it follows that the z-transform relationship is

$$F_m(z) = A_m(z)X(z)$$

or, equivalently,

$$A_m(z) = \frac{F_m(z)}{X(z)} = \frac{F_m(z)}{F_0(z)} \tag{9.2.32}$$

The other output component from the lattice, namely, $g_m(n)$, can also be expressed in the form of a convolution sum as in (9.2.31), by using another set of coefficients, say $\{\beta_m(k)\}$. That this in fact is the case becomes apparent from observation of (9.2.23) and (9.2.24). From (9.2.23) we note that the filter coefficients for the lattice filter that produces $f_1(n)$ are $\{1, K_1\} = \{1, \alpha_1(1)\}$ while the coefficients for the filter with output $g_1(n)$ are $\{K_1, 1\} = \{\alpha_1(1), 1\}$. We note that these two sets of coefficients are in reverse order. If we consider the two-stage lattice filter, with the output given by (9.2.24), we find that $g_2(n)$ can be expressed in the form

$$\begin{aligned} g_2(n) &= K_2 f_1(n) + g_1(n-1) \\ &= K_2[x(n) + K_1 x(n-1)] + K_1 x(n-1) + x(n-2) \\ &= K_2 x(n) + K_1(1 + K_2)x(n-1) + x(n-2) \\ &= \alpha_2(2)x(n) + \alpha_2(1)x(n-1) + x(n-2) \end{aligned}$$

Consequently, the filter coefficients are $\{\alpha_2(2), \alpha_2(1), 1\}$, whereas the coefficients for the filter that produces the output $f_2(n)$ are $\{1, \alpha_2(1), \alpha_2(2)\}$. Here, again, the two sets of filter coefficients are in reverse order.

From this development it follows that the output $g_m(n)$ from an m-stage lattice filter can be expressed by the convolution sum of the form

$$g_m(n) = \sum_{k=0}^{m} \beta_m(k)x(n-k) \tag{9.2.33}$$

where the filter coefficients $\{\beta_m(k)\}$ are associated with a filter that produces $f_m(n) = y(n)$ but operates in reverse order. Consequently,

$$\beta_m(k) = \alpha_m(m-k), \qquad k = 0, 1, \ldots, m \tag{9.2.34}$$

with $\beta_m(m) = 1$.

In the context of linear prediction, suppose that the data $x(n)$, $x(n-1)$, ..., $x(n-m+1)$ is used to linearly predict the signal value $x(n-m)$ by use of a linear filter with coefficients $\{-\beta_m(k)\}$. Thus the predicted value is

$$\hat{x}(n-m) = -\sum_{k=0}^{m-1} \beta_m(k)x(n-k) \tag{9.2.35}$$

Since the data are run in reverse order through the predictor, the prediction performed in (9.2.35) is called *backward prediction.* In contrast, the FIR filter with system function $A_m(z)$ is called a *forward predictor.*

In the z-transform domain, (9.2.33) becomes

$$G_m(z) = B_m(z)X(z) \tag{9.2.36}$$

or, equivalently,

$$B_m(z) = \frac{G_m(z)}{X(z)} \tag{9.2.37}$$

where $B_m(z)$ represents the system function of the FIR filter with coefficients $\{\beta_m(k)\}$, that is,

$$B_m(z) = \sum_{k=0}^{m} \beta_m(k)z^{-k} \tag{9.2.38}$$

Since $\beta_m(k) = \alpha_m(m-k)$, (9.2.38) may be expressed as

$$\begin{aligned} B_m(z) &= \sum_{k=0}^{m} \alpha_m(m-k)z^{-k} \\ &= \sum_{l=0}^{m} \alpha_m(l)z^{l-m} \\ &= z^{-m}\sum_{l=0}^{m} \alpha_m(l)z^{l} \\ &= z^{-m}A_m(z^{-1}) \end{aligned} \tag{9.2.39}$$

The relationship in (9.2.39) implies that the zeros of the FIR filter with system function $B_m(z)$ are simply the reciprocals of the zeros of $A_m(z)$. Hence $B_m(z)$ is called the reciprocal or *reverse* polynomial of $A_m(z)$.

Now that we have established these interesting relationships between the direct-form FIR filter and the lattice structure, let us return to the recursive lattice equations in (9.2.28) through (9.2.30) and transfer them to the z-domain. Thus we have

$$F_0(z) = G_0(z) = X(z) \tag{9.2.40}$$

$$F_m(z) = F_{m-1}(z) + K_m z^{-1} G_{m-1}(z), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.41}$$

$$G_m(z) = K_m F_{m-1}(z) + z^{-1} G_{m-1}(z), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.42}$$

If we divide each equation by $X(z)$, we obtain the desired results in the form

$$A_0(z) = B_0(z) = 1 \tag{9.2.43}$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.44}$$

$$B_m(z) = K_m A_{m-1}(z) + z^{-1} B_{m-1}(z), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.45}$$

Thus a lattice stage is described in the z-domain by the matrix equation

$$\begin{bmatrix} A_m(z) \\ B_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_m \\ K_m & 1 \end{bmatrix} \begin{bmatrix} A_{m-1}(z) \\ z^{-1} B_{m-1}(z) \end{bmatrix} \tag{9.2.46}$$

Before concluding this discussion, it is desirable to develop the relationships for converting the lattice parameters $\{K_i\}$, that is, the reflection coefficients, to the direct-form filter coefficients $\{\alpha_m(k)\}$, and vice versa.

Conversion of lattice coefficients to direct-form filter coefficients. The direct-form FIR filter coefficients $\{\alpha_m(k)\}$ can be obtained from the lattice coefficients $\{K_i\}$ by using the following relations:

$$A_0(z) = B_0(z) = 1 \tag{9.2.47}$$

$$A_m(z) = A_{m-1}(z) + K_m z^{-1} B_{m-1}(z), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.48}$$

$$B_m(z) = z^{-m} A_m(z^{-1}), \qquad m = 1, 2, \ldots, M-1 \tag{9.2.49}$$

The solution is obtained recursively, beginning with $m = 1$. Thus we obtain a sequence of $(M-1)$ FIR filters, one for each value of m. The procedure is best illustrated by means of an example.

EXAMPLE 9.2.2

Given a three-stage lattice filter with coefficients $K_1 = \frac{1}{4}$, $K_2 = \frac{1}{4}$, $K_3 = \frac{1}{3}$, determine the FIR filter coefficients for the direct-form structure.

Solution. We solve the problem recursively, beginning with (9.2.48) for $m = 1$. Thus we have

$$\begin{aligned} A_1(z) &= A_0(z) + K_1 z^{-1} B_0(z) \\ &= 1 + K_1 z^{-1} = 1 + \tfrac{1}{4} z^{-1} \end{aligned}$$

Hence the coefficients of an FIR filter corresponding to the single-stage lattice are $\alpha_1(0) = 1$, $\alpha_1(1) = K_1 = \frac{1}{4}$. Since $B_m(z)$ is the reverse polynomial of $A_m(z)$, we have

$$B_1(z) = \tfrac{1}{4} + z^{-1}$$

Next we add the second stage to the lattice. For $m = 2$, (9.2.48) yields

$$\begin{aligned} A_2(z) &= A_1(z) + K_2 z^{-1} B_1(z) \\ &= 1 + \tfrac{3}{8} z^{-1} + \tfrac{1}{2} z^{-2} \end{aligned}$$

Hence the FIR filter parameters corresponding to the two-stage lattice are $\alpha_2(0) = 1$, $\alpha_2(1) = \frac{3}{8}$, $\alpha_2(2) = \frac{1}{2}$. Also,

$$B_2(z) = \tfrac{1}{2} + \tfrac{3}{8} z^{-1} + z^{-2}$$

Finally, the addition of the third stage to the lattice results in the polynomial

$$\begin{aligned} A_3(z) &= A_2(z) + K_3 z^{-1} B_2(z) \\ &= 1 + \tfrac{13}{24} z^{-1} + \tfrac{5}{8} z^{-2} + \tfrac{1}{3} z^{-3} \end{aligned}$$

Consequently, the desired direct-form FIR filter is characterized by the coefficients

$$\alpha_3(0) = 1, \qquad \alpha_3(1) = \tfrac{13}{24}, \qquad \alpha_3(2) = \tfrac{5}{8}, \qquad \alpha_3(3) = \tfrac{1}{3}$$

As this example illustrates, the lattice structure with parameters $K_1, K_2, \ldots, K_m$, corresponds to a class of m direct-form FIR filters with system functions $A_1(z)$, $A_2(z), \ldots, A_m(z)$. It is interesting to note that a characterization of this class of m FIR filters in direct form requires $m(m + 1)/2$ filter coefficients. In contrast, the lattice-form characterization requires only the m reflection coefficients $\{K_i\}$. The reason that the lattice provides a more compact representation for the class of m FIR filters is simply that the addition of stages to the lattice does not alter the parameters of the previous stages. On the other hand, the addition of the mth stage to a lattice with $(m - 1)$ stages results in an FIR filter with system function $A_m(z)$ that has coefficients totally different from the coefficients of the lower-order FIR filter with system function $A_{m-1}(z)$.

A formula for determining the filter coefficients $\{\alpha_m(k)\}$ recursively can be easily derived from polynomial relationships in (9.2.47) through (9.2.49). From the relationship in (9.2.48) we have

$$\begin{aligned} A_m(z) &= A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \\ \sum_{k=0}^{m} \alpha_m(k) z^{-k} &= \sum_{k=0}^{m-1} \alpha_{m-1}(k) z^{-k} + K_m \sum_{k=0}^{m-1} \alpha_{m-1}(m-1-k) z^{-(k+1)} \end{aligned} \tag{9.2.50}$$

By equating the coefficients of equal powers of z^{-1} and recalling that $\alpha_m(0) = 1$ for $m = 1, 2, \ldots, M-1$, we obtain the desired recursive equation for the FIR filter coefficients in the form

$$\alpha_m(0) = 1 \tag{9.2.51}$$

$$\alpha_m(m) = K_m \tag{9.2.52}$$

$$\begin{aligned}\alpha_m(k) &= \alpha_{m-1}(k) + K_m\alpha_{m-1}(m-k) \\ &= \alpha_{m-1}(k) + \alpha_m(m)\alpha_{m-1}(m-k), \qquad \begin{matrix}1 \le k \le m-1 \\ m = 1, 2, \ldots, M-1\end{matrix}\end{aligned} \tag{9.2.53}$$

We note that (9.2.51) through (9.2.53) are simply the Levinson–Durbin recursive equations given in Chapter 12.

Conversion of direct-form FIR filter coefficients to lattice coefficients. Suppose that we are given the FIR coefficients for the direct-form realization or, equivalently, the polynomial $A_m(z)$, and we wish to determine the corresponding lattice filter parameters $\{K_i\}$. For the m-stage lattice we immediately obtain the parameter $K_m = \alpha_m(m)$. To obtain K_{m-1} we need the polynomials $A_{m-1}(z)$ since, in general, K_m is obtained from the polynomial $A_m(z)$ for $m = M-1, M-2, \ldots, 1$. Consequently, we need to compute the polynomials $A_m(z)$ starting from $m = M-1$ and "stepping down" successively to $m = 1$.

The desired recursive relation for the polynomials is easily determined from (9.2.44) and (9.2.45). We have

$$\begin{aligned}A_m(z) &= A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \\ &= A_{m-1}(z) + K_m[B_m(z) - K_m A_{m-1}(z)]\end{aligned}$$

If we solve for $A_{m-1}(z)$, we obtain

$$A_{m-1}(z) = \frac{A_m(z) - K_m B_m(z)}{1 - K_m^2}, \qquad m = M-1, M-2, \ldots, 1 \tag{9.2.54}$$

Thus we compute all lower-degree polynomials $A_m(z)$ beginning with $A_{M-1}(z)$ and obtain the desired lattice coefficients from the relation $K_m = \alpha_m(m)$. We observe that the procedure works as long as $|K_m| \neq 1$ for $m = 1, 2, \ldots, M-1$.

EXAMPLE 9.2.3

Determine the lattice coefficients corresponding to the FIR filter with system function

$$H(z) = A_3(z) = 1 + \tfrac{13}{24}z^{-1} + \tfrac{5}{8}z^{-2} + \tfrac{1}{3}z^{-3}$$

Solution. First we note that $K_3 = \alpha_3(3) = \frac{1}{3}$. Furthermore,

$$B_3(z) = \tfrac{1}{3} + \tfrac{5}{8}z^{-1} + \tfrac{13}{24}z^{-2} + z^{-3}$$

The step-down relationship in (9.2.54) with $m = 3$ yields

$$A_2(z) = \frac{A_3(z) - K_3 B_3(z)}{1 - K_3^2}$$

$$= 1 + \tfrac{3}{8}z^{-1} + \tfrac{1}{2}z^{-2}$$

Hence $K_2 = \alpha_2(2) = \frac{1}{2}$ and $B_2(z) = \frac{1}{2} + \frac{3}{8}z^{-1} + z^{-2}$. By repeating the step-down recursion in (9.2.51), we obtain

$$A_1(z) = \frac{A_2(z) - K_2 B_2(z)}{1 - K_2^2}$$

$$= 1 + \tfrac{1}{4}z^{-1}$$

Hence $K_1 = \alpha_1(1) = \frac{1}{4}$.

From the step-down recursive equation in (9.2.54), it is relatively easy to obtain a formula for recursively computing K_m, beginning with $m = M - 1$ and stepping down to $m = 1$. For $m = M - 1, M - 2, \ldots, 1$ we have

$$K_m = \alpha_m(m), \qquad \alpha_{m-1}(0) = 1 \tag{9.2.55}$$

$$\alpha_{m-1}(k) = \frac{\alpha_m(k) - K_m \beta_m(k)}{1 - K_m^2}$$

$$= \frac{\alpha_m(k) - \alpha_m(m)\alpha_m(m-k)}{1 - \alpha_m^2(m)}, \qquad 1 \le k \le m - 1 \tag{9.2.56}$$

As indicated above, the recursive equation in (9.2.56) breaks down if any lattice parameters $|K_m| = 1$. If this occurs, it is indicative of the fact that the polynomial $A_{m-1}(z)$ has a root on the unit circle. Such a root can be factored out from $A_{m-1}(z)$ and the iterative process in (9.2.56) is carried out for the reduced-order system.

9.3 Structures for IIR Systems

In this section we consider different IIR systems structures described by the difference equation in (9.1.1) or, equivalently, by the system function in (9.1.2). Just as in the case of FIR systems, there are several types of structures or realizations, including direct-form structures, cascade-form structures, lattice structures, and lattice-ladder structures. In addition, IIR systems lend themselves to a parallel-form realization. We begin by describing two direct-form realizations.

9.3.1 Direct-Form Structures

The rational system function as given by (9.1.2) that characterizes an IIR system can be viewed as two systems in cascade, that is,

$$H(z) = H_1(z)H_2(z) \tag{9.3.1}$$

where $H_1(z)$ consists of the zeros of $H(z)$, and $H_2(z)$ consists of the poles of $H(z)$,

$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k} \tag{9.3.2}$$

and

$$H_2(z) = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{9.3.3}$$

In Section 2.5 we described two different direct-form realizations, characterized by whether $H_1(z)$ precedes $H_2(z)$, or vice versa. Since $H_1(z)$ is an FIR system, its direct-form realization was illustrated in Fig. 9.2.1. By attaching the all-pole system in cascade with $H_1(z)$, we obtain the direct form I realization depicted in Fig. 9.3.1. This realization requires $M+N+1$ multiplications, $M+N$ additions, and $M+N+1$ memory locations.

If the all-pole filter $H_2(z)$ is placed before the all-zero filter $H_1(z)$, a more compact structure is obtained as illustrated in Section 2.5. Recall that the difference

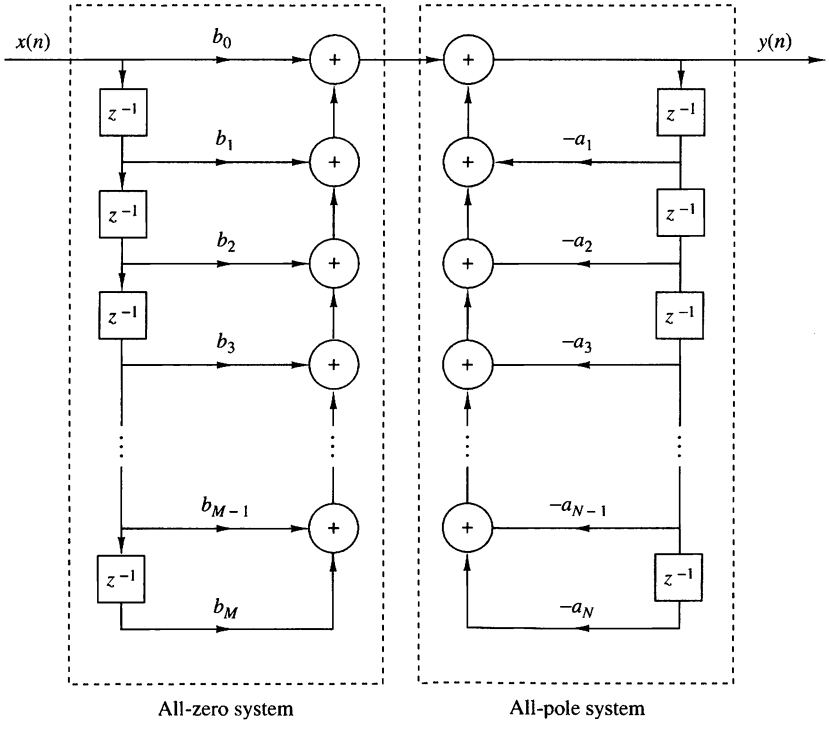


Figure 9.3.1 Direct form I realization.

equation for the all-pole filter is

$$w(n) = -\sum_{k=1}^{N} a_k w(n-k) + x(n) \tag{9.3.4}$$

Since $w(n)$ is the input to the all-zero system, its output is

$$y(n) = \sum_{k=0}^{M} b_k w(n-k) \tag{9.3.5}$$

We note that both (9.3.4) and (9.3.5) involve delayed versions of the sequence $\{w(n)\}$. Consequently, only a single delay line or a single set of memory locations is required for storing the past values of $\{w(n)\}$. The resulting structure that implements (9.3.4) and (9.3.5) is called a direct form II realization and is depicted in Fig. 9.3.2. This structure requires $M+N+1$ multiplications, $M+N$ additions, and the maximum of $\{M, N\}$ memory locations. Since the direct form II realization minimizes the number of memory locations, it is said to be *canonic*. However, we should indicate that other IIR structures also possess this property, so that this terminology is perhaps unjustified.

The structures in Figs. 9.3.1 and 9.3.2 are both called "direct-form" realizations because they are obtained directly from the system function $H(z)$ without any rearrangement of $H(z)$. Unfortunately, both structures are extremely sensitive to parameter quantization, in general, and are not recommended in practical applications. This topic is discussed in detail in Section 9.6, where we demonstrate that when N is large, a small change in a filter coefficient due to parameter quantization results in a large change in the location of the poles and zeros of the system.

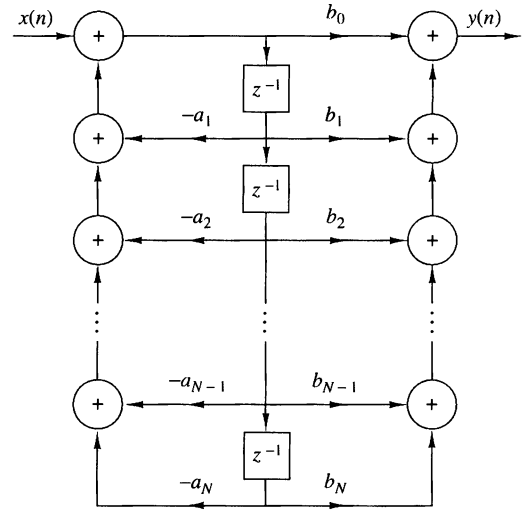


Figure 9.3.2
Direct form II realization ($N = M$).

9.3.2 Signal Flow Graphs and Transposed Structures

A signal flow graph provides an alternative, but equivalent, graphical representation to a block diagram structure that we have been using to illustrate various system realizations. The basic elements of a flow graph are branches and nodes. A signal flow graph is basically a set of directed branches that connect at nodes. By definition, the signal out of a branch is equal to the branch gain (system function) times the signal into the branch. Furthermore, the signal at a node of a flow graph is equal to the sum of the signals from all branches connecting to the node.

To illustrate these basic notions, let us consider the two-pole and two-zero IIR system depicted in block diagram form in Fig. 9.3.3(a). The system block diagram can be converted to the signal flow graph shown in Fig. 9.3.3(b). We note that the flow graph contains five nodes labeled 1 through 5. Two of the nodes (1, 3) are summing nodes (i.e., they contain adders), while the other three nodes represent branching points. Branch transmittances are indicated for the branches in the flow graph. Note that a delay is indicated by the branch transmittance z^{-1}. When the branch transmittance is unity, it is left unlabeled. The input to the system originates at a *source node* and the output signal is extracted at a *sink node.*

We observe that the signal flow graph contains the same basic information as the block diagram realization of the system. The only apparent difference is that *both*

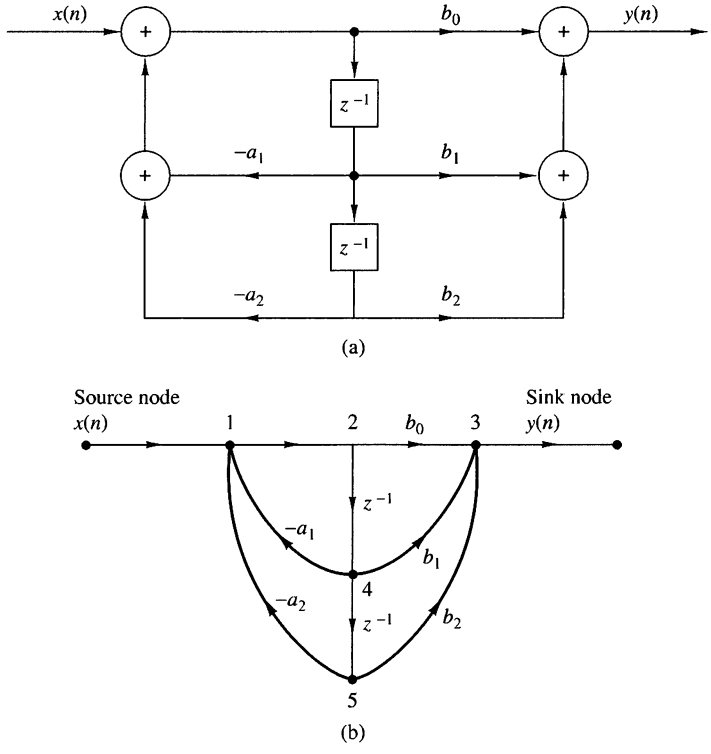


Figure 9.3.3 Second-order filter structure (a) and its signal flow graph (b).

branch points and adders in the block diagram are represented by nodes in the signal flow graph.

The subject of linear signal flow graphs is an important one in the treatment of networks, and many interesting results are available. One basic notion involves the transformation of one flow graph into another without changing the basic input–output relationship. Specifically, one technique that is useful in deriving new system structures for FIR and IIR systems stems from the *transposition* or *flow-graph reversal theorem*. This theorem simply states that if we reverse the directions of all branch transmittances and interchange the input and output in the flow graph, the system function remains unchanged. The resulting structure is called a *transposed structure* or a *transposed form*.

For example, the transposition of the signal flow graph in Fig. 9.3.3(b) is illustrated in Fig. 9.3.4(a). The corresponding block diagram realization of the transposed form is depicted in Fig. 9.3.4(b). It is interesting to note that the transposition of the original flow graph resulted in branching nodes becoming adder nodes, and vice versa.

Let us apply the transposition theorem to the direct form II structure. First, we reverse all the signal flow directions in Fig. 9.3.2. Second, we change nodes into adders and adders into nodes, and finally, we interchange the input and the output.

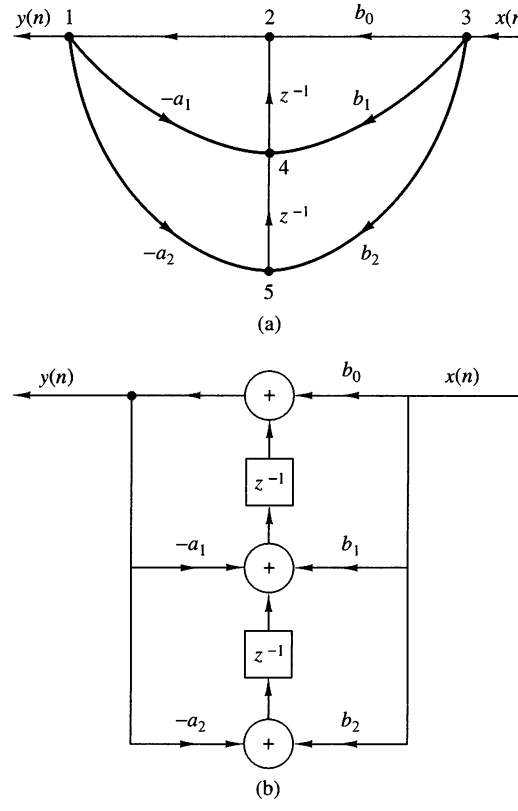


Figure 9.3.4
Signal flow graph of transposed structure (a) and its realization (b).

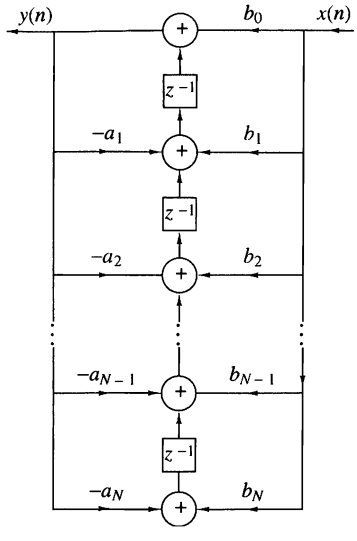


Figure 9.3.5
Transposed direct form II structure.

These operations result in the transposed direct form II structure shown in Fig. 9.3.5. This structure can be redrawn as in Fig. 9.3.6, which shows the input on the left and the output on the right.

The transposed direct form II realization that we have obtained can be described by the set of difference equations

$$y(n) = w_1(n-1) + b_0 x(n) \tag{9.3.6}$$

$$w_k(n) = w_{k+1}(n-1) - a_k y(n) + b_k x(n), \qquad k = 1, 2, \ldots, N-1 \tag{9.3.7}$$

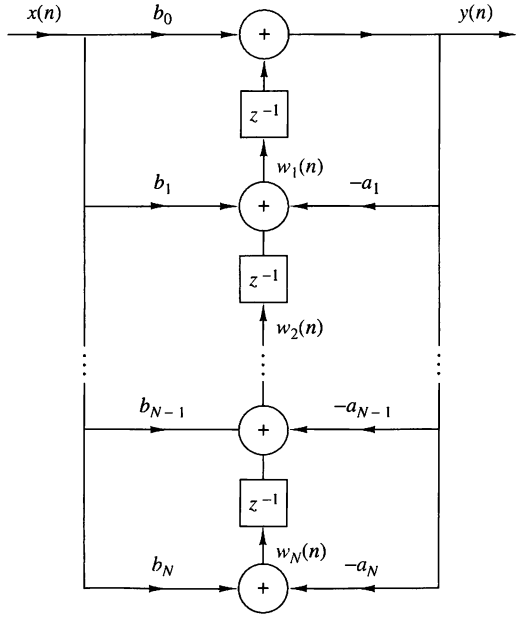


Figure 9.3.6
Transposed direct form II structure.

$$w_N(n) = b_N x(n) - a_N y(n) \tag{9.3.8}$$

Without loss of generality, we have assumed that $M = N$ in writing equations. It is also clear from observation of Fig. 9.3.6 that this set of difference equations is equivalent to the single difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k) \tag{9.3.9}$$

Finally, we observe that the transposed direct form II structure requires the same number of multiplications, additions, and memory locations as the original direct-form II structure.

Although our discussion of transposed structures has been concerned with the general form of an IIR system, it is interesting to note that an FIR system, obtained from (9.3.9) by setting the $a_k = 0$, $k = 1, 2, \ldots, N$, also has a transposed direct form as illustrated in Fig. 9.3.7. This structure is simply obtained from Fig. 9.3.6 by setting $a_k = 0$, $k = 1, 2, \ldots, N$. This transposed-form realization may be described by the set of difference equations

$$w_M(n) = b_M x(n) \tag{9.3.10}$$

$$w_k(n) = w_{k+1}(n-1) + b_k x(n), \qquad k = M-1, M-2, \ldots, 1 \tag{9.3.11}$$

$$y(n) = w_1(n-1) + b_0 x(n) \tag{9.3.12}$$

In summary, Table 9.1 illustrates the direct-form structures and the corresponding difference equations for a basic two-pole and two-zero IIR system with system function

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \tag{9.3.13}$$

This is the basic building block in the cascade realization of high-order IIR systems, as described in the following section. Of the three direct-form structures given in Table 9.1, the direct form II structures are preferable due to the smaller number of memory locations required in their implementation.

Finally, we note that in the z-domain, the set of difference equations describing a linear signal flow graph constitute a linear set of equations. Any rearrangement of such a set of equations is equivalent to a rearrangement of the signal flow graph to obtain a new structure, and vice versa.

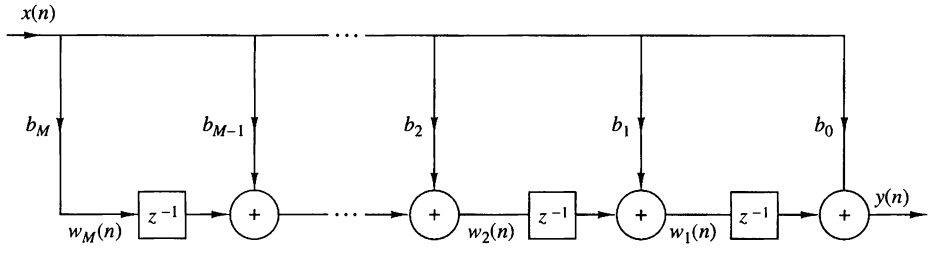
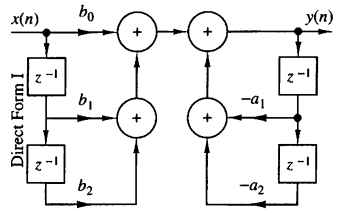
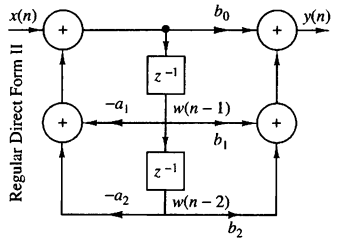
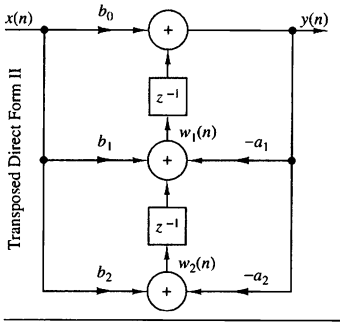


Figure 9.3.7 Transposed FIR structure.

TABLE 9.1 Some Second-Order Modules for Discrete-Time Systems

Structure	Implementation Equations	System Function
	$y(n) = b_0x(n) + b_1x(n-1) + b_2x(n-2) - a_1y(n-1) - a_2y(n-2)$	$\frac{H(z) = b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$
	$w(n) = -a_1w(n-1) - a_2w(n-2) + x(n)$ $y(n) = b_0w(n) + b_1w(n-1) + b_2w(n-2)$	$\frac{H(z) = b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$
	$y(n) = b_0x(n) + w_1(n-1)$ $w_1(n) = b_1x(n) - a_1y(n) + w_2(n-1)$ $w_2(n) = b_2x(n) - a_2y(n)$	$\frac{H(z) = b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}}$

9.3.3 Cascade-Form Structures

Let us consider a high-order IIR system with system function given by (9.1.2). Without loss of generality we assume that $N \geq M$. The system can be factored into a cascade of second-order subsystems, such that $H(z)$ can be expressed as

$$H(z) = \prod_{k=1}^{K} H_k(z) \tag{9.3.14}$$

where K is the integer part of $(N+1)/2$. $H_k(z)$ has the general form

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}} \tag{9.3.15}$$

As in the case of FIR systems based on a cascade-form realization, the parameter b_0 can be distributed equally among the K filter sections so that $b_0 = b_{10}b_{20}\ldots b_{K0}$.

The coefficients $\{a_{ki}\}$ and $\{b_{ki}\}$in the second-order subsystems are real. This implies that in forming the second-order subsystems or quadratic factors in (9.3.15),

we should group together a pair of complex-conjugate poles and we should group together a pair of complex-conjugate zeros. However, the pairing of two complex-conjugate poles with a pair of complex-conjugate zeros or real-valued zeros to form a subsystem of the type given by (9.3.15) can be done arbitrarily. Furthermore, any two real-valued zeros can be paired together to form a quadratic factor and, likewise, any two real-valued poles can be paired together to form a quadratic factor. Consequently, the quadratic factor in the numerator of (9.3.15) may consist of either a pair of real roots or a pair of complex-conjugate roots. The same statement applies to the denominator of (9.3.15).

If $N > M$, some of the second-order subsystems have numerator coefficients that are zero, that is, either $b_{k2} = 0$ or $b_{k1} = 0$ or both $b_{k2} = b_{k1} = 0$ for some k. Furthermore, if N is odd, one of the subsystems, say $H_k(z)$, must have $a_{k2} = 0$, so that the subsystem is of first order. To preserve the modularity in the implementation of $H(z)$, it is often preferable to use the basic second-order subsystems in the cascade structure and have some zero-valued coefficients in some of the subsystems.

Each of the second-order subsystems with system function of the form (9.3.15) can be realized in either direct form I, or direct form II, or transposed direct form II. Since there are many ways to pair the poles and zeros of $H(z)$ into a cascade of second-order sections, and several ways to order the resulting subsystems, it is possible to obtain a variety of cascade realizations. Although all cascade realizations are equivalent for infinite-precision arithmetic, the various realizations may differ significantly when implemented with finite-precision arithmetic.

The general form of the cascade structure is illustrated in Fig. 9.3.8. If we use the direct form II structure for each of the subsystems, the computational algorithm for realizing the IIR system with system function $H(z)$ is described by the following set of equations.

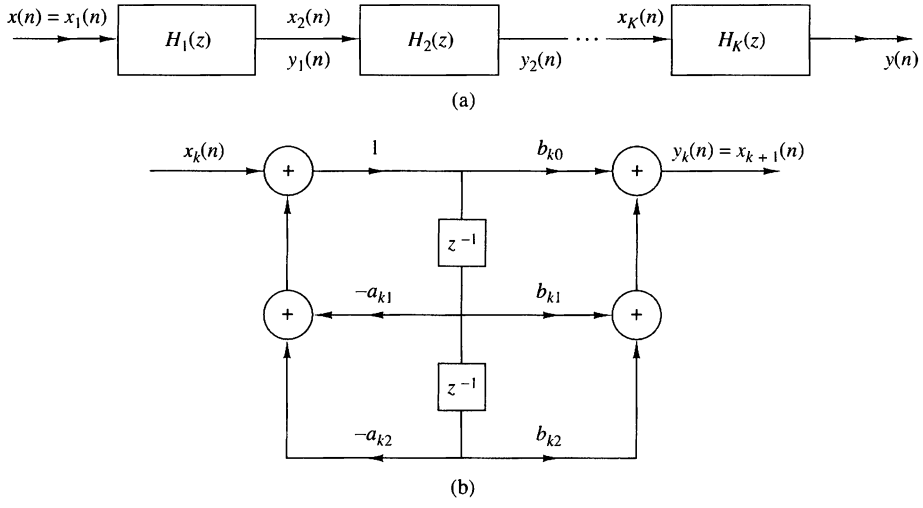


Figure 9.3.8 Cascade structure of second-order systems and a realization of each second-order section.

$$y_0(n) = x(n) \tag{9.3.16}$$

$$w_k(n) = -a_{k1}w_k(n-1) - a_{k2}w_k(n-2) + y_{k-1}(n), \qquad k = 1, 2, \ldots, K \tag{9.3.17}$$

$$y_k(n) = b_{k0}w_k(n) + b_{k1}w_k(n-1) + b_{k2}w_k(n-2), \qquad k = 1, 2, \ldots, K \tag{9.3.18}$$

$$y(n) = y_K(n) \tag{9.3.19}$$

Thus this set of equations provides a complete description of the cascade structure based on direct form II sections.

9.3.4 Parallel-Form Structures

A parallel-form realization of an IIR system can be obtained by performing a partial-fraction expansion of $H(z)$. Without loss of generality, we again assume that $N \geq M$ and that the poles are distinct. Then, by performing a partial-fraction expansion of $H(z)$, we obtain the result

$$H(z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}} \tag{9.3.20}$$

where $\{p_k\}$ are the poles, $\{A_k\}$ are the coefficients (residues) in the partial-fraction expansion, and the constant C is defined as $C = b_N/a_N$. The structure implied by (9.3.20) is shown in Fig. 9.3.9. It consists of a parallel bank of single-pole filters.

In general, some of the poles of $H(z)$ may be complex valued. In such a case, the corresponding coefficients A_k are also complex valued. To avoid multiplications by complex numbers, we can combine pairs of complex-conjugate poles to form two-pole subsystems. In addition, we can combine, in an arbitrary manner, pairs of real-valued poles to form two-pole subsystems. Each of these subsystems has the form

$$H_k(z) = \frac{b_{k0} + b_{k1}z^{-1}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}} \tag{9.3.21}$$

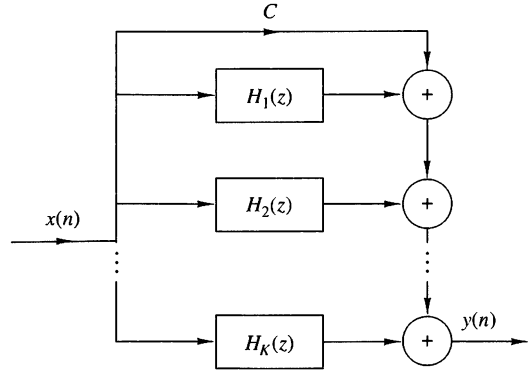


Figure 9.3.9
Parallel structure of IIR system.

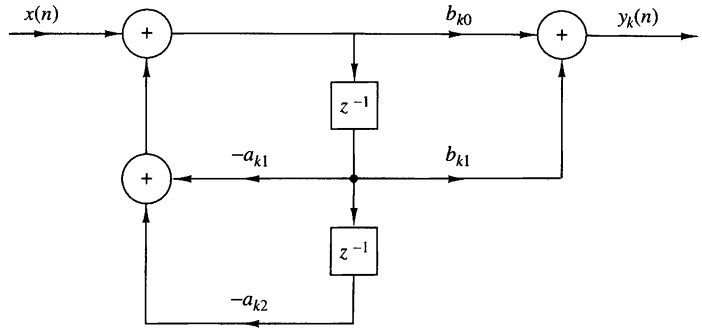


Figure 9.3.10 Structure of second-order section in a parallel IIR system realization.

where the coefficients $\{b_{ki}\}$ and $\{a_{ki}]$ are real-valued system parameters. The overall function can now be expressed as

$$H(z) = C + \sum_{k=1}^{K} H_k(z) \tag{9.3.22}$$

where K is the integer part of $(N+1)/2$. When N is odd, one of the $H_k(z)$ is really a single-pole system (i.e., $b_{k1} = a_{k2} = 0$).

The individual second-order sections which are the basic building blocks for $H(z)$ can be implemented in either of the direct forms or in a transposed direct form. The direct form II structure is illustrated in Fig. 9.3.10. With this structure as a basic building block, the parallel-form realization of the FIR system is described by the following set of equations:

$$w_k(n) = -a_{k1}w_k(n-1) - a_{k2}w_k(n-2) + x(n), \qquad k = 1, 2, \ldots, K \tag{9.3.23}$$

$$y_k(n) = b_{k0}w_k(n) + b_{k1}w_k(n-1), \qquad k = 1, 2, \ldots, K \tag{9.3.24}$$

$$y(n) = Cx(n) + \sum_{k=1}^{K} y_k(n) \tag{9.3.25}$$

EXAMPLE 9.3.1

Determine the cascade and parallel realizations for the system described by the system function

$$H(z) = \frac{10(1 - \frac{1}{2}z^{-1})(1 - \frac{2}{3}z^{-1})(1 + 2z^{-1})}{(1 - \frac{3}{4}z^{-1})(1 - \frac{1}{8}z^{-1})[1 - (\frac{1}{2} + j\frac{1}{2})z^{-1}][1 - (\frac{1}{2} - j\frac{1}{2})z^{-1}]}$$

Solution. The cascade realization is easily obtained from this form. One possible pairing of poles and zeros is

$$H_1(z) = \frac{1 - \frac{2}{3}z^{-1}}{1 - \frac{7}{8}z^{-1} + \frac{3}{32}z^{-2}}$$

$$H_2(z) = \frac{1 + \frac{3}{2}z^{-1} - z^{-2}}{1 - z^{-1} + \frac{1}{2}z^{-2}}$$

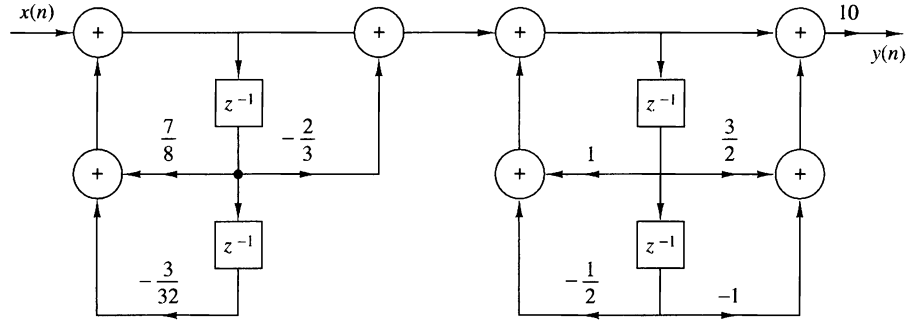
and hence

$$H(z) = 10H_1(z)H_2(z)$$

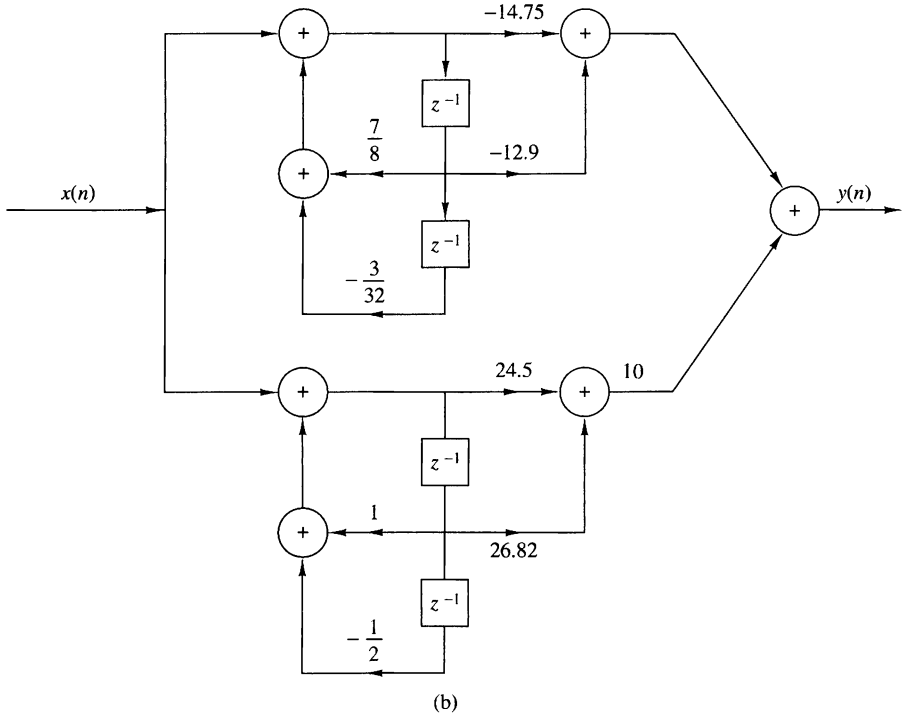
The cascade realization is depicted in Fig. 9.3.11(a).

To obtain the parallel-form realization, $H(z)$ must be expanded in partial fractions. Thus we have

$$H(z) = \frac{A_1}{1 - \frac{3}{4}z^{-1}} + \frac{A_2}{1 - \frac{1}{8}z^{-1}} + \frac{A_3}{1 - (\frac{1}{2} + j\frac{1}{2})z^{-1}} + \frac{A_3^*}{1 - (\frac{1}{2} - j\frac{1}{2})z^{-1}}$$



(a)



(b)

Figure 9.3.11 Cascade and parallel realizations for the system in Example 9.3.1.

where A_1, A_2, A_3, and A_3^* are to be determined. After some arithmetic we find that

$$A_1 = 2.93, \quad A_2 = -17.68, \quad A_3 = 12.25 - j14.57, \quad A_3^* = 12.25 + j14.57$$

Upon recombining pairs of poles, we obtain

$$H(z) = \frac{-14.75 - 12.90z^{-1}}{1 - \frac{7}{8}z^{-1} + \frac{3}{32}z^{-2}} + \frac{24.50 + 26.82z^{-1}}{1 - z^{-1} + \frac{1}{2}z^{-2}}$$

The parallel-form realization is illustrated in Fig. 9.3.11(b).

9.3.5 Lattice and Lattice-Ladder Structures for IIR Systems

In Section 9.2.4 we developed a lattice filter structure that is equivalent to an FIR system. In this section we extend the development to IIR systems.

Let us begin with an all-pole system with system function

$$H(z) = \frac{1}{1 + \sum_{k=1}^{N} a_N(k)z^{-k}} = \frac{1}{A_N(z)} \tag{9.3.26}$$

The direct-form realization of this system is illustrated in Fig. 9.3.12. The difference equation for this IIR system is

$$y(n) = -\sum_{k=1}^{N} a_N(k)y(n-k) + x(n) \tag{9.3.27}$$

It is interesting to note that if we interchange the roles of input and output [i.e., interchange $x(n)$ with $y(n)$ in (9.3.27)], we obtain

$$x(n) = -\sum_{k=1}^{N} a_N(k)x(n-k) + y(n)$$

or, equivalently,

$$y(n) = x(n) + \sum_{k=1}^{N} a_N(k)x(n-k) \tag{9.3.28}$$

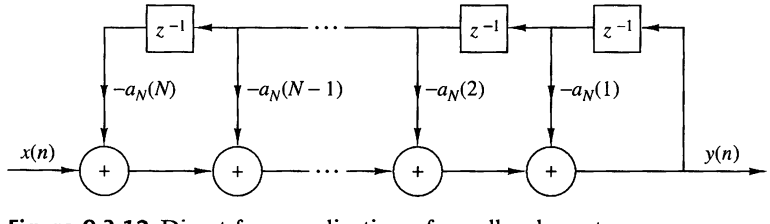


Figure 9.3.12 Direct-form realization of an all-pole system.

We note that the equation in (9.3.28) describes an FIR system having the system function $H(z) = A_N(z)$, while the system described by the difference equation in (9.3.27) represents an IIR system with system function $H(z) = 1/A_N(z)$. One system can be obtained from the other simply by interchanging the roles of the input and output.

Based on this observation, we shall use the all-zero (FIR) lattice described in Section 9.2.4 to obtain a lattice structure for an all-pole IIR system by interchanging the roles of the input and output. First, we take the all-zero lattice filter illustrated in Fig. 9.2.11 and then redefine the input as

$$x(n) = f_N(n) \tag{9.3.29}$$

and the output as

$$y(n) = f_0(n) \tag{9.3.30}$$

These are exactly the opposite of the definitions for the all-zero lattice filter. These definitions dictate that the quantities $\{f_m(n)\}$ be computed in descending order [i.e., $f_N(n)$, $f_{N-1}(n), \ldots$]. This computation can be accomplished by rearranging the recursive equation in (9.2.29) and thus solving for $f_{m-1}(n)$ in terms of $f_m(n)$, that is,

$$f_{m-1}(n) = f_m(n) - K_m g_{m-1}(n-1), \qquad m = N, N-1, \ldots, 1$$

The equation (9.2.30) for $g_m(n)$ remains unchanged.

The result of these changes is the set of equations

$$f_N(n) = x(n) \tag{9.3.31}$$

$$f_{m-1}(n) = f_m(n) - K_m g_{m-1}(n-1), \qquad m = N, N-1, \ldots, 1 \tag{9.3.32}$$

$$g_m(n) = K_m f_{m-1}(n) + g_{m-1}(n-1), \qquad m = N, N-1, \ldots, 1 \tag{9.3.33}$$

$$y(n) = f_0(n) = g_0(n) \tag{9.3.34}$$

which correspond to the structure shown in Fig. 9.3.13.

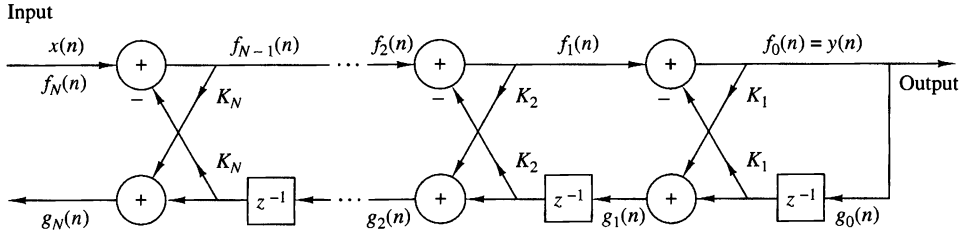


Figure 9.3.13 Lattice structure for an all-pole IIR system.

To demonstrate that the set of equations (9.3.31) through (9.3.34) represent an all-pole IIR system, let us consider the case where $N = 1$. The equations reduce to

$$\begin{aligned} x(n) &= f_1(n) \\ f_0(n) &= f_1(n) - K_1 g_0(n-1) \\ g_1(n) &= K_1 f_0(n) + g_0(n-1) \\ y(n) &= f_0(n) \\ &= x(n) - K_1 y(n-1) \end{aligned} \tag{9.3.35}$$

Furthermore, the equation for $g_1(n)$ can be expressed as

$$g_1(n) = K_1 y(n) + y(n-1) \tag{9.3.36}$$

We observe that (9.3.35) represents a first-order all-pole IIR system while (9.3.36) represents a first-order FIR system. The pole is a result of the feedback introduced by the solution of the $\{f_m(n)\}$ in descending order. This feedback is depicted in Fig. 9.3.14(a).

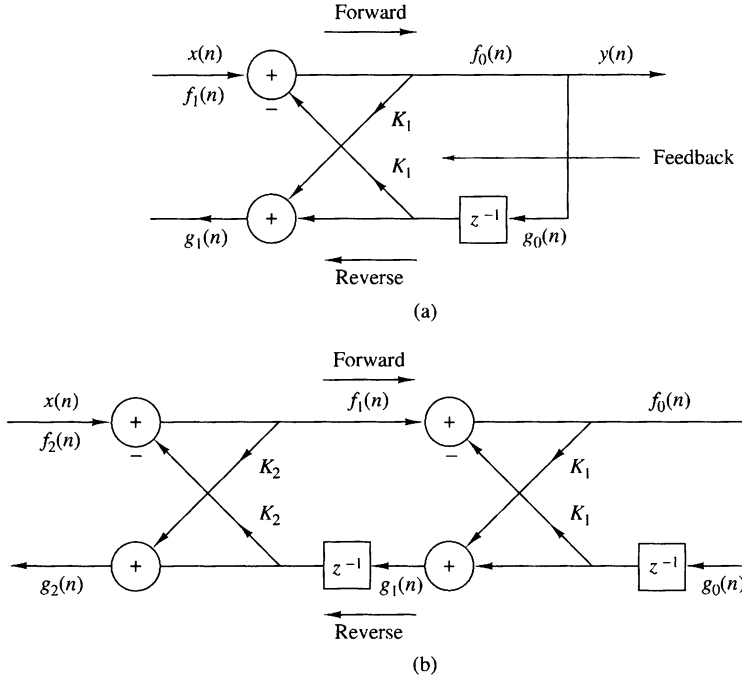


Figure 9.3.14 Single-pole and two-pole lattice system.

Next, let us consider the case $N = 2$, which corresponds to the structure in Fig. 9.3.14(b). The equations corresponding to this structure are

$$\begin{aligned} f_2(n) &= x(n) \\ f_1(n) &= f_2(n) - K_2 g_1(n-1) \\ g_2(n) &= K_2 f_1(n) + g_1(n-1) \\ f_0(n) &= f_1(n) - K_1 g_0(n-1) \\ g_1(n) &= K_1 f_0(n) + g_0(n-1) \\ y(n) &= f_0(n) = g_0(n) \end{aligned} \tag{9.3.37}$$

After some simple substitutions and manipulations we obtain

$$y(n) = -K_1(1 + K_2)y(n-1) - K_2 y(n-2) + x(n) \tag{9.3.38}$$

$$g_2(n) = K_2 y(n) + K_1(1 + K_2)y(n-1) + y(n-2) \tag{9.3.39}$$

Clearly, the difference equation in (9.3.38) represents a two-pole IIR system, and the relation in (9.3.39) is the input–output equation for a two-zero FIR system. Note that the coefficients for the FIR system are identical to those in the IIR system except that they occur in reverse order.

In general, these conclusions hold for any N. Indeed, with the definition of $A_m(z)$ given in (9.2.32), the system function for the all-pole IIR system is

$$H_a(z) = \frac{Y(z)}{X(z)} = \frac{F_0(z)}{F_m(z)} = \frac{1}{A_m(z)} \tag{9.3.40}$$

Similarly, the system function of the all-zero (FIR) system is

$$H_b(z) = \frac{G_m(z)}{Y(z)} = \frac{G_m(z)}{G_0(z)} = B_m(z) = z^{-m} A_m(z^{-1}) \tag{9.3.41}$$

where we used the previously established relationships in (9.2.36) through (9.2.42). Thus the coefficients in the FIR system $H_b(z)$ are identical to the coefficients in $A_m(z)$, except that they occur in reverse order.

It is interesting to note that the all-pole lattice structure has an all-zero path with input $g_0(n)$ and output $g_N(n)$, which is identical to its counterpart all-zero path in the all-zero lattice structure. The polynomial $B_m(z)$, which represents the system function of the all-zero path common to both lattice structures, is usually called the *backward system function*, because it provides a backward path in the all-pole lattice structure.

From this discussion the reader should observe that the all-zero and all-pole lattice structures are characterized by the same set of lattice parameters, namely, $K_1, K_2, \ldots, K_N$. The two lattice structures differ only in the interconnections of their signal flow graphs. Consequently, the algorithms for converting between the

system parameters $\{\alpha_m(k)\}$ in the direct-form realization of an FIR system, and the parameters of its lattice counterpart apply as well to the all-pole structure.

We recall that the roots of the polynomial $A_N(z)$ lie inside the unit circle if and only if the lattice parameters $|K_m| < 1$ for all $m = 1, 2, \ldots, N$. Therefore, the all-pole lattice structure is a stable system if and only if its parameters $|K_m| < 1$ for all m.

In practical applications the all-pole lattice structure has been used to model the human vocal tract and a stratified earth. In such cases the lattice parameters, $\{K_m\}$, have the physical significance of being identical to reflection coefficients in the physical medium. This is the reason that the lattice parameters are often called *reflection coefficients*. In such applications, a stable model of the medium requires that the reflection coefficients, obtained by performing measurements on output signals from the medium, be less than unity.

The all-pole lattice provides the basic building block for lattice-type structures that implement IIR systems that contain both poles and zeros. To develop the appropriate structure, let us consider an IIR system with system function

$$H(z) = \frac{\sum_{k=0}^{M} c_M(k) z^{-k}}{1 + \sum_{k=1}^{N} a_N(k) z^{-k}} = \frac{C_M(z)}{A_N(z)} \tag{9.3.42}$$

where the notation for the numerator polynomial has been changed to avoid confusion with our previous development. Without loss of generality, we assume that $N \geq M$.

In the direct form II structure, the system in (9.3.42) is described by the difference equations

$$w(n) = -\sum_{k=1}^{N} a_N(k) w(n-k) + x(n) \tag{9.3.43}$$

$$y(n) = \sum_{k=0}^{M} c_M(k) w(n-k) \tag{9.3.44}$$

Note that (9.3.43) is the input–output of an all-pole IIR system and that (9.3.44) is the input–output of an all-zero system. Furthermore, we observe that the output of the all-zero system is simply a linear combination of delayed outputs from the all-pole system. This is easily seen by observing the direct form II structure redrawn as in Fig. 9.3.15.

Since zeros result from forming a linear combination of previous outputs, we can carry over this observation to construct a pole–zero IIR system using the all-pole lattice structure as the basic building block. We have already observed that $g_m(n)$ is a linear combination of present and past outputs. In fact, the system

$$H_b(z) = \frac{G_m(z)}{Y(z)} = B_m(z)$$

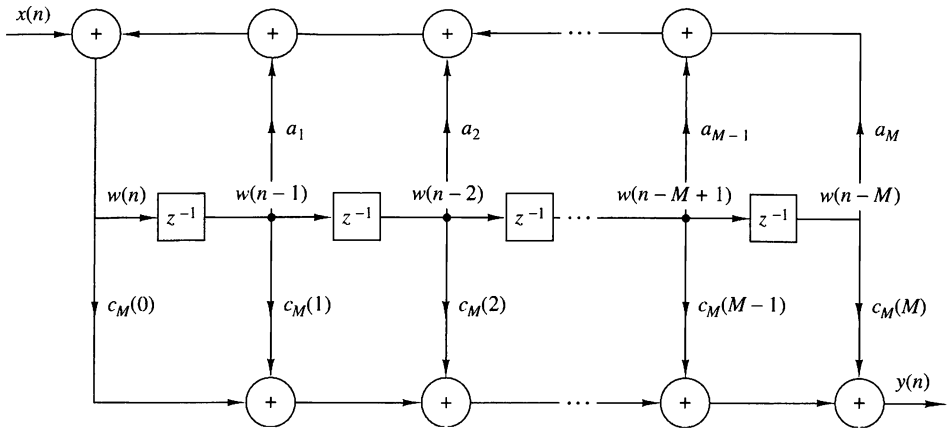


Figure 9.3.15 Direct form II realization of IIR system.

is an all-zero system. Therefore, any linear combination of $\{g_m(n)\}$ is also an all-zero system.

Thus we begin with an all-pole lattice structure with parameters K_m, $1 \leq m \leq N$, and we add a *ladder* part by taking as the output a weighted linear combination of $\{g_m(n)\}$. The result is a pole–zero IIR system which has the *lattice-ladder* structure shown in Fig. 9.3.16 for $M = N$. Its output is

$$y(n) = \sum_{m=0}^{M} v_m g_m(n) \tag{9.3.45}$$

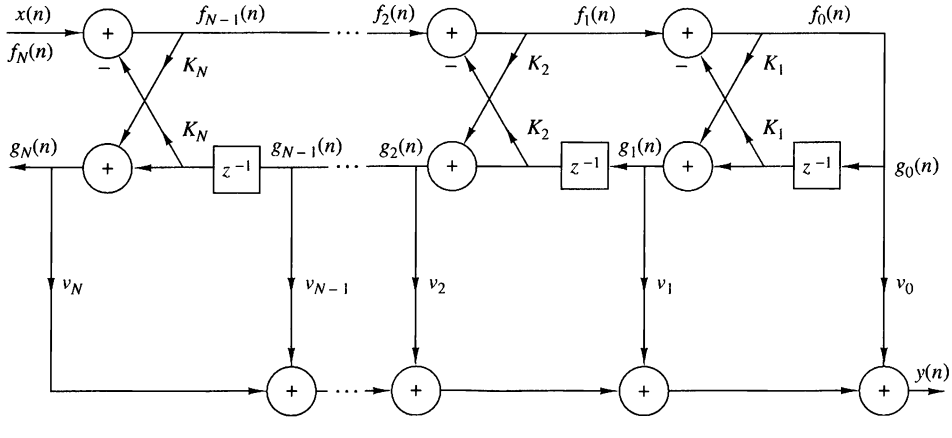


Figure 9.3.16 Lattice-ladder structure for the realization of a pole–zero system.

where $\{v_m\}$ are the parameters that determine the zeros of the system. The system function corresponding to (9.3.45) is

$$\begin{aligned} H(z) &= \frac{Y(z)}{X(z)} \\ &= \sum_{m=0}^{M} v_m \frac{G_m(z)}{X(z)} \end{aligned} \tag{9.3.46}$$

Since $X(z) = F_N(z)$ and $F_0(z) = G_0(z)$, (9.3.46) can be written as

$$\begin{aligned} H(z) &= \sum_{m=0}^{M} v_m \frac{G_m(z)}{G_0(z)} \frac{F_0(z)}{F_N(z)} \\ &= \sum_{m=0}^{M} v_m \frac{B_m(z)}{A_N(z)} \\ &= \frac{\sum_{m=0}^{M} v_m B_m(z)}{A_N(z)} \end{aligned} \tag{9.3.47}$$

If we compare (9.3.41) with (9.3.47), we conclude that

$$C_M(z) = \sum_{m=0}^{M} v_m B_m(z) \tag{9.3.48}$$

This is the desired relationship that can be used to determine the weighting coefficients $\{v_m\}$. Thus, we have demonstrated that the coefficients of the numerator polynomial $C_M(z)$ determine the ladder parameters $\{v_m\}$, whereas the coefficients in the denominator polynomial $A_N(z)$ determine the lattice parameters $\{K_m\}$.

Given the polynomials $C_M(z)$ and $A_N(z)$, where $N \geq M$, the parameters of the all-pole lattice are determined first, as described previously, by the conversion algorithm given in Section 9.2.4, which converts the direct-form coefficients into lattice parameters. By means of the step-down recursive relations given by (9.2.54), we obtain the lattice parameters $\{K_m\}$ and the polynomials $B_m(z)$, $m = 1, 2, \ldots, N$.

The ladder parameters are determined from (9.3.48), which can be expressed as

$$C_m(z) = \sum_{k=0}^{m-1} v_k B_k(z) + v_m B_m(z) \tag{9.3.49}$$

or, equivalently, as

$$C_m(z) = C_{m-1}(z) + v_m B_m(z) \tag{9.3.50}$$

Thus $C_m(z)$ can be computed recursively from the reverse polynomials $B_m(z)$, $m = 1, 2, \ldots, M$. Since $\beta_m(m) = 1$ for all m, the parameters v_m, $m = 0, 1, \ldots, M$ can be determined by first noting that

$$v_m = c_m(m), \qquad m = 0, 1, \ldots, M \tag{9.3.51}$$

Then, by rewriting (9.3.50) as

$$C_{m-1}(z) = C_m(z) - v_m B_m(z) \tag{9.3.52}$$

and running this recursive relation backward in m (i.e., $m = M, M-1, \ldots, 2$), we obtain $c_m(m)$ and therefore the ladder parameters according to (9.3.51).

The lattice-ladder filter structures that we have presented require the minimum amount of memory but not the minimum number of multiplications. Although lattice structures with only one multiplier per lattice stage exist, the two multiplier-per-stage lattice that we have described is by far the most widely used in practical applications. In conclusion, the modularity, the built-in stability characteristics embodied in the coefficients $\{K_m\}$, and its robustness to finite-word-length effects make the lattice structure very attractive in many practical applications, including speech processing systems, adaptive filtering, and geophysical signal processing.

9.4 Representation of Numbers

Up to this point we have considered the implementation of discrete-time systems without being concerned about the finite-word-length effects that are inherent in any digital realization, whether it be in hardware or in software. In fact, we have analyzed systems that are modeled as linear when, in fact, digital realizations of such systems are inherently nonlinear.

In this and the following two sections, we consider the various forms of quantization effects that arise in digital signal processing. Although we describe floating-point arithmetic operations briefly, our major concern is with fixed-point realizations of digital filters.

In this section we consider the representation of numbers for digital computations. The main characteristic of digital arithmetic is the limited (usually fixed) number of digits used to represent numbers. This constraint leads to finite numerical precision in computations, which leads to round-off errors and nonlinear effects in the performance of digital filters. We now provide a brief introduction to digital arithmetic.

9.4.1 Fixed-Point Representation of Numbers

The representation of numbers in a fixed-point format is a generalization of the familiar decimal representation of a number as a string of digits with a decimal point. In this notation, the digits to the left of the decimal point represent the integer part

of the number, and the digits to the right of the decimal point represent the fractional part of the number. Thus a real number X can be represented as

$$\begin{aligned} X &= (b_{-A}, \ldots, b_{-1}, b_0, b_1, \ldots, b_B)_r \\ &= \sum_{i=-A}^{B} b_i r^{-i}, \qquad 0 \le b_i \le (r-1) \end{aligned} \tag{9.4.1}$$

where b_i represents the digit, r is the radix or base, A is the number of integer digits, and B is the number of fractional digits. As an example, the decimal number $(123.45)_{10}$ and the binary number $(101.0)_2$ represent the following sums:

$$(123.45)_{10} = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

$$(101.01)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

Let us focus our attention on the binary representation since it is the most important for digital signal processing. In this case $r = 2$ and the digits $\{b_i\}$ are called binary digits or bits and take the values $\{0, 1\}$. The binary digit b_{-A} is called the most significant bit (MSB) of the number, and the binary digit b_B is called the least significant bit (LSB). The "binary point" between the digits b_0 and b_1 does not exist physically in the computer. Simply, the logic circuits of the computer are designed so that the computations result in numbers that correspond to the assumed location of this point.

By using an n-bit integer format ($A = n - 1$, $B = 0$), we can represent unsigned integers with magnitude in the range 0 to $2^n - 1$. Usually, we use the fraction format ($A = 0$, $B = n - 1$), with a binary point between b_0 and b_1, that permits numbers in the range from 0 to $1 - 2^{-n}$. Note that any integer or mixed number can be represented in a fraction format by factoring out the term r^A in (9.4.1). In the sequel we focus our attention on the binary fraction format because mixed numbers are difficult to multiply and the number of bits representing an integer cannot be reduced by truncation or rounding.

There are three ways to represent negative numbers. This leads to three formats for the representation of signed binary fractions. The format for positive fractions is the same in all three representations, namely,

$$X = 0.b_1 b_2 \cdots b_B = \sum_{i=1}^{B} b_i \cdot 2^{-i}, \qquad X \ge 0 \tag{9.4.2}$$

Note that the MSB b_0 is set to zero to represent the positive sign. Consider now the negative fraction

$$X = -0.b_1 b_2 \cdots b_B = -\sum_{i=1}^{B} b_i \cdot 2^{-i} \tag{9.4.3}$$

This number can be represented using one of the following three formats.

Sign-magnitude format. In this format, the MSB is set to 1 to represent the negative sign,

$$X_{\text{SM}} = 1.b_1 b_2 \cdots b_B, \qquad \text{for } X \leq 0 \tag{9.4.4}$$

One's-complement format. In this format the negative numbers are represented as

$$X_{1\text{C}} = 1.\bar{b}_1 \bar{b}_2 \cdots \bar{b}_B, \qquad X \leq 0 \tag{9.4.5}$$

where $\bar{b}_i = 1 - b_i$ is the one's complement of b_i. Thus if X is a positive number, the corresponding negative number is determined by complementing (changing 1's to 0's and 0's to 1's) all the bits. An alternative definition for $X_{1\text{C}}$ can be obtained by noting that

$$X_{1\text{C}} = 1 \times 2^0 + \sum_{i=1}^{B} (1 - b_i) \cdot 2^{-i} = 2 - 2^{-B}|X| \tag{9.4.6}$$

Two's-complement format. In this format a negative number is represented by forming the two's complement of the corresponding positive number. In other words, the negative number is obtained by subtracting the positive number from 2.0. More simply, the two's complement is formed by complementing the positive number and adding one LSB. Thus

$$X_{2\text{C}} = 1.\bar{b}_1 \bar{b}_2 \cdots \bar{b}_B + 00 \cdots 01, \qquad X < 0 \tag{9.4.7}$$

where $+$ represents modulo-2 addition that ignores any carry generated from the sign bit. For example, the number $-\frac{3}{8}$ is simply obtained by complementing 0011 ($\frac{3}{8}$) to obtain 1100 and then adding 0001. This yields 1101, which represents $-\frac{3}{8}$ in two's complement.

From (9.4.6) and (9.4.7) it can easily be seen that

$$X_{2\text{C}} = X_{1\text{C}} + 2^{-B} = 2 - |X| \tag{9.4.8}$$

To demonstrate that (9.4.7) truly represents a negative number, we use the identity

$$1 = \sum_{i=1}^{B} 2^{-i} + 2^{-B} \tag{9.4.9}$$

The negative number X in (9.4.3) can be expressed as

$$\begin{aligned} X_{2\text{C}} &= -\sum_{i=1}^{B} b_i \cdot 2^{-i} + 1 - 1 \\ &= -1 + \sum_{i=1}^{B} (1 - b_i) 2^{-i} + 2^{-B} \\ &= -1 + \sum_{i=1}^{B} \bar{b}_i \cdot 2^{-1} + 2^{-B} \end{aligned}$$

which is exactly the two's-complement representation of (9.4.7).

In summary, the value of a binary string $b_0 b_1 \cdots b_B$ depends on the format used. For positive numbers, $b_0 = 0$, and the number is given by (9.4.2). For negative numbers, we use these corresponding formulas for the three formats.

EXAMPLE 9.4.1

Express the fraction $\frac{7}{8}$ and $-\frac{7}{8}$ in sign-magnitude, two's-complement, and one's-complement format.

Solution. $X = \frac{7}{8}$ is represented as $2^{-1} + 2^{-2} + 2^{-3}$, so that $X = 0.111$. In sign-magnitude format, $X = -\frac{7}{8}$ is represented as 1.111. In one's complement, we have

$$X_{1C} = 1.000$$

In two's complement, the result is

$$X_{2C} = 1.000 + 0.001 = 1.001$$

The basic arithmetic operations of addition and multiplication depend on the format used. For one's-complement and two's-complement formats, addition is carried out by adding the numbers bit by bit. The formats differ only in the way in which a carry bit affects the MSB. For example, $\frac{4}{8} - \frac{3}{8} = \frac{1}{8}$. In two's complement, we have

$$0100 \oplus 1101 = 0001$$

where $\oplus$ indicates modulo-2 addition. Note that the carry bit, if present in the MSB, is dropped. On the other hand, in one's-complement arithmetic, the carry in the MSB, if present, is carried around to the LSB. Thus the computation $\frac{4}{8} - \frac{3}{8} = \frac{1}{8}$ becomes

$$0100 \oplus 1100 = 0000 \oplus 0001 = 0001$$

Addition in the sign-magnitude format is more complex and can involve sign checks, complementing, and the generation of a carry. On the other hand, direct multiplication of two sign-magnitude numbers is relatively straightforward, whereas a special algorithm is usually employed for one's-complement and two's-complement multiplication.

Most fixed-point digital signal processors use two's-complement arithmetic. Hence, the range for $(B + 1)$-bit numbers is from -1 to $1 - 2^{-B}$. These numbers can be viewed in a wheel format as shown in Fig. 9.4.1 for $B = 2$. Two's-complement arithmetic is basically arithmetic modulo 2^{B+1} [i.e., any number that falls outside the range (overflow or underflow) is reduced to this range by subtracting an appropriate multiple of 2^{B+1}]. This type of arithmetic can be viewed as counting using the wheel of Fig. 9.4.1. A very important property of two's-complement addition is that if the final sum of a string of numbers $X_1, X_2, \ldots, X_N$ is within the range, it will be computed correctly, even if individual partial sums result in overflows. This and other characteristics of two's-complement arithmetic are considered in Problem 9.29.

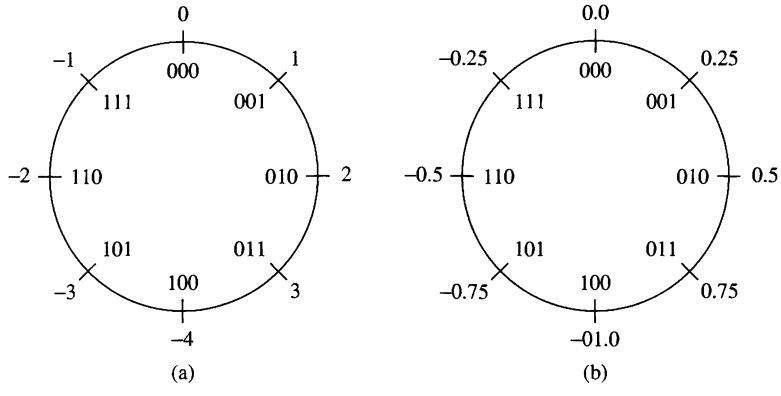


Figure 9.4.1 Counting wheel for 3-bit two's-complement numbers: (a) integers and (b) fractions.

In general, the multiplication of two fixed-point numbers each b bits in length results in a product of $2b$ bits in length. In fixed-point arithmetic, the product is either truncated or rounded back to b bits. As a result we have a truncation or round-off error in the b least significant bits. The characterization of such errors is treated below.

9.4.2 Binary Floating-Point Representation of Numbers

A fixed-point representation of numbers allows us to cover a range of numbers, say, $x_{\max} - x_{\min}$ with a resolution

$$\Delta = \frac{x_{\max} - x_{\min}}{m - 1}$$

where $m = 2^b$ is the number of levels and b is the number of bits. A basic characteristic of the fixed-point representation is that the resolution is fixed. Furthermore, Δ increases in direct proportion to an increase in the dynamic range.

A floating-point representation can be employed as a means for covering a larger dynamic range. The binary floating-point representation commonly used in practice consists of a mantissa M, which is the fractional part of the number and falls in the range $\frac{1}{2} \leq M < 1$, multiplied by the exponential factor 2^E, where the exponent E is either a positive or negative integer. Hence a number X is represented as

$$X = M \cdot 2^E$$

The mantissa requires a sign bit for representing positive and negative numbers, and the exponent requires an additional sign bit. Since the mantissa is a signed fraction, we can use any of the four fixed-point representations just described.

For example, the number $X_1 = 5$ is represented by the following mantissa and exponent:

$$M_1 = 0.101000$$

$$E_1 = 011$$

while the number $X_2 = \frac{3}{8}$ is represented by the following mantissa and exponent

$$M_2 = 0.110000$$

$$E_2 = 101$$

where the leftmost bit in the exponent represents the sign bit.

If the two numbers are to be multiplied, the mantissas are multiplied and the exponents are added. Thus the product of these two numbers is

$$\begin{aligned} X_1X_2 &= M_1M_2 \cdot 2^{E_1+E_2} \\ &= (0.011110) \cdot 2^{010} \\ &= (0.111100) \cdot 2^{001} \end{aligned}$$

On the other hand, the addition of the two floating-point numbers requires that the exponents be equal. This can be accomplished by shifting the mantissa of the smaller number to the right and compensating by increasing the corresponding exponent. Thus the number X_2 can be expressed as

$$M_2 = 0.000011$$

$$E_2 = 011$$

With $E_2 = E_1$, we can add the two numbers X_1 and X_2. The result is

$$X_1 + X_2 = (0.101011) \cdot 2^{011}$$

It should be observed that the shifting operation required to equalize the exponent of X_2 with that for X_1 results in loss of precision, in general. In this example the six-bit mantissa was sufficiently long to accommodate a shift of four bits to the right for M_2 without dropping any of the ones. However, a shift of five bits would have caused the loss of a single bit and a shift of six bits to the right would have resulted in a mantissa of $M_2 = 0.000000$, unless we round upward after shifting so that $M_2 = 0.000001$.

Overflow occurs in the multiplication of two floating-point numbers when the sum of the exponents exceeds the dynamic range of the fixed-point representation of the exponent.

In comparing a fixed-point representation with a floating-point representation, each with the same number of total bits, it is apparent that the floating-point representation allows us to cover a larger dynamic range by varying the resolution across the range. The resolution decreases with an increase in the size of successive numbers. In other words, the distance between two successive floating-point numbers increases as the numbers increase in size. It is this variable resolution that results in a larger dynamic range. Alternatively, if we wish to cover the same dynamic range with both fixed-point and floating-point representations, the floating-point representation provides finer resolution for small numbers but coarser resolution for the larger

numbers. In contrast, the fixed-point representation provides a uniform resolution throughout the range of numbers.

For example, if we have a computer with a word size of 32 bits, it is possible to represent 2^{32} numbers. If we wish to represent the positive integers beginning with zero, the largest possible integer that can be accommodated is

$$2^{32} - 1 = 4{,}294{,}967{,}295$$

The distance between successive numbers (the resolution) is 1. Alternatively, we can designate the leftmost bit as the sign bit and use the remaining 31 bits for the magnitude. In such a case a fixed-point representation allows us to cover the range

$$-(2^{31} - 1) = -2{,}147{,}483{,}647 \quad \text{to} \quad (2^{31} - 1) = 2{,}147{,}483{,}647$$

again with a resolution of 1.

On the other hand, suppose that we increase the resolution by allocating 10 bits for a fractional part, 21 bits for the integer part, and 1 bit for the sign. Then this representation allows us to cover the dynamic range

$$-(2^{31} - 1) \cdot 2^{-10} = -(2^{21} - 2^{-10}) \quad \text{to} \quad (2^{31} - 1) \cdot 2^{-10} = 2^{21} - 2^{-10}$$

or, equivalently,

$$-2{,}097{,}151.999 \quad \text{to} \quad 2{,}097{,}151.999$$

In this case, the resolution is 2^{-10}. Thus, the dynamic range has been decreased by a factor of approximately 1000 (actually 2^{10}), while the resolution has been increased by the same factor.

For comparison, suppose that the 32-bit word is used to represent floating- point numbers. In particular, let the mantissa be represented by 23 bits plus a sign bit and let the exponent be represented by 7 bits plus a sign bit. Now, the smallest number in magnitude will have the representation,

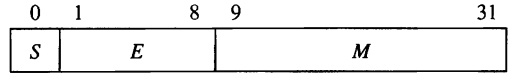
sign	23 bits	sign	7 bits	
0.	$100\cdots0$	1	1111111	$= \frac{1}{2} \times 2^{-127} \approx 0.3 \times 10^{-38}$

At the other extreme, the largest number that can be represented with this floating-point representation is

sign	23 bits	sign	7 bits	
0	$111\cdots1$	0	1111111	$= (1 - 2^{-23}) \times 2^{127} \approx 1.7 \times 10^{38}$

Thus, we have achieved a dynamic range of approximately 10^{76}, but with varying resolution. In particular, we have fine resolution for small numbers and coarse resolution for larger numbers.

The representation of zero poses some special problems. In general, only the mantissa has to be zero, but not the exponent. The choice of M and E, the representation of zero, the handling of overflows, and other related issues have resulted in various floating-point representations on different digital computers. In an effort to define a common floating-point format, the Institute of Electrical and Electronic Engineers (IEEE) introduced the IEEE 754 standard, which is widely used in practice. For a 32-bit machine, the IEEE 754 standard single-precision, floating-point number is represented as $X = (-1)^s \cdot 2^{E-127}(M)$, where



This number has the following interpretations:

If $E = 255$ and $M \neq 0$, then X is not a number

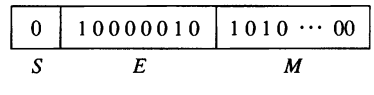
If $E = 255$ and $M = 0$, then $X = (-1)^S \cdot \infty$

If $0 < E < 255$, then $X = (-1)^S \cdot 2^{E-127}(1.M)$

If $E = 0$ and $M \neq 0$, then $X = (-1)^S \cdot 2^{-126}(0.M)$

If $E = 0$ and $M = 0$, then $X = (-1)^S \cdot 0$

where $0.M$ is a fraction and $1.M$ is a mixed number with one integer bit and 23 fractional bits. For example, the number



has the value $X = -1^0 \times 2^{130-127} \times 1.1010\ldots0 = 2^3 \times \frac{13}{8} = 13$. The magnitude range of the 32-bit IEEE 754 floating-point numbers is from $2^{-126} \times 2^{-23}$ to $(2-2^{-23}) \times 2^{127}$ (i.e., from 1.18×10^{-38} to 3.40×10^{38}). Computations with numbers outside this range result in either underflow or overflow.

9.4.3 Errors Resulting from Rounding and Truncation

In performing computations such as multiplications with either fixed-point or floating-point arithmetic, we are usually faced with the problem of quantizing a number via truncation or rounding, from a given level of precision to a level of lower precision. The effect of rounding and truncation is to introduce an error whose value depends on the number of bits in the original number relative to the number of bits after quantization. The characteristics of the errors introduced through either truncation or rounding depend on the particular form of number representation.

To be specific, let us consider a fixed-point representation in which a number x is quantized from b_u bits to b bits. Thus the number

$$x = \overbrace{0.1011\cdots01}^{b_u}$$

consisting of b_u bits prior to quantization is represented as

$$x = \overbrace{0.101\cdots1}^{b}$$

after quantization, where $b < b_u$. For example, if x represents the sample of an analog signal, then b_u may be taken as infinite. In any case if the quantizer truncates the value of x, the truncation error is defined as

$$E_t = Q_t(x) - x \tag{9.4.10}$$

First, we consider the range of values of the error for sign-magnitude and two's-complement representation. In both of these representations, the positive numbers have identical representations. For positive numbers, truncation results in a number that is smaller than the unquantized number. Consequently, the truncation error resulting from a reduction of the number of significant bits from b_u to b is

$$-(2^{-b} - 2^{-b_u}) \leq E_t \leq 0 \tag{9.4.11}$$

where the largest error arises from discarding $b_u - b$ bits, all of which are ones.

In the case of negative fixed-point numbers based on the sign-magnitude representation, the truncation error is positive, since truncation basically reduces the magnitude of the numbers. Consequently, for negative numbers, we have

$$0 \leq E_t \leq (2^{-b} - 2^{-b_u}) \tag{9.4.12}$$

In the two's-complement representation, the negative of a number is obtained by subtracting the corresponding positive number from 2. As a consequence, the effect of truncation on a negative number is to increase the magnitude of the negative number. Consequently, $x > Q_t(x)$ and hence

$$-(2^{-b} - 2^{-b_u}) \leq E_t \leq 0 \tag{9.4.13}$$

Hence we conclude that *the truncation error for the sign-magnitude representation is symmetric about zero and falls in the range*

$$-(2^{-b} - 2^{-b_u}) \leq E_t \leq (2^{-b} - 2^{-b_u}) \tag{9.4.14}$$

On the other hand, *for two's-complement representation, the truncation error is always negative and falls in the range*

$$-(2^{-b} - 2^{-b_u}) \leq E_t \leq 0 \tag{9.4.15}$$

Next, let us consider the quantization errors due to rounding of a number. A number x, represented by b_u bits before quantization and b bits after quantization, incurs a quantization error

$$E_r = Q_r(x) - x \tag{9.4.16}$$

Basically, rounding involves only the magnitude of the number and, consequently, the round-off error is independent of the type of fixed-point representation. The maximum error that can be introduced through rounding is $(2^{-b} - 2^{-bu})/2$ and this can be either positive or negative, depending on the value of x. Therefore, *the round-off error is symmetric about zero and falls in the range*

$$-\tfrac{1}{2}(2^{-b} - 2^{-bu}) \leq E_r \leq \tfrac{1}{2}(2^{-b} - 2^{-bu}) \tag{9.4.17}$$

These relationships are summarized in Fig. 9.4.2 when x is a continuous signal amplitude ($b_u = \infty$).

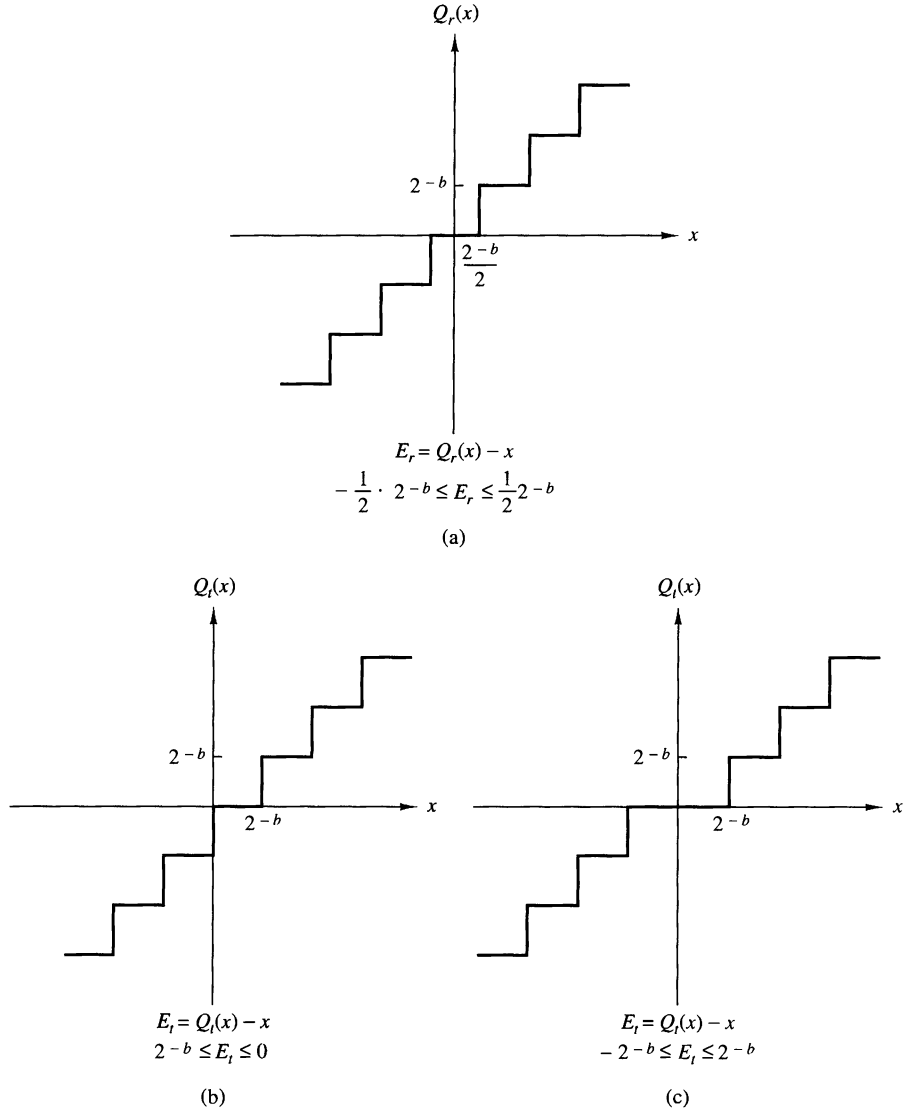


Figure 9.4.2 Quantization errors in rounding and truncation: (a) rounding; (b) truncation in two's complement; (c) truncation in sign-magnitude.

In a floating-point representation, the mantissa is either rounded or truncated. Due to the nonuniform resolution, the corresponding error in a floating-point representation is proportional to the number being quantized. An appropriate representation for the quantized value is

$$Q(x) = x + ex \tag{9.4.18}$$

where e is called the relative error. Now

$$Q(x) - x = ex \tag{9.4.19}$$

In the case of truncation based on two's-complement representation of the mantissa, we have

$$-2^E 2^{-b} < e_t x < 0 \tag{9.4.20}$$

for positive numbers. Since $2^{E-1} \leq x < 2^E$, it follows that

$$-2^{-b+1} < e_t \leq 0, \qquad x > 0 \tag{9.4.21}$$

On the other hand, for a negative number in two's-complement representation, the error is

$$0 \leq e_t x < 2^E 2^{-b}$$

and hence

$$0 \leq e_t < 2^{-b+1}, \qquad x < 0 \tag{9.4.22}$$

In the case where the mantissa is rounded, the resulting error is symmetric relative to zero and has a maximum value of $\pm 2^{-b}/2$. Consequently, the round-off error becomes

$$-2^E \cdot 2^{-b}/2 < e_r x \leq 2^E \cdot 2^{-b}/2 \tag{9.4.23}$$

Again, since x falls in the range $2^{E-1} \leq x < 2^E$, we divide through by 2^{E-1} so that

$$-2^{-b} < e_r \leq 2^{-b} \tag{9.4.24}$$

In arithmetic computations involving quantization via truncation and rounding, it is convenient to adopt a statistical approach to the characterization of such errors. The quantizer can be modeled as introducing an additive noise to the unquantized value x. Thus we can write

$$Q(x) = x + \epsilon$$

where $\epsilon = E_r$ for rounding and $\epsilon = E_t$ for truncation. This model is illustrated in Fig. 9.4.3.

Since x can be any number that falls within any of the levels of the quantizer, the quantization error is usually modeled as a random variable that falls within the limits specified. This random variable is assumed to be uniformly distributed within the ranges specified for the fixed-point representations. Furthermore, in practice, $b_u >> b$, so that we can neglect the factor of 2^{-b_u} in the formulas given below. Under these conditions, the probability density functions for the round-off and truncation errors in the two fixed-point representations are illustrated in Fig. 9.4.4. We note that

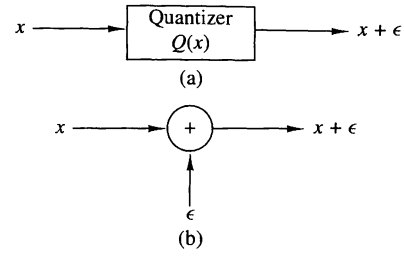


Figure 9.4.3
Additive noise model for the nonlinear quantization process: (a) actual system; (b) model for quantization.

in the case of truncation of the two's-complement representation of the number, the average value of the error has a bias of $2^{-b}/2$, whereas in all other cases just illustrated, the error has an average value of zero.

We shall use this statistical characterization of the quantization errors in our treatment of such errors in digital filtering and in the computation of the DFT for fixed-point implementation.

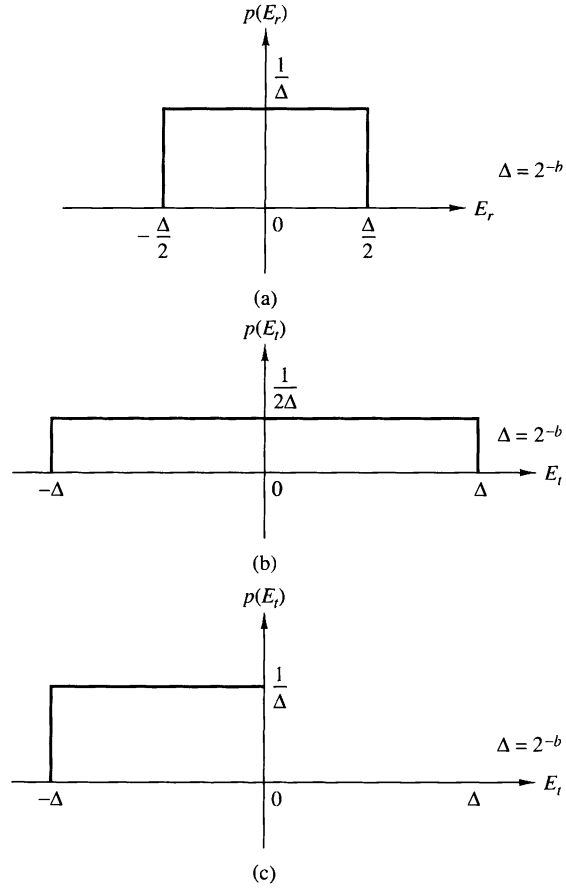


Figure 9.4.4
Statistical characterization of quantization errors: (a) round-off error; (b) truncation error for sign-magnitude; (c) truncation error for two's complement.

9.5 Quantization of Filter Coefficients

In the realization of FIR and IIR filters in hardware or in software on a general-purpose computer, the accuracy with which filter coefficients can be specified is limited by the word length of the computer or the length of the register provided to store the coefficients. Since the coefficients used in implementing a given filter are not exact, the poles and zeros of the system function will, in general, be different from the desired poles and zeros. Consequently, we obtain a filter having a frequency response that is different from the frequency response of the filter with unquantized coefficients.

In Section 9.5.1, we demonstrate that the sensitivity of the filter frequency response characteristics to quantization of the filter coefficients is minimized by realizing a filter having a large number of poles and zeros as an interconnection of second-order filter sections. This leads us to the parallel-form and cascade-form realizations in which the basic building blocks are second-order filter sections.

9.5.1 Analysis of Sensitivity to Quantization of Filter Coefficients

To illustrate the effect of quantization of the filter coefficients in a direct-form realization of an IIR filter, let us consider a general IIR filter with system function

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \tag{9.5.1}$$

The direct-form realization of the IIR filter with quantized coefficients has the system function

$$\overline{H}(z) = \frac{\sum_{k=0}^{M} \overline{b}_k z^{-k}}{1 + \sum_{k=1}^{N} \overline{a}_k z^{-k}} \tag{9.5.2}$$

where the quantized coefficients $\{\overline{b}_k\}$and $\{\overline{a}_k\}$ can be related to the unquantized coefficients $\{b_k\}$ and $\{a_k\}$ by the relations

$$\begin{aligned} \overline{a}_k &= a_k + \Delta a_k, \qquad k = 1, 2, \ldots, N \\ \overline{b}_k &= b_k + \Delta b_k, \qquad k = 0, 1, \ldots, M \end{aligned} \tag{9.5.3}$$

and $\{\Delta a_k\}$ and $\{\Delta b_k\}$ represent the quantization errors.

The denominator of $H(z)$ may be expressed in the form

$$D(z) = 1 + \sum_{k=0}^{N} a_k z^{-k} = \prod_{k=1}^{N} (1 - p_k z^{-1}) \tag{9.5.4}$$

where $\{p_k\}$ are the poles of $H(z)$. Similarly, we can express the denominator of $\overline{H}(z)$ as

$$\overline{D}(z) = \prod_{k=1}^{N}(1 - \overline{p}_k z^{-1}) \tag{9.5.5}$$

where $\overline{p}_k = p_k + \Delta p_k$, $k = 1, 2, \ldots, N$, and Δp_k is the error or perturbation resulting from the quantization of the filter coefficients.

We shall now relate the perturbation Δp_k to the quantization errors in the $\{a_k\}$. The perturbation error Δp_i can be expressed as

$$\Delta p_i = \sum_{k=1}^{N} \frac{\partial p_i}{\partial a_k} \Delta a_k \tag{9.5.6}$$

where $\partial p_i / \partial a_k$, the partial derivative of p_i with respect to a_k, represents the incremental change in the pole p_i due to a change in the coefficient a_k. Thus the total error Δp_i is expressed as a sum of the incremental errors due to changes in each of the coefficients $\{a_k\}$.

The partial derivatives $\partial p_i / \partial a_k$, $k = 1, 2, \ldots, N$, can be obtained by differentiating $D(z)$ with respect to each of the $\{a_k\}$. First we have

$$\left(\frac{\partial D(z)}{\partial a_k}\right)_{z=p_i} = \left(\frac{\partial D(z)}{\partial z}\right)_{z=p_i} \left(\frac{\partial p_i}{\partial a_k}\right) \tag{9.5.7}$$

Then

$$\frac{\partial p_i}{\partial a_k} = \frac{(\partial D(z)/\partial a_k)_{z=p_i}}{(\partial D(z)/\partial z)_{z=p_i}} \tag{9.5.8}$$

The numerator of (9.5.8) is

$$\left(\frac{\partial D(z)}{\partial a_k}\right)_{z=p_i} = -z^{-k}|_{z=p_i} = -p_i^{-k} \tag{9.5.9}$$

The denominator of (9.5.8) is

$$\begin{aligned}\left(\frac{\partial D(z)}{\partial z}\right)_{z=p_i} &= \left\{\frac{\partial}{\partial z}\left[\prod_{l=1}^{N}(1 - p_l z^{-1})\right]\right\}_{z=pi} \\ &= \left\{\sum_{k=1}^{N} \frac{p_k}{z^2} \prod_{\substack{l=1 \\ l\neq i}}^{N}(1 - p_l z^{-1})\right\}_{z=p_i} \\ &= \frac{1}{p_i^N} \prod_{\substack{l=1 \\ l\neq i}}^{N}(p_i - p_l)\end{aligned} \tag{9.5.10}$$

Therefore, (9.5.8) can be expressed as

$$\frac{\partial p_i}{\partial a_k} = \frac{-p_i^{N-k}}{\prod_{\substack{l=1 \\ l \neq i}}^{N} (p_i - p_l)} \tag{9.5.11}$$

Substitution of the result in (9.5.11) into (9.5.6) yields the total perturbation error Δp_i in the form

$$\Delta p_i = -\sum_{k=1}^{N} \frac{p_i^{N-k}}{\prod_{\substack{l=1 \\ l \neq i}}^{N} (p_i - p_l)} \Delta a_k \tag{9.5.12}$$

This expression provides a measure of the sensitivity of the ith pole to changes in the coefficients $\{a_k\}$. An analogous result can be obtained for the sensitivity of the zeros to errors in the parameters $\{b_k\}$.

The terms $(p_i - p_l)$ in the denominator of (9.5.12) represent vectors in the z-plane from the poles $\{p_l\}$ to the pole p_i. If the poles are tightly clustered as they are in a narrowband filter, as illustrated in Fig. 9.5.1, the lengths $|p_i - p_l|$ are small for the poles in the vicinity of p_i. These small lengths will contribute to large errors and hence a large perturbation error Δp_i results.

The error Δp_i can be minimized by maximizing the lengths $|p_i - p_l|$. This can be accomplished by realizing the high-order filter with either single-pole or double-pole filter sections. In general, however, single-pole (and single-zero) filter sections have complex-valued poles and require complex-valued arithmetic operations for their realization. This problem can be avoided by combining complex-valued poles (and zeros) to form second-order filter sections. Since the complex-valued poles are

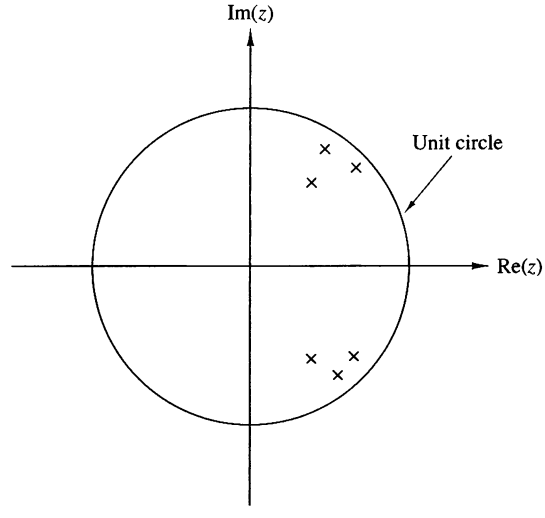


Figure 9.5.1
Pole positions for a bandpass IIR filter.

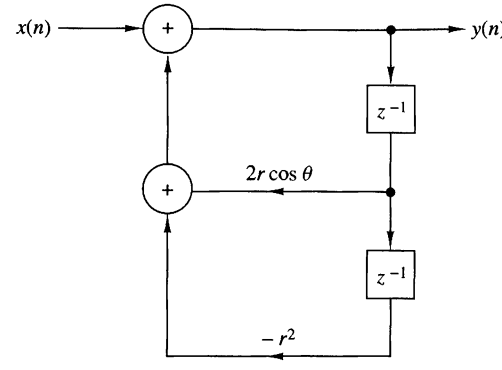


Figure 9.5.2
Realization of a two-pole IIR filter.

usually sufficiently far apart, the perturbation errors $\{\Delta p_i\}$ are minimized. As a consequence, the resulting filter with quantized coefficients more closely approximates the frequency response characteristics of the filter with unquantized coefficients.

It is interesting to note that even in the case of a two-pole filter section, the structure used to realize the filter section plays an important role in the errors caused by coefficient quantization. To be specific, let us consider a two-pole filter with system function

$$H(z) = \frac{1}{1 - (2r\cos\theta)z^{-1} + r^2 z^{-2}} \tag{9.5.13}$$

This filter has poles at $z = re^{\pm j\theta}$. When realized as shown in Fig. 9.5.2, it has two coefficients, $a_1 = 2r\cos\theta$ and $a_2 = -r^2$. With infinite precision it is possible to achieve an infinite number of pole positions. Clearly, with finite precision (i.e., quantized coefficients a_1 and a_2), the possible pole positions are also finite. In fact, when b bits are used to represent the magnitudes of a_1 and a_2, there are at most $(2^b - 1)^2$ possible positions for the poles in each quandrant, excluding the case $a_1 = 0$ and $a_2 = 0$.

For example, suppose that $b = 4$. Then there are 15 possible nonzero values for a_1. There are also 15 possible values for r^2. We illustrate these possible values in Fig. 9.5.3 for the first quandrant of the z-plane only. There are 169 possible pole positions in this case. The nonuniformity in their positions is due to the fact that we are quantizing r^2, whereas the pole positions lie on a circular arc of radius r. Of particular significance is the sparse set of poles for values of θ near zero and, due to symmetry, near $\theta = \pi$. This situation would be highly unfavorable for lowpass filters and highpass filters which normally have poles clustered near $\theta = 0$ and $\theta = \pi$.

An alternative realization of the two-pole filter is the coupled-form realization illustrated in Fig. 9.5.4. The two coupled equations are

$$\begin{aligned} y_1(n) &= x(n) + r\cos\theta\ y_1(n-1) - r\sin\theta\ y(n-1) \\ y(n) &= r\sin\theta\ y_1(n-1) + r\cos\theta\ y(n-1) \end{aligned} \tag{9.5.14}$$

By transforming these two equations into the z-domain, it is a simple matter to show that

$$\frac{Y(z)}{X(z)} = H(z) = \frac{(r\sin\theta)z^{-1}}{1 - (2r\cos\theta)z^{-1} + r^2 z^{-2}} \tag{9.5.15}$$

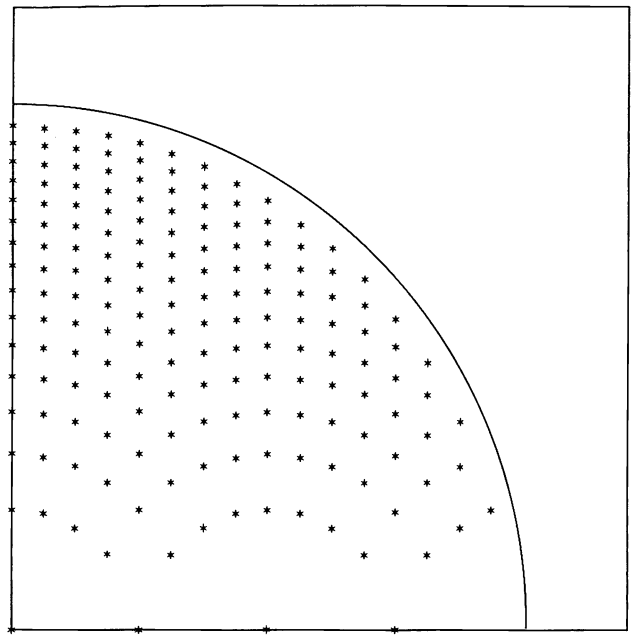


Figure 9.5.3
Possible pole positions for two-pole IIR filter realization in Fig. 9.5.2.

In the coupled form we observe that there are also two coefficients, $\alpha_1 = r \sin\theta$ and $\alpha_2 = r \cos\theta$. Since they are both linear in r, the possible pole positions are now equally spaced points on a rectangular grid, as shown in Fig. 9.5.5. As a consequence, the pole positions are now uniformly distributed inside the unit circle, which is a more desirable situation than the previous realization, especially for lowpass filters. (There are 198 possible pole positions in this case.) However, the price that we pay for this uniform distribution of pole positions is an increase in computations. The coupled-form realization requires four multiplications per output point, whereas the realization in Fig. 9.5.2 requires only two multiplications per output point.

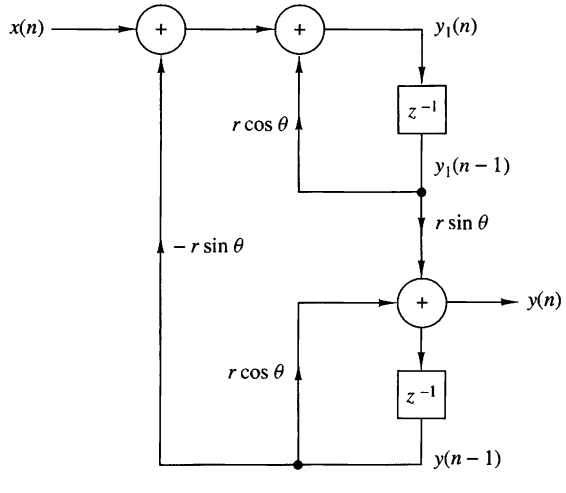


Figure 9.5.4
Coupled-form realization of a two-pole IIR filter.

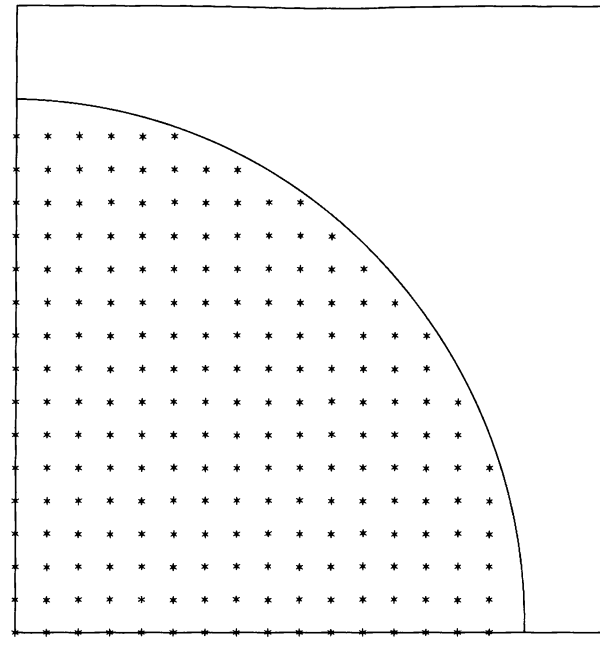


Figure 9.5.5
Possible pole positions for the coupled-form two-pole filter in Fig. 9.5.4.

Since there are various ways in which one can realize a second-order filter section, there are obviously many possibilities for different pole locations with quantized coefficients. Ideally, we should select a structure that provides us with a dense set of points in the regions where the poles lie. Unfortunately, however, there is no simple and systematic method for determining the filter realization that yields this desired result.

Given that a higher-order IIR filter should be implemented as a combination of second-order sections, we still must decide whether to employ a parallel configuration or a cascade configuration. In other words, we must decide between the realization

$$H(z) = \prod_{k=1}^{K} \frac{b_{k0} + b_{k1}z^{-1} + b_{k2}z^{-2}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}} \tag{9.5.16}$$

and the realization

$$H(z) = \sum_{k=1}^{K} \frac{c_{k0} + c_{k1}z^{-1}}{1 + a_{k1}z^{-1} + a_{k2}z^{-2}} \tag{9.5.17}$$

If the IIR filter has zeros on the unit circle, as is generally the case with elliptic and Chebyshev type II filters, each second-order section in the cascade configuration of (9.5.16) contains a pair of complex-conjugate zeros. The coefficients $\{b_k\}$ directly determine the location of these zeros. If the $\{b_k\}$ are quantized, the sensitivity of the system response to the quantization errors is easily and directly controlled by allocating a sufficiently large number of bits to the representation of the $\{b_{ki}\}$. In fact, we can easily evaluate the perturbation effect resulting from quantizing the

coefficients $\{b_{ki}\}$ to some specified precision. Thus we have direct control of both the poles and the zeros that result from the quantization process.

On the other hand, the parallel realization of $H(z)$ provides direct control of the poles of the system only. The numerator coefficients $\{c_{k0}\}$ and $\{c_{k1}\}$ do not specify the location of the zeros directly. In fact, the $\{c_{k0}\}$ and $\{c_{k1}\}$ are obtained by performing a partial-fraction expansion of $H(z)$. Hence they do not directly influence the location of the zeros, but only indirectly through a combination of all the factors of $H(z)$. As a consequence, it is more difficult to determine the effect of quantization errors in the coefficients $\{c_{ki}\}$ on the location of the zeros of the system.

It is apparent that quantization of the parameters $\{c_{ki}\}$ is likely to produce a significant perturbation of the zero positions and usually, it is sufficiently large in fixed-point implementations to move the zeros off the unit circle. This is a highly undesirable situation, which can be easily remedied by use of a floating-point representation. In any case the cascade form is more robust in the presence of coefficient quantization and should be the preferred choice in practical applications, especially where a fixed-point representation is employed.

EXAMPLE 9.5.1

Determine the effect of parameter quantization on the frequency response of the seventh-order elliptic filter given in Table 10.6 when it is realized as a cascade of second-order sections.

Solution. The coefficients for the elliptic filter given in Table 10.6 are specified for the cascade form to six significant digits. We quantized these coefficients to four and then three significant digits (by rounding) and plotted the magnitude (in decibels) and the phase of

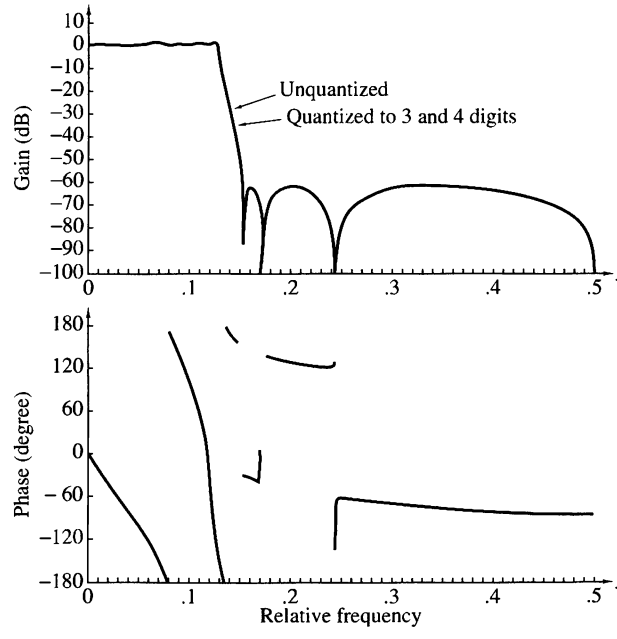


Figure 9.5.6
Effect of coefficient quantization of the magnitude and phase response of an $N = 7$ elliptic filter realized in cascade form.

the frequency response. The results are shown in Fig. 9.5.6 along the frequency response of the filter with unquantized (six significant digits) coefficients. We observe that there is an insignificant degradation due to coefficient quantization for the cascade realization.

EXAMPLE 9.5.2

Repeat the computation of the frequency response for the elliptic filter considered in Example 9.5.1 when it is realized in the parallel form with second-order sections.

Solution. The system function for the 7-order elliptic filter given in Table 10.6 is

$$H(z) = \frac{0.2781304 + 0.0054373108z^{-1}}{1 - 0.790103z^{-1}} + \frac{-0.3867805 + 0.3322229z^{-1}}{1 - 1.517223z^{-1} + 0.714088z^{-2}} + \frac{0.1277036 - 0.1558696z^{-1}}{1 - 1.421773z^{-1} + 0.861895z^{-2}} + \frac{-0.015824186 + 0.38377356z^{-1}}{1 - 1.387447z^{-1} + 0.962242z^{-2}}$$

The frequency response of this filter with coefficients quantized to four digits is shown in Fig. 9.5.7(a). When this result is compared with the frequency response in Fig. 9.5.6, we observe that the zeros in the parallel realization have been perturbed sufficiently so that the nulls in the magnitude response are now at -80, -85, and -92 dB. The phase response has also been perturbed by a small amount.

When the coefficients are quantized to three significant digits, the frequency response characteristic deteriorates significantly, in both magnitude and phase, as illustrated in Fig. 9.5.7(b). It is apparent from the magnitude response that the zeros are no longer on the unit circle as a result of the quantization of the coefficients. This result clearly illustrates the sensitivity of the zeros to quantization of the coefficients in the parallel form.

When compared with the results of Example 9.5.1, it is also apparent that the cascade form is definitely more robust to parameter quantization than the parallel form.

9.5.2 Quantization of Coefficients in FIR Filters

As indicated in the preceding section, the sensitivity analysis performed on the poles of a system also applies directly to the zeros of the IIR filters. Consequently, an expression analogous to (9.5.12) can be obtained for the zeros of an FIR filter. In effect, we should generally realize FIR filters with a large number of zeros as a cascade of second-order and first-order filter sections to minimize the sensitivity to coefficient quantization.

Of particular interest in practice is the realization of linear-phase FIR filters. The direct-form realizations shown in Figs. 9.2.1 and 9.2.2 maintain the linear-phase property even when the coefficients are quantized. This follows easily from the observation that the system function of a linear-phase FIR filter satisfies the property

$$H(z) = \pm z^{-(M-1)} H(z^{-1})$$

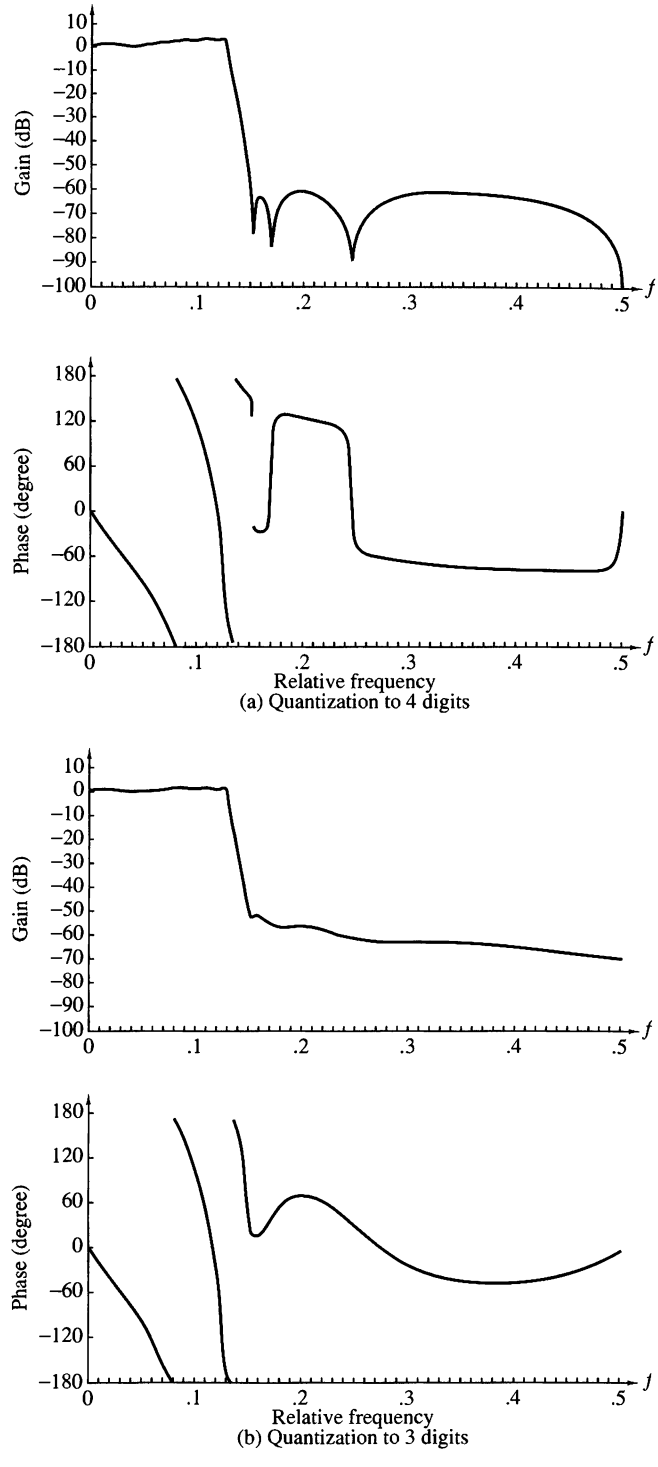


Figure 9.5.7
Effect of coefficient quantization of the magnitude and phase response of an $N = 7$ elliptic filter realized in parallel form: (a) quantization to four digits; (b) quantization to three digits.

independent of whether the coefficients are quantized or unquantized (see Section 10.2). Consequently, coefficient quantization does not affect the phase characteristic of the FIR filter, but affects only the magnitude. As a result, coefficient quantization effects are not as severe on a linear-phase FIR filter, since the only effect is in the magnitude.

EXAMPLE 9.5.3

Determine the effect of parameter quantization on the frequency response of an $M = 32$ linear-phase FIR bandpass filter. The filter is realized in the direct form.

Solution. The frequency response of a linear-phase FIR bandpass filter with unquantized coefficients is illustrated in Fig. 9.5.8(a). When the coefficients are quantized to four significant digits, the effect on the frequency response is insignificant. However, when the coefficients are quantized to three significant digits, the sidelobes increase by several decibels, as illustrated in Fig. 9.5.8(b). This result indicates that we should use a minimum of 10 bits to represent the coefficients of this FIR filter and, preferably, 12 to 14 bits, if possible.

From this example we learn that a minimum of 10 bits is required to represent the coefficients in a direct-form realization of an FIR filter of moderate length. As the filter length increases, the number of bits per coefficient must be increased to maintain the same error in the frequency response characteristic of the filter.

For example, suppose that each filter coefficient is rounded to $(b+1)$ bits. Then the maximum error in a coefficient value is bounded as

$$-2^{-(b+1)} < e_h(n) < 2^{-(b+1)}$$

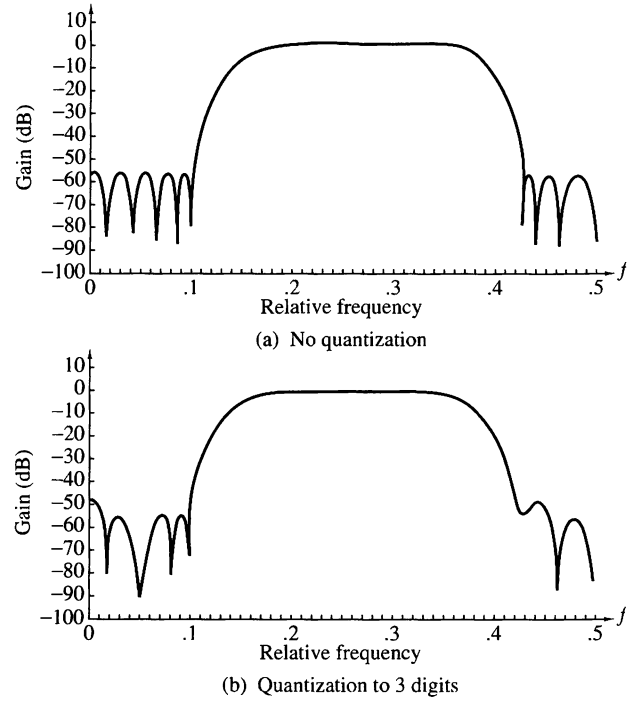


Figure 9.5.8
Effect of coefficient quantization of the magnitude of an $M = 32$ linear-phase FIR filter realized in direct form: (a) no quantization; (b) quantization to three digits.

Since the quantized values may be represented as $\overline{h}(n) = h(n) + e_h(n)$, the error in the frequency response is

$$E_M(\omega) = \sum_{n=0}^{M-1} e_h(n)e^{-j\omega n}$$

Since $e_h(n)$ is zero mean, it follows that $E_M(\omega)$ is also zero mean. Assuming that the coefficient error sequence $e_h(n)$, $0 \le n \le M-1$, is uncorrelated, the variance of the error $E_M(\omega)$ in the frequency response is just the sum of the variances of the M terms. Thus we have

$$\sigma_E^2 = \frac{2^{-2(b+1)}}{12}M = \frac{2^{-2(b+2)}}{3}M$$

Here we note that the variance of the error in $H(\omega)$ increases linearly with M. Hence the standard deviation of the error in $H(\omega)$ is

$$\sigma_E = \frac{2^{-(b+2)}}{\sqrt{3}}\sqrt{M}$$

Consequently, for every factor-of-4 increase in M, the precision in the filter coefficients must be increased by one additional bit to maintain the standard deviation fixed. This result, taken together with the results of Example 9.5.3, implies that the frequency error remains tolerable for filter lengths up to 256, provided that filter coefficients are represented by 12 to 13 bits. If the word length of the digital signal processor is less than 12 bits or if the filter length exceeds 256, the filter should be implemented as a cascade of smaller length filters to reduce the precision requirements.

In a cascade realization of the form

$$H(z) = G\prod_{k=1}^{K} H_k(z) \tag{9.5.18}$$

where the second-order sections are given as

$$H_k(z) = 1 + b_{k1}z^{-1} + b_{k2}z^{-2} \tag{9.5.19}$$

the coefficients of complex-valued zeros are expressed as $b_{k1} = -2r_k\cos\theta_k$ and $b_{k2} = r_k^2$. Quantization of b_{k1} and b_{k2} results in zero locations as shown in Fig. 9.5.3, except that the grid extends to points outside the unit circle.

A problem may arise, in this case, in maintaining the linear-phase property, because the quantized pair of zeros at $z = (1/r_k)e^{\pm j\theta_k}$ may not be the mirror image of the quantized zeros at $z = r_k e^{\pm j\theta_k}$. This problem can be avoided by rearranging the factors corresponding to the mirror-image zero. That is, we can write the mirror-image factor as

$$\left(1 - \frac{2}{r_k}\cos\theta_k z^{-1} + \frac{1}{r_k^2}z^{-2}\right) = \frac{1}{r_k^2}(r_k^2 - 2r_k\cos\theta_k z^{-1} + z^{-2}) \tag{9.5.20}$$

The factors $\{1/r_k^2\}$ can be combined with the overall gain factor G, or they can be distributed in each of the second-order filters. The factor in (9.5.20) contains exactly the same parameters as the factor $(1 - 2r_k \cos\theta_k z^{-1} + r_k^2 z^{-2})$, and consequently, the zeros now occur in mirror-image pairs even when the parameters are quantized.

In this brief treatment we have given the reader an introduction to the problems of coefficient quantization in IIR and FIR filters. We have demonstrated that a high-order filter should be reduced to a cascade (for FIR or IIR filters) or a parallel (for IIR filters) realization to minimize the effects of quantization errors in the coefficients. This is especially important in fixed-point realizations in which the coefficients are represented by a relatively small number of bits.

9.6 Round-Off Effects in Digital Filters

In Section 9.4 we characterized the quantization errors that occur in arithmetic operations performed in a digital filter. The presence of one or more quantizers in the realization of a digital filter results in a nonlinear device with characteristics that may be significantly different from the ideal linear filter. For example, a recursive digital filter may exhibit undesirable oscillations in its output, as shown in the following section, even in the absence of an input signal.

As a result of the finite-precision arithmetic operations performed in the digital filter, some registers may overflow if the input signal level becomes large. Overflow represents another form of undesirable nonlinear distortion on the desired signal at the output of the filter. Consequently, special care must be exercised to scale the input signal properly, either to prevent overflow completely or, at least, to minimize its rate of occurrence.

The nonlinear effects due to finite-precision arithmetic make it extremely difficult to precisely analyze the performance of a digital filter. To perform an analysis of quantization effects, we adopt a statistical characterization of quantization errors which, in effect, results in a linear model for the filter. Thus we are able to quantify the effects of quantization errors in the implementation of digital filters. Our treatment is limited to fixed-point realizations where quantization effects are very important.

9.6.1 Limit-Cycle Oscillations in Recursive Systems

In the realization of a digital filter, either in digital hardware or in software on a digital computer, the quantization inherent in the finite-precision arithmetic operations renders the system nonlinear. In recursive systems, the nonlinearities due to the finite-precision arithmetic operations often cause periodic oscillations to occur in the output, even when the input sequence is zero or some nonzero constant value. Such oscillations in recursive systems are called *limit cycles* and are directly attributable to round-off errors in multiplication and overflow errors in addition.

To illustrate the characteristics of a limit-cycle oscillation, let us consider a single-pole system described by the linear difference equation

$$y(n) = ay(n-1) + x(n) \tag{9.6.1}$$

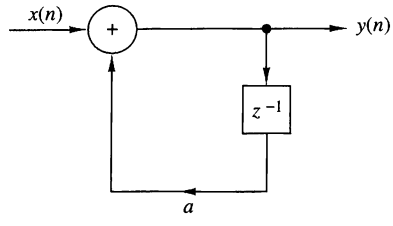


Figure 9.6.1
Ideal single-pole recursive system.

where the pole is at $z = a$. The ideal system is realized as shown in Fig. 9.6.1. On the other hand, the actual system, which is described by the nonlinear difference equation

$$v(n) = Q[av(n-1)] + x(n) \tag{9.6.2}$$

is realized as shown in Fig. 9.6.2.

Suppose that the actual system in Fig. 9.6.2 is implemented with fixed-point arithmetic based on four bits for the magnitude plus a sign bit. The quantization that takes place after multiplication is assumed to round the resulting product upward.

In Table 9.2 we list the response of the actual system for four different locations of the pole $z = a$, and an input $x(n) = \beta\delta(n)$, where $\beta = 15/16$, which has the binary representation 0.1111. Ideally, the response of the system should decay toward zero exponentially [i.e., $y(n) = a^n \to 0$ as $n \to \infty$]. In the actual system, however, the response $v(n)$ reaches a steady-state periodic output sequence with a period that depends on the value of the pole. When the pole is positive, the oscillations occur with a period $N_p = 1$, so that the output reaches a constant value of $\frac{1}{16}$ for $a = \frac{1}{2}$ and $\frac{1}{8}$ for $a = \frac{3}{4}$. On the other hand, when the pole is negative, the output sequence oscillates between positive and negative values ($\pm\frac{1}{16}$ for $a = -\frac{1}{2}$ and $\pm\frac{1}{8}$ for $a = -\frac{3}{4}$). Hence the period is $N_p = 2$.

These limit cycles occur as a result of the quantization effects in multiplications. When the input sequence $x(n)$ to the filter becomes zero, the output of the filter then, after a number of iterations, enters into the limit cycle. The output remains in the limit cycle until another input of sufficient size is applied that drives the system out of the limit cycle. Similarly, zero-input limit cycles occur from nonzero initial conditions with the input $x(n) = 0$. The amplitudes of the output during a limit cycle are confined to a range of values that is called the *dead band* of the filter.

It is interesting to note that when the response of the single-pole filter is in the limit cycle, the actual nonlinear system operates as an equivalent linear system with

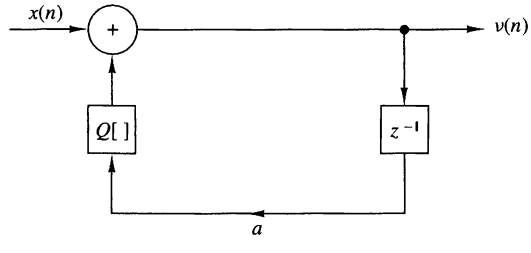


Figure 9.6.2
Actual nonlinear system.

TABLE 9.2 Limit Cycles for Lowpass Single-Pole Filter

n	$a = 0.1000 = \frac{1}{2}$		$a = 1.1000 = -\frac{1}{2}$		$a = 0.1100 = \frac{3}{4}$		$a = 1.1100 = -\frac{3}{4}$	
0	0.1111	$(\frac{15}{16})$	0.1111	$(\frac{15}{16})$	0.1011	$(\frac{11}{16})$	0.1011	$(\frac{11}{16})$
1	0.1000	$(\frac{8}{16})$	1.1000	$(-\frac{8}{16})$	0.1000	$(\frac{8}{16})$	1.1000	$(-\frac{8}{16})$
2	0.0100	$(\frac{4}{16})$	0.0100	$(\frac{4}{16})$	0.0110	$(\frac{6}{16})$	0.0110	$(\frac{6}{16})$
3	0.0010	$(\frac{2}{16})$	1.0010	$(-\frac{2}{16})$	0.0101	$(\frac{5}{16})$	1.0101	$(-\frac{5}{16})$
4	0.0001	$(\frac{1}{16})$	0.0001	$(\frac{1}{16})$	0.0100	$(\frac{4}{16})$	0.0100	$(\frac{4}{16})$
5	0.0001	$(\frac{1}{16})$	1.0001	$(-\frac{1}{16})$	0.0011	$(\frac{3}{16})$	1.0011	$(-\frac{3}{16})$
6	0.0001	$(\frac{1}{16})$	0.0001	$(\frac{1}{16})$	0.0010	$(\frac{2}{16})$	0.0010	$(\frac{2}{16})$
7	0.0001	$(\frac{1}{16})$	1.0001	$(-\frac{1}{16})$	0.0010	$(\frac{2}{16})$	1.0010	$(-\frac{2}{16})$
8	0.0001	$(\frac{1}{16})$	0.0001	$(\frac{1}{16})$	0.0010	$(\frac{2}{16})$	0.0010	$(\frac{2}{16})$

a pole at $z = 1$ when the pole is positive and $z = -1$ when the pole is negative. That is,

$$Q_r[av(n-1)] = \begin{cases} v(n-1), & a > 0 \\ -v(n-1), & a < 0 \end{cases} \tag{9.6.3}$$

Since the quantized product $av(n-1)$ is obtained by rounding, it follows that the quantization error is bounded as

$$|Q_r[av(n-1)] - av(n-1)| \leq \tfrac{1}{2} \cdot 2^{-b} \tag{9.6.4}$$

where b is the number of bits (exclusive of sign) used in the representation of the pole a and $v(n)$. Consequently, (9.6.4) and (9.6.3) lead to

$$|v(n-1)| - |av(n-1)| \leq \tfrac{1}{2} \cdot 2^{-b}$$

and hence

$$|v(n-1)| \leq \frac{\frac{1}{2} \cdot 2^{-b}}{1 - |a|} \tag{9.6.5}$$

The expression in (9.6.5) defines the dead band for a single-pole filter. For example, when $b = 4$ and $|a| = \frac{1}{2}$, we have a dead band with a range of amplitudes $(-\frac{1}{16}, \frac{1}{16})$. When $b = 4$ and $|a| = \frac{3}{4}$, the dead band increases to $(-\frac{1}{8}, \frac{1}{8})$.

The limit-cycle behavior in a two-pole filter is much more complex and a larger variety of oscillations can occur. In this case the ideal two-pole system is described by the linear difference equation,

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + x(n) \tag{9.6.6}$$

whereas the actual system is described by the nonlinear difference equation

$$v(n) = Q_r[a_1 v(n-1)] + Q_r[a_2 v(n-2)] + x(n) \tag{9.6.7}$$

When the filter coefficients satisfy the condition $a_1^2 < -4a_2$, the poles of the system occur at

$$z = re^{\pm j\theta}$$

where $a_2 = -r^2$ and $a_1 = 2r\cos\theta$. As in the case of the single-pole filter, when the system is in a zero-input or zero-state limit cycle,

$$Q_r[a_2 v(n-2)] = -v(n-2) \tag{9.6.8}$$

In other words, the system behaves as an oscillator with complex-conjugate poles on the unit circle (i.e., $a_2 = -r^2 = -1$). Rounding the product $a_2 v(n-2)$ implies that

$$|Q_r[a_2 v(n-2)] - a_2 v(n-2)| \leq \tfrac{1}{2} \cdot 2^{-b} \tag{9.6.9}$$

Upon substitution of (9.6.8) into (9.6.9), we obtain the result

$$|v(n-2)| - |a_2 v(n-2)| \leq \tfrac{1}{2} \cdot 2^{-b}$$

or equivalently,

$$|v(n-2)| \leq \frac{\frac{1}{2} \cdot 2^{-b}}{1 - |a_2|} \tag{9.6.10}$$

The expression in (9.6.10) defines the dead band of the two-pole filter with complex-congugate poles. We observe that the dead-band limits depend only on $|a_2|$. The parameter $a_1 = 2r\cos\theta$ determines the frequency of oscillation.

Another possible limit-cycle mode with zero input, which occurs as a result of rounding the multiplications, corresponds to an equivalent second-order system with poles at $z = \pm 1$. In this case it was shown by Jackson (1969) that the two-pole filter exhibits oscillations with an amplitude that falls in the dead band bounded by $2^{-b}/(1 - |a_1| - a_2)$.

It is interesting to note that these limit cycles result from rounding the product of the filter coefficients with the previous outputs, $v(n-1)$ and $v(n-2)$. Instead of rounding, we may choose to truncate the products to b bits. With truncation, we can eliminate many, although not all, of the limit cycles as shown by Claasen et al. (1973). However, recall that truncation results in a biased error unless the sign-magnitude representation is used, in which case the truncation error is symmetric about zero. In general, this bias is undesirable in digital filter implementation.

In a parallel realization of a high-order IIR system, each second-order filter section exhibits its own limit-cycle behavior, with no interaction among the second-order filter sections. Consequently, the output is the sum of the zero-input limit cycles from the individual sections. In the case of a cascade realization for a high-order IIR system, the limit cycles are much more difficult to analyze. In particular, when the first filter section exhibits a zero-input limit cycle, the output limit cycle is filtered by the succeeding sections. If the frequency of the limit cycle falls near a resonance frequency in a succeeding filter section, the amplitude of the sequence is enhanced by the resonance characteristic. In general, we must be careful to avoid such situations.

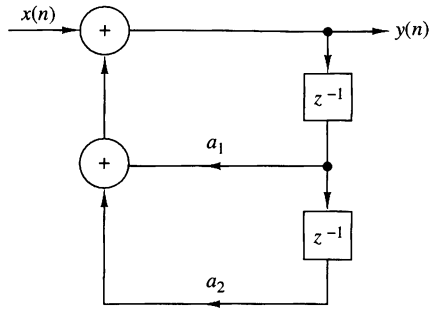


Figure 9.6.3
Two-pole filter realization.

In addition to limit cycles caused by rounding the result of multiplications, there are limit cycles caused by overflows in addition. An overflow in addition of two or more binary numbers occurs when the sum exceeds the word size available in the digital implementation of the system. For example, let us consider the second-order filter section illustrated in Fig. 9.6.3, in which the addition is performed in two's-complement arithmetic. Thus we can write the output $y(n)$ as

$$y(n) = g[a_1 y(n-1) + a_2 y(n-2) + x(n)] \tag{9.6.11}$$

where the function $g[\cdot]$ represents the two's-complement addition. It is easily verified that the function $g(v)$ versus v is described by the graph in Fig. 9.6.4.

Recall that the range of values of the parameters (a_1, a_2) for a stable filter is given by the stability triangle in Fig. 3.5.1. However, these conditions are no longer sufficient to prevent overflow oscillation with two's-complement arithmetic. In fact, it can easily be shown that a necessary and sufficient condition for ensuring that no zero-input overflow limit cycles occur is

$$|a_1| + |a_2| < 1 \tag{9.6.12}$$

which is extremely restrictive and hence an unreasonable constraint to impose on any second-order section.

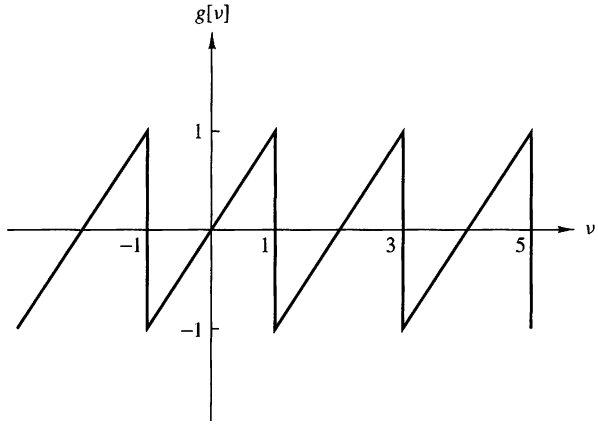


Figure 9.6.4
Characteristic functional relationship for two's-complement addition of two or more numbers.

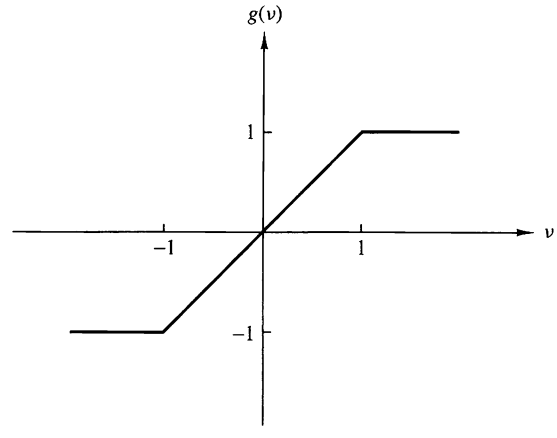


Figure 9.6.5
Characteristic functional relationship for addition with clipping at ±1.

An effective remedy for curing the problem of overflow oscillations is to modify the adder characteristic, as illustrated in Fig. 9.6.5, so that it performs saturation arithmetic. Thus when an overflow (or underflow) is sensed, the output of the adder will be the full-scale value of ±1. The distortion caused by this nonlinearity in the adder is usually small, provided that saturation occurs infrequently. The use of such a nonlinearity does not preclude the need for scaling of the signals and the system parameters, as described in the following section.

9.6.2 Scaling to Prevent Overflow

Saturation arithmetic as just described eliminates limit cycles due to overflow, on the one hand, but on the other hand, it causes undesirable signal distortion due to the nonlinearity of the clipper. In order to limit the amount of nonlinear distortion, it is important to scale the input signal and the unit sample response, between the input and any internal summing node in the system, such that overflow becomes a rare event.

For fixed-point arithmetic, let us first consider the extreme condition that overflow is not permitted at any node of the system. Let $y_k(n)$ denote the response of the system at the kth node when the input sequence is $x(n)$ and the unit sample response between the node and the input is $h_k(n)$. Then

$$|y_k(n)| = \left| \sum_{m=-\infty}^{\infty} h_k(m)x(n-m) \right| \leq \sum_{m=-\infty}^{\infty} |h_k(m)||x(n-m)|$$

Suppose that $x(n)$ is upper bounded by A_x. Then

$$|y_k(n)| \leq A_x \sum_{m=-\infty}^{\infty} |h_k(m)|, \qquad \text{for all } n \tag{9.6.13}$$

Now, if the dynamic range of the computer is limited to $(-1, 1)$, the condition

$$|y_k(n)| < 1$$

can be satisfied by requiring that the input $x(n)$ be scaled such that

$$A_x < \frac{1}{\sum_{m=-\infty}^{\infty} |h_k(m)|} \tag{9.6.14}$$

for all possible nodes in the system. The condition in (9.6.14) is both necessary and sufficient to prevent overflow.

The condition in (9.6.14) is overly conservative, however, to the point where the input signal may be scaled too much. In such a case, much of the precision used to represent $x(n)$ is lost. This is especially true for narrowband sequences, such as sinusoids, where the scaling implied by (9.6.14) is extremely severe. For narrowband signals we can use the frequency response characteristics of the system in determining the appropriate scaling. Since $|H(\omega)|$ represents the gain of the system at frequency ω, a less severe and reasonably adequate scaling is to require that

$$A_x < \frac{1}{\max\limits_{0 \le \omega \le \pi} |H_k(\omega)|} \tag{9.6.15}$$

where $H_k(\omega)$ is the Fourier transform of $\{h_k(n)\}$.

In the case of an FIR filter, the condition in (9.6.14) reduces to

$$A_x < \frac{1}{\sum_{m=0}^{M-1} |h_k(m)|} \tag{9.6.16}$$

which is now a sum over the M nonzero terms of the filter unit sample response.

Another approach to scaling is to scale the input so that

$$\sum_{n=-\infty}^{\infty} |y_k(n)|^2 \le C^2 \sum_{n=-\infty}^{\infty} |x(n)|^2 = C^2 E_x \tag{9.6.17}$$

From Parseval's theorem we have

$$\begin{aligned} \sum_{n=-\infty}^{\infty} |y_k(n)|^2 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\omega)X(\omega)|^2 d\omega \\ &\le E_x \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\omega)|^2 d\omega \end{aligned} \tag{9.6.18}$$

By combining (9.6.17) with (9.6.18), we obtain

$$C^2 \le \frac{1}{\sum_{n=-\infty}^{\infty} |h_k(n)|^2} = \frac{1}{(1/2\pi) \int_{-\pi}^{\pi} |H(\omega)|^2 d\omega} \tag{9.6.19}$$

If we compare the different scaling factors given above, we find that

$$\left[\sum_{n=-\infty}^{\infty} |h_k(n)|^2\right]^{1/2} \leq \max_{\omega} |H_k(\omega)| \leq \sum_{n=-\infty}^{\infty} |h_k(n)| \tag{9.6.20}$$

Clearly, (9.6.14) is the most pessimistic constraint.

In the following section we observe the ramifications of this scaling on the output signal-to-noise (power) ratio (SNR) from a first-order and a second-order filter section.

9.6.3 Statistical Characterization of Quantization Effects in Fixed-Point Realizations of Digital Filters

It is apparent from our treatment in the previous section that an analysis of quantization errors in digital filtering, based on deterministic models of quantization effects, is not a very fruitful approach. The basic problem is that the nonlinear effects in quantizing the products of two numbers and in clipping the sum of two numbers to prevent overflow are not easily modeled in large systems that contain many multipliers and many summing nodes.

To obtain more general results on the quantization effects in digital filters, we shall model the quantization errors in multiplication as an additive noise sequence $e(n)$, just as we did in characterizing the quantization errors in A/D conversion of an analog signal. For addition, we consider the effect of scaling the input signal to prevent overflow.

Let us begin our treatment with the characterization of the round-off noise in a single-pole filter which is implemented in fixed-point arithmetic and is described by the nonlinear difference equation

$$v(n) = Q_r[av(n-1)] + x(n) \tag{9.6.21}$$

The effect of rounding the product $av(n-1)$ is modeled as a noise sequence $e(n)$ added to the actual product $av(n-1)$, that is,

$$Q_r[av(n-1)] = av(n-1) + e(n) \tag{9.6.22}$$

With this model for the quantization error, the system under consideration is described by the *linear difference equation*

$$v(n) = av(n-1) + x(n) + e(n) \tag{9.6.23}$$

The corresponding system is illustrated in block diagram form in Fig. 9.6.6.

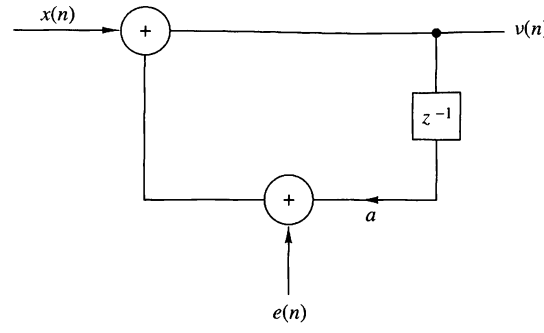


Figure 9.6.6
Additive noise model for the quantization error in a single-pole filter.

It is apparent from (9.6.23) that the output sequence $v(n)$ of the filter can be separated into two components. One is the response of the system to the input sequence $x(n)$. The second is the response of the system to the additive quantization noise $e(n)$. In fact, we can express the output sequence $v(n)$ as a sum of these two components, that is,

$$v(n) = y(n) + q(n) \tag{9.6.24}$$

where $y(n)$ represents the response of the system to $x(n)$, and $q(n)$ represents the response of the system to the quantization error $e(n)$. Upon substitution from (9.6.24) for $v(n)$ into (9.6.23), we obtain

$$y(n) + q(n) = ay(n-1) + aq(n-1) + x(n) + e(n) \tag{9.6.25}$$

To simplify the analysis, we make the following assumptions about the error sequence $e(n)$.

1. For any n, the error sequence $\{e(n)\}$ is uniformly distributed over the range $(-\frac{1}{2} \cdot 2^{-b}, \frac{1}{2} \cdot 2^{-b})$. This implies that the mean value of $e(n)$ is zero and its variance is
$$\sigma_e^2 = \frac{2^{-2b}}{12} \tag{9.6.26}$$
2. The error $\{e(n)\}$ is a stationary white noise sequence. In other words, the error $e(n)$ and the error $e(m)$ are uncorrelated for $n \neq m$.
3. The error sequence $\{e(n)\}$ is uncorrelated with the signal sequence $\{x(n)\}$.

The last assumption allows us to separate the difference equation in (9.6.25) into two uncoupled difference equations, namely,

$$y(n) = ay(n-1) + x(n) \tag{9.6.27}$$

$$q(n) = aq(n-1) + e(n) \tag{9.6.28}$$

The difference equation in (9.6.27) represents the input–output relation for the desired system and the difference equation in (9.6.28) represents the relation for the quantization error at the output of the system.

To complete the analysis, we make use of two important relationships developed in Section 12.1. The first is the relationship for the mean value of the output $q(n)$ of a linear shift-invariant filter with impulse response $h(n)$ when excited by a random sequence $e(n)$ having a mean value m_e. The result is

$$m_q = m_e \sum_{n=-\infty}^{\infty} h(n) \tag{9.6.29}$$

or, equivalently,

$$m_q = m_e H(0) \tag{9.6.30}$$

where $H(0)$ is the value of the frequency response $H(\omega)$ of the filter evaluated at $\omega = 0$.

The second important relationship is the expression for the autocorrelation sequence of the output $q(n)$ of the filter with impulse response $h(n)$ when the input random sequence $e(n)$ has an autocorrelation $\gamma_{ee}(n)$. This result is

$$\gamma_{qq}(n) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} h(k)h(l)\gamma_{ee}(k - l + n) \tag{9.6.31}$$

In the important special case where the random sequence is white (spectrally flat), the autocorrelation $\gamma_{ee}(n)$ is a unit sample sequence scaled by the variance σ_e^2, that is,

$$\gamma_{ee}(n) = \sigma_e^2 \delta(n) \tag{9.6.32}$$

Upon substituting (9.6.32) into (9.6.31), we obtain the desired result for the autocorrelation sequence at the output of a filter excited by white noise, namely,

$$\gamma_{qq}(n) = \sigma_e^2 \sum_{k=-\infty}^{\infty} h(k)h(k + n) \tag{9.6.33}$$

The variance σ_q^2 of the output noise is simply obtained by evaluating $\gamma_{qq}(n)$ at $n = 0$. Thus

$$\sigma_q^2 = \sigma_e^2 \sum_{k=-\infty}^{\infty} h^2(k) \tag{9.6.34}$$

and with the aid of Parseval's theorem, we have the alternative expression

$$\sigma_q^2 = \frac{\sigma_e^2}{2\pi} \int_{-\pi}^{\pi} |H(\omega)|^2 d\omega \tag{9.6.35}$$

In the case of the single-pole filter under consideration, the unit sample response is

$$h(n) = a^n u(n) \tag{9.6.36}$$

Since the quantization error due to rounding has zero mean, the mean value of the error at the output of the filter is $m_q = 0$. The variance of the error at the output of the filter is

$$\begin{aligned}\sigma_q^2 &= \sigma_e^2 \sum_{k=0}^{\infty} a^{2k} \\ &= \frac{\sigma_e^2}{1 - a^2}\end{aligned} \tag{9.6.37}$$

We observe that the noise power σ_q^2 at the output of the filter is enhanced relative to the input noise power σ_e^2 by the factor $1/(1 - a^2)$. This factor increases as the pole is moved closer to the unit circle.

To obtain a clearer picture of the effect of the quantization error, we should also consider the effect of scaling the input. Let us assume that the input sequence $\{x(n)\}$ is a white noise sequence (wideband signal), whose amplitude has been scaled according to (9.6.14) to prevent overflows in addition. Then

$$A_x < 1 - |a|$$

If we assume that $x(n)$ is uniformly distributed in the range $(-A_x, A_x)$, then, according to (9.6.31) and (9.6.34), the signal power at the output of the filter is

$$\begin{aligned}\sigma_y^2 &= \sigma_x^2 \sum_{k=0}^{\infty} a^{2k} \\ &= \frac{\sigma_x^2}{1 - a^2}\end{aligned} \tag{9.6.38}$$

where $\sigma_x^2 = (1 - |a|)^2/3$ is the variance of the input signal. The ratio of the signal power σ_y^2 to the quantization error power σ_q^2, which is called the signal-to-noise ratio (SNR), is simply

$$\begin{aligned}\frac{\sigma_y^2}{\sigma_q^2} &= \frac{\sigma_x^2}{\sigma_e^2} \\ &= (1 - |a|)^2 \cdot 2^{2(b+1)}\end{aligned} \tag{9.6.39}$$

This expression for the output SNR clearly illustrates the severe penalty paid as a consequence of the scaling of the input, especially when the pole is near the unit circle. By comparison, if the input is not scaled and the adder has a sufficient number of bits to avoid overflow, then the signal amplitude may be confined to the range $(-1, 1)$. In this case, $\sigma_x^2 = \frac{1}{3}$, which is independent of the pole position. Then

$$\frac{\sigma_y^2}{\sigma_q^2} = 2^{2(b+1)} \tag{9.6.40}$$

The difference between the SNRs in (9.6.40) and (9.6.39) clearly demonstrates the need to use more bits in addition than in multiplication. The number of additional

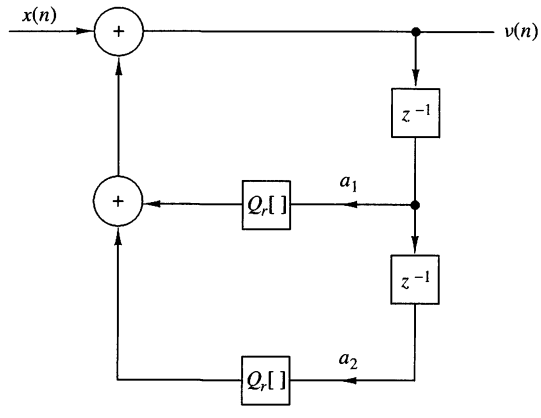


Figure 9.6.7
Two-pole digital filter with rounding quantizers.

bits depends on the position of the pole and should be increased as the pole is moved closer to the unit circle.

Next, let us consider a two-pole filter with infinite precision which is described by the linear difference equation

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + x(n) \tag{9.6.41}$$

where $a_1 = 2r\cos\theta$ and $a_2 = -r^2$. When the two products are rounded, we have a system which is described by the nonlinear difference equation

$$v(n) = Q_r[a_1 v(n-1)] + Q_r[a_2 v(n-2)] + x(n) \tag{9.6.42}$$

This system is illustrated in block diagram form in Fig. 9.6.7.

Now there are two multiplications, and hence two quantization errors are produced for each output. Consequently, we should introduce two noise sequences $e_1(n)$ and $e_2(n)$, which correspond to the quantizer outputs

$$\begin{aligned} Q_r[a_1 v(n-1)] &= a_1 v(n-1) + e_1(n) \\ Q_r[a_2 v(n-2)] &= a_2 v(n-2) + e_2(n) \end{aligned} \tag{9.6.43}$$

A block diagram for the corresponding model is shown in Fig. 9.6.8. Note that the error sequences $e_1(n)$ and $e_2(n)$ can be moved directly to the input of the filter.

As in the case of the first-order filter, the output of the second-order filter can be separated into two components, the desired signal component and the quantization error component. The former is described by the difference equation

$$y(n) = a_1 y(n-1) + a_2 y(n-2) + x(n) \tag{9.6.44}$$

while the latter satisfies the difference equation

$$q(n) = a_1 q(n-1) + a_2 q(n-2) + e_1(n) + e_2(n) \tag{9.6.45}$$

It is reasonable to assume that the two sequences $e_1(n)$ and $e_2(n)$ are uncorrelated.

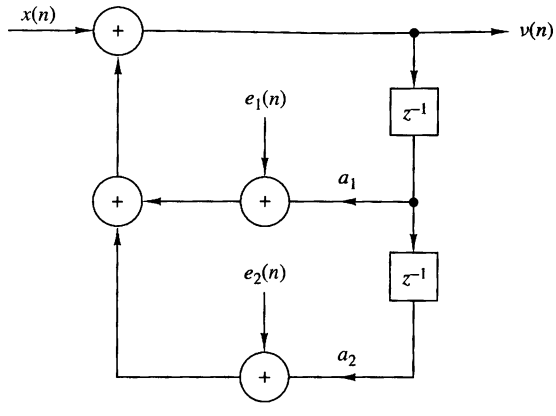


Figure 9.6.8
Additive noise model for the quantization errors in a two-pole filter realization.

Now the second-order filter has a unit sample response

$$h(n) = \frac{r^n}{\sin\theta}\sin(n+1)\theta u(n) \tag{9.6.46}$$

Hence

$$\sum_{n=0}^{\infty} h^2(n) = \frac{1+r^2}{1-r^2}\frac{1}{r^4+1-2r^2\cos 2\theta} \tag{9.6.47}$$

By applying (9.6.34), we obtain the variance of the quantization errors at the output of the filter in the form

$$\sigma_q^2 = \sigma_e^2\left(\frac{1+r^2}{1-r^2}\frac{1}{r^4+1-2r^2\cos 2\theta}\right) \tag{9.6.48}$$

In the case of the signal component, if we scale the input as in (9.6.14) to avoid overflow, the power in the output signal is

$$\sigma_y^2 = \sigma_x^2\sum_{n=0}^{\infty} h^2(n) \tag{9.6.49}$$

where the power in the input signal $x(n)$ is given by the variance

$$\sigma_x^2 = \frac{1}{3\left[\sum_{n=0}^{\infty}|h(n)|\right]^2} \tag{9.6.50}$$

Consequently, the SNR at the output of the two-pole filter is

$$\frac{\sigma_y^2}{\sigma_q^2} = \frac{\sigma_x^2}{\sigma_e^2} = \frac{2^{2(b+1)}}{\left[\sum_{n=0}^{\infty}|h(n)|\right]^2} \tag{9.6.51}$$

Although it is difficult to determine the exact value of the denominator term in (9.6.51), it is easy to obtain an upper and a lower bound. In particular, $|h(n)|$ is upper bounded as

$$|h(n)| \leq \frac{1}{\sin\theta} r^n, \qquad n \geq 0 \tag{9.6.52}$$

so that

$$\sum_{n=0}^{\infty} |h(n)| \leq \frac{1}{\sin\theta} \sum_{n=0}^{\infty} r^n = \frac{1}{(1-r)\sin\theta} \tag{9.6.53}$$

The lower bound may be obtained by noting that

$$|H(\omega)| = \left| \sum_{n=0}^{\infty} h(n) e^{-j\omega n} \right| \leq \sum_{n=0}^{\infty} |h(n)|$$

But

$$H(\omega) = \frac{1}{(1 - re^{j\theta}e^{-j\omega})(1 - re^{-j\theta}e^{-j\omega})}$$

At $\omega = \theta$, which is the resonant frequency of the filter, we obtain the largest value of $|H(\omega)|$. Hence

$$\sum_{n=0}^{\infty} |h(n)| \geq |H(\theta)| = \frac{1}{(1-r)\sqrt{1 + r^2 - 2r\cos 2\theta}} \tag{9.6.54}$$

Therefore, the SNR is bounded from above and below according to the relation

$$2^{2(b+1)}(1-r)^2 \sin^2\theta \leq \frac{\sigma_y^2}{\sigma_q^2} \leq 2^{2(b+1)}(1-r)^2(1 + r^2 - 2r\cos 2\theta) \tag{9.6.55}$$

For example, when $\theta = \pi/2$, the expression in (9.6.55) reduces to

$$2^{2(b+1)}(1-r)^2 \leq \frac{\sigma_y^2}{\sigma_q^2} \leq 2^{2(b+1)}(1-r)^2(1+r)^2 \tag{9.6.56}$$

The dominant term in this bound is $(1-r)^2$ which acts to reduce the SNR dramatically as the poles move toward the unit circle. Hence the effect of scaling in the second-order filter is more severe than in the single-pole filter. Note that if $d = 1 - r$ is the distance of the pole from the unit circle, the SNR in (9.6.56) is reduced by d^2, whereas in the single-pole filter the reduction is proportional to d. These results serve to reinforce the earlier statement regarding the use of more bits in addition than in multiplication as a mechanism for avoiding the severe penalty due to scaling.

The analysis of the quantization effects in a second-order filter can be applied directly to higher-order filters based on a parallel realization. In this case each second-order filter section is independent of all the other sections, and therefore the total

quantization noise power at the output of the parallel bank is simply the linear sum of the quantization noise powers of each of the individual sections. On the other hand, the cascade realization is more difficult to analyze. For the cascade interconnection, the noise generated in any second-order filter section is filtered by the succeeding sections. As a consequence, there is the issue of how to pair together real-valued poles to form second-order sections and how to arrange the resulting second-order filters to minimize the total noise power at the output of the high-order filter. This general topic was investigated by Jackson (1970a, b), who showed that poles close to the unit circle should be paired with nearby zeros to reduce the gain of each second-order section. In ordering the second-order sections in cascade, a reasonable strategy is to place the sections in the order of decreasing maximum frequency gain. In this case the noise power generated in the early high-gain section is not boosted significantly by the latter sections.

The following example illustrates the point that proper ordering of sections in a cascade realization is important in controlling the round-off noise at the output of the overall filter.

EXAMPLE 9.6.1

Determine the variance of the round-off noise at the output of the two cascade realizations of the filter with system function

$$H(z) = H_1(z)H_2(z)$$

where

$$H_1(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

$$H_2(z) = \frac{1}{1 - \frac{1}{4}z^{-1}}$$

Solution. Let $h(n)$, $h_1(n)$, and $h_2(n)$ represent the unit sample responses corresponding to the system functions $H(z)$, $H_1(z)$, and $H_2(z)$, respectively. It follows that

$$h_1(n) = (\tfrac{1}{2})^n u(n), \qquad h_2(n) = (\tfrac{1}{4})^n u(n)$$

$$h(n) = [2(\tfrac{1}{2})^n - (\tfrac{1}{4})^n]u(n)$$

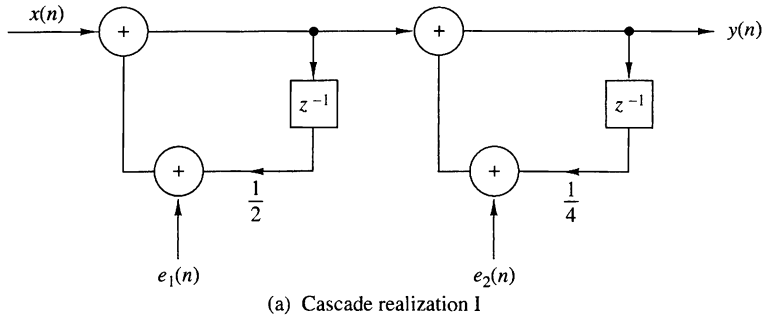
The two cascade realizations are shown in Fig. 9.6.9.

In the first cascade realization, the variance of the output is

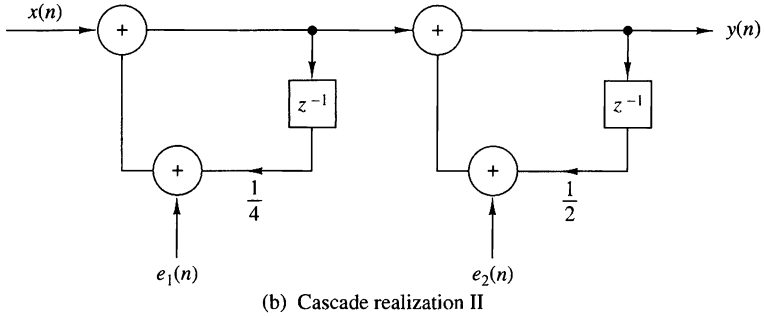
$$\sigma_{q1}^2 = \sigma_e^2 \left[\sum_{n=0}^{\infty} h^2(n) + \sum_{n=0}^{\infty} h_2^2(n) \right]$$

In the second cascade realization, the variance of the output noise is

$$\sigma_{q2}^2 = \sigma_e^2 \left[\sum_{n=0}^{\infty} h^2(n) + \sum_{n=0}^{\infty} h_1^2(n) \right]$$



(a) Cascade realization I



(b) Cascade realization II

Figure 9.6.9 Two cascade realizations in Example 9.6.1: (a) cascade realization I; (b) cascade realization II.

Now

$$\sum_{n=0}^{\infty} h_1^2(n) = \frac{1}{1-\frac{1}{4}} = \frac{4}{3}$$

$$\sum_{n=0}^{\infty} h_2^2(n) = \frac{1}{1-\frac{1}{16}} = \frac{16}{15}$$

$$\sum_{m=0}^{\infty} h^2(n) = \frac{4}{1-\frac{1}{4}} - \frac{4}{1-\frac{1}{8}} + \frac{1}{1-\frac{1}{16}} = 1.83$$

Therefore,

$$\sigma_{q1}^2 = 2.90\sigma_e^2$$

$$\sigma_{q2}^2 = 3.16\sigma_e^2$$

and the ratio of noise variances is

$$\frac{\sigma_{q2}^2}{\sigma_{q1}^2} = 1.09$$

Consequently, the noise power in the second cascade realization is 9% larger than in the first realization.

9.7 Summary and References

From the treatment in this chapter we have seen that there are various realizations of discrete-time systems. FIR systems can be realized in a direct form, a cascade form, a frequency sampling form, and a lattice form. IIR systems can also be realized in a direct form, a cascade form, a lattice or a lattice-ladder form, and a parallel form.

For any given system described by a linear constant-coefficient difference equation, these realizations are equivalent in that they represent the same system and produce the same output for any given input, provided that the internal computations are performed with infinite precision. However, the various structures are not equivalent when they are realized with finite-precision arithmetic.

Additional FIR and IIR filter structures can be obtained by adopting a state-space formulation that provides an internal description of a system. Such state-space realizations were treated in previous editions of this book, but have been dropped from this edition due to space limitations. The use of state-space filter structures in the realization of IIR systems has been proposed by Mullis and Roberts (1976a,b), and further developed by Hwang (1977), Jackson et al. (1979), Jackson (1979), Mills et al. (1981), and Bomar (1985).

Three important factors are presented for choosing among the various FIR and IIR system realizations. These factors are computational complexity, memory requirements, and finite-word-length effects. Depending on either the time-domain or the frequency-domain characteristics of a system, some structures may require less computation and/or less memory than others. Hence our selection must consider these two important factors.

In deriving the transposed structures in Section 9.3, we introduced several concepts and operations on signal flow graphs. Signal flow graphs are treated in depth in the books by Mason and Zimmerman (1960) and Chow and Cassignol (1962).

Another important structure for IIR systems, a *wave digital filter*, has been investigated by Fettweis (1971) and further developed by Sedlmeyer and Fettweis (1973). A treatment of this filter structure can also be found in the book by Antoniou (1979).

Finite-word-length effects are an important factor in the implementation of digital signal processing systems. In this chapter we described the effects of a finite word length in digital filtering. In particular, we considered the following problems dealing with finite-word length effects:

1. Parameter quantization in digital filters
2. Round-off noise in multiplication
3. Overflow in addition
4. Limit cycles

These four effects are internal to the filter and influence the method by which the system will be implemented. In particular, we demonstrated that high-order systems, especially IIR systems, should be realized by using second-order sections as building blocks. We advocated the use of the direct form II realization, either the conventional or the transposed form.

Effects of round-off errors in fixed-point implementations of FIR and IIR filter structures have been investigated by many researchers. We cite the papers by Gold and Rader (1966), Rader and Gold (1967b), Jackson (1970a,b), Liu (1971), Chan and Rabiner (1973a,b,c), and Oppenheim and Weinstein (1972).

Limit-cycle oscillations occur in IIR filters as a result of quantization effects in fixed-point multiplication and rounding. Investigation of limit cycles in digital filtering and their characteristic behavior is treated in the papers by Parker and Hess (1971), Brubaker and Gowdy (1972), Sandberg and Kaiser (1972), and Jackson (1969, 1979). The latter paper deals with limit cycles in state-space structures. Methods have also been devised to eliminate limit cycles caused by round-off errors. For example, the papers by Barnes and Fam (1977), Fam and Barnes (1979), Chang (1981), Butterweck et al. (1984), and Auer (1987) discuss this problem. Overflow oscillations have been treated in the paper by Ebert et al. (1969).

The effects of parameter quantization have been treated in a number of papers. We cite for reference the work of Rader and Gold (1967b), Knowles and Olcayto (1968), Avenhaus and Schuessler (1970), Herrmann and Schuessler (1970b), Chan and Rabiner (1973c), and Jackson (1976).

Finally, we mention that the lattice and lattice-ladder filter structures are known to be robust in fixed-point implementations. For a treatment of these types of filters, the reader is referred to the papers of Gray and Markel (1973), Makhoul (1978), and Morf et al. (1977) and to the book by Markel and Gray (1976).

Problems

1 Determine a direct-form realization for the following linear phase filters.

(a) $h(n) = \{\underset{\uparrow}{1}, 2, 3, 4, 3, 2, 1\}$

(b) $h(n) = \{\underset{\uparrow}{1}, 2, 3, 3, 2, 1\}$

2 Consider an FIR filter with system function

$$H(z) = 1 + 2.88z^{-1} + 3.4048z^{-2} + 1.74z^{-3} + 0.4z^{-4}$$

Sketch the direct-form and lattice realizations of the filter and determine in detail the corresponding input–output equations. Is the system minimum phase?

9.3 Determine the system function and the impulse response of the system shown in Fig. P9.3.

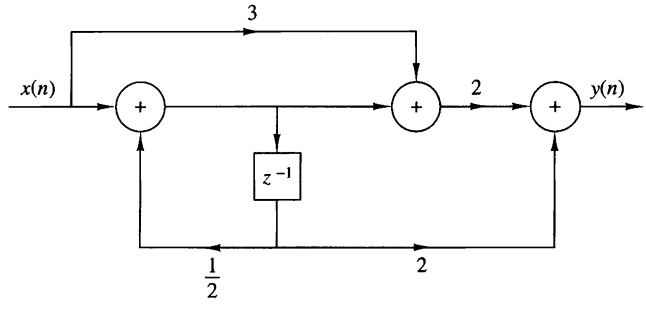


Figure P9.3

9.4 Determine the system function and the impulse response of the system shown in Fig. P9.4.

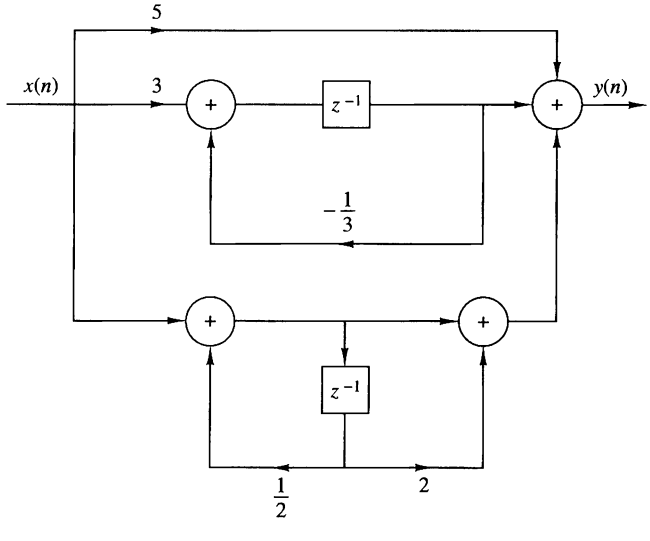


Figure P9.4

9.5 Determine the transposed structure of the system in Fig. P9.4 and verify that both the original and the transposed system have the same system function.

9.6 Determine a_1, a_2 and c_1, and c_0 in terms of b_1 and b_2 so that the two systems in Fig. P9.6 are equivalent.

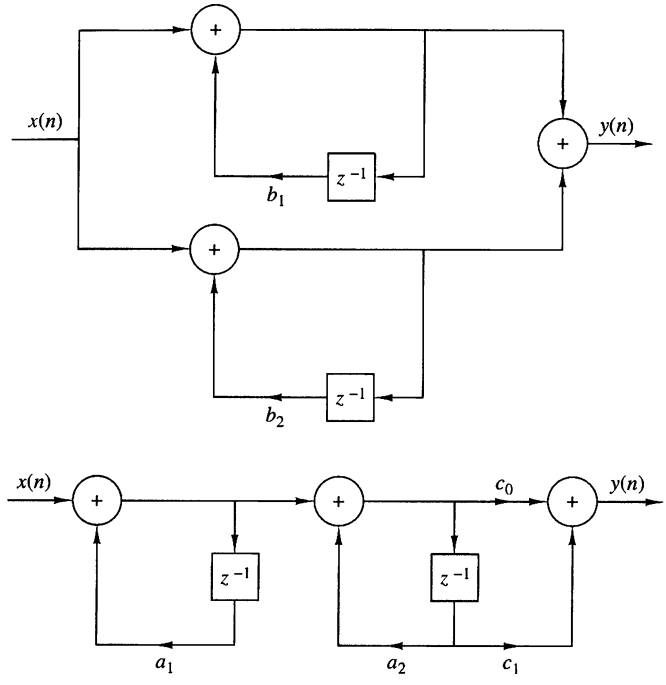


Figure P9.6

9.7 Consider the filter shown in Fig. P9.7.

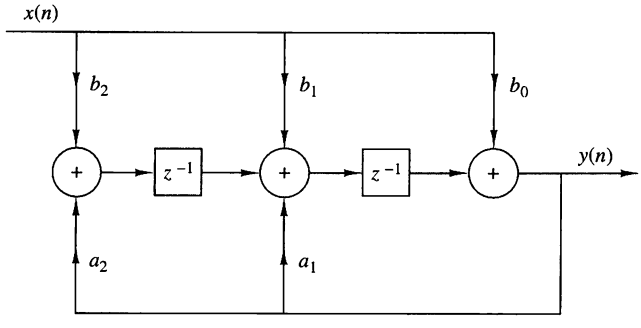


Figure P9.7

(a) Determine its system function.

(b) Sketch the pole–zero plot and check for stability if

1. $b_0 = b_2 = 1, \quad b_1 = 2, \quad a_1 = 1.5, a_2 = -0.9$

2. $b_0 = b_2 = 1, \quad b_1 = 2, \quad a_1 = 1, a_2 = -2$

(c) Determine the response to $x(n) = \cos(\pi n/3)$ if $b_0 = 1$, $b_1 = b_2 = 0$, $a_1 = 1$, and $a_2 = -0.99$.

9.8 Consider an LTI system, initially at rest, described by the difference equation

$$y(n) = \tfrac{1}{4}y(n-2) + x(n)$$

(a) Determine the impulse response, $h(n)$, of the system.

(b) What is the response of the system to the input signal

$$x(n) = [(\tfrac{1}{2})^n + (-\tfrac{1}{2})^n]u(n)$$

(c) Determine the direct form II, parallel-form, and cascade-form realizations for this system.

(d) Sketch roughly the magnitude response $|H(\omega)|$ of this system.

9.9 Obtain the direct form I, direct form II, cascade, and parallel structures for the following systems.

(a) $y(n) = \frac{3}{4}y(n-1) - \frac{1}{8}y(n-2) + x(n) + \frac{1}{3}x(n-1)$

(b) $y(n) = -0.1y(n-1) + 0.72y(n-2) + 0.7x(n) - 0.252x(n-2)$

(c) $y(n) = -0.1y(n-1) + 0.2y(n-2) + 3x(n) + 3.6x(n-1) + 0.6x(n-2)$

(d) $H(z) = \dfrac{2(1-z^{-1})(1+\sqrt{2}z^{-1}+z^{-2})}{(1+0.5z^{-1})(1-0.9z^{-1}+0.81z^{-2})}$

(e) $y(n) = \frac{1}{2}y(n-1) + \frac{1}{4}y(n-2) + x(n) + x(n-1)$

(f) $y(n) = y(n-1) - \frac{1}{2}y(n-2) + x(n) - x(n-1) + x(n-2)$

Which of the systems above are stable?

9.10 Show that the systems in Fig. P9.10 are equivalent.

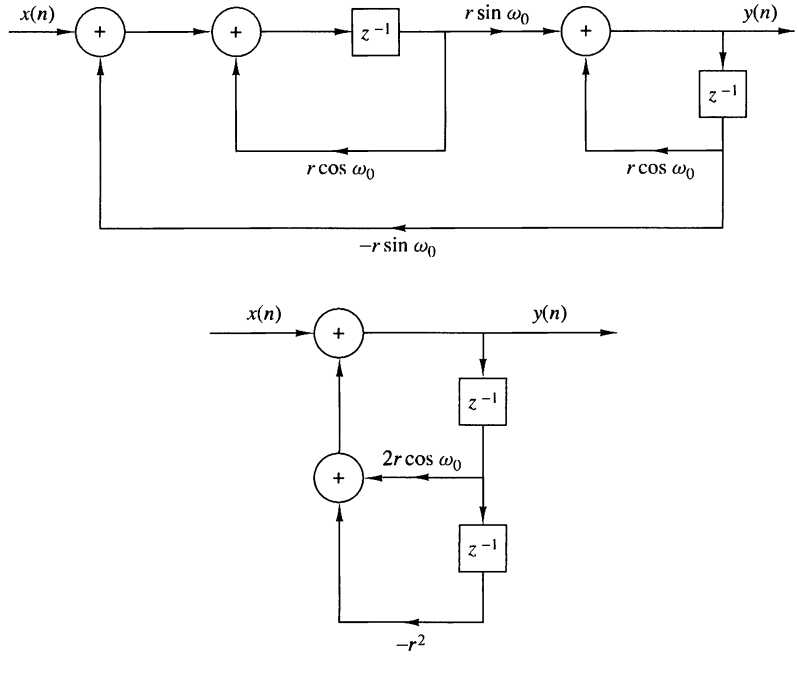


Figure P9.10

9.11 Determine all the FIR filters which are specified by the lattice parameters $K_1 = \frac{1}{2}$, $K_2 = 0.6$, $K_3 = -0.7$, and $K_4 = \frac{1}{3}$.

9.12 Determine the set of difference equations for describing a realization of an IIR system based on the use of the transposed direct form II structure for the second-order subsystems.

***9.13** Write a program that implements a parallel-form realization based on transposed direct form II second-order modules.

***9.14** Write a program that implements a cascade-form realization based on regular direct form II second-order modules.

9.15 Determine the parameters $\{K_m\}$ of the lattice filter corresponding to the FIR filter described by the system function

$$H(z) = A_2(z) = 1 + 2z^{-1} + z^{-2}$$

9.16 **(a)** Determine the zeros and sketch the zero pattern for the FIR lattice filter with parameters

$$K_1 = \tfrac{1}{2}, \qquad K_2 = -\tfrac{1}{3}, \qquad K_3 = 1$$

(b) The same as in part (a) but with $K_3 = -1$.

(c) You should have found that all the zeros lie exactly on the unit circle. Can this result be generalized? How?

(d) Sketch the phase response of the filters in parts (a) and (b). What did you notice? Can this result be generalized? How?

9.17 Consider an FIR lattice filter with coefficients $K_1 = 0.65$, $K_2 = -0.34$, and $K_3 = 0.8$.

(a) Find its impulse response by tracing a unit impulse input through the lattice structure.

(b) Draw the equivalent direct-form structure.

9.18 Consider a causal IIR system with system function

$$H(z) = \frac{1 + 2z^{-1} + 3z^{-2} + 2z^{-3}}{1 + 0.9z^{-1} - 0.8z^{-2} + 0.5z^{-3}}$$

(a) Determine the equivalent lattice-ladder structure.

(b) Check if the system is stable.

9.19 Determine the input–output relationship, the system function, and plot the pole–zero pattern for the discrete-time system shown in Fig. P9.19.

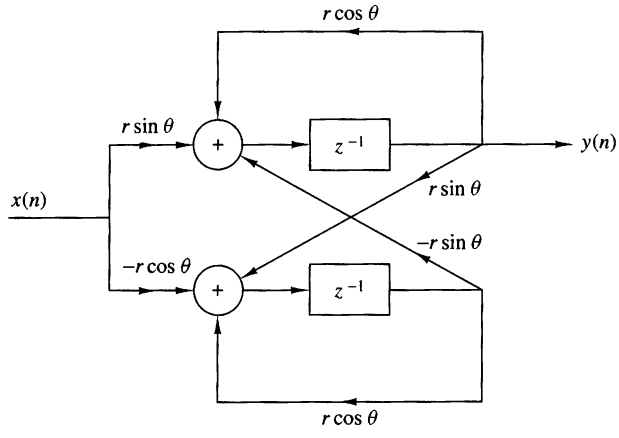


Figure P9.19

9.20 Determine the lattice realization for the digital resonator

$$H(z) = \frac{1}{1 - (2r\cos\omega_0)z^{-1} + r^2z^{-2}}$$

9.21

(a) Determine the impulse response of an FIR lattice filter with parameters $K_1 = 0.6$, $K_2 = 0.3$, $K_3 = 0.5$, and $K_4 = 0.9$.

(b) Sketch the direct-form and lattice all-zero and all-pole filters specified by the K-parameters given in part (a).

9.22 **(a)** Determine the lattice-ladder realization for the resonator

$$H(z) = \frac{1 - z^{-1}}{1 - (2r\cos\omega_0)z^{-1} + r^2z^{-2}}$$

(b) What happens if $r = 1$?

9.23 Sketch the lattice-ladder structure for the system

$$H(z) = \frac{1 - 0.8z^{-1} + 0.15z^{-2}}{1 + 0.1z^{-1} - 0.72z^{-2}}$$

9.24 Consider a pole–zero system with system function

$$H(z) = \frac{(1 - 0.5e^{j\pi/4}z^{-1})(1 - 0.5e^{-j\pi/4}z^{-1})}{(1 - 0.8e^{j\pi/3}z^{-1})(1 - 0.8e^{-j\pi/3}z^{-1})}$$

Sketch the regular and transpose direct form II realizations of the system.

9.25 Determine a parallel and a cascade realization of the system

$$H(z) = \frac{1 + z^{-1}}{(1 - z^{-1})(1 - 0.8e^{j\pi/4}z^{-1})(1 - 0.8e^{-j\pi/4}z^{-1})}$$

9.26 The generic floating-point format for a DSP microprocessor is the following:

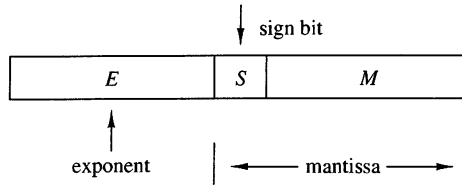


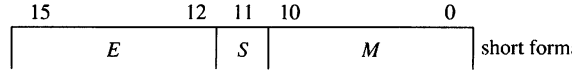
Figure P9.26

The value of the number X is given by

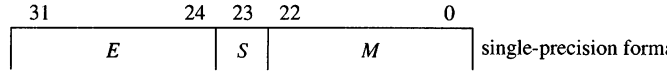
$$X = \begin{cases} 01.M \times 2^E & \text{if } S = 0 \\ 10.M \times 2^E & \text{if } S = 1 \\ 0 & \text{if } E \text{ is the most negative two's-complement value} \end{cases}$$

Determine the range of positive and negative numbers for the following two formats:

(a)



(b)



9.27 Consider the IIR recursive filter shown in Fig. P9.27 and let $h_F(n)$, $h_R(n)$, and $h(n)$ denote the impulse responses of the FIR section, the recursive section, and the overall filter, respectively.

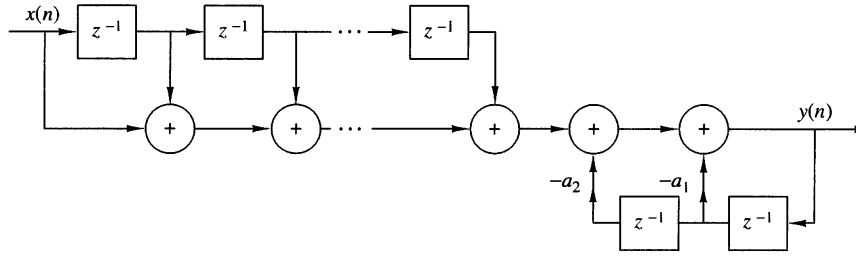


Figure P9.27

(a) Find all the causal and stable recursive second-order sections with integer coefficients (a_1, a_2) and determine and sketch their impulse responses and frequency responses. These filters do not require complicated multiplications or quantization after multiplications.

(b) Show that three of the sections obtained in part (a) can be obtained by interconnection of other sections.

(c) Find a difference equation that describes the impulse response $h(n)$ of the filter and determine the conditions for the overall filter to be FIR.

(d) Rederive the results in parts (a) to (c) using z-domain considerations.

9.28 This problem illustrates the development of digital filter structures using Horner's rule for polynomial evaluation. To this end consider the polynomial

$$p(x) = \alpha_p x^p + a_{p-1}x^{p-1} + \cdots + a_1 x + a_0$$

which computes $p(x)$ with the minimum cost of p multiplications and p additions.

(a) Draw the structures corresponding to the factorizations

$$H_1(z) = b_0(1 + b_1 z^{-1}(1 + b_2 z^{-1}(1 + b_3 z^{-1})))$$
$$H(z) = b_0(z^{-3} + (b_1 z^{-2} + (b_2 z^{-1} + b_3)))$$

and determine the system function, number of delay elements, and arithmetic operations for each structure

(b) Draw the Horner structure for the following linear-phase system:

$$H(z) = z^{-1}\left[\alpha_0 + \sum_{k=1}^{3}(z^{-k} + z^k)\alpha_k\right]$$

9.29 Let x_1 and x_2 be $(b+1)$-bit binary numbers with magnitude less than 1. To compute the sum of x_1 and x_2 using two's-complement representation we treat them as $(b+1)$-bit unsigned numbers, perform addition modulo-2 and ignore any carry after the sign bit.

(a) Show that if the sum of two numbers with the same sign has the opposite sign, this corresponds to overflow.

(b) Show that when we compute the sum of several numbers using two's-complement representation, the result will be correct, even if there are overflows, if the correct sum is less than 1 in magnitude. Illustrate this argument by constructing a simple example with three numbers.

9.30 Consider the system described by the difference equation

$$y(n) = ay(n-1) - ax(n) + x(n-1)$$

(a) Show that it is allpass.

(b) Obtain the direct form II realization of the system

(c) If you quantize the coefficients of the system in part (b), is it still allpass?

(d) Obtain a realization by rewriting the difference equation as

$$y(n) = a[y(n-1) - x(n)] + x(n-1)$$

(e) If you quantize the coefficients of the system in part (d), is it still allpass?

9.31 Consider the system

$$y(n) = \tfrac{1}{2}y(n-1) + x(n)$$

(a) Compute its response to the input $x(n) = (\frac{1}{4})^n u(n)$ assuming infinite-precision arithmetic.

(b) Compute the response of the system $y(n)$, $0 \leq n \leq 5$ to the same input, assuming finite-precision sign-and-magnitude fractional arithmetic with five bits (i.e., the sign bit plus four fractional bits). The quantization is performed by truncation.

(c) Compare the results obtained in parts (a) and (b).

9.32 The input to the system

$$y(n) = 0.999y(n-1) + x(n)$$

is quantized to $b = 8$ bits. What is the power produced by the quantization noise at the output of the filter?

9.33 Consider the system

$$y(n) = 0.875y(n-1) - 0.125y(n-2) + x(n)$$

(a) Compute its poles and design the cascade realization of the system.

(b) Quantize the coefficients of the system using truncation, maintaining a sign bit plus three other bits. Determine the poles of the resulting system.

(c) Repeat part (b) for the same precision using rounding.

(d) Compare the poles obtained in parts (b) and (c) with those in part (a). Which realization is better? Sketch the frequency responses of the systems in parts (a), (b), and (c).

9.34 Consider the system

$$H(z) = \frac{1 - \frac{1}{2}z^{-1}}{(1 - \frac{1}{4}z^{-1})(1 + \frac{1}{4}z^{-1})}$$

(a) Draw all possible realizations of the system.

(b) Suppose that we implement the filter with fixed-point sign-and-magnitude fractional arithmetic using $(b + 1)$ bits (one bit is used for the sign). Each resulting product is rounded into b bits. Determine the variance of the round-off noise created by the multipliers at the output of each one of the realizations in part (a).

9.35 The first-order filter shown in Fig. P9.35 is implemented in four-bit (including sign) fixed-point two's-complement fractional arithmetic. Products are rounded to four-bit representation. Using the input $x(n) = 0.10\delta(n)$, determine:

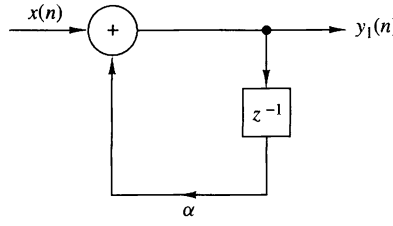


Figure P9.35

(a) The first five outputs if $\alpha = 0.5$. Does the filter go into a limit cycle?

(b) The first five outputs if $\alpha = 0.75$. Does the filter go into a limit cycle?

9.36 The digital system shown in Fig. P9.36 uses a six-bit (including sign) fixed-point two's-complement A/D converter with rounding, and the filter $H(z)$ is implemented using eight-bit (including sign) fixed-point two's-complement fractional arithmetic with rounding. The input $x(t)$ is a zero-mean uniformly distributed random process having autocorrelation $\gamma_{xx}(\tau) = 3\delta(\tau)$. Assume that the A/D converter can handle input values up to ± 1.0 without overflow.

(a) What value of attenuation should be applied prior to the A/D converter to assure that it does not overflow?

(b) With the attenuation above, what is the signal-to-quantization-noise ratio (SQNR) at the A/D converter output?

(c) The six-bit A/D samples can be left justified, right justified, or centered in the eight-bit word used as the input to the digital filter. What is the correct strategy to use for maximum SNR at the filter output without overflow?

(d) What is the SNR at the output of the filter due to all quantization noise sources?

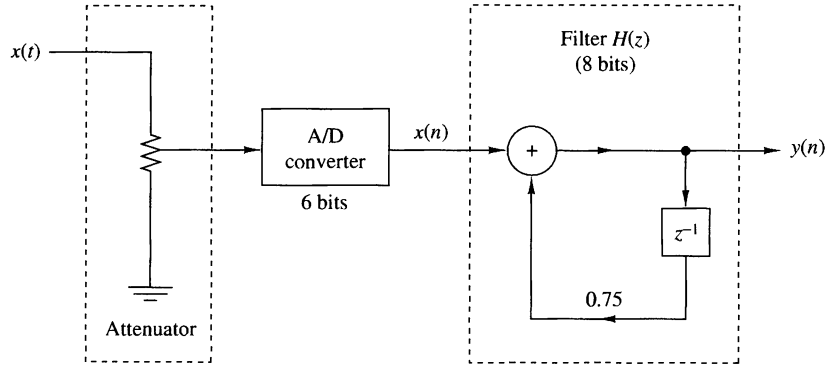


Figure P9.36

9.37 Shown in Fig. P9.37 is the coupled-form implementation of a two-pole filter with poles at $x = re^{\pm j\theta}$. There are four real multiplications per output point. Let $e_i(n)$, $i = 1, 2, 3, 4$ represent the round-off noise in a fixed-point implementation of the filter. Assume that the noise sources are zero-mean mutually uncorrelated stationary white noise sequences. For each n the probability density function $p(e)$ is uniform in the range $-\Delta/2 \leq e \leq \Delta/2$, where $\Delta = 2^{-b}$.

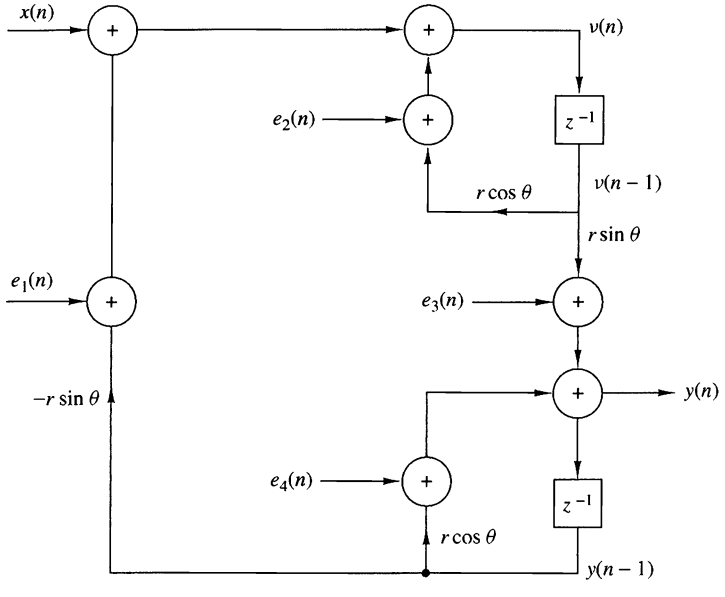


Figure P9.37

(a) Write the two coupled difference equations for $y(n)$ and $v(n)$, including the noise sources and the input sequence $x(n)$.

(b) From these two difference equations, show that the filter system functions $H_1(z)$ and $H_2(z)$ between the input noise terms $e_1(n) + e_2(n)$ and $e_3(n) + e_4(n)$ and the output $y(n)$ are:

$$H_1(z) = \frac{r \sin\theta z^{-1}}{1 - 2r\cos\theta z^{-1} + r^2 z^{-2}}$$

$$H_2(z) = \frac{1 - r\cos\theta z^{-1}}{1 - 2r\cos\theta z^{-1} + r^2 z^{-2}}$$

We know that

$$H(z) = \frac{1}{1 - 2r\cos\theta z^{-1} + r2z^{-2}} \Rightarrow h(n) = \frac{1}{\sin\theta} r^n \sin(n+1)\theta u(n)$$

Determine $h_1(n)$ and $h_2(n)$.

(c) Determine a closed-form expression for the variance of the total noise from $e_i(n)$, $i = 1, 2, 3, 4$ at the output of the filter.

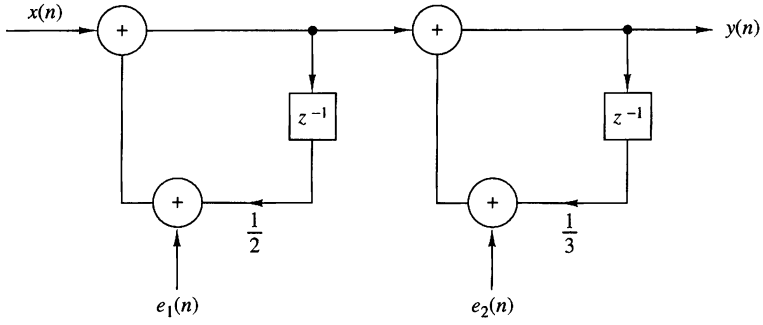
9.38 Determine the variance of the round-off noise at the output of the two cascade realizations of the filter shown in Fig. P9.38, with system function

$$H(z) = H_1(z)H_2(z)$$

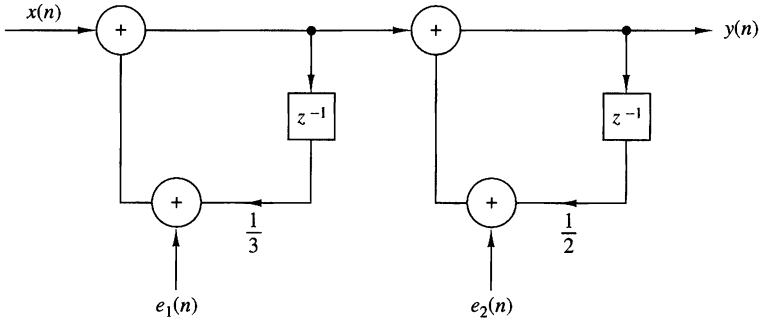
where

$$H_1(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

$$H_2(z) = \frac{1}{1 - \frac{1}{3}z^{-1}}$$



(a) Cascade realization I



(b) Cascade realization II

Figure P9.38

9.39 *Quantization effects in direct-form FIR filters* Consider a direct-form realization of an FIR filter of length M. Suppose that the multiplication of each coefficient with the corresponding signal sample is performed in fixed-point arithmetic with b bits and each product is rounded to b bits. Determine the variance of the quantization noise at the output of the filter by using a statistical characterization of the round-off noise as in Section 9.6.3.

9.40 Consider the system specified by the system function

$$H(z) = \frac{B(z)}{A(z)}$$

$$= \left[G_1 \frac{(1 - 0.8e^{j\pi/4}z^{-1})(1 - 0.8e^{-j\pi/4}z^{-1})}{(1 - \frac{1}{2}z^{-1})(1 + \frac{1}{3}z^{-1})} \right] \left[G_2 \frac{(1 + \frac{1}{4}z^{-1})(1 - \frac{5}{8}z^{-1})}{(1 - 0.8e^{j\pi/3}z^{-1})(1 - 0.8e^{-j\pi/3}z^{-1})} \right]$$

(a) Choose G_1 and G_2 so that the gain of each second-order section at $\omega = 0$ is equal to 1.

(b) Sketch the direct form 1, direct form 2, and cascade realizations of the system.

(c) Write a program that implements the direct form 1 and direct form 2, and compute the first 100 samples of the impulse response and the step response of the system.

(d) Plot the results in part (c) to illustrate the proper functioning of the programs.

9.41 Consider the system given in Problem 9.40 with $G_1 = G_2 = 1$.

(a) Determine a lattice realization for the system

$$H(z) = B(z)$$

(b) Determine a lattice realization for the system

$$H(z) = \frac{1}{A(z)}$$

(c) Determine a lattice-ladder realization for the system $H(z) = B(z)/A(z)$.

(d) Write a program for the implementation of the lattice-ladder structure in part (c).

(e) Determine and sketch the first 100 samples of the impulse responses of the systems in parts (a) through (c) by working with the lattice structures.

(f) Compute and sketch the first 100 samples of the convolution of impulse responses in parts (a) and (b). What did you find? Explain your results.

9.42 Consider the system given in Problem 9.40. Determine the parallel-form structure and write a program for its implementation.

CHAPTER

10

Design of Digital Filters

With the background that we have developed in the preceding chapters, we are now in a position to treat the subject of digital filter design. We shall describe several methods for designing FIR and IIR digital filters.

In the design of frequency-selective filters, the desired filter characteristics are specified in the frequency domain in terms of the desired magnitude and phase response of the filter. In the filter design process, we determine the coefficients of a causal FIR or IIR filter that closely approximates the desired frequency response specifications. The issue of which type of filter to design, FIR or IIR, depends on the nature of the problem and on the specifications of the desired frequency response.

In practice, FIR filters are employed in filtering problems where there is a requirement for a linear-phase characteristic within the passband of the filter. If there is no requirement for a linear-phase characteristic, either an IIR or an FIR filter may be employed. However, as a general rule, an IIR filter has lower sidelobes in the stopband than an FIR filter having the same number of parameters. For this reason, if some phase distortion is either tolerable or unimportant, an IIR filter is preferable, primarily because its implementation involves fewer parameters, requires less memory and has lower computational complexity.

In conjunction with our discussion of digital filter design, we describe frequency transformations in both the analog and digital domains for transforming a lowpass prototype filter into either another lowpass, bandpass, bandstop, or highpass filter.

Today, FIR and IIR digital filter design is greatly facilitated by the availability of numerous computer software programs. In describing the various digital filter design methods in this chapter, our primary objective is to give the reader the background necessary to select the filter that best matches the application and satisfies the design requirements.

10.1 General Considerations

In Section 5.4, we described the characteristics of ideal filters and demonstrated that such filters are not causal and therefore are not physically realizable. In this section, the issue of causality and its implications is considered in more detail. Following this discussion, we present the frequency response characteristics of causal FIR and IIR digital filters.

10.1.1 Causality and Its Implications

Let us consider the issue of causality in more detail by examining the impulse response $h(n)$ of an ideal lowpass filter with frequency response characteristic

$$H(\omega) = \begin{cases} 1, & |\omega| \le \omega_c \\ 0, & \omega_c < \omega \le \pi \end{cases} \tag{10.1.1}$$

The impulse response of this filter is

$$h(n) = \begin{cases} \dfrac{\omega_c}{\pi}, & n = 0 \\ \dfrac{\omega_c}{\pi} \dfrac{\sin \omega_c n}{\omega_c n}, & n \neq 0 \end{cases} \tag{10.1.2}$$

A plot of $h(n)$ for $\omega_c = \pi/4$ is illustrated in Fig. 10.1.1. It is clear that the ideal lowpass filter is noncausal and hence it cannot be realized in practice.

One possible solution is to introduce a large delay n_0 in $h(n)$ and arbitrarily to set $h(n) = 0$ for $n < n_0$. However, the resulting system no longer has an ideal frequency response characteristic. Indeed, if we set $h(n) = 0$ for $n < n_0$, the Fourier series expansion of $H(\omega)$ results in the Gibbs phenomenon, as will be described in Section 10.2.

Although this discussion is limited to the realization of a lowpass filter, our conclusions hold, in general, for all the other ideal filter characteristics. In brief,

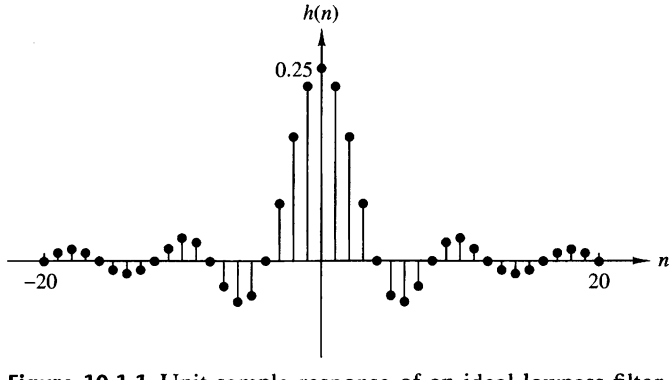


Figure 10.1.1 Unit sample response of an ideal lowpass filter.

none of the ideal filter characteristics previously illustrated in Fig. 5.4.1 are causal, hence all are physically unrealizable.

A question that naturally arises at this point is the following: What are the necessary and sufficient conditions that a frequency response characteristic $H(\omega)$ must satisfy in order for the resulting filter to be causal? The answer to this question is given by the Paley–Wiener theorem, which can be stated as follows:

Paley–Wiener Theorem. If $h(n)$ has finite energy and $h(n) = 0$ for $n < 0$, then [for a reference, see Wiener and Paley (1934)]

$$\int_{-\pi}^{\pi} |ln|H(\omega)||d\omega < \infty \tag{10.1.3}$$

Conversely, if $|H(\omega)|$ is square integrable and if the integral in (10.1.3) is finite, then we can associate with $|H(\omega)|$ a phase response $\Theta(\omega)$, so that the resulting filter with frequency response

$$H(\omega) = |H(\omega)|e^{j\Theta(\omega)}$$

is causal.

One important conclusion that we draw from the Paley–Wiener theorem is that the magnitude function $|H(\omega)|$ can be zero at some frequencies, but it cannot be zero over any finite band of frequencies, since the integral then becomes infinite. Consequently, any ideal filter is noncausal.

Apparently, causality imposes some tight constraints on a linear time-invariant system. In addition to the Paley–Wiener condition, causality also implies a strong relationship between $H_R(\omega)$ and $H_I(\omega)$, the real and imaginary components of the frequency response $H(\omega)$. To illustrate this dependence, we decompose $h(n)$ into an even and an odd sequence, that is,

$$h(n) = h_e(n) + h_o(n) \tag{10.1.4}$$

where

$$h_e(n) = \frac{1}{2}[h(n) + h(-n)] \tag{10.1.5}$$

and

$$h_o(n) = \frac{1}{2}[h(n) - h(-n)] \tag{10.1.6}$$

Now, if $h(n)$ is causal, it is possible to recover $h(n)$ from its even part $h_e(n)$ for $0 \leq n \leq \infty$ or from its odd component $h_o(n)$ for $1 \leq n \leq \infty$.

Indeed, it can be easily seen that

$$h(n) = 2h_e(n)u(n) - h_e(0)\delta(n), \qquad n \geq 0 \tag{10.1.7}$$

and

$$h(n) = 2h_o(n)u(n) + h(0)\delta(n), \qquad n \geq 1 \tag{10.1.8}$$

Since $h_o(n) = 0$ for $n = 0$, we cannot recover $h(0)$ from $h_o(n)$ and hence we also must know $h(0)$. In any case, it is apparent that $h_o(n) = h_e(n)$ for $n \geq 1$, so there is a strong relationship between $h_o(n)$ and $h_e(n)$.

If $h(n)$ is absolutely summable (i.e., BIBO stable), the frequency response $H(\omega)$ exists, and

$$H(\omega) = H_R(\omega) + jH_I(\omega) \tag{10.1.9}$$

In addition, if $h(n)$ is real valued and causal, the symmetry properties of the Fourier transform imply that

$$\begin{aligned} h_e(n) &\overset{F}{\longleftrightarrow} H_R(\omega) \\ h_o(n) &\overset{F}{\longleftrightarrow} H_I(\omega) \end{aligned} \tag{10.1.10}$$

Since $h(n)$ is completely specified by $h_e(n)$, it follows that $H(\omega)$ is completely determined if we know $H_R(\omega)$. Alternatively, $H(\omega)$ is completely determined from $H_I(\omega)$ and $h(0)$. In short, $H_R(\omega)$ and $H_I(\omega)$ are interdependent and cannot be specified independently if the system is causal. Equivalently, the magnitude and phase responses of a causal filter are interdependent and hence cannot be specified independently.

Given $H_R(\omega)$ for a corresponding real, even, and absolutely summable sequence $h_e(n)$, we can determine $H(\omega)$. The following example illustrates the procedure.

EXAMPLE 10.1.1

Consider a stable LTI system with real and even impulse response $h(n)$. Determine $H(\omega)$ if

$$H_R(\omega) = \frac{1 - a\cos\omega}{1 - 2a\cos\omega + a^2}, \qquad |a| < 1$$

Solution. The first step is to determine $h_e(n)$. This can be done by noting that

$$H_R(\omega) = H_R(z)|_{z=e^{j\omega}}$$

where

$$H_R(z) = \frac{1 - a(z + z^{-1})/2}{1 - a(z + z^{-1}) + a^2} = \frac{z - a(z^2 + 1)/2}{(z - a)(1 - az)}$$

The ROC has to be restricted by the poles at $p_1 = a$ and $p_2 = 1/a$ and should include the unit circle. Hence the ROC is $|a| < |z| < 1/|a|$. Consequently, $h_e(n)$ is a two-sided sequence, with the pole at $z = a$ contributing to the causal part and $p_2 = 1/a$ contributing to the anticausal part. By using a partial-fraction expansion, we obtain

$$h_e(n) = \tfrac{1}{2}a^{|n|} + \tfrac{1}{2}\delta(n) \tag{10.1.11}$$

By substituting (10.1.11) into (10.1.7), we obtain $h(n)$ as

$$h(n) = a^n u(n)$$

Finally, we obtain the Fourier transform of $h(n)$ as

$$H(\omega) = \frac{1}{1 - ae^{-j\omega}}$$

The relationship between the real and imaginary components of the Fourier transform of an absolutely summable, causal, and real sequence can be easily established from (10.1.7). The Fourier transform relationship for (10.1.7) is

$$H(\omega) = H_R(\omega) + jH_I(\omega) = \frac{1}{\pi}\int_{-\pi}^{\pi} H_R(\lambda)U(\omega - \lambda)d\lambda - h_e(0) \tag{10.1.12}$$

where $U(\omega)$ is the Fourier transform of the unit step sequence $u(n)$. Although the unit step sequence is not absolutely summable, it has a Fourier transform (see Section 4.2.8).

$$\begin{aligned} U(\omega) &= \pi\delta(\omega) + \frac{1}{1 - e^{-j\omega}} \\ &= \pi\delta(\omega) + \frac{1}{2} - j\frac{1}{2}\cot\frac{\omega}{2}, \qquad -\pi \le \omega \le \pi \end{aligned} \tag{10.1.13}$$

By substituting (10.1.13) into (10.1.12) and carrying out the integration, we obtain the relation between $H_R(\omega)$ and $H_I(\omega)$ as

$$H_I(\omega) = -\frac{1}{2\pi}\int_{-\pi}^{\pi} H_R(\lambda)\cot\frac{\omega - \lambda}{2}d\lambda \tag{10.1.14}$$

Thus $H_I(\omega)$ is uniquely determined from $H_R(\omega)$ through this integral relationship. The integral is called a *discrete Hilbert transform*. It is left as an exercise to the reader to establish the relationship for $H_R(\omega)$ in terms of the discrete Hilbert transform of $H_I(\omega)$.

To summarize, causality has very important implications in the design of frequency-selective filters. These are: (a) the frequency response $H(\omega)$ cannot be zero, except at a finite set of points in frequency; (b) the magnitude $|H(\omega)|$ cannot be constant in any finite range of frequencies and the transition from passband to stopband cannot be infinitely sharp [this is a consequence of the Gibbs phenomenon, which results from the truncation of $h(n)$ to achieve causality]; and (c) the real and imaginary parts of $H(\omega)$ are interdependent and are related by the discrete Hilbert transform. As a consequence, the magnitude $|H(\omega)|$ and phase $\Theta(\omega)$ of $H(\omega)$ cannot be chosen arbitrarily.

Now that we know the restrictions that causality imposes on the frequency response characteristic and the fact that ideal filters are not achievable in practice, we limit our attention to the class of linear time-invariant systems specified by the difference equation

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M-1} b_k x(n-k)$$

which are causal and physically realizable. As we have demonstrated, such systems have a frequency response

$$H(\omega) = \frac{\sum_{k=0}^{M-1} b_k e^{-j\omega k}}{1 + \sum_{k=1}^{N} a_k e^{-j\omega k}} \tag{10.1.15}$$

The basic digital filter design problem is to approximate any of the ideal frequency response characteristics with a system that has the frequency response (10.1.15), by properly selecting the coefficients $\{a_k\}$ and $\{b_k\}$. The approximation problem is treated in detail in Sections 10.2 and 10.3, where we discuss techniques for digital filter design.

10.1.2 Characteristics of Practical Frequency-Selective Filters

As we observed from our discussion of the preceding section, ideal filters are noncausal and hence physically unrealizable for real-time signal processing applications. Causality implies that the frequency response characteristic $H(\omega)$ of the filter cannot be zero, except at a finite set of points in the frequency range. In addition, $H(\omega)$ cannot have an infinitely sharp cutoff from passband to stopband, that is, $H(\omega)$ cannot drop from unity to zero abruptly.

Although the frequency response characteristics possessed by ideal filters may be desirable, they are not absolutely necessary in most practical applications. If we relax these conditions, it is possible to realize causal filters that approximate the ideal filters as closely as we desire. In particular, it is not necessary to insist that the magnitude $|H(\omega)|$ be constant in the entire passband of the filter. A small amount of ripple in the passband, as illustrated in Fig. 10.1.2, is usually tolerable. Similarly,

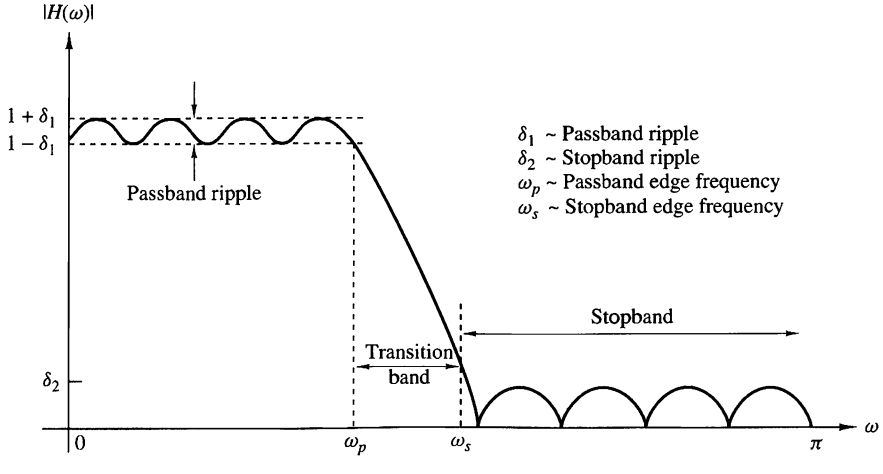


Figure 10.1.2 Magnitude characteristics of physically realizable filters.

it is not necessary for the filter response $|H(\omega)$ to be zero in the stopband. A small, nonzero value or a small amount of ripple in the stopband is also tolerable.

The transition of the frequency response from passband to stopband defines the *transition band* or *transition region* of the filter, as illustrated in Fig. 10.1.2. The band-edge frequency ω_p defines the edge of the passband, while the frequency ω_s denotes the beginning of the stopband. Thus the width of the transition band is $\omega_s - \omega_p$. The width of the passband is usually called the *bandwidth* of the filter. For example, if the filter is lowpass with a passband edge frequency ω_p, its bandwidth is ω_p.

If there is ripple in the passband of the filter, its value is denoted as δ_1, and the magnitude $|H(\omega)|$ varies between the limits $1 \pm \delta_1$. The ripple in the stopband of the filter is denoted as δ_2.

To accommodate a large dynamic range in the graph of the frequency response of any filter, it is common practice to use a logarithmic scale for the magnitude $|H(\omega)|$. Consequently, the ripple in the passband is $20 \log_{10} \delta_1$ decibels, and that in the stopband is $20 \log_{10} \delta_2$.

In any filter design problem we can specify (1) the maximum tolerable passband ripple, (2) the maximum tolerable stopband ripple, (3) the passband edge frequency ω_p, and (4) the stopband edge frequency ω_s. Based on these specifications, we can select the parameters $\{a_k\}$ and $\{b_k\}$ in the frequency response characteristic, given by (10.1.15), which best approximate the desired specification. The degree to which $H(\omega)$ approximates the specifications depends in part on the criterion used in the selection of the filter coefficients $\{a_k\}$ and $\{b_k\}$ as well as on the numbers (M, N) of coefficients.

In the following section we present a method for designing linear-phase FIR filters.

10.2 Design of FIR Filters

In this section we describe several methods for designing FIR filters. Our treatment is focused on the important class of linear-phase FIR filters.

10.2.1 Symmetric and Antisymmetric FIR Filters

An FIR filter of length M with input $x(n)$ and output $y(n)$ is described by the difference equation

$$\begin{aligned} y(n) &= b_0 x(n) + b_1 x(n-1) + \cdots + b_{M-1} x(n-M+1) \\ &= \sum_{k=0}^{M-1} b_k x(n-k) \end{aligned} \tag{10.2.1}$$

where $\{b_k\}$ is the set of filter coefficients. Alternatively, we can express the output sequence as the convolution of the unit sample response $h(n)$ of the system with the input signal. Thus we have

$$y(n) = \sum_{k=0}^{M-1} h(k) x(n-k) \tag{10.2.2}$$

where the lower and upper limits on the convolution sum reflect the causality and finite-duration characteristics of the filter. Clearly, (10.2.1) and (10.2.2) are identical in form and hence it follows that $b_k = h(k)$, $k = 0, 1, \ldots, M-1$.

The filter can also be characterized by its system function

$$H(z) = \sum_{k=0}^{M-1} h(k) z^{-k} \tag{10.2.3}$$

which we view as a polynomial of degree $M-1$ in the variable z^{-1}. The roots of this polynomial constitute the zeros of the filter.

An FIR filter has linear phase if its unit sample response satisfies the condition

$$h(n) = \pm h(M-1-n), \qquad n = 0, 1, \ldots, M-1 \tag{10.2.4}$$

When the symmetry and antisymmetry conditions in (10.2.4) are incorporated into (10.2.3), we have

$$\begin{aligned} H(z) &= h(0) + h(1)z^{-1} + h(2)z^{-2} + \cdots + h(M-2)z^{-(M-2)} + h(M-1)z^{-(M-1)} \\ &= z^{-(M-1)/2} \left\{ h\left(\frac{M-1}{2}\right) + \sum_{n=0}^{(M-3)/2} h(n) \left[z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2} \right] \right\}, \quad M \text{ odd} \\ &= z^{-(M-1)/2} \sum_{n=0}^{(M/2)-1} h(n) [z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2}], \qquad M \text{ even} \end{aligned} \tag{10.2.5}$$

Now, if we substitute z^{-1} for z in (10.2.3) and multiply both sides of the resulting equation by $z^{-(M-1)}$, we obtain

$$z^{-(M-1)} H(z^{-1}) = \pm H(z) \tag{10.2.6}$$

This result implies that the roots of the polynomial $H(z)$ are identical to the roots of the polynomial $H(z^{-1})$. Consequently, the roots of $H(z)$ must occur in reciprocal pairs. In other words, if z_1 is a root or a zero of $H(z)$, then $1/z_1$ is also a root. Furthermore, if the unit sample response $h(n)$ of the filter is real, complex-valued roots must occur in complex-conjugate pairs. Hence, if z_1 is a complex-valued root, z_1^* is also a root. As a consequence of (10.2.6), $H(z)$ also has a zero at $1/z_1^*$. Figure 10.2.1 illustrates the symmetry that exists in the location of the zeros of a linear-phase FIR filter.

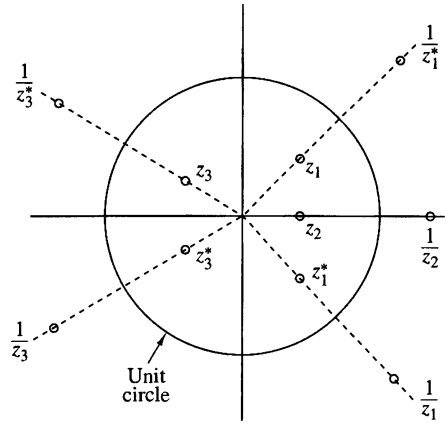


Figure 10.2.1
Symmetry of zero locations for a linear-phase FIR filter.

The frequency response characteristics of linear-phase FIR filters are obtained by evaluating (10.2.5) on the unit circle. This substitution yields the expression for $H(\omega)$.

When $h(n) = h(M-1-n)$, $H(\omega)$ can be expressed as

$$H(\omega) = H_r(\omega)e^{-j\omega(M-1)/2} \tag{10.2.7}$$

where $H_r(\omega)$ is a real function of ω and can be expressed as

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n)\cos\omega\left(\frac{M-1}{2} - n\right), \quad M \text{ odd} \tag{10.2.8}$$

$$H_r(\omega) = 2\sum_{n=0}^{(M/2)-1} h(n)\cos\omega\left(\frac{M-1}{2} - n\right), \quad M \text{ even} \tag{10.2.9}$$

The phase characteristic of the filter for both M odd and M even is

$$\Theta(\omega) = \begin{cases} -\omega\left(\dfrac{M-1}{2}\right), & \text{if } H_r(\omega) > 0 \\ -\omega\left(\dfrac{M-1}{2}\right) + \pi, & \text{if } H_r(\omega) < 0 \end{cases} \tag{10.2.10}$$

When

$$h(n) = -h(M-1-n)$$

the unit sample response is *antisymmetric*. For M odd, the center point of the antisymmetric $h(n)$ is $n = (M-1)/2$. Consequently,

$$h\left(\frac{M-1}{2}\right) = 0$$

However, if M is even, each term in $h(n)$ has a matching term of opposite sign.

It is straightforward to show that the frequency response of an FIR filter with an antisymmetric unit sample response can be expressed as

$$H(\omega) = H_r(\omega)e^{j[-\omega(M-1)/2+\pi/2]} \tag{10.2.11}$$

where

$$H_r(\omega) = 2\sum_{n=0}^{(M-3)/2} h(n)\sin\omega\left(\frac{M-1}{2}-n\right), \qquad M \text{ odd} \tag{10.2.12}$$

$$H_r(\omega) = 2\sum_{n=0}^{(M/2)-1} h(n)\sin\omega\left(\frac{M-1}{2}-n\right), \qquad M \text{ even} \tag{10.2.13}$$

The phase characteristic of the filter for both M odd and M even is

$$\Theta(\omega) = \begin{cases} \dfrac{\pi}{2} - \omega\left(\dfrac{M-1}{2}\right), & \text{if } H_r(\omega) > 0 \\ \dfrac{3\pi}{2} - \omega\left(\dfrac{M-1}{2}\right), & \text{if } H_r(\omega) < 0 \end{cases} \tag{10.2.14}$$

These general frequency response formulas can be used to design linear-phase FIR filters with symmetric and antisymmetric unit sample responses. We note that, for a symmetric $h(n)$, the number of filter coefficients that specify the frequency response is $(M+1)/2$ when M is odd or $M/2$ when M is even. On the other hand, if the unit sample response is antisymmetric,

$$h\left(\frac{M-1}{2}\right) = 0$$

so that there are $(M-1)/2$ filter coefficients when M is odd and $M/2$ coefficients when M is even to be specified.

The choice of a symmetric or antisymmetric unit sample response depends on the application. As we shall see later, a symmetric unit sample response is suitable for some applications, while an antisymmetric unit sample response is more suitable for other applications. For example, if $h(n) = -h(M-1-n)$ and M is odd, (10.2.12) implies that $H_r(0) = 0$ and $H_r(\pi) = 0$. Consequently, (10.2.12) is not suitable as either a lowpass filter or a highpass filter. Similarly, the antisymmetric unit sample response with M even also results in $H_r(0) = 0$, as can be easily verified from (10.2.13). Consequently, we would not use the antisymmetric condition in the design of a lowpass linear-phase FIR filter. On the other hand, the symmetry condition $h(n) = h(M-1-n)$ yields a linear-phase FIR filter with a nonzero response at $\omega = 0$, if desired, that is,

$$H_r(0) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n), \qquad M \text{ odd} \tag{10.2.15}$$

$$H_r(0) = 2\sum_{n=0}^{(M/2)-1} h(n), \qquad M \text{ even} \tag{10.2.16}$$

In summary, the problem of FIR filter design is simply to determine the M coefficients $h(n)$, $n = 0, 1, \ldots, M-1$, from a specification of the desired frequency response $H_d(\omega)$ of the FIR filter. The important parameters in the specification of $H_d(\omega)$ are given in Fig. 10.1.2.

In the following subsections we describe design methods based on specification of $H_d(\omega)$.

10.2.2 Design of Linear-Phase FIR Filters Using Windows

In this method we begin with the desired frequency response specification $H_d(\omega)$ and determine the corresponding unit sample response $h_d(n)$. Indeed, $h_d(n)$ is related to $H_d(\omega)$ by the Fourier transform relation

$$H_d(\omega) = \sum_{n=0}^{\infty} h_d(n) e^{-j\omega n} \tag{10.2.17}$$

where

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n} d\omega \tag{10.2.18}$$

Thus, given $H_d(\omega)$, we can determine the unit sample response $h_d(n)$ by evaluating the integral in (10.2.17).

In general, the unit sample response $h_d(n)$ obtained from (10.2.17) is infinite in duration and must be truncated at some point, say at $n = M-1$, to yield an FIR filter of length M. Truncation of $h_d(n)$ to a length $M-1$ is equivalent to multiplying $h_d(n)$ by a "rectangular window," defined as

$$w(n) = \begin{cases} 1, & n = 0, 1, \ldots, M-1 \\ 0, & \text{otherwise} \end{cases} \tag{10.2.19}$$

Thus the unit sample response of the FIR filter becomes

$$\begin{aligned} h(n) &= h_d(n) w(n) \\ &= \begin{cases} h_d(n), & n = 0, 1, \ldots, M-1 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{10.2.20}$$

It is instructive to consider the effect of the window function on the desired frequency response $H_d(\omega)$. Recall that multiplication of the window function $w(n)$ with $h_d(n)$ is equivalent to convolution of $H_d(\omega)$ with $W(\omega)$, where $W(\omega)$ is the frequency-domain representation (Fourier transform) of the window function, that is,

$$W(\omega) = \sum_{n=0}^{M-1} w(n) e^{-j\omega n} \tag{10.2.21}$$

Thus the convolution of $H_d(\omega)$ with $W(\omega)$ yields the frequency response of the (truncated) FIR filter. That is,

$$H(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\nu)W(\omega - \nu)d\nu \tag{10.2.22}$$

The Fourier transform of the rectangular window is

$$\begin{aligned} W(\omega) &= \sum_{n=0}^{M-1} e^{-j\omega n} \\ &= \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}} = e^{-j\omega(M-1)/2}\,\frac{\sin(\omega M/2)}{\sin(\omega/2)} \end{aligned} \tag{10.2.23}$$

This window function has a magnitude response

$$|W(\omega)| = \frac{|\sin(\omega M/2)|}{|\sin(\omega/2)|}, \qquad -\pi \le \omega \le \pi \tag{10.2.24}$$

and a piecewise linear phase

$$\Theta(\omega) = \begin{cases} -\omega\left(\dfrac{M-1}{2}\right), & \text{when } \sin(\omega M/2) \ge 0 \\ -\omega\left(\dfrac{M-1}{2}\right) + \pi, & \text{when } \sin(\omega M/2) < 0 \end{cases} \tag{10.2.25}$$

The magnitude response of the window function is illustrated in Fig. 10.2.2 for $M = 31$ and 61. The width of the main lobe [width is measured to the first zero of $W(\omega)$] is $4\pi/M$. Hence, as M increases, the main lobe becomes narrower. However, the sidelobes of $|W(\omega)|$ are relatively high and remain unaffected by an increase in M. In fact, even though the width of each sidelobe decreases with an increase in M, the height of each sidelobe increases with an increase in M in such a manner that the area under each sidelobe remains invariant to changes in M. This characteristic behavior is not evident from observation of Fig. 10.2.2 because $W(\omega)$ has been normalized by M such that the normalized peak values of the sidelobes remain invariant to an increase in M.

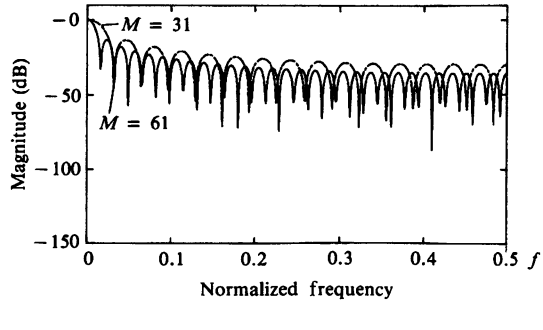


Figure 10.2.2
Frequency response for rectangular window of lengths (a) $M = 31$, (b) $M = 61$.

The characteristics of the rectangular window play a significant role in determining the resulting frequency response of the FIR filter obtained by truncating $h_d(n)$ to length M. Specifically, the convolution of $H_d(\omega)$ with $W(\omega)$ has the effect of smoothing $H_d(\omega)$. As M is increased, $W(\omega)$ becomes narrower, and the smoothing provided by $W(\omega)$ is reduced. On the other hand, the large sidelobes of $W(\omega)$ result in some undesirable ringing effects in the FIR filter frequency response $H(\omega)$, and also in relatively larger sidelobes in $H(\omega)$. These undesirable effects are best alleviated by the use of windows that do not contain abrupt discontinuities in their time-domain characteristics, and have correspondingly low sidelobes in their frequency-domain characteristics.

Table 10.1 lists several window functions that possess desirable frequency response characteristics. Figure 10.2.3 illustrates the time-domain characteristics of the windows. The frequency response characteristics of the Hanning, Hamming, and Blackman windows are illustrated in Figs. 10.2.4 through 10.2.6. All of these window

TABLE 10.1 Window Functions for FIR Filter Design

Name of window	Time-domain sequence, $h(n), 0 \le n \le M-1$
Bartlett (triangular)	$1 - \dfrac{2\left\lvert n - \dfrac{M-1}{2}\right\rvert}{M-1}$
Blackman	$0.42 - 0.5\cos\dfrac{2\pi n}{M-1} + 0.08\cos\dfrac{4\pi n}{M-1}$
Hamming	$0.54 - 0.46\cos\dfrac{2\pi n}{M-1}$
Hanning	$\dfrac{1}{2}\left(1 - \cos\dfrac{2\pi n}{M-1}\right)$
Kaiser	$\dfrac{I_0\left[\alpha\sqrt{\left(\dfrac{M-1}{2}\right)^2 - \left(n - \dfrac{M-1}{2}\right)^2}\right]}{I_0\left[\alpha\left(\dfrac{M-1}{2}\right)\right]}$
Lanczos	$\left\{\dfrac{\sin\left[2\pi\left(n - \dfrac{M-1}{2}\right)\Big/(M-1)\right]}{2\pi\left(n - \dfrac{M-1}{2}\right)\Big/\left(\dfrac{M-1}{2}\right)}\right\}^L, \quad L > 0$
	$1, \quad \left\lvert n - \dfrac{M-1}{2}\right\rvert \le \alpha\dfrac{M-1}{2}, \qquad 0 < \alpha < 1$
Tukey	$\dfrac{1}{2}\left[1 + \cos\left(\dfrac{n - (1+a)(M-1)/2}{(1-\alpha)(M-1)/2}\pi\right)\right]$
	$\alpha(M-1)/2 \le \left\lvert n - \dfrac{M-1}{2}\right\rvert \le \dfrac{M-1}{2}$

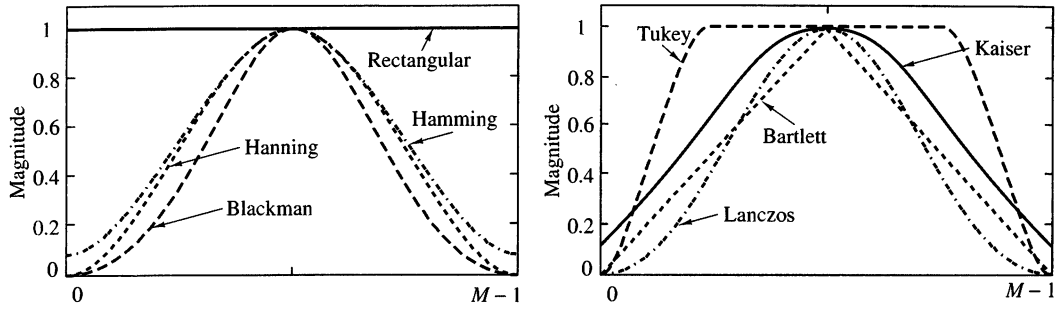


Figure 10.2.3 Shapes of several window functions.

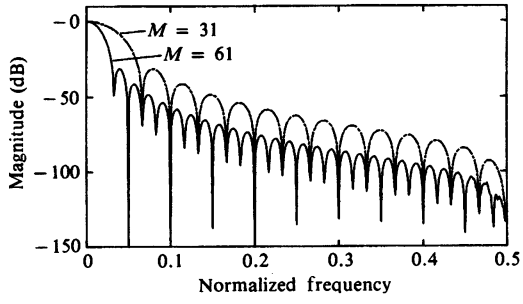


Figure 10.2.4
Frequency responses of Hanning window for (a) $M = 31$ and (b) $M = 61$.

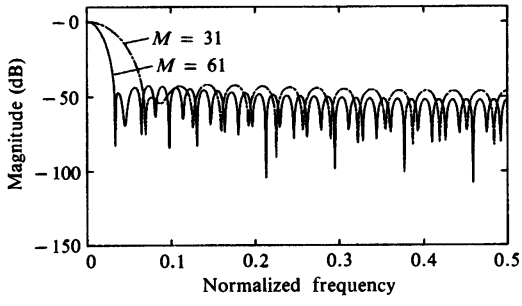


Figure 10.2.5
Frequency responses for Hamming window for (a) $M = 31$ and (b) $M = 61$.

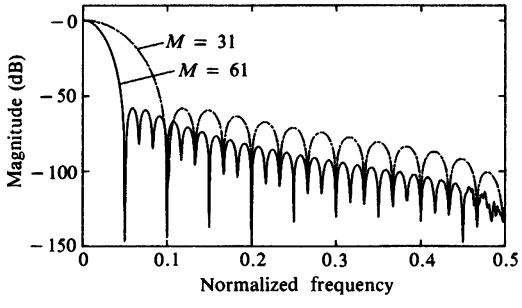


Figure 10.2.6
Frequency responses for Blackman window for (a) $M = 31$ and (b) $M = 61$.

functions have significantly lower sidelobes compared with the rectangular window. However, for the same value of M, the width of the main lobe is also wider for these windows compared to the rectangular window. Consequently, these window functions provide more smoothing through the convolution operation in the frequency domain, and as a result, the transition region in the FIR filter response is wider. To reduce the width of this transition region, we can simply increase the length of the window, which results in a larger filter. Table 10.2 summarizes these important frequency-domain features of the various window functions.

The window technique is best described in terms of a specific example. Suppose that we want to design a symmetric lowpass linear-phase FIR filter having a desired frequency response

$$H_d(\omega) = \begin{cases} 1e^{-j\omega(M-1)/2}, & 0 \le |\omega| \le \omega_c \\ 0, & \text{otherwise} \end{cases} \tag{10.2.26}$$

A delay of $(M-1)/2$ units is incorporated into $H_d(\omega)$ in anticipation of forcing the filter to be of length M. The corresponding unit sample response, obtained by evaluating the integral in (10.2.18), is

$$\begin{aligned} h_d(n) &= \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega\left(n-\frac{M-1}{2}\right)}d\omega \\ &= \frac{\sin\omega_c\left(n-\dfrac{M-1}{2}\right)}{\pi\left(n-\dfrac{M-1}{2}\right)}, \qquad n \neq \frac{M-1}{2} \end{aligned} \tag{10.2.27}$$

Clearly, $h_d(n)$ is noncausal and infinite in duration.

If we multiply $h_d(n)$ by the rectangular window sequence in (10.2.19), we obtain

TABLE 10.2 Important Frequency-Domain Characteristics of Some Window Functions

Type of window	Approximate transition width of main lobe	Peak sidelobe (dB)
Rectangular	$4\pi/M$	−13
Bartlett	$8\pi/M$	−25
Hanning	$8\pi/M$	−31
Hamming	$8\pi/M$	−41
Blackman	$12\pi/M$	−57

an FIR filter of length M having the unit sample response

$$h(n) = \frac{\sin \omega_c \left(n - \dfrac{M-1}{2}\right)}{\pi \left(n - \dfrac{M-1}{2}\right)}, \qquad 0 \le n \le M-1, \quad n \ne \frac{M-1}{2} \tag{10.2.28}$$

If M is selected to be odd, the value of $h(n)$ at $n = (M-1)/2$ is

$$h\left(\frac{M-1}{2}\right) = \frac{\omega_c}{\pi} \tag{10.2.29}$$

The magnitude of the frequency response $H(\omega)$ of this filter is illustrated in Fig. 10.2.7 for $M = 61$ and $M = 101$. We observe that relatively large oscillations or ripples occur near the band edge of the filter. The oscillations increase in frequency as M increases, but they do not diminish in amplitude. As indicated previously, these large oscillations are the direct result of the large sidelobes existing in the frequency characteristic $W(\omega)$ of the rectangular window. As this window function is convolved with the desired frequency response characteristic $H_d(\omega)$, the oscillations occur as the large constant-area sidelobes of $W(\omega)$ move across the discontinuity that exists in $H_d(\omega)$. Since (10.2.17) is basically a Fourier series representation of $H_d(\omega)$, the multiplication of $h_d(n)$ with a rectangular window is identical to truncating the Fourier series representation of the desired filter characteristic $H_d(\omega)$. The truncation of the Fourier series is known to introduce ripples in the frequency response characteristic $H(\omega)$ due to the nonuniform convergence of the Fourier series at a discontinuity. The oscillatory behavior near the band edge of the filter is called the *Gibbs phenomenon*.

To alleviate the presence of large oscillations in both the passband and the stopband, we should use a window function that contains a taper and decays toward zero gradually, instead of abruptly, as it occurs in a rectangular window. Figures 10.2.8 through 10.2.11 illustrate the frequency response of the resulting filter when some of the window functions listed in Table 10.1 are used to taper $h_d(n)$. As illustrated in Figs. 10.2.8 through 10.2.11, the window functions do indeed eliminate the ringing effects at the band edge and do result in lower sidelobes at the expense of an increase in the width of the transition band of the filter.

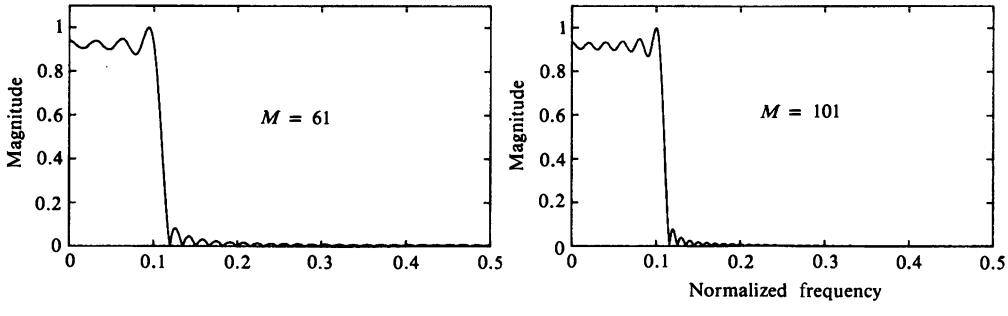


Figure 10.2.7 Lowpass filter designed with a rectangular window: (a) $M = 61$ and (b) $M = 101$.

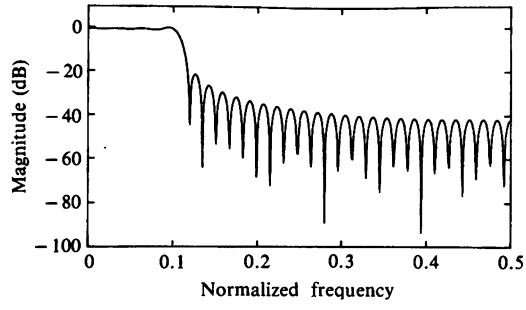


Figure 10.2.8
Lowpass FIR filter designed with rectangular window ($M = 61$).

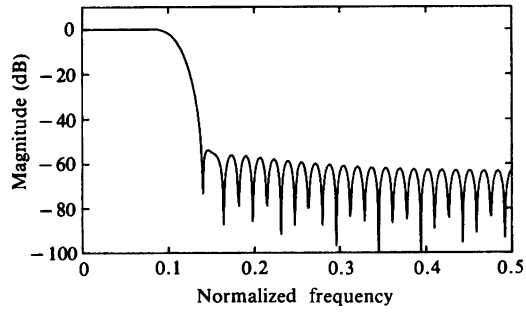


Figure 10.2.9
Lowpass FIR filter designed with Hamming window ($M = 61$).

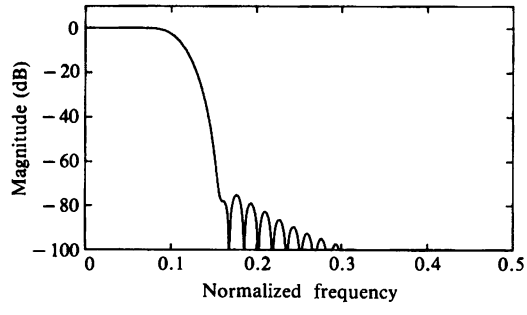


Figure 10.2.10
Lowpass FIR filter designed with Blackman window ($M = 61$).

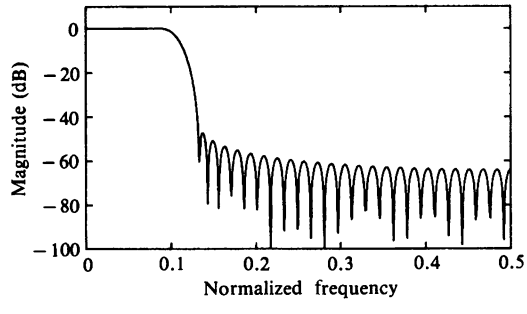


Figure 10.2.11
Lowpass FIR filter designed with $\alpha = 4$ Kaiser window ($M = 61$).

10.2.3 Design of Linear-Phase FIR Filters by the Frequency-Sampling Method

In the frequency-sampling method for FIR filter design, we specify the desired frequency response $H_d(\omega)$ at a set of equally spaced frequencies, namely

$$\omega_k = \frac{2\pi}{M}(k+\alpha), \qquad \begin{aligned} &k = 0, 1, \ldots, \frac{M-1}{2}, && M \text{ odd} \\ &k = 0, 1, \ldots, \frac{M}{2} - 1, && M \text{ even} \\ &\alpha = 0 \quad \text{or} \quad \tfrac{1}{2} \end{aligned} \tag{10.2.30}$$

and solve for the unit sample response $h(n)$ of the FIR filter from these equally spaced frequency specifications. To reduce sidelobes, it is desirable to optimize the frequency specification in the transition band of the filter. This optimization can be accomplished numerically on a digital computer by means of linear programming techniques as shown by Rabiner et al. (1970).

In this section we exploit a basic symmetry property of the sampled frequency response function to simplify the computations. Let us begin with the desired frequency response of the FIR filter, which is [for simplicity, we drop the subscript in $H_d(\omega)$],

$$H(\omega) = \sum_{n=0}^{M-1} h(n)e^{-j\omega n} \tag{10.2.31}$$

Suppose that we specify the frequency response of the filter at the frequencies given by (10.2.30). Then from (10.2.31) we obtain

$$H(k+\alpha) \equiv H\left(\frac{2\pi}{M}(k+\alpha)\right)$$

$$H(k+\alpha) \equiv \sum_{n=0}^{M-1} h(n)e^{-j2\pi(k+\alpha)n/M}, \qquad k = 0, 1, \ldots, M-1 \tag{10.2.32}$$

It is a simple matter to invert (10.2.32) and express $h(n)$ in terms of $H(k+\alpha)$. If we multiply both sides of (10.2.32) by the exponential, $\exp(j2\pi km/M)$, $m = 0, 1, \ldots, M-1$, and sum over $k = 0, 1, \ldots, M-1$, the right-hand side of (10.2.32) reduces to $Mh(m)\exp(-j2\pi\alpha m/M)$. Thus we obtain

$$h(n) = \frac{1}{M}\sum_{k=0}^{M-1} H(k+\alpha)e^{j2\pi(k+\alpha)n/M}, \qquad n = 0, 1, \ldots, M-1 \tag{10.2.33}$$

The relationship in (10.2.33) allows us to compute the values of the unit sample response $h(n)$ from the specification of the frequency samples $H(k+\alpha)$, $k = 0, 1, \ldots, M-1$. Note that when $\alpha = 0$, (10.2.32) reduces to the discrete Fourier

transform (DFT) of the sequence $\{h(n)\}$ and (10.2.33) reduces to the inverse DFT (IDFT).

Since $\{h(n)\}$ is real, we can easily show that the frequency samples $\{H(k+\alpha)\}$ satisfy the symmetry condition

$$H(k+\alpha) = H^*(M-k-\alpha) \tag{10.2.34}$$

This symmetry condition, along with the symmetry conditions for $\{h(n)\}$, can be used to reduce the frequency specifications from M points to $(M+1)/2$ points for M odd and $M/2$ points for M even. Thus the linear equations for determining $\{h(n)\}$ from $\{H(k+\alpha)\}$ are considerably simplified.

In particular, if (10.2.11) is sampled at the frequencies $\omega_k = 2\pi(k+\alpha)/M$, $k = 0, 1, \ldots, M-1$, we obtain

$$H(k+\alpha) = H_r\left(\frac{2\pi}{M}(k+\alpha)\right) e^{j[\beta\pi/2 - 2\pi(k+\alpha)(M-1)/2M]} \tag{10.2.35}$$

where $\beta = 0$ when $\{h(n)\}$ is symmetric and $\beta = 1$ when $\{h(n)\}$ is antisymmetric. A simplification occurs by defining a set of real frequency samples $\{G(k+\alpha)\}$

$$G(k+\alpha) = (-1)^k H_r\left(\frac{2\pi}{M}(k+\alpha)\right), \qquad k = 0, 1, \ldots, M-1 \tag{10.2.36}$$

We use (10.2.36) in (10.2.35) to eliminate $H_r(\omega_k)$. Thus we obtain

$$H(k+\alpha) = G(k+\alpha)e^{j\pi k} e^{j[\beta\pi/2 - 2\pi(k+\alpha)(M-1)/2M]} \tag{10.2.37}$$

Now the symmetry condition for $H(k+\alpha)$ given in (10.2.34) translates into a corresponding symmetry condition for $G(k+\alpha)$, which can be exploited by substituting into (10.2.33), to simplify the expressions for the FIR filter impulse response $\{h(n)\}$ for the four cases $\alpha = 0$, $\alpha = \frac{1}{2}$, $\beta = 0$, and $\beta = 1$. The results are summarized in Table 10.3. The detailed derivations are left as exercises for the reader.

Although the frequency sampling method provides us with another means for designing linear-phase FIR filters, its major advantage lies in the efficient frequency-sampling structure, which is obtained when most of the frequency samples are zero, as demonstrated in Section 9.2.3.

The following examples illustrate the design of linear-phase FIR filters based on the frequency-sampling method. The optimum values for the samples in the transition band are obtained from the tables in Appendix B, which are taken from the paper by Rabiner et al. (1970).

EXAMPLE 10.2.1

Determine the coefficients of a linear-phase FIR filter of length $M = 15$ which has a symmetric unit sample response and a frequency response that satisfies the conditions

$$H_r\left(\frac{2\pi k}{15}\right) = \begin{cases} 1, & k = 0, 1, 2, 3 \\ 0.4, & k = 4 \\ 0, & k = 5, 6, 7 \end{cases}$$

TABLE 10.3 Unit Sample Response: $h(n) = \pm h(M-1-n)$

	Symmetric
$\alpha = 0$	$H(k) = G(k)e^{j\pi k/M}, \quad k = 0, 1, \ldots, M-1$
	$G(k) = (-1)^k H_r\left(\frac{2\pi k}{M}\right), \quad G(k) = -G(M-k)$
	$h(n) = \frac{1}{M}\left\{G(0) + 2\sum_{k=1}^{U} G(k)\cos\frac{2\pi k}{M}(n+\frac{1}{2})\right\}$
	$U = \begin{cases} \frac{M-1}{2}, & M \text{ odd} \\ \frac{M}{2}-1, & M \text{ even} \end{cases}$
$\alpha = \frac{1}{2}$	$H\left(k+\frac{1}{2}\right) = G\left(k+\frac{1}{2}\right)e^{-j\pi/2}e^{j\pi(2k+1)/2M}$
	$G\left(k+\frac{1}{2}\right) = (-1)^k H_r\left[\frac{2\pi}{M}\left(k+\frac{1}{2}\right)\right]$
	$G\left(k+\frac{1}{2}\right) = G\left(M-k-\frac{1}{2}\right)$
	$h(n) = \frac{2}{M}\sum_{k=0}^{U} G\left(k+\frac{1}{2}\right)\sin\frac{2\pi}{M}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)$
	Antisymmetric
$\alpha = 0$	$H(k) = G(k)e^{j\pi/2}e^{j\pi k/M}, \quad k = 0, 1, \ldots, M-1$
	$G(k) = (-1)^k H_r\left(\frac{2\pi k}{M}\right), \quad G(k) = G(M-k)$
	$h(n) = -\frac{2}{M}\sum_{k=1}^{(M-1)/2} G(k)\sin\frac{2\pi k}{M}\left(n+\frac{1}{2}\right), \quad M \text{ odd}$
	$h(n) = \frac{1}{M}\left\{(-1)^{n+1}G(M/2) - 2\sum_{k=1}^{(M/2)-1} G(k)\sin\frac{2\pi}{M}k\left(n+\frac{1}{2}\right)\right\}, \quad M \text{ even}$
$\alpha = \frac{1}{2}$	$H\left(k+\frac{1}{2}\right) = G\left(k+\frac{1}{2}\right)e^{j\pi(2k+1)/2M}$
	$G\left(k+\frac{1}{2}\right) = (-1)^k H_r\left[\frac{2\pi}{M}\left(k+\frac{1}{2}\right)\right]$
	$G\left(k+\frac{1}{2}\right) = -G\left(M-k-\frac{1}{2}\right); \quad G(M/2) = 0 \text{ for } M \text{ odd}$
	$h(n) = \frac{2}{M}\sum_{k=0}^{V} G\left(k+\frac{1}{2}\right)\cos\frac{2\pi}{M}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)$
	$V = \begin{cases} \frac{M-3}{2}, & M \text{ odd} \\ \frac{M}{2}-1, & M \text{ even} \end{cases}$

Solution. Since $h(n)$ is symmetric and the frequencies are selected to correspond to the case $\alpha = 0$, we use the corresponding formula in Table 10.3 to evaluate $h(n)$. In this case

$$G(k) = (-1)^k H_r\left(\frac{2\pi k}{15}\right), \qquad k = 0, 1, \ldots, 7$$

The result of this computation is

$$\begin{aligned}
h(0) &= h(14) = -0.014112893 \\
h(1) &= h(13) = -0.001945309 \\
h(2) &= h(12) = 0.04000004 \\
h(3) &= h(11) = 0.01223454 \\
h(4) &= h(10) = -0.09138802 \\
h(5) &= h(9) \;\; = -0.01808986 \\
h(6) &= h(8) \;\; = 0.3133176 \\
h(7) &\qquad\;\; = 0.52
\end{aligned}$$

The frequency response characteristic of this filter is shown in Fig. 10.2.12. We should emphasize that $H_r(\omega)$ is exactly equal to the values given by the specifications above at $\omega_k = 2\pi k/15$.

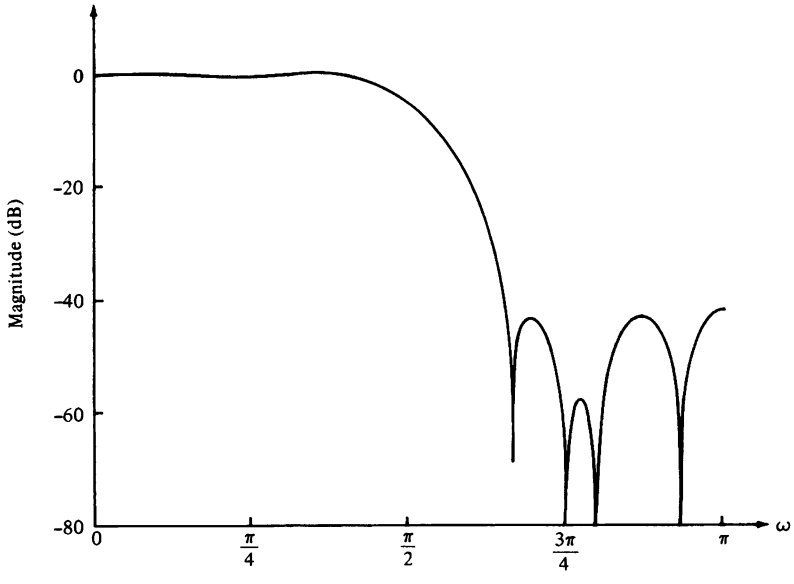


Figure 10.2.12 Frequency response of linear-phase FIR filter in Example 10.2.1.

EXAMPLE 10.2.2

Determine the coefficients of a linear-phase FIR filter of length $M = 32$ which has a symmetric unit sample response and a frequency response that satisfies the condition

$$H_r\left(\frac{2\pi(k+\alpha)}{32}\right) = \begin{cases} 1, & k = 0, 1, 2, 3, 4, 5 \\ T_1, & k = 6 \\ 0, & k = 7, 8, \ldots, 15 \end{cases}$$

where $T_1 = 0.3789795$ for $\alpha = 0$, and $T_1 = 0.3570496$ for $\alpha = \frac{1}{2}$. These values of T_1 were obtained from the tables of optimum transition parameters given in Appendix B.

Solution. The appropriate equations for this computation are given in Table 10.3 for $\alpha = 0$ and $\alpha = \frac{1}{2}$. These computations yield the unit sample responses shown in Table 10.4. The corresponding frequency response characteristics are illustrated in Figs. 10.2.13 and 10.2.14, respectively. Note that the bandwidth of the filter for $\alpha = \frac{1}{2}$ is wider than that for $\alpha = 0$.

TABLE 10.4

M = 32	M = 32
ALPHA = 0.	ALPHA = 0.5
T1 = 0.3789795E+00	T1 = 0.3570496E+00
h(0) = −0.7141978E-02	h(0) = −0.4089120E-02
h(1) = −0.3070801E-02	h(1) = −0.9973779E-02
h(2) = 0.5891327E-02	h(2) = −0.7379891E-02
h(3) = 0.1349923E-01	h(3) = 0.5949799E-02
h(4) = 0.8087033E-02	h(4) = 0.1727056E-01
h(5) = −0.1107258E-01	h(5) = 0.7878412E-02
h(6) = −0.2420687E-01	h(6) = −0.1798590E-01
h(7) = −0.9446550E-02	h(7) = −0.2670584E-01
h(8) = 0.2544464E-01	h(8) = 0.3778549E-02
h(9) = 0.3985050E-01	h(9) = 0.4191022E-01
h(10) = 0.2753036E-02	h(10) = 0.2839344E-01
h(11) = −0.5913959E-01	h(11) = −0.4163144E-01
h(12) = −0.6841660E-01	h(12) = −0.8254962E-01
h(13) = 0.3175741E-01	h(13) = 0.2802212E-02
h(14) = 0.2080981E+00	h(14) = 0.2013655E+00
h(15) = 0.3471138E+00	h(15) = 0.3717532E+00

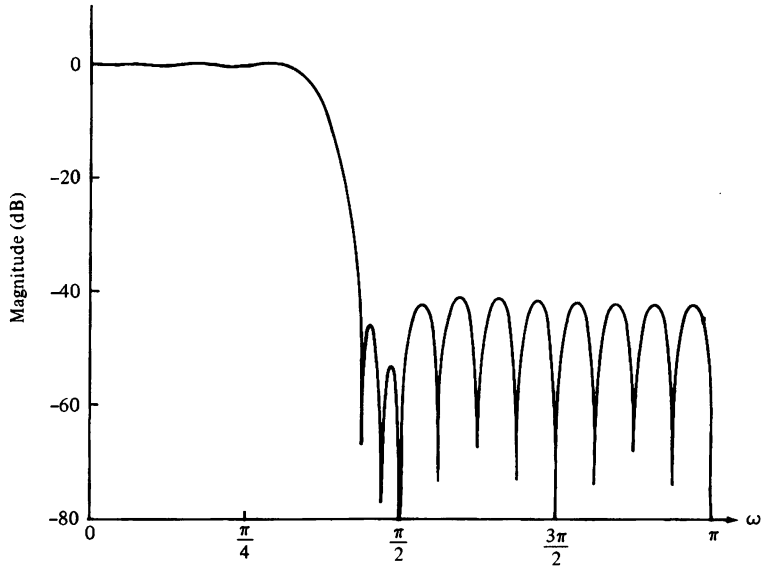


Figure 10.2.13 Frequency response of linear-phase FIR filter in Example 10.2.2 ($M = 32$ and $\alpha = 0$).

The optimization of the frequency samples in the transition region of the frequency response can be explained by evaluating the system function $H(z)$, given by (9.2.12), on the unit circle and using the relationship in (10.2.37) to express $H(\omega)$ in terms of $G(k + \alpha)$. Thus for the symmetric filter we obtain

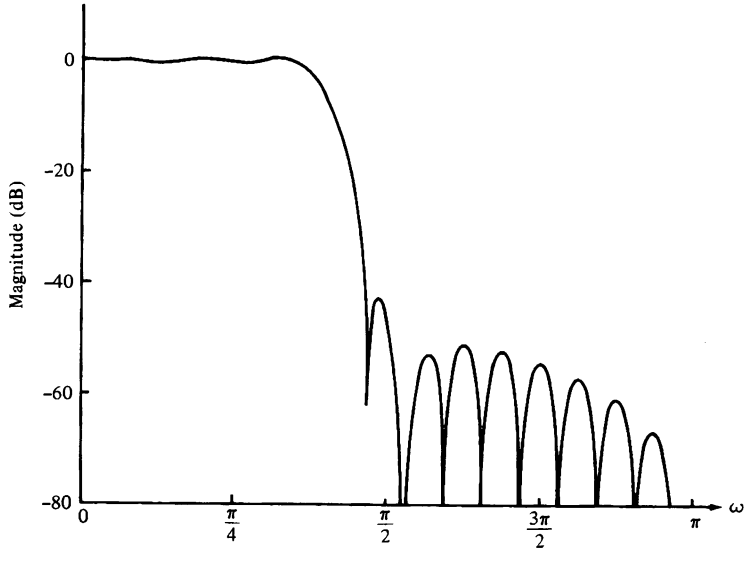


Figure 10.2.14 Frequency response of linear-phase FIR filter in Example 10.2.2 ($M = 32$ and $\alpha = \frac{1}{2}$).

$$H(\omega) = \left\{ \frac{\sin\left(\frac{\omega M}{2} - \pi\alpha\right)}{M} \sum_{k=0}^{M-1} \frac{G(k+\alpha)}{\sin\left[\frac{\omega}{2} - \frac{\pi}{M}(k+\alpha)\right]} \right\} e^{-j\omega(M-1)/2} \tag{10.2.38}$$

where

$$G(k+\alpha) = \begin{cases} -G(M-k), & \alpha = 0 \\ G(M-k-\frac{1}{2}), & \alpha = \frac{1}{2} \end{cases} \tag{10.2.39}$$

Similarly, for the antisymmetric linear-phase FIR filter we obtain

$$H(\omega) = \left\{ \frac{\sin\left(\frac{\omega M}{2} - \pi\alpha\right)}{M} \sum_{k=0}^{M-1} \frac{G(k+\alpha)}{\sin\left[\frac{\omega}{2} - \frac{\pi}{M}(k+\alpha)\right]} \right\} e^{-j\omega(M-1)/2} e^{j\pi/2} \tag{10.2.40}$$

where

$$G(k+\alpha) = \begin{cases} G(M-k), & \alpha = 0 \\ -G\left(M-k-\frac{1}{2}\right), & \alpha = \frac{1}{2} \end{cases} \tag{10.2.41}$$

With these expressions for the frequency response $H(\omega)$ given in terms of the desired frequency samples $\{G(k+\alpha)\}$, we can easily explain the method for selecting the parameters $\{G(k+\alpha)\}$ in the transition band which result in minimizing the peak sidelobe in the stopband. In brief, the values of $G(k+\alpha)$ in the passband are set to $(-1)^k$ and those in the stopband are set to zero. For any choice of $G(k+\alpha)$ in the transition band, the value of $H(\omega)$ is computed at a dense set of frequencies (e.g., at $\omega_n = 2\pi n/K$, $n = 0, 1, \ldots, K-1$, where, for example, $K = 10M$). The value of the maximum sidelobe is determined, and the values of the parameters $\{G(k+\alpha)\}$ in the transition band are changed in a direction of steepest descent, which, in effect, reduces the maximum sidelobe. The computation of $H(\omega)$ is now repeated with the new choice of $\{G(k+\alpha)\}$. The maximum sidelobe of $H(\omega)$ is again determined and the values of the parameters $\{G(k+\alpha)\}$ in the transition band are adjusted in a direction of steepest descent, which, in turn, reduces the sidelobe. This iterative process is performed until it converges to the optimum choice of the parameters $\{G(k+\alpha)\}$ in the transition band.

There is a potential problem in the frequency-sampling realization of the FIR linear-phase filter. The frequency-sampling realization of the FIR filter introduces poles and zeros at equally spaced points on the unit circle. In the ideal situation, the zeros cancel the poles and, consequently, the actual zeros of $H(z)$ are determined by the selection of the frequency samples $\{H(k+\alpha)\}$. In a practical implementation of the frequency-sampling realization, however, quantization effects preclude a perfect cancellation of the poles and zeros. In fact, the location of poles on the unit circle provide no damping of the round-off noise that is introduced in the computations. As a result, such noise tends to increase with time and, ultimately, may destroy the normal operation of the filter.

To mitigate this problem, we can move both the poles and zeros from the unit circle to a circle just inside the unit circle, say at radius $r = 1 - \epsilon$, where ϵ is a very small number. Thus the system function of the linear-phase FIR filter becomes

$$H(z) = \frac{1 - r^M z^{-M} e^{j2\pi\alpha}}{M} \sum_{k=0}^{M-1} \frac{H(k+\alpha)}{1 - re^{j2\omega\pi(k+\alpha)/M} z^{-1}} \tag{10.2.42}$$

The corresponding two-pole filter realization given in Section 9.2.3 can be modified accordingly. The damping provided by selecting $r < 1$ ensures that roundoff noise will be bounded and thus instability is avoided.

10.2.4 Design of Optimum Equiripple Linear-Phase FIR Filters

The window method and the frequency-sampling method are relatively simple techniques for designing linear-phase FIR filters. However, they also possess some minor disadvantages, described in Section 10.2.6, which may render them undesirable for some applications. A major problem is the lack of precise control of the critical frequencies such as ω_p and ω_s.

The filter design method described in this section is formulated as a Chebyshev approximation problem. It is viewed as an optimum design criterion in the sense that the weighted approximation error between the desired frequency response and the actual frequency response is spread evenly across the passband and evenly across the stopband of the filter minimizing the maximum error. The resulting filter designs have ripples in both the passband and the stopband.

To describe the design procedure, let us consider the design of a lowpass filter with passband edge frequency ω_p and stopband edge frequency ω_s. From the general specifications given in Fig. 10.1.2, in the passband, the filter frequency response satisfies the condition

$$1 - \delta_1 \le H_r(\omega) \le 1 + \delta_1, \qquad |\omega| \le \omega_p \tag{10.2.43}$$

Similarly, in the stopband, the filter frequency response is specified to fall between the limits $\pm\delta_2$, that is,

$$-\delta_2 \le H_r(\omega) \le \delta_2, \qquad |\omega| > \omega_s \tag{10.2.44}$$

Thus δ_1 represents the ripple in the passband and δ_2 represents the attenuation or ripple in the stopband. The remaining filter parameter is M, the filter length or the number of filter coefficients.

Let us focus on the four different cases that result in a linear-phase FIR filter. These cases were treated in Section 10.2.2 and are summarized below.

Case 1: Symmetric unit sample response. $h(n) = h(M-1-n)$ and M odd. In this case, the real-valued frequency response characteristic $H_r(\omega)$ is

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n)\cos\omega\left(\frac{M-1}{2} - n\right) \tag{10.2.45}$$

If we let $k = (M-1)/2 - n$ and define a new set of filter parameters $\{a(k)\}$ as

$$a(k) = \begin{cases} h\left(\dfrac{M-1}{2}\right), & k = 0 \\ 2h\left(\dfrac{M-1}{2} - k\right), & k = 1, 2, \ldots, \dfrac{M-1}{2} \end{cases} \tag{10.2.46}$$

then (10.2.45) reduces to the compact form

$$H_r(\omega) = \sum_{k=0}^{(M-1)/2} a(k) \cos \omega k \tag{10.2.47}$$

Case 2: Symmetric unit sample response. $h(n) = h(M-1-n)$ and M even. In this case, $H_r(\omega)$ is expressed as

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \cos \omega \left(\frac{M-1}{2} - n\right) \tag{10.2.48}$$

Again, we change the summation index from n to $k = M/2 - n$ and define a new set of filter parameters $\{b(k)\}$ as

$$b(k) = 2h\left(\frac{M}{2} - k\right), \qquad k = 1, 2, \ldots, M/2 \tag{10.2.49}$$

With these substitutions (10.2.48) becomes

$$H_r(\omega) = \sum_{k=1}^{M/2} b(k) \cos \omega \left(k - \frac{1}{2}\right) \tag{10.2.50}$$

In carrying out the optimization, it is convenient to rearrange (10.2.50) further into the form

$$H_r(\omega) = \cos \frac{\omega}{2} \sum_{k=0}^{(M/2)-1} \tilde{b}(k) \cos \omega k \tag{10.2.51}$$

where the coefficients $\{\tilde{b}(k)\}$ are linearly related to the coefficients $\{b(k)\}$. In fact, it can be shown that the relationship is

$$\begin{aligned} &\tilde{b}(0) = \tfrac{1}{2} b(1), \qquad \tilde{b}(1) = 2b(1) - 2b(0) \\ &\tilde{b}(k) = 2b(k) - \tilde{b}(k-1), \qquad k = 1, 2, 3, \ldots, \frac{M}{2} - 2 \\ &\tilde{b}\left(\frac{M}{2} - 1\right) = 2b\left(\frac{M}{2}\right) \end{aligned} \tag{10.2.52}$$

Case 3: Antisymmetric unit sample response. $h(n) = -h(M-1-n)$ **and** M **odd.** The real-valued frequency response characteristic $H_r(\omega)$ for this case is

$$H_r(\omega) = 2 \sum_{n=0}^{(M-3)/2} h(n) \sin \omega \left(\frac{M-1}{2} - n \right) \tag{10.2.53}$$

If we change the summation in (10.2.53) from n to $k = (M-1)/2 - n$ and define a new set of filter parameters $\{c(k)\}$ as

$$c(k) = 2h\left(\frac{M-1}{2} - k \right), \qquad k = 1, 2, \ldots, (M-1)/2 \tag{10.2.54}$$

then (10.2.53) becomes

$$H_r(\omega) = \sum_{k=1}^{(M-1)/2} c(k) \sin \omega k \tag{10.2.55}$$

As in the previous case, it is convenient to rearrange (10.2.55) into the form

$$H_r(\omega) = \sin \omega \sum_{k=0}^{(M-3)/2} \tilde{c}(k) \cos \omega k \tag{10.2.56}$$

where the coefficients $\{\tilde{c}(k)\}$ are linearly related to the parameters $\{c(k)\}$. This desired relationship can be derived from (10.2.55) and (10.2.56) and is simply given as

$$\begin{aligned}
\tilde{c}\left(\frac{M-3}{2} \right) &= c\left(\frac{M-1}{2} \right) \\
\tilde{c}\left(\frac{M-5}{2} \right) &= 2c\left(\frac{M-3}{2} \right) \\
&\vdots \qquad\qquad \vdots \\
\tilde{c}(k-1) - \tilde{c}(k+1) &= 2c(k), \qquad 2 \le k \le \frac{M-5}{2} \\
\tilde{c}(0) - \tfrac{1}{2}\tilde{c}(2) &= c(1)
\end{aligned} \tag{10.2.57}$$

Case 4: Antisymmetric unit sample response. $h(n) = -h(M-1-n)$ **and** M **even.** In this case, the real-valued frequency response characteristic $H_r(\omega)$ is

$$H_r(\omega) = 2 \sum_{n=0}^{(M/2)-1} h(n) \sin \omega \left(\frac{M-1}{2} - n \right) \tag{10.2.58}$$

A change in the summation index from n to $k = M/2 - n$ combined with a definition of a new set of filter coefficients $\{d(k)\}$, related to $\{h(n)\}$ according to

$$d(k) = 2h\left(\frac{M}{2} - k\right), \qquad k = 1, 2, \ldots, \frac{M}{2} \tag{10.2.59}$$

results in the expression

$$H_r(\omega) = \sum_{k=1}^{M/2} d(k) \sin \omega \left(k - \frac{1}{2}\right) \tag{10.2.60}$$

As in the previous two cases, we find it convenient to rearrange (10.2.60) into the form

$$H_r(\omega) = \sin \frac{\omega}{2} \sum_{k=0}^{(M/2)-1} \tilde{d}(k) \cos \omega k \tag{10.2.61}$$

where the new filter parameters $\{\tilde{d}(k)\}$ are related to $\{d(k)\}$ as follows:

$$\begin{aligned} \tilde{d}\left(\frac{M}{2} - 1\right) &= 2d\left(\frac{M}{2}\right) \\ \tilde{d}(k-1) - \tilde{d}(k) &= 2d(k), \qquad 2 \le k \le \frac{M}{2} - 1 \\ \tilde{d}(0) - \tfrac{1}{2}\tilde{d}(1) &= d(1) \end{aligned} \tag{10.2.62}$$

TABLE 10.5 Real-Valued Frequency Response Functions for Linear-Phase FIR Filters

Filter type	$Q(\omega)$	$P(\omega)$
$h(n) = h(M-1-n)$ M odd (Case 1)	1	$\sum_{k=0}^{(M-1)/2} a(k) \cos \omega k$
$h(n) = h(M-1-n)$ M even (Case 2)	$\cos \frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{b}(k) \cos \omega k$
$h(n) = -h(M-1-n)$ M odd (Case 3)	$\sin \omega$	$\sum_{k=0}^{(M-3)/2} \tilde{c}(k) \cos \omega k$
$h(n) = -h(M-1-n)$ M even (Case 4)	$\sin \frac{\omega}{2}$	$\sum_{k=0}^{(M/2)-1} \tilde{d}(k) \cos \omega k$

The expressions for $H_r(\omega)$ in these four cases are summarized in Table 10.5. We note that the rearrangements that we made in Cases 2, 3, and 4 have allowed us to express $H_r(\omega)$ as

$$H_r(\omega) = Q(\omega)P(\omega) \tag{10.2.63}$$

where

$$Q(\omega) = \begin{cases} 1 & \text{Case 1} \\ \cos\dfrac{\omega}{2} & \text{Case 2} \\ \sin\omega & \text{Case 3} \\ \sin\dfrac{\omega}{2} & \text{Case 4} \end{cases} \tag{10.2.64}$$

and $P(\omega)$ has the common form

$$P(\omega) = \sum_{k=0}^{L} \alpha(k)\cos\omega k \tag{10.2.65}$$

with $\{\alpha(k)\}$ representing the parameters of the filter, which are linearly related to the unit sample response $h(n)$ of the FIR filter. The upper limit L in the sum is $L = (M-1)/2$ for Case 1, $L = (M-3)/2$ for Case 3, and $L = M/2 - 1$ for Case 2 and Case 4.

In addition to the common framework given above for the representation of $H_r(\omega)$, we also define the real-valued desired frequency response $H_{\text{dr}}(\omega)$ and the weighting function $W(\omega)$ on the approximation error. The real-valued desired frequency response $H_{\text{dr}}(\omega)$ is simply defined to be unity in the passband and zero in the stopband. For example, Fig. 10.2.15 illustrates several different types of characteristics for $H_{\text{dr}}(\omega)$. The weighting function on the approximation error allows us to choose the relative size of the errors in the different frequency bands (i.e., in the passband and in the stopband). In particular, it is convenient to normalize $W(\omega)$ to unity in the stopband and set $W(\omega) = \delta_2/\delta_1$ in the passband, that is,

$$W(\omega) = \begin{cases} \delta_2/\delta_1, & \omega \text{ in the passband} \\ 1, & \omega \text{ in the stopband} \end{cases} \tag{10.2.66}$$

Then we simply select $W(\omega)$ in the passband to reflect our emphasis on the relative size of the ripple in the stopband to the ripple in the passband.

With the specification of $H_{\text{dr}}(\omega)$ and $W(\omega)$, we can now define the weighted approximation error as

$$\begin{aligned} E(\omega) &= W(\omega)[H_{\text{dr}}(\omega) - H_r(\omega)] \\ &= W(\omega)[H_{\text{dr}}(\omega) - Q(\omega)P(\omega)] \\ &= W(\omega)Q(\omega)\left[\frac{H_{\text{dr}}(\omega)}{Q(\omega)} - P(\omega)\right] \end{aligned} \tag{10.2.67}$$

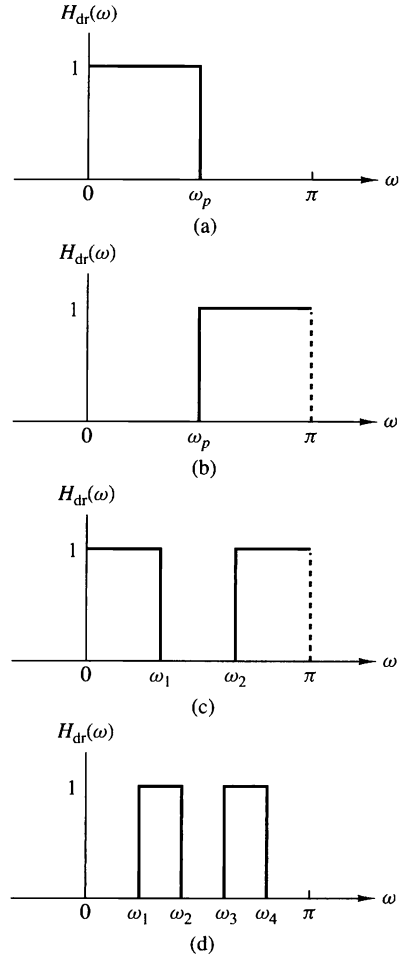


Figure 10.2.15
Desired frequency response characteristics for different types of filters.

For mathematical convenience, we define a modified weighting function $\hat{W}(\omega)$ and a modified desired frequency response $\hat{H}_{dr}(\omega)$ as

$$\begin{aligned} \hat{W}(\omega) &= W(\omega)Q(\omega) \\ \hat{H}_{dr}(\omega) &= \frac{H_{dr}(\omega)}{Q(\omega)} \end{aligned} \tag{10.2.68}$$

Then the weighted approximation error may be expressed as

$$E(\omega) = \hat{W}(\omega)[\hat{H}_{dr}(\omega) - P(\omega)] \tag{10.2.69}$$

for all four different types of linear-phase FIR filters.

Given the error function $E(\omega)$, the Chebyshev approximation problem is basically to determine the filter parameters $\{\alpha(k)\}$ that minimize the maximum absolute

value of $E(\omega)$ over the frequency bands in which the approximation is to be performed. In mathematical terms, we seek the solution to the problem

$$\min_{\text{over } \{\alpha(k)\}} \left[\max_{\omega \epsilon S} |E(\omega)| \right] = \min_{\text{over } \{a(k)\}} \left[\max_{\omega \epsilon S} |\hat{W}(\omega)[\hat{H}_{\text{dr}}(\omega) - \sum_{k=0}^{L} \alpha(k) \cos \omega k]| \right] \tag{10.2.70}$$

where S represents the set (disjoint union) of frequency bands over which the optimization is to be performed. Basically, the set S consists of the passbands and stopbands of the desired filter.

The solution to this problem is due to Parks and McClellan (1972a), who applied a theorem in the theory of Chebyshev approximation. It is called the *alternation theorem*, which we state without proof.

Alternation Theorem. Let S be a compact subset of the interval $[0, \pi)$. A necessary and sufficient condition for

$$P(\omega) = \sum_{k=0}^{L} \alpha(k) \cos \omega k$$

to be the unique, best weighted Chebyshev approximation to $\hat{H}_{\text{dr}}(\omega)$ in S, is that the error function $E(\omega)$ exhibit at least $L + 2$ extremal frequencies in S. That is, there must exist at least $L + 2$ frequencies $\{\omega_i\}$ in S such that $\omega_1 < \omega_2 < \cdots < \omega_{L+2}$, $E(\omega_i) = -E(\omega_{i+1})$, and

$$|E(\omega_i)| = \max_{\omega \epsilon S} |E(\omega)|, \qquad i = 1, 2, \ldots, L + 2$$

We note that the error function $E(\omega)$ alternates in sign between two successive extremal frequencies. Hence the theorem is called the alternation theorem.

To elaborate on the alternation theorem, let us consider the design of a lowpass filter with passband $0 \leq \omega \leq \omega_p$ and stopband $\omega_s \leq \omega \leq \pi$. Since the desired frequency response $H_{\text{dr}}(\omega)$ and the weighting function $W(\omega)$ are piecewise constant, we have

$$\frac{dE(\omega)}{d\omega} = \frac{d}{d\omega}\{W(\omega)[H_{\text{dr}}(\omega) - H_r(\omega)]\}$$

$$= -\frac{dH_r(\omega)}{d\omega} = 0$$

Consequently, the frequencies $\{\omega_i\}$ corresponding to the peaks of $E(\omega)$ also correspond to peaks at which $H_r(\omega)$ meets the error tolerance. Since $H_r(\omega)$ is a trigonometric polynomial of degree L, for Case 1, for example,

$$\begin{aligned} H_r(\omega) &= \sum_{k=0}^{L} \alpha(k) \cos \omega k \\ &= \sum_{k=0}^{L} \alpha(k) \left[\sum_{n=0}^{k} \beta_{nk} (\cos \omega)^n \right] \\ &= \sum_{k=0}^{L} \alpha'(k) (\cos \omega)^k \end{aligned} \tag{10.2.71}$$

it follows that $H_r(\omega)$ can have at most $L-1$ local maxima and minima in the open interval $0 < \omega < \pi$. In addition, $\omega = 0$ and $\omega = \pi$ are usually extrema of $H_r(\omega)$ and, also, of $E(\omega)$. Therefore, $H_r(\omega)$ has at most $L+1$ extremal frequencies. Furthermore, the band-edge frequencies ω_p and ω_s are also extrema of $E(\omega)$, since $|E(\omega)|$ is maximum at $\omega = \omega_p$ and $\omega = \omega_s$. As a consequence, there are at most $L+3$ extremal frequencies in $E(\omega)$ for the unique, best approximation of the ideal lowpass filter. On the other hand, the alternation theorem states that there are at least $L+2$ extremal frequencies in $E(\omega)$. Thus the error function for the lowpass filter design has either $L+3$ or $L+2$ extrema. In general, filter designs that contain more than $L+2$ alternations or ripples are called *extra ripple filters*. When the filter design contains the maximum number of alternations, it is called a *maximal ripple filter*.

The alternation theorem guarantees a unique solution for the Chebyshev optimization problem in (10.2.70). At the desired extremal frequencies $\{\omega_n\}$, we have the set of equations

$$\hat{W}(\omega_n)[\hat{H}_{\text{dr}}(\omega_n) - P(\omega_n)] = (-1)^n \delta, \qquad n = 0, 1, \ldots, L+1 \tag{10.2.72}$$

where δ represents the maximum value of the error function $E(\omega)$. In fact, if we select $W(\omega)$ as indicated by (10.2.66), it follows that $\delta = \delta_2$.

The set of linear equations in (10.2.72) can be rearranged as

$$P(\omega_n) + \frac{(-1)^n \delta}{\hat{W}(\omega_n)} = \hat{H}_{\text{dr}}(\omega_n), \qquad n = 0, 1, \ldots, L+1$$

or, equivalently, in the form

$$\sum_{k=0}^{L} \alpha(k) \cos \omega_n k + \frac{(-1)^n \delta}{\hat{W}(\omega_n)} = \hat{H}_{\text{dr}}(\omega_n), \qquad n = 0, 1, \ldots, L+1 \tag{10.2.73}$$

If we treat the $\{a(k)\}$ and δ as the parameters to be determined, (10.2.73) can be expressed in matrix form as

$$\begin{bmatrix} 1 & \cos\omega_0 & \cos 2\omega_0 & \cdots & \cos L\omega_0 & \dfrac{1}{\hat{W}(\omega_0)} \\ 1 & \cos\omega_1 & \cos 2\omega_1 & \cdots & \cos L\omega_1 & \dfrac{-1}{\hat{W}(\omega_1)} \\ \vdots & & & & & \\ & & & & & \\ \vdots & & & & & \\ 1 & \cos\omega_{L+1} & \cos 2\omega_{L+1} & \cdots & \cos L\omega_{L+1} & \dfrac{(-1)^{L+1}}{\hat{W}(\omega_{L+1})} \end{bmatrix} \begin{bmatrix} \alpha(0) \\ \alpha(1) \\ \vdots \\ \alpha(L) \\ \delta \end{bmatrix} = \begin{bmatrix} \hat{H}_{\mathrm{dr}}(\omega_0) \\ \hat{H}_{\mathrm{dr}}(\omega_1) \\ \vdots \\ \vdots \\ \hat{H}_{\mathrm{dr}}(\omega_{L+1}) \end{bmatrix} \tag{10.2.74}$$

Initially, we know neither the set of extremal frequencies $\{\omega_n\}$ nor the parameters $\{\alpha(k)\}$ and δ. To solve for the parameters, we use an iterative algorithm, called the *Remez exchange algorithm* [see Rabiner et al. (1975)], in which we begin by guessing at the set of extremal frequencies, determine $P(\omega)$ and δ, and then compute the error function $E(\omega)$. From $E(\omega)$ we determine another set of $L+2$ extremal frequencies and repeat the process iteratively until it converges to the optimal set of extremal frequencies. Although the matrix equation in (10.2.74) can be used in the iterative procedure, matrix inversion is time consuming and inefficient.

A more efficient procedure, suggested in the paper by Rabiner et al. (1975), is to compute δ analytically, according to the formula

$$\delta = \frac{\gamma_0 \hat{H}_{\mathrm{dr}}(\omega_0) + \gamma_1 \hat{H}_{\mathrm{dr}}(\omega_1) + \cdots + \gamma_{L+1} \hat{H}_{\mathrm{dr}}(\omega_{L+1})}{\dfrac{\gamma_0}{\hat{W}(\omega_0)} - \dfrac{\gamma_1}{\hat{W}(\omega_1)} + \cdots + \dfrac{(-1)^{L+1}\gamma_{L+1}}{\hat{W}(\omega_{L+1})}} \tag{10.2.75}$$

where

$$\gamma_k = \prod_{\substack{n=0 \\ n \neq k}}^{L+1} \frac{1}{\cos\omega_k - \cos\omega_n} \tag{10.2.76}$$

The expression for δ in (10.2.75) follows immediately from the matrix equation in (10.2.74). Thus with an initial guess at the $L+2$ extremal frequencies, we compute δ.

Now since $P(\omega)$ is a trigonometric polynomial of the form

$$P(\omega) = \sum_{k=0}^{L} \alpha(k) x^k, \qquad x = \cos\omega$$

and since we know that the polynomial at the points $x_n \equiv \cos\omega_n$, $n = 0, 1, \ldots, L+1$, has the corresponding values

$$P(\omega_n) = \hat{H}_{\mathrm{dr}}(\omega_n) - \frac{(-1)^n \delta}{\hat{W}(\omega_n)}, \qquad n = 0, 1, \ldots, L+1 \tag{10.2.77}$$

we can use the Lagrange interpolation formula for $P(\omega)$. Thus $P(\omega)$ can be expressed as [see Hamming (1962)]

$$P(\omega) = \frac{\sum_{k=0}^{L} P(\omega_k)[\beta_k/(x - x_k)]}{\sum_{k=0}^{L} [\beta_k/(x - x_k)]} \tag{10.2.78}$$

where $P(\omega_n)$ is given by (10.2.77), $x = \cos\omega$, $x_k = \cos\omega_k$, and

$$\beta_k = \prod_{\substack{n=0 \\ n \neq k}}^{L} \frac{1}{x_k - x_n} \tag{10.2.79}$$

Having the solution for $P(\omega)$, we can now compute the error function $E(\omega)$ from

$$E(\omega) = \hat{W}(\omega)[\hat{H}_{\text{dr}}(\omega) - P(\omega)] \tag{10.2.80}$$

on a dense set of frequency points. Usually, a number of points equal to $16M$, where M is the length of the filter, suffices. If $|E(\omega)| \geq \delta$ for some frequencies on the dense set, then a new set of frequencies corresponding to the $L + 2$ largest peaks of $|E(\omega)|$ are selected and the computational procedure beginning with (10.2.75) is repeated. Since the new set of $L + 2$ extremal frequencies is selected to correspond to the peaks of the error function $|E(\omega)|$, the algorithm forces δ to increase in each iteration until it converges to the upper bound and hence to the optimum solution for the Chebyshev approximation problem. In other words, when $|E(\omega)| \leq \delta$ for all frequencies on the dense set, the optimal solution has been found in terms of the polynomial $H(\omega)$. A flowchart of the algorithm is shown in Fig. 10.2.16 and is due to Remez (1957).

Once the optimal solution has been obtained in terms of $P(\omega)$, the unit sample response $h(n)$ can be computed directly, without having to compute the parameters $\{\alpha(k)\}$. In effect, we have determined

$$H_r(\omega) = Q(\omega)P(\omega)$$

which can be evaluated at $\omega = 2\pi k/M$, $k = 0, 1, \ldots, (M - 1)/2$, for M odd, or $M/2$ for M even. Then, depending on the type of filter being designed, $h(n)$ can be determined from the formulas given in Table 10.3.

A computer program written by Parks and McClellan (1972b) is available for designing linear-phase FIR filters based on the Chebyshev approximation criterion and implemented with the Remez exchange algorithm. This program can be used to design lowpass, highpass, or bandpass filters, differentiators, and Hilbert transformers. The latter two types of filters are described in the following sections. A number of software packages for designing equiripple linear-phase FIR filters are now available.

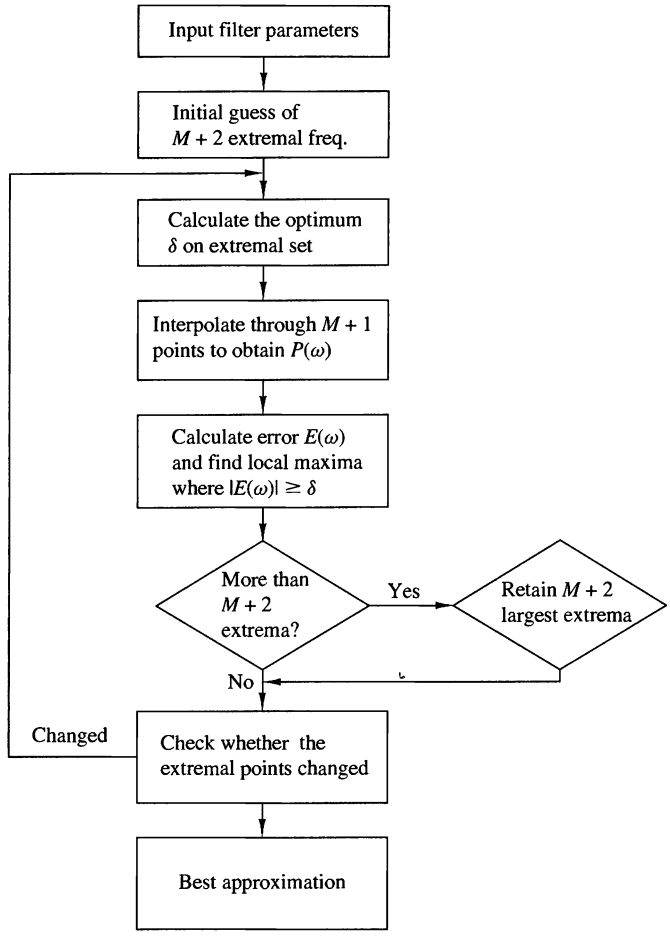


Figure 10.2.16 Flowchart of Remez algorithm.

The Parks–McClellan program requires a number of input parameters which determine the filter characteristics. In particular, the following parameters must be specified:

NFILT: The filter length, denoted above as M.

JTYPE: Type of filter:
JTYPE = 1 results in a multiple passband/stopband filter.
JTYPE = 2 results in a differentiator.
JTYPE = 3 results in a Hilbert transformer.

NBANDS: The number of frequency bands from 2 (for a lowpass filter) to a maximum of 10 (for a multiple-band filter).

LGRID: The grid density for interpolating the error function $E(\omega)$. The default value is 16 if left unspecified.

EDGE: The frequency bands specified by lower and upper cutoff frequencies,

up to a maximum of 10 bands (an array of size 20, maximum). The frequencies are given in terms of the variable $f = \omega/2\pi$, where $f = 0.5$ corresponds to the folding frequency.

FX: An array of maximum size 10 that specifies the desired frequency response $H_{\text{dr}}(\omega)$ in each band.

WTX: An array of maximum size 10 that specifies the weight function in each band.

The following examples demonstrate the use of this program to design a lowpass and a bandpass filter.

EXAMPLE 10.2.3

Design a lowpass filter of length $M = 61$ with a passband edge frequency $f_p = 0.1$ and a stopband edge frequency $f_s = 0.15$.

Solution. The lowpass filter is a two-band filter with passband edge frequencies $(0, 0.1)$ and stopband edge frequencies $(0.15, 0.5)$. The desired response is $(1, 0)$ and the weight function is arbitrarily selected as $(1, 1)$.

```
61, 1, 2
0.0, 0.1, 0.15, 0.5
1.0, 0.0
1.0, 1.0
```

The impulse response and frequency response are shown in Fig. 10.2.17. The resulting filter has a stopband attenuation of -56 dB and a passband ripple of 0.0135 dB.

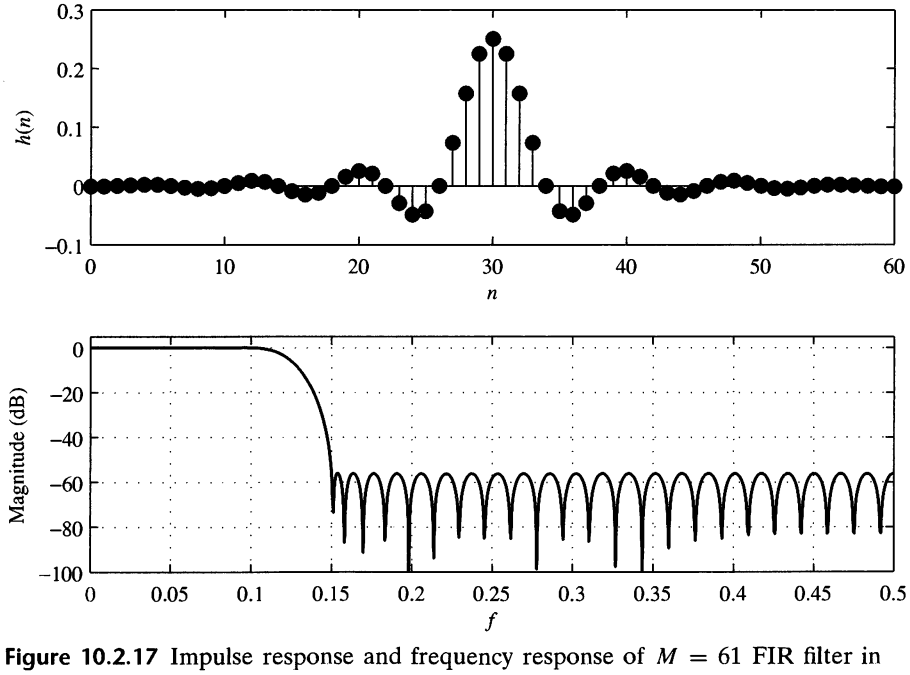


Figure 10.2.17 Impulse response and frequency response of $M = 61$ FIR filter in Example 10.2.3.

If we increase the length of the filter to $M = 101$ while maintaining all the other parameters given above the same, the resulting filter has the impulse response and frequency response characteristics shown in Fig. 10.2.18. Now, the stopband attenuation is -85 dB and the passband ripple is reduced to 0.00046 dB.

We should indicate that it is possible to increase the attenuation in the stopband by keeping the filter length fixed, say at $M = 61$, and decreasing the weighting function $W(\omega) = \delta_2/\delta_1$ in the passband. With $M = 61$ and a weighting function $(0.1, 1)$, we obtain a filter that has a stopband attenuation of -65 dB and a passband ripple of 0.049 dB.

EXAMPLE 10.2.4

Design a bandpass filter of length $M = 32$ with passband edge frequencies $f_{p1} = 0.2$ and $f_{p2} = 0.35$ and stopband edge frequencies of $f_{s1} = 0.1$ and $f_{s2} = 0.425$.

Solution. This passband filter is a three-band filter with a stopband range of $(0, 0.1)$, a passband range of $(0.2, 0.35)$, and a second stopband range of $(0.425, 0.5)$. The weighting function is selected as $(10.0, 1.0, 10.0)$, or as $(1.0, 0.1, 1.0)$, and the desired response in the three bands is $(0.0, 1.0, 0.0)$. Thus the input parameters to the program are

32, 1, 3
0.0, 0.1, 0.2, 0.35, 0.425, 0.5
0.0, 1.0, 0.0
10.0, 1.0, 10.0

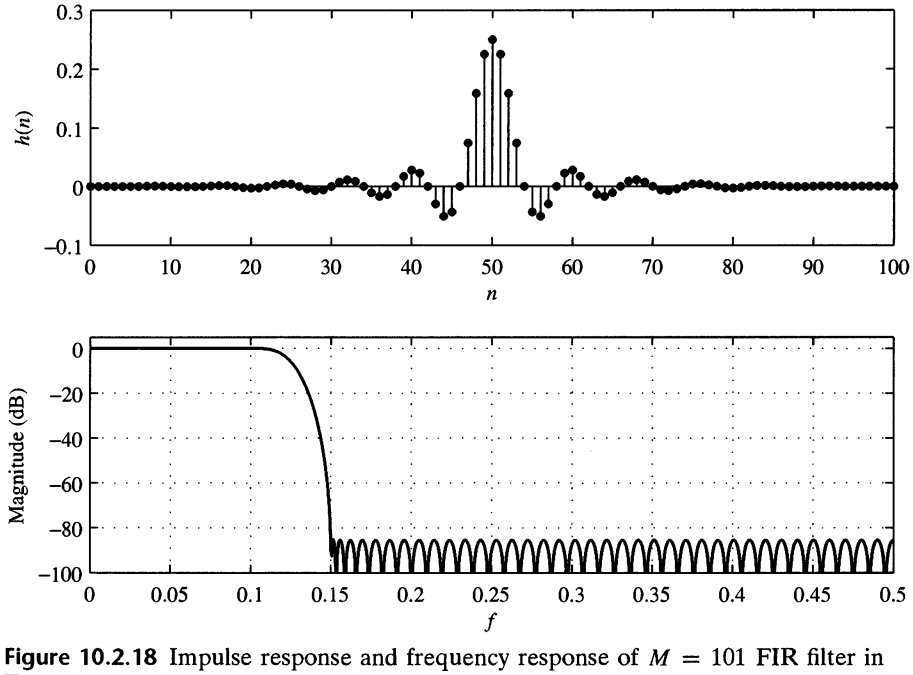


Figure 10.2.18 Impulse response and frequency response of $M = 101$ FIR filter in Example 10.2.3.

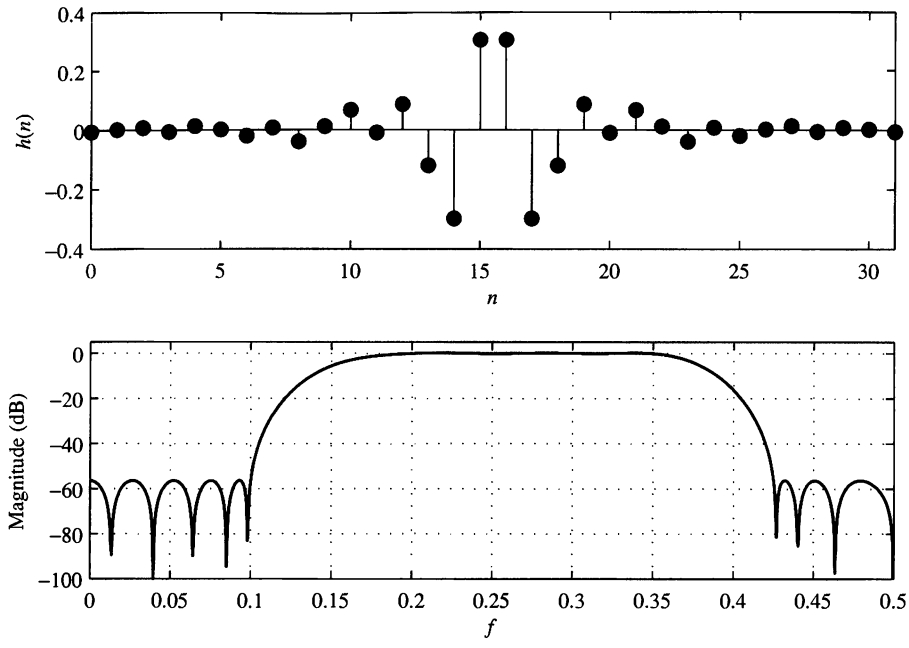


Figure 10.2.19 Impulse response and frequency response of $M = 32$ FIR filter in Example 10.2.4.

We note that the ripple in the stopbands δ_2 is 10 times smaller than the ripple in the passband due to the fact that errors in the stopband were given a weight of 10 compared to the passband weight of unity. The impulse response and frequency response of the bandpass filter are illustrated in Fig. 10.2.19.

These examples serve to illustrate the relative ease with which optimal lowpass, highpass, bandstop, bandpass, and more general multiband linear-phase FIR filters can be designed based on the Chebyshev approximation criterion implemented by means of the Remez exchange algorithm. In the next two sections we consider the design of differentiators and Hilbert transformers.

10.2.5 Design of FIR Differentiators

Differentiators are used in many analog and digital systems to take the derivative of a signal. An ideal differentiator has a frequency response that is linearly proportional to frequency. Similarly, an ideal digital differentiator is defined as one that has the frequency response

$$H_d(\omega) = j\omega, \qquad -\pi \leq \omega \leq \pi \tag{10.2.81}$$

The unit sample response corresponding to $H_d(\omega)$ is

$$\begin{aligned} h_d(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\omega)e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} j\omega e^{j\omega n}d\omega \\ &= \frac{\cos \pi n}{n}, \qquad -\infty < n < \infty, \qquad n \neq 0 \end{aligned} \tag{10.2.82}$$

We observe that the ideal differentiator has an antisymmetric unit sample response [i.e., $h_d(n) = -h_d(-n)$]. Hence, $h_d(0) = 0$.

In this section we consider the design of linear-phase FIR differentiators based on the Chebyshev approximation criterion. In view of the fact that the ideal differentiator has an antisymmetric unit sample response, we shall confine our attention to FIR designs in which $h(n) = -h(M-1-n)$. Hence we consider the filter types classified in the preceding section as Case 3 and Case 4.

We recall that in Case 3, where M is odd, the real-valued frequency response of the FIR filter $H_r(\omega)$ has the characteristic that $H_r(0) = 0$. A zero response at zero frequency is just the condition that the differentiator should satisfy, and we see from Table 10.5 that both filter types satisfy this condition. However, if a full-band differentiator is desired, this is impossible to achieve with an FIR filter having an odd number of coefficients, since $H_r(\pi) = 0$ for M odd. In practice, however, full-band differentiators are rarely required.

In most cases of practical interest, the desired frequency response characteristic need only be linear over the limited frequency range $0 \leq \omega \leq 2\pi f_p$, where f_p is called the bandwidth of the differentiator. In the frequency range $2\pi f_p < \omega \leq \pi$, the desired response may be either left unconstrained or constrained to be zero.

In the design of FIR differentiators based on the Chebyshev approximation criterion, the weighting function $W(\omega)$ is specified in the program as

$$W(\omega) = \frac{1}{\omega}, \qquad 0 \leq \omega \leq 2\pi f_p \tag{10.2.83}$$

in order that the relative ripple in the passband be a constant. Thus the absolute error between the desired response ω and the approximation $H_r(\omega)$ increases as ω varies from 0 to $2\pi f_p$. However, the weighting function in (10.2.83) ensures that the relative error

$$\begin{aligned} \delta &= \max_{0\leq\omega\leq 2\pi f_p} \{W(\omega)[\omega - H_r(\omega)]\} \\ &= \max_{0\leq\omega\leq 2\pi f_p} \left[1 - \frac{H_r(\omega)}{\omega}\right] \end{aligned} \tag{10.2.84}$$

is fixed within the passband of the differentiator.

EXAMPLE 10.2.5

Use the Remez algorithm to design a linear-phase FIR differentiator of length $M = 60$. The passband edge frequency is 0.1 and the stopband edge frequency is 0.15.

Solution. The input parameters to the program are

```
60,   2,    2
0.0,  0.1,  0.15,  0.5
1.0,  0.0
1.0,  1.0
```

The frequency response characteristic is illustrated in Fig. 10.2.20. Also shown in the same figure is the approximation error over the passband $0 \leq f \leq 0.1$ of the filter.

The important parameters in a differentiator are its length M, its bandwidth {band-edge frequency} f_p, and the peak relative error δ of the approximation. The interrelationship among these three parameters can be easily displayed parametrically. In particular, the value of $20 \log_{10} \delta$ versus f_p with M as a parameter is shown in Fig. 10.2.21 for M even and in Fig. 10.2.22 for M odd. These results, due to Rabiner and Schafer (1974a), are useful in the selection of the filter length, given specifications on the inband ripple and the cutoff frequency f_p.

A comparison of the graphs in Figs. 10.2.21 and 10.2.22 reveals that even-length differentiators result in a significantly smaller approximation error δ than comparable odd-length differentiators. Designs based on M odd are particularly poor if the bandwidth exceeds $f_p = 0.45$. The problem is basically the zero in the frequency response at $\omega = \pi (f = 1/2)$. When $f_p < 0.45$, good designs are obtained for M odd, but comparable-length differentiators with M even are always better in the sense that the approximation error is smaller.

In view of the obvious advantage of even-length over odd-length differentiators, a conclusion might be that even-length differentiators are always preferable in practical systems. This is certainly true for many applications. However, we should note that the signal delay introduced by any linear-phase FIR filter is $(M-1)/2$, which is not an integer when M is even. In many practical applications, this is unimportant. In some applications where it is desirable to have an integer-valued delay in the signal at the output of the differentiator, we must select M to be odd.

These numerical results are based on designs resulting from the Chebyshev approximation criterion. We wish to indicate it is also possible and relatively easy to design linear-phase FIR differentiators based on the frequency-sampling method. For example, Fig. 10.2.23 illustrates the frequency response characteristics of a wideband ($f_p = 0.5$) differentiator, of length $M = 30$. The graph of the absolute value of the approximation error as a function of frequency is also shown in this figure.

10.2.6 Design of Hilbert Transformers

An ideal Hilbert transformer is an all-pass filter that imparts a 90° phase shift on the signal at its input. Hence the frequency response of the ideal Hilbert transformer is specified as

$$H_d(\omega) = \begin{cases} -j, & 0 < \omega \leq \pi \\ j, & -\pi < \omega < 0 \end{cases} \tag{10.2.85}$$

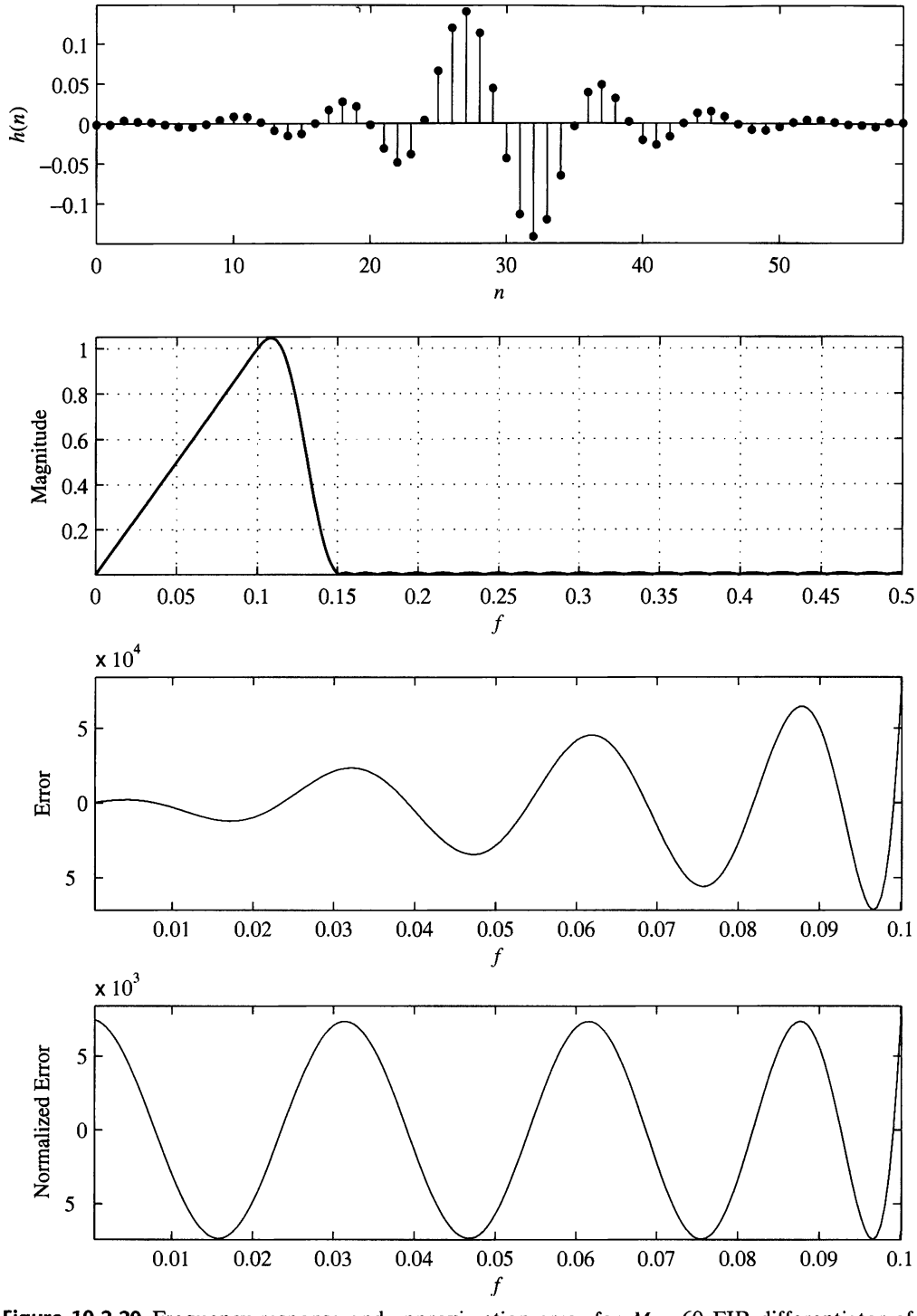


Figure 10.2.20 Frequency response and approximation error for $M = 60$ FIR differentiator of Example 10.2.5.

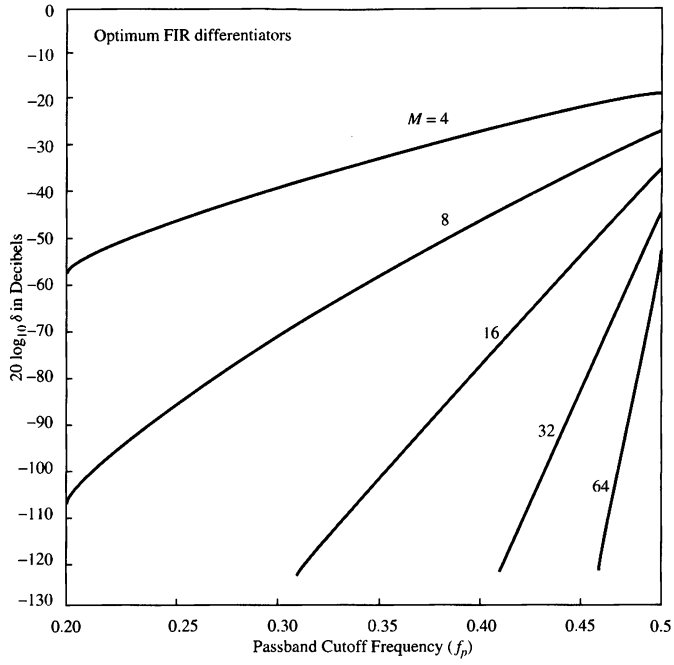


Figure 10.2.21 Curves of $20\log_{10}\delta$ versus f_p for $M = 4$, 8, 16, 32, and 64. [From paper by Rabiner and Schafer (1974a). Reprinted with permission of AT&T.]

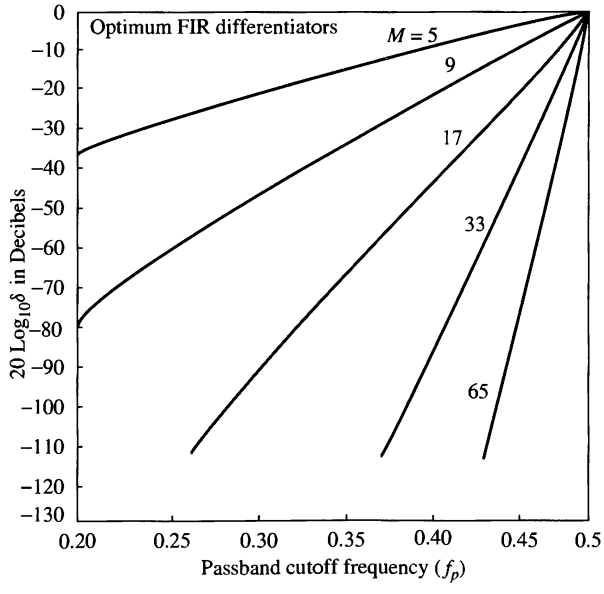


Figure 10.2.22 Curves of $20\log_10\delta$ versus F_p for $M = 5$, 9, 17,33 and 65 [From paper by Rabiner and Schafer (1974a). Reprinted with permission of AT&T.]

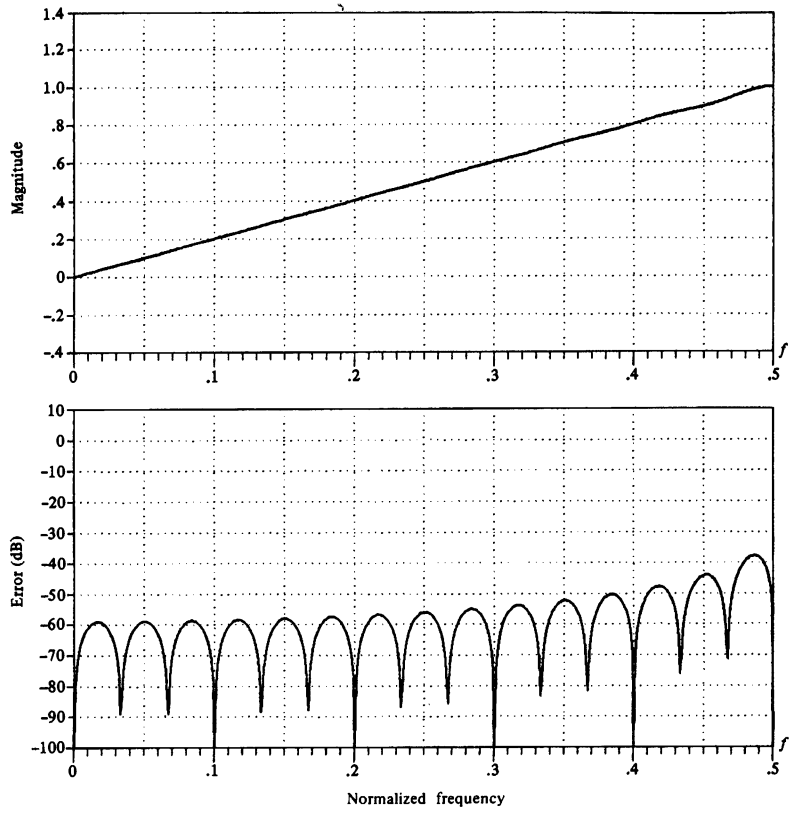


Figure 10.2.23 Frequency response and approximation error for $M = 30$ FIR differentiator designed by frequency-sampling method.

Hilbert transformers are frequently used in communication systems and signal processing, as, for example, in the generation of single-sideband modulated signals, radar signal processing, and speech signal processing.

The unit sample response of an ideal Hilbert transformer is

$$\begin{aligned} h_d(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\omega)e^{j\omega n}d\omega \\ &= \frac{1}{2\pi}\left(\int_{-\pi}^{0} je^{j\omega n}d\omega - \int_{0}^{\pi} je^{j\omega n}d\omega\right) \\ &= \begin{cases} \dfrac{2}{\pi}\dfrac{\sin^2(\pi n/2)}{n}, & n \neq 0 \\ 0, & n = 0 \end{cases} \end{aligned} \tag{10.2.86}$$

As expected, $h_d(n)$ is infinite in duration and noncausal. We note that $h_d(n)$ is antisymmetric [i.e., $h_d(n) = -h_d(-n)$]. In view of this characteristic, we focus our attention on the design of linear-phase FIR Hilbert transformers with antisymmetric

unit sample response [i.e., $h(n) = -h(M-1-n)$]. We also observe that our choice of an antisymmetric unit sample response is consistent with having a purely imaginary frequency response characteristic $H_d(\omega)$.

We recall once again that when $h(n)$ is antisymmetric, the real-valued frequency response characteristic $H_r(\omega)$ is zero at $\omega = 0$ for both M odd and even and at $\omega = \pi$ when M is odd. Clearly, then, it is impossible to design an all-pass digital Hilbert transformer. Fortunately, in practical signal processing applications, an all-pass Hilbert transformer is unnecessary. Its bandwidth need only cover the bandwidth of the signal to be phase shifted. Consequently, we specify the desired real-valued frequency response of a Hilbert transform filter as

$$H_{\text{dr}}(\omega) = 1, \qquad 2\pi f_l \leq \omega \leq 2\pi f_u \tag{10.2.87}$$

where f_l and f_u are the lower and upper cutoff frequencies, respectively.

It is interesting to note that the ideal Hilbert transformer with unit sample response $h_d(n)$ as given in (10.2.86) is zero for n even. This property is retained by the FIR Hilbert transformer under some symmetry conditions. In particular, let us consider the Case 3 filter type for which

$$H_r(\omega) = \sum_{k=1}^{(M-1)/2} c(k) \sin \omega k \tag{10.2.88}$$

and suppose that $f_l = 0.5 - f_u$. This ensures a symmetric passband about the midpoint frequency $f = 0.25$. If we have this symmetry in the frequency response, $H_r(\omega) = H_r(\pi - \omega)$ and hence (10.2.88) yields

$$\begin{aligned}
\sum_{k=1}^{(M-1)/2} c(k) \sin \omega k &= \sum_{k=1}^{(M-1)/2} c(k) \sin k(\pi - \omega) \\
&= \sum_{k=1}^{(M-1)/2} c(k) \sin \omega k \cos \pi k \\
&= \sum_{k=1}^{(M-1)/2} c(k)(-1)^{k+1} \sin \omega k
\end{aligned}$$

or equivalently,

$$\sum_{k=1}^{(M-1)/2} [1 - (-1)^{k+1}] c(k) \sin \omega k = 0 \tag{10.2.89}$$

Clearly, $c(k)$ must be equal to zero for $k = 0, 2, 4, \ldots$.

Now, the relationship between $\{c(k)\}$ and the unit sample response $\{h(n)\}$ is, from (10.2.54),

$$c(k) = 2h\left(\frac{M-1}{2} - k\right)$$

or, equivalently,

$$h\left(\frac{M-1}{2}-k\right)=\frac{1}{2}c(k) \tag{10.2.90}$$

If $c(k)$ is zero for $k = 0, 2, 4, \ldots$, then (10.2.90) yields

$$h(k)=\begin{cases}0, & k=0,2,4,\ldots, \quad \text{for } \dfrac{M-1}{2} \text{ even} \\ 0, & k=1,3,5,\ldots, \quad \text{for } \dfrac{M-1}{2} \text{ odd}\end{cases} \tag{10.2.91}$$

Unfortunately, (10.2.91) holds only for M odd. It does not hold for M even. This means that for comparable values of M, the case M odd is preferable since the computational complexity (number of multiplications and additions per output point) is roughly one half of that for M even.

When the design of the Hilbert transformer is performed by the Chebyshev approximation criterion using the Remez algorithm, we select the filter coefficients to minimize the peak approximation error

$$\begin{aligned}\delta &= \max_{2\pi f_l \le \omega \le 2\pi f_u} [H_{\mathrm{dr}}(\omega) - H_r(\omega)] \\ &= \max_{2\pi f_l \le \omega \le 2\pi f_u} [1 - H_r(\omega)]\end{aligned} \tag{10.2.92}$$

Thus the weighting function is set to unity and the optimization is performed over the single frequency band (i.e., the passband of the filter).

EXAMPLE 10.2.6

Design a Hilbert transformer with parameters $M = 31$, $f_l = 0.05$, and $f_u = 0.45$.

Solution. We observe that the frequency response is symmetric, since $f_u = 0.5 - f_l$. The parameters for executing the Remez algorithm are

```
31,     3,  1
0.05,  0.45
1.0
1.0
```

The result of this design is the unit sample response and frequency response shown in Fig, 10.2.24. We observe that, indeed, every other value of $h(n)$ is essentially zero.

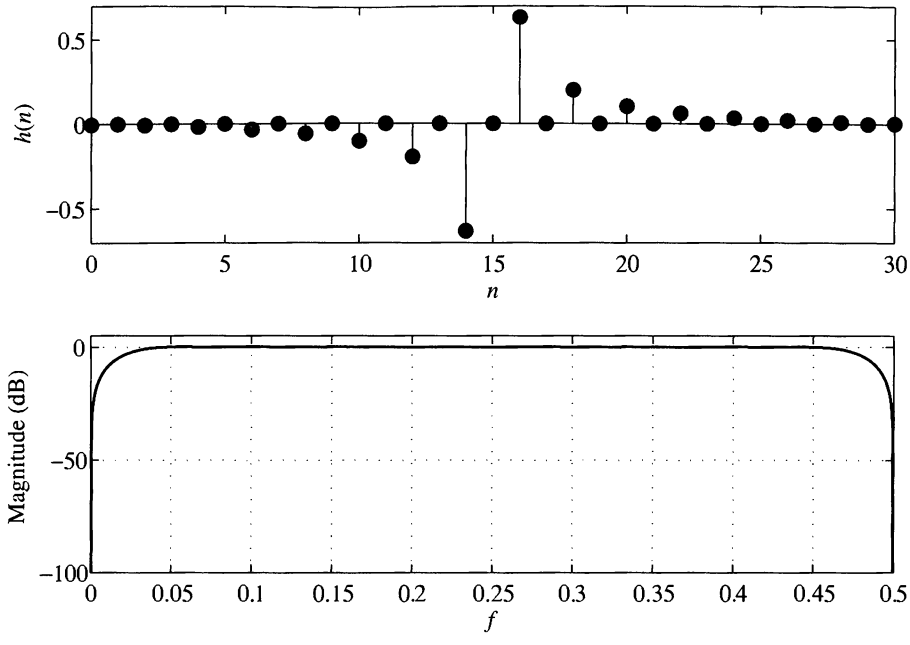


Figure 10.2.24 Frequency of FIR Hilbert transform filter in Example 10.2.6.

Rabiner and Schafer (1974b) have investigated the characteristics of Hilbert transformer designs for both M odd and M even. If the filter design is restricted to a symmetric frequency response, then there are basically three parameters of interest, M, δ, and f_l. Figure 10.2.25 is a plot of $20\log_{10}\delta$ versus f_l (the transition width) with M as a parameter. We observe that for comparable values of M, there is no performance advantage of using M odd over M even, and vice versa. However, the computational complexity in implementing a filter for M odd is less by a factor of 2 over M even as previously indicated. Therefore, M odd is preferable in practice.

For design purposes, the graphs in Fig. 10.2.25 suggest that, as a rule of thumb,

$$Mf_l \approx -0.61\log_{10}\delta \tag{10.2.93}$$

Hence this formula can be used to estimate the size of one of the three basic filter parameters when the other two parameters are specified.

10.2.7 Comparison of Design Methods for Linear-Phase FIR Filters

Historically, the design method based on the use of windows to truncate the impulse response $h_d(n)$ and obtain the desired spectral shaping was the first method proposed for designing linear-phase FIR filters. The frequency-sampling method and the Chebyshev approximation method were developed in the 1970s and have since become very popular in the design of practical linear-phase FIR filters.

The major disadvantage of the window design method is the lack of precise control of the critical frequencies, such as ω_p and ω_s, in the design of a lowpass FIR

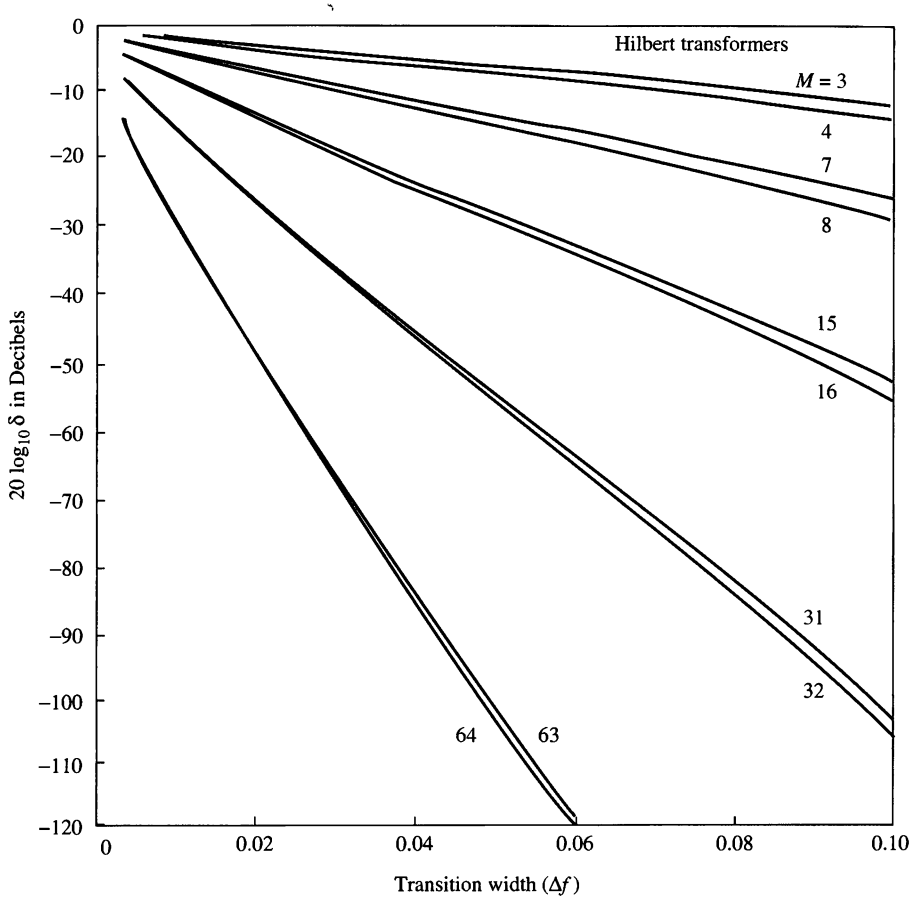


Figure 10.2.25 Curves of $20\log_{10}\delta$ versus Δf for $M = 3, 4, 7, 8, 15, 16, 31, 32, 63, 64$. [From paper by Rabiner and Schafer (1974b). Reprinted with permission of AT&T.]

filter. The values of ω_p and ω_s, in general, depend on the type of window and the filter length M.

The frequency-sampling method provides an improvement over the window design method, since $H_r(\omega)$ is specified at the frequencies $\omega_k = 2\pi k/M$ or $\omega_k = \pi(2k+1)/M$ and the transition band is a multiple of $2\pi/M$. This filter design method is particularly attractive when the FIR filter is realized either in the frequency domain by means of the DFT or in any of the frequency-sampling realizations. The attractive feature of these realizations is that $H_r(\omega_k)$ is either zero or unity at all frequencies, except in the transition band.

The Chebyshev approximation method provides total control of the filter specifications, and, as a consequence, it is usually preferable over the other two methods. For a lowpass filter, the specifications are given in terms of the parameters ω_p, ω_s, δ_1, δ_2, and M. We can specify the parameters ω_p, ω_s, M, and δ, and optimize the filters relative to δ_2. By spreading the approximation error over the passband and

the stopband of the filter, this method results in an optimal filter design, in the sense that for a given set of specifications just described, the maximum sidelobe level is minimized.

The Chebyshev design procedure based on the Remez exchange algorithm requires that we specify the length of the filter, the critical frequencies ω_p and ω_s, and the ratio δ_2/δ_1. However, it is more natural in filter design to specify ω_p, ω_s, δ_1, and δ_2 and to determine the filter length that satisfies the specifications. Although there is no simple formula to determine the filter length from these specifications, a number of approximations have been proposed for estimating M from ω_p, ω_s, δ_1, and δ_2. A particularly simple formula attributed to Kaiser for approximating M is

$$\hat{M} = \frac{-20\log_{10}\left(\sqrt{\delta_1\delta_2}\right) - 13}{14.6\,\Delta f} + 1 \tag{10.2.94}$$

where Δf is the transition band, defined as $\Delta f = (\omega_s - \omega_p)/2\pi$. This formula has been given in the paper by Rabiner et al. (1975). A more accurate formula proposed by Herrmann et al. (1973) is

$$\hat{M} = \frac{D_\infty(\delta_1, \delta_2) - f(\delta_1, \delta_2)(\Delta f)^2}{\Delta f} + 1 \tag{10.2.95}$$

where, by definition,

$$\begin{aligned} D_\infty(\delta_1, \delta_2) &= [0.005309(\log_{10}\delta_1)^2 + 0.07114(\log_{10}\delta_1) - 0.4761](\log_{10}\delta_2) \\ &\quad - [0.00266(\log_{10}\delta_1)^2 + 0.5941\log_{10}\delta_1 + 0.4278] \end{aligned} \tag{10.2.96}$$

$$f(\delta_1, \delta_2) = 11.012 + 0.51244(\log_{10}\delta_1 - \log_{10}\delta_2) \tag{10.2.97}$$

These formulas are extremely useful in obtaining a good estimate of the filter length required to achieve the given specifications Δf, δ_1, and δ_2. The estimate is used to carry out the design, and if the resulting δ exceeds the specified δ_2, the length can be increased until we obtain a sidelobe level that meets the specifications.

10.3 Design of IIR Filters From Analog Filters

Just as in the design of FIR filters, there are several methods that can be used to design digital filters having an infinite-duration unit sample response. The techniques described in this section are all based on converting an analog filter into a digital filter. Analog filter design is a mature and well-developed field, so it is not surprising that we begin the design of a digital filter in the analog domain and then convert the design into the digital domain.

An analog filter can be described by its system function,

$$H_a(s) = \frac{B(s)}{A(s)} = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k} \tag{10.3.1}$$

where $\{\alpha_k\}$ and $\{\beta_k\}$ are the filter coefficients, or by its impulse response, which is related to $H_a(s)$ by the Laplace transform

$$H_a(s) = \int_{-\infty}^{\infty} h(t)e^{-st}\,dt \tag{10.3.2}$$

Alternatively, the analog filter having the rational system function $H(s)$ given in (10.3.1) can be described by the linear constant-coefficient differential equation

$$\sum_{k=0}^{N} \alpha_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} \beta_k \frac{d^k x(t)}{dt^k} \tag{10.3.3}$$

where $x(t)$ denotes the input signal and $y(t)$ denotes the output of the filter.

Each of these three equivalent characterizations of an analog filter leads to alternative methods for converting the filter into the digital domain, as will be described in Sections 10.3.1 through 10.3.3. We recall that an analog linear time-invariant system with system function $H(s)$ is stable if all its poles lie in the left half of the s-plane. Consequently, if the conversion technique is to be effective, it should possess the following desirable properties:

1. The $j\Omega$ axis in the s-plane should map into the unit circle in the z-plane. Thus there will be a direct relationship between the two frequency variables in the two domains.
2. The left-half plane (LHP) of the s-plane should map into the inside of the unit circle in the z-plane. Thus a stable analog filter will be converted to a stable digital filter.

We mentioned in the preceding section that physically realizable and stable IIR filters cannot have linear phase. Recall that a linear-phase filter must have a system function that satisfies the condition

$$H(z) = \pm z^{-N} H(z^{-1}) \tag{10.3.4}$$

where z^{-N} represents a delay of N units of time. But if this were the case, the filter would have a mirror-image pole outside the unit circle for every pole inside the unit circle. Hence the filter would be unstable. Consequently, a causal and stable IIR filter cannot have linear phase.

If the restriction on physical realizability is removed, it is possible to obtain a linear-phase IIR filter, at least in principle. This approach involves performing a time reversal of the input signal $x(n)$, passing $x(-n)$ through a digital filter $H(z)$, time-reversing the output of $H(z)$, and finally, passing the result through $H(z)$ again. This signal processing is computationally cumbersome and appears to offer no advantages over linear-phase FIR filters. Consequently, when an application requires a linear-phase filter, it should be an FIR filter.

In the design of IIR filters, we shall specify the desired filter characteristics for the magnitude response only. This does not mean that we consider the phase response unimportant. Since the magnitude and phase characteristics are related, as indicated in Section 10.1, we specify the desired magnitude characteristics and accept the phase response that is obtained from the design methodology.

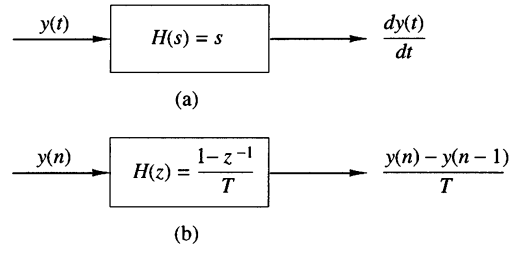


Figure 10.3.1
Substitution of the backward difference for the derivative implies the mapping $s = (1 - z^{-1})/T$.

10.3.1 IIR Filter Design by Approximation of Derivatives

One of the simplest methods for converting an analog filter into a digital filter is to approximate the differential equation in (10.3.3) by an equivalent difference equation. This approach is often used to solve a linear constant-coefficient differential equation numerically on a digital computer.

For the derivative $dy(t)/dt$ at time $t = nT$, we substitute the *backward difference* $[y(nT) - y(nT - 1)]/T$. Thus

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} = \frac{y(nT) - y(nT - T)}{T} = \frac{y(n) - y(n - 1)}{T} \tag{10.3.5}$$

where T represents the sampling interval and $y(n) \equiv y(nT)$. The analog differentiator with output $dy(t)/dt$ has the system function $H(s) = s$, while the digital system that produces the output $[y(n) - y(n - 1)]/T$ has the system function $H(z) = (1 - z^{-1})/T$. Consequently, as shown in Fig. 10.3.1, the frequency-domain equivalent for the relationship in (10.3.5) is

$$s = \frac{1 - z^{-1}}{T} \tag{10.3.6}$$

The second derivative $d^2y(t)/dt^2$ is replaced by the second difference, which is derived as follows:

$$\begin{aligned}\left.\frac{d^2y(t)}{dt^2}\right|_{t=nT} &= \frac{d}{dt}\left[\frac{dy(t)}{dt}\right]_{t=nT} \\ &= \frac{[y(nT) - y(nT - T)]/T - [y(nT - T) - y(nT - 2T)]/T}{T} \\ &= \frac{y(n) - 2y(n - 1) + y(n - 2)}{T^2}\end{aligned} \tag{10.3.7}$$

In the frequency domain, (10.3.7) is equivalent to

$$s^2 = \frac{1 - 2z^{-1} + z^{-2}}{T^2} = \left(\frac{1 - z^{-1}}{T}\right)^2 \tag{10.3.8}$$

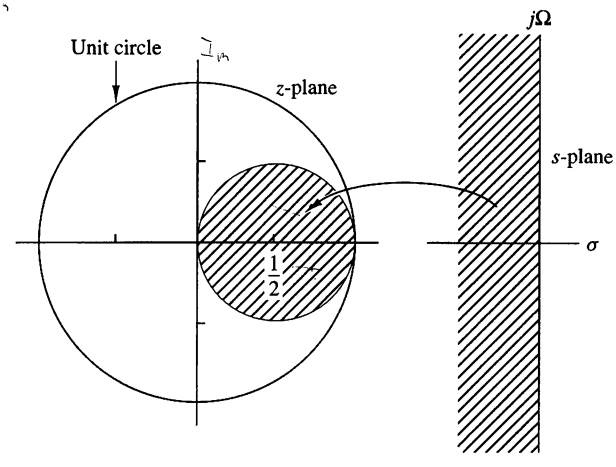


Figure 10.3.2
The mapping $s = (1 - z^{-1})/T$ takes LHP in the s-plane into points inside the circle of radius $\frac{1}{2}$ and center $z = \frac{1}{2}$ in the z-plane.

It easily follows from the discussion that the substitution for the kth derivative of $y(t)$ results in the equivalent frequency-domain relationship

$$s^k = \left(\frac{1 - z^{-1}}{T}\right)^k \tag{10.3.9}$$

Consequently, the system function for the digital IIR filter obtained as a result of the approximation of the derivatives by finite differences is

$$H(z) = H_a(s)|_{s=(1-z^{-1})/T} \tag{10.3.10}$$

where $H_a(s)$ is the system function of the analog filter characterized by the differential equation given in (10.3.3).

Let us investigate the implications of the mapping from the s-plane to the z-plane as given by (10.3.6) or, equivalently,

$$z = \frac{1}{1 - sT} \tag{10.3.11}$$

If we substitute $s = j\Omega$ in (10.2.11), we find that

$$\begin{aligned} z &= \frac{1}{1 - j\Omega T} \\ &= \frac{1}{1 + \Omega^2 T^2} + j\frac{\Omega T}{1 + \Omega^2 T^2} \end{aligned} \tag{10.3.12}$$

As Ω varies from $-\infty$ to ∞, the corresponding locus of points in the z-plane is a circle of radius $\frac{1}{2}$ and with center at $z = \frac{1}{2}$, as illustrated in Fig. 10.3.2.

It is easily demonstrated that the mapping in (10.3.11) takes points in the LHP of the s-plane into corresponding points inside this circle in the z-plane, and points in the right-hand plane (RHP) of the s-plane are mapped into points outside this circle. Consequently, this mapping has the desirable property that a stable analog filter is transformed into a stable digital filter. However, the possible location of the poles of the digital filter are confined to relatively small frequencies and, as a consequence, the mapping is restricted to the design of lowpass filters and bandpass filters having relatively small resonant frequencies. It is not possible, for example, to transform a highpass analog filter into a corresponding highpass digital filter.

In an attempt to overcome the limitations in the mapping given above, more complex substitutions for the derivatives have been proposed. In particular, an Lth-order difference of the form

$$\left.\frac{dy(t)}{dt}\right|_{t=nT} = \frac{1}{T}\sum_{k=1}^{L}\alpha_k \frac{y(nT+kT) - y(nT-kT)}{T} \tag{10.3.13}$$

has been proposed, where $\{\alpha_k\}$ are a set of parameters that can be selected to optimize the approximation. The resulting mapping between the s-plane and the z-plane is now

$$s = \frac{1}{T}\sum_{k=1}^{L}\alpha_k(z^k - z^{-k}) \tag{10.3.14}$$

When $z = e^{j\omega}$, we have

$$s = j\frac{2}{T}\sum_{k=1}^{L}\alpha_k \sin \omega k \tag{10.3.15}$$

which is purely imaginary. Thus

$$\Omega = \frac{2}{T}\sum_{k=1}^{L}\alpha_k \sin \omega k \tag{10.3.16}$$

is the resulting mapping between the two frequency variables. By proper choice of the coefficients $\{\alpha_k\}$ it is possible to map the $j\Omega$-axis into the unit circle. Furthermore, points in the LHP in s can be mapped into points inside the unit circle in z.

Despite achieving the two desirable characteristics with the mapping of (10.3.16), the problem of selecting the set of coefficients $\{\alpha_k\}$ remains. In general, this is a difficult problem. Since simpler techniques exist for converting analog filters into IIR digital filters, we shall not emphasize the use of the Lth-order difference as a substitute for the derivative.

EXAMPLE 10.3.1

Convert the analog bandpass filter with system function

$$H_a(s) = \frac{1}{(s+0.1)^2 + 9}$$

into a digital IIR filter by use of the backward difference for the derivative.

Solution. Substitution for s from (10.3.6) into $H(s)$ yields

$$H(z) = \frac{1}{\left(\dfrac{1-z^{-1}}{T}+0.1\right)^2+9}$$

$$= \frac{T^2/(1+0.2T+9.01T^2)}{1-\dfrac{2(1+0.1T)}{1+0.2T+9.01T^2}z^{-1}+\dfrac{1}{1+0.2T+9.01T^2}z^{-2}}$$

The system function $H(z)$ has the form of a resonator provided that T is selected small enough (e.g., $T \leq 0.1$), in order for the poles to be near the unit circle. Note that the condition $a_1^2 < 4a_2$ is satisfied, so that the poles are complex valued.

For example, if $T = 0.1$, the poles are located at

$$p_{1,2} = 0.91 \pm j0.27$$

$$= 0.949e^{\pm j16.5^\circ}$$

We note that the range of resonant frequencies is limited to low frequencies, due to the characteristics of the mapping. The reader is encouraged to plot the frequency response $H(\omega)$ of the digital filter for different values of T and compare the results with the frequency response of the analog filter.

EXAMPLE 10.3.2

Convert the analog bandpass filter in Example 10.3.1 into a digital IIR filter by use of the mapping

$$s = \frac{1}{T}(z - z^{-1})$$

Solution. By substituting for s in $H(s)$, we obtain

$$H(z) = \frac{1}{\left(\dfrac{z-z^{-1}}{T}+0.1\right)^2+9}$$

$$= \frac{z^2T^2}{z^4+0.2Tz^3+(2+9.01T^2)z^2-0.2Tz+1}$$

We observe that this mapping has introduced two additional poles in the conversion from $H_a(s)$ to $H(z)$. As a consequence, the digital filter is significantly more complex than the analog filter. This is a major drawback to the mapping given above.

10.3.2 IIR Filter Design by Impulse Invariance

In the impulse invariance method, our objective is to design an IIR filter having a unit sample response $h(n)$ that is the sampled version of the impulse response of the analog filter. That is,

$$h(n) \equiv h(nT), \qquad n = 0, 1, 2, \ldots \tag{10.3.17}$$

where T is the sampling interval.

To examine the implications of (10.3.17), we refer back to Section 6.1. Recall that when a continuous time signal $x_a(t)$ with spectrum $X_a(F)$ is sampled at a rate $F_s = 1/T$ samples per second, the spectrum of the sampled signal is the periodic repetition of the scaled spectrum $F_s X_a(F)$ with period F_s. Specifically, the relationship is

$$X(f) = F_s \sum_{k=-\infty}^{\infty} X_a[(f-k)F_s] \tag{10.3.18}$$

where $f = F/F_s$ is the normalized frequency. Aliasing occurs if the sampling rate F_s is less than twice the highest frequency contained in $X_a(F)$.

Expressed in the context of sampling the impulse response of an analog filter with frequency response $H_a(F)$, the digital filter with unit sample response $h(n) \equiv h_a(nT)$ has the frequency response

$$H(f) = F_s \sum_{k=-\infty}^{\infty} H_a[(f-k)F_s] \tag{10.3.19}$$

or, equivalently,

$$H(\omega) = F_s \sum_{k=-\infty}^{\infty} H_a[(\omega - 2\pi k)F_s] \tag{10.3.20}$$

or

$$H(\Omega T) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(\Omega - \frac{2\pi k}{T}\right) \tag{10.3.21}$$

Figure 10.3.3 depicts the frequency response of a lowpass analog filter and the frequency response of the corresponding digital filter.

It is clear that the digital filter with frequency response $H(\omega)$ has the frequency response characteristics of the corresponding analog filter if the sampling interval T is selected sufficiently small to completely avoid or at least minimize the effects of aliasing. It is also clear that the impulse invariance method is inappropriate for designing highpass filters due to the spectrum aliasing that results from the sampling process.

To investigate the mapping of points between the z-plane and the s-plane implied by the sampling process, we rely on a generalization of (10.3.21) which relates the z-transform of $h(n)$ to the Laplace transform of $h_a(t)$. This relationship is

$$H(z)|_{z=e^{sT}} = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a\left(s - j\frac{2\pi k}{T}\right) \tag{10.3.22}$$

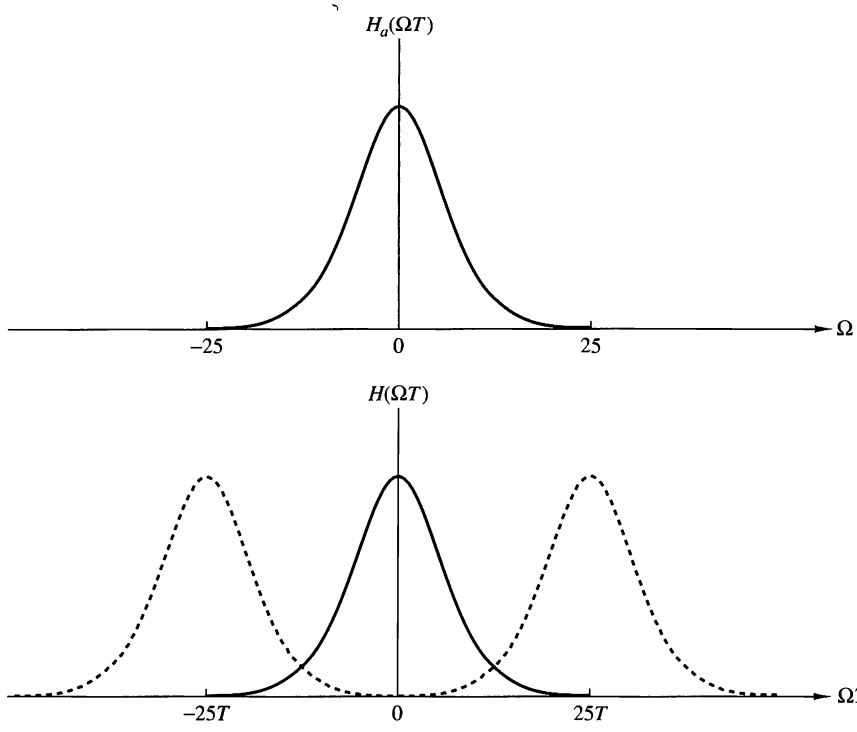


Figure 10.3.3 Frequency response $H_a(\Omega)$ of the analog filter and frequency response of the corresponding digital filter with aliasing.

where

$$H(z) = \sum_{n=0}^{\infty} h(n)z^{-n}$$
$$H(z)|_{z=e^{sT}} = \sum_{n=0}^{\infty} h(n)e^{-sTn} \tag{10.3.23}$$

Note that when $s = j\Omega$, (10.3.22) reduces to (10.3.21), where the factor of j in $H_a(\Omega)$ is suppressed in our notation.

Let us consider the mapping of points from the s-plane to the z-plane implied by the relation

$$z = e^{sT} \tag{10.3.24}$$

If we substitute $s = \sigma + j\Omega$ and express the complex variable z in polar form as $z = re^{j\omega}$, (10.3.24) becomes

$$re^{j\omega} = e^{\sigma T}e^{j\Omega T}$$

Clearly, we must have

$$r = e^{\sigma T}$$
$$\omega = \Omega T \tag{10.3.25}$$

Consequently, $\sigma < 0$ implies that $0 < r < 1$ and $\sigma > 0$ implies that $r > 1$. When $\sigma = 0$, we have $r = 1$. Therefore, the LHP in s is mapped inside the unit circle in z and the RHP in s is mapped outside the unit circle in z.

Also, the $j\Omega$-axis is mapped into the unit circle in z as indicated above. However, the mapping of the $j\Omega$-axis into the unit circle is not one-to-one. Since ω is unique over the range $(-\pi, \pi)$, the mapping $\omega = \Omega T$ implies that the interval $-\pi/T \leq \Omega \leq \pi/T$ maps into the corresponding values of $-\pi \leq \omega \leq \pi$. Furthermore, the frequency interval $\pi/T \leq \Omega \leq 3\pi/T$ also maps into the interval $-\pi \leq \omega \leq \pi$ and, in general, so does the interval $(2k-1)\pi/T \leq \Omega \leq (2k+1)\pi/T$, when k is an integer. Thus the mapping from the analog frequency Ω to the frequency variable ω in the digital domain is many-to-one, which simply reflects the effects of aliasing due to sampling. Figure 10.3.4 illustrates the mapping from the s-plane to the z-plane for the relation in (10.3.24).

To explore further the effect of the impulse invariance design method on the characteristics of the resulting filter, let us express the system function of the analog filter in partial-fraction form. On the assumption that the poles of the analog filter are distinct, we can write

$$H_a(s) = \sum_{k=1}^{N} \frac{c_k}{s - p_k} \tag{10.3.26}$$

where $\{p_k\}$ are the poles of the analog filter and $\{c_k\}$ are the coefficients in the partial-fraction expansion. Consequently,

$$h_a(t) = \sum_{k=1}^{N} c_k e^{p_k t}, \qquad t \geq 0 \tag{10.3.27}$$

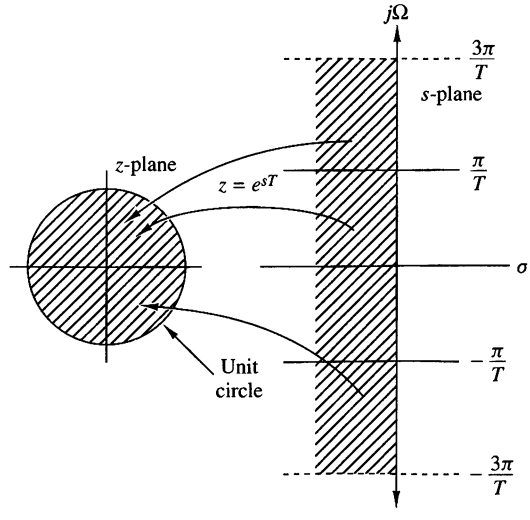


Figure 10.3.4
The mapping of $z = e^{sT}$ maps strips of width $2\pi/T$ (for $\sigma < 0$) in the s-plane into points in the unit circle in the z-plane.

If we sample $h_a(t)$ periodically at $t = nT$, we have

$$\begin{aligned} h(n) &= h_a(nT) \\ &= \sum_{k=1}^{N} c_k e^{p_k T n} \end{aligned} \tag{10.3.28}$$

Now, with the substitution of (10.3.28), the system function of the resulting digital IIR filter becomes

$$\begin{aligned} H(z) &= \sum_{n=0}^{\infty} h(n) z^{-n} \\ &= \sum_{n=0}^{\infty} \left(\sum_{k=1}^{N} c_k e^{p_k T n} \right) z^{-n} \\ &= \sum_{k=1}^{N} c_k \sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n \end{aligned} \tag{10.3.29}$$

The inner sum in (10.3.29) converges because $p_k < 0$ and yields

$$\sum_{n=0}^{\infty} (e^{p_k T} z^{-1})^n = \frac{1}{1 - e^{p_k T} z^{-1}} \tag{10.3.30}$$

Therefore, the system function of the digital filter is

$$H(z) = \sum_{k=1}^{N} \frac{c_k}{1 - e^{p_k T} z^{-1}} \tag{10.3.31}$$

We observe that the digital filter has poles at

$$z_k = e^{p_k T}, \qquad k = 1, 2, \ldots, N \tag{10.3.32}$$

Although the poles are mapped from the s-plane to the z-plane by the relationship in (10.3.32), we should emphasize that the zeros in the two domains do not satisfy the same relationship. Therefore, the impulse invariance method does not correspond to the simple mapping of points given by (10.3.24).

The development that resulted in $H(z)$ given by (10.3.31) was based on a filter having distinct poles. It can be generalized to include multiple-order poles. For brevity, however, we shall not attempt to generalize (10.3.31).

EXAMPLE 10.3.3

Convert the analog filter with system function

$$H_a(s) = \frac{s + 0.1}{(s + 0.1)^2 + 9}$$

into a digital IIR filter by means of the impulse invariance method.

Solution. We note that the analog filter has a zero at $s = -0.1$ and a pair of complex-conjugate poles at

$$p_k = -0.1 \pm j3$$

as illustrated in Fig. 10.3.5.

We do not have to determine the impulse response $h_a(t)$ in order to design the digital IIR filter based on the method of impulse invariance. Instead, we directly determine $H(z)$, as given by (10.3.31), from the partial-fraction expansion of $H_a(s)$. Thus we have

$$H(s) = \frac{\frac{1}{2}}{s + 0.1 - j3} + \frac{\frac{1}{2}}{s + 0.1 + j3}$$

Then

$$H(z) = \frac{\frac{1}{2}}{1 - e^{-0.1T}e^{j3T}z^{-1}} + \frac{\frac{1}{2}}{1 - e^{-0.1T}e^{-j3T}z^{-1}}$$

Since the two poles are complex conjugates, we can combine them to form a single two-pole filter with system function

$$H(z) = \frac{1 - (e^{-0.1T}\cos 3T)z^{-1}}{1 - (2e^{-0.1T}\cos 3T)z^{-1} + e^{-0.2T}z^{-1}}$$

The magnitude of the frequency response characteristic of this filter is plotted in Fig. 10.3.6 for $T = 0.1$ and $T = 0.5$. For purpose of comparison, we have also plotted the magnitude of the frequency response of the analog filter in Fig. 10.3.7. We note that aliasing is significantly more prevalent when $T = 0.5$ than when $T = 0.1$. Also, note the shift of the resonant frequency as T changes.

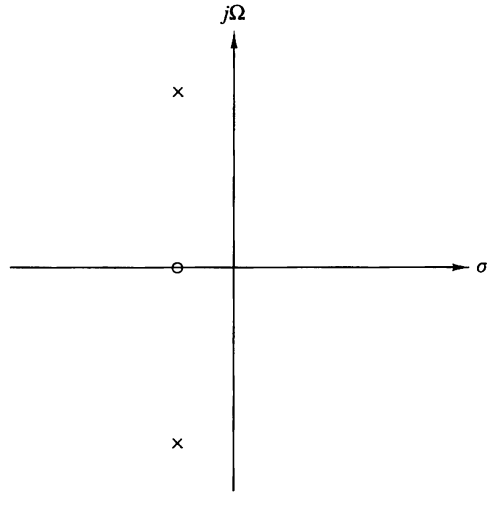


Figure 10.3.5
Pole–zero locations for analog filter in Example 10.3.3.

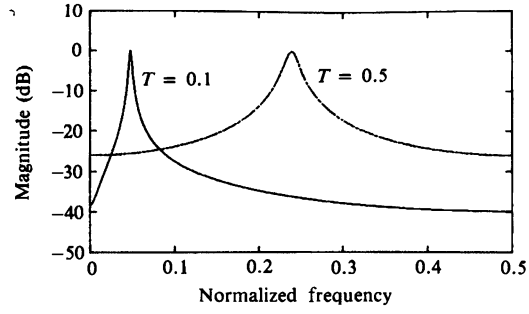


Figure 10.3.6
Frequency response of digital filter in Example 10.3.3.

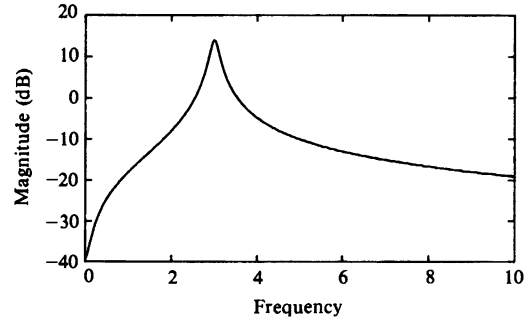


Figure 10.3.7
Frequency response of analog filter in Example 10.3.3.

The preceding example illustrates the importance of selecting a small value for T to minimize the effect of aliasing. Due to the presence of aliasing, the impulse invariance method is appropriate for the design of lowpass and bandpass filters only.

10.3.3 IIR Filter Design by the Bilinear Transformation

The IIR filter design techniques described in the preceding two sections have a severe limitation in that they are appropriate only for lowpass filters and a limited class of bandpass filters.

In this section we describe a mapping from the s-plane to the z-plane, called the bilinear transformation, that overcomes the limitation of the other two design methods described previously. The bilinear transformation is a conformal mapping that transforms the $j\Omega$-axis into the unit circle in the z-plane only once, thus avoiding aliasing of frequency components. Furthermore, all points in the LHP of s are mapped inside the unit circle in the z-plane and all points in the RHP of s are mapped into corresponding points outside the unit circle in the z-plane.

The bilinear transformation can be linked to the trapezoidal formula for numerical integration. For example, let us consider an analog linear filter with system function

$$H(s) = \frac{b}{s + a} \tag{10.3.33}$$

This system is also characterized by the differential equation

$$\frac{dy(t)}{dt} + ay(t) = bx(t) \tag{10.3.34}$$

Instead of substituting a finite difference for the derivative, suppose that we integrate the derivative and approximate the integral by the trapezoidal formula. Thus

$$y(t) = \int_{t_0}^{t} y'(\tau)d\tau + y(t_0) \tag{10.3.35}$$

where $y'(t)$ denotes the derivative of $y(t)$. The approximation of the integral in (10.3.35) by the trapezoidal formula at $t = nT$ and $t_0 = nT - T$ yields

$$y(nT) = \frac{T}{2}[y'(nT) + y'(nT - T)] + y(nT - T) \tag{10.3.36}$$

Now the differential equation in (10.3.34) evaluated at $t = nT$ yields

$$y'(nT) = -ay(nT) + bx(nT) \tag{10.3.37}$$

We use (10.3.37) to substitute for the derivative in (10.3.36) and thus obtain a difference equation for the equivalent discrete-time system. With $y(n) \equiv y(nT)$ and $x(n) \equiv x(nT)$, we obtain the result

$$\left(1 + \frac{aT}{2}\right) y(n) - \left(1 - \frac{aT}{2}\right) y(n-1) = \frac{bT}{2}[x(n) + x(n-1)] \tag{10.3.38}$$

The z-transform of this difference equation is

$$\left(1 + \frac{aT}{2}\right) Y(z) - \left(1 - \frac{aT}{2}\right) z^{-1}Y(z) = \frac{bT}{2}(1 + z^{-1})X(z)$$

Consequently, the system function of the equivalent digital filter is

$$H(z) = \frac{Y(z)}{X(z)} = \frac{(bT/2)(1 + z^{-1})}{1 + aT/2 - (1 - aT/2)z^{-1}}$$

or, equivalently,

$$H(z) = \frac{b}{\frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) + a} \tag{10.3.39}$$

Clearly, the mapping from the s-plane to the z-plane is

$$s = \frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) \tag{10.3.40}$$

This is called the *bilinear transformation.*

Although our derivation of the bilinear transformation was performed for a first-order differential equation, it holds, in general, for an Nth-order differential equation.

To investigate the characteristics of the bilinear transformation, let

$$z = re^{j\omega}$$

$$s = \sigma + j\Omega$$

Then (10.3.40) can be expressed as

$$\begin{aligned} s &= \frac{2}{T}\frac{z-1}{z+1} \\ &= \frac{2}{T}\frac{re^{j\omega}-1}{re^{j\omega}+1} \\ &= \frac{2}{T}\left(\frac{r^2-1}{1+r^2+2r\cos\omega} + j\frac{2r\sin\omega}{1+r^2+2r\cos\omega}\right) \end{aligned}$$

Consequently,

$$\sigma = \frac{2}{T}\frac{r^2-1}{1+r^2+2r\cos\omega} \tag{10.3.41}$$

$$\Omega = \frac{2}{T}\frac{2r\sin\omega}{1+r^2+2r\cos\omega} \tag{10.3.42}$$

First, we note that if $r < 1$, then $\sigma < 0$, and if $r > 1$, then $\sigma > 0$. Consequently, the LHP in s maps into the inside of the unit circle in the z-plane and the RHP in s maps into the outside of the unit circle. When $r = 1$, then $\sigma = 0$ and

$$\begin{aligned} \Omega &= \frac{2}{T}\frac{\sin\omega}{1+\cos\omega} \\ &= \frac{2}{T}\tan\frac{\omega}{2} \end{aligned} \tag{10.3.43}$$

or, equivalently,

$$\omega = 2\tan^{-1}\frac{\Omega T}{2} \tag{10.3.44}$$

The relationship in (10.3.44) between the frequency variables in the two domains is illustrated in Fig. 10.3.8. We observe that the entire range in Ω is mapped only once into the range $-\pi \le \omega \le \pi$. However, the mapping is highly nonlinear. We observe a frequency compression or *frequency warping*, as it is usually called, due to the nonlinearity of the arctangent function.

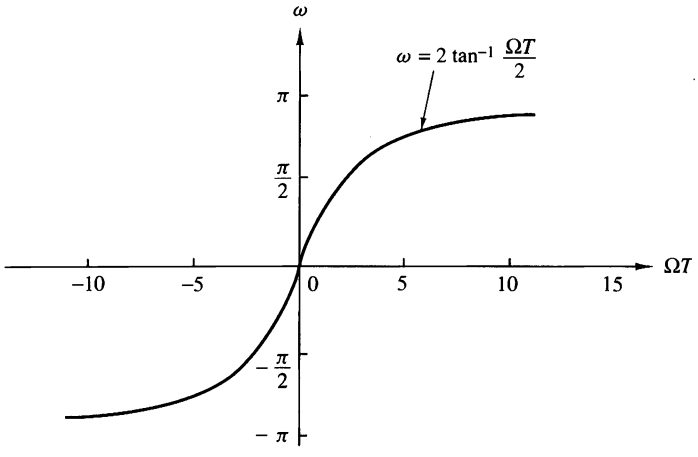


Figure 10.3.8 Mapping between the frequency variables ω and Ω resulting from the bilinear transformation.

It is also interesting to note that the bilinear transformation maps the point $s = \infty$ into the point $z = -1$. Consequently, the single-pole lowpass filter in (10.3.33), which has a zero at $s = \infty$, results in a digital filter that has a zero at $z = -1$.

EXAMPLE 10.3.4

Convert the analog filter with system function

$$H_a(s) = \frac{s + 0.1}{(s + 0.1)^2 + 16}$$

into a digital IIR filter by means of the bilinear transformation. The digital filter is to have a resonant frequency of $\omega_r = \pi/2$.

Solution. First, we note that the analog filter has a resonant frequency $\Omega_r = 4$. This frequency is to be mapped into $\omega_r = \pi/2$ by selecting the value of the parameter T. From the relationship in (10.3.43), we must select $T = \frac{1}{2}$ in order to have $\omega_r = \pi/2$. Thus the desired mapping is

$$s = 4\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

The resulting digital filter has the system function

$$H(z) = \frac{0.128 + 0.006z^{-1} - 0.122z^{-1}}{1 + 0.0006z^{-1} + 0.975z^{-2}}$$

We note that the coefficient of the z^{-1} term in the denominator of $H(z)$ is extremely small and can be approximated by zero. Thus we have the system function

$$H(z) = \frac{0.128 + 0.006z^{-1} - 0.122z^{-2}}{1 + 0.975z^{-2}}$$

This filter has poles at

$$p_{1,2} = 0.987e^{\pm j\pi/2}$$

and zeros at

$$z_{1,2} = -1, 0.95$$

Therefore, we have succeeded in designing a two-pole filter that resonates near $\omega = \pi/2$.

In this example the parameter T was selected to map the resonant frequency of the analog filter into the desired resonant frequency of the digital filter. Usually, the design of the digital filter begins with specifications in the digital domain, which involve the frequency variable ω. These specifications in frequency are converted to the analog domain by means of the relation in (10.3.43). The analog filter is then designed that meets these specifications and converted to a digital filter by means of the bilinear transformation in (10.3.40). In this procedure, the parameter T is transparent and may be set to any arbitrary value (e.g., $T = 1$). The following example illustrates this point.

EXAMPLE 10.3.5

Design a single-pole lowpass digital filter with a 3-dB bandwidth of 0.2π, using the bilinear transformation applied to the analog filter

$$H(s) = \frac{\Omega_c}{s + \Omega_c}$$

where Ω_c is the 3-dB bandwidth of the analog filter.

Solution. The digital filter is specified to have its -3-dB gain at $\omega_c = 0.2\pi$. In the frequency domain of the analog filter $\omega_c = 0.2\pi$ corresponds to

$$\Omega_c = \frac{2}{T}\tan 0.1\pi$$

$$= \frac{0.65}{T}$$

Thus the analog filter has the system function

$$H(s) = \frac{0.65/T}{s + 0.65/T}$$

This represents our filter design in the analog domain.

Now, we apply the bilinear transformation given by (10.3.40) to convert the analog filter into the desired digital filter. Thus we obtain

$$H(z) = \frac{0.245(1 + z^{-1})}{1 - 0.509z^{-1}}$$

where the parameter T has been divided out.

The frequency response of the digital filter is

$$H(\omega) = \frac{0.245(1 + e^{-j\omega})}{1 - 0.509e^{-j\omega}}$$

At $\omega = 0$, $H(0) = 1$, and at $\omega = 0.2\pi$, we have $|H(0.2\pi)| = 0.707$, which is the desired response.

10.3.4 Characteristics of Commonly Used Analog Filters

As we have seen from our discussion above, IIR digital filters can easily be obtained by beginning with an analog filter and then using a mapping to transform the s-plane into the z-plane. Thus the design of a digital filter is reduced to designing an appropriate analog filter and then performing the conversion from $H(s)$ to $H(z)$, in such a way so as to preserve, as much as possible, the desired characteristics of the analog filter.

Analog filter design is a well-developed field and many books have been written on the subject. In this section we briefly describe the important characteristics of commonly used analog filters and introduce the relevant filter parameters. Our discussion is limited to lowpass filters. Subsequently, we describe several frequency transformations that convert a lowpass prototype filter into either a bandpass, highpass, or band-elimination filter.

Butterworth filters. Lowpass Butterworth filters are all-pole filters characterized by the magnitude-squared frequency response

$$|H(\Omega)|^2 = \frac{1}{1 + (\Omega/\Omega_c)^{2N}} = \frac{1}{1 + \epsilon^2(\Omega/\Omega_p)^{2N}} \tag{10.3.45}$$

where N is the order of the filter, Ω_c is its -3-dB frequency (usually called the cutoff frequency), Ω_p is the passband edge frequency, and $1/(1+\epsilon^2)$ is the band-edge value of $|H(\Omega)|^2$. Since $H(s)H(-s)$ evaluated at $s = j\Omega$ is simply equal to $|H(\Omega)|^2$, it follows that

$$H(s)H(-s) = \frac{1}{1 + (-s^2/\Omega_c^2)^N} \tag{10.3.46}$$

The poles of $H(s)H(-s)$ occur on a circle of radius Ω_c at equally spaced points. From (10.3.46) we find that

$$\frac{-s^2}{\Omega_c^2} = (-1)^{1/N} = e^{j(2k+1)\pi/N}, \qquad k = 0, 1, \ldots, N-1$$

and hence

$$s_k = \Omega_c e^{j\pi/2} e^{j(2k+1)\pi/2N}, \qquad k = 0, 1, \ldots, N-1 \tag{10.3.47}$$

For example, Fig. 10.3.9 illustrates the pole positions for $N = 4$ and $N = 5$ Butterworth filters.

The frequency response characteristics of the class of Butterworth filters are shown in Fig. 10.3.10 for several values of N. We note that $|H(\Omega)|^2$ is monotonic in both the passband and stopband. The order of the filter required to meet an attenuation δ_2 at a specified frequency Ω_s is easily determined from (10.3.45). Thus at $\Omega = \Omega_s$ we have

$$\frac{1}{1 + \epsilon^2(\Omega_s/\Omega_p)^{2N}} = \delta_2^2$$

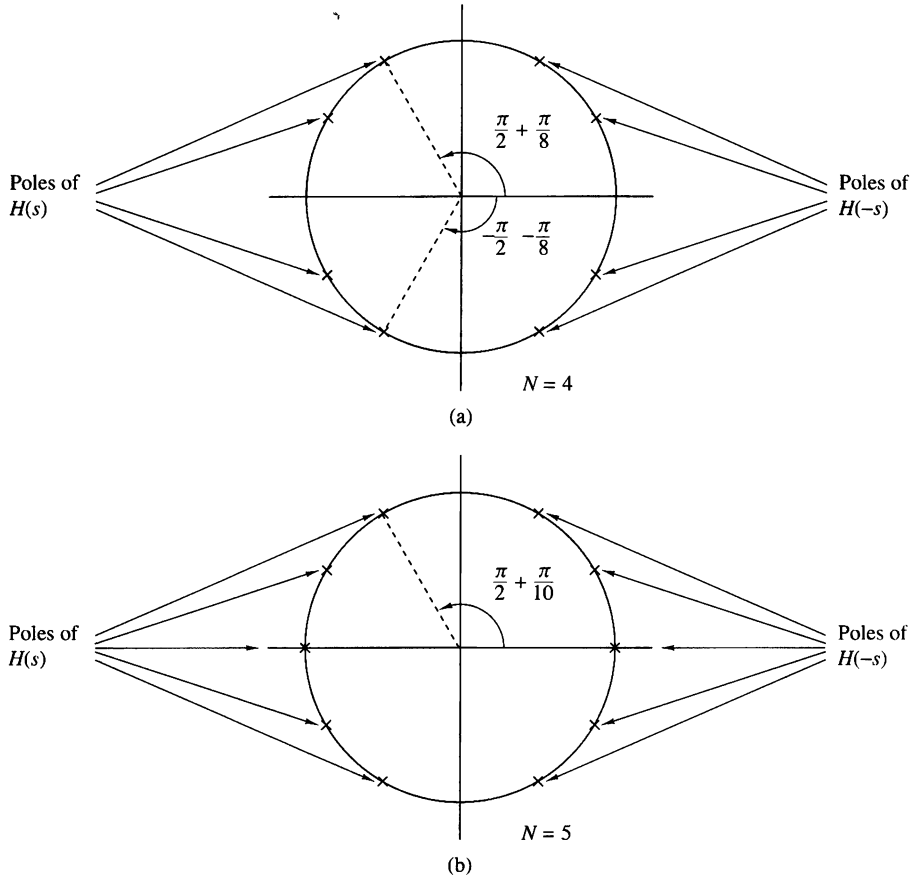


Figure 10.3.9 Pole positions for Butterworth filters.

and hence

$$N = \frac{\log[(1/\delta_2^2) - 1]}{2\log(\Omega_s/\Omega_c)} = \frac{\log(\delta/\epsilon)}{\log(\Omega_s/\Omega_p)} \tag{10.3.48}$$

where, by definition, $\delta_2 = 1/\sqrt{1+\delta^2}$. Thus the Butterworth filter is completely characterized by the parameters N, δ_2, ϵ, and the ratio Ω_s/Ω_p.

EXAMPLE 10.3.6

Determine the order and the poles of a lowpass Butterworth filter that has a −3-dB bandwidth of 500 Hz and an attenuation of 40 dB at 1000 Hz.

Solution. The critical frequencies are the −3-dB frequency Ω_c and the stopband frequency Ω_s, which are

$$\Omega_c = 1000\pi$$

$$\Omega_s = 2000\pi$$

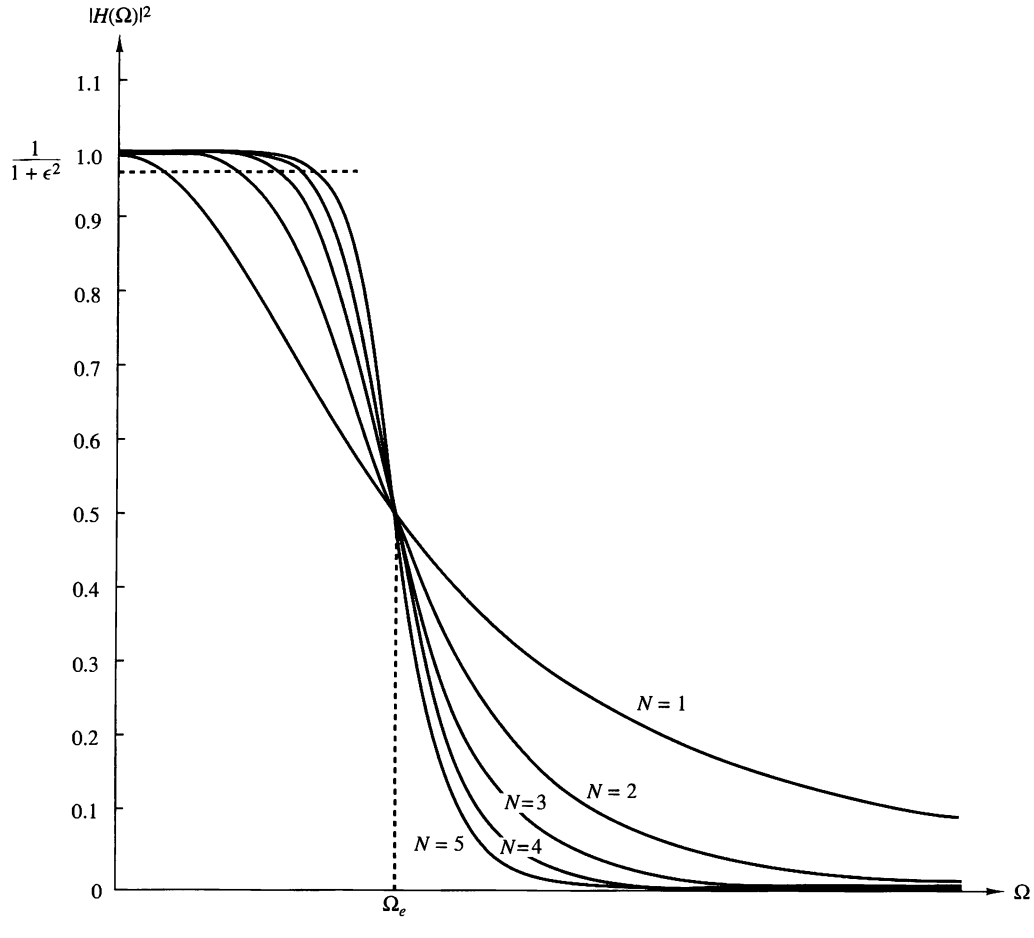


Figure 10.3.10 Frequency response of Butterworth filters.

For an attenuation of 40 dB, $\delta_2 = 0.01$. Hence from (10.3.48) we obtain

$$N = \frac{\log_{10}(10^4 - 1)}{2\log_{10} 2}$$
$$= 6.64$$

To meet the desired specifications, we select $N = 7$. The pole positions are

$$s_k = 1000\pi e^{j[\pi/2+(2k+1)\pi/14]}, \qquad k = 0, 1, 2, \ldots, 6$$

Chebyshev filters. There are two types of Chebyshev filters. Type I Chebyshev filters are all-pole filters that exhibit equiripple behavior in the passband and a monotonic

characteristic in the stopband. On the other hand, the family of type II Chebyshev filters contains both poles and zeros and exhibits a monotonic behavior in the passband and an equiripple behavior in the stopband. The zeros of this class of filters lie on the imaginary axis in the s-plane.

The magnitude squared of the frequency response characteristic of a type I Chebyshev filter is given as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 T_N^2(\Omega/\Omega_p)} \tag{10.3.49}$$

where ϵ is a parameter of the filter related to the ripple in the passband and $T_N(x)$ is the Nth-order Chebyshev polynomial defined as

$$T_N(x) = \begin{cases} \cos(N \cos^{-1} x), & |x| \leq 1 \\ \cosh(N \cosh^{-1} x), & |x| > 1 \end{cases} \tag{10.3.50}$$

The Chebyshev polynomials can be generated by the recursive equation

$$T_{N+1}(x) = 2xT_N(x) - T_{N-1}(x), \qquad N = 1, 2, \ldots \tag{10.3.51}$$

where $T_0(x) = 1$ and $T_1(x) = x$. From (10.3.51) we obtain $T_2(x) = 2x^2 - 1$, $T_3(x) = 4x^3 - 3x$, and so on.

Some of the properties of these polynomials are as follows:

1. $|T_N(x)| \leq 1$ for all $|x| \leq 1$.
2. $T_N(1) = 1$ for all N.
3. All the roots of the polynomial $T_N(x)$ occur in the interval $-1 \leq x \leq 1$.

The filter parameter ϵ is related to the ripple in the passband, as illustrated in Fig. 10.3.11, for N odd and N even. For N odd, $T_N(0) = 0$ and hence $|H(0)|^2 = 1$. On the other hand, for N even, $T_N(0) = 1$ and hence $|H(0)|^2 = 1/(1 + \epsilon^2)$. At the band-edge frequency $\Omega = \Omega_p$, we have $T_N(1) = 1$, so that

$$\frac{1}{\sqrt{1 + \epsilon^2}} = 1 - \delta_1$$

or, equivalently,

$$\epsilon^2 = \frac{1}{(1 - \delta_1)^2} - 1 \tag{10.3.52}$$

where δ_1 is the value of the passband ripple.

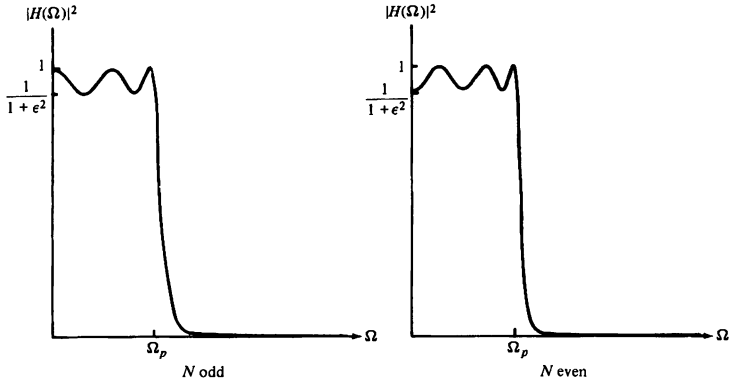


Figure 10.3.11 Type I Chebyshev filter characteristic.

The poles of a type I Chebyshev filter lie on an ellipse in the s-plane with major axis

$$r_1 = \Omega_p \frac{\beta^2 + 1}{2\beta} \tag{10.3.53}$$

and minor axis

$$r_2 = \Omega_p \frac{\beta^2 - 1}{2\beta} \tag{10.3.54}$$

where β is related to ϵ according to the equation

$$\beta = \left[\frac{\sqrt{1+\epsilon^2} + 1}{\epsilon} \right]^{1/N} \tag{10.3.55}$$

The pole locations are most easily determined for a filter of order N by first locating the poles for an equivalent Nth-order Butterworth filter that lie on circles of radius r_1 or radius r_2, as illustrated in Fig. 10.3.12. If we denote the angular positions of the poles of the Butterworth filter as

$$\phi_k = \frac{\pi}{2} + \frac{(2k+1)\pi}{2N}, \qquad k = 0, 1, 2, \ldots, N-1 \tag{10.3.56}$$

then the positions of the poles for the Chebyshev filter lie on the ellipse at the coordinates (x_k, y_k), $k = 0, 1, \ldots, N-1$, where

$$\begin{aligned} x_k &= r_2 \cos \phi_k, \quad k = 0, 1, \ldots, N-1 \\ y_k &= r_1 \sin \phi_k, \quad k = 0, 1, \ldots, N-1 \end{aligned} \tag{10.3.57}$$

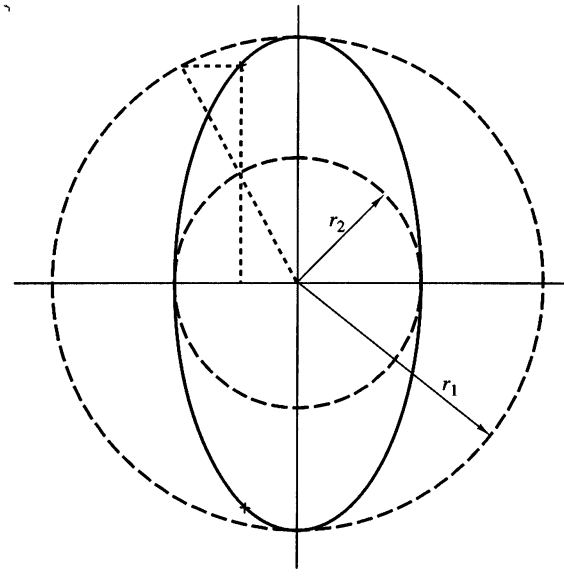


Figure 10.3.12
Determination of the pole locations for a Chebyshev filter.

A type II Chebyshev filter contains zeros as well as poles. The magnitude squared of its frequency response is given as

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2[T_N^2(\Omega_s/\Omega_p)/T_N^2(\Omega_s/\Omega)]} \tag{10.3.58}$$

where $T_N(x)$ is, again, the Nth-order Chebyshev polynomial and Ω_s is the stopband frequency as illustrated in Fig. 10.3.13. The zeros are located on the imaginary axis at the points

$$s_k = j\frac{\Omega_s}{\sin\phi_k}, \qquad k = 0, 1, \ldots, N-1 \tag{10.3.59}$$

The poles are located at the points (v_k, w_k), where

$$v_k = \frac{\Omega_s x_k}{\sqrt{x_k^2 + y_k^2}}, \qquad k = 0, 1, \ldots, N-1 \tag{10.3.60}$$

$$w_k = \frac{\Omega_s y_k}{\sqrt{x_k^2 + y_k^2}}, \qquad k = 0, 1, \ldots, N-1 \tag{10.3.61}$$

where $\{x_k\}$ and $\{y_k\}$ are defined in (10.3.57) with β now related to the ripple in the stopband through the equation

$$\beta = \left[\frac{1 + \sqrt{1 - \delta_2^2}}{\delta_2}\right]^{1/N} \tag{10.3.62}$$

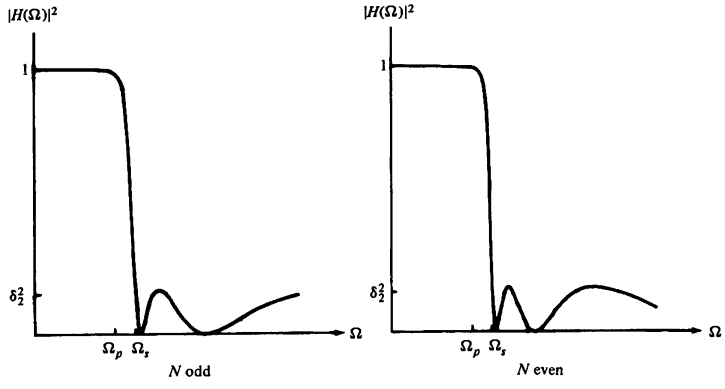


Figure 10.3.13 Type II Chebyshev filters.

From this description, we observe that the Chebyshev filters are characterized by the parameters N, ϵ, δ_2, and the ratio Ω_s/Ω_p. For a given set of specifications on ϵ, δ_2, and Ω_s/Ω_p, we can determine the order of the filter from the equation

$$N = \frac{\log\left[\left(\sqrt{1-\delta_2^2} + \sqrt{1-\delta_2^2(1+\epsilon^2)}\right)/\epsilon\delta_2\right]}{\log\left[(\Omega_s/\Omega_p) + \sqrt{(\Omega_s/\Omega_p)^2 - 1}\right]} \tag{10.3.63}$$

$$= \frac{\cosh^{-1}(\delta/\epsilon)}{\cosh^{-1}(\Omega_s/\Omega_p)}$$

where, by definition, $\delta_2 = 1/\sqrt{1+\delta^2}$.

EXAMPLE 10.3.7

Determine the order and the poles of a type I lowpass Chebyshev filter that has a 1-dB ripple in the passband, a cutoff frequency $\Omega_p = 1000\pi$, a stopband frequency of 2000π, and an attenuation of 40 dB or more for $\Omega \geq \Omega_s$.

Solution. First, we determine the order of the filter. We have

$$10\log_{10}(1+\epsilon^2) = 1$$

$$1+\epsilon^2 = 1.259$$

$$\epsilon^2 = 0.259$$

$$\epsilon = 0.5088$$

Also,

$$20\log_{10}\delta_2 = -40$$

$$\delta_2 = 0.01$$

Hence from (10.3.63) we obtain

$$N = \frac{\log_{10} 196.54}{\log_{10}(2+\sqrt{3})}$$

$$= 4.0$$

Thus a type I Chebyshev filter having four poles meets the specifications.

The pole positions are determined from the relations in (10.3.53) through (10.3.57). First, we compute β, r_1, and r_2. Hence

$$\beta = 1.429$$

$$r_1 = 1.06\Omega_p$$

$$r_2 = 0.365\Omega_p$$

The angles $\{\phi_k\}$ are

$$\phi_k = \frac{\pi}{2} + \frac{(2k+1)\pi}{8}, \qquad k = 0, 1, 2, 3$$

Therefore, the poles are located at

$$x_1 + jy_1 = -0.1397\Omega_p \pm j0.979\Omega_p$$

$$x_2 + jy_2 = -0.337\Omega_p \pm j0.4056\Omega_p$$

The filter specifications in Example 10.3.7 are very similar to the specifications given in Example 10.3.6, which involved the design of a Butterworth filter. In that case the number of poles required to meet the specifications was seven. On the other hand, the Chebyshev filter required only four. This result is typical of such comparisons. In general, the Chebyshev filter meets the specifications with fewer poles than the corresponding Butterworth filter. Alternatively, if we compare a Butterworth filter to a Chebyshev filter having the same number of poles and the same passband and stopband specifications, the Chebyshev filter will have a smaller transition bandwidth. For a tabulation of the characteristics of Chebyshev filters and their pole–zero locations, the interested reader is referred to the handbook of Zverev (1967).

Elliptic filters. Elliptic (or Cauer) filters exhibit equiripple behavior in both the passband and the stopband, as illustrated in Fig. 10.3.14 for N odd and N even. This class of filters contains both poles and zeros and is characterized by the magnitude-squared frequency response

$$|H(\Omega)|^2 = \frac{1}{1 + \epsilon^2 U_N(\Omega/\Omega_p)} \tag{10.3.64}$$

where $U_N(x)$ is the Jacobian elliptic function of order N, which has been tabulated by Zverev (1967), and ϵ is a parameter related to the passband ripple. The zeros lie on the $j\Omega$-axis.

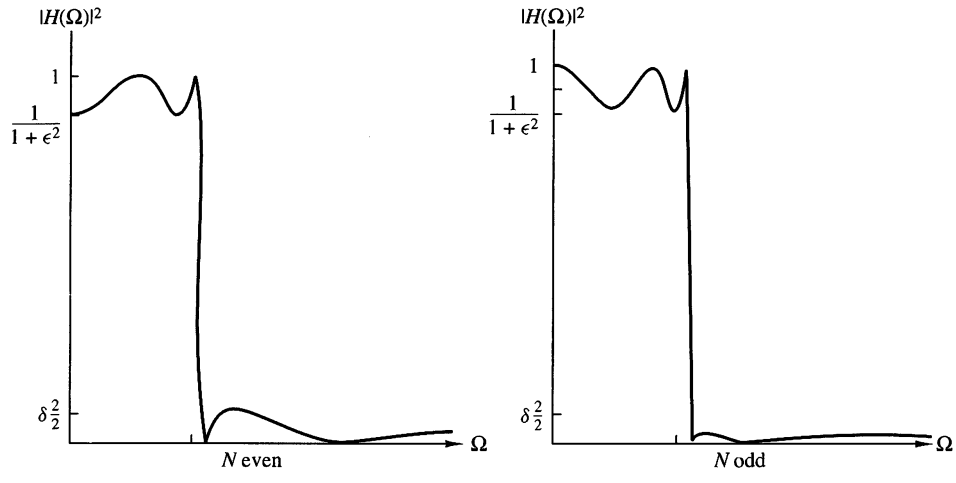


Figure 10.3.14 Magnitude-squared frequency characteristics of elliptic filters.

We recall from our discussion of FIR filters that the most efficient designs occur when we spread the approximation error equally over the passband and the stopband. Elliptic filters accomplish this objective and, as a consequence, are the most efficient from the viewpoint of yielding the smallest-order filter for a given set of specifications. Equivalently, we can say that for a given order and a given set of specifications, an elliptic filter has the smallest transition bandwidth.

The filter order required to achieve a given set of specifications in passband ripple δ_1, stopband ripple δ_2, and transition ratio Ω_p/Ω_s is given as

$$N = \frac{K(\Omega_p/\Omega_s)K\left(\sqrt{1-(\epsilon^2/\delta^2)}\right)}{K(\epsilon/\delta)K\left(\sqrt{1-(\Omega_p/\Omega_s)^2}\right)} \tag{10.3.65}$$

where $K(x)$ is the complete elliptic integral of the first kind, defined as

$$K(x) = \int_0^{\pi/2} \frac{d\theta}{\sqrt{1-x^2\sin^2\theta}} \tag{10.3.66}$$

and $\delta_2 = 1/\sqrt{1+\delta^2}$. Values of this integral have been tabulated in a number of texts [e.g., the books by Jahnke and Emde (1945) and Dwight (1957)]. The passband ripple is $10\log_{10}(1+\epsilon^2)$.

We shall not attempt to describe elliptic functions in any detail because such a discussion would take us too far afield. Suffice to say that computer programs are available for designing elliptic filters from the frequency specifications indicated above.

In view of the optimality of elliptic filters, the reader may question the reason for considering the class of Butterworth or the class of Chebyshev filters in practical applications. One important reason that these other types of filters might be preferable in some applications is that they possess better phase response characteristics. The phase response of elliptic filters is more nonlinear in the passband than a comparable Butterworth filter or a Chebyshev filter, especially near the band edge.

Bessel filters. Bessel filters are a class of all-pole filters that are characterized by the system function

$$H(s) = \frac{1}{B_N(s)} \tag{10.3.67}$$

where $B_N(s)$ is the Nth-order Bessel polynomial. These polynomials can be expressed in the form

$$B_N(s) = \sum_{k=0}^{N} a_k s^k \tag{10.3.68}$$

where the coefficients $\{a_k\}$ are given as

$$a_k = \frac{(2N-k)!}{2^{N-k}k!(N-k)!}, \qquad k = 0, 1, \ldots, N \tag{10.3.69}$$

Alternatively, the Bessel polynomials may be generated recursively from the relation

$$B_N(s) = (2N-1)B_{N-1}(s) + s^2 B_{N-2}(s) \tag{10.3.70}$$

with $B_0(s) = 1$ and $B_1(s) = s + 1$ as initial conditions.

An important characteristic of Bessel filters is the linear-phase response over the passband of the filter. For example, Fig. 10.3.15 shows a comparison of the magnitude

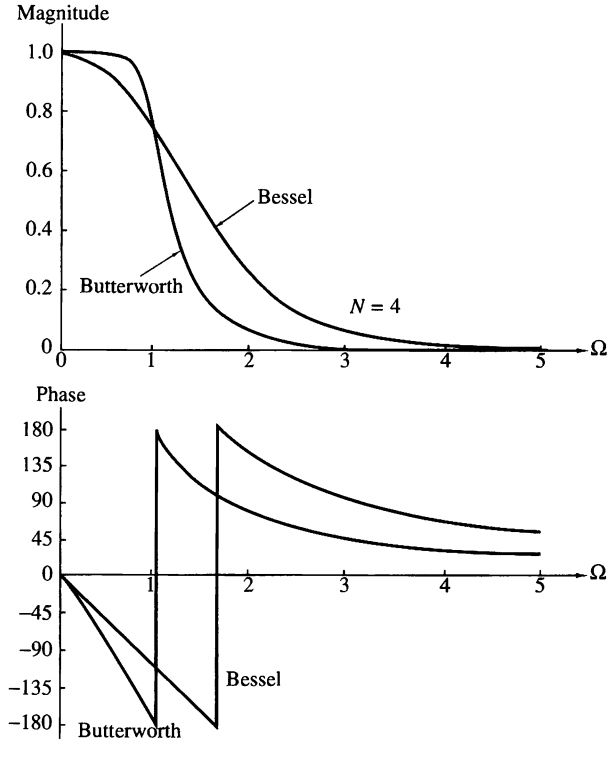


Figure 10.3.15
Magnitude and phase responses of Bessel and Butterworth filters of order $N = 4$.

and phase responses of a Bessel filter and Butterworth filter of order $N = 4$. We note that the Bessel filter has a larger transition bandwidth, but its phase is linear within the passband. However, we should emphasize that the linear-phase chacteristics of the analog filter are destroyed in the process of converting the filter into the digital domain by means of the transformations described previously.

10.3.5 Some Examples of Digital Filter Designs Based on the Bilinear Transformation

In this section we present several examples of digital filter designs obtained from analog filters by applying the bilinear transformation to convert $H(s)$ to $H(z)$. These filter designs are performed with the aid of one of several software packages now available for use on a personal computer.

A lowpass filter is designed to meet specifications of a maximum ripple of $\frac{1}{2}$ dB in the passband, 60-dB attenuation in the stopband, a passband edge frequency of $\omega_p = 0.25\pi$, and a stopband edge frequency of $\omega_s = 0.30\pi$.

A Butterworth filter of order $N = 37$ is required to satisfy the specifications. Its frequency response characteristics are illustrated in Fig. 10.3.16. If a Chebyshev filter

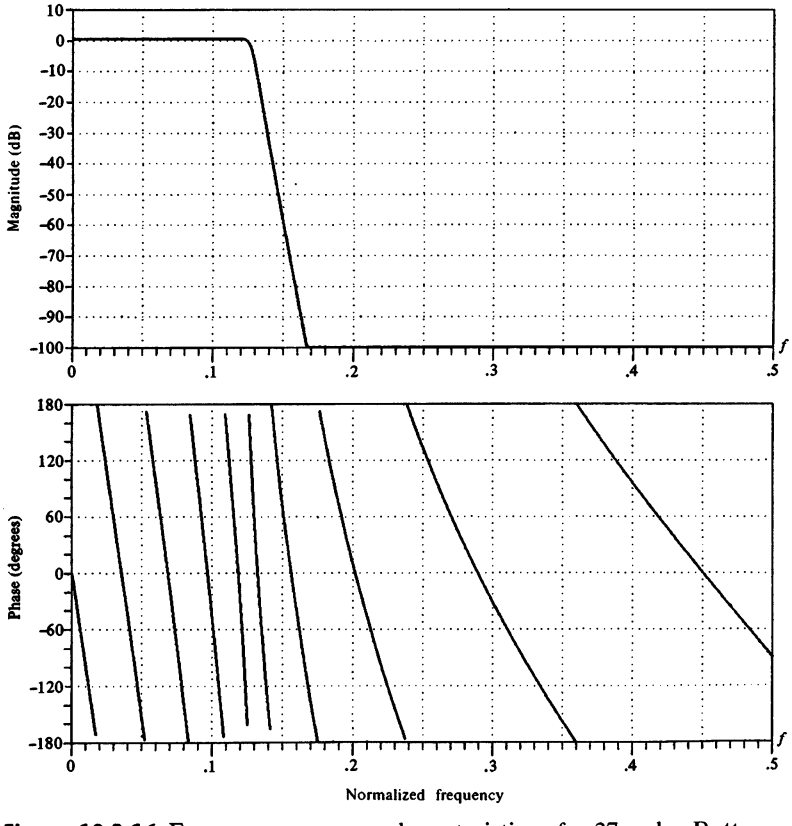


Figure 10.3.16 Frequency response characteristics of a 37-order Butterworth filter.

is used, a filter of order $N = 13$ satisfies the specifications. The frequency response characteristics for a type I Chebyshev filter are shown in Fig. 10.3.17. The filter has a passband ripple of 0.31 dB. Finally, an elliptic filter of order $N = 7$ is designed which also satisfies the specifications. For illustrative purposes, we show in Table 10.6, the numerical values for the filter parameters, and the resulting frequency specifications are shown in Fig. 10.3.18. The following notation is used for the parameters in the function $H(z)$:

$$H(z) = \prod_{i=1}^{K} \frac{b(i,0) + b(i,1)z^{-1} + b(i,2)z^{-2}}{1 + a(i,1)z^{-1} + a(i,2)z^{-2}} \tag{10.3.71}$$

Although we have described only lowpass analog filters in the preceding section, it is a simple matter to convert a lowpass analog filter into a bandpass, bandstop, or highpass analog filter by a frequency transformation, as is described in Section 10.4. The bilinear transformation is then applied to convert the analog filter into an equivalent digital filter. As in the case of the lowpass filters described above, the entire design can be carried out on a computer.

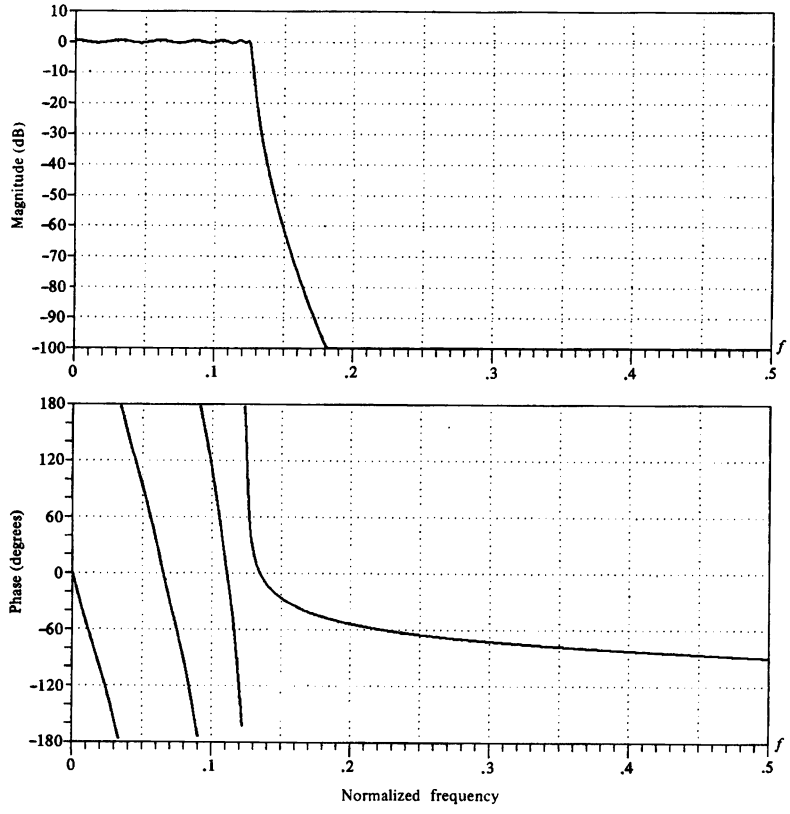


Figure 10.3.17 Frequency response characteristics of a 13-order type I Chebyshev filter.

TABLE 10.6 Filter Coefficients for a 7-Order Elliptic Filter

```
            INFINITE IMPULSE RESPONSE (IIR)
               ELLIPTIC LOWPASS FILTER
              UNQUANTIZED COEFFICIENTS
FILTER ORDER = 7
SAMPLING FREQUENCY = 2.000 KILOHERTZ
I.  A(I, 1)     A(I, 2)   B(I, 0)   B(I, 1)    B(I, 2)
1    -.790103   .000000   .104948    .104948   .000000
2   -1.517223   .714088   .102450   -.007817   .102232
3   -1.421773   .861895   .420100   -.399842   .419864
4   -1.387447   .962252   .714929   -.826743   .714841
     *** CHARACTERISTICS OF DESIGNED FILTER ***
                  BAND 1      BAND 2
LOWER BAND EDGE    .00000      .30000
UPPER BAND EDGE    .25000     1.00000
NOMINAL GAIN      1.00000      .00000
NOMINAL RIPPLE     .05600      .00100
MAXIMUM RIPPLE     .04910      .00071
RIPPLE IN DB       .41634   -63.00399
```

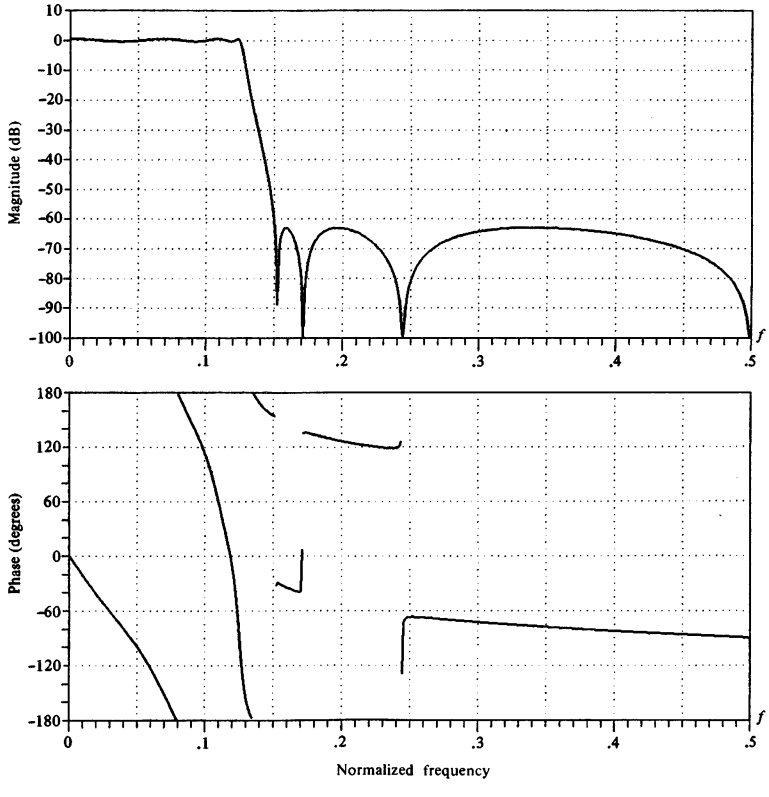


Figure 10.3.18 Frequency response characteristics of a 7-order elliptic filter.

10.4 Frequency Transformations

The treatment in the preceding section is focused primarily on the design of lowpass IIR filters. If we wish to design a highpass or a bandpass or a bandstop filter, it is a simple matter to take a lowpass prototype filter (Butterworth, Chebyshev, elliptic, Bessel) and perform a frequency transformation.

One possibility is to perform the frequency transformation in the analog domain and then to convert the analog filter into a corresponding digital filter by a mapping of the s-plane into the z-plane. An alternative approach is first to convert the analog lowpass filter into a lowpass digital filter and then to transform the lowpass digital filter into the desired digital filter by a digital transformation. In general, these two approaches yield different results, except for the bilinear transformation, in which case the resulting filter designs are identical. These two approaches are described below.

10.4.1 Frequency Transformations in the Analog Domain

First, we consider frequency transformations in the analog domain. Suppose that we have a lowpass filter with passband edge frequency Ω_p and we wish to convert it to another lowpass filter with passband edge frequency Ω_p'. The transformation that accomplishes this is

$$s \longrightarrow \frac{\Omega_p}{\Omega_p'} s, \qquad \text{(lowpass to lowpass)} \tag{10.4.1}$$

Thus we obtain a lowpass filter with system function $H_l(s) = H_p[(\Omega_p/\Omega_p')s]$, where $H_p(s)$ is the system function of the prototype filter with passband edge frequency Ω_p.

If we wish to convert a lowpass filter into a highpass filter with passband edge frequency Ω_p', the desired transformation is

$$s \longrightarrow \frac{\Omega_p \Omega_p'}{s}, \qquad \text{(lowpass to highpass)} \tag{10.4.2}$$

The system function of the highpass filter is $H_h(s) = H_p(\Omega_p \Omega_p'/s)$.

The transformation for converting a lowpass analog filter with passband edge frequency Ω_c into a band filter, having a lower band edge frequency Ω_l and an upper band edge frequency Ω_u, can be accomplished by first converting the lowpass filter into another lowpass filter having a band edge frequency $\Omega_p' = 1$ and then performing the transformation

$$s \longrightarrow \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}, \qquad \text{(lowpass to bandpass)} \tag{10.4.3}$$

Equivalently, we can accomplish the same result in a single step by means of the transformation

$$s \longrightarrow \Omega_p \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}, \qquad \text{(lowpass to bandpass)} \tag{10.4.4}$$

TABLE 10.7 Frequency Transformations for Analog Filters (Prototype Lowpass Filter Has Band Edge Frequency Ω_p)

Type of transformation	Transformation	Band edge frequencies of new filter
Lowpass	$s \longrightarrow \frac{\Omega_p}{\Omega_p'} s$	Ω_p'
Highpass	$s \longrightarrow \frac{\Omega_p \Omega_p'}{s}$	Ω_p'
Bandpass	$s \longrightarrow \Omega_p \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}$	Ω_l, Ω_u
Bandstop	$s \longrightarrow \Omega_p \frac{s(\Omega_u - \Omega_c)}{s^2 + \Omega_u \Omega_l}$	Ω_l, Ω_u

where

$$\Omega_l = \text{lower band edge frequency}$$

$$\Omega_u = \text{upper band edge frequency}$$

Thus we obtain

$$H_b(s) = H_p\left(\Omega_p \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}\right)$$

Finally, if we wish to convert a lowpass analog filter with band-edge frequency Ω_p into a bandstop filter, the transformation is simply the inverse of (10.4.3) with the additional factor Ω_p serving to normalize for the band-edge frequency of the lowpass filter. Thus the transformation is

$$s \longrightarrow \Omega_p \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l}, \qquad \text{(lowpass to bandstop)} \tag{10.4.5}$$

which leads to

$$H_{bs}(s) = H_p\left(\Omega_p \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l}\right)$$

The mappings in (10.4.1), (10.4.2), (10.4.3), and (10.4.5) are summarized in Table 10.7. The mappings in (10.4.4) and (10.4.5) are nonlinear and may appear to distort the frequency response characteristics of the lowpass filter. However, the effects of the nonlinearity on the frequency response are minor, primarily affecting the frequency scale but preserving the amplitude response characteristics of the filter. Thus an equiripple lowpass filter is transformed into an equiripple bandpass or bandstop or highpass filter.

EXAMPLE 10.4.1

Transform the single-pole lowpass Butterworth filter with system function

$$H(s) = \frac{\Omega_p}{s + \Omega_p}$$

into a bandpass filter with upper and lower band edge frequencies Ω_u and Ω_l, respectively.

Solution. The desired transformation is given by (10.4.4). Thus we have

$$H(s) = \frac{1}{\dfrac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)} + 1}$$

$$= \frac{(\Omega_u - \Omega_l)s}{s^2 + (\Omega_u - \Omega_l)s + \Omega_l \Omega_u}$$

The resulting filter has a zero at $s = 0$ and poles at

$$s = \frac{-(\Omega_u - \Omega_l) \pm \sqrt{\Omega_u^2 + \Omega_l^2 - 6\Omega_u \Omega_l}}{2}$$

10.4.2 Frequency Transformations in the Digital Domain

As in the analog domain, frequency transformations can be performed on a digital lowpass filter to convert it to either a bandpass, bandstop, or highpass filter. The transformation involves replacing the variable z^{-1} by a rational function $g(z^{-1})$, which must satisfy the following properties:

1. The mapping $z^{-1} \longrightarrow g(z^{-1})$ must map points inside the unit circle in the z-plane into itself.
2. The unit circle must also be mapped into itself.

Condition (2) implies that for $r = 1$,

$$e^{-j\omega} = g(e^{-j\omega}) \equiv g(\omega)$$

$$= |g(\omega)| e^{j \arg[g(\omega)]}$$

It is clear that we must have $|g(\omega)| = 1$ for all ω. That is, the mapping must be all pass. Hence it is of the form

$$g(z^{-1}) = \pm \prod_{k=1}^{n} \frac{z^{-1} - a_k}{1 - a_k z^{-1}} \tag{10.4.6}$$

where $|a_k| < 1$ to ensure that a stable filter is transformed into another stable filter (i.e., to satisfy condition 1).

From the general form in (10.4.6), we obtain the desired set of digital transformations for converting a prototype digital lowpass filter into either a bandpass, a bandstop, a highpass, or another lowpass digital filter. These transformations are tabulated in Table 10.8.

TABLE 10.8 Frequency Transformation for Digital Filters (Prototype Lowpass Filter Has Band Edge Frequency ω_p)

Type of transformation	Transformation	Parameters
Lowpass	$z^{-1} \longrightarrow \dfrac{z^{-1}-a}{1-az^{-1}}$	ω_p' = band edge frequency new filter $a = \dfrac{\sin[(\omega_p-\omega_p')/2]}{\sin[(\omega_p+\omega_p')/2]}$
Highpass	$z^{-1} \longrightarrow -\dfrac{z^{-1}+a}{1+az^{-1}}$	ω_p' = band edge frequency new filter $a = -\dfrac{\cos[(\omega_p+\omega_p')/2]}{\cos[(\omega_p-\omega_p')/2]}$
Bandpass	$z^{-1} \longrightarrow -\dfrac{z^{-2}-a_1z^{-1}+a_2}{a_2z^{-2}-a_1z^{-1}+1}$	ω_l = lower band edge frequency ω_u = upper band edge frequency $a_1 = 2\alpha K/(K+1)$ $a_2 = (K-1)/(K+1)$ $\alpha = \dfrac{\cos[(\omega_u+\omega_l)/2]}{\cos[(\omega_u-\omega_l)/2]}$ $K = \cot\dfrac{\omega_u-\omega_l}{2}\tan\dfrac{\omega_p}{2}$
Bandstop	$z^{-1} \longrightarrow \dfrac{z^{-2}-a_1z^{-1}+a_2}{a_2z^{-2}-a_1z^{-1}+1}$	ω_l = lower band edge frequency ω_u = upper band edge frequency $a_1 = 2\alpha/(K+1)$ $a_2 = (1-K)/(1+K)$ $\alpha = \dfrac{\cos[(\omega_u+\omega_l)/2]}{\cos[(\omega_u-\omega_l)/2]}$ $K = \tan\dfrac{\omega_u-\omega_l}{2}\tan\dfrac{\omega_p}{2}$

EXAMPLE 10.4.2

Convert the single-pole lowpass Butterworth filter with system function

$$H(z) = \frac{0.245(1+z^{-1})}{1-0.509z^{-1}}$$

into a bandpass filter with upper and lower cutoff frequencies ω_u and ω_l, respectively. The lowpass filter has 3-dB bandwidth, $\omega_p = 0.2\pi$ (see Example 10.3.5).

Solution. The desired transformation is

$$z^{-1} \longrightarrow -\frac{z^{-2}-a_1z^{-1}+a_2}{a_2z^{-2}-a_1z^{-1}+1}$$

where a_1 and a_2 are defined in Table 10.8. Substitution into $H(z)$ yields

$$H(z) = \frac{0.245\left[1 - \dfrac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}\right]}{1 + 0.509\left(\dfrac{z^{-2} - a_1 z^{-1} + a_2}{a_2 z^{-2} - a_1 z^{-1} + 1}\right)}$$

$$= \frac{0.245(1 - a_2)(1 - z^{-2})}{(1 + 0.509a_2) - 1.509a_1 z^{-1} + (a_2 + 0.509)z^{-2}}$$

Note that the resulting filter has zeros at $z = \pm 1$ and a pair of poles that depend on the choice of ω_u and ω_l.

For example, suppose that $\omega_u = 3\pi/5$ and $\omega_l = 2\pi/5$. Since $\omega_p = 0.2\pi$, we find that $K = 1$, $a_2 = 0$, and $a_1 = 0$. Then

$$H(z) = \frac{0.245(1 - z^{-2})}{1 + 0.509z^{-2}}$$

This filter has poles at $z = \pm j0.713$ and hence resonates at $\omega = \pi/2$.

Since a frequency transformation can be performed either in the analog domain or in the digital domain, the filter designer has a choice as to which approach to take. However, some caution must be exercised depending on the types of filters being designed. In particular, we know that the impulse invariance method and the mapping of derivatives are inappropriate to use in designing highpass and many bandpass filters, due to the aliasing problem. Consequently, one would not employ an analog frequency transformation followed by conversion of the result into the digital domain by use of these two mappings. Instead, it is much better to perform the mapping from an analog lowpass filter into a digital lowpass filter by either of these mappings, and then to perform the frequency transformation in the digital domain. Thus the problem of aliasing is avoided.

In the case of the bilinear transformation, where aliasing is not a problem, it does not matter whether the frequency transformation is performed in the analog domain or in the digital domain. In fact, in this case only, the two approaches result in identical digital filters.

10.5 Summary and References

We have described in some detail the most important techniques for designing FIR and IIR digital filters based either on frequency-domain specifications expressed in terms of a desired frequency response $H_d(\omega)$ or on the desired impulse response $h_d(n)$.

As a general rule, FIR filters are used in applications where there is a need for a linear-phase filter. This requirement occurs in many applications, especially in telecommunications, where there is a requirement to separate (demultiplex) signals such as data that have been frequency-division multiplexed, without distorting these

signals in the process of demultiplexing. Of the several methods described for designing FIR filters, the frequency-sampling design method and the optimum Chebyshev approximation method yield the best designs.

IIR filters are generally used in applications where some phase distortion is tolerable. Of the class of IIR filters, elliptic filters are the most efficient to implement in the sense that for a given set of specifications, an elliptic filter has a lower order or fewer coefficients than any other IIR filter type. When compared with FIR filters, elliptic filters are also considerably more efficient. In view of this, one might consider the use of an elliptic filter to obtain the desired frequency selectivity, followed then by an all-pass phase equalizer that compensates for the phase distortion in the elliptic filter. However, attempts to accomplish this have resulted in filters with a number of coefficients in the cascade combination that equaled or exceeded the number of coefficients in an equivalent linear-phase FIR filter. Consequently, no reduction in complexity is achievable in using phase-equalized elliptic filters.

Such a rich literature now exists on the design of digital filters that it is not possible to cite all the important references. We shall cite only a few. Some of the early work on digital filter design was done by Kaiser (1963, 1966), Steiglitz (1965), Golden and Kaiser (1964), Rader and Gold (1967a), Shanks (1967), Helms (1968), Gibbs (1969, 1970), and Gold and Rader (1969).

The design of analog filters is treated in the classic books by Storer (1957), Guillemin (1957), Weinberg (1962), and Daniels (1974).

The frequency-sampling method for filter design was first proposed by Gold and Jordan (1968, 1969), and optimized by Rabiner et al. (1970). Additional results were published by Herrmann (1970), Herrmann and Schuessler (1970a), and Hofstetter et al. (1971). The Chebyshev (minimax) approximation method for designing linear-phase FIR filters was proposed by Parks and McClellan (1972a,b) and discussed further by Rabiner et al. (1975). The design of elliptic digital filters is treated in the book by Gold and Rader (1969) and in the paper by Gray and Markel (1976). The latter includes a computer program for designing digital elliptic filters.

The use of frequency transformations in the digital domain was proposed by Constantinides (1967, 1968, 1970). These transformations are appropriate only for IIR filters. The reader should note that when these transformations are applied to a lowpass FIR filter, the resulting filter is IIR.

Direct design techniques for digital filters have been considered in a number of papers, including Shanks (1967), Burrus and Parks (1970), Steiglitz (1970), Deczky (1972), Brophy and Salazar (1973), and Bandler and Bardakjian (1973).

Problems

10.1 Design an FIR linear-phase, digital filter approximating the ideal frequency response

$$H_d(\omega) = \begin{cases} 1, & \text{for } |\omega| \leq \dfrac{\pi}{6} \\ 0, & \text{for } \dfrac{\pi}{6} < |\omega| \leq \pi \end{cases}$$

(a) Determine the coefficients of a 25-tap filter based on the window method with a rectangular window.

(b) Determine and plot the magnitude and phase response of the filter.

(c) Repeat parts (a) and (b) using the Hamming window.

(d) Repeat parts (a) and (b) using a Bartlett window.

10.2 Repeat Problem 10.1 for a bandstop filter having the ideal response

$$H_d(\omega) = \begin{cases} 1, & \text{for } |\omega| \leq \dfrac{\pi}{6} \\ 0, & \text{for } \dfrac{\pi}{6} < |\omega| < \dfrac{\pi}{3} \\ 1, & \text{for } \dfrac{\pi}{3} \leq |\omega| \leq \pi \end{cases}$$

10.3 Redesign the filter of Problem 10.1 using the Hanning and Blackman windows.

10.4 Redesign the filter of Problem 10.2 using the Hanning and Blackman windows.

10.5 Determine the unit sample response $\{h(n)\}$ of a linear-phase FIR filter of length $M = 4$ for which the frequency response at $\omega = 0$ and $\omega = \pi/2$ is specified as

$$H_r(0) = 1, \qquad H_r\left(\frac{\pi}{2}\right) = \frac{1}{2}$$

10.6 Determine the coefficients $\{h(n)\}$ of a linear-phase FIR filter of length $M = 15$ which has a symmetric unit sample response and a frequency response that satisfies the condition

$$H_r\left(\frac{2\pi k}{15}\right) = \begin{cases} 1, & k = 0, 1, 2, 3 \\ 0, & k = 4, 5, 6, 7 \end{cases}$$

10.7 Repeat the filter design problem in Problem 10.6 with the frequency response specifications

$$H_r\left(\frac{2\pi k}{15}\right) = \begin{cases} 1, & k = 0, 1, 2, 3 \\ 0.4, & k = 4 \\ 0, & k = 5, 6, 7 \end{cases}$$

10.8 The ideal analog differentiator is described by

$$y_a(t) = \frac{dx_a(t)}{dt}$$

where $x_a(t)$ is the input and $y_a(t)$ the output signal.

(a) Determine its frequency response by exciting the system with the input $x_a(t) = e^{j2\pi Ft}$.

(b) Sketch the magnitude and phase response of an ideal analog differentiator bandlimited to B hertz.

(c) The ideal digital differentiator is defined as

$$H(\omega) = j\omega, \qquad |\omega| \leq \pi$$

Justify this definition by comparing the frequency response $|H(\omega)|$, $\measuredangle H(\omega)$ with that in part (b).

(d) By computing the frequency response $H(\omega)$, show that the discrete-time system

$$y(n) = x(n) - x(n-1)$$

is a good approximation of a differentiator at low frequencies.

(e) Compute the response of the system to the input

$$x(n) = A\cos(\omega_0 n + \theta)$$

10.9 Use the window method with a Hamming window to design a 21-tap differentiator as shown in Fig. P10.9. Compute and plot the magnitude and phase response of the resulting filter.

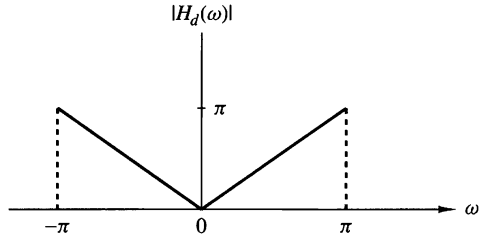


Figure P10.9

10.10 Use the bilinear transformation to convert the analog filter with system function

$$H(s) = \frac{s + 0.1}{(s + 0.1)^2 + 9}$$

into a digital IIR filter. Select $T = 0.1$ and compare the location of the zeros in $H(z)$ with the locations of the zeros obtained by applying the impulse invariance method in the conversion of $H(s)$.

10.11 Convert the analog bandpass filter designed in Example 10.4.1 into a digital filter by means of the bilinear transformation. Thereby derive the digital filter characteristic obtained in Example 10.4.2 by the alternative approach and verify that the bilinear transformation applied to the analog filter results in the same digital bandpass filter.

10.12 An ideal analog integrator is described by the system function $H_a(s) = 1/s$. A digital integrator with system function $H(z)$ can be obtained by use of the bilinear transformation. That is,

$$H(z) = \frac{T}{2}\frac{1 + z^{-1}}{1 - z^{-1}} \equiv H_a(s)|_{s=(2/T)(1-z^{-1})/(1+z^{-1})}$$

(a) Write the difference equation for the digital integrator relating the input $x(n)$ to the output $y(n)$.

(b) Roughly sketch the magnitude $|H_a(j\Omega)|$ and phase $\Theta(\Omega)$ of the analog integrator.

(c) It is easily verified that the frequency response of the digital integrator is

$$H(\omega) = -j\frac{T}{2}\frac{\cos(\omega/2)}{\sin(\omega/2)} = -j\frac{T}{2}\cot\frac{\omega}{2}$$

Roughly sketch $|H(\omega)|$ and $\theta(\omega)$.

(d) Compare the magnitude and phase characteristics obtained in parts (b) and (c). How well does the digital integrator match the magnitude and phase characteristics of the analog integrator?

(e) The digital integrator has a pole at $z = 1$. If you implement this filter on a digital computer, what restrictions might you place on the input signal sequence $x(n)$ to avoid computational difficulties?

10.13 A z-plane pole–zero plot for a certain digital filter is shown in Fig. P10.13. The filter has unity gain at dc.

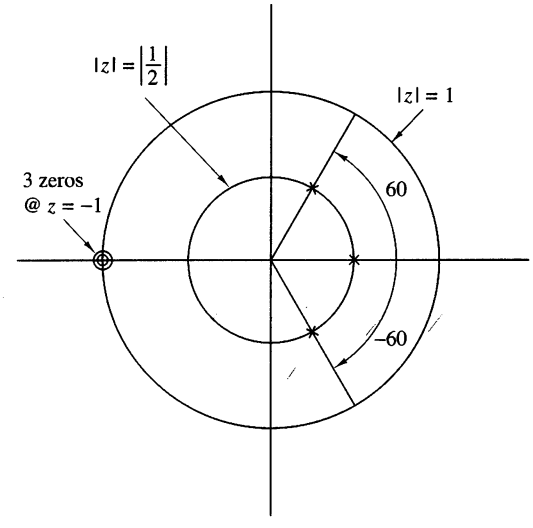


Figure P10.13

(a) Determine the system function in the form

$$H(z) = A\left[\frac{(1 + a_1z^{-1})(1 + b_1z^{-1} + b_2z^{-2})}{(1 + c_1z^{-1})(1 + d_1z^{-1} + d_2z^{-2})}\right]$$

giving numerical values for the parameters A, a_1, b_1, b_2, c_1, d_1, and d_2.

(b) Draw block diagrams showing numerical values for path gains in the following forms:

(a) Direct form II (canonic form)

(b) Cascade form (make each section canonic, with real coefficients)

10.14 Consider the pole–zero plot shown in Fig. P10.14.

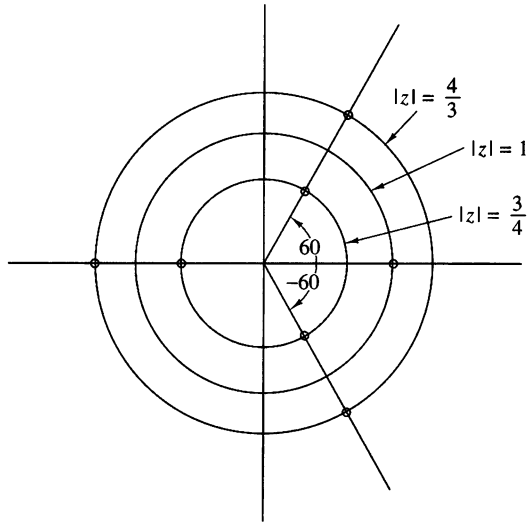


Figure P10.14

(a) Does it represent an FIR filter?

(b) Is it a linear-phase system?

(c) Give a direct-form realization that exploits all symmetries to minimize the number of multiplications. Show all path gains.

10.15 A digital low-pass filter is required to meet the following specifications:

Passband ripple: ≤ 1 dB
Passband edge: 4 kHz
Stopband attenuation: ≥ 40 dB
Stopband edge: 6 kHz
Sample rate: 24 kHz

The filter is to be designed by performing a bilinear transformation on an analog system function. Determine what order Butterworth, Chebyshev, and elliptic analog designs must be used to meet the specifications in the digital implementation.

10.16 An IIR digital lowpass filter is required to meet the following specifications:

Passband ripple (or peak-to-peak ripple): ≤ 0.5 dB
Passband edge: 1.2 kHz
Stopband attenuation: ≥ 40 dB
Stopband edge: 2.0 kHz
Sample rate: 8.0 kHz

Use the design formulas in the book to determine the required filter order for

(a) A digital Butterworth filter

(b) A digital Chebyshev filter

(c) A digital elliptic filter

10.17 Determine the system function $H(z)$ of the lowest-order Chebyshev digital filter that meets the following specifications:

(a) 1-dB ripple in the passband $0 \leq |\omega| \leq 0.3\pi$.

(b) At least 60 dB attentuation in the stopband $0.35\pi \leq |\omega| \leq \pi$. Use the bilinear transformation.

10.18 Determine the system function $H(z)$ of the lowest-order Chebyshev digital filter that meets the following specifications:

(a) $\frac{1}{2}$-dB ripple in the passband $0 \leq |\omega| \leq 0.24\pi$.

(b) At least 50-dB attenuation in the stopband $0.35\pi \leq |\omega| \leq \pi$. Use the bilinear transformation.

10.19 An analog signal $x(t)$ consists of the sum of two components $x_1(t)$ and $x_2(t)$. The spectral characteristics of $x(t)$ are shown in the sketch in Fig. P10.19. The signal $x(t)$ is bandlimited to 40 kHz and it is sampled at a rate of 100 kHz to yield the sequence $x(n)$.

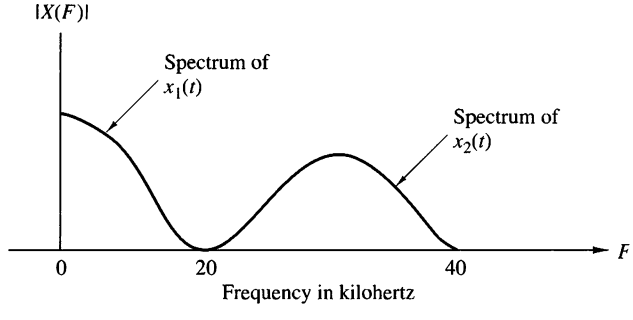


Figure P10.19

It is desired to suppress the signal $x_2(t)$ by passing the sequence $x(n)$ through a digital lowpass filter. The allowable amplitude distortion on $|X_1(f)|$ is $\pm 2\%(\delta_1 = 0.02)$ over the range $0 \leq |F| \leq 15$ kHz. Above 20 kHz, the filter must have an attenuation of at least 40 dB ($\delta_2 = 0.01$).

(a) Use the Remez exchange algorithm to design the *minimum*-order linear-phase FIR filter that meets the specifications above. From the plot of the magnitude characteristic of the filter frequency response, give the actual specifications achieved by the filter.

(b) Compare the order M obtained in part (a) with the approximate formulas given in equations (10.2.94) and (10.2.95).

(c) For the order M obtained in part (a), design an FIR digital lowpass filter using the window technique and the Hamming window. Compare the frequency response characteristics of this design with those obtained in part (a).

(d) Design the *minimum*-order elliptic filter that meets the given amplitude specifications. Compare the frequency response of the elliptic filter with that of the FIR filter in part (a).

(e) Compare the complexity of implementing the FIR filter in part (a) versus the elliptic filter obtained in part (d). Assume that the FIR filter is implemented in the direct form and the elliptic filter is implemented as a cascade of two-pole filters. Use storage requirements and the number of multiplications per output point in the comparison of complexity.

10.20 The impulse response of an analog filter is shown in Fig. P10.20.

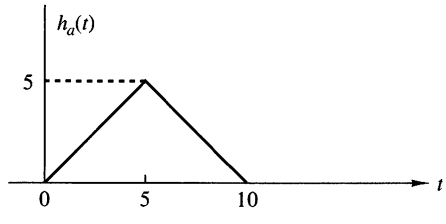


Figure P10.20

(a) Let $h(n) = h_a(nT)$, where $T = 1$, be the impulse response of a discrete-time filter. Determine the system function $H(z)$ and the frequency response $H(\omega)$ for this FIR filter.

(b) Sketch (roughly) $|H(\omega)$ and compare this frequency response characteristic with $|H_a(j\Omega)|$.

10.21 In this problem you will be comparing some of the characteristics of analog and digital implementations of the single-pole low-pass analog system

$$H_a(s) = \frac{\alpha}{s + \alpha} \Leftrightarrow h_a(t) = e^{-\alpha t}$$

(a) What is the gain at dc? At what radian frequency is the analog frequency response 3 dB down from its dc value? At what frequency is the analog frequency response zero? At what time has the analog impulse response decayed to $1/e$ of its initial value?

(b) Give the digital system function $H(z)$ for the impulse-invariant design for this filter. What is the gain at dc? Give an expression for the 3-dB radian frequency. At what (real-valued) frequency is the response zero? How many samples are there in the unit sample time-domain response before it has decayed to $1/e$ of its initial value?

(c) "Prewarp" the parameter α and perform the bilinear transformation to obtain the digital system function $H(z)$ from the analog design. What is the gain at dc? At what (real-valued) frequency is the response zero? Give an expression for the 3-dB radian frequency. How many samples are there in the unit sample time-domain response before it has decayed to $1/e$ of its initial value?

10.22 We wish to design a FIR bandpass filter having a duration $M = 201$. $H_d(\omega)$ represents the ideal characteristic of the noncausal bandpass filter as shown in Fig. P10.22.

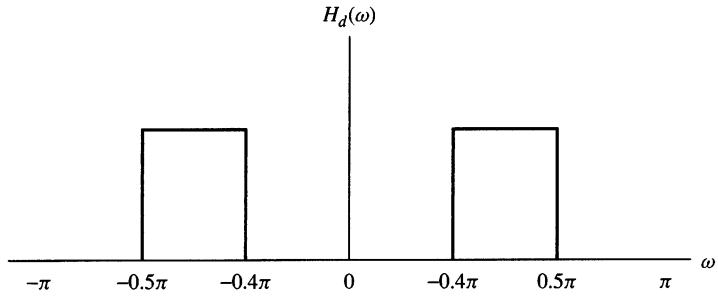


Figure P10.22

(a) Determine the unit sample (impulse) response $h_d(n)$ corresponding to $H_d(\omega)$.

(b) Explain how you would use the Hamming window

$$w(n) = 0.54 + 0.46\cos\left(\frac{2\pi n}{N-1}\right), \qquad -\frac{M-1}{2} \le n \le \frac{M-1}{2}$$

to design a FIR bandpass filter having an impulse response $h(n)$ for $0 \le n \le 200$.

(c) Suppose that you were to design the FIR filter with $M = 201$ by using the frequency-sampling technique in which the DFT coefficients $H(k)$ are specified instead of $h(n)$. Give the values of $H(k)$ for $0 \le k \le 200$ corresponding to $H_d(e^{j\omega})$ and indicate how the frequency response of the actual filter will differ from the ideal. Would the actual filter represent a good design? Explain your answer.

10.23 We wish to design a digital bandpass filter from a second-order analog lowpass Butterworth filter prototype using the bilinear transformation. The specifications on the digital filter are shown in Fig. P10.23(a). The cutoff frequencies (measured at the half-power points) for the digital filter should lie at $\omega_l = 5\pi/12$ and $\omega_u = 7\pi/12$.

The analog prototype is given by

$$H_a(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

with the half-power point at $\Omega = 1$.

(a) Determine the system function for the digital bandpass filter.

(b) Using the same specs on the digital filter as in part (a), determine which of the analog bandpass prototype filters shown in Fig. P10.23(b) could be transformed directly using the bilinear transformation to give the proper digital filter. Only the plot of the magnitude squared of the frequency is given.

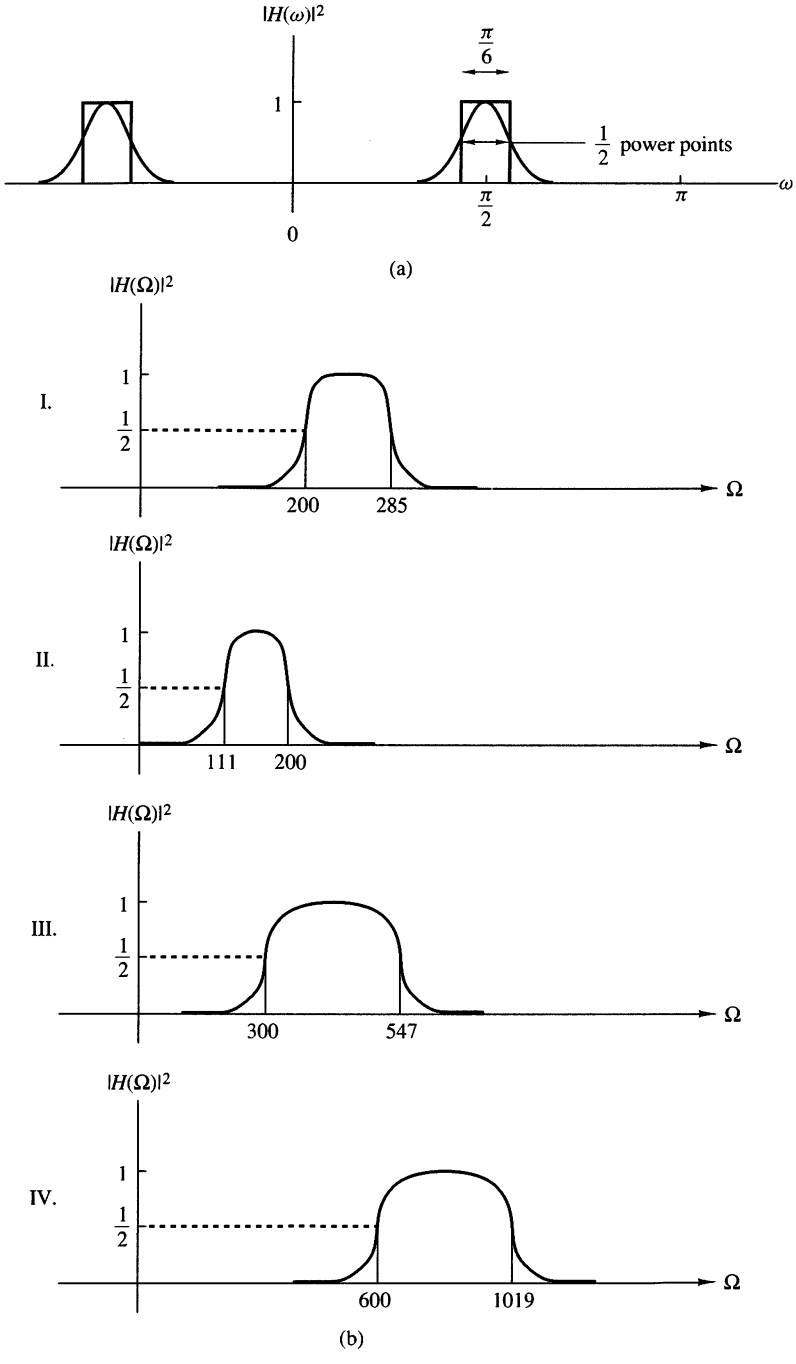


Figure P10.23

10.24 Figure P10.24 shows a digital filter designed using the frequency-sampling method.

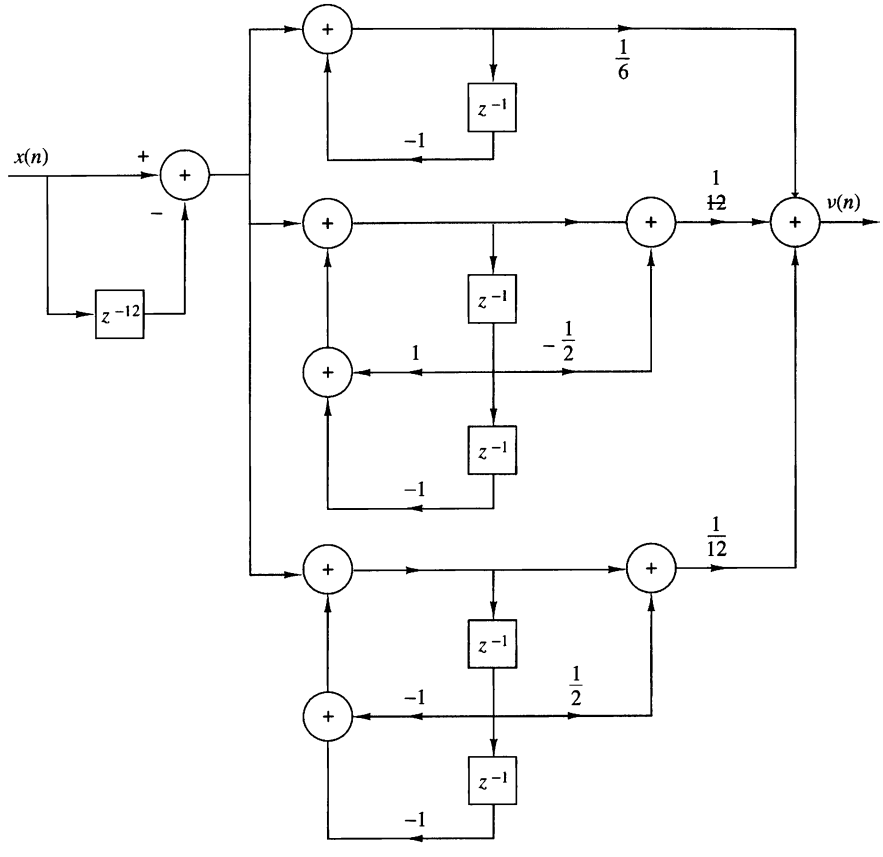


Figure P10.24

(a) Sketch a z-plane pole–zero plot for this filter.

(b) Is the filter lowpass, highpass, or bandpass?

(c) Determine the magnitude response $|H(\omega)|$ at the frequencies $\omega_k = \pi k/6$ for $k = 0, 1, 2, 3, 4, 5, 6$.

(d) Use the results of part (c) to sketch the magnitude response for $0 \leq \omega \leq \pi$ and confirm your answer to part (b).

10.25 An analog signal of the form $x_a(t) = a(t)\cos 2000\pi t$ is bandlimited to the range $900 \leq F \leq 1100$ Hz. It is used as an input to the system shown in Fig. P10.25.

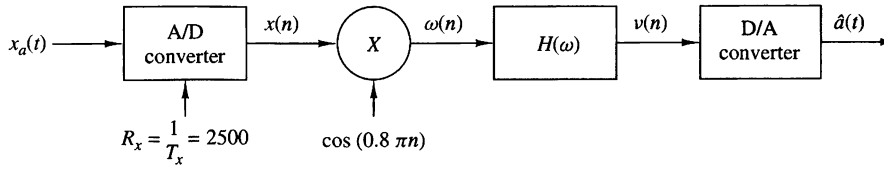


Figure P10.25

(a) Determine and sketch the spectra for the signals $x(n)$ and $w(n)$.

(b) Use a Hamming window of length $M = 31$ to design a lowpass linear phase FIR filter $H(\omega)$ that passes $\{a(n)\}$.

(c) Determine the sampling rate of the A/D converter that would allow us to eliminate the frequency conversion in Fig. P10.25.

10.26 *System identification* Consider an unknown LTI system and an FIR system model as shown in Fig. P10.26. Both systems are excited by the same input sequence $\{x(n)\}$. The problem is to determine the coefficients $\{h(n), 0 \leq n \leq M-1\}$ of the FIR model of the system to minimize the average squared error between the outputs of the two systems.

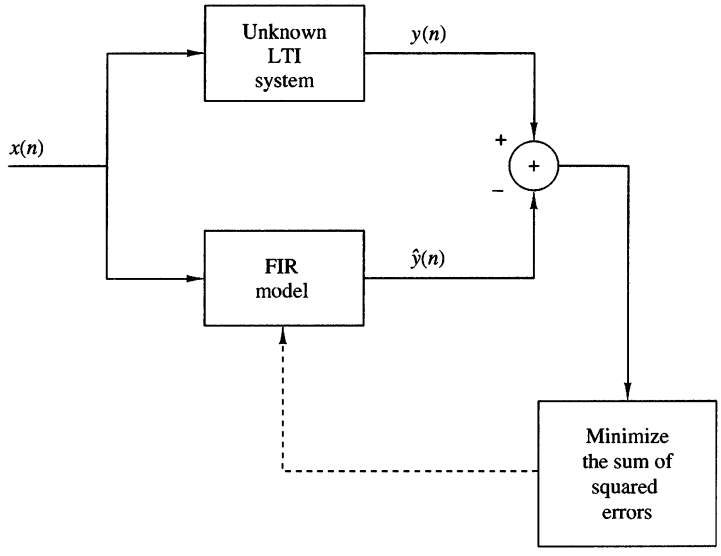


Figure P10.26

(a) Determine the equation for the FIR filter coefficients $\{h(n), 0 \leq n \leq M-1\}$ that minimize the least-squares error

$$\mathcal{E} = \sum_{n=0}^{N} [y(n) - \hat{y}(n)]^2$$

where

$$\hat{y}(n) = \sum_{k=0}^{M-1} h(k)x(n-k), \quad n = K, K+1, \ldots, N$$

and $N \gg M$.

(b) Repeat part (a) if the output of the unknown system is corrupted by an additive white noise $\{w(n)\}$ sequence with variance σ_w^2.

10.27 A linear time-invariant system has an input sequence $x(n)$ and an output sequence $y(n)$. The user has access only to the system output $y(n)$. In addition, the following information is available:

The input signal is periodic with a given fundamental period N and has a flat spectral envelope, that is,

$$x(n) = \sum_{k=0}^{N-1} c_k^x e^{j(2\pi/N)kn}, \qquad \text{all } n$$

where $c_k^x = 1$ for all k.

The system $H(z)$ is all pole, that is,

$$H(z) = \frac{1}{1 + \sum_{k=1}^{P} a_k z^{-k}}$$

but the order p and the coefficients $(a_k, 1 \leq k \leq p)$ are unknown. Is it possible to determine the order p and the numerical values of the coefficients $\{a_k, 1 \leq k \leq p\}$ by taking measurements on the output $y(n)$? If yes, explain how. Is this possible for every value of p?

10.28 *FIR system modeling* Consider an "unknown" FIR system with impulse response $h(n)$, $0 \leq n \leq 11$, given by

$$\begin{aligned}
h(0) = h(11) &= 0.309828 \times 10^{-1} \\
h(1) = h(10) &= 0.416901 \times 10^{-1} \\
h(2) = h(9) &= -0.577081 \times 10^{-1} \\
h(3) = h(8) &= -0.852502 \times 10^{-1} \\
h(4) = h(7) &= 0.147157 \times 10^{0} \\
h(5) = h(6) &= 0.449188 \times 10^{0}
\end{aligned}$$

A potential user has access to the input and output of the system but does not have any information about its impulse response other than that it is FIR. In an effort to determine the impulse response of the system, the user excites it with a zero-mean, random sequence $x(n)$ uniformly distributed in the range $[-0.5, 0.5]$, and records the signal $x(n)$ and the corresponding output $y(n)$ for $0 \leq n \leq 199$.

(a) By using the available information that the unknown system is FIR, the user employs the method of least squares to obtain an FIR model $h(n)$, $0 \leq n \leq M-1$. Set up the system of linear equations, specifying the parameters $h(0)$, $h(1), \ldots, h(M-1)$. Specify formulas we should use to determine the necessary autocorrelation and crosscorrelation values.

(b) Since the order of the system is unknown, the user decides to try models of different orders and check the corresponding total squared error. Clearly, this error will be zero (or very close to it if the order of the model becomes equal to the order of the system). Compute the FIR models $h_M(n)$, $0 \leq n \leq M-1$ for $M = 8, 9, 10, 11, 12, 13, 14$ as well as the corresponding total squared errors E_M, $M = 8, 9, \ldots, 14$. What do you observe?

(c) Determine and plot the frequency response of the system and the models for $M = 11, 12, 13$. Comment on the results.

(d) Suppose now that the output of the system is corrupted by additive noise, so instead of the signal $y(n)$, $0 \leq n \leq 199$, we have available the signal

$$v(n) = y(n) + 0.01w(n)$$

where $w(n)$ is a Gaussian random sequence with zero mean and variance $\sigma^2 = 1$.

Repeat part (b) of Problem 10.27 by using $v(n)$ instead of $y(n)$ and comment on the results. The quality of the model can be also determined by the quantity

$$Q = \frac{\sum_{n=0}^{\infty}[h(n) - \hat{h}(n)]^2}{\sum_{n=0}^{\infty} h^2(n)}$$

10.29 *Filter design by Padé approximation* Let the desired impulse response $h_d(n)$, $n \geq 0$, of an IIR filter be specified. The filter to be designed to approximate $\{h_d(n)\}$ has the system function

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} = \sum_{k=0}^{\infty} h(k) z^{-k}$$

$H(z)$ has $L = M + N + 1$ parameters, namely, the coefficients $\{a_k\}$ and $\{b_k\}$ to be determined. Suppose the input to the filter is $x(n) = \delta(n)$. Then, the response of the filter is $y(n) = h(n)$, and, hence,

$$\begin{aligned} h(n) = &-a_1 h(n-1) - a_2 h(n-2) - \ldots - a_N h(n-N) \\ &+ b_0\delta(n) + b_1\delta(n-1) + \ldots + b_M\delta(n-M) \end{aligned} \tag{1}$$

(a) Show that equation (1) reduces to

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \ldots - a_N h(n-N) + b_n, \quad 0 \leq n \leq M \tag{2}$$

(b) Show that for $n > M$, equation (1) reduces to

$$h(n) = -a_1 h(n-1) - a_2 h(n-2) - \ldots - a_N h(n-N), \quad n > M \tag{3}$$

(c) Explain how equations (2) and (3) can be used to determine $\{a_k\}$ and $\{b_k\}$ by letting $h(n) = h_d(n)$ for $0 \leq n \leq N + M$. (This filter design method in which $h(n)$ exactly matches the desired response $h_d(n)$ for $0 \leq n \leq M + N$ is called the Padé approximation method.)

10.30 Suppose the desired unit sample response is

$$h_d(n) = 2\left(\frac{1}{2}\right)^n u(n)$$

(a) Use the Padé approximation described in Problem 10.29 to determine $h(n)$.

(b) Compare the frequency response of $H(\omega)$ with that of the desired filter response $H_d(\omega)$.

10.31 *Shanks method for least-squares filter design* Suppose that we are given the desired response $h_d(n)$, $n \geq 0$, and we wish to determine the coefficients $\{a_k\}$ and $\{b_k\}$ of an IIR filter with system function

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

such that the sum of squared errors between $h_d(n)$ and $h(n)$ is minimized.

(a) If the input to $H(z)$ is $x(n) = \delta(n)$, what is the difference equation satisfied by the filter $H(z)$?

(b) Show that for $n > M$, an estimate of $h_d(n)$ is

$$\hat{h}_d(n) = -\sum_{k=1}^{N} a_k h_d(n-k)$$

and determine the equations for the coefficients $\{a_k\}$ by minimizing the sum of squared errors

$$\mathcal{E}_1 = \sum_{n=M+1}^{\infty} [h_d(n) - \hat{h}_d(n)]^2$$

Thus, the filter coefficients $\{a_k\}$ that specify the denominator of $H(z)$ are determined.

(c) To determine the parameters $\{b_k\}$ consider the system shown in Fig. P10.31, where

$$H_1(z) = \frac{1}{1 + \sum_{k=1}^{N} \hat{a}_k z^{-k}}$$

$$H_2(z) = \sum_{k=1}^{N} b_k z^{-k}$$

and $\{\hat{a}_k\}$ are the coefficients determined in part (b).

If the response of $H_1(z)$ to the input $\delta(n)$ is denoted as

$$v(n) = -\sum_{k=1}^{N} \hat{a}_k v(n-k) + \delta(n)$$

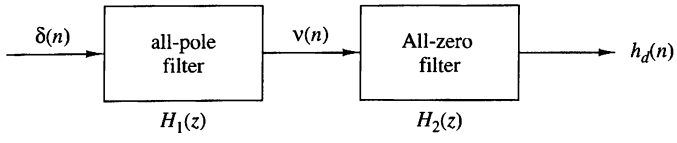


Figure P10.31

and the output of $H_2(z)$ is denoted as $\hat{h}_d(n)$, determine the equation for the parameters $\{b_k\}$ that minimize the sum of squared errors

$$\mathcal{E}_2 = \sum_{n=0}^{\infty}[h_d(n) - \hat{h}_d(n)]^2$$

(The preceding least-squares method for filter design is due to Shanks (1967).)

10.32 Use the Shanks method for filter design as described in Problem 10.31 to determine the parameters $\{a_k\}$ and $\{b_k\}$ of

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

when the desired response is the impulse response of the three-pole and three-zero type II lowpass Chebyshev digital filter having a system function

$$H_d(z) = \frac{0.3060(1 + z^{-1})(0.2652 - 0.09z^{-1} + 0.2652z^{-2})}{(1 - 0.3880z^{-1})(1 - 1.1318z^{-1} + 0.5387z^{-2})}$$

(a) Plot $h_d(n)$ and then observe that $h_d(n) \approx 0$ for $n > 50$.

(b) Determine the pole and zero positions for the filter $H(z)$ obtained by the Shanks method for $(N, M) = (3, 2)$, $(N, M) = (3, 3)$ and $(N, M) = (4, 3)$, and compare these results with the poles and zeros of $H_d(z)$. Comment on the similarities and differences.

CHAPTER

11

Multirate Digital Signal Processing

In many practical applications of digital signal processing, one is faced with the problem of changing the sampling rate of a signal, either increasing it or decreasing it by some amount. For example, in telecommunication systems that transmit and receive different types of signals (e.g., teletype, facsimile, speech, video, etc.), there is a requirement to process the various signals at different rates commensurate with the corresponding bandwidths of the signals. The process of converting a signal from a given rate to a different rate is called *sampling rate conversion*. In turn, systems that employ multiple sampling rates in the processing of digital signals are called *multirate digital signal processing systems*.

Sampling rate conversion of a digital signal can be accomplished in one of two general methods. One method is to pass the digital signal through a D/A converter, filter it if necessary, and then to resample the resulting analog signal at the desired rate (i.e., to pass the analog signal through an A/D converter). The second method is to perform the sampling rate conversion entirely in the digital domain.

One apparent advantage of the first method is that the new sampling rate can be arbitrarily selected and need not have any special relationship to the old sampling rate. A major disadvantage, however, is the signal distortion, introduced by the D/A converter in the signal reconstruction, and by the quantization effects in the A/D conversion. Sampling rate conversion performed in the digital domain avoids this major disadvantage.

In this chapter we describe sampling rate conversion and multirate signal processing in the digital domain. First we describe sampling rate conversion by a rational factor and present several methods for implementing the rate converter, including single-stage and multistage implementations. Then, we describe a method for sampling rate conversion by an arbitrary factor and discuss its implementation. We

present several applications of sampling rate conversion in multirate signal processing systems, which include the implementation of narrowband filters, digital filter banks, subband coding, transmultiplexers, and quadrature mirror filters.

11.1 Introduction

The process of sampling rate conversion can be developed and understood using the idea of "resampling after reconstruction." In this theoretical approach, a discrete-time signal is ideally reconstructed and the resulting continuous-time signal is resampled at a different sampling rate. This idea leads to a mathematical formulation that enables the realization of the entire process by means of digital signal processing.

Let $x(t)$ be a continuous-time signal that is sampled at a rate $F_x = 1/T_x$ to generate a discrete-time signal $x(nT_x)$. From the samples $x(nT_x)$ we can generate a continuous-time signal using the interpolation formula

$$y(t) = \sum_{n=-\infty}^{\infty} x(nT_x)g(t - nT_x) \tag{11.1.1}$$

If the bandwidth of $x(t)$ is less than $F_x/2$ and the interpolation function is given by

$$g(t) = \frac{\sin(\pi t/T_x)}{\pi t/T_x} \xleftrightarrow{\mathcal{F}} G(F) = \begin{cases} T_x, & |F| \le F_x/2 \\ 0, & \text{otherwise} \end{cases} \tag{11.1.2}$$

then $y(t) = x(t)$; otherwise $y(t) \neq x(t)$. In practice, perfect recovery of $x(t)$ is not possible because the infinite summation in (11.1.1) should be replaced by a finite summation.

To perform sampling rate conversion we simply evaluate (11.1.1) at time instants $t = mT_y$, where $F_y = 1/T_y$ is the desired sampling frequency. Therefore, the general formula for sampling rate conversion becomes

$$y(mT_y) = \sum_{n=-\infty}^{\infty} x(nT_x)g(mT_y - nT_x) \tag{11.1.3}$$

which expresses directly the samples of the desired sequence in terms of the samples of the original sequence and sampled values of the reconstruction function at positions $(mT_y - nT_x)$. The computation of $y(nT_y)$ requires (a) the input sequence $x(nT_x)$, (b) the reconstruction function $g(t)$, and (c) the time instants nT_x and mT_y of the input and output samples. The values $y(mT_y)$ calculated by this equation are accurate only if $F_y > F_x$. If $F_y < F_x$, we should filter out the frequency components of $x(t)$ above $F_y/2$ before resampling in order to prevent aliasing. Therefore, the sampling rate conversion formula (11.1.3) yields $y(mT_y) = x(mT_y)$ if we use (11.1.2) and $X(F) = 0$ for $|F| \ge \min\{F_x/2, F_y/2\}$.

If $T_y = T_x$, equation (11.1.3) becomes a convolution summation, which corresponds to an LTI system. To understand the meaning of (11.1.3) for $T_y \neq T_x$, we rearrange the argument of $g(t)$ as follows:

$$y(mT_y) = \sum_{n=-\infty}^{\infty} x(nT_x)g\left(T_x\left(\frac{mT_y}{T_x} - n\right)\right) \tag{11.1.4}$$

The term mT_y/T_x can be decomposed into an integer part k_m and a fractional part Δ_m, $0 \leq \Delta_m < 1$, as

$$\frac{mT_y}{T_x} = k_m + \Delta_m \tag{11.1.5}$$

where

$$k_m = \left\lfloor \frac{mT_y}{T_x} \right\rfloor \tag{11.1.6}$$

and

$$\Delta_m = \frac{mT_y}{T_x} - \left\lfloor \frac{mT_y}{T_x} \right\rfloor \tag{11.1.7}$$

The symbol $\lfloor a \rfloor$ denotes the largest integer contained in a. The quantity Δ_m specifies the position of the current sample within the sample period T_x. Substituting (11.1.5) into (11.1.4) we obtain

$$y(mT_y) = \sum_{n=-\infty}^{\infty} x(nT_x)g((k_m + \Delta_m - n)T_x) \tag{11.1.8}$$

If we change the index of summation in (11.1.8) from n to $k = k_m - n$, we have

$$\begin{aligned} y(mT_y) &= y((k_m + \Delta_m)T_x) \\ &= \sum_{k=-\infty}^{\infty} g(kT_x + \Delta_m T_x)x((k_m - k)T_x) \end{aligned} \tag{11.1.9}$$

Equation (11.1.9) provides the fundamental equation for the discrete-time implementation of sampling rate conversion. This process is illustrated in Figure 11.1.1. We note that (a) given T_x and T_y the input and output sampling times are fixed, (b) the function $g(t)$ is shifted for each m such that the value $g(\Delta_m T_x)$ is positioned at $t = mT_y$, and (c) the required values of $g(t)$ are determined at the input sampling times. For each value of m, the fractional interval Δ_m determines the impulse response coefficients whereas the index k_m specifies the corresponding input samples

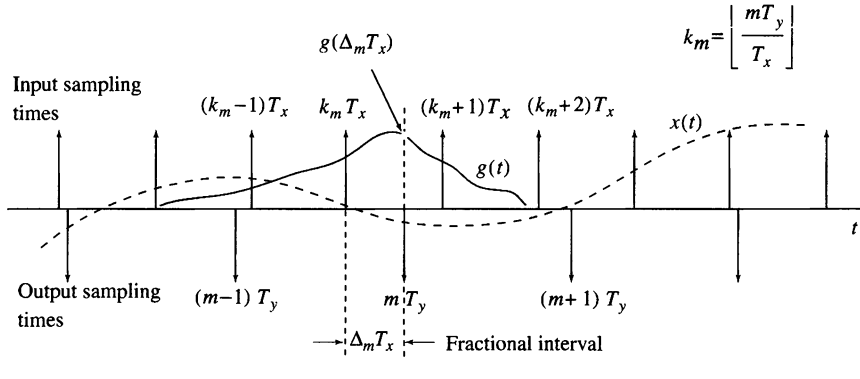


Figure 11.1.1 Illustration of timing relations for sampling rate conversion.

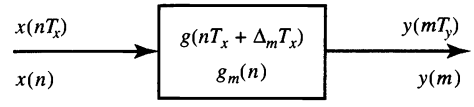


Figure 11.1.2
Discrete-time linear time-varying system for sampling rate conversion.

needed to compute the sample $y(mT_y)$. Since for any given value of m the index k_m is an integer number, $y(mT_y)$ is the convolution between the input sequence $x(nT_x)$ and an impulse response $g((n + \Delta_m)T_x)$. The difference between (11.1.8) and (11.1.9) is that the first shifts a "changing" reconstruction function whereas the second shifts the "fixed" input sequence.

The sampling rate conversion process defined by (11.1.9) is a discrete-time *linear and time-varying* system, because it requires a different impulse response

$$g_m(nT_x) = g((n + \Delta_m)T_x) \tag{11.1.10}$$

for each output sample $y(mT_y)$. Therefore, a new set of coefficients should be computed or retrieved from storage for the computation of *every* output sample (see Figure 11.1.2). This procedure may be inefficient when the function $g(t)$ is complicated and the number of required values is large. This dependence on m prohibits the use of recursive structures because the required past output values must be computed using an impulse response for the present value of Δ_m.

A significant simplification results when the ratio T_y/T_x is constrained to be a rational number, that is,

$$\frac{T_y}{T_x} = \frac{F_x}{F_y} = \frac{D}{I} \tag{11.1.11}$$

where D and I are relatively prime integers. To this end, we express the offset Δ_m as

$$\Delta_m = \frac{mD}{I} - \left\lfloor \frac{mD}{I} \right\rfloor = \frac{1}{I}\left(mD - \left\lfloor \frac{mD}{I} \right\rfloor I\right) = \frac{1}{I}(mD)_I \tag{11.1.12}$$

where $(k)_I$ denotes the value of k modulo I. From (11.1.12) it is clear that Δ_m can take on only I unique values $0, 1/I, \ldots, (I-1)/I$, so that there are only I distinct impulse responses possible. Since $g_m(nT_x)$ can take on I distinct sets of values, it is periodic in m; that is,

$$g_m(nT_x) = g_{m+rI}(nT_x), \quad r = 0, \pm 1, \pm 2, \ldots \tag{11.1.13}$$

Thus the system $g_m(nT_x)$ is a linear and *periodically time-varying* discrete-time system. Such systems have been widely studied for a wide range of applications (Meyers and Burrus, 1975). This is a great simplification compared to the *continuously* time-varying discrete-time system in (11.1.10).

To illustrate these concepts we consider two important special cases. We start with the process of reducing the sampling rate by an integer factor D, which is known as *decimation* or *downsampling*. If we set $T_y = DT_x$ in (11.1.3), we have

$$y(mT_y) = y(mDT_x) = \sum_{k=-\infty}^{\infty} x(kT_x)g((mD - k)T_x) \tag{11.1.14}$$

We note that the input signal and the impulse response are sampled with a period T_x. However, the impulse response is shifted at increments of $T_y = DT_x$ because we need to compute only one out of every D samples. Since $I = 1$, we have $\Delta_m = 0$ and therefore there is only one impulse response $g(nT_x)$, for all m. This process is illustrated in Figure 11.1.3 for $D = 2$.

We consider now the process of increasing the sampling rate by an integer factor I, which is called *upsampling* or *interpolation*. If we set $T_y = T_x/I$ in (11.1.3), we have

$$y(mT_y) = \sum_{k=-\infty}^{\infty} x(kT_x)g(m(T_x/I) - kT_x) \tag{11.1.15}$$

We note that both $x(t)$ and $g(t)$ are sampled with a period T_x; however, the impulse response is shifted at increments of $T_y = T_x/I$ for the computation of each output sample. This is required to "fill-in" an additional number of $(I - 1)$ samples within each period T_x. This is illustrated in Figure 11.1.4(a)–(b) for $I = 2$. Each "fractional shifting" requires that we resample $g(t)$, resulting in a new impulse response $g_m(nT_x) = g(nT_x + mT_x/I)$, for $m = 0, 1, \ldots, I-1$ in agreement with (11.1.14). Careful inspection of Figure 11.1.4(a)–(b) shows that if we determine an impulse response sequence $g(nT_y)$ and create a new sequence $v(nT_y)$ by inserting $(I - 1)$ zero samples between successive samples of $x(nT_x)$, we can compute $y(mT_y)$ as the convolution of the sequences $g(nT_y)$ and $x(nT_y)$. This idea is illustrated in Figure 11.1.4(c) for $I = 2$.

In the next few sections we discuss in more detail the properties, design, and structures for the implementation of sampling rate conversion entirely in the discrete-time domain. For convenience, we usually drop the sampling periods T_x and T_y from the argument of discrete-time signals. However, occasionally, it will be beneficial to the reader to reintroduce and think in terms of continuous-time quantities and units.

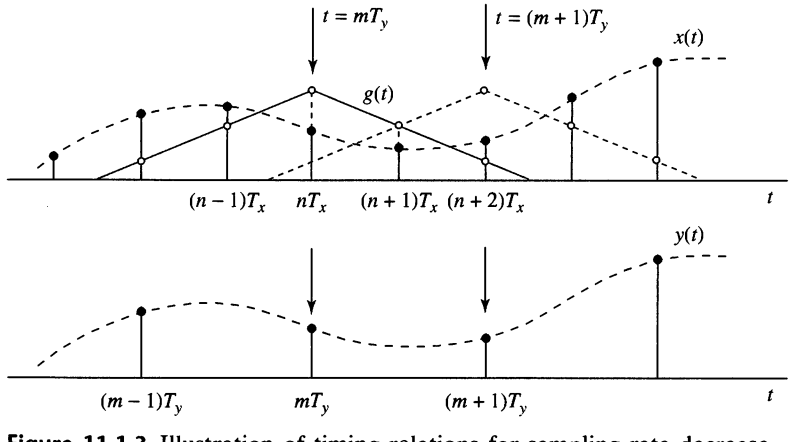


Figure 11.1.3 Illustration of timing relations for sampling rate decrease by an integer factor $D = 2$. A single impulse response, sampled with period T_x, is shifted at steps equal to $T_y = DT_x$ to generate the output samples.

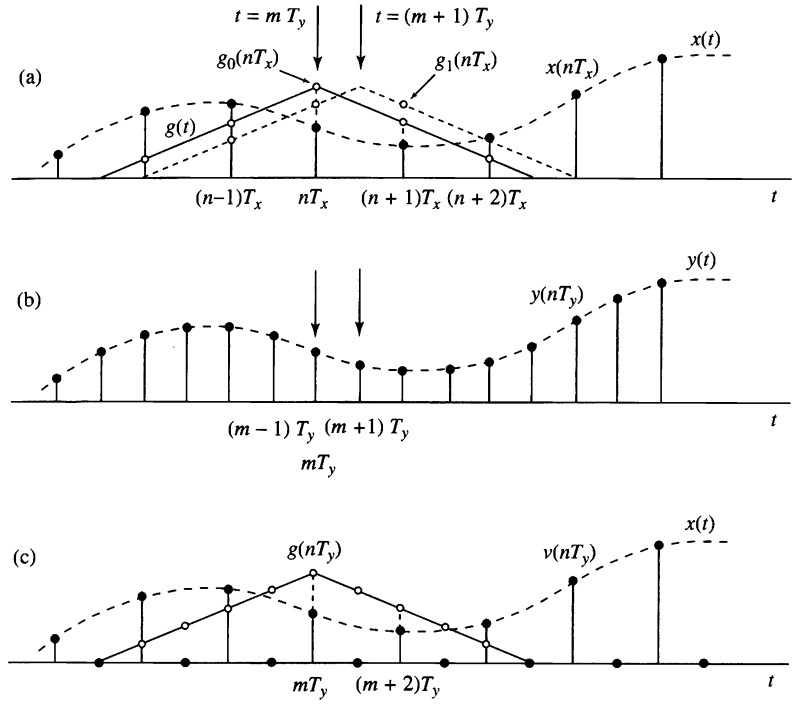


Figure 11.1.4 Illustration of timing relations for sampling rate increase by an integer factor $I = 2$. The approach in (a) requires one impulse response for the even-numbered and one for the odd-numbered output samples. The approach in (c) requires only one impulse response, obtained by interleaving the impulse responses in (a).

11.2 Decimation by a Factor D

Let us assume that the signal $x(n)$ with spectrum $X(\omega)$ is to be downsampled by an integer factor D. The spectrum $X(\omega)$ is assumed to be nonzero in the frequency interval $0 \leq |\omega| \leq \pi$ or, equivalently, $|F| \leq F_x/2$. We know that if we reduce the sampling rate simply by selecting every Dth value of $x(n)$, the resulting signal will be an aliased version of $x(n)$, with a folding frequency of $F_x/2D$. To avoid aliasing, we must first reduce the bandwidth of $x(n)$ to $F_{\max} = F_x/2D$ or, equivalently, to $\omega_{\max} = \pi/D$. Then we may downsample by D and thus avoid aliasing.

The decimation process is illustrated in Fig. 11.2.1. The input sequence $x(n)$ is passed through a lowpass filter, characterized by the impulse response $h(n)$ and a frequency response $H_D(\omega)$, which ideally satisfies the condition

$$H_D(\omega) = \begin{cases} 1, & |\omega| \leq \pi/D \\ 0, & \text{otherwise} \end{cases} \tag{11.2.1}$$

Thus the filter eliminates the spectrum of $X(\omega)$ in the range $\pi/D < \omega < \pi$. Of course, the implication is that only the frequency components of $x(n)$ in the range $|\omega| \leq \pi/D$ are of interest in further processing of the signal.

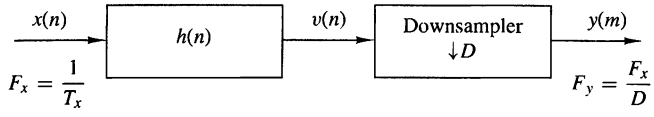


Figure 11.2.1 Decimation by a factor D.

The output of the filter is a sequence $v(n)$ given as

$$v(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{11.2.2}$$

which is then downsampled by the factor D to produce $y(m)$. Thus

$$\begin{aligned} y(m) &= v(mD) \\ &= \sum_{k=0}^{\infty} h(k)x(mD-k) \end{aligned} \tag{11.2.3}$$

Although the filtering operation on $x(n)$ is linear and time invariant, the downsampling operation in combination with the filtering results in a time-variant system. This is easily verified. Given the fact that $x(n)$ produces $y(m)$, we note that $x(n-n_0)$ does not imply $y(n-n_0)$ unless n_0 is a multiple of D. Consequently, the overall linear operation (linear filtering followed by downsampling) on $x(n)$ is not time invariant.

The frequency-domain characteristics of the output sequence $y(m)$ can be obtained by relating the spectrum of $y(m)$ to the spectrum of the input sequence $x(n)$. First, it is convenient to define a sequence $\tilde{v}(n)$ as

$$\tilde{v}(n) = \begin{cases} v(n), & n = 0, \pm D, \pm 2D, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{11.2.4}$$

Clearly, $\tilde{v}(n)$ can be viewed as a sequence obtained by multiplying $v(n)$ with a periodic train of impulses $p(n)$, with period D, as illustrated in Fig. 11.2.2. The discrete Fourier series representation of $p(n)$ is

$$p(n) = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi kn/D} \tag{11.2.5}$$

Hence

$$\tilde{v}(n) = v(n)p(n) \tag{11.2.6}$$

and

$$y(m) = \tilde{v}(mD) = v(mD)p(mD) = v(mD) \tag{11.2.7}$$

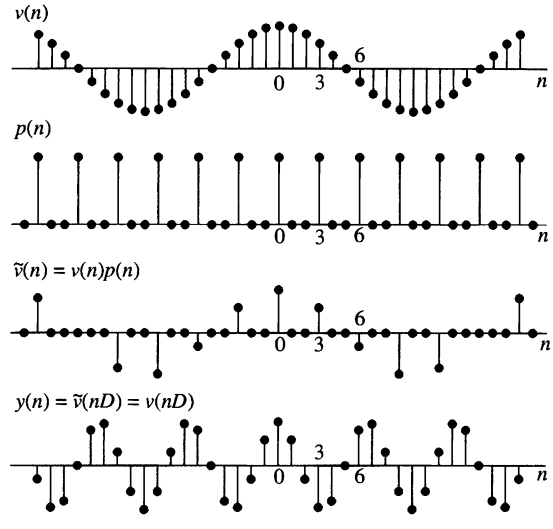


Figure 11.2.2
Steps required to facilitate the mathematical description of downsampling by a factor D, using a sinusoidal sequence for illustration.

Now the z-transform of the output sequence $y(m)$ is

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} y(m)z^{-m} \\ &= \sum_{m=-\infty}^{\infty} \tilde{v}(mD)z^{-m} \\ Y(z) &= \sum_{m=-\infty}^{\infty} \tilde{v}(m)z^{-m/D} \end{aligned} \tag{11.2.8}$$

where the last step follows from the fact that $\tilde{v}(m) = 0$, except at multiples of D. By making use of the relations in (11.2.5) and (11.2.6) in (11.2.8), we obtain

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} v(m) \left[\frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi mk/D} \right] z^{-m/D} \\ &= \frac{1}{D} \sum_{k=0}^{D-1} \sum_{m=-\infty}^{\infty} v(m)(e^{-j2\pi k/D} z^{1/D})^{-m} \\ &= \frac{1}{D} \sum_{k=0}^{D-1} V(e^{-j2\pi k/D} z^{1/D}) \\ &= \frac{1}{D} \sum_{k=0}^{D-1} H_D(e^{-j2\pi k/D} z^{1/D}) X(e^{-j2\pi k/D} z^{1/D}) \end{aligned} \tag{11.2.9}$$

where the last step follows from the fact that $V(z) = H_D(z)X(z)$.

By evaluating $Y(z)$ in the unit circle, we obtain the spectrum of the output signal $y(m)$. Since the rate of $y(m)$ is $F_y = 1/T_y$, the frequency variable, which we denote as ω_y, is in radians and is relative to the sampling rate F_y,

$$\omega_y = \frac{2\pi F}{F_y} = 2\pi F T_y \tag{11.2.10}$$

Since the sampling rates are related by the expression

$$F_y = \frac{F_x}{D} \tag{11.2.11}$$

it follows that the frequency variables ω_y and

$$\omega_x = \frac{2\pi F}{F_x} = 2\pi F T_x \tag{11.2.12}$$

are related by

$$\omega_y = D\omega_x \tag{11.2.13}$$

Thus, as expected, the frequency range $0 \leq |\omega_x| \leq \pi/D$ is stretched into the corresponding frequency range $0 \leq |\omega_y| \leq \pi$ by the downsampling process.

We conclude that the spectrum $Y(\omega_y)$, which is obtained by evaluating (11.2.9) on the unit circle, can be expressed as

$$Y(\omega_y) = \frac{1}{D}\sum_{k=0}^{D-1} H_D\left(\frac{\omega_y - 2\pi k}{D}\right) X\left(\frac{\omega_y - 2\pi k}{D}\right) \tag{11.2.14}$$

With a properly designed filter $H_D(\omega)$, the aliasing is eliminated and, consequently, all but the first term in (11.2.14) vanish. Hence

$$\begin{aligned} Y(\omega_y) &= \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) X\left(\frac{\omega_y}{D}\right) \\ &= \frac{1}{D} X\left(\frac{\omega_y}{D}\right) \end{aligned} \tag{11.2.15}$$

for $0 \leq |\omega_y| \leq \pi$. The spectra for the sequences $x(n)$, $v(n)$, and $y(m)$ are illustrated in Fig. 11.2.3.

EXAMPLE 11.2.1

Design a decimator that downsamples an input signal $x(n)$ by a factor $D = 2$. Use the Remez algorithm to determine the coefficients of the FIR filter that has a 0.1-dB ripple in the passband and is down by at least 30 dB in the stopband.

Solution. A filter length $M = 30$ achieves the design specifications given above. The frequency response is illustrated in Fig. 11.2.4. Note that the cutoff frequency is $\omega_c = \pi/2$.

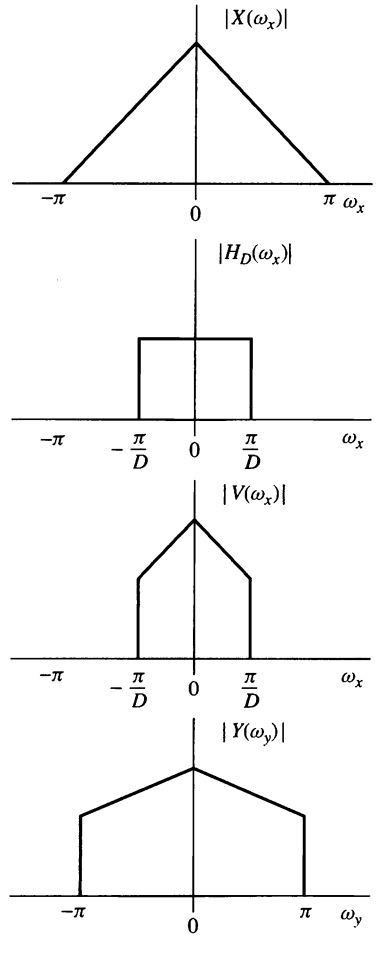


Figure 11.2.3
Spectra of signals in the decimation of $x(n)$ by a factor D.

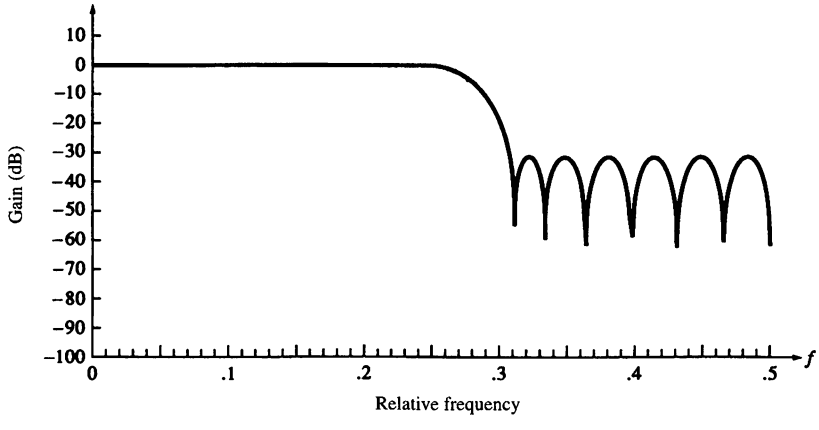


Figure 11.2.4 Magnitude response of linear-phase FIR filter of length $M = 30$ in Example 11.2.1.

11.3 Interpolation by a Factor I

An increase in the sampling rate by an integer factor of I can be accomplished by interpolating $I - 1$ new samples between successive values of the signal. The interpolation process can be accomplished in a variety of ways. We shall describe a process that preserves the spectral shape of the signal sequence $x(n)$.

Let $v(m)$ denote a sequence with a rate $F_y = IF_x$, which is obtained from $x(n)$ by adding $I - 1$ zeros between successive values of $x(n)$. Thus

$$v(m) = \begin{cases} x(m/I), & m = 0, \pm I, \pm 2I, \dots \\ 0, & \text{otherwise} \end{cases} \tag{11.3.1}$$

and its sampling rate is identical to the rate of $y(m)$. This sequence has a z-transform

$$\begin{aligned} V(z) &= \sum_{m=-\infty}^{\infty} v(m) z^{-m} \\ &= \sum_{m=-\infty}^{\infty} x(m) z^{-mI} \\ &= X(z^I) \end{aligned} \tag{11.3.2}$$

The corresponding spectrum of $v(m)$ is obtained by evaluating (11.3.2) on the unit circle. Thus

$$V(\omega_y) = X(\omega_y I) \tag{11.3.3}$$

where ω_y denotes the frequency variable relative to the new sampling rate F_y (i.e., $\omega_y = 2\pi F/F_y$). Now the relationship between sampling rates is $F_y = IF_x$ and hence, the frequency variables ω_x and ω_y are related according to the formula

$$\omega_y = \frac{\omega_x}{I} \tag{11.3.4}$$

The spectra $X(\omega_x)$ and $V(\omega_y)$ are illustrated in Fig. 11.3.1. We observe that the sampling rate increase, obtained by the addition of $I - 1$ zero samples between successive values of $x(n)$, results in a signal whose spectrum $V(\omega_y)$ is an I-fold periodic repetition of the input signal spectrum $X(\omega_x)$.

Since only the frequency components of $x(n)$ in the range $0 \le \omega_y \le \pi/I$ are unique, the images of $X(\omega)$ above $\omega_y = \pi/I$ should be rejected by passing the sequence $v(m)$ through a lowpass filter with frequency response $H_I(\omega_y)$ that ideally has the characteristic

$$H_I(\omega_y) = \begin{cases} C, & 0 \le |\omega_y| \le \pi/I \\ 0, & \text{otherwise} \end{cases} \tag{11.3.5}$$

where C is a scale factor required to properly normalize the output sequence $y(m)$. Consequently, the output spectrum is

$$Y(\omega_y) = \begin{cases} CX(\omega_y I), & 0 \le |\omega_y| \le \pi/I \\ 0, & \text{otherwise} \end{cases} \tag{11.3.6}$$

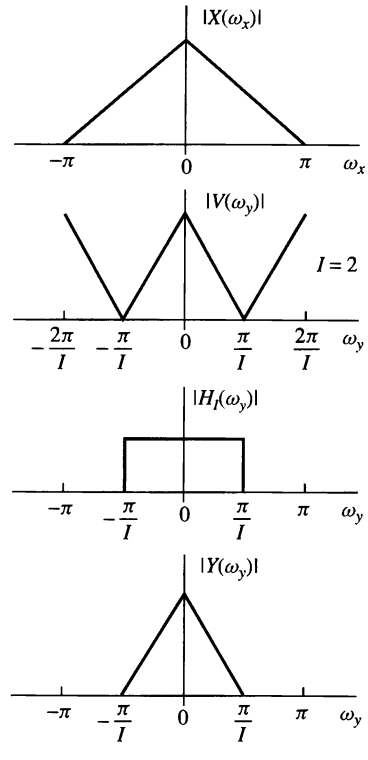


Figure 11.3.1
Spectra of $x(n)$ and $v(n)$ where $V(\omega_y) = X(\omega_y l)$.

The scale factor C is selected so that the output $y(m) = x(m/I)$ for $m = 0, \pm I, +2I, \ldots$. For mathematical convenience, we select the point $m = 0$. Thus

$$\begin{aligned} y(0) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} Y(\omega_y)d\omega_y \\ &= \frac{C}{2\pi}\int_{-\pi/I}^{\pi/I} X(\omega_y I)d\omega_y \end{aligned} \tag{11.3.7}$$

Since $\omega_y = \omega_x/I$, (11.3.7) can be expressed as

$$\begin{aligned} y(0) &= \frac{C}{I}\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega_x)d\omega_x \\ &= \frac{C}{I}x(0) \end{aligned} \tag{11.3.8}$$

Therefore, $C = I$ is the desired normalization factor.

Finally, we indicate that the output sequence $y(m)$ can be expressed as a convolution of the sequence $v(n)$ with the unit sample response $h(n)$ of the lowpass filter. Thus

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)v(k) \tag{11.3.9}$$

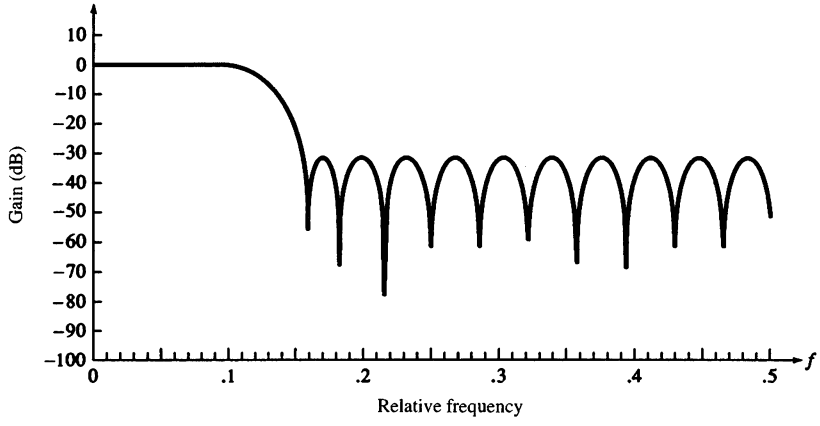


Figure 11.3.2 Magnitude response of linear-phase FIR filter of length $M = 30$ in Example 11.3.1.

Since $v(k) = 0$ except at multiples of I, where $v(kI) = x(k)$, (11.3.9) becomes

$$y(m) = \sum_{k=-\infty}^{\infty} h(m - kI)x(k) \tag{11.3.10}$$

EXAMPLE 11.3.1

Design a interpolator that increases the input sampling rate by a factor of $I = 5$. Use the Remez algorithm to determine the coefficients of the FIR filter that has a 0.1-dB ripple in the passband and is down by at least 30 dB in the stopband.

Solution. A filter length $M = 30$ achieves the design specifications given above. The frequency response of the FIR filter is illustrated in Fig. 11.3.2. Note that the cutoff frequency is $\omega_c = \pi/5$.

11.4 Sampling Rate Conversion by a Rational Factor I/D

Having discussed the special cases of decimation (downsampling by a factor D) and interpolation (upsampling by a factor I), we now consider the general case of sampling rate conversion by a rational factor I/D. Basically, we can achieve this sampling rate conversion by first performing interpolation by the factor I and then decimating the output of the interpolator by the factor D. In other words, a sampling rate conversion by the rational factor I/D is accomplished by cascading an interpolator with a decimator, as illustrated in Fig. 11.4.1.

We emphasize that the importance of performing the interpolation first and the decimation second is to preserve the desired spectral characteristics of $x(n)$. Furthermore, with the cascade configuration illustrated in Fig. 11.4.1, the two filters with impulse response $\{h_u(k)\}$ and $\{h_d(k)\}$ are operated at the same rate, namely IF_x, and hence can be combined into a single lowpass filter with impulse response $h(k)$ as illustrated in Fig. 11.4.2. The frequency response $H(\omega_v)$ of the combined filter

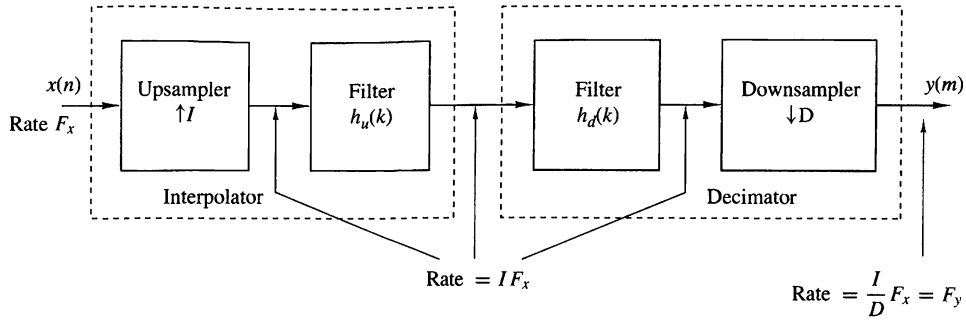


Figure 11.4.1 Method for sampling rate conversion by a factor I/D.

must incorporate the filtering operations for both interpolation and decimation, and hence it should ideally possess the frequency response characteristic

$$H(\omega_v) = \begin{cases} I, & 0 \le |\omega_v| \le \min(\pi/D, \pi/I) \\ 0, & \text{otherwise} \end{cases} \tag{11.4.1}$$

where $\omega_v = 2\pi F/F_v = 2\pi F/IF_x = \omega_x/I$.

In the time domain, the output of the upsampler is the sequence

$$v(l) = \begin{cases} x(l/I), & l = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{11.4.2}$$

and the output of the linear time-invariant filter is

$$\begin{aligned} w(l) &= \sum_{k=-\infty}^{\infty} h(l-k)v(k) \\ &= \sum_{k=-\infty}^{\infty} h(l-kI)x(k) \end{aligned} \tag{11.4.3}$$

Finally, the output of the sampling rate converter is the sequence $\{y(m)\}$, which is

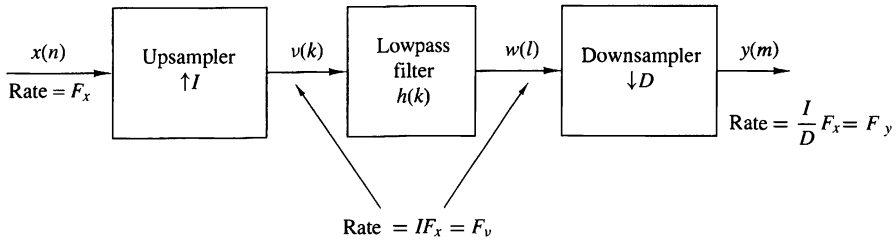


Figure 11.4.2 Method for sampling rate conversion by a factor I/D.

obtained by downsampling the sequence $\{w(l)\}$ by a factor of D. Thus

$$\begin{aligned} y(m) &= w(mD) \\ &= \sum_{k=-\infty}^{\infty} h(mD - kI)x(k) \end{aligned} \tag{11.4.4}$$

It is illuminating to express (11.4.4) in a different form by making a change in variable. Let

$$k = \left\lfloor \frac{mD}{I} \right\rfloor - n \tag{11.4.5}$$

where the notation $\lfloor r \rfloor$ denotes the largest integer contained in r. With this change in variable, (11.4.4) becomes

$$y(m) = \sum_{n=-\infty}^{\infty} h\left(mD - \left\lfloor \frac{mD}{I} \right\rfloor I + nI\right) x\left(\left\lfloor \frac{mD}{I} \right\rfloor - n\right) \tag{11.4.6}$$

We note that

$$\begin{aligned} mD - \left\lfloor \frac{mD}{I} \right\rfloor I &= mD, \qquad \text{modulo } I \\ &= (mD)_I \end{aligned}$$

Consequently, (11.4.6) can be expressed as

$$y(m) = \sum_{n=-\infty}^{\infty} h(nI + (mD)_I)x\left(\left\lfloor \frac{mD}{I} \right\rfloor - n\right) \tag{11.4.7}$$

which is the discrete-time version of (11.1.9).

It is apparent from this form that the output $y(m)$ is obtained by passing the input sequence $x(n)$ through a time-variant filter with impulse response

$$g(n, m) = h(nI + (mD)_I), \qquad -\infty < m, n < \infty \tag{11.4.8}$$

where $h(k)$ is the impulse response of the time-invariant lowpass filter operating at the sampling rate IF_x. We further observe that for any integer k,

$$\begin{aligned} g(n, m + kI) &= h(nI + (mD + kDI)_I) \\ &= h(nI + (mD)_I) \\ &= g(n, m) \end{aligned} \tag{11.4.9}$$

Hence $g(n, m)$ is periodic in the variable m with period I.

The frequency-domain relationships can be obtained by combining the results of the interpolation and decimation processes. Thus the spectrum at the output of the linear filter with impulse response $h(l)$ is

$$\begin{aligned} V(\omega_v) &= H(\omega_v)X(\omega_v I) \\ &= \begin{cases} IX(\omega_v I), & 0 \le |\omega_v| \le \min(\pi/D, \pi/I) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{11.4.10}$$

The spectrum of the output sequence $y(m)$, obtained by decimating the sequence $v(n)$ by a factor of D, is

$$Y(\omega_y) = \frac{1}{D}\sum_{k=0}^{D-1} V\left(\frac{\omega_y - 2\pi k}{D}\right) \tag{11.4.11}$$

where $\omega_y = D\omega_v$. Since the linear filter prevents aliasing as implied by (11.4.10), the spectrum of the output sequence given by (11.4.11) reduces to

$$Y(\omega_y) = \begin{cases} \dfrac{I}{D}X\left(\dfrac{\omega_y}{D}\right), & 0 \le |\omega_y| \le \min\left(\pi, \dfrac{\pi D}{I}\right) \\ 0, & \text{otherwise} \end{cases} \tag{11.4.12}$$

EXAMPLE 11.4.1

Design a sample rate converter that increases the sampling rate by a factor of 2.5. Use the Remez algorithm to determine the coefficients of the FIR filter that has a 0.1-dB ripple in the passband and is down by at least 30 dB in the stopband. Specify the sets of time-varying coefficients $g(n, m)$ used in the realization of the sampling rate converter.

Solution. The FIR filter that meets the specifications of this problem is exactly the same as the filter designed in Example 11.3.1. Its bandwidth is $\pi/5$.

The coefficients of the FIR filter are given by (11.4.8) as

$$\begin{aligned} g(n, m) &= h(nI + (mD)_I) \\ &= h\left(nI + mD - \lfloor \frac{mD}{I} \rfloor I\right) \end{aligned}$$

By substituting $I = 5$ and $D = 2$ we obtain

$$g(n, m) = h\left(5n + 2m - 5\lfloor \frac{2m}{5} \rfloor\right)$$

By evaluating $g(n, m)$ for $n = 0, 1, \ldots, 5$ and $m = 0, 1, \ldots, 4$ we obtain the following coefficients for the time-variant filter:

$$\begin{aligned} g(0, m) &= \{h(0) \quad h(2) \quad h(4) \quad h(1) \quad h(3)\ \} \\ g(1, m) &= \{h(5) \quad h(7) \quad h(9) \quad h(6) \quad h(8)\ \} \\ g(2, m) &= \{h(10) \quad h(12) \quad h(14) \quad h(11) \quad h(13)\} \\ g(3, m) &= \{h(15) \quad h(17) \quad h(19) \quad h(16) \quad h(18)\} \\ g(4, m) &= \{h(20) \quad h(22) \quad h(24) \quad h(21) \quad h(23)\} \\ g(0, m) &= \{h(25) \quad h(27) \quad h(29) \quad h(26) \quad h(28)\} \end{aligned}$$

In summary, sampling rate conversion by the factor I/D can be achieved by first increasing the sampling rate by I, accomplished by inserting $I-1$ zeros between successive values of the input signal $x(n)$, followed by linear filtering of the resulting sequence to eliminate unwanted images of $X(\omega)$ and, finally, by downsampling the filtered signal by the factor D. When $F_y > F_x$, the lowpass filter acts as an anti-imaging postfilter that removes the spectral replicas at multiples of F_x, but not at multiples of IF_x. When $F_y < F_x$, the lowpass filter acts as an anti-aliasing prefilter that removes the down-shifted spectral replicas at multiples of F_y to avoid overlapping. The design of the lowpass filter can be accomplished by the filter design techniques described in Chapter 10.

11.5 Implementation of Sampling Rate Conversion

In this section we consider the efficient implementation of sampling rate conversion systems using polyphase filter structures. Additional computational simplifications can be obtained using the multistage approach described in Section 11.6.

11.5.1 Polyphase Filter Structures

Polyphase structures for FIR filters were developed for the efficient implementation of sampling rate converters; however, they can be used in other applications. The polyphase structure is based on the fact that any system function can be split as

$$\begin{aligned} H(z) = &\cdots + h(0) && + h(M)z^{-M} + \cdots \\ &\cdots + h(1)z^{-1} && + h(M+1)z^{-(M+1)} + \cdots \\ & && \vdots \\ &\cdots + h(M-1)z^{-(M-1)} && + h(2M-1)z^{-(2M-1)} + \cdots \end{aligned}$$

If we next factor out the term $z^{-(i-1)}$ at the ith row, we obtain

$$\begin{aligned} H(z) = &[\cdots + h(0) && + h(M)z^{-M} + \cdots] \\ &+ z^{-1}[\cdots + h(1) && + h(M+1)z^{-M} + \cdots] \\ & && \vdots \\ &+ z^{-(M-1)}[\cdots + h(M-1) && + h(2M-1)z^{-M} + \cdots] \end{aligned}$$

The last equation can be compactly expressed as

$$H(z) = \sum_{i=0}^{M-1} z^{-i} P_i(z^M) \tag{11.5.1}$$

where

$$P_i(z) = \sum_{n=-\infty}^{\infty} h(nM+i)z^{-n} \tag{11.5.2}$$

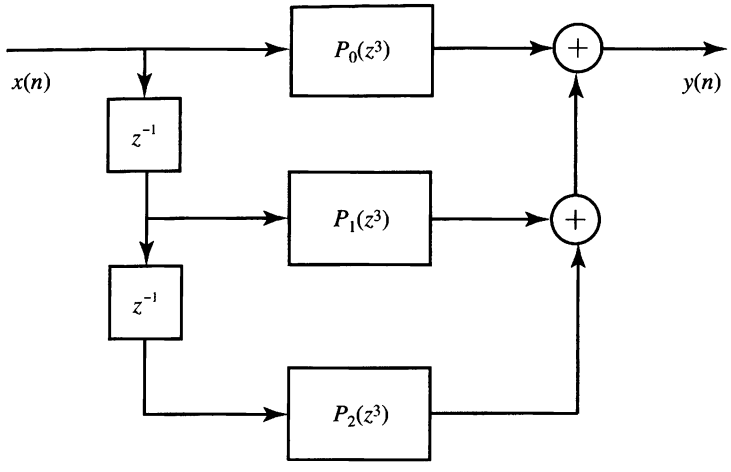


Figure 11.5.1 Block diagram of polyphase filter structure for $M = 3$.

Relation (11.5.1) is called the *M-component polyphase decomposition* and $P_i(z)$ the polyphase components of $H(z)$. Each subsequence

$$p_i(n) = h(nM + i), \quad i = 0, 1, \ldots, M - 1 \tag{11.5.3}$$

is obtained by downsampling a delayed ("phase-shifted") version of the original impulse response.

To develop an M-component polyphase filter structure, we use (11.5.1) for $M = 3$ to express the z-transform of the output sequence as

$$\begin{aligned} Y(z) &= H(z)X(z) \\ &= P_0(z^3)X(z) + z^{-1}P_1(z^3)X(z) + z^{-2}P_2(z^3)X(z) \qquad (11.5.4) \\ &= P_0(z^3)X(z) + z^{-1}\{P_1(z^3)X(z) + z^{-1}[P_2(z^3)X(z)]\} \qquad (11.5.5) \end{aligned}$$

Equation (11.5.4) leads to the polyphase structure of Figure 11.5.1. Similarly, (11.5.5) yields the polyphase structure shown in Figure 11.5.2. This is known as transpose polyphase structure because it is similar to the transpose FIR filter realization. The obtained polyphase structures are valid for any filter, FIR or IIR, and any finite value of M and are sufficient for our needs. Additional structures and details can be found in Vaidyanathan (1993).

11.5.2 Interchange of Filters and Downsamplers/Upsamplers

In general, the order of a sampling rate converter (which is a linear time-varying system) and a linear time-invariant system *cannot* be interchanged. We next derive two identities, known as *noble identities*, that help to swap the position of a filter with a downsampler or an upsampler by properly modifying the filter.

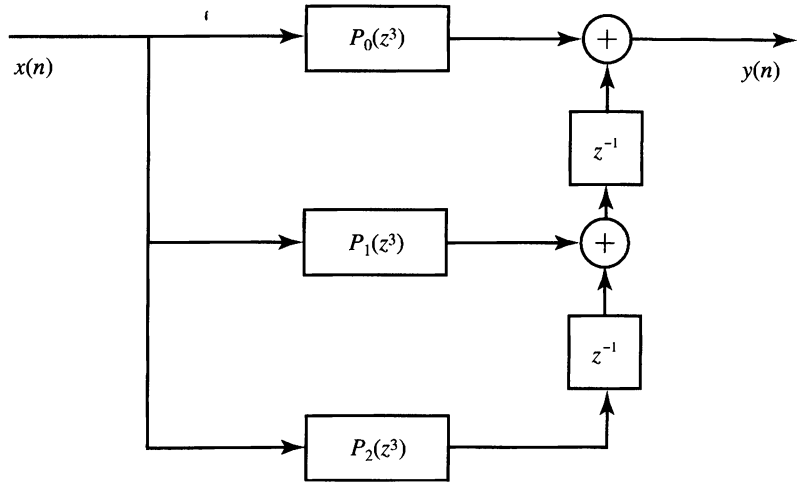


Figure 11.5.2 Illustration of transpose polyphase filter structure for $M = 3$.

To prove the first identity (see Figure 11.5.3), we recall that the input/output description of a downsampler is

$$y(n) = x(nD) \xleftrightarrow{\mathcal{Z}} Y(z) = \frac{1}{D}\sum_{i=0}^{D-1} X(z^{1/D}W_D^i) \tag{11.5.6}$$

where $W_D = e^{-j2\pi/D}$. The output of the system in Figure 11.5.3(a) can be written as

$$Y(z) = \frac{1}{D}\sum_{i=0}^{D-1} V_1(z^{1/D}W_D^i) = \frac{1}{D}\sum_{i=0}^{D-1} H(zW_D^{iD})X(z^{1/D}W_D^i) \tag{11.5.7}$$

because $V_1(z) = H(z^D)X(z)$. Taking into consideration $W_D^{iD} = 1$ and Figure 11.5.3(b), relation (11.5.7) results in

$$Y(z) = \frac{1}{D}H(z)\sum_{i=0}^{D-1} X(z^{1/D}W_D^i) = H(z)V_2(z) \tag{11.5.8}$$

which shows the equivalence of the two structures in Figure 11.5.3.

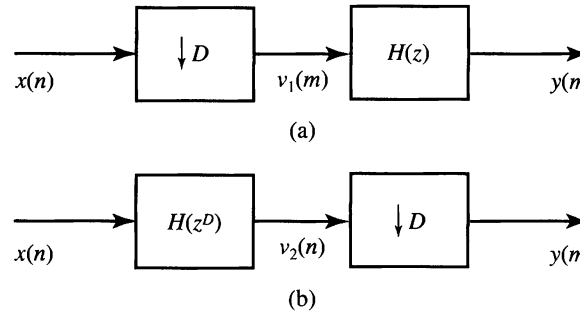


Figure 11.5.3 Two equivalent downsampling systems (first noble identity).

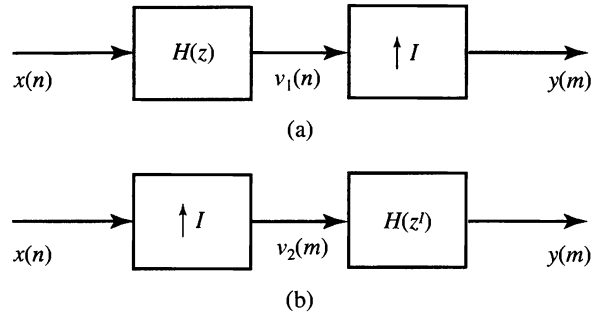


Figure 11.5.4
Two equivalent upsampling systems (second noble identity).

A similar identity can be shown to hold for upsampling. To start, we recall that the input/output description of an upsampler is

$$y(n) = \begin{cases} x\left(\frac{n}{I}\right), & n = 0, \pm I, \pm 2I, \dots \\ 0, & \text{otherwise} \end{cases} \xleftrightarrow{z} Y(z) = X(z^I) \tag{11.5.9}$$

The output of the system in Figure 11.5.4(a) can be written as

$$Y(z) = H(z^I)V_1(z) = H(z^I)X(z^I) \tag{11.5.10}$$

because $V_1(z) = X(z^I)$. The output of the system in Figure 11.5.4(b) is given by

$$Y(z) = V_2(z^I) = H(z^I)X(z^I) \tag{11.5.11}$$

which is equal to (11.5.10). This shows that the two systems in Figure 11.5.4 are identical.

In conclusion, we have shown that it is possible to interchange the operation of linear filtering and downsampling or upsampling if we properly modify the system function of the filter.

11.5.3 Sampling Rate Conversion with Cascaded Integrator Comb Filters

The hardware implementation of the lowpass filter required for sampling rate conversion can be significantly simplified if we choose a comb filter with system function (see Section 5.4.5)

$$H(z) = \sum_{k=0}^{M-1} z^{-k} = \frac{1 - z^{-M}}{1 - z^{-1}} \tag{11.5.12}$$

This system can be implemented by cascading either the "integrator" $1/(1-z^{-1})$ with the comb filter $(1 - z^{-M})$ or vice versa. This leads to the name *cascaded integrator comb* (CIC) filter structure. The CIC structure does not require any multiplications or storage for the filter coefficients.

To obtain an efficient decimation structure, we start with an integrator-comb CIC filter followed by the downsampler and then we apply the first noble identity, as shown in Figure 11.5.5. For the interpolation case, we use an upsampler followed

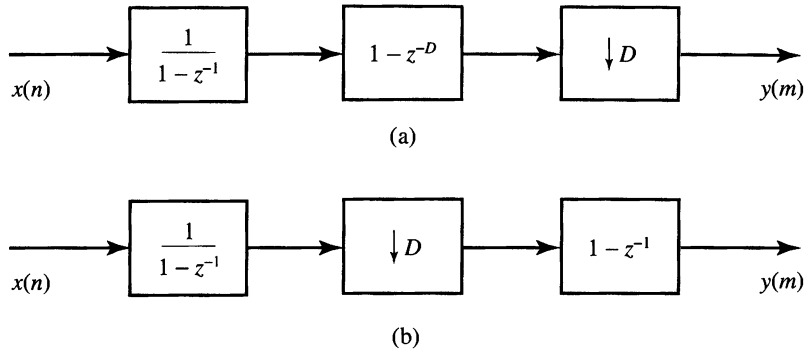


Figure 11.5.5 Using the first noble identity to obtain an efficient CIC filter structure for decimation.

by a comb-integrator CIC filter and then we use the second noble identity, as shown in Figure 11.5.6. To improve the lowpass frequency response required for sampling rate conversion, we can cascade K CIC filters. In this case, we order all integrators on one side of the filter and the comb filters on the other side, and then we apply the noble identities as in the single-stage case. The integrator $1/(1-z^{-1})$ is an unstable system. Therefore, its output may grow without limits, resulting in overflow when the integrator section comes first, as in the decimation structure shown in Figure 11.5.5(b). However, this overflow can be tolerated if the entire filter is implemented using two's-complement fixed-point arithmetic. If $D \neq M$ or $I \neq M$, the comb filter $1-z^{-1}$ in Figures 11.5.5(a) and 11.5.6(b) should be replaced by $1-z^{-M/D}$ or $1-z^{-M/I}$, respectively. A detailed treatment of CIC filters for decimation and interpolation can be found in Hogenauer (1981). Finally, we note that CIC filters are special cases of the frequency sampling structure discussed in Section 10.2.3.

If the CIC filter order is a power of 2, that is, $M = 2^K$, we can decompose the system function (11.5.12) as follows:

$$H(z) = (1+z^{-1})(1+z^{-2})(1+z^{-4})\ldots(1+z^{-2^{K-1}}) \tag{11.5.13}$$

Using this decomposition we can develop decimator structures using nonrecursive

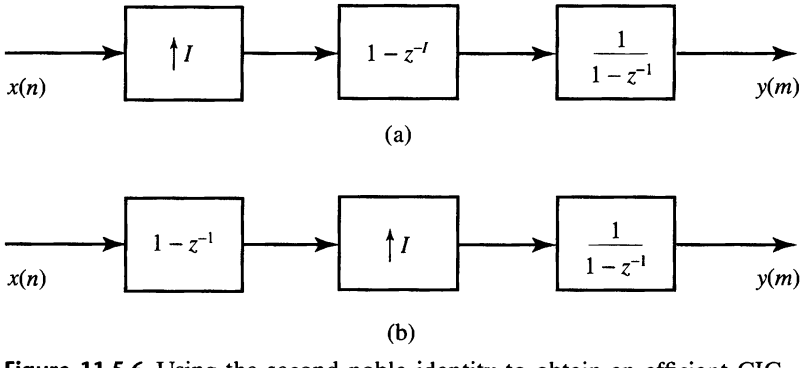


Figure 11.5.6 Using the second noble identity to obtain an efficient CIC filter structure for interpolation.

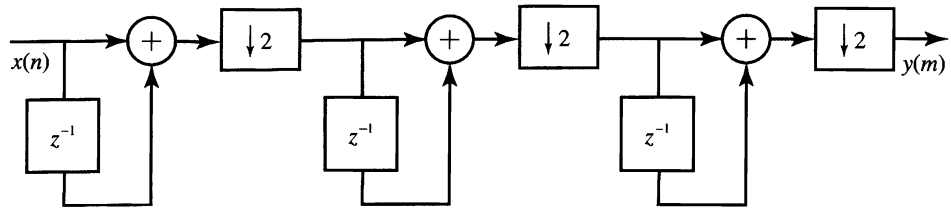


Figure 11.5.7 Efficient filter structure for decimation by $D = 8$ using comb filters.

CIC filters. Figure 11.5.7 shows an example for a decimator with $D = M = 8$. Cascading of N CIC filters can be obtained by providing M first-order sections $(1 - z^{-1})$ between each decimation stage. The constraint $M = 2^K$ can be relaxed by factoring M into a product of prime numbers, as shown in Jang and Yang (2001).

11.5.4 Polyphase Structures for Decimation and Interpolation Filters

To develop a polyphase structure for decimation, we start with the straightforward implementation of the decimation process shown in Figure 11.5.8. The decimated sequence is obtained by passing the input sequence $x(n)$ through a linear filter and then downsampling the filter output by a factor D. In this configuration, the filter is operating at the high sampling rate F_x, while only one out of every D output samples is actually needed. A logical solution would be to find a structure where only the needed samples are computed. We will develop such an efficient implementation by exploitating the polyphase structure in Figure 11.5.1. Since downsampling commutes with addition, combining the structures in Figures 11.5.8 and 11.5.1 yields the structure in Figure 11.5.9(a). If we next apply the identity in Figure 11.5.3, we obtain the desired implementation structure shown in Figure 11.5.9(b). In this filtering structure, only the needed samples are computed and all multiplications and additions are performed at the lower sampling rate F_x/D. Thus, we have achieved the desired efficiency. Additional reduction in computation can be achieved by using an FIR filter with linear phase and exploiting the symmetry of its impulse response.

In practice it is more convenient to implement the polyphase decimator using a *commutator model* as shown in Figure 11.5.10. The commutator rotates *counterclockwise* starting at time $n = 0$ and distributes a block of D input samples to the polyphase filters starting at filter $i = D - 1$ and continuing in reverse order until $i = 0$. For every block of D input samples, the polyphase filters receive a new input and their outputs are computed and summed to produce one sample of the output signal $y(m)$. The operation of this realization can be also understood by a careful inspection of Figure 11.1.3.

Next, let us consider the efficient implementation of an interpolator, which is realized by first inserting $I - 1$ zeros between successive samples of $x(n)$ and then

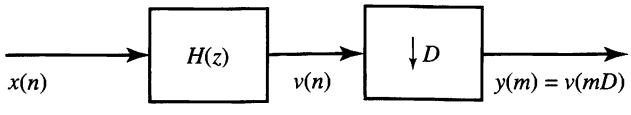


Figure 11.5.8 Decimation system.

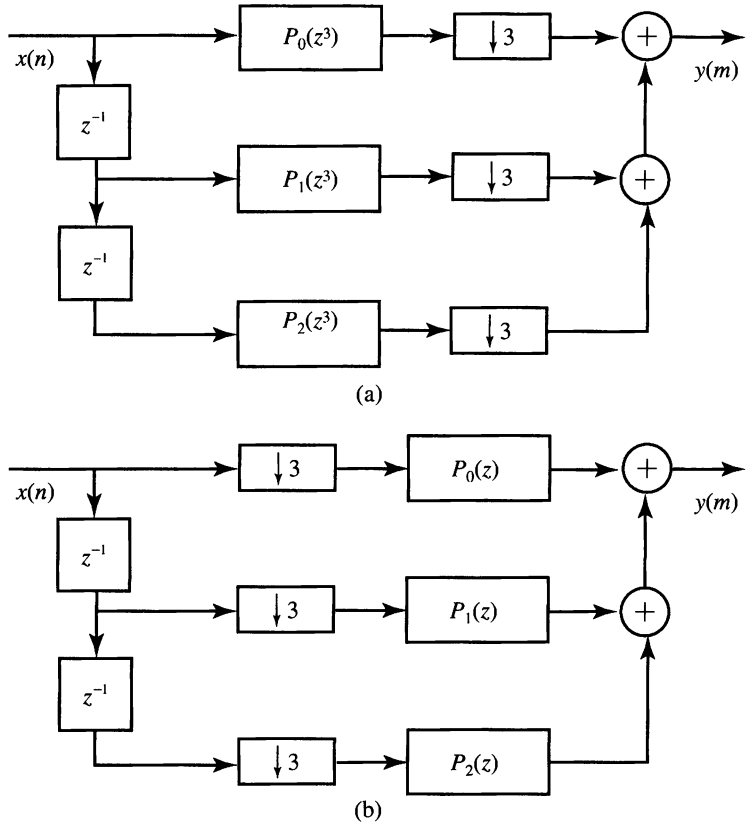


Figure 11.5.9 Implementation of a decimation system using a polyphase structure before (a) and after (b) the use of the first noble identity.

filtering the resulting sequence (see Figure 11.5.11). The major problem with this structure is that the filter computations are performed at the high sampling rate IF_x. The desired simplification is achieved by first replacing the filter in Figure 11.5.11 with the transpose polyphase structure in Figure 11.5.2, as illustrated in Figure 11.5.12(a).

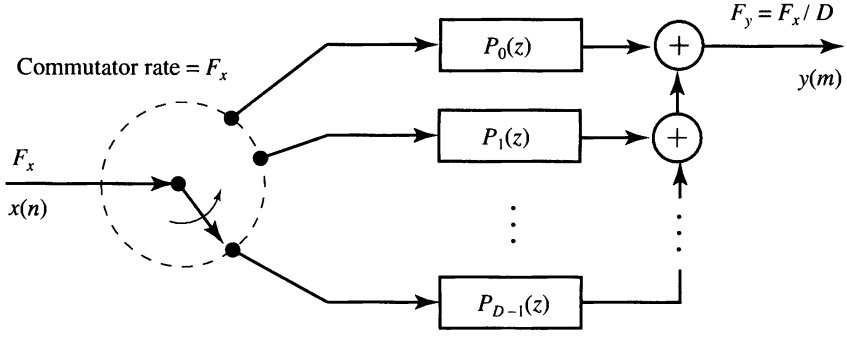


Figure 11.5.10 Decimation using a polyphase filter and a commutator.

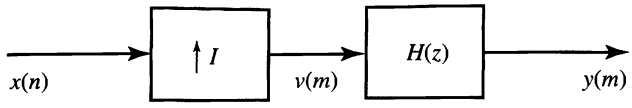
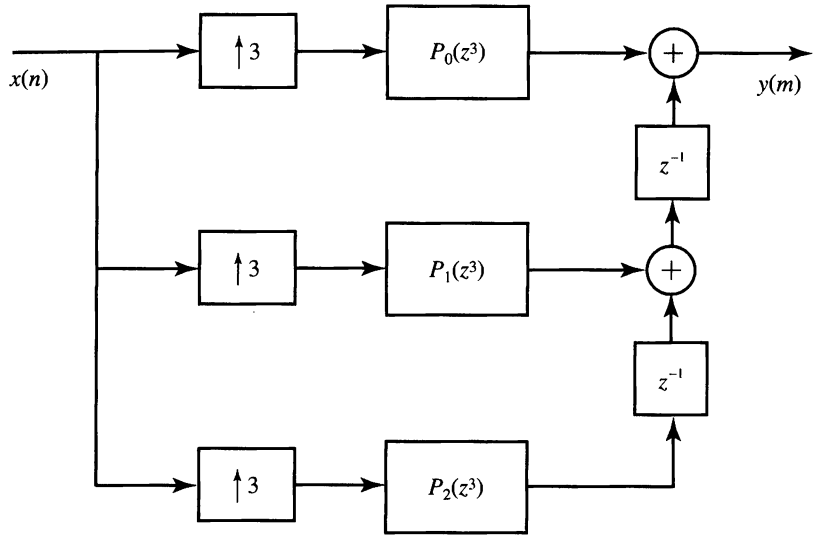
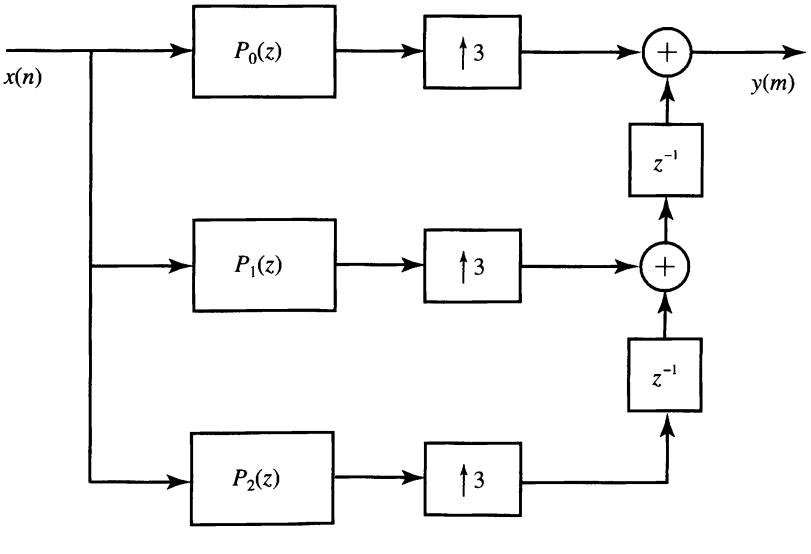


Figure 11.5.11 Interpolation system.



(a)



(b)

Figure 11.5.12 Implementation of an interpolation system using a polyphase structure before (a) and after (b) the use of the second noble identity.

Then, we use the second noble identity (see Figure 11.5.4) to obtain the structure in Figure 11.5.12(b). Thus, all the filtering multiplications are performed at the low rate F_x. It is interesting to note that the structure for an interpolator can be obtained by transposing the structure of a decimator, and vice versa (Crochiere and Rabiner, 1981).

For every input sample, the polyphase filters produce I output samples $y_0(n)$, $y_1(n), \ldots, y_{I-1}(n)$. Because the output $y_i(n)$ of the ith filter is followed by $(I-1)$ zeros and it is delayed by i th samples, the polyphase filters contribute nonzero samples at different time slots. In practice, we can implement the part of the structure including the 1-to-I expanders, delays, and adders using the commutator model shown in Figure 11.5.13. The commutator rotates *counterclockwise* starting at time $n = 0$ at branch $i = 0$. For each input sample $x(n)$ the commutator reads the output of every polyphase filter to obtain I samples of the output (interpolated) signal $y(m)$. The operation of this realization can be also understood by a careful inspection of Figure 11.1.4. Each polyphase filter in Figure 11.5.13 operates on the same input data using its unique set of coefficients. Therefore, we can obtain the same results using a single filter by sequentially loading a different set of coefficients.

11.5.5 Structures for Rational Sampling Rate Conversion

A sampling rate converter with a ratio I/D can be efficiently implemented using a polyphase interpolator followed by a downsampler. However, since the downsampler keeps only every Dth polyphase subfilter output, it is not necessary to compute all I interpolated values between successive input samples. To determine which polyphase subfilter outputs to compute, we consider an example with $I = 5$ and $D = 3$. The interpolator polyphase structure has $I = 5$ subfilters which provide interpolated samples at an effective sampling period $T = T_x/I$. The downsampler picks every Dth of these samples, resulting in a discrete-time signal with sampling

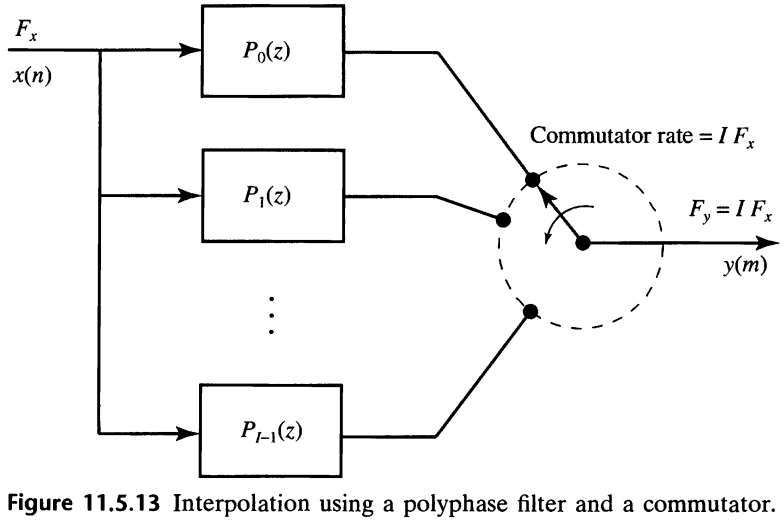
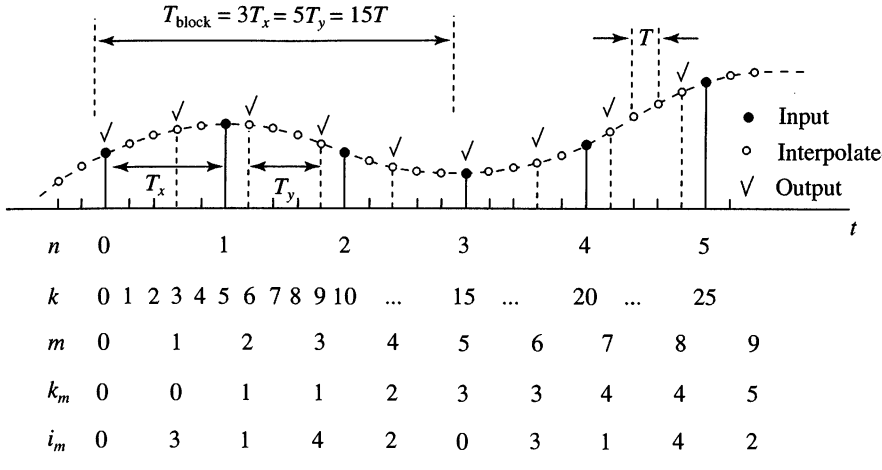


Figure 11.5.13 Interpolation using a polyphase filter and a commutator.



n	0		1		2		3		4		5
k	0	1	2	3	4	5	6	7	8	9	10 ... 15 ... 20 ... 25
m	0		1		2		3		4		5	6	7		8		9
k_m	0		0		1		1		2		3	3	4		4		5
i_m	0		3		1		4		2		0	3	1		4		2

Figure 11.5.14 Illustration of index computation for polyphase implementation of sampling rate conversion for a rational ratio $I/D = 5/3$.

period $T_y = DT = DT_x/I$. It is convenient to think in terms of blocks of duration

$$T_{\text{block}} = IT_y = DT_x = IDT \tag{11.5.14}$$

which contain L output samples or I input samples. The relative time positions of the various sequences and a block of data are illustrated in Figure 11.5.14. The input sequence $x(nT_x)$ is interpolated to produce a sequence $v(kT)$, which is then decimated to yield $y(mT_y)$. If we use an FIR filter with $M = KI$ coefficients, the polyphase subfilters are given by $p_i(n) = h(nI + i)$, $i = 0, 1, \ldots, I - 1$, where $n = 0, 1, \ldots, K - 1$. To compute the output sample $y(m)$, we use the polyphase subfilter with index i_m which requires the input samples $x(k_m), x(k_m - 1), \ldots, x(k_m - K + 1)$. From relation (11.1.9) and Figure 11.5.14, we can easily deduce that

$$k_m = \left\lfloor \frac{mD}{I} \right\rfloor \quad \text{and} \quad i_m = (Dm)_I \tag{11.5.15}$$

For $D = 3$ and $I = 5$ the first data block includes $D = 3$ input samples and $I = 5$ output samples. To compute the samples $\{y(0), y(1), y(2), y(3), y(4)\}$, we use the polyphase subfilter specified by the index $i_m = \{0, 3, 1, 4, 2\}$, respectively. The samples in the filter memory are updated only when k_m changes value. This discussion provides the basic ideas for the efficient software implementation of rational sampling rate conversion using FIR filters.

11.6 Multistage Implementation of Sampling Rate Conversion

In practical applications of sampling-rate conversion we often encounter decimation factors and interpolation factors that are much larger than unity. For example, suppose that we are given the task of altering the sampling rate by the factor

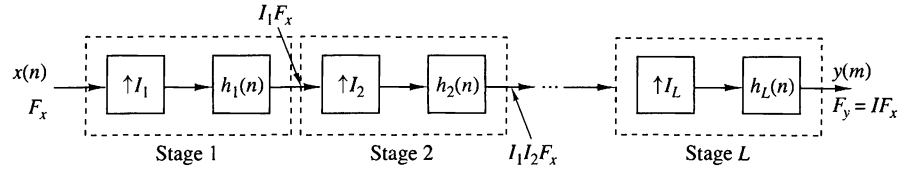


Figure 11.6.1 Multistage implementation of interpolation by a factor I.

$I/D = 130/63$. Although, in theory, this rate alteration can be achieved exactly, the implementation would require a bank of 130 polyphase filters and may be computationally inefficient. In this section we consider methods for performing sampling rate conversion for either $D \gg 1$ and/or $I \gg 1$ in multiple stages.

First, let us consider interpolation by a factor $I \gg 1$ and let us assume that I can be factored into a product of positive integers as

$$I = \prod_{i=1}^{L} I_i \tag{11.6.1}$$

Then, interpolation by a factor I can be accomplished by cascading L stages of interpolation and filtering, as shown in Fig. 11.6.1. Note that the filter in each of the interpolators eliminates the images introduced by the upsampling process in the corresponding interpolator.

In a similar manner, decimation by a factor D, where D may be factored into a product of positive integers as

$$D = \prod_{i=1}^{J} D_i \tag{11.6.2}$$

can be implemented as a cascade of J stages of filtering and decimation as illustrated in Fig. 11.6.2. Thus the sampling rate at the output of the ith stage is

$$F_i = \frac{F_{i-1}}{D_i}, \qquad i = 1, 2, \ldots, J \tag{11.6.3}$$

where the input rate for the sequence $\{x(n)\}$ is $F_0 = F_x$.

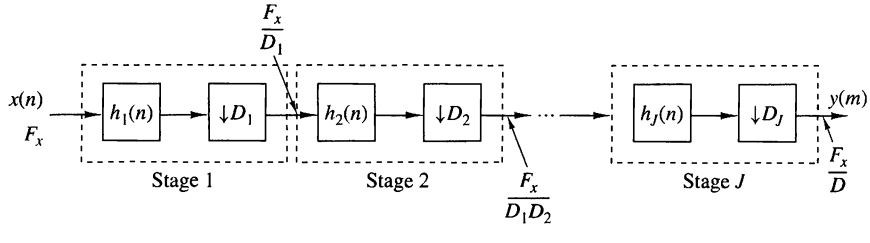


Figure 11.6.2 Multistage implementation of decimation by a factor D.

To ensure that no aliasing occurs in the overall decimation process, we can design each filter stage to avoid aliasing within the frequency band of interest. To elaborate, let us define the desired passband and the transition band in the overall decimator as

$$\begin{aligned} \text{Passband: } & 0 \le F \le F_{pc} \\ \text{Transition band: } & F_{pc} \le F \le F_{sc} \end{aligned} \tag{11.6.4}$$

where $F_{sc} \le F_x/2D$. Then, aliasing in the band $0 \le F \le F_{sc}$ is avoided by selecting the frequency bands of each filter stage as follows:

$$\begin{aligned} \text{Passband: } & 0 \le F \le F_{pc} \\ \text{Transition band: } & F_{pc} \le F \le F_i - F_{sc} \\ \text{Stopband: } & F_i - F_{sc} \le F \le \frac{F_{i-1}}{2} \end{aligned} \tag{11.6.5}$$

For example, in the first filter stage we have $F_1 = F_x/D_1$, and the filter is designed to have the following frequency bands:

$$\begin{aligned} \text{Passband: } & 0 \le F \le F_{pc} \\ \text{Transition band: } & F_{pc} \le F \le F_1 - F_{sc} \\ \text{Stopband: } & F_1 - F_{sc} \le F \le \frac{F_0}{2} \end{aligned} \tag{11.6.6}$$

After decimation by D_1, there is aliasing from the signal components that fall in the filter transition band, but the aliasing occurs at frequencies above F_{sc}. Thus there is no aliasing in the frequency band $0 \le F \le F_{sc}$. By designing the filters in the subsequent stages to satisfy the specifications given in (11.6.5), we ensure that no aliasing occurs in the primary frequency band $0 \le F \le F_{sc}$.

EXAMPLE 11.6.1

Consider an audio-band signal with a nominal bandwidth of 4 kHz that has been sampled at a rate of 8 kHz. Suppose that we wish to isolate the frequency components below 80 Hz with a filter that has a passband $0 \le F \le 75$ and a transition band $75 \le F \le 80$. Hence $F_{pc} = 75$ Hz and $F_{sc} = 80$. The signal in the band $0 \le F \le 80$ may be decimated by the factor $D = F_x/2F_{sc} = 50$. We also specify that the filter have a passband ripple $\delta_1 = 10^{-2}$ and a stopband ripple of $\delta_2 = 10^{-4}$.

The length of the linear phase FIR filter required to satisfy these specifications can be estimated from one of the well-known formulas given in the Section 10.2.7. Recall that a particularly simple formula for approximating the length M, attributed to Kaiser, is

$$\hat{M} = \frac{-10 \log_{10} \delta_1 \delta_2 - 13}{14.6 \Delta f} + 1 \tag{11.6.7}$$

where Δf is the normalized (by the sampling rate) width of the transition region [i.e., $\Delta f = (F_{sc} - F_{pc})/F_x$]. A more accurate formula proposed by Herrmann et al. (1973) is

$$\hat{M} = \frac{D_\infty(\delta_1, \delta_2) - f(\delta_1, \delta_2)(\Delta f)^2}{\Delta f} + 1 \tag{11.6.8}$$

where $D_\infty(\delta_1, \delta_2)$ and $f(\delta_1, \delta_2)$ are defined as

$$D_\infty(\delta_1, \delta_2) = [0.005309(\log_{10}\delta_1)^2 + 0.07114(\log_{10}\delta_1) - 0.4761]\log_{10}\delta_2 - [0.00266(\log_{10}\delta_1)^2 + 0.5941\log_{10}\delta_1 + 0.4278] \tag{11.6.9}$$

$$f(\delta_1, \delta_2) = 11.012 + 0.51244[\log_{10}\delta_1 - \log_{10}\delta_2] \tag{11.6.10}$$

Now a single FIR filter followed by a decimator would require (using the Kaiser formula) a filter of (approximate) length

$$\hat{M} = \frac{-10\log_{10}10^{-6} - 13}{14.6(5/8000)} + 1 \approx 5152$$

As an alternative, let us consider a two-stage decimation process with $D_1 = 25$ and $D_2 = 2$. In the first stage we have the specifications $F_1 = 320$ Hz and

$$\text{Passband: } 0 \le F \le 75$$

$$\text{Transition band: } 75 < F \le 240$$

$$\Delta f = \frac{165}{8000}$$

$$\delta_{11} = \frac{\delta_1}{2}, \qquad \delta_{21} = \delta_2$$

Note that we have reduced the passband ripple δ_1 by a factor of 2, so that the total passband ripple in the cascade of the two filters does not exceed δ_1. On the other hand, the stopband ripple is maintained at δ_2 in both stages. Now the Kaiser formula yields an estimate of M_1 as

$$\hat{M}_1 = \frac{-10\log_{10}\delta_{11}\delta_{21} - 13}{14.6\Delta f} + 1 \approx 167$$

For the second stage, we have $F_2 = F_1/2 = 160$ and the specifications

$$\text{Passband: } 0 \le F \le 75$$

$$\text{Transition band: } 75 < F \le 80$$

$$\Delta f = \frac{5}{320}$$

$$\delta_{12} = \frac{\delta_1}{2}, \qquad \delta_{22} = \delta_2$$

Hence the estimate of the length M_2 of the second filter is

$$\hat{M}_2 \approx 220$$

Therefore, the total length of the two FIR filters is approximately $\hat{M}_1 + \hat{M}_2 = 387$. This represents a reduction in the filter length by a factor of more than 13.

The reader is encouraged to repeat the computation above with $D_1 = 10$ and $D_2 = 5$.

It is apparent from the computations in Example 11.6.1 that the reduction in the filter length results from increasing the factor Δf, which appears in the denominator in (11.6.7) and (11.6.8). By decimating in multiple stages, we are able to increase the width of the transition region through a reduction in the sampling rate.

In the case of a multistage interpolator, the sampling rate at the output of the ith stage is

$$F_{i-1} = I_i F_i, \qquad i = J, J-1, \ldots, 1$$

and the output rate is $F_0 = I F_J$ when the input sampling rate is F_J. The corresponding frequency band specifications are

$$\text{Passband: } 0 \le F \le F_p$$

$$\text{Transition band: } F_p < F \le F_i - F_{\text{sc}}$$

The following example illustrates the advantages of multistage interpolation.

EXAMPLE 11.6.2

Let us reverse the filtering problem described in Example 11.6.1 by beginning with a signal having a passband $0 \le F \le 75$ and a transition band of $75 \le F \le 80$. We wish to interpolate by a factor of 50. By selecting $I_1 = 2$ and $I_2 = 25$, we have basically a transposed form of the decimation problem considered in Example 11.6.1. Thus we can simply transpose the two-stage decimator to achieve the two-stage interpolator with $I_1 = 2$, $I_2 = 25$, $\hat{M}_1 \approx 220$, and $\hat{M}_2 \approx 167$.

11.7 Sampling Rate Conversion of Bandpass Signals

In this section we consider the decimation and interpolation of bandpass signals. We begin by noting that any bandpass signal can be converted into an equivalent lowpass signal (see Section 6.5.2) whose sampling rate can be changed using the already developed techniques. However, a simpler and more widely used approach concerns bandpass discrete-time signals with *integer-band positioning*. The concept is similar to the one discussed for continuous-time bandpass signals in Section 6.4.

To be specific, suppose that we wish to decimate by a factor D an integer-positioned bandpass signal with spectrum confined to the bands

$$(k-1)\frac{\pi}{D} < |\omega| < k\frac{\pi}{D} \tag{11.7.1}$$

where k is a positive integer. A bandpass filter defined by

$$H_{\text{BP}}(\omega) = \begin{cases} 1, & (k-1)\frac{\pi}{D} < |\omega| < k\frac{\pi}{D} \\ 0, & \text{otherwise} \end{cases} \tag{11.7.2}$$

would normally be used to eliminate signal frequency components outside the desired frequency range. Then direct decimation of the filtered signal $v(n)$ by the factor D

results in a periodic replication of the bandpass spectrum $V(\omega)$ every $2\pi/D$ radians according to (11.2.14). The spectrum of the decimated signal $y(m)$ is obtained by scaling the frequency axis by $\omega_y = D\omega_x$. This process is illustrated in Figure 11.7.1 for bandpass signal with odd band positioning ($k = 3$) and in Figure 11.7.2 for signals with even band positioning ($k = 4$). In the case where k is odd, there is an inversion of the spectrum of the signal as in the continuous-time case (see Figure 6.4.1(b)). The inversion can be undone by the simple process $y'(m) = (-1)^m y(m)$. Note that violation of the bandwidth constraint given by (11.7.1) results in signal aliasing.

The process of bandpass interpolation by an integer factor I is the inverse of that of bandpass decimation and can be accomplished in a similar manner. The process of upsampling by inserting zeros between the samples of $x(n)$ produces I images in the band $0 \leq \omega \leq \pi$. The desired image can be selected by bandpass filtering. This can be seen by "reversing" the process shown in Figure 11.7.1. Note that the process of interpolation also provides us with the opportunity to achieve frequency translation of the spectrum.

Finally, sampling rate conversion for a bandpass signal by a rational factor I/D can be accomplished by cascading a decimator with an interpolator in a manner that depends on the choice of the parameters D and I. A bandpass filter preceding the sampling converter is usually required to isolate the signal frequency band of

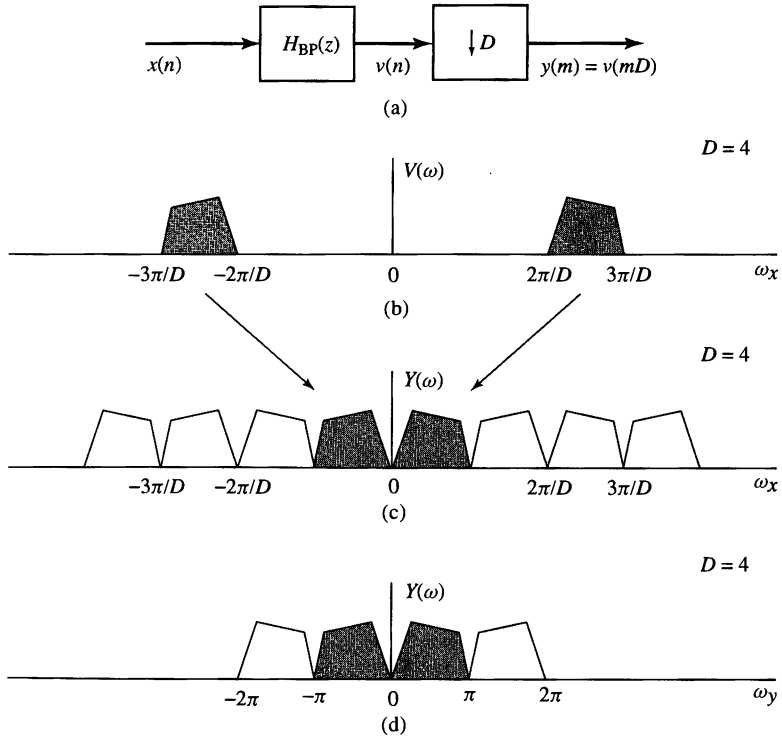


Figure 11.7.1 Spectral interpretation of bandpass decimation for integer-band positioning (odd integer positioning).

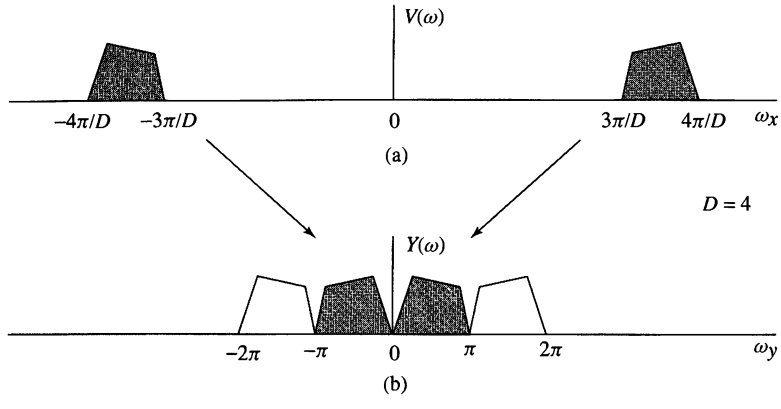


Figure 11.7.2 Spectral interpretation of bandpass decimation for integer-band positioning (even integer positioning).

interest. Note that this approach provides us with a modulation-free method for achieving frequency translation of a signal by selecting $D = I$.

11.8 Sampling Rate Conversion by an Arbitrary Factor

Efficient implementation of sampling rate conversion by a polyphase structure requires that the rates F_x and F_y are fixed and related by a rational factor I/D. In some applications, it is either inefficient or sometimes impossible to use such an exact rate conversion scheme.

For example, suppose we need to perform rate conversion by the rational number I/D, where I is a large integer (e.g., $I/D = 1023/511$). Although we can achieve exact rate conversion by this number, we would need a polyphase filter with 1023 subfilters. Such an implementation is obviously inefficient in memory usage because we need to store a large number of filter coefficients.

In some applications, the exact conversion rate is not known when we design the rate converter, or the rate is continuously changing during the conversion process. For example, we may encounter the situation where the input and output samples are controlled by two independent clocks. Even though it is still possible to define a nominal conversion rate that is a rational number, the actual rate would be slightly different, depending on the frequency difference between the two clocks. Obviously, it is not possible to design an exact converter in this case.

In principle, we can convert from any rate F_x to any rate F_y (fixed or variable) using formula (11.1.9), which we repeat here for convenience:

$$y(mT_y) = \sum_{k=K_1}^{K_2} g(kT_x + \Delta_m T_x)x((k_m - k)T_x) \tag{11.8.1}$$

This requires the computation of a new impulse response $p_m(k) = g(kT_x + \Delta_m T_x)$ for each output sample. However, if Δ_m is measured with finite accuracy, there is only a finite set of impulse responses, which may be precomputed and loaded from memory

as needed. We next discuss two practical approaches for sampling rate conversion by an arbitrary factor.

11.8.1 Arbitrary Resampling with Polyphase Interpolators

If we use a polyphase interpolator with I subfilters we can generate samples with spacing T_x/I. Therefore, the number I of stages determines the granularity of the interpolating process. If T_x/I is sufficiently small so that successive values of the signal do not change significantly or the change is less than the quantization step, we can determine the value at any location $t = nT_x + \Delta T_x$, $0 \leq \Delta \leq 1$, using the value of the nearest neighbor (zero-order hold interpolation).

Additional improvement can be obtained using two-point linear interpolation

$$y(nT_x + \Delta T_x) = (1 - \Delta)x(n) + \Delta x(n + 1) \tag{11.8.2}$$

The performance of these interpolation techniques has been discussed in Section 6.5 by analyzing their frequency-domain characteristics. Additional practical details can be found in Ramstad (1984).

11.8.2 Arbitrary Resampling with Farrow Filter Structures

In practice, we typically implement polyphase sampling rate converters by a rational factor using causal FIR lowpass filters. If we use an FIR filter with $M = KI$ coefficients, the coefficients of the polyphase filters are obtained by the mapping

$$p_i(n) = h(nI + i), \quad i = 0, 1, \ldots, I - 1 \tag{11.8.3}$$

This mapping can be easily visualized as mapping the one-dimensional sequence $h(n)$ into a two-dimensional array with I rows and K columns by filling successive columns in natural order as follows:

$$\begin{array}{llllll}
p_0(k) & \mapsto h(0) & h(I) & \ldots & h((K-1)I) & \\
p_1(k) & \mapsto h(1) & h(I+1) & \ldots & h((K-1)I+1) & \\
 & \vdots & & & & \\
p_i(k) & \mapsto h(i) & h(I+i) & \ldots & h((K-1)I+i) & \\
p_{i+1}(k) & \mapsto h(i+1) & h(I+i+1) & \ldots & & \\
 & \vdots & & & & \\
p_{I-1}(k) & \mapsto h(I-1) & h(2I-1) & \ldots & h(KI-1) &
\end{array} \tag{11.8.4}$$

The polyphase filters $p_i(n)$ are used to compute samples at I equidistant locations $t = nT_x + i(T_x/I)$, $i = 0, 1, \ldots, I-1$ covering each input sampling interval. Suppose now that we wish to compute a sample at $t = nT_x + \Delta T_x$, where $\Delta \neq i/I$ and $0 \leq \Delta \leq 1$. This requires a nonexisting polyphase subfilter, denoted as $p_\Delta(k)$, which would "fall" between two existing subfilters, say, $p_i(k)$ and $p_{i+1}(k)$. This set of coefficients would create a row between the rows with indexes i and $i + 1$. We note that each column of (11.8.4) consists of a segment of I consecutive samples of

the impulse response $h(n)$ and covers one sampling interval T_x. Suppose next that we can approximate the coefficient set in each column by an L-degree polynomial

$$B_k(\Delta) = \sum_{\ell=0}^{L} b_\ell^{(k)} \Delta^\ell, \quad k = 0, 1, \ldots, K-1 \tag{11.8.5}$$

We note that evaluating (11.8.5) at $\Delta = i/I$ will provide the coefficients of $p_i(k)$ polyphase subfilter. The type of the polynomial (Lagrange, Chebyschev, etc.) and the order L can be chosen to avoid any performance degradation compared to the original filter $h(n)$. The sample at location $t = nT_x + \Delta T_x$ is determined by

$$y((n+\Delta)T_x) = \sum_{k=0}^{K-1} B_k(\Delta) x((n-k)T_x), \quad 0 \le \Delta \le 1 \tag{11.8.6}$$

where the required filter coefficients are computed using (11.8.5). If we substitute the polynomials (11.8.5) into the filtering formula (11.8.6) and we change the order of summations, we obtain

$$\begin{aligned} y((n+\Delta)T_x) &= \sum_{k=0}^{K-1} \sum_{\ell=0}^{L} b_\ell^{(k)} \Delta^\ell x((n-k)T_x) \\ &= \sum_{\ell=0}^{L} \Delta^\ell \sum_{k=0}^{K-1} b_\ell^{(k)} x((n-k)T_x) \end{aligned}$$

The last equation can be written as

$$y((n+\Delta)T_x) = \sum_{\ell=0}^{L} v(\ell) \Delta^\ell \tag{11.8.7}$$

where

$$v(\ell) = \sum_{k=0}^{K-1} b_\ell^{(k)} x((n-k)T_x), \quad \ell = 0, 1, \ldots, L \tag{11.8.8}$$

Equation (11.8.7) can be interpreted as a Taylor series representation of the output sequence, where the terms $v(\ell)$ are successive local derivatives determined from the input sequence. Relation (11.8.8) can be implemented using FIR filtering structures with system functions

$$H_\ell(z) = \sum_{k=0}^{K-1} b_\ell^{(k)} z^{-k} \tag{11.8.9}$$

The most efficient computation of polynomial (11.8.7) can be done using the nested Horner's rule, which is illustrated next for $L = 4$:

$$\begin{aligned} y(\Delta) &= c_0 + c_1\Delta + c_2\Delta^2 + c_3\Delta^3 + c_4\Delta^4 \\ &= c_0 + \Delta(c_1 + \Delta(c_2 + \Delta(c_3 + \Delta c_4))) \end{aligned} \tag{11.8.10}$$

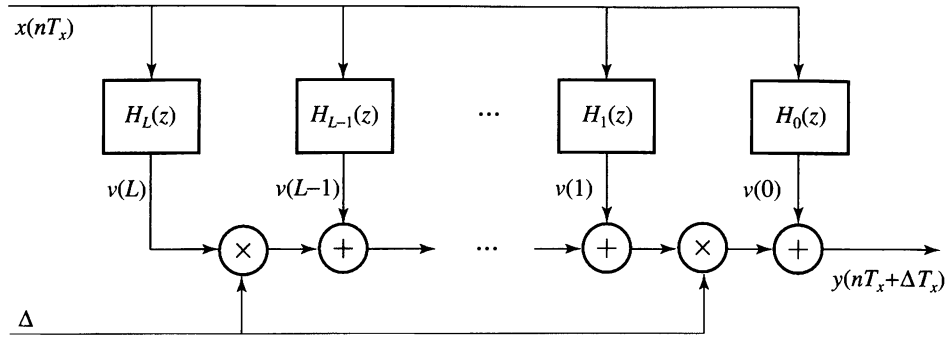


Figure 11.8.1 Block diagram of the Farrow structure for sampling rate change by an arbitrary factor.

This approach leads to the block diagram realization shown in Figure 11.8.1, which is known as Farrow structure (Farrow 1988). Basically, the Farrow structure performs interpolation between signal values by interpolating between filter coefficients. More details can be found in Gardner (1993), Erup et al. (1993), Ramstad (1984), Harris (1997), and Laakso et al. (1996).

11.9 Applications of Multirate Signal Processing

There are numerous practical applications of multirate signal processing. In this section we describe a few of these applications.

11.9.1 Design of Phase Shifters

Suppose that we wish to design a network that delays the signal $x(n)$ by a fraction of a sample. Let us assume that the delay is a rational fraction of a sampling interval T_x [i.e., $d = (k/I)T_x$, where k and I are relatively prime positive integers]. In the frequency domain, the delay corresponds to a linear phase shift of the form

$$\Theta(\omega) = -\frac{k\omega}{I} \tag{11.9.1}$$

The design of an all-pass linear-phase filter is relatively difficult. However, we can use the methods of sample-rate conversion to achieve a delay of $(k/I)T_x$, exactly, without introducing any significant distortion in the signal. To be specific, let us consider the system shown in Fig. 11.9.1. The sampling rate is increased by a factor I using a standard interpolator. The lowpass filter eliminates the images in the spectrum of the interpolated signal, and its output is delayed by k samples at the sampling rate IF_x. The delayed signal is decimated by a factor $D = I$. Thus we have achieved the desired delay of $(k/I)T_x$.

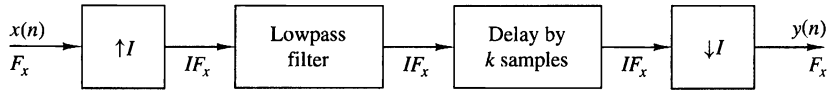


Figure 11.9.1 Method for generating a delay in a discrete-time signal.

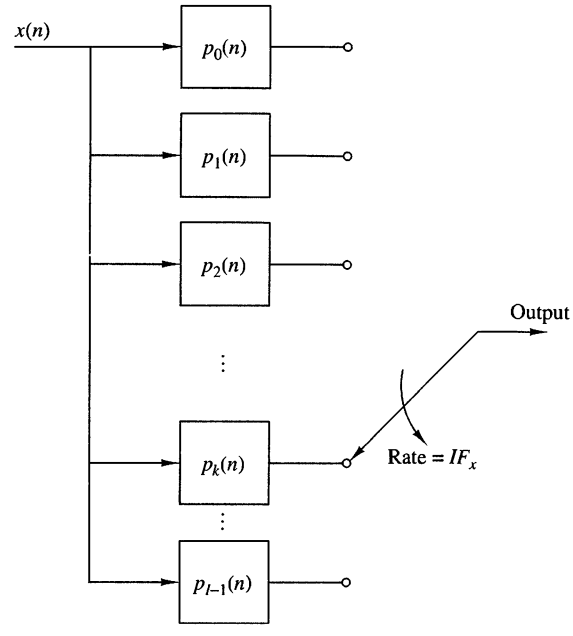


Figure 11.9.2
Polyphase filter structure for implementing the system shown in Fig. 11.9.1.

An efficient implementation of the interpolator is the polyphase filter illustrated in Fig. 11.9.2. The delay of k samples is achieved by placing the initial position of the commutator at the output of the kth subfilter. Since decimation by $D = I$ means that we take one out of every I samples from the polyphase filter, the commutator position can be fixed to the output of the kth subfilter. Thus a delay in k/I can be achieved by using only the kth subfilter of the polyphase filter. We note that the polyphase filter introduces an additional delay of $(M-1)/2$ samples, where M is the length of its impulse response.

Finally, we mention that if the desired delay is a nonrational factor of the sample interval T_x, the methods described in Section 11.8 can be used to obtain the delay.

11.9.2 Interfacing of Digital Systems with Different Sampling Rates

In practice we frequently encounter the problem of interfacing two digital systems that are controlled by independently operating clocks. An analog solution to this problem is to convert the signal from the first system to analog form and then resample it at the input to the second system using the clock in this system. However, a simpler approach is one where the interfacing is done by a digital method using the basic sample-rate conversion methods described in this chapter.

To be specific, let us consider interfacing the two systems with independent clocks as shown in Fig. 11.9.3. The output of system A at rate F_x is fed to an interpolator which increases the sampling rate by I. The output of the interpolator is fed at the rate IF_x to a digital sample-and-hold which serves as the interface to system B at the high sampling rate IF_x. Signals from the digital sample-and-hold are read out into system B at the clock rate DF_y of system B. Thus the output rate from the sample-and-hold is not synchronized with the input rate.

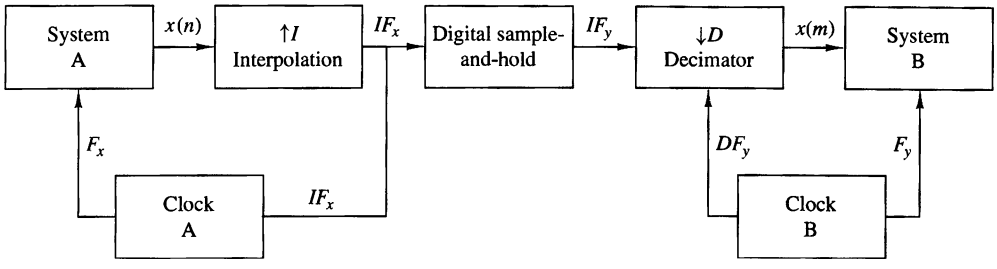


Figure 11.9.3 Interfacing of two digital systems with different sampling rates.

In the special case where $D = I$ and the two clock rates are comparable but not identical, some samples at the output of the sample-and-hold may be repeated or dropped at times. The amount of signal distortion resulting from this method can be kept small if the interpolator/decimator factor is large. By using linear interpolation in place of the digital sample-and-hold we can further reduce the distortion and thus reduce the size of the interpolator factor.

11.9.3 Implementation of Narrowband Lowpass Filters

In Section 11.6 we demonstrated that a multistage implementation of sampling-rate conversion often provides for a more efficient realization, especially when the filter specifications are very tight (e.g., a narrow passband and a narrow transition band). Under similar conditions, a lowpass, linear-phase FIR filter may be more efficiently implemented in a multistage decimator-interpolator configuration. To be more specific, we can employ a multistage implementation of a decimator of size D, followed by a multistage implementation of an interpolator of size I, where $I = D$.

We demonstrate the procedure by means of an example for the design of a lowpass filter which has the same specifications as the filter that is given in Example 11.6.1.

EXAMPLE 11.9.1

Design a linear-phase FIR filter that satisfies the following specifications:

Sampling frequency:	8000 Hz
Passband:	$0 \leq F \leq 75$ Hz
Transition band:	$75 \text{ Hz} \leq F \leq 80$ Hz
Stopband	$80 \text{ Hz} \leq F \leq 4000$ Hz
Passband ripple:	$\delta_1 = 10^{-2}$
Stopband ripple:	$\delta_2 = 10^{-4}$

Solution. If this filter were designed as a single-rate linear-phase FIR filter, the length of the filter required to meet the specifications is (from Kaiser's formula)

$$\hat{M} \approx 5152$$

Now, suppose that we employ a multirate implementation of the lowpass filter based on a decimation and interpolation factor of $D = I = 100$. A single-stage implementation of the decimator-interpolator requires an FIR filter of length

$$\hat{M}_1 = \frac{-10\log_{10}(\delta_1\delta_2/2) - 13}{14.6\Delta f} + 1 \approx 5480$$

However, there is a significant savings in computational complexity by implementing the decimator and interpolator filters using their corresponding polyphase filters. If we employ linear-phase (symmetric) decimation and interpolation filters, the use of polyphase filters reduces the multiplication rate by a factor of 100.

A significantly more efficient implementation is obtained by using two stages of decimation followed by two stages of interpolation. For example, suppose that we select $D_1 = 50$, $D_2 = 2$, $I_1 = 2$, and $I_2 = 50$. Then the required filter lengths are

$$\hat{M}_1 = \frac{-10\log(\delta_1\delta_2/4) - 13}{14.6\Delta f} + 1 \approx 177$$

$$\hat{M}_1 = \frac{-10\log_{10}(\delta_1\delta_2/4) - 13}{14.6\Delta f} + 1 \approx 233$$

Thus we obtain a reduction in the overall filter length of $2(5480)/2(177 + 233) \approx 13.36$. In addition, we obtain further reduction in the multiplication rate by using polyphase filters. For the first stage of decimation, the reduction in multiplication rate is 50, while for the second stage the reduction in multiplication rate is 100. Further reductions can be obtained by increasing the number of stages of decimation and interpolation.

11.9.4 Subband Coding of Speech Signals

A variety of techniques have been developed to efficiently represent speech signals in digital form for either transmission or storage. Since most of the speech energy is contained in the lower frequencies, we would like to encode the lower-frequency band with more bits than the high-frequency band. Subband coding is a method where the speech signal is subdivided into several frequency bands and each band is digitally encoded separately.

An example of a frequency subdivision is shown in Fig. 11.9.4(a). Let us assume that the speech signal is sampled at a rate F_s samples per second. The first frequency subdivision splits the signal spectrum into two equal-width segments, a lowpass signal ($0 \le F \le F_s/4$) and a highpass signal ($F_s/4 \le F \le F_s/2$). The second frequency subdivision splits the lowpass signal from the first stage into two equal bands, a lowpass signal ($0 < F \le F_s/8$) and a highpass signal ($F_s/8 \le F \le F_s/4$). Finally, the third frequency subdivision splits the lowpass signal from the second stage into two equal-bandwidth signals. Thus the signal is subdivided into four frequency bands, covering three octaves, as shown in Fig. 11.9.4(b).

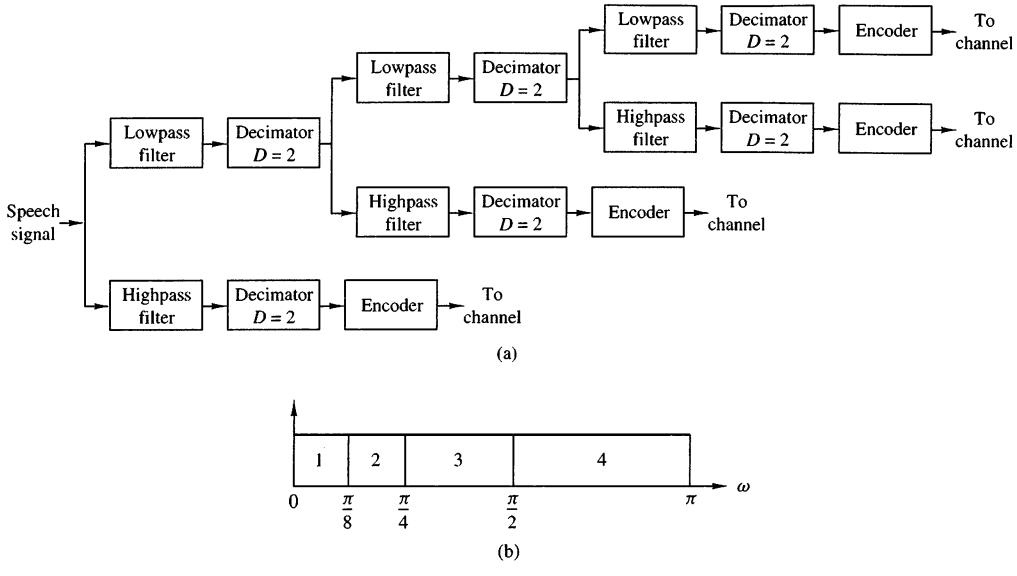


Figure 11.9.4 Block diagram of a subband speech coder.

Decimation by a factor of 2 is performed after frequency subdivision. By allocating a different number of bits per sample to the signal in the four subbands, we can achieve a reduction in the bit rate of the digitalized speech signal.

Filter design is particularly important in achieving good performance in subband coding. Aliasing resulting from decimation of the subband signals must be negligible. It is clear that we cannot use brickwall filter characteristics as shown in Fig. 11.9.5(a), since such filters are physically unrealizable. A particularly practical solution to the aliasing problem is to use *quadrature mirror filters* (QMF), which have the frequency response characteristics shown in Fig. 11.9.5(b). These filters are described in Section 11.11.

The synthesis method for the subband encoded speech signal is basically the reverse of the encoding process. The signals in adjacent lowpass and highpass frequency bands are interpolated, filtered, and combined as shown in Fig. 11.9.6. A pair of QMF is used in the signal synthesis for each octave of the signal.

Subband coding is also an effective method to achieve data compression in image signal processing. By combining subband coding with vector quantization for each subband signal, Safranek et al. (1988) have obtained coded images with approximately $\frac{1}{2}$ bit per pixel, compared with 8 bits per pixel for the uncoded image.

In general, subband coding of signals is an effective method for achieving bandwidth compression in a digital representation of the signal, when the signal energy is concentrated in a particular region of the frequency band. Multirate signal processing notions provide efficient implementations of the subband encoder.

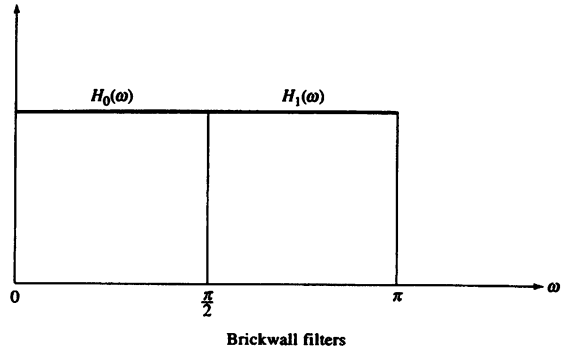


Figure 11.9.5
Filter characteristics for subband coding.

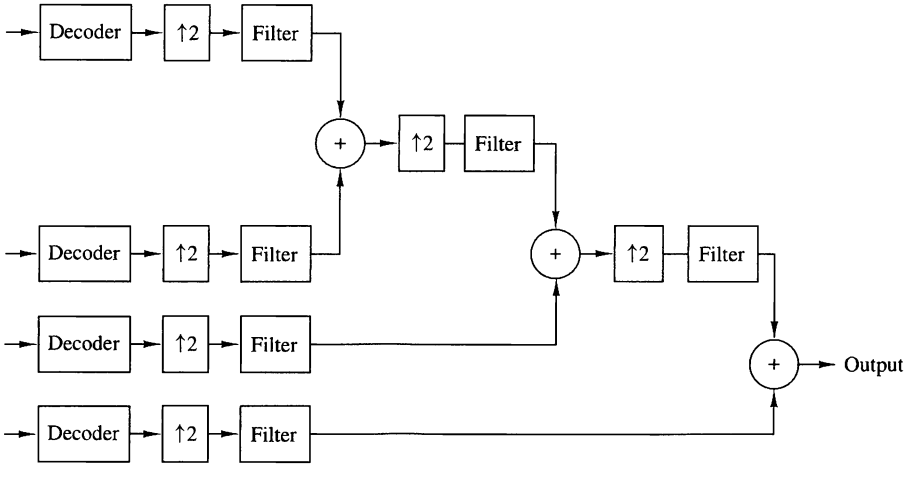
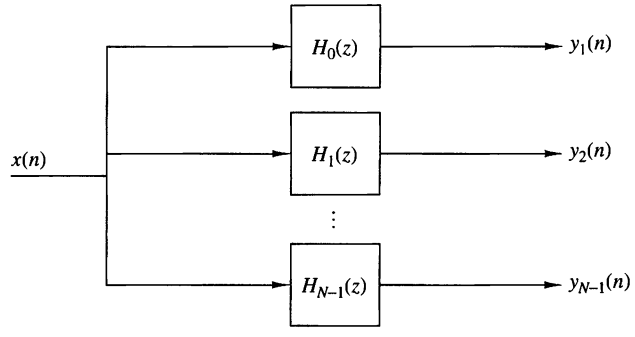


Figure 11.9.6 Synthesis of subband-encoded signals.

11.10 Digital Filter Banks

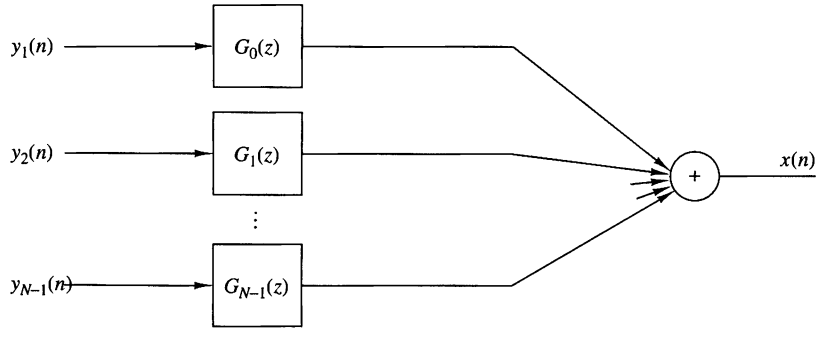
Filter banks are generally categorized as two types, *analysis filter banks* and *synthesis filter banks*. An analysis filter bank consists of a set of filters, with system functions $\{H_k(z)\}$, arranged in a parallel bank as illustrated in Fig. 11.10.1(a). The frequency response characteristics of this filter bank split the signal into a corresponding number of subbands. On the other hand, a synthesis filter bank consists of a set of filters with system functions $\{G_k(z)\}$, arranged as shown in Fig. 11.10.1(b), with corresponding inputs $\{y_k(n)\}$. The outputs of the filters are summed to form the synthesized signal $\{x(n)\}$.

Filter banks are often used for performing spectrum analysis and signal synthesis. When a filter bank is employed in the computation of the discrete Fourier transform (DFT) of a sequence $\{x(n)\}$, the filter bank is called a DFT filter bank. An analysis filter bank consisting of N filters $\{H_k(z),\ k = 0, 1, \ldots, N-1\}$ is called a *uniform DFT filter bank* if $H_k(z)$, $k = 1, 2, \ldots, N-1$, are derived from a prototype filter



Analysis filter bank

(a)



Synthesis filter bank

(b)

Figure 11.10.1 A digital filter bank.

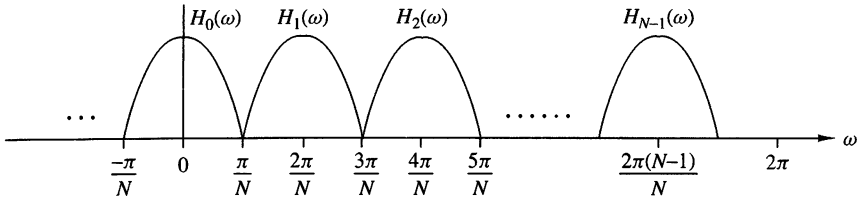


Figure 11.10.2 Illustration of frequency response characteristics of the N filters.

$H_0(z)$, where

$$H_k(\omega) = H_0\left(\omega - \frac{2\pi k}{N}\right), \qquad k = 1, 2, \ldots, N-1 \tag{11.10.1}$$

Hence the frequency response characteristics of the filters $\{H_k(z), k = 0, 1, \ldots, N-1\}$ are simply obtained by uniformly shifting the frequency response of the prototype filter by multiples of $2\pi/N$. In the time domain the filters are characterized by their impulse responses, which can be expressed as

$$h_k(n) = h_0(n)e^{j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1 \tag{11.10.2}$$

where $h_0(n)$ is the impulse response of the prototype filter, which in general may be either an FIR or an IIR filter. If $H_0(z)$ denotes the transfer function of the prototype filter, the transfer function of the kth filter is

$$H_k(z) = H_0(ze^{-j2\pi k/N}), \qquad 1 \le k \le N-1 \tag{11.10.3}$$

Figure 11.10.2 provides a conceptual illustration of the frequency response characteristics of the N filters.

The uniform DFT analysis filter bank can be realized as shown in Fig. 11.10.3(a), where the frequency components in the sequence $\{x(n)\}$ are translated in frequency to lowpass by multiplying $x(n)$ with the complex exponentials $\exp(-j2\pi nk/N)$, $k = 1, \ldots, N-1$, and the resulting product signals are passed through a lowpass filter with impulse response $h_0(n)$. Since the output of the lowpass filter is relatively narrow in bandwidth, the signal can be decimated by a factor $D \le N$. The resulting decimated output signal can be expressed as

$$X_k(m) = \sum_n h_0(mD - n)x(n)e^{-j2\pi nk/N}, \qquad \begin{matrix} k = 0, 1, \ldots, N-1 \\ m = 0, 1, \ldots \end{matrix} \tag{11.10.4}$$

where $\{X_k(m)\}$ are samples of the DFT at frequencies $\omega_k = 2\pi k/N$.

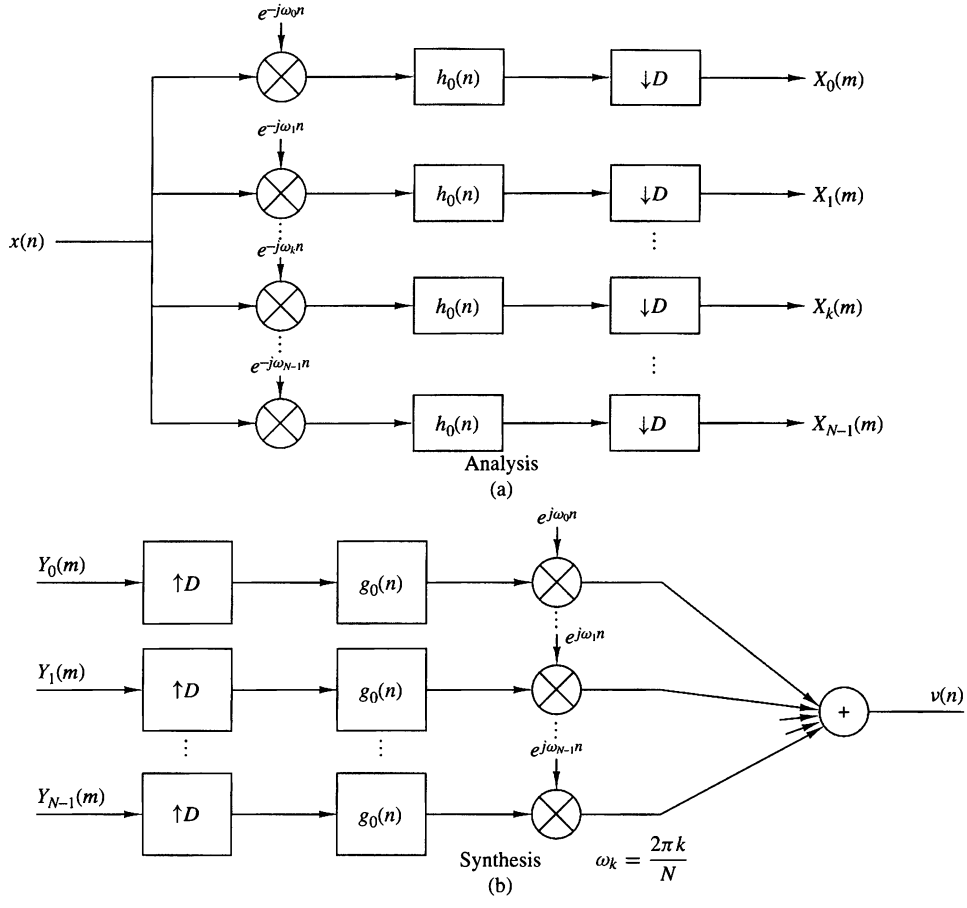


Figure 11.10.3 A uniform DFT filter bank.

The corresponding synthesis filter for each element in the filter bank can be viewed as shown in Fig. 11.10.3(b), where the input signal sequences $\{Y_k(m), k = 0, 1, \ldots, N-1\}$ are upsampled by a factor of $I = D$, filtered to remove the images, and translated in frequency by multiplication by the complex exponentials $\{\exp(j2\pi nk/N), k = 0, 1, \ldots, N-1\}$. The resulting frequency-translated signals from the N filters are then summed. Thus we obtain the sequence

$$\begin{aligned} v(n) &= \frac{1}{N}\sum_{k=0}^{N-1} e^{j2\pi nk/N}\left[\sum_m Y_k(m)g_0(n-mI)\right] \\ &= \sum_m g_0(n-mI)\left[\frac{1}{N}\sum_{k=0}^{N-1} Y_k(m)e^{j2\pi nk/N}\right] \\ &= \sum_m g_0(n-mI)y_n(m) \end{aligned} \tag{11.10.5}$$

where the factor $1/N$ is a normalization factor, $\{y_n(m)\}$ represent samples of the inverse DFT sequence corresponding to $\{Y_k(m)\}$, $\{g_0(n)\}$ is the impulse response of the interpolation filter, and $I = D$.

The relationship between the output $\{X_k(n)\}$ of the analysis filter bank and the input $\{Y_k(m)\}$ to the synthesis filter bank depends on the application. Usually, $\{Y_k(m)\}$ is a modified version of $\{X_k(m)\}$, where the specific modification is determined by the application.

An alternative realization of the analysis and synthesis filter banks is illustrated in Fig. 11.10.4. The filters are realized as bandpass filters with impulse responses

$$h_k(n) = h_0(n)e^{j2\pi nk/N}, \qquad k = 0, 1, \ldots, N-1 \tag{11.10.6}$$

The output of each bandpass filter is decimated by a factor D and multiplied by $\exp(-j2\pi mk/N)$ to produce the DFT sequence $\{X_k(m)\}$. The modulation by the complex exponential allows us to shift the spectrum of the signal from $\omega_k = 2\pi k/N$ to $\omega_0 = 0$. Hence this realization is equivalent to the realization given in Fig. 11.10.3.

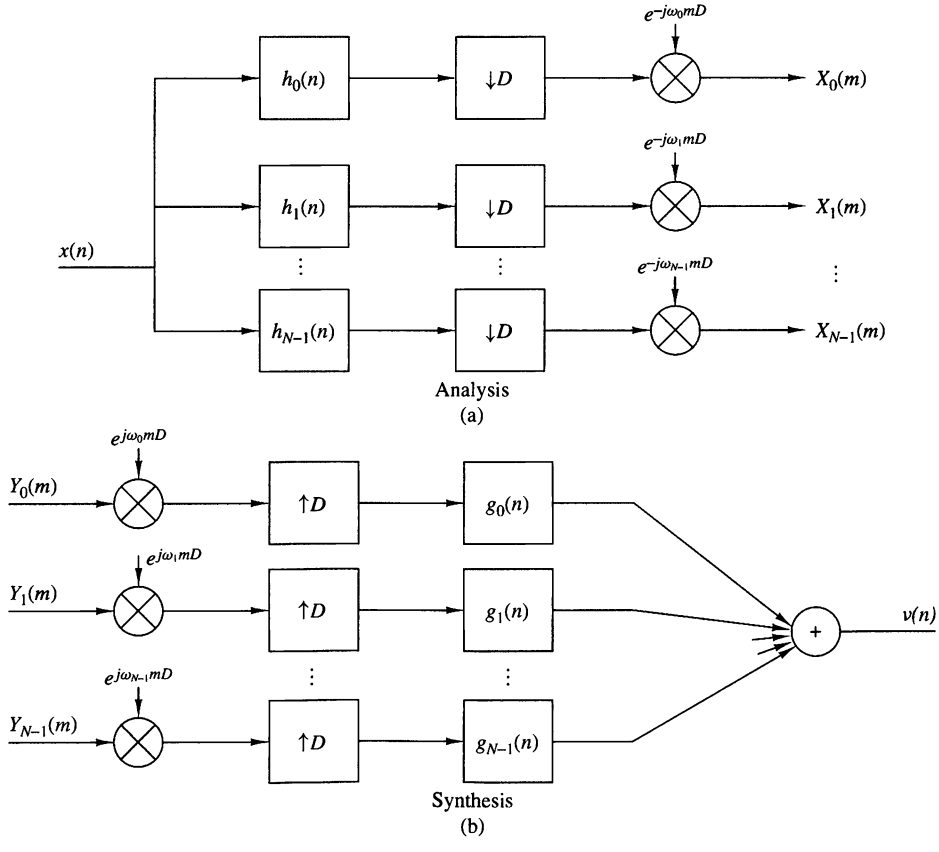


Figure 11.10.4 Alternative realization of a uniform DFT filter bank.

The analysis filter bank output can be written as

$$X_k(m) = \left[\sum_n x(n)h_0(mD-n)e^{j2\pi k(mD-n)/N}\right] e^{-j2\pi mkD/N} \tag{11.10.7}$$

The corresponding filter bank synthesizer can be realized as shown in Fig. 11.10.4-(b), where the input sequences are first multiplied by the exponential factors $[\exp(j2\pi kmD/N)]$, upsampled by the factor $I = D$, and the resulting sequences are filtered by the bandpass interpolation filters with impulse responses

$$g_k(n) = g_0(n)e^{j2\pi nk/N} \tag{11.10.8}$$

where $\{g_0(n)\}$ is the impulse response of the prototype filter. The outputs of these filters are then summed to yield

$$v(n) = \frac{1}{N}\sum_{k=0}^{N-1}\left\{\sum_m [Y_k(m)e^{j2\pi kmI/N}]g_k(n-mI)\right\} \tag{11.10.9}$$

where $I = D$.

In the implementation of digital filter banks, computational efficiency can be achieved by use of polyphase filters for decimation and interpolation. Of particular interest is the case where the decimation factor D is selected to be equal to the number N of frequency bands. When $D = N$, we say that the filter bank is *critically sampled.*

11.10.1 Polyphase Structures of Uniform Filter Banks

For the analysis filter bank, let us define a set of $N = D$ polyphase filters with impulse responses

$$p_k(n) = h_0(nN-k), \qquad k = 0, 1, \ldots, N-1 \tag{11.10.10}$$

and the corresponding set of decimated input sequences

$$x_k(n) = x(nN+k), \qquad k = 0, 1, \ldots, N-1 \tag{11.10.11}$$

Note that this definition of $\{p_k(n)\}$ implies that the commutator for the decimator rotates clockwise.

The structure of the analysis filter bank based on the use of polyphase filters can be obtained by substituting (11.10.10) and (11.10.11) into (11.10.7) and rearranging the summation into the form

$$X_k(m) = \sum_{n=0}^{N-1}\left[\sum_l p_n(l)x_n(m-l)\right] e^{-j2\pi nk/N}, \qquad k = 0, 1, \ldots, D-1 \tag{11.10.12}$$

where $N = D$. Note that the inner summation represents the convolution of $\{p_n(l)\}$ with $\{x_n(l)\}$. The outer summation represents the N-point DFT of the filter outputs. The filter structure corresponding to this computation is illustrated in Fig. 11.10.5. Each sweep of the commutator results in N outputs, denoted as $\{r_n(m),\ n = 0, 1, \ldots, N-1\}$ from the N polyphase filters. The N-point DFT of this sequence yields the spectral samples $\{X_k(m)\}$. For large values of N, the FFT algorithm provides an efficient means for computing the DFT.

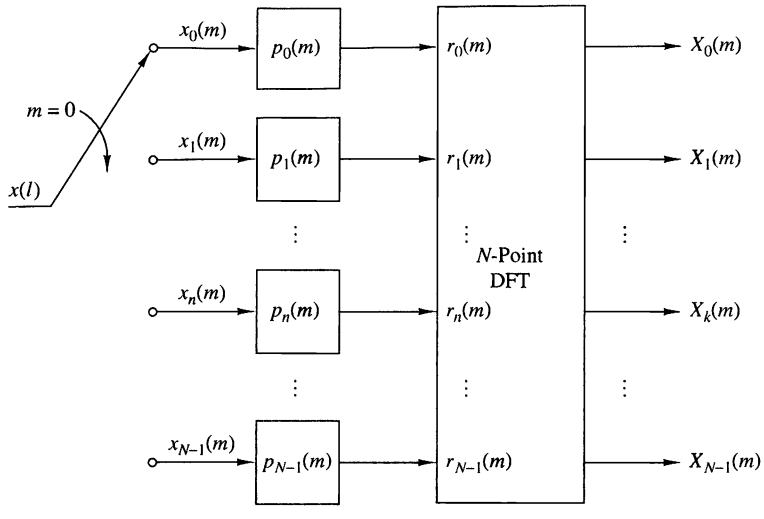


Figure 11.10.5 Digital filter bank structure for the computation of (11.10.12).

Now suppose that the spectral samples $\{X_k(m)\}$ are modified in some manner, prescribed by the application, to produce $\{Y_k(m)\}$. A filter bank synthesis filter based on a polyphase filter structure can be realized in a similar manner. First, we define the impulse response of the N $(D = I = N)$ polyphase filters for the interpolation filter as

$$q_k(n) = g_0(nN + k), \qquad k = 0, 1, \ldots, N-1 \tag{11.10.13}$$

and the corresponding set of output signals as

$$v_k(n) = v(nN + k), \qquad k = 0, 1, \ldots, N-1 \tag{11.10.14}$$

Note that this definition of $\{q_k(n)\}$ implies that the commutator for the interpolator rotates counterclockwise.

By substituting (11.10.13) into (11.10.5), we can express the output $v_l(n)$ of the lth polyphase filter as

$$v_l(n) = \sum_m q_l(n-m)\left[\frac{1}{N}\sum_{k=0}^{N-1} Y_k(m)e^{j2\pi kl/N}\right], \qquad l = 0, 1, \ldots, N-1 \tag{11.10.15}$$

The term in brackets is the N-point inverse DFT of $\{Y_k(m)\}$, which we denote as $\{y_l(m)\}$. Hence

$$v_l(n) = \sum_m q_l(n-m)y_l(m), \qquad l = 0, 1, \ldots, N-1 \tag{11.10.16}$$

The synthesis structure corresponding to (11.10.16) is shown in Fig. 11.10.6. It is interesting to note that by defining the polyphase interpolation filter as in (11.10.13), the structure in Fig. 11.10.6 is the transpose of the polyphase analysis filter shown in Fig. 11.10.5.

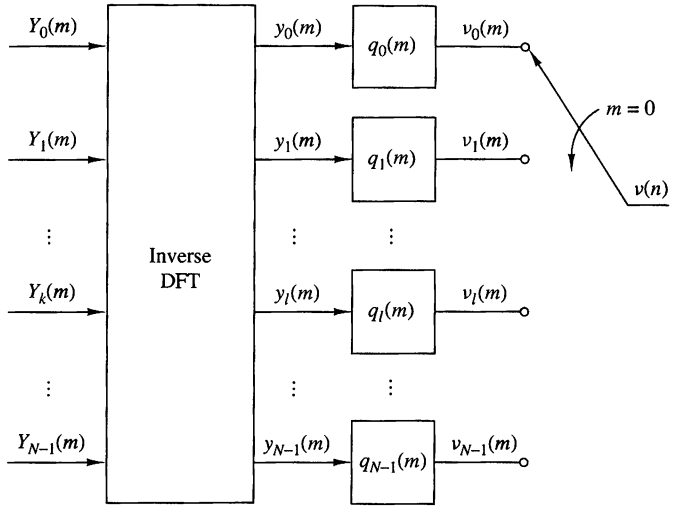


Figure 11.10.6 Digital filter bank structure for the computation of (11.10.16).

In our treatment of digital filter banks we considered the important case of critically sampled DFT filter banks, where $D = N$. Other choices of D and N can be employed in practice, but the implementation of the filters becomes more complex. Of particular importance is the oversampled DFT filter bank, where $N = KD$, D denotes the decimation factor and K is an integer that specifies the oversampling factor. In this case it can be shown that the polyphase filter bank structures for the analysis and synthesis filters can be implemented by use of N subfilters and N-point DFTs and inverse DFTs.

11.10.2 Transmultiplexers

An application of digital filter banks is in the design and implementation of digital transmultiplexers, which are devices for converting between time-division-multiplexed (TDM) signals and frequency-division-multiplexed (FDM) signals.

In a transmultiplexer for TDM-to-FDM conversion, the input signal $\{x(n)\}$ is a time-division-multiplexed signal consisting of L signals, which are separated by a commutator switch. Each of these L signals is then modulated on a different carrier frequency to obtain an FDM signal for transmission. In a transmultiplexer for FDM-to-TDM conversion, the composite signal is separated by filtering into the L signal components which are then time-division multiplexed.

In telephony, single-sideband transmission is used with channels spaced at a nominal 4-kHz bandwidth. Twelve channels are usually stacked in frequency to form a basic group channel, with a bandwidth of 48 kHz. Larger-bandwidth FDM signals are formed by frequency translation of multiple groups into adjacent frequency bands. We shall confine our discussion to digital transmultiplexers for 12-channel FDM and TDM signals.

Let us first consider FDM-to-TDM conversion. The analog FDM signal is passed through an A/D converter as shown in Fig. 11.10.7(a). The digital signal is then

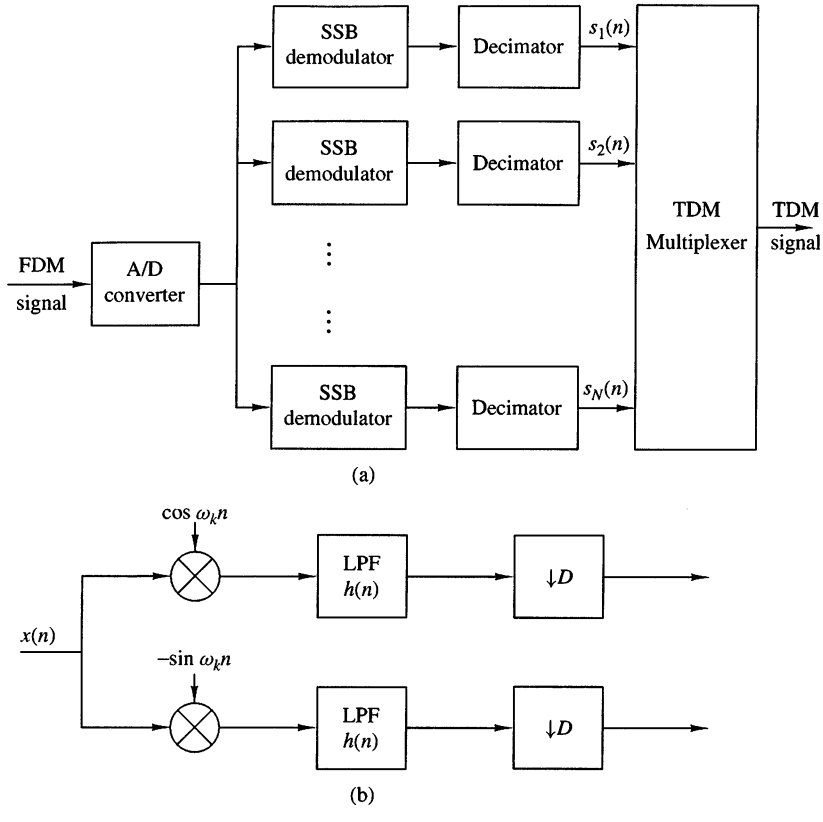


Figure 11.10.7 Block diagram of FDM-to-TDM transmultiplexer.

demodulated to baseband by means of single-sideband demodulators. The output of each demodulator is decimated and fed to the commutator of the TDM system.

To be specific, let us assume that the 12-channel FDM signal is sampled at the Nyquist rate of 96 kHz and passed through a filter-bank demodulator. The basic building block in the FDM demodulator consists of a frequency converter, a lowpass filter, and a decimator, as illustrated in Fig. 11.10.7(b). Frequency conversion can be efficiently implemented by the DFT filter bank described previously. The lowpass filter and decimator are efficiently implemented by use of the polyphase filter structure. Thus the basic structure for the FDM-to-TDM converter has the form of a DFT filter bank analyzer. Since the signal in each channel occupies a 4-kHz bandwidth, its Nyquist rate is 8 kHz, and hence the polyphase filter output can be decimated by a factor of 12. Consequently, the TDM commutator is operating at a rate of 12×8 kHz or 96 kHz.

In TDM-to-FDM conversion, the 12-channel TDM signal is demultiplexed into the 12 individual signals, where each signal has a rate of 8 kHz. The signal in each channel is interpolated by a factor of 12 and frequency converted by a single-sideband modulator, as shown in Fig. 11.10.8. The signal outputs from the 12 single-sideband modulators are summed and fed to the D/A converter. Thus we obtain the analog

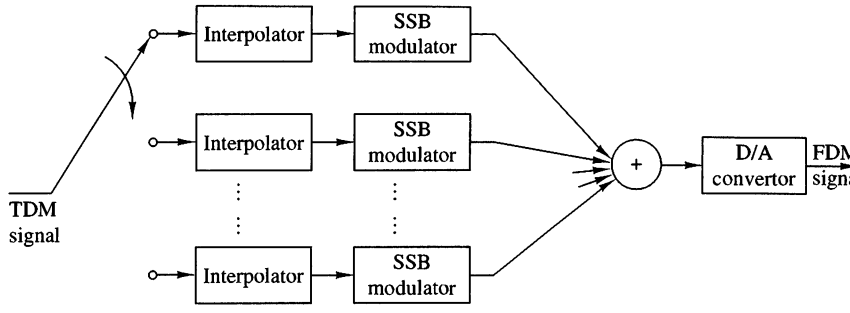


Figure 11.10.8 Block diagram of TDM-to-FDM transmultiplexer.

FDM signal for transmission. As in the case of FDM-to-TDM conversion, the interpolator and the modulator filter are combined and efficiently implemented by use of a polyphase filter. The frequency translation can be accomplished by the DFT. Consequently, the TDM-to-FDM converter encompasses the basic principles introduced previously in our discussion of DFT filter bank synthesis.

11.11 Two-Channel Quadrature Mirror Filter Bank

The basic building block in applications of quadrature mirror filters (QMF) is the two-channel QMF bank shown in Fig. 11.11.1. This is a multirate digital filter structure that employs two decimators in the "signal analysis" section and two interpolators in the "signal synthesis" section. The lowpass and highpass filters in the analysis section have impulse responses $h_0(n)$ and $h_1(n)$, respectively. Similarly, the lowpass and highpass filters contained in the synthesis section have impulse responses $g_0(n)$ and $g_1(n)$, respectively

The Fourier transforms of the signals at the outputs of the two decimators are

$$\begin{aligned} X_{a0}(\omega) &= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right)H_0\left(\frac{\omega}{2}\right) + X\left(\frac{\omega-2\pi}{2}\right)H_0\left(\frac{\omega-2\pi}{2}\right)\right] \\ X_{a1}(\omega) &= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right)H_1\left(\frac{\omega}{2}\right) + X\left(\frac{\omega-2\pi}{2}\right)H_1\left(\frac{\omega-2\pi}{2}\right)\right] \end{aligned} \tag{11.11.1}$$

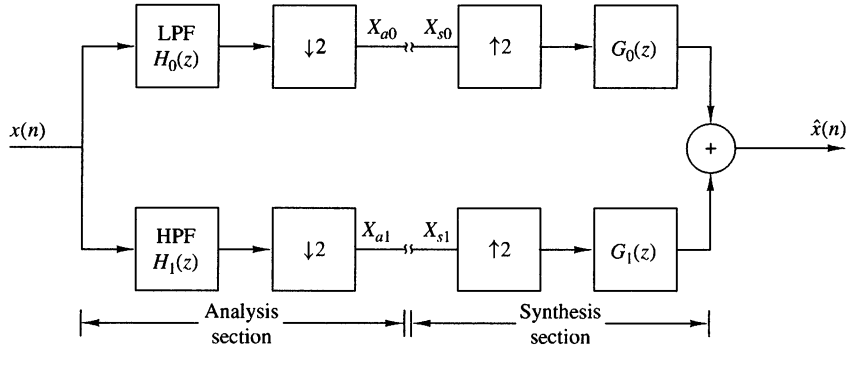


Figure 11.11.1 Two-channel QMF bank.

If $X_{s0}(\omega)$ and $X_{s1}(\omega)$ represent the two inputs to the synthesis section, the output is simply

$$\hat{X}(\omega) = X_{s0}(2\omega)G_0(\omega) + X_{s_1}(2\omega)G_1(\omega) \tag{11.11.2}$$

Now, suppose that we connect the analysis filter to the corresponding synthesis filter, so that $X_{a0}(\omega) = X_{s0}(\omega)$ and $X_{a1}(\omega) = X_{s1}(\omega)$. Then, by substituting from (11.11.1) into (11.11.2), we obtain

$$\begin{aligned}\hat{X}(\omega) &= \frac{1}{2}\left[H_0(\omega)G_0(\omega) + H_1(\omega)G_1(\omega)\right]X(\omega) \\ &\quad + \frac{1}{2}\left[H_0(\omega-\pi)G_0(\omega) + H_1(\omega-\pi)G_1(\omega)\right]X(\omega-\pi)\end{aligned} \tag{11.11.3}$$

The first term in (11.11.3) is the desired signal output from the QMF bank. The second term represents the effect of aliasing, which we would like to eliminate.

In the z-transform domain (11.11.3) is expressed as

$$\begin{aligned}\hat{X}(z) &= \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right]X(z) \\ &\quad + \frac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]X(-z) \\ &= Q(z)X(z) + A(z)X(-z)\end{aligned} \tag{11.11.4}$$

where, by definition,

$$\begin{aligned}Q(z) &= \frac{1}{2}\left[H_0(z)G_0(z) + H_1(z)G_1(z)\right] \\ A(z) &= \frac{1}{2}\left[H_0(-z)G_0(z) + H_1(-z)G_1(z)\right]\end{aligned} \tag{11.11.5}$$

11.11.1 Elimination of Aliasing

To eliminate aliasing, we require that $A(z) = 0$, i.e.,

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0 \tag{11.11.6}$$

In the frequency domain, this condition becomes

$$H_0(\omega-\pi)G_0(\omega) + H_1(\omega-\pi)G_1(\omega) = 0 \tag{11.11.7}$$

This condition can be simply satisfied by selecting $G_0(\omega)$ and $G_1(\omega)$ as

$$G_0(\omega) = H_1(\omega-\pi), \qquad G_1(\omega) = -H_0(\omega-\pi) \tag{11.11.8}$$

Thus, the second term in (11.11.3) vanishes and the filter bank is alias-free.

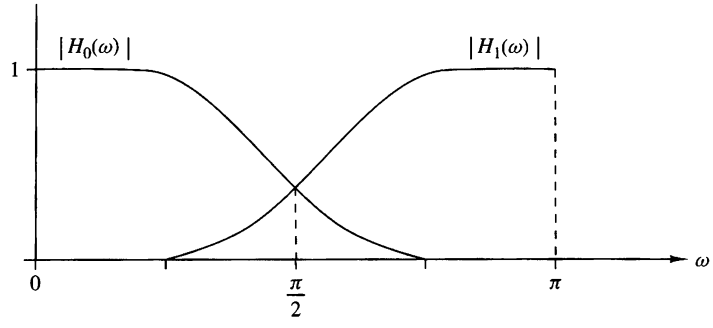


Figure 11.11.2 Mirror image characteristics of the analysis filters $H_0(\omega)$ and $H_1(\omega)$.

To elaborate, let us assume that $H_0(\omega)$ is a lowpass filter and $H_1(\omega)$ is a mirror-image highpass filter as shown in Fig. 11.11.2. Then we can express $H_0(\omega)$ and $H_1(\omega)$ as

$$\begin{aligned} H_0(\omega) &= H(\omega) \\ H_1(\omega) &= H(\omega - \pi) \end{aligned} \tag{11.11.9}$$

where $H(\omega)$ is the frequency response of a lowpass filter. In the time domain, the corresponding relations are

$$\begin{aligned} h_0(n) &= h(n) \\ h_1(n) &= (-1)^n h(n) \end{aligned} \tag{11.11.10}$$

As a consequence, $H_0(\omega)$ and $H_1(\omega)$ have mirror-image symmetry about the frequency $\omega = \pi/2$, as shown in Fig.11.11.2. To be consistent with the constraint in (11.11.8), we select the lowpass filter $G_0(\omega)$ as

$$G_0(\omega) = H(\omega) \tag{11.11.11}$$

and the highpass filter $G_1(\omega)$ as

$$G_1(\omega) = -H(\omega - \pi) \tag{11.11.12}$$

In the time domain, these relations become

$$\begin{aligned} g_0(n) &= h(n) \\ g_1(n) &= (-1)^n h(n) \end{aligned} \tag{11.11.13}$$

In the z-transform domain, the relations for the elimination of aliasing are:

$$\begin{aligned} H_0(z) &= H(z) \\ H_1(z) &= H(-z) \\ G_0(z) &= H(z) \\ G_1(z) &= -H(-z) \end{aligned} \tag{11.11.14}$$

11.11.2 Condition for Perfect Reconstruction

With $A(z) = 0$, we now consider the condition for which the output $x(n)$ of the QMF bank is identical to the input $x(n)$, except for an arbitrary delay, for all possible inputs. When this condition is satisfied, the filter bank is called a perfect reconstruction QMF bank. Thus, we require that

$$Q(z) = \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)] = z^{-k} \tag{11.11.15}$$

By making use of the relations in (11.11.14), the condition for perfect reconstruction may be expressed as

$$H^2(z) - H^2(-z) = 2z^{-k} \tag{11.11.16}$$

or, equivalently,

$$H^2(\omega) - H^2(\omega - \pi) = 2e^{-j\omega k} \tag{11.11.17}$$

Therefore, for perfect reconstruction, the frequency response $H(\omega)$ of the lowpass filter in the two-channel QMF bank must satisfy the magnitude condition

$$\left|H^2(\omega) - H^2(\omega - \pi)\right| = C \tag{11.11.18}$$

where C is a positive constant, e.g., $C = 2$. We note that if $H(\omega)$ satisfies the magnitude condition in (11.11.18) and is designed to have linear phase, then the QMF output $x(n)$ is simply a delayed version of the input sequence $x(n)$. However, linear phase is not a necessary condition for perfect reconstruction.

11.11.3 Polyphase Form of the QMF Bank

The two-channel alias-free QMF bank can be realized efficiently by employing polyphase filters. Toward this goal, $H_0(z)$ and $H_1(z)$ can be expressed as

$$\begin{aligned} H_0(z) &= P_0(z^2) + z^{-1}P_1(z^2) \\ H_1(z) &= P_0(z^2) - z^{-1}P_1(z^2) \end{aligned} \tag{11.11.19}$$

where we have used the relationship given in (11.11.14).

Similarly, using the relations given in (11.11.14), we obtain the polyphase representation of the filters $G_0(z)$ and $G_1(z)$ as

$$\begin{aligned} G_0(z) &= P_0(z^2) + z^{-1}P_1(z^2) \\ G_1(z) &= -\left[P_0(z^2) - z^{-1}P_1(z^2)\right] \end{aligned} \tag{11.11.20}$$

Thus, we obtain the polyphase realization of the QMF bank as shown in Figure 11.11.3(a). The corresponding computationally efficient polyphase realization is shown in Figure 11.11.3(b).

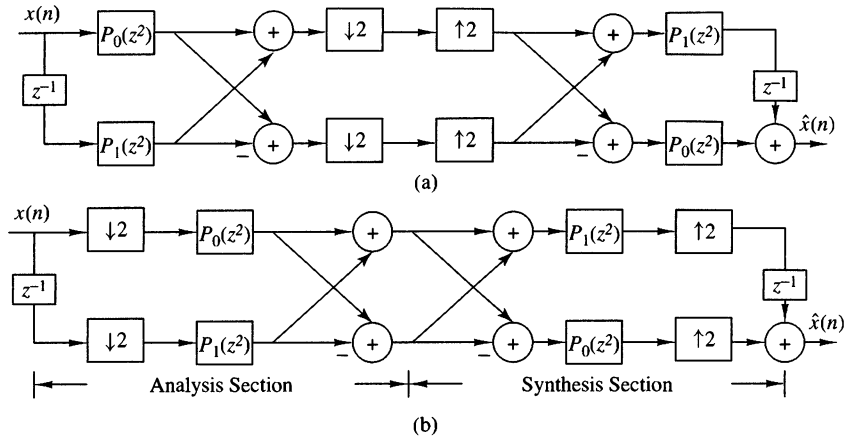


Figure 11.11.3 Polyphase realization of the two-channel QMF bank.

11.11.4 Linear Phase FIR QMF Bank

Now, let us consider the use of a linear phase filter $H(\omega)$. Hence $H(\omega)$ may be expressed in the form

$$H(\omega) = H_r(\omega)e^{-j\omega(N-1)/2} \tag{11.11.21}$$

where N is the filter length. Then

$$\begin{aligned} H^2(\omega) &= H_r^2(\omega)e^{-j\omega(N-1)} \\ &= |H(\omega)|^2 e^{-j\omega(N-1)} \end{aligned} \tag{11.11.22}$$

and

$$\begin{aligned} H^2(\omega-\pi) &= H_r^2(\omega-\pi)e^{-j(\omega-\pi)(N-1)} \\ &= (-1)^{N-1}|H(\omega-\pi)|^2 e^{-j\omega(N-1)} \end{aligned} \tag{11.11.23}$$

Therefore, the overall transfer function of the two-channel QMF which employs linear-phase FIR filters is

$$\frac{\hat{X}(\omega)}{X(\omega)} = \left[|H(\omega)|^2 - (-1)^{N-1}|H(\omega-\pi)|^2\right] e^{-j\omega(N-1)} \tag{11.11.24}$$

Note that the overall filter has a delay of $N-1$ samples and a magnitude characteristic

$$M(\omega) = |H(\omega)|^2 - (-1)^{N-1}|H(\omega-\pi)|^2 \tag{11.11.25}$$

We also note that when N is odd, $M(\pi/2) = 0$, because $|H(\pi/2)| = |H(3\pi/2)|$. This is an undesirable property for a QMF design. On the other hand, when N is even,

$$M(\omega) = |H(\omega)|^2 + |H(\omega-\pi)|^2 \tag{11.11.26}$$

which avoids the problem of a zero at $\omega = \pi/2$. For N even, the ideal two-channel QMF should satisfy the condition

$$M(\omega) = |H(\omega)|^2 + |H(\omega - \pi)|^2 = 1 \quad \text{for all } \omega \tag{11.11.27}$$

which follows from (11.11.25). Unfortunately, the only filter frequency response function that satisfies (11.11.27) is the trivial function $|H(\omega)|^2 = \cos^2 a\omega$. Consequently, any nontrivial linear-phase FIR filter $H(\omega)$ introduces some amplitude distortion.

The amount of amplitude distortion introduced by a nontrivial linear phase FIR filter in the QMF can be minimized by optimizing the FIR filter coefficients. A particularly effective method is to select the filter coefficients of $H(\omega)$ such that $M(\omega)$ is made as flat as possible while simultaneously minimizing (or constraining) the stopband energy of $H(\omega)$. This approach leads to the minimization of the integral squared error

$$J = w \int_{\omega_s}^{\pi} |H(\omega)|^2 d\omega + (1 - w) \int_0^{\pi} [M(\omega) - 1]^2 d\omega \tag{11.11.28}$$

where w is a weighting factor in the range $0 < w < 1$. In performing the optimization, the filter impulse response is constrained to be symmetric (linear phase). This optimization is easily done numerically on a digital computer. This approach has been used by Johnston (1980) and Jain and Crochiere (1984) to design two-channel QMFs. Tables of optimum filter coefficients have been tabulated by Johnston (1980).

11.11.5 IIR QMF Bank

As an alternative to linear-phase FIR filters, we can design an IIR filter that satisfies the all-pass constraint given by (11.11.18). For this purpose, elliptic filters provide especially efficient designs. Since the QMF would introduce some phase distortion, the signal at the output of the QMF can be passed through an all-pass phase equalizer designed to minimize phase distortion.

11.11.6 Perfect Reconstruction Two-Channel FIR QMF Bank

We have observed that neither linear-phase FIR filters nor IIR filters described above yield perfect reconstruction in a two-channel QMF bank. However, as shown by Smith and Barnwell (1984), perfect reconstruction can be achieved by designing $H(\omega)$ as a linear-phase FIR half-band filter of length $2N - 1$.

A half-band filter is defined as a zero-phase FIR filter whose impulse response $\{b(n)\}$ satisfies the condition

$$b(2n) = \begin{cases} \text{constant}, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{11.11.29}$$

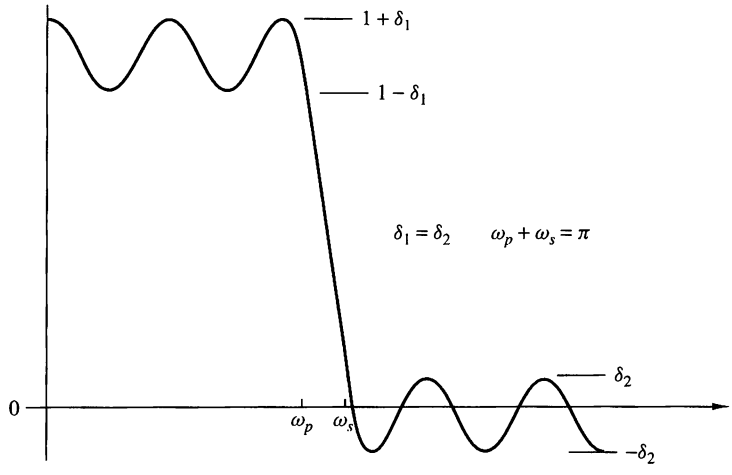


Figure 11.11.4 Frequency response characteristic of FIR half-band filter.

Hence all the even-numbered samples are zero except at $n = 0$. The zero-phase requirement implies that $b(n) = b(-n)$. The frequency response of such a filter is

$$B(\omega) = \sum_{n=-K}^{K} b(n)e^{-j\omega n} \tag{11.11.30}$$

where K is odd. Furthermore, $B(\omega)$ satisfies the condition that $B(\omega) + B(\pi - \omega)$ is equal to a constant for all frequencies. The typical frequency response characteristic of a half-band filter is shown in Fig. 11.11.4. We note that the filter response is symmetric with respect to $\pi/2$, the band edges frequencies ω_p and ω_s are symmetric about $\omega = \pi/2$, and the peak passband and stopband errors are equal. We also note that the filter can be made causal by introducing a delay of K samples.

Now, suppose that we design an FIR half-band filter of length $2N - 1$, where N is even, with frequency response as shown in Fig. 11.11.5(a). From $B(\omega)$ we construct another half-band filter with frequency response

$$B_+(\omega) = B(\omega) + \delta e^{-j\omega(N-1)} \tag{11.11.31}$$

as shown in Fig. 11.11.5(b). Note that $B_+(\omega)$ is nonnegative and hence it has the spectral factorization

$$B_+(z) = H(z)H(z^{-1})z^{-(N-1)} \tag{11.11.32}$$

or, equivalently,

$$B_+(\omega) = |H(\omega)|^2 e^{-j\omega(N-1)} \tag{11.11.33}$$

where $H(\omega)$ is the frequency response of an FIR filter of length N with real coefficients. Due to the symmetry of $B_+(\omega)$ with respect to $\omega = \pi/2$, we also have

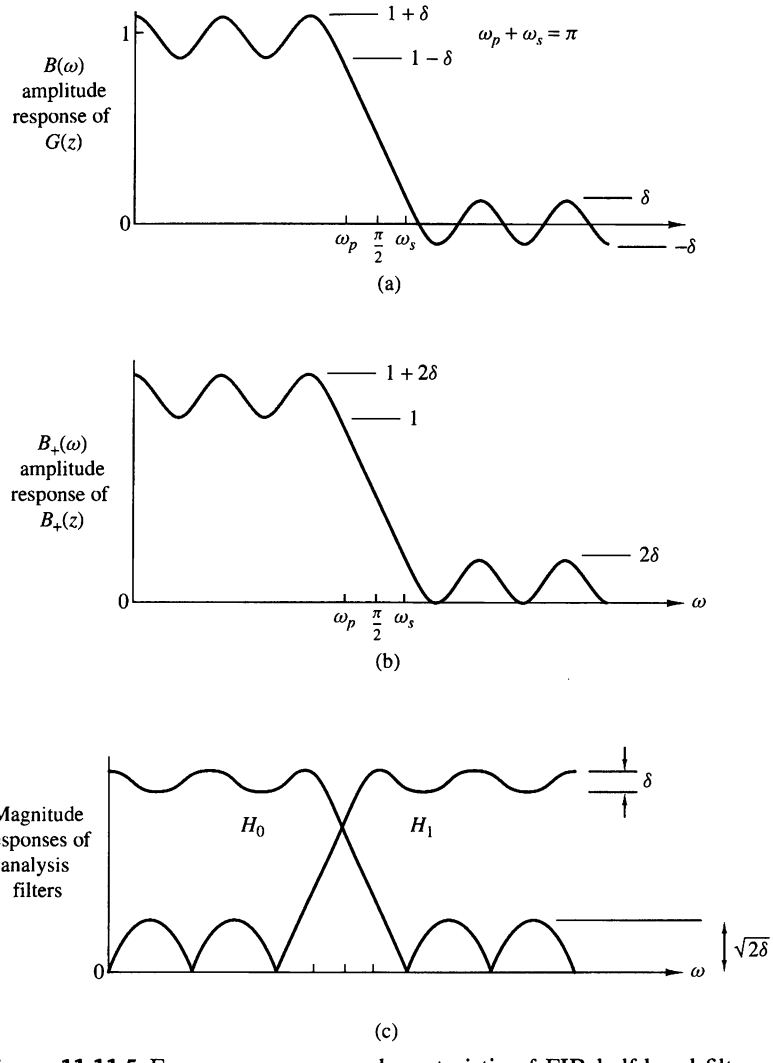


Figure 11.11.5 Frequency response characteristic of FIR half-band filters $B(\omega)$ and $B_+(\omega)$. (From Vaidyanathan (1987))

$$B_+(z) + (-1)^{N-1}B_+(-z) = \alpha z^{-(N-1)} \tag{11.11.34}$$

or, equivalently,

$$B_+(\omega) + (-1)^{N-1}B_+(\omega - \pi) = \alpha e^{-j\omega(N-1)} \tag{11.11.35}$$

where α is a constant. Thus, by substituting (11.11.32) into (11.11.34), we obtain

$$H(z)H(z^{-1}) + H(-z)H(-z^{-1}) = \alpha \tag{11.11.36}$$

Since $H(z)$ satisfies (11.11.36) and since aliasing is eliminated when we have $G_0(z) = H_1(-z)$ and $G_1(z) = -H_0(-z)$, it follows that these conditions are satisfied by choosing $H_1(z)$, $G_0(z)$, and $G_1(z)$ as

$$\begin{aligned} H_0(z) &= H(z) \\ H_1(z) &= -z^{-(N-1)} H_0(-z^{-1}) \\ G_0(z) &= z^{-(N-1)} H_0(z^{-1}) \\ G_1(z) &= z^{-(N-1)} H_1(z^{-1}) = -H_0(-z) \end{aligned} \tag{11.11.37}$$

Thus aliasing distortion is eliminated and, since $\hat{X}(\omega)/X(\omega)$ is a constant, the QMF performs perfect reconstruction so that $x(n) = \alpha x(n - N + 1)$. However, we note that $H(z)$ is not a linear-phase filter. The FIR filters $H_0(z)$, $H_1(z)$, $G_0(z)$, and $G_1(z)$ in the two-channel QMF bank are efficiently realized as polyphase filters as shown previously.

11.11.7 Two-Channel QMF Banks in Subband Coding

In Section 11.9.4 we described a method for efficient encoding of a speech signal based on subdividing the signal into several subbands and encoding each subband separately. For example, in Figure 11.9.4 we illustrated the separation of a signal into four subbands, namely, $0 \le F \le F_s/16$, $F_s/16 < F \le F_s/8$, $F_s/8 < F \le F_s/4$, and $F_s/4 < F \le F_s/2$, where F_s is the sampling frequency. The subdivision into four subbands can be accomplished by use of three two-channel QMF analysis sections. After encoding and transmission through the channel, each subband signal is decoded and reconstructed by passing the subbands through three two-channel QMF synthesis filters. The system configuration for subband coding employing four subbands is illustrated in Figure 11.11.6.

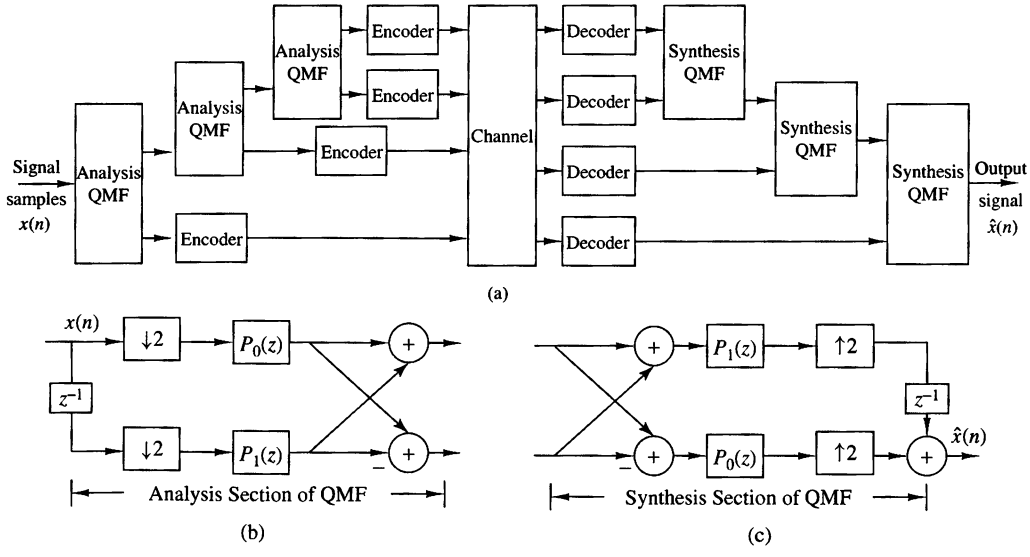


Figure 11.11.6 System for subband coding using two-channel QMF banks.

11.12 M-Channel QMF Bank

In this section, we consider the generalization of the QMF bank to M channels. Figure 11.12.1 illustrates the structure of an M-channel QMF bank, where $x(n)$ is the input to the analysis section, $x_k^{(a)}(n)$, $0 \le k \le M-1$, are the outputs of the analysis filters, $x_k^{(s)}(n)$, $0 \le k \le M-1$, are the inputs to the synthesis filters and $\hat{x}(n)$ is the output of the synthesis section.

The M outputs from the analysis filters may be expressed in the z-transform domain as

$$X_k^{(a)}(z) = \frac{1}{M} \sum_{m=0}^{M-1} H_k\left(z^{1/M} W_M^m\right) X\left(z^{1/M} W_M^m\right), \quad 0 \le k \le M-1 \tag{11.12.1}$$

where $W_M = e^{-j2\pi/M}$. The output from the synthesis section is

$$\hat{X}(z) = \sum_{k=0}^{M-1} X_k^{(s)}\left(z^M\right) G_k(z) \tag{11.12.2}$$

As in the case of the two-channel QMF bank, we set $X_k^{(a)}(z) = X_k^{(s)}(z)$. Then, if we substitute (11.12.1) into (11.12.2), we obtain

$$\begin{aligned} \hat{X}(z) &= \sum_{k=0}^{M-1} G_k(z) \left[\frac{1}{M} \sum_{m=0}^{M-1} H_k\left(zW_M^m\right) X\left(zW_M^m\right) \right] \\ &= \sum_{m=0}^{M-1} \left[\frac{1}{M} \sum_{k=0}^{M-1} G_k(z) H_k\left(zW_M^m\right) \right] X\left(zW_M^m\right) \end{aligned} \tag{11.12.3}$$

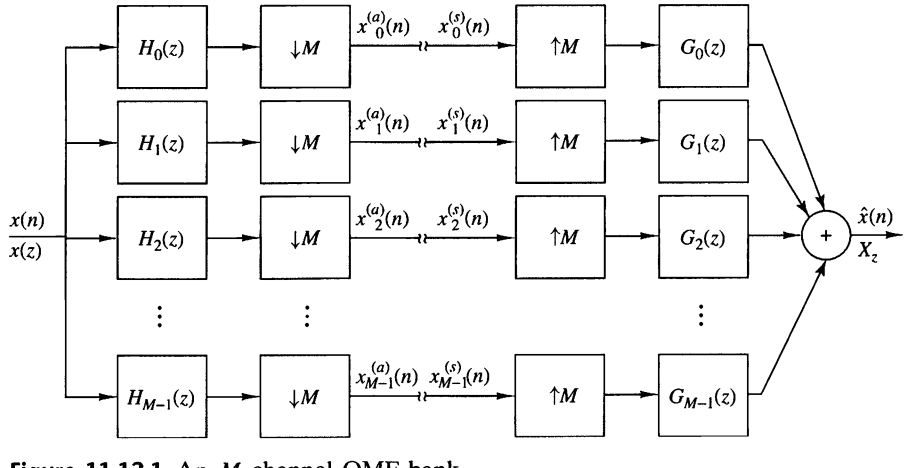


Figure 11.12.1 An M-channel QMF bank.

It is convenient to define the term in the brackets as

$$R_m(z) = \frac{1}{M}\sum_{k=0}^{M-1} G_k(z)H_k(zW_M^m), \quad 0 \le m \le M-1 \tag{11.12.4}$$

Then, (11.12.3) may be expressed as

$$\begin{aligned}\hat{X}(z) &= \sum_{m=0}^{M-1} R_m(z)X(zW_N^m) \\ &= R_0(z)X(z) + \sum_{m=1}^{M-1} R_m(z)X\left(zW_M^m\right)\end{aligned} \tag{11.12.5}$$

We note the first term in (11.12.5) is the alias-free component of the QMF bank and the second term is the aliasing component.

11.12.1 Alias-Free and Perfect Reconstruction Condition

From (11.12.5), it is clear that aliasing is eliminated by forcing the condition

$$R_m(z) = 0, \quad 1 \le m \le M-1 \tag{11.12.6}$$

With the elimination of the alias terms, the M-channel QMF bank becomes a linear time-invariant system that satisfies the input-output relation

$$\hat{X}(z) = R_0(z)X(z) \tag{11.12.7}$$

where

$$R_0(z) = \frac{1}{M}\sum_{k=0}^{M-1} H_k(z)G_k(z) \tag{11.12.8}$$

Then, the condition for a perfect reconstruction M-channel QMF bank becomes

$$R_0(z) = Cz^{-k} \tag{11.12.9}$$

where C and k are positive constants.

11.12.2 Polyphase Form of the M-Channel QMF Bank

An efficient implementation of the M-channel QMF bank is achieved by employing polyphase filters. To obtain the polyphase form for the analysis filter bank, the kth filter $H_k(z)$ is represented as

$$H_k(z) = \sum_{m=0}^{M-1} z^{-m}P_{km}(z), \qquad 0 \le k \le M-1 \tag{11.12.10}$$

We may express the equations for the M polyphase filter in matrix form as

$$\mathbf{H}(z) = \mathbf{P}(z^M)\mathbf{a}(z) \tag{11.12.11}$$

where

$$\begin{aligned} \mathbf{H}(z) &= [H_0(z)\ H_1(z)\ \cdots\ H_{M-1}(z)]^t \\ \mathbf{a}(z) &= \left[1\ z^{-1}\ z^{-2}\ \cdots\ z^{-(M-1)}\right]^t \end{aligned} \tag{11.12.12}$$

and

$$\mathbf{P}(z) = \begin{bmatrix} P_{00}(z) & P_{01}(z) & \cdots & P_{0M-1}(z) \\ P_{10}(z) & P_{11}(z) & \cdots & P_{1M-1}(z) \\ \vdots & & & \\ P_{M-1\,0}(z) & P_{M-1\,1}(z) & \cdots & P_{M-1\,M-1}(z) \end{bmatrix} \tag{11.12.13}$$

The polyphase form of the analysis filter bank is shown in Figure 11.12.2(a), and after applying the first noble identity we obtain the structure shown in Figure 11.12.2(b).

The synthesis section can be constructed in a similar manner. Suppose we use a type II (transpose) form (see Problem 11.15) for the polyphase representation of the filters $\{G_k(z)\}$. Thus,

$$G_k(z) = \sum_{m=0}^{M-1} z^{-(M-1-m)} Q_{km}(z^M), \qquad 0 \le k \le M-1 \tag{11.12.14}$$

When expressed in matrix form, (11.12.14) becomes

$$\mathbf{G}(z) = z^{-(M-1)}\mathbf{Q}^t(z^M)\mathbf{a}(z^{-1}) \tag{11.12.15}$$

where $\mathbf{a}(z)$ is defined in (11.12.12) and

$$\begin{aligned} \mathbf{G}(z) &= [\,G_0(z)\quad G_1(z)\quad \cdots\quad G_{M-1}(z)\,]^t \\ \mathbf{Q}(z) &= \begin{bmatrix} Q_{00}(z) & Q_{01}(z) & \cdots & Q_{0\,M-1}(z) \\ Q_{10}(z) & Q_{11}(z) & \cdots & Q_{1\,M-1}(z) \\ \vdots & & & \\ Q_{M-1\,0}(z) & Q_{M-1\,1}(z) & \cdots & Q_{M-1\,M-1}(z) \end{bmatrix} \end{aligned} \tag{11.12.16}$$

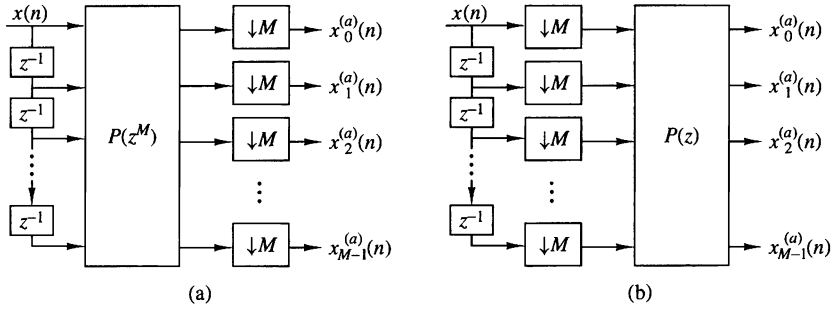


Figure 11.12.2 Polyphase structure of the analysis section of an M-channel QMF bank (a) before and (b) after applying the first noble identity.

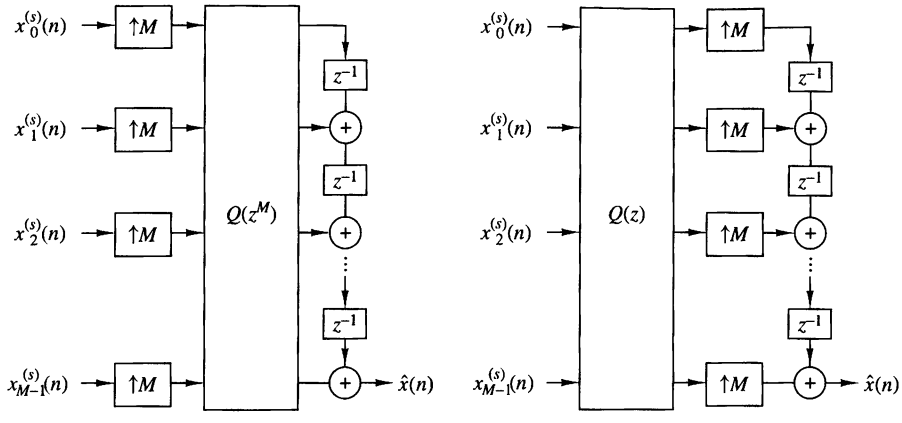


Figure 11.12.3 Polyphase structure of the synthesis section of an M-channel QMF bank (a) before and (b) after applying the first noble identity.

Therefore, the synthesis section of the M-channel QMF bank is realized as shown in Figure 11.11.3. By combining Figures 11.12.2(b) and 11.12.3(b), we obtain the polyphase structure of the complete M-channel QMF bank shown in Figure 11.12.4.

From the structure of the M-channel QMF bank shown in Figure 11.12.4, we observe that the perfect reconstruction condition can be restated as

$$\mathbf{Q}(z)\mathbf{P}(z) = Cz^{-k}\mathbf{I} \tag{11.12.17}$$

where $\mathbf{I}$ is the $M \times M$ identity matrix. Hence, if the polyphase matrix $\mathbf{P}(z)$ is known, then the polyphase synthesis matrix $\mathbf{Q}(z)$ is

$$\mathbf{Q}(z) = Cz^{-k}[\mathbf{P}(z)]^{-1} \tag{11.12.18}$$

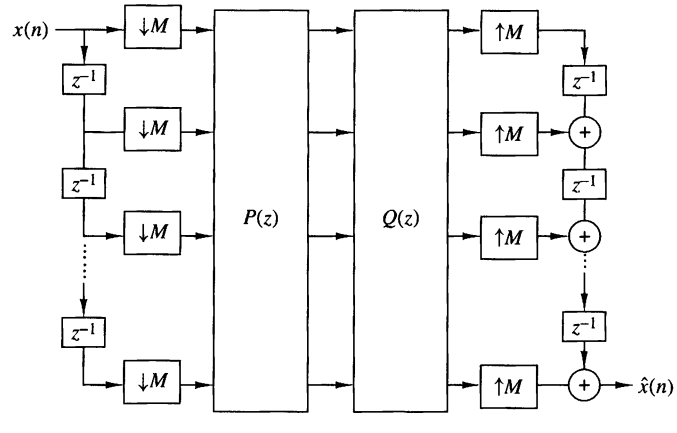


Figure 11.12.4 Polyphase realization of the M-channel QMF bank

EXAMPLE 11.12.1

Suppose the polyphase matrix for a three-channel perfect reconstruction FIR QMF bank is

$$\mathbf{P}(z^3) = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 1 \\ 1 & 2 & 1 \end{bmatrix}$$

Determine the analysis and the synthesis filters in the QMF bank.

Solution. The analysis filters are given by (11.12.11) as

$$\begin{bmatrix} H_0(z) \\ H_1(z) \\ H_2(z) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 3 & 1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ z^{-1} \\ z^{-2} \end{bmatrix}$$

Hence,

$$H_0(z) = 1 + z^{-1} + 2z^{-2}, \qquad H_1(z) = 2 + 3z^{-1} + z^{-2}, \qquad H_2(z) = 1 + 2z^{-1} + z^{-2}$$

The inverse of $\mathbf{P}(z^3)$ is

$$[\mathbf{P}(z^3)]^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 3 & -5 \\ -1 & -1 & 3 \\ 1 & -1 & 1 \end{bmatrix}$$

We may scale this inverse by the factor 2, so that

$$\mathbf{Q}(z^3) = 2[\mathbf{P}(z^3)]^{-1}$$

Then, by applying (11.12.15), we obtain the synthesis filters as

$$\begin{bmatrix} G_0(z) \\ G_1(z) \\ G_2(z) \end{bmatrix} = z^{-2} \begin{bmatrix} 1 & -1 & 1 \\ 3 & -1 & -1 \\ -5 & 3 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ z \\ z^2 \end{bmatrix}$$

Hence,

$$G_0(z) = 1 - z^{-1} + z^{-2} \qquad G_1(z) = -1 - z^{-1} + 3z^{-2} \qquad G_3(z) = 1 + 3z^{-1} - 5z^{-2}$$

Vaidyanathan (1992) treats the design of M-channel perfect reconstruction QMF banks by selecting the analysis filters $H_k(z)$ to be FIR with a paraunitary polyphase structure, i.e.,

$$\tilde{\mathbf{P}}(z)\mathbf{P}(z) = d\mathbf{I}, \qquad d > 0 \tag{11.12.19}$$

where $\tilde{\mathbf{P}}(z)$ is the paraconjugate of $\mathbf{P}(z)$. That is, $\tilde{\mathbf{P}}(z)$ is formed by the transpose of $\mathbf{P}(1/z)$, with the coefficients of $\mathbf{P}(z)$ replaced by their complex conjugates. Then, the polyphase filters in the synthesis section are designed to satisfy

$$\mathbf{Q}(z) = Cz^{-k}\tilde{\mathbf{P}}(z), \qquad C > 0, k > 0 \tag{11.12.20}$$

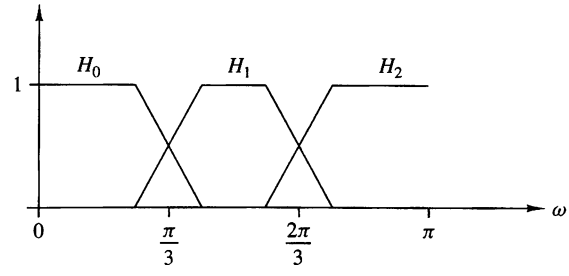


Figure 11.12.5
Magnitude response for analysis filters in an $M = 3$ QMF bank.

It can be shown (see Vaidyanathan et al. (1989)) that any causal Lth-degree FIR paraunitary matrix $\mathbf{P}(z)$ can be expressed in a product form as

$$\mathbf{P}(z) = \mathbf{V}_L(z)\mathbf{V}_{L-1}(z)\cdots\mathbf{V}_1(z)\mathbf{U} \tag{11.12.21}$$

where $\mathbf{U}$ is a unitary matrix and $\{\mathbf{V}_m(z)\}$ are paraunitary matrices that have the form

$$\mathbf{V}_m(z) = \mathbf{I} - \boldsymbol{v}_m\boldsymbol{v}_m^H + z^{-1}\boldsymbol{v}_m\boldsymbol{v}_m^H \tag{11.12.22}$$

where $\boldsymbol{v}_m$ is an M-dimensional vector of unit norm. Then, the design of the synthesis filters reduces to the optimization of the components of $\boldsymbol{v}_m$ and $\mathbf{u}_i$ by minimizing an objective function, which may be a generalization of the two-channel QMF objective function given by (11.11.28). In particular, we may minimize the objective function

$$J = \sum_{k=0}^{M-1} \int_{k\text{th stopband}} |H_k(\omega)|^2 d\omega \tag{11.12.23}$$

by employing a nonlinear optimization technique to determine the components of $\boldsymbol{v}_m$ and $\mathbf{u}_i$. Thus, the vectors $\boldsymbol{v}_m$ and $\mathbf{u}_i$ completely determine $\mathbf{P}(z)$ and, hence, the analysis filters $H_k(z)$. The synthesis filters are then determined from (11.12.20).

For example, consider the design of a perfect reconstruction three-channel QMF bank with magnitude responses as shown in Fig. 11.12.5. The design procedure described above yields the magnitude responses for the analysis filters that are shown in Fig. 11.12.6. The filter length is $N = 14$. We observe that the stopband attenuation of the filters is approximately 20 dB.

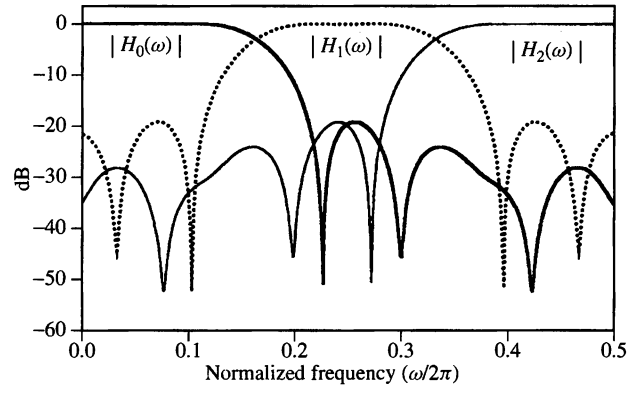


Figure 11.12.6
Magnitude response of optimized analysis filters for an $M = 3$ FIR perfect reconstruction QMF bank. (*From Multirate Systems and Filter Banks*, by P.P. Vaidyanathan, ©1993 by Prentice Hall. Reprinted with permission of the publisher.)

11.13 Summary and References

The need for sampling rate conversion arises frequently in digital signal processing applications. In this chapter we first treated sampling rate reduction (decimation) and sampling rate increase (interpolation) by integer factors and then demonstrated how the two processes can be combined to obtain sampling rate conversion by any rational factor. Then, we described a method to achieve sampling rate conversion by an arbitrary factor. In the special case where the signal to be resampled is a bandpass signal, we described methods for performing the sampling rate conversion.

In general, the implementation of sampling rate conversion requires the use of a linear time-variant filter. We described methods for implementing such filters, including the class of polyphase filter structures, which are especially simple to implement. We also described the use of multistage implementations of multirate conversion as a means of simplifying the complexity of the filter required to meet the specifications.

We also described a number of applications that employ multirate signal processing, including the implementation of narrowband filters, phase shifters, filter banks, subband speech coders, quadrature mirror filters and transmultiplexers. These are just a few of the many applications encountered in practice where multirate signal processing is used.

The first comprehensive treatment of multirate signal processing was given in the book by Crochiere and Rabiner (1983). In the technical literature, we cite the papers by Schafer and Rabiner (1973), and Crochiere and Rabiner (1975, 1976, 1981, 1983). The use of interpolation methods to achieve sampling rate conversion by an arbitrary factor is treated in a paper by Ramstad (1984). A thorough tutorial treatment of multirate digital filters and filter banks, including quadrature mirror filters, is given by Vetterli (1987), and by Vaidyanathan (1990, 1993), where many references on various applications are cited. A comprehensive survey of digital transmultiplexing methods is found in the paper by Scheuermann and Gockler (1981). Subband coding of speech has been considered in many publications. The pioneering work on this topic was done by Crochiere (1977, 1981) and by Garland and Esteban (1980). Subband coding has also been applied to coding of images. We mention the papers by Vetterli (1984), Woods and O'Neil (1986), Smith and Eddins (1988), and Safranek et al. (1988) as just a few examples. In closing, we wish to emphasize that multirate signal processing continues to be a very active research area.

Problems

11.1 An analog signal $x_a(t)$ is bandlimited to the range $900 \leq F \leq 1100$ Hz. It is used as an input to the system shown in Fig. P11.1. In this system, $H(\omega)$ is an ideal lowpass filter with cutoff frequency $F_c = 125$ Hz.

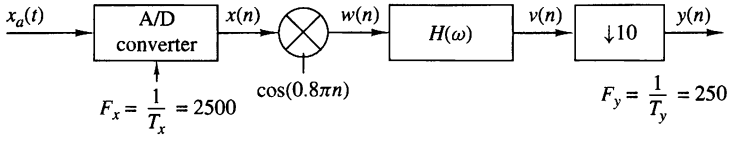


Figure P11.1

(a) Determine and sketch the spectra for the signals $x(n)$, $w(n)$, $v(n)$, and $y(n)$.

(b) Show that it is possible to obtain $y(n)$ by sampling $x_a(t)$ with period $T = 4$ milliseconds.

11.2 Consider the signal $x(n) = a^n u(n)$, $|a| < 1$.

(a) Determine the spectrum $X(\omega)$.

(b) The signal $x(n)$ is applied to a decimator that reduces the rate by a factor of 2. Determine the output spectrum.

(c) Show that the spectrum in part (b) is simply the Fourier transform of $x(2n)$.

11.3 The sequence $x(n)$ is obtained by sampling an analog signal with period T. From this signal a new signal is derived having the sampling period $T/2$ by use of a linear interpolation method described by the equation

$$y(n) = \begin{cases} x(n/2), & n \text{ even} \\ \frac{1}{2}\left[x\left(\frac{n-1}{2}\right) + x\left(\frac{n+1}{2}\right)\right], & n \text{ odd} \end{cases}$$

(a) Show that this linear interpolation scheme can be realized by basic digital signal processing elements.

(b) Determine the spectrum of $y(n)$ when the spectrum of $x(n)$ is

$$X(\omega) = \begin{cases} 1, & 0 \le |\omega| \le 0.2\pi \\ 0, & \text{otherwise} \end{cases}$$

(c) Determine the spectrum of $y(n)$ when the spectrum of $x(n)$ is

$$X(\omega) = \begin{cases} 1, & 0.7\pi \le |\omega| \le 0.9\pi \\ 0, & \text{otherwise} \end{cases}$$

11.4 Consider a signal $x(n)$ with Fourier transform

$$X(\omega) = 0, \qquad \begin{matrix} \omega_n < |\omega| \le \pi \\ f_m < |f| \le \frac{1}{2} \end{matrix}$$

(a) Show that the signal $x(n)$ can be recovered from its samples $x(mD)$ if the sampling frequency $\omega_s = 2\pi/D \le 2\omega_m (f_s = 1/D \ge 2f_m)$.

(b) Show that $x(n)$ can be reconstructed using the formula

$$x(n) = \sum_{k=-\infty}^{\infty} x(kD) h_r(n - kD)$$

where

$$h_r(n) = \frac{\sin(2\pi f_c n)}{2\pi n}, \qquad \begin{matrix} f_m < f_c < f_s - f_m \\ \omega_m < \omega_c < \omega_s - \omega_m \end{matrix}$$

(c) Show that the bandlimited interpolation in part (b) can be thought of as a two-step process, first, increasing the sampling rate by a factor of D by inserting $(D-1)$ zero samples between successive samples of the decimated signal $x_a(n) = x(nD)$, and second, filtering the resulting signal using an ideal lowpass filter with cutoff frequency ω_c.

11.5 In this problem we illustrate the concepts of sampling and decimation for discrete-time signals. To this end consider a signal $x(n)$ with Fourier transform $X(\omega)$ as in Fig. P11.5.

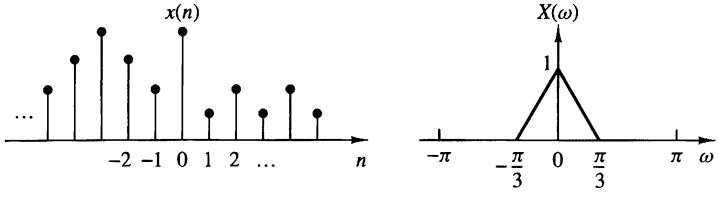


Figure P11.5

(a) Sampling $x(n)$ with a sampling period $D = 2$ results in the signal

$$x_s(n) = \begin{cases} x(n), & n = 0, \pm 2, \pm 4, \ldots \\ 0, & n = \pm 1, \pm 3, \pm 5, \ldots \end{cases}$$

Compute and sketch the signal $x_s(n)$ and its Fourier transform $X_s(\omega)$. Can we reconstruct $x(n)$ from $x_s(n)$? How?

(b) Decimating $x(n)$ by a factor of $D = 2$ produces the signal

$$x_d(n) = x(2n), \qquad \text{all } n$$

Show that $X_d(\omega) = X_s(\omega/2)$. Plot the signal $x_d(n)$ and its transform $X_d(\omega)$. Do we lose any information when we decimate the sampled signal $x_s(n)$?

11.6 Design a decimator that downsamples an input signal $x(n)$ by a factor $D = 5$. Use the Remez algorithm to determine the coefficients of the FIR filter that has a 0.1-dB ripple in the passband ($0 \leq \omega \leq \pi/5$) and is down by at least 30 dB in the stopband. Also determine the corresponding polyphase filter structure for implementing the decimator.

11.7 Design an interpolator that increases the input sampling rate by a factor of $I = 2$. Use the Remez algorithm to determine the coefficients of the FIR filter that has a 0.1-dB ripple in the passband ($0 \leq \omega \leq \pi/2$) and is down by at least 30 dB in the stopband. Also, determine the corresponding polyphase filter structure for implementing the interpolator.

11.8 Design a sample rate converter that reduces the sampling rate by a factor $\frac{2}{5}$. Use the Remez algorithm to determine the coefficients of the FIR filter that has a 0.1-dB ripple in the passband and is down by at least 30 dB in the stopband. Specify the sets of time-variant coefficients $g(n, m)$ and the corresponding coefficients in the polyphase filter realization of the sample rate converter.

11.9 Consider the two different ways of cascading a decimator with an interpolator shown in Fig. P11.9.

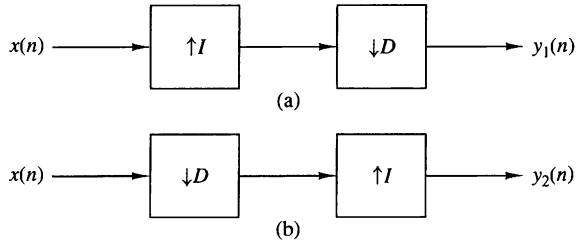


Figure P11.9

(a) If $D = I$, show that the outputs of the two configurations are different. Hence, in general, the two systems are not identical.

(b) Show that the two systems are identical if and only if D and I are relatively prime.

11.10 Prove the equivalence of the two decimator and interpolator configurations shown in Fig. P11.10 (noble identities) (see Vaidyanathan, 1990).

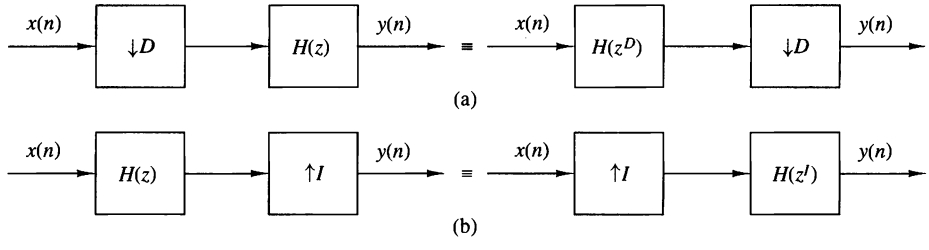


Figure P11.10

11.11 Consider an arbitrary digital filter with transfer function

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n}$$

(a) Perform a two-component polyphase decomposition of $H(z)$ by grouping the even-numbered samples $h_0(n) = h(2n)$ and the odd-numbered samples $h_1(n) = h(2n+1)$. Thus show that $H(z)$ can be expressed as

$$H(z) = H_0(z^2) + z^{-1}H_1(z^2)$$

and determine $H_0(z)$ and $H_1(z)$.

(b) Generalize the result in part (a) by showing that $H(z)$ can be decomposed into a D-component polyphase filter structure with transfer function

$$H(z) = \sum_{k=0}^{D-1} z^{-k} H_k(z^D)$$

Determine $H_k(z)$.

(c) For the IIR filter with transfer function

$$H(z) = \frac{1}{1 - az^{-1}}$$

determine $H_0(z)$ and $H_1(z)$ for the two-component decomposition.

11.12 A sequence $x(n)$ is upsampled by $I = 2$, it passes through an LTI system $H_1(z)$, and then it is downsampled by $D = 2$. Can we replace this process with a single LTI system $H_2(z)$? If the answer is positive, determine the system function of this system.

11.13 Plot the signals and their corresponding spectra for rational sampling rate conversion by (a) $I/D = 5/3$ and (b) $I/D = 3/5$. Assume that the spectrum of the input signal $x(n)$ occupies the entire range $-\pi \leq \omega_x \leq \pi$.

11.14 We wish to design an efficient nonrecursive decimator for $D = 8$ using the factorization

$$H(z) = [(1 + z^{-1})(1 + z^{-2})(1 + z^{-4}) \ldots (1 + z^{-2^{K-1}})]^5$$

(a) Derive an efficient implementation using filters with system function $H_k(z) = (1 + z^{-1})^5$.

(b) Show that each stage of the obtained decimator can be implemented more efficiently using a polyphase decomposition.

11.15 Let us consider an alternative polyphase decomposition by defining new polyphase filters $Q_m(z^N)$ as

$$H(z) = \sum_{m=0}^{N-1} z^{-(N-1-m)} Q_m(z^N)$$

This polyphase decomposition is called a type II form to distinguish it from the conventional decomposition based on polyphase filters $P_m(z^N)$.

(a) Show that the type II form polyphase filters $Q_m(z)$ are related to the polyphase filters $P_m(z^N)$ as follows:

$$Q_m(z^N) = P_{N-1-m}(z^N)$$

(b) Sketch the polyphase filter structure for $H(z)$ based in the polyphase filters $Q_m(z^N)$ and, thus, show that this structure is an alternative transpose form.

11.16 Use the result in Problem 11.15 to determine the type II form of the $I = 3$ interpolator in Figure 11.5.9.

11.17 Design a two-stage decimator for the following specifications:

$$D = 100$$

Passband: $0 \leq F \leq 50$

Transition band: $50 \leq F \leq 55$

Input sampling rate: 10,000 Hz

Ripple: $\delta_1 = 10^{-1}, \delta_2 = 10^{-3}$

11.18 Design a linear-phase FIR filter that satisfies the following specifications based on a single-stage and a two-stage multirate structure:

Sampling rate: 10,000 Hz

Passband: $0 \leq F \leq 60$

Transition band: $60 \leq F \leq 65$

Ripple: $\delta_1 = 10^{-1}, \delta_2 = 10^{-3}$

11.19 Prove that the half-band filter that satisfies (11.11.35) is always odd and the even coefficients are zero.

11.20 Design one-stage and two-stage interpolators to meet the following specifications:

$$I = 20$$

Input sampling rate: 10,000 Hz

Passband: $0 \leq F \leq 90$

Transition band: $90 \leq F \leq 100$

Ripple: $\delta_1 = 10^{-2}, \delta_2 = 10^{-3}$

11.21 By using (11.10.15) derive the equations corresponding to the structure for the polyphase synthesis section shown in Fig. 11.10.6.

11.22 Show that the transpose of an L-stage interpolator for increasing the sampling rate by an integer factor I is equivalent to an L-stage decimator that decreases the sampling rate by a factor $D = I$.

11.23 Sketch the polyphase filter structure for achieving a time advance of $(k/I)T_x$ in a sequence $x(n)$.

11.24 Prove the following expressions for an interpolator of order I.

(a) The impulse response $h(n)$ can be expressed as

$$h(n) = \sum_{k=0}^{I-1} p_k(n-k)$$

where

$$p_k(n) = \begin{cases} p_k(n/I), & n = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases}$$

(b) $H(z)$ may be expressed as

$$H(z) = \sum_{k=0}^{I-1} z^{-k} p_k(z)$$

(c) $$P_k(z) = \frac{1}{I} \sum_{n=-\infty}^{\infty} \sum_{l=0}^{I-1} h(n) e^{j2\pi l(n-k)/I} z^{-(n-k)/I}$$

$$P_k(\omega) = \frac{1}{I} \sum_{l=0}^{I-1} H\left(\omega - \frac{2\pi l}{I}\right) e^{j(\omega - 2\pi l)k/I}$$

11.25 Consider the interpolation of a signal by a factor I. The interpolation filter with transfer function $H(z)$ is implemented by a polyphase filter structure based on an alternative (type II) decomposition, namely

$$H(z) = \sum_{m=0}^{I-1} z^{-(I-1-m)} Q_m(z^I)$$

Determine and sketch the structure of the interpolator that employs the polyphase filters $Q_m(z)$, $0 \le m \le I-1$.

11.26 In the polyphase filter structures for a uniform DFT filter bank described in Section 11.10.1, the polyphase filters in the analysis section were defined by equation (11.10.10). Instead of this definition, suppose we define the N-band polyphase filters for the lowpass prototype filter, $H_0(z)$, as

$$H_0(z) = \sum_{i=0}^{N-1} z^{-i} P_i(z^N)$$

where

$$P_i(z) = \sum_{n=0}^{\infty} h_0(nN + i) z^{-n}, \qquad 0 \le i \le N-1$$

Then,

$$H_k(z) = H_0(z\, e^{-j2\pi k/N}) = H_0(z W_N^k)$$

where $W_N = e^{-j2\pi/N}$.

(a) Show that the filters $H_k(z)$, $1 \le k \le N-1$, can be expressed as

$$H_k(z) = \begin{bmatrix} 1 & W_N^{-k} & W_N^{-2k} & \cdots & W_N^{-(N-1)k} \end{bmatrix} \begin{bmatrix} P_0(z^N) \\ z^{-1} P_1(z^N) \\ \vdots \\ z^{-(N-1)} P_{N-1}(z^N) \end{bmatrix}$$

(b) Show that

$$\begin{bmatrix} H_0(z) \\ H_1(z) \\ \vdots \\ H_{N-1}(z) \end{bmatrix} = N\mathbf{W}^{-1} \begin{bmatrix} P_0(z^N) \\ z^{-1}P_1(z^N) \\ \vdots \\ z^{-(N-1)}P_{N-1}(z^N) \end{bmatrix}$$

where $\mathbf{W}$ is the DFT matrix

$$\mathbf{W} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}$$

(c) Sketch the analysis section of the uniform DFT filter bank.

(d) Determine and sketch the synthesis section of the uniform DFT filter bank.

11.27 The prototype filter in a four-channel uniform DFT filter bank is characterized by the transfer function

$$H_0(z) = 1 + z^{-1} + 3z^{-2} + 4z^{-4}$$

(a) Determine the transfer functions of the filters $H_1(z)$, $H_2(z)$ and $H_3(z)$ in the analysis section.

(b) Determine the transfer functions of the filters in the synthesis section.

(c) Sketch the analysis and synthesis sections of the uniform DFT filter bank.

11.28 Consider the following FIR filter transfer function:

$$H(z) = -3 + 19z^{-2} + 32z^{-3} + 19z^{-4} - 3z^{-6}$$

(a) Show that $H(z)$ is a linear-phase filter.

(b) Show that $H(z)$ is a half-band filter.

(c) Plot the magnitude and phase responses of the filter.

11.29 The analysis filter $H_0(z)$ in a two-channel QMF has the transfer function

$$H_0(z) = 1 + z^{-1}$$

(a) Determine the polyphase filters $P_0(z^2)$ and $P_1(z^2)$.

(b) Determine the analysis filter $H_1(z)$ and sketch the two-channel analysis section that employs polyphase filters.

(c) Determine the synthesis filters $G_0(z)$ and $G_1(z)$ and sketch the entire two-channel QMF based on polyphase filters.

(d) Show that the QMF bank results in perfect reconstruction.

11.30 The analysis filters in a three-channel QMF bank have the transfer functions

$$H_0(z) = 1 + z^{-1} + z^{-2}$$
$$H_1(z) = 1 - z^{-1} + z^{-2}$$
$$H_2(z) = 1 - z^{-2}$$

(a) Determine the polyphase matrix $\mathbf{P}(z^3)$ and express the analysis filters in the form (11.12.11).

(b) Determine the synthesis filters $G_0(z)$, $G_1(z)$, and $G_2(z)$ that result in perfect reconstruction.

(c) Sketch the three-channel QMF bank that employs polyphase filters in the analysis and synthesis sections.

11.31 *Zoom-frequency analysis* Consider the system in Fig. P11.31(a).

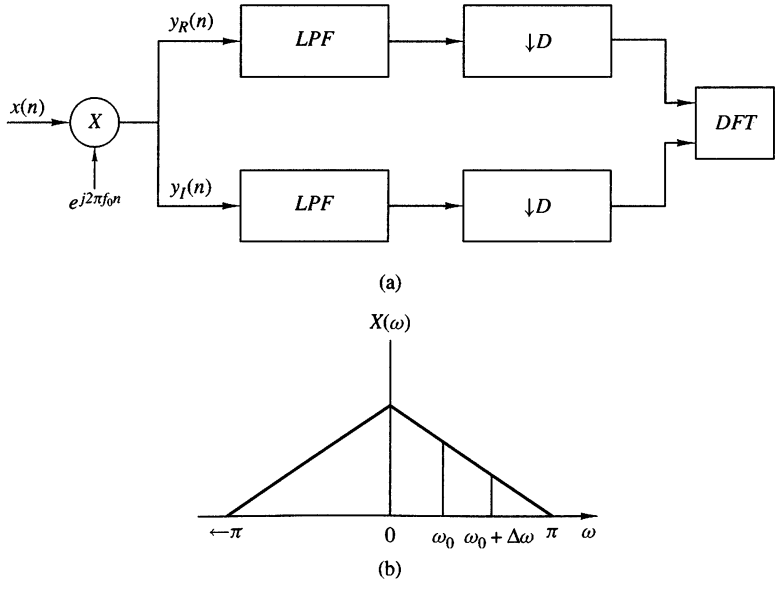


Figure P11.31

(a) Sketch the spectrum of the signal $y(n) = y_R(n) + jy_I(n)$ if the input signal $x(n)$ has the spectrum shown in Fig. P11.31(b).

(b) Suppose that we are interested in the analysis of the frequencies in the band $f_0 \leq f \leq f_0 + \Delta f$, where $f_0 = 1/6$ and $\Delta f = 1/3$. Determine the cutoff of a lowpass filter and the decimation factor D required to retain the information contained in this band of frequencies.

(c) Assume that

$$x(n) = \sum_{k=0}^{p-1} \left(1 - \frac{k}{2p}\right) \cos 2\pi f_k n$$

where $p = 40$ and $f_k = k/p$, $k = 0, 1, \ldots, p-1$. Compute and plot the 1024-point DFT of $x(n)$.

(d) Repeat part (b) for the signal $x(n)$ given in part (c) by using an appropriately designed lowpass linear phase FIR filter to determine the decimated signal $s(n) = s_R(n) + js_I(n)$.

(e) Compute the 1024-point DFT of $s(n)$ and investigate to see if you have obtained the expected results.

CHAPTER

12

Linear Prediction and Optimum Linear Filters

The design of filters to perform signal estimation is a problem that frequently arises in the design of communication systems, control systems, in geophysics, and in many other applications and disciplines. In this chapter we treat the problem of optimum filter design from a statistical viewpoint. The filters are constrained to be linear and the optimization criterion is based on the minimization of the mean-square error. As a consequence, only the second-order statistics (autocorrelation and crosscorrelation functions) of a stationary process are required in the determination of the optimum filters.

Included in this treatment is the design of optimum filters for linear prediction. Linear prediction is a particularly important topic in digital signal processing, with applications in a variety of areas, such as speech signal processing, image processing, and noise suppression in communication systems. As we shall observe, determination of the optimum linear filter for prediction requires the solution of a set of linear equations that have some special symmetry. To solve these linear equations, we describe two algorithms, the Levinson–Durbin algorithm and the Schur algorithm, which provide the solution to the equations through computationally efficient procedures that exploit the symmetry properties.

The last section of the chapter treats an important class of optimum filters called Wiener filters. Wiener filters are widely used in many applications involving the estimation of signals corrupted with additive noise.

12.1 Random Signals, Correlation Functions, and Power Spectra

We begin with a brief review of the characterization of random signals in terms of statistical averages expressed in both the time domain and the frequency domain. The

reader is assumed to have a background in probability theory and random processes, at the level given in the books of Helstrom (1990), Peebles (1987), and Stark and Woods (1994).

12.1.1 Random Processes

Many physical phenomena encountered in nature are best characterized in statistical terms. For example, meteorological phenomena such as air temperature and air pressure fluctuate randomly as a function of time. Thermal noise voltages generated in the resistors of electronic devices, such as a radio or television receiver, are also randomly fluctuating phenomena. These are just a few examples of random signals. Such signals are usually modeled as infinite-duration infinite-energy signals.

Suppose that we take the set of waveforms corresponding to the air temperature in different cities around the world. For each city there is a corresponding waveform that is a function of time, as illustrated in Fig. 12.1.1. The set of all possible waveforms

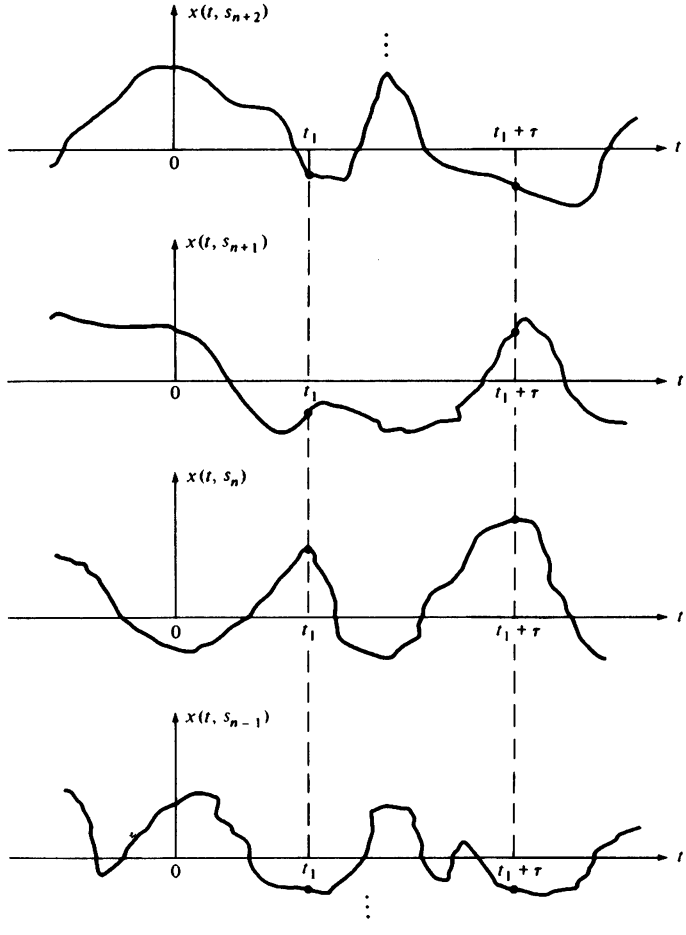


Figure 12.1.1 Sample functions of a random process.

is called an *ensemble* of time functions or, equivalently, a *random process*. The waveform for the temperature in any particular city is a *single realization* or a *sample function* of the random process.

Similarly, the thermal noise voltage generated in a resistor is a single realization or a sample function of the random process consisting of all noise voltage waveforms generated by the set of all resistors.

The set (ensemble) of all possible noise waveforms of a random process is denoted as $X(t, S)$, where t represents the time index and S represents the set (sample space) of all possible sample functions. A single waveform in the ensemble is denoted by $x(t, s)$. Usually, we drop the variable s (or S) for notational convenience, so that the random process is denoted as $X(t)$ and a single realization is denoted as $x(t)$.

Having defined a random process $X(t)$ as an ensemble of sample functions, let us consider the values of the process for any set of time instants $t_1 > t_2 > \cdots > t_n$, where n is any positive integer. In general, the samples $X_{t_i} \equiv x(t_i)$, $i = 1, 2, \ldots, n$ are n random variables characterized statistically by their joint probability density function (PDF) denoted as $p(x_{t_1}, x_{t_2}, \ldots, x_{t_n})$ for any n.

12.1.2 Stationary Random Processes

Suppose that we have n samples of the random process $X(t)$ at $t = t_i$, $i = 1, 2, \ldots, n$, and another set of n samples displaced in time from the first set by an amount τ. Thus the second set of samples are $X_{t_i+\tau} \equiv X(t_i + \tau)$, $i = 1, 2, \ldots, n$, as shown in Fig. 12.1.1. This second set of n random variables is characterized by the joint probability density function $p(x_{t_i+\tau}, \ldots, x_{t_n+\tau})$. The joint PDFs of the two sets of random variables may or may not be identical. When they are identical, then

$$p(x_{t_1}, x_{t_2}, \ldots, x_{t_n}) = p(x_{t_1+\tau}, x_{t_2+\tau}, \ldots, x_{t_n+\tau}) \tag{12.1.1}$$

for all τ and all n, and the random process is said to be *stationary in the strict sense*. In other words, the statistical properties of a stationary random process are invariant to a translation of the time axis. On the other hand, when the joint PDFs are different, the random process is nonstationary.

12.1.3 Statistical (Ensemble) Averages

Let us consider a random process $X(t)$ sampled at time instant $t = t_i$. Thus $X(t_i)$ is a random variable with PDF $p(x_{t_i})$. The lth *moment* of the random variable is defined as the *expected value* of $X^l(t_i)$, that is,

$$E(X_{t_i}^l) = \int_{-\infty}^{\infty} x_{t_i}^l \, p(x_{t_i}) dx_{t_i} \tag{12.1.2}$$

In general, the value of the lth moment depends on the time instant t_i, if the PDF of X_{t_i} depends on t_i. When the process is stationary, however, $p(x_{t_i+\tau}) = p(x_{t_i})$ for all τ. Hence the PDF is independent of time and, consequently, the lth moment is independent of time (a constant).

Next, let us consider the two random variables $X_{t_i} = X(t_i)$, $i = 1, 2$, corresponding to samples of $X(t)$ taken at $t = t_1$ and $t = t_2$. The statistical (ensemble) correlation between X_{t_1} and X_{t_2} is measured by the joint moment

$$E(X_{t_1} X_{t_2}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{t_1} x_{t_2} p(x_{t_1} x_{t_2})\, dx_1\, dx_2 \tag{12.1.3}$$

Since the joint moment depends on the time instants t_1 and t_2, it is denoted as $\gamma_{xx}(t_1, t_2)$ and is called the *autocorrelation function* of the random process. When the process $X(t)$ is stationary, the joint PDF of the pair (X_{t_1}, X_{t_2}) is identical to the joint PDF of the pair $(X_{t_1+\tau}, X_{t_2+\tau})$ for any arbitrary τ. This implies that the autocorrelation function of $X(t)$ depends on the time difference $t_1 - t_2 = \tau$. Hence for a stationary real-valued random process the autocorrelation function is

$$\gamma_{xx}(\tau) = E[X_{t_1+\tau} X_{t_1}] \tag{12.1.4}$$

On the other hand,

$$\gamma_{xx}(-\tau) = E(X_{t_1-\tau} X_{t_1}) = E(X_{t_1'} X_{t_1'+\tau}) = \gamma_{xx}(\tau) \tag{12.1.5}$$

Therefore, $\gamma_{xx}(\tau)$ is an even function. We also note that $\gamma_{xx}(0) = E(X_{t_1}^2)$ is the *average power* of the random process.

There exist nonstationary processes with the property that the mean value of the process is a constant and the autocorrelation function satisfies the property $\gamma_{xx}(t_1, t_2) = \gamma_{xx}(t_1 - t_2)$. Such a process is called *wide-sense stationary*. Clearly, wide-sense stationarity is a less stringent condition than strict-sense stationarity. In our treatment we shall require only that the processes be wide-sense stationary.

Related to the autocorrelation function is the autocovariancefunction, which is defined as

$$\begin{aligned} c_{xx}(t_1, t_2) &= E\{[X_{t_1} - m(t_1)][X_{t_2} - m(t_2)]\} \\ &= \gamma_{xx}(t_1, t_2) - m(t_1)m(t_2) \end{aligned} \tag{12.1.6}$$

where $m(t_1) = E(X_{t_1})$ and $m(t_2) = E(X_{t_2})$ are the mean values of X_{t_1} and X_{t_2}, respectively. When the process is stationary,

$$c_{xx}(t_1, t_2) = c_{xx}(t_1 - t_2) = c_{xx}(\tau) = \gamma_{xx}(\tau) - m_x^2 \tag{12.1.7}$$

where $\tau = t_1 - t_2$. Furthermore, the variance of the process is $\sigma_x^2 = c_{xx}(0) = \gamma_{xx}(0) - m_x^2$.

12.1.4 Statistical Averages for Joint Random Processes

Let $X(t)$ and $Y(t)$ be two random processes and let $X_{t_i} \equiv X(t_i)$, $i = 1, 2, \ldots, n$, and $Y_{t_j'} \equiv Y(t_j')$, $j = 1, 2, \ldots, m$, represent the random variables at times $t_1 > t_2 >$

$\cdots > t_n$ and $t'_1 > t'_2 > \cdots > t'_m$, respectively. The two sets of random variables are characterized statistically by the joint PDF

$$p(x_{t_1}, x_{t_2}, \ldots, x_{t_n}, y_{t'_1}, y_{t'_2}, \ldots, y_{t'_m})$$

for any set of time instants $\{t_i\}$ and $\{t'_j\}$ and for any positive integer values of m and n.

The *crosscorrelation function* of $X(t)$ and $Y(t)$, denoted as $\gamma_{xy}(t_1, t_2)$, is defined by the joint moment

$$\gamma_{xy}(t_1, t_2) \equiv E(X_{t_1} Y_{t_2}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_{t_1} y_{t_2} p(x_{t_1}, y_{t_2}) dx_{t_1} dy_{t_2} \tag{12.1.8}$$

and the crosscovariance is

$$c_{xy}(t_1, t_2) = \gamma_{xy}(t_1, t_2) - m_x(t_1) m_y(t_2) \tag{12.1.9}$$

When the random processes are jointly and individually stationary, we have $\gamma_{xy}(t_1, t_2) = \gamma_{xy}(t_1 - t_2)$ and $c_{xy}(t_1, t_2) = c_{xy}(t_1 - t_2)$. In this case

$$\gamma_{xy}(-\tau) = E(X_{t_1} Y_{t_1+\tau}) = E(X_{t'_1-\tau} Y_{t'_1}) = \gamma_{yx}(\tau) \tag{12.1.10}$$

The random processes $X(t)$ and $Y(t)$ are said to be statistically independent if and only if

$$p(x_{t_1}, x_{t_2}, \ldots, x_{t_n}, y_{t'_1}, y_{t'_2}, \ldots, y_{t'_m}) = p(x_{t_1}, \ldots, x_{t_n}) p(y_{t'_1}, \ldots, y_{t'_m})$$

for all choices of t_i, t'_i and for all positive integers n and m. The processes are said to be uncorrelated if

$$\gamma_{xy}(t_1, t_2) = E(X_{t_1}) E(Y_{t_2}) \tag{12.1.11}$$

so that $c_{xy}(t_1, t_2) = 0$.

A complex-valued random process $Z(t)$ is defined as

$$Z(t) = X(t) + jY(t) \tag{12.1.12}$$

where $X(t)$ and $Y(t)$ are random processes. The joint PDF of the complex-valued random variables $Z_{t_i} \equiv Z(t_i)$, $i = 1, 2, \ldots$, is given by the joint PDF of the components (X_{t_i}, Y_{t_i}), $i = 1, 2, \ldots, n$. Thus the PDF that characterizes Z_{t_i}, $i = 1, 2, \ldots, n$ is

$$p(x_{t_1}, x_{t_2}, \ldots, x_{t_n}, y_{t_1}, y_{t_2}, \ldots, y_{t_n})$$

A complex-valued random process $Z(t)$ is encountered in the representation of the in-phase and quadrature components of the lowpass equivalent of a narrowband random signal or noise. An important characteristic of such a process is its autocorrelation function, which is defined as

$$\begin{aligned} \gamma_{zz}(t_1, t_2) &= E(Z_{t_1} Z^*_{t_2}) \\ &= E[(X_{t_1} + jY_{t_1})(X_{t_2} - jY_{t_2})] \\ &= \gamma_{xx}(t_1, t_2) + \gamma_{yy}(t_1, t_2) + j[\gamma_{yx}(t_1, t_2) - \gamma_{xy}(t_1, t_2)] \end{aligned} \tag{12.1.13}$$

When the random processes $X(t)$ and $Y(t)$ are jointly and individually stationary, the autocorrelation function of $Z(t)$ becomes

$$\gamma_{zz}(t_1, t_2) = \gamma_{zz}(t_1 - t_2) = \gamma_{zz}(\tau)$$

where $\tau = t_1 - t_2$. The complex conjugate of (12.1.13) is

$$\gamma_{zz}^*(\tau) = E(Z_{t_1}^* Z_{t_1-\tau}) = \gamma_{zz}(-\tau) \tag{12.1.14}$$

Now, suppose that $Z(t) = X(t) + jY(t)$ and $W(t) = U(t) + jV(t)$ are two complex-valued random processes. Their crosscorrelation function is defined as

$$\begin{aligned}\gamma_{zw}(t_1, t_2) &= E(Z_{t_1} W_{t_2}^*) \\ &= E[(X_{t_1} + jY_{t_1})(U_{t_2} - jV_{t_2})] \\ &= \gamma_{xu}(t_1, t_2) + \gamma_{yv}(t_1, t_2) + j[\gamma_{yu}(t_1, t_2) - \gamma_{xv}(t_1, t_2)]\end{aligned} \tag{12.1.15}$$

When $X(t)$, $Y(t)$, $U(t)$, and $V(t)$ are pairwise stationary, the crosscorrelation functions in (12.1.15) become functions of the time difference $\tau = t_1 - t_2$. In addition, we have

$$\gamma_{zw}^*(\tau) = E(Z_{t_1}^* W_{t_1-\tau}) = E(Z_{t_1+\tau}^* W_{t_1}) = \gamma_{wz}(-\tau) \tag{12.1.16}$$

12.1.5 Power Density Spectrum

A stationary random process is an infinite-energy signal and hence its Fourier transform does not exist. The spectral characteristic of a random process is obtained according to the Wiener–Khintchine theorem, by computing the Fourier transform of the autocorrelation function. That is, the distribution of power with frequency is given by the function

$$\Gamma_{xx}(F) = \int_{-\infty}^{\infty} \gamma_{xx}(\tau) e^{-j2\pi F\tau} d\tau \tag{12.1.17}$$

The inverse Fourier transform is given as

$$\gamma_{xx}(\tau) = \int_{-\infty}^{\infty} \Gamma_{xx}(F) e^{j2\pi F\tau} dF \tag{12.1.18}$$

We observe that

$$\begin{aligned}\gamma_{xx}(0) &= \int_{-\infty}^{\infty} \Gamma_{xx}(F) dF \\ &= E(X_t^2) \geq 0\end{aligned} \tag{12.1.19}$$

Since $E(X_t^2) = \gamma_{xx}(0)$ represents the average power of the random process, which is the area under $\Gamma_{xx}(F)$, it follows that $\Gamma_{xx}(F)$ is the distribution of power as a function of frequency. For this reason, $\Gamma_{xx}(F)$ is called the *power density spectrum* of the random process.

If the random process is real, $\gamma_{xx}(\tau)$ is real and even and hence $\Gamma_{xx}(F)$ is real and even. If the random process is complex valued, $\gamma_{xx}(\tau) = \gamma_{xx}^*(-\tau)$ and, hence

$$\Gamma_{xx}^*(F) = \int_{-\infty}^{\infty} \gamma_{xx}^*(\tau) e^{j2\pi F\tau} d\tau = \int_{-\infty}^{\infty} \gamma_{xx}^*(-\tau) e^{-j2\pi F\tau} d\tau$$
$$= \int_{-\infty}^{\infty} \gamma_{xx}(\tau) e^{-j2\pi F\tau} d\tau = \Gamma_{xx}(F)$$

Therefore, $\Gamma_{xx}(F)$ is always real.

The definition of the power density spectrum can be extended to two jointly stationary random processes $X(t)$ and $Y(t)$, which have a crosscorrelation function $\gamma_{xy}(\tau)$. The Fourier transform of $\gamma_{xy}(\tau)$ is

$$\Gamma_{xy}(F) = \int_{-\infty}^{\infty} \gamma_{xy}(\tau) e^{-j2\pi F\tau} d\tau \tag{12.1.20}$$

which is called the *cross-power density spectrum*. It is easily shown that $\Gamma_{xy}^*(F) = \Gamma_{yx}(-F)$. For real random processes, the condition is $\Gamma_{yx}(F) = \Gamma_{xy}(-F)$.

12.1.6 Discrete-Time Random Signals

This characterization of continuous-time random signals can be easily carried over to discrete-time signals. Such signals are usually obtained by uniformly sampling a continuous-time random process.

A discrete-time random process $X(n)$ consists of an ensemble of sample sequences $x(n)$. The statistical properties of $X(n)$ are similar to the characterization of $X(t)$, with the restriction that n is now an integer (time) variable. To be specific, we state the form for the important moments that we use in this text.

The lth moment of $X(n)$ is defined as

$$E(X_n^l) = \int_{-\infty}^{\infty} x_n^l p(x_n) dx_n \tag{12.1.21}$$

and the autocorrelation sequence is

$$\gamma_{xx}(n, k) = E(X_n X_k) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_n x_k p(x_n, x_k) dx_n dx_k \tag{12.1.22}$$

Similarly, the autocovariance is

$$c_{xx}(n, k) = \gamma_{xx}(n, k) - E(X_n)E(X_k) \tag{12.1.23}$$

For a stationary process, we have the special forms ($m = n - k$)

$$\begin{aligned} \gamma_{xx}(n - k) &= \gamma_{xx}(m) \\ c_{xx}(n - k) &= c_{xx}(m) = \gamma_{xx}(m) - m_x^2 \end{aligned} \tag{12.1.24}$$

where $m_x = E(X_n)$ is the mean of the random process. The variance is defined as $\sigma^2 = c_{xx}(0) = \gamma_{xx}(0) - m_x^2$.

For a complex-valued stationary process $Z(n) = X(n) + jY(n)$, we have

$$\gamma_{zz}(m) = \gamma_{xx}(m) + \gamma_{yy}(m) + j[\gamma_{yx}(m) - \gamma_{xy}(m)] \tag{12.1.25}$$

and the crosscorrelation sequence of two complex-valued stationary sequences is

$$\gamma_{zw}(m) = \gamma_{xu}(m) + \gamma_{yv}(m) + j[\gamma_{yu}(m) - \gamma_{xv}(m)] \tag{12.1.26}$$

As in the case of a continuous-time random process, a discrete-time random process has infinite energy but a finite average power and is given as

$$E(X_n^2) = \gamma_{xx}(0) \tag{12.1.27}$$

By use of the Wiener–Khintchine theorem, we obtain the power density spectrum of the discrete-time random process by computing the Fourier transform of the autocorrelation sequence $\gamma_{xx}(m)$, that is,

$$\Gamma_{xx}(f) = \sum_{m=-\infty}^{\infty} \gamma_{xx}(m)e^{-j2\pi fm} \tag{12.1.28}$$

The inverse transform relationship is

$$\gamma_{xx}(m) = \int_{-1/2}^{1/2} \Gamma_{xx}(f)e^{j2\pi fm}df \tag{12.1.29}$$

We observe that the average power is

$$\gamma_{xx}(0) = \int_{-1/2}^{1/2} \Gamma_{xx}(f)df \tag{12.1.30}$$

so that $\Gamma_{xx}(f)$ is the distribution of power as a function of frequency, that is, $\Gamma_{xx}(f)$ is the power density spectrum of the random process $X(n)$. The properties we have stated for $\Gamma_{xx}(F)$ also hold for $\Gamma_{xx}(f)$.

12.1.7 Time Averages for a Discrete-Time Random Process

Although we have characterized a random process in terms of statistical averages, such as the mean and the autocorrelation sequence, in practice, we usually have available a single realization of the random process. Let us consider the problem of obtaining the averages of the random process from a single realization. To accomplish this, the random process must be *ergodic*.

By definition, a random process $X(n)$ is ergodic if, with probability 1, all the statistical averages can be determined from a single sample function of the process. In effect, the random process is ergodic if time averages obtained from a single

realization are equal to the statistical (ensemble) averages. Under this condition we can attempt to estimate the ensemble averages using time averages from a single realization.

To illustrate this point, let us consider the estimation of the mean and the autocorrelation of the random process from a single realization $x(n)$. Since we are interested only in these two moments, we define ergodicity with respect to these parameters. For additional details on the requirements for mean ergodicity and autocorrelation ergodicity which are given below, the reader is referred to the book of Papoulis (1984).

12.1.8 Mean-Ergodic Process

Given a stationary random process $X(n)$ with mean

$$m_x = E(X_n)$$

let us form the *time average*

$$\hat{m}_x = \frac{1}{2N+1} \sum_{n=-N}^{N} x(n) \tag{12.1.31}$$

In general, we view $\hat{m}_x$ in (12.1.31) as an estimate of the statistical mean whose value will vary with the different realizations of the random process. Hence $\hat{m}_x$ is a random variable with a PDF $p(\hat{m}_x)$. Let us compute the expected value of $\hat{m}_x$ over all possible realizations of $X(n)$. Since the summation and the expectation are linear operations, we can interchange them, so that

$$E(\hat{m}_x) = \frac{1}{2N+1} \sum_{n=-N}^{N} E[x(n)] = \frac{1}{2N+1} \sum_{n=-N}^{N} m_x = m_x \tag{12.1.32}$$

Since the mean value of the estimate is equal to the statistical mean, we say that the estimate $\hat{m}_x$ is unbiased.

Next, we compute the variance of $\hat{m}_x$. We have

$$\text{var}(\hat{m}_x) = E(|\hat{m}_x|^2) - |m_x|^2$$

But

$$E(|\hat{m}_x|^2) = \frac{1}{(2N+1)^2} \sum_{n=-N}^{N} \sum_{k=-N}^{N} E[x^*(n)x(k)]$$

$$= \frac{1}{(2N+1)^2} \sum_{n=-N}^{N} \sum_{k=-N}^{N} \gamma_{xx}(k-n)$$

$$= \frac{1}{2N+1} \sum_{m=-2N}^{2N} \left(1 - \frac{|m|}{2N+1}\right) \gamma_{xx}(m)$$

Therefore,

$$\begin{aligned}\text{var}(\hat{m}_x) &= \frac{1}{2N+1}\sum_{m=-2N}^{2N}\left(1-\frac{|m|}{2N+1}\right)\gamma_{xx} - |m_x|^2 \\ &= \frac{1}{2N+1}\sum_{m=-2N}^{2N}\left(1-\frac{|m|}{2N+1}\right)c_{xx}(m)\end{aligned} \tag{12.1.33}$$

If $\text{var}(m_x) \to 0$ as $N \to \infty$, the estimate converges with probability 1 to the statistical mean m_x. Therefore, the process $X(n)$ is mean ergodic if

$$\lim_{N\to\infty}\frac{1}{2N+1}\sum_{m=-2N}^{2N}\left(1-\frac{|m|}{2N+1}\right)c_{xx}(m) = 0 \tag{12.1.34}$$

Under this condition, the estimate $\hat{m}_x$ in the limit as $N \to \infty$ becomes equal to the statistical mean, that is,

$$m_x = \lim_{N\to\infty}\frac{1}{2N+1}\sum_{n=-N}^{N}x(n) \tag{12.1.35}$$

Thus the time-average mean, in the limit as $N \to \infty$, is equal to the ensemble mean.

A sufficient condition for (12.1.34) to hold is if

$$\sum_{m=-\infty}^{\infty}|c_{xx}(m)| < \infty \tag{12.1.36}$$

which implies that $c_{xx}(m) \to 0$ as $m \to \infty$. This condition holds for most zero-mean processes encountered in the physical world.

12.1.9 Correlation-Ergodic Processes

Now, let us consider the estimate of the autocorrelation $\gamma_{xx}(m)$ from a single realization of the process. Following our previous notation, we denote the estimate (for a complex-valued signal, in general) as

$$r_{xx}(m) = \frac{1}{2N+1}\sum_{n=-N}^{N}x^*(n)x(n+m) \tag{12.1.37}$$

Again, we regard $r_{xx}(m)$ as a random variable for any given lag m, since it is a function of the particular realization. The expected value (mean value over all realizations) is

$$\begin{aligned}E[r_{xx}(m)] &= \frac{1}{2N+1}\sum_{n=-N}^{N}E[x^*(n)x(n+m)] \\ &= \frac{1}{2N+1}\sum_{n=-N}^{N}\gamma_{xx}(m) = \gamma_{xx}(m)\end{aligned} \tag{12.1.38}$$

Therefore, the expected value of the time-average autocorrelation is equal to the statistical average. Hence we have an unbiased estimate of $\gamma_{xx}(m)$.

To determine the variance of the estimate $r_{xx}(m)$, we compute the expected value of $|r_{xx}(m)|^2$ and subtract the square of the mean value. Thus

$$\text{var}[r_{xx}(m)] = E[|r_{xx}(m)|^2] - |\gamma_{xx}(m)|^2 \tag{12.1.39}$$

But

$$E[|r_{xx}(m)|^2] = \frac{1}{(2N+1)^2} \sum_{n=-N}^{N} \sum_{k=-N}^{N} E[x^*(n)x(n+m)x(k)x^*(k+m)] \tag{12.1.40}$$

The expected value of the term $x^*(n)x(n+m)x(k)x^*(k+m)$ is just the autocorrelation sequence of a random process defined as

$$v_m(n) = x^*(n)x(n+m)$$

Hence (12.1.40) may be expressed as

$$\begin{aligned} E[|r_{xx}(m)|^2] &= \frac{1}{(2N+1)^2} \sum_{n=-N}^{N} \sum_{k=-N}^{N} \gamma_{vv}^{(m)}(n-k) \\ &= \frac{1}{2N+1} \sum_{n=-2N}^{2N} \left(1 - \frac{|n|}{2N+1}\right) \gamma_{vv}^{(m)}(n) \end{aligned} \tag{12.1.41}$$

and the variance is

$$\text{var}[r_{xx}(m)] = \frac{1}{2N+1} \sum_{n=-2N}^{2N} \left(1 - \frac{|n|}{2N+1}\right) \gamma_{vv}^{(m)}(n) - |\gamma_{xx}(m)|^2 \tag{12.1.42}$$

If $\text{var}[r_{xx}(m)] \to 0$ as $N \to \infty$, the estimate $r_{xx}(m)$ converges with probability 1 to the statistical autocorrelation $\gamma_{xx}(m)$. Under these conditions, the process is correlation ergodic and the time-average correlation is identical to the statistical average, that is,

$$\lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} x^*(n)x(n+m) = \gamma_{xx}(m) \tag{12.1.43}$$

In our treatment of random signals, we assume that the random processes are mean ergodic and correlation ergodic, so that we can deal with time averages of the mean and the autocorrelation obtained from a single realization of the process.

12.2 Innovations Representation of a Stationary Random Process

In this section we demonstrate that a wide-sense stationary random process can be represented as the output of a causal and causally invertible linear system excited by a white noise process. The condition that the system is causally invertible also allows us to represent the wide-sense stationary random process by the output of the inverse system, which is a white noise process.

Let us consider a wide-sense stationary process $\{x(n)\}$ with autocorrelation sequence $\{\gamma_{xx}(m)\}$ and power spectral density $\Gamma_{xx}(f)$, $|f| \leq \frac{1}{2}$. We assume that $\Gamma_{xx}(f)$ is real and continuous for all $|f| \leq \frac{1}{2}$. The z-transform of the autocorrelation sequence $\{\gamma_{xx}(m)\}$ is

$$\Gamma_{xx}(z) = \sum_{m=-\infty}^{\infty} \gamma_{xx}(m) z^{-m} \tag{12.2.1}$$

from which we obtain the power spectral density by evaluating $\Gamma_{xx}(z)$ on the unit circle [i.e., by substituting $z = \exp(j2\pi f)$].

Now, let us assume that $\log \Gamma_{xx}(z)$ is analytic (possesses derivatives of all orders) in an annular region in the z-plane that includes the unit circle (i.e., $r_1 < |z| < r_2$ where $r_1 < 1$ and $r_2 > 1$). Then, $\log \Gamma_{xx}(z)$ can be expanded in a Laurent series of the form

$$\log \Gamma_{xx}(z) = \sum_{m=-\infty}^{\infty} v(m) z^{-m} \tag{12.2.2}$$

where the $\{v(m)\}$ are the coefficients in the series expansion. We can view $\{v(m)\}$ as the sequence with z-transform $V(z) = \log \Gamma_{xx}(z)$. Equivalently, we can evaluate $\log \Gamma_{xx}(z)$ on the unit circle,

$$\log \Gamma_{xx}(f) = \sum_{m=-\infty}^{\infty} v(m) e^{-j2\pi f m} \tag{12.2.3}$$

so that the $\{v(m)\}$ are the Fourier coefficients in the Fourier series expansion of the periodic function $\log \Gamma_{xx}(f)$. Hence

$$v(m) = \int_{-\frac{1}{2}}^{\frac{1}{2}} [\log \Gamma_{xx}(f)] e^{j2\pi f m} \, df, \qquad m = 0, \pm 1, \ldots \tag{12.2.4}$$

We observe that $v(m) = v(-m)$, since $\Gamma_{xx}(f)$ is a real and even function of f.

From (12.2.2) it follows that

$$\begin{aligned} \Gamma_{xx}(z) &= \exp\left[\sum_{m=-\infty}^{\infty} v(m) z^{-m}\right] \\ &= \sigma_w^2 H(z) H(z^{-1}) \end{aligned} \tag{12.2.5}$$

where, by definition, $\sigma_w^2 = \exp[v(0)]$ and

$$H(z) = \exp\left[\sum_{m=1}^{\infty} v(m)z^{-m}\right], \qquad |z| > r_1 \tag{12.2.6}$$

If (12.2.5) is evaluated on the unit circle, we have the equivalent representation of the power spectral density as

$$\Gamma_{xx}(f) = \sigma_w^2 |H(f)|^2 \tag{12.2.7}$$

We note that

$$\begin{aligned}\log \Gamma_{xx}(f) &= \log \sigma_w^2 + \log H(f) + \log H^*(f) \\ &= \sum_{m=-\infty}^{\infty} v(m)e^{-j2\pi fm}\end{aligned}$$

From the definition of $H(z)$ given by (12.2.6), it is clear that the causal part of the Fourier series in (12.2.3) is associated with $H(z)$ and the anticausal part is associated with $H(z^{-1})$. The Fourier series coefficients $\{v(m)\}$ are the *cepstral coefficients* and the sequence $\{v(m)\}$ is called the cepstrum of the sequence $\{\gamma_{xx}(m)\}$, as defined in Section 4.2.7.

The filter with system function $H(z)$ given by (12.2.6) is analytic in the region $|z| > r_1 < 1$. Hence, in this region, it has a Taylor series expansion as a causal system of the form

$$H(z) = \sum_{m=0}^{\infty} h(n)z^{-n} \tag{12.2.8}$$

The output of this filter in response to a white noise input sequence $w(n)$ with power spectral density σ_w^2 is a stationary random process $\{x(n)\}$ with power spectral density $\Gamma_{xx}(f) = \sigma_w^2|H(f)|^2$. Conversely, the stationary random process $\{x(n)\}$ with power spectral density $\Gamma_{xx}(f)$ can be transformed into a white noise process by passing $\{x(n)\}$ through a linear filter with system function $1/H(z)$. We call this filter a noise *whitening filter*. Its output, denoted as $\{w(n)\}$ is called the *innovations process* associated with the stationary random process $\{x(n)\}$. These two relationships are illustrated in Fig. 12.2.1.

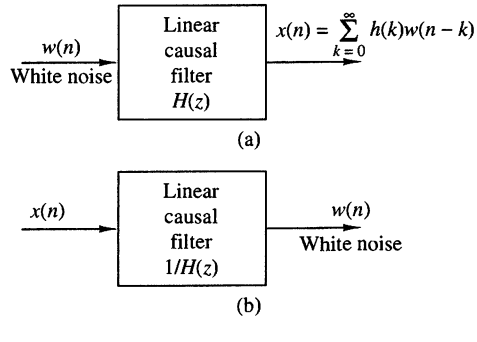


Figure 12.2.1
Filters for generating (a) the random process $x(n)$ from white noise and (b) the inverse filter.

The representation of the stationary stochastic process $\{x(n)\}$ as the output of an IIR filter with system function $H(z)$ given by (12.2.8) and excited by a white noise sequence $\{w(n)\}$ is called the *Wold representation.*

12.2.1 Rational Power Spectra

Let us now restrict our attention to the case where the power spectral density of the stationary random process $\{x(n)\}$ is a rational function, expressed as

$$\Gamma_{xx}(z) = \sigma_w^2 \frac{B(z)B(z^{-1})}{A(z)A(z^{-1})}, \qquad r_1 < |z| < r_2 \tag{12.2.9}$$

where the polynomials $B(z)$ and $A(z)$ have roots that fall inside the unit circle in the z-plane. Then the linear filter $H(z)$ for generating the random process $\{x(n)\}$ from the white noise sequence $\{w(n)\}$ is also rational and is expressed as

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \qquad |z| > r_1 \tag{12.2.10}$$

where $\{b_k\}$ and $\{a_k\}$ are the filter coefficients that determine the location of the zeros and poles of $H(z)$, respectively. Thus $H(z)$ is causal, stable, and minimum phase. Its reciprocal $1/H(z)$ is also a causal, stable, and minimum-phase linear system. Therefore, the random process $\{x(n)\}$ uniquely represents the statistical properties of the innovations process $\{w(n)\}$, and vice versa.

For the linear system with the rational system function $H(z)$ given by (12.2.10), the output $x(n)$ is related to the input $w(n)$ by the difference equation

$$x(n) + \sum_{k=1}^{p} a_k x(n-k) = \sum_{k=0}^{q} b_k w(n-k) \tag{12.2.11}$$

We will distinguish among three specific cases.

Autoregressive (AR) process. $b_0 = 1$, $b_k = 0$, $k > 0$. In this case, the linear filter $H(z) = 1/A(z)$ is an all-pole filter and the difference equation for the input–output relationship is

$$x(n) + \sum_{k=1}^{p} a_k x(n-k) = w(n) \tag{12.2.12}$$

In turn, the noise-whitening filter for generating the innovations process is an all-zero filter.

Moving average (MA) process. $a_k = 0$, $k \geq 1$. In this case, the linear filter $H(z) = B(z)$ is an all-zero filter and the difference equation for the input–output relationship is

$$x(n) = \sum_{k=0}^{q} b_k w(n-k) \tag{12.2.13}$$

The noise-whitening filter for the MA process is an all-pole filter.

Autoregressive, moving average (ARMA) process. In this case, the linear filter $H(z) = B(z)/A(z)$ has both finite poles and zeros in the z-plane and the corresponding difference equation is given by (12.2.11). The inverse system for generating the innovations process from $x(n)$ is also a pole–zero system of the form $1/H(z) = A(z)/B(z)$.

12.2.2 Relationships Between the Filter Parameters and the Autocorrelation Sequence

When the power spectral density of the stationary random process is a rational function, there is a basic relationship between the autocorrelation sequence $\{\gamma_{xx}(m)\}$ and the parameters $\{a_k\}$ and $\{b_k\}$ of the linear filter $H(z)$ that generates the process by filtering the white noise sequence $w(n)$. This relationship can be obtained by multiplying the difference equation in (12.2.11) by $x^*(n-m)$ and taking the expected value of both sides of the resulting equation. Thus we have

$$\begin{aligned} E[x(n)x^*(n-m)] &= -\sum_{k=1}^{p} a_k E[x(n-k)x^*(n-m)] \\ &\quad + \sum_{k=0}^{q} b_k E[w(n-k)x^*(n-m)] \end{aligned} \tag{12.2.14}$$

Hence

$$\gamma_{xx}(m) = -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k) + \sum_{k=0}^{q} b_k \gamma_{wx}(m-k) \tag{12.2.15}$$

where $\gamma_{wx}(m)$ is the crosscorrelation sequence between $w(n)$ and $x(n)$.

The crosscorrelation $\gamma_{wx}(m)$ is related to the filter impulse response. That is,

$$\begin{aligned} \gamma_{wx}(m) &= E[x^*(n)w(n+m)] \\ &= E\left[\sum_{k=0}^{\infty} h(k)w^*(n-k)w(n+m)\right] \\ &= \sigma_w^2 h(-m) \end{aligned} \tag{12.2.16}$$

where, in the last step, we have used the fact that the sequence $w(n)$ is white. Hence

$$\gamma_{wx}(m) = \begin{cases} 0, & m > 0 \\ \sigma_w^2 h(-m), & m \leq 0 \end{cases} \tag{12.2.17}$$

By combining (12.2.17) with (12.2.15), we obtain the desired relationship:

$$\gamma_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k), & m > q \\ -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k) + \sigma_w^2 \sum_{k=0}^{q-m} h(k) b_{k+m}, & 0 \le m \le q \\ \gamma_{xx}^*(-m), & m < 0 \end{cases} \tag{12.2.18}$$

This represents a nonlinear relationship between $\gamma_{xx}(m)$ and the parameters $\{a_k\}$, $\{b_k\}$.

The relationship in (12.2.18) applies, in general, to the ARMA process. For an AR process, (12.2.18) simplifies to

$$\gamma_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k), & m > 0 \\ -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k) + \sigma_w^2, & m = 0 \\ \gamma_{xx}^*(-m), & m < 0 \end{cases} \tag{12.2.19}$$

Thus we have a linear relationship between $\gamma_{xx}(m)$ and the $\{a_k\}$ parameters. These equations, called the *Yule–Walker* equations, can be expressed in the matrix form

$$\begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \cdots & \gamma_{xx}(-p) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \cdots & \gamma_{xx}(-p+1) \\ \vdots & \vdots & \vdots & & \vdots \\ \gamma_{xx}(p) & \gamma_{xx}(p-1) & \gamma_{xx}(p-2) & \cdots & \gamma_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{12.2.20}$$

This correlation matrix is Toeplitz, and hence it can be efficiently inverted by use of the algorithms described in Section 12.4.

Finally, by setting $a_k = 0$, $1 \le k \le p$, and $h(k) = b_k$, $0 \le k \le q$, in (12.2.18), we obtain the relationship for the autocorrelation sequence in the case of a MA process, namely,

$$\gamma_{xx}(m) = \begin{cases} \sigma_w^2 \sum_{k=0}^{q} b_k b_{k+m}, & 0 \le m \le q \\ 0, & m > q \\ \gamma_{xx}^*(-m), & m < 0 \end{cases} \tag{12.2.21}$$

12.3 Forward and Backward Linear Prediction

Linear prediction is an important topic in digital signal processing with many practical applications. In this section we consider the problem of linearly predicting the value of a stationary random process either forward in time or backward in time. This formulation leads to lattice filter structures and to some interesting connections to parametric signal models.

12.3.1 Forward Linear Prediction

Let us begin with the problem of predicting a future value of a stationary random process from observation of past values of the process. In particular, we consider the *one-step forward linear predictor*, which forms the prediction of the value $x(n)$ by a weighted linear combination of the past values $x(n-1), x(n-2), \ldots, x(n-p)$. Hence the linearly predicted value of $x(n)$ is

$$\hat{x}(n) = -\sum_{k=1}^{p} a_p(k)x(n-k) \tag{12.3.1}$$

where the $\{-a_p(k)\}$ represent the weights in the linear combination. These weights are called the *prediction coefficients* of the one-step forward linear predictor of *order* p. The negative sign in the definition of $x(n)$ is for mathematical convenience and conforms with current practice in the technical literature.

The difference between the value $x(n)$ and the predicted value $x(n)$ is called the *forward prediction error*, denoted as $f_p(n)$:

$$\begin{aligned} f_p(n) &= x(n) - \hat{x}(n) \\ &= x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \end{aligned} \tag{12.3.2}$$

We view linear prediction as equivalent to linear filtering where the predictor is embedded in the linear filter, as shown in Fig. 12.3.1. This is called a *prediction-error filter* with input sequence $\{x(n)\}$ and output sequence $\{f_p(n)\}$. An equivalent realization for the prediction-error filter is shown in Fig. 12.3.2. This realization is a direct-form FIR filter with system function

$$A_p(z) = \sum_{k=0}^{p} a_p(k)z^{-k} \tag{12.3.3}$$

where, by definition, $a_p(0) = 1$.

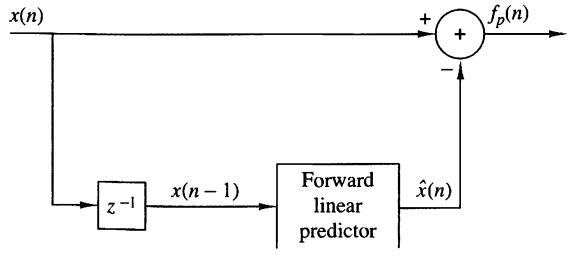


Figure 12.3.1
Forward linear prediction.

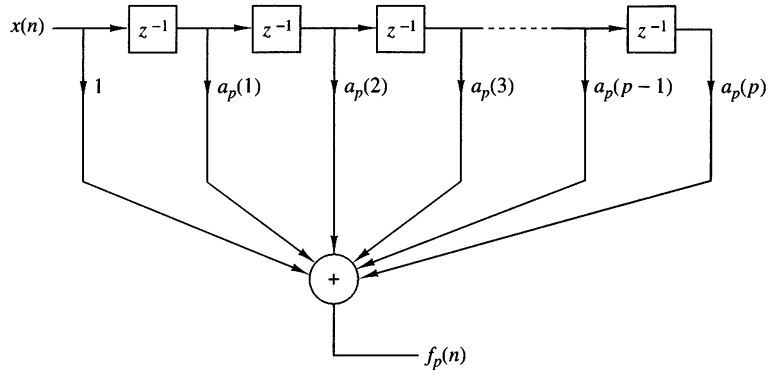


Figure 12.3.2 Prediction-error filter.

As shown in Section 9.2.4, the direct-form FIR filter is equivalent to an all-zero lattice filter. The lattice filter is generally described by the following set of *order-recursive equations*:

$$\begin{aligned} f_0(n) &= g_0(n) = x(n) \\ f_m(n) &= f_{m-1}(n) + K_m g_{m-1}(n-1) \qquad m = 1, 2, \ldots, p \\ g_m(n) &= K_m^* f_{m-1}(n) + g_{m-1}(n-1) \qquad m = 1, 2, \ldots, p \end{aligned} \tag{12.3.4}$$

where $\{K_m\}$ are the reflection coefficients and $g_m(n)$ is the backward prediction error defined in the following section. Note that for complex-valued data, the conjugate of K_m is used in the equation for $g_m(n)$. Figure 12.3.3 illustrates a p-stage lattice filter in block diagram form along with a typical stage showing the computations given by (12.3.4).

As a consequence of the equivalence between the direct-form prediction-error FIR filter and the FIR lattice filter, the output of the p-stage lattice filter is expressed

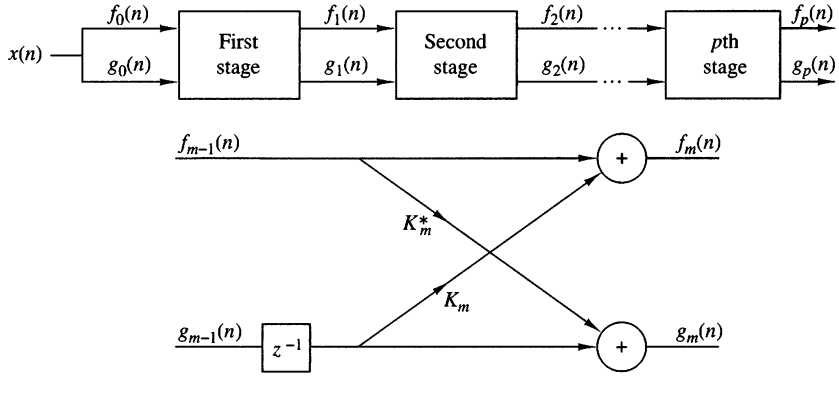


Figure 12.3.3 p-stage lattice filter.

as

$$f_p(n) = \sum_{k=0}^{p} a_p(k)x(n-k), \qquad a_p(0) = 1 \tag{12.3.5}$$

Since (12.3.5) is a convolution sum, the z-transform relationship is

$$F_p(z) = A_p(z)X(z) \tag{12.3.6}$$

or, equivalently,

$$A_p(z) = \frac{F_p(z)}{X(z)} = \frac{F_p(z)}{F_0(z)} \tag{12.3.7}$$

The mean-square value of the forward linear prediction error $f_p(n)$ is

$$\begin{aligned} \mathcal{E}_p^f &= E[|f_p(n)|^2] \\ &= \gamma_{xx}(0) + 2\Re\left[\sum_{k=1}^{p} a_p^*(k)\gamma_{xx}(k)\right] + \sum_{k=1}^{p}\sum_{l=1}^{p} a_p^*(l)a_p(k)\gamma_{xx}(l-k) \end{aligned} \tag{12.3.8}$$

$\mathcal{E}_p^f$ is a quadratic function of the predictor coefficients and its minimization leads to the set of linear equations

$$\gamma_{xx}(l) = -\sum_{k=1}^{p} a_p(k)\gamma_{xx}(l-k), \qquad l = 1, 2, \ldots, p \tag{12.3.9}$$

These are called the *normal equations* for the coefficients of the linear predictor. The minimum mean-square prediction error is simply

$$\min[\mathcal{E}_p^f] \equiv E_p^f = \gamma_{xx}(0) + \sum_{k=1}^{p} a_p(k)\gamma_{xx}(-k) \tag{12.3.10}$$

In the following section we extend the development above to the problem of predicting the value of a time series in the opposite direction, namely, backward in time.

12.3.2 Backward Linear Prediction

Let us assume that we have the data sequence $x(n), x(n-1), \ldots, x(n-p+1)$ from a stationary random process and we wish to predict the value $x(n-p)$ of the process. In this case we employ a *one-step backward linear predictor* of order p. Hence

$$\hat{x}(n-p) = -\sum_{k=0}^{p-1} b_p(k)x(n-k) \tag{12.3.11}$$

The difference between the value $x(n-p)$ and the estimate $\hat{x}(n-p)$ is called the *backward prediction error*, denoted as $g_p(n)$:

$$\begin{aligned} g_p(n) &= x(n-p) + \sum_{k=0}^{p-1} b_p(k)x(n-k) \\ &= \sum_{k=0}^{p} b_p(k)x(n-k), \qquad b_p(p) = 1 \end{aligned} \tag{12.3.12}$$

The backward linear predictor can be realized either by a direct-form FIR filter structure similar to the structure shown in Fig. 12.3.1 or as a lattice structure. The lattice structure shown in Fig. 12.3.3 provides the backward linear predictor as well as the forward linear predictor.

The weighting coefficients in the backward linear predictor are the complex conjugates of the coefficients for the forward linear predictor, but they occur in reverse order. Thus we have

$$b_p(k) = a_p^*(p-k), \qquad k = 0, 1, \ldots, p \tag{12.3.13}$$

In the z-domain, the convolution sum in (12.3.12) becomes

$$G_p(z) = B_p(z)X(z) \tag{12.3.14}$$

or, equivalently,

$$B_p(z) = \frac{G_p(z)}{X(z)} = \frac{G_p(z)}{G_0(z)} \tag{12.3.15}$$

where $B_p(z)$ represents the system function of the FIR filter with coefficients $b_p(k)$.

Since $b_p(k) = a^*(p-k)$, $G_p(z)$ is related to $A_p(z)$

$$\begin{aligned} B_p(z) &= \sum_{k=0}^{p} b_p(k)z^{-k} \\ &= \sum_{k=0}^{p} a_p^*(p-k)z^{-k} \\ &= z^{-p}\sum_{k=0}^{p} a_p^*(k)z^{k} \\ &= z^{-p}A_p^*(z^{-1}) \end{aligned} \tag{12.3.16}$$

The relationship in (12.3.16) implies that the zeros of the FIR filter with system function $B_p(z)$ are simply the (conjugate) reciprocals of the zeros of $A_p(z)$. Hence $B_p(z)$ is called the reciprocal or *reverse polynomial* of $A_p(z)$.

Now that we have established these interesting relationships between the forward and backward one-step predictors, let us return to the recursive lattice equations in (12.3.4) and transform them to the z-domain. Thus we have

$$\begin{aligned} F_0(z) &= G_0(z) = X(z) \\ F_m(z) &= F_{m-1}(z) + K_m z^{-1} G_{m-1}(z), \qquad m = 1, 2, \ldots, p \\ G_m(z) &= K_m^* F_{m-1}(z) + z^{-1} G_{m-1}(z), \qquad m = 1, 2, \ldots, p \end{aligned} \tag{12.3.17}$$

If we divide each equation by $X(z)$, we obtain the desired results in the form

$$\begin{aligned} A_0(z) &= B_0(z) = 1 \\ A_m(z) &= A_{m-1}(z) + K_m z^{-1} B_{m-1}(z), \qquad m = 1, 2, \ldots, p \\ B_m(z) &= K_m^* A_{m-1}(z) + z^{-1} B_{m-1}(z), \qquad m = 1, 2, \ldots, p \end{aligned} \tag{12.3.18}$$

Thus a lattice filter is described in the z-domain by the matrix equation

$$\begin{bmatrix} A_m(z) \\ B_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_m z^{-1} \\ K_m^* & z^{-1} \end{bmatrix} \begin{bmatrix} A_{m-1}(z) \\ B_{m-1}(z) \end{bmatrix} \tag{12.3.19}$$

The relations in (12.3.18) for $A_m(z)$ and $B_m(z)$ allow us to obtain the direct-form FIR filter coefficients $\{a_m(k)\}$ from the reflection coefficients $\{K_m\}$, and vice versa. These relationships were given in Section 9.2.4 by (9.2.51) through (9.2.53).

The lattice structure with parameters $K_1, K_2, \ldots, K_p$ corresponds to a class of p direct-form FIR filters with system functions $A_1(z), A_2(z), \ldots, A_p(z)$. It is interesting to note that a characterization of this class of p FIR filters in direct form requires $p(p+1)/2$ filter coefficients. In contrast, the lattice-form characterization requires only the p reflection coefficients $\{K_i\}$. The reason the lattice provides a more compact representation for the class of p FIR filters is because appending stages to the lattice does not alter the parameters of the previous stages. On the other hand, appending the pth stage to a lattice with $(p-1)$ stages is equivalent to increasing the length of an FIR filter by one coefficient. The resulting FIR filter with system function $A_p(z)$ has coefficients totally different from the coefficients of the lower-order FIR filter with system function $A_{p-1}(z)$.

The formula for determining the filter coefficients $\{a_p(k)\}$ recursively is easily derived from polynomial relationships (12.3.18). We have

$$\begin{aligned} A_m(z) &= A_{m-1}(z) + K_m z^{-1} B_{m-1}(z) \\ \sum_{k=0}^{m} a_m(k) z^{-k} &= \sum_{k=0}^{m-1} a_{m-1}(k) z^{-k} + K_m \sum_{k=0}^{m-1} a_{m-1}^*(m-1-k) z^{-(k+1)} \end{aligned} \tag{12.3.20}$$

By equating the coefficients of equal powers of z^{-1} and recalling that $a_m(0) = 1$ for $m = 1, 2, \ldots, p$, we obtain the desired recursive equation for the FIR filter

coefficients in the form

$$\begin{aligned} a_m(0) &= 1 \\ a_m(m) &= K_m \\ a_m(k) &= a_{m-1}(k) + K_m a_{m-1}^*(m-k) \\ &= a_{m-1}(k) + a_m(m) a_{m-1}^*(m-k), \quad \begin{matrix} 1 \le k \le m-1 \\ m = 1, 2, \ldots, p \end{matrix} \end{aligned} \tag{12.3.21}$$

The conversion formula from the direct-form FIR filter coefficients $\{a_p(k)\}$ to the lattice reflection coefficients $\{K_i\}$ is also very simple. For the p-stage lattice we immediately obtain the reflection coefficient $K_p = a_p(p)$. To obtain $K_{p-1}, \ldots, K_1$, we need the polynomials $A_m(z)$ for $m = p-1, \ldots, 1$. From (12.3.19) we obtain

$$A_{m-1}(z) = \frac{A_m(z) - K_m B_m(z)}{1 - |K_m|^2}, \qquad m = p, \ldots, 1 \tag{12.3.22}$$

which is just a step-down recursion. Thus we compute all lower-degree polynomials $A_m(z)$ beginning with $A_{p-1}(z)$ and obtain the desired lattice reflection coefficients from the relation $K_m = a_m(m)$. We observe that the procedure works as long as $|K_m| \neq 1$ for $m = 1, 2, \ldots, p-1$. From this step-down recursion for the polynomials, it is relatively easy to obtain a formula for recursively and directly computing K_m, $m = p-1, \ldots, 1$. For $m = p-1, \ldots, 1$, we have

$$\begin{aligned} K_m &= a_m(m) \\ a_{m-1}(k) &= \frac{a_m(k) - K_m b_m(k)}{1 - |K_m|^2} \\ &= \frac{a_m(k) - a_m(m) a_m^*(m-k)}{1 - |a_m(m)|^2} \end{aligned} \tag{12.3.23}$$

which is just the recursion in the Schur–Cohn stability test for the polynomial $A_m(z)$.

As just indicated, the recursive equation in (12.3.23) breaks down if any of the lattice parameters $|K_m| = 1$. If this occurs, it is indicative that the polynomial $A_{m-1}(z)$ has a root located on the unit circle. Such a root can be factored out from $A_{m-1}(z)$ and the iterative process in (12.3.23) carried out for the reduced-order system.

Finally, let us consider the minimization of the mean-square error in a backward linear predictor. The backward prediction error is

$$\begin{aligned} g_p(n) &= x(n-p) + \sum_{k=0}^{p-1} b_p(k) x(n-k) \\ &= x(n-p) + \sum_{k=1}^{p} a_p^*(k) x(n-p+k) \end{aligned} \tag{12.3.24}$$

and its mean-square value is

$$\mathcal{E}_p^b = E[|g_p(n)|^2] \tag{12.3.25}$$

The minimization of $\mathcal{E}_p^b$ with respect to the prediction coefficients yields the same set of linear equations as in (12.3.9). Hence the minimum mean-square error (MSE) is

$$\min[\mathcal{E}_p^b] \equiv E_p^b = E_p^f \tag{12.3.26}$$

which is given by (12.3.10).

12.3.3 The Optimum Reflection Coefficients for the Lattice Forward and Backward Predictors

In Sections 12.3.1 and 12.3.2 we derived the set of linear equations which provide the predictor coefficients that minimize the mean-square value of the prediction error. In this section we consider the problem of optimizing the reflection coefficients in the lattice predictor and expressing the reflection coefficients in terms of the forward and backward prediction errors.

The forward prediction error in the lattice filter is expressed as

$$f_m(n) = f_{m-1}(n) + K_m g_{m-1}(n-1) \tag{12.3.27}$$

The minimization of $E[|f_m(n)|^2]$ with respect to the reflection coefficient K_m yields the result

$$K_m = \frac{-E[f_{m-1}(n)g_{m-1}^*(n-1)]}{E[|g_{m-1}(n-1)|^2]} \tag{12.3.28}$$

or, equivalently,

$$K_m = \frac{-E[f_{m-1}(n)g_{m-1}^*(n-1)]}{\sqrt{E_{m-1}^f E_{m-1}^b}} \tag{12.3.29}$$

where $E_{m-1}^f = E_{m-1}^b = E[|g_{m-1}(n-1)|^2] = E[|f_{m-1}(n)|^2]$.

We observe that the optimum choice of the reflection coefficients in the lattice predictor is the negative of the (normalized) crosscorrelation coefficients between the forward and backward errors in the lattice.[1] Since it is apparent from (12.3.28) that $|K_m| \leq 1$, it follows that the minimum mean-square value of the prediction error, which can be expressed recursively as

$$E_m^f = (1 - |K_m|^2)E_{m-1}^f \tag{12.3.30}$$

is a monotonically decreasing sequence.

[1] The normalized crosscorrelation coefficients between the forward and backward error in the lattice (i.e., $\{-K_m\}$) are often called the partial correlation (PARCOR) coefficients.

12.3.4 Relationship of an AR Process to Linear Prediction

The parameters of an AR(p) process are intimately related to a predictor of order p for the same process. To see the relationship, we recall that in an AR(p) process, the autocorrelation sequence $\{\gamma_{xx}(m)\}$ is related to the parameters $\{a_k\}$ by the Yule–Walker equations given in (12.2.19) or (12.2.20). The corresponding equations for the predictor of order p are given by (12.3.9) and (12.3.10).

A direct comparison of these two sets of relations reveals that there is a one-to-one correspondence between the parameters $\{a_k\}$ of the AR(p) process and the predictor coefficients $\{a_p(k)\}$ of the pth-order predictor. In fact, if the underlying process $\{x(n)\}$ is AR(p), the prediction coefficients of the pth-order predictor are identical to $\{a_k\}$. Furthermore, the minimum mean-square error (MMSE) in the pth-order predictor E_p^f is identical to σ_w^2, the variance of the white noise process. In this case, the prediction-error filter is a noise-whitening filter which produces the innovations sequence $\{w(n)\}$.

12.4 Solution of the Normal Equations

In the preceding section we observed that the minimization of the mean-square value of the forward prediction error resulted in a set of linear equations for the coefficients of the predictor given by (12.3.9). These equations, called the normal equations, may be expressed in the compact form

$$\begin{aligned} &\sum_{k=0}^{p} a_p(k)\gamma_{xx}(l-k) = 0 \qquad l = 1, 2, \ldots, p \\ &\qquad\qquad\qquad\qquad\qquad\qquad a_p(0) = 1 \end{aligned} \tag{12.4.1}$$

The resulting minimum MSE (MMSE) is given by (12.3.10). If we augment (12.3.10) to the normal equations given by (12.4.1) we obtain the set of *augmented normal equations*, which may be expressed as

$$\sum_{k=0}^{p} a_p(k)\gamma_{xx}(l-k) = \begin{cases} E_p^f, & l = 0 \\ 0, & l = 1, 2, \ldots, p \end{cases} \tag{12.4.2}$$

We also noted that if the random process is an AR(p) process, the MMSE $E_p^f = \sigma_w^2$.

In this section we describe two computationally efficient algorithms for solving the normal equations. One algorithm, originally due to Levinson (1947) and modified by Durbin (1959), is called the Levinson–Durbin algorithm. This algorithm is suitable for serial processing and has a computation complexity of $O(p^2)$. The second algorithm, due to Schur (1917), also computes the reflection coefficients in $O(p^2)$ operations but with parallel processors, the computations can be performed in $O(p)$ time. Both algorithms exploit the Toeplitz symmetry property inherent in the autocorrelation matrix.

We begin by describing the Levinson–Durbin algorithm.

12.4.1 The Levinson–Durbin Algorithm

The Levinson-Durbin algorithm is a computationally efficient algorithm for solving the normal equations in (12.4.1) for the prediction coefficients. This algorithm exploits the special symmetry in the autocorrelation matrix

$$\mathbf{\Gamma}_p = \begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}^*(1) & \cdots & \gamma_{xx}^*(p-1) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \cdots & \gamma_{xx}^*(p-2) \\ \vdots & \vdots & & \\ \gamma_{xx}(p-1) & \gamma_{xx}(p-2) & \cdots & \gamma_{xx}(0) \end{bmatrix} \tag{12.4.3}$$

Note that $\Gamma_p(i, j) = \Gamma_p(i-j)$, so that the autocorrelation matrix is a *Toeplitz matrix*. Since $\Gamma_p(i, j) = \mathbf{\Gamma}_p^*(j, i)$, the matrix is also Hermitian.

The key to the Levinson–Durbin method of solution, which exploits the Toeplitz property of the matrix, is to proceed recursively, beginning with a predictor of order $m = 1$ (one coefficient), and then to increase the order recursively, using the lower-order solutions to obtain the solution to the next-higher order. Thus the solution to the first-order predictor obtained by solving (12.4.1) is

$$a_1(1) = -\frac{\gamma_{xx}(1)}{\gamma_{xx}(0)} \tag{12.4.4}$$

and the resulting MMSE is

$$\begin{aligned} E_1^f &= \gamma_{xx}(0) + a_1(1)\gamma_{xx}(-1) \\ &= \gamma_{xx}(0)[1 - |a_1(1)|^2] \end{aligned} \tag{12.4.5}$$

Recall that $a_1(1) = K_1$, the first reflection coefficient in the lattice filter.

The next step is to solve for the coefficients $\{a_2(1), a_2(2)\}$ of the second-order predictor and express the solution in terms of $a_1(1)$. The two equations obtained from (12.4.1) are

$$\begin{aligned} a_2(1)\gamma_{xx}(0) + a_2(2)\gamma_{xx}^*(1) &= -\gamma_{xx}(1) \\ a_2(1)\gamma_{xx}(1) + a_2(2)\gamma_{xx}(0) &= -\gamma_{xx}(2) \end{aligned} \tag{12.4.6}$$

By using the solution in (12.4.4) to eliminate $\gamma_{xx}(1)$, we obtain the solution

$$\begin{aligned} a_2(2) &= -\frac{\gamma_{xx}(2) + a_1(1)\gamma_{xx}(1)}{\gamma_{xx}(0)[1 - |a_1(1)|^2]} \\ &= -\frac{\gamma_{xx}(2) + a_1(1)\gamma_{xx}(1)}{E_1^f} \\ a_2(1) &= a_1(1) + a_2(2)a_1^*(1) \end{aligned} \tag{12.4.7}$$

Thus we have obtained the coefficients of the second-order predictor. Again, we note that $a_2(2) = K_2$, the second reflection coefficient in the lattice filter.

Proceeding in this manner, we can express the coefficients of the mth-order predictor in terms of the coefficients of the $(m-1)$st-order predictor. Thus we can write the coefficient vector $\mathbf{a}_m$ as the sum of two vectors, namely,

$$\mathbf{a}_m = \begin{bmatrix} a_m(1) \\ a_m(2) \\ \vdots \\ a_m(m) \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{m-1} \\ \cdots \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{d}_{m-1} \\ \cdots \\ K_m \end{bmatrix} \tag{12.4.8}$$

where $\mathbf{a}_{m-1}$ is the predictor coefficient vector of the $(m-1)$st-order predictor and the vector $\mathbf{d}_{m-1}$ and the scalar K_m are to be determined. Let us also partition the $m \times m$ autocorrelation matrix Γ_{xx} as

$$\mathbf{\Gamma}_m = \begin{bmatrix} \mathbf{\Gamma}_{m-1} & \boldsymbol{\gamma}_{m-1}^{b*} \\ \boldsymbol{\gamma}_{m-1}^{bt} & \gamma_{xx}(0) \end{bmatrix} \tag{12.4.9}$$

where $\boldsymbol{\gamma}_{m-1}^{bt} = [\gamma_{xx}(m-1) \quad \gamma_{xx}(m-2) \quad \cdots \quad \gamma_{xx}(1)] = (\boldsymbol{\gamma}_{m-1}^{b})^t$, the asterisk (*) denotes the complex conjugate, and $\boldsymbol{\gamma}_m^t$ denotes the transpose of $\boldsymbol{\gamma}_m$. The superscript b on $\boldsymbol{\gamma}_{m-1}$ denotes the vector $\boldsymbol{\gamma}_{m-1}^t = [\gamma_{xx}(1) \quad \gamma_{xx}(2) \quad \cdots \quad \gamma_{xx}(m-1)]$ with elements taken in reverse order.

The solution to the equation $\mathbf{\Gamma}_m \mathbf{a}_m = -\boldsymbol{\gamma}_m$ can be expressed as

$$\begin{bmatrix} \mathbf{\Gamma}_{m-1} & \boldsymbol{\gamma}_{m-1}^{b*} \\ \boldsymbol{\gamma}_{m-1}^{bt} & \gamma_{xx}(0) \end{bmatrix} \left\{ \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{d}_{m-1} \\ K_m \end{bmatrix} \right\} = - \begin{bmatrix} \boldsymbol{\gamma}_{m-1} \\ \gamma_{xx}(m) \end{bmatrix} \tag{12.4.10}$$

This is the key step in the Levinson–Durbin algorithm. From (12.4.10) we obtain two equations, namely,

$$\mathbf{\Gamma}_{m-1}\mathbf{a}_{m-1} + \mathbf{\Gamma}_{m-1}\mathbf{d}_{m-1} + K_m \boldsymbol{\gamma}_{m-1}^{b*} = -\boldsymbol{\gamma}_{m-1} \tag{12.4.11}$$

$$\boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1} + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{d}_{m-1} + K_m \gamma_{xx}(0) = -\gamma_{xx}(m) \tag{12.4.12}$$

Since $\mathbf{\Gamma}_{m-1}\mathbf{a}_{m-1} = -\boldsymbol{\gamma}_{m-1}$, (12.4.11) yields the solution

$$\mathbf{d}_{m-1} = -K_m \mathbf{\Gamma}_{m-1}^{-1} \boldsymbol{\gamma}_{m-1}^{b*} \tag{12.4.13}$$

But $\boldsymbol{\gamma}_{m-1}^{b*}$ is just $\boldsymbol{\gamma}_{m-1}$ with elements taken in reverse order and conjugated. Therefore, the solution in (12.4.13) is simply

$$\mathbf{d}_{m-1} = K_m \mathbf{a}_{m-1}^{b*} = K_m \begin{bmatrix} a_{m-1}^*(m-1) \\ a_{m-1}^*(m-2) \\ \vdots \\ a_{m-1}^*(1) \end{bmatrix} \tag{12.4.14}$$

The scalar equation (12.4.12) can now be used to solve for K_m. If we eliminate $\mathbf{d}_{m-1}$ in (12.4.12) by using (12.4.14), we obtain

$$K_m[\gamma_{xx}(0) + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1}^{b*}] + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1} = -\gamma_{xx}(m)$$

Hence

$$K_m = -\frac{\gamma_{xx}(m) + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1}}{\gamma_{xx}(0) + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1}^{b*}} \tag{12.4.15}$$

Therefore, by substituting the solutions in (12.4.14) and (12.4.15) into (12.4.8), we obtain the desired recursion for the predictor coefficients in the Levinson–Durbin algorithm as

$$a_m(m) = K_m = -\frac{\gamma_{xx}(m) + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1}}{\gamma_{xx}(0) + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1}^{b*}} = -\frac{\gamma_{xx}(m) + \boldsymbol{\gamma}_{m-1}^{bt}\mathbf{a}_{m-1}}{E_{m-1}^{f}} \tag{12.4.16}$$

$$\begin{aligned} a_m(k) &= a_{m-1}(k) + K_m a_{m-1}^*(m-k) \\ &= a_{m-1}(k) + a_m(m)a_{m-1}^*(m-k), \qquad \begin{matrix} k = 1, 2, \ldots, m-1 \\ m = 1, 2, \ldots, p \end{matrix} \end{aligned} \tag{12.4.17}$$

The reader should note that the recursive relation in (12.4.17) is identical to the recursive relation in (12.3.21) for the predictor coefficients, obtained from the polynomials $A_m(z)$ and $B_m(z)$. Furthermore, K_m is the reflection coefficient in the mth stage of the lattice predictor. This development clearly illustrates that the Levinson–Durbin algorithm produces the reflection coefficients for the optimum lattice prediction filter as well as the coefficients of the optimum direct-form FIR predictor.

Finally, let us determine the expression for the MMSE. For the mth-order predictor, we have

$$\begin{aligned} E_m^f &= \gamma_{xx}(0) + \sum_{k=1}^{m} a_m(k)\gamma_{xx}(-k) \\ &= \gamma_{xx}(0) + \sum_{k=1}^{m}[a_{m-1}(k) + a_m(m)a_{m-1}^*(m-k)]\gamma_{xx}(-k) \\ &= E_{m-1}^f[1 - |a_m(m)|^2] = E_{m-1}^f(1 - |K_m|^2), \qquad m = 1, 2, \ldots, p \end{aligned} \tag{12.4.18}$$

where $E_0^f = \gamma_{xx}(0)$. Since the reflection coefficients satisfy the property that $|K_m| \leq 1$, the MMSE for the sequence of predictors satisfies the condition

$$E_0^f \geq E_1^f \geq E_2^f \geq \cdots \geq E_p^f \tag{12.4.19}$$

This concludes the derivation of the Levinson–Durbin algorithm for solving the linear equations $\boldsymbol{\Gamma}_m\mathbf{a}_m = -\boldsymbol{\gamma}_m$, for $m = 0, 1, \ldots, p$. We observe that the linear equations have the special property that the vector on the right-hand side also appears as a vector in $\boldsymbol{\Gamma}_m$. In the more general case, where the vector on the right-hand side is some other vector, say $\mathbf{c}_m$, the set of linear equations can be solved recursively by introducing a second recursive equation to solve the more general linear

equations $\mathbf{\Gamma}_m \mathbf{b}_m = \mathbf{c}_m$. The result is a *generalized Levinson–Durbin* algorithm (see Problem 12.12).

The Levinson–Durbin recursion given by (12.4.17) requires $O(m)$ multiplications and additions (operations) to go from stage m to stage $m+1$. Therefore, for p stages it takes on the order of $1+2+3+\cdots+p(p+1)/2$, or $O(p^2)$, operations to solve for the prediction filter coefficients, or the reflection coefficients, compared with $O(p^3)$ operations if we did not exploit the Toeplitz property of the correlation matrix.

If the Levinson–Durbin algorithm is implemented on a serial computer or signal processor, the required computation time is on the order of $O(p^2)$ time units. On the other hand, if the processing is performed on a parallel processing machine utilizing as many processors as necessary to exploit the full parallelism in the algorithm, the multiplications as well as the additions required to compute (12.4.17) can be carried out simultaneously. Therefore, this computation can be performed in $O(p)$ time units. However, the computation in (12.4.16) for the reflection coefficients takes additional time. Certainly, the inner products involving the vectors $\mathbf{a}_{m-1}$ and $\boldsymbol{\gamma}_{m-1}^b$ can be computed simultaneously by employing parallel processors. However, the addition of these products cannot be done simultaneously, but instead, requires $O(\log p)$ time units. Hence, the computations in the Levinson–Durbin algorithm, when performed by p parallel processors, can be accomplished in $O(p \log p)$ time.

In the following section we describe another algorithm, due to Schur (1917), that avoids the computation of inner products, and therefore is more suitable for parallel computation of the reflection coefficients.

12.4.2 The Schur Algorithm

The Schur algorithm is intimately related to a recursive test for determining the positive definiteness of a correlation matrix. To be specific, let us consider the autocorrelation matrix $\mathbf{\Gamma}_{p+1}$ associated with the augmented normal equations given by (12.4.2). From the elements of this matrix we form the function

$$R_0(z) = \frac{\gamma_{xx}(1)z^{-1} + \gamma_{xx}(2)z^{-2} + \cdots + \gamma_{xx}(p)z^{-p}}{\gamma_{xx}(0) + \gamma_{xx}(1)z^{-1} + \cdots + \gamma_{xx}(p)z^{-p}} \tag{12.4.20}$$

and the sequence of functions $R_m(z)$ defined recursively as

$$R_m(z) = \frac{R_{m-1}(z) - R_{m-1}(\infty)}{z^{-1}[1 - R_{m-1}^*(\infty)R_{m-1}(z)]}, \qquad m = 1, 2, \ldots \tag{12.4.21}$$

Schur's theorem states that a necessary and sufficient condition for the correlation matrix to be positive definite is that $|R_m(\infty)| < 1$ for $m = 1, 2, \ldots, p$.

Let us demonstrate that the condition for positive definiteness of the autocorrelation matrix $\mathbf{\Gamma}_{p+1}$ is equivalent to the condition that the reflection coefficients in the equivalent lattice filter satisfy the condition $|K_m| < 1$, $m = 1, 2, \ldots, p$.

First, we note that $R_0(\infty) = 0$. Then, from (12.4.21) we have

$$R_1(z) = \frac{\gamma_{xx}(1) + \gamma_{xx}(2)z^{-1} + \cdots + \gamma_{xx}(p)z^{-p+1}}{\gamma_{xx}(0) + \gamma_{xx}(1)z^{-1} + \cdots + \gamma_{xx}(p)z^{-p}} \tag{12.4.22}$$

Hence $R_1(\infty) = \gamma_{xx}(1)/\gamma_{xx}(0)$. We observe that $R_1(\infty) = -K_1$.

Second, we compute $R_2(z)$ according to (12.4.21) and evaluate the result at $z = \infty$. Thus we obtain

$$R_2(\infty) = \frac{\gamma_{xx}(2) + K_1\gamma_{xx}(1)}{\gamma_{xx}(0)(1 - |K_1|^2)}$$

Again, we observe that $R_2(\infty) = -K_2$. By continuing this procedure, we find that $R_m(\infty) = -K_m$, for $m = 1, 2, \ldots, p$. Hence the condition $|R_m(\infty)| < 1$ for $m = 1, 2, \ldots, p$, is identical to the condition $|K_m| < 1$ for $m = 1, 2, \ldots, p$, and ensures the positive definiteness of the autocorrelation matrix $\mathbf{\Gamma}_{p+1}$.

Since the reflection coefficients can be obtained from the sequence of functions $R_m(z)$, $m = 1, 2, \ldots, p$, we have another method for solving the normal equations. We call this method the *Schur algorithm.*

Schur algorithm. Let us first rewrite $R_m(z)$ as

$$R_m(z) = \frac{P_m(z)}{Q_m(z)}, \qquad m = 0, 1, \ldots, p \tag{12.4.23}$$

where

$$\begin{aligned} P_0(z) &= \gamma_{xx}(1)z^{-1} + \gamma_{xx}(2)z^{-2} + \cdots + \gamma_{xx}(p)z^{-p} \\ Q_0(z) &= \gamma_{xx}(0) + \gamma_{xx}(1)z^{-1} + \cdots + \gamma_{xx}(p)z^{-p} \end{aligned} \tag{12.4.24}$$

Since $K_0 = 0$ and $K_m = -R_m(\infty)$ for $m = 1, 2, \ldots, p$, the recursive equation (12.4.21) implies the following recursive equations for the polynomials $P_m(z)$ and $Q_m(z)$:

$$\begin{bmatrix} P_m(z) \\ Q_m(z) \end{bmatrix} = \begin{bmatrix} 1 & K_{m-1} \\ K_{m-1}^* z^{-1} & z^{-1} \end{bmatrix} \begin{bmatrix} P_{m-1}(z) \\ Q_{m-1}(z) \end{bmatrix}, \qquad m = 1, 2, \ldots, p \tag{12.4.25}$$

Thus we have

$$\begin{aligned} P_1(z) &= P_0(z) = \gamma_{xx}(1)z^{-1} + \gamma_{xx}(2)z^{-2} + \cdots + \gamma_{xx}(p)z^{-p} \\ Q_1(z) &= z^{-1}Q_0(z) = \gamma_{xx}(0)z^{-1} + \gamma_{xx}(1)z^{-2} + \cdots + \gamma_{xx}(p)z^{-p-1} \end{aligned} \tag{12.4.26}$$

and

$$K_1 = -\left.\frac{P_1(z)}{Q_1(z)}\right|_{z=\infty} = -\frac{\gamma_{xx}(1)}{\gamma_{xx}(0)} \tag{12.4.27}$$

Next the reflection coefficient K_2 is obtained by determining $P_2(z)$ and $Q_2(z)$ from (12.4.25), dividing $P_2(z)$ by $Q_2(z)$ and evaluating the result at $z = \infty$. Thus we find

that

$$\begin{aligned} P_2(z) &= P_1(z) + K_1 Q_1(z) \\ &= [\gamma_{xx}(2) + K_1\gamma_{xx}(1)]z^{-2} + \cdots \\ &\quad + [\gamma_{xx}(p) + K_1\gamma_{xx}(p-1)]z^{-p} \\ Q_2(z) &= z^{-1}[Q_1(z) + K_1^* P_1(z)] \\ &= [\gamma_{xx}(0) + K_1^*\gamma_{xx}(1)]z^{-2} + \cdots \\ &\quad + [\gamma_{xx}(p-2) + K_1^*\gamma_{xx}(p-1)]z^{-p} \end{aligned} \tag{12.4.28}$$

where the terms involving z^{-p-1} have been dropped. Thus we observe that the recursive equation in (12.4.25) is equivalent to (12.4.21).

Based on these relationships, the Schur algorithm is described by the following recursive procedure.

Initialization. Form the $2x(p+1)$ generator matrix

$$\mathbf{G}_0 = \begin{bmatrix} 0 & \gamma_{xx}(1) & \gamma_{xx}(2) & \cdots & \gamma_{xx}(p) \\ \gamma_{xx}(0) & \gamma_{xx}(1) & \gamma_{xx}(2) & \cdots & \gamma_{xx}(p) \end{bmatrix} \tag{12.4.29}$$

where the elements of the first row are the coefficients of $P_0(z)$ and the elements of the second row are the coefficients of $Q_0(z)$.

Step 1. Shift the second row of the generator matrix to the right by one place and discard the last element of this row. A zero is placed in the vacant position. Thus we obtain a new generator matrix,

$$\mathbf{G}_1 = \begin{bmatrix} 0 & \gamma_{xx}(1) & \gamma_{xx}(1) & \gamma_{xx}(2) & \cdots & \gamma_{xx}(p) \\ 0 & \gamma_{xx}(0) & \gamma_{xx}(1) & \gamma_{xx}(1) & \cdots & \gamma_{xx}(p-1) \end{bmatrix} \tag{12.4.30}$$

The (negative) ratio of the elements in the second column yields the reflection coefficient $K_1 = -\gamma_{xx}(1)/\gamma_{xx}(0)$.

Step 2. Multiply the generator matrix by the 2×2 matrix

$$\mathbf{V}_1 = \begin{bmatrix} 1 & K_1 \\ K_1^* & 1 \end{bmatrix} \tag{12.4.31}$$

Thus we obtain

$$\mathbf{V}_1\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & \gamma_{xx}(2) + K_1\gamma_{xx}(1) & \gamma_{xx}(p) + K_1\gamma_{xx}(p-1) \\ 0 & \gamma_{xx}(0) + K_1^*\gamma_{xx}(1) & \cdots & \gamma_{xx}(p-1) + K_1^*\gamma_{xx}(p) \end{bmatrix} \tag{12.4.32}$$

Step 3. Shift the second row of $\mathbf{V}_1\mathbf{G}_1$ by one place to the right and thus form the new generator matrix

$$\mathbf{G}_2 = \begin{bmatrix} 0 & 0 & \gamma_{xx}(2) + K_1\gamma_{xx}(1) & \cdots & \gamma_{xx}(p) + K_1\gamma_{xx}(p-1) \\ 0 & 0 & \gamma_{xx}(0) + K_1^*\gamma_{xx}(1) & \cdots & \gamma_{xx}(p-2) + K_1^*\gamma_{xx}(p-1) \end{bmatrix} \tag{12.4.33}$$

The negative ratio of the elements in the third column of G_2 yields K_2.

Steps 2 and 3 are repeated until we have solved for all p reflection coefficients. In general, the 2×2 matrix in Step 2 is

$$\mathbf{V}_m = \begin{bmatrix} 1 & K_m \\ K_m^* & 1 \end{bmatrix} \tag{12.4.34}$$

and multiplication of $\mathbf{V}_m$ by $\mathbf{G}_m$ yields $\mathbf{V}_m\mathbf{G}_m$. In Step 3 we shift the second row of $\mathbf{V}_m\mathbf{G}_m$ one place to the right and obtain the new generator matrix $\mathbf{G}_{m+1}$.

We observe that the shifting operation of the second row in each iteration is equivalent to multiplication by the delay operator z^{-1} in the second recursive equation in (12.4.25). We also note that the division of the polynomial $P_m(z)$ by the polynomial $Q_m(z)$ and the evaluation of the quotient at $z = \infty$ is equivalent to dividing the elements in the $(m + 1)$st column of $\mathbf{G}_m$. The computation of the p reflection coefficients can be accomplished by use of parallel processors in $O(p)$ time units. Below we describe a pipelined architecture for performing these computations.

Another way of demonstrating the relationship of the Schur algorithm to the Levinson–Durbin algorithm and the corresponding lattice predictor is to determine the output of the lattice filter obtained when the input sequence is the correlation sequence $\{\gamma_{xx}(m), m = 0, 1, \ldots\}$. Thus, the first input to the lattice filter is $\gamma_{xx}(0)$, the second input is $\gamma_{xx}(1)$, and so on [i.e., $f_0(n) = \gamma_{xx}(n)$]. After the delay in the first stage, we have $g_0(n-1) = \gamma_{xx}(n-1)$. Hence, for $n = 1$, the ratio $f_0(1)/g_0(0) = \gamma_{xx}(1)/\gamma_{xx}(0)$, which is the negative of the reflection coefficient K_1. Alternatively, we can express this relationship as

$$f_0(1) + K_1 g_0(0) = \gamma_{xx}(1) + K_1\gamma_{xx}(0) = 0$$

Furthermore, $g_0(0) = \gamma_{xx}(0) = E_0^f$. At time $n = 2$, the input to the second stage is, according to (12.3.4),

$$f_1(2) = f_0(2) + K_1 g_0(1) = \gamma_{xx}(2) + K_1\gamma_{xx}(1)$$

and after the unit of delay in the second stage, we have

$$g_1(1) = K_1^* f_0(1) + g_0(0) = K_1^*\gamma_{xx}(1) + \gamma_{xx}(0)$$

Now the ratio $f_1(2)/g_1(1)$ is

$$\frac{f_1(2)}{g_1(1)} = \frac{\gamma_{xx}(2) + K_1\gamma_{xx}(1)}{\gamma_{xx}(0) + K_1^*\gamma_{xx}(1)} = \frac{\gamma_{xx}(2) + K_1\gamma_{xx}(1)}{E_1^f} = -K_2$$

Hence

$$f_1(2) + K_2 g_1(1) = 0$$

$$g_1(1) = E_1^f$$

By continuing in this manner, we can show that at the input to the mth lattice stage, the ratio $f_{m-1}(m)/g_{m-1}(m-1) = -K_m$ and $g_{m-1}(m-1) = E_{m-1}^f$. Consequently, the lattice filter coefficients obtained from the Levinson algorithm are identical to the coefficients obtained in the Schur algorithm. Furthermore, the lattice filter structure provides a method for computing the reflection coefficients in the lattice predictor.

A pipelined architecture for implementing the Schur algorithm. Kung and Hu (1983) developed a pipelined lattice-type processor for implementing the Schur algorithm. The processor consists of a cascade of p lattice-type stages, where each stage consists of two processing elements (PEs), which we designate as upper PEs denoted as $A_1, A_2, \ldots, A_p$, and lower PEs denoted as $B_1, B_2, \ldots, B_p$, as shown in Fig. 12.4.1. The PE designated as A_1 is assigned the task of performing divisions. The remaining PEs perform one multiplication and one addition per iteration (one clock cycle).

Initially, the upper PEs are loaded with the elements of the first row of the generator matrix $\mathbf{G}_0$, as illustrated in Fig. 12.4.1. The lower PEs are loaded with the elements of the second row of the generator matrix $\mathbf{G}_0$. The computational process begins with the division PE, A_1, which computes the first reflection coefficient as $K_1 = -\gamma_{xx}(1)/\gamma_{xx}(0)$. The value of K_1 is sent simultaneously to all the PEs in the upper branch and lower branch.

The second step in the computation is to update the contents of all processing elements simultaneously. The contents of the upper and lower PEs are updated as follows:

$$\text{PE } A_m\text{: } A_m \leftarrow A_m + K_1 B_m, \quad m = 2, 3, \ldots, p$$
$$\text{PE } B_m\text{: } B_m \leftarrow B_m + K_1^* A_m, \quad m = 1, 2, \ldots, p$$

The third step involves the shifting of the contents of the upper PEs one place to the left. Thus we have

$$\text{PE } A_m\text{: } A_{m-1} \leftarrow A_m, \qquad m = 2, 3, \ldots, p$$

At this point, PE A_1 contains $\gamma_{xx}(2) + K_1\gamma_{xx}(1)$ while PE B_1 contains $\gamma_{xx}(0) + K_1^*\gamma_{xx}(1)$. Hence the processor A_1 is ready to begin the second cycle by computing the second reflection coefficient $K_2 = -A_1/B_1$. The three computational steps beginning with the division A_1/B_1 are repeated until all p reflection coefficients are computed. Note that PE B_1 provides the minimum mean-square error E_m^f for each iteration.

If τ_d denotes the time for PE A_1 to perform a (complex) division and τ_{ma} is the time required to perform a (complex) multiplication and an addition, the time required to compute all p reflection coefficients is $p(\tau_d + \tau_{\text{ma}})$ for the Schur algorithm.

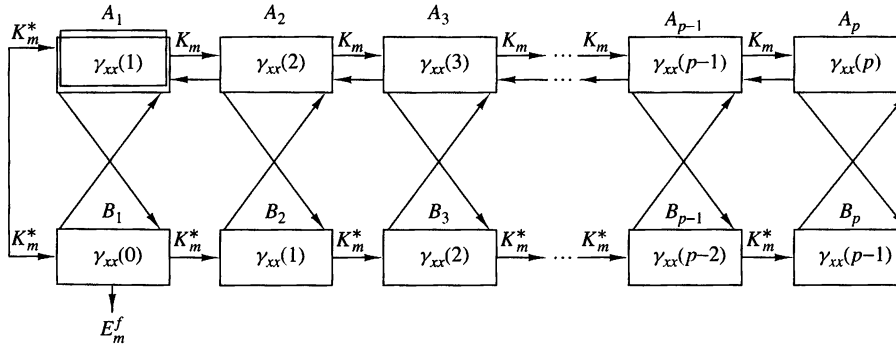


Figure 12.4.1 Pipelined parallel processor for computing the reflection coefficients.

12.5 Properties of the Linear Prediction-Error Filters

Linear prediction filters possess several important properties, which we now describe. We begin by demonstrating that the forward prediction-error filter is minimum phase.

Minimum-phase property of the forward prediction-error filter. We have already demonstrated that the reflection coefficients $\{K_i\}$ are correlation coefficients, and consequently, $|K_i| \leq 1$ for all i. This condition and the relation $E_m^f = (1 - |K_m|^2)E_{m-1}^f$ can be used to show that the zeros of the prediction-error filter are either all inside the unit circle or they are all on the unit circle.

First, we show that if $E_p^f > 0$, the zeros $|z_i| < 1$ for every i. The proof is by induction. Clearly, for $p = 1$ the system function for the prediction-error filter is

$$A_1(z) = 1 + K_1 z^{-1} \tag{12.5.1}$$

Hence $z_1 = -K_1$ and $E_1^f = (1 - |K_1|^2)E_0^f > 0$. Now, suppose that the hypothesis is true for $p - 1$. Then, if z_i is a root of $A_p(z)$, we have from (12.3.16) and (12.3.18),

$$\begin{aligned} A_p(z_i) &= A_{p-1}(z_i) + K_p z_i^{-1} B_{p-1}(z_i) \\ &= A_{p-1}(z_i) + K_p z_i^{-p} A_{p-1}^*\left(\frac{1}{z_i}\right) = 0 \end{aligned} \tag{12.5.2}$$

Hence

$$\frac{1}{K_p} = -\frac{z_i^{-p} A_{p-1}^*(1/z_i)}{A_{p-1}(z_i)} \equiv Q(z_i) \tag{12.5.3}$$

We note that the function $Q(z)$ is all pass. In general, an all-pass function of the form

$$P(z) = \prod_{k=1}^{N} \frac{z z_k^* + 1}{z + z_k}, \qquad |z_k| < 1 \tag{12.5.4}$$

satisfies the property that $|P(z)| > 1$ for $|z| < 1$, $|P(z)| = 1$ for $|z| = 1$, and $|P(z)| < 1$ for $|z| > 1$. Since $Q(z) = -P(z)/z$, it follows that $|z_i| < 1$ if $|Q(z)| > 1$. Clearly, this is the case since $Q(z_i) = 1/K_p$ and $E_p^f > 0$.

On the other hand, suppose that $E_{p-1}^f > 0$ and $E_p^f = 0$. In this case $|K_p| = 1$ and $|Q(z_i)| = 1$. Since the MMSE is zero, the random process $x(n)$ is called *predictable* or *deterministic*. Specifically, a purely sinusoidal random process of the form

$$x(n) = \sum_{k=1}^{M} \alpha_k e^{j(n\omega_k + \theta_k)} \tag{12.5.5}$$

where the phases $\{\theta_k\}$ are statistically independent and uniformly distributed over $(0, 2\pi)$, has the autocorrelation

$$\gamma_{xx}(m) = \sum_{k=1}^{M} \alpha_k^2 e^{jm\omega_k} \tag{12.5.6}$$

and the power density spectrum

$$\Gamma_{xx}(f) = \sum_{k=1}^{M} \alpha_k^2 \delta(f - f_k), \qquad f_k = \frac{\omega_k}{2\pi} \tag{12.5.7}$$

This process is predictable with a predictor of order $p \geq M$.

To demonstrate the validity of the statement, consider passing this process through a prediction error filter of order $p \geq M$. The MSE at the output of this filter is

$$\begin{aligned} \mathcal{E}_p^f &= \int_{-1/2}^{1/2} \Gamma_{xx}(f)|A_p(f)|^2 df \\ &= \int_{-1/2}^{1/2} \left[\sum_{k=1}^{M} \alpha_k^2 \delta(f - f_k) \right] |A_p(f)|^2 df \\ &= \sum_{k=1}^{M} \alpha_k^2 |A_p(f_k)|^2 \end{aligned} \tag{12.5.8}$$

By choosing M of the p zeros of the prediction-error filter to coincide with the frequencies $\{f_k\}$, the MSE $\mathcal{E}_p^f$ can be forced to zero. The remaining $p - M$ zeros can be selected arbitrarily to be anywhere inside the unit circle.

Finally, the reader can prove that if a random process consists of a mixture of a continuous power spectral density and a discrete spectrum, the prediction-error filter must have all its roots inside the unit circle.

Maximum-phase property of the backward prediction-error filter. The system function for the backward prediction-error filter of order p is

$$B_p(z) = z^{-p} A_p^*(z^{-1}) \tag{12.5.9}$$

Consequently, the roots of $B_p(z)$ are the reciprocals of the roots of the forward prediction-error filter with system function $A_p(z)$. Hence if $A_p(z)$ is minimum phase, then $B_p(z)$ is maximum phase. However, if the process $x(n)$ is predictable, all the roots of $B_p(z)$ lie on the unit circle.

Whitening property. Suppose that the random process $x(n)$ is an AR(p) stationary random process that is generated by passing white noise with variance σ_w^2 through an all-pole filter with system function

$$H(z) = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{12.5.10}$$

Then the prediction-error filter of order p has the system function

$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k) z^{-k} \tag{12.5.11}$$

where the predictor coefficients $a_p(k) = a_k$. The response of the prediction-error filter is a white noise sequence $\{w(n)\}$. In this case the prediction-error filter whitens the input random process $x(n)$ and is called a whitening filter, as indicated in Section 12.3.4.

More generally, even if the input process $x(n)$ is not an AR process, the prediction-error filter attempts to remove the correlation among the signal samples of the input process. As the order of the predictor is increased, the predictor output $\hat{x}(n)$ becomes a closer approximation to $x(n)$ and hence the difference $f(n) = \hat{x}(n) - x(n)$ approaches a white noise sequence.

Orthogonality of the backward prediction errors. The backward prediction errors $\{g_m(k)\}$ from different stages in the FIR lattice filter are orthogonal. That is,

$$E[g_m(n)g_l^*(n)] = \begin{cases} 0, & 0 \le l \le m-1 \\ E_m^b, & l = m \end{cases} \tag{12.5.12}$$

This property is easily proved by substituting for $g_m(n)$ and $g_l^*(n)$ into (12.5.12) and carrying out the expectation. Thus

$$\begin{aligned} E[g_m(n)g_l^*(n)] &= \sum_{k=0}^{m} b_m(k) \sum_{j=0}^{l} b_l^*(j) E[x(n-k)x^*(n-j)] \\ &= \sum_{j=0}^{l} b_l^*(j) \sum_{k=0}^{m} b_m(k) \gamma_{xx}(j-k) \end{aligned} \tag{12.5.13}$$

But the normal equations for the backward linear predictor require that

$$\sum_{k=0}^{m} b_m(k) \gamma_{xx}(j-k) = \begin{cases} 0, & j = 1, 2, \ldots, m-1 \\ E_m^b, & j = m \end{cases} \tag{12.5.14}$$

Therefore,

$$E[g_m(n)g_l^*(n)] = \begin{cases} E_m^b = E_m^f, & m = l \\ 0, & 0 \le l \le m-1 \end{cases} \tag{12.5.15}$$

Additional properties. There are a number of other interesting properties regarding the forward and backward prediction errors in the FIR lattice filter. These are given here for real-valued data. Their proof is left as an exercise for the reader.

(a) $E[f_m(n)x(n-i)] = 0, \qquad 1 \le i \le m$

(b) $E[g_m(n)x(n-i)] = 0, \qquad 0 \le i \le m-1$

(c) $E[f_m(n)x(n)] = E[g_m(n)x(n-m)] = E_m$

(d) $E[f_i(n)f_j(n)] = E_{\max}(i, j)$

(e) $E[f_i(n)f_j(n-t)] = 0$, for $\begin{cases} 1 \le t \le i-j, & i > j \\ -1 \ge t \ge i-j, & i < j \end{cases}$

(f) $E[g_i(n)g_j(n-t)] = 0$, for $\begin{cases} 0 \le t \le i-j, & i > j \\ 0 \ge t \ge i-j+1, & i < j \end{cases}$

(g) $E[f_i(n+i)f_j(n+j)] = \begin{cases} E_i, & i = j \\ 0, & i \ne j \end{cases}$

(h) $E[g_i(n+i)g_j(n+j)] = E_{\max\,(i,j)}$

(i) $E[f_i(n)g_j(n)] = \begin{cases} K_j E_i, & i \ge j, \quad i, j \ge 0, \quad K_0 = 1 \\ 0, & i < j \end{cases}$

(j) $E[f_i(n)g_i(n-1)] = -K_{i+1}E_i$

(k) $E[g_i(n-1)x(n)] = E[f_i(n+1)x(n-1)] = -K_{i+1}E_i$

(l) $E[f_i(n)g_j(n-1)] = \begin{cases} 0, & i > j \\ -K_{j+1}E_i, & i \le j \end{cases}$

12.6 AR Lattice and ARMA Lattice-Ladder Filters

In Section 12.3 we showed the relationship of the all-zero FIR lattice to linear prediction. The linear predictor with transfer function,

$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k)z^{-k} \tag{12.6.1}$$

when excited by an input random process $\{x(n)\}$, produces an output that approaches a white noise sequence as $p \to \infty$. On the other hand, if the input process is an AR(p), the output of $A_p(z)$ is white. Since $A_p(z)$ generates an MA(p) process when excited with a white noise sequence, the all-zero lattice is sometimes called an MA lattice.

In the following section, we develop the lattice structure for the inverse filter $1/A_p(z)$, called the AR lattice, and the lattice-ladder structure for an ARMA process.

12.6.1 AR Lattice Structure

Let us consider an all-pole system with system function

$$H(z) = \frac{1}{1 + \sum_{k=1}^{p} a_p(k)z^{-k}} \tag{12.6.2}$$

The difference equation for this IIR system is

$$y(n) = -\sum_{k=1}^{p} a_p(k)y(n-k) + x(n) \tag{12.6.3}$$

Now suppose that we interchange the roles of the input and output [i.e., interchange $x(n)$ with $y(n)$ in (12.6.3)], obtaining the difference equation

$$x(n) = -\sum_{k=1}^{p} a_p(k)x(n-k) + y(n)$$

or, equivalently,

$$y(n) = x(n) + \sum_{k=1}^{p} a_p(k)x(n-k) \tag{12.6.4}$$

We observe that (12.6.4) is a difference equation for an FIR system with system function $A_p(z)$. Thus an all-pole IIR system can be converted to an all-zero system by interchanging the roles of the input and output.

Based on this observation, we can obtain the structure of an AR(p) lattice from an MA(p) lattice by interchanging the input with the output. Since the MA(p) lattice has $y(n) = f_p(n)$ as its output and $x(n) = f_0(n)$ is the input, we let

$$\begin{aligned} x(n) &= f_p(n) \\ y(n) &= f_0(n) \end{aligned} \tag{12.6.5}$$

These definitions dictate that the quantities $\{f_m(n)\}$ be computed in descending order. This computation can be accomplished by rearranging the recursive equation for $\{f_m(n)\}$ in (12.3.4) and solving for $f_{m-1}(n)$ in terms of $f_m(n)$. Thus we obtain

$$f_{m-1}(n) = f_m(n) - K_m g_{m-1}(n-1), \qquad m = p, p-1, \ldots, 1$$

The equation for $g_m(n)$ remains unchanged. The result of these changes is the set of equations

$$\begin{aligned} x(n) &= f_p(n) \\ f_{m-1}(n) &= f_m(n) - K_m g_{m-1}(n-1) \\ g_m(n) &= K_m^* f_{m-1}(n) + g_{m-1}(n-1) \\ y(n) &= f_0(n) = g_0(n) \end{aligned} \tag{12.6.6}$$

The corresponding structure for the AR(p) lattice is shown in Fig. 12.6.1. Note that the all-pole lattice structure has an all-zero path with input $g_0(n)$ and output $g_p(n)$, which is identical to the all-zero path in the MA(p) lattice structure. This is not surprising, since the equation for $g_m(n)$ is identical in the two lattice structures.

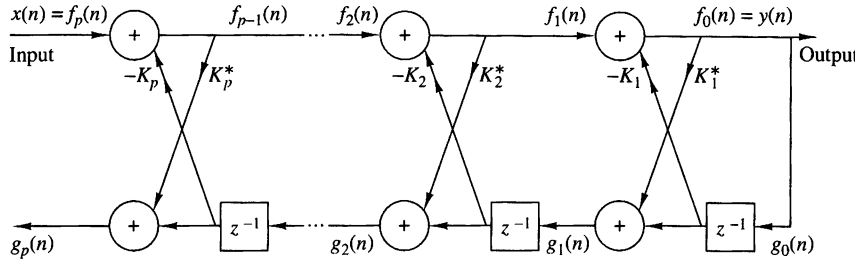


Figure 12.6.1 Lattice structure for an all-pole system.

We also observe that the AR(p) and MA(p) lattice structures are characterized by the same parameters, namely, the reflection coefficients $\{K_i\}$. Consequently, the equations given in (12.3.21) and (12.3.23) for converting between the system parameters $\{a_p(k)\}$ in the direct-form realizations of the all-zero system $A_p(z)$ and the lattice parameters $\{K_i\}$ of the MA(p) lattice structure apply as well to the all-pole structures.

12.6.2 ARMA Processes and Lattice-Ladder Filters

The all-pole lattice provides the basic building block for lattice-type structures that implement IIR systems that contain both poles and zeros. To construct the appropriate structure, let us consider an IIR system with system function

$$H(z) = \frac{\sum_{k=0}^{q} c_q(k) z^{-k}}{1 + \sum_{k=1}^{p} a_p(k) z^{-k}} = \frac{C_q(z)}{A_p(z)} \tag{12.6.7}$$

Without loss of generality, we assume that $p \geq q$.

This system is described by the difference equations

$$\begin{aligned} v(n) &= -\sum_{k=1}^{p} a_p(k) v(n-k) + x(n) \\ y(n) &= \sum_{k=0}^{q} c_q(k) v(n-k) \end{aligned} \tag{12.6.8}$$

obtained by viewing the system as a cascade of an all-pole system followed by an all-zero system. From (12.6.8) we observe that the output $y(n)$ is simply a linear combination of delayed outputs from the all-pole system.

Since zeros result from forming a linear combination of previous outputs, we can carry over this observation to construct a pole–zero system using the all-pole lattice structure as the basic building block. We have clearly observed that $g_m(n)$ in the

all-pole lattice can be expressed as a linear combination of present and past outputs. In fact, the system

$$H_b(z) \equiv \frac{G_m(z)}{Y(z)} = B_m(z) \tag{12.6.9}$$

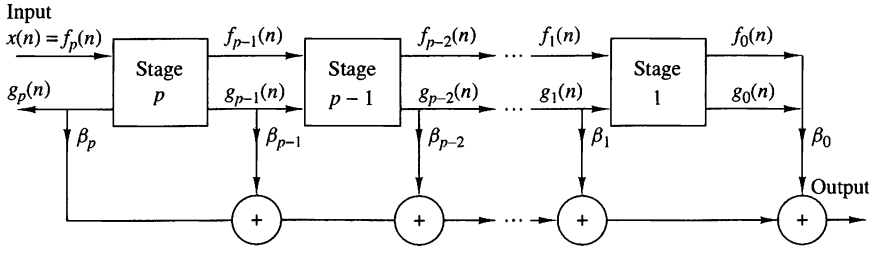
is an all-zero system. Therefore, any linear combination of $\{g_m(n)\}$ is also an all-zero filter.

Let us begin with an all-pole lattice filter with coefficients K_m, $1 \leq m \leq p$, and add a *ladder* part by taking as the output a weighted linear combination of $\{g_m(n)\}$. The result is a pole–zero filter that has the *lattice-ladder* structure shown in Fig. 12.6.2. Its output is

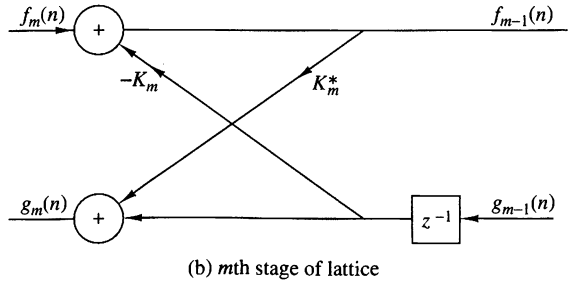
$$y(n) = \sum_{k=0}^{q} \beta_k g_k(n) \tag{12.6.10}$$

where $\{\beta_k\}$ are the parameters that determine the zeros of the system. The system function corresponding to (12.6.10) is

$$H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{q} \beta_k \frac{G_k(z)}{X(z)} \tag{12.6.11}$$



(a) Pole–zero system



(b) mth stage of lattice

Figure 12.6.2 Lattice-ladder structure for pole–zero system.

Since $X(z) = F_p(z)$ and $F_0(z) = G_0(z)$, (12.6.11), can be expressed as

$$\begin{aligned} H(z) &= \sum_{k=0}^{q} \beta_k \frac{G_k(z)}{G_0(z)} \frac{F_0(z)}{F_p(z)} \\ &= \frac{1}{A_p(z)} \sum_{k=0}^{q} \beta_k B_k(z) \end{aligned} \tag{12.6.12}$$

Therefore,

$$C_q(z) = \sum_{k=0}^{q} \beta_k B_k(z) \tag{12.6.13}$$

This is the desired relationship that can be used to determine the weighting coefficients $\{\beta_k\}$ as previously shown in Section 9.3.5.

Given the polynomials $C_q(z)$ and $A_p(z)$, where $p \geq q$, the reflection coefficients $\{K_i\}$ are determined first from the coefficients $\{a_p(k)\}$. By means of the step-down recursive relation given by (12.3.22), we also obtain the polynomials $B_k(z)$, $k = 1, 2, \ldots, p$. Then the ladder parameters can be obtained from (12.6.13), which can be expressed as

$$\begin{aligned} C_m(z) &= \sum_{k=0}^{m-1} \beta_k B_k(z) + \beta_m B_m(z) \\ &= C_{m-1}(z) + \beta_m B_m(z) \end{aligned} \tag{12.6.14}$$

or, equivalently,

$$C_{m-1}(z) = C_m(z) - \beta_m B_m(z), \qquad m = p, p-1, \ldots, 1 \tag{12.6.15}$$

By running this recursive relation backward, we can generate all the lower-degree polynomials, $C_m(z)$, $m = p-1, \ldots, 1$. Since $b_m(m) = 1$, the parameters β_m are determined from (12.6.15) by setting

$$\beta_m = c_m(m), \qquad m = p, p-1, \ldots, 1, 0$$

When excited by a white noise sequence, this lattice-ladder filter structure generates an ARMA(p, q) process that has a power density spectrum

$$\Gamma_{xx}(f) = \sigma_w^2 \frac{|C_q(f)|^2}{|A_p(f)|^2} \tag{12.6.16}$$

and an autocorrelation function that satisfies (12.2.18), where σ_w^2 is the variance of the input white noise sequence.

12.7 Wiener Filters for Filtering and Prediction

In many practical applications we are given an input signal $\{x(n)\}$, consisting of the sum of a desired signal $\{s(n)\}$ and an undesired noise or interference $\{w(n)\}$, and we are asked to design a filter that suppresses the undesired interference component. In such a case, the objective is to design a system that filters out the additive interference while preserving the characteristics of the desired signal $\{s(n)\}$.

In this section we treat the problem of signal estimation in the presence of an additive noise disturbance. The estimator is constrained to be a linear filter with impulse response $\{h(n)\}$, designed so that its output approximates some specified desired signal sequence $\{d(n)\}$. Figure 12.7.1 illustrates the linear estimation problem.

The input sequence to the filter is $x(n) = s(n) + w(n)$, and its output sequence is $y(n)$. The difference between the desired signal and the filter output is the error sequence $e(n) = d(n) - y(n)$.

We distinguish three special cases:

1. If $d(n) = s(n)$, the linear estimation problem is referred to as *filtering*.
2. If $d(n) = s(n + D)$, where $D > 0$, the linear estimation problem is referred to as signal *prediction*. Note that this problem is different than the prediction considered earlier in this chapter, where $d(n) = x(n + D)$, $D \geq 0$.
3. If $d(n) = s(n - D)$, where $D > 0$, the linear estimation problem is referred to as signal *smoothing*.

Our treatment will concentrate on filtering and prediction.

The criterion selected for optimizing the filter impulse response $\{h(n)\}$ is the minimization of the mean-square error. This criterion has the advantages of simplicity and mathematical tractability.

The basic assumptions are that the sequences $\{s(n)\}$, $\{w(n)\}$, and $\{d(n)\}$ are zero mean and wide-sense stationary. The linear filter will be assumed to be either FIR or IIR. If it is IIR, we assume that the input data $\{x(n)\}$ are available over the infinite past. We begin with the design of the optimum FIR filter. The optimum linear filter, in the sense of minimum mean-square error (MMSE), is called a *Wiener filter*.

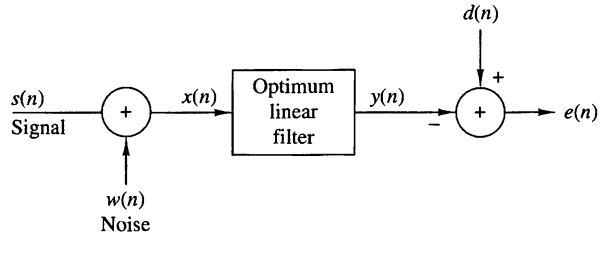


Figure 12.7.1
Model for linear estimation problem.

12.7.1 FIR Wiener Filter

Suppose that the filter is constrained to be of length M with coefficients $\{h_k, 0 \leq k \leq M-1\}$. Hence its output $y(n)$ depends on the finite data record $x(n)$, $x(n-1), \ldots, x(n-M+1)$,

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{12.7.1}$$

The mean-square value of the error between the desired output $d(n)$ and $y(n)$ is

$$\begin{aligned} \mathcal{E}_M &= E|e(n)|^2 \\ &= E\left|d(n) - \sum_{k=0}^{M-1} h(k)x(n-k)\right|^2 \end{aligned} \tag{12.7.2}$$

Since this is a quadratic function of the filter coefficients, the minimization of $\mathcal{E}_M$ yields the set of linear equations

$$\sum_{k=0}^{M-1} h(k)\gamma_{xx}(l-k) = \gamma_{dx}(l), \qquad l = 0, 1, \ldots, M-1 \tag{12.7.3}$$

where $\gamma_{xx}(k)$ is the autocorrelation of the input sequence $\{x(n)\}$ and $\gamma_{dx}(k) = E[d(n)x^*(n-k)]$ is the crosscorrelation between the desired sequence $\{d(n)\}$ and the input sequence $\{x(n), 0 \leq n \leq M-1\}$. The set of linear equations that specify the optimum filter is called the *Wiener–Hopf equation.* These equations are also called the normal equations, encountered earlier in the chapter in the context of linear one-step prediction.

In general, the equations in (12.7.3) can be expressed in matrix form as

$$\mathbf{\Gamma}_M \mathbf{h}_M = \boldsymbol{\gamma}_d \tag{12.7.4}$$

where $\mathbf{\Gamma}_M$ is an $M \times M$ (Hermitian) Toeplitz matrix with elements $\Gamma_{lk} = \gamma_{xx}(l-k)$ and $\boldsymbol{\gamma}_d$ is the $M \times 1$ crosscorrelation vector with elements $\gamma_{dx}(l)$, $l = 0, 1, \ldots, M-1$. The solution for the optimum filter coefficients is

$$\mathbf{h}_{\text{opt}} = \mathbf{\Gamma}_M^{-1} \boldsymbol{\gamma}_d \tag{12.7.5}$$

and the resulting minimum MSE achieved by the Wiener filter is

$$\text{MMSE}_M = \min_{\mathbf{h}_M} \mathcal{E}_M = \sigma_d^2 - \sum_{k=0}^{M-1} h_{\text{opt}}(k)\gamma_{dx}^*(k) \tag{12.7.6}$$

or, equivalently,

$$\text{MMSE}_M = \sigma_d^2 - \boldsymbol{\gamma}_d^{*t} \mathbf{\Gamma}_M^{-1} \boldsymbol{\gamma}_d \tag{12.7.7}$$

where $\sigma_d^2 = E|d(n)|^2$.

Let us consider some special cases of (12.7.3). If we are dealing with filtering, the $d(n) = s(n)$. Furthermore, if $s(n)$ and $w(n)$ are uncorrelated random sequences, as is usually the case in practice, then

$$\begin{aligned} \gamma_{xx}(k) &= \gamma_{ss}(k) + \gamma_{ww}(k) \\ \gamma_{dx}(k) &= \gamma_{ss}(k) \end{aligned} \tag{12.7.8}$$

and the normal equations in (12.7.3) become

$$\sum_{k=0}^{M-1} h(k)[\gamma_{ss}(l-k) + \gamma_{ww}(l-k)] = \gamma_{ss}(l), \qquad l = 0, 1, \ldots, M-1 \tag{12.7.9}$$

If we are dealing with prediction, then $d(n) = s(n+D)$ where $D > 0$. Assuming that $s(n)$ and $w(n)$ are uncorrelated random sequences, we have

$$\gamma_{dx}(k) = \gamma_{ss}(l+D) \tag{12.7.10}$$

Hence the equations for the Wiener prediction filter become

$$\sum_{k=0}^{M-1} h(k)[\gamma_{ss}(l-k) + \gamma_{ww}(l-k)] = \gamma_{ss}(l+D), \qquad l = 0, 1, \ldots, M-1 \tag{12.7.11}$$

In all these cases, the correlation matrix to be inverted is Toeplitz. Hence the (generalized) Levinson–Durbin algorithm may be used to solve for the optimum filter coefficients.

EXAMPLE 12.7.1

Let us consider a signal $x(n) = s(n) + w(n)$, where $s(n)$ is an AR(1) process that satisfies the difference equation

$$s(n) = 0.6s(n-1) + v(n)$$

where $\{v(n)\}$ is a white noise sequence with variance $\sigma_v^2 = 0.64$, and $\{w(n)\}$ is a white noise sequence with variance $\sigma_w^2 = 1$. We will design a Wiener filter of length $M = 2$ to estimate $\{s(n)\}$.

Solution. Since $\{s(n)\}$ is obtained by exciting a single-pole filter by white noise, the power spectral density of $s(n)$ is

$$\begin{aligned} \Gamma_{ss}(f) &= \sigma_v^2 |H(f)|^2 \\ &= \frac{0.64}{|1 - 0.6e^{-j2\pi f}|^2} \\ &= \frac{0.64}{1.36 - 1.2\cos 2\pi f} \end{aligned}$$

The corresponding autocorrelation sequence $\{\gamma_{ss}(m)\}$ is

$$\gamma_{ss}(m) = (0.6)^{|m|}$$

The equations for the filter coefficients are

$$2h(0) + 0.6h(1) = 1$$

$$0.6h(0) + 2h(1) = 0.6$$

Solution of these equations yields the result

$$h(0) = 0.451, \qquad h(1) = 0.165$$

The corresponding minimum MSE is

$$\begin{aligned} \text{MMSE}_2 &= 1 - h(0)\gamma_{ss}(0) - h(1)\gamma_{ss}(1) \\ &= 1 - 0.451 - (0.165)(0.6) \\ &= 0.45 \end{aligned}$$

This error can be reduced further by increasing the length of the Wiener filter (see Problem 12.35).

12.7.2 Orthogonality Principle in Linear Mean-Square Estimation

The normal equations for the optimum filter coefficients given by (12.7.3) can be obtained directly by applying the orthogonality principle in linear mean-square estimation. Simply stated, the mean-square error $\mathcal{E}_M$ in (12.7.2) is a minimum if the filter coefficients $\{h(k)\}$ are selected such that the error is orthogonal to each of the data points in the estimate,

$$E[e(n)x^*(n-l)] = 0, \qquad l = 0, 1, \ldots, M-1 \tag{12.7.12}$$

where

$$e(n) = d(n) - \sum_{k=0}^{M-1} h(k)x(n-k) \tag{12.7.13}$$

Conversely, if the filter coefficients satisfy (12.7.12), the resulting MSE is a minimum.

When viewed geometrically, the output of the filter, which is the estimate

$$\hat{d}(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{12.7.14}$$

is a vector in the subspace spanned by the data $\{x(k), 0 \le k \le M-1\}$. The error $e(n)$ is a vector from $d(n)$ to $\hat{d}(n)$ [i.e., $d(n) = e(n) + \hat{d}(n)$], as shown in Fig. 12.7.2. The orthogonality principle states that the length $\mathcal{E}_M = E|e(n)|^2$ is a minimum when $e(n)$ is perpendicular to the data subspace [i.e., $e(n)$ is orthogonal to each data point $x(k)$, $0 \le k \le M-1$].

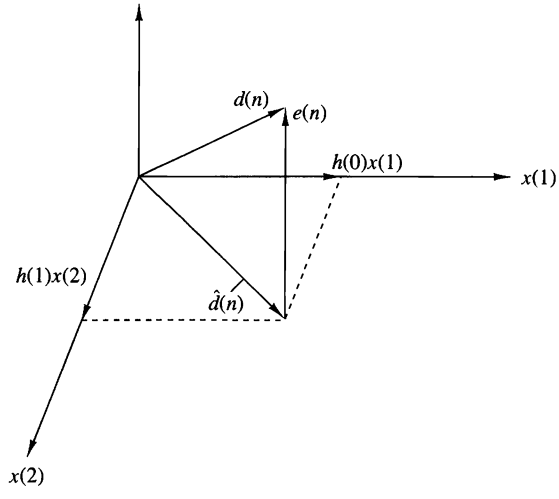


Figure 12.7.2
Geometric interpretation of linear MSE problem.

We note that the solution obtained from the normal equations in (12.7.3) is unique if the data $\{x(n)\}$ in the estimate $\hat{d}(n)$ are *linearly independent.* In this case, the correlation matrix $\mathbf{\Gamma}_M$ is nonsingular. On the other hand, if the data are linearly dependent, the rank of $\mathbf{\Gamma}_M$ is less than M and therefore the solution is not unique. In this case, the estimate $\hat{d}(n)$ can be expressed as a linear combination of a reduced set of linearly independent data points equal to the rank of $\mathbf{\Gamma}_M$.

Since the MSE is minimized by selecting the filter coefficients to satisfy the orthogonality principle, the residual minimum MSE is simply

$$\text{MMSE}_M = E[e(n)d^*(n)] \tag{12.7.15}$$

which yields the result given in (12.7.6).

12.7.3 IIR Wiener Filter

In the preceding section we constrained the filter to be FIR and obtained a set of M linear equations for the optimum filter coefficients. In this section we allow the filter to be infinite in duration (IIR) and the data sequence to be infinite as well. Hence the filter output is

$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{12.7.16}$$

The filter coefficients are selected to minimize the mean-square error between the desired output $d(n)$ and $y(n)$, that is,

$$\begin{aligned} \mathcal{E}_\infty &= E|e(n)|^2 \\ &= E\left|d(n) - \sum_{k=0}^{\infty} h(k)x(n-k)\right|^2 \end{aligned} \tag{12.7.17}$$

Application of the orthogonality principle leads to the Wiener–Hopf equation,

$$\sum_{k=0}^{\infty} h(k)\gamma_{xx}(l-k) = \gamma_{dx}(l), \qquad l \geq 0 \tag{12.7.18}$$

The residual MMSE is simply obtained by application of the condition given by (12.7.15). Thus we obtain

$$\text{MMSE}_{\infty} = \min_{\mathbf{h}} \mathcal{E}_{\infty} = \sigma_d^2 - \sum_{k=0}^{\infty} h_{\text{opt}}(k)\gamma_{dx}^*(k) \tag{12.7.19}$$

The Wiener–Hopf equation given by (12.7.18) cannot be solved directly with z-transform techniques, because the equation holds only for $l \geq 0$. We shall solve for the optimum IIR Wiener filter based on the innovations representation of the stationary random process $\{x(n)\}$.

Recall that a stationary random process $\{x(n)\}$ with autocorrelation $\gamma_{xx}(k)$ and power spectral density $\Gamma_{xx}(f)$ can be represented by an equivalent innovations process, $\{i(n)\}$, by passing $\{x(n)\}$ through a noise-whitening filter with system function $1/G(z)$, where $G(z)$ is the minimum-phase part obtained from the spectral factorization of $\Gamma_{xx}(z)$:

$$\Gamma_{xx}(z) = \sigma_i^2 G(z)G(z^{-1}) \tag{12.7.20}$$

Hence $G(z)$ is analytic in the region $|z| > r_1$, where $r_1 < 1$.

Now, the optimum Wiener filter can be viewed as the cascade of the whitening filter $1/G(z)$ with a second filter, say $Q(z)$, whose output $y(n)$ is identical to the output of the optimum Wiener filter. Since

$$y(n) = \sum_{k=0}^{\infty} q(k)i(n-k) \tag{12.7.21}$$

and $e(n) = d(n) - y(n)$, application of the orthogonality principle yields the new Wiener–Hopf equation as

$$\sum_{k=0}^{\infty} q(k)\gamma_{ii}(l-k) = \gamma_{di}(l), \qquad l \geq 0 \tag{12.7.22}$$

But since $\{i(n)\}$ is white, it follows that $\gamma_{ii}(l-k) = 0$ unless $l = k$. Thus we obtain the solution as

$$q(l) = \frac{\gamma_{di}(l)}{\gamma_{ii}(0)} = \frac{\gamma_{di}(l)}{\sigma_i^2}, \qquad l \geq 0 \tag{12.7.23}$$

The z-transform of the sequence $\{q(l)\}$ is

$$\begin{aligned} Q(z) &= \sum_{k=0}^{\infty} q(k)z^{-k} \\ &= \frac{1}{\sigma_i^2} \sum_{k=0}^{\infty} \gamma_{di}(k)z^{-k} \end{aligned} \tag{12.7.24}$$

If we denote the z-transform of the two-sided crosscorrelation sequence $\gamma_{di}(k)$ by $\Gamma_{di}(z)$

$$\Gamma_{di}(z) = \sum_{k=-\infty}^{\infty} \gamma_{di}(k) z^{-k} \tag{12.7.25}$$

and define $[\Gamma_{di}(z)]_+$ as

$$[\Gamma_{di}(z)]_+ = \sum_{k=0}^{\infty} \gamma_{di}(k) z^{-k} \tag{12.7.26}$$

then

$$Q(z) = \frac{1}{\sigma_i^2} [\Gamma_{di}(z)]_+ \tag{12.7.27}$$

To determine $[\Gamma_{di}(z)]_+$, we begin with the output of the noise-whitening filter, which can be expressed as

$$i(n) = \sum_{k=0}^{\infty} v(k) x(n-k) \tag{12.7.28}$$

where $\{v(k), k \geq 0\}$ is the impulse response of the noise-whitening filter,

$$\frac{1}{G(z)} \equiv V(z) = \sum_{k=0}^{\infty} v(k) z^{-k} \tag{12.7.29}$$

Then

$$\begin{aligned} \gamma_{di}(k) &= E[d(n) i^*(n-k)] \\ &= \sum_{m=0}^{\infty} v(m) E[d(n) x^*(n-m-k)] \\ &= \sum_{m=0}^{\infty} v(m) \gamma_{dx}(k+m) \end{aligned} \tag{12.7.30}$$

The z-transform of the crosscorrelation $\gamma_{di}(k)$ is

$$\begin{aligned} \Gamma_{di}(z) &= \sum_{k=-\infty}^{\infty} \left[\sum_{m=0}^{\infty} v(m) \gamma_{dx}(k+m) \right] z^{-k} \\ &= \sum_{m=0}^{\infty} v(m) \sum_{k=-\infty}^{\infty} \gamma_{dx}(k+m) z^{-k} \\ &= \sum_{m=0}^{\infty} v(m) z^m \sum_{k=-\infty}^{\infty} \gamma_{dx}(k) z^{-k} \\ &= V(z^{-1}) \Gamma_{dx}(z) = \frac{\Gamma_{dx}(z)}{G(z^{-1})} \end{aligned} \tag{12.7.31}$$

Therefore,

$$Q(z) = \frac{1}{\sigma_i^2}\left[\frac{\Gamma_{dx}(z)}{G(z^{-1})}\right]_+ \tag{12.7.32}$$

Finally, the optimum IIR Wiener filter has the system function

$$\begin{aligned} H_{\text{opt}}(z) &= \frac{Q(z)}{G(z)} \\ &= \frac{1}{\sigma_i^2 G(z)}\left[\frac{\Gamma_{dx}(z)}{G(z^{-1})}\right]_+ \end{aligned} \tag{12.7.33}$$

In summary, the solution for the optimum IIR Wiener filter requires that we perform a spectral factorization of $\Gamma_{xx}(z)$ to obtain $G(z)$, the minimum-phase component, and then we solve for the causal part of $\Gamma_{dx}(z)/G(z^{-1})$. The following example illustrates the procedure.

EXAMPLE 12.7.2

Let us determine the optimum IIR Wiener filter for the signal given in Example 12.7.1.

Solution. For this signal we have

$$\Gamma_{xx}(z) = \Gamma_{ss}(z) + 1 = \frac{1.8(1 - \frac{1}{3}z^{-1})(1 - \frac{1}{3}z)}{(1 - 0.6z^{-1})(1 - 0.6z)}$$

where $\sigma_i^2 = 1.8$ and

$$G(z) = \frac{1 - \frac{1}{3}z^{-1}}{1 - 0.6z^{-1}}$$

The z-transform of the crosscorrelation $\gamma_{dx}(m)$ is

$$\Gamma_{dx}(z) = \Gamma_{ss}(z) = \frac{0.64}{(1 - 0.6z^{-1})(1 - 0.6z)}$$

Hence

$$\begin{aligned} \left[\frac{\Gamma_{dx}(z)}{G(z^{-1})}\right]_+ &= \left[\frac{0.64}{(1 - \frac{1}{3}z)(1 - 0.6z^{-1})}\right]_+ \\ &= \left[\frac{0.8}{1 - 0.6z^{-1}} + \frac{0.266z}{1 - \frac{1}{3}z}\right]_+ \\ &= \frac{0.8}{1 - 0.6z^{-1}} \end{aligned}$$

The optimum IIR filter has the system function

$$\begin{aligned} H_{\text{opt}}(z) &= \frac{1}{1.8}\left(\frac{1 - 0.6z^{-1}}{1 - \frac{1}{3}z^{-1}}\right)\left(\frac{0.8}{1 - 0.6z^{-1}}\right) \\ &= \frac{\frac{4}{9}}{1 - \frac{1}{3}z^{-1}} \end{aligned}$$

and an impulse response

$$h_{\text{opt}}(n) = \tfrac{4}{9}\left(\tfrac{1}{3}\right)^n, \qquad n \geq 0$$

We conclude this section by expressing the minimum MSE given by (12.7.19) in terms of the frequency-domain characteristics of the filter. First, we note that $\sigma_d^2 \equiv E|d(n)|^2$ is simply the value of the autocorrelation sequence $\{\gamma_{dd}(k)\}$ evaluated at $k = 0$. Since

$$\gamma_{dd}(k) = \frac{1}{2\pi j}\oint_C \Gamma_{dd}(z) z^{k-1} dz \tag{12.7.34}$$

it follows that

$$\sigma_d^2 = \gamma_{dd}(0) = \frac{1}{2\pi j}\oint_C \frac{\Gamma_{dd}(z)}{z} dz \tag{12.7.35}$$

where the contour integral is evaluated along a closed path encircling the origin in the region of convergence of $\Gamma_{dd}(z)$.

The second term in (12.7.19) is also easily transformed to the frequency domain by application of Parseval's theorem. Since $h_{\text{opt}}(k) = 0$ for $k < 0$, we have

$$\sum_{k=-\infty}^{\infty} h_{\text{opt}}(k)\gamma_{dx}^*(k) = \frac{1}{2\pi j}\oint_C H_{\text{opt}}(z)\Gamma_{dx}(z^{-1}) z^{-1} dz \tag{12.7.36}$$

where C is a closed contour encircling the origin that lies within the common region of convergence of $H_{\text{opt}}(z)$ and $\Gamma_{dx}(z^{-1})$.

By combining (12.7.35) with (12.7.36), we obtain the desired expression for the MMSE_∞ in the form

$$\text{MMSE}_\infty = \frac{1}{2\pi j}\oint_C [\Gamma_{dd}(z) - H_{\text{opt}}(z)\Gamma_{dx}(z^{-1})] z^{-1} dz \tag{12.7.37}$$

EXAMPLE 12.7.3

For the optimum Wiener filter derived in Example 12.7.2, the minimum MSE is

$$\text{MMSE}_\infty = \frac{1}{2\pi j}\oint_C \left[\frac{0.3555}{(z - \frac{1}{3})(1 - 0.6z)}\right] dz$$

There is a single pole inside the unit circle at $z = \frac{1}{3}$. By evaluating the residue at the pole, we obtain

$$\text{MMSE}_\infty = 0.444$$

We observe that this MMSE is only slightly smaller than that for the optimum two-tap Wiener filter in Example 12.7.1.

12.7.4 Noncausal Wiener Filter

In the preceding section we constrained the optimum Wiener filter to be causal [i.e., $h_{\text{opt}}(n) = 0$ for $n < 0$]. In this section we drop this condition and allow the filter to include both the infinite past and the infinite future of the sequence $\{x(n)\}$ in forming the output $y(n)$, that is,

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{12.7.38}$$

The resulting filter is physically unrealizable. It can also be viewed as a *smoothing filter* in which the infinite future signal values are used to smooth the estimate $\hat{d}(n) = y(n)$ of the desired signal $d(n)$.

Application of the orthogonality principle yields the Wiener–Hopf equation for the noncausal filter in the form

$$\sum_{k=-\infty}^{\infty} h(k)\gamma_{xx}(l-k) = \gamma_{dx}(l), \qquad -\infty < l < \infty \tag{12.7.39}$$

and the resulting MMSE_{nc} as

$$\text{MMSE}_{\text{nc}} = \sigma_d^2 - \sum_{k=-\infty}^{\infty} h(k)\gamma_{dx}^*(k) \tag{12.7.40}$$

Since (12.7.39) holds for $-\infty < l < \infty$, this equation can be transformed directly to yield the optimum noncausal Wiener filter as

$$H_{\text{nc}}(z) = \frac{\Gamma_{dx}(z)}{\Gamma_{xx}(z)} \tag{12.7.41}$$

The MMSE_{nc} can also be simply expressed in the z-domain as

$$\text{MMSE}_{\text{nc}} = \frac{1}{2\pi j}\oint_C [\Gamma_{dd}(z) - H_{\text{nc}}(z)\Gamma_{dx}(z^{-1})]z^{-1}dz \tag{12.7.42}$$

In the following example we compare the form of the optimal noncausal filter with the optimal causal filter obtained in the previous section.

EXAMPLE 12.7.4

The optimum noncausal Wiener filter for the signal characteristics given in Example 12.7.1 is given by (12.7.41), where

$$\Gamma_{dx}(z) = \Gamma_{ss}(z) = \frac{0.64}{(1-0.6z^{-1})(1-0.6z)}$$

and

$$\begin{aligned}\Gamma_{xx}(z) &= \Gamma_{ss}(z) + 1 \\ &= \frac{2(1-0.3z^{-1}-0.3z)}{(1-0.6z^{-1})(1-0.6z)}\end{aligned}$$

Then,

$$H_{nc}(z) = \frac{0.3555}{(1 - \frac{1}{3}z^{-1})(1 - \frac{1}{3}z)}$$

This filter is clearly noncausal.

The minimum MSE achieved by this filter is determined from evaluating (12.7.42). The integrand is

$$\frac{1}{z}\Gamma_{ss}(z)[1 - H_{nc}(z)] = \frac{0.3555}{(z - \frac{1}{3})(1 - \frac{1}{3}z)}$$

The only pole inside the unit circle is $z = \frac{1}{3}$. Hence the residue is

$$\left.\frac{0.3555}{1 - \frac{1}{3}z}\right|_{z=\frac{1}{3}} = \frac{0.3555}{8/9} = 0.40$$

Hence the minimum achievable MSE obtained with the optimum noncausal Wiener filter is

$$\text{MMSE}_{nc} = 0.40$$

Note that this is lower than the MMSE for the causal filter, as expected.

12.8 Summary and References

The major focal point in this chapter is the design of optimum linear systems for linear prediction and filtering. The criterion for optimality is the minimization of the mean-square error between a specified desired filter output and the actual filter output.

In the development of linear prediction, we demonstrated that the equations for the forward and backward prediction errors specified a lattice filter whose parameters, the reflection coefficients $\{K_m\}$, were simply related to the filter coefficients $\{a_m(k)\}$ of the direct-form FIR linear predictor and the associated prediction-error filter. The optimum filter coefficients $\{K_m\}$ and $\{a_m(k)\}$ are easily obtained from the solution of the normal equations.

We described two computationally efficient algorithms for solving the normal equations, the Levinson–Durbin algorithm and the Schur algorithm. Both algorithms are suitable for solving a Toeplitz system of linear equations and have a computational complexity of $O(p^2)$ when executed on a single processor. However, with full parallel processing, the Schur algorithm solves the normal equations in $O(p)$ time, whereas the Levinson–Durbin algorithm requires $O(p \log p)$ time.

In addition to the all-zero lattice filter resulting from linear prediction, we also derived the AR lattice (all-pole) filter structure and the ARMA lattice-ladder (pole–zero) filter structure. Finally, we described the design of the class of optimum linear filters, called Wiener filters.

Linear estimation theory has had a long and rich history of development over the past four decades. Kailath (1974) presents a historical account of the first three

decades. The pioneering work of Wiener (1949) on optimum linear filtering for statistically stationary signals is especially significant. The generalization of the Wiener filter theory to dynamical systems with random inputs was developed by Kalman (1960) and Kalman and Bucy (1961). Kalman filters are treated in the books by Meditch (1969), Brown (1983), and Chui and Chen (1987). The monograph by Kailath (1981) treats both Wiener and Kalman filters.

There are numerous references on linear prediction and lattice filters. Tutorial treatments on these subjects have been published in the journal papers by Makhoul (1975, 1978) and Friedlander (1982a, b). The books by Haykin (1991), Markel and Gray 1976), and Tretter (1976) provide comprehensive treatments of these subjects. Applications of linear prediction to spectral analysis are found in the books by Kay (1988) and Marple (1987), to geophysics in the book by Robinson and Treitel (1980), and to adaptive filtering in the book by Haykin (1991).

The Levinson–Durbin algorithm for solving the normal equations recursively was given by Levinson (1947) and later modified by Durbin (1959). Variations of this classical algorithm, called *split Levinson algorithms*, have been developed by Delsarte and Genin (1986) and by Krishna (1988). These algorithms exploit additional symmetries in the Toeplitz correlation matrix and save about a factor of 2 in the number of multiplications.

The Schur algorithm was originally described by Schur (1917) in a paper published in German. An English translation of this paper appears in the book edited by Gohberg (1986). The Schur algorithm is intimately related to the polynomials $\{A_m(z)\}$, which can be interpreted as orthogonal polynomials. A treatment of orthogonal polynomials is given in the books by Szegö (1967), Grenander and Szegö (1958), and Geronimus (1958). The thesis of Vieira (1977) and the papers by Kailath et al. (1978), Delsarte et al. (1978), and Youla and Kazanjian (1978) provide additional results on orthogonal polynomials. Kailath (1985, 1986) provides tutorial treatments of the Schur algorithm and its relationship to orthogonal polynomials and the Levinson–Durbin algorithm. The pipelined parallel processing structure for computing the reflection coefficients based on the Schur algorithm and the related problem of solving Toeplitz systems of linear equations is described in the paper by Kung and Hu (1983). Finally, we should mention that some additional computational efficiency can be achieved in the Schur algorithm, by further exploiting symmetry properties of Toeplitz matrices, as described by Krishna (1988). This leads to the so-called split-Schur algorithm, which is analogous to the split-Levinson algorithm.

Problems

12.1 The power density spectrum of an AR process $\{x(n)\}$ is given as

$$\Gamma_{xx}(\omega) = \frac{\sigma_w^2}{|A(\omega)|^2}$$

$$= \frac{25}{|1 - e^{-j\omega} + \frac{1}{2}e^{-j2\omega}|^2}$$

where σ_w^2 is the variance of the input sequence.

(a) Determine the difference equation for generating the AR process when the excitation is white noise.

(b) Determine the system function for the whitening filter.

An ARMA process has an autocorrelation $\{\gamma_{xx}(m)\}$ whose z-transform is given as

$$\Gamma_{xx}(z) = 9\frac{(z-\frac{1}{3})(z-3)}{(z-\frac{1}{2})(z-2)}, \quad \tfrac{1}{2} < |z| < 2$$

(a) Determine the filter $H(z)$ for generating $\{x(n)\}$ from a white noise input sequence. Is $H(z)$ unique? Explain.

(b) Determine a stable linear whitening filter for the sequence $\{x(n)\}$.

Consider the ARMA process generated by the difference equation

$$x(n) = 1.6x(n-1) - 0.63x(n-2) + w(n) + 0.9w(n-1)$$

(a) Determine the system function of the whitening filter and its poles and zeros.

(b) Determine the power density spectrum of $\{x(n)\}$.

Determine the lattice coefficients corresponding to the FIR filter with system function

$$H(z) = A_3(z) = 1 + \tfrac{13}{24}z^{-1} + \tfrac{5}{8}z^{-2} + \tfrac{1}{3}z^{-3}$$

Determine the reflection coefficients $\{K_m\}$ of the lattice filter corresponding to the FIR filter described by the system function

$$H(z) = A_2(z) = 1 + 2z^{-1} + \tfrac{1}{3}z^{-2}$$

(a) Determine the zeros and sketch the zero pattern for the FIR lattice filter with reflection coefficients

$$K_1 = \tfrac{1}{2}, \qquad K_2 = -\tfrac{1}{3}, \qquad K_3 = 1$$

(b) Repeat part (a) but with $K_3 = -1$.

(c) You should have found that the zeros lie on the unit circle. Can this result be generalized? How?

Determine the impulse response of the FIR filter that is described by the lattice coefficients $K_1 = 0.6$, $K_2 = 0.3$, $K_3 = 0.5$, and $K_4 = 0.9$.

In Section 12.3.4 we indicated that the noise-whitening filter $A_p(z)$ for a causal AR(p) process is a forward linear prediction-error filter of order p. Show that the backward linear prediction-error filter of order p is the noise-whitening filter of the corresponding anticausal AR(p) process.

Use the orthogonality principle to determine the normal equations and the resulting minimum MSE for a forward predictor of order p that predicts m samples ($m > 1$) into the future (m-step forward predictor). Sketch the prediction-error filter.

Repeat Problem 12.9 for an m-step backward predictor.

12.11 Determine a Levinson–Durbin recursive algorithm for solving for the coefficients of a backward prediction-error filter. Use the result to show that coefficients of the forward and backward predictors can be expressed recursively as

$$\mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + K_m \begin{bmatrix} \mathbf{b}_{m-1} \\ 1 \end{bmatrix}$$

$$\mathbf{b}_m = \begin{bmatrix} \mathbf{b}_{m-1} \\ 0 \end{bmatrix} + K_m^* \begin{bmatrix} \mathbf{a}_{m-1} \\ 1 \end{bmatrix}$$

12.12 The Levinson–Durbin algorithm described in Section 12.4.1 solved the linear equations

$$\mathbf{\Gamma}_m \mathbf{a}_m = -\boldsymbol{\gamma}_m$$

where the right-hand side of this equation has elements of the autocorrelation sequence that are also elements of the matrix $\mathbf{\Gamma}$. Let us consider the more general problem of solving the linear equations

$$\mathbf{\Gamma}_m \mathbf{b}_m = \mathbf{c}_m$$

where $\mathbf{c}_m$ is an arbitrary vector. (The vector $\mathbf{b}_m$ is not related to the coefficients of the backward predictor.) Show that the solution to $\mathbf{\Gamma}_m \mathbf{b}_m = \mathbf{c}_m$ can be obtained from a *generalized Levinson–Durbin* algorithm which is given recursively as

$$b_m(m) = \frac{c(m) - \boldsymbol{\gamma}_{m-1}^{bt} \mathbf{b}_{m-1}}{E_{m-1}^f}$$

$$b_m(k) = b_{m-1}(k) - b_m(m) a_{m-1}^*(m-k), \qquad \begin{matrix} k = 1, 2, \ldots, m-1 \\ m = 1, 2, \ldots, p \end{matrix}$$

where $b_1(1) = c(1)/\gamma_{xx}(0) = c(1)/E_0^f$ and $a_m(k)$ is given by (12.4.17). Thus a second recursion is required to solve the equation $\mathbf{\Gamma}_m \mathbf{b}_m = \mathbf{c}_m$.

12.13 Use the generalized Levinson–Durbin algorithm to solve the normal equations recursively for the m-step forward and backward predictors.

12.14 Consider the AR(3) process generated by the equation

$$x(n) = \tfrac{14}{24} x(n-1) + \tfrac{9}{24} x(n-2) - \tfrac{1}{24} x(n-3) + w(n)$$

where $w(n)$ is a stationary white noise process with variance σ_w^2.

(a) Determine the coefficients of the optimum $p = 3$ linear predictor.

(b) Determine the autocorrelation sequence $\gamma_{xx}(m)$, $0 \leq m \leq 5$.

(c) Determine the reflection coefficients corresponding to the $p = 3$ linear predictor.

12.15 The z-transform of the autocorrelation $\gamma_{xx}(m)$ of an ARMA(1, 1) process is

$$\Gamma_{xx}(z) = \sigma_w^2 H(z) H(z^{-1})$$

$$\Gamma_{xx}(z) = \frac{4\sigma_w^2}{9} \frac{5 - 2z - 2z^{-1}}{10 - 3z^{-1} - 3z}$$

(a) Determine the minimum-phase system function $H(z)$.

(b) Determine the system function $H(z)$ for a mixed-phase stable system.

12.16 Consider an FIR filter with coefficient vector

$$[1 \quad -2r\cos\theta \quad r^2]$$

(a) Determine the reflection coefficients for the corresponding FIR lattice filter.

(b) Determine the values of the reflection coefficients in the limit as $r \to 1$.

12.17 An AR(3) process is characterized by the prediction coefficients

$$a_3(1) = -1.25, \qquad a_3(2) = 1.25, \qquad a_3(3) = -1$$

(a) Determine the reflection coefficients.

(b) Determine $\gamma_{xx}(m)$ for $0 \leq m \leq 3$.

(c) Determine the mean-square prediction error.

12.18 The autocorrelation sequence for a random process is

$$\gamma_{xx}(m) = \begin{cases} 1, & m = 0 \\ -0.5, & m = \pm 1 \\ 0.625, & m = \pm 2 \\ -0.6875, & m = \pm 3 \\ 0 & \text{otherwise} \end{cases}$$

Determine the system functions $A_m(z)$ for the prediction-error filters for $m = 1, 2, 3$, the reflection coefficients $\{K_m\}$, and the corresponding mean-square prediction errors.

12.19 The autocorrelation sequence for an AR process $x(n)$ is

$$\gamma_{xx}(m) = (\tfrac{1}{4})^{|m|}$$

(a) Determine the difference equation for $x(n)$.

(b) Is your answer unique? If not, give any other possible solutions.

12.20 Repeat Problem 12.19 for an AR process with autocorrelation

$$\gamma_{xx}(m) = a^{|m|}\cos\frac{\pi m}{2}$$

where $0 < a < 1$.

12.21 Prove that a FIR filter with system function

$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k)z^{-k}$$

and reflection coefficients $|K_k| < 1$ for $1 \leq k \leq p-1$ and $|K_p| > 1$ is maximum phase [all the roots of $A_p(z)$ lie outside the unit circle].

12.22 Show that the transformation

$$\mathbf{V}_m = \begin{bmatrix} 1 & K_m \\ K_m^* & 1 \end{bmatrix}$$

in the Schur algorithm satisfies the special property

$$\mathbf{V}_m \mathbf{J} \mathbf{V}_m^t = (1 - |K_m|^2)\mathbf{J}$$

where

$$\mathbf{J} = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Thus $\mathbf{V}_m$ is called a J-rotation matrix. Its role is to rotate or hyperbolate the row of $\mathbf{G}_m$ to lie along the first coordinate direction (Kailath, 1985).

12.23 Prove the additional properties (a) through (l) of the prediction-error filters given in Section 12.5.

12.24 Extend the additional properties (a) through (l) of the prediction-error filters given in Section 12.5 to complex-valued signals.

12.25 Determine the reflection coefficient K_3 in terms of the autocorrelations $\{\gamma_{xx}(m)\}$ from the Schur algorithm and compare your result with the expression for K_3 obtained from the Levinson–Durbin algorithm.

12.26 Consider an infinite-length ($p = \infty$) one-step forward predictor for a stationary random process $\{x(n)\}$ with a power density spectrum of $\Gamma_{xx}(f)$. Show that the mean-square error of the prediction-error filter can be expressed as

$$E_\infty^f = 2\pi \exp\{\int_{-1/2}^{1/2} \ln \Gamma_{xx}(f)\, df\}$$

12.27 Determine the output of an infinite-length ($p = \infty$) m-step forward predictor and the resulting mean-square error when the input signal is a first-order autoregressive process of the form

$$x(n) = ax(n-1) + w(n)$$

12.28 An AR(3) process $\{x(n)\}$ is characterized by the autocorrelation sequence $\gamma_{xx}(0) = 1$, $\gamma_{xx}(1) = \frac{1}{2}$, $\gamma_{xx}(2) = \frac{1}{8}$, and $\gamma_{xx}(3) = \frac{1}{64}$.

(a) Use the Schur algorithm to determine the three reflection coefficients K_1, K_2, and K_3.

(b) Sketch the lattice filter for synthesizing $\{x(n)\}$ from a white noise excitation.

12.29 The purpose of this problem is to show that the polynomials $\{A_m(z)\}$, which are the system functions of the forward prediction-error filters of order m, $m = 0, 1, \ldots, p$, can be interpreted as orthogonal on the unit circle. Toward this end, suppose that $\Gamma_{xx}(f)$ is the power spectral density of a zero-mean random process $\{x(n)\}$ and let $\{A_m(z)\}$, $m = 0, 1, \ldots, p\}$, be the system functions of the corresponding prediction-error filters. Show that the polynomials $\{A_m(z)\}$ satisfy the orthogonality property

$$\int_{-1/2}^{1/2} \Gamma_{xx}(f) A_m(f) A_n^*(f) df = E_m^f \delta_{mn}, \qquad m, n = 0, 1, \ldots, p$$

12.30 Determine the system function of the all-pole filter described by the lattice coefficients $K_1 = 0.6$, $K_2 = 0.3$, $K_3 = 0.5$, and $K_4 = 0.9$.

12.31 Determine the parameters and sketch the lattice-ladder filter structure for the system with system function

$$H(z) = \frac{1 - 0.8z^{-1} + 0.15z^{-2}}{1 + 0.1z^{-1} - 0.72z^{-2}}$$

12.32 Consider a signal $x(n) = s(n) + w(n)$, where $s(n)$ is an AR(1) process that satisfies the difference equation

$$s(n) = 0.8s(n-1) + v(n)$$

where $\{v(n)\}$ is a white noise sequence with variance $\sigma_v^2 = 0.49$ and $\{w(n)\}$ is a white noise sequence with variance $\sigma_w^2 = 1$. The processes $\{v(n)\}$ and $\{w(n)\}$ are uncorrelated.

(a) Determine the autocorrelation sequences $\{\gamma_{ss}(m)\}$ and $\{\gamma_{xx}(m)\}$.

(b) Design a Wiener filter of length $M = 2$ to estimate $\{s(n)\}$.

(c) Determine the MMSE for $M = 2$.

12.33 Determine the optimum causal IIR Wiener filter for the signal given in Problem 12.32 and the corresponding MMSE_∞.

12.34 Determine the system function for the noncausal IIR Wiener filter for the signal given in Problem 12.32 and the corresponding MMSE_{nc}.

12.35 Determine the optimum FIR Wiener filter of length $M = 3$ for the signal in Example 12.7.1 and the corresponding MMSE_3. Compare MMSE_3 with MMSE_2 and comment on the difference.

12.36 An AR(2) process is defined by the difference equation

$$x(n) = x(n-1) - 0.6x(n-2) + w(n)$$

where $\{w(n)\}$ is a white noise process with variance σ_w^2. Use the Yule–Walker equations to solve for the values of the autocorrelation $\gamma_{xx}(0)$, $\gamma_{xx}(1)$, and $\gamma_{xx}(2)$.

12.37 An observed random process $\{x(n)\}$ consists of the sum of an AR(p) process of the form

$$s(n) = -\sum_{k=1}^{p} a_p(k)s(n-k) + v(n)$$

and a white noise process $\{w(n)\}$ with variance σ_w^2. The random process $\{v(n)\}$ is also white with variance σ_v^2. The sequences $\{v(n)\}$ and $\{w(n)\}$ are uncorrelated.

Show that the observed process $\{x(n) = s(n) + w(n)\}$ is ARMA(p, p) and determine the coefficients of the numerator polynomial (MA component) in the corresponding system function.

CHAPTER

13

Adaptive Filters

In contrast to filter design techniques described in Chapter 12, which were based on knowledge of the second-order statistics of the signals, there are many digital signal processing applications in which these statistics cannot be specified a priori. Such applications include channel equalization, echo cancellation, and system modeling among others, as described in this chapter. In these applications, filters with adjustable coefficients, called *adaptive filters*, are employed. Such filters incorporate algorithms that allow the filter coefficients to adapt to the signal statistics.

Adaptive filters have received considerable attention from researchers over the past 25 years. As a result, many computationally efficient algorithms for adaptive filtering have been developed. In this chapter, we describe two basic algorithms: the least-mean-square (LMS) algorithm, which is based on a gradient optimization for determining the coefficients, and the class of recursive least-squares algorithms, which includes both direct-form FIR and lattice realizations. Before we describe the algorithms, we present several practical applications in which adaptive filters have been successfully used in the estimation of signals corrupted by noise and other interference.

13.1 Applications of Adaptive Filters

Adaptive filters have been used widely in communication systems, control systems, and various other systems in which the statistical characteristics of the signals to be filtered are either unknown a priori or, in some cases, are slowly time-variant (nonstationary signals). Numerous applications of adaptive filters have been described in the literature. Some of the more noteworthy applications include: (1) adaptive antenna systems, in which adaptive filters are used for beam steering and for providing

nulls in the beam pattern to remove undesired interference (Widrow, Mantey, and Griffiths (1967)); (2) digital communication receivers, in which adaptive filters are used to provide equalization of intersymbol interference and for channel identification (Lucky (1965), Proakis and Miller (1969), Gersho (1969), George, Bowen, and Storey (1971), Proakis (1970; 1975), Magee and Proakis (1973), Picinbono (1978) and Nichols, Giordano, and Proakis (1977)); (3) adaptive noise cancelling techniques, in which adaptive filters are used to estimate and eliminate a noise component in a desired signal (Widrow et al. (1975), Hsu and Giordano (1978), and Ketchum and Proakis (1982)); (4) system modeling, in which adaptive filters are used as models to estimate the characteristics of an unknown system. These are just a few of the best-known examples of the use of adaptive filters.

Although both IIR and FIR filters have been considered for adaptive filtering, the FIR filter is by far the most practical and widely used. The reason for this preference is quite simple: the FIR filter has only adjustable zeros; hence, it is free of stability problems associated with adaptive IIR filters, which have adjustable poles as well as zeros. We should not conclude, however, that adaptive FIR filters are always stable. On the contrary, the stability of the filter depends critically on the algorithm for adjusting its coefficients, as will be demonstrated in Sections 13.2 and 13.3.

Of the various FIR filter structures that are possible, the direct form and the lattice form are the ones used in adaptive filtering applications. The direct-form FIR filter structure with adjustable coefficients $h(n)$ is illustrated in Figure 13.1.1. On the other hand, the adjustable parameters in an FIR lattice structure are the reflection coefficients K_n.

An important consideration in the use of an adaptive filter is the criterion for optimizing the adjustable filter parameters. The criterion must not only provide a meaningful measure of filter performance, but it must also result in a practically realizable algorithm.

For example, a desirable performance index in a digital communication system is the average probability of error. Consequently, in implementing an adaptive equalizer, we might consider the selection of the equalizer coefficients to minimize the average probability of error as the basis for our optimization criterion. Unfortunately,

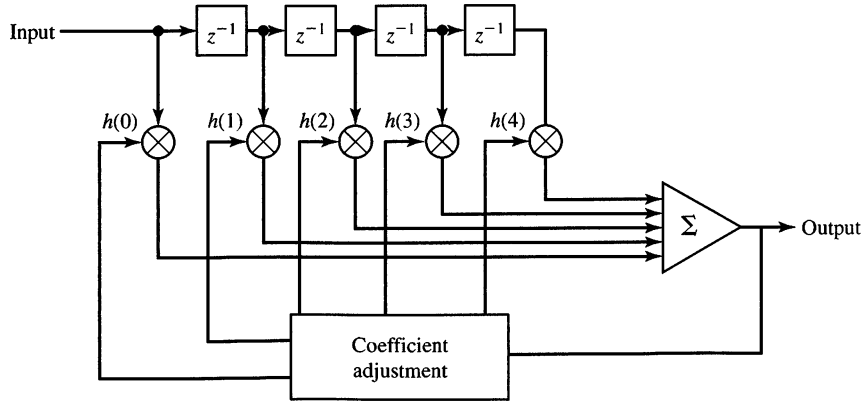


Figure 13.1.1 Direct-form adaptive FIR filter.

however, the performance index (average probability of error) for this criterion is a highly nonlinear function of the filter coefficients and the signal statistics. As a consequence, the implementation of an adaptive filter that optimizes such a performance index is complex and impractical.

In some cases, a performance index that is a nonlinear function of the filter parameters possesses many relative minima (or maxima), so that one is not certain whether the adaptive filter has converged to the optimum solution or to one of the relative minima (or maxima). For such reasons, some desirable performance indices, such as the average probability of error in a digital communication system, must be rejected on the grounds that they are impractical to implement.

Two criteria that provide good measures of performance in adaptive filtering applications are the least-squares criterion and its counterpart in a statistical formulation of the problem, namely, the mean-square-error (MSE) criterion. The least-squares (and MSE) criterion results in a quadratic performance index as a function of the filter coefficients and, hence, it possesses a single minimum. The resulting algorithms for adjusting the coefficients of the filter are relatively easy to implement, as we demonstrate in Sections 13.2 and 13.3.

In the following section, we describe several applications of adaptive filters that serve as a motivation for the mathematical development of algorithms derived in Sections 13.2 and 13.3. We find it convenient to use the direct-form FIR structure in these examples. Although we will not develop the recursive algorithms for automatically adjusting the filter coefficients in this section, it is instructive to formulate the optimization of the filter coefficients as a least-squares optimization problem. This development will serve to establish a common framework for the algorithms derived in the next two sections.

13.1.1 System Identification or System Modeling

In the formulation of this problem we have an unknown system, called a *plant*, that we wish to identify. The system is modeled by an FIR filter with M adjustable coefficients. Both the plant and model are excited by an input sequence $x(n)$. If $y(n)$ denotes the output of the plant and $\hat{y}(n)$ denotes the output of the model,

$$\hat{y}(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{13.1.1}$$

We may form the error sequence

$$e(n) = y(n) - \hat{y}(n), \qquad n = 0, 1, \ldots \tag{13.1.2}$$

and select the coefficients $h(k)$ to minimize

$$\mathscr{E}_M = \sum_{n=0}^{N} \left[y(n) - \sum_{k=0}^{M-1} h(k)x(n-k) \right]^2 \tag{13.1.3}$$

where $N + 1$ is the number of observations.

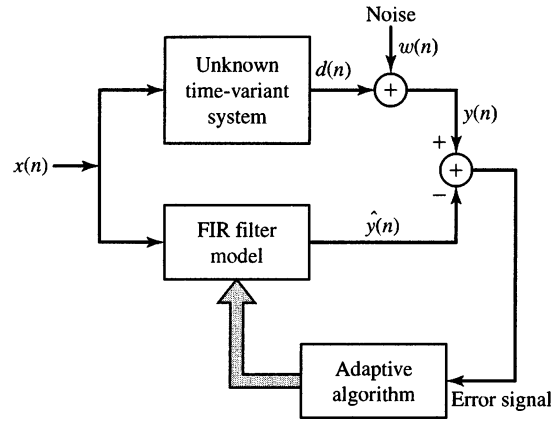


Figure 13.1.2
Application of adaptive filtering to system identification.

The least-squares criterion leads to the set of linear equations for determining the filter coefficients, namely,

$$\sum_{k=0}^{M-1} h(k)r_{xx}(l-k) = r_{yx}(l), \qquad l = 0, 1, \ldots, M-1 \tag{13.1.4}$$

In (13.1.4), $r_{xx}(l)$ is the autocorrelation of the sequence $x(n)$ and $r_{yx}(l)$ is the cross-correlation of the system output with the input sequence.

By solving (13.1.4), we obtain the filter coefficients for the model. Since the filter parameters are obtained directly from measurement data at the input and output of the system, without prior knowledge of the plant, we call the FIR filter model an adaptive filter.

If our only objective were to identify the system by use of the FIR model, the solution of (13.1.4) would suffice. In control systems applications, however, the system being modeled may be time variant, changing slowly with time, and our purpose for having a model is to ultimately use it for designing a controller that controls the plant. Furthermore, measurement noise is usually present at the output of the plant. This noise introduces uncertainty in the measurements and corrupts the estimates of the filter coefficients in the model. Such a scenario is illustrated in Figure 13.1.2. In this case, the adaptive filter must identify and track the time-variant characteristics of the plant in the presence of measurement noise at the output of the plant. The algorithms to be described in Sections 13.2 and 13.3 are applicable to this system identification problem.

13.1.2 Adaptive Channel Equalization

Figure 13.1.3 shows a block diagram of a digital communication system in which an adaptive equalizer is used to compensate for the distortion caused by the transmission medium (channel). The digital sequence of information symbols $a(n)$ is fed to the transmitting filter, whose output is

$$s(t) = \sum_{k=0}^{\infty} a(k)p(t - kT_s) \tag{13.1.5}$$

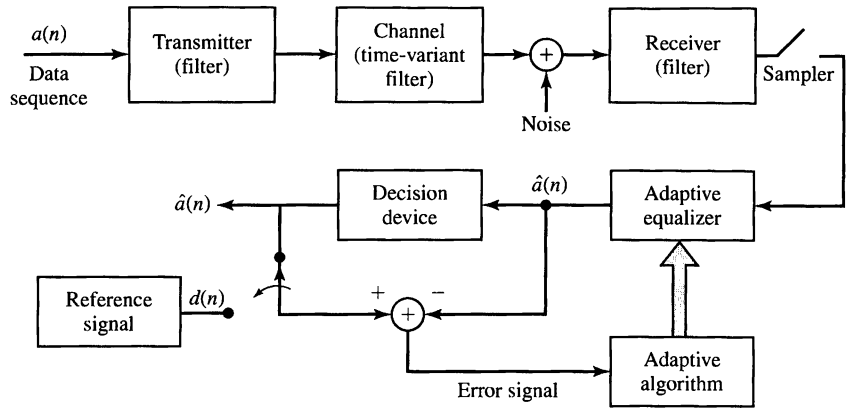


Figure 13.1.3 Application of adaptive filtering to adaptive channel equalization.

where $p(t)$ is the impulse response of the filter at the transmitter and T_s is the time interval between information symbols; that is, $1/T_s$ is the symbol rate. For purposes of this discussion, we may assume that $a(n)$ is a multilevel sequence that takes on values from the set $\pm 1, \pm 3, \pm 5, \ldots, \pm(K-1)$, where K is the number of possible symbol values.

Typically, the pulse $p(t)$ is designed to have the characteristics illustrated in Figure 13.1.4. Note that $p(t)$ has amplitude $p(0) = 1$ at $t = 0$ and $p(nT_s) = 0$ at $t = nT_s$, $n = \pm 1, \pm 2, \ldots$. As a consequence, successive pulses transmitted sequentially every T_s seconds do not interfere with one another when sampled at the time instants $t = nT_s$. Thus, $a(n) = s(nT_s)$.

The channel, which is usually well modeled as a linear filter, distorts the pulse and, thus, causes intersymbol interference. For example, in telephone channels, filters are used throughout the system to separate signals in different frequency ranges. These filters cause phase and amplitude distortion. Figure 13.1.5 illustrates the effect of channel distortion on the pulse $p(t)$ as it might appear at the output of a telephone channel. Now, we observe that the samples taken every T_s seconds are corrupted by

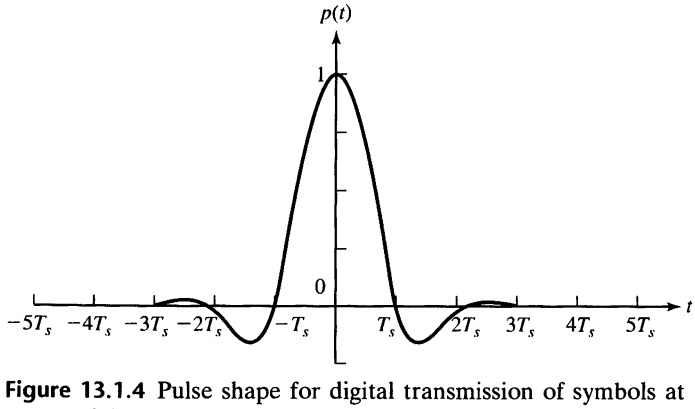


Figure 13.1.4 Pulse shape for digital transmission of symbols at a rate of $1/T_s$ symbols per second.

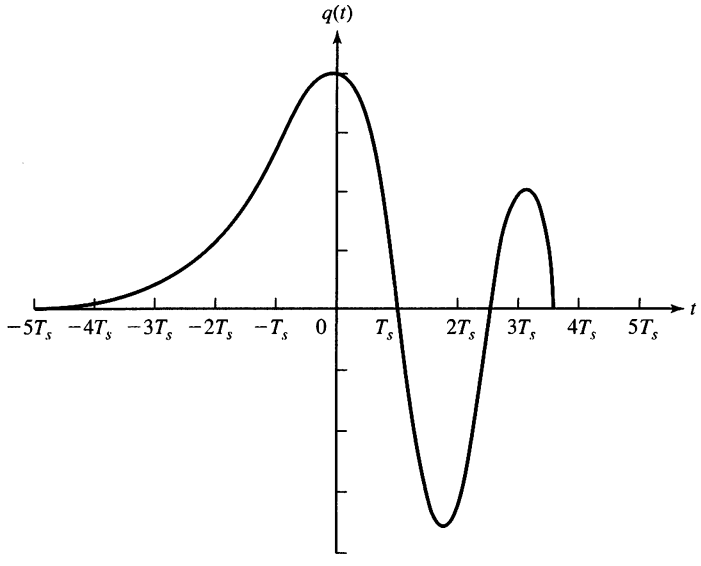


Figure 13.1.5 Effect of channel distortion on the signal pulse in Figure 13.1.4.

interference from several adjacent symbols. The distorted signal is also corrupted by additive noise, which is usually wideband.

At the receiving end of the communication system, the signal is first passed through a filter that is designed primarily to eliminate the noise outside of the frequency band occupied by the signal. We may assume that this filter is a linear-phase FIR filter that limits the bandwidth of the noise but causes negligible additional distortion on the channel-corrupted signal.

Samples of the received signal at the output of this filter reflect the presence of intersymbol interference and additive noise. If we ignore for the moment the possible time variations in the channel, we may express the sampled output at the receiver as

$$\begin{aligned} x(nT_s) &= \sum_{k=0}^{\infty} a(k)q(nT_s - kT_s) + w(nT_s) \\ &= a(n)q(0) + \sum_{\substack{k=0 \\ k\neq n}}^{\infty} a(k)q(nT_s - kT_s) + w(nT_s) \end{aligned} \tag{13.1.6}$$

where $w(t)$ represents the additive noise and $q(t)$ represents the distorted pulse at the output of the receiver filter.

To simplify our discussion, we assume that the sample $q(0)$ is normalized to unity by means of an automatic gain control (AGC) contained in the receiver. Then, the

sampled signal given in (13.1.6) may be expressed as

$$x(n) = a(n) + \sum_{\substack{k=0 \\ k \neq n}}^{\infty} a(k)q(n-k) + w(n) \tag{13.1.7}$$

where $x(n) \equiv x(nT_s)$, $q(n) \equiv q(nT_s)$, and $w(n) \equiv w(nT_s)$. The term $a(n)$ in (13.1.7) is the desired symbol at the nth sampling instant. The second term,

$$\sum_{\substack{k=0 \\ k \neq n}}^{\infty} a(k)q(n-k)$$

constitutes the intersymbol interference due to the channel distortion, and $w(n)$ represents the additive noise in the system.

In general, the channel distortion effects embodied through the sampled values $q(n)$ are unknown at the receiver. Furthermore, the channel may vary slowly with time such that the intersymbol interference effects are time variant. The purpose of the adaptive equalizer is to compensate the signal for the channel distortion, so that the resulting signal can be detected reliably. Let us assume that the equalizer is an FIR filter with M adjustable coefficients $h(n)$. Its output may be expressed as

$$\hat{a}(n) = \sum_{k=0}^{M-1} h(k)x(n + D - k) \tag{13.1.8}$$

where D is some nominal delay in processing the signal through the filter and $\hat{a}(n)$ represents an estimate of the nth information symbol. Initially, the equalizer is trained by transmitting a known data sequence $d(n)$. Then, the equalizer output, $\hat{a}(n)$, is compared with $d(n)$ and an error is generated that is used to optimize the filter coefficients.

If we again adopt the least-squares error criterion, we select the coefficients $h(k)$ to minimize the quantity

$$\mathscr{E}_M = \sum_{n=0}^{N} [d(n) - \hat{a}(n)]^2 = \sum_{n=0}^{N} \left[d(n) - \sum_{k=0}^{M-1} h(k)x(n + D - k) \right]^2 \tag{13.1.9}$$

The result of the optimization is a set of linear equations of the form

$$\sum_{k=0}^{M-1} h(k)r_{xx}(l-k) = r_{dx}(l-D), \qquad l = 0, 1, 2, \ldots, M-1 \tag{13.1.10}$$

where $r_{xx}(l)$ is the autocorrelation of the sequence $x(n)$ and $r_{dx}(l)$ is the crosscorrelation between the desired sequence $d(n)$ and the received sequence $x(n)$.

Although the solution of (13.1.10) is obtained recursively in practice (as demonstrated in the following two sections), in principle, we observe that these equations result in values of the coefficients for the initial adjustment of the equalizer. After the short training period, which usually lasts less than one second for most channels, the transmitter begins to transmit the information sequence $a(n)$. In order to track the possible time variations in the channel, the equalizer coefficients must continue to be adjusted in an adaptive manner while receiving data. As illustrated in Figure 13.1.3, this is usually accomplished by treating the decisions at the output of the decision device as correct, and using the decisions in place of the reference $d(n)$ to generate the error signal. This approach works quite well when decision errors occur infrequently (for example, less than one decision error per hundred symbols). The occasional decision errors cause only small misadjustments in the equalizer coefficients. In Sections 13.2 and 13.3, we describe the adaptive algorithms for recursively adjusting the equalizer coefficients.

13.1.3 Echo Cancellation in Data Transmission over Telephone Channels

In the transmission of data over telephone channels, modems (modulator/demodulators) are used to provide an interface between the digital data sequence and the analog channel. Shown in Figure 13.1.6 is a block diagram of a communication system in which two terminals, labeled A and B, transmit data by using modems A and B to interface to a telephone channel. As shown, a digital sequence $a(n)$ is transmitted from terminal A to terminal B while another digital sequence $b(n)$ is transmitted from terminal B to A. This simultaneous transmission in both directions is called *full-duplex transmission.*

As described, the two transmitted signals may be represented as

$$s_A(t) = \sum_{k=0}^{\infty} a(k)p(t - kT_s) \tag{13.1.11}$$

$$s_B(t) = \sum_{k=0}^{\infty} b(k)p(t - kT_s) \tag{13.1.12}$$

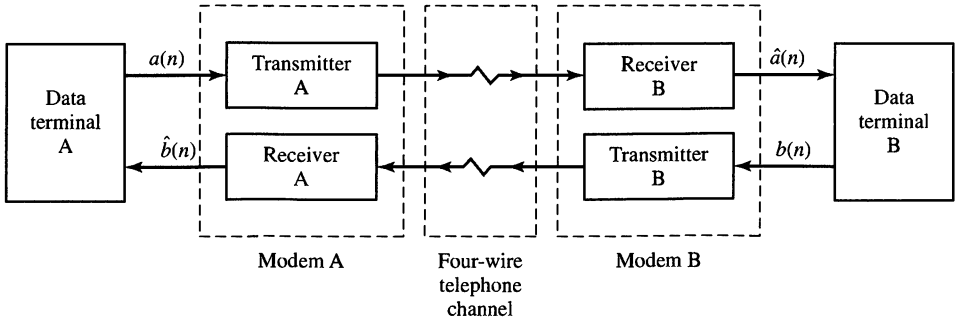


Figure 13.1.6 Full-duplex data transmission over telephone channels.

where $p(t)$ is a pulse as shown in Figure 13.1.4.

When a subscriber leases a private line from a telephone company for the purpose of transmitting data between terminals A and B, the telephone line provided is a four-wire line, which is equivalent to having two dedicated telephone (two-wire) channels, one (pair of wires) for transmitting data in one direction and one (pair of wires) for receiving data from the other direction. In such a case the two transmission paths are isolated and, consequently, there is no "crosstalk" or mutual interference between the two signal paths. Channel distortion is compensated by use of an adaptive equalizer, as previously described, at the receiver of each modem.

The major problem with the system shown in Figure 13.1.6 is the cost of leasing a four-wire telephone channel. If the volume of traffic is high and the telephone channel is used either continuously or frequently, as in banking transactions systems or airline reservation systems, the system pictured in Figure 13.1.6 may be cost effective. Otherwise, it will not be.

An alternative solution for low-volume, infrequent transmission of data is to use the dial-up switched telephone network. In this case, the local communication link between the subscriber and the local central telephone office is a two-wire line, called the *local loop*. At the central office, the subscriber two-wire line is connected to the main four-wire telephone channels that interconnect different central offices, called *trunk lines*, by a device called a *hybrid*. By using transformer coupling, the hybrid is tuned to provide isolation between the transmission and reception channels in full-duplex operation. However, due to impedance mismatch between the hybrid and the telephone channel, the level of isolation is often insufficient and, consequently, some of the signal on the transmitter side leaks back and corrupts the signal on the receiver side, causing an "echo" that is often heard in voice communications over telephone channels.

To mitigate the echoes in voice transmissions, the telephone companies employ a device called an *echo suppressor*. In data transmission, the solution is to use an *echo canceller* within each modem. The echo cancellers are implemented as adaptive filters with automatically adjustable coefficients, just as in the case of transversal equalizers.

With the use of hybrids to couple a two-wire to a four-wire channel, and echo cancellers at each modem to estimate and subtract the echoes, the data communication system for the dial-up switched network takes the form shown in Figure 13.1.7. A hybrid is needed at each modem to isolate the transmitter from the receiver and to couple to the two-wire local loop. Hybrid A is physically located at the central office of subscriber A while hybrid B is located at the central office to which subscriber B is connected. The two central offices are connected by a four-wire line, one pair used for transmission from A to B and the other pair used for transmission in the reverse direction, from B to A. An echo at terminal A due to the hybrid A is called a *near-end echo,* while an echo at terminal A due to the hybrid B is termed a *far-end echo*. Both types of echoes are usually present in data transmission and must be removed by the echo canceller.

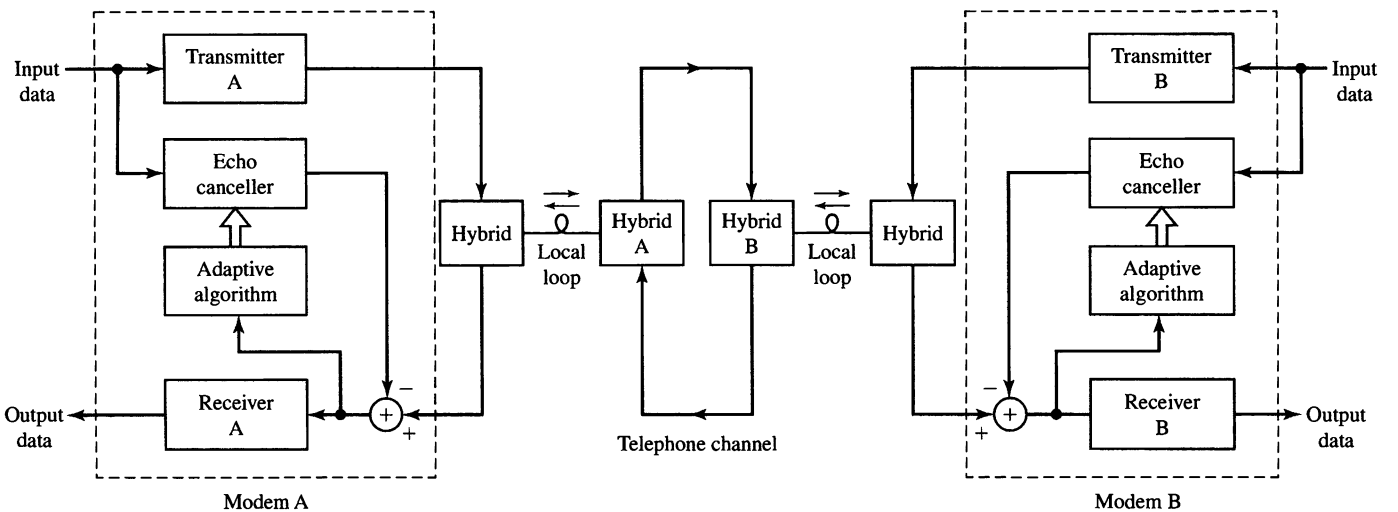


Figure 13.1.7 Block diagram model of a digital communication system that uses echo cancellers in the modems.

Suppose we neglect the channel distortion for purposes of this discussion, and deal with the echoes only. The signal received at modem A may be expressed as

$$s_{RA}(t) = A_1 s_B(t) + A_2 s_A(t - d_1) + A_3 s_A(t - d_2) \tag{13.1.13}$$

where $s_B(t)$ is the desired signal to be demodulated at modem A; $s_A(t-d_1)$ is the near-end echo due to hybrid A, $s_A(t-d_2)$ is the far-end echo due to hybrid B; A_i, $i = 1, 2, 3$, are the corresponding amplitudes of the three signal components; and d_1 and d_2 are the delays associated with the echo components. A further disturbance that corrupts the received signal is additive noise, so that the received signal at modem A is

$$r_A(t) = s_{RA}(t) + w(t) \tag{13.1.14}$$

where $w(t)$ represents the additive noise process.

The adaptive echo canceller attempts to estimate adaptively the two echo components. If its coefficients are $h(n)$, $n = 0, 1, \ldots, M-1$, its output is

$$\hat{s}_A(n) = \sum_{k=0}^{M-1} h(k)a(n-k) \tag{13.1.15}$$

which is an estimate of the echo signal components. This estimate is subtracted from the sampled received signal, and the resulting error signal can be minimized in the least-squares sense to optimally adjust the coefficients of the echo canceller. There are several possible configurations for placement of the echo canceller in the modem, and for forming the corresponding error signal. Figure 13.1.8 illustrates one configuration, in which the canceller output is subtracted from the sampled output of the receiver filter with input $r_A(t)$. Figure 13.1.9 illustrates a second configuration, in which the echo canceller is generating samples at the Nyquist rate instead of the symbol rate; in this case the error signal used to adjust the coefficients is simply the difference between $r_A(n)$, the sampled received signal, and the canceller output. Finally, Figure 13.1.10 illustrates the canceller operating in combination with an adaptive equalizer.

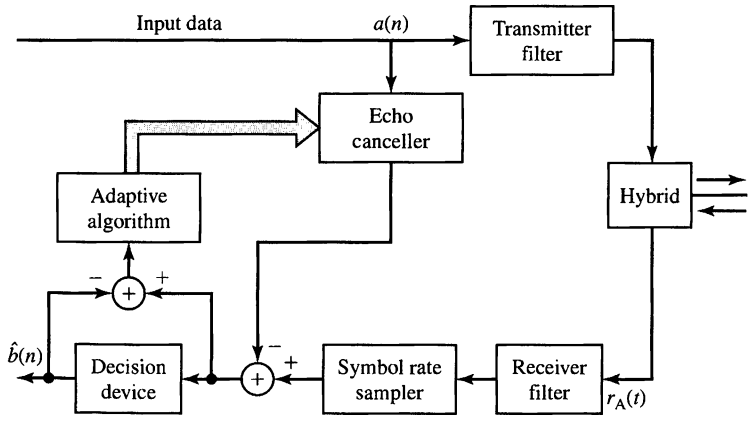


Figure 13.1.8 Symbol rate echo canceller.

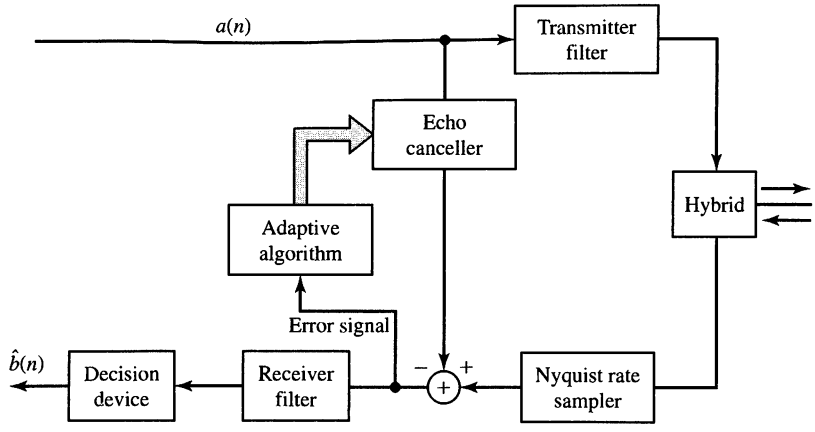


Figure 13.1.9 Nyquist rate echo canceller.

Application of the least-squares criterion in any of the configurations shown in Figures 13.1.8–13.1.10 leads to a set of linear equations for the coefficients of the echo canceller. The reader is encouraged to derive the equations corresponding to the three configurations.

13.1.4 Suppression of Narrowband Interference in a Wideband Signal

We now discuss a problem that arises in practice, especially in signal detection and in digital communications. Let us assume that we have a signal sequence $v(n)$ that

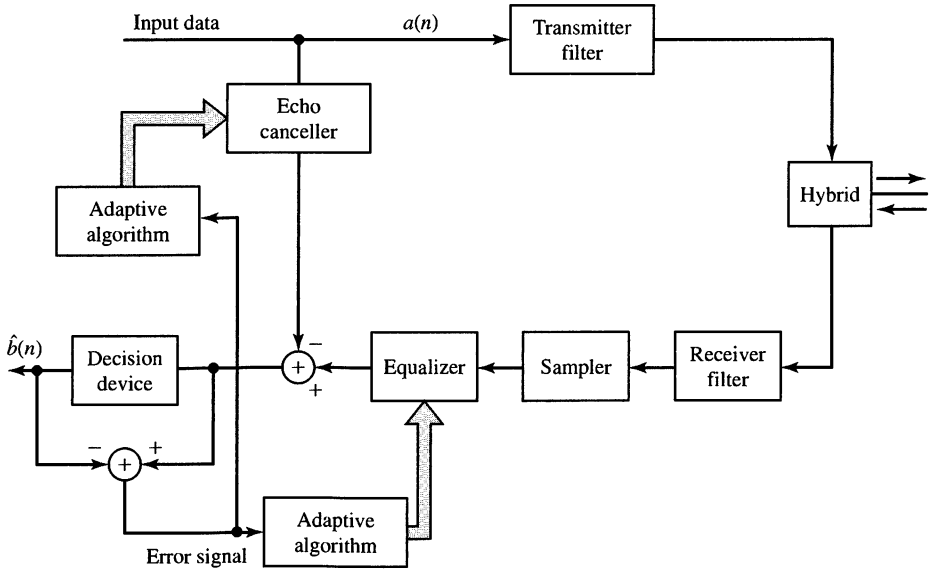


Figure 13.1.10 Modem with adaptive equalizer and echo canceller.

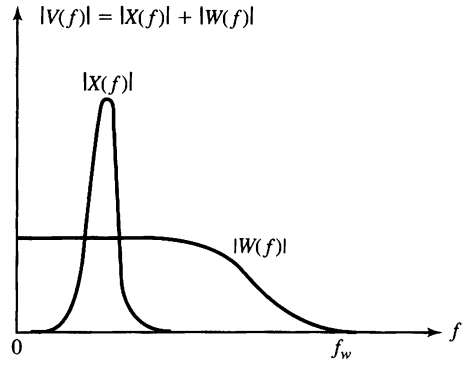


Figure 13.1.11
Strong narrowband interference $X(f)$ in a wideband signal $W(f)$.

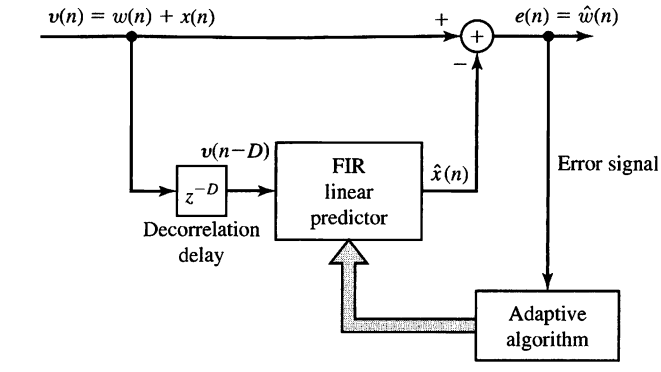
consists of a desired wideband signal sequence $w(n)$ corrupted by an additive narrowband interference sequence $x(n)$. The two sequences are uncorrelated. These sequences result from sampling an analog signal $v(t)$ at the Nyquist rate (or faster) of the wideband signal $w(t)$. Figure 13.1.11 illustrates the spectral characteristics of $w(n)$ and $x(n)$. Usually, the interference $|X(f)|$ is much larger than $|W(f)|$ within the narrow frequency band that it occupies.

In digital communications and signal detection problems that fit the above model, the desired signal sequence $w(n)$ is often a *spread-spectrum signal*, while the narrowband interference represents a signal from another user of the frequency band, or intentional interference from a jammer who is trying to disrupt the communications or detection system.

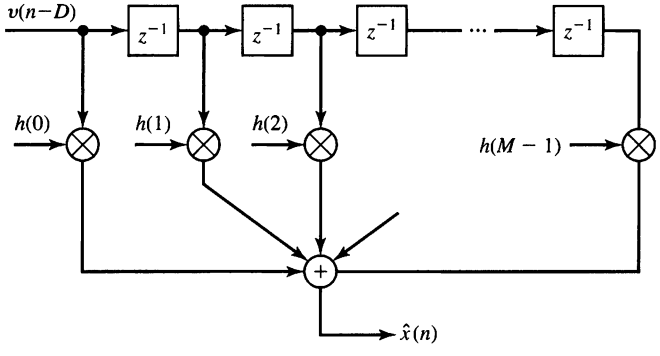
Our objective from a filtering viewpoint is to employ a filter that suppresses the narrowband interference. In effect, such a filter will have a notch in the frequency band occupied by $|X(f)|$, and in practice, the band occupied by $|X(f)|$ is unknown. Moreover, if the interference is nonstationary, its frequency band occupancy may vary with time. Hence, an adaptive filter is desired.

From another viewpoint, the narrowband characteristics of the interference allow us to estimate $x(n)$ from past samples of the sequence $v(n)$ and to subtract the estimate from $v(n)$. Since the bandwidth of $x(n)$ is narrow compared to the bandwidth of the sequence $w(n)$, the samples $x(n)$ are highly correlated due to the high sampling rate. On the other hand, the samples $w(n)$ are not highly correlated, since the samples are taken at the Nyquist rate of $w(n)$. By exploiting the high correlation between $x(n)$ and past samples of the sequence $v(n)$, it is possible to obtain an estimate of $x(n)$, which can be subtracted from $v(n)$.

The general configuration is illustrated in Figure 13.1.12. The signal $v(n)$ is delayed by D samples, where D is selected sufficiently large so that the wideband signal components $w(n)$ and $w(n - D)$ contained in $v(n)$ and $v(n - D)$, respectively, are uncorrelated. Usually, a choice of $D = 1$ or 2 is adequate. The delayed signal sequence $v(n-D)$ is passed through an FIR filter, which is best characterized as a linear predictor of the value $x(n)$ based on M samples $v(n - D - k), k = 0, 1, \ldots, M - 1$. The output of the linear predictor is



(a)



(b)

Figure 13.1.12 Adaptive filter for estimating and suppressing a narrowband interference in a wideband signal.

$$\hat{x}(n) = \sum_{k=0}^{M-1} h(k)v(n - D - k) \tag{13.1.16}$$

This predicted value of $x(n)$ is subtracted from $v(n)$ to yield an estimate of $w(n)$, as illustrated in Figure 13.1.12. Clearly, the quality of the estimate $\hat{x}(n)$ determines how well the narrowband interference is suppressed. It is also apparent that the delay D must be kept as small as possible in order to obtain a good estimate of $x(n)$, but must be sufficiently large so that $w(n)$ and $w(n - D)$ are uncorrelated.

Let us define the error sequence

$$\begin{aligned} e(n) &= v(n) - \hat{x}(n) \\ &= v(n) - \sum_{k=0}^{M-1} h(k)v(n - D - k) \end{aligned} \tag{13.1.17}$$

If we apply the least-squares criterion to optimally select the prediction coefficients,

we obtain the set of linear equations

$$\sum_{k=0}^{M-1} h(k) r_{vv}(l-k) = r_{vv}(l+D), \qquad l = 0, 1, \ldots, M-1 \tag{13.1.18}$$

where $r_{vv}(l)$ is the autocorrelation sequence of $v(n)$. Note, however, that the right-hand side of (13.1.18) may be expressed as

$$\begin{aligned} r_{vv}(l+D) &= \sum_{n=0}^{N} v(n)v(n-l-D) \\ &= \sum_{n=0}^{N} [w(n)+x(n)][w(n-l-D)+x(n-l-D)] \\ &= r_{ww}(l+D) + r_{xx}(l+D) + r_{wx}(l+D) + r_{xw}(l+D) \end{aligned} \tag{13.1.19}$$

The correlations in (13.1.19) are time-average correlation sequences. The expected value of $r_{ww}(l+D)$ is

$$E[r_{ww}(l+D)] = 0, \qquad l = 0, 1, \ldots, M-1 \tag{13.1.20}$$

because $w(n)$ is wideband and D is large enough that $w(n)$ and $w(n-D)$ are uncorrelated. Also,

$$E[r_{xw}(l+D)] = E[r_{wx}(l+D)] = 0 \tag{13.1.21}$$

by assumption. Finally,

$$E[r_{xx}(l+D)] = \gamma_{xx}(l+D) \tag{13.1.22}$$

Therefore, the expected value of $r_{vv}(l+D)$ is simply the statistical autocorrelation of the narrowband signal $x(n)$. Furthermore, if the wideband signal is weak relative to the interference, the autocorrelation $r_{vv}(l)$ in the left-hand side of (13.1.18) is approximately $r_{xx}(l)$. The major influence of $w(n)$ is to the diagonal elements of $r_{vv}(l)$. Consequently, the values of the filter coefficients determined from the linear equations in (13.1.18) are a function of the statistical characteristics of the interference $x(n)$.

The overall filter structure in Figure 13.1.12 is an adaptive FIR prediction-error filter with coefficients

$$h'(k) = \begin{cases} 1, & k = 0 \\ -h(k-D), & k = D, D+1, \ldots, D+M-1 \\ 0, & \text{otherwise} \end{cases} \tag{13.1.23}$$

and a frequency response

$$H(\omega) = \sum_{k=0}^{M-1} h'(k+D) e^{-j\omega k} \tag{13.1.24}$$

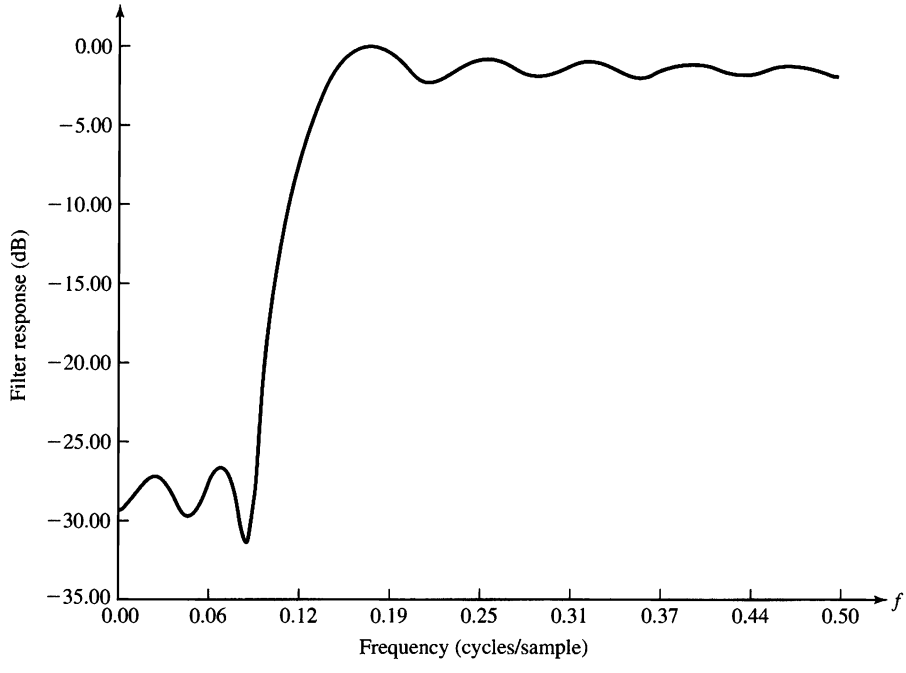


Figure 13.1.13 Frequency response characteristics of an adaptive notch filter.

This overall filter acts as a notch filter for the interference. For example, Figure 13.1.13 illustrates the magnitude of the frequency response of an adaptive filter with $M = 15$ coefficients, which attempts to suppress a narrowband interference that occupies 20 percent of the frequency band of a desired spread-spectrum signal sequence. The data were generated pseudorandomly by adding a narrowband interference consisting of 100 randomly phased, equal-amplitude sinusoids to a pseudonoise spread-spectrum signal. The coefficients of the filter were obtained by solving the equations in (13.1.18), with $D = 1$, where the correlation $r_{vv}(l)$ was obtained from the data. We observe that the overall interference suppression filter has the characteristics of a notch filter. The depth of the notch depends on the power of the interference relative to the wideband signal. The stronger the interference, the deeper the notch.

The algorithms presented in Sections 13.2 and 13.3 are appropriate for estimating the predictor coefficients continuously, in order to track a nonstationary narrowband interference signal.

13.1.5 Adaptive Line Enhancer

In the preceding example, the adaptive linear predictor was used to estimate the narrowband interference for the purpose of suppressing the interference from the input sequence $v(n)$. An adaptive line enhancer (ALE) has the same configuration as the interference suppression filter in Figure 13.1.12, except that the objective is different.

In the adaptive line enhancer, $x(n)$ is the desired signal and $w(n)$ represents

a wideband noise component that masks $x(n)$. The desired signal $x(n)$ is either a spectral line or a relatively narrowband signal. The linear predictor shown in Figure 13.1.12(b) operates in exactly the same fashion as that in Figure 13.1.12(a), and provides an estimate of the narrowband signal $x(n)$. It is apparent that the ALE (i.e., the FIR prediction filter) is a self-tuning filter that has a peak in its frequency response at the frequency of the sinusoid or, equivalently, in the frequency band of the narrowband signal $x(n)$. By having a narrow bandwidth, the noise $w(n)$ outside of the band is suppressed and, thus, the spectral line is enhanced in amplitude relative to the noise power in $w(n)$. This explains why the FIR predictor is called an ALE. Its coefficients are determined by the solution of (13.1.18).

13.1.6 Adaptive Noise Cancelling

Echo cancellation, the suppression of narrowband interference in a wideband signal, and the ALE are related to another form of adaptive filtering called *adaptive noise cancelling*. A model for the adaptive noise canceller is illustrated in Figure 13.1.14.

The primary input signal consists of a desired signal sequence $x(n)$ corrupted by an additive noise sequence $w_1(n)$ and an additive interference (noise) $w_2(n)$. The additive interference (noise) is also observable after it has been filtered by some unknown linear system that yields $v_2(n)$ and is further corrupted by an additive noise sequence $w_3(n)$. Thus, we have available a secondary signal sequence, which may be expressed as $v(n) = v_2(n) + w_3(n)$. The sequences $w_1(n)$, $w_2(n)$, and $w_3(n)$ are assumed to be mutually uncorrelated and zero mean.

As shown in Figure 13.1.14, an adaptive FIR filter is used to estimate the interference sequence $w_2(n)$ from the secondary signal $v(n)$ and subtract the estimate $w_2(n)$ from the primary signal. The output sequence, which represents an estimate of the desired signal $x(n)$, is the error signal

$$\begin{aligned} e(n) &= y(n) - \hat{w}_2(n) \\ &= y(n) - \sum_{k=0}^{M-1} h(k)v(n-k) \end{aligned} \tag{13.1.25}$$

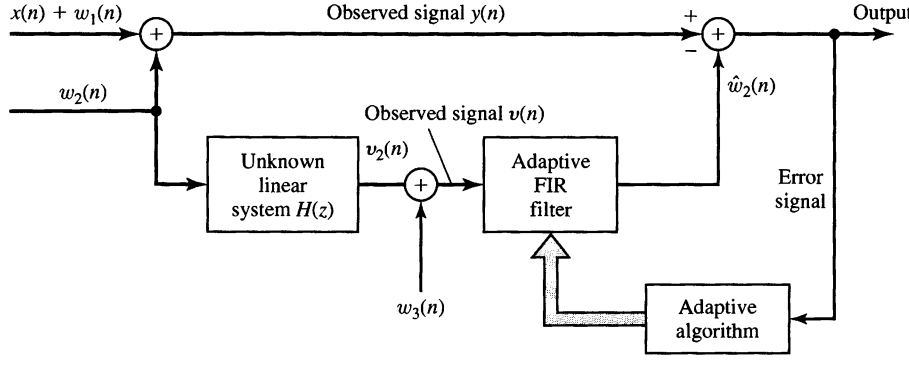


Figure 13.1.14 Example of an adaptive noise-cancelling system.

This error sequence is used to adaptively adjust the coefficients of the FIR filter.

If the least-squares criterion is used to determine the filter coefficients, the result of the optimization is the set of linear equations

$$\sum_{k=0}^{M-1} h(k) r_{vv}(l-k) = r_{yv}(l), \qquad l = 0, 1, \ldots, M-1 \tag{13.1.26}$$

where $r_{vv}(l)$ is the sample (time-average) autocorrelation of the sequence $v(n)$ and $r_{yv}(l)$ is the sample crosscorrelation of the sequences $y(n)$ and $v(n)$. Clearly, the noise cancelling problem is similar to the last three adaptive filtering applications described above.

13.1.7 Linear Predictive Coding of Speech Signals

Various methods have been developed over the past four decades for digital encoding of speech signals. In the telephone system, for example, two commonly used methods for speech encoding are pulse code modulation (PCM) and differential PCM (DPCM). These are examples of *waveform coding* methods. Other waveform coding methods have also been developed, such as delta modulation (DM) and adaptive DPCM.

Since the digital speech signal is ultimately transmitted from the source to a destination, a primary objective in devising speech encoders is to minimize the number of bits required to represent the speech signal, while maintaining speech intelligibility. This objective has led to the development of a class of low bit-rate (10,000 bits per second and below) speech encoding methods, which are based on constructing a model of the speech source and transmitting the model parameters. Adaptive filtering finds application in these model-based speech coding systems. We describe one very effective method called *linear predictive coding* (LPC).

In LPC, the vocal tract is modeled as a linear all-pole filter having the system function

$$H(z) = \frac{G}{1 - \sum_{k=1}^{p} a_k z^{-k}} \tag{13.1.27}$$

where p is the number of poles, G is the filter gain, and a_k are the parameters that determine the poles. There are two mutually exclusive excitation functions, used to model voiced and unvoiced speech sounds. On a short-time basis, voiced speech is periodic with a fundamental frequency F_0, or a pitch period $1/F_0$, which depends on the speaker. Thus, voiced speech is generated by exciting the all-pole filter model by a periodic impulse train with a period equal to the desired pitch period. Unvoiced speech sounds are generated by exciting the all-pole filter model by the output of a random-noise generator. This model is shown in Figure 13.1.15.

Given a short-time segment of a speech signal, the speech encoder at the transmitter must determine the proper excitation function, the pitch period for voiced speech, the gain parameter G, and the coefficients $\{a_k\}$. A block diagram that illustrates the source encoding system is given in Figure 13.1.16. The parameters of the model are determined adaptively from the data. Then, the speech samples are

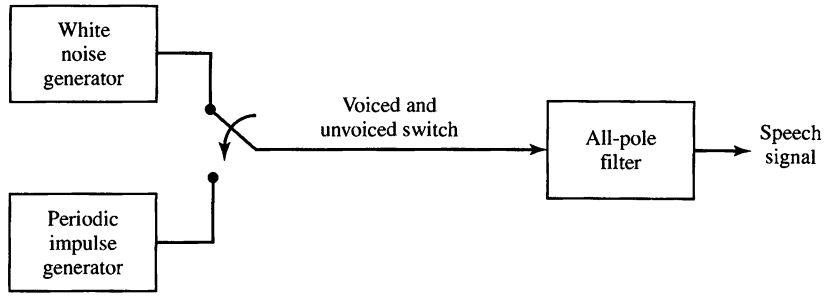


Figure 13.1.15 Block diagram model for the generation of a speech signal.

synthesized by using the model, and an error signal sequence is generated (as shown in Figure 13.1.16) by taking the difference between the actual and the synthesized sequence. The error signal and the model parameters are encoded into a binary sequence and transmitted to the destination. At the receiver, the speech signal is synthesized from the model and the error signal.

The parameters of the all-pole filter model are easily determined from the speech samples by means of linear prediction. To be specific, consider the system shown in Figure 13.1.17 and assume that we have N signal samples. The output of the FIR filter is

$$\hat{x}(n) = \sum_{k=1}^{p} a_k x(n-k) \tag{13.1.28}$$

and the corresponding error between the observed sample $x(n)$ and the estimate $\hat{x}(n)$ is

$$e(n) = x(n) - \sum_{k=1}^{p} a_k x(n-k) \tag{13.1.29}$$

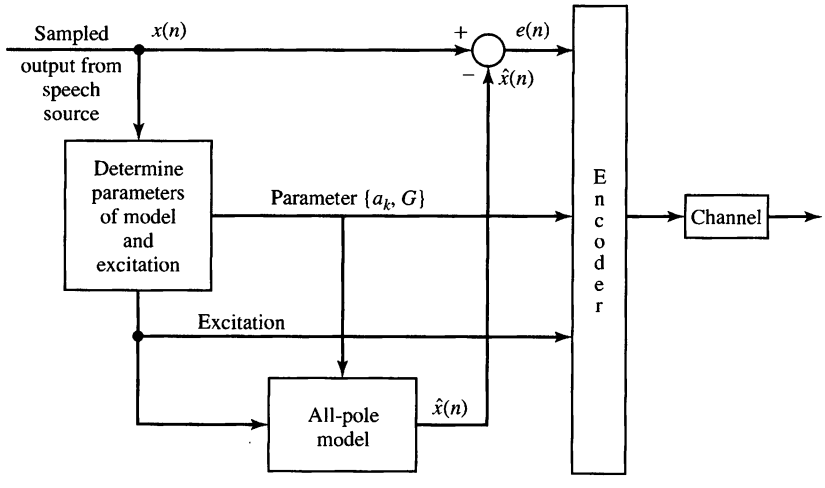


Figure 13.1.16 Source encoder for a speech signal.

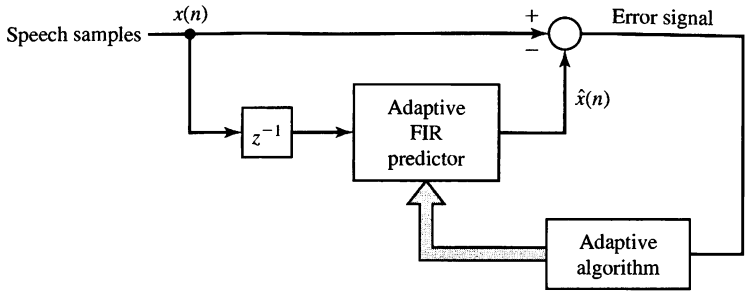


Figure 13.1.17 Estimation of pole parameters in LPC.

By applying the least-squares criterion, we can determine the model parameters a_k. The result of this optimization is a set of linear equations

$$\sum_{k=1}^{p} a_k r_{xx}(l-k) = r_{xx}(l), \qquad l = 1, 2, \ldots, p \tag{13.1.30}$$

where $r_{xx}(l)$ is the time-average autocorrelation of the sequence $x(n)$. The gain parameter for the filter can be obtained by noting that its input–output equation is

$$x(n) = \sum_{k=1}^{p} a_k x(n-k) + Gv(n) \tag{13.1.31}$$

where $v(n)$ is the input sequence. Clearly,

$$\begin{aligned} Gv(n) &= x(n) - \sum_{k=1}^{p} a_k x(n-k) \\ &= e(n) \end{aligned}$$

Then,

$$G^2 \sum_{n=0}^{N-1} v^2(n) = \sum_{n=0}^{N-1} e^2(n) \tag{13.1.32}$$

If the input excitation is normalized to unit energy by design, then

$$\begin{aligned} G^2 &= \sum_{n=0}^{N-1} e^2(n) \\ &= r_{xx}(0) - \sum_{k=1}^{p} a_k r_{xx}(k) \end{aligned} \tag{13.1.33}$$

Thus, G^2 is set equal to the residual energy resulting from the least-squares optimization.

In this development, we have described the use of linear prediction to adaptively determine the pole parameters and the gain of an all-pole filter model for speech generation. In practice, due to the nonstationary character of speech signals, this model is applied to short-time segments (10 to 20 milliseconds) of a speech signal. Usually, a new set of parameters is determined for each short-time segment. However, it is often advantageous to use the model parameters measured from previous segments to smooth out sharp discontinuities that usually exist in estimates of model parameters obtained from segment to segment. Although our discussion was totally in terms of the FIR filter structure, we should mention that speech synthesis is usually performed by using the FIR lattice structure and the reflection coefficients K_i. Since the dynamic range of the K_i is significantly smaller than that of the a_k, the reflection coefficients require fewer bits to represent them. Hence, the K_i are transmitted over the channel. Consequently, it is natural to synthesize the speech at the destination using the all-pole lattice structure.

In our treatment of LPC for speech coding, we have not considered algorithms for the estimation of the excitation and the pitch period. A discussion of appropriate algorithms for these parameters of the model would take us too far afield and, hence, is omitted. The interested reader is referred to Rabiner and Schafer (1978) and Deller, Hansen and Proakis (2000) for a detailed treatment of speech analysis and synthesis methods.

13.1.8 Adaptive Arrays

In the previous examples, we considered adaptive filtering performed on a single data sequence. However, adaptive filtering has also been widely applied to multiple data sequences that result from antenna, hydrophone, and seismometer arrays, where the sensors (antennas, hydrophones, or seismometers) are arranged in some spatial configuration. Each element of the array of sensors provides a signal sequence. By properly combining the signals from the various sensors, it is possible to change the directivity pattern of the array. For example, consider a linear antenna array consisting of five elements, as shown in Figure 13.1.18(a). If the signals are simply linearly summed, we obtain the sequence

$$x(n) = \sum_{k=1}^{5} x_k(n) \tag{13.1.34}$$

which results in the antenna directivity pattern shown in Figure 13.1.18(a). Now, suppose that an interference signal is received from a direction corresponding to one of the sidelobes in the array. By properly weighting the sequences $x_k(n)$ prior to combining, it is possible to alter the sidelobe pattern such that the array contains a null in the direction of the interference, as shown in Figure 13.1.18(b). Thus, we obtain

$$x(n) = \sum_{k=1}^{5} h_k x_k(n) \tag{13.1.35}$$

where the h_k are the weights.

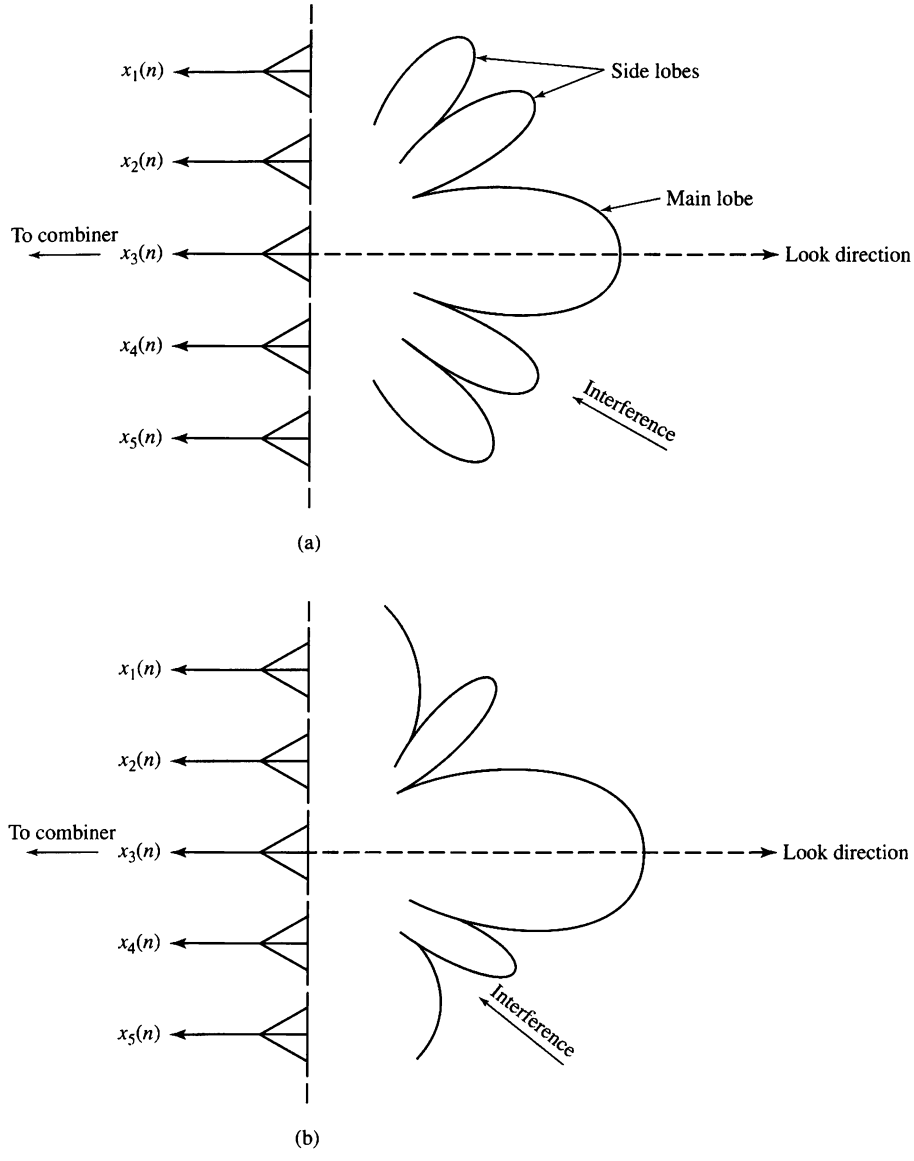


Figure 13.1.18 Linear antenna array: (a) linear antenna array with antenna pattern; (b) linear antenna array with a null placed in the direction of the interference.

We may also change or steer the direction of the main antenna lobe by simply introducing delays in the output of the sensor signals prior to combining. Hence, from K sensors we have a combined signal of the form

$$x(n) = \sum_{k=1}^{K} h_k x_k(n - n_k) \tag{13.1.36}$$

where the h_k are the weights and n_k corresponds to an n_k-sample delay in the signal $x(n)$. The choice of weights may be used to place nulls in specific directions.

More generally, we may simply filter each sequence prior to combining. In such a case, the output sequence has the general form

$$\begin{aligned} y(n) &= \sum_{k=1}^{K} y_k(n) \\ &= \sum_{k=1}^{K} \sum_{l=0}^{M-1} h_k(l) x_k(n - n_k - l) \end{aligned} \tag{13.1.37}$$

where $h_k(l)$ is the impulse response of the filter for processing the kth sensor output and the n_k are the delays that steer the beam pattern.

The LMS algorithm described in Section 13.2.2 is frequently used in adaptively selecting the weights h_k or the impulse responses $h_k(l)$. The more powerful recursive least-squares algorithms described in Section 13.3 can also be applied to the multisensor (multichannel) data problem.

13.2 Adaptive Direct-Form FIR Filters—The LMS Algorithm

From the examples of the previous section, we observed that there is a common framework in all the adaptive filtering applications. The least-squares criterion that we have adopted leads to a set of linear equations for the filter coefficients, which may be expressed as

$$\sum_{k=0}^{M-1} h(k) r_{xx}(l - k) = r_{dx}(l + D), \qquad l = 0, 1, 2, \ldots, M - 1 \tag{13.2.1}$$

where $r_{xx}(l)$ is the autocorrelation of the sequence $x(n)$ and $r_{dx}(l)$ is the crosscorrelation of the sequences $d(n)$ and $x(n)$. The delay parameter D is zero in some cases and nonzero in others.

We observe that the autocorrelation $r_{xx}(l)$ and the crosscorrelation $r_{dx}(l)$ are obtained from the data and, hence, represent estimates of the true (statistical) autocorrelation and crosscorrelation sequences. As a result, the coefficients $h(k)$ obtained from (13.2.1) are estimates of the true coefficients. The quality of the estimates depends on the length of the data record that is available for estimating $r_{xx}(l)$ and $r_{dx}(l)$. This is one problem that must be considered in the implementation of an adaptive filter.

A second problem that must be considered is that the underlying random process $x(n)$ is usually nonstationary. For example, in channel equalization, the frequency response characteristics of the channel may vary with time. As a consequence, the statistical autocorrelation and crosscorrelation sequences—and, hence, their estimates—vary with time. This implies that the coefficients of the adaptive filter must change with time to incorporate the time-variant statistical characteristics

of the signal into the filter. This also implies that the quality of the estimates cannot be made arbitrarily high by simply increasing the number of signal samples used in the estimation of the autocorrelation and crosscorrelation sequences.

There are several ways by which the coefficients of the adaptive filter can be varied with time to track the time-variant statistical characteristics of the signal. The most popular method is to adapt the filter recursively on a sample-by-sample basis, as each new signal sample is received. A second approach is to estimate $r_{xx}(l)$ and $r_{dx}(l)$ on a block-by-block basis, with no attempt to maintain continuity in the values of the filter coefficients from one block of data to another. In such a scheme, the block size must be relatively small, encompassing a time interval that is short compared to the time interval over which the statistical characteristics of the data change significantly. In addition to this block processing method, other block processing schemes can be devised that incorporate some block-to-block continuity in the filter coefficients.

In our treatment of adaptive filtering algorithms, we consider only time-recursive algorithms that update the filter coefficients on a sample-by-sample basis. In particular, we consider two types of algorithms, the LMS algorithm, which is based on a gradient-type search for tracking the time-variant signal characteristics, and the class of recursive least-squares algorithms, which are significantly more complex than the LMS algorithm, but which provide faster convergence to changes in signal statistics.

13.2.1 Minimum Mean-Square-Error Criterion

The LMS algorithm that is described in the following subsection is most easily obtained by formulating the optimization of the FIR filter coefficients as an estimation problem based on the minimization of the mean-square error. Let us assume that we have available the (possibly complex-valued) data sequence $x(n)$, which consists of samples from a stationary random process with autocorrelation sequence

$$\gamma_{xx}(m) = E[x(n)x^*(n-m)] \tag{13.2.2}$$

From these samples, we form an estimate of the desired sequence $d(n)$ by passing the observed data $x(n)$ through an FIR filter with coefficients $h(n), 0 \leq n \leq M-1$. The filter output may be expressed as

$$\hat{d}(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \tag{13.2.3}$$

where $\hat{d}(n)$ represents an estimate of $d(n)$. The estimation error is defined as

$$\begin{aligned} e(n) &= d(n) - \hat{d}(n) \\ &= d(n) - \sum_{k=0}^{M-1} h(k)x(n-k) \end{aligned} \tag{13.2.4}$$

The mean-square error as a function of the filter coefficients is

$$\begin{aligned}\mathscr{E}_M &= E[|e(n)|^2] \\ &= E\left[\left|d(n) - \sum_{k=0}^{M-1} h(k)x(n-k)\right|^2\right] \\ &= E\left\{|d(n)|^2 - 2\mathrm{Re}\left[\sum_{k=0}^{M-1} h^*(l)d(n)x^*(n-l)\right] + \sum_{k=0}^{M-1}\sum_{l=0}^{M-1} h^*(l)h(k)x^*(n-l)x(n-k)\right\} \\ &= \sigma_d^2 - 2\mathrm{Re}\left[\sum_{l=0}^{M-1} h^*(l)\gamma_{dx}(l)\right] + \sum_{l=0}^{M-1}\sum_{k=0}^{M-1} h^*(l)h(k)\gamma_{xx}(l-k) \qquad (13.2.5)\end{aligned}$$

where, by definition, $\sigma_d^2 = E[|d(n)|^2]$.

We observe that the MSE is a quadratic function of the filter coefficients. Consequently, the minimization of $\mathscr{E}_M$ with respect to the coefficients leads to the set of M linear equations,

$$\sum_{k=0}^{M-1} h(k)\gamma_{xx}(l-k) = \gamma_{dx}(l), \qquad l = 0, 1, \ldots, M-1 \qquad (13.2.6)$$

The filter with coefficients obtained from (13.2.6), which is the Wiener–Hopf equation previously derived in Section 12.7.1, is called the *Wiener filter*.

If we compare (13.2.6) with (13.2.1), it is apparent that these equations are similar in form. In (13.2.1), we use estimates of the autocorrelation and crosscorrelation to determine the filter coefficients, whereas in (13.2.6) the statistical autocorrelation and crosscorrelation are employed. Hence, (13.2.6) yields the optimum (Wiener) filter coefficients in the MSE sense, whereas (13.2.1) yields estimates of the optimum coefficients.

The equations in (13.2.6) may be expressed in matrix form as

$$\mathbf{\Gamma}_M \mathbf{h}_M = \boldsymbol{\gamma}_d \qquad (13.2.7)$$

where $\mathbf{h}_M$ denotes the vector of coefficients, $\mathbf{\Gamma}_M$ is an $M \times M$ (Hermitian) Toeplitz matrix with elements $\Gamma_{lk} = \gamma_{xx}(l-k)$, and $\boldsymbol{\gamma}_d$ is an $M \times 1$ crosscorrelation vector with elements $\gamma_{dx}(l), l = 0, 1, \ldots, M-1$. The complex-conjugate of $\mathbf{h}_M$ is denoted as $\mathbf{h}_M^*$ and the transpose as $\mathbf{h}_M^t$. The solution for the optimum filter coefficients is

$$\mathbf{h}_{\text{opt}} = \mathbf{\Gamma}_M^{-1}\boldsymbol{\gamma}_d \qquad (13.2.8)$$

and the resulting minimum MSE achieved with the optimum coefficients given by (13.2.8) is

$$\begin{aligned}\mathscr{E}_{M\,\min} &= \sigma_d^2 - \sum_{k=0}^{M-1} h_{\text{opt}}(k)\gamma_{dx}^*(k) \\ &= \sigma_d^2 - \boldsymbol{\gamma}_d^H \mathbf{\Gamma}_M^{-1}\boldsymbol{\gamma}_d \qquad (13.2.9)\end{aligned}$$

where the exponent H denotes the conjugate transpose.

Recall that the set of linear equations in (13.2.6) can also be obtained by invoking the orthogonality principle in mean-square estimation (see Section 12.7.2). According to the orthogonality principle, the mean-square estimation error is minimized when the error $e(n)$ is orthogonal, in the statistical sense, to the estimate $\hat{d}(n)$, that is,

$$E[e(n)\hat{d}^*(n)] = 0 \tag{13.2.10}$$

But the condition in (13.2.10) implies that

$$E\left[\sum_{k=0}^{M-1} h(k)e(n)x^*(n-k)\right] = \sum_{k=0}^{M-1} h(k)E[e(n)x^*(n-k)] = 0$$

or, equivalently,

$$E[e(n)x^*(n-l)] = 0, \qquad l = 0, 1, \ldots, M-1 \tag{13.2.11}$$

If we substitute for $e(n)$ in (13.2.11) using the expression for $e(n)$ given in (13.2.4), and perform the expectation operation, we obtain the equations given in (13.2.6).

Since $\hat{d}(n)$ is orthogonal to $e(n)$, the residual (minimum) mean-square error is

$$\begin{aligned} \mathcal{E}_{M\,\min} &= E[e(n)d^*(n)] \\ &= E[|d(n)|^2] - \sum_{k=0}^{M-1} h_{\text{opt}}(k)\gamma_{dx}^*(k) \end{aligned} \tag{13.2.12}$$

which is the result given in (13.2.9).

The optimum filter coefficients given by (13.2.8) can be solved efficiently by using the Levinson-Durbin algorithm. However, we shall consider the use of a gradient method for solving for $\mathbf{h}_{\text{opt}}$, iteratively. This development leads to the LMS algorithm for adaptive filtering.

13.2.2 The LMS Algorithm

There are various numerical methods that can be used to solve the set of linear equations given by (13.2.6) or (13.2.7) for the optimum FIR filter coefficients. In the following, we consider recursive methods that have been devised for finding the minimum of a function of several variables. In our problem, the performance index is the MSE given by (13.2.5), which is a quadratic function of the filter coefficients. Hence, this function has a unique minimum, which we shall determine by an iterative search.

For the moment, let us assume that the autocorrelation matrix $\mathbf{\Gamma}_M$ and the cross-correlation vector $\boldsymbol{\gamma}_d$ are known. Hence, $\mathcal{E}_M$ is a known function of the coefficients $h(n), 0 \leq n \leq M-1$. Algorithms for recursively computing the filter coefficients and, thus, searching for the minimum of $\mathcal{E}_M$, have the form

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \frac{1}{2}\Delta(n)\mathbf{S}(n), \qquad n = 0, 1, \ldots \tag{13.2.13}$$

where $\mathbf{h}_M(n)$ is the vector of filter coefficients at the nth iteration, $\Delta(n)$ is a step size at the nth iteration, and $\mathbf{S}(n)$ is a direction vector for the nth iteration. The initial vector $\mathbf{h}_M(0)$ is chosen arbitrarily. In this treatment we exclude methods that require the computations of $\mathbf{\Gamma}_M^{-1}$, such as Newton's method, and consider only search methods based on the use of gradient vectors.

The simplest method for finding the minimum of $\mathscr{E}_M$ recursively is based on a steepest-descent search (see Murray (1972)). In the method of steepest descent, the direction vector $\mathbf{S}(n) = -\mathbf{g}(n)$, where $\mathbf{g}(n)$ is the gradient vector at the nth iteration, defined as

$$\begin{aligned}\mathbf{g}(n) &= \frac{d\mathscr{E}_M(n)}{d\mathbf{h}_M(n)} \\ &= 2[\mathbf{\Gamma}_M\mathbf{h}_M(n) - \boldsymbol{\gamma}_d], \qquad n = 0, 1, 2, \ldots \end{aligned} \tag{13.2.14}$$

Hence, we compute the gradient vector at each iteration and change the values of $\mathbf{h}_M(n)$ in a direction opposite the gradient. Thus, the recursive algorithm based on the method of steepest descent is

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) - \frac{1}{2}\Delta(n)\mathbf{g}(n) \tag{13.2.15}$$

or, equivalently,

$$\mathbf{h}_M(n+1) = [\mathbf{I} - \Delta(n)\mathbf{\Gamma}_M]\mathbf{h}_M(n) + \Delta(n)\boldsymbol{\gamma}_d \tag{13.2.16}$$

We state without proof that the algorithm leads to the convergence of $\mathbf{h}_M(n)$ to $\mathbf{h}_{\text{opt}}$ in the limit as $n \to \infty$, provided that the sequence of step sizes $\Delta(n)$ is absolutely summable, with $\Delta(n) \to 0$ as $n \to \infty$. It follows that as $n \to \infty$, $\mathbf{g}(n) \to \mathbf{0}$.

Other candidate algorithms that provide faster convergence are the *conjugate-gradient* algorithm and the Fletcher–Powell algorithm. In the conjugate-gradient algorithm, the direction vectors are given as

$$\mathbf{S}(n) = \beta(n-1)\mathbf{S}(n-1) - \mathbf{g}(n) \tag{13.2.17}$$

where $\beta(n)$ is a scalar function of the gradient vectors (see Beckman (1960)). In the Fletcher–Powell algorithm, the direction vectors are given as

$$\mathbf{S}(n) = -\mathbf{H}(n)\mathbf{g}(n) \tag{13.2.18}$$

where $\mathbf{H}(n)$ is an $M \times M$ positive definite matrix, computed iteratively, that converges to the inverse of $\mathbf{\Gamma}_M$ (see Fletcher and Powell (1963)). Clearly, the three algorithms differ in the manner in which the direction vectors are computed.

These three algorithms are appropriate when $\mathbf{\Gamma}_M$ and $\boldsymbol{\gamma}_d$ are known. However, this is not the case in adaptive filtering applications, as we have previously indicated. In the absence of knowledge of $\mathbf{\Gamma}_M$ and $\boldsymbol{\gamma}_d$, we may substitute estimates $\hat{\mathbf{S}}(n)$ of the direction vectors in place of the actual vectors $\mathbf{S}(n)$. We consider this approach for the steepest-descent algorithm.

First, we note that the gradient vector given by (13.2.14) may also be expressed in terms of the orthogonality conditions given by (13.2.11). In fact, the conditions in (13.2.11) are equivalent to the expression

$$E[e(n)\mathbf{X}_M^*(n)] = \boldsymbol{\gamma}_d - \boldsymbol{\Gamma}_M \mathbf{h}_M(n) \tag{13.2.19}$$

where $\mathbf{X}_M(n)$ is the vector with elements $x(n-l), l = 0, 1, \ldots, M-1$. Therefore, the gradient vector is simply

$$\mathbf{g}(n) = -2E[e(n)\mathbf{X}_M^*(n)] \tag{13.2.20}$$

Clearly, the gradient vector $\mathbf{g}(n) = \mathbf{0}$ when the error is orthogonal to the data in the estimate $\hat{d}(n)$.

An unbiased estimate of the gradient vector at the nth iteration is simply obtained from (13.2.20) as

$$\hat{\mathbf{g}}(n) = -2e(n)\mathbf{X}_M^*(n) \tag{13.2.21}$$

where $e(n) = d(n) - \hat{d}(n)$, and $\mathbf{X}_M(n)$ is the set of M signal samples in the filter at the nth iteration. Thus, with $\hat{\mathbf{g}}(n)$ substituted for $\mathbf{g}(n)$, we have the algorithm

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta(n)e(n)\mathbf{X}_M^*(n) \tag{13.2.22}$$

This is called a *stochastic-gradient-descent algorithm.* As given by (13.2.22), it has a variable step size.

It has become common practice in adaptive filtering to use a fixed-step-size algorithm for two reasons. The first is that a fixed-step-size algorithm is easily implemented in either hardware or software. The second is that a fixed step size is appropriate for tracking time-variant signal statistics, whereas if $\Delta(n) \to 0$ as $n \to \infty$, adaptation to signal variations cannot occur. For these reasons, (13.2.22) is modified to the algorithm

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta e(n)\mathbf{X}_M^*(n) \tag{13.2.23}$$

where Δ is now the fixed step size. This algorithm was first proposed by Widrow and Hoff (1960) and is now widely known as the *LMS (least-mean-squares) algorithm.* Clearly, it is a stochastic-gradient algorithm.

The LMS algorithm is relatively simple to implement. For this reason, it has been widely used in many adaptive filtering applications. Its properties and limitations have also been thoroughly investigated. In the following section, we provide a brief treatment of its important properties concerning convergence, stability, and the noise resulting from the use of estimates of the gradient vectors. Subsequently, we compare its properties with the more complex recursive least-squares algorithms.

13.2.3 Related Stochastic Gradient Algorithms

Several variations of the basic LMS algorithm have been proposed in the literature and implemented in adaptive filtering applications. One variation is obtained if we

average the gradient vectors over several iterations prior to making adjustments of the filter coefficients. For example, the average over K gradient vectors is

$$\bar{\hat{\mathbf{g}}}(nK) = -\frac{2}{K}\sum_{k=0}^{K-1} e(nK+k)\mathbf{X}_M^*(nK+k) \tag{13.2.24}$$

and the corresponding recursive equation for updating the filter coefficients once every K iterations is

$$\mathbf{h}_M((n+1)K) = \mathbf{h}_M(nK) - \frac{1}{2}\Delta\bar{\hat{\mathbf{g}}}(nK) \tag{13.2.25}$$

In effect, the averaging operation performed in (13.2.24) reduces the noise in the estimate of the gradient vector, as shown by Gardner (1984).

An alternative approach is to filter the gradient vectors by a lowpass filter and use the output of the filter as an estimate of the gradient vector. For example, a simple lowpass filter for the gradients yields as an output

$$\hat{\mathbf{S}}(n) = \beta\hat{\mathbf{S}}(n-1) - \hat{\mathbf{g}}(n), \qquad \mathbf{S}(0) = -\hat{\mathbf{g}}(0) \tag{13.2.26}$$

where the choice of $0 \leq \beta < 1$ determines the bandwidth of the lowpass filter. When β is close to unity, the filter bandwidth is small and the effective averaging is performed over many gradient vectors. On the other hand, when β is small, the lowpass filter has a large bandwidth and, hence, it provides little averaging of the gradient vectors. With the filtered gradient vectors given by (13.2.26) in place of $\hat{\mathbf{g}}(n)$, we obtain the filtered version of the LMS algorithm given by

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \frac{1}{2}\Delta\hat{\mathbf{S}}(n) \tag{13.2.27}$$

An analysis of the filtered-gradient LMS algorithm is given in Proakis (1974).

Three other variations of the basic LMS algorithm given in (13.2.23) are obtained by using sign information contained in the error signal sequence $e(n)$ and/or in the components of the signal vector $\mathbf{X}_M(n)$. Hence, the three possible variations are

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta\text{csgn}[e(n)]\mathbf{X}_M^*(n) \tag{13.2.28}$$

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta e(n)\text{csgn}[\mathbf{X}_M^*(n)] \tag{13.2.29}$$

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta\text{csgn}[e(n)]\text{csgn}[\mathbf{X}_M^*(n)] \tag{13.2.30}$$

where csgn$[x]$ is the complex sign function defined as

$$\text{csgn}[x] = \begin{cases} 1+j, & \text{if Re}(x) > 0 \text{ and Im}(x) > 0 \\ 1-j, & \text{if Re}(x) > 0 \text{ and Im}(x) < 0 \\ -1+j, & \text{if Re}(x) < 0 \text{ and Im}(x) > 0 \\ -1-j, & \text{if Re}(x) < 0 \text{ and Im}(x) < 0 \end{cases}$$

and csgn[$\mathbf{X}$] denotes the complex sign function applied to each element of the vector $\mathbf{X}$. These three variations of the LMS algorithm may be called reduced complexity LMS algorithms, since multiplications are completely avoided in (13.2.30), and can be completely avoided in (13.2.28) and (13.2.29) by selecting Δ to be a power of 1/2. The price paid for the reduction in computational complexity is a slower convergence of the filter coefficients to their optimum values.

Another version of the LMS algorithms, called a *normalized LMS*(NLMS) algorithm, that is frequently used in practice is given as

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \frac{\Delta}{||\mathbf{X}_M(n)||^2} e(n)\mathbf{X}_M^*(n) \tag{13.2.31}$$

By dividing the step size by the norm of the data vector $\mathbf{X}_M(n)$, the NLMS algorithm is equivalent to employing a variable step size of the form

$$\Delta(n) = \frac{\Delta}{||\mathbf{X}_M(n)||^2} \tag{13.2.32}$$

Thus, the step size at each iteration is inversely proportional to the energy in the received data vector $\mathbf{X}_M(n)$. This scaling is advantageous in adaptive filtering applications where the dynamic range of the input to the adaptive filter is large, as would be the case, for example, in the implementation of adaptive equalizers for slowly fading communication channels. In such applications, it may be advantageous to add a small positive constant to the denominator of (13.2.32) to avoid numerical instabilities that may result when the norm of $\mathbf{X}_M(n)$ is small. Thus, another version of the NLMS algorithm may employ a variable step size of the form

$$\Delta(n) = \frac{\Delta}{\delta + ||\mathbf{X}_M(n)||^2} \tag{13.2.33}$$

where δ is a small positive number.

13.2.4 Properties of the LMS Algorithm

In this section, we consider the basic properties of the LMS algorithm given by (13.2.23). In particular, we focus on its convergence properties, its stability, and the excess noise generated as a result of using noisy gradient vectors in place of the actual gradient vectors. The use of noisy estimates of the gradient vectors implies that the filter coefficients will fluctuate randomly and, hence, an analysis of the characteristics of the algorithm should be performed in statistical terms.

The convergence and stability of the LMS algorithm may be investigated by determining how the mean value of $\mathbf{h}_M(n)$ converges to the optimum coefficients $\mathbf{h}_{\text{opt}}$. If we take the expected value of (13.2.23), we obtain

$$\begin{aligned} \bar{\mathbf{h}}_M(n+1) &= \bar{\mathbf{h}}_M(n) + \Delta E[e(n)\mathbf{X}_M^*(n)] \\ &= \bar{\mathbf{h}}_M(n) + \Delta[\boldsymbol{\gamma}_d - \boldsymbol{\Gamma}_M \bar{\mathbf{h}}_M(n)] \\ &= (\mathbf{I} - \Delta\boldsymbol{\Gamma}_M)\bar{\mathbf{h}}_M(n) + \Delta\boldsymbol{\gamma}_d \end{aligned} \tag{13.2.34}$$

where $\bar{\mathbf{h}}_M(n) = \mathbf{E}[\mathbf{h}_M(n)]$, and $\mathbf{I}$ is the identity matrix.

The recursive relation in (13.2.34) may be represented as a closed-loop control system, as shown in Figure 13.2.1. The convergence rate and the stability of this closed-loop system are governed by our choice of the step-size parameter Δ. To determine the convergence behavior, it is convenient to decouple the M simultaneous difference equations given in (13.2.34), by performing a linear transformation of the mean coefficient vector $\bar{\mathbf{h}}_M(n)$. The appropriate transformation is obtained by noting that the autocorrelation matrix $\mathbf{\Gamma}_M$ is Hermitian and, hence, it can be represented (see Gantmacher (1960)) as

$$\mathbf{\Gamma}_M = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H \tag{13.2.35}$$

where $\mathbf{U}$ is the normalized modal matrix of $\mathbf{\Gamma}_M$ and $\mathbf{\Lambda}$ is a diagonal matrix with diagonal elements λ_k, $0 \le k \le M-1$, equal to the eigenvalues of $\mathbf{\Gamma}_M$.

When (13.2.35) is substituted into (13.2.34), the latter may be expressed as

$$\bar{\mathbf{h}}_M^0(n+1) = (\mathbf{I} - \Delta\mathbf{\Lambda})\bar{\mathbf{h}}_M^0(n) + \Delta\boldsymbol{\gamma}_d^0 \tag{13.2.36}$$

where the transformed (orthogonalized) vectors are $\bar{\mathbf{h}}_M^0(n) = \mathbf{U}^H\bar{\mathbf{h}}_M(n)$ and $\boldsymbol{\gamma}_d^0 = \mathbf{U}^H\boldsymbol{\gamma}_d$. The set of M first-order difference equations in (13.2.36) are now decoupled. Their convergence and their stability are determined from the homogeneous equation

$$\bar{\mathbf{h}}_M^0(n+1) = (\mathbf{I} - \Delta\mathbf{\Lambda})\bar{\mathbf{h}}_M^0(n) \tag{13.2.37}$$

If we focus our attention on the solution of the kth equation in (13.2.37), we observe that

$$\bar{h}^0(k,n) = C(1-\Delta\lambda_k)^n u(n), \qquad k = 0, 1, 2, \ldots, M-1 \tag{13.2.38}$$

where C is an arbitrary constant and $u(n)$ is the unit step sequence. Clearly, $\bar{h}^0(k,n)$ converges to zero exponentially, provided that

$$|1 - \Delta\lambda_k| < 1$$

or, equivalently,

$$0 < \Delta < \frac{2}{\lambda_k}, \qquad k = 0, 1, \ldots, M-1 \tag{13.2.39}$$

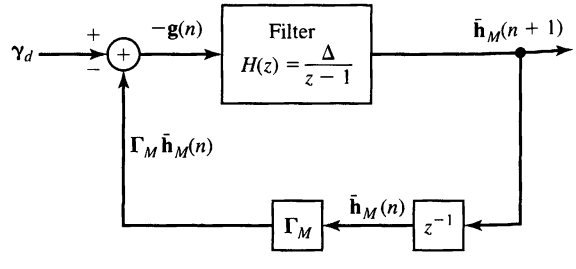


Figure 13.2.1
Closed-loop control system representation of recursive Equation (13.2.34).

The condition given by (13.2.39) for convergence of the homogeneous difference equation for the kth normalized filter coefficient (kth mode of the closed-loop system) must be satisfied for all $k = 0, 1, \ldots, M-1$. Therefore, the range of values of Δ that ensures the convergence of the mean of the coefficient vector in the LMS algorithm is

$$0 < \Delta < \frac{2}{\lambda_{\max}} \tag{13.2.40}$$

where $\lambda_{\max}$ is the largest eigenvalue of $\mathbf{\Gamma}_M$.

Since $\mathbf{\Gamma}_M$ is an autocorrelation matrix, its eigenvalues are nonnegative. Hence, an upper bound on $\lambda_{\max}$ is

$$\lambda_{\max} < \sum_{k=0}^{M-1} \lambda_k = \text{trace } \mathbf{\Gamma}_M = M\gamma_{xx}(0) \tag{13.2.41}$$

where $\gamma_{xx}(0)$ is the input signal power, which is easily estimated from the received signal. Therefore, an upper bound on the step size Δ is $2/M\gamma_{xx}(0)$.

From (13.2.38), we observe that rapid convergence of the LMS algorithm occurs when $|1 - \Delta\lambda_k|$ is small, that is, when the poles of the closed-loop system in Figure 13.2.1 are far from the unit circle. However, we cannot achieve this desirable condition and still satisfy the upper bound in (13.2.39) when there is a large difference between the largest and smallest eigenvalues of $\mathbf{\Gamma}_M$. In other words, even if we select Δ to be $1/\lambda_{\max}$, the convergence rate of the LMS algorithm will be determined by the decay of the mode corresponding to the smallest eigenvalue $\lambda_{\min}$. For this mode, with $\Delta = 1/\lambda_{\max}$ substituted in (13.2.38), we have

$$h_M^0(k, n) = C\left(1 - \frac{\lambda_{\min}}{\lambda_{\max}}\right)^n u(n) \tag{13.2.42}$$

Consequently, the ratio $\lambda_{\min}/\lambda_{\max}$ ultimately determines the convergence rate. If $\lambda_{\min}/\lambda_{\max}$ is small (much smaller than unity), the convergence rate will be slow. On the other hand, if $\lambda_{\min}/\lambda_{\max}$ is close to unity, the convergence rate of the algorithm is fast.

The other important characteristic of the LMS algorithm is the noise resulting from the use of estimates of the gradient vectors. The noise in the gradient-vector estimates causes random fluctuations in the coefficients about their optimal values and, thus, leads to an increase in the MMSE at the output of the adaptive filter. Hence, the total MSE is $\mathcal{E}_{M\,\min} + \mathcal{E}_\Delta$, where $\mathcal{E}_\Delta$ is called the *excess mean-square error.*

For any given set of filter coefficients $\mathbf{h}_M(n)$, the total MSE at the output of the adaptive filter may be expressed as

$$\mathcal{E}_t(n) = \mathcal{E}_{M\,\min} + (\mathbf{h}_M(n) - \mathbf{h}_{\text{opt}})^t \mathbf{\Gamma}_M (\mathbf{h}_M(n) - \mathbf{h}_{\text{opt}})^* \tag{13.2.43}$$

where $\mathbf{h}_{\text{opt}}$ represents the optimum filter coefficients defined by (13.2.8). A plot of $\mathcal{E}_t(n)$ as a function of the iteration n is called a *learning curve.* If we substitute

(13.2.35) for $\mathbf{\Gamma}_M$ and perform the linear orthogonal transformation used previously, we obtain

$$\mathscr{E}_t(n) = \mathscr{E}_{M\min} + \sum_{k=0}^{M-1} \lambda_k \left| h^0(k,n) - h^0_{\text{opt}}(k) \right|^2 \tag{13.2.44}$$

where the term $h^0(k,n) - h^0_{\text{opt}}(k)$ represents the error in the kth filter coefficient (in the orthogonal coordinate system). The excess MSE is defined as the expected value of the second term in (13.2.44),

$$\mathscr{E}_\Delta = \sum_{k=0}^{M-1} \lambda_k E\left[\left| h^0(k,n) - h^0_{\text{opt}}(k) \right|^2 \right] \tag{13.2.45}$$

To derive an expression for the excess MSE $\mathscr{E}_\Delta$, we assume that the mean values of the filter coefficients $\mathbf{h}_M(n)$ have converged to their optimum values $\mathbf{h}_{\text{opt}}$. Then, the term $\Delta e(n)\mathbf{X}^*_M(n)$ in the LMS algorithm given by (13.2.23) is a zero-mean noise vector. Its covariance is

$$\text{cov}[\Delta e(n)\mathbf{X}^*_M(n)] = \Delta^2 E\left[|e(n)|^2 \mathbf{X}_M(n)\mathbf{X}^H_M(n) \right] \tag{13.2.46}$$

To a first approximation, we assume that $|e(n)|^2$ is uncorrelated with the signal vector. Although this assumption is not strictly true, it simplifies the derivation and yields useful results. (The reader may refer to Mazo (1979), Jones, Cavin, and Reed (1982), and Gardner (1984) for further discussion on this assumption). Then,

$$\begin{aligned} \text{cov}[\Delta e(n)\mathbf{X}^*_M(n)] &= \Delta^2 E[|e(n)|^2] E\left[\mathbf{X}_M(n)\mathbf{X}^H_M(n) \right] \\ &= \Delta^2 \mathscr{E}_{M\min} \mathbf{\Gamma}_M \end{aligned} \tag{13.2.47}$$

For the orthogonalized coefficient vector $\mathbf{h}^0_M(n)$ with additive noise, we have the equation

$$\mathbf{h}^0_M(n+1) = (\mathbf{I} - \Delta\mathbf{\Lambda})\mathbf{h}^0_M(n) + \Delta\boldsymbol{\gamma}^0_d + \mathbf{w}^0(n) \tag{13.2.48}$$

where $\mathbf{w}^0(n)$ is the additive noise vector, which is related to the noise vector $\Delta e(n)\mathbf{X}^*_M(n)$ through the transformation

$$\begin{aligned} \mathbf{w}^0(n) &= \mathbf{U}^H[\Delta e(n)\mathbf{X}^*_M(n)] \\ &= \Delta e(n)\mathbf{U}^H\mathbf{X}^*_M(n) \end{aligned} \tag{13.2.49}$$

It is easily seen that the covariance matrix of the noise vector is

$$\begin{aligned} \text{cov}[\mathbf{w}^0(n)] &= \Delta^2 \mathscr{E}_{M\min} \mathbf{U}^H \mathbf{\Gamma}_M \mathbf{U} \\ &= \Delta^2 \mathscr{E}_{M\min} \mathbf{\Lambda} \end{aligned} \tag{13.2.50}$$

Therefore, the M components of $\mathbf{w}^0(n)$ are uncorrelated and each component has the variance $\sigma_k^2 = \Delta^2 \mathscr{E}_{M\min} \lambda_k$, $k = 0, 1, \ldots, M-1$.

Since the noise components of $\mathbf{w}^0(n)$ are uncorrelated, we may consider the M uncoupled difference equations in (13.2.48) separately. Each first-order difference equation represents a filter with impulse response $(1 - \Delta\lambda_k)^n$. When such a filter is excited with a noise sequence $w_k^0(n)$, the variance of the noise at the output of the filter is

$$E\left[\left|h^0(k,n) - h_{\text{opt}}^0(k)\right|^2\right] = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} (1 - \Delta\lambda_k)^n (1 - \Delta\lambda_k)^m E\left[w_k^0(n) w_k^{0*}(m)\right] \quad (13.2.51)$$

We make the simplifying assumption that the noise sequence $w_k^0(n)$ is white. Then, (13.2.51) reduces to

$$E\left[\left|h^0(k,n) - h_{\text{opt}}^0(k)\right|^2\right] = \frac{\sigma_k^2}{1 - (1 - \Delta\lambda_k)^2} = \frac{\Delta^2 \mathscr{E}_{M\min} \lambda_k}{1 - (1 - \Delta\lambda_k)^2} \quad (13.2.52)$$

If we substitute the result of (13.2.52) into (13.2.45), we obtain the expression for the excess MSE as

$$\mathscr{E}_\Delta = \Delta^2 \mathscr{E}_{M\min} \sum_{k=0}^{M-1} \frac{\lambda_k^2}{1 - (1 - \Delta\lambda_k)^2} \quad (13.2.53)$$

This expression can be simplified if we assume that Δ is selected such that $\Delta\lambda_k \ll 1$ for all k. Then,

$$\begin{aligned} \mathscr{E}_\Delta &\approx \Delta^2 \mathscr{E}_{M\min} \sum_{k=0}^{M-1} \frac{\lambda_k^2}{2\Delta\lambda_k} \\ &\approx \frac{1}{2} \Delta \mathscr{E}_{M\min} \sum_{k=0}^{M-1} \lambda_k \\ &\approx \frac{\Delta M \mathscr{E}_{M\min} \gamma_{xx}(0)}{2} \end{aligned} \quad (13.2.54)$$

where $\gamma_{xx}(0)$ is the power of the input signal.

The expression for $\mathscr{E}_\Delta$ indicates that the excess MSE is proportional to the step-size parameter Δ. Hence, our choice of Δ must be based on a compromise between fast convergence and a small excess MSE. In practice, it is desirable to have $\mathscr{E}_\Delta < \mathscr{E}_{M\min}$. Hence,

$$\frac{\mathscr{E}_\Delta}{\mathscr{E}_{M\min}} \approx \frac{\Delta M \gamma_{xx}(0)}{2} < 1$$

or, equivalently,

$$\Delta < \frac{2}{M\gamma_{xx}(0)} \tag{13.2.55}$$

But, this is just the upper bound that we had obtained previously for $\lambda_{\max}$. In steady-state operation, Δ should satisfy the upper bound in (13.2.55); otherwise the excess MSE causes significant degradation in the performance of the adaptive filter.

The preceding analysis of the excess MSE is based on the assumption that the mean values of the filter coefficients have converged to the optimum solution $\mathbf{h}_{\text{opt}}$. Under this condition, the step size Δ should satisfy the bound in (13.2.55). On the other hand, we have determined that convergence of the mean coefficient vector requires that $\Delta < 2/\lambda_{\max}$. While a choice of Δ near the upper bound $2/\lambda_{\max}$ may lead to initial convergence of the deterministic (known) gradient algorithm, such a large value of Δ will usually result in instability of the stochastic-gradient LMS algorithm.

The initial convergence or transient behavior of the LMS algorithm has been investigated by several researchers. Their results clearly indicate that the step size must be reduced in direct proportion to the length of the adaptive filter, as in (13.2.55). The upper bound given in (13.2.55) is necessary to ensure the initial convergence of the stochastic-gradient LMS algorithm. In practice, a choice of $\Delta < 1/M\gamma_{xx}(0)$ is usually made. Sayed (2003), Gitlin and Weinstein (1979), and Ungerboeck (1972) have given an analysis of the transient behavior and the convergence properties of the LMS algorithm.

In a digital implementation of the LMS algorithm, the choice of the step-size parameter becomes even more critical. In an attempt to reduce the excess MSE, it is possible to reduce the step-size parameter to the point at which the total output MSE actually increases. This condition occurs when the estimated gradient components $e(n)x^*(n-l), l = 0, 1, M-1$, after multiplication by the small step-size parameter Δ, are smaller than one-half of the least significant bit in the fixed-point representation of the filter coefficients. In such a case, adaptation ceases. Consequently, it is important for the step size to be large enough to bring the filter coefficients in the vicinity of $\mathbf{h}_{\text{opt}}$. If it is desired to decrease the step size significantly, it is necessary to increase the precision in the filter coefficients. Typically, sixteen bits of precision may be used for the filter coefficients, with the twelve most significant bits used for arithmetic operations in the filtering of the data. The four least significant bits are required to provide the necessary precision for the adaptation process. Thus, the scaled, estimated gradient components $\Delta e(n)x^*(n-l)$ usually affect only the least significant bits. In effect, the added precision also allows for the noise to be averaged out, since several incremental changes in the least significant bits are required before any change occurs in the upper, more significant bits used in arithmetic operations for filtering of the data. For an analysis of round-off errors in a digital implementation of the LMS algorithm, the reader is referred to Gitlin and Weinstein (1979), Gitlin, Meadors, and Weinstein (1982), and Caraiscos and Liu (1984).

As a final point, we should indicate that the LMS algorithm is appropriate for tracking slowly time-variant signal statistics. In such a case, the minimum MSE and the optimum coefficient vector will be time variant. In other words, $\mathcal{E}_{M\,\min}$ is a function of time, and the M-dimensional error surface is moving with the time

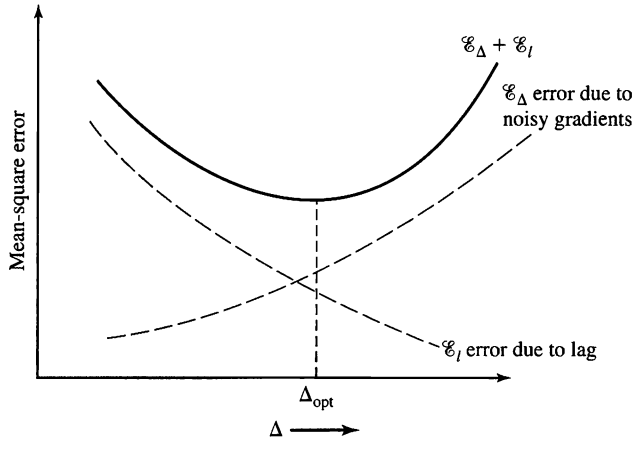


Figure 13.2.2
Excess mean-square error $\mathcal{E}_\Delta$ and lag error $\mathcal{E}_l$ as a function of the step size Δ.

index n. The LMS algorithm attempts to follow the moving minimum $\mathcal{E}_{M\,\min}$ in the M-dimensional space, but it is always lagging behind due to its use of (estimated) gradient vectors. As a consequence, the LMS algorithm incurs another form of error, called the *lag error*, whose mean-square value decreases with an increase in the step size Δ. The total MSE can now be expressed as

$$\mathcal{E}_{\text{total}} = \mathcal{E}_{M\,\min} + \mathcal{E}_\Delta + \mathcal{E}_l \tag{13.2.56}$$

where $\mathcal{E}_l$ denotes the MSE due to the lag.

In any given nonstationary adaptive filtering problem, if we plot the $\mathcal{E}_\Delta$ and $\mathcal{E}_l$ as a function of Δ, we expect these errors to behave as illustrated in Figure 13.2.2. We observe that $\mathcal{E}_\Delta$ increases with an increase in Δ, whereas $\mathcal{E}_l$ decreases with an increase in Δ. The total error will exhibit a minimum, which will determine the optimum choice of the step-size parameter.

When the statistical time variations of the signals occur rapidly, the lag error will dominate the performance of the adaptive filter. In such a case, $\mathcal{E}_l \gg \mathcal{E}_{M\,\min} + \mathcal{E}_\Delta$, even when the largest possible value of Δ is used.

EXAMPLE 13.2.1

Learning curves for the LMS algorithm, when used to adaptively equalize a communication channel, are illustrated in Figure 13.2.3. The FIR equalizer was realized in direct form and had a length $M = 11$. The autocorrelation matrix $\mathbf{\Gamma}_M$ has an eigenvalue spread of $\lambda_{\max}/\lambda_{\min} = 11$. These three learning curves have been obtained with step sizes $\Delta = 0.045, 0.09$, and 0.115, by averaging the (estimated) MSE in 200 simulation runs. The input signal power was normalized to unity. Hence, the upper bound in (13.2.55) is equal to 0.18. By selecting $\Delta = 0.09$ (one-half of the upper bound), we obtain a rapidly decaying learning curve, as shown in Figure 13.2.3. If we divide Δ by 2 to obtain 0.045, the convergence rate is reduced but the excess MSE is also reduced, so the algorithm performs better in the time-invariant signal environment. Finally, we note that a choice of $\Delta = 0.115$ causes large undesirable fluctuations in the output MSE of the algorithm. Note that $\Delta = 0.115$ is significantly lower than the upper bound given in (13.2.55).

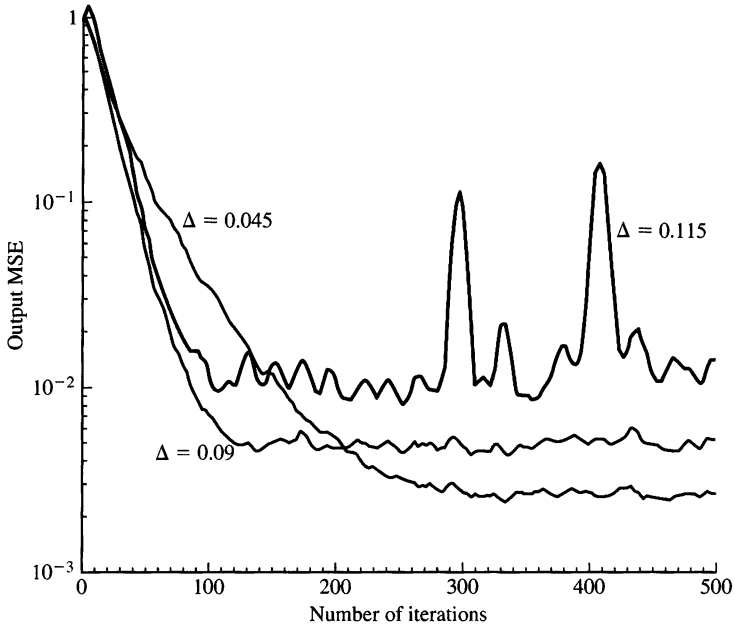


Figure 13.2.3 Learning curves for the LMS algorithm applied to an adaptive equalizer of length $M = 11$ and a channel with eigenvalue spread $\lambda_{max}/\lambda_{min} = 11$.

13.3 Adaptive Direct-Form Filters—RLS Algorithms

The major advantage of the LMS algorithm lies in its computational simplicity. However, the price paid for this simplicity is slow convergence, especially when the eigenvalues of the autocorrelation matrix $\mathbf{\Gamma}_M$ have a large spread, that is, when $\lambda_{max}/\lambda_{min} \gg 1$. From another point of view, the LMS algorithm has only a single adjustable parameter for controlling the convergence rate, namely, the step-size parameter Δ. Since Δ is limited for purposes of stability to be less than the upper bound in (13.2.55), the modes corresponding to the smaller eigenvalues converge very slowly.

To obtain faster convergence, it is necessary to devise more complex algorithms, which involve additional parameters. In particular, if the correlation matrix $\mathbf{\Gamma}_M$ has unequal eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{M-1}$, we should use an algorithm that contains M parameters, one for each of the eigenvalues. In deriving more rapidly converging adaptive filtering algorithms, we adopt the least-squares criterion instead of the statistical approach based on the MSE criterion. Thus, we deal directly with the data sequence $x(n)$ and obtain estimates of correlations from the data.

13.3.1 RLS Algorithm

It is convenient to express the least-squares algorithms in matrix form, in order to simplify the notation. Since the algorithms will be recursive in time, it is also necessary to introduce a time index in the filter-coefficient vector and in the error sequence.

Hence, we define the filter-coefficient vector at time n as

$$\mathbf{h}_M(n) = \begin{bmatrix} h(0,n) \\ h(1,n) \\ h(2,n) \\ \vdots \\ h(M-1,n) \end{bmatrix} \tag{13.3.1}$$

where the subscript M denotes the length of the filter. Similarly, the input signal vector to the filter at time n is denoted as

$$\mathbf{X}_M(n) = \begin{bmatrix} x(n) \\ x(n-1) \\ x(n-2) \\ \vdots \\ x(n-M+1) \end{bmatrix} \tag{13.3.2}$$

We assume that $x(n) = 0$ for $n < 0$. This is usually called *prewindowing* of the input data.

The recursive least-squares problem may now be formulated as follows. Suppose that we have observed the vectors $\mathbf{X}_M(l), l = 0, 1, \ldots, n$, and we wish to determine the filter-coefficient vector $\mathbf{h}_M(n)$ that minimizes the weighted sum of magnitude-squared errors

$$\mathscr{E}_M = \sum_{l=0}^{n} w^{n-l} |e_M(l,n)|^2 \tag{13.3.3}$$

where the error is defined as the difference between the desired sequence $d(l)$ and the estimate $\hat{d}(l,n)$,

$$\begin{aligned} e_M(l,n) &= d(l) - \hat{d}(l,n) \\ &= d(l) - \mathbf{h}_M^t(n)\mathbf{X}_M(l) \end{aligned} \tag{13.3.4}$$

and w is a weighting factor in the range $0 < w < 1$.

The purpose of the factor w is to weight the most recent data points more heavily and, thus, allow the filter coefficients to adapt to time-varying statistical characteristics of the data. This is accomplished by using the exponential weighting factor with the past data. Alternatively, we may use a finite-duration sliding window with uniform weighting over the window length. We find the exponential weighting factor more convenient, both mathematically and practically. For comparison, an exponentially weighted window sequence has an effective memory of

$$\bar{N} = \frac{\sum_{n=0}^{\infty} n w^n}{\sum_{n=0}^{\infty} w^n} = \frac{w}{1-w} \tag{13.3.5}$$

and, hence, should be approximately equivalent to a sliding window of length $\bar{N}$.

The minimization of $\mathcal{E}_M$ with respect to the filter-coefficient vector $\mathbf{h}_M(n)$ yields the set of linear equations

$$\mathbf{R}_M(n)\mathbf{h}_M(n) = \mathbf{D}_M(n) \tag{13.3.6}$$

where $\mathbf{R}_M(n)$ is the signal (estimated) correlation matrix defined as

$$\mathbf{R}_M(n) = \sum_{l=0}^{n} w^{n-l}\mathbf{X}_M^*(l)\mathbf{X}_M^t(l) \tag{13.3.7}$$

and $\mathbf{D}_M(n)$ is the (estimated) crosscorrelation vector

$$\mathbf{D}_M(n) = \sum_{l=0}^{n} w^{n-l}\mathbf{X}_M^*(l)d(l) \tag{13.3.8}$$

The solution of (13.3.6) is

$$\mathbf{h}_M(n) = \mathbf{R}_M^{-1}(n)\mathbf{D}_M(n) \tag{13.3.9}$$

Clearly, the matrix $\mathbf{R}_M(n)$ is akin to the statistical autocorrelation matrix $\mathbf{\Gamma}_M$, and the vector $\mathbf{D}_M(n)$ is akin to the crosscorrelation vector $\boldsymbol{\gamma}_d$, defined previously. We emphasize, however, that $\mathbf{R}_M(n)$ is not a Toeplitz matrix, whereas $\mathbf{\Gamma}_M$ is. We should also mention that for small values of n, $\mathbf{R}_M(n)$ may be ill conditioned, so that its inverse is not computable. In such a case, it is customary to initially add the matrix $\delta\mathbf{I}_M$ to $\mathbf{R}_M(n)$, where $\mathbf{I}_M$ is an identity matrix and δ is a small positive constant. With exponential weighting into the past, the effect of adding $\delta\mathbf{I}_M$ dissipates with time.

Now, suppose that we have the solution of (13.3.9) at time $n-1$—that is, we have $\mathbf{h}_M(n-1)$, and we wish to compute $\mathbf{h}_M(n)$. It is inefficient and, hence, impractical to solve the set of M linear equations for each new signal component. Instead, we may compute the matrix and vectors recursively. First, $\mathbf{R}_M(n)$ may be computed recursively as

$$\mathbf{R}_M(n) = w\mathbf{R}_M(n-1) + \mathbf{X}_M^*(n)\mathbf{X}_M^t(n) \tag{13.3.10}$$

We call (13.3.10) the *time-update equation* for $\mathbf{R}_M(n)$.

Since the inverse of $\mathbf{R}_M(n)$ is needed, we use the matrix inversion lemma (see Householder (1964)),

$$\mathbf{R}_M^{-1}(n) = \frac{1}{w}\left[\mathbf{R}_M^{-1}(n-1) - \frac{\mathbf{R}_M^{-1}(n-1)\mathbf{X}_M^*(n)\mathbf{X}_M^t(n)\mathbf{R}_M^{-1}(n-1)}{w + \mathbf{X}_M^t(n)\mathbf{R}_M^{-1}(n-1)\mathbf{X}_M^*(n)}\right] \tag{13.3.11}$$

Thus, $\mathbf{R}_M^{-1}(n)$ may be computed recursively.

For convenience, we define $\mathbf{P}_M(n) = \mathbf{R}_M^{-1}(n)$. It is also convenient to define an M-dimensional vector $\mathbf{K}_M(n)$, sometimes called the *Kalman gain vector*, as

$$\mathbf{K}_M(n) = \frac{1}{w + \mu_M(n)}\mathbf{P}_M(n-1)\mathbf{X}_M^*(n) \tag{13.3.12}$$

where $\mu_M(n)$ is a scalar defined as

$$\mu_M(n) = \mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n) \tag{13.3.13}$$

With these definitions, (13.3.11) becomes

$$\mathbf{P}_M(n) = \frac{1}{w}\left[\mathbf{P}_M(n-1) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\right] \tag{13.3.14}$$

Let us postmultiply (13.3.14) by $\mathbf{X}_M^*(n)$. Then,

$$\begin{aligned}\mathbf{P}_M(n)\mathbf{X}_M^*(n) &= \frac{1}{w}\left[\mathbf{P}_M(n-1)\mathbf{X}_M^*(n) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)\right] \\ &= \frac{1}{w}\{[w+\mu_M(n)]\mathbf{K}_M(n) - \mathbf{K}_M(n)\mu_M(n)\} = \mathbf{K}_M(n)\end{aligned} \tag{13.3.15}$$

Therefore, the Kalman gain vector may also be defined as $\mathbf{P}_M(n)\mathbf{X}_M^*(n)$.

Now, we may use the matrix inversion lemma to derive an equation for computing the filter coefficients recursively. Since

$$\mathbf{h}_M(n) = \mathbf{P}_M(n)\mathbf{D}_M(n) \tag{13.3.16}$$

and

$$\mathbf{D}_M(n) = w\mathbf{D}_M(n-1) + d(n)\mathbf{X}_M^*(n) \tag{13.3.17}$$

we have, upon substitution of (13.3.14) and (13.3.17) into (13.3.9),

$$\begin{aligned}\mathbf{h}_M(n) &= \frac{1}{w}\left[\mathbf{P}_M(n-1) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\right] \\ &\quad \times \left[w\mathbf{D}_M(n-1) + d(n)\mathbf{X}_M^*(n)\right] \\ &= \mathbf{P}_M(n-1)\mathbf{D}_M(n-1) + \frac{1}{w}d(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n) \\ &\quad - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{D}_M(n-1) \\ &\quad - \frac{1}{w}d(n)\mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n) \\ &= \mathbf{h}_M(n-1) + \mathbf{K}_M(n)\left[d(n) - \mathbf{X}_M^t(n)\mathbf{h}_M(n-1)\right]\end{aligned} \tag{13.3.18}$$

We observe that $\mathbf{X}_M^t(n)\mathbf{h}_M(n-1)$ is the output of the adaptive filter at time n based on use of the filter coefficients at time $n-1$. Since

$$\mathbf{X}_M^t(n)\mathbf{h}_M(n-1) = \hat{d}(n, n-1) \equiv \hat{d}(n) \tag{13.3.19}$$

and

$$e_M(n, n-1) = d(n) - \hat{d}(n, n-1) \equiv e_M(n) \tag{13.3.20}$$

it follows that the time-update equation for $\mathbf{h}_M(n)$ may be expressed as

$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)e_M(n) \tag{13.3.21}$$

or, equivalently,

$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{P}_M(n)\mathbf{X}_M^*(n)e_M(n) \tag{13.3.22}$$

To summarize, suppose we have the optimum filter coefficients $\mathbf{h}_M(n-1)$, the matrix $\mathbf{P}_M(n-1)$, and the vector $\mathbf{X}_M(n-1)$. When the new signal component $x(n)$ is obtained, we form the vector $\mathbf{X}_M(n)$ by dropping the term $x(n-M)$ from $\mathbf{X}_M(n-1)$ and adding the term $x(n)$ as the first element. Then, the recursive computation for the filter coefficients proceeds as follows:

1. Compute the filter output:

$$\hat{d}(n) = \mathbf{X}_M^t(n)\mathbf{h}_M(n-1) \tag{13.3.23}$$

2. Compute the error:

$$e_M(n) = d(n) - \hat{d}(n) \tag{13.3.24}$$

3. Compute the Kalman gain vector:

$$\mathbf{K}_M(n) = \frac{\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)}{w + \mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)} \tag{13.3.25}$$

4. Update the inverse of the correlation matrix

$$\mathbf{P}_M(n) = \frac{1}{w}\left[\mathbf{P}_M(n-1) - \mathbf{K}_M(n)\mathbf{X}_M^t(n)\mathbf{P}_M(n-1)\right] \tag{13.3.26}$$

5. Update the coefficient vector of the filter

$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)e_M(n) \tag{13.3.27}$$

The recursive algorithm specified by (13.3.23) through (13.3.27) is called *the direct-form recursive least-squares* (RLS) *algorithm*. It is initialized by setting $\mathbf{h}_M(-1) = \mathbf{0}$ and $\mathbf{P}_M(-1) = 1/\delta \mathbf{I}_M$, where δ is a small positive number.

The residual MSE resulting from the preceding optimization is

$$\mathscr{E}_{M\min} = \sum_{l=0}^{n} w^{n-l}|d(l)|^2 - \mathbf{h}_M^t(n)\mathbf{D}_M^*(n) \tag{13.3.28}$$

From (13.3.27), we observe that the filter coefficients vary with time by an amount equal to the error $e_M(n)$ multiplied by the Kalman gain vector $\mathbf{K}_M(n)$. Since $\mathbf{K}_M(n)$ is an M-dimensional vector, each filter coefficient is controlled by one of the elements of $\mathbf{K}_M(n)$. Consequently, rapid convergence is obtained. In contrast, the time-update equation for the coefficients of the filter adjusted by use of the LMS algorithm is

$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \Delta \mathbf{X}^*(n)e_M(n) \tag{13.3.29}$$

which has only the single parameter Δ for controlling the adjustment rate of the coefficients.

13.3.2 The LDU Factorization and Square-Root Algorithms

The RLS algorithm is very susceptible to round-off noise in an implementation of the algorithm with finite-precision arithmetic. The major problem with round-off errors occurs in the updating of $\mathbf{P}_M(n)$. To remedy this problem, we may perform a decomposition of either the correlation matrix $\mathbf{R}_M(n)$ or its inverse $\mathbf{P}_M(n)$.

To be specific, let us consider an LDU (lower-triangular/ diagonal/upper-triangular) decomposition of $\mathbf{P}_M(n)$. We may write

$$\mathbf{P}_M(n) = \mathbf{L}_M(n)\bar{\mathbf{D}}_M(n)\mathbf{L}_M^H(n) \tag{13.3.30}$$

where $\mathbf{L}_M(n)$ is a lower-triangular matrix with elements l_{ik}, $\bar{\mathbf{D}}_M(n)$ is a diagonal matrix with elements δ_k, and $\mathbf{L}_M^H(n)$ is an upper-triangular matrix. The diagonal elements of $\mathbf{L}_M(n)$ are set to unity (i.e., $l_{ii} = 1$). Now, instead of computing $\mathbf{P}_M(n)$ recursively, we can determine a formula for updating the factors $\mathbf{L}_M(n)$ and $\bar{\mathbf{D}}_M(n)$ directly, thus avoiding the computation of $\mathbf{P}_M(n)$.

The desired update formula is obtained by substituting the factored form of $\mathbf{P}_M(n)$ into (13.3.26). Thus, we have

$$\begin{aligned}&\mathbf{L}_M(n)\bar{\mathbf{D}}_M(n)\mathbf{L}_M^H(n)\\ &\quad = \frac{1}{w}\mathbf{L}_M(n-1)\left[\bar{\mathbf{D}}_M(n-1) - \frac{1}{w+\mu_M(n)}\mathbf{V}_M(n-1)\mathbf{V}_M^H(n-1)\right]\mathbf{L}_M^H(n-1)\end{aligned} \tag{13.3.31}$$

where, by definition,

$$\mathbf{V}_M(n-1) = \bar{\mathbf{D}}_M(n-1)\mathbf{L}_M^H(n-1)\mathbf{X}_M^*(n) \tag{13.3.32}$$

The term inside the brackets in (13.3.31) is a Hermitian matrix and may be expressed in an LDU factored form as

$$\begin{aligned}&\hat{\mathbf{L}}_M(n-1)\hat{\mathbf{D}}_M(n-1)\hat{\mathbf{L}}_M^H(n-1)\\ &\quad = \bar{\mathbf{D}}_M(n-1) - \frac{1}{w+\mu_M(n)}\mathbf{V}_M(n-1)\mathbf{V}_M^H(n-1)\end{aligned} \tag{13.3.33}$$

Then, if we substitute (13.3.33) into (13.3.31), we obtain

$$\mathbf{L}_M(n)\bar{\mathbf{D}}_M(n)\hat{\mathbf{L}}_M^H(n) = \frac{1}{w}\left[\mathbf{L}_M(n-1)\hat{\mathbf{L}}_M(n-1)\hat{\mathbf{D}}_M(n-1)\hat{\mathbf{L}}_M^H(n-1)\mathbf{L}_M^H(n-1)\right] \tag{13.3.34}$$

Consequently, the desired update relations are

$$\begin{aligned}\mathbf{L}_M(n) &= \mathbf{L}_M(n-1)\hat{\mathbf{L}}_M(n-1)\\ \bar{\mathbf{D}}_M(n) &= \frac{1}{w}\hat{\mathbf{D}}_M(n-1)\end{aligned} \tag{13.3.35}$$

To determine the factors $\hat{\mathbf{L}}_M(n-1)$ and $\hat{\mathbf{D}}_M(n-1)$, we need to factor the matrix on the right-hand side (RHS) of (13.3.33). This factorization may be expressed by the set of linear equations

$$\sum_{k=1}^{j} l_{ik}d_k l_{jk}^* = p_{ij}, \qquad 1 \le j \le i-1, \qquad i \ge 2 \tag{13.3.36}$$

where $\{d_k\}$ are the elements of $\hat{\mathbf{D}}_M(n-1)$, $\{l_{ik}\}$ are elements of $\hat{\mathbf{L}}_M(n-1)$ and $\{p_{ij}\}$ are the elements of the matrix on the RHS of (13.3.33). Then, $\{l_{ik}\}$ and $\{d_k\}$ are determined as follows:

$$\begin{aligned}d_1 &= p_{11}\\ l_{ij}d_j &= p_{ij} - \sum_{k=1}^{j-1} l_{ik}d_k l_{jk}^*, \qquad 1 \le j \le i-1, \quad 2 \le i \le M\\ d_i &= p_{ii} - \sum_{k=1}^{i-1} |l_{ik}|^2 d_k, \qquad 2 \le i \le M\end{aligned} \tag{13.3.37}$$

TABLE 13.1 LDU Form of Square-Root RLS Algorithm

for $j = 1, \ldots, 2, \ldots, M$ do
  $f_j = x_j^*(n)$
end loop j
for $j = 1, 2, \ldots, M-1$ do
  for $i = j+1, j+2, \ldots, M$ do
    $f_j = f_j + l_{ij}(n-1) f_i$
end loop j
for $j = 1, 2, \ldots, M$ do
    $\bar{d}_j(n) = d_j(n-1)/w$
    $v_j = \bar{d}_j(n) f_j$
end loop j
$\alpha_M = 1 + v_M f_M^*$
$d_M(n) = \bar{d}_M(n)/\alpha_M$
$\bar{k}_M = v_M$
for $j = M-1, M-2, \ldots, 1$ do
  $\bar{k}_j = v_j$
  $\alpha_j = \alpha_{j+1} + v_j f_j^*$
  $\lambda_j = f_j/\alpha_{j+1}$
  $d_j(n) = \bar{d}_j(n)\alpha_{j+1}/\alpha_1$
  for $i = M, M-1, \ldots, j+1$ do
    $l_{ij}(n) = l_{ij}(n-1) + \bar{k}_i^* \lambda_j$
    $\bar{k}_i = \bar{k}_i + v_j l_{ij}^*(n-1)$ down to $j = 2$)
  end loop i
end loop j
$\bar{\mathbf{K}}_M(n) = [\bar{k}_1, \bar{k}_2, \ldots, \bar{k}_M]^t$
$e_M(n) = d(n) - \bar{d}(n)$
$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + [e_M(n)/\alpha_1]\bar{\mathbf{K}}_M(n)$

The resulting algorithm, obtained from the time-update equations in (13.3.35), depends directly on the data vector $\mathbf{X}_M(n)$ and not on the "square" of the data vector. Thus, the squaring operation of the data vector is avoided and, consequently, the effect of round-off errors is significantly reduced.

The RLS algorithms obtained from an LDU decomposition of either $\mathbf{R}_M(n)$ or $\mathbf{P}_M(n)$ are called *square-root RLS algorithms*. Bierman (1977), Carlson and Culmone (1979), and Hsu (1982) treat these types of algorithms. A square-root RLS algorithm based on the LDU decomposition of $\mathbf{P}_M(n)$, as just described, is given in Table 13.1. Its computational complexity is proportional to M^2.

13.3.3 Fast RLS Algorithms

The RLS direct-form algorithm and the square-root algorithms have a computational complexity proportional to M^2, as indicated. On the other hand, the RLS lattice algorithms derived in Section 13.4 have a computational complexity proportional to M. Basically, the lattice algorithms avoid the matrix multiplications involved in computing the Kalman gain vector $\mathbf{K}_M(n)$.

TABLE 13.2 Fast RLS Algorithm: Version A

$$f_{M-1}(n) = x(n) + \mathbf{a}_{M-1}^t(n-1)\mathbf{X}_{M-1}(n-1)$$

$$g_{M-1}(n) = x(n-M+1) + \mathbf{b}_{M-1}^t(n-1)\mathbf{X}_{M-1}(n)$$

$$\mathbf{a}_{M-1}(n) = \mathbf{a}_{M-1}(n-1) - \mathbf{K}_{M-1}(n-1)f_{M-1}(n)$$

$$f_{M-1}(n,n) = x(n) + \mathbf{a}_{M-1}^t(n)\mathbf{X}_{M-1}(n-1)$$

$$E_{M-1}^f(n) = wE_{M-1}^f(n-1) + f_{M-1}(n)f_{M-1}^*(n,n)$$

$$\begin{bmatrix} \mathbf{C}_{M-1}(n) \\ c_{\mathrm{MM}}(n) \end{bmatrix} \equiv \mathbf{K}_M(n) = \begin{bmatrix} 0 \\ \mathbf{K}_{M-1}(n-1) \end{bmatrix} + \frac{f_{M-1}^*(n,n)}{E_{M-1}^f(n)} \begin{bmatrix} 1 \\ \mathbf{a}_{M-1}(n) \end{bmatrix}$$

$$\mathbf{K}_{M-1}(n) = \frac{\mathbf{C}_{M-1}(n) - c_{\mathrm{MM}}(n)\mathbf{b}_{M-1}(n-1)}{1 - c_{\mathrm{MM}}(n)g_{M-1}(n)}$$

$$\mathbf{b}_{M-1}(n) = \mathbf{b}_{M-1}(n-1) - \mathbf{K}_{M-1}(n)g_{M-1}(n)$$

$$\hat{d}(n) = \mathbf{h}_M^t(n-1)\mathbf{X}_M(n)$$

$$e_M(n) = d(n) - \hat{d}(n)$$

$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)e_M(n)$$

Initialization

$$\mathbf{a}_{M-1}(-1) = \mathbf{b}_{M-1}(-1) = \mathbf{0}$$

$$\mathbf{K}_{M-1}(-1) = \mathbf{0}$$

$$\mathbf{h}_{M-1}(-1) = \mathbf{0}$$

$$E_{M-1}^f(-1) = \epsilon, \quad \epsilon > 0$$

By using the forward and backward prediction formulas derived in Section 13.4 for the RLS lattice, it is possible to obtain time-update equations for the Kalman gain vector that completely avoid matrix multiplications. The resulting algorithms have a complexity that is proportional to M (multiplications and divisions) and, hence, they are called *fast RLS algorithms* for direct-form FIR filters.

There are several versions of fast algorithms, which differ in minor ways. Two versions are given in Tables 13.2 and 13.3 for complex-valued signals. The variables used in the fast algorithms listed in these tables are defined in Section 13.4. The computational complexity for Version A is $10M - 4$ (complex) multiplications and divisions, whereas Version B has a complexity of $9M + 1$ multiplications and divisions. Further reduction of computational complexity to $7M$ is possible. For example, Carayannis, Manolakis, and Kalouptsidis (1983) describe a fast RLS algorithm, termed the FAEST (fast a posteriori error sequential technique) algorithm, with a computational complexity of $7M$; this algorithm is given in Section 13.4. Other versions of these algorithms with a complexity of $7M$ have been proposed, but many of these algorithms are extremely sensitive to round-off noise and exhibit instability problems (Falconer and Ljung (1978), Carayannis, Manolakis, and Kalouptsidis (1983; 1986) and Cioffi and Kailath (1984)). Slock and Kailath (1988; 1991) have shown how to

TABLE 13.3 Fast RLS Algorithm: Version B

$$f_{M-1}(n) = x(n) + \mathbf{a}'_{M-1}(n-1)\mathbf{X}_{M-1}(n-1)$$

$$g_{M-1}(n) = x(n-M+1) + \mathbf{b}'_{M-1}(n-1)\mathbf{X}_{M-1}(n)$$

$$\mathbf{a}_{M-1}(n) = \mathbf{a}_{M-1}(n-1) - \mathbf{K}_{M-1}(n-1)f_{M-1}(n)$$

$$f_{M-1}(n,n) = \alpha_{M-1}(n-1)f_{M-1}(n)$$

$$E^f_{M-1}(n) = wE^f_{M-1}(n-1) + \alpha_{M-1}(n-1)|f_{M-1}(n)|^2$$

$$\begin{bmatrix} \mathbf{C}_{M-1}(n) \\ c_{\mathrm{MM}}(n) \end{bmatrix} \equiv \mathbf{K}_M(n) = \begin{bmatrix} 0 \\ \mathbf{K}_{M-1}(n-1) \end{bmatrix} + \frac{f^*_{M-1}(n,n)}{E^f_{M-1}(n)} \begin{bmatrix} 1 \\ \mathbf{a}_{M-1}(n) \end{bmatrix}$$

$$\mathbf{K}_{M-1}(n) = \frac{\mathbf{C}_{M-1}(n) - c_{\mathrm{MM}}(n)\mathbf{b}_{M-1}(n-1)}{1 - c_{\mathrm{MM}}(n)g_{M-1}(n)}$$

$$\mathbf{b}_{M-1}(n) = \mathbf{b}_{M-1}(n-1) - \mathbf{K}_{M-1}(n)g_{M-1}(n)$$

$$\alpha_{M-1}(n) = \alpha_{M-1}(n-1)\left[\frac{1 - \frac{f_{M-1}(n)f^*_{M-1}(n,n)}{E^f_{M-1}(n)}}{1 - c_{\mathrm{MM}}(n)g_{M-1}(n)}\right]$$

$$\hat{d}(n) = \mathbf{h}'_M(n-1)\mathbf{X}_M(n)$$

$$e_M(n) = d(n) - \hat{d}(n)$$

$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \mathbf{K}_M(n)e_M(n)$$

Initialization

$$\mathbf{a}_{M-1}(-1) = \mathbf{b}_{M-1}(-1) = \mathbf{0}$$

$$\mathbf{K}_{M-1}(-1) = \mathbf{0}, \quad \mathbf{h}_{M-1}(-1) = \mathbf{0}$$

$$E^f_{M-1}(-1) = \epsilon > 0$$

stabilize these fast ($7M$) algorithms with a relatively small increase in the number of computations; two stabilized fast RLS algorithms are given in Section 13.4.

13.3.4 Properties of the Direct-Form RLS Algorithms

A major advantage of the direct-form RLS algorithms over the LMS algorithm is their faster convergence rate. This characteristic behavior is illustrated in Figure 13.3.1, which shows the convergence rate of the LMS and direct-form RLS algorithms for an adaptive FIR channel equalizer of length $M = 11$. The statistical autocorrelation matrix $\mathbf{\Gamma}_M$ for the received signal has an eigenvalue ratio of $\lambda_{\max}/\lambda_{\min} = 11$. All the equalizer coefficients were initially set to zero. The step size for the LMS algorithm was selected as $\Delta = 0.02$, which represents a good compromise between convergence rate and excess MSE.

The superiority of the RLS algorithm in achieving faster convergence is clearly evident. The algorithm converges in less than 70 iterations (70 signal samples) while

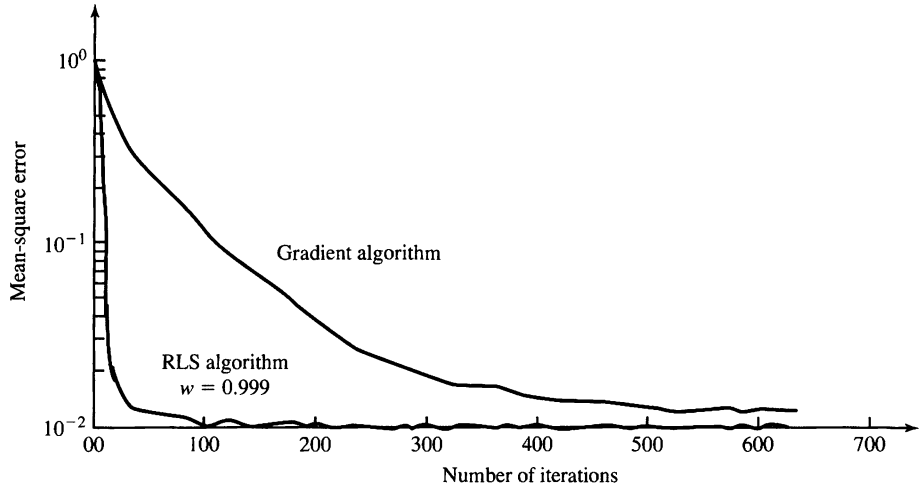


Figure 13.3.1 Learning curves for RLS and LMS algorithms for adaptive equalizer of length $M = 11$. The eigenvalue spread of the channel is $\lambda_{max}/\lambda_{min} = 11$. The step size for the LMS algorithm is $\Delta = 0.02$. (From *Digital Communication* by John G. Proakis, © 1983 by McGraw-Hill Book Company. Reprinted with permission of the publisher.)

the LMS algorithm has not converged in over 600 iterations. This rapid rate of convergence of the RLS algorithm is extremely important in applications in which the signal statistics vary rapidly with time. For example, the time variations of the characteristics of an ionospheric high-frequency (HF) radio channel result in frequent fading of the signal to the point where the signal strength is comparable to or even lower than the additive noise. During a signal fade, both the LMS and RLS algorithms are unable to track the channel characteristics. As the signal emerges from the fade, the channel characteristics are generally different than those prior to the fade. In such a case, the LMS algorithm is slow to adapt to the new channel characteristics. On the other hand, the RLS algorithm adapts sufficiently fast to track such rapid variations (Hsu (1982)).

Despite their superior convergence rate, the RLS algorithms for FIR adaptive filtering described in the previous section have two important disadvantages. One is their computational complexity. The square-root algorithms have a complexity proportional to M^2. The fast RLS algorithms have a computational complexity proportional to M, but the proportionality factor is four to five times that of the LMS algorithm.

The second disadvantage of the algorithms is their sensitivity to round-off errors that accumulate as a result of the recursive computations. In some cases, the round-off errors cause these algorithms to become unstable.

The numerical properties of the RLS algorithms have been investigated by several researchers, including Ling and Proakis (1984a), Ljung and Ljung (1985), and Cioffi (1987b). For illustrative purposes, Table 13.4 includes simulation results on the steady-state (time-averaged) square error for the RLS square-root algorithm, the fast RLS algorithm in Table 13.2, and the LMS algorithm, for different word lengths.

TABLE 13.4 Numerical Accuracy of FIR Adaptive Filtering Algorithms (Least-Squares Error $\times 10^{-3}$)

Number of bits (including sign)	Algorithm		
	RLS square root	Fast RLS	LMS
16	2.17	2.17	2.30
13	2.33	2.21	2.30
11	6.14	3.34	19.0
10	17.6	[a]	77.2
9	75.3	[a]	311.0
8	[a]	[a]	1170.0

[a] Algorithm does not converge to optimum coefficients.

The simulation was performed with a linear adaptive equalizer having $M = 11$ coefficients. The channel had an eigenvalue ratio of $\lambda_{max}/\lambda_{min} = 11$. The exponential weighting factor used in the RLS algorithms was $w = 0.975$ and the step size for the LMS algorithm was $\Delta = 0.025$. The additive noise has a variance of 0.001. The output MSE with infinite precision is 2.1×10^{-3}.

We should indicate that the direct-form RLS algorithm becomes unstable and, hence, does not work properly with 16-bit fixed-point arithmetic. For this algorithm, we found experimentally that approximately 20–24 bits of precision are needed for the algorithm to work properly. On the other hand, the square-root algorithm works down to about 9 bits, but the degradation in performance below 11 bits is significant. The fast RLS algorithm works well down to 11 bits for short durations of the order of 500 iterations. For a much larger number of iterations, the algorithm becomes unstable due to the accumulation of round-off errors. In such a case, several methods have been proposed to restart the algorithm in order to prevent overflow in the coefficients. The interested reader may refer to Eleftheriou and Falconer (1987), Cioffi and Kailath (1984), and Hsu (1982). Alternatively, one may modify the algorithm as proposed by Slock and Kailath (1988; 1991) and, thus, stabilize it.

We also observe from the results of Table 13.4 that the LMS algorithm is quite robust to round-off noise. It deteriorates as expected with a decrease in the precision of the filter coefficients, but no catastrophic failure (instability) occurs with 8 or 9 bits of precision. However, the degradation in performance below 12 bits is significant.

13.4 Adaptive Lattice-Ladder Filters

In Chapters 9 and 12, we demonstrated that an FIR filter may also be realized as a lattice structure in which the lattice parameters, called the *reflection coefficients,* are related to the filter coefficients in the direct-form FIR structure. The method for converting the FIR filter coefficients into the reflection coefficients (and vice versa) was also described.

In this section, we derive adaptive filtering algorithms in which the filter structure is a lattice or a lattice-ladder. These adaptive lattice-ladder filter algorithms,

based on the method of least squares, have several desirable properties, including computational efficiency and robustness to round-off errors. From the development of the RLS lattice-ladder algorithms, we obtain the fast RLS algorithms that were described in Section 13.3.3.

13.4.1 Recursive Least-Squares Lattice-Ladder Algorithms

In Chapter 12, we showed the relationship between the lattice filter structure and a linear predictor, and derived the equations that relate the predictor coefficients to the reflection coefficients of the lattice (and vice versa). We also established the relationship between the Levinson-Durbin recursions for the linear predictor coefficients and the reflection coefficients in the lattice filter. From these developments, we would expect to obtain the recursive least-squares lattice filter by formulating the least-squares estimation problem in terms of linear prediction. This is the approach that we take.

The recursive least-squares algorithms for the direct-form FIR structures described in Section 13.3.1 are recursive in time only. The length of the filter is fixed. A change (increase or decrease) in the filter length results in a new set of filter coefficients that are totally different from the previous set.

In contrast, the lattice filter is order recursive. As a consequence, the number of sections that it contains can be easily increased or decreased without affecting the reflection coefficients of the remaining sections. This and several other advantages (described in this and subsequent sections) make the lattice filter very attractive for adaptive filtering applications.

To begin, suppose that we observe the signal $x(n-l)$, $l = 1, 2, \ldots, m$, and let us consider the linear prediction of $x(n)$. Let $f_m(l, n)$ denote the forward prediction error for an mth-order predictor, defined as

$$f_m(l, n) = x(l) + \mathbf{a}_m^t(n)\mathbf{X}_m(l-1) \tag{13.4.1}$$

where the vector $-\mathbf{a}_m(n)$ consists of the forward prediction coefficients, that is,

$$\mathbf{a}_m^t(n) = [a_m(1, n)\ a_m(2, n) \cdots a_m(m, n)] \tag{13.4.2}$$

and the data vector $\mathbf{X}_m(l-1)$ is

$$\mathbf{X}_m^t(l-1) = [x(l-1)\ x(l-2) \cdots x(l-m)] \tag{13.4.3}$$

The predictor coefficients $\mathbf{a}_m(n)$ are selected to minimize the time-averaged weighted squared error

$$\mathcal{E}_m^f(n) = \sum_{l=0}^{n} w^{n-l} |f_m(l, n)|^2 \tag{13.4.4}$$

The minimization of $\mathcal{E}_m^f(n)$ with respect to $\mathbf{a}_m(n)$ leads to the set of linear equations

$$\mathbf{R}_m(n-1)\mathbf{a}_m(n) = -\mathbf{Q}_m(n) \tag{13.4.5}$$

where $\mathbf{R}_m(n)$ is defined by (13.3.7) and $\mathbf{Q}_m(n)$ is defined as

$$\mathbf{Q}_m(n) = \sum_{l=0}^{n} w^{n-l} x(l) \mathbf{X}_m^*(l-1) \tag{13.4.6}$$

The solution of (13.4.5) is

$$\mathbf{a}_m(n) = -\mathbf{R}_m^{-1}(n-1)\mathbf{Q}_m(n) \tag{13.4.7}$$

The minimum value of $\mathscr{E}_m^f(n)$, obtained with the linear predictor specified by (13.4.7), is denoted as $E_m^f(n)$ and is given by

$$\begin{aligned} E_m^f(n) &= \sum_{l=0}^{n} w^{n-l} x^*(l) \left[x(l) + \mathbf{a}_m^t(n) \mathbf{X}_m(l-1) \right] \\ &= q(n) + \mathbf{a}_m^t(n) \mathbf{Q}_m^*(n) \end{aligned} \tag{13.4.8}$$

where $q(n)$ is defined as

$$q(n) = \sum_{l=0}^{n} w^{n-l} |x(l)|^2 \tag{13.4.9}$$

The linear equations in (13.4.5) and the equation for $E_m^f(n)$ in (13.4.8) can be combined in a single matrix equation of the form

$$\begin{bmatrix} q(n) & \mathbf{Q}_m^H(n) \\ \mathbf{Q}_m(n) & \mathbf{R}_m(n-1) \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{a}_m(n) \end{bmatrix} = \begin{bmatrix} E_m^f(n) \\ \mathbf{O}_m \end{bmatrix} \tag{13.4.10}$$

where $\mathbf{O}_m$ is the m-dimensional null vector. It is interesting to note that

$$\begin{aligned} \mathbf{R}_{m+1}(n) &= \sum_{l=0}^{n} w^{n-l} \mathbf{X}_{m+1}^*(l) \mathbf{X}_{m+1}^t(l) \\ &= \sum_{l=0}^{n} w^{n-l} \begin{bmatrix} x^*(l) \\ \mathbf{X}_m^*(l-1) \end{bmatrix} \left[x(l) \mathbf{X}_m^t(l-1) \right] \\ &= \begin{bmatrix} q(n) & \mathbf{Q}_m^H(n) \\ \mathbf{Q}_m(n) & \mathbf{R}_m(n-1) \end{bmatrix} \end{aligned} \tag{13.4.11}$$

which is the matrix in (13.4.10).

In a completely parallel development to (13.4.1) through (13.4.11), we minimize the backward time-averaged weighted squared error for an mth-order backward predictor defined as

$$\mathscr{E}_m^b(n) = \sum_{l=0}^{n} w^{n-l} |g_m(l, n)|^2 \tag{13.4.12}$$

where the backward error is defined as

$$g_m(l, n) = x(l - m) + \mathbf{b}_m^t(n)\mathbf{X}_m(l) \tag{13.4.13}$$

and $\mathbf{b}_m^t(n) = [b_m(1, n)\ b_m(2, n) \ldots b_m(m, n)]$ is the vector of coefficients for the backward predictor. The minimization of $\mathscr{E}_m^b(n)$ leads to the equation

$$\mathbf{R}_m(n)\mathbf{b}_m(n) = -\mathbf{V}_m(n) \tag{13.4.14}$$

and, hence, to the solution

$$\mathbf{b}_m(n) = -\mathbf{R}_m^{-1}(n)\mathbf{V}_m(n) \tag{13.4.15}$$

where

$$\mathbf{V}_m(n) = \sum_{l=0}^{n} w^{n-l} x(l - m)\mathbf{X}_m^*(l) \tag{13.4.16}$$

The minimum value of $\mathscr{E}_m^b(n)$, denoted as $E_m^b(n)$, is

$$\begin{aligned} E_m^b(n) &= \sum_{l=0}^{n} w^{n-l}\left[x(l - m) + \mathbf{b}_m^t(n)\mathbf{X}_m(l)\right]x^*(l - m) \\ &= v(n) + \mathbf{b}_m^t(n)\mathbf{V}_m^*(n) \end{aligned} \tag{13.4.17}$$

where the scalar quantity $v(n)$ is defined as

$$v(n) = \sum_{l=0}^{n} w^{n-l}|x(l - m)|^2 \tag{13.4.18}$$

If we combine (13.4.14) and (13.4.17) into a single equation, we obtain

$$\begin{bmatrix} \mathbf{R}_m(n) & \mathbf{V}_m(n) \\ \mathbf{V}_m^H(n) & v(n) \end{bmatrix} \begin{bmatrix} \mathbf{b}_m(n) \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{O}_m \\ E_m^b(n) \end{bmatrix} \tag{13.4.19}$$

We also note that the (estimated) autocorrelation matrix $\mathbf{R}_{m+1}(n)$ can be expressed as

$$\begin{aligned} \mathbf{R}_{m+1}(n) &= \sum_{l=0}^{n} w^{n-l} \begin{bmatrix} \mathbf{X}_m^*(l) \\ x^*(l - m) \end{bmatrix} \left[\mathbf{X}_m^t(l)x(l - m)\right] \\ &= \begin{bmatrix} \mathbf{R}_m(n) & \mathbf{V}_m(n) \\ \mathbf{V}_m^H(n) & v(n) \end{bmatrix} \end{aligned} \tag{13.4.20}$$

Thus, we have obtained the equations for the forward and backward least-squares predictors of order m.

Next, we derive the order-update equations for these predictors, which will lead us to the lattice filter structure. In deriving the order-update equations for $\mathbf{a}_m(n)$ and

$\mathbf{b}_m(n)$, we will make use of the two matrix inversion identities for a matrix of the form

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \tag{13.4.21}$$

where $\mathbf{A}$, $\mathbf{A}_{11}$, and $\mathbf{A}_{22}$ are square matrices. The inverse of $\mathbf{A}$ is expressible in two different forms, namely,

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} + \mathbf{A}_{11}^{-1}\mathbf{A}_{12}\tilde{\mathbf{A}}_{22}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\tilde{\mathbf{A}}_{22}^{-1} \\ -\tilde{\mathbf{A}}_{22}^{-1}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \tilde{\mathbf{A}}_{22}^{-1} \end{bmatrix} \tag{13.4.22}$$

and

$$\mathbf{A}^{-1} = \begin{bmatrix} \tilde{\mathbf{A}}_{11}^{-1} & -\tilde{\mathbf{A}}_{11}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\tilde{\mathbf{A}}_{11}^{-1} & \mathbf{A}_{22}^{-1}\mathbf{A}_{21}\tilde{\mathbf{A}}_{11}^{-1}\mathbf{A}_{12}\mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1} \end{bmatrix} \tag{13.4.23}$$

where $\tilde{\mathbf{A}}_{11}$ and $\tilde{\mathbf{A}}_{12}$ are defined as

$$\begin{aligned} \tilde{\mathbf{A}}_{11} &= \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} \\ \tilde{\mathbf{A}}_{22} &= \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \end{aligned} \tag{13.4.24}$$

Order-Update Recursions. Now, let us use the formula in (13.4.22) to obtain the inverse of $\mathbf{R}_{m+1}(n)$ by using the form in (13.4.20). First, we have

$$\begin{aligned} \tilde{\mathbf{A}}_{22} &= v(n) - \mathbf{V}_m^H(n)\mathbf{R}_m^{-1}(n)\mathbf{V}_m(n) \\ &= v(n) + \mathbf{b}_m^t(n)\mathbf{V}_m^*(n) = E_m^b(n) \end{aligned} \tag{13.4.25}$$

and

$$\mathbf{A}_{11}^{-1}\mathbf{A}_{12} = \mathbf{R}_m^{-1}(n)\mathbf{V}_m(n) = -\mathbf{b}_m(n) \tag{13.4.26}$$

Hence,

$$\mathbf{R}_{m+1}^{-1}(n) \equiv \mathbf{P}_{m+1}(n) = \begin{bmatrix} \mathbf{P}_m(n) + \dfrac{\mathbf{b}_m(n)\mathbf{b}_m^H(n)}{E_m^b(n)} & \dfrac{\mathbf{b}_m(n)}{E_m^b(n)} \\ \dfrac{\mathbf{b}_m^H(n)}{E_m^b(n)} & \dfrac{1}{E_m^b(n)} \end{bmatrix}$$

or, equivalently,

$$\mathbf{P}_{m+1}(n) = \begin{bmatrix} \mathbf{P}_m(n) & 0 \\ 0 & 0 \end{bmatrix} + \frac{1}{E_m^b(n)} \begin{bmatrix} \mathbf{b}_m(n) \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_m^H(n) & 1 \end{bmatrix} \tag{13.4.27}$$

By substituting $n-1$ for n in (13.4.27) and postmultiplying the result by $-\mathbf{Q}_{m+1}(n)$, we obtain the order update for $\mathbf{a}_m(n)$. Thus,

$$\begin{aligned} \mathbf{a}_{m+1}(n) &= -\mathbf{P}_{m+1}(n-1)\mathbf{Q}_{m+1}(n) \\ &= \begin{bmatrix} \mathbf{P}_m(n-1) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -\mathbf{Q}_m^{(n)} \\ \cdots \end{bmatrix} \\ &\quad - \frac{1}{E_m^b(n-1)} \begin{bmatrix} \mathbf{b}_m(n-1) \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_m^H(n-1) & 1 \end{bmatrix} \mathbf{Q}_{m+1}(n) \\ &= \begin{bmatrix} \mathbf{a}_m(n) \\ 0 \end{bmatrix} - \frac{k_{m+1}(n)}{E_m^b(n-1)} \begin{bmatrix} \mathbf{b}_m(n-1) \\ 1 \end{bmatrix} \end{aligned} \tag{13.4.28}$$

where the scalar quantity $k_{m+1}(n)$ is defined as

$$k_{m+1}(n) = \begin{bmatrix} \mathbf{b}_m^H(n-1) & 1 \end{bmatrix} \mathbf{Q}_{m+1}(n) \tag{13.4.29}$$

The reader should observe that (13.4.28) is a Levinson-type recursion for the predictor coefficients.

To obtain the corresponding order update for $\mathbf{b}_m(n)$, we use the matrix inversion formula in (13.4.23) for the inverse of $\mathbf{R}_{m+1}(n)$, along with the form in (13.4.11). In this case, we have

$$\begin{aligned} \tilde{\mathbf{A}}_{11} &= q(n) - \mathbf{Q}_m^H(n)\mathbf{R}_m^{-1}(n-1)\mathbf{Q}_m(n) \\ &= q(n) + \mathbf{a}_m^t(n)\mathbf{Q}_m^*(n) = E_m^f(n) \end{aligned} \tag{13.4.30}$$

and

$$\mathbf{A}_{22}^{-1}\mathbf{A}_{21} = \mathbf{R}_m^{-1}(n-1)\mathbf{Q}_m(n) = -\mathbf{a}_m(n) \tag{13.4.31}$$

Hence,

$$\mathbf{P}_{m+1}(n) = \begin{bmatrix} \dfrac{1}{E_m^f(n)} & \dfrac{\mathbf{a}_m^H(n)}{E_m^f(n)} \\ \dfrac{\mathbf{a}_m(n)}{E_m^f(n)} & \mathbf{P}_m(n-1) + \dfrac{\mathbf{a}_m(n)\mathbf{a}_m^H(n)}{E_m^f(n)} \end{bmatrix}$$

or, equivalently,

$$\mathbf{P}_{m+1}(n) = \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{P}_m(n-1) \end{bmatrix} + \frac{1}{E_m^f(n)} \begin{bmatrix} 1 \\ \mathbf{a}_m(n) \end{bmatrix} \begin{bmatrix} 1 & \mathbf{a}_m^H(n) \end{bmatrix} \tag{13.4.32}$$

Now, if we postmultiply (13.4.32) by $-\mathbf{V}_{m+1}(n)$, we obtain

$$\begin{aligned} \mathbf{b}_{m+1}(n) &= \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{P}_m(n-1) \end{bmatrix} \begin{bmatrix} \cdots \\ -\mathbf{V}_m(n-1) \end{bmatrix} \\ &\quad - \frac{1}{E_m^f(n)} \begin{bmatrix} 1 \\ \mathbf{a}_m(n) \end{bmatrix} \begin{bmatrix} 1 & \mathbf{a}_m^H(n) \end{bmatrix} \mathbf{V}_{m+1}(n) \\ &= \begin{bmatrix} 0 \\ \mathbf{b}_m(n-1) \end{bmatrix} - \frac{k_{m+1}^*(n)}{E_m^f(n)} \begin{bmatrix} 1 \\ \mathbf{a}_m(n) \end{bmatrix} \end{aligned} \tag{13.4.33}$$

where

$$\begin{bmatrix} 1 & \mathbf{a}_m^H(n) \end{bmatrix} \mathbf{V}_{m+1}(n) = \begin{bmatrix} \mathbf{b}_m^t(n-1) & 1 \end{bmatrix} \mathbf{Q}_{m+1}^*(n) = k_{m+1}^*(n) \tag{13.4.34}$$

The proof of (13.4.34) and its relation to (13.4.29) is left as an excercise for the reader. Thus, (13.4.28) and (13.4.33) specify the order-update equations for $\mathbf{a}_m(n)$ and $\mathbf{b}_m(n)$, respectively.

The order-update equations for $E_m^f(n)$ and $E_m^b(n)$ may now be obtained. From the definition of $E_m^f(n)$ given by (13.4.8), we have

$$E_{m+1}^f(n) = q(n) + \mathbf{a}_{m+1}^t(n)\mathbf{Q}_{m+1}^*(n) \tag{13.4.35}$$

By substituting for $\mathbf{a}_{m+1}(n)$ from (13.4.28) into (13.4.35), we obtain

$$E_{m+1}^{f}(n) = q(n) + [\mathbf{a}_m'(n) \quad 0]\begin{bmatrix} \mathbf{Q}_m^*(n) \\ \cdots \end{bmatrix} - \frac{k_{m+1}(n)}{E_m^b(n-1)}[\mathbf{b}_m'(n-1) \quad 1]\mathbf{Q}_{m+1}^*(n)$$

$$= E_m^f(n) - \frac{|k_{m+1}(n)|^2}{E_m^b(n-1)} \tag{13.4.36}$$

Similarly, by using (13.4.17) and (13.4.33), we obtain the order update for $E_{m+1}^b(n)$, in the form

$$E_{m+1}^b(n) = E_m^b(n-1) - \frac{|k_{m+1}(n)|^2}{E_m^f(n)} \tag{13.4.37}$$

The lattice filter is specified by two coupled equations involving the forward and backward errors $f_m(n, n-1)$ and $g_m(n, n-1)$, respectively. From the definition of the forward error in (13.4.1), we have

$$f_{m+1}(n, n-1) = x(n) + \mathbf{a}_{m+1}^t(n-1)\mathbf{X}_{m+1}(n-1) \tag{13.4.38}$$

Substituting for $\mathbf{a}_{m+1}^t(n-1)$ from (13.4.28) into (13.4.38) yields

$$f_{m+1}(n, n-1) = x(n) + [\mathbf{a}_m^t(n-1) \quad 0]\begin{bmatrix} \mathbf{X}_m(n-1) \\ \cdots \end{bmatrix}$$

$$- \frac{k_{m+1}(n-1)}{E_m^b(n-2)}[\mathbf{b}_m^t(n-2) \quad 1]\mathbf{X}_{m+1}(n-1)$$

$$= f_m(n, n-1) - \frac{k_{m+1}(n-1)}{E_m^b(n-2)} \times [x(n-m-1) + \mathbf{b}_m^t(n-2)\mathbf{X}_m(n-1)]$$

$$= f_m(n, n-1) - \frac{k_{m+1}(n-1)}{E_m^b(n-2)} g_m(n-1, n-2) \tag{13.4.39}$$

To simplify the notation, we define

$$\begin{aligned} f_m(n) &= f_m(n, n-1) \\ g_m(n) &= g_m(n, n-1) \end{aligned} \tag{13.4.40}$$

Then, (13.4.39) may be expressed as

$$f_{m+1}(n) = f_m(n) - \frac{k_{m+1}(n-1)}{E_m^b(n-2)} g_m(n-1) \tag{13.4.41}$$

Similarly, beginning with the definition of the backward error given by (13.4.13), we have

$$g_{m+1}(n, n-1) = x(n-m-1) + \mathbf{b}_{m+1}^t(n-1)\mathbf{X}_{m+1}(n) \tag{13.4.42}$$

Substituting for $\mathbf{b}_{m+1}(n-1)$ from (13.4.33) and simplifying the result, we obtain

$$g_{m+1}(n, n-1) = g_m(n-1, n-2) - \frac{k_{m+1}^*(n-1)}{E_m^f(n-1)} f_m(n, n-1) \tag{13.4.43}$$

or, equivalently,

$$g_{m+1}(n) = g_m(n-1) - \frac{k_{m+1}^*(n-1)}{E_m^f(n-1)} f_m(n) \tag{13.4.44}$$

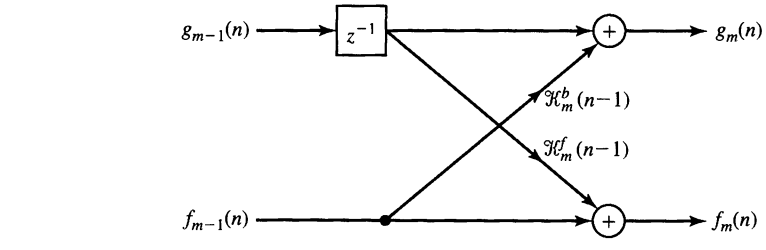
The two recursive equations in (13.4.41) and (13.4.44) specify the lattice filter illustrated in Figure 13.4.1 where, for notational convenience, we have defined the reflection coefficients for the lattice as

$$\begin{aligned} \mathcal{K}_m^f(n) &= \frac{-k_m(n)}{E_{m-1}^b(n-1)} \\ \mathcal{K}_m^b(n) &= \frac{-k_m^*(n)}{E_{m-1}^f(n)} \end{aligned} \tag{13.4.45}$$

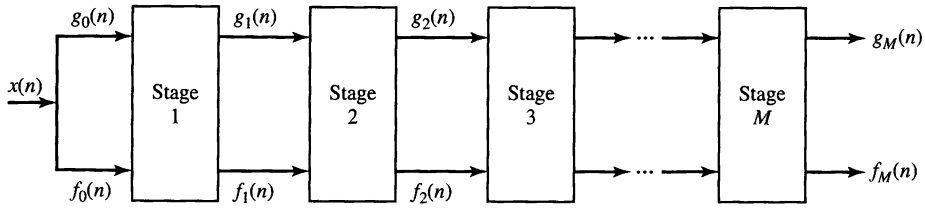
The initial conditions on the order updates are

$$\begin{aligned} f_0(n) &= g_0(n) = x(n) \\ E_0^f(n) &= E_0^b(n) = \sum_{l=0}^{n} w^{n-l}|x(l)|^2 = wE_0^f(n-1) + |x(n)|^2 \end{aligned} \tag{13.4.46}$$

We note that (13.4.46) is also a time-update equation for $E_0^f(n)$ and $E_0^b(n)$.



(a)



(b)

Figure 13.4.1 Least-squares lattice filter.

Time-Update Recursions. Our goal is to determine a time-update equation for $k_m(n)$, which is necessary if the lattice filter is to be adaptive. This derivation will require time-update equations for the prediction coefficients. We begin with the form

$$k_{m+1}(n) = -\mathbf{V}_{m+1}^H(n) \begin{bmatrix} 1 \\ \mathbf{a}_m(n) \end{bmatrix} \tag{13.4.47}$$

The time-update equation for $\mathbf{V}_{m+1}(n)$ is

$$\mathbf{V}_{m+1}(n) = w\mathbf{V}_{m+1}(n-1) + x(n-m-1)\mathbf{X}_{m+1}^*(n) \tag{13.4.48}$$

The time-update equations for the prediction coefficients are determined as follows. From (13.4.6), (13.4.7), and (13.3.14), we have

$$\begin{aligned} \mathbf{a}_m(n) &= -\mathbf{P}_m(n-1)\mathbf{Q}_m(n) \\ &= -\frac{1}{w}\left[\mathbf{P}_m(n-2) - \mathbf{K}_m(n-1)\mathbf{X}_m^t(n-1)\mathbf{P}_m(n-2)\right] \\ &\quad \times \left[wQ_m(n-1) + x(n)X_m^*(n-1)\right] \\ &= \mathbf{a}_m(n-1) - \mathbf{K}_m(n-1)\left[x(n) + \mathbf{a}_m^t(n-1)\mathbf{X}_m(n-1)\right] \end{aligned} \tag{13.4.49}$$

where $\mathbf{K}_m(n-1)$ is the Kalman gain vector at iteration $n-1$. But, from (13.4.38), we have

$$x(n) + \mathbf{a}_m^t(n-1)\mathbf{X}_m(n-1) = f_m(n, n-1) \equiv f_m(n)$$

Therefore, the time-update equation for $\mathbf{a}_m(n)$ is

$$\mathbf{a}_m(n) = \mathbf{a}_m(n-1) - \mathbf{K}_m(n-1)f_m(n) \tag{13.4.50}$$

In a parallel development, using (13.4.15), (13.4.16), and (13.3.14), we obtain the time-update equations for the coefficients of the backward predictor, in the form

$$\mathbf{b}_m(n) = \mathbf{b}_m(n-1) - \mathbf{K}_m(n)g_m(n) \tag{13.4.51}$$

Now, from (13.4.48) and (13.4.50), the time-update equation for $k_{m+1}(n)$ is

$$\begin{aligned} k_{m+1}(n) &= -\left[w\mathbf{V}_{m+1}^H(n-1) + x^*(n-m-1)\mathbf{X}_{m+1}^t(n)\right] \\ &\quad \times \left(\begin{bmatrix} 1 \\ \mathbf{a}_m(n-1) \end{bmatrix} - \begin{bmatrix} 0 \\ \mathbf{K}_m(n-1)f_m(n) \end{bmatrix}\right) \\ &= wk_{m+1}(n-1) - w\mathbf{V}_{m+1}^H(n-1)\begin{bmatrix} 0 \\ \mathbf{K}_m(n-1) \end{bmatrix} f_m(n) \\ &\quad + x^*(n-m-1)\mathbf{X}_{m+1}^t(n)\begin{bmatrix} 1 \\ \mathbf{a}_m(n-1) \end{bmatrix} \\ &\quad - x^*(n-m-1)\mathbf{X}_{m+1}^t(n)\begin{bmatrix} 0 \\ \mathbf{K}_m(n-1) \end{bmatrix} f_m(n) \end{aligned} \tag{13.4.52}$$

But

$$\mathbf{X}_{m+1}^t(n)\begin{bmatrix}1\\ \mathbf{a}_m(n-1)\end{bmatrix} = [x(n)\ \mathbf{X}_m^t(n-1)]\begin{bmatrix}1\\ \mathbf{a}_m(n-1)\end{bmatrix} = f_m(n) \tag{13.4.53}$$

and

$$\begin{aligned}\mathbf{V}_{m+1}^H(n-1)\begin{bmatrix}0\\ \mathbf{K}_m(n-1)\end{bmatrix} &= \mathbf{V}_m^H(n-2)\mathbf{K}_m(n-1)\\ &= \frac{\mathbf{V}_m^H(n-2)\mathbf{P}_m(n-2)\mathbf{X}_m^*(n-1)}{w+\mu_m(n-1)}\\ &= \frac{-\mathbf{b}_m^H(n-2)\mathbf{X}_m^*(n-1)}{w+\mu_m(n-1)}\\ &= -\frac{g_m^*(n-1)-x^*(n-m-1)}{w+\mu_m(n-1)}\end{aligned} \tag{13.4.54}$$

where $\mu_m(n)$ is as previously defined in (13.3.13). Finally,

$$\mathbf{X}_{m+1}^t(n)\begin{bmatrix}0\\ \mathbf{K}_m(n-1)\end{bmatrix} = \frac{\mathbf{X}_m^t(n-1)\mathbf{P}_m(n-2)\mathbf{X}_m^*(n-1)}{w+\mu_m(n-1)} = \frac{\mu_m(n-1)}{w+\mu_m(n-1)} \tag{13.4.55}$$

Substituting the results of (13.4.53), (13.4.54), and (13.4.55) into (13.4.52) we obtain the desired time-update equation in the form

$$k_{m+1}(n) = wk_{m+1}(n-1) + \frac{w}{w+\mu_m(n-1)}f_m(n)g_m^*(n-1) \tag{13.4.56}$$

It is convenient to define a new variable

$$\alpha_m(n) = \frac{w}{w+\mu_m(n)} \tag{13.4.57}$$

Clearly, $\alpha_m(n)$ is real-valued and has a range $0 < \alpha_m(n) < 1$. Then, the time-update equation (13.4.56) becomes

$$k_{m+1}(n) = wk_{m+1}(n-1) + \alpha_m(n-1)f_m(n)g_m^*(n-1) \tag{13.4.58}$$

Order Update for $\alpha_m(n)$. Although $\alpha_m(n)$ can be computed directly for each value of m and for each n, it is more efficient to use an order-update equation, which is determined as follows. First, from the definition of $\mathbf{K}_m(n)$ given in (13.3.12), it is easily seen that

$$\alpha_m(n) = 1 - \mathbf{X}_m^t(n)\mathbf{K}_m(n) \tag{13.4.59}$$

To obtain an order-update equation for $\alpha_m(n)$, we need an order-update equation for the Kalman gain vector $\mathbf{K}_m(n)$. But $\mathbf{K}_{m+1}(n)$ may be expressed as

$$\begin{aligned}\mathbf{K}_{m+1}(n) &= \mathbf{P}_{m+1}(n)\mathbf{X}^*_{m+1}(n)\\ &= \left(\begin{bmatrix}\mathbf{P}_m(n) & 0\\ 0 & 0\end{bmatrix} + \frac{1}{E^b_m(n)}\begin{bmatrix}\mathbf{b}_m(n)\\ 1\end{bmatrix}\begin{bmatrix}\mathbf{b}^H_m(n) & 1\end{bmatrix}\right)\begin{bmatrix}\mathbf{X}^*_m(n)\\ x^*(n-m)\end{bmatrix}\\ &= \begin{bmatrix}\mathbf{K}_m(n)\\ 0\end{bmatrix} + \frac{g^*_m(n,n)}{E^b_m(n)}\begin{bmatrix}\mathbf{b}_m(n)\\ 1\end{bmatrix}\end{aligned} \tag{13.4.60}$$

The term $g_m(n, n)$ may also be expressed as

$$\begin{aligned}g_m(n,n) &= x(n-m) + \mathbf{b}^t_m(n)\mathbf{X}_m(n)\\ &= x(n-m) + \left[\mathbf{b}^t_m(n-1) - \mathbf{K}^t_m(n)g_m(n)\right]\mathbf{X}_m(n)\\ &= x(n-m) + \mathbf{b}^t_m(n-1)\mathbf{X}_m(n) - g_m(n)\mathbf{K}^t_m(n)\mathbf{X}_m(n)\\ &= g_m(n)\left[1 - \mathbf{K}^t_m(n)\mathbf{X}_m(n)\right]\\ &= \alpha_m(n)g_m(n)\end{aligned} \tag{13.4.61}$$

Hence, the order-update equation for $\mathbf{K}_m(n)$ in (13.4.60) may also be written as

$$\mathbf{K}_{m+1}(n) = \begin{bmatrix}\mathbf{K}_m(n)\\ 0\end{bmatrix} + \frac{\alpha_m(n)g^*_m(n)}{E^b_m(n)}\begin{bmatrix}\mathbf{b}_m(n)\\ 1\end{bmatrix} \tag{13.4.62}$$

By using (13.4.62) and the relation in (13.4.59), we obtain the order-update equation for $\alpha_m(n)$ as follows:

$$\begin{aligned}\alpha_{m+1}(n) &= 1 - \mathbf{X}^t_{m+1}(n)\mathbf{K}_{m+1}(n) = 1 - \left[\mathbf{X}^t_m(n)x(n-m)\right]\\ &\quad\times\left(\begin{bmatrix}\mathbf{K}_m(n)\\ 0\end{bmatrix} + \frac{\alpha_m(n)g^*_m(n)}{E^b_m(n)}\begin{bmatrix}\mathbf{b}_m(n)\\ 1\end{bmatrix}\right)\\ &= \alpha_m(n) - \frac{\alpha_m(n)g^*_m(n)}{E^b_m(n)}\left[\mathbf{X}^t_m(n)\ x(n-m)\right]\begin{bmatrix}\mathbf{b}_m(n)\\ 1\end{bmatrix}\\ &= \alpha_m(n) - \frac{\alpha_m(n)g^*_m(n)}{E^b_m(n)}g_m(n,\ n)\\ &= \alpha_m(n) - \frac{\alpha^2_m(n)|g_m(n)|^2}{E^b_m(n)}\end{aligned} \tag{13.4.63}$$

Thus, we have obtained both the order-update and time-update equations for the basic least-squares lattice shown in Figure 13.4.1. The basic equations are (13.4.41) and (13.4.44) for the forward and backward errors, usually called the *residuals,* (13.4.36) and (13.4.37) for the corresponding least-squares errors, the time-update

Equation (13.4.58) for $k_m(n)$, and the order-update Equation (13.4.63) for the parameter $\alpha_m(n)$. Initially, we have

$$\begin{aligned} E_m^f(-1) &= E_m^b(-1) = E_m^b(-2) = \epsilon > 0 \\ f_m(-1) &= g_m(-1) = k_m(-1) = 0 \\ \alpha_m(-1) &= 1, \quad \alpha_{-1}(n) = \alpha_{-1}(n-1) = 1 \end{aligned} \tag{13.4.64}$$

Joint Process Estimation. The last step in the derivation is to obtain the least-squares estimate of the desired signal $d(n)$ from the lattice. Suppose that the adaptive filter has $m+1$ coefficients, which are determined to minimize the average weighted squared error

$$\mathscr{E}_{m+1} = \sum_{l=0}^{n} w^{n-l} |e_{m+1}(l, n)|^2 \tag{13.4.65}$$

where

$$e_{m+1}(l, n) = d(l) - \mathbf{h}_{m+1}^t(n)\mathbf{X}_{m+1}(l) \tag{13.4.66}$$

The linear estimate

$$\hat{d}(l, n) = \mathbf{h}_{m+1}^t(n)\mathbf{X}_{m+1}(l) \tag{13.4.67}$$

which will be obtained from the lattice by using the residuals $g_m(n)$, is called the *joint process estimate.*

From the results of Section 13.3.1, we have already established that the coefficients of the adaptive filter that minimize (13.4.65) are given by the equation

$$\mathbf{h}_{m+1}(n) = \mathbf{P}_{m+1}(n)\mathbf{D}_{m+1}(n) \tag{13.4.68}$$

We have also established that $\mathbf{h}_m(n)$ satisfies the time-update equation given in (13.3.27).

Now, let us obtain an order-update equation for $\mathbf{h}_m(n)$. From (13.4.68) and (13.4.27), we have

$$\mathbf{h}_{m+1}(n) = \begin{bmatrix} \mathbf{P}_m(n) & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{D}_m(n) \\ \cdots \end{bmatrix} + \frac{1}{E_m^b(n)} \begin{bmatrix} \mathbf{b}_m(n) \\ 1 \end{bmatrix} \begin{bmatrix} \mathbf{b}_m^H(n) & 1 \end{bmatrix} \mathbf{D}_{m+1}(n) \tag{13.4.69}$$

We define a complex-valued scalar quantity $\delta_m(n)$ as

$$\delta_m(n) = \begin{bmatrix} \mathbf{b}_m^H(n) & 1 \end{bmatrix} \mathbf{D}_{m+1}(n) \tag{13.4.70}$$

Then, (13.4.69) may be expressed as

$$\mathbf{h}_{m+1}(n) = \begin{bmatrix} \mathbf{h}_m(n) \\ 0 \end{bmatrix} + \frac{\delta_m(n)}{E_m^b(n)} \begin{bmatrix} \mathbf{b}_m(n) \\ 1 \end{bmatrix} \tag{13.4.71}$$

The scalar $\delta_m(n)$ satisfies a time-update equation that is obtained from the time-update equations for $\mathbf{b}_m(n)$ and $\mathbf{D}_m(n)$, given by (13.4.51) and (13.3.17), respectively. Thus,

$$\begin{aligned}\delta_m(n) &= \begin{bmatrix}\mathbf{b}_m^H(n-1) - \mathbf{K}_m^H(n)g_m^*(n) & 1\end{bmatrix}\begin{bmatrix}w\mathbf{D}_{m+1}(n-1) + d(n)\mathbf{X}_{m+1}^*(n)\end{bmatrix} \\ &= w\delta_m(n-1) + \begin{bmatrix}\mathbf{b}_m^H(n-1) & 1\end{bmatrix}\mathbf{X}_{m+1}^*(n)d(n) \\ &\quad - wg_m^*(n)\begin{bmatrix}\mathbf{K}_m^H(n) & 0\end{bmatrix}\mathbf{D}_{m+1}(n-1) - g_m^*(n)d(n)\begin{bmatrix}\mathbf{K}_m^H(n) & 0\end{bmatrix}\mathbf{X}_{m+1}^*(n)\end{aligned} \tag{13.4.72}$$

But

$$\begin{bmatrix}\mathbf{b}_m^H(n-1) & 1\end{bmatrix}\mathbf{X}_{m+1}^*(n) = x^*(n-m) + \mathbf{b}_m^H(n-1)\mathbf{X}_m^*(n) = g_m^*(n) \tag{13.4.73}$$

Also,

$$\begin{aligned}\begin{bmatrix}\mathbf{K}_m^H(n) & 0\end{bmatrix}\mathbf{D}_{m+1}(n-1) &= \frac{1}{w+\mu_m(n)}\begin{bmatrix}\mathbf{X}_m^t(n)\mathbf{P}_m(n-1) & 0\end{bmatrix}\begin{bmatrix}\mathbf{D}_m(n-1) \\ \cdots\end{bmatrix} \\ &= \frac{1}{w+\mu_m(n)}\mathbf{X}_m^t(n)\mathbf{h}_m(n-1)\end{aligned} \tag{13.4.74}$$

The last term in (13.4.72) may be expressed as

$$\begin{aligned}\begin{bmatrix}\mathbf{K}_m^H(n) & 0\end{bmatrix}\begin{bmatrix}\mathbf{X}_m^*(n) \\ \cdots\end{bmatrix} &= \frac{1}{w+\mu_m(n)}\mathbf{X}_m^t(n)\mathbf{P}_m(n-1)\mathbf{X}_m^*(n) \\ &= \frac{\mu_m(n)}{w+\mu_m(n)}\end{aligned} \tag{13.4.75}$$

Upon substituting the results in (13.4.73–13.4.75) into (13.4.72), we obtain the desired time-update equation for $\delta_m(n)$ as

$$\delta_m(n) = w\delta_m(n-1) + \alpha_m(n)g_m^*(n)e_m(n) \tag{13.4.76}$$

Order-update equations for $\alpha_m(n)$ and $g_m(n)$ have already been derived. With $e_0(n) = d(n)$, the order-update equation for $e_m(n)$ is obtained as follows:

$$\begin{aligned}e_m(n) = e_m(n, n-1) &= d(n) - \mathbf{h}_m^t(n-1)\mathbf{X}_m(n) \\ &= d(n) - \begin{bmatrix}\mathbf{h}_{m-1}^t(n-1) & 0\end{bmatrix}\begin{bmatrix}\mathbf{X}_{m-1}(n) \\ \cdots\end{bmatrix} \\ &\quad - \frac{\delta_{m-1}(n-1)}{E_{m-1}^b(n-1)}\begin{bmatrix}\mathbf{b}_{m-1}^t(n-1) & 1\end{bmatrix}\mathbf{X}_m(n) \\ &= e_{m-1}(n) - \frac{\delta_{m-1}(n-1)g_{m-1}(n)}{E_{m-1}^b(n-1)}\end{aligned} \tag{13.4.77}$$

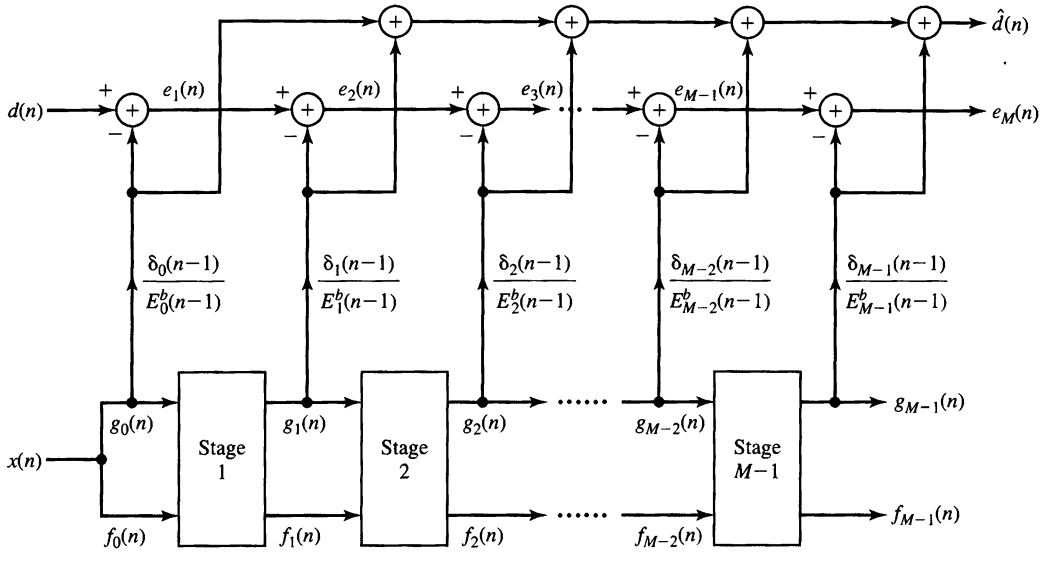


Figure 13.4.2 Adaptive RLS lattice-ladder filter.

Finally, the output estimate $d(n)$ of the least-squares lattice is

$$\hat{d}(n) = \mathbf{h}_{m+1}^t(n-1)\mathbf{X}_{m+1}(n) \tag{13.4.78}$$

But $\mathbf{h}_{m+1}^t(n-1)$ is not computed explicitly. By repeated substitution of the order-update equation for $\mathbf{h}_{m+1}(n)$ given by (13.4.71) into (13.4.78), we obtain the desired expression for $\hat{d}(n)$ in the form

$$\hat{d}(n) = \sum_{k=0}^{M-1} \frac{\delta_k(n-1)}{E_k^b(n-1)} g_k(n) \tag{13.4.79}$$

In other words, the output estimate $\hat{d}(n)$ is a linear weighted sum of the backward residuals $g_k(n)$.

The adaptive least-squares lattice/joint-process (ladder) estimator is illustrated in Figure 13.4.2. This lattice-ladder structure is mathematically equivalent to the RLS direct-form FIR filter. The recursive equations are sumarized in Table 13.5. This is called the *a priori form* of the *RLS lattice-ladder algorithm* in order to distinguish it from another form of the algorithm, called the *a posteriori form,* in which the coefficient vector $\mathbf{h}_M(n)$ is used in place of $\mathbf{h}_M(n-1)$ to compute the estimate $d(n)$. In many adaptive filtering problems, such as channel equalization and echo cancellation, the a posteriori form cannot be used, because $\mathbf{h}_M(n)$ cannot be computed prior to the computation of $d(n)$.

We now describe a number of modifications that can be made to the "conventional" RLS lattice-ladder algorithm given in Table 13.5.

TABLE 13.5 A Priori Form of the RLS Lattice-Ladder Algorithm

Lattice predictor: Begin with $n = 1$ and compute the order updates for $m = 0, 1, \ldots, M-2$

$$k_{m+1}(n-1) = wk_{m+1}(n-2) + \alpha_m(n-2)f_m(n-1)g_m^*(n-2)$$

$$\mathcal{K}_{m+1}^f(n-1) = -\frac{k_{m+1}(n-1)}{E_m^b(n-2)}$$

$$\mathcal{K}_{m+1}^b(n-1) = -\frac{k_{m+1}^*(n-1)}{E_m^f(n-1)}$$

$$f_{m+1}(n) = f_m(n) + \mathcal{K}_{m+1}^f(n-1)g_m(n-1)$$

$$g_{m+1}(n) = g_m(n-1) + \mathcal{K}_{m+1}^b(n-1)f_m(n)$$

$$E_{m+1}^f(n-1) = E_m^f(n-1) - \frac{|k_{m+1}(n-1)|^2}{E_m^b(n-2)}$$

$$E_{m+1}^b(n-1) = E_m^b(n-2) - \frac{|k_{m+1}(n-1)|^2}{E_m^f(n-1)}$$

$$\alpha_{m+1}(n-1) = \alpha_m(n-1) - \frac{\alpha_m^2(n-1)|g_m(n-1)|^2}{E_m^b(n-1)}$$

Ladder filter: Begin with $n = 1$ and compute the order updates for $m = 0, 1, \ldots, M-1$

$$\delta_m(n-1) = w\delta_m(n-2) + \alpha_m(n-1)g_m^*(n-1)e_m(n-1)$$

$$\xi_m(n-1) = -\frac{\delta_m(n-1)}{E_m^b(n-1)}$$

$$e_{m+1}(n) = e_m(n) + \xi_m(n-1)g_m(n)$$

Initialization

$$\alpha_0(n-1) = 1, \quad e_0(n) = d(n), \quad f_0(n) = g_0(n) = x(n)$$

$$E_0^f(n) = E_0^b(n) = wE_0^f(n-1) + |x(n)|^2$$

$$\alpha_m(-1) = 1, \quad k_m(-1) = 0$$

$$E_m^b(-1) = E_m^f(0) = \epsilon > 0; \quad \delta_m(-1) = 0$$

Modified RLS Lattice Algorithms. The recursive equations in the RLS lattice algorithm given in Table 13.5 are by no means unique. Modifications can be made to some of the equations without affecting the optimality of the algorithm. However, some modifications result in algorithms that are more numerically robust when fixed-point arithmetic is used in the implementation of the algorithms. We give a number of basic relationships that are easily established from the above developments.

First, we have a relationship between the a priori and a posteriori error residuals.

A priori errors:

$$\begin{aligned} f_m(n, n-1) &\equiv f_m(n) = x(n) + \mathbf{a}_m^t(n-1)\mathbf{X}_m(n-1) \\ g_m(n, n-1) &\equiv g_m(n) = x(n-m) + \mathbf{b}_m^t(n-1)\mathbf{X}_m(n) \end{aligned} \tag{13.4.80}$$

A posteriori errors:

$$\begin{aligned} f_m(n,n) &= x(n) + \mathbf{a}_m^t(n)\mathbf{X}_m(n-1) \\ g_m(n,n) &= x(n-m) + \mathbf{b}_m^t(n)\mathbf{X}_m(n) \end{aligned} \tag{13.4.81}$$

The basic relations between (13.4.80) and (13.4.81) are

$$\begin{aligned} f_m(n,n) &= \alpha_m(n-1) f_m(n) \\ g_m(n,n) &= \alpha_m(n) g_m(n) \end{aligned} \tag{13.4.82}$$

These relations follow easily by using (13.4.50) and (13.4.51) in (13.4.81).

Second, we may obtain time-update equations for the least-squares forward and backward errors. For example, from (13.4.8) and (13.4.50) we obtain

$$\begin{aligned} E_m^f(n) &= q(n) + \mathbf{a}_m^t(n)\mathbf{Q}_m^*(n) \\ &= q(n) + \left[\mathbf{a}_m^t(n-1) - \mathbf{K}_m^t(n-1) f_m(n)\right]\left[w\mathbf{Q}_m^*(n-1) + x^*(n)\mathbf{X}_m(n-1)\right] \\ &= wE_m^f(n-1) + \alpha_m(n-1)|f_m(n)|^2 \end{aligned} \tag{13.4.83}$$

Similarly, from (13.4.17) and (13.4.51) we obtain

$$E_m^b(n) = wE_m^b(n-1) + \alpha_m(n)|g_m(n)|^2 \tag{13.4.84}$$

Usually, (13.4.83) and (13.4.84) are used in place of the sixth and seventh equations in Table 13.5.

Third, we obtain a time-update equation for the Kalman gain vector, which is not explicitly used in the lattice algorithm, but which is used in the fast FIR filter algorithms. For this derivation, we also use the time-update equations for the forward and backward prediction coefficients given by (13.4.50) and (13.4.51). Thus, we have

$$\begin{aligned} \mathbf{K}_m(n) &= \mathbf{P}_m(n)\mathbf{X}_m^*(n) \\ &= \begin{bmatrix} 0 & 0 \\ 0 & \mathbf{P}_{m-1}(n-1) \end{bmatrix}\begin{bmatrix} x^*(n) \\ \mathbf{X}_{m-1}^*(n-1) \end{bmatrix} \\ &\quad + \frac{1}{E_{m-1}^f(n)}\begin{bmatrix} 1 \\ \mathbf{a}_{m-1}(n) \end{bmatrix}\begin{bmatrix} 1 & \mathbf{a}_{m-1}^H(n) \end{bmatrix}\begin{bmatrix} x^*(n) \\ \mathbf{X}_{m-1}^*(n-1) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \mathbf{K}_{m-1}(n-1) \end{bmatrix} + \frac{f_{m-1}^*(n,n)}{E_{m-1}^f(n)}\begin{bmatrix} 1 \\ \mathbf{a}_{m-1}(n) \end{bmatrix} \\ &\equiv \begin{bmatrix} \mathbf{C}_{m-1}(n) \\ c_{mm}(n) \end{bmatrix} \end{aligned} \tag{13.4.85}$$

where, by definition, $\mathbf{C}_{m-1}(n)$ consists of the first $(m-1)$ elements of $\mathbf{K}_m(n)$ and $c_{\text{mm}}(n)$ is the last element. From (13.4.60), we also have the order-update equation for $\mathbf{K}_m(n)$ as

$$\mathbf{K}_m(n) = \begin{bmatrix} \mathbf{K}_{m-1}(n) \\ 0 \end{bmatrix} + \frac{g^*_{m-1}(n,n)}{E^b_{m-1}(n)} \begin{bmatrix} \mathbf{b}_{m-1}(n) \\ 1 \end{bmatrix} \tag{13.4.86}$$

By equating (13.4.85) to (13.4.86), we obtain the result

$$c_{\text{mm}}(n) = \frac{g^*_{m-1}(n,n)}{E^b_{m-1}(n)} \tag{13.4.87}$$

and, hence,

$$\mathbf{K}_{m-1}(n) + c_{\text{mm}}(n)\mathbf{b}_{m-1}(n) = \mathbf{C}_{m-1}(n) \tag{13.4.88}$$

By substituting for $\mathbf{b}_{m-1}(n)$ from (13.4.51) into (13.4.88), we obtain the time-update equation for the Kalman gain vector in (13.4.85) as

$$\mathbf{K}_{m-1}(n) = \frac{\mathbf{C}_{m-1}(n) - c_{\text{mm}}(n)\mathbf{b}_{m-1}(n-1)}{1 - c_{\text{mm}}(n)g_{m-1}(n)} \tag{13.4.89}$$

There is also a time-update equation for the scalar $\alpha_m(n)$. From (13.4.63), we have

$$\begin{aligned} \alpha_m(n) &= \alpha_{m-1}(n) - \frac{\alpha^2_{m-1}(n)|g_{m-1}(n)|^2}{E^b_{m-1}(n)} \\ &= \alpha_{m-1}(n)[1 - c_{\text{mm}}(n)g_{m-1}(n)] \end{aligned} \tag{13.4.90}$$

A second relation is obtained by using (13.4.85) to eliminate $\mathbf{K}_{m-1}(n)$ in the expression for $\alpha_m(n)$. Then,

$$\begin{aligned} \alpha_m(n) &= 1 - \mathbf{X}^t_m(n)\mathbf{K}_m(n) \\ &= \alpha_{m-1}(n-1)\left[1 - \frac{f^*_{m-1}(n,n)f_{m-1}(n)}{E^f_{m-1}(n)}\right] \end{aligned} \tag{13.4.91}$$

By equating (13.4.90) to (13.4.91), we obtain the desired time-update equation for $\alpha_m(n)$ as

$$\alpha_{m-1}(n) = \alpha_{m-1}(n-1)\left[\frac{1 - \frac{f^*_{m-1}(n,n)f_{m-1}(n)}{E^f_{m-1}(n)}}{1 - c_{\text{mm}}(n)g_{m-1}(n)}\right] \tag{13.4.92}$$

Finally, we wish to distinguish between two different methods for updating the reflection coefficients in the lattice filter and the ladder part: the *conventional (indirect) method* and the *direct method.* In the conventional (indirect) method,

$$\mathcal{K}_{m+1}^{f}(n) = -\frac{k_{m+1}(n)}{E_m^b(n-1)} \tag{13.4.93}$$

$$\mathcal{K}_{m+1}^{b}(n) = -\frac{k_{m+1}^*(n)}{E_m^f(n)} \tag{13.4.94}$$

$$\xi_m(n) = -\frac{\delta_m(n)}{E_m^b(n)} \tag{13.4.95}$$

where $k_{m+1}(n)$ is time-updated from (13.4.58), $\delta_m(n)$ is updated according to (13.4.76), and $E_m^f(n)$ and $E_m^b(n)$ are updated according to (13.4.83) and (13.4.84). By substituting for $k_{m+1}(n)$ from (13.4.58) into (13.4.93), and using (13.4.84) and the eighth equation in Table 13.5, we obtain

$$\begin{aligned}
\mathcal{K}_{m+1}^{f}(n) &= -\frac{k_{m+1}(n-1)}{E_m^b(n-2)}\left(\frac{wE_m^b(n-2)}{E_m^b(n-1)}\right) - \frac{\alpha_m(n-1)f_m(n)g_m^*(n-1)}{E_m^b(n-1)} \\
&= \mathcal{K}_{m+1}^{f}(n-1)\left(1 - \frac{\alpha_m(n-1)|g_m(n-1)|^2}{E_m^b(n-1)}\right) \\
&\quad - \frac{\alpha_m(n-1)f_m(n)g_m^*(n-1)}{E_m^b(n-1)} \\
&= \mathcal{K}_{m+1}^{f}(n-1) - \frac{\alpha_m(n-1)f_{m+1}(n)g_m^*(n-1)}{E_m^b(n-1)}
\end{aligned} \tag{13.4.96}$$

which is a formula for directly updating the reflection coefficients in the lattice. Similarly, by substituting (13.4.58) into (13.4.94), and using (13.4.83) and the eighth equation in Table 13.5, we obtain

$$\mathcal{K}_{m+1}^{b}(n) = \mathcal{K}_{m+1}^{b}(n-1) - \frac{\alpha_m(n-1)f_m^*(n)g_{m+1}(n)}{E_m^f(n)} \tag{13.4.97}$$

Finally, the ladder gain can also be updated directly according to the relation

$$\xi_m(n) = \xi_m(n-1) - \frac{\alpha_m(n)g_m^*(n)e_{m+1}(n)}{E_m^b(n)} \tag{13.4.98}$$

The RLS lattice-ladder algorithm that uses the direct update relations in (13.4.96–13.4.98) and (13.4.83–13.4.84) is listed in Table 13.6.

An important characteristic of the algorithm in Table 13.6 is that the forward and backward residuals are fed back to time-update the reflection coefficients in the lattice stage, and $e_{m+1}(n)$ is fed back to update the ladder gain $\xi_m(n)$. For this reason, this RLS lattice-ladder algorithm has been called the *error-feedback form.* A similar form can be obtained for the a posteriori RLS lattice-ladder algorithm. For more details on the error-feedback form of RLS lattice-ladder algorithms, the interested reader is referred to Ling, Manolakis, and Proakis (1986).

TABLE 13.6 Direct Update (Error-Feedback) Form of the A Priori RLS Lattice-Ladder Algorithm

Lattice predictor: Begin with $n = 1$ and compute the order updates for $m = 0, 1, \ldots, M-2$

$$\mathcal{K}^f_{m+1}(n-1) = \mathcal{K}^f_{m+1}(n-2) - \frac{\alpha_m(n-2) f_{m+1}(n-1) g^*_m(n-2)}{E^b_m(n-2)}$$

$$\mathcal{K}^b_{m+1}(n-1) = \mathcal{K}^b_{m+1}(n-2) - \frac{\alpha_m(n-2) f^*_m(n-1) g_{m+1}(n-1)}{E^f_m(n-1)}$$

$$f_{m+1}(n) = f_m(n) + \mathcal{K}^f_{m+1}(n-1) g_m(n-1)$$

$$g_{m+1}(n) = g_m(n-1) + \mathcal{K}^b_{m+1}(n-1) f_m(n)$$

$$E^f_{m+1}(n-1) = w E^f_{m+1}(n-2) + \alpha_{m+1}(n-2) |f_{m+1}(n-1)|^2$$

$$\alpha_{m+1}(n-1) = \alpha_m(n-1) - \frac{\alpha^2_m(n-1) |g_m(n-1)|^2}{E^b_m(n-1)}$$

$$E^b_{m+1}(n-1) = w E^b_{m+1}(n-2) + \alpha_{m+1}(n-1) |g_{m+1}(n-1)|^2$$

Ladder filter: Begin with $n = 1$ and compute the order updates $m = 0, 1, \ldots, M-1$

$$\xi_m(n-1) = \xi_m(n-2) - \frac{\alpha_m(n-1) g^*_m(n-1) e_{m+1}(n-1)}{E^b_m(n-1)}$$

$$e_{m+1}(n) = e_m(n) + \xi_m(n-1) g_m(n)$$

Initialization

$$\alpha_0(n-1) = 1, \quad e_0(n) = d(n), \quad f_0(n) = g_0(n) = x(n)$$

$$E^f_0(n) = E^b_0(n) = w E^f_0(n-1) + |x(n)|^2$$

$$\alpha_m(-1) = 1, \quad \mathcal{K}^f_m(-1) = \mathcal{K}^b_m(-1) = 0$$

$$E^b_m(-1) = E^f_m(0) = \epsilon > 0$$

Fast RLS Algorithms. The two versions of the fast RLS algorithms given in Section 13.3.3 follow directly from the relationships that we have obtained in this section. In particular, we fix the size of the lattice and the associated forward and backward predictors at $M-1$ stages. Thus, we obtain the first seven recursive equations in the two versions of the algorithm. The remaining problem is to determine the time-update equation for the Kalman gain vector, which was determined in (13.4.85–13.4.89). In version B of the algorithm, given in Table 13.3, we used the scalar $\alpha_m(n)$ to reduce the computations from $10M$ to $9M$. Version A of the algorithm, given in Table 13.2, avoids the use of this parameter. Since these algorithms provide a direct updating of the Kalman gain vector, they have been called *fast Kalman algorithms* (for reference, see Falconer and Ljung (1978) and Proakis (1989)).

Further reduction of computational complexity to $7M$ is possible by directly updating the following *alternative (Kalman) gain vector* (see Carayannis, Manolakis, and Kalouptsidis (1983)) defined as

$$\tilde{\mathbf{K}}_M(n) = \frac{1}{w} \mathbf{P}_M(n-1) \mathbf{X}^*_M(n) \tag{13.4.99}$$

Several fast algorithms using this gain vector have been proposed, with complexities ranging from $7M$ to $10M$. Table 13.7 lists the FAEST (Fast A Posteriori Error Sequential Technique) algorithm with a computational complexity $7M$ (for a derivation, see Carayannis, Manolakis, and Kalouptsidis (1983; 1986) and Problem 13.7).

In general, the $7M$ fast RLS algorithms and some variations are very sensitive to round-off noise and exhibit instability problems (Falconer and Ljung (1978), Carayannis, Manolakis, and Kalouptsidis (1983; 1986), and Cioffi and Kailath (1984)). The instability problem in the $7M$ algorithms has been addressed by Slock and Kailath (1988; 1991), and modifications have been proposed that stabilize these algorithms. The resulting stabilized algorithms have a computational complexity ranging from $8M$ to $9M$. Thus, their computational complexity is increased by a relatively small amount compared to the unstable $7M$ algorithms.

To understand the stabilized fast RLS algorithms, we begin by comparing the fast RLS algorithm given in Table 13.3 and the FAEST algorithm in Table 13.7. As

TABLE 13.7 FAEST Algorithm

$$
\begin{aligned}
f_{M-1}(n) &= x(n) + \mathbf{a}_{M-1}^t(n-1)\mathbf{X}_{M-1}(n-1) \\
\bar{f}_{M-1}(n,n) &= \frac{f_{M-1}(n)}{\bar{\alpha}_{M-1}(n-1)} \\
\mathbf{a}_{M-1}(n) &= \mathbf{a}_{M-1}(n-1) - \bar{\mathbf{K}}_{M-1}(n-1)\bar{f}_{M-1}(n,n) \\
E_{M-1}^f(n) &= wE_{M-1}^f(n-1) + f_{M-1}(n)f_{M-1}^*(n,n) \\
\bar{\mathbf{K}}_M(n) \equiv \begin{bmatrix} \bar{\mathbf{C}}_{M-1}(n) \\ \bar{c}_{\mathrm{MM}}(n) \end{bmatrix} &= \begin{bmatrix} 0 \\ \bar{\mathbf{K}}_{M-1}(n-1) \end{bmatrix} + \frac{f_{M-1}^*(n)}{wE_{M-1}^f(n-1)} \begin{bmatrix} 1 \\ \mathbf{a}_{M-1}(n-1) \end{bmatrix} \\
g_{M-1}(n) &= -wE_{M-1}^b(n-1)\bar{c}_{\mathrm{MM}}^*(n) \\
\bar{\mathbf{K}}_{M-1}(n) &= \bar{\mathbf{C}}_{M-1}(n) - \mathbf{b}_{M-1}(n-1)\bar{c}_{\mathrm{MM}}(n) \\
\bar{\alpha}_M(n) &= \bar{\alpha}_{M-1}(n-1) + \frac{|f_{M-1}(n)|^2}{wE_{M-1}^f(n-1)} \\
\bar{\alpha}_{M-1}(n) &= \bar{\alpha}_M(n) + g_{M-1}(n)\bar{c}_{\mathrm{MM}}(n) \\
\bar{g}_{M-1}(n,n) &= \frac{g_{M-1}(n)}{\bar{\alpha}_{M-1}(n)} \\
E_{M-1}^b(n) &= wE_{M-1}^b(n-1) + g_{M-1}(n)\bar{g}_{M-1}^*(n,n) \\
\mathbf{b}_{M-1}(n) &= \mathbf{b}_{M-1}(n-1) + \bar{\mathbf{K}}_{M-1}(n)\bar{g}_{M-1}(n,n) \\
e_M(n) &= d(n) - \mathbf{h}_M^t(n-1)\mathbf{X}_M(n) \\
\bar{e}_M(n,n) &= \frac{e_M(n)}{\bar{\alpha}_M(n)} \\
\mathbf{h}_M(n) &= \mathbf{h}_M(n-1) + \bar{\mathbf{K}}_M(n)\bar{e}_M(n,n)
\end{aligned}
$$

Initialization: Set all vectors to zero

$$
\begin{aligned}
E_{M-1}^f(-1) &= E_{M-1}^b(-1) = \epsilon > 0 \\
\bar{\alpha}_{M-1}(-1) &= 1
\end{aligned}
$$

indicated, there are two major differences between these two algorithms. First, the FAEST algorithm uses the alternative (Kalman) gain vector instead of the Kalman gain vector. Second, the fast RLS algorithm computes the a priori backward prediction error $g_{M-1}(n)$ through FIR filtering using the backward prediction coefficient vector $\mathbf{b}_{m-1}(n-1)$, whereas the FAEST algorithm computes the same quantity through a scalar operation by noticing that the last element of the alternative gain vector, $\tilde{c}_{\text{MM}}(n)$, is equal to $-wE^b_{M-1}g_{M-1}(n)$. Since these two algorithms are algebraically equivalent, the backward prediction errors calculated in different ways should be identical if infinite precision is used in the computation. Practically, when finite-precision arithmetic is used, the backward prediction errors computed using different formulas are only approximately equal. In what follows, we denote them by $g^{(f)}_{M-1}$ and $g^{(s)}_{M-1}(n)$, respectively. The superscripts (f) and (s) indicate that they are computed using the filtering approach and scalar operation, respectively.

There are other quantities in the algorithms that can also be computed in different ways. In particular, the parameter $\alpha_{M-1}(n)$ can be computed from the vector quantities $\tilde{\mathbf{K}}_{M-1}(n)$ and $\mathbf{X}_{M-1}(n)$ as

$$\alpha_{M-1}(n) = 1 + \tilde{\mathbf{K}}^t_{M-1}(n)\mathbf{X}_{M-1}(n) \tag{13.4.100}$$

or from scalar quantities. We denote these values as $\tilde{\alpha}^{(f)}_{M-1}(n)$ and $\tilde{\alpha}^{(s)}_{M-1}(n)$, respectively. Finally, the last element of $\tilde{\mathbf{K}}_M(n)$, denoted as $\tilde{c}^{(f)}_{\text{MM}}(n)$, may be computed from the relation

$$\tilde{c}^{(f)}_{\text{MM}}(n) = \frac{-g^{(f)}_{M-1}(n)}{wE^b_{M-1}(n-1)} \tag{13.4.101}$$

The two quantities in each of the three pairs $[g^{(f)}_{M-1}(n),\ g^{(s)}_{M-1}(n)]$, $[\alpha^{(f)}_{M-1}(n),\ \alpha^{(s)}_{M-1}(n)]$, and $[\tilde{c}^{(f)}_{\text{MM}}(n),\ \tilde{c}^{(s)}_{\text{MM}}(n)]$ are algebraically equivalent. Hence, either of the two quantities or their linear combination (of the form $k\beta^{(s)} + (1-k)\beta^{(f)}$, where β represents any of the three parameters) are algebraically equivalent to the original quantities, and may be used in the algorithm. Slock and Kailath (1988; 1991) found that by using the appropriate quantity or its linear combination in the fast RLS algorithm, it was sufficient to correct for the positive feedback inherent in the fast RLS algorithms. Implementation of this basic notion leads to the stabilized fast RLS algorithm given in Table 13.8.

We observe from Table 13.8 that the stabilized fast RLS algorithm employs constants $k_i, i = 1, 2, \ldots, 5$, to form five linear combinations of the three pairs of quantities just described. The best values of the k_i found by Slock and Kailath resulted from computer search, and are given as $k_1 = 1.5, k_2 = 2.5, k_3 = 1, k_4 = 0, k_5 = 1$. When $k_i = 0$ or 1, we use only one of the quantities in the linear combination. Hence, some of the parameters in the three pairs need not be computed. It was also found that the stability of the algorithm is only slightly affected if $\alpha^{(f)}_{M-1}(n)$ is not used. These simplifications result in the algorithm given in Table 13.9, which has a complexity of $8M$ and is numerically stable.

The performance of the stabilized fast RLS algorithms depends highly on proper initialization. On the other hand, an algorithm that uses $g^{(f)}_{M-1}(n)$ in its computations

TABLE 13.8 The Stabilized Fast RLS Algorithm

$$f_{M-1}(n) = x(n) + \mathbf{a}_{M-1}^t(n-1)\mathbf{X}_{M-1}(n-1)$$

$$f_{M-1}(n,n) = \frac{f_{M-1}(n)}{\bar{\alpha}_{M-1}(n-1)}$$

$$\mathbf{a}_{M-1}(n) = \mathbf{a}_{M-1}(n-1) - \bar{\mathbf{K}}_{M-1}(n-1)f_{M-1}(n,n)$$

$$\bar{c}_{M1}(n) = \frac{f_{M-1}^*(n)}{wE_{M-1}^f(n-1)}$$

$$\begin{bmatrix} \bar{\mathbf{C}}_{M-1}(n) \\ \bar{c}_{\mathrm{MM}}^{(s)}(n) \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{\mathbf{K}}_{M-1}(n-1) \end{bmatrix} + \bar{c}_{M1}(n)\begin{bmatrix} 1 \\ \mathbf{a}_{M-1}(n-1) \end{bmatrix}$$

$$g_{M-1}^{(f)}(n) = x(n-M+1) + \mathbf{b}_{M-1}^t(n-1)\mathbf{X}_{M-1}(n)$$

$$\bar{c}_{\mathrm{MM}}^{(f)}(n) = -\frac{g_{M-1}^{(f)*}(n)}{wE_{M-1}^b(n-1)}$$

$$\bar{c}_{\mathrm{MM}}(n) = k_4\bar{c}_{\mathrm{MM}}^{(f)}(n) + (1-k_4)\bar{c}_{\mathrm{MM}}^{(s)}(n)$$

$$\bar{\mathbf{K}}_M(n) = \begin{bmatrix} \bar{\mathbf{C}}_{M-1}(n) \\ \bar{c}_{\mathrm{MM}}(n) \end{bmatrix}$$

$$g_{M-1}^{(s)}(n) = -wE_{M-1}^b(n-1)\bar{c}_{\mathrm{MM}}^{(s)*}(n)$$

$$g_{M-1}^{(i)}(n) = k_i g_{M-1}^{(f)}(n) + (1-k_i)g_{M-1}^{(s)}(n), \quad i = 1, 2, 5$$

$$\bar{\mathbf{K}}_{M-1}(n) = \bar{\mathbf{C}}_{M-1}(n) - \mathbf{b}_{M-1}(n-1)\bar{c}_{\mathrm{MM}}(n)$$

$$\bar{\alpha}_M(n) = \bar{\alpha}_{M-1}(n-1) + \bar{c}_{M1}(n)f_{M-1}(n)$$

$$\bar{\alpha}_{M-1}^{(s)}(n) = \bar{\alpha}_M(n) + g_{M-1}^{(s)}(n)\bar{c}_{\mathrm{MM}}^{(s)}(n)$$

$$\bar{\alpha}_{M-1}^{(f)}(n) = 1 + \bar{\mathbf{K}}_{M-1}^t(n)\mathbf{X}_{M-1}(n)$$

$$\bar{\alpha}_{M-1}(n) = k_3\bar{\alpha}_{M-1}^{(f)}(n) + (1-k_3)\bar{\alpha}_{M-1}^{(s)}(n)$$

$$E_{M-1}^f(n) = wE_{M-1}^f(n-1) + f_{M-1}(n)f_{M-1}^*(n,n)$$

$$\left[\text{or,} \quad \frac{1}{E_{M-1}^f(n)} = \frac{1}{w}\frac{1}{E_{M-1}^f(n-1)} - \frac{|\bar{c}_{M1}(n)|^2}{\bar{\alpha}_{M-1}^{(s)}(n)}\right]$$

$$g_{M-1}^{(i)}(n,n) = \frac{g_{M-1}^{(i)}(n)}{\bar{\alpha}_{M-1}(n)}, \quad i = 1, 2$$

$$\mathbf{b}_{M-1}(n) = \mathbf{b}_{M-1}(n-1) + \bar{\mathbf{K}}_{M-1}(n)g_{M-1}^{(1)}(n,n)$$

$$E_{M-1}^b(n) = wE_{M-1}^b(n-1) + g_{M-1}^{(2)}(n)g_{M-1}^{(2)*}(n,n)$$

$$e_M(n) = d(n) - \mathbf{h}_M^t(n-1)\mathbf{X}_M(n)$$

$$e_M(n,n) = \frac{e_M(n)}{\bar{\alpha}_M(n)}$$

$$\mathbf{h}_M(n) = \mathbf{h}_M(n-1) + \bar{\mathbf{K}}_M(n)e_M(n,n)$$

TABLE 13.9 A Simplified Stabilized Fast RLS Algorithm

$$
\begin{aligned}
f_{M-1}(n) &= x(n) + \mathbf{a}'_{M-1}(n-1)\mathbf{X}_{M-1}(n-1) \\
f_{M-1}(n,n) &= \frac{f_{M-1}(n)}{\bar{\alpha}_{M-1}(n-1)} \\
\mathbf{a}_{M-1}(n) &= \mathbf{a}_{M-1}(n-1) - \bar{\mathbf{K}}_{M-1}(n-1) f_{M-1}(n,n) \\
\bar{c}_{M1}(n) &= \frac{f^*_{M-1}(n)}{w E^f_{M-1}(n-1)} \\
\bar{\mathbf{K}}_M(n) \equiv \begin{bmatrix} \bar{\mathbf{C}}_{M-1}(n) \\ \bar{c}_{\mathrm{MM}}(n) \end{bmatrix} &= \begin{bmatrix} 0 \\ \bar{\mathbf{K}}_{M-1}(n-1) \end{bmatrix} + \frac{f^*_{M-1}(n)}{w E^f_{M-1}(n-1)} \begin{bmatrix} 1 \\ \mathbf{a}_{M-1}(n-1) \end{bmatrix} \\
g^{(f)}_{M-1}(n) &= x(n-M+1) + \mathbf{b}'_{M-1}(n-1)\mathbf{X}_{M-1}(n) \\
g^{(s)}_{M-1}(n) &= -w E^b_{M-1}(n-1)\bar{c}^*_{\mathrm{MM}}(n) \\
g^{(i)}_{M-1}(n) &= k_i g^{(f)}_{M-1}(n) + (1-k_i) g^{(s)}_{M-1}(n), \quad i = 1,2 \\
\bar{\mathbf{K}}_{M-1}(n) &= \bar{\mathbf{C}}_{M-1}(n) - \mathbf{b}_{M-1}(n-1)\bar{c}_{\mathrm{MM}}(n) \\
\bar{\alpha}_M(n) &= \bar{\alpha}_{M-1}(n-1) + \bar{c}_{M1}(n) f_{M-1}(n) \\
\bar{\alpha}_{M-1}(n) &= \bar{\alpha}_M(n) + g^{(f)}_{M-1}(n)\bar{c}_{\mathrm{MM}}(n) \\
E^f_{M-1}(n) &= w E^f_{M-1}(n-1) + f_{M-1}(n) f^*_{M-1}(n,n) \\
g^{(i)}_{M-1}(n,n) &= \frac{g^{(i)}_{M-1}(n)}{\bar{\alpha}_{M-1}(n)}, \quad i = 1,2 \\
\mathbf{b}_{M-1}(n) &= \mathbf{b}_{M-1}(n-1) + \bar{\mathbf{K}}_{M-1}(n) g^{(1)}_{M-1}(n,n) \\
E^b_{M-1}(n) &= w E^b_{M-1}(n-1) + g^{(2)}_{M-1}(n) g^{(2)*}_{M-1}(n,n) \\
e_M(n) &= d(n) - \mathbf{h}'_M(n-1)\mathbf{X}_M(n) \\
e_M(n,n) &= \frac{e_M(n)}{\bar{\alpha}_M(n)} \\
\mathbf{h}_M(n) &= \mathbf{h}_M(n-1) + \bar{\mathbf{K}}_M(n) e_M(n,n)
\end{aligned}
$$

is not critically affected by proper initialization (although, enventually, it will diverge). Consequently, we may initially use $g^{(f)}_{M-1}(n)$ in place of $g^{(s)}_{M-1}(n)$ (or their linear combination) for the first few hundred iterations, and then switch to the form for the stabilized fast RLS algorithm. By doing so, we obtain a stabilized fast RLS algorithm that is also insensitive to initial conditions.

13.4.2 Other Lattice Algorithms

Another type of RLS lattice algorithm is obtained by normalizing the forward and backward prediction errors through division of the errors by $\sqrt{E^f_m(n)}$ and $\sqrt{E^b_m(n)}$, respectively, and multiplication by $\sqrt{\alpha_m(n-1)}$ and $\sqrt{\alpha_m(n)}$, respectively. The resulting lattice algorithm is called a square-root or angle-and-power normalized RLS

lattice algorithm. This algorithm has a more compact form than the other forms of RLS lattice algorithms. However, the algorithm requires many square-root operations, which can be computationally complex. This problem can be solved by using CORDIC processors, which compute a square root in N clock cycles, where N is the number of bits of the computer word length. A description of the square-root/normalized RLS lattice algorithm and the CORDIC algorithm is given in the book by Proakis et al. (2002).

It is also possible to simplify the computational complexity of the RLS algorithms described in the previous section at the expense of compromising the convergence rate. One such algorithm is called *the gradient-lattice algorithm.* In this algorithm each stage of the lattice filter is characterized by the output–input relations

$$\begin{aligned} f_m(n) &= f_{m-1}(n) - k_m(n)g_{m-1}(n-1) \\ g_m(n) &= g_{m-1}(n-1) - k_m^*(n)f_{m-1}(n) \end{aligned} \tag{13.4.102}$$

where $k_m(n)$ is the reflection coefficient in the mth stage of the lattice and $f_m(n)$ and $g_m(n)$ are the forward and backward residuals.

This form of the lattice filter is identical to the Levinson-Durbin algorithm, except that now $k_m(n)$ is allowed to vary with time so that the lattice filter adapts to the time variations in the signal statistics. The reflection coefficients $\{k_m(n)\}$ may be optimized by employing the method of least squares, which results in the solution

$$k_m(n) = \frac{2\sum_{l=0}^{n} w^{n-l} f_{m-1}(l)g_{m-1}^*(l-1)}{\sum_{l=0}^{n} w^{n-l}[|f_{m-1}(l)|^2 + |g_{m-1}(l-1)|^2]}, \quad m = 1, 2, \ldots M-1 \tag{13.4.103}$$

These coefficients can also be updated recursively in time. The ladder coefficients are computed recursively in time by employing an LMS-type algorithm that is obtained by applying the mean-square-error criterion. A description of this algorithm is given in the paper by Griffiths (1978) and in the book by Proakis et al. (2002).

13.4.3 Properties of Lattice-Ladder Algorithms

The lattice algorithms that we have derived in the two previous subsections have a number of desirable properties. In this subsection, we consider the properties of these algorithms and compare them with the corresponding properties of the LMS algorithm and the RLS direct-form FIR filtering algorithms.

Convergence Rate. The RLS lattice-ladder algorithms basically have the same convergence rate as the RLS direct-form FIR filter structures. This characteristic behavior is not surprising, since both filter structures are optimum in the least-squares sense. Although the gradient lattice algorithm retains some of the optimal characteristics of the RLS lattice, nevertheless the former is not optimum in the least-squares sense and, hence, its convergence rate is slower.

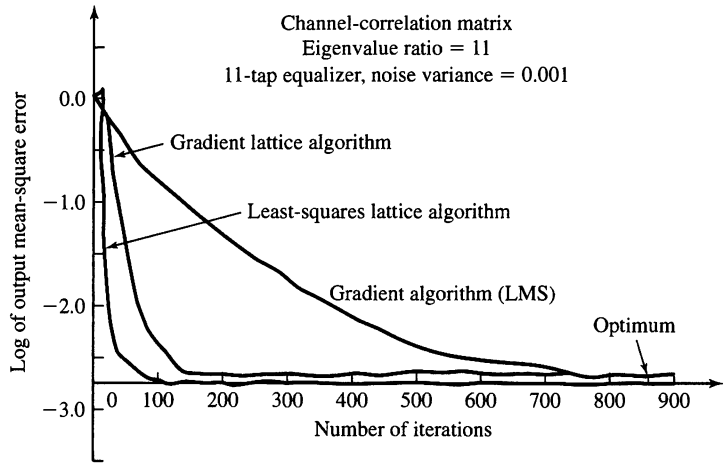


Figure 13.4.3 Learning curves for RLS lattice, gradient lattice, and LMS algorithms for adaptive equalizer of length $M = 11$. (From *Digital Communications* by John G. Proakis. ©1989 by McGraw-Hill Book Company. Reprinted with permission of the publisher.)

For comparison purposes, Figures 13.4.3 and 13.4.4 illustrate the learning curves for an adaptive equalizer of length $M = 11$, implemented as an RLS lattice-ladder filter, a gradient lattice-ladder filter, and a direct-form FIR filter using the LMS algorithm, for a channel autocorrelation matrix that has eigenvalue ratios of $\lambda_{max}/\lambda_{min} = 11$ and $\lambda_{max}/\lambda_{min} = 21$, respectively. From these learning curves, we observe that the gradient lattice algorithm takes about twice as many iterations to converge as the optimum RLS lattice algorithm. Furthermore, the gradient lattice algorithm provides significantly faster convergence than the LMS algorithm. For both lattice structures, the convergence rate does not depend on the eigenvalue spread of the correlation matrix.

Computational Requirements. The RLS lattice algorithms described in the previous subsection have a computational complexity that is proportional to M. In contrast, the computational complexity of the RLS square-root algorithms is proportional to M^2. On the other hand, the direct-form fast algorithms, which are a derivative of the lattice algorithm, have a complexity proportional to M, and they are a little more efficient than the lattice-ladder algorithms.

In Figure 13.4.5 we illustrate the computational complexity (number of complex multiplications and divisions) of the various adaptive filtering algorithms that we have described. Clearly, the LMS algorithm requires the fewest computations. The fast RLS algorithms in Tables 13.3 and 13.9 are the most efficient of the RLS algorithms shown, closely followed by the gradient lattice algorithm, then the RLS lattice algorithms and, finally, the square-root algorithms. Note that for small values of M, there is little difference in complexity among the rapidly convergent algorithms.

Numerical Properties. In addition to providing fast convergence, the RLS and gradient lattice algorithms are numerically robust. First, these lattice algorithms are

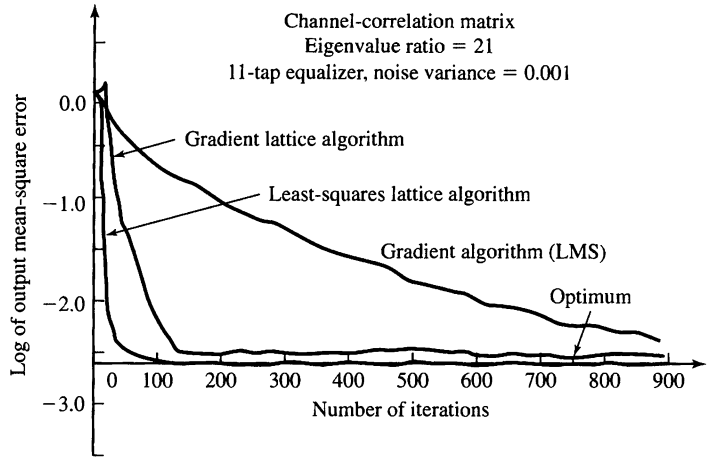


Figure 13.4.4 Learning curves for RLS lattice, gradient lattice, and LMS algorithms for adaptive equalizer of length $M = 11$. (From *Digital Communications* by John G. Proakis. ©1989 by McGraw-Hill Book Company. Reprinted with permission of the publisher.)

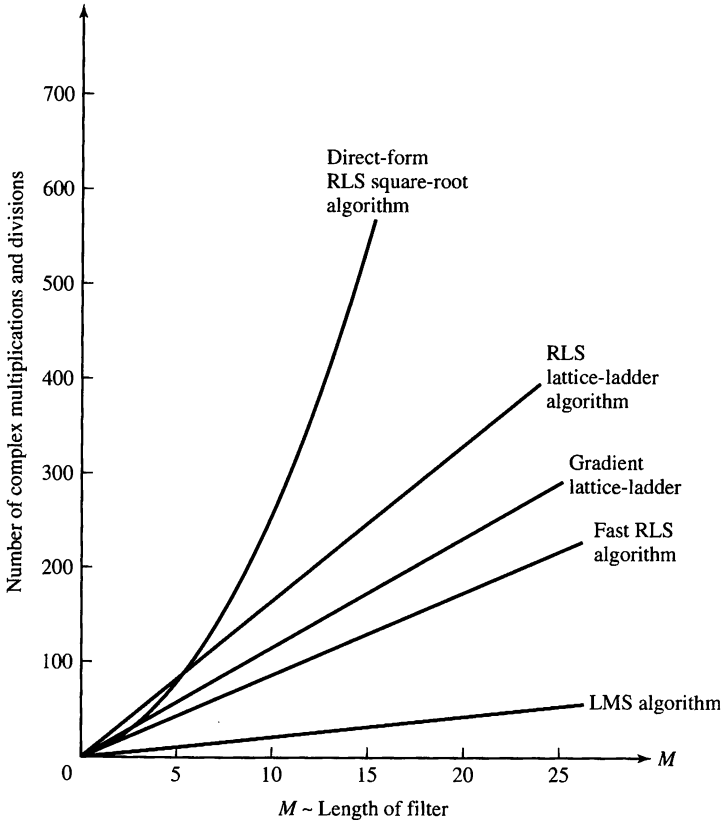


Figure 13.4.5 Computational complexity of adaptive filter algorithms.

TABLE 13.10 Numerical Accuracy, in Terms of Output MSE for Channel with $\lambda_{max}/\lambda_{min} = 11$ and $w = 0.975$, MSE $\times 10^{-3}$

Number of bits (including sign)	Algorithm				
	RLS square root	Fast RLS	Conventional RLS lattice	Error feedback RLS lattice	LMS
16	2.17	2.17	2.16	2.16	2.30
13	2.33	2.21	3.09	2.22	2.30
11	6.14	3.34	25.2	3.09	19.0
9	75.3	[a]	365	31.6	311

[a] Algorithm did not converge.

numerically stable, which means that the output estimation error from the computational procedure is bounded when a bounded error signal is introduced at the input. Second, the numerical accuracy of the optimum solution is also relatively good when compared to the LMS and the RLS direct-form FIR algorithms.

For purposes of comparison, we illustrate in Table 13.10 the steady-state average squared error or (estimated) minimum MSE obtained through computer simulation from the two RLS lattice algorithms and the direct-form FIR filter algorithms described in Section 13.2. The striking result in Table 13.10 is the superior performance obtained with the RLS lattice-ladder algorithm, in which the reflection coefficients and the ladder gain are updated directly according to (13.4.96–13.4.98). This is the error-feedback form of the RLS lattice algorithm. It is clear that the direct updating of these coefficients is significantly more robust to round-off errors than all the other adaptive algorithms, including the LMS algorithm. It is also apparent that the two-step process used in the conventional RLS lattice algorithm to estimate the reflection coefficients is not as accurate. Furthermore, the estimation errors that are generated in the coefficients at each stage propagate from stage to stage, causing additional errors.

The effect of changing the weighting factor w is illustrated in the numerical results given in Table 13.11. In this table, we give the minimum (estimated) MSE

TABLE 13.11 Numerical Accuracy, in Terms of Output MSE, of A Priori Least-Squares Lattice Algorithm with Different Values of the Weighting Factor w, MSE, $\times 10^{-3}$

Number of bits with sign	Algorithm					
	$w = 0.99$		$w = 0.975$		$w = 0.95$	
	Conventional	Error feedback	Conventional	Error feedback	Conventional	Error feedback
16	2.14	2.08	2.18	2.16	2.66	2.62
13	7.08	2.11	3.09	2.22	3.65	2.66
11	39.8	3.88	25.2	3.09	15.7	2.78
9	750	44.1	365	31.6	120	15.2

obtained with the conventional and error-feedback forms of the RLS lattice algorithm. We observe that the output MSE decreases with an increase in the weighting factor when the precision is high (13 bits and 16 bits). This reflects the improvement in performance obtained by increasing the observation interval. As the number of bits of precision is decreased, we observe that the weighting factor should also be decreased in order to maintain good performance. In effect, with low precision, the effect of a longer averaging time results in a larger round-off noise. Of course, these results were obtained with time-invariant signal statistics. If the signal statistics are time-variant, the rate of the time variations will also influence the choice of w.

In the gradient lattice algorithm, the reflection coefficients and the ladder gains are also updated directly. Consequently, the numerical accuracy of the gradient lattice algorithm is comparable to that obtained with the direct update form of the RLS lattice.

Analytical and simulation results on numerical stability and numerical accuracy in fixed-point implementation of these algorithms can be found in Ling and Proakis (1984), Ling, Manolakis, and Proakis (1985; 1986), Ljung and Ljung (1985), and Gardner (1984).

Implementation Considerations. As we have observed, the lattice filter structure is highly modular and allows for the computations to be pipelined. Because of the high degree of modularity, the RLS and gradient lattice algorithms are particularly suitable for implementation in VLSI. As a result of this advantage in implementation and the desirable properties of stability, excellent numerical accuracy, and fast convergence, we anticipate that adaptive filters will be increasingly implemented as lattice-ladder structures in the near future.

13.5 Summary and References

We have presented adaptive algorithms for direct-form FIR and lattice filter structures. The algorithms for the direct-form FIR filter consisted of the simple LMS algorithm due to Widrow and Hoff (1960) and the direct-form, time-recursive least-squares algorithms, including the conventional RLS form given by (13.3.23–13.3.27), the square-root RLS forms described by Bierman (1977), Carlson and Culmone (1979), and Hsu (1982), and the RLS fast Kalman algorithms, one form of which was described by Falconer and Ljung (1978), and other forms later derived by Carayannis, Manolakis, and Kalouptsidis (1983), Proakis (1989), and Cioffi and Kailath (1984).

Of these algorithms, the LMS algorithm is the simplest. It is used in many applications where its slow convergence is adequate. Of the direct-form RLS algorithms, the square-root algorithms have been used in applications where fast convergence is required. The algorithms have good numerical properties. The family of stabilized fast RLS algorithms is very attractive from the viewpoint of computational efficiency. Methods to avoid instability due to round-off errors have been proposed by Hsu (1982), Cioffi and Kailath (1984), Lin (1984), Eleftheriou and Falconer (1987), and Slock and Kailath (1988; 1991).

The adaptive lattice-ladder filter algorithm derived in this chapter is the optimum RLS lattice-ladder algorithms (in both conventional and error-feedback form). Only the a priori form of the lattice-ladder algorithm was derived, which is the form most

often used in applications. In addition, there is an a posteriori form of the RLS lattice-ladder algorithms (both conventional and error-feedback), as described by Ling, Manolakis, and Proakis (1986). The error-feedback form of the RLS lattice-ladder algorithm has excellent numerical properties, and is particularly suitable for implementation in fixed-point arithmetic and in VLSI.

In the direct-form and lattice RLS algorithms, we used exponential weighting into the past in order to reduce the effective memory in the adaptation process. As an alternative to exponential weighting, we may employ finite-length uniform weighting into the past. This approach leads to the class of finite-memory RLS direct-form and lattice structures described in Cioffi and Kalaith (1985) and Manolakis, Ling, and Proakis (1987).

In addition to the various algorithms that we have presented in this chapter, there has been considerable research into efficient implementation of these algorithms using systolic arrays and other parallel architectures. The reader is referred to Kung (1982), and Kung, Whitehouse, and Kailath (1985).

Problems

13.1 Use the least-squares criterion to determine the equations for the parameters of the FIR filter model in Figure 13.1.2, when the plant output is corrupted by additive noise, $w(n)$.

13.2 Determine the equations for the coefficients of an adaptive echo canceler based on the least-squares criterion. Use the configuration in Figure 13.1.8 and assume the presence of a near-end echo only.

13.3 If the sequences $w_1(n)$, $w_2(n)$, and $w_3(n)$ in the adaptive noise-canceling system shown in Figure 13.1.14 are mutually uncorrelated, determine the expected value of the estimated correlation sequences $r_{vv}(k)$ and $r_{yv}(k)$ contained in (13.1.26).

13.4 Prove the result in (13.4.34).

13.5 Derive the equation for the direct update of the ladder gain given by (13.4.98).

13.6 Derive the equation for the reflection coefficients in a gradient lattice-algorithm given in (13.4.103).

13.7 Derive the FAEST algorithm given in Table 13.7 by using the alternative Kalman gain vector

$$\tilde{\mathbf{K}}_M(n) = \frac{1}{w}\mathbf{P}_M(n-1)\mathbf{X}_M^*(n)$$

instead of the Kalman gain vector $\mathbf{K}_M(n)$.

13.8 The tap-leaky LMS algorithm proposed by Gitlin, Meadors, and Weinstein (1982) may be expressed as

$$\mathbf{h}_M(n+1) = w\mathbf{h}_M(n) + \Delta e(n)\mathbf{X}_M^*(n)$$

where $0 < w < 1$, Δ is the step size, and $\mathbf{X}_M(n)$ is the data vector at time n. Determine the condition for the convergence of the mean value of $\mathbf{h}_M(n)$.

13.9 The tap-leaky LMS algorithm in Problem 13.8 may be obtained by minimizing the cost function

$$\mathcal{E}(n) = |e(n)|^2 + c||\mathbf{h}_M(n)||^2$$

where c is a constant and $e(n)$ is the error between the desired filter output and the actual filter output. Show that the minimization of $\mathcal{E}(n)$ with respect to the filter coefficient vector $\mathbf{h}_M(n)$ leads to the following tap-leaky LMS algorithm.

$$\mathbf{h}_M(n+1) = (1 - \Delta c)\mathbf{h}_M(n) + \Delta e(n)\mathbf{X}_M^*(n)$$

13.10 For the normalized LMS algorithm given by (13.2.31), determine the range of values of step size Δ to ensure the stability of the algorithm in the mean-square-error sense.

13.11 By using the alternative Kalman gain vector given in Problem 13.8, modify the a priori fast least-squares algorithms given in Tables 13.2 and 13.3, and thus reduce the number of computations.

13.12 Consider the random process

$$x(n) = gv(n) + w(n), \qquad n = 0, 1, \ldots, M-1$$

where $v(n)$ is a known sequence, g is a random variable with $E[g] = 0$, and $E[g^2] = G$. The process $w(n)$ is a white noise sequence with

$$\gamma_{ww}(m) = \sigma_w^2 \delta(m)$$

Determine the coefficients of the linear estimator for g, that is,

$$\hat{g} = \sum_{n=0}^{M-1} h(n)x(n)$$

that minimizes the mean-square error

$$\mathcal{E} = e[(g - \hat{g})^2]$$

13.13 Recall that an FIR filter can be realized in the frequency-sampling form with system function

$$H(z) = \frac{1 - z^{-M}}{M} \sum_{k=0}^{M-1} \frac{H_k}{1 - e^{j2\pi k/M} z^{-1}}$$

$$= H_1(z)H_2(z)$$

where $H_1(z)$ is the comb filter and $H_2(z)$ is the parallel bank of resonators.

(a) Suppose that this structure is implemented as an adaptive filter using the LMS algorithm to adjust the filter (DFT) parameters H_k. Give the time-update equation for these parameters. Sketch the adaptive filter structure.

(b) Suppose that this structure is used as an adaptive channel equalizer, in which the desired signal is

$$d(n) = \sum_{k=0}^{M-1} A_k \cos \omega_k n, \qquad \omega_k = \frac{2\pi k}{M}$$

With this form for the desired signal, what advantages are there in the LMS adaptive algorithm for the DFT coefficients H_k over the direct-form structure with coefficients $h(n)$? (*Hint*: Refer to Proakis (1970).)

13.14 Consider the performance index

$$J = h^2 - 40h + 28$$

Suppose that we search for the minimum of J by using the steepest-descent algorithm

$$h(n+1) = h(n) - \frac{1}{2}\Delta g(n)$$

where $g(n)$ is the gradient.

(a) Determine the range of values of Δ that provides an overdamped system for the adjustment process.

(b) Plot the expression for J as a function of n for a value of Δ in this range.

13.15 Consider the noise-canceling adaptive filter shown in Figure 13.1.14. Assume that the additive noise processes are white and mutually uncorrelated, with equal variances σ_w^2. Suppose that the linear system has a known system function

$$H(z) = \frac{1}{1 - \frac{1}{2}z^{-1}}$$

Determine the optimum weights of a three-tap noise canceler that minimizes the MSE.

13.16 Determine the coefficients a_1 and a_2 for the linear predictor shown in Figure P13.16, given that the autocorrelation $\gamma_{xx}(n)$ of the input signal is

$$\gamma_{xx}(m) = a^{|m|}, \qquad 0 < a < 1$$

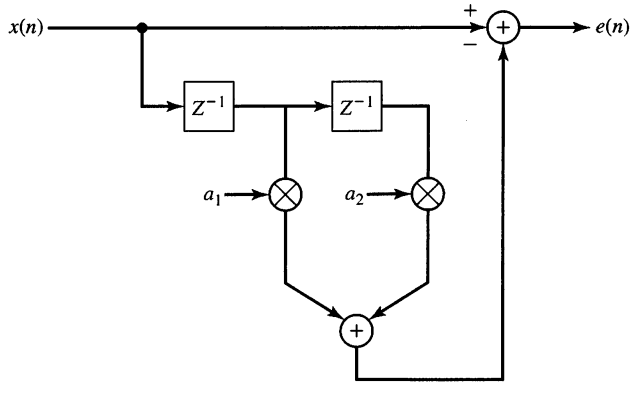


Figure P13.16

13.17 Determine the lattice filter and its optimum reflection coefficients corresponding to the linear predictor in Problem 13.16.

13.18 Consider the adaptive FIR filter shown in Figure P13.18. The system $C(z)$ is characterized by the system function

$$C(z) = \frac{1}{1 + 0.9z^{-1}}$$

Determine the optimum coefficients of the adaptive FIR filter $B(z) = b_0 + b_1 z^{-1}$ that minimize the MSE. The additive noise is white with variance $\sigma_w^2 = 0.1$.

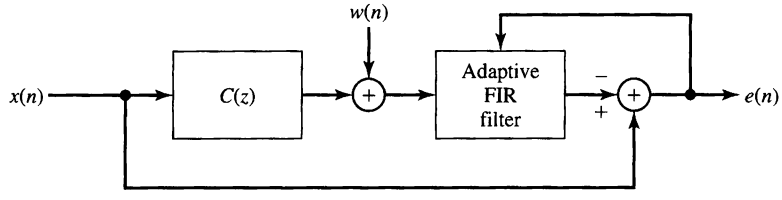


Figure P13.18

13.19 In the gradient lattice algorithm, the forward and backward prediction errors are given by (13.4.102).

(a) Show that the minimization of the least-squares error

$$\mathcal{E}_m^{\text{LS}} = \sum_{l=0}^{n} w^{n-l}[|fm(l)|^2 + |gm(l)|^2]$$

with respect to the reflection coefficients $\{k_m(n)\}$ results in the equation given by (13.4.103).

(b) To determine an equation for the recursive computation of the reflection coefficients given by (13.4.103) define

$$u_m(n) = wu_m(n-1) + 2f_{m-1}(n)g_{m-1}^*(n-1)$$

$$v_m(n) = wv_m(n-1) + |f_{m-1}(n)|^2 + |g_{m-1}(n-1)|^2$$

so that $k_m(n) = u_m(n)/v_m(n)$. Then, show that $k_m(n)$ can be computed recursively by the relation

$$k_m(n) = k_m(n-1) + \frac{f_m(n)g_{m-1}^*(n-1) + g_m^*(n)f_{m-1}(n)}{wv_m(n-1)}$$

13.20 Consider the adaptive predictor shown in Figure P13.16.

(a) Determine the quadratic performance index and the optimum parameters for the signal

$$x(n) = \sin\frac{n\pi}{4} + w(n)$$

where $w(n)$ is white noise with variance $\sigma_w^2 = 0.1$.

(b) Generate a sequence of 1000 samples of $x(n)$, and use the LMS algorithm to adaptively obtain the predictor coefficients. Compare the experimental results with the theoretical values obtained in part (a). Use a step size of $\Delta \leq \frac{1}{10}\Delta_{\max}$.

(c) Repeat the experiment in part (b) for $N = 10$ trials with different noise sequences, and compute the average values of the predictor coefficients. Comment on how these results compare with the theoretical values in part (a).

13.21 An autoregressive process is described by the difference equation

$$x(n) = 1.26x(n-1) - 0.81x(n-2) + w(n)$$

(a) Generate a sequence of $N = 1000$ samples of $x(n)$, where $w(n)$ is a white noise sequence with variance $\sigma_w^2 = 0.1$. Use the LMS algorithm to determine the parameters of a second-order ($p = 2$) linear predictor. Begin with $a_1(0) = a_2(0) = 0$. Plot the coefficients $a_1(n)$ and $a_2(n)$ as a function of the iteration number.

(b) Repeat part (a) for 10 trials, using different noise sequences, and superimpose the 10 plots of $a_1(n)$ and $a_2(n)$.

(c) Plot the learning curve for the average (over the 10 trials) MSE for the data in part (b).

13.22 A random process $x(n)$ is given as

$$\begin{aligned} x(n) &= s(n) + w(n) \\ &= \sin(\omega_0 n + \phi) + w(n), \qquad \omega_0 = \pi/4, \quad \phi = 0 \end{aligned}$$

where $w(n)$ is an additive white noise sequence with variance $\sigma_w^2 = 0.1$.

(a) Generate $N = 1000$ samples of $x(n)$ and simulate an adaptive line enhancer of length $L = 4$. Use the LMS algorithm to adapt the ALE.

(b) Plot the output of the ALE.

(c) Compute the autocorrelation $\gamma_{xx}(m)$ of the sequence $x(n)$.

(d) Determine the theoretical values of the ALE coefficients and compare them with the experimental values.

(e) Compute and plot the frequency response of the linear predictor (ALE).

(f) Compute and plot the frequency response of the prediction-error filter.

(g) Compute and plot the experimental values of the autocorrelation $r_{ee}(m)$ of the output error sequence for $0 \leq m < 10$.

(h) Repeat the experiment for 10 trials, using different noise sequences, and superimpose the frequency response plots on the same graph.

(i) Comment on the result in parts (a) through (h).

CHAPTER

14

Power Spectrum Estimation

In this chapter we are concerned with the estimation of the spectral characteristics of signals characterized as random processes. Many of the phenomena that occur in nature are best characterized statistically in terms of averages. For example, meteorological phenomena such as the fluctuations in air temperature and pressure are best characterized statistically as random processes. Thermal noise voltages generated in resistors and electronic devices are additional examples of physical signals that are well modeled as random processes.

Due to the random fluctuations in such signals, we must adopt a statistical viewpoint, which deals with the average characteristics of random signals. In particular, the autocorrelation function of a random process is the appropriate statistical average that we will use for characterizing random signals in the time domain, and the Fourier transform of the autocorrelation function, which yields the power density spectrum, provides the transformation from the time domain to the frequency domain.

Power spectrum estimation methods have a relatively long history. For a historical perspective, the reader is referred to the paper by Robinson (1982) and the book by Marple (1987). Our treatment of this subject covers the classical power spectrum estimation methods based on the periodogram, originally introduced by Schuster (1898), and by Yule (1927), who originated the modern model-based or parametric methods. These methods were subsequently developed and applied by Walker (1931), Bartlett (1948), Parzen (1957), Blackman and Tukey (1958), Burg (1967), and others. We also describe the method of Capon (1969) and methods based on eigenanalysis of the data correlation matrix.

14.1 Estimation of Spectra from Finite-Duration Observations of Signals

The basic problem that we consider in this chapter is the estimation of the power density spectrum of a signal from observation of the signal over a finite time interval. As we will see, the finite record length of the data sequence is a major limitation on the quality of the power spectrum estimate. When dealing with signals that are statistically stationary, the longer the data record, the better the estimate that can be extracted from the data. On the other hand, if the signal statistics are nonstationary, we cannot select an arbitrarily long data record to estimate the spectrum. In such a case, the length of the data record that we select is determined by the rapidity of the time variations in the signal statistics. Ultimately, our goal is to select as short a data record as possible that still allows us to resolve the spectral characteristics of different signal components in the data record that have closely spaced spectra.

One of the problems that we encounter with classical power spectrum estimation methods based on a finite-length data record is the distortion of the spectrum that we are attempting to estimate. This problem occurs in both the computation of the spectrum for a deterministic signal and the estimation of the power spectrum of a random signal. Since it is easier to observe the effect of the finite length of the data record on a deterministic signal, we treat this case first. Thereafter, we consider only random signals and the estimation of their power spectra.

14.1.1 Computation of the Energy Density Spectrum

Let us consider the computation of the spectrum of a deterministic signal from a finite sequence of data. The sequence $x(n)$ is usually the result of sampling a continuous-time signal $x_a(t)$ at some uniform sampling rate F_s. Our objective is to obtain an estimate of the true spectrum from a finite-duration sequence $x(n)$.

Recall that if $x(t)$ is a finite-energy signal, that is,

$$E = \int_{-\infty}^{\infty} |x_a(t)|^2 dt < \infty$$

then its Fourier transform exists and is given as

$$X_a(F) = \int_{-\infty}^{\infty} x_a(t) e^{-j2\pi Ft} dt$$

From Parseval's theorem we have

$$E = \int_{-\infty}^{\infty} |x_a(t)|^2 dt = \int_{-\infty}^{\infty} |X_a(F)|^2 dF \tag{14.1.1}$$

The quantity $|X_a(F)|^2$ represents the distribution of signal energy as a function of frequency, and hence it is called the energy density spectrum of the signal, that is,

$$S_{xx}(F) = |X_a(F)|^2 \tag{14.1.2}$$

as described in Chapter 4. Thus the total energy in the signal is simply the integral of $S_{xx}(F)$ over all F [i.e., the total area under $S_{xx}(F)$].

It is also interesting to note that $S_{xx}(F)$ can be viewed as the Fourier transform of another function, $R_{xx}(\tau)$, called the *autocorrelation function* of the finite-energy signal $x_a(t)$, defined as

$$R_{xx}(\tau) = \int_{-\infty}^{\infty} x_a^*(t)x_a(t+\tau)\,dt \tag{14.1.3}$$

Indeed, it easily follows that

$$\int_{-\infty}^{\infty} R_{xx}(\tau)e^{-j2\pi F\tau}\,d\tau = S_{xx}(F) = |X_a(F)|^2 \tag{14.1.4}$$

so that $R_{xx}(\tau)$ and $S_{xx}(F)$ are a Fourier transform pair.

Now suppose that we compute the energy density spectrum of the signal $x_a(t)$ from its samples taken at the rate F_s samples per second. To ensure that there is no spectral aliasing resulting from the sampling process, the signal is assumed to be prefiltered, so that, for practical purposes, its bandwidth is limited to B hertz. Then the sampling frequency F_s is selected such that $F_s > 2B$.

The sampled version of $x_a(t)$ is a sequence $x(n)$, $-\infty < n < \infty$, which has a Fourier transform (voltage spectrum)

$$X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$$

or, equivalently,

$$X(f) = \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fn} \tag{14.1.5}$$

Recall that $X(f)$ can be expressed in terms of the voltage spectrum of the analog signal $x_a(t)$ as

$$X(F) = F_s \sum_{k=-\infty}^{\infty} X_a(F - kF_s) \tag{14.1.6}$$

where $f = F/F_s$ is the normalized frequency variable.

In the absence of aliasing, within the fundamental range $|F| \leq F_s/2$, we have

$$X(F) = F_s X_a(F), \quad |F| \leq F_s/2 \tag{14.1.7}$$

Hence the voltage spectrum of the sampled signal is identical to the voltage spectrum of the analog signal. As a consequence, the energy density spectrum of the sampled signal is

$$S_{xx}(F) = |X(F)|^2 = F_s^2|X_a(F)|^2 \tag{14.1.8}$$

We can proceed further by noting that the autocorrelation of the sampled signal, which is defined as

$$r_{xx}(k) = \sum_{n=-\infty}^{\infty} x^*(n)x(n+k) \tag{14.1.9}$$

has the Fourier transform (Wiener–Khintchine theorem)

$$S_{xx}(f) = \sum_{k=-\infty}^{\infty} r_{xx}(k)e^{-j2\pi kf} \tag{14.1.10}$$

Hence the energy density spectrum can be obtained by the Fourier transform of the autocorrelation of the sequence $\{x(n)\}$.

The relations above lead us to distinguish between two distinct methods for computing the energy density spectrum of a signal $x_a(t)$ from its samples $x(n)$. One is the *direct method*, which involves computing the Fourier transform of $\{x(n)\}$, and then

$$\begin{aligned} S_{xx}(f) &= |X(f)|^2 \\ &= \left| \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fn} \right|^2 \end{aligned} \tag{14.1.11}$$

The second approach is called the *indirect method* because it requires two steps. First, the autocorrelation $r_{xx}(k)$ is computed from $x(n)$ and then the Fourier transform of the autocorrelation is computed as in (14.1.10) to obtain the energy density spectrum.

In practice, however, only the finite-duration sequence $x(n)$, $0 \le n \le N-1$, is available for computing the spectrum of the signal. In effect, limiting the duration of the sequence $x(n)$ to N points is equivalent to multiplying $x(n)$ by a rectangular window. Thus we have

$$\tilde{x}(n) = x(n)w(n) = \begin{cases} x(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{14.1.12}$$

From our discussion of FIR filter design based on the use of windows to limit the duration of the impulse response, we recall that multiplication of two sequences is equivalent to convolution of their voltage spectra. Consequently, the frequency-domain relation corresponding to (14.1.12) is

$$\begin{aligned} \tilde{X}(f) &= X(f) * W(f) \\ &= \int_{-1/2}^{1/2} X(\alpha)W(f-\alpha)d\alpha \end{aligned} \tag{14.1.13}$$

Recall from our discussion in Section 10.2.1 that convolution of the window function $W(f)$ with $X(f)$ smooths the spectrum $X(f)$, provided that the spectrum $W(f)$ is relatively narrow compared to $X(f)$. But this condition implies that the window $w(n)$ be sufficiently long (i.e., N must be sufficiently large) such that $W(f)$

is narrow compared to $X(f)$. Even if $W(f)$ is narrow compared to $X(f)$, the convolution of $X(f)$ with the sidelobes of $W(f)$ results in sidelobe energy in $\tilde{X}(f)$, in frequency bands where the true signal spectrum $X(f) = 0$. This sidelobe energy is called *leakage*. The following example illustrates the leakage problem.

EXAMPLE 14.1.1

A signal with (voltage) spectrum

$$X(f) = \begin{cases} 1, & |f| \leq 0.1 \\ 0, & \text{otherwise} \end{cases}$$

is convolved with the rectangular window of length $N = 61$. Determine the spectrum of $\tilde{X}(f)$ given by (14.1.13).

Solution. The spectral characteristic $W(f)$ for the length $N = 61$ rectangular window is illustrated in Fig. 10.2.2. Note that the width of the main lobe of the window function is $\Delta\omega = 4\pi/61$ or $\Delta f = 2/61$, which is narrow compared to $X(f)$.

The convolution of $X(f)$ with $W(f)$ is illustrated in Fig. 14.1.1. We note that energy has leaked into the frequency band $0.1 < |f| \leq 0.5$, where $X(f) = 0$. A part of this is due to the width of the main lobe in $W(f)$, which causes a broadening or smearing of $X(f)$ outside the range $|f| \leq 0.1$. However, the sidelobe energy in $\tilde{X}(f)$ is due to the presence of the sidelobes of $W(f)$, which are convolved with $X(f)$. The smearing of $X(f)$ for $|f| > 0.1$ and the sidelobes in the range $0.1 \leq |f| \leq 0.5$ constitute the leakage.

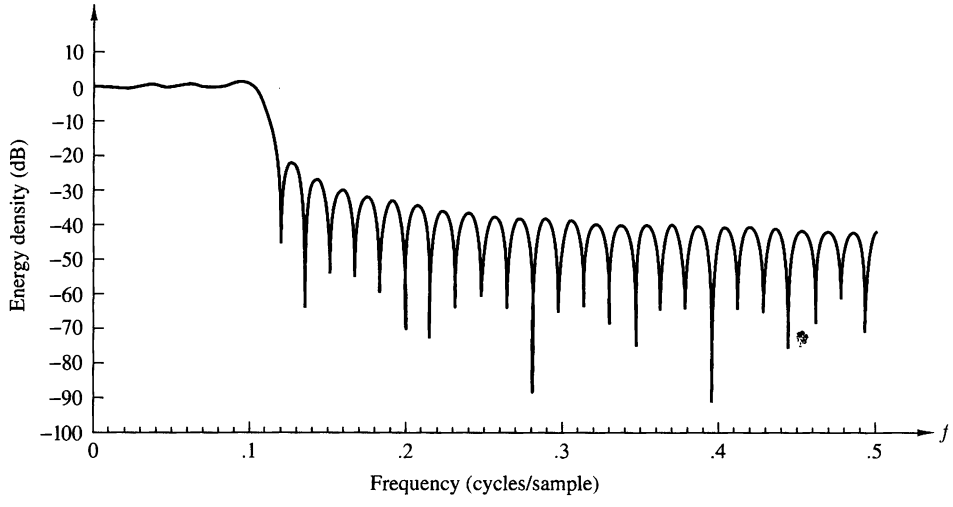


Figure 14.1.1 Spectrum obtained by convolving an $M = 61$ rectangular window with the ideal lowpass spectrum in Example 14.1.1.

Just as in the case of FIR filter design, we can reduce sidelobe leakage by selecting windows that have low sidelobes. This implies that the windows have a smooth time-domain cutoff instead of the abrupt cutoff in the rectangular window. Although such window functions reduce sidelobe leakage, they result in an increase in smoothing or

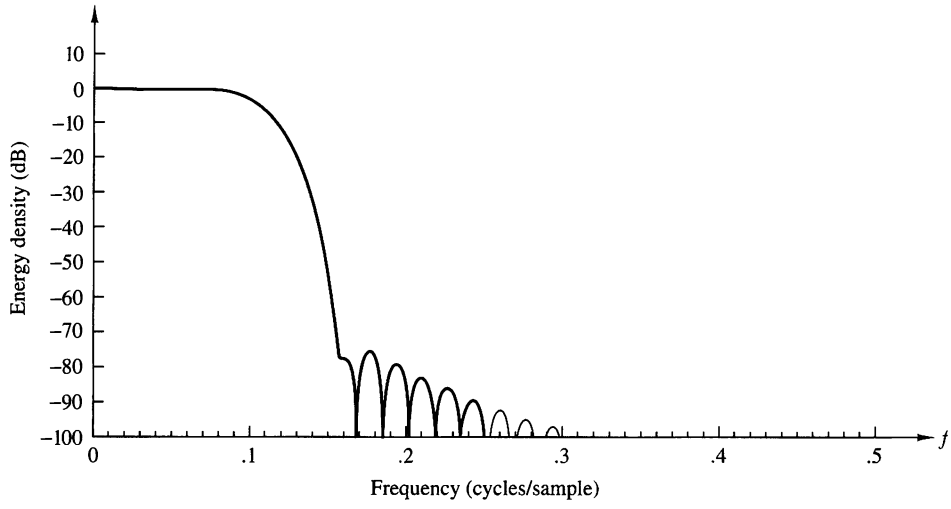


Figure 14.1.2 Spectrum obtained by convolving an $M = 61$ Blackman window with the ideal lowpass spectrum in Example 14.1.1.

broadening of the spectral characteristic $X(f)$. For example, the use of a Blackman window of length $N = 61$ in Example 14.1.1 results in the spectral characteristic $\tilde{X}(f)$ shown in Fig. 14.1.2. The sidelobe leakage has certainly been reduced, but the spectral width has been increased by about 50%.

The broadening of the spectrum being estimated due to windowing is particularly a problem when we wish to resolve signals with closely spaced frequency components. For example, the signal with spectral characteristic $X(f) = X_1(f) + X_2(f)$, as shown in Fig. 14.1.3, cannot be resolved as two separate signals unless the width of the window function is significantly narrower than the frequency separation Δf. Thus we observe that using smooth time-domain windows reduces leakage at the expense of a decrease in frequency resolution.

It is clear from this discussion that the energy density spectrum of the windowed sequence $\{\tilde{x}(n)\}$ is an approximation of the desired spectrum of the sequence $\{x(n)\}$.

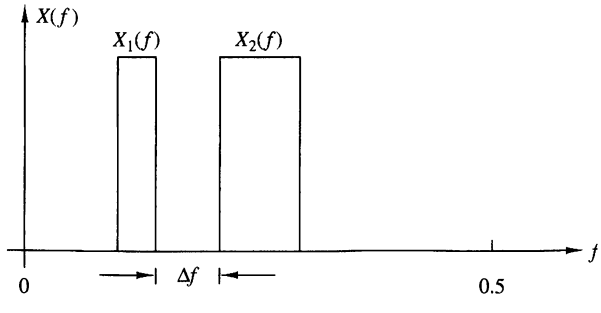


Figure 14.1.3
Two narrowband signal spectra.

The spectral density obtained from $\{\tilde{x}(n)\}$ is

$$S_{\tilde{x}\tilde{x}}(f) = |\tilde{X}(f)|^2 = \left|\sum_{n=0}^{N-1} \tilde{x}(n)e^{-j2\pi fn}\right|^2 \tag{14.1.14}$$

The spectrum given by (14.1.14) can be computed numerically at a set of N frequency points by means of the DFT. Thus

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n)e^{-j2\pi kn/N} \tag{14.1.15}$$

Then

$$|\tilde{X}(k)|^2 = S_{\tilde{x}\tilde{x}}(f)|_{f=k/N} = S_{\tilde{x}\tilde{x}}\left(\frac{k}{N}\right) \tag{14.1.16}$$

and hence

$$S_{\tilde{x}\tilde{x}}\left(\frac{k}{N}\right) = \left|\sum_{n=0}^{N-1} \tilde{x}(n)e^{-j2\pi kn/N}\right|^2 \tag{14.1.17}$$

which is a distorted version of the true spectrum $S_{xx}(k/N)$.

14.1.2 Estimation of the Autocorrelation and Power Spectrum of Random Signals: The Periodogram

The finite-energy signals considered in the preceding section possess a Fourier transform and are characterized in the spectral domain by their energy density spectrum. On the other hand, the important class of signals characterized as stationary random processes do not have finite energy and hence do not possess a Fourier transform. Such signals have finite average power and hence are characterized by a *power density spectrum*. If $x(t)$ is a stationary random process, its autocorrelation function is

$$\gamma_{xx}(\tau) = E[x^*(t)x(t+\tau)] \tag{14.1.18}$$

where $E[\cdot]$ denotes the statistical average. Then, via the Wiener–Khintchine theorem, the power density spectrum of the stationary random process is the Fourier transform of the autocorrelation function, that is,

$$\Gamma_{xx}(F) = \int_{-\infty}^{\infty} \gamma_{xx}(\tau)e^{-j2\pi F\tau}\, dt \tag{14.1.19}$$

In practice, we deal with a single realization of the random process from which we estimate the power spectrum of the process. We do not know the true autocorrelation function $\gamma_{xx}(\tau)$ and as a consequence, we cannot compute the Fourier transform in

(14.1.19) to obtain $\Gamma_{xx}(F)$. On the other hand, from a single realization of the random process we can compute the time-average autocorrelation function

$$R_{xx}(\tau) = \frac{1}{2T_0} \int_{-T_0}^{T_0} x^*(t)x(t+\tau)\, dt \tag{14.1.20}$$

where $2T_0$ is the observation interval. If the stationary random process is *ergodic* in the first and second moments (mean and autocorrelation function), then

$$\begin{aligned} \gamma_{xx}(\tau) &= \lim_{T_0 \to \infty} R_{xx}(\tau) \\ &= \lim_{T_0 \to \infty} \frac{1}{2T_0} \int_{-T_0}^{T_0} x^*(t)x(t+\tau)\, dt \end{aligned} \tag{14.1.21}$$

This relation justifies the use of the time-average autocorrelation $R_{xx}(\tau)$ as an estimate of the statistical autocorrelation function $\gamma_{xx}(\tau)$. Furthermore, the Fourier transform of $R_{xx}(\tau)$ provides an estimate $P_{xx}(F)$ of the power density spectrum, that is,

$$\begin{aligned} P_{xx}(F) &= \int_{-T_0}^{T_0} R_{xx}(\tau) e^{-j2\pi F\tau}\, d\tau \\ &= \frac{1}{2T_0} \int_{-T_0}^{T_0} \left[\int_{-T_0}^{T_0} x^*(t)x(t+\tau)dt \right] e^{-j2\pi F\tau} d\tau \\ &= \frac{1}{2T_0} \left| \int_{-T_0}^{T_0} x(t) e^{-j2\pi Ft} dt \right|^2 \end{aligned} \tag{14.1.22}$$

The actual power density spectrum is the expected value of $P_{xx}(F)$ in the limit as $T_0 \to \infty$,

$$\begin{aligned} \Gamma_{xx}(F) &= \lim_{T_0 \to \infty} E[P_{xx}(F)] \\ &= \lim_{T_0 \to \infty} E\left[\frac{1}{2T_0} \left| \int_{-T_0}^{T_0} x(t) e^{-j2\pi Ft} dt \right|^2 \right] \end{aligned} \tag{14.1.23}$$

From (14.1.20) and (14.1.22) we again note the two possible approaches to computing $P_{xx}(F)$, the direct method as given by (14.1.22) or the indirect method, in which we obtain $R_{xx}(\tau)$ first and then compute the Fourier transform.

We shall consider the estimation of the power density spectrum from samples of a single realization of the random process. In particular, we assume that $x_a(t)$ is sampled at a rate $F_s > 2B$, where B is the highest frequency contained in the power density spectrum of the random process. Thus we obtain a finite-duration

sequence $x(n)$, $0 \le n \le N-1$, by sampling $x_a(t)$. From these samples we compute the time-average autocorrelation sequence

$$r'_{xx}(m) = \frac{1}{N-m} \sum_{n=0}^{N-m-1} x^*(n)x(n+m), \qquad m = 0, 1, \ldots, N-1 \tag{14.1.24}$$

and for negative values of m we have $r'_{xx}(m) = [r'_{xx}(-m)]^*$. Then, we compute the Fourier transform

$$P'_{xx}(f) = \sum_{m=-N+1}^{N-1} r'_{xx}(m)e^{-j2\pi fm} \tag{14.1.25}$$

The normalization factor $N-m$ in (14.1.24) results in an estimate with mean value

$$\begin{aligned} E[r'_{xx}(m)] &= \frac{1}{N-m} \sum_{n=0}^{N-m-1} E[x^*(n)x(n+m)] \\ &= \gamma_{xx}(m) \end{aligned} \tag{14.1.26}$$

where $\gamma_{xx}(m)$ is the true (statistical) autocorrelation sequence of $x(n)$. Hence $r'_{xx}(m)$ is an unbiased estimate of the autocorrelation function $\gamma_{xx}(m)$. The variance of the estimate $r'_{xx}(m)$ is approximately

$$\text{var}[r'_{xx}(m)] \approx \frac{N}{[N-m]^2} \sum_{n=-\infty}^{\infty} \left[|\gamma_{xx}(n)|^2 + \gamma^*_{xx}(n-m)\gamma_{xx}(n+m)\right] \tag{14.1.27}$$

which is a result given by Jenkins and Watts (1968). Clearly,

$$\lim_{N\to\infty} \text{var}[r'_{xx}(m)] = 0 \tag{14.1.28}$$

provided that

$$\sum_{n=-\infty}^{\infty} |\gamma_{xx}(n)|^2 < \infty$$

Since $E[r'_{xx}(m)] = \gamma_{xx}(m)$ and the variance of the estimate converges to zero as $N \to \infty$, the estimate $r'_{xx}(m)$ is said to be *consistent.*

For large values of the lag parameter m, the estimate $r'_{xx}(m)$ given by (14.1.24) has a large variance, especially as m approaches N. This is due to the fact that fewer data points enter into the estimate for large lags. As an alternative to (14.1.24) we can use the estimate

$$\begin{aligned} r_{xx}(m) &= \frac{1}{N} \sum_{n=0}^{N-m-1} x^*(n)x(n+m), \qquad 0 \le m \le N-1 \\ r_{xx}(m) &= \frac{1}{N} \sum_{n=|m|}^{N-1} x^*(n)x(n+m), \qquad m = -1, -2, \ldots, 1-N \end{aligned} \tag{14.1.29}$$

which has a bias of $|m|\gamma_{xx}(m)/N$, since its mean value is

$$\begin{aligned} E[r_{xx}(m)] &= \frac{1}{N}\sum_{n=0}^{N-m-1} E[x^*(n)x(n+m)] \\ &= \frac{N-|m|}{N}\gamma_{xx}(m) = \left(1-\frac{|m|}{N}\right)\gamma_{xx}(m) \end{aligned} \tag{14.1.30}$$

However, this estimate has a smaller variance, given approximately as

$$\text{var}[r_{xx}(m)] \approx \frac{1}{N}\sum_{n=-\infty}^{\infty}[\gamma_{xx}(n)|^2 + \gamma_{xx}^*(n-m)\gamma_{xx}(n+m)] \tag{14.1.31}$$

We observe that $r_{xx}(m)$ is *asymptotically unbiased*, that is,

$$\lim_{N\to\infty} E[r_{xx}(m)] = \gamma_{xx}(m) \tag{14.1.32}$$

and its variance converges to zero as $N \to \infty$. Therefore, the estimate $r_{xx}(m)$ is also a *consistent estimate* of $\gamma_{xx}(m)$.

We shall use the estimate $r_{xx}(m)$ given by (14.1.29) in our treatment of power spectrum estimation. The corresponding estimate of the power density spectrum is

$$P_{xx}(f) = \sum_{m=-(N-1)}^{N-1} r_{xx}(m)e^{-j2\pi fm} \tag{14.1.33}$$

If we substitute for $r_{xx}(m)$ from (14.1.29) into (14.1.33), the estimate $P_{xx}(f)$ can also be expressed as

$$P_{xx}(f) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi fn}\right|^2 = \frac{1}{N}|X(f)|^2 \tag{14.1.34}$$

where $X(f)$ is the Fourier transform of the sample sequence $x(n)$. This well-known form of the power density spectrum estimate is called the *periodogram*. It was originally introduced by Schuster (1898) to detect and measure "hidden periodicities" in data.

From (14.1.33), the average value of the periodogram estimate $P_{xx}(f)$ is

$$\begin{aligned} E[P_{xx}(f)] &= E\left[\sum_{m=-(N-1)}^{N-1} r_{xx}(m)e^{-j2\pi fm}\right] = \sum_{m=-(N-1)}^{N-1} E[r_{xx}(m)]e^{-j2\pi fm} \\ E[P_{xx}(f)] &= \sum_{m=-(N-1)}^{N-1}\left(1-\frac{|m|}{N}\right)\gamma_{xx}(m)e^{-j2\pi fm} \end{aligned} \tag{14.1.35}$$

The interpretation that we give to (14.1.35) is that the mean of the estimated spectrum is the Fourier transform of the windowed autocorrelation function

$$\tilde{\gamma}_{xx}(m) = \left(1 - \frac{|m|}{N}\right)\gamma_{xx}(m) \tag{14.1.36}$$

where the window function is the (triangular) Bartlett window. Hence the mean of the estimated spectrum is

$$\begin{aligned} E[P_{xx}(f)] &= \sum_{m=-\infty}^{\infty} \tilde{\gamma}_{xx}(m)e^{-j2\pi fm} \\ &= \int_{-1/2}^{1/2} \Gamma_{xx}(\alpha)W_B(f-\alpha)d\alpha \end{aligned} \tag{14.1.37}$$

where $W_B(f)$ is the spectral characteristic of the Bartlett window. The relation (14.1.37) illustrates that the mean of the estimated spectrum is the convolution of the true power density spectrum $\Gamma_{xx}(f)$ with the Fourier transform $W_B(f)$ of the Bartlett window. Consequently, the mean of the estimated spectrum is a smoothed version of the true spectrum and suffers from the same spectral leakage problems, which are due to the finite number of data points.

We observe that the estimated spectrum is asymptotically unbiased, that is,

$$\lim_{N\to\infty} E\left[\sum_{m=-(N-1)}^{N-1} r_{xx}(m)e^{-j2\pi fm}\right] = \sum_{m=-\infty}^{\infty} \gamma_{xx}(m)e^{-j2\pi fm} = \Gamma_{xx}(f)$$

However, in general, the variance of the estimate $P_{xx}(f)$ does not decay to zero as $N \to \infty$. For example, when the data sequence is a Gaussian random process, the variance is easily shown to be (see Problem 14.4)

$$\text{var}[P_{xx}(f)] = \Gamma_{xx}^2(f)\left[1 + \left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2\right] \tag{14.1.38}$$

which, in the limit as $N \to \infty$, becomes

$$\lim_{N\to\infty} \text{var}[P_{xx}(f)] = \Gamma_{xx}^2(f) \tag{14.1.39}$$

Hence we conclude that the *periodogram is not a consistent estimate of the true power density spectrum* (i.e., it does not converge to the true power density spectrum).

In summary, the estimated autocorrelation $r_{xx}(m)$ is a consistent estimate of the true autocorrelation function $\gamma_{xx}(m)$. However, its Fourier transform $P_{xx}(f)$, the periodogram, is not a consistent estimate of the true power density spectrum. We observed that $P_{xx}(f)$ is an asymptotically unbiased estimate of $\Gamma_{xx}(f)$, but for a finite-duration sequence, the mean value of $P_{xx}(f)$ contains a bias, which from (14.1.37) is evident as a distortion of the true power density spectrum. Thus the

estimated spectrum suffers from the smoothing effects and the leakage embodied in the Bartlett window. The smoothing and leakage ultimately limit our ability to resolve closely spaced spectra.

The problems of leakage and frequency resolution that we have just described, as well as the problem that the periodogram is not a consistent estimate of the power spectrum, provide the motivation for the power spectrum estimation methods described in Sections 14.2, 14.3, and 14.4. The methods described in Section 14.2 are classical nonparametric methods, which make no assumptions about the data sequence. The emphasis of the classical methods is on obtaining a consistent estimate of the power spectrum through some averaging or smoothing operations performed directly on the periodogram or on the autocorrelation. As we will see, the effect of these operations is to reduce the frequency resolution further, while the variance of the estimate is decreased.

The spectrum estimation methods described in Section 14.3 are based on some model of how the data were generated. In general, the model-based methods that have been developed over the past few decades provide significantly higher resolution than do the classical methods.

Additional methods are described in Sections 14.4 and 14.5. In Section 14.4, we consider filter bank methods for the estimation of the power spectrum. The methods described in Section 14.5 are based on an eigenvalue/eigenvector decomposition of the data correlation matrix.

14.1.3 The Use of the DFT in Power Spectrum Estimation

As given by (14.1.14) and (14.1.34), the estimated energy density spectrum $S_{xx}(f)$ and the periodogram $P_{xx}(f)$, respectively, can be computed by use of the DFT, which in turn is efficiently computed by an FFT algorithm. If we have N data points, we compute as a minimum the N-point DFT. For example, the computation yields samples of the periodogram

$$P_{xx}\left(\frac{k}{N}\right) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}\right|^2, \qquad k = 0, 1, \ldots, N-1 \tag{14.1.40}$$

at the frequencies $f_k = k/N$.

In practice, however, such a sparse sampling of the spectrum does not provide a very good representation or a good picture of the continuous spectrum estimate $P_{xx}(f)$. This is easily remedied by evaluating $P_{xx}(f)$ at additional frequencies. Equivalently, we can effectively increase the length of the sequence by means of zero padding and then evaluate $P_{xx}(f)$ at a more dense set of frequencies. Thus if we increase the data sequence length to L points by means of zero padding and evaluate the L-point DFT, we have

$$P_{xx}\left(\frac{k}{L}\right) = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/L}\right|^2, \qquad k = 0, 1, \ldots, L-1 \tag{14.1.41}$$

We emphasize that zero padding and evaluating the DFT at $L > N$ points does not improve the frequency resolution in the spectral estimate. It simply provides us with a method for interpolating the values of the measured spectrum at more frequencies. The frequency resolution in the spectral estimate $P_{xx}(f)$ is determined by the length N of the data record.

EXAMPLE 14.1.2

A sequence of $N = 16$ samples is obtained by sampling an analog signal consisting of two frequency components. The resulting discrete-time sequence is

$$x(n) = \sin 2\pi(0.135)n + \cos 2\pi(0.135 + \Delta f)n, \qquad n = 0, 1, \ldots, 15$$

where Δf is the frequency separation. Evaluate the power spectrum $P(f) = (1/N)|X(f)|^2$ at the frequencies $f_k = k/L$, $k = 0, 1, \ldots, L - 1$, for $L = 8$, 16, 32, and 128 for values of $\Delta f = 0.06$ and $\Delta f = 0.01$.

Solution. By zero padding, we increase the data sequence to obtain the power spectrum estimate $P_{xx}(k/L)$. The results for $\Delta f = 0.06$ are plotted in Fig. 14.1.4. Note that zero padding

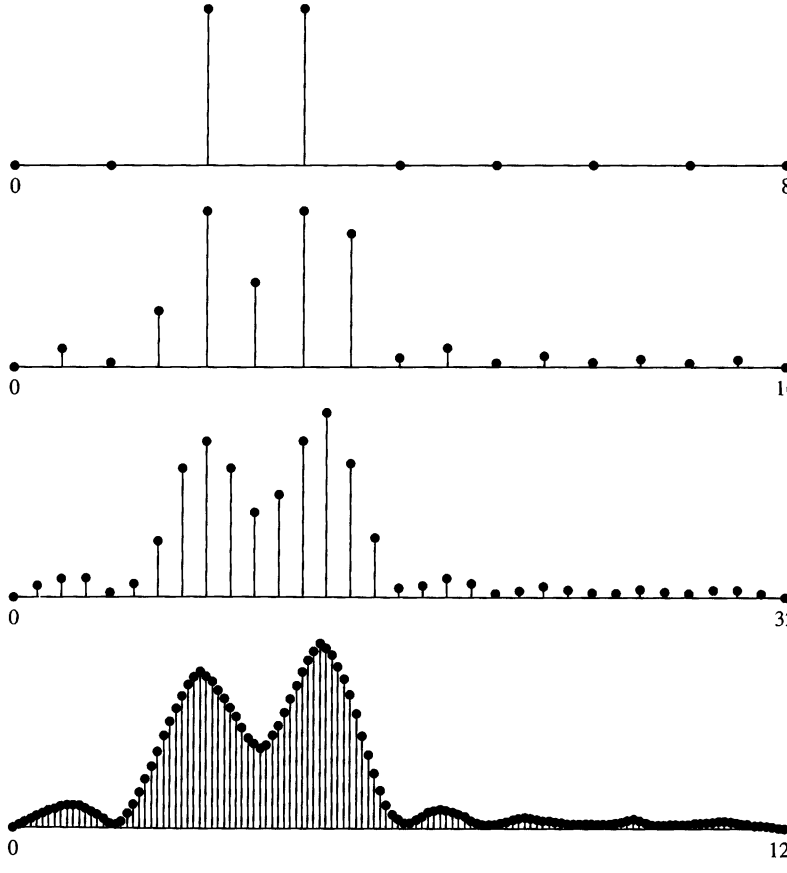


Figure 14.1.4 Spectra of two sinusoids with frequency separation $\Delta f = 0.06$.

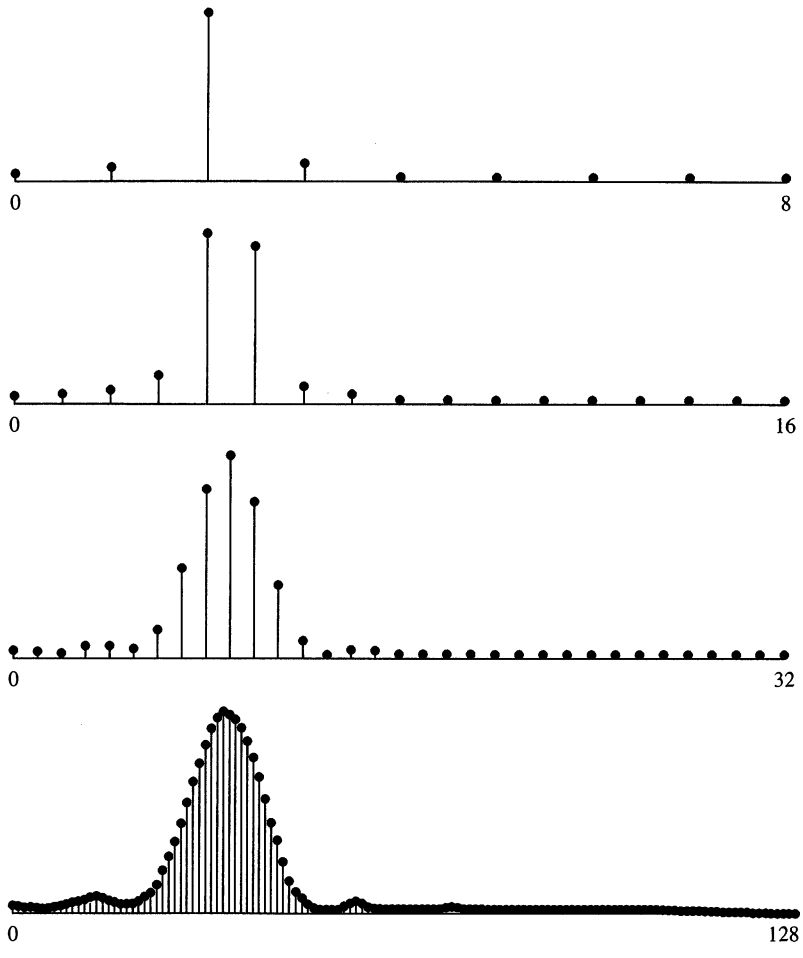


Figure 14.1.5 Spectra of two sinusoids with frequency separation $\Delta f = 0.01$.

does not change the resolution, but it does have the effect of interpolating the spectrum $P_{xx}(f)$. In this case the frequency separation Δf is sufficiently large so that the two frequency components are resolvable.

The spectral estimates for $\Delta f = 0.01$ are shown in Fig. 14.1.5. In this case the two spectral components are not resolvable. Again, the effect of zero padding is to provide more interpolation, thus giving us a better picture of the estimated spectrum. It does not improve the frequency resolution.

When only a few points of the periodogram are needed, the Goertzel algorithm described in Chapter 8 may provide a more efficient computation. Since the Goertzel algorithm has been interpreted as a linear filtering approach to computing the DFT, it is clear that the periodogram estimate can be obtained by passing the signal through a bank of parallel tuned filters and squaring their outputs (see Problem 14.5).

14.2 Nonparametric Methods for Power Spectrum Estimation

The power spectrum estimation methods described in this section are the classical methods developed by Bartlett (1948), Blackman and Tukey (1958), and Welch (1967). These methods make no assumption about how the data were generated and hence are called *nonparametric*.

Since the estimates are based entirely on a finite record of data, the frequency resolution of these methods is, at best, equal to the spectral width of the rectangular window of length N, which is approximately $1/N$ at the -3-dB points. We shall be more precise in specifying the frequency resolution of the specific methods. All the estimation techniques described in this section decrease the frequency resolution in order to reduce the variance in the spectral estimate.

First, we describe the estimates and derive the mean and variance of each. A comparison of the three methods is given in Section 14.2.4. Although the spectral estimates are expressed as a function of the continuous frequency variable f, in practice, the estimates are computed at discrete frequencies via the FFT algorithm. The FFT-based computational requirements are considered in Section 14.2.5.

14.2.1 The Bartlett Method: Averaging Periodograms

Bartlett's method for reducing the variance in the periodogram involves three steps. First, the N-point sequence is subdivided into K nonoverlapping segments, where each segment has length M. This results in the K data segments

$$x_i(n) = x(n + iM), \qquad \begin{matrix} i = 0, 1, \ldots, K-1 \\ n = 0, 1, \ldots, M-1 \end{matrix} \tag{14.2.1}$$

For each segment, we compute the periodogram

$$P_{xx}^{(i)}(f) = \frac{1}{M}\left|\sum_{n=0}^{M-1} x_i(n)e^{-j2\pi fn}\right|^2, \qquad i = 0, 1, \ldots, K-1 \tag{14.2.2}$$

Finally, we average the periodograms for the K segments to obtain the Bartlett power spectrum estimate [Bartlett (1948)]

$$P_{xx}^{B}(f) = \frac{1}{K}\sum_{i=0}^{K-1} P_{xx}^{(i)}(f) \tag{14.2.3}$$

The statistical properties of this estimate are easily obtained. First, the mean value is

$$\begin{aligned} E[P_{xx}^{B}(f)] &= \frac{1}{K}\sum_{i=0}^{K-1} E[P_{xx}^{(i)}(f)] \\ &= E[P_{xx}^{(i)}(f)] \end{aligned} \tag{14.2.4}$$

From (14.1.35) and (14.1.37) we have the expected value for the single periodogram as

$$\begin{aligned} E[P_{xx}^{(i)}(f)] &= \sum_{m=-(M-1)}^{M-1} \left(1 - \frac{|m|}{M}\right) \gamma_{xx}(m) e^{-j2\pi fm} \\ &= \frac{1}{M} \int_{-1/2}^{1/2} \Gamma_{xx}(\alpha) \left(\frac{\sin \pi(f-\alpha)M}{\sin \pi(f-\alpha)}\right)^2 d\alpha \end{aligned} \tag{14.2.5}$$

where

$$W_B(f) = \frac{1}{M} \left(\frac{\sin \pi f M}{\sin \pi f}\right)^2 \tag{14.2.6}$$

is the frequency characteristic of the Bartlett window

$$w_B(n) = \begin{cases} 1 - \dfrac{|m|}{M}, & |m| \le M-1 \\ 0, & \text{otherwise} \end{cases} \tag{14.2.7}$$

From (14.2.5) we observe that the true spectrum is now convolved with the frequency characteristic $W_B(f)$ of the Bartlett window. The effect of reducing the length of the data from N points to $M = N/K$ results in a window whose spectral width has been increased by a factor of K. Consequently, the frequency resolution has been reduced by a factor K.

In return for this reduction in resolution, we have reduced the variance. The variance of the Bartlett estimate is

$$\begin{aligned} \text{var}[P_{xx}^B(f)] &= \frac{1}{K^2} \sum_{i=0}^{K-1} \text{var}[P_{xx}^{(i)}(f)] \\ &= \frac{1}{K} \text{var}[P_{xx}^{(i)}(f)] \end{aligned} \tag{14.2.8}$$

If we make use of (14.1.38) in (14.2.8), we obtain

$$\text{var}[P_{xx}^B(f)] = \frac{1}{K} \Gamma_{xx}^2(f) \left[1 + \left(\frac{\sin 2\pi f M}{M \sin 2\pi f}\right)^2\right] \tag{14.2.9}$$

Therefore, the variance of the Bartlett power spectrum estimate has been reduced by the factor K.

14.2.2 The Welch Method: Averaging Modified Periodograms

Welch (1967) made two basic modifications to the Bartlett method. First, he allowed the data segments to overlap. Thus the data segments can be represented as

$$x_i(n) = x(n + iD), \qquad \begin{matrix} n = 0, 1, \ldots, M-1 \\ i = 0, 1, \ldots, L-1 \end{matrix} \tag{14.2.10}$$

where iD is the starting point for the ith sequence. Observe that if $D = M$, the segments do not overlap and the number L of data segments is identical to the number K in the Bartlett method. However, if $D = M/2$, there is 50% overlap between successive data segments and $L = 2K$ segments are obtained. Alternatively, we can form K data segments each of length $2M$.

The second modification made by Welch to the Bartlett method is to window the data segments prior to computing the periodogram. The result is a "modified" periodogram

$$\tilde{P}_{xx}^{(i)}(f) = \frac{1}{MU}\left|\sum_{n=0}^{M-1} x_i(n)w(n)e^{-j2\pi fn}\right|^2, \qquad i = 0, 1, \ldots, L-1 \tag{14.2.11}$$

where U is a normalization factor for the power in the window function and is selected as

$$U = \frac{1}{M}\sum_{n=0}^{M-1} w^2(n) \tag{14.2.12}$$

The Welch power spectrum estimate is the average of these modified periodograms, that is,

$$P_{xx}^{W}(f) = \frac{1}{L}\sum_{i=0}^{L-1} \tilde{P}_{xx}^{(i)}(f) \tag{14.2.13}$$

The mean value of the Welch estimate is

$$\begin{aligned} E[P_{xx}^{W}(f)] &= \frac{1}{L}\sum_{i=0}^{L-1} E[\tilde{P}_{xx}^{(i)}(f)] \\ &= E[\tilde{P}_{xx}^{(i)}(f)] \end{aligned} \tag{14.2.14}$$

But the expected value of the modified periodogram is

$$\begin{aligned} E[\tilde{P}_{xx}^{(i)}(f)] &= \frac{1}{MU}\sum_{n=0}^{M-1}\sum_{m=0}^{M-1} w(n)w(m)E[x_i(n)x_i^*(m)]e^{-j2\pi f(n-m)} \\ &= \frac{1}{MU}\sum_{n=0}^{M-1}\sum_{m=0}^{M-1} w(n)w(m)\gamma_{xx}(n-m)e^{-j2\pi f(n-m)} \end{aligned} \tag{14.2.15}$$

Since

$$\gamma_{xx}(n) = \int_{-1/2}^{1/2} \Gamma_{xx}(\alpha)e^{j2\pi\alpha n}\,d\alpha \tag{14.2.16}$$

substitution for $\gamma_{xx}(n)$ from (14.2.16) into (14.2.15) yields

$$\begin{aligned} E[\tilde{P}_{xx}^{(i)}(f)] &= \frac{1}{MU}\int_{-1/2}^{1/2}\Gamma_{xx}(\alpha)\left[\sum_{n=0}^{M-1}\sum_{m=0}^{M-1}w(n)w(m)e^{-j2\pi(n-m)(f-\alpha)}\right]d\alpha \\ &= \int_{-1/2}^{1/2}\Gamma_{xx}(\alpha)W(f-\alpha)d\alpha \end{aligned} \tag{14.2.17}$$

where, by definition,

$$W(f) = \frac{1}{MU}\left|\sum_{n=0}^{M-1}w(n)e^{-j2\pi fn}\right|^2 \tag{14.2.18}$$

The normalization factor U ensures that

$$\int_{-1/2}^{1/2}W(f)df = 1 \tag{14.2.19}$$

The variance of the Welch estimate is

$$\text{var}[P_{xx}^{W}(f)] = \frac{1}{L^2}\sum_{i=0}^{L-1}\sum_{j=0}^{L-1}E[\tilde{P}_{xx}^{(i)}(f)\tilde{P}_{xx}^{(j)}(f)] - \{E[P_{xx}^{W}(f)]\}^2 \tag{14.2.20}$$

In the case of no overlap between successive data segments ($L = K$), Welch has shown that

$$\begin{aligned} \text{var}[P_{xx}^{W}(f)] &= \frac{1}{L}\text{var}[\tilde{P}_{xx}^{(i)}(f)] \\ &\approx \frac{1}{L}\Gamma_{xx}^2(f) \end{aligned} \tag{14.2.21}$$

In the case of 50% overlap between successive data segments ($L = 2K$), the variance of the Welch power spectrum estimate with the Bartlett (triangular) window, also derived in the paper by Welch, is

$$\text{var}[P_{xx}^{W}(f)] \approx \frac{9}{8L}\Gamma_{xx}^2(f) \tag{14.2.22}$$

Although we considered only the triangular window in the computation of the variance, other window functions may be used. In general, they will yield a different variance. In addition, one may vary the data segment overlapping by either more or less than the 50% considered in this section in an attempt to improve the relevant characteristics of the estimate.

14.2.3 The Blackman and Tukey Method: Smoothing the Periodogram

Blackman and Tukey (1958) proposed and analyzed the method in which the sample autocorrelation sequence is windowed first and then Fourier transformed to yield the estimate of the power spectrum. The rationale for windowing the estimated autocorrelation sequence $r_{xx}(m)$ is that, for large lags, the estimates are less reliable because a smaller number $(N-m)$ of data points enter into the estimate. For values of m approaching N, the variance of these estimates is very high, and hence these estimates should be given a smaller weight in the formation of the estimated power spectrum. Thus the Blackman–Tukey estimate is

$$P_{xx}^{\mathrm{BT}}(f) = \sum_{m=-(M-1)}^{M-1} r_{xx}(m)w(m)e^{-j2\pi fm} \tag{14.2.23}$$

where the window function $w(n)$ has length $2M-1$ and is zero for $|m| \geq M$. With this definition for $w(n)$, the limits on the sum in (14.2.23) can be extended to $(-\infty, \infty)$. Hence the frequency-domain equivalent expression for (14.2.23) is the convolution integral

$$P_{xx}^{\mathrm{BT}}(f) = \int_{-1/2}^{1/2} P_{xx}(\alpha)W(f-\alpha)d\alpha \tag{14.2.24}$$

where $P_{xx}(f)$ is the periodogram. It is clear from (14.2.24) that the effect of windowing the autocorrelation is to smooth the periodogram estimate, thus decreasing the variance in the estimate at the expense of reducing the resolution.

The window sequence $w(n)$ should be symmetric (even) about $m=0$ to ensure that the estimate of the power spectrum is real. Furthermore, it is desirable to select the window spectrum to be nonnegative, that is,

$$W(f) \geq 0, \qquad |f| \leq 1/2 \tag{14.2.25}$$

This condition ensures that $P_{xx}^{\mathrm{BT}}(f) \geq 0$ for $|f| \leq 1/2$, which is a desirable property for any power spectrum estimate. We should indicate, however, that some of the window functions we have introduced do not satisfy this condition. For example, in spite of their low sidelobe levels, the Hamming and Hann (or Hanning) windows do not satisfy the property in (14.2.25) and, consequently, may result in negative spectrum estimates in some parts of the frequency range.

The expected value of the Blackman–Tukey power spectrum estimate is

$$E[P_{xx}^{\mathrm{BT}}(f)] = \int_{-1/2}^{1/2} E[P_{xx}(\alpha)]W(f-\alpha)d\alpha \tag{14.2.26}$$

where from (14.1.37) we have

$$E[P_{xx}(\alpha)] = \int_{-1/2}^{1/2} \Gamma_{xx}(\theta)W_B(\alpha-\theta)d\theta \tag{14.2.27}$$

and $W_B(f)$ is the Fourier transform of the Bartlett window. Substitution of (14.2.27) into (14.2.26) yields the double convolution integral

$$E[P_{xx}^{\mathrm{BT}}(f)] = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} \Gamma_{xx}(\theta) W_B(\alpha - \theta) W(f - \alpha) d\alpha d\theta \tag{14.2.28}$$

Equivalently, by working in the time domain, the expected value of the Blackman–Tukey power spectrum estimate is

$$\begin{aligned} E[P_{xx}^{\mathrm{BT}}(f)] &= \sum_{m=-(M-1)}^{M-1} E[r_{xx}(m)] w(m) e^{-j2\pi fm} \\ &= \sum_{m=-(M-1)}^{M-1} \gamma_{xx}(m) w_B(m) w(m) e^{-j2\pi fm} \end{aligned} \tag{14.2.29}$$

where the Bartlett window is

$$w_B(m) = \begin{cases} 1 - \dfrac{|m|}{N}, & |m| < N \\ 0, & \text{otherwise} \end{cases} \tag{14.2.30}$$

Clearly, we should select the window length for $w(n)$ such that $M << N$, that is, $w(n)$ should be narrower than $w_B(m)$ to provide additional smoothing of the periodogram. Under this condition, (14.2.28) becomes

$$E[P_{xx}^{\mathrm{BT}}(f)] \approx \int_{-1/2}^{1/2} \Gamma_{xx}(\theta) W(f - \theta) d\theta \tag{14.2.31}$$

since

$$\begin{aligned} \int_{-1/2}^{1/2} W_B(\alpha - \theta) W(f - \alpha) d\alpha &= \int_{-1/2}^{1/2} W_B(\alpha) W(f - \theta - \alpha) d\alpha \\ &\approx W(f - \theta) \end{aligned} \tag{14.2.32}$$

The variance of the Blackman–Tukey power spectrum estimate is

$$\mathrm{var}[P_{xx}^{\mathrm{BT}}(f)] = E\{[P_{xx}^{\mathrm{BT}}(f)]^2\} - \{E[P_{xx}^{\mathrm{BT}}(f)]\}^2 \tag{14.2.33}$$

where the mean can be approximated as in (14.2.31). The second moment in (14.2.33) is

$$E\{[P_{xx}^{\mathrm{BT}}(f)]^2\} = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} E[P_{xx}(\alpha) P_{xx}(\theta)] W(f - \alpha) W(f - \theta) d\alpha d\theta \tag{14.2.34}$$

On the assumption that the random process is Gaussian (see Problem 14.5), we find that

$$E[P_{xx}(\alpha)P_{xx}(\theta)] = \Gamma_{xx}(\alpha)\Gamma_{xx}(\theta)\left\{1 + \left[\frac{\sin\pi(\theta+\alpha)N}{N\sin\pi(\theta+\alpha)}\right]^2 + \left[\frac{\sin\pi(\theta-\alpha)N}{N\sin\pi(\theta-\alpha)}\right]^2\right\} \tag{14.2.35}$$

Substitution of (14.2.35) into (14.2.34) yields

$$\begin{aligned} E\{[P_{xx}^{\mathrm{BT}}(f)]^2\} = &\left[\int_{-1/2}^{1/2}\Gamma_{xx}(\theta)W(f-\theta)d\theta\right]^2 \\ &+ \int_{-1/2}^{1/2}\int_{-1/2}^{1/2}\Gamma_{xx}(\alpha)\Gamma_{xx}(\theta)W(f-\alpha)W(f-\theta) \\ &\times\left\{\left[\frac{\sin\pi(\theta+\alpha)N}{N\sin\pi(\theta+\alpha)}\right]^2 + \left[\frac{\sin\pi(\theta-\alpha)N}{N\sin\pi(\theta-\alpha)}\right]^2\right\} d\alpha\, d\theta \end{aligned} \tag{14.2.36}$$

The first term in (14.2.36) is simply the square of the mean of $P_{xx}^{\mathrm{BT}}(f)$, which is to be subtracted out according to (14.2.33). This leaves the second term in (14.2.36), which constitutes the variance. For the case in which $N \gg M$, the functions $\sin\pi(\theta+\alpha)N/N\sin\pi(\theta+\alpha)$ and $\sin\pi(\theta-\alpha)N/N\sin\pi(\theta-\alpha)$ are relatively narrow compared to $W(f)$ in the vicinity of $\theta = -\alpha$ and $\theta = \alpha$, respectively. Therefore,

$$\begin{aligned} &\int_{-1/2}^{1/2}\Gamma_{xx}(\theta)W(f-\theta)\left\{\left[\frac{\sin\pi(\theta+\alpha)N}{N\sin\pi(\theta+\alpha)}\right]^2 + \left[\frac{\sin\pi(\theta-\alpha)N}{N\sin\pi(\theta-\alpha)}\right]^2\right\} d\theta \\ &\qquad \approx \frac{\Gamma_{xx}(-\alpha)W(f+\alpha) + \Gamma_{xx}(\alpha)W(f-\alpha)}{N} \end{aligned} \tag{14.2.37}$$

With this approximation, the variance of $P_{xx}^{\mathrm{BT}}(f)$ becomes

$$\begin{aligned} \mathrm{var}[P_{xx}^{\mathrm{BT}}(f)] &\approx \frac{1}{N}\int_{-1/2}^{1/2}\Gamma_{xx}(\alpha)W(f-\alpha)[\Gamma_{xx}(-\alpha)W(f+\alpha) + \Gamma_{xx}(\alpha)W(f-\alpha)]d\alpha \\ &\approx \frac{1}{N}\int_{-1/2}^{1/2}\Gamma_{xx}^2(\alpha)W^2(f-\alpha)d\alpha \end{aligned} \tag{14.2.38}$$

where in the last step we made the approximation

$$\int_{-1/2}^{1/2}\Gamma_{xx}(\alpha)\Gamma_{xx}(-\alpha)W(f-\alpha)W(f+\alpha)d\alpha \approx 0 \tag{14.2.39}$$

We shall make one additional approximation in (14.2.38). When $W(f)$ is narrow compared to the true power spectrum $\Gamma_{xx}(f)$, (14.2.38) is further approximated as

$$\begin{aligned} \text{var}[P_{xx}^{\text{BT}}(f)] &\approx \Gamma_{xx}^2(f)\left[\frac{1}{N}\int_{-1/2}^{1/2} W^2(\theta)d\theta\right] \\ &\approx \Gamma_{xx}^2(f)\left[\frac{1}{N}\sum_{m=-(M-1)}^{M-1} w^2(m)\right] \end{aligned} \tag{14.2.40}$$

14.2.4 Performance Characteristics of Nonparametric Power Spectrum Estimators

In this section we compare the quality of the Bartlett, Welch, and Blackman and Tukey power spectrum estimates. As a measure of quality, we use the ratio of its variance to the square of the mean of the power spectrum estimate, that is,

$$Q_A = \frac{\{E[P_{xx}^A(f)]\}^2}{var[P_{xx}^A(f)]} \tag{14.2.41}$$

where $A = B$, W, or BT for the three power spectrum estimates. The reciprocal of this quantity, called the *variability*, can also be used as a measure of performance.

For reference, the periodogram has a mean and variance

$$E[P_{xx}(f)] = \int_{-1/2}^{1/2} \Gamma_{xx}(\theta)W_B(f-\theta)d\theta \tag{14.2.42}$$

$$\text{var}[P_{xx}(f)] = \Gamma_{xx}^2(f)\left[1 + \left(\frac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2\right] \tag{14.2.43}$$

where

$$W_B(f) = \frac{1}{N}\left(\frac{\sin \pi fN}{\sin \pi f}\right)^2 \tag{14.2.44}$$

For large N (i.e., $N \to \infty$),

$$\begin{aligned} &E[P_{xx}(f)] \to \Gamma_{xx}(f), \quad \int_{-1/2}^{1/2} W_B(\theta)d\theta = w_B(0)\Gamma_{xx}(f) = \Gamma_{xx}(f) \\ &\text{var}[P_{xx}(f)] \to \Gamma_{xx}^2(f) \end{aligned} \tag{14.2.45}$$

Hence, as indicated previously, the periodogram is an asymptotically unbiased estimate of the power spectrum, but it is not consistent because its variance does not approach zero as N increases toward infinity.

Asymptotically, the periodogram is characterized by the quality factor

$$Q_P = \frac{\Gamma_{xx}^2(f)}{\Gamma_{xx}^2(f)} = 1 \tag{14.2.46}$$

The fact that Q_P is fixed and independent of the data length N is another indication of the poor quality of this estimate.

Bartlett power spectrum estimate. The mean and variance of the Bartlett power spectrum estimate are

$$E[P_{xx}^B(f)] = \int_{-1/2}^{1/2} \Gamma_{xx}(\theta) W_B(f-\theta) d\theta \tag{14.2.47}$$

$$\text{var}[P_{xx}^B(f)] = \frac{1}{K}\Gamma_{xx}^2(f)\left[1 + \left(\frac{\sin 2\pi f M}{M \sin 2\pi f}\right)^2\right] \tag{14.2.48}$$

and

$$W_B(f) = \frac{1}{M}\left(\frac{\sin \pi f M}{\sin \pi f}\right)^2 \tag{14.2.49}$$

As $N \to \infty$ and $M \to \infty$, while $K = N/M$ remains fixed, we find that

$$\begin{aligned} E[P_{xx}^B(f)] &\to \Gamma_{xx}(f), \quad \int_{-1/2}^{1/2} W_B(f) df = \Gamma_{xx}(f) w_B(0) = \Gamma_{xx}(f) \\ \text{var}[P_{xx}^B(f)] &\to \frac{1}{K}\Gamma_{xx}^2(f) \end{aligned} \tag{14.2.50}$$

We observe that the Bartlett power spectrum estimate is asymptotically unbiased and if K is allowed to increase with an increase in N, the estimate is also consistent. Hence, asymptotically, this estimate is characterized by the quality factor

$$Q_B = K = \frac{N}{M} \tag{14.2.51}$$

The frequency resolution of the Bartlett estimate, measured by taking the 3-dB width of the main lobe of the rectangular window, is

$$\Delta f = \frac{0.9}{M} \tag{14.2.52}$$

Hence, $M = 0.9/\Delta f$ and the quality factor becomes

$$Q_B = \frac{N}{0.9/\Delta f} = 1.1 N \Delta f \tag{14.2.53}$$

Welch power spectrum estimate. The mean and variance of the Welch power spectrum estimate are

$$E[P_{xx}^W(f)] = \int_{-1/2}^{1/2} \Gamma_{xx}(\theta) W(f-\theta) d\theta \tag{14.2.54}$$

where

$$W(f) = \frac{1}{MU}\left|\sum_{n=0}^{M-1} w(n) e^{-j2\pi f n}\right|^2 \tag{14.2.55}$$

and

$$\text{var}[P_{xx}^{W}(f)] = \begin{cases} \dfrac{1}{L}\Gamma_{xx}^2(f), & \text{for no overlap} \\ \dfrac{9}{8L}\Gamma_{xx}^2(f), & \text{for 50\% overlap and triangular window} \end{cases} \tag{14.2.56}$$

As $N \to \infty$ and $M \to \infty$, the mean converges to

$$E[P_{xx}^{W}(f)] \to \Gamma_{xx}(f) \tag{14.2.57}$$

and the variance converges to zero, so that the estimate is consistent.

Under the two conditions given by (14.2.56) the quality factor is

$$Q_W = \begin{cases} L = \dfrac{N}{M}, & \text{for no overlap} \\ \dfrac{8L}{9} = \dfrac{16N}{9M}, & \text{for 50\% overlap and triangular window} \end{cases} \tag{14.2.58}$$

On the other hand, the spectral width of the triangular window at the 3-dB points is

$$\Delta f = \frac{1.28}{M} \tag{14.2.59}$$

Consequently, the quality factor expressed in terms of Δf and N is

$$Q_W = \begin{cases} 0.78N\Delta f, & \text{for no overlap} \\ 1.39N\Delta f, & \text{for 50\% overlap and triangular window} \end{cases} \tag{14.2.60}$$

Blackman–Tukey power spectrum estimate. The mean and variance of this estimate are approximated as

$$\begin{aligned} E[P_{xx}^{\text{BT}}(f)] &\approx \int_{-1/2}^{1/2} \Gamma_{xx}(\theta) W(f-\theta)\, d\theta \\ \text{var}[P_{xx}^{\text{BT}}(f)] &\approx \Gamma_{xx}^2(f) \left[\frac{1}{N} \sum_{m=-(M-1)}^{M-1} w^2(m) \right] \end{aligned} \tag{14.2.61}$$

where $w(m)$ is the window sequence used to taper the estimated autocorrelation sequence. For the rectangular and Bartlett (triangular) windows we have

$$\frac{1}{N} \sum_{n=-(M-1)}^{M-1} w^2(n) = \begin{cases} 2M/N, & \text{rectangular window} \\ 2M/3N, & \text{triangular window} \end{cases} \tag{14.2.62}$$

It is clear from (14.2.61) that the mean value of the estimate is asymptotically unbiased. Its quality factor for the triangular window is

$$Q_{\text{BT}} = 1.5\frac{N}{M} \tag{14.2.63}$$

TABLE 14.1 Quality of Power Spectrum Estimates

Estimate	Quality Factor
Bartlett	$1.11N\Delta f$
Welch (50% overlap)	$1.39N\Delta f$
Blackman–Tukey	$2.34N\Delta f$

Since the window length is $2M - 1$, the frequency resolution measured at the 3-dB points is

$$\Delta f = \frac{1.28}{2M} = \frac{0.64}{M} \tag{14.2.64}$$

and hence

$$Q_{\text{BT}} = \frac{1.5}{0.64} N\Delta f = 2.34N\Delta f \tag{14.2.65}$$

These results are summarized in Table 14.1. It is apparent from the results we have obtained that the Welch and Blackman–Tukey power spectrum estimates are somewhat better than the Bartlett estimate. However, the differences in performance are relatively small. The main point is that the quality factor increases with an increase in the length N of the data. This characteristic behavior is not shared by the periodogram estimate. Furthermore, the quality factor depends on the product of the data length N and the frequency resolution Δf. For a desired level of quality, Δf can be decreased (frequency resolution increased) by increasing the length N of the data, and vice versa.

14.2.5 Computational Requirements of Nonparametric Power Spectrum Estimates

The other important aspect of the nonparametric power spectrum estimates is their computational requirements. For this comparison we assume the estimates are based on a fixed amount of data N and a specified resolution Δf. The radix-2 FFT algorithm is assumed in all the computations. We shall count only the number of complex multiplications required to compute the power spectrum estimate.

Bartlett power spectrum estimate.

$$\text{FFT length} = M = 0.9/\Delta f$$

$$\text{Number of FFTs} = \frac{N}{M} = 1.11N\Delta f$$

$$\text{Number of computations} = \frac{N}{M}\left(\frac{M}{2}\log_2 M\right) = \frac{N}{2}\log_2 \frac{0.9}{\Delta f}$$

Welch power spectrum estimate (50% overlap).

$$\text{FFT length} = M = 1.28/\Delta f$$

$$\text{Number of FFTs} = \frac{2N}{M} = 1.56N\Delta f$$

$$\text{Number of computations} = \frac{2N}{M}\left(\frac{M}{2}\log_2 M\right) = N\log_2\frac{1.28}{\Delta f}$$

In addition to the $2N/M$ FFTs, there are additional multiplications required for windowing the data. Each data record requires M multiplications. Therefore, the total number of computations is

$$\text{Total computations} = 2N + N\log_2\frac{1.28}{\Delta f} = N\log_2\frac{5.12}{\Delta f}$$

Blackman–Tukey power spectrum estimate. In the Blackman–Tukey method, the autocorrelation $r_{xx}(m)$ can be computed efficiently via the FFT algorithm. However, if the number of data points is large, it may not be possible to compute one N-point DFT. For example, we may have $N = 10^5$ data points but only the capacity to perform 1024-point DFTs. Since the autocorrelation sequence is windowed to $2M - 1$ points where $M \ll N$, it is possible to compute the desired $2M - 1$ points of $r_{xx}(m)$ by segmenting the data into $K = N/2M$ records and then computing $2M$-point DFTs and one $2M$-point IDFT via the FFT algorithm. Rader (1970) has described a method for performing this computation (see Problem 14.7).

If we base the computational complexity of the Blackman–Tukey method on this approach, we obtain the following computational requirements.

$$\text{FFT length} = 2M = 1.28/\Delta f$$

$$\text{Number of FFTs} = 2K + 1 = 2\left(\frac{N}{2M}\right) + 1 \approx \frac{N}{M}$$

$$\text{Number of computations} = \frac{N}{M}(M\log_2 2M) = N\log_2\frac{1.28}{\Delta f}$$

We can neglect the additional M multiplications required to window the autocorrelation sequence $r_{xx}(m)$, since it is a relatively small number. Finally, there is the additional computation required to perform the Fourier transform of the windowed autocorrelation sequence. The FFT algorithm can be used for this computation with some zero padding for purposes of interpolating the spectral estimate. As a result of these additional computations, the number of computations is increased by a small amount.

From these results we conclude that the Welch method requires a little more computational power than do the other two methods. The Bartlett method apparently requires the smallest number of computations. However, the differences in the computational requirements of the three methods are relatively small.

14.3 Parametric Methods for Power Spectrum Estimation

The nonparametric power spectrum estimation methods described in the preceding section are relatively simple, well understood, and easy to compute using the FFT algorithm. However, these methods require the availability of long data records in order to obtain the necessary frequency resolution required in many applications. Furthermore, these methods suffer from spectral leakage effects, due to windowing, that are inherent in finite-length data records. Often, the spectral leakage masks weak signals that are present in the data.

From one point of view, the basic limitation of the nonparametric methods is the inherent assumption that the autocorrelation estimate $r_{xx}(m)$ is zero for $m \geq N$, as implied by (14.1.33). This assumption severely limits the frequency resolution and the quality of the power spectrum estimate that is achieved. From another viewpoint, the inherent assumption in the periodogram estimate is that the data are periodic with period N. Neither one of these assumptions is realistic.

In this section we describe power spectrum estimation methods that do not require such assumptions. In fact, these methods *extrapolate* the values of the autocorrelation for lags $m \geq N$. Extrapolation is possible if we have some *a priori* information on how the data were generated. In such a case a model for the signal generation can be constructed with a number of parameters that can be estimated from the observed data. From the model and the estimated parameters, we can compute the power density spectrum implied by the model.

In effect, the modeling approach eliminates the need for window functions and the assumption that the autocorrelation sequence is zero for $|m| \geq N$. As a consequence, *parametric* (model-based) power spectrum estimation methods avoid the problem of leakage and provide better frequency resolution than do the FFT-based, nonparametric methods described in the preceding section. This is especially true in applications where short data records are available due to time-variant or transient phenomena.

The parametric methods considered in this section are based on modeling the data sequence $x(n)$ as the output of a linear system characterized by a rational system function of the form

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}} \tag{14.3.1}$$

The corresponding difference equation is

$$x(n) = -\sum_{k=1}^{p} a_k x(n-k) + \sum_{k=0}^{q} b_k w(n-k) \tag{14.3.2}$$

where $w(n)$ is the input sequence to the system and the observed data, $x(n)$, represent the output sequence.

In power spectrum estimation, the input sequence is not observable. However, if the observed data are characterized as a stationary random process, then the input sequence is also assumed to be a stationary random process. In such a case the power density spectrum of the data is

$$\Gamma_{xx}(f) = |H(f)|^2 \Gamma_{ww}(f)$$

where $\Gamma_{ww}(f)$ is the power density spectrum of the input sequence and $H(f)$ is the frequency response of the model.

Since our objective is to estimate the power density spectrum $\Gamma_{xx}(f)$, it is convenient to assume that the input sequence $w(n)$ is a zero-mean white noise sequence with autocorrelation

$$\gamma_{ww}(m) = \sigma_w^2 \delta(m)$$

where σ_w^2 is the variance (i.e., $\sigma_w^2 = E[|w(n)|^2]$). Then the power density spectrum of the observed data is simply

$$\Gamma_{xx}(f) = \sigma_w^2 |H(f)|^2 = \sigma_w^2 \frac{|B(f)|^2}{|A(f)|^2} \tag{14.3.3}$$

In Section 12.2 we described the representation of a stationary random process as given by (14.3.3).

In the model-based approach, the spectrum estimation procedure consists of two steps. Given the data sequence $x(n), 0 \leq n \leq N-1$, we estimate the parameters $\{a_k\}$ and $\{b_k\}$ of the model. Then from these estimates, we compute the power spectrum estimate according to (14.3.3).

Recall that the random process $x(n)$ generated by the pole–zero model in (14.3.1) or (14.3.2) is called an *autoregressive moving average* (ARMA) *process* of order (p, q) and it is usually denoted as ARMA (p, q). If $q = 0$ and $b_0 = 1$, the resulting system model has a system function $H(z) = 1/A(z)$ and its output $x(n)$ is called an *autoregressive* (AR) *process* of order p. This is denoted as AR(p). The third possible model is obtained by setting $A(z) = 1$, so that $H(z) = B(z)$. Its output $x(n)$ is called a *moving average* (MA) *process* of order q and denoted as MA(q).

Of these three linear models the AR model is by far the most widely used. The reasons are twofold. First, the AR model is suitable for representing spectra with narrow peaks (resonances). Second, the AR model results in very simple linear equations for the AR parameters. On the other hand, the MA model, as a general rule, requires many more coefficients to represent a narrow spectrum. Consequently, it is rarely used by itself as a model for spectrum estimation. By combining poles and zeros, the ARMA model provides a more efficient representation, from the viewpoint of the number of model parameters, of the spectrum of a random process.

The decomposition theorem due to Wold (1938) asserts that any ARMA or MA process can be represented uniquely by an AR model of possibly infinite order, and any ARMA or AR process can be represented by an MA model of possibly infinite order. In view of this theorem, the issue of model selection reduces to selecting the model that requires the smallest number of parameters that are also easy to compute.

Usually, the choice in practice is the AR model. The ARMA model is used to a lesser extent.

Before describing methods for estimating the parameters in AR(p), MA(q), and ARMA(p, q) models, it is useful to establish the basic relationships between the model parameters and the autocorrelation sequence $\gamma_{xx}(m)$. In addition, we relate the AR model parameters to the coefficients in a linear predictor for the process $x(n)$.

14.3.1 Relationships Between the Autocorrelation and the Model Parameters

In Section 12.2.2 we established the basic relationships between the autocorrelation $\{\gamma_{xx}(m)\}$ and the model parameters $\{a_k\}$ and $\{b_k\}$. For the ARMA(p, q) process, the relationship given by (12.2.18) is

$$\gamma_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k), & m > q \\ -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k) + \sigma_w^2 \sum_{k=0}^{q-m} h(k) b_{k+m}, & 0 \le m \le q \\ \gamma_{xx}^*(-m), & m < 0 \end{cases} \tag{14.3.4}$$

The relationships in (14.3.4) provide a formula for determining the model parameters $\{a_k\}$ by restricting our attention to the case $m > q$. Thus the set of linear equations

$$\begin{bmatrix} \gamma_{xx}(q) & \gamma_{xx}(q-1) & \cdots & \gamma_{xx}(q-p+1) \\ \gamma_{xx}(q+1) & \gamma_{xx}(q) & \cdots & \gamma_{xx}(q+p+2) \\ \vdots & \vdots & & \\ \gamma_{xx}(q+p-1) & \gamma_{xx}(q+p-2) & \cdots & \gamma_{xx}(q) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} \gamma_{xx}(q+1) \\ \gamma_{xx}(q+2) \\ \vdots \\ \gamma_{xx}(q+p) \end{bmatrix} \tag{14.3.5}$$

can be used to solve for the model parameters $\{a_k\}$ by using estimates of the autocorrelation sequence in place of $\gamma_{xx}(m)$ for $m \ge q$. This problem is discussed in Section 14.3.8.

Another interpretation of the relationship in (14.3.5) is that the values of the autocorrelation $\gamma_{xx}(m)$ for $m > q$ are uniquely determined from the pole parameters $\{a_k\}$ and the values of $\gamma_{xx}(m)$ for $0 \le m \le p$. Consequently, the linear system model automatically extends the values of the autocorrelation sequence $\gamma_{xx}(m)$ for $m > p$.

If the pole parameters $\{a_k\}$ are obtained from (14.3.5), the result does not help us in determining the MA parameters $\{b_k\}$, because the equation

$$\sigma_w^2 \sum_{k=0}^{q-m} h(k) b_{k+m} = \gamma_{xx}(m) + \sum_{k=1}^{p} a_k \gamma_{xx}(m-k), \qquad 0 \le m \le q$$

depends on the impulse response $h(n)$. Although the impulse response can be expressed in terms of the parameters $\{b_k\}$ by long division of $B(z)$ with the known $A(z)$, this approach results in a set of nonlinear equations for the MA parameters.

If we adopt an AR(p) model for the observed data, the relationship between the AR parameters and the autocorrelation sequence is obtained by setting $q = 0$ in (14.3.4). Thus we obtain

$$\gamma_{xx}(m) = \begin{cases} -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k), & m > 0 \\ -\sum_{k=1}^{p} a_k \gamma_{xx}(m-k) + \sigma_w^2, & m = 0 \\ \gamma_{xx}^*(-m), & m < 0 \end{cases} \tag{14.3.6}$$

In this case, the AR parameters $\{a_k\}$ are obtained from the solution of the Yule–Walker or normal equations

$$\begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(-1) & \vdots & \gamma_{xx}(-p+1) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \vdots & \gamma_{xx}(-p+2) \\ \cdots & \cdots & & \vdots \\ \gamma_{xx}(p-1) & \gamma_{xx}(p-2) & \vdots & \gamma_{xx}(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = - \begin{bmatrix} \gamma_{xx}(1) \\ \gamma_{xx}(2) \\ \vdots \\ \gamma_{xx}(p) \end{bmatrix} \tag{14.3.7}$$

and the variance σ_w^2 can be obtained from the equation

$$\sigma_w^2 = \gamma_{xx}(0) + \sum_{k=1}^{p} a_k \gamma_{xx}(-k) \tag{14.3.8}$$

The equations in (14.3.7) and (14.3.8) are usually combined into a single matrix equation of the form

$$\begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(-1) & \cdots & \gamma_{xx}(-p) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \cdots & \gamma_{xx}(-p+1) \\ \vdots & \vdots & & \vdots \\ \gamma_{xx}(p) & \gamma_{xx}(p-1) & \cdots & \gamma_{xx}(0) \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{14.3.9}$$

Since the correlation matrix in (14.3.7), or in (14.3.9), is Toeplitz, it can be efficiently inverted by use of the Levinson–Durbin algorithm.

Thus all the system parameters in the AR(p) model are easily determined from knowledge of the autocorrelation sequence $\gamma_{xx}(m)$ for $0 \le m \le p$. Furthermore, (14.3.6) can be used to extend the autocorrelation sequence for $m > p$, once the $\{a_k\}$ are determined.

Finally, for completeness, we indicate that in an MA(q) model for the observed data, the autocorrelation sequence $\gamma_{xx}(m)$ is related to the MA parameters $\{b_k\}$ by the equation

$$\gamma_{xx}(m) = \begin{cases} \sigma_w^2 \sum_{k=0}^{q} b_k b_{k+m}, & 0 \le m \le q \\ 0, & m > q \\ \gamma_{xx}^*(-m), & m < 0 \end{cases} \tag{14.3.10}$$

which was established in Section 12.2.

With this background established, we now describe the power spectrum estimation methods for the AR(p), ARMA(p, q), and MA(q) models.

14.3.2 The Yule–Walker Method for the AR Model Parameters

In the Yule–Walker method we simply estimate the autocorrelation from the data and use the estimates in (14.3.7) to solve for the AR model parameters. In this method it is desirable to use the biased form of the autocorrelation estimate,

$$r_{xx}(m) = \frac{1}{N} \sum_{n=0}^{N-m-1} x^*(n)x(n+m), \qquad m \ge 0 \tag{14.3.11}$$

to ensure that the autocorrelation matrix is positive semidefinite. The result is a stable AR model. Although stability is not a critical issue in power spectrum estimation, it is conjectured that a stable AR model best represents the data.

The Levinson–Durbin algorithm described in Chapter 12 with $r_{xx}(m)$ substituted for $\gamma_{xx}(m)$ yields the AR parameters. The corresponding power spectrum estimate is

$$P_{xx}^{\text{YW}}(f) = \frac{\hat{\sigma}_{wp}^2}{|1 + \sum_{k=1}^{p} \hat{a}_p(k) e^{-j2\pi fk}|^2} \tag{14.3.12}$$

where $\hat{a}_p(k)$ are estimates of the AR parameters obtained from the Levinson–Durbin recursions and

$$\hat{\sigma}_{wp}^2 = \hat{E}_p^f = r_{xx}(0) \prod_{k=1}^{p} [1 - |\hat{a}_k(k)|^2] \tag{14.3.13}$$

is the estimated minimum mean-square value for the pth-order predictor. An example illustrating the frequency resolution capabilities of this estimator is given in Section 14.3.9.

In estimating the power spectrum of sinusoidal signals via AR models, Lacoss (1971) showed that spectral peaks in an AR spectrum estimate are proportional to the square of the power of the sinusoidal signal. On the other hand, the area under the peak in the power density spectrum is linearly proportional to the power of the sinusoid. This characteristic behavior holds for all AR model-based estimation methods.

14.3.3 The Burg Method for the AR Model Parameters

The method devised by Burg (1968) for estimating the AR parameters can be viewed as an order-recursive least-squares lattice method, based on the minimization of the forward and backward errors in linear predictors, with the constraint that the AR parameters satisfy the Levinson–Durbin recursion.

To derive the estimator, suppose that we are given the data $x(n)$, $n = 0, 1, \ldots, N-1$, and let us consider the forward and backward linear prediction estimates of order m, given as

$$\begin{aligned} \hat{x}(n) &= -\sum_{k=1}^{m} a_m(k)x(n-k) \\ \hat{x}(n-m) &= -\sum_{k=1}^{m} a_m^*(k)x(n+k-m) \end{aligned} \tag{14.3.14}$$

and the corresponding forward and backward errors $f_m(n)$ and $g_m(n)$ defined as $f_m(n) = x(n) - \hat{x}(n)$ and $g_m(n) = x(n-m) - \hat{x}(n-m)$ where $a_m(k)$, $0 \le k \le m-1$, $m = 1, 2, \ldots, p$, are the prediction coefficients. The least-squares error is

$$\mathcal{E}_m = \sum_{n=m}^{N-1} [|f_m(n)|^2 + |g_m(n)|^2] \tag{14.3.15}$$

This error is to be minimized by selecting the prediction coefficients, subject to the constraint that they satisfy the Levinson–Durbin recursion given by

$$a_m(k) = a_{m-1}(k) + K_m a_{m-1}^*(m-k), \qquad \begin{matrix} 1 \le k \le m-1 \\ 1 \le m \le p \end{matrix} \tag{14.3.16}$$

where $K_m = a_m(m)$ is the mth reflection coefficient in the lattice filter realization of the predictor. When (14.3.16) is substituted into the expressions for $f_m(n)$ and $g_m(n)$, the result is the pair of order-recursive equations for the forward and backward prediction errors given by (12.3.4).

Now, if we substitute from (12.3.4) into (14.3.16) and perform the minimization of $\mathcal{E}_m$ with respect to the complex-valued reflection coefficient K_m, we obtain the result

$$\hat{K}_m = \frac{-\sum_{n=m}^{N-1} f_{m-1}(n) g_{m-1}^*(n-1)}{\frac{1}{2}\sum_{n=m}^{N-1} [|f_{m-1}(n)|^2 + |g_{m-1}(n-1)|^2]}, \qquad m = 1, 2, \ldots, p \tag{14.3.17}$$

The term in the numerator of (14.3.17) is an estimate of the crosscorrelation between the forward and backward prediction errors. With the normalization factors in the denominator of (14.3.17), it is apparent that $|K_m| < 1$, so that the all-pole model

obtained from the data is stable. The reader should note the similarity of (14.3.17) to its statistical counterparts given by (12.3.28).

We note that the denominator in (14.3.17) is simply the least-squares estimate of the forward and backward errors, E_{m-1}^f and E_{m-1}^b, respectively. Hence (14.3.17) can be expressed as

$$\hat{K}_m = \frac{-\sum_{n=m}^{N-1} f_{m-1}(n) g_{m-1}^*(n-1)}{\frac{1}{2}[\hat{E}_{m-1}^f + \hat{E}_{m-1}^b]}, \qquad m = 1, 2, \ldots, p \tag{14.3.18}$$

where $\hat{E}_{m-1}^f + \hat{E}_{m-1}^b$ is an estimate of the total squared error E_m. We leave it as an exercise for the reader to verify that the denominator term in (14.3.18) can be computed in an order-recursive fashion according to the relation

$$\hat{E}_m = (1 - |\hat{K}_m|^2)\hat{E}_{m-1} - |f_{m-1}(m-1)|^2 - |g_{m-1}(m-2)|^2 \tag{14.3.19}$$

where $\hat{E}_m \equiv \hat{E}_m^f + \hat{E}_m^b$ is the total least-squares error. This result is due to Andersen (1978).

To summarize, the Burg algorithm computes the reflection coefficients in the equivalent lattice structure as specified by (14.3.18) and (14.3.19), and the Levinson–Durbin algorithm is used to obtain the AR model parameters. From the estimates of the AR parameters, we form the power spectrum estimate

$$P_{xx}^{\text{BU}}(f) = \frac{\hat{E}_p}{\left|1 + \sum_{k=1}^{p} \hat{a}_p(k) e^{-j2\pi fk}\right|^2} \tag{14.3.20}$$

The major advantages of the Burg method for estimating the parameters of the AR model are (1) it results in high frequency resolution, (2) it yields a stable AR model, and (3) it is computationally efficient.

The Burg method is known to have several disadvantages, however. First, it exhibits spectral line splitting at high signal-to-noise ratios [see the paper by Fougere et al. (1976)]. By line splitting, we mean that the spectrum of $x(n)$ may have a single sharp peak, but the Burg method may result in two or more closely spaced peaks. For high-order models, the method also introduces spurious peaks. Furthermore, for sinusoidal signals in noise, the Burg method exhibits a sensitivity to the initial phase of a sinusoid, especially in short data records. This sensitivity is manifest as a frequency shift from the true frequency, resulting in a phase-dependent frequency bias. For more details on some of these limitations the reader is referred to the papers of Chen and Stegen (1974), Ulrych and Clayton (1976), Fougere et al. (1976), Kay and Marple (1979), Swingler (1979a, 1980), Herring (1980), and Thorvaldsen (1981).

Several modifications have been proposed to overcome some of the more important limitations of the Burg method: namely, the line splitting, spurious peaks,

and frequency bias. Basically, the modifications involve the introduction of a weighting (window) sequence on the squared forward and backward errors. That is, the least-squares optimization is performed on the weighted squared errors

$$\mathcal{E}_m^{\text{WB}} = \sum_{n=m}^{N-1} w_m(n)[|f_m(n)|^2 + |g_m(n)|^2] \tag{14.3.21}$$

which, when minimized, results in the reflection coefficient estimates

$$\hat{K}_m = \frac{-\sum_{n=m}^{N-1} w_{m-1}(n) f_{m-1}(n) g_{m-1}^*(n-1)}{\frac{1}{2}\sum_{n=m}^{N-1} w_{m-1}(n)[|f_{m-1}(n)|^2 + |g_{m-1}(n-1)|^2]} \tag{14.3.22}$$

In particular, we mention the use of a Hamming window used by Swingler (1979b), a quadratic or parabolic window used by Kaveh and Lippert (1983), the energy weighting method used by Nikias and Scott (1982), and the data-adaptive energy weighting used by Helme and Nikias (1985).

These windowing and energy weighting methods have proved effective in reducing the occurrence of line splitting and spurious peaks, and are also effective in reducing frequency bias.

The Burg method for power spectrum estimation is usually associated with *maximum entropy spectrum estimation*, a criterion used by Burg (1967, 1975) as a basis for AR modeling in parametric spectrum estimation. The problem considered by Burg was how best to extrapolate from the given values of the autocorrelation sequence $\gamma_{xx}(m)$, $0 \leq m \leq p$, the values for $m > p$, such that the entire autocorrelation sequence is positive semidefinite. Since an infinite number of extrapolations are possible, Burg postulated that the extrapolations be made on the basis of maximizing uncertainty (entropy) or randomness, in the sense that the spectrum $\Gamma_{xx}(f)$ of the process is the flattest of all spectra which have the given autocorrelation values $\gamma_{xx}(m)$, $0 \leq m \leq p$. In particular the entropy per sample is proportional to the integral [see Burg (1975)]

$$\int_{-1/2}^{1/2} \ln \Gamma_{xx}(f) df \tag{14.3.23}$$

Burg found that the maximum of this integral subject to the $(p+1)$ constraints

$$\int_{-1/2}^{1/2} \Gamma_{xx}(f) e^{j2\pi fm} df = \gamma_{xx}(m), \qquad 0 \leq m \leq p \tag{14.3.24}$$

is the AR(p) process for which the given autocorrelation sequence $\gamma_{xx}(m)$, $0 \leq m \leq p$ is related to the AR parameters by the equation (14.3.6). This solution provides an additional justification for the use of the AR model in power spectrum estimation.

In view of Burg's basic work in maximum entropy spectral estimation, the Burg power spectrum estimation procedure is often called the *maximum entropy method*

(MEM). We should emphasize, however, that the maximum entropy spectrum is identical to the AR-model spectrum only when the exact autocorrelation $\gamma_{xx}(m)$ is known. When only an estimate of $\gamma_{xx}(m)$ is available for $0 \leq m \leq p$, the AR-model estimates of Yule–Walker and Burg are not maximum entropy spectral estimates. The general formulation for the maximum entropy spectrum based on estimates of the autocorrelation sequence results in a set of nonlinear equations. Solutions for the maximum entropy spectrum with measurement errors in the correlation sequence have been obtained by Newman (1981) and Schott and McClellan (1984).

14.3.4 Unconstrained Least-Squares Method for the AR Model Parameters

As described in the preceding section, the Burg method for determining the parameters of the AR model is basically a least-squares lattice algorithm with the added constraint that the predictor coefficients satisfy the Levinson recursion. As a result of this constraint, an increase in the order of the AR model requires only a single parameter optimization at each stage. In contrast to this approach, we may use an unconstrained least-squares algorithm to determine the AR parameters.

To elaborate, we form the forward and backward linear prediction estimates and their corresponding forward and backward errors as indicated in (14.3.14) and (14.3.15). Then we minimize the sum of squares of both errors, that is,

$$\begin{aligned} \mathcal{E}_p &= \sum_{n=p}^{N-1} [|f_p(n)|^2 + |g_p(n)|^2] \\ &= \sum_{n=p}^{N-1} \left[\left| x(n) + \sum_{k=1}^{p} a_p(k) x(n-k) \right|^2 \right. \\ &\quad \left. + \left| x(n-p) + \sum_{k=1}^{p} a_p^*(k) x(n+k-p) \right|^2 \right] \end{aligned} \tag{14.3.25}$$

which is the same performance index as in the Burg method. However, we do not impose the Levinson–Durbin recursion in (14.3.25) for the AR parameters. The unconstrained minimization of $\mathcal{E}_p$ with respect to the prediction coefficients yields the set of linear equations

$$\sum_{k=1}^{p} a_p(k) r_{xx}(l,k) = -r_{xx}(l,0), \qquad l = 1, 2, \ldots, p \tag{14.3.26}$$

where, by definition, the autocorrelation $r_{xx}(l,k)$ is

$$r_{xx}(l,k) = \sum_{n=p}^{N-1} [x(n-k)x^*(n-l) + x(n-p+l)x^*(n-p+k)] \tag{14.3.27}$$

The resulting residual least-squares error is

$$\mathcal{E}_p^{\text{LS}} = r_{xx}(0, 0) + \sum_{k=1}^{p} \hat{a}_p(k) r_{xx}(0, k) \tag{14.3.28}$$

Hence the unconstrained least-squares power spectrum estimate is

$$P_{xx}^{\text{LS}}(f) = \frac{\mathcal{E}_p^{\text{LS}}}{\left| 1 + \sum_{k=1}^{p} \hat{a}_p(k) e^{-j2\pi fk} \right|^2} \tag{14.3.29}$$

The correlation matrix in (14.3.27), with elements $r_{xx}(l, k)$, is not Toeplitz, so that the Levinson–Durbin algorithm cannot be applied. However, the correlation matrix has sufficient structure to make it possible to devise computationally efficient algorithms with computational complexity proportional to p^2. Marple (1980) devised such an algorithm, which has a lattice structure and employs Levinson–Durbin-type order recursions and additional time recursions.

This form of the unconstrained least-squares method described has also been called the *unwindowed data* least-squares method. It has been proposed for spectrum estimation in several papers, including the papers by Burg (1967), Nuttall (1976), and Ulrych and Clayton (1976). Its performance characteristics have been found to be superior to the Burg method, in the sense that the unconstrained least-squares method does not exhibit the same sensitivity to such problems as line splitting, frequency bias, and spurious peaks. In view of the computational efficiency of Marple's algorithm, which is comparable to the efficiency of the Levinson–Durbin algorithm, the unconstrained least-squares method is very attractive. With this method there is no guarantee that the estimated AR parameters yield a stable AR model. However, in spectrum estimation, this is not considered to be a problem.

14.3.5 Sequential Estimation Methods for the AR Model Parameters

The three power spectrum estimation methods described in the preceding sections for the AR model can be classified as block processing methods. These methods obtain estimates of the AR parameters from a block of data, say $x(n)$, $n = 0, 1, \ldots, N-1$. The AR parameters, based on the block of N data points, are then used to obtain the power spectrum estimate.

In situations where data are available on a continuous basis, we can still segment the data into blocks of N points and perform spectrum estimation on a block-by-block basis. This is often done in practice, for both real-time and non-real-time applications. However, in such applications, there is an alternative approach based on sequential (in time) estimation of the AR model parameters as each new data point becomes available. By introducing a weighting function into past data samples, it is possible to deemphasize the effect of older data samples as new data are received.

Sequential lattice methods based on recursive least squares can be employed to optimally estimate the prediction and reflection coefficients in the lattice realization

of the forward and backward linear predictors. The recursive equations for the prediction coefficients relate directly to the AR model parameters. In addition to the order-recursive nature of these equations, as implied by the lattice structure, we can also obtain time-recursive equations for the reflection coefficients in the lattice and for the forward and backward prediction coefficients.

Sequential recursive least-squares algorithms are equivalent to the unconstrained least-squares, block processing method described in the preceding section. Hence the power spectrum estimates obtained by the sequential recursive least-squares method retain the desirable properties of the block processing algorithm described in Section 14.3.4. Since the AR parameters are being continuously estimated in a sequential estimation algorithm, power spectrum estimates can be obtained as often as desired, from once per sample to once every N samples. By properly weighting past data samples, the sequential estimation methods are particularly suitable for estimating and tracking time-variant power spectra resulting from nonstationary signal statistics.

The computational complexity of sequential estimation methods is generally proportional to p, the order of the AR process. As a consequence, sequential estimation algorithms are computationally efficient and, from this viewpoint, may offer some advantage over the block processing methods.

There are numerous references on sequential estimation methods. The papers by Griffiths (1975), Friedlander (1982a, b), and Kalouptsidis and Theodoridis (1987) are particularly relevant to the spectrum estimation problem.

14.3.6 Selection of AR Model Order

One of the most important aspects of the use of the AR model is the selection of the order p. As a general rule, if we select a model with too low an order, we obtain a highly smoothed spectrum. On the other hand, if p is selected too high, we run the risk of introducing spurious low-level peaks in the spectrum. We mentioned previously that one indication of the performance of the AR model is the mean-square value of the residual error, which, in general, is different for each of the estimators described above. The characteristic of this residual error is that it decreases as the order of the AR model is increased. We can monitor the rate of decrease and decide to terminate the process when the rate of decrease becomes relatively slow. It is apparent, however, that this approach may be imprecise and ill defined, and other methods should be investigated.

Much work has been done by various researchers on this problem and many experimental results have been given in the literature [e.g., the papers by Gersch and Sharpe (1973), Ulrych and Bishop (1975), Tong (1975, 1977), Jones (1976), Nuttall (1976), Berryman (1978), Kaveh and Bruzzone (1979), and Kashyap (1980)].

Two of the better-known criteria for selecting the model order have been proposed by Akaike (1969, 1974). With the first, called the *final prediction error (FPE) criterion*, the order is selected to minimize the performance index

$$\text{FPE}(p) = \hat{\sigma}_{wp}^2 \left(\frac{N + p + 1}{N - p - 1} \right) \tag{14.3.30}$$

where $\hat{\sigma}_{wp}^2$ is the estimated variance of the linear prediction error. This performance index is based on minimizing the mean-square error for a one-step predictor.

The second criterion proposed by Akaike (1974), called the *Akaike information criterion* (AIC), is based on selecting the order that minimizes

$$\text{AIC}(p) = \ln \hat{\sigma}_{wp}^2 + 2p/N \tag{14.3.31}$$

Note that the term $\hat{\sigma}_{wp}^2$ decreases and therefore $\ln \hat{\sigma}_{wp}^2$ also decreases as the order of the AR model is increased. However, $2p/N$ increases with an increase in p. Hence a minimum value is obtained for some p.

An alternative information criterion, proposed by Rissanen (1983), is based on selecting the order that *minimizes the description length* (MDL), where MDL is defined as

$$\text{MDL}(p) = N \ln \hat{\sigma}_{wp}^2 + p \ln N \tag{14.3.32}$$

A fourth criterion has been proposed by Parzen (1974). This is called the *criterion autoregressive transfer* (CAT) function and is defined as

$$\text{CAT}(p) = \left(\frac{1}{N} \sum_{k=1}^{p} \frac{1}{\overline{\sigma}_{wk}^2} \right) - \frac{1}{\hat{\sigma}_{wp}^2} \tag{14.3.33}$$

where

$$\overline{\sigma}_{wk}^2 = \frac{N}{N-k} \hat{\sigma}_{wk}^2 \tag{14.3.34}$$

The order p is selected to minimize $\text{CAT}(p)$.

In applying this criterion, the mean should be removed from the data. Since $\hat{\sigma}_{wk}^2$ depends on the type of spectrum estimate we obtain, the model order is also a function of the criterion.

The experimental results given in the references just cited indicate that the model-order selection criteria do not yield definitive results. For example, Ulrych and Bishop (1975), Jones (1976), and Berryman (1978) found that the $\text{FPE}(p)$ criterion tends to underestimate the model order. Kashyap (1980) showed that the AIC criterion is statistically inconsistent as $N \to \infty$. On the other hand, the MDL information criterion proposed by Rissanen is statistically consistent. Other experimental results indicate that for small data lengths, the order of the AR model should be selected to be in the range $N/3$ to $N/2$ for good results. It is apparent that in the absence of any prior information regarding the physical process that resulted in the data, one should try different model orders and different criteria and, finally, consider the different results.

14.3.7 MA Model for Power Spectrum Estimation

As shown in Section 14.3.1, the parameters in an $\text{MA}(q)$ model are related to the statistical autocorrelation $\gamma_{xx}(m)$ by (14.3.10). However,

$$B(z)B(z^{-1}) = D(z) = \sum_{m=-q}^{q} d_m z^{-m} \tag{14.3.35}$$

where the coefficients $\{d_m\}$ are related to the MA parameters by the expression

$$d_m = \sum_{k=0}^{q-|m|} b_k b_{k+m}, \qquad |m| \le q \tag{14.3.36}$$

Clearly, then,

$$\gamma_{xx}(m) = \begin{cases} \sigma_w^2 d_m, & |m| \le q \\ 0, & |m| > q \end{cases} \tag{14.3.37}$$

and the power spectrum for the MA(q) process is

$$\Gamma_{xx}^{\text{MA}}(f) = \sum_{m=-q}^{q} \gamma_{xx}(m) e^{-j2\pi fm} \tag{14.3.38}$$

It is apparent from these expressions that we do not have to solve for the MA parameters $\{b_k\}$ to estimate the power spectrum. The estimates of the autocorrelation $\gamma_{xx}(m)$ for $|m| \le q$ suffice. From such estimates we compute the estimated MA power spectrum, given as

$$P_{xx}^{\text{MA}}(f) = \sum_{m=-q}^{q} r_{xx}(m) e^{-j2\pi fm} \tag{14.3.39}$$

which is identical to the classical (nonparametric) power spectrum estimate described in Section 14.1.

There is an alternative method for determining $\{b_k\}$ based on a high-order AR approximation to the MA process. To be specific, let the MA(q) process be modeled by an AR(p) model, where $p >> q$. Then $B(z) = 1/A(z)$, or equivalently, $B(z)A(z) = 1$. Thus the parameters $\{b_k\}$ and $\{a_k\}$ are related by a convolution sum, which can be expressed as

$$\hat{a}_n + \sum_{k=1}^{q} b_k \hat{a}_{n-k} = \begin{cases} 1, & n = 0 \\ 0, & n \ne 0 \end{cases} \tag{14.3.40}$$

where $\{\hat{a}_n\}$ are the parameters obtained by fitting the data to an AR(p) model.

Although this set of equations can be easily solved for the $\{b_k\}$, a better fit is obtained by using a least-squares error criterion. That is, we form the squared error

$$\mathcal{E} = \sum_{n=0}^{p} \left[\hat{a}_n + \sum_{k=1}^{q} b_k \hat{a}_{n-k} \right]^2 - 1, \qquad \hat{a}_0 = 1, \quad \hat{a}_k = 0, \quad k < 0 \tag{14.3.41}$$

which is minimized by selecting the MA(q) parameters $\{b_k\}$. The result of this minimization is

$$\hat{\mathbf{b}} = -\mathbf{R}_{aa}^{-1} \mathbf{r}_{aa} \tag{14.3.42}$$

where the elements of $\mathbf{R}_{aa}$ and $\mathbf{r}_{aa}$ are given as

$$\begin{aligned} R_{aa}(|i-j|) &= \sum_{n=0}^{p-|i-j|} \hat{a}_n \hat{a}_{n+|i-j|}, \qquad i, j = 1, 2, \ldots, q \\ r_{aa}(i) &= \sum_{n=0}^{p-i} \hat{a}_n \hat{a}_{n+i}, \qquad i = 1, 2, \ldots, q \end{aligned} \tag{14.3.43}$$

This least-squares method for determining the parameters of the MA(q) model is attributed to Durbin (1959). It has been shown by Kay (1988) that this estimation method is approximately maximum likelihood under the assumption that the observed process is Gaussian.

The order q of the MA model may be determined empirically by several methods. For example, the AIC for MA models has the same form as for AR models,

$$\text{AIC}(q) = \ln \sigma_{wq}^2 + \frac{2q}{N} \tag{14.3.44}$$

where σ_{wq}^2 is an estimate of the variance of the white noise. Another approach, proposed by Chow (1972b), is to filter the data with the inverse MA(q) filter and test the filtered output for whiteness.

14.3.8 ARMA Model for Power Spectrum Estimation

The Burg algorithm, its variations, and the least-squares method described in the previous sections provide reliable high-resolution spectrum estimates based on the AR model. An ARMA model provides us with an opportunity to improve on the AR spectrum estimate, perhaps, by using fewer model parameters.

The ARMA model is particularly appropriate when the signal has been corrupted by noise. For example, suppose that the data $x(n)$ are generated by an AR system, where the system output is corrupted by additive white noise. The z-transform of the autocorrelation of the resultant signal can be expressed as

$$\begin{aligned} \Gamma_{xx}(z) &= \frac{\sigma_w^2}{A(z)A(z^{-1})} + \sigma_n^2 \\ &= \frac{\sigma_w^2 + \sigma_n^2 A(z)A(z^{-1})}{A(z)A(z^{-1})} \end{aligned} \tag{14.3.45}$$

where σ_n^2 is the variance of the additive noise. Therefore, the process $x(n)$ is ARMA(p, p), where p is the order of the autocorrelation process. This relationship provides some motivation for investigating ARMA models for power spectrum estimation.

As we have demonstrated in Section 14.3.1, the parameters of the ARMA model are related to the autocorrelation by the equation in (14.3.4). For lags $|m| > q$, the equation involves only the AR parameters $\{a_k\}$. With estimates substituted in place

of $\gamma_{xx}(m)$, we can solve the p equations in (14.3.5) to obtain $\hat{a}_k$. For high-order models, however, this approach is likely to yield poor estimates of the AR parameters due to the poor estimates of the autocorrelation for large lags. Consequently, this approach is not recommended.

A more reliable method is to construct an overdetermined set of linear equations for $m > q$, and to use the method of least squares on the set of overdetermined equations, as proposed by Cadzow (1979). To elaborate, suppose that the autocorrelation sequence can be accurately estimated up to lag M, where $M > p + q$. Then we can write the following set of linear equations:

$$\begin{bmatrix} r_{xx}(q) & r_{xx}(q-1) & \cdots & r_{xx}(q-p+1) \\ r_{xx}(q+1) & r_{xx}(q) & \cdots & r_{xx}(q-p+2) \\ \vdots & \vdots & & \\ r_{xx}(M-1) & r_{xx}(M-2) & & r_{xx}(M-p) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_{xx}(q+1) \\ r_{xx}(q+2) \\ \vdots \\ r_{xx}(M) \end{bmatrix} \tag{14.3.46}$$

or equivalently,

$$\mathbf{R}_{xx}\mathbf{a} = -\mathbf{r}_{xx} \tag{14.3.47}$$

Since $\mathbf{R}_{xx}$ is of dimension $(M-q) \times p$, and $M - q > p$ we can use the least-squares criterion to solve for the parameter vector $\mathbf{a}$. The result of this minimization is

$$\hat{\mathbf{a}} = -(\mathbf{R}_{xx}^t \mathbf{R}_{xx})^{-1} \mathbf{R}_{xx}^t \mathbf{r}_{xx} \tag{14.3.48}$$

This procedure is called the *least-squares modified Yule–Walker method.* A weighting factor can also be applied to the autocorrelation sequence to deemphasize the less reliable estimates for large lags.

Once the parameters for the AR part of the model have been estimated as indicated above, we have the system

$$\hat{A}(z) = 1 + \sum_{k=1}^{p} \hat{a}_k z^{-k} \tag{14.3.49}$$

The sequence $x(n)$ can now be filtered by the FIR filter $\hat{A}(z)$ to yield the sequence

$$v(n) = x(n) + \sum_{k=1}^{p} \hat{a}_k x(n-k), \qquad n = 0, 1, \ldots, N-1 \tag{14.3.50}$$

The cascade of the ARMA(p, q) model with $\hat{A}(z)$ is approximately the MA(q) process generated by the model $B(z)$. Hence we can apply the MA estimate given in the preceding section to obtain the MA spectrum. To be specific, the filtered sequence

$v(n)$ for $p \leq n \leq N-1$ is used to form the estimated correlation sequences $r_{vv}(m)$, from which we obtain the MA spectrum

$$P_{vv}^{\mathrm{MA}}(f) = \sum_{m=-q}^{q} r_{vv}(m) e^{-j2\pi fm} \tag{14.3.51}$$

First, we observe that the parameters $\{b_k\}$ are not required to determine the power spectrum. Second, we observe that $r_{vv}(m)$ is an estimate of the autocorrelation for the MA model given by (14.3.10). In forming the estimate $r_{vv}(m)$, weighting (e.g., with the Bartlett window) may be used to deemphasize correlation estimates for large lags. In addition, the data may be filtered by a backward filter, thus creating another sequence, say $v^b(n)$, so that both $v(n)$ and $v^b(n)$ can be used in forming the estimate of the autocorrelation $r_{vv}(m)$, as proposed by Kay (1980). Finally, the estimated ARMA power spectrum is

$$\hat{P}_{xx}^{\mathrm{ARMA}}(f) = \frac{P_{vv}^{\mathrm{MA}}(f)}{\left|1 + \sum_{k=1}^{p} \hat{a}_k e^{-j2\pi fk}\right|^2} \tag{14.3.52}$$

The problem of order selection for the ARMA (p, q) model has been investigated by Chow (1972a, b) and Bruzzone and Kaveh (1980). For this purpose the minimum of the AIC index

$$\mathrm{AIC}(p, q) = \ln \hat{\sigma}_{wpq}^2 + \frac{2(p+q)}{N} \tag{14.3.53}$$

can be used, where $\hat{\sigma}_{wpq}^2$ is an estimate of the variance of the input error. An additional test on the adequacy of a particular ARMA (p, q) model is to filter the data through the model and test for whiteness of the output data. This requires that the parameters of the MA model be computed from the estimated autocorrelation, using spectral factorization to determine $B(z)$ from $D(z) = B(z)B(z^{-1})$.

For additional reading on ARMA power spectrum estimation, the reader is referred to the papers by Graupe et al. (1975), Cadzow (1981, 1982), Kay (1980), and Friedlander (1982b).

14.3.9 Some Experimental Results

In this section we present some experimental results on the performance of AR and ARMA power spectrum estimates obtained by using artificially generated data. Our objective is to compare the spectral estimation methods on the basis of their frequency resolution, bias, and their robustness in the presence of additive noise.

The data consist of either one or two sinusoids and additive Gaussian noise. The two sinusoids are spaced Δf apart. Clearly, the underlying process is ARMA (4, 4). The results that are shown employ an AR (p) model for these data. For high signal-to-noise ratios (SNRs) we expect the AR(4) to be adequate. However, for low SNRs, a higher-order AR model is needed to approximate the ARMA (4, 4) process.

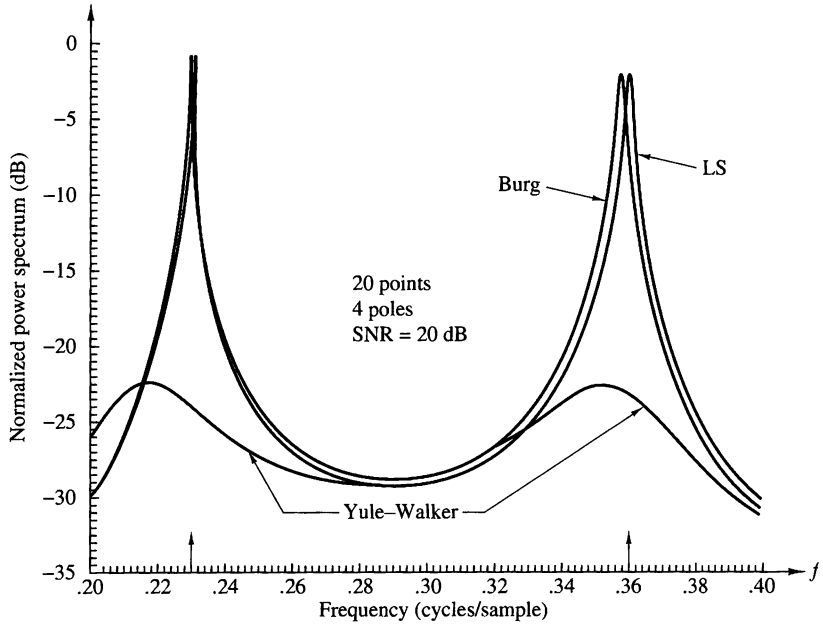


Figure 14.3.1 Comparison of AR spectrum estimation methods.

The results given below are consistent with this statement. The SNR is defined as $10 \log_{10} A^2/2\sigma^2$, where σ^2 is variance of the additive noise and A is the amplitude of the sinusoid.

In Fig. 14.3.1 we illustrate the results for $N = 20$ data points based on an AR(4) model with an SNR = 20 dB and $\Delta f = 0.13$. Note that the Yule–Walker method gives an extremely smooth (broad) spectral estimate with small peaks. If Δf is decreased to $\Delta f = 0.07$, the Yule–Walker method no longer resolves the peaks as illustrated in Fig. 14.3.2. Some bias is also evident in the Burg method. Of course, by increasing the number of data points the Yule–Walker method eventually is able to resolve the peaks. However, the Burg and least-squares methods are clearly superior for short data records.

The effect of additive noise on the estimate is illustrated in Fig. 14.3.3 for the least-squares method. The effect of filter order on the Burg and least-squares methods is illustrated in Figs. 14.3.4 and 14.3.5, respectively. Both methods exhibit spurious peaks as the order is increased to $p = 12$.

The effect of initial phase is illustrated in Figs. 14.3.6 and 14.3.7 for the Burg and least-squares methods. It is clear that the least-squares method exhibits less sensitivity to initial phase than the Burg algorithm.

An example of line splitting for the Burg method is shown in Fig. 14.3.8 with $p = 12$. It does not occur for the AR(8) model. The least-squares method did not exhibit line splitting under the same conditions. On the other hand, the line splitting on the Burg method disappeared with an increase in the number of data points N.

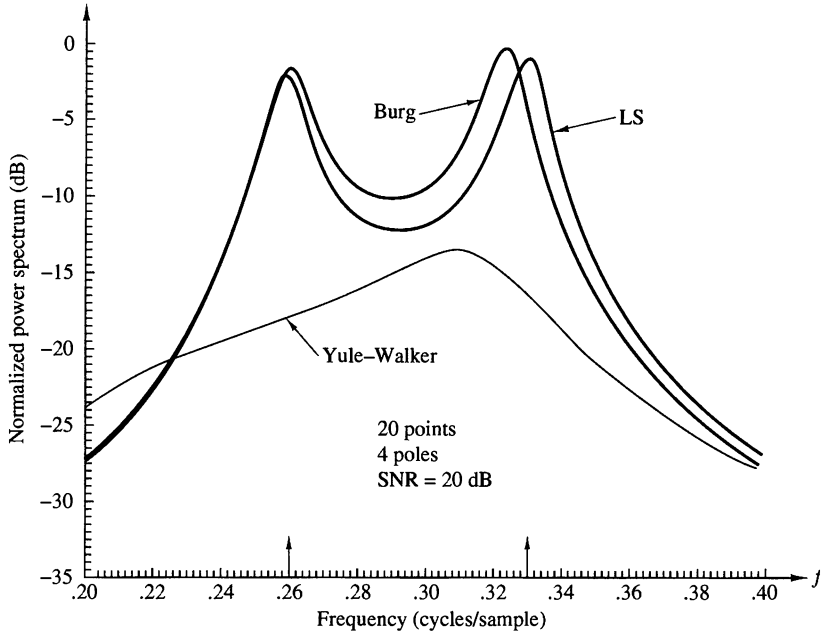


Figure 14.3.2 Comparison of AR spectrum estimation methods.

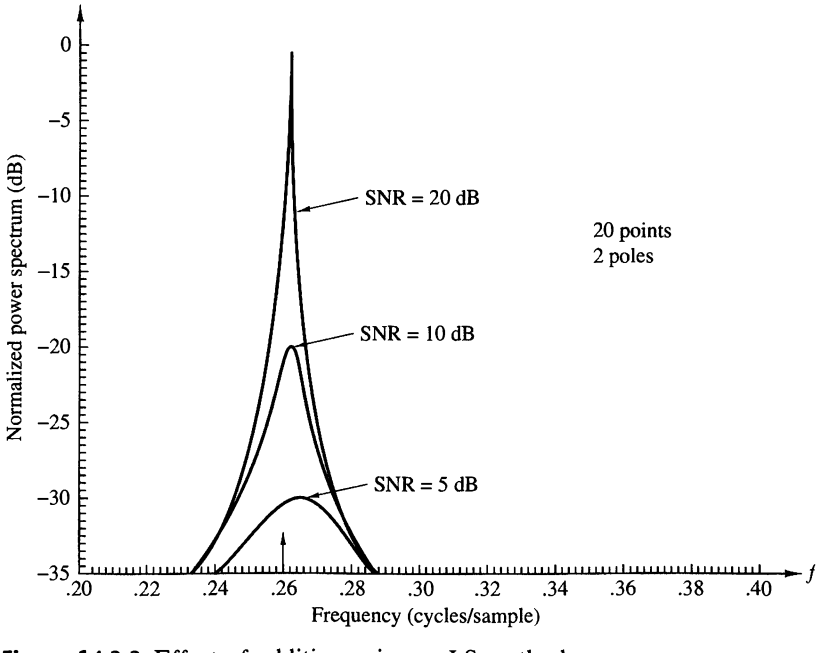


Figure 14.3.3 Effect of additive noise on LS method.

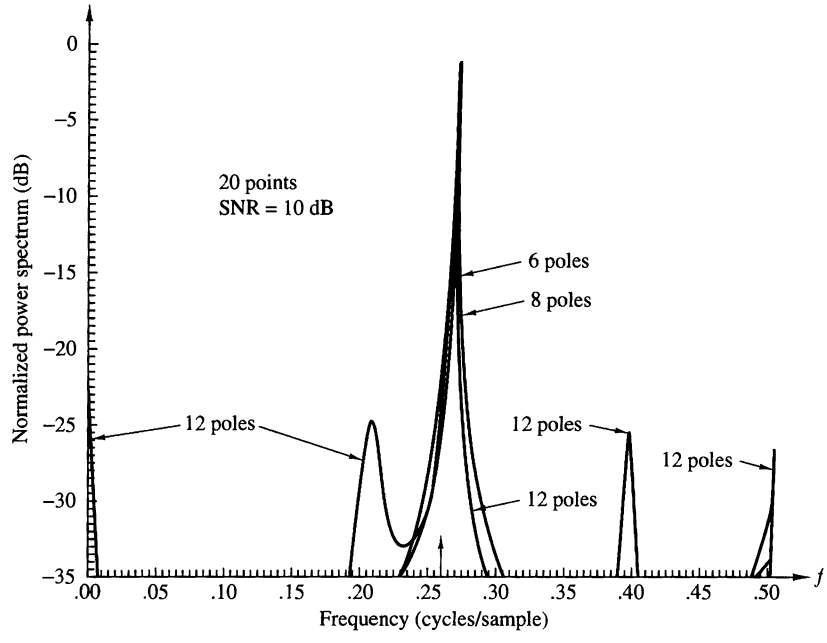


Figure 14.3.4 Effect of filter order of Burg method.

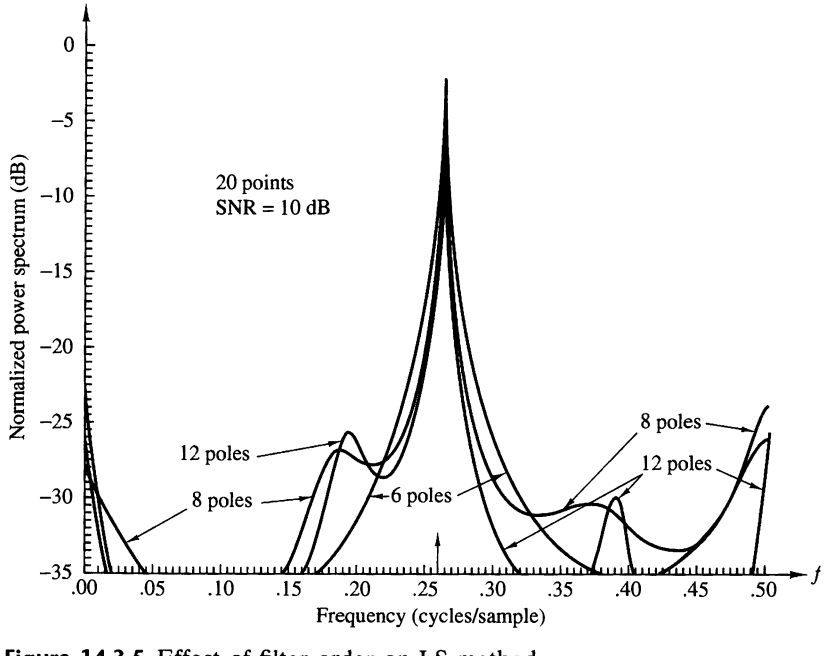


Figure 14.3.5 Effect of filter order on LS method.

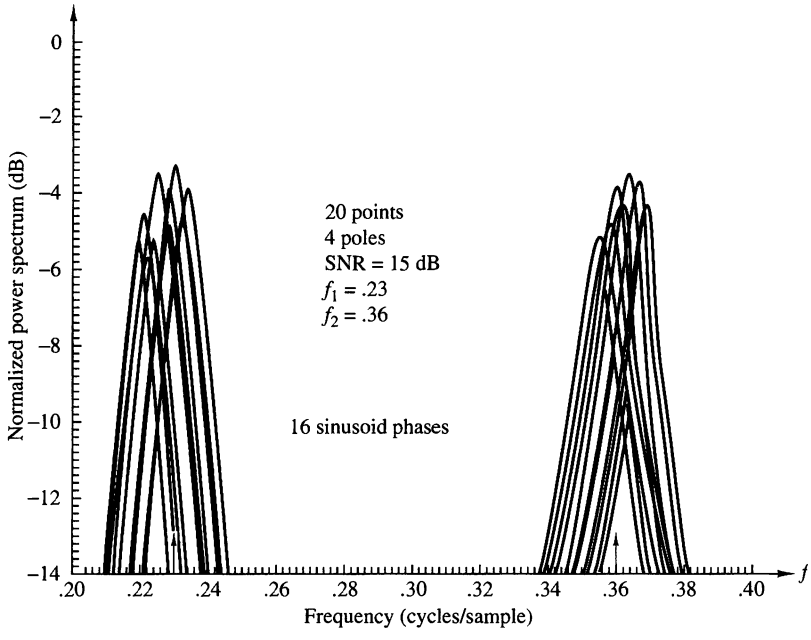


Figure 14.3.6 Effect of initial phase on Burg method.

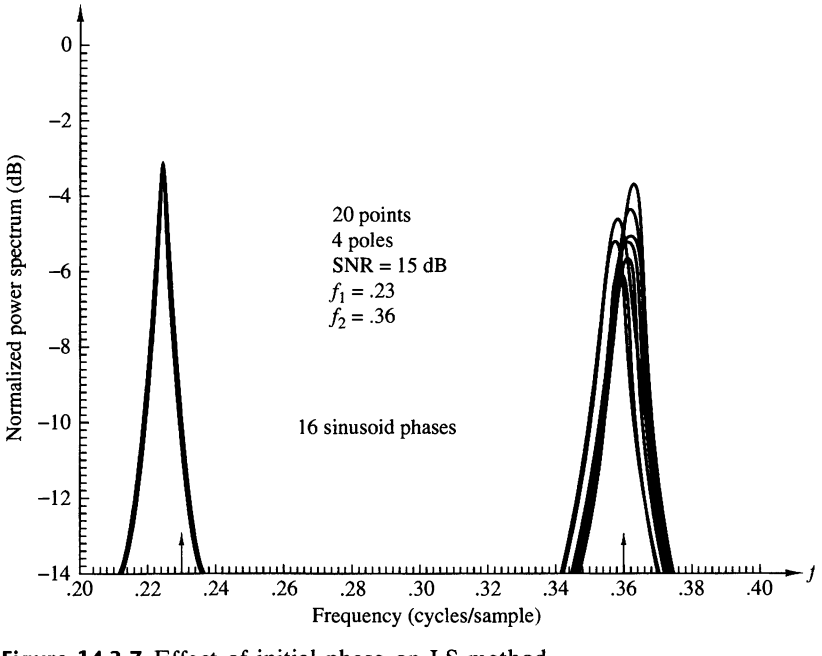


Figure 14.3.7 Effect of initial phase on LS method.

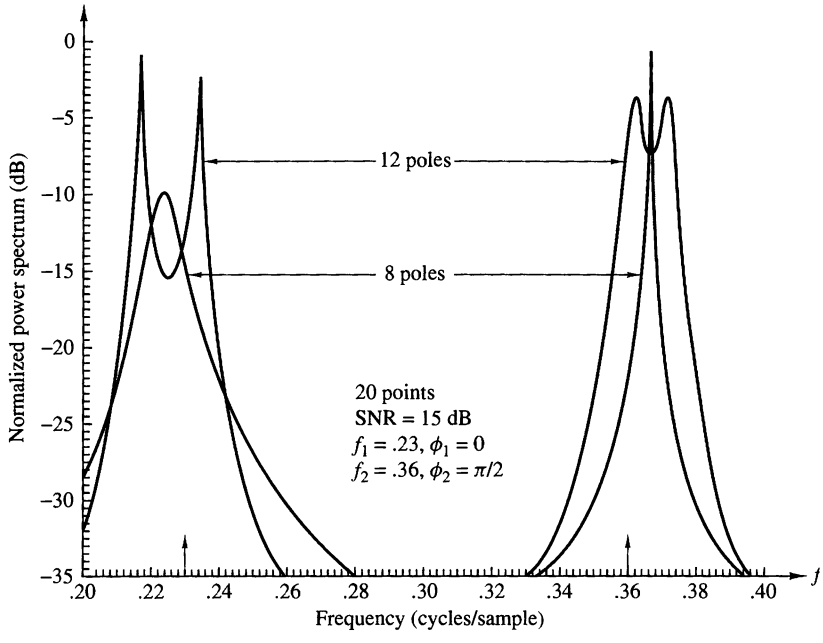


Figure 14.3.8 Line splitting in Burg method.

Figures 14.3.9 and 14.3.10 illustrate the resolution properties of the Burg and least-squares methods for $\Delta f = 0.07$ and $N = 20$ points at low SNR (3 dB). Since the additive noise process is ARMA, a higher-order AR model is required to provide a good approximation at low SNR. Hence the frequency resolution improves as the order is increased.

The FPE for the Burg method is illustrated in Fig. 14.3.11 for an SNR= 3 dB. For this SNR the optimum value is $p = 12$ according to the FPE criterion.

The Burg and least-squares methods were also tested with data from a narrowband process, obtained by exciting a four-pole (two pairs of complex-conjugate poles) narrowband filter and selecting a portion of the output sequence for the data record. Figure 14.3.12 illustrates the superposition of 20 data records of 20 points each. We observe a relatively small variability. In contrast, the Burg method exhibited a much larger variability, approximately a factor of 2 compared to the least-squares method. The results shown in Figs. 14.3.1 through 14.3.12 are taken from Poole (1981).

Finally, we show in Fig. 14.3.13 the ARMA(10, 10) spectral estimates obtained by Kay (1980) for two sinusoids in noise using the least-squares ARMA method described in Section 14.3.8, as an illustration of the quality of power spectrum estimation obtained with the ARMA model.

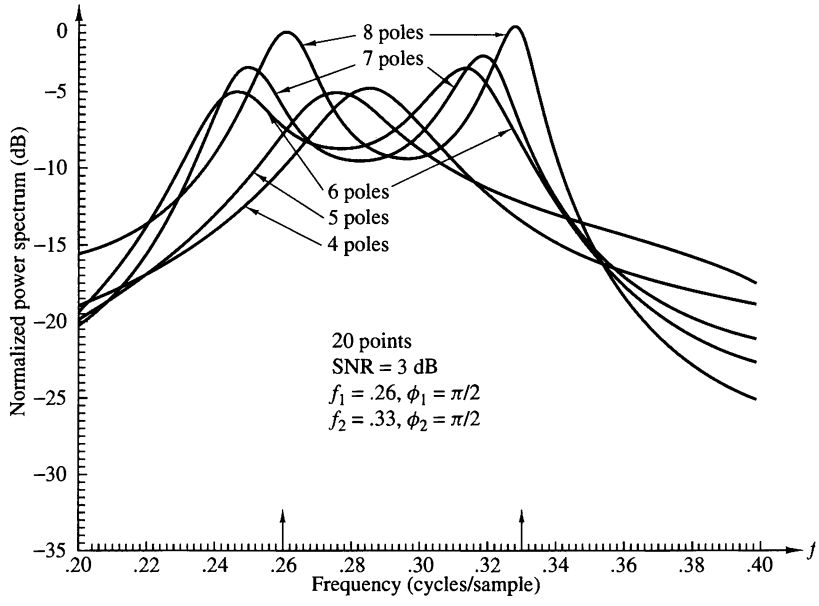


Figure 14.3.9 Frequency resolution of Burg method with $N = 20$ points.

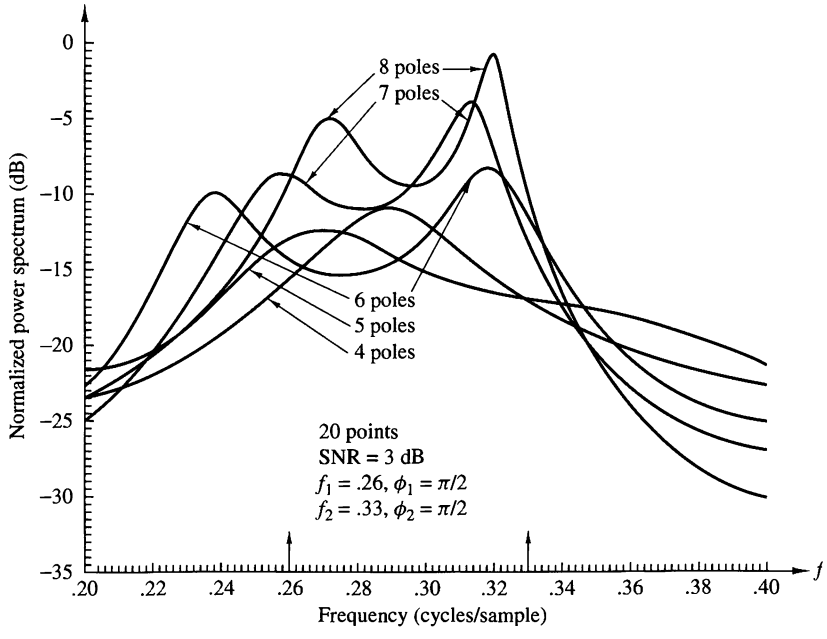


Figure 14.3.10 Frequency resolution of LS method with $N = 20$ points.

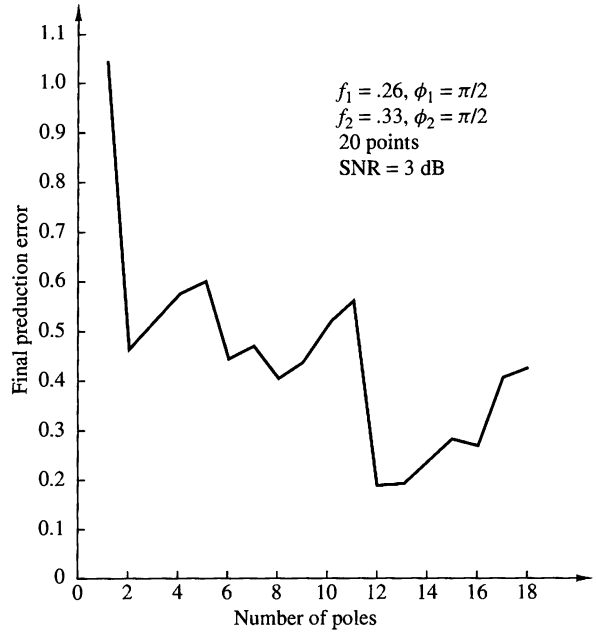


Figure 14.3.11 Final prediction error for Burg estimate.

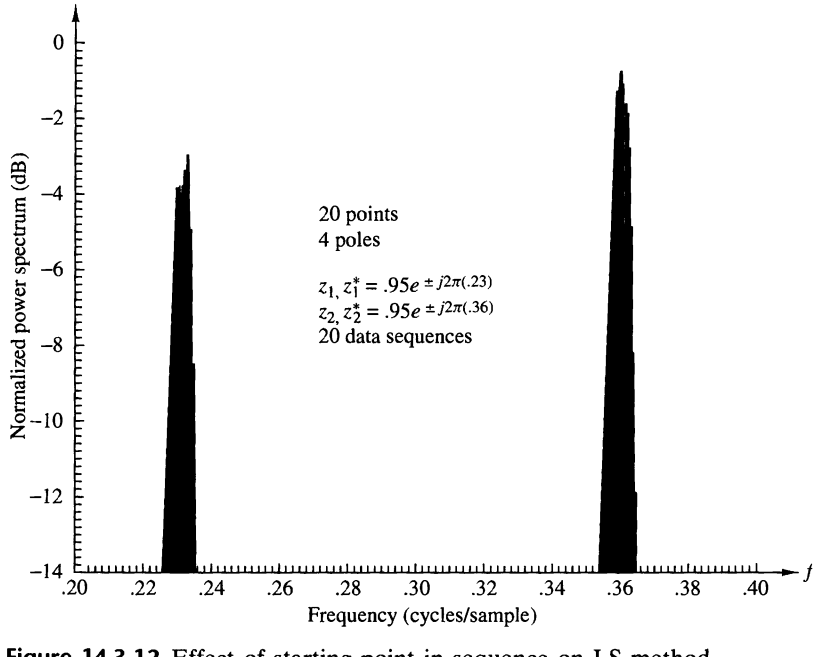


Figure 14.3.12 Effect of starting point in sequence on LS method.

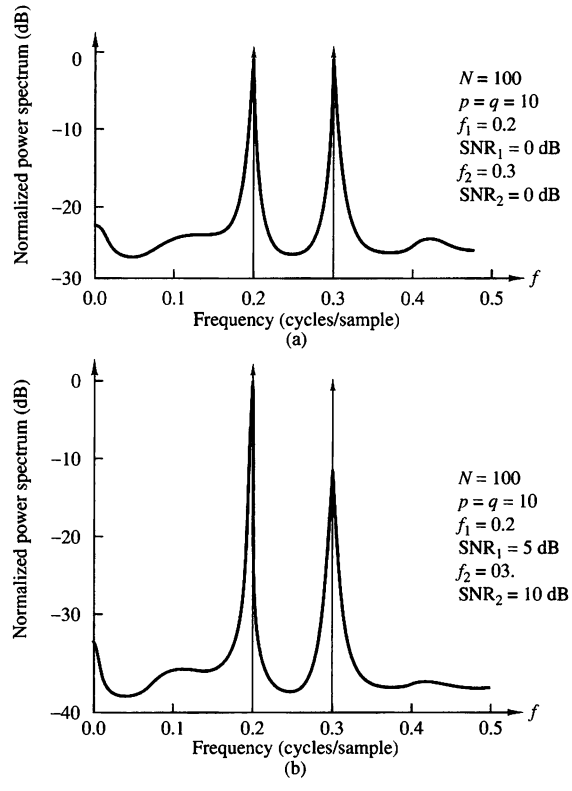


Figure 14.3.13
ARMA (10, 10) power spectrum estimates from paper by Kay (1980). Reprinted with permission from the IEEE.

14.4 Filter Bank Methods

In this section, we consider the estimation of the power spectrum of a signal sequence $\{x(n)\}$ by filtering $\{x(n)\}$ with a parallel bank of FIR filters tuned to the desired frequencies and rectifying the filter outputs. Thus, an estimate of the power spectrum at a particular frequency is obtained as illustrated in the system shown in Figure 14.4.1.

We begin by showing that the conventional periodogram can be computed by using a filter bank, where the FIR filters basically function as rectangular windows on the signal sequence $\{x(n)\}$. Then, we describe a method for designing the filters that exploit the characteristics of the data. This approach leads to a high resolution spectral estimation method that is data adaptive.

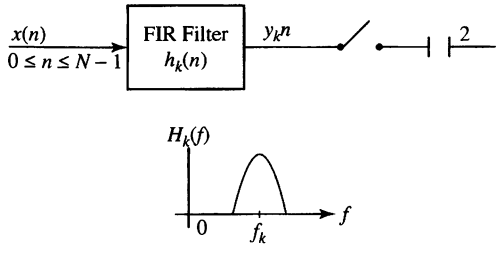


Figure 14.4.1
Measurement of signal power at a frequency in the vicinity of f_k.

14.4.1 Filter Bank Realization of the Periodogram

The power spectrum estimate at a given frequency $f_k = k/N$ that is obtained from the periodogram is

$$\begin{aligned} P_{xx}(f_k) &= \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi f_k n}\right|^2 \\ &= \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{j2\pi f_k (N-n)}\right|^2 \end{aligned} \tag{14.4.1}$$

where $\{x(n), 0 \le n \le N-1\}$ is the signal sequence. Following an approach similar to the development of the Goertzel algorithm (Section 8.3.1), we define a linear filter with impulse response.

$$h_k(n) = \begin{cases} \frac{1}{\sqrt{N}} e^{j2\pi f_k n}, & n \ge 0 \\ 0, & n < 0 \end{cases} \tag{14.4.2}$$

Then, after substituting $h_k(n)$ in equation (14.4.1), we obtain the power spectral estimate at f_k as

$$\begin{aligned} P_{xx}(f) &= \left|\sum_{n=0}^{N-1} x(n)h_k(N-n)\right|^2 \\ &= \left|\sum_{n=0}^{N-1} x(n)h_k(l-n)\right|^2_{l=N} \end{aligned} \tag{14.4.3}$$

Thus, the power spectral estimate at $f = f_k = k/N$ may be obtained by passing the signal sequence $\{x(n), 0 \le n \le N-1\}$ through a linear FIR filter with impulse response $h_k(n)$, as given by equation (14.4.2), evaluating the filter output at $l = N$, and computing the square magnitude of the filter output.

The system function of the filter is

$$\begin{aligned} H_k(z) &= \sum_{n=0}^{\infty} h_k(n)z^{-n} \\ &= \frac{1/\sqrt{N}}{1 - e^{j2\pi f_k} z^{-1}} \end{aligned} \tag{14.4.4}$$

This filter is a single pole filter with the pole on the unit circle, corresponding to the frequency f_k.

The above development suggests that the periodogram can be computed by linearly filtering the signal sequence $\{x(n), 0 \le n \le N-1\}$ by a bank of parallel filters with impulse responses $\{h_k(n), 0 \le n \le N-1\}$, as illustrated in Figure 14.4.2. The 3-dB bandwidth of each of these filters is approximately $1/N$ cycles per sample interval.

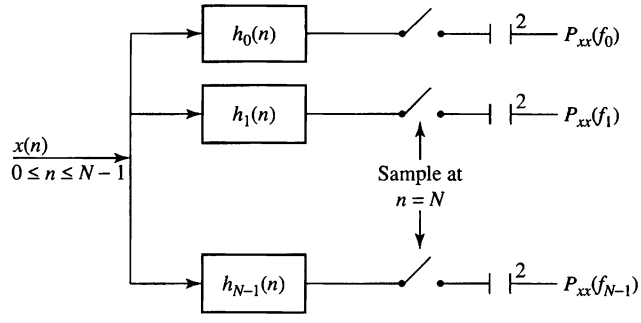


Figure 14.4.2
Filter bank implementation of periodogram estimator.

Another equivalent filter bank implementation that produces the periodogram is shown in Figure 14.4.3. In this filter bank, the signal sequence $\{x(n), 0 \le n \le N-1\}$ is multiplied by the exponential factors $\{e^{-j2\pi kn/N}, 0 \le k \le N-1\}$ and each of the resultant products is passed through a rectangular lowpass filter with impulse response

$$h_0(n) = \begin{cases} 1/\sqrt{N}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} \tag{14.4.5}$$

whose complex-valued output at $l = N-1$ is magnitude squared to yield the spectral estimate at the corresponding frequency $f = f_k$. We may call this spectral estimate the *unwindowed periodogram*.

We may replace the rectangular window in Figure 14.4.3 with another temporal window $h_0(n)$, say a Bartlett or a Hamming or a Kaiser window spanning the same time duration $0 \le n \le N-1$. This would generate a *windowed periodogram*, which may be expressed as

$$P_{xx}(f) = \frac{1}{N}\left|\sum_{n=1}^{N-1} x(n)e^{j2\pi fn}h_0(N-n)\right|^2 \tag{14.4.6}$$

Applying a window function to the signal sequence is similar to the Welch method. However, in the latter, the window function is applied to each of several

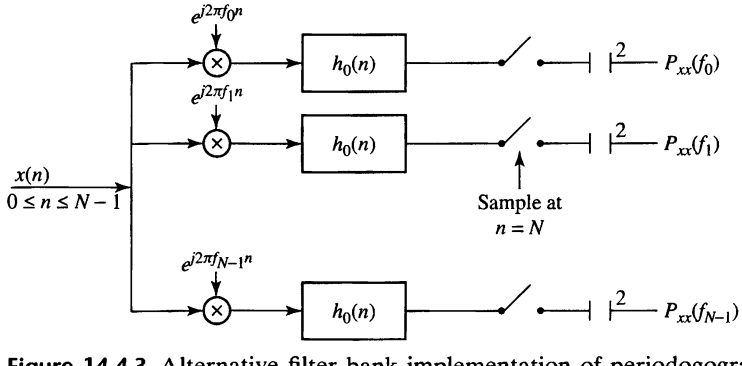


Figure 14.4.3 Alternative filter bank implementation of periodogram estimator.

segments of the data sequence and the windowed periodograms of the individual segments are averaged. Consequently, the variability of the spectral estimate obtained by the Welch method is significantly less than the variability of the spectral estimate obtained by the windowed periodogram in equation (14.4.6). On the other hand, the resolution resulting from the windowed periodogram is higher than the resolution obtained with the Welch method. As we observed in our treatment of nonparametric methods, there is a trade-off between frequency resolution and the variance of the spectral estimates. By segmenting the signal sequence into several smaller sequences, computing the windowed (modified) periodogram for each of the subsequences and, finally, averaging the windowed periodograms, we were able to reduce the variance of the estimate at the cost of decreasing the frequency resolution.

The filter bank interpretation of the periodogram leads us to consider other possible filter characteristics that may result in high resolution spectral estimates. We note that the filters used in the filter bank implementations shown in Figures 14.4.2 and 14.4.3 were not optimized in any manner. Thus, we may say that the filters were selected without regard to the characteristics of the data (i.e., the filters used are data independent). In the following section, we describe a filter bank implementation of the spectral estimator due to Capon (1969), in which the filters are designed based on the statistical characteristics of the data (i.e., the filter impulse responses are adapted to the characteristics of the data).

14.4.2 Minimum Variance Spectral Estimates

The spectral estimator proposed by Capon (1969) was intended for use in large seismic arrays for frequency-wave number estimation. It was later adapted to single-time-series spectrum estimation by Lacoss (1971), who demonstrated that the method provides a minimum-variance unbiased estimate of the spectral components in the signal. This estimator has also been called a "maximum-likelihood spectral estimator" because the filter design method results in the minimum-variance unbiased spectral estimate when the signal is corrupted by additive Gaussian noise.

Following the development of Lacoss, let us consider the design of an FIR filter with coefficients $h(k)$, $0 \le k \le p$, to be determined. Then, if the observed data $x(n)$, $0 \le n \le N-1$, are passed through the filter, the response is

$$y(n) = \sum_{k=0}^{p} h(k)x(n-k) \equiv \mathbf{X}^t(n)\mathbf{h} \tag{14.4.7}$$

where $\mathbf{X}^t(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-p)]$ is the data vector and $\mathbf{h}$ is the filter coefficient vector. If we assume that $E[x(n)] = 0$, the variance of the output sequence is

$$\begin{aligned} \sigma_y^2 &= E[|y(n)|^2] = E[\mathbf{h}^H \mathbf{X}^*(n)\mathbf{X}^t(n)\mathbf{h}] \\ &= \mathbf{h}^H \mathbf{\Gamma}_{xx}\mathbf{h} \end{aligned} \tag{14.4.8}$$

where $\mathbf{\Gamma}_{xx}$ is the autocorrelation matrix of the sequence $x(n)$, with elements $\gamma_{xx}(m)$.

The filter coefficients are selected so that at the frequency f_l, the frequency response of the FIR filter is normalized to unity, that is,

$$\sum_{k=0}^{p} h(k)e^{-j2\pi kf_l} = 1$$

This constraint can also be written in matrix form as

$$\mathbf{E}^H(f_l)\mathbf{h} = 1 \tag{14.4.9}$$

where

$$\mathbf{E}^t(f_l) = [\,1 \quad e^{j2\pi f_l} \quad \cdots \quad e^{j2\pi pf_l}\,]$$

By minimizing the variance σ_y^2 subject to the constraint (14.4.9), we obtain an FIR filter that passes the frequency component f_l undistorted, while components distant from f_l are severely attenuated. The result of this minimization leads to the coefficient vector

$$\hat{\mathbf{h}}_{\text{opt}} = \boldsymbol{\Gamma}_{xx}^{-1}\mathbf{E}(f_l)/\mathbf{E}^H(f_l)\boldsymbol{\Gamma}_{xx}^{-1}\mathbf{E}(f_l) \tag{14.4.10}$$

If $\hat{\mathbf{h}}$ is substituted into (14.4.8), we obtain the minimum variance

$$\sigma_{\text{min}}^2 = \frac{1}{\mathbf{E}^H(f_l)\boldsymbol{\Gamma}_{xx}^{-1}\mathbf{E}(f_l)} \tag{14.4.11}$$

The expression in (14.4.11) is the minimum-variance power spectrum estimate at the frequency f_l. By changing f_l over the range $0 \leq f_l \leq 0.5$, we can obtain the power spectrum estimate. Thus, the minimum-variance method is basically a filter bank implementation for the spectrum estimator. It differs basically from the filter bank interpretation of the periodogram in that the filter coefficients in the Capon method are optimized. It should be noted that although $\mathbf{E}(f)$ changes with the choice of frequency, $\boldsymbol{\Gamma}_{xx}^{-1}$ is computed only once. As demonstrated by Lacoss (1971), the computation of the quadratic form $\mathbf{E}^H(f)\boldsymbol{\Gamma}_{xx}^{-1}\mathbf{E}(f)$ can be done with a single DFT.

With an estimate $\mathbf{R}_{xx}$ of the autocorrelation matrix substituted in place of $\boldsymbol{\Gamma}_{xx}$, we obtain the minimum variance power spectrum estimate of Capon as

$$P_{xx}^{\text{MV}}(f) = \frac{1}{\mathbf{E}^H(f)\mathbf{R}_{xx}^{-1}\mathbf{E}(f)} \tag{14.4.12}$$

It has been shown by Lacoss (1971) that this power spectrum estimator yields estimates of the spectral peaks proportional to the power at that frequency. In constrast, the AR methods described in Section 14.3 result in estimates of the spectral peaks proportional to the square of the power at that frequency.

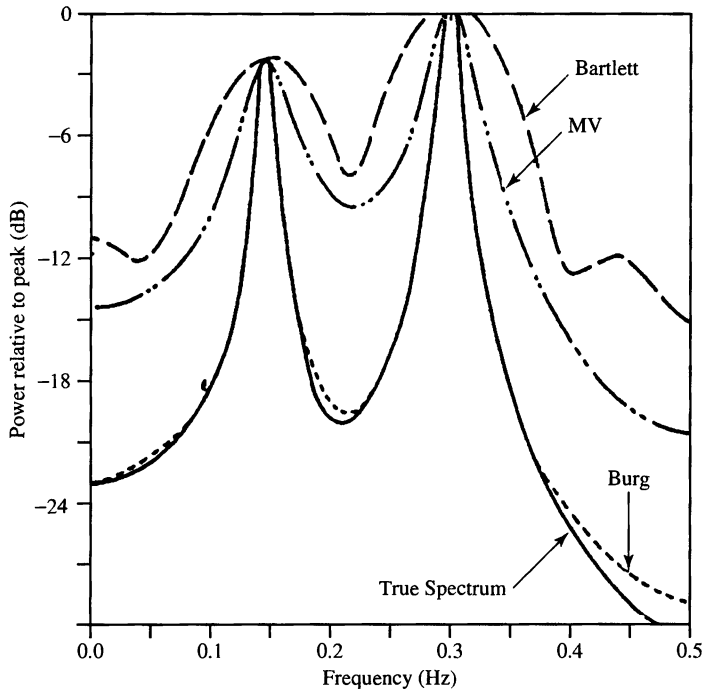


Figure 14.4.4 A comparison of three spectral estimates (Barlett, minimum variance (MV) and Burg) for a signal with two narrowband peaks at frequencies 0.15 and 0.30. (Taken from R.I. Lacoss, "Data Adaptive Spectral Analysis Methods," *Geophysics*, Vol. 36, pp. 661–675, August 1971. Reproduced by permission.)

Experiments on the performance of this method compared with the performance of the Burg method have been done by Lacoss (1971) and others. For example, Figure 14.4.4 taken from the paper by Lacoss (1971) illustrates a comparison of three spectral estimates, the Bartlett estimate, the minimum variance estimate, and the Burg estimate for a signal containing two narrowband peaks at frequencies 0.15 and 0.30. This figure illustrates that the minimum variance spectral estimator yields a better spectral estimate than the Bartlett estimate, but a poorer estimate compared to the Burg estimate. The true power spectrum of the signal is also shown in this figure. In general, the minimum-variance estimate in (14.4.12) outperforms the nonparametric spectral estimators in frequency resolution, but it does not provide the high frequency resolution obtained from the AR methods of Burg and the unconstrained least squares. Furthermore, Burg (1972) demonstrated that for a known correlation sequence, the minimum variance spectrum is related to the AR model spectrum through the equation

$$\frac{1}{\Gamma_{xx}^{\mathrm{MV}}(f)} = \frac{1}{p}\sum_{k=0}^{p}\frac{1}{\Gamma_{xx}^{\mathrm{AR}}(f,k)} \tag{14.4.13}$$

where $\Gamma_{xx}^{\mathrm{AR}}(f,k)$ is the AR power spectrum obtained with an AR(k) model. Thus

the reciprocal of the minimum variance estimate is equal to the average of the reciprocals of all spectra obtained with AR(k) models for $1 \leq k \leq p$. Since low-order AR models, in general, do not provide good resolution, the averaging operation in (14.4.13) reduces the frequency resolution in the spectral estimate. Hence we conclude that the AR power spectrum estimate of order p is superior to the minimum variance estimate of order $p + 1$.

The relationship given by (14.4.13) represents a frequency-domain relationship between the Capon minimum variance estimate and the Burg AR estimate. A time-domain relationship between these two estimates also can be established as shown by Musicus (1985). This has led to a computationally efficient algorithm for the minimum variance estimate.

Additional references to the method of Capon and comparisons with other estimators can be found in the literature. We cite the papers of Capon and Goodman (1971), Marzetta (1983), Marzetta and Lang (1983, 1984), Capon (1983), and McDonough (1983).

14.5 Eigenanalysis Algorithms for Spectrum Estimation

In Section 14.3.8 we demonstrated that an AR(p) process corrupted by additive (white) noise is equivalent to an ARMA(p, p) process. In this section we consider the special case in which the signal components are sinusoids corrupted by additive white noise. The algorithms are based on an eigen-decomposition of the correlation matrix of the noise-corrupted signal.

From our previous discussion on the generation of sinusoids in Chapter 5, we recall that a real sinusoidal signal can be generated via the difference equation,

$$x(n) = -a_1 x(n-1) - a_2 x(n-2) \tag{14.5.1}$$

where $a_1 = 2\cos 2\pi f_k$, $a_2 = 1$, and initially, $x(-1) = -1$, $x(-2) = 0$. This system has a pair of complex-conjugate poles (at $f = f_k$ and $f = -f_k$) and therefore generates the sinusoid $x(n) = \cos 2\pi f_k n$, for $n \geq 0$.

In general, a signal consisting of p sinusoidal components satisfies the difference equation

$$x(n) = -\sum_{m=1}^{2p} a_m x(n-m) \tag{14.5.2}$$

and corresponds to the system with system function

$$H(z) = \frac{1}{1 + \sum_{m=1}^{2p} a_m z^{-m}} \tag{14.5.3}$$

The polynomial

$$A(z) = 1 + \sum_{m=1}^{2p} a_m z^{-m} \tag{14.5.4}$$

has $2p$ roots on the unit circle which correspond to the frequencies of the sinusoids.

Now, suppose that the sinusoids are corrupted by a white noise sequence $w(n)$ with $E[|w(n)|^2] = \sigma_w^2$. Then we observe that

$$y(n) = x(n) + w(n) \tag{14.5.5}$$

If we substitute $x(n) = y(n) - w(n)$ in (14.5.2), we obtain

$$y(n) - w(n) = -\sum_{m=1}^{2p} [y(n-m) - w(n-m)]a_m$$

or, equivalently,

$$\sum_{m=0}^{2p} a_m y(n-m) = \sum_{m=0}^{2p} a_m w(n-m) \tag{14.5.6}$$

where, by definition, $a_0 = 1$.

We observe that (14.5.6) is the difference equation for an ARMA$(2p, 2p)$ process in which both the AR and MA parameters are identical. This symmetry is a characteristic of the sinusoidal signals in white noise. The difference equation in (14.5.6) may be expressed in matrix form as

$$\mathbf{Y}^t\mathbf{a} = \mathbf{W}^t\mathbf{a} \tag{14.5.7}$$

where $\mathbf{Y}^t = [\, y(n) \quad y(n-1) \quad \cdots \quad y(n-2p) \,]$ is the observed data vector of dimension $(2p+1)$, $\mathbf{W}^t = [\, w(n) \quad w(n-1) \quad \cdots \quad w(n-2p) \,]$ is the noise vector, and $\mathbf{a} = [\, 1 \quad a_1 \quad \cdots \quad a_{2p} \,]$ is the coefficient vector.

If we premultiply (14.5.7) by $\mathbf{Y}$ and take the expected value, we obtain

$$\begin{aligned} E(\mathbf{Y}\mathbf{Y}^t)\mathbf{a} &= E(\mathbf{Y}\mathbf{W}^t)\mathbf{a} = E[(\mathbf{X} + \mathbf{W})\mathbf{W}^t]\mathbf{a} \\ \mathbf{\Gamma}_{yy}\mathbf{a} &= \sigma_w^2\mathbf{a} \end{aligned} \tag{14.5.8}$$

where we have used the assumption that the sequence $w(n)$ is zero mean and white, and $\mathbf{X}$ is a deterministic signal.

The equation in (14.5.8) is in the form of an eigenequation, that is,

$$(\mathbf{\Gamma}_{yy} - \sigma_w^2\mathbf{I})\mathbf{a} = \mathbf{0} \tag{14.5.9}$$

where σ_w^2 is an eigenvalue of the autocorrelation matrix $\mathbf{\Gamma}_{yy}$. Then the parameter vector $\mathbf{a}$ is an eigenvector associated with the eigenvalue σ_w^2. The eigenequation in (14.5.9) forms the basis for the Pisarenko harmonic decomposition method.

14.5.1 Pisarenko Harmonic Decomposition Method

For p randomly phased sinusoids in additive white noise, the autocorrelation values are

$$\begin{aligned} \gamma_{yy}(0) &= \sigma_w^2 + \sum_{i=1}^{p} P_i \\ \gamma_{yy}(k) &= \sum_{i=1}^{p} P_i \cos 2\pi f_i k, \qquad k \neq 0 \end{aligned} \tag{14.5.10}$$

where $P_i = A_i^2/2$ is the average power in the ith sinusoid and A_i is the corresponding amplitude. Hence we may write

$$\begin{bmatrix} \cos 2\pi f_1 & \cos 2\pi f_2 & \cdots & \cos 2\pi f_p \\ \cos 4\pi f_1 & \cos 4\pi f_2 & \cdots & \cos 4\pi f_p \\ \vdots & \vdots & & \vdots \\ \cos 2\pi p f_1 & \cos 2\pi p f_2 & \cdots & \cos 2\pi p f_p \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \\ \vdots \\ P_p \end{bmatrix} = \begin{bmatrix} \gamma_{yy}(1) \\ \gamma_{yy}(2) \\ \vdots \\ \gamma_{yy}(p) \end{bmatrix} \tag{14.5.11}$$

If we know the frequencies f_i, $1 \leq i \leq p$, we can use this equation to determine the powers of the sinusoids. In place of $\gamma_{xx}(m)$, we use the estimates $r_{xx}(m)$. Once the powers are known, the noise variance can be obtained from (14.5.10) as

$$\sigma_w^2 = r_{yy}(0) - \sum_{i=1}^{p} P_i \tag{14.5.12}$$

The problem that remains is to determine the p frequencies f_i, $1 \leq i \leq p$, which, in turn, require knowledge of the eigenvector $\mathbf{a}$ corresponding to the eigenvalue σ_w^2. Pisarenko (1973) observed [see also Papoulis (1984) and Grenander and Szegö (1958)] that for an ARMA process consisting of p sinusoids in additive white noise, the variance σ_w^2 corresponds to the minimum eigenvalue of Γ_{yy} when the dimension of the autocorrelation matrix equals or exceeds $(2p + 1) \times (2p + 1)$. The desired ARMA coefficient vector corresponds to the eigenvector associated with the minimum eigenvalue. Therefore, the frequencies f_i, $1 \leq i \leq p$ are obtained from the roots of the polynomial in (14.5.4), where the coefficients are the elements of the eigenvector $\mathbf{a}$ corresponding to the minimum eigenvalue σ_w^2.

In summary, the Pisarenko harmonic decomposition method proceeds as follows. First we estimate $\mathbf{\Gamma}_{yy}$ from the data (i.e., we form the autocorrelation matrix $\mathbf{R}_{yy}$). Then we find the minimum eigenvalue and the corresponding minimum eigenvector. The minimum eigenvector yields the parameters of the ARMA$(2p, 2p)$ model. From (14.5.4) we can compute the roots that constitute the frequencies $\{f_i\}$. By using these frequencies, we can solve (14.5.11) for the signal powers $\{P_i\}$ by substituting the estimates $r_{yy}(m)$ for $\gamma_{yy}(m)$.

As will be seen in the following example, the Pisarenko method is based on the use of a noise subspace eigenvector to estimate the frequencies of the sinusoids.

EXAMPLE 14.5.1

Suppose that we are given the autocorrelation values $\gamma_{yy}(0) = 3$, $\gamma_{yy}(1) = 1$, and $\gamma_{yy}(2) = 0$ for a process consisting of a single sinusoid in additive white noise. Determine the frequency, its power, and the variance of the additive noise.

Solution. The correlation matrix is

$$\mathbf{\Gamma}_{yy} = \begin{bmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

The minimum eigenvalue is the smallest root of the characteristic polynomial

$$g(\lambda) = \begin{vmatrix} 3-\lambda & 1 & 0 \\ 1 & 3-\lambda & 1 \\ 0 & 1 & 3-\lambda \end{vmatrix} = (3-\lambda)(\lambda^2 - 6\lambda + 7) = 0$$

Therefore, the eigenvalues are $\lambda_1 = 3$, $\lambda_2 = 3 + \sqrt{2}$, $\lambda_3 = 3 - \sqrt{2}$.

The variance of the noise is

$$\sigma_w^2 = \lambda_{\min} = 3 - \sqrt{2}$$

The corresponding eigenvalue is the vector that satisfies (14.5.9), that is,

$$\begin{bmatrix} \sqrt{2} & 1 & 0 \\ 1 & \sqrt{2} & 1 \\ 0 & 1 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The solution is $a_1 = -\sqrt{2}$ and $a_2 = 1$.

The next step is to use the value a_1 and a_2 to determine the roots of the polynomial in (14.5.4). We have

$$z^2 - \sqrt{2}z + 1 = 0$$

Thus

$$z_1, z_2 = \frac{1}{\sqrt{2}} \pm j\frac{1}{\sqrt{2}}$$

Note that $|z_1| = |z_2| = 1$, so that the roots are on the unit circle. The corresponding frequency is obtained from

$$z_i = e^{j2\pi f_1} = \frac{1}{\sqrt{2}} + j\frac{1}{\sqrt{2}}$$

which yields $f_1 = \frac{1}{8}$. Finally, the power of the sinusoid is

$$P_1 \cos 2\pi f_1 = \gamma_{yy}(1) = 1$$

$$P_1 = \sqrt{2}$$

and its amplitude is $A = \sqrt{2P_1} = \sqrt{2\sqrt{2}}$.

As a check on our computations, we have

$$\sigma_w^2 = \gamma_{yy}(0) - P_1$$

$$= 3 - \sqrt{2}$$

which agrees with $\lambda_{\min}$.

14.5.2 Eigen-decomposition of the Autocorrelation Matrix for Sinusoids in White Noise

In the previous discussion we assumed that the sinusoidal signal consists of p real sinusoids. For mathematical convenience we shall now assume that the signal consists of p complex sinusoids of the form

$$x(n) = \sum_{i=1}^{p} A_i e^{j(2\pi f_i n + \phi_i)} \tag{14.5.13}$$

where the amplitudes $\{A_i\}$ and the frequencies $\{f_i\}$ are unknown and the phases $\{\phi_i\}$ are statistically independent random variables uniformly distributed on $(0, 2\pi)$. Then the random process $x(n)$ is wide-sense stationary with autocorrelation function

$$\gamma_{xx}(m) = \sum_{i=1}^{p} P_i e^{j2\pi f_i m} \tag{14.5.14}$$

where, for complex sinusoids, $P_i = A_i^2$ is the power of the ith sinusoid.

Since the sequence observed is $y(n) = x(m) + w(n)$, where $w(n)$ is a white noise sequence with spectral density σ_w^2, the autocorrelation function for $y(n)$ is

$$\gamma_{yy}(m) = \gamma_{xx}(m) + \sigma_w^2 \delta(m), \qquad m = 0, \pm 1, \ldots, \pm(M-1) \tag{14.5.15}$$

Hence the $M \times M$ autocorrelation matrix for $y(n)$ can be expressed as

$$\mathbf{\Gamma}_{yy} = \mathbf{\Gamma}_{xx} + \sigma_w^2 \mathbf{I} \tag{14.5.16}$$

where $\mathbf{\Gamma}_{xx}$ is the autocorrelation matrix for the signal $x(n)$ and $\sigma_w^2\mathbf{I}$ is the autocorrelation matrix for the noise. Note that if we select $M > p$, $\mathbf{\Gamma}_{xx}$ which is of dimension $M \times M$ is not of full rank, because its rank is p. However, $\mathbf{\Gamma}_{yy}$ is full rank because $\sigma_w^2\mathbf{I}$ is of rank M.

In fact, the signal matrix $\mathbf{\Gamma}_{xx}$ can be represented as

$$\mathbf{\Gamma}_{xx} = \sum_{i=1}^{p} P_i \mathbf{s}_i \mathbf{s}_i^H \tag{14.5.17}$$

where H denotes the conjugate transpose and $\mathbf{s}_i$ is a signal vector of dimension M defined as

$$\mathbf{s}_i = [1, e^{j2\pi f_i}, e^{j4\pi f_i}, \ldots, e^{j2\pi(M-1)f_i}] \tag{14.5.18}$$

Since each vector (outer product) $\mathbf{s}_i\mathbf{s}_i^H$ is a matrix of rank 1 and since there are p vector products, the matrix $\mathbf{\Gamma}_{xx}$ is of rank p. Note that if the sinusoids were real, the correlation matrix $\mathbf{\Gamma}_{xx}$ would have rank $2p$.

Now, let us perform an eigen-decomposition of the matrix $\mathbf{\Gamma}_{yy}$. Let the eigenvalues $\{\lambda_i\}$ be ordered in decreasing value with $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \cdots \geq \lambda_M$ and let the corresponding eigenvectors be denoted as $\{\mathbf{v}_i,\ i = 1, \ldots, M\}$. We assume that

the eigenvectors are normalized so that $\mathbf{v}_i^H \cdot \mathbf{v}_j = \delta_{ij}$. In the absence of noise the eigenvalues λ_i, $i = 1, 2, \ldots, p$, are nonzero while $\lambda_{p+1} = \lambda_{p+2} = \cdots = \lambda_M = 0$. Furthermore, it follows that the signal correlation matrix can be expressed as

$$\mathbf{\Gamma}_{xx} = \sum_{i=1}^{p} \lambda_i \mathbf{v}_i \mathbf{v}_i^H \tag{14.5.19}$$

Thus, the eigenvectors $\mathbf{v}_i$, $i = 1, 2, \ldots, p$ span the signal subspace as do the signal vectors $\mathbf{s}_i$, $i = 1, 2, \ldots, p$. These p eigenvectors for the signal subspace are called the *principal eigenvectors* and the corresponding eigenvalues are called the *principal eigenvalues*.

In the presence of noise, the noise autocorrelation matrix in (14.5.16) can be represented as

$$\sigma_w^2 \mathbf{I} = \sigma_w^2 \sum_{i=1}^{M} \mathbf{v}_i \mathbf{v}_i^H \tag{14.5.20}$$

By substituting (14.5.19) and (14.5.20) into (14.5.16), we obtain

$$\begin{aligned} \mathbf{\Gamma}_{yy} &= \sum_{i=1}^{p} \lambda_i \mathbf{v}_i \mathbf{v}_i^H + \sum_{i=1}^{M} \sigma_w^2 \mathbf{v}_i \mathbf{v}_i^H \\ &= \sum_{i=1}^{p} (\lambda_i + \sigma_w^2) \mathbf{v}_i \mathbf{v}_i^H + \sum_{i=p+1}^{M} \sigma_w^2 \mathbf{v}_i \mathbf{v}_i^H \end{aligned} \tag{14.5.21}$$

This eigen-decomposition separates the eigenvectors into two sets. The set $\{\mathbf{v}_i, i = 1, 2, \ldots, p\}$, which are the principal eigenvectors, span the signal subspace, while the set $\{\mathbf{v}_i, i = p + 1, \ldots, M\}$, which are orthogonal to the principal eigenvectors, are said to belong to the noise subspace. Since the signal vectors $\{\mathbf{s}_i, i = 1, 2, \ldots, p\}$ are in the signal subspace, it follows that the $\{\mathbf{s}_i\}$ are simply linear combinations of the principal eigenvectors and are also orthogonal to the vectors in the noise subspace.

In this context we see that the Pisarenko method is based on an estimation of the frequencies by using the orthogonality property between the signal vectors and the vectors in the noise subspace. For complex sinusoids, if we select $M = p + 1$ (for real sinusoids we select $M = 2p + 1$), there is only a single eigenvector in the noise subspace (corresponding to the minimum eigenvalue) which must be orthogonal to the signal vectors. Thus we have

$$\mathbf{s}_i^H \mathbf{v}_{p+1} = \sum_{k=0}^{p} v_{p+1}(k+1) e^{-j2\pi f_i k} = 0, \qquad i = 1, 2, \ldots, p \tag{14.5.22}$$

But (14.5.22) implies that the frequencies $\{f_i\}$ can be determined by solving for the zeros of the polynomial

$$V(z) = \sum_{k=0}^{p} v_{p+1}(k+1) z^{-k} \tag{14.5.23}$$

all of which lie on the unit circle. The angles of these roots are $2\pi f_i$, $i = 1, 2, \ldots, p$.

When the number of sinusoids is unknown, the determination of p may prove to be difficult, especially if the signal level is not much higher than the noise level. In theory, if $M > p + 1$, there is a multiplicity $(M - p)$ of the minimum eigenvalue. However, in practice the $(M - p)$ small eigenvalues of $\mathbf{R}_{yy}$ will probably be different. By computing all the eigenvalues it may be possible to determine p by grouping the $M - p$ small (noise) eigenvalues into a set and averaging them to obtain an estimate of σ_w^2. Then, the average value can be used in (14.5.9) along with $\mathbf{R}_{yy}$ to determine the corresponding eigenvector.

14.5.3 MUSIC Algorithm

The multiple signal classification (MUSIC) method is also a noise subspace frequency estimator. To develop the method, let us first consider the "weighted" spectral estimate

$$P(f) = \sum_{k=p+1}^{M} w_k |\mathbf{s}^H(f)\mathbf{v}_k|^2 \tag{14.5.24}$$

where $\{\mathbf{v}_k, k = p + 1, \ldots, M\}$ are the eigenvectors in the noise subspace, $\{w_k\}$ are a set of positive weights, and $\mathbf{s}(f)$ is the complex sinusoidal vector

$$\mathbf{s}(f) = [1, e^{j2\pi f}, e^{j4\pi f}, \ldots, e^{j2\pi(M-1)f}] \tag{14.5.25}$$

Note that at $f = f_i$, $\mathbf{s}(f_i) \equiv \mathbf{s}_i$, so that at any one of the p sinusoidal frequency components of the signal, we have

$$P(f_i) = 0, \qquad i = 1, 2, \ldots, p \tag{14.5.26}$$

Hence, the reciprocal of $P(f)$ is a sharply peaked function of frequency and provides a method for estimating the frequencies of the sinusoidal components. Thus

$$\frac{1}{P(f)} = \frac{1}{\sum_{k=p+1}^{M} w_k \left|\mathbf{s}^H(f)\mathbf{v}_k\right|^2} \tag{14.5.27}$$

Although theoretically $1/P(f)$ is infinite at $f = f_i$, in practice the estimation errors result in finite values for $1/P(f)$ at all frequencies.

The MUSIC sinusoidal frequency estimator proposed by Schmidt (1981, 1986) is a special case of (14.5.27) in which the weights $w_k = 1$ for all k. Hence

$$P_{\text{MUSIC}}(f) = \frac{1}{\sum_{k=p+1}^{M} \left|\mathbf{s}^H(f)\mathbf{v}_k\right|^2} \tag{14.5.28}$$

The estimates of the sinusoidal frequencies are the peaks of $P_{\text{MUSIC}}(f)$. Once the sinusoidal frequencies are estimated, the power of each of the sinusoids can be obtained by solving (14.5.11).

EXAMPLE 14.5.2

The autocorrelation matrix for a signal consisting of a complex exponential in white noise is given as

$$\mathbf{\Gamma}_{yy} = \begin{bmatrix} 3 & -2j & -2 \\ 2j & 3 & -2j \\ -2 & 2j & 3 \end{bmatrix}$$

Use the MUSIC method to determine the frequencies and the powers of the complex exponential and the variance of the additive noise.

Solution. By solving for the roots of the polynomial

$$g(\lambda) = \begin{vmatrix} 3-\lambda & -2j & -2 \\ 2j & 3-\lambda & -2j \\ -2 & 2j & 3-\lambda \end{vmatrix} = \lambda^3 - 9\lambda^2 + 15\lambda - 7 = 0$$

we find that the eigenvalues are $\lambda_1 = 7$, $\lambda_2 = 1$, and $\lambda_3 = 1$. Hence, we conclude that there is a single complex exponential, corresponding to the eigenvalue $\lambda = 7$. The eigenvectors corresponding to the signal and the noise subspaces are, respectively,

$$\mathbf{v}_1 = \begin{bmatrix} 1/\sqrt{3} \\ j/\sqrt{3} \\ -1/3 \end{bmatrix}, \qquad \mathbf{v}_2 = \begin{bmatrix} 0 \\ j/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \qquad \mathbf{v}_3 = \begin{bmatrix} 2j/\sqrt{6} \\ 1/\sqrt{6} \\ j/\sqrt{6} \end{bmatrix}$$

By computing the denominator of (14.5.28) we obtain

$$\sum_{k=2}^{3} |\mathbf{s}^H(f)\mathbf{v}_k|^2 = 2 + \frac{5}{3}\cos\left(2\pi f + \frac{\pi}{2}\right) + \frac{2}{3}\cos 4\pi f + \frac{1}{3}\cos\left(2\pi f - \frac{\pi}{2}\right)$$

It is easily verified that this term is zero at $f = 1/4$. Furthermore, from the relation (14.5.15) and our knowledge that $\sigma_w^2 = 1$, we conclude that the power of the complex exponential signal is $P = 2$.

In comparing the Pisarenko method to the MUSIC algorithm, we note that the former selects $M = p + 1$ and projects the signal vectors onto a single noise eigenvector. Hence, the Pisarenko method assumes precise knowledge of the number of sinusoidal components in the signal. In contrast, the MUSIC method selects $M > p + 1$ and, after performing the eigenanalysis, subdivides the eigenvalues into two groups, those (p) corresponding to the signal subspace and those $(M - p)$ corresponding to the noise subspace. Then, the signal vectors are projected onto the $M - p$ eigenvectors in the noise subspace. Hence precise knowledge of p is not necessary. The order selection methods described in Section 14.5.5 may be used to obtain an estimate of p and to select M such that $M > p + 1$.

14.5.4 ESPRIT Algorithm

ESPRIT (estimation of signal parameters via rotational invariance techniques) is yet another method for estimating frequencies of a sum of sinusoids by use of an eigendecomposition approach. As we observe from the development that follows, which

is due to Roy et al. (1986), ESPRIT exploits an underlying rotational invariance of signal subspaces spanned by two temporally displaced data vectors.

We again consider the estimation of p complex-valued sinusoids in additive white noise. The received sequence is given by the vector

$$\begin{aligned}\mathbf{y}(n) &= [y(n), y(n+1), \ldots, y(n+M-1)]^t \\ &= \mathbf{x}(n) + \mathbf{w}(n)\end{aligned} \tag{14.5.29}$$

where $\mathbf{x}(n)$ is the signal vector and $\mathbf{w}(n)$ is the noise vector. To exploit the deterministic character of the sinusoids, we define the time-displaced vector $\mathbf{z}(n) = \mathbf{y}(n+1)$. Thus

$$\begin{aligned}\mathbf{z}(n) &= [z(n), z(n+1), \ldots, z(n+M-1)]^t \\ &= [y(n+1), y(n+2), \ldots, y(n+M)]^t\end{aligned} \tag{14.5.30}$$

With these definitions we can express the vectors $\mathbf{y}(n)$ and $\mathbf{z}(n)$ as

$$\begin{aligned}\mathbf{y}(n) &= \mathbf{S}\mathbf{a} + \mathbf{w}(n) \\ \mathbf{z}(n) &= \mathbf{S}\boldsymbol{\Phi}\mathbf{a} + \mathbf{w}(n)\end{aligned} \tag{14.5.31}$$

where $\mathbf{a} = [a_1, a_2, \ldots, a_p]^t$, $a_i = A_i e^{j\phi i}$, and $\boldsymbol{\Phi}$ is a diagonal $p \times p$ matrix consisting of the relative phase between adjacent time samples of each of the complex sinusoids,

$$\boldsymbol{\Phi} = \text{diag}[e^{j2\pi f_1}, e^{j2\pi f_2}, \ldots, e^{j2\pi f_p}] \tag{14.5.32}$$

Note that the matrix $\boldsymbol{\Phi}$ relates the time-displaced vectors $\mathbf{y}(n)$ and $\mathbf{z}(n)$ and can be called a rotation operator. We also note that $\boldsymbol{\Phi}$ is unitary. The matrix $\mathbf{S}$ is the $M \times p$ Vandermonde matrix specified by the column vectors

$$\mathbf{s}_i = [1, e^{j2\pi f_i}, e^{j4\pi f_i}, \ldots, e^{j2\pi(M-1)f_i}], \qquad i = 1, 2, \ldots, p \tag{14.5.33}$$

Now the autocovariance matrix for the data vector $\mathbf{y}(n)$ is

$$\begin{aligned}\boldsymbol{\Gamma}_{yy} &= E[\mathbf{y}(n)\mathbf{y}^H(n)] \\ &= \mathbf{S}\mathbf{P}\mathbf{S}^H + \sigma_w^2\mathbf{I}\end{aligned} \tag{14.5.34}$$

where $\mathbf{P}$ is the $p \times p$ diagonal matrix consisting of the powers of the complex sinusoids,

$$\begin{aligned}\mathbf{P} &= \text{diag}[|a_1|^2, |a_2|^2, \ldots, |a_p|^2] \\ &= \text{diag}[P_1, P_2, \ldots, P_p]\end{aligned} \tag{14.5.35}$$

We observe that $\mathbf{P}$ is a diagonal matrix since complex sinusoids of different frequencies are orthogonal over the infinite interval. However, we should emphasize that the ESPRIT algorithm does not require $\mathbf{P}$ to be diagonal. Hence the algorithm

is applicable to the case in which the covariance matrix is estimated from finite data records.

The crosscovariance matrix of the signal vectors $\mathbf{y}(n)$ and $\mathbf{z}(n)$ is

$$\mathbf{\Gamma}_{yz} = E[\mathbf{y}(n)\mathbf{z}^H(n)] = \mathbf{SP}\mathbf{\Phi}^H\mathbf{S}^H + \mathbf{\Gamma}_w \tag{14.5.36}$$

where

$$\begin{aligned}\mathbf{\Gamma}_w &= E[\mathbf{w}(n)\mathbf{w}^H(n+1)] \\ &= \sigma_w^2 \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \equiv \sigma_w^2\mathbf{Q}\end{aligned} \tag{14.5.37}$$

The auto and crosscovariance matrices $\mathbf{\Gamma}_{yy}$ and $\mathbf{\Gamma}_{yz}$ are given as

$$\mathbf{\Gamma}_{yy} = \begin{bmatrix} \gamma_{yy}(0) & \gamma_{yy}(1) & \cdots & \gamma_{yy}(M-1) \\ \gamma_{yy}^*(1) & \gamma_{yy}(0) & \cdots & \gamma_{yy}(M-2) \\ \vdots & \vdots & & \vdots \\ \gamma_{yy}^*(M-1) & \gamma_{yy}(M-2) & \cdots & \gamma_{yy}(0) \end{bmatrix} \tag{14.5.38}$$

$$\mathbf{\Gamma}_{yz} = \begin{bmatrix} \gamma_{yy}(1) & \gamma_{yy}(2) & \cdots & \gamma_{yy}(M) \\ \gamma_{yy}(0) & \gamma_{yy}(1) & \cdots & \gamma_{yy}(M-1) \\ \vdots & \vdots & & \vdots \\ \gamma_{yy}^*(M-2) & \gamma_{yy}^*(M-3) & \cdots & \gamma_{yy}(1) \end{bmatrix} \tag{14.5.39}$$

where $\gamma_{yy}(m) = E[y^*(n)y(n+m)]$. Note that both $\mathbf{\Gamma}_{yy}$ and $\mathbf{\Gamma}_{yz}$ are Toeplitz matrices.

Based on this formulation, the problem is to determine the frequencies $\{f_i\}$ and their powers $\{P_i\}$ from the autocorrelation sequence $\{\gamma_{yy}(m)\}$.

From the underlying model, it is clear that the matrix $\mathbf{SPS}^H$ has rank p. Consequently, $\mathbf{\Gamma}_{yy}$ given by (14.5.34) has $(M-p)$ identical eigenvalues equal to σ_w^2. Hence

$$\mathbf{\Gamma}_{yy} - \sigma_w^2\mathbf{I} = \mathbf{SPS}^H \equiv \mathbf{C}_{yy} \tag{14.5.40}$$

From (14.5.36) we also have

$$\mathbf{\Gamma}_{yz} - \sigma_w^2\mathbf{\Gamma}_w = \mathbf{SP}\mathbf{\Phi}^H\mathbf{S}^H \equiv \mathbf{C}_{yz} \tag{14.5.41}$$

Now, let us consider the matrix $\mathbf{C}_{yy} - \lambda\mathbf{C}_{yz}$, which can be written as

$$\mathbf{C}_{yy} - \lambda\mathbf{C}_{yz} = \mathbf{SP}(\mathbf{I} - \lambda\mathbf{\Phi}^H)\mathbf{S}^H \tag{14.5.42}$$

Clearly, the column space of $\mathbf{SPS}^H$ is identical to the column space of $\mathbf{SP}\mathbf{\Phi}^H\mathbf{S}^H$. Consequently, the rank of $\mathbf{C}_{yy} - \lambda\mathbf{C}_{yz}$ is equal to p. However, we note that if $\lambda = \exp(j2\pi f_i)$, the ith row of $(\mathbf{I} - \lambda\mathbf{\Phi}^H)$ is zero and hence the rank of $[\mathbf{I} - \mathbf{\Phi}^H \exp(j2\pi f_i)]$ is $p-1$. But $\lambda_i = \exp(j2\pi f_i)$, $i = 1, 2, \ldots, p$, are the generalized eigenvalues of the

matrix pair $(\mathbf{C}_{yy}, \mathbf{C}_{yz})$. Thus the p generalized eigenvalues $\{\lambda_i\}$ that lie on the unit circle correspond to the elements of the rotation operator $\mathbf{\Phi}$. The remaining $M - p$ generalized eigenvalues of the pair $\{\mathbf{C}_{yy}, \mathbf{C}_{yz}\}$, which correspond to the common null space of these matrices, are zero [i.e., the $(M - p)$ eigenvalues are at the origin in the complex plane].

Based on these mathematical relationships we can formulate an algorithm (ESPRIT) for estimating the frequencies $\{f_i\}$. The procedure is as follows:

1. From the data, compute the autocorrelation values $r_{yy}(m)$, $m = 1, 2, \ldots, M$, and form the matrices $\mathbf{R}_{yy}$ and $\mathbf{R}_{yz}$ corresponding to estimates of $\mathbf{\Gamma}_{yy}$ and $\mathbf{\Gamma}_{yz}$.
2. Compute the eigenvalues of $\mathbf{R}_{yy}$. For $M > p$, the minimum eigenvalue is an estimate of σ_w^2.
3. Compute $\hat{\mathbf{C}}_{yy} = \mathbf{R}_{yy} - \hat{\sigma}_w^2\mathbf{I}$ and $\hat{\mathbf{C}}_{yz} = \mathbf{R}_{yz} - \hat{\sigma}_w^2\mathbf{Q}$, where $\mathbf{Q}$ is defined in (14.5.37).
4. Compute the generalized eigenvalues of the matrix pair $\{\hat{\mathbf{C}}_{yy}, \hat{\mathbf{C}}_{yz}\}$. The p generalized eigenvalues of these matrices that lie on (or near) the unit circle determine the (estimate) elements of $\mathbf{\Phi}$ and hence the sinusoidal frequencies. The remaining $M - p$ eigenvalues will lie at (or near) the origin.

One method for determining the power in the sinusoidal components is to solve the equation in (14.5.11) with $r_{yy}(m)$ substituted for $\gamma_{yy}(m)$.

Another method is based on the computation of the generalized eigenvectors $\{\mathbf{v}_i\}$ corresponding to the generalized eigenvalues $\{\lambda_i\}$. We have

$$(\mathbf{C}_{yy} - \lambda_i\mathbf{C}_{yz})\mathbf{v}_i = \mathbf{SP}(\mathbf{I} - \lambda_i\mathbf{\Phi}^H)\mathbf{S}^H\mathbf{v}_i = 0 \tag{14.5.43}$$

Since the column space of $(\mathbf{C}_{yy} - \lambda_i \cdot \mathbf{C}_{yz})$ is identical to the column space spanned by the vectors $\{\mathbf{s}_j, j \neq i\}$ given by (14.5.33), it follows that the generalized eigenvector $\mathbf{v}_i$ is orthogonal to $\mathbf{s}_j$, $j \neq i$. Since $\mathbf{P}$ is diagonal, it follows from (14.5.43) that the signal powers are

$$P_i = \frac{\mathbf{v}_i^H\mathbf{C}_{yy}\mathbf{v}_i}{|\mathbf{v}_i^H\mathbf{s}_i|^2}, \qquad i = 1, 2, \ldots, p \tag{14.5.44}$$

14.5.5 Order Selection Criteria

The eigenanalysis methods described in this section for estimating the frequencies and the powers of the sinusoids also provide information about the number of sinusoidal components. If there are p sinusoids, the eigenvalues associated with the signal subspace are $\{\lambda_i + \sigma_w^2, i = 1, 2 \ldots, p\}$ while the remaining $(M - p)$ eigenvalues are all equal to σ_w^2. Based on this eigenvalue decomposition, a test can be designed that compares the eigenvalues with a specified threshold. An alternative method also uses the eigenvector decomposition of the estimated autocorrelation matrix of the observed signal and is based on matrix perturbation analysis. This method is described in the paper by Fuchs (1988).

Another approach based on an extension and modification of the AIC criterion to the eigen-decomposition method has been proposed by Wax and Kailath (1985).

If the eigenvalues of the sample autocorrelation matrix are ranked so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M$, where $M > p$, the number of sinusoids in the signal subspace is estimated by selecting the minimum value of MDL(p), given as

$$\text{MDL}(p) = -\log\left[\frac{G(p)}{A(p)}\right]^N + E(p) \tag{14.5.45}$$

where

$$\begin{aligned} G(p) &= \prod_{i=p+1}^{M} \lambda_i, \qquad p = 0, 1, \ldots, M-1 \\ A(p) &= \left[\frac{1}{M-p}\sum_{i=p+1}^{M}\lambda_i\right]^{M-p} \\ E(p) &= \frac{1}{2}p(2M-p)\log N \end{aligned} \tag{14.5.46}$$

N : number of samples used to estimate the M autocorrelation lags

Some results on the quality of this order selection criterion are given in the paper by Wax and Kailath (1985). The MDL criterion is guaranteed to be consistent.

14.5.6 Experimental Results

In this section we illustrate with an example the resolution characteristics of the eigenanalysis-based spectral estimation algorithms and compare their performance with the model-based methods and nonparametric methods. The signal sequence is

$$x(n) = \sum_{i=1}^{4} A_i e^{j(2\pi f_i n + \phi_i)} + w(n)$$

where $A_i = 1$, $i = 1, 2, 3, 4$, $\{\phi_i\}$ are statistically independent random variables uniformly distributed on $(0, 2\pi)$, $\{w(n)\}$ is a zero-mean, white noise sequence with variance σ_w^2, and the frequencies are $f_1 = -0.222$, $f_2 = -0.166$, $f_3 = 0.10$, and $f_4 = 0.122$. The sequence $\{x(n), 0 \leq n \leq 1023\}$ is used to estimate the number of frequency components and the corresponding values of their frequencies for $\sigma_w^2 =$ 0.1, 0.5, 1.0, and $M = 12$ (length of the estimated autocorrelation).

Figures 14.5.1, 14.5.2, 14.5.3, and 14.5.4 illustrate the estimated power spectra of the signal using the Blackman–Tukey method, the minimum-variance method of Capon, the AR Yule–Walker method, and the MUSIC algorithm, respectively. The results from the ESPRIT algorithm are given in Table 14.2. From these results it is apparent that (1) the Blackman–Tukey method does not provide sufficient resolution to estimate the sinusoids from the data; (2) the minimum-variance method of Capon resolves only the frequencies f_1, f_2 but not f_3 and f_4; (3) the AR methods resolve all

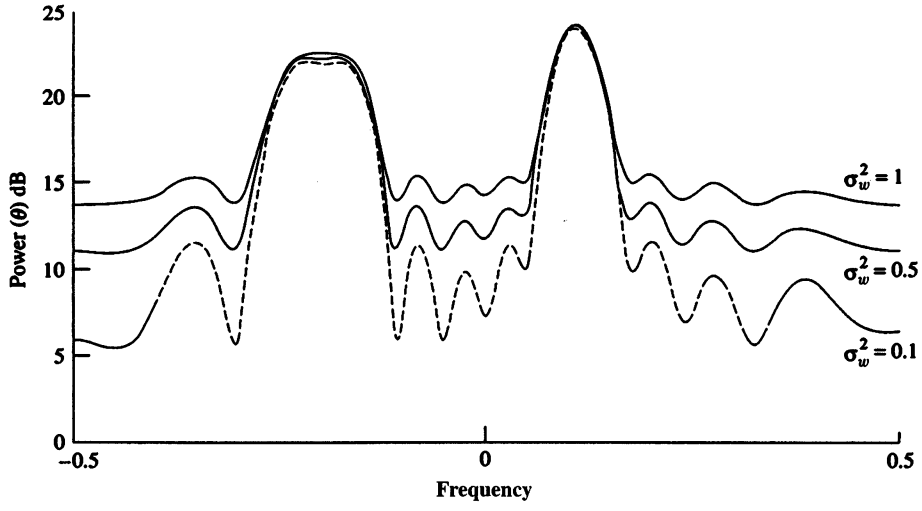


Figure 14.5.1 Power spectrum estimates from Blackman–Tukey method.

frequencies for $\sigma_w^2 = 0.1$ and $\sigma_w^2 = 0.5$; and (4) the MUSIC and ESPRIT algorithms not only recover all four sinusoids, but their performance for different values of σ_w^2 is essentially indistinguishable. We further observe that the resolution properties of the minimum variance method and the AR methods are functions of the noise variance. These results clearly demonstrate the power of the eigenanalysis-based algorithms in resolving sinusoids in additive noise.

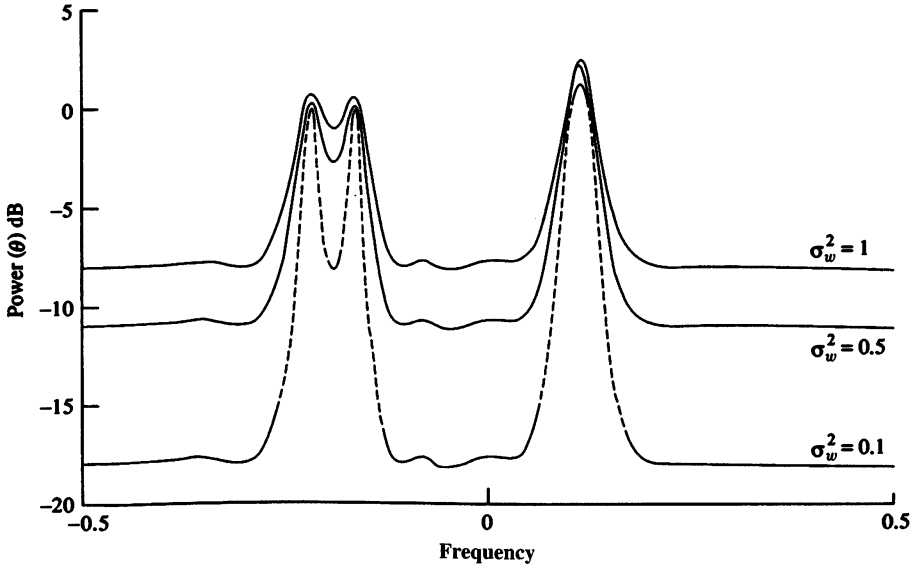


Figure 14.5.2 Power spectrum estimates from minimum-variance method.

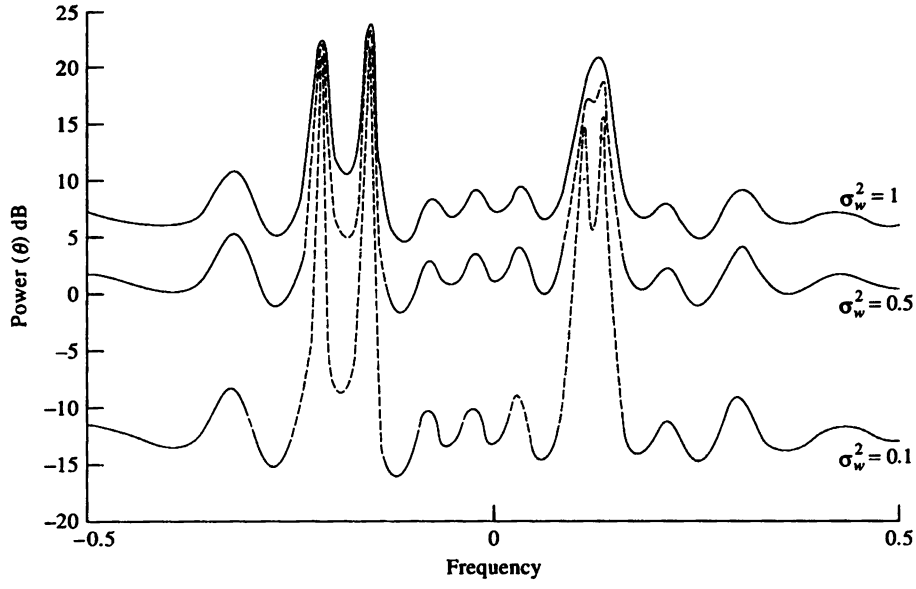


Figure 14.5.3 Power spectrum estimates from Yule–Walker AR method.

In conclusion, we should emphasize that the high-resolution, eigenanalysis-based spectral estimation methods described in this section, namely MUSIC and ESPRIT, are not only applicable to sinusoidal signals, but apply more generally to the estimation of narrowband signals.

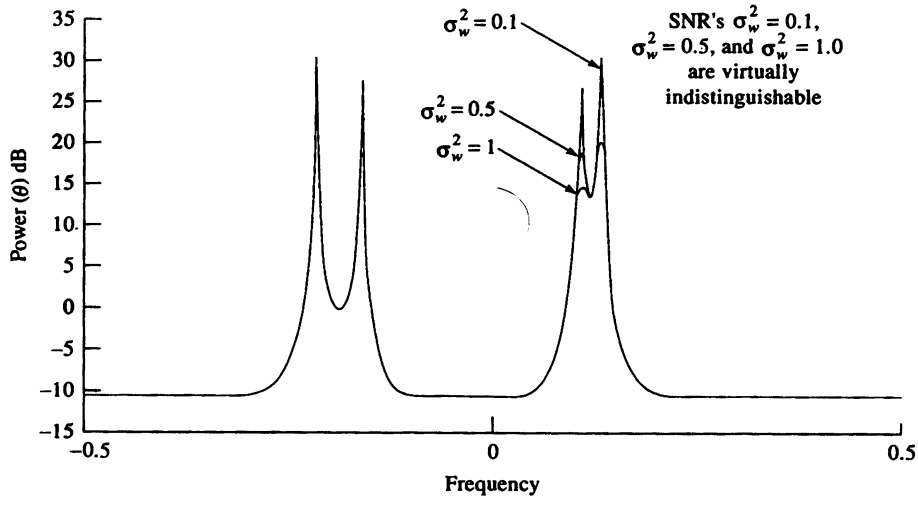


Figure 14.5.4 Power spectrum estimates from MUSIC algorithm.

TABLE 14.2 ESPRIT Algorithm

σ_w^2	$\hat{f}_1$	$\hat{f}_2$	$\hat{f}_3$	$\hat{f}_4$
0.1	−0.2227	−0.1668	−0.1224	−0.10071
0.5	−0.2219	−0.167	−0.121	0.0988
1.0	−0.222	−0.167	0.1199	0.1013
True values	−0.222	−0.166	0.122	0.100

14.6 Summary and References

Power spectrum estimation is one of the most important areas of research and applications in digital signal processing. In this chapter we have described the most important power spectrum estimation techniques and algorithms that have been developed over the past century, beginning with the nonparametric or classical methods based on the periodogram and concluding with the more modern parametric methods based on AR, MA, and ARMA linear models. Our treatment is limited in scope to single-time-series spectrum estimation methods, based on second moments (autocorrelation) of the statistical data.

The parametric and nonparametric methods that we described have been extended to multichannel and multidimensional spectrum estimation. The tutorial paper by McClellan (1982) treats the multidimensional spectrum estimation problem, while the paper by Johnson (1982) treats the multichannel spectrum estimation problem. Additional spectrum estimation methods have been developed for use with higher-order cumulants that involve the bispectrum and the trispectrum. A tutorial paper on these topics has been published by Nikias and Raghuveer (1987).

As evidenced from our previous discussion, power spectrum estimation is an area that has attracted many researchers and, as a result, thousands of papers have been published in the technical literature on this subject. Much of this work has been concerned with new algorithms and techniques, and modifications of existing techniques. Other work has been concerned with obtaining an understanding of the capabilities and limitations of the various power spectrum methods. In this context the statistical properties and limitations of the classical nonparametric methods have been thoroughly analyzed and are well understood. The parametric methods have also been investigated by many researchers, but the analysis of their performance is difficult and, consequently, fewer results are available. Some of the papers that have addressed the problem of performance characteristics of parametric methods are those of Kromer (1969), Lacoss (1971), Berk (1974), Baggeroer (1976), Sakai (1979), Swingler (1980), Lang and McClellan (1980), and Tufts and Kumaresan (1982).

In addition to the references already given in this chapter on the various methods for spectrum estimation and their performance, we should include for reference some of the tutorial and survey papers. In particular, we cite the tutorial paper by Kay and Marple (1981), which includes about 280 references, the paper by Brillinger (1974), and the Special Issue on Spectral Estimation of the *IEEE Proceedings*, September

1982. Another indication of the widespread interest in the subject of spectrum estimation and analysis is the publication of texts by Gardner (1987), Kay (1988), and Marple (1987), and the IEEE books edited by Childers (1978) and Kesler (1986).

Many computer programs as well as software packages that implement the various spectrum estimation methods described in this chapter are available. One software package is available through the IEEE (*Programs for Digital Signal Processing*. IEEE Press, 1979); others are available commercially.

Problems

14.1 **(a)** By expanding (14.1.23), taking the expected value, and finally taking the limit as $T_0 \to \infty$, show that the right-hand side converges to $\Gamma_{xx}(F)$.

(b) Prove that

$$\sum_{m=-N}^{N} r_{xx}(m)e^{-j2\pi fm} = \frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-j2\pi fn}\right|^2$$

14.2 For zero-mean, jointly Gaussian random variables, X_1, X_2, X_3, X_4, it is well known [see Papoulis (1984)] that

$$E(X_1X_2X_3X_4) = E(X_1X_2)E(X_3X_4) + E(X_1X_3)E(X_2X_4) + E(X_1X_4)E(X_2X_3)$$

Use this result to derive the mean-square value of $r'_{xx}(m)$, given by (14.1.27) and the variance, which is

$$\text{var}[r'_{xx}(m)] = E[|r'_{xx}(m)|^2] - |E[r'_{xx}(m)]|^2$$

14.3 By use of the expression for the fourth joint moment for Gaussian random variables, show that

(a) $E[P_{xx}(f_1)P_{xx}(f_2)] = \sigma_x^4\left\{1 + \left[\dfrac{\sin\pi(f_1+f_2)N}{N\sin\pi(f_1+f_2)}\right]^2 + \left[\dfrac{\sin\pi(f_1-f_2)N}{N\sin\pi(f_1-f_2)}\right]^2\right\}$

(b) $\text{cov}[P_{xx}(f_1)P_{xx}(f_2)] = \sigma_x^4\left\{\left[\dfrac{\sin\pi(f_1+f_2)N}{N\sin\pi(f_1+f_2)}\right]^2 + \left[\dfrac{\sin\pi(f_1-f_2)N}{N\sin\pi(f_1-f_2)}\right]^2\right\}$

(c) $\text{var}[P_{xx}(f)] = \sigma_x^4\left\{1 + \left(\dfrac{\sin 2\pi fN}{N\sin 2\pi f}\right)^2\right\}$ under the condition that the sequence $x(n)$ is a zero-mean white Gaussian noise sequence with variance σ_x^2.

14.4 Generalize the results in Problem 14.3 to a zero-mean Gaussian noise process with power density spectrum $\Gamma_{xx}(f)$. Then derive the variance of the periodogram $P_{xx}(f)$, as given by (14.1.38). (*Hint:* Assume that the colored Gaussian noise process is the output of a linear system excited by white Gaussian noise. Then use the appropriate relations given in Section 12.1.)

14.5 Show that the periodogram values at frequencies $f_k = k/L$, $k = 0, 1, \ldots, L-1$, given by (14.1.41) can be computed by passing the sequence through a bank of N IIR filters, where each filter has an impulse response

$$h_k(n) = e^{-j2\pi nk/N}u(n)$$

and then compute the magnitude-squared value of the filter outputs at $n = N$. Note that each filter has a pole on the unit circle at the frequency f_k.

14.6 Prove that the normalization factor given by (14.2.12) ensures that (14.2.19) is satisfied.

14.7 Let us consider the use of the DFT (computed via the FFT algorithm) to compute the autocorrelation of the complex-valued sequence $x(n)$, that is,

$$r_{xx}(m) = \frac{1}{N}\sum_{n=0}^{N-m-1} x^*(n)x(n+m), \qquad m \geq 0$$

Suppose the size M of the FFT is much smaller than that of the data length N. Specifically, assume that $N = KM$.

(a) Determine the steps needed to section $x(n)$ and compute $r_{xx}(m)$ for $-(M/2)+1 \leq m \leq (M/2)-1$, by using $4K$ M-point DFTs and one M-point IDFT.

(b) Now consider the following three sequences $x_1(n)$, $x_2(n)$, and $x_3(n)$, each of duration M. Let the sequences $x_1(n)$ and $x_2(n)$ have arbitrary values in the range $0 \leq n \leq (M/2)-1$, but be zero for $(M/2) \leq n \leq M-1$. The sequence $x_3(n)$ is defined as

$$x_3(n) = \begin{cases} x_1(n), & 0 \leq \dfrac{M}{2} - 1 \\ x_2\left(n - \dfrac{M}{2}\right), & \dfrac{M}{2} \leq n \leq M-1 \end{cases}$$

Determine a simple relationship among the M-point DFTs $X_1(k)$, $X_2(k)$, and $X_3(k)$.

(c) By using the result in part (b), show how the computation of the DFTs in part (a) can be reduced in number from $4K$ to $2K$.

14.8 The Bartlett method is used to estimate the power spectrum of a signal $x(n)$. We know that the power spectrum consists of a single peak with a 3-dB bandwidth of 0.01 cycle per sample, but we do not know the location of the peak.

(a) Assuming that N is large, determine the value of $M = N/K$ so that the spectral window is narrower than the peak.

(b) Explain why it is not advantageous to increase M beyond the value obtained in part (a).

14.9 Suppose we have $N = 1000$ samples from a sample sequence of a random process.

(a) Determine the frequency resolution of the Bartlett, Welch (50% overlap), and Blackman–Tukey methods for a quality factor $Q = 10$.

(b) Determine the record lengths (M) for the Bartlett, Welch (50% overlap), and Blackman–Tukey methods.

14.10 Consider the problem of continuously estimating the power spectrum from a sequence $x(n)$ based on averaging periodograms with exponential weighting into the past. Thus with $P_{xx}^{(0)}(f) = 0$, we have

$$P_{xx}^{(m)}(f) = wP_{xx}^{(m-1)}(f) + \frac{1-w}{M}\left|\sum_{n=0}^{M-1} x_m(n)e^{-j2\pi fn}\right|^2$$

where successive periodograms are assumed to be uncorrelated and w is the (exponential) weighting factor.

(a) Determine the mean and variance of $P_{xx}^{(m)}(f)$ for a Gaussian random process.

(b) Repeat the analysis of part (a) for the case in which the modified periodogram defined by Welch is used in the averaging with no overlap.

14.11 The periodogram in the Bartlett method can be expressed as

$$P_{xx}^{(i)}(f) = \sum_{m=-(M-1)}^{M-1}\left(1 - \frac{|m|}{M}\right)r_{xx}^{(i)}(m)e^{-j2\pi fm}$$

where $r_{xx}^{(i)}(m)$ is the estimated autocorrelation sequence obtained from the ith block of data. Show that $P_{xx}^{(i)}(f)$ can be expressed as

$$P_{xx}^{(i)}(f) = \mathbf{E}^H(f)\mathbf{R}_{xx}^{(i)}\mathbf{E}(f)$$

where

$$\mathbf{E}(f) = [1 \quad e^{j2\pi f} \quad e^{j4\pi f} \quad \cdots \quad e^{j2\pi(M-1)f}]^t$$

and therefore,

$$P_{xx}^{B}(f) = \frac{1}{K}\sum_{k=1}^{K}\mathbf{E}^H(f)\mathbf{R}_{xx}^{(k)}\mathbf{E}(f)$$

14.12 Derive the recursive order-update equation given in (14.3.19).

14.13 Determine the mean and the autocorrelation of the sequence $x(n)$, which is the output of an ARMA (1, 1) process described by the difference equation

$$x(n) = \tfrac{1}{2}x(n-1) + w(n) - w(n-1)$$

where $w(n)$ is a white noise process with variance σ_w^2.

14.14 Determine the mean and the autocorrelation of the sequence $x(n)$ generated by the MA(2) process described by the difference equation

$$x(n) = w(n) - 2w(n-1) + w(n-2)$$

where $w(n)$ is a white noise process with variance σ_w^2.

14.15 An MA(2) process has the autocorrelation sequence

$$\gamma_{xx}(m) = \begin{cases} 6\sigma_w^2, & m = 0 \\ -4\sigma_w^2, & m = \pm 1 \\ -2\sigma_w^2, & m = \pm 2 \\ 0, & \text{otherwise} \end{cases}$$

(a) Determine the coefficients of the MA(2) process that have the foregoing autocorrelation.

(b) Is the solution unique? If not, give all the possible solutions.

14.16 An MA(2) process has the autocorrelation sequence

$$\gamma_{xx}(m) = \begin{cases} \sigma_w^2, & m = 0 \\ -\dfrac{35}{62}\sigma_w^2, & m = \pm 1 \\ \dfrac{6}{62}\sigma_w^2, & m = \pm 2 \end{cases}$$

(a) Determine the coefficients of the minimum-phase system for the MA(2) process.

(b) Determine the coefficients of the maximum-phase system for the MA(2) process.

(c) Determine the coefficients of the mixed-phase system for the MA(2) process.

14.17 Consider the linear system described by the difference equation

$$y(n) = 0.8y(n-1) + x(n) + x(n-1)$$

where $x(n)$ is a wide-sense stationary random process with zero mean and autocorrelation

$$\gamma_{xx}(m) = \left(\tfrac{1}{2}\right)^{|m|}$$

(a) Determine the power density spectrum of the output $y(n)$.

(b) Determine the autocorrelation $\gamma_{yy}(m)$ of the output.

(c) Determine the variance σ_y^2 of the output.

14.18 From (14.3.6) and (14.3.9) we note that an AR(p) stationary random process satisfies the equation

$$\gamma_{xx}(m) + \sum_{k=1}^{p} a_p(k)\gamma_{xx}(m-k) = \begin{cases} \sigma_w^2, & m = 0, \\ 0, & 1 \le m \le p, \end{cases}$$

where $a_p(k)$ are the prediction coefficients of the linear predictor of order p and σ_w^2 is the minimum mean-square prediction error. If the $(p+1)\times(p+1)$ autocorrelation matrix $\mathbf{\Gamma}_{xx}$ in (14.3.9) is positive definite, prove that:

(a) The reflection coefficients $|K_m| < 1$ for $1 \leq m \leq p$.

(b) The polynomial

$$A_p(z) = 1 + \sum_{k=1}^{p} a_p(k) z^{-k}$$

has all its roots inside the unit circle (i.e., it is minimum phase).

14.19 An AR(2) process is described by the difference equation

$$x(n) = 0.81x(n-2) + w(n)$$

where $w(n)$ is a white noise process with variance σ_w^2.

(a) Determine the parameters of the MA(2), MA(4), and MA(8) models which provide a minimum mean-square error fit to the data $x(n)$.

(b) Plot the true spectrum and those of the MA(q), $q = 2, 4, 8$, spectra and compare the results. Comment on how well the MA(q) models approximate the AR(2) process.

14.20 An MA(2) process is described by the difference equation

$$x(n) = w(n) + 0.81w(n-2)$$

where $w(n)$ is a white noise process with variance σ_w^2.

(a) Determine the parameters of the AR(2), AR(4), and AR(8) models that provide a minimum mean-square error fit to the data $x(n)$.

(b) Plot the true spectra and those of the AR(p), $p = 2, 4, 8$, and compare the results. Comment on how well the AR(p) models approximate the MA(2) process.

14.21 **(a)** Determine the power spectra for the random processes generated by the following difference equations.

1. $x(n) = -0.81x(n-2) + w(n) - w(n-1)$
2. $x(n) = w(n) - w(n-2)$
3. $x(n) = -0.81x(n-2) + w(n)$

where $w(n)$ is a white noise process with variance σ_w^2.

(b) Sketch the spectra for the processes given in part (a).

(c) Determine the autocorrelation $\gamma_{xx}(m)$ for the processes in (2) and (3).

14.22 The Bartlett method is used to estimate the power spectrum of a signal from a sequence $x(n)$ consisting of $N = 2400$ samples.

(a) Determine the smallest length M of each segment in the Bartlett method that yields a frequency resolution of $\Delta f = 0.01$.

(b) Repeat part (a) for $\Delta f = 0.02$.

(c) Determine the quality factors Q_B for parts (a) and (b).

14.23 A random process $x(n)$ is characterized by the power density spectrum

$$\Gamma_{xx}(f) = \sigma_w^2 \frac{|e^{j2\pi f} - 0.9|^2}{|e^{j2\pi f} - j0.9|^2 |e^{j2\pi f} + j0.9|^2}$$

where σ_w^2 is a constant (scale factor).

(a) If we view $\Gamma_{xx}(f)$ as the power spectrum at the output of a linear pole–zero system $H(z)$ driven by white noise, determine $H(z)$.

(b) Determine the system function of a stable system (noise-whitening filter) that produces a white noise output when excited by $x(n)$.

14.24 The N-point DFT of a random sequence $x(n)$ is

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N}$$

Assume that $E[x(n)] = 0$ and $E[x(n)x(n+m)] = \sigma_x^2 \delta(m)$ [i.e., $x(n)$ is a white noise process].

(a) Determine the variance of $X(k)$.

(b) Determine the autocorrelation of $X(k)$.

14.25 Suppose that we represent an ARMA(p, q) process as a cascade of an MA(q) followed by an AR(p) model. The input–output equation for the MA(q) model is

$$v(n) = \sum_{k=0}^{q} b_k w(n-k)$$

where $w(n)$ is a white noise process. The input–output equation for the AR(p) model is

$$x(n) + \sum_{k=1}^{p} a_k x(n-k) = v(n)$$

(a) By computing the autocorrelation of $v(n)$, show that

$$\gamma_{vv}(m) = \sigma_w^2 \sum_{k=0}^{q-m} b_k^* b_{k+m}$$

(b) Show that

$$\gamma_{vv}(m) = \sum_{k=0}^{p} a_k \gamma_{vx}(m+k), \qquad a_o = 1$$

where $\gamma_{vx}(m) = E[v(n+m)x^*(n)]$.

14.26 Determine the autocorrelation $\gamma_{xx}(m)$ of the random sequence

$$x(n) = A\cos(\omega_1 n + \phi)$$

where the amplitude A and the frequency ω_1 are (known) constants and ϕ is a uniformly distributed random phase over the interval $(0, 2\pi)$.

14.27 Suppose that the AR(2) process in Problem 14.19 is corrupted by an additive white noise process $v(n)$ with variance σ_v^2. Thus we have

$$y(n) = x(n) + v(n)$$

(a) Determine the difference equation for $y(n)$ and thus demonstrate that $y(n)$ is an ARMA(2, 2) process. Determine the coefficients of the ARMA process.

(b) Generalize the result in part (a) to an AR(p) process

$$x(n) = -\sum_{k=1}^{p} a_k(xn - k) + w(n)$$

and

$$y(n) = x(n) + v(n)$$

14.28 **(a)** Determine the autocorrelation of the random sequence

$$x(n) = \sum_{k=1}^{K} A_k \cos(\omega_k n + \phi_k) + w(n)$$

where $\{A_k\}$ are constant amplitudes, $\{\omega_k\}$ are constant frequencies, and $\{\phi_k\}$ are mutually statistically independent and uniformly distributed random phases. The noise sequence $w(n)$ is white with variance σ_w^2.

(b) Determine the power density spectrum of $x(n)$.

14.29 The harmonic decomposition problem considered by Pisarenko can be expressed as the solution to the equation

$$\mathbf{a}^H \mathbf{\Gamma}_{yy} \mathbf{a} = \sigma_w^2 \mathbf{a}^H \mathbf{a}$$

The solution for $\mathbf{a}$ can be obtained by minimizing the quadratic form $\mathbf{a}^H \mathbf{\Gamma}_{yy} \mathbf{a}$ subject to the constraint that $\mathbf{a}^H \mathbf{a} = 1$. The constraint can be incorporated into the performance index by means of a Lagrange multiplier. Thus the performance index becomes

$$\mathcal{E} = \mathbf{a}^H \mathbf{\Gamma}_{yy} \mathbf{a} + \lambda(1 - \mathbf{a}^H \mathbf{a})$$

By minimizing $\mathcal{E}$ with respect to $\mathbf{a}$, show that this formulation is equivalent to the Pisarenko eigenvalue problem given in (14.5.9) with the Lagrange multiplier playing the role of the eigenvalue. Thus show that the minimum of $\mathcal{E}$ is the minimum eigenvalue σ_w^2.

14.30 The autocorrelation of a sequence consisting of a sinusoid with random phase in noise is

$$\gamma_{xx}(m) = P\cos 2\pi f_1 m + \sigma_w^2\delta(m)$$

where f_1 is the frequency of the sinusoidal, P is its power, and σ_w^2 is the variance of the noise. Suppose that we attempt to fit an AR(2) model to the data.

(a) Determine the optimum coefficients of the AR(2) model as a function of σ_w^2 and f_1.

(b) Determine the reflection coefficients K_1 and K_2 corresponding to the AR(2) model parameters.

(c) Determine the limiting values of the AR(2) parameters and (K_1, K_2) as $\sigma_w^2 \to 0$.

14.31 The minimum variance power spectrum estimate described in Section 14.4 is determined by minimizing the variance

$$\sigma_y^2 = \mathbf{h}^H \mathbf{\Gamma}_{xx}\mathbf{h}$$

subject to the constraint

$$\mathbf{E}^H(f)\mathbf{h} = 1$$

where $\mathbf{E}(f)$ is defined as the vector

$$\mathbf{E}^t(f) = [1 \quad e^{j2\pi f} \quad e^{j4\pi f} \quad \cdots \quad e^{j2\pi pf}]$$

To determine the optimum filter that minimizes σ_y^2, let us define the function

$$\mathcal{E}(\mathbf{h}) = \mathbf{h}^H\mathbf{\Gamma}_{xx}\mathbf{h} + \mu(1 - \mathbf{E}^H(f)\mathbf{h}) + \mu^*(1 - \mathbf{h}^H\mathbf{E}(f))$$

where μ is a Lagrange multiplier

(a) By differentiating $\mathcal{E}(\mathbf{h})$ and setting the derivative to zero, show that

$$\mathbf{h}_{\text{opt}} = \mu^*\mathbf{\Gamma}_{xx}^{-1}\mathbf{E}(f)$$

(b) Solve for μ^* by using the constraint and, thus, show that

$$\mathbf{h}_{\text{opt}} = \frac{\mathbf{\Gamma}_{xx}^{-1}\mathbf{E}(f)}{\mathbf{E}^H(f)\mathbf{\Gamma}_{xx}^{-1}\mathbf{E}(f)}$$

14.32 The periodogram spectral estimate is expressed as

$$P_{xx}(f) = \frac{1}{N}|X(f)|^2$$

where

$$X(f) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi fn}$$

$$= \mathbf{E}^H(f)\mathbf{X}(n)$$

$\mathbf{X}(f)$ is the data vector and $\mathbf{E}(f)$ is defined as

$$E^t(f) = [1 \quad e^{j2\pi f} \quad e^{j4\pi f} \quad \cdots \quad e^{j2\pi f(N-1)}]$$

Show that

$$E[P_{xx}(f)] = \frac{1}{N}\mathbf{E}^H(f)\mathbf{\Gamma}_{xx}\mathbf{E}(f)$$

where $\mathbf{\Gamma}_{xx}$ is the autocorrelation matrix of the data vector $\mathbf{X}(n)$.

14.33 Determine the frequency and power of a single real sinusoid in white noise. The signal and noise correlation function is given as

$$\gamma_{yy}(m) = \begin{cases} 3, & m = 0 \\ 0, & m = 1 \\ -2, & m = 2 \\ 0, & |m| > 2 \end{cases}$$

14.34 The signal $y(n)$ consists of complex exponentials in white noise. Its autocorrelation matrix is

$$\Gamma_{yy} = \begin{bmatrix} 2 & -j & -1 \\ j & 2 & -j \\ -1 & j & 2 \end{bmatrix}$$

Use the MUSIC algorithm to determine the frequencies of the exponentials and their power levels.

14.35 Show that $P_{\text{MUSIC}}(f)$ can be expressed as

$$P_{\text{MUSIC}}(f) = \frac{1}{\mathbf{s}^H(f)\left(\sum_{k=p+1}^{M} \boldsymbol{v}_k \boldsymbol{v}_k^H\right)\mathbf{s}(f)}$$

14.36 *Root MUSIC algorithm* Let us define the noise subspace polynomials as

$$V_k(z) = \sum_{n=0}^{M-1} v_k(n+1)z^{-n}, \quad k = p+1, \ldots, M$$

where $v_k(n)$ are the elements of the noise subspace vector $\boldsymbol{v}_k$.

(a) Show that $P_{\text{MUSIC}}(f)$ can be expressed as

$$P_{\text{MUSIC}}(f) = \frac{1}{\sum_{k=p+1}^{M} V_k(f)V_k^*(f)}$$

$$= \frac{1}{\sum_{k=p+1}^{M} V_k(z)V_k^*\left(\frac{1}{z^*}\right)\Big|_{z=e^{j2\pi f}}}$$

(b) For p complex sinusoids in white noise, the polynomial

$$Q(z) = \sum_{k=p+1}^{M} V_k(z) V_k^*\left(\frac{1}{z^*}\right)$$

goes to zero at $z = e^{j2\pi f_i}$, $i = 1, 2, \ldots, p$. Hence, this polynomial has p roots on the unit circle in the z-plane, where each root is a double root. The determination of the frequencies by factoring the polynomial $Q(z)$ is called the *root MUSIC method.* For the correlation matrix given in Problem 14.34, determine the frequencies of the exponentials by the root MUSIC method. The extra roots obtained by this method are spurious roots and may be discarded.

14.37 This problem involves the use of crosscorrelation to detect a signal in noise and estimate the time delay in the signal. A signal $x(n)$ consists of a pulsed sinusoid corrupted by a stationary zero-mean white noise sequence. That is,

$$x(n) = y(n - n_0) + w(n), \qquad 0 \le n \le N - 1$$

where $w(n)$ is the noise with variance σ_w^2 and the signal is

$$\begin{aligned} y(n) &= A\cos\omega_0 n, && 0 \le n \le M - 1 \\ &= 0, && \text{otherwise} \end{aligned}$$

The frequency ω_0 is known but the delay n_0, which is a positive integer, is unknown, and is to be determined by crosscorrelating $x(n)$ with $y(n)$. Assume that $N > M + n_0$. Let

$$r_{xy}(m) = \sum_{n=0}^{N-1} y(n - m)x(n)$$

denote the crosscorrelation sequence between $x(n)$ and $y(n)$. In the absence of noise this function exhibits a peak at delay $m = n_0$. Thus n_0 is determined with no error. The presence of noise can lead to errors in determining the unknown delay.

(a) For $m = n_0$, determine $E[r_{xy}(n_0)]$. Also, determine the variance, $\text{var}[r_{xy}(n_0)]$, due to the presence of the noise. In both calculations, assume that the double frequency term averages to zero. That is, $M \gg 2\pi/\omega_0$.

(b) Determine the signal-to-noise ratio, defined as

$$\text{SNR} = \frac{\{E[r_{xy}(n_0)]\}^2}{\text{var}[r_{xy}(n_0)]}$$

(c) What is the effect of the pulse duration M on the SNR?

14.38 Generate 100 samples of a zero-mean white noise sequence $w(n)$ with variance $\sigma_w^2 = \frac{1}{12}$, by using a uniform random number generator.

(a) Compute the autocorrelation of $w(n)$ for $0 \leq m \leq 15$.

(b) Compute the periodogram estimate $P_{xx}(f)$ and plot it.

(c) Generate 10 different realizations of $w(n)$ and compute the corresponding sample autocorrelation sequences $r_k(m)$, $1 \leq k \leq 10$ and $0 \leq m \leq 15$.

(d) Compute and plot the average autocorrelation sequence for part (c):

$$r_{\text{av}}(m) = \frac{1}{10} \sum_{k=1}^{10} r_k(m)$$

and the periodogram corresponding to $r_{av}(m)$.

(e) Comment on the results in parts (a) through (d).

14.39 A random signal is generated by passing zero-mean white Gaussian noise with unit variance through a filter with system function

$$H(z) = \frac{1}{(1 + az^{-1} + 0.99z^{-2})(1 - az^{-1} + 0.98z^{-2})}$$

(a) Sketch a typical plot of the theoretical power spectrum $\Gamma_{xx}(f)$ for a small value of the parameter a (i.e., $0 < a < 0.1$). Pay careful attention to the value of the two spectral peaks and the value of $P_{xx}(\omega)$ for $\omega = \pi/2$.

(b) Let $a = 0.1$. Determine the section length M required to resolve the spectral peaks of $\Gamma_{xx}(f)$ when using Bartlett's method.

(c) Consider the Blackman–Tukey method of smoothing the periodogram. How many lags of the correlation estimate must be used to obtain resolution comparable to that of the Bartlett estimate considered in part (b)? How many data points must be used if the variance of the estimate is to be comparable to that of a four-section Bartlett estimate?

(d) For $a = 0.05$, fit an AR(4) model to 100 samples of the data based on the Yule–Walker method and plot the power spectrum. Avoid transient effects by discarding the first 200 samples of the data.

(e) Repeat part (d) with the Burg method.

(f) Repeat parts (d) and (e) for 50 data samples and comment on similarities and differences in the results.

APPENDIX

A

Random Number Generators

In some of the examples given in the text, random numbers are generated to simulate the effect of noise on signals and to illustrate how the method of correlation can be used to detect the presence of a signal buried in noise. In the case of periodic signals, the correlation technique also allowed us to estimate the period of the signal.

In practice, random number generators are often used to simulate the effect of noiselike signals and other random phenomena encountered in the physical world. Such noise is present in electronic devices and systems and usually limits our ability to communicate over large distances and to be able to detect relatively weak signals. By generating such noise on a computer, we are able to study its effects through simulation of communication systems, radar detection systems, and the like and to assess the performance of such systems in the presence of noise.

Most computer software libraries include a uniform random number generator. Such a random number generator generates a number between zero and 1 with equal probability. We call the output of the random number generator a random variable. If A denotes such a random variable, its range is the interval $0 \leq A \leq 1$.

We know that the numerical output of a digital computer has limited precision, and as a consequence, it is impossible to represent the continuum of numbers in the interval $0 \leq A \leq 1$. However, we can assume that our computer represents each output by a large number of bits in either fixed point or floating point. Consequently, for all practical purposes, the number of outputs in the interval $0 \leq A \leq 1$ is sufficiently large, so that we are justified in assuming that any value in the interval is a possible output from the generator.

The uniform probability density function for the random variable A, denoted as $p(A)$, is illustrated in Fig. A.1(a). We note that the average value or mean value of A, denoted as m_A, is $m_A = \frac{1}{2}$. The integral of the probability density function, which

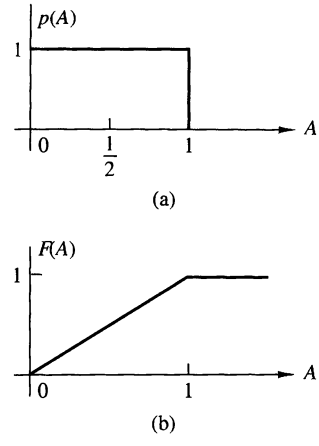


Figure A.1

represents the area under $p(A)$, is called the probability distribution function of the random variable A and is defined as

$$F(A) = \int_{-\infty}^{A} p(x)dx \tag{A.1}$$

For any random variable, this area must always be unity, which is the maximum value that can be achieved by a distribution function. Hence

$$F(1) = \int_{-\infty}^{1} p(x)dx = 1 \tag{A.2}$$

and the range of $F(A)$ is $0 \leq F(A) \leq 1$ for $0 \leq A \leq 1$.

If we wish to generate uniformly distributed noise in an interval $(b, b+1)$ it can simply be accomplished by using the output A of the random number generator and shifting it by an amount b. Thus a new random variable B can be defined as

$$B = A + b \tag{A.3}$$

which now has a mean value $m_B = b + \frac{1}{2}$. For example, if $b = -\frac{1}{2}$, the random variable B is uniformly distributed in the interval $(-\frac{1}{2}, \frac{1}{2})$, as shown in Fig A.2(a). Its probability distribution function $F(B)$ is shown in Fig A.2(b).

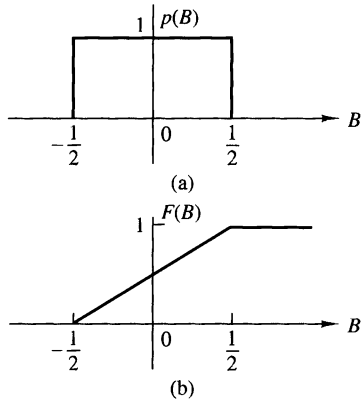


Figure A.2

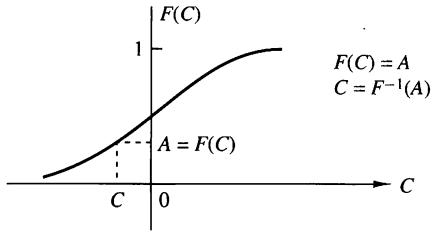


Figure A.3

A uniformly distributed random variable in the range (0, 1) can be used to generate random variables with other probability distribution functions. For example, suppose that we wish to generate a random variable C with probability distribution function $F(C)$, as illustrated in Fig A.3. Since the range of $F(C)$ is the interval (0, 1), we begin by generating a uniformly distributed random variable A in the range (0, 1). If we set

$$F(C) = A \tag{A.4}$$

then

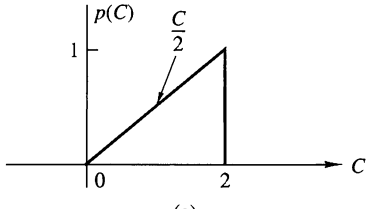
$$C = F^{-1}(A) \tag{A.5}$$

Thus we solve (A.4) for C, and the solution in (A.5) provides the value of C for which $F(C) = A$. By this means we obtain a new random variable C with probability distribution $F(C)$. This inverse mapping from A to C is illustrated in Fig A.3.

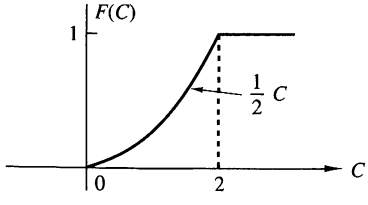
EXAMPLE A.1

Generate a random variable C that has the linear probability density function shown in Fig A.4(a), that is,

$$p(C) = \begin{cases} \dfrac{C}{2}, & 0 \le C \le 2 \\ 0, & \text{otherwise} \end{cases}$$



(a)



(b)

Figure A.4

Solution. This random variable has a probability distribution function

$$F(C) = \begin{cases} 0, & C < 0 \\ \frac{1}{4}C^2, & 0 \le C \le 2 \\ 1, & C > 2 \end{cases}$$

which is illustrated in Fig A.4(b). We generate a uniformly distributed random variable A and set $F(C) = A$. Hence

$$F(C) = \tfrac{1}{4}C^2 = A$$

Upon solving for C, we obtain

$$C = 2\sqrt{A}$$

Thus we generate a random variable C with probability function $F(C)$, as shown in Fig A.4(b).

In Example A.1 the inverse mapping $C = F^{-1}(A)$ was simple. In some cases it is not. This problem arises in trying to generate random numbers that have a normal distribution function.

Noise encountered in physical systems is often characterized by the normal or Gaussian probability distribution, which is illustrated in Fig A.5. The probability density function is given by

$$p(C) = \frac{1}{\sqrt{2\pi}\sigma} e^{-C^2/2\sigma^2}, \qquad -\infty < C < \infty \tag{A.6}$$

where σ^2 is the variance of C, which is a measure of the spread of the probability density function $p(C)$. The probability distribution function $F(C)$ is the area under $p(C)$ over the range $(-\infty, C)$. Thus

$$F(C) = \int_{-\infty}^{C} p(x)dx \tag{A.7}$$

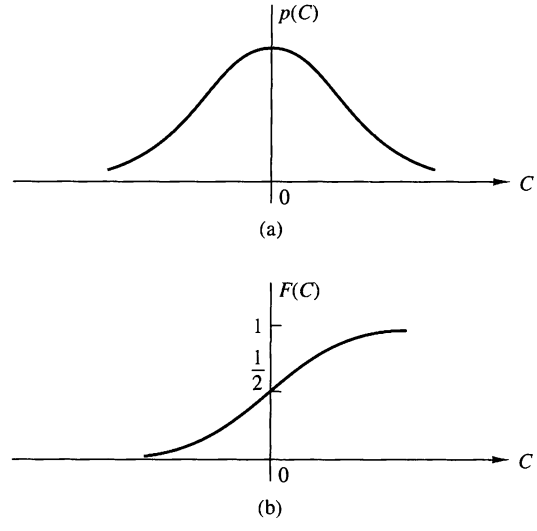


Figure A.5

Unfortunately, the integral in (A.7) cannot be expressed in terms of simple functions. Consequently, the inverse mapping is difficult to achieve.

A way has been found to circumvent this problem. From probability theory it is known that a (Rayleigh distributed) random variable R, with probability distribution function

$$F(R) = \begin{cases} 0, & R < 0 \\ 1 - e^{-R^2/2\sigma^2}, & R \geq 0 \end{cases} \tag{A.8}$$

is related to a pair of Gaussian random variables C and D, through the transformation

$$C = R\cos\Theta \tag{A.9}$$

$$D = R\sin\Theta \tag{A.10}$$

where Θ is a uniformly distributed variable in the interval $(0, 2\pi)$. The parameter σ^2 is the variance of C and D. Since (A.8) is easily inverted, we have

$$F(R) = 1 - e^{-R^2/2\sigma^2} = A$$

and hence

$$R = \sqrt{2\sigma^2 \ln[1/(1-A)]} \tag{A.11}$$

where A is a uniformly distributed random variable in the interval $(0, 1)$. Now if we generate a second uniformly distributed random variable B and define

$$\Theta = 2\pi B \tag{A.12}$$

then from (A.9) and (A.10), we obtain two statistically independent Gaussian distributed random variables C and D.

```
C      SUBROUTINE GAUSS CONVERTS A UNIFORM RANDOM
C      SEQUENCE XIN IN [0,1] TO A GAUSSIAN RANDOM
C      SEQUENCE WITH G(0,SIGMA**2)
C      PARAMETERS   :
C            XIN    :UNIFORM IN [0,1] RANDOM NUMBER
C              B    :UNIFORM IN [0,1] RANDOM NUMBER
C          SIGMA    :STANDARD DEVIATION OF THE GAUSSIAN
C           YOUT    :OUTPUT FROM THE GENERATOR
C
       SUBROUTINE GAUSS 9XIN,B,SIGMA,YOUT)
       PI=4.0*ATAN (1.0)
       B=2.0*PI*B
       R=SQRT (2.0*(SIGMA**2)*ALOG(1.0/(1.0-XIN)))
       YOUT=R*COS(B)
       RETURN
       END
C      NOTE: TO USE THE ABOVE SUBROUTINE FOR A
C            GAUSSIAN RANDOM NUMBER GENERATOR
C            YOU MUST PROVIDE AS INPUT TWO UNIFORM RANDOM NUMBERS
C            XIN AND B
C            XIN AND B MUST BE STATISTICALLY INDEPENDENT
C
```

Figure A.6 Subroutine for generating Gaussian random variables

The method described above is often used in practice to generate Gaussian distributed random variables. As shown in Fig A.5, these random variables have a mean value of zero and a variance σ^2. If a nonzero mean Gaussian random variable is desired, then C and D can be translated by the addition of the mean value.

A subroutine implementing this method for generating Gaussian distributed random variables is given in Fig A.6.

APPENDIX

B

Tables of Transition Coefficients for the Design of Linear-Phase FIR Filters

In Section 10.2.3 we described a design method for linear-phase FIR filters that involved the specification of $H_r(\omega)$ at a set of equally spaced frequencies $\omega_k = 2\pi(k + \alpha)/M$, where $\alpha = 0$ or $\alpha = \frac{1}{2}$, $k = 0, 1, \ldots, (M - 1)/2$ for M odd and $k = 0, 1, 2, \ldots, (M/2) - 1$ for M even, where M is the length of the filter. Within the passband of the filter, we select $H_r(\omega_k) = 1$, and in the stopband, $H_r(\omega_k) = 0$. For frequencies in the transition band, the values of $H_r(\omega_k)$ are optimized to minimize the maximum sidelobe in the stopband. This is called a *minimax optimization criterion.*

The optimization of the values of $H_r(\omega)$ in the transition band has been performed by Rabiner et al. (1970) and tables of transition values have been provided in the published paper. A selected number of the tables for lowpass FIR filters are included in this appendix.

Four tables are given. Table B.1 lists the transition coefficients for the case $\alpha = 0$ and one coefficient in the transition band for both M odd and M even. Table B.2 lists the transition coefficients for the case $\alpha = 0$, and two coefficients in the transition band for M odd and M even. Table B.3 lists the transition coefficients for the case $\alpha = \frac{1}{2}$, M even and one coefficient in the transition band. Finally, Table B.4 lists the transition coefficients for the case $\alpha = \frac{1}{2}$, M even, and two coefficients in the transition band. The tables also include the level of the maximum sidelobe and a bandwidth parameter, denoted as BW.

To use the tables, we begin with a set of specifications, including (1) the bandwidth of the filter, which can be defined as $(2\pi/M)(\text{BW}+\alpha)$, where BW is the number of consecutive frequencies at which $H(\omega_k) = 1$, (2) the width of the transition region, which is roughly $2\pi/M$ times the number of transition coefficients, and (3) the maximum tolerable sidelobe in the stopband. The length of the filter can be selected from the tables to satisfy the specifications.

TABLE B.1 Transition Coefficients for $\alpha = 0$

M Odd			M Even		
BW	Minimax	T_1	BW	Minimax	T_1
	$M = 15$			$M = 16$	
1	−42.30932283	0.43378296	1	−39.75363827	0.42631836
2	−41.26299286	0.41793823	2	−37.61346340	0.40397949
3	−41.25333786	0.41047636	3	−36.57721567	0.39454346
4	−41.94907713	0.40405884	4	−35.87249756	0.38916626
5	−44.37124538	0.39268189	5	−35.31695461	0.38840332
6	−56.01416588	0.35766525	6	−35.51951933	0.40155639
	$M = 33$			$M = 32$	
1	−43.03163004	0.42994995	1	−42.24728918	0.42856445
2	−42.42527962	0.41042481	2	−41.29370594	0.40773926
3	−42.40898275	0.40141601	3	−41.03810358	0.39662476
4	−42.45948601	0.39641724	4	−40.93496323	0.38925171
6	−42.52403450	0.39161377	5	−40.85183477	0.37897949
8	−42.44085121	0.39039917	8	−40.75032616	0.36990356
10	−42.11079407	0.39192505	10	−40.54562140	0.35928955
12	−41.92705250	0.39420166	12	−39.93450451	0.34487915
14	−44.69430351	0.38552246	14	−38.91993237	0.34407349
15	−56.18293285	0.35360718			
	$M = 65$			$M = 64$	
1	−43.16935968	0.42919312	1	−42.96059322	0.42882080
2	−42.61945581	0.40903320	2	−42.30815172	0.40830689
3	−42.70906305	0.39920654	3	−42.32423735	0.39807129
4	−42.86997318	0.39335937	4	−42.43565893	0.39177246
5	−43.01999664	0.38950806	5	−42.55461407	0.38742065
6	−43.14578819	0.38679809	6	−42.66526604	0.38416748
10	−43.44808340	0.38129272	10	−43.01104736	0.37609863
14	−43.54684496	0.37946167	14	−43.28309965	0.37089233
18	−43.48173618	0.37955322	18	−43.56508827	0.36605225
22	−43.19538212	0.38162842	22	−43.96245098	0.35977783
26	−42.44725609	0.38746948	26	−44.60516977	0.34813232
30	−44.76228619	0.38417358	30	−43.81448936	0.29973144
31	−59.21673775	0.35282745			
	$M = 125$			$M = 128$	
1	−43.20501566	0.42899170	1	−43.15302420	0.42889404
2	−42.66971111	0.40867310	2	−42.59092569	0.40847778
3	−42.77438974	0.39868774	3	−42.67634487	0.39838257
4	−42.95051050	0.39268189	4	−42.84038544	0.39226685
6	−43.25854683	0.38579101	5	−42.99805641	0.38812256
8	−43.47917461	0.38195801	7	−43.25537014	0.38281250
10	−43.63750410	0.37954102	10	−43.52547789	0.3782638
18	−43.95589399	0.37518311	18	−43.93180990	0.37251587
26	−44.05913115	0.37384033	26	−44.18097305	0.36941528
34	−44.05672455	0.37371826	34	−44.40153408	0.36686401
42	−43.94708776	0.37470093	42	−44.67161417	0.36394653
50	−43.58473492	0.37797851	50	−45.17186594	0.35902100
58	−42.14925432	0.39086304	58	−46.92415667	0.34273681
59	−42.60623264	0.39063110	62	−49.46298973	0.28751221
60	−44.78062010	0.38383713			
61	−56.22547865	0.35263062			

Source: Rabiner et al. (1970); © 1970 IEEE; reprinted with permission

TABLE B.2 Transition Coefficients for $\alpha = 0$

M Odd				M Even			
BW	Minimax	T_1	T_2	BW	Minimax	T_1	T_2
	$M = 15$				$M = 16$		
1	−70.60540585	0.09500122	0.58995418	1	−65.27693653	0.10703125	0.60559357
2	−69.26168156	0.10319824	0.59357118	2	−62.85937929	0.12384644	0.62201631
3	−69.91973495	0.10083618	0.58594327	3	−62.96594906	0.12827148	0.62855407
4	−75.51172256	0.08407953	0.55715312	4	−66.03942485	0.12130127	0.61952704
5	−103.45078300	0.05180206	0.49917424	5	−71.73997498	0.11066284	0.60979204
	$M = 33$				$M = 32$		
1	−70.60967541	0.09497070	0.58985167	1	−67.37020397	0.09610596	0.59045212
2	−68.16726971	0.10585937	0.59743846	2	−63.93104696	0.11263428	0.60560235
3	−67.13149548	0.10937500	0.59911696	3	−62.49787903	0.11931763	0.61192546
5	−66.53917217	0.10965576	0.59674101	5	−61.28204536	0.12541504	0.61824023
7	−67.23387909	0.10902100	0.59417456	7	−60.82049131	0.12907715	0.62307031
9	−67.85412312	0.10502930	0.58771575	9	−59.74928167	0.12068481	0.60685586
11	−69.08597469	0.10219727	0.58216391	11	−62.48683357	0.13004150	0.62821502
13	−75.86953640	0.08137207	0.54712777	13	−70.64571857	0.11017914	0.60670943
14	−104.04059029	0.05029373	0.49149549				
	$M = 65$				$M = 64$		
1	−70.66014957	0.09472656	0.58945943	1	−70.26372528	0.09376831	0.58789222
2	−68.89622307	0.10404663	059.476127	2	−67.20729542	0.10411987	0.59421778
3	−67.90234470	0.10720215	0.59577449	3	−65.80684280	0.10850220	0.59666158
4	−67.24003792	0.10726929	0.59415763	4	−64.95227051	0.11038818	0.59730067
5	−66.86065960	0.10689087	0.59253047	5	−64.42742348	0.11113281	0.59698496
9	−66.27561188	0.10548706	0.58845983	9	−63.41714096	0.10936890	0.59088884
13	−65.96417046	0.10466309	0.58660485	13	−62.72142410	0.10828857	0.58738641
17	−66.16404629	0.10649414	0.58862042	17	−62.37051868	0.11031494	0.58968142
21	−66.76456833	0.10701904	0.58894575	21	−62.04848146	0.11254273	0.59249461
25	−68.13407993	0.10327148	0.58320831	25	−61.88074064	0.11994629	0.60564501
29	−75.98313046	0.08069458	0.54500379	29	−70.05681992	0.10717773	0.59842159
30	−104.92083740	0.04978485	0.48965181				
	$M = 125$				$M = 128$		
1	−70.68010235	0.09464722	0.58933268	1	−70.58992958	0.09445190	0.58900996
2	−68.94157696	0.10390015	0.59450024	2	−68.62421608	0.10349731	0.59379058
3	−68.19352627	0.10682373	0.59508549	3	−67.66701698	0.10701294	0.59506081
5	−67.34261131	0.10668945	0.59187505	4	−66.95196629	0.10685425	0.59298926
7	−67.09767151	0.10587158	0.59821869	6	−66.32718945	0.10596924	0.58953845
9	−67.05801296	0.10523682	0.58738706	9	−66.01315498	0.10471191	0.58593906
17	−67.17504501	0.10372925	0.58358265	17	−65.89422417	0.10288086	0.58097354
25	−67.22918987	0.10316772	0.58224835	25	−65.92644215	0.10182495	0.57812308
33	−67.11609936	0.10303955	0.58198956	33	−65.95577812	0.10096436	0.57576437
41	−66.71271324	0.10313721	0.58245499	41	−65.97698021	0.10094604	0.57451694
49	−66.62364197	0.10561523	0.58629534	49	−65.67919827	0.09865112	0.56927420
57	−69.28378487	0.10061646	0.57812192	57	−64.61514568	0.09845581	0.56604486
58	−70.35782337	0.09663696	0.57121235	61	−71.76589394	0.10496826	0.59452277
59	−75.94707718	0.08054886	0.54451285				
60	−104.09012318	0.04991760	0.48963264				

Source: Rabiner et al. (1970); © 1970 IEEE; reprinted with permission.

TABLE B.3 Transition Coefficients for $\alpha = \frac{1}{2}$

BW	Minimax	T_1
	$M = 16$	
1	−51.60668707	0.26674805
2	−47.48000240	0.32149048
3	−45.19746828	0.34810181
4	−44.32862616	0.36308594
5	−45.68347692	0.36661987
6	−56.63700199	0.34327393
	$M = 32$	
1	−52.64991188	0.26073609
2	−49.39390278	0.30878296
3	−47.72596645	0.32984619
4	−46.68811989	0.34217529
6	−45.33436489	0.35704956
8	−44.30730963	0.36750488
10	−43.11168003	0.37810669
12	−42.97900438	0.38465576
14	−56.32780266	0.35030518
	$M = 64$	
1	−52.90375662	0.25923462
2	−49.74046421	0.30603638
3	−48.38088989	0.32510986
4	−47.47863007	0.33595581
5	−46.88655186	0.34287720
6	−46.46230555	0.34774170
10	−45.46141434	0.35859375
14	−44.85988188	0.36470337
18	−44.34302616	0.36983643
22	−43.69835377	0.37586059
26	−42.45641375	0.38624268
30	−56.25024033	0.35200195
	$M = 128$	
1	−52.96778202	0.25885620
2	−49.82771969	0.30534668
3	−48.51341629	0.32404785
4	−47.67455149	0.33443604
5	−47.11462021	0.34100952
7	−46.43420267	0.34880371
10	−45.88529110	0.35493774
18	−45.21660566	0.36182251
26	−44.87959814	0.36521607
34	−44.61497784	0.36784058
42	−44.32706451	0.37066040
50	−43.87646437	0.37500000
58	−42.30969715	0.38807373
62	−56.23294735	0.35241699

Source: Rabiner et al. (1970); © 1970 IEEE; reprinted with permission.

TABLE B.4 Transition Coefficients for $\alpha = \frac{1}{2}$

BW	Minimax	T_1	T_2
		$M = 16$	
1	−77.26126766	0.05309448	0.41784180
2	−73.81026745	0.07175293	0.49369211
3	−73.02352142	0.07862549	0.51966134
4	−77.95156193	0.07042847	0.51158076
5	−105.23953247	0.04587402	0.46967784
		$M = 32$	
1	−80.49464130	0.04725342	0.40357383
2	−73.92513466	0.07094727	0.49129255
3	−72.40863037	0.08012695	0.52153983
5	−70.95047379	0.08935547	0.54805908
7	−70.22383976	0.09403687	0.56031410
9	−69.94402790	0.09628906	0.56637987
11	−70.82423878	0.09323731	0.56226952
13	−104.85642624	0.04882812	0.48479068
		$M = 64$	
1	−80.80974960	0.04658203	0.40168723
2	−75.11772251	0.06759644	0.48390015
3	−72.66662025	0.07886963	0.51850058
4	−71.85610867	0.08393555	0.53379876
5	−71.34401417	0.08721924	0.54311474
9	−70.32861614	0.09371948	0.56020256
13	−69.34809303	0.09761963	0.56903714
17	−68.06440258	0.10051880	0.57543691
21	−67.99149132	0.10289307	0.58007699
25	−69.32065105	0.10068359	0.57729656
29	−105.72862339	0.04923706	0.48767025
		$M = 128$	
1	−80.89347839	0.04639893	0.40117195
2	−77.22580583	0.06295776	0.47399521
3	−73.43786240	0.07648926	0.51361278
4	−71.93675232	0.08345947	0.53266251
6	−71.10850430	0.08880615	0.54769675
9	−70.53600121	0.09255371	0.55752959
17	−69.95890045	0.09628906	0.56676912
25	−69.29977322	0.09834595	0.57137301
33	−68.75139713	0.10077515	0.57594641
41	−67.89687920	0.10183716	0.57863142
49	−66.76120186	0.10264282	0.58123560
57	−69.21525860	0.10157471	0.57946395
61	−104.57432938	0.04970703	0.48900685

Source: Rabiner et al. (1970); © 1970 IEEE; reprinted with permission.

As an illustration, the filter design for which $M = 15$ and

$$H_r\left(\frac{2\pi k}{M}\right) = \begin{cases} 1, & k = 0, 1, 2, 3 \\ T_1, & k = 4 \\ 0, & k = 5, 6, 7 \end{cases}$$

corresponds to $\alpha = 0$, BW $= 4$, since $H_r(\omega_k) = 1$ at the four consecutive frequencies $\omega_k = 2\pi k/15$, $k = 0, 1, 2, 3$, and the transition coefficient is T_1 at the frequency $\omega_k = 8\pi/15$. The value given in Table B.1 for $M = 15$ and BW $= 4$ is $T_1 = 0.40405884$. The maximum sidelobe is at -41.9 dB, according to Table B.1.

References and Bibliography

AKAIKE, H. 1969. "Power Spectrum Estimation Through Autoregression Model Fitting," *Ann. Inst. Stat. Math.*, Vol. 21, pp. 407–149.

AKAIKE, H. 1974. "A New Look at the Statistical Model Identification," *IEEE Trans. Automatic Control*, Vol. AC-19, pp. 716–723, December.

ANDERSEN, N. O. 1978. "Comments on the Performance of Maximum Entropy Algorithm," *Proc. IEEE*, Vol. 66, pp. 1581–1582, November.

ANTONIOU, A. 1979. *Digital Filters: Analysis and Design*, McGraw-Hill, New York.

AUER, E. 1987. "A Digital Filter Structure Free of Limit Cycles," *Proc. 1987 ICASSP*, pp. 21.11.1–21.11.4, Dallas, TX, April.

AVENHAUS, E., and SCHUESSLER, H. W. 1970. "On the Approximation Problem in the Design of Digital Filters with Limited Wordlength," *Arch. Elek. Ubertragung*, Vol. 24, pp. 571–572.

BAGGEROER, A. B. 1976. "Confidence Intervals for Regression (MEM) Spectral Estimates," *IEEE Trans. Information Theory*, Vol. IT-22, pp. 534–545, September.

BANDLER, J. W., and BARDAKJIAN, B. J. 1973. "Least pth Optimization of Recursive Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-21, pp. 460–470, October.

BARNES, C. W., and FAM, A. T. 1977. "Minimum Norm Recursive Digital Filters That Are Free of Overflow Limit Cycles," *IEEE Trans. Circuits and Systems*, Vol. CAS-24, pp. 569–574, October.

BARTLETT, M. S. 1948. "Smoothing Periodograms from Time Series with Continuous Spectra," *Nature* (London), Vol. 161, pp. 686–687, May.

BARTLETT, M. S. 1961. *Stochastic Processes*, Cambridge University Press, Cambridge, UK.

BECKMAN, F. S. 1960 The Solution of Linear Equations by the conjugate Gradient Method" in *Mathematical Methods for Digital Computers*, A. Ralston and H.S. Wilf, eds., Wiley, New York

BERGLAND, G. D. 1969. "A Guided Tour of the Fast Fourier Transform," *IEEE Spectrum*, Vol. 6, pp. 41–52, July.

BERK, K. N. 1974. "Consistent Autoregressive Spectral Estimates," *Ann. Stat.*, Vol. 2, pp. 489–502.

BERNHARDT, P. A., ANTONIADIS, D. A., and DA ROSA, A. V. 1976. "Lunar Perturbations in Columnar Electron Content and Their Interpretation in Terms of Dynamo Electrostatic Fields," *J. Geophys. Res.*, Vol. 81, pp. 5957–5963, December.

BERRYMAN, J. G. 1978. "Choice of Operator Length for Maximum Entropy Spectral Analysis," *Geophysics*, Vol. 43, pp. 1384–1391, December.

BIERMAN, G. J. 1977. *Factorization Methods for Discrete Sequential Estimation*, Academic, New York.

BLACKMAN, R. B., and TUKEY, J. W. 1958. *The Measurement of Power Spectra*, Dover, New York.

BLAHUT, R. E. 1985. *Fast Algorithms for Digital Signal Processing*, Addison-Wesley, Reading, MA.

BLUESTEIN, L. I. 1970. "A Linear Filtering Approach to the Computation of the Discrete Fourier Transform," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-18, pp. 451–455, December.

BOLT, B. A. 1988. *Earthquakes*, W. H. Freeman and Co., New York.

BOMAR, B. W. 1985. "New Second-Order State-Space Structures for Realizing Low Roundoff Noise Digital Filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, pp. 106–110, February.

BRACEWELL, R. N. 1978. *The Fourier Transform and Its Applications*, 2d ed., McGraw-Hill, New York.

BRIGHAM, E. O. 1988. *The Fast Fourier Transform and Its Applications*, Prentice Hall, Upper Saddle River, NJ.

BRIGHAM, E. O., and MORROW, R. E. 1967. "The Fast Fourier Transform," *IEEE Spectrum*, Vol. 4, pp. 63–70, December.

BRILLINGER, D. R. 1974. "Fourier Analysis of Stationary Processes," *Proc. IEEE*, Vol. 62, pp. 1628–1643, December.

BROPHY, F., and SALAZAR, A. C. 1973. "Considerations of the Padé Approximant Technique in the Synthesis of Recursive Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-21, pp. 500–505, December.

BROWN, J. L., JR., 1980. "First-Order Sampling of Bandpass Signals—A New Approach," *IEEE Trans. Information Theory*, Vol. IT-26, pp. 613–615, September.

BROWN, R. C. 1983. *Introduction to Random Signal Analysis and Kalman Filtering*, Wiley, New York.

BRUBAKER, T. A., and GOWDY, J. N. 1972. "Limit Cycles in Digital Filters," *IEEE Trans. Automatic Control*, Vol. AC-17, pp. 675–677, October.

BRUZZONE, S. P., and KAVEH, M. 1980. "On Some Suboptimum ARMA Spectral Estimators," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, pp. 753–755, December.

BURG, J. P. 1967. "Maximum Entropy Spectral Analysis," *Proc. 37th Meeting of the Society of Exploration Geophysicists*, Oklahoma City, OK, October. Reprinted in *Modern Spectrum Analysis*, D. G. Childers, ed., IEEE Press, New York.

BURG, J. P. 1968. "A New Analysis Technique for Time Series Data," NATO Advanced Study Institute on Signal Processing with Emphasis on Underwater Acoustics, August 12–23. Reprinted in *Modern Spectrum Analysis*, D. G. Childers, ed., IEEE Press, New York.

BURG, J. P. 1972. "The Relationship Between Maximum Entropy and Maximum Likelihood Spectra," *Geophysics*, Vol. 37, pp. 375–376, April.

BURG, J. P. 1975. "Maximum Entropy Spectral Analysis," Ph.D. dissertation, Department of Geophysics, Stanford University, Stanford, CA, May.

BURRUS, C. S., and PARKS, T. W. 1970. "Time-Domain Design of Recursive Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. 18, pp. 137–141, June.

BURRUS, C. S. and PARKS, T. W. 1985. *DFT/FFT and Convolution Algorithms*, Wiley, New York.

BUTTERWECK, H. J., VAN MEER, A. C. P., and VERKROOST, G. 1984. "New Second-Order Digital Filter Sections Without Limit Cycles," *IEEE Trans. Circuits and Systems*, Vol. CAS-31, pp. 141–146, February.

CADZOW, J. A. 1979. "ARMA Spectral Estimation: An Efficient Closed-Form Procedure," *Proc. RADC Spectrum Estimation Workshop*, pp. 81–97, Rome, NY, October.

CADZOW, J. A. 1981. "Autoregressive–Moving Average Spectral Estimation: A Model Equation Error Procedure," *IEEE Trans. Geoscience Remote Sensing*, Vol. GE-19, pp. 24–28, January.

CADZOW, J. A. 1982. "Spectral Estimation: An Overdetermined Rational Model Equation Approach," *Proc. IEEE*, Vol. 70, pp. 907–938, September.

CANDY, J. C. 1986. "Decimation for Sigma Delta Modulation," *IEEE Trans. Communications*, Vol. COM-34, pp. 72–76, January.

CANDY, J. C., WOOLEY, B. A., and BENJAMIN, D. J. 1981. "A Voiceband Codec with Digital Filtering," *IEEE Trans. Communications*, Vol. COM-29, pp. 815–830, June.

CAPON, J. 1969. "High-Resolution Frequency–Wavenumber Spectrum Analysis," *Proc. IEEE*, Vol. 57, pp. 1408–1418, August.

CAPON, J. 1983. "Maximum-Likelihood Spectral Estimation," in *Nonlinear Methods of Spectral Analysis*, 2d ed., S. Haykin, ed., Springer-Verlag, New York.

CAPON, J., and GOODMAN, N. R. 1971. "Probability Distribution for Estimators of the Frequency-Wavenumber Spectrum," *Proc. IEEE*, Vol. 58, pp. 1785–1786, October.

CARAISCOS, C., and LIU, B. 1984. "A Roundoff Error Analysis of the LMS Adaptive Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 34–41, January.

CARAYANNIS, G., MANOLAKIS, D. G., and KALOUPTSIDIS, N. 1983. "A Fast Sequential Algorithm for Least-Squares Filtering and Prediction," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, pp. 1394–1402, December.

CARAYANNIS, G., MANOLAKIS, D. G., and KALOUPTSIDIS, N. 1986. "A Unified View of Parametric Processing Algorithms for Prewindowed Signals," *Signal Processing*, Vol. 10, pp. 335–368, June.

CARLSON, N. A., and CULMONE, A. F. 1979. "Efficient Algorithms for On-Board Array Processing," *Record 1979 International Conference on Communications*, pp. 58.1.1–58.1.5, Boston, June 10–14.

CHAN, D. S. K., and RABINER, L. R. 1973a. "Theory of Roundoff Noise in Cascade Realizations of Finite Impulse Response Digital Filters," *Bell Syst. Tech. J.*, Vol. 52, pp. 329–345, March.

CHAN, D. S. K., and RABINER, L. R. 1973b. "An Algorithm for Minimizing Roundoff Noise in Cascade Realizations of Finite Impulse Response Digital Filters," *Bell Sys. Tech. J.*, Vol. 52, pp. 347–385, March.

CHAN, D. S. K., and RABINER, L. R. 1973c. "Analysis of Quantization Errors in the Direct Form for Finite Impulse Response Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-21, pp. 354–366, August.

CHANG, T. 1981. "Suppression of Limit Cycles in Digital Filters Designed with One Magnitude-Truncation Quantizer," *IEEE Trans. Circuits and Systems*, Vol. CAS-28, pp. 107–111, February.

CHEN, C. T. 1970. *Introduction to Linear System Theory*, Holt, Rinehart and Winston, New York.

CHEN, W. Y., and STEGEN, G. R. 1974. "Experiments with Maximum Entropy Power Spectra of Sinusoids," *J. Geophys. Res.*, Vol. 79, pp. 3019–3022, July.

CHILDERS, D. G., ed. 1978. *Modern Spectrum Analysis*, IEEE Press, New York.

CHOW, J. C. 1972a. "On the Estimation of the Order of a Moving-Average Process," *IEEE Trans. Automatic Control*, Vol. AC-17, pp. 386–387, June.

CHOW, J. C. 1972b. "On Estimating the Orders of an Autoregressive-Moving Average Process with Uncertain Observations," *IEEE Trans. Automatic Control*, Vol. AC-17, pp. 707–709, October.

CHOW, Y., and CASSIGNOL, E. 1962. *Linear Signal Flow Graphs and Applications*, Wiley, New York.

CHUI, C. K., and CHEN, G. 1987. *Kalman Filtering*, Springer-Verlag, New York.

CIOFFI, J. M. 1987, "Limited Precision Effects in Adaptive Filtering," *IEEE Trans. Circuits and Systems*, Vol. CAS-34, pp. 821–833, July.

CIOFFI, J. M., and KAILATH, T. 1984. "Fast Recursive-Least-Squares Transversal Filters for Adaptive Filtering," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 304–337, April.

CIOFFI, J. M., and KAILATH, T. 1985. "Windowed Fast Transversal Filters Adaptive Algorithms with Normalization," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, pp. 607–625, June.

CLAASEN, T. A. C. M., MECKLENBRAUKER, W. F. G., and PEEK, J. B. H. 1973. "Second-Order Digital Filter with Only One Magnitude-Truncation Quantizer and Having Practically No Limit Cycles," *Electron. Lett.*, Vol. 9, November.

CLARKE, R. J. 1985. *Transform Coding of Images*. Academic Press, London, UK.

COCHRAN, W. T., COOLEY, J. W., FAVIN, D. L., HELMS, H. D., KAENEL, R. A., LANG, W. W., MALING, G. C., NELSON, D. E., RADER, C. E., and WELCH, P. D. 1967. "What Is the Fast Fourier Transform," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-15, pp. 45–55, June.

CONSTANTINIDES, A. G. 1967. "Frequency Transformations for Digital Filters," *Electron. Lett.*, Vol. 3, pp. 487–489, November.

CONSTANTINIDES, A. G. 1968. "Frequency Transformations for Digital Filters," *Electron. Lett.*, Vol. 4, pp. 115–116, April.

CONSTANTINIDES, A. G. 1970. "Spectral Transformations for Digital Filters," *Proc. IEEE*, Vol. 117, pp. 1585–1590, August.

COOLEY, J. W., and TUKEY, J. W. 1965. "An Algorithm for the Machine Computation of Complex Fourier Series," *Math. Comp.*, Vol. 19, pp. 297–301, April.

COOLEY, J. W., LEWIS, P., and WELCH, P. D. 1967. "Historical Notes on the Fast Fourier Transform," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-15, pp. 76–79, June.

COOLEY, J. W., LEWIS, P., and WELCH, P. D. 1969. "The Fast Fourier Transform and Its Applications," *IEEE Trans. Education*, Vol. E-12, pp. 27–34, March.

COULSON, A. 1995. "A Generalization of Nonuniform Bandpass Sampling." *IEEE Trans. on Signal Processing*, Vol. 43(3), pp. 694–704, March.

COULSON, A., VAUGHAM, R., and POULETTI, M, 1994. "Frequency Shifting Using Bandpass Sampling." *IEEE Trans. on Signal Processing*, Vol 42(6), pp. 1556–1559, June.

CROCHIERE, R. E. 1977. "On the Design of Sub-Band Coders for Low Bit Rate Speech Communication," *Bell Syst. Tech. J.*, Vol. 56, pp. 747–711, May–June.

CROCHIERE, R. E. 1981. "Sub-Band Coding," *Bell Syst. Tech. J.*, Vol. 60, pp. 1633–1654, September.

CROCHIERE, R. E., and RABINER, L. R. 1975. "Optimum FIR Digital Filter Implementations for Decimation, Interpolation, and Narrowband Filtering," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-23, pp. 444–456, October.

CROCHIERE, R. E., and RABINER, L. R. 1976. "Further Considerations in the Design of Decimators and Interpolators," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-24, pp. 296–311, August.

CROCHIERE, R. E., and RABINER, L. R. 1981. "Interpolation and Decimation of Digital Signals—A Tutorial Review," *Proc. IEEE*, Vol. 69, pp. 300–331, March.

CROCHIERE, R. E., and RABINER, L. R. 1983. *Multirate Digital Signal Processing*, Prentice Hall, Upper Saddle River, NJ.

DANIELS, R. W. 1974. *Approximation Methods for the Design of Passive, Active and Digital Filters*, McGraw-Hill, New York.

DAVENPORT, W. B., JR. 1970. *Probability and Random Processes: An Introduction for Applied Scientists and Engineers*, McGraw-Hill, New York.

DAVIS, H. F. 1963. *Fourier Series and Orthogonal Functions*, Allyn and Bacon, Boston.

DECZKY, A. G. 1972. "Synthesis of Recursive Digital Filters Using the Minimum p-Error Criterion," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-20, pp. 257–263, October.

DELLER, J. R. Jr., HANSEN, J. H. L., and PROAKIS, J. G. 2000. *Discrete-Time Processing of Speech Signals*, Wiley, New York.

DELSARTE, P., and GENIN, Y. 1986. "The Split Levinson Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp. 470–478, June.

DELSARTE, P., GENIN, Y., and KAMP, Y. 1978. "Orthogonal Polynomial Matrices on the Unit Circle," *IEEE Trans. Circuits and Systems*, Vol. CAS-25, pp. 149–160. January.

DERUSSO, P. M., ROY, R. J., and CLOSE, C. M. 1965. *State Variables for Engineers*, Wiley, New York.

DUHAMEL, P. 1986. "Implementation of Split-Radix FFT Algorithms for Complex, Real, and Real-Symmetric Data," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp. 285–295, April.

DUHAMEL, P., and HOLLMANN, H. 1984. "Split-Radix FFT Algorithm," *Electron. Lett.*, Vol. 20, pp. 14–16, January.

DURBIN, J. 1959. "Efficient Estimation of Parameters in Moving-Average Models," *Biometrika*, Vol. 46, pp. 306–316.

DWIGHT, H. B. 1957. *Tables of Integrals and Other Mathematical Data*, 3d ed., Macmillan, New York.

DYM, H., and MCKEAN, H. P. 1972. *Fourier Series and Integrals*, Academic, New York.

EBERT, P. M., MAZO, J. E., and TAYLOR, M. G. 1969. "Overflow Oscillations in Digital Filters," *Bell Syst. Tech. J.*, Vol. 48, pp. 2999–3020, November.

ELEFTHERIOU, E., and FALCONER, D. D. 1987. "Adaptive Equalization Techniques for HF Channels," *IEEE J. Selected Areas in Communications*, Vol. SAC-5, pp. 238–247, February.

ELUP, L., GARDNER, F. M., and HARRIS F. A., 1993 "Interpolation in digital Modems, Part II: Fundamentals and performance." *IEEE trans. on Communications*, Vol, 41(6), pp. 998–1008, June.

FALCONER, D. D., and LJUNG, L. 1978. "Application of Fast Kalman Estimation to Adaptive Equalization," *IEEE Trans. Communications*, Vol. COM-26, pp. 1439–1446, October.

FAM, A. T., and BARNES, C. W. 1979. "Non-minimal Realizations of Fixed-Point Digital Filters That Are Free of All Finite Wordlength Limit Cycles," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, pp. 149–153, April.

FARROW, C. W. 1998. " A Continiously Variable Digital Delay Element." *Proc. IEEE Intern. Symposium on Circuits and Systems*, pp. 2641–2645.

FETTWEIS, A. 1971. "Some Principles of Designing Digital Filters Imitating Classical Filter Structures," *IEEE Trans. Circuit Theory*, Vol. CT-18, pp. 314–316, March.

FLETCHER, R., and POWELL, M. J. D. 1963. "A Rapidly Convergent Descent Method for Minimization," *Comput. J.*, Vol. 6, pp. 163–168.

FOUGERE, P. F., ZAWALICK, E. J., and RADOSKI, H. R. 1976. "Spontaneous Line Splitting in Maximum Entropy Power Spectrum Analysis," *Phys. Earth Planet. Inter.*, Vol. 12, 201–207, August.

FRERKING, M. E. 1994. *Digital Signal Processing in Communication Systems*, Kluwer Academic Publishers, Boston.

FRIEDLANDER, B. 1982a. "Lattice Filters for Adaptive Processing," *Proc. IEEE*, Vol. 70, pp. 829–867, August.

FRIEDLANDER, B. 1982b. "Lattice Methods for Spectral Estimation," *Proc. IEEE*, Vol. 70, pp. 990–1017, September.

FUCHS, J. J. 1988. "Estimating the Number of Sinusoids in Additive White Noise," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-36, pp. 1846–1853, December.

GANTMACHER, F. R. 1960. *The Theory of Matrices*, Vol. I., Chelsea, New York.

GARDNER, F. M. 1993. "Interpolation in Digital Modems, Part I: Fundamentals." *IEEE Trans. on Communications*, Vol. 41(3), pp. 502–508, March.

GARDNER, W. A. 1984. "Learning Characteristics of Stochastic-Gradient-Descent Algorithms: A General Study, Analysis and Critique," *Signal Processing*, Vol. 6, pp. 113–133, April.

GARDNER, W. A. 1987. *Statistical Spectral Analysis: A Nonprobabilistic Theory*, Prentice Hall, Upper Saddle River, NJ.

GARLAN, C., and ESTEBAN, D. 1980. "16 Kbps Real-Time QMF Sub-Band Coding Implementation," *Proc. 1980 International Conference on Acoustics, Speech, and Signal Processing*, pp. 332–335, April.

GEORGE, D. A., BOWEN, R. R., and STOREY, J. R. 1971. "An Adaptive Decision-Feedback Equalizer," *IEEE Trans. Communication Technology*, Vol. COM-19, pp. 281–293, June.

GERONIMUS, L. Y. 1958. *Orthogonal Polynomials* (in Russian) (English translation by Consultants Bureau, New York, 1961).

GERSCH, W., and SHARPE, D. R. 1973. "Estimation of Power Spectra with Finite-Order Autoregressive Models," *IEEE Trans. Automatic Control*, Vol. AC-18, pp. 367–369, August.

GERSHO, A. 1969. "Adaptive Equalization of Highly Dispersive Channels for Data Transmission," *Bell Syst. Tech. J.*, Vol. 48, pp. 55–70, January.

GIBBS, A. J. 1969. "An Introduction to Digital Filters," *Aust. Telecommun. Res.*, Vol. 3, pp. 3–14, November.

GIBBS, A. J. 1970. "The Design of Digital Filters," *Aust. Telecommun. Res.*, Vol. 4, pp. 29–34, May.

GITLIN, R. D., and WEINSTEIN, S. B. 1979. "On the Required Tap-Weight Precision for Digitally Implemented Mean-Squared Equalizers," *Bell Syst. Tech. J.*, Vol. 58, pp. 301–321, February.

GITLIN, R. D., MEADORS, H. C., and WEINSTEIN, S. B. 1982. "The Tap-Leakage Algorithm: An Algorithm for the Stable Operation of a Digitally Implemented Fractionally Spaced, Adaptive Equalizer," *Bell Syst. Tech. J.*, Vol. 61, pp. 1817–1839, October.

GOERTZEL, G. 1968. "An Algorithm for the Evaluation of Finite Trigonometric Series," *Am. Math. Monthly*, Vol. 65, pp. 34–35, January.

GOHBERG, I., ed. 1986. *I. Schür Methods in Operator Theory and Signal Processing*, Birkhauser Verlag, Stuttgart, Germany.

GOLD, B., and JORDAN, K. L., JR. 1986. "A Note on Digital Filter Synthesis," *Proc. IEEE*, Vol. 56, pp. 1717–1718, October.

GOLD, B., and JORDAN, K. L., JR. 1969. "A Direct Search Procedure for Designing Finite Duration Impulse Response Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-17, pp. 33–36, March.

GOLD, B., and RADER, C. M. 1966. "Effects of Quantization Noise in Digital Filters." *Proc. AFIPS 1966 Spring Joint Computer Conference*, Vol. 28, pp. 213–219.

GOLD, B., and RADER, C. M. 1969. *Digital Processing of Signals*, McGraw-Hill, New York.

GOLDEN, R. M., and KAISER, J. F. 1964. "Design of Wideband Sampled Data Filters," *Bell Syst. Tech. J.*, Vol. 43, pp. 1533–1546, July.

GOOD, I. J. 1971. "The Relationship Between Two Fast Fourier Transforms," *IEEE Trans. Computers*, Vol. C-20, pp. 310–317.

GORSKI-POPIEL, J., ed. 1975. *Frequency Synthesis: Techniques and Applications*, IEEE Press, New York.

GOYAL, V. 2001. "Theoretical Foundations of Transform Coding." *IEEE Signal Processing Magazine*, pp 9-21, September.

GRACE, O. D., and PITT, S. P. 1969. "Sampling and Interpolation of Bandlimited Signals by Quadrature Methods." *J. Acoust. Soc. Amer.*, Vol. 48(6), pp. 1311–1318, November.

GRAUPE, D., KRAUSE, D. J., and MOORE, J. B. 1975. "Identification of Autoregressive–Moving Average Parameters of Time Series," *IEEE Trans. Automatic Control*, Vol. AC-20, pp. 104–107, February.

GRAY, A. H., and MARKEL, J. D. 1973. "Digital Lattice and Ladder Filter Synthesis," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-21, pp. 491–500, December.

GRAY, A. H., and MARKEL, J. D. 1976. "A Computer Program for Designing Digital Elliptic Filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-24, pp. 529–538, December.

GRAY, R. M. 1990. *Source Coding Theory*, Kluwer, Boston, MA.

GRENANDER, O., and SZEGO, G. 1958. *Toeplitz Forms and Their Applications*, University of California Press, Berkeley, CA.

GRIFFITHS, L. J. 1975. "Rapid Measurements of Digital Instantaneous Frequency," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-23, pp. 207–222, April.

GRIFFITHS, L. J. 1978. "An Adaptive Lattice Structure for Noise Cancelling Applications," Proc. ICASSP-78, pp. 87–90. Tulsa, OK, April.

GUILLEMIN, E. A. 1957. *Synthesis of Passive Networks*, Wiley, New York.

GUPTA, S. C. 1966. *Transform and State Variable Methods in Linear Systems*, Wiley, New York.

HAMMING, R. W. 1962. *Numerical Methods for Scientists and Engineers*, McGraw-Hill, New York.

HARRIS, F. 1997. *Performance and Design of Farrow Filter Used for Arbitrary Resampling.* 31st Conference on Signals, Systems, and Computers, Pacific Grove, CA, pp. 595–599.

HAYKIN, S. 1991. *Adaptive Filter Theory*, 2d ed., Prentice Hall, Upper Saddle River, NJ.

HELME, B., and NIKIAS, C. S. 1985. "Improved Spectrum Performance via a Data-Adaptive Weighted Burg Technique," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, pp. 903–910, August.

HELMS, H. D. 1967. "Fast Fourier Transforms Method of Computing Difference Equations and Simulating Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-15, pp. 85–90, June.

HELMS, H. D. 1968. "Nonrecursive Digital Filters: Design Methods for Achieving Specifications on Frequency Response," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-16, pp. 336–342, September.

HELSTROM, C. W. 1990. *Probability and Stochastic Processes for Engineers*, 2d ed., Macmillan, New York.

HERRING, R. W. 1980. "The Cause of Line Splitting in Burg Maximum-Entropy Spectral Analysis," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, pp. 692–701, December.

HERMANN, O. 1970. "Design of Nonrecursive Digital Filters with Linear Phase," *Electron. Lett.*, Vol. 6, pp. 328–329, November.

HERMANN, O., and SCHÜESSLER, H. W. 1970a. "Design of Nonrecursive Digital Filters with Minimum Phase," *Electron. Lett.*, Vol. 6, pp. 329–330, November.

HERRMANN, O., and SCHÜESSLER, H. W. 1970b. "On the Accuracy Problem in the Design of Nonrecursive Digital Filters," *Arch. Elek. Ubertragung*, Vol. 24, pp. 525–526.

HERRMANN, O., RABINER, L. R., and CHAN, D. S. K. 1973. "Practical Design Rules for Optimum Finite Impulse Response Lowpass Digital Filters," *Bell Syst. Tech. J.*, Vol. 52, pp. 769–799, July–August.

HILDEBRAND, F. B. 1952. *Methods of Applied Mathematics*, Prentice Hall, Upper Saddle River, NJ.

HOFSTETTER, E., OPPENHEIM, A. V., and SIEGEL, J. 1971. "A New Technique for the Design of Nonrecursive Digital Filters," *Proc. 5th Annual Princeton Conference on Information Sciences and Systems*, pp. 64–72.

HOGENAUER, E.B. 1981. "An Economical Class of Digital Filters for Decimation and Interpolation" *IEEE Trans. on ASSP*, Vol. 29(2), pp. 155–162, April.

HOUSEHOLDER, A. S. 1964. *The Theory of Matrices in Numerical Analysis*, Blaisdell, Waltham, MA.

HSU, F. M. 1982. "Square-Root Kalman Filtering for High-Speed Data Received Over Fading Dispersive HF Channels," *IEEE Trans. Information Theory*, Vol. IT-28, pp. 753–763, September.

HSU, F. M., and GIORDANO, A. A. 1978. "Digital Whitening Techniques for Improving Spread Spectrum Communications Performance in the Presence of Narrowband Jamming and Interference," *IEEE Trans. Communications*, Vol. COM-26, pp. 209–216, February.

HWANG, S. Y. 1977. "Minimum Uncorrelated Unit Noise in State Space Digital Filtering," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-25, pp. 273–281, August.

INCINBONO, B.1978. " adaptive Signal Processing for Detection and Communication," in *communication Systems and Random Process Theory*, J.K. Skwirzynski, ed., Sijthoff en Noordhoff, Alphen aan den Rijn, The Netherlands.

JAYANT, N. S., and NOLL, P. 1984. *Digital Coding of waveforms*, Prentice Hall, Upper Saddle River, NJ.

JACKSON, L. B. 1969. "An Analysis of Limit Cycles Due to Multiplication Rounding in Recursive Digital (Sub) Filters," *Proc. 7th Annual Allerton Conference on Circuit and System Theory*, pp. 69–78.

JACKSON, L. B. 1970a. "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," *Bell Syst. Tech. J.*, Vol. 49, pp. 159–184, February.

JACKSON, L. B. 1970b. "Roundoff Noise Analysis for Fixed-Point Digital Filters Realized in Cascade or Parallel Form," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-18, pp. 107–122, June.

JACKSON, L. B. 1976. "Roundoff Noise Bounds Derived from Coefficients Sensitivities in Digital Filters," *IEEE Trans. Circuits and Systems*, Vol. CAS-23, pp. 481–485, August.

JACKSON, L. B. 1979. "Limit Cycles on State-Space Structures for Digital Filters," *IEEE Trans. Circuits and Systems*, Vol. CAS-26, pp. 67–68, January.

JACKSON, L. B., LINDGREN, A. G., and KIM, Y. 1979. "Optimal Synthesis of Second-Order State-Space Structures for Digital Filters," *IEEE Trans. Circuits and Systems*, Vol. CAS-26, pp. 149–153, March.

JACKSON, M., and MATTHEWSON, P. 1986. "Digital Processing of Bandpass Signals." *GEC Journal of Research*, Vol. 4(1), pp. 32–41.

JAHNKE, E., and EMDE, F. 1945. *Tables of Functions*, 4th ed., Dover, New York.

JAIN, V. K., and CROCHIERE, R. E. 1984. "Quadrature Mirror Filter Design in the Time Domain," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 353–361, April.

JANG, Y., and YANG, S. 2001. *Recursive Cascaded Integrator-Comb Decimation Filters with Integer Multiple factors*, 44th IEEE Midwest Symposium on circuits and Systems, Daytona, OH, August.

JENKINS, G. M., and WATTS, D. G. 1968. *Spectral Analysis and Its Applications*, Holden-Day, San Francisco.

JOHNSON, D. H. 1982. "The Application of Spectral Estimation Methods to Bearing Estimation Problems," *Proc. IEEE*, Vol. 70, pp. 1018–1028, September.

JOHNSTON, J. D. 1980. "A Filter Family Designed for Use in Quadrature Mirror Filter Banks," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 291–294, April.

JONES, R. H. 1976. "Autoregression Order Selection," *Geophysics*, Vol. 41, pp. 771–773, August.

JONES, S. K., CAVIN, R. K., and REED, W. M. 1982. "Analysis of Error-Gradient Adaptive Linear Equalizers for a Class of Stationary-Dependent Processes," *IEEE Trans. Information Theory*, Vol. IT-28, pp. 318–329, March.

JURY, E. I. 1964. *Theory and Applications of the z-Transform Method*, Wiley, New York.

KAILATH, T. 1974. "A View of Three Decades of Linear Filter Theory," *IEEE Trans. Information Theory*, Vol. IT-20, pp. 146–181, March.

KAILATH, T. 1981. *Lectures on Wiener and Kalman Filtering*, 2d printing, Springer-Verlag, New York.

KAILATH, T. 1985. "Linear Estimation for Stationary and Near-Stationary Processes," in *Modern Signal Processing*, T. Kailath, ed., Hemisphere Publishing Corp., Washington, DC.

KAILATH, T. 1986. "A Theorem of I. Schür and Its Impact on Modern Signal Processing," in Gohberg 1986.

KAILATH, T., VIEIRA, A. C. G, and MORF, M. 1978. "Inverses of Toeplitz Operators, Innovations, and Orthogonal Polynomials," *SIAM Rev.*, Vol. 20, pp. 1006–1019.

KAISER, J. F. 1963. "Design Methods for Sampled Data Filters," *Proc. First Allerton Conference on Circuit System Theory*, pp. 221–236, November.

KAISER, J. F. 1966. "Digital Filters," in *System Analysis by Digital Computer*, F. F. Kuo and J. F. Kaiser, eds., Wiley, New York.

KALOUPTSIDIS, N., and THEODORIDIS, S. 1987. "Fast Adaptive Least-Squares Algorithms for Power Spectral Estimation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, pp. 661–670, May.

KALMAN, R. E. 1960. "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME, J. Basic Eng.*, Vol. 82D, pp. 35–45, March.

KALMAN, R. E., and BUCY, R. S. 1961. "New Results in Linear Filtering Theory," *Trans. ASME, J. Basic Eng.*, Vol. 83, pp. 95–108.

KASHYAP, R. L. 1980. "Inconsistency of the AIC Rule for Estimating the Order of Autoregressive Models," *IEEE Trans. Automatic Control*, Vol. AC-25, pp. 996–998, October.

KAVEH, J., and BRUZZONE, S. P. 1979. "Order Determination for Autoregressive Spectral Estimation," *Record of the 1979 RADC Spectral Estimation Workshop*, pp. 139–145, Griffin Air Force Base, Rome, NY.

KAVEH, M., and LIPPERT, G. A. 1983. "An Optimum Tapered Burg Algorithm for Linear Prediction and Spectral Analysis," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, pp. 438–444, April.

KAY, S. M. 1980. "A New ARMA Spectral Estimator," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, pp. 585–588, October.

KAY, S. M. 1988. *Modern Spectral Estimation*, Prentice Hall, Upper Saddle River, NJ.

KAY, S. M., and MARPLE, S. L., JR. 1979. "Sources of and Remedies for Spectral Line Splitting in Autoregressive Spectrum Analysis," *Proc. 1979 ICASSP*, pp. 151–154.

KAY, S. M., and MARPLE, S. L., JR. 1981. "Spectrum Analysis: A Modern Perspective," *Proc. IEEE*, Vol. 69, pp. 1380–1419, November.

KESLER, S. B., ed. 1986. *Modern Spectrum Analysis II*, IEEE Press, New York.

KETCHUM, J. W., and PROAKIS, J. G. 1982. "Adaptive Algorithms for Estimating and Suppressing Narrow-Band Interference in PN Spread-Spectrum Systems," *IEEE Trans. Communications*, Vol. COM-30, pp. 913–923, May.

KNOWLES, J. B., and OLCAYTO, E. M. 1968. "Coefficient Accuracy and Digital Filter Response," *IEEE Trans. Circuit Theory*, Vol. CT-15, pp. 31–41, March.

KOHLENBURG, A. 1953. "Exact Interpolation of Bandlimited Functions." *Journal of Applied Physics*, Vol. 24(12), pp. 1432–1436, May.

KRISHNA, H. 1988. "New Split Levinson, Schür, and Lattice Algorithms for Digital Signal Processing," *Proc. 1988 International Conference on Acoustics, Speech, and Signal Processing*, pp. 1640–1642, New York, April.

KROMER, R. E. 1969. "Asymptotic Properties of the Autoregressive Spectral Estimator," Ph.D. dissertation, Department of Statistics, Stanford University, Stanford, CA.

KUNG, H. T. 1982. "Why Systolic Architectures"? *IEEE Computer*, Vol. 15, pp. 37–46.

KUNG, S. Y., and HU, Y. H. 1983. "A Highly Concurrent Algorithm and Pipelined Architecture for Solving Toeplitz Systems," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, pp. 66–76, January.

KUNG, S. Y., WHITEHOUSE, H. J., and KAILATH, T., eds. 1985. *VLSI and Modern Signal Processing*, Prentice Hall, Upper Saddle River, NJ.

LAAKSO, T., VALIMAKI, V., KARJALAINEN, M., and LAINE, U. 1996. "Splitting the Unit Delay." *IEEE Signal Processing Magazine*, No. 1, pp. 30–54, January.

LACOSS, R. T. 1971. "Data Adaptive Spectral Analysis Methods," *Geophysics*, Vol. 36, pp. 661–675, August.

LANG, S. W., and MCCLELLAN, J. H. 1980. "Frequency Estimation with Maximum Entropy Spectral Estimators," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, pp. 716–724, December.

LEVINSON, N. 1947. "The Wiener RMS Error Criterion in Filter Design and Prediction," *J. Math. Phys.*, Vol. 25, pp. 261–278.

LEVY, H., and LESSMAN, F. 1961. *Finite Difference Equations*, Macmillan, New York.

LIN, D. W. 1984. "On Digital Implementation of the Fast Kalman Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 998–1005, October.

LINDEN, D. A. 1959. " A Discussion of Sampling Theorems." *Proc. of the IRE*, Vol. 47(11), pp. 1219–1226, November.

LING, F., and PROAKIS, J. G. (1984a). "Numerical Accuracy and Stability: Two Problems of Adaptive Estimation Algorithms Caused by Round-Off Error," *Proc. ICASSP-84*, pp. 30.3.1–30.3.4, San Diego, CA, March.

LING, F., and PROAKIS, J. G. (1984b). "Nonstationary Learning Characteristics of Least-Squares Adaptive Estimation Algorithms," *Proc. ICASSP-84*, pp. 3.7.1–3.7.4, San Diego, CA, March.

LING, F., MANOLAKIS, D., and PROAKIS, J. G. 1985, "New Forms of LS Lattice Algorithms and Analysis of Their Round-Off Error Characteristics," *Proc. ICASSP-85*, pp. 1739–1742, Tampa, FL, April.

LING, F., MANOLAKIS, D., and PROAKIS, J. G. 1986. "Numerically Robust Least-Squares Lattice-Ladder Algorithms with Direct Updating of the Reflection Coefficients," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp. 837–845, August.

LIU, B. 1971. "Effect of Finite Word Length on the Accuracy of Digital Filters—A Review," *IEEE Trans. Circuit Theory*, Vol. CT-18, pp. 670–677, November.

LJUNG, S., and LJUNG, L. 1985. "Error Propagation Properties of Recursive Least-Squares Adaptation Algorithms," *Automatica*, Vol. 21, pp. 157–167.

LUCKY, R. W. 1965. "Automatic Equalization for Digital Communications," *Bell Syst. Tech. J.*, Vol. 44, pp. 547–588, April.

MAGEE, F. R. and PROAKIS, J. G. 1973. "Adaptive Maximum-Likelihood Sequence Estimation for Digital Signaling in the Presence of Intersymbol Interference," *IEEE Trans. Information Theory*, Vol. IT-19, pp. 120–124, January.

MAKHOUL, J. 1975. "Linear Prediction: A Tutorial Review," *Proc. IEEE*, Vol. 63, pp. 561–580, April.

MAKHOUL, J. 1978. "A Class of All-Zero Lattice Digital Filters: Properties and Applications," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-26, pp. 304–314, August.

MAKHOUL, J. 1980. "A Fast Cosine Transform In One and Two Dimentions." *IEEE Trans. on ASSP*, Vol. 28(1), pp. 27–34, February.

MARKEL, J. D., and GRAY, A. H., JR. 1976. *Linear Prediction of Speech*, Springer-Verlag, New York.

MANOLAKIS, D., LING, F., and PROAKIS, J. G. 1987. "Efficient Time-Recursive Least-Squares Algorithms for Finite-Memory Adaptive Filtering," *IEEE Trans. Circuits and Systems*, Vol. CAS-34, pp. 400–408, April.

MARPLE, S. L., JR. 1980. "A New Autoregressive Spectrum Analysis Algorithm," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, pp. 441–454, August.

MARPLE, S. L., JR. 1987. *Digital Spectral Analysis with Applications*, Prentice Hall, Upper Saddle River, NJ.

MARTUCCI, S. A. 1994. "Symmetric Convolution and the Discrete Sine and Cosine Transforms." *IEEE Trans. on Signal Processing*, Vol. 42(5), pp. 1038–1051, May.

MARZETTA, T. L. 1983. "A New Interpretation for Capon's Maximum Likelihood Method of Frequency–Wavenumber Spectral Estimation," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-31, pp. 445–449, April.

MARZETTA, T. L., and LANG, S. W. 1983. "New Interpretations for the MLM and DASE Spectral Estimators," *Proc. 1983 ICASSP*, pp. 844–846, Boston, April.

MARZETTA, T. L., and LANG, S. W. 1984. "Power Spectral Density Bounds," *IEEE Trans. Information Theory*, Vol. IT-30, pp. 117–122, January.

MASON, S. J., and ZIMMERMAN, H. J. 1960. *Electronic Circuits, Signals and Systems*, Wiley, New York.

MAZO, J. E. 1979. "On the Independence Theory of Equalizer Convergence," *Bell Syst. Tech. J.*, Vol. 58, pp. 963–993, May.

MCCLELLAN, J. H. 1982. "Multidimensional Spectral Estimation," *Proc. IEEE*, Vol. 70, pp. 1029–1039, September.

MCDONOUGH, R. N. 1983. "Application of the Maximum-Likelihood Method and the Maximum Entropy Method to Array Processing," in *Nonlinear Methods of Spectral Analysis*, 2d ed., S. Haykin, ed., Springer-Verlag, New York.

MCGILLEM, C. D., and COOPER, G. R. 1984. *Continuous and Discrete Signal and System Analysis*, 2d ed., Holt Rinehart and Winston, New York.

MEDITCH, J. E. 1969. *Stochastic Optimal Linear Estimation and Control*, McGraw-Hill, New York.

MEYER, R.,and BURRUS, S. 1975. "A Unified Analysis of Multirate and Periodically Time-Varying Digital Filters." *IEEE Trans. on Circuits and Systems*, Vol. 22(3), pp. 162–168, March.

MILLS, W. L., MULLIS, C. T., and ROBERTS, R. A. 1981. "Low Roundoff Noise and Normal Realizations of Fixed-Point IIR Digital Filters," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, pp. 893–903, August.

MOORER, J. A. 1977. "Signal Aspects of Computer Music; A Survey," *Proc. IEEE*, Vol. 65, pp. 1108–1137, August.

MORF, M., VIEIRA, A., and LEE, D. T. 1977. "Ladder Forms for Identification and Speech Processing," *Proc. 1977 IEEE Conference Decision and Control*, pp. 1074–1078, New Orleans, LA, December.

MULLIS, C. T., and ROBERTS, R. A. 1976a. "Synthesis of Minimum Roundoff Noise Fixed-Point Digital Filters," *IEEE Trans. Circuits and Systems*, Vol. CAS-23, pp. 551–561, September.

MULLIS, C. T., and ROBERTS, R. A. 1976b. "Roundoff Noise in Digital Filters: Frequency Transformations and Invariants," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-24, pp. 538–549, December.

MURRAY, W., ed. 1972 *Numerical Methods for Unconstrained Minimization*, Acedemic, New York.

MUSICUS, B. 1985. "Fast MLM Power Spectrum Estimation from Uniformly Spaced Correlations," *IEEE Trans. Acoustics, Speech, and Signal Proc.*, Vol. ASSP-33, pp. 1333–1335, October.

NEWMAN, W. I. 1981. "Extension to the Maximum Entropy Method III," *Proc. 1st ASSP Workshop on Spectral Estimation*, pp. 1.7.1–1.7.6, Hamilton, ON, August.

NICHOLS, H. E., GIORDANO, A. A., and PROAKIS, J. G. 1977. "MLD and MSE Algorithms for Adaptive Detection of Digital Signals in the Presence of Interchannel Interference," *IEEE Trans. Information Theory*, Vol. IT-23, pp. 563–575, September.

NIKIAS, C. L., and RAGHUVEER, M. R. 1987. "Bispectrum Estimation: A Digital Signal Processing Framework," *Proc. IEEE*, Vol. 75, pp. 869–891, July.

NIKIAS, C. L., and SCOTT, P. D. 1982. "Energy-Weighted Linear Predictive Spectral Estimation: A New Method Combining Robustness and High Resolution," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, pp. 287–292, April.

NUTTALL, A. H. 1976. "Spectral Analysis of a Univariate Process with Bad Data Points, via Maximum Entropy and Linear Predictive Techniques," *NUSC Technical Report TR-5303*, New London, CT, March.

NYQUIST, H. 1928. "Certain Topics in Telegraph Transmission Theory," *Trans. AIEE*, Vol. 47, pp. 617–644, April.

OPPENHEIM, A. V. 1978. *Applications of Digital Signal Processing*, Prentice Hall, Upper Saddle River, NJ.

OPPENHEIM, A. V., and SCHAFER, R. W. 1989. *Discrete-Time Signal Processing*, Prentice Hall, Upper Saddle River, NJ.

OPPENHEIM, A. V., and WEINSTEIN, C. W. 1972. "Effects of Finite Register Length in Digital Filters and the Fast Fourier Transform," *Proc. IEEE*, Vol. 60, pp. 957–976. August.

OPPENHEIM, A. V., and WILLSKY, A. S. 1983. *Signals and Systems*, Prentice Hall, Upper Saddle River, NJ.

PAPOULIS, A. 1962 *The Fourier Integral and Its Applications*, McGraw-Hall, New York.

PAPOULIS, A. 1984. *Probability, Random Variables, and Stochastic Processes*, 2d ed., McGraw-Hill, New York.

PARKER, S. R., and HESS, S. F. 1971. "Limit-Cycle Oscillations in Digital Filters," *IEEE Trans. Circuit Theory*, Vol. CT-18, pp. 687–696, November.

PARKS, T. W., and MCCLELLAN, J. H. 1972a. "Chebyshev-Approximation for Nonrecursive Digital Filters with Linear Phase," *IEEE Trans. Circuit Theory*, Vol. CT-19, pp. 189–194, March.

PARKS, T. W., and MCCLELLAN, J. H. 1972b. "A Program for the Design of Linear Phase Finite Impulse Response Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-20, pp. 195–199, August.

PARZEN, E. 1957. "On Consistent Estimates of the Spectrum of a Stationary Time Series," *Am. Math. Stat.*, Vol. 28, pp. 329–348.

PARZEN, E. 1974. "Some Recent Advances in Time Series Modeling," *IEEE Trans. Automatic Control*, Vol. AC-19, pp. 723–730, December.

PEACOCK, K. L, and TREITEL, S. 1969. "Predictive Deconvolution—Theory and Practice," *Geophysics*, Vol. 34, pp. 155–169.

PEEBLES, P. Z., JR. 1987. *Probability, Random Variables, and Random Signal Principles*, 2d ed., McGraw-Hill, New York.

PICINBONO, B. 1978. "Adaptive Signal Processing for Detection and Communication," in *Communication Systems and Random Process Theory*, J. K. Skwirzynski, ed., Sijthoff en Noordhoff, Alphen aan den Rijn, The Netherlands.

PISARENKO, V. F. 1973. "The Retrieval of Harmonics from a Convariance Function," *Geophys. J. R. Astron. Soc.*, Vol. 33, pp. 347–366.

POOLE, M. A. 1981. *Autoregressive Methods of Spectral Analysis*, E.E. degree thesis, Department of Electrical and Computer Engineering, Northeastern University, Boston, May.

PRICE, R. 1990. "Mirror FFT and Phase FFT Algorithms," unpublished work, Raytheon Research Division, May.

PROAKIS, J. G. 1970. "Adaptive Digital Filters for Equalization of Telephone Channels," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-18, pp. 195–200, June.

PROAKIS, J. G. 1974, "Channel Identification for High Speed Digital Communications," *IEEE Trans. Automatic Control*, Vol. AC-19, pp. 916–922, December.

PROAKIS, J. G. 1975. "Advances in Equalization for Intersymbol Interference," in *Advances in Communication Systems*, Vol. 4, A. J. Viterbi, ed., Academic, New York.

PROAKIS, J. G. 1989. *Digital Communications*, 2nd ed., McGraw-Hill, New York.

PROAKIS, J. G., and MILLER, J. H. 1969. "Adaptive Receiver for Digital Signaling through Channels with Intersymbol Interference," *IEEE Trans. Information Theory*, Vol. IT-15, pp. 484–497, July.

PROAKIS, J. G., RADER, C. M., LING, F., NIKIAS, C. L., MOONEN, M. and PROUDLER, I. K. 2002. *Algorithms for Statistical Signal Processing*, Prentice-Hall, Upper Saddle River, NJ.

PROAKIS, J. G. 2001. *Digital Communications*, 4th ed., McGraw-Hill, New York.

QI, R., COAKLEY, F., and EVANS, B. 1996. "Practical Consideration for Bandpass Sampling." *Electronics Letters*, Vol. 32(20), pp. 1861–1862, September.

RABINER, L. R., and SCHAEFER, R. W. 1974a. "On the Behavior of Minimax Relative Error FIR Digital Differentiators," *Bell Syst. Tech. J.*, Vol. 53, pp. 333–362, February.

RABINER, L. R., and SCHAFER, R. W. 1974b. "On the Behavior of Minimax FIR Digital Hilbert Transformers," *Bell Syst. Tech. J.*, Vol. 53, pp. 363–394, February.

RABINER, L. R., and SCHAFER, R. W. 1978. *Digital Processing of Speech Signals*, Prentice Hall, Upper Saddle River, NJ.

RABINER, L. R., SCHAFER, R. W., and RADER, C. M. 1969. "The Chirp z-Transform Algorithm and Its Applications," *Bell Syst. Tech. J.*, Vol. 48, pp. 1249–1292, May–June.

RABINER, L. R., GOLD, B., and MCGONEGAL, C. A. 1970. "An Approach to the Approximation Problem for Nonrecursive Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-18, pp. 83–106, June.

RABINER, L. R., MCCLELLAN, J. H., and PARKS, T. W. 1975. "FIR Digital Filter Design Techniques Using Weighted Chebyshev Approximation," *Proc. IEEE*, Vol. 63, pp. 595–610, April.

RADER, C. M. 1970. "An Improved Algorithm for High-Speed Auto-correlation with Applications to Spectral Estimation," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-18, pp. 439–441, December.

RADER, C. M., and BRENNER, N. M. 1976. "A New Principle for Fast Fourier Transformation," *IEEE Trans. Acoustics, Speech and Signal Processing*, Vol. ASSP-24, pp. 264–266, June.

RADER, C. M., and GOLD, B. 1967a. "Digital Filter Design Techniques in the Frequency Domain," *Proc. IEEE*, Vol. 55, pp. 149–171, February.

RADER, C. M., and GOLD, B. 1967b. "Effects of Parameter Quantization on the Poles of a Digital Filter," *Proc. IEEE*, Vol. 55, pp. 688–689, May.

RAMSTAD, T. A. 1984. "Digital Methods for Conversion Between Arbitrary Sampling Frequencies," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 577–591, June.

RAO, K., and HUANG, J. 1996. *Techniques and Standards for Image, Video, and Audio Coding*, Prentice Hall, Upper Saddle River, NJ.

RAO, K. R., and YIP, P 1990. *Discrete Cosine Transform*, Academic Press, Boston.

REMEZ, E. YA. 1957. *General Computational Methods of Chebyshev Approximation*, Atomic Energy Translation 4491, Kiev, USSR.

RICE, D., and WU, K. 1982. "Quatature Sampling with High Dynamic Range."*IEEE Trans. Aerospace and Electronic Systems*, Vol. 18(4), pp. 736–739, November.

RISSANEN, J. 1983. "A Universal Prior for the Integers and Estimation by Minimum Description Length," *Ann. Stat.*, Vol. 11, pp. 417–431.

ROBERTS, R. A., and MULLIS, C. T. 1987. *Digital Signal Processing*, Addison-Wesley, Reading, MA.

ROBINSON, E. A. 1962. *Random Wavelets and Cybernetic Systems*, Charles Griffin, London.

ROBINSON, E. A. 1982. "A Historical Perspective of Spectrum Estimation," *Proc. IEEE*, Vol. 70, pp. 885–907, September.

ROBINSON, E. A., and TREITEL, S. 1978. "Digital Signal Processing in Geophysics," in *Applications of Digital Signal Processing*, A. V. Oppenheim, ed., Prentice Hall, Upper Saddle River, NJ.

ROBINSON, E. A., and TREITEL, S. 1980. *Geophysical Signal Analysis*, Prentice Hall, Upper Saddle River, NJ.

ROY, R., PAULRAJ, A., and KAILATH, T. 1986. "ESPRIT: A Subspace Rotation Approach to Estimation of Parameters of Cisoids in Noise," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp. 1340–1342, October.

SAFRANEK, R. J., MACKAY, K. JAYANT, N. W., and KIM, T. 1988. "Image Coding Based on Selective Quantization of the Reconstruction Noise in the Dominant Sub-Band," *Proc. 1988 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 765–768, April.

SAKAI, H. 1979. "Statistical Properties of AR Spectral Analysis," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-27, pp. 402–409, August.

SANDBERG, I. W., and KAISER, J. F. 1972. "A Bound on Limit Cycles in Fixed-Point Implementations of Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-20, pp. 110–112, June.

SATORIUS, E. H., and ALEXANDER J. T. 1978. "High Resolution Spectral Analysis of Sinusoids in Correlated Noise," *Proc. 1978 ICASSP*, pp. 349–351, Tulsa, OK, April 10–12.

SAYED, A. H. 2003 *Adaptive Filters*, Wiley, New York.

SCHAFER, R. W., and RABINER, L. R. 1973. "A Digital Signal Processing Approach to Interpolation," *Proc. IEEE*, Vol. 61, pp. 692–702, June.

SCHEUERMANN, H., and GOCKLER, H. 1981. "A Comprehensive Survey of Digital Transmultiplexing Methods," *Proc. IEEE*, Vol. 69, pp. 1419–1450.

SCHMIDT, R. D. 1981. "A Signal Subspace Approach to Multiple Emitter Location and Spectral Estimation," Ph.D. dissertation, Department of Electrical Engineering, Stanford University, Stanford, CA, November.

SCHMIDT, R. D. 1986. "Multiple Emitter Location and Signal Parameter Estimation," *IEEE Trans. Antennas and Propagation*, Vol. AP 34, pp. 276–280, March.

SCHOTT, J. P., and MCCLELLAN, J. H. 1984. "Maximum Entropy Power Spectrum Estimation with Uncertainty in Correlation Measurements," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 410–418, April.

SCHUR, I. 1917. "On Power Series Which Are Bounded in the Interior of the Unit Circle," *J. Reine Angew. Math.*, Vol. 147, pp. 205–232, Berlin. For an English translation of the paper, see Gohberg 1986.

SCHUSTER, SIR ARTHUR. 1898. "On the Investigation of Hidden Periodicities with Application to a Supposed Twenty-Six-Day Period of Meteorological Phenomena," *Terr. Mag.*, Vol. 3, pp. 13–41, March.

SEDLMEYER, A., and FETTWEIS, A. 1973. "Digital Filters with True Ladder Configuration," *Int. J. Circuit Theory Appl.*, Vol. 1, pp. 5–10, March.

SHANKS, J. L. 1967. "Recursion Filters for Digital Processing," *Geophysics*, Vol. 32, pp. 33–51, February.

SHANNON, C. E. 1949. "Communication in the Presence of Noise," *Proc. IRE*, pp. 10–21, January.

SHEINGOLD, D. H., ed. 1986. *Analog-Digital Conversion Handbook*, Prentice Hall, Upper Saddle River, NJ.

SIEBERT, W. M. 1986. *Circuits, Signals and Systems*, McGraw-Hill, New York.

SINGLETON, R. C. 1967. "A Method for Computing the Fast Fourier Transform with Auxiliary Memory and Limit High Speed Storage," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-15, pp. 91–98, June.

SINGLETON, R. C. 1969. "An Algorithm for Computing the Mixed Radix Fast Fourier Transform," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-17, pp. 93–103, June.

SLOCK, D. T. M., and KAILATH, T 1988. "Numerically Stable Fast Recursive Least Squares Transversal Filters," *Proc. 1988 Int. Conf. Acoustics, Speech, and Signal Processing*, pp. 1364–1368, NY, April.

SLOCK, D. T. M., and KAILATH, T 1991. "Numerically Stable Fast Transversal Filters for Recursive Least Squares Adaptive Filtering," *IEEE Trans. Signal Processing*, Vol. 39, pp. 92–114, January.

SMITH, M. J. T., and BARWELL, T. P. 1984. "A Procedure for Designing Exact Reconstruction Filter Banks for Tree Structured Subband Coders," *Proc. 1984 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 27.1.1–27.1.4, San Diego, March.

SMITH, M. J. T., and EDDINS, S. L. 1988. "Subband Coding of Images with Octave Band Tree Structures," *Proc. 1987 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1382–1385, Dallas, April.

STARK, H, and WOODS, J. W. 1994. *Probability, Random Processes, and Estimation Theory for Engineers*, 2nd Ed., Prentice Hall, Upper Saddle River, NJ.

STEIGLITZ, K. 1965. "The Equivalence of Digital and Analog Signal Processing," *Inf. Control*, Vol. 8, pp. 455–467, October.

STEIGLITZ, K. 1970. "Computer-Aided Design of Recursive Digital Filters," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-18, pp. 123–129, June.

STOCKHAM, T. G. 1966. "High Speed Convolution and Correlation," *1966 Spring Joint Computer Conference, AFIPS Proc.*, Vol. 28, pp. 229–233.

STORER, J. E. 1957. *Passive Network Synthesis*, McGraw-Hill, New York.

STRANG, G. 1999. "The Discrete Cosine Transform." *SIAM Review*, Vol. 41(1), pp. 135–137.

SWARZTRAUBER, P. 1986. "Symmetric FFT's," *Mathematics of Computation*, Vol. 47, pp. 323–346, July.

SWINGLER, D. N. 1979a. "A Comparison Between Burg's Maximum Entropy Method and a Nonrecursive Technique for the Spectral Analysis of Deterministic Signals," *J. Geophys. Res.*, Vol. 84, pp. 679–685, February.

SWINGLER, D. N. 1979b. "A Modified Burg Algorithm for Maximum Entropy Spectral Analysis," *Proc. IEEE*, Vol. 67, pp. 1368–1369, September.

SWINGLER, D. N. 1980. "Frequency Errors in MEM Processing," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, pp. 257–259, April.

SZEGO, G. 1967. *Orthogonal Polynomials*, 3d ed., Colloquium Publishers, No. 23, American Mathematical Society, Providence, RI.

THORVALDSEN, T. 1981. "A Comparison of the Least-Squares Method and the Burg Method for Autoregressive Spectral Analysis," *IEEE Trans. Antennas and Propagation*, Vol. AP-29, pp. 675–679, July.

TONG, H. 1975. "Autoregressive Model Fitting with Noisy Data by Akaike's Information Criterion," *IEEE Trans. Information Theory*, Vol. IT-21, pp. 476–480, July.

TONG, H. 1977. "More on Autoregressive Model Fitting with Noise Data by Akaike's Information Criterion," *IEEE Trans. Information Theory*, Vol. IT-23, pp. 409–410, May.

TRETTER, S. A. 1976. *Introduction to Discrete-Time Signal Processing*, Wiley, New York.

TUFTS, D. W., and KUMARESAN, R. 1982. "Estimation of Frequencies of Multiple Sinusoids: Making Linear Prediction Perform Like Maximum Likelihood," *Proc. IEEE*, Vol. 70, pp. 975–989, September.

ULRYCH, T. J., and BISHOP, T. N. 1975. "Maximum Entropy Spectral Analysis and Autoregressive Decomposition," *Rev. Geophys. Space Phys.*, Vol. 13, pp. 183–200, February.

ULRYCH, T. J., and CLAYTON, R. W. 1976. "Time Series Modeling and Maximum Entropy," *Phys. Earth Planet. Inter.*, Vol. 12, pp. 188–200, August.

UNGERBOECK, G. 1972. "Theory on the Speed of Convergence in Adaptive Equalizers for Digital Communication," *IBM J. Res. Devel.*, Vol. 16, pp. 546–555, November.

VAIDYANATHAN, P. P. 1990. "Multirate Digital Filters, Filter Banks, Polyphase Networks, and Applications: A Tutorial," *Proc. IEEE*, Vol. 78, pp. 56–93, January.

VAIDYANATHAN, P. P. 1993. *Multirate Systems and Filter Banks*, Prentice Hall, Upper Saddle River, NJ.

VAUGHAN, R., SCOTT, N., and WHITE, D. 1991. "The Theory of Bandpass Sampling." *IEEE Trans. on signal Procesing*, Vol. 39(9), pp. 1973–1984.

VETTERLI, J. 1984. "Multi-dimensional Sub-Band Coding: Some Theory and Algorithms," *Signal Processing*, Vol. 6, pp. 97–112, April.

VETTERLI, J. 1987. "A Theory of Multirate Filter Banks," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, pp. 356–372, March.

VIEIRA, A. C. G. 1977. "Matrix Orthogonal Polynomials with Applications to Autoregressive Modeling and Ladder Forms," Ph.D. dissertation, Department of Electrical Engineering, Stanford University, Stanford, CA, December.

WALKER, G. 1931. "On Periodicity in Series of Related Terms," *Proc. R. Soc.*, Ser. A, Vol. 313, pp. 518–532.

WANG, Z. 1984. "Fast Algorithms for the Discrete *W* Transform for the Discrete foruier Transform." *IEEE Trans. on ASSP*, Vol. 32(4), pp. 803–816, August.

WATERS, W,, and JARRETT, B. 1982. "Bandpass Signal Sampling and Coherent Dection." *IEEE Trans. on Aerospace and Electronic Systems*, Vol. 18(4), pp. 731–736, November.

WAX, M., and KAILATH, T. 1985. "Detection of Signals by Information Theoretic Criteria," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, pp. 387–392, April.

WEINBERG, L. 1962. *Network Analysis and Synthesis*, McGraw-Hill, New York.

WELCH, P. D. 1967. "The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short Modified Periodograms," *IEEE Trans. Audio and Electroacoustics*, Vol. AU-15, pp. 70–73, June.

WIDROW, B., and HOFF, M. E., Jr. 1960. "Adaptive Switching Circuits," *IRE WESCON Conv. Rec.*, pt. 4, pp. 96–104.

WIDROW, B., MANTEY, P., and GRIFFITHS, L. J. 1967. "Adaptive Antenna Systems," *Proc. IEEE*, Vol. 55, pp. 2143–2159, December.

WIDROW, B. et al. 1975. "Adaptive Noise Cancelling Principles and Applications," *Proc. IEEE*, Vol. 63, pp. 1692–1716, December.

WIDROW, B., MANTEY, P., and GRIFFITHS, L. J. 1967. "Adaptive Antenna syatems." *Proc. IEEE*, Vol. 55, pp. 2143–2159, December.

WIENER, N. 1949. *Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications*, Wiley, New York.

WIENER, N., and PALEY, R. E. A. C. 1934. *Fourier Transforms in the Complex Domain*, American Mathematical Society, Providence, RI.

WINOGRAD, S. 1976. "On Computing the Discrete Fourier Transform," *Proc. Natl. Acad. Sci.*, Vol. 73, pp. 105–106.

WINOGRAD, S. 1978. "On Computing the Discrete Fourier Transform," *Math. Comp.*, Vol. 32, pp. 177–199.

WOLD, H. 1938. *A Study in the Analysis of Stationary Time Series*, reprinted by Almqvist & Wiksell, Stockholm, 1954.

WOOD, L. C., and TREITEL, S. 1975. "Seismic Signal Processing," *Proc. IEEE*, Vol. 63, pp. 649–661, April.

WOODS, J. W., and O'NEIL, S. D. 1986. "Subband Coding of Images," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, pp. 1278–1288, October.

YOULA, D., and KAZANJIAN, N. 1978. "Bauer-Type Factorization of Positive Matrices and the Theory of Matrices Orthogonal on the Unit Circle," *IEEE Trans. Circuits and Systems*, Vol. CAS-25, pp. 57–69, January.

YULE, G. U. 1927. "On a Method of Investigating Periodicities in Disturbed Series with Special References to Wolfer's Sunspot Numbers," *Philos. Trans. R. Soc. London*, Ser. A, Vol. 226, pp. 267–298, July.

ZADEH, L. A., and DESOER, C. A. 1963. *Linear System Theory: The State-Space Approach*, McGraw-Hill, New York.

ZVEREV, A. I. 1967. *Handbook of Filter Synthesis*, Wiley, New York.

Answers to Selected Problems

Chapter 1

1.1 **(a)** One dimensional, multichannel, discrete time, and digital.

(b) Multi dimensional, single channel, continuous-time, analog.

(c) One dimensional, single channel, continuous-time, analog.

(d) One dimensional, single channel, continuous-time, analog.

(e) One dimensional, multichannel, discrete-time, digital.

1.3 **(a)** Periodic with period $T_p = \frac{2\pi}{5}$.

(c) $f = \frac{1}{12\pi} \Rightarrow$ non-periodic.

(e) $\cos\left(\frac{\pi n}{2}\right)$ is periodic with period $N_p = 4$; $\sin\left(\frac{\pi n}{8}\right)$ is periodic with period $N_p = 16$; $\cos\left(\frac{\pi n}{4} + \frac{\pi}{3}\right)$ is periodic with period $N_p = 8$. Therefore, $x(n)$ is periodic with period $N_p = 16$ (16 is the least common multiple of 4, 8, 16).

1.4 **(b)** $N = 7; k = 0\ 1\ 2\ 3\ 4\ 5\ 6\ 7$; GCD$(k, N) = 7\ 1\ 1\ 1\ 1\ 1\ 1\ 7$; $N_p = 1\ 7\ 7\ 7\ 7\ 7\ 1$.

1.5 **(a)** Yes. $x(1) = 3 = 3\sin\left(\frac{100\pi}{F_s}\right) \Rightarrow 200$ samples/sec.

1.6 **(b)** If $x(n)$ is periodic, then $f = k/N$ where N is the period. Then, $T_d = \left(\frac{k}{f}T\right) = k\left(\frac{T_p}{T}\right)T = kT_p$. Thus, it takes k periods (kT_p) of the analog signal to make 1 period (T_d) of the discrete signal.

1.8 **(a)** $F_{\max} = 100Hz$, $F_s \geq 2F_{\max} = 200$ Hz.

(b) $F_{fold} = \frac{F_s}{2} = 125$ Hz.

1.10 **(a)** $F_s = 1000$ samples/sec; $F_{fold} = 500$ Hz.

(b) $F_{\max} = 900$ Hz; $F_N = 2F_{\max} = 1800$ Hz.

(c) $f_1 = 0.3$; $f_2 = 0.9$; But $f_2 = 0.9 > 0.5 \Rightarrow f_2 = 0.1$. Hence, $x(n) = 3\cos[(2\pi)(0.3)n] + 2\cos[(2\pi)(0.1)n]$.

(d) $\Delta = \frac{10}{1023}$.

Chapter 2

2.7 **(a)** Static, nonlinear, time invariant, causal, stable.

(c) Static, linear, time variant, causal, stable.

(e) Static, nonlinear, time invariant, causal, stable.

(h) Static, linear, time invariant, causal, stable.

(k) Static, nonlinear, time invariant, causal, stable.

2.11 Since the system is linear and $x_1(n) + x_2(n) = \delta(n)$, it follows that the impulse response of the system is $y_1(n) + y_2(n) = \left\{0, \underset{\uparrow}{3}, -1, 2, 1\right\}$.

If the system were time invariant, the response to $x_3(n)$ would be $\left\{\underset{\uparrow}{3}, 2, 1, 3, 1\right\}$. But this is not the case.

2.16 **(b)** **(1)** $y(n) = h(n) * x(n) = \{1, 3, 7, 7, 7, 6, 4\}$; $\sum_n y(n) = 35$, $\sum_k h(k) = 5$, $\sum_k x(k) = 7$.
(4) $y(n) = \{1, 2, 3, 4, 5\}$; $\sum_n y(n) = 15$, $\sum_n h(n) = 1$, $\sum_n x(n) = 15$
(7) $y(n) = \{0, 1, 4, -4, -5, -1, 3\}$ $\sum_n y(n) = -2$, $\sum_n h(n) = -1$, $\sum_n x(n) = 2$
(10) $y(n) = \{1, 4, 4, 4, 10, 4, 4, 4, 1\}$; $\sum_n y(n) = 36$, $\sum_n h(n) = 6$, $\sum_n x(n) = 6$

2.19 $y(n) = \sum_{k=0}^{4} h(k)x(n-k)$, $x(n) = \left\{a^{-3}, a^{-2}, a^{-1}, \underset{\uparrow}{1}, a, \ldots, a^5\right\}$, $h(n) = \left\{\underset{\uparrow}{1}, 1, 1, 1, 1, \right\}$; $y(n) = \sum_{k=0}^{4} x(n-k)$, $-3 \le n \le 9$; $y(n) = 0$, otherwise

2.22 **(a)** $y_1(n) = x(n) + x(n-1) = \{1, 5, 6, 5, 8, 8, 6, 7, 9, 12, 15, 9\}$
$y_4(n) = \{0.25, 1.5, 2.75, 2.75, 3.25, 4, 3.5, 3.25, 3.75, 5.25, 6.25, 7, 6, 2.25\}$

2.26 With $x(n) = 0$, we have $y(n-1) + \frac{4}{3}y(n-1) = 0$ $y(-1) = -\frac{4}{3}y(-2)$; $y(0) = \left(-\frac{4}{3}\right)^2 y(-2)$; $y(1) = \left(-\frac{4}{3}\right)^3 y(-2)$

Therefore, $y(k) = \left(-\frac{4}{3}\right)^{k+2} y(-2)$ ← zero-input response.

2.32 **(a)** $L_1 = N_1 + M_1$ and $L_2 = N_2 + M_2$

(c) $N_1 = -2$, $N_2 = 4$, $M_1 = -1$, $M_2 = 2$

Partial overlap from left: $n = -3$ $n = -1$ $L_1 = -3$; Full overlap: $n = 0$ $n = 3$; Partial overlap from right: $n = 4$ $n = 6$ $L_2 = 6$

2.34 $h(n) = \left\{\underset{\uparrow}{1}, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}\right\}$; $y(n) = \left\{\underset{\uparrow}{1}, 2, 2, 5, 3, 3, 3, 2, 1, 0\right\}$

Then, $x(n) = \left\{1.\frac{3}{2}, \frac{3}{2}, \frac{7}{4}, \frac{3}{2}\right\}$

2.38 $\sum_{n=-\infty}^{\infty} |h(n)| = \sum_{n=0,\text{neven}}^{\infty} |a|^n$; $\sum_{n=-}^{\infty} |a|^{2n}$; $= \frac{1}{1-|a|^2}$

Stable if $|a| < 1$

2.40 $y(n) = 2\left[1 - (\frac{1}{2})^{n+1}\right]u(n) - 2\left[1 - (\frac{1}{2})^{a-9}\right]u(n-10)$

2.41 **(a)** $y(n) = \frac{2}{3}\left[2^{n+1} - (\frac{1}{2})^{n+1}\right]u(n)$

2.42 **(a)** $h_c(n) = h_1(n) * h_2(n) * h_3(n) = [\delta(n) - \delta(n-1)] * u(n) * h(n) = h(n)$

(b) No.

2.45 $y(n) = -\frac{1}{2}y(n-1) + z(n) + 2x(n-2)$; $y(-2) = 1$, $y(-1) = \frac{3}{2}$, $y(9) = \frac{17}{4}$, $y(1) = \frac{47}{8}, \cdots$

2.48 **(a)** $y(n) = ay(n-1) + bx(n) \Rightarrow h(n) = ba^n u(n)$ $\sum_{n=0}^{\infty} h(n) = \frac{b}{1-a} = 1 \Rightarrow b = 1 - a$.

(b) $s(n) = \sum_{k=0}^{n} h(n-k) = b\left[\frac{1-a^{n+1}}{1-a}\right]u(n)$

$s(\infty) = \frac{b}{1-a} = 1 \Rightarrow b = 1 - a$

2.51 **(a)** $y(n) = \frac{1}{3}x(n) + \frac{1}{3}x(n-3) + y(n-1)$ for $x(n) = \delta(n)$, we have
$h(n) = \left\{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3} \ldots\right\}$

(b) $y(n) = \frac{1}{2}y(n-1) + \frac{1}{8}y(n-2) + \frac{1}{2}x(n-2)$, $y(-1) = y(-2) = 0$
with $x(n) = \delta(n)$, $h(n) = \left\{0, 0, \frac{1}{2}, \frac{1}{4}, \frac{3}{16}, \frac{1}{9}, \frac{11}{128}, \frac{15}{256}, \frac{41}{1024} \ldots\right\}$

(c) $y(n) = 1.4y(n-1) - 0.48y(n-2) + x(n)$, $y(-1) - y(-2) = 0$
with $x(n)\delta(n)$, $h(n) = \{1, 1, 4, 1.48, 1.4, 1.2496, 1.0774, 0.9086, \ldots\}$

(d) All three systems are IIR.

2.54 **(a)** convolution: $y_1(n) = \left\{\underset{\rightarrow}{1}3, 7, 7, 7, 7, 4\right\}$

correlation: $\gamma_1(n) = \left\{1, 3, 7, 7, \underset{\rightarrow}{7}, 6, 4\right\}$

(c) convolution: $y_4(n) = \left\{\underset{\rightarrow}{1}, 4, 10, 20, 25, 24, 16\right\}$

correlation: $\gamma_4(n) = \left\{4, 11, 20, \underset{\rightarrow}{30}, 20, 11, 4\right\}$

2.58 $h(n) = [c_1 2^n + c_2 n 2^n]u(n)$

With $y(0) = 1$, $y(1) = 3$, we have, $c_1 = 1$, and $c_2 = \frac{1}{2}$.

2.61 $x(n) = \begin{cases} 1, & n_0 - N \le n \le n_0 + N \\ 0 & \text{otherwise} \end{cases}$.

$r_{xx}(l) = \begin{cases} 2N + 1 - |l|, & -2N \le l \le 2N \\ 0, & \text{otherwise} \end{cases}$.

2.63 $r_{xx}(l) = \sum_{n=-\infty}^{\infty} x(n)x(n-l) = \begin{cases} 2N + 1 - |l|, & -2N \le l \le 2N \\ 0, & \text{otherwise} \end{cases}$.

Since $r_{xx}(0) = 2N + 1$, the normalized autocorrelation is

$\rho_{xx}(l) = \begin{cases} \frac{1}{2N+1}(2N + 1 - |l|), & -2N \le l \le 2N \\ 0, & \text{otherwise} \end{cases}$

Chapter 3

3.1 **(a)** $X(z) = 3z^5 + 6 + z^{-1} - 4z^{-2}$ ROC $: 0 < |z| < \infty$

3.2 **(a)** $X(z) = \frac{1}{(1-z^{-1})^2}$

(d) $X(z) = \frac{\left[az^{-1} - (az^{-1})^3\right]\sin w_0}{\left[1 - (2a\cos w_0)z^{-1} + a^2 z^{-2}\right]^2}$, $|z| < a$

(h) $X(z) = \frac{1 - \left(\frac{1}{2}z^{-1}\right)^{10}}{1 - \frac{1}{2}z^{-1}}$, $|z| > \frac{1}{2}$

3.4 **(a)** $X(z) = \frac{z^{-1}}{(1+z^{-1})^2}$, $|z| > 1$

(f) $X(z) = 1 - z^{-2} + z^{-4} - z^{-5}$, $z \neq 0$

3.8 **(a)** $Y(z) = \frac{X(z)}{1-z^{-1}}$

3.12 **(a)** $x(n) = [4(2)^n - 3 - n]u(n)$

3.14 **(a)** $x(n) = [2(-1)^n - (-2)^n]u(n)$

(c) $x(n) = -\left[\frac{3}{5}\left(\frac{1}{\sqrt{2}}\right)^n \cos\frac{\pi}{4}n + \frac{23}{10}\left(1\sqrt{2}\right)^n \sin\frac{\pi}{4}n + \frac{17}{20}(12)^n\right]u(n)$

(j) $x(n) = \left(-\frac{1}{a}\right)^{n+1}u(n) + \left(\frac{1}{a}\right)^{n-1}u(n-1)$

3.16 **(a)** $y(n) = \left[-\frac{4}{3}\left(\frac{1}{4}\right)^n + \frac{1}{3} + \left(\frac{1}{2}\right)^n\right]u(n)$

(d) $y(n) = [-2(n+1) + 2^{n+1}]u(n)$

3.19 **(b)** $x(n) = -\left(\frac{1}{n}\right)\left(\frac{1}{2}\right)^n u(n-1)$

3.24 **(a)** $x(n) = [0.136(0.28)^n + 0.864(-1.78)^n]u(n)$

3.35 **(a)** $y(n) = \left[\frac{1}{7}\left(\frac{1}{3}\right)^n \frac{6}{7}\left(\frac{1}{2}\right)^n \cos\frac{\pi n}{3} + \frac{3\sqrt{3}}{7}\left(\frac{1}{2}\right)^n \sin\frac{\pi n}{3}\right]u(n)$

(d) $y(n) = \frac{10}{\sqrt{2}}\sin\left(\frac{\pi n}{2} + \frac{\pi}{4}\right)u(n)$

(h) $y(n) = \left[4\left(\frac{1}{2}\right)^n - n\left(\frac{1}{4}\right)^n - 3\left(\frac{1}{4}\right)^n\right]u(n)$

3.38 **(a)** $h(n) = \left[2\left(\frac{1}{2}\right)^n - \left(\frac{1}{4}\right)^n\right]u(n)$

$y(n) = \left[\frac{8}{3} - 2\left(\frac{1}{2}\right)^n + \frac{1}{3}\left(\frac{1}{4}\right)^n\right]u(n)$

(d) $h(n) = \left[2\left(\frac{2}{5}\right)^n - \left(\frac{1}{5}\right)^n\right]u(n)$

$y(n) = \left[\frac{25}{12} + \frac{1}{4}\left(\frac{1}{5}\right)^n - \frac{4}{3}\left(\frac{2}{5}\right)^n\right]u(n)$

3.42 **(a)** $h(n) = \left[-\frac{7}{2}\left(\frac{1}{5}\right)^{n-1} + \frac{9}{2}\left(\frac{2}{5}\right)^{n-1}\right]u(n-1)$

3.44 **(a)** $h(n) = b_0\delta(n) + (b1 - b_0a_1)(-a_1)^{n-1}u(n-1)$

(b) $y(n) = \left[\frac{b_0+b_1}{1+a_1} + \frac{a_1b_0-b_1}{1+a_1}(-a_1)^n\right]u(n)$

3.49 **(d)** $y(n) = \left[\frac{4}{3} - \frac{3}{8}\left(\frac{1}{2}\right)^n + \frac{7}{24}\left(-\frac{1}{2}\right)^n\right]u(n)$

3.56 **(a)** $x(n) = \left(\frac{1}{2}\right)^n u(n)$

(d) $x(n) = \left[\frac{3}{10}\left(\frac{1}{2}\right)^n \frac{7}{10}\left(-\frac{1}{3}\right)^n\right]u(n)$

3.58 ROC: $a < |z| < 1/a$

$x(-18) = -32/15309$

Chapter 4

4.2 **(a)** $X_a(f) = \frac{A}{a+j2\pi\pi F}$; $|X_a(F)| = \frac{A}{\sqrt{a^2+(2\pi F)}}$; $\angle X_a(f) = -\tan -1\frac{2\pi f}{a}$

4.5 **(a)** $x(n) = \left\{\frac{11}{2}, 2 + \frac{3}{4}\sqrt{2}, 1, 2 - \frac{3}{4}\sqrt{2}, \frac{1}{2}, 2 - \frac{3}{4}\sqrt{2}, 1, 2 + \frac{3}{4}\sqrt{2}\right\}$

Hence, $c_0 = 2$, $c_1 = c_7 = 1$, $c_2 = c_6 = \frac{1}{2}$, $c_3 = c_5 = \frac{1}{4}$, $c_4 = 0$

(b) $P = \frac{53}{8}$

4.6 **(a)**

$$c_k = \begin{cases} \frac{1}{2j}, & k = 3 \\ \frac{1}{2}, & k = 5 \\ \frac{1}{2}, & k = 10 \\ \frac{-1}{2j}, & k = 12 \\ 0, & \text{otherwise} \end{cases}$$

(d) $c_0 = 0$,

$c_1 = \frac{2j}{5}\left[-\sin\left(\frac{2\pi}{5}\right) + 2\sin\left(\frac{4\pi}{5}\right)\right]$

$c_2 = \frac{2j}{5}\left[\sin\left(\frac{4\pi}{5}\right) - 2\sin\left(\frac{2\pi}{5}\right)\right]$;

$c_3 = -c_2$; $c_4 = -c_1$

(h) $N = 2$; $c_k = \frac{1}{2}(1 - e^{-j\pi k})$; $c_0 = 0$, $c_1 = -1$

4.9 **(a)** $X(\omega) = \frac{1-e^{-j6\omega}}{1-e-j\omega}$

(e) $\sum_{n=-\infty}^{\infty} |x(n)| \to \infty$.

Therefore, the Fourier transform does not exist.

(h) $X(\omega) = (2M+1)A + 2A\sum_{k=1}^{M}(2M+1-k)\cos wk$

4.14 **(a)** $X(0) = \sum_n x(n) = -1$;

(e) $\int_{-\pi}^{\pi} |X(\omega|^2 d\omega = 2\pi\sum_n |x(n)|^2 = (2\pi)(19) = 38\pi$

4.17 **(a)** $\sum_n x^*(n)e^{-j\omega n} = \left(\sum_n x(n)e^{-j(-\omega)n)}\right)^* = X^*(-\omega)$

(c) $Y(\omega) = X(\omega) + X(\omega)e^{-j\omega} = (1 - e^{-j\omega})X(\omega)$

(f) $Y(\omega) = X(2\omega)$

4.19 **(a)** $X_1(\omega) = \frac{1}{2j}\left[X\left(\omega - \frac{\pi}{4}\right) + X\left(\omega + \frac{\pi}{4}\right)\right]$

(d) $X_4(\omega) = \frac{1}{2}[X(\omega - \pi) + X(\omega + \pi)] = X(\omega - \pi)$

4.22 **(a)** $X_1(\omega) = \frac{e^{j\omega/2}}{1 - ae^{j\omega/2}}$

(d) $X_4(\omega) = \frac{1}{2}[X(\omega - 0.3\pi) + X(\omega + 0.3\pi)]$

4.23 **(b)** $Y_2(\omega) = X\left(\frac{\omega}{2}\right)$

Chapter 5

5.1 **(a)** Because the range of n is $(-\infty, \infty)$, the fourier transforms of $x(n)$ and $y(n)$ do not exist. However, the relationship implied by the forms of $x(n)$ and $y(n)$ is $y(n) = x^3(n)$. In this case, the system H_1 is non-linear.

(b) In this case, $H(\omega) = \frac{1 - \frac{1}{2}e^{-j\omega}}{1 - \frac{1}{8}e^{-j\omega}}$, so the system is LTI.

5.3 **(a)** $|H(\omega)| = \frac{1}{\left[\frac{5}{4} - \cos\omega\right]^{\frac{1}{2}}}$ $\angle H(\omega) = -\tan^{-1}\frac{\frac{1}{2}\sin\omega}{1 - \frac{1}{2}\cos\omega}$

5.4 **(a)** $H(\omega) = (\sin\omega)e^{j\pi/2}$

(j) $H(\omega) = (\cos\omega)\left(\cos\frac{\omega}{2}\right)e^{-j3\omega/2}$

(n) $H(\omega) = (\sin^2\omega/2)e^{-j(\omega-\pi)}$

5.8 $y_{ss}(n) = 3\cos\left(\frac{\pi}{2}n + 60^0\right)$; $y_{tr}(n) = 0$

5.12 **(a)** $H(\omega) = \frac{b}{1 - 0.9e^{-j\omega}}$; $|H(0)| = 1, \Rightarrow b = \pm 0.1$

$$\Theta(\omega) = \begin{cases} -\frac{\omega M}{2}, & \cos\frac{\omega M}{2} > 0 \\ \pi - \frac{\omega M}{2}, & \cos\frac{\omega M}{2} < 0 \end{cases}$$

(b) $|H(\omega_0)|^2 = \frac{1}{2} \Rightarrow \frac{b^2}{1.81 - 1.8\cos\omega_0} = \frac{1}{2} \Rightarrow \omega_0 = 0.105$

5.16 **(a)** $|H(\omega)| = \frac{4}{5 - 3\cos\omega}$; $\angle H(\omega) = 0$

5.18 **(a)** $H(\omega) = 2e^{-j2\omega}e^{j\pi/2}\sin 2\omega$

(b) $y(n) = 2\cos\pi n/4$, $H\left(\frac{\pi}{4}\right) = 2$, $\angle H\left(\frac{\pi}{4}\right) = 0$

5.20 **(a)** $Y(\omega) = X\left(\frac{\omega}{2}\right) = \begin{cases} 1, & |\omega| \le \frac{\pi}{2} \\ 0, & \frac{\pi}{2} \le |\omega| \le p \end{cases}$

(b) $Y(\omega) = \frac{1}{2}[X(\omega - \pi) + X(\omega + \pi)] = \begin{cases} 0, & |\omega| \le \frac{3\pi}{4} \\ \frac{1}{2}, & \frac{3\pi}{4} \le |\omega| < \pi \end{cases}$

5.22 **(a)** $|H(\omega)| = 2|\sin 5\omega\}$,

$$\Theta(\omega) = \begin{cases} \frac{\pi}{2} - 5\omega & \text{for } \sin 5\omega > 0 \\ \frac{\pi}{2} - 5\omega + \pi, & \text{for } \sin 5\omega < 0 \end{cases}$$

(b) $\left|H\left(\frac{\pi}{10}\right)\right| = 2$, $\angle H\left(\frac{\iota}{\pi}10\right) = 0$; $\left|H\left(\frac{\pi}{3}\right)\right| = \sqrt{3}$, $\Theta\left(\frac{\pi}{3}\right) = \angle H\left(\frac{\pi}{3}\right) = -\frac{\pi}{6}$

(1) Hence, $y(n) = 2\cos\frac{\pi}{10}n + 3\sqrt{3}\sin\left(\frac{\pi}{3}n - \frac{\pi}{15}\right)$

(2) $H(0) = 0$, $H\left(\frac{2\pi}{5}\right) = 0$; Hence, $y(n) = 0$)

5.24 **(a)** $H(z) = \frac{2}{1 - \frac{1}{2}z^{-1}} - 1$; $h(n) = 2\left(\frac{1}{2}\right)^n u(n) - \delta(n)$

5.30 $|H(\omega)| = \cos^2\frac{\omega}{2}$; $\Theta(\omega) = \angle H(\omega) = -\omega$

5.33 **(a)** $H(\omega) = \frac{1}{2M+1}\left[1 + 2\sum_{k=1}^{M-1}\cos\omega k\right]$

(b) $H(\omega) = \frac{1}{2M}\cos M\omega + \frac{1}{2M}\left[1 + 2\sum_{k=1}^{M-1}\cos\omega k\right]$

5.39 **(a)** $|H_1(\omega)|^2 = \frac{1}{2} \Rightarrow \cos w_2 = \frac{4a-1-a^2}{2a}$

(b) $|H_2(\omega)|^2 = (\frac{1}{2} \Rightarrow \cos\omega \frac{2a}{1+a^2}$

By comparing the results of (a) and (b) we conclude that $\omega_2 < \omega_1$. Therefore, the second filter has a smaller $3dB$ bandwidth.

5.49 **(a)** Replace z by z^8. We need 8 zeros at the frequencies $\omega = 0, \pm\frac{\pi}{4}, \pm\frac{\pi}{3}, \pm\frac{3\pi}{4}, \pi$. Hence,

$H(z) = \frac{1-z^{-8}}{1-az^{-8}}$; $y(n) = ay(n-8) + x(n) - x(n-8)$

5.53

$H_r(\omega) = 2\left[h(0)\sin\frac{3\omega}{2} + h(1)\sin\frac{\omega}{2}\right]$; $H_r\left(\frac{\pi}{4}\right) = \frac{1}{2}$; $H_r\left(\frac{3\pi}{4}\right) = 1$

$H_r\left(\frac{3\pi}{4}\right) = 2h(0)\sin\frac{9\pi}{8} + 2h(1)\sin\frac{3\pi}{8} = 1$

$1.85h(0) + 0.765h(1) = \frac{1}{2}$

$-0.765h(0) + 1.85h(1) = 1$

$h(1) = 0.56$, $h(0) = 0.04$

5.59 **(a)** $H(z) = \frac{0.1}{1-0.9z^{-1}}$; $H_{bp}(\omega) = H\left(\omega - \frac{\pi}{2}\right) = \frac{0.1}{1-0.9e^{-j\pi-\frac{\pi}{2}}}$

(b) $h(n) = 0.1\left(0.9\,e^{j\pi/2}\right)^n u(n)$

5.61 **(a)** $H(z) = \frac{b}{1=az^{-1}}$; $H(\omega) = \frac{b}{1=ae^{-j\omega}}$; $b = \pm(1-a)$

(b) $|H(\omega)|^2 = \frac{b^2}{1+a^2-2a\cos\omega} = \frac{1}{2}$; $\omega_3 = \cos^{-1}\left(\frac{4a-1-a^2}{2a}\right)$

5.66 **(a)** $h(n) = \left[-\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n\right]u(-n-1)$

$y(n) = y(n-1) + y(n-2) + x(n-1)$

5.68 $r_{xx}(n) = \frac{16}{15}\left(\frac{1}{4}\right)^n u(n) + \frac{16}{15}(4)^n u(-n-1)$; $r_{hh}(n) = \frac{4}{3}\left(\frac{1}{2}\right)^n u(n) + \frac{4}{3}(2)^n u(-n-1)$

$r_{xy}(n) = \frac{16}{17}(2)^n u(n-1) - \frac{16}{15}(4)^n u(-n-1) - \frac{128}{105}\left(\frac{1}{4}\right)^n u(n)$

$r_{yy}(n) = \frac{64}{21}(2)^n u(-n-1) - \frac{128}{105}(4)^n u(-n-1) + \frac{64}{21}\left(\frac{1}{2}\right)^n u(n) - \frac{128}{105}\left(\frac{1}{4}\right)^n u(n)$

5.77 **(a)** $H(z)H(z^{-1}) = \frac{\frac{5}{4} - \frac{1}{2}(z+z^{-1})}{\frac{10}{9} - \frac{1}{3}(z+z^{-1})}$

$H(z) = \frac{1-\frac{1}{2}z^{-1}}{1-\frac{1}{3}z^{-1}}$

5.82 **(a)** $B = 10\,\text{kHz}$; $F_s = 20\,\text{kHz}$; $z_1 = \frac{10k}{20k} = 0.5$; $z_2 = \frac{7.777k}{20k} = 0.3889$; $z_3 = \frac{8.889k}{20k} = 0.4445$;

$z_4 = \frac{6.667k}{20k} = 0.3334$;

$H(z) = (z-0.5)(z-0.3889)(z-0.4445)(z-0.3334)$

Chapter 6

6.9 **(a)** $X_a(F) = \frac{1}{j2\pi(F+F_0)}$

(b) $X(f) = \frac{1}{1-e^{-j2\pi(F+F_0/F_s)}}$

6.10

Since $\frac{F_c+\frac{B}{2}}{B} = \frac{50+10}{20} = 3$ is an integer, then $F_s = 2B = 40$ Hz

6.14

$\sum_{n=-\infty}^{\infty} x^2(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi}|X(w)|^2 dw = \frac{E_d}{T}$

6.18

Let P_d denote the power spectral density of the quantization noise. Then

$P_n = \int_{-\frac{B}{F_s}}^{\frac{B}{F_s}} P_d df = \frac{2B}{F_s}P_d = \sigma_e^2$

$\text{SQNR} = 10\log 10\frac{\sigma_x^2}{\sigma_e^2} = 10\log_{10}\frac{\sigma_x^2 F_s}{2BP_d} = 10\log_{10}\frac{\sigma_x^2 F_s}{2BP_d} + 10\log 10F_s$

Thus, SQNR will increase by 3 dB if F_s is doubled.

6.20

$H_s(z) = z^{-1}$; $H_n(z) = (1-z^{-1})^2$

Chapter 7

7.1 Since $x(n)$ is real, the real part of the DFT is even, imaginary part odd. Thus, the remaining points are $\{0.125 + j0.0518, 0, 0.125 + j0.3018\}$

7.2 **(a)** $\tilde{x}_2(l) = \sin\left(\frac{3\pi}{8}\right)|l|,\ |l| \le 7$
Therefore, $x_1(n)\ 8\ x_2(n) = \{1.25, 2.55, 2.55, 1.25, 0.25, -1.06, -1.06, 0.25\}$

(c) $\tilde{R}_{xx}(k) = X_1(k)X_1^*(k) = \frac{N^2}{4}[\delta(k-1) + \delta(k+1)]$
$\Rightarrow \tilde{r}_{xx}(n) = \frac{N}{2}\cos\left(\frac{2\pi}{N}n\right)$

(d) $\tilde{R}_{yy}(k) = X_2(k)X_2^*(k) = \frac{N^2}{4}[\delta(k-1) + \delta(k+1)]$
$\Rightarrow \tilde{r}_{yy}(n) = \frac{N}{2}\cos\left(\frac{2\pi}{N}n\right)$

7.5 **(a)** $\sum_{n=0}^{N-1} x_1(n)x_2^*(n) = \frac{1}{4}\sum_{n=0}^{N-1}\left(e^{j\frac{2\pi}{N}n} + e^{-j\frac{2\pi}{N}n}\right)^2 = \frac{N}{2}$

7.9 $X_3(k) = \{17, 19, 22, 19\}$

7.12 **(a)** $s(k) = W_2^k X(k)$
$s(n) = \{3, 4, 0, 0, 1, 2\}$

7.14 **(a)** $y(n) = x_1(n)\ 5\ x_2(n) = \{4, 0, 1, 2, 3\}$

7.21 **(a)** $F_s \equiv F_N = 2B = 6000$ samples/sec

(b) $L = 120$ samples

(c) $LT = \frac{1}{6000} \times 120 = 0.02$ seconds

7.23 **(a)** $X(k) = \sum_{n=0}^{N-1}\delta(n)e^{-j\frac{2\pi}{N}kn} = 1,\ 0 \le k \le N-1$

(e) $X(k) = N\delta(k - k_0)$

(h) $X(k) = \frac{1}{1 - e^{-j\frac{2\pi}{N}k}}$

7.25 **(a)** $X(w) = 3 + 2\cos(2w) + 4\cos(4w)$

7.31 **(a)** $c_k = \left(\frac{2}{\pi}, -\frac{1}{\pi}, \frac{2}{3\pi}, -\frac{1}{2\pi}\cdots\right)$

7.32 **(a)** $Y(j\Omega) = T_0 \sin c\left(\frac{T_0(\Omega - \Omega_0)}{2}\right)e^{-j\frac{T_0(\Omega-\Omega_0)}{2}}$

(c) $Y(w) = \frac{\sin\frac{N}{2}(w-w_0)}{\sin\frac{w-w_0}{2}}e^{-j\frac{N-1}{2}(w-w_0)}$

Chapter 8

8.5 $X(k) = 2 + 2e^{-j\frac{2\pi}{5}} + e^{-j\frac{4\pi}{5}} + \cdots + e^{-j\frac{8\pi}{5}}$
$x'(n) = \{2, 2, 1, 1, 1\}$
$x'(n) = \sum_m x(n + 7m),\ n = 0, 1, \cdots 4$

8.8 $W_8 = \frac{1}{\sqrt{2}}(1 - j)$
Refer to Fig. 8.1.9. The first stage of butterflies produces $\{2, 2, 2, 2, 0, 0, 0, 0\}$. The phase factor multiplications do not change this sequence. The next stage produces $\{4, 4, 0, 0, 0, 0, 0.0\}$ which again remains unchanged by the phase factors. The last state produces $\{8, 0, 0, 0, 0, 0, 0, 0\}$. The bit reversal to permute the sequence into proper order unscrambles only zeros so the result remains $\{8, 0, 0, 0, 0, 0, 0, 0\}$.

8.13 **(a)** "gain" $= W_8^0 W_8^0(-1)W_8^2 = -W_8^2 = j$

(b) Given a certain output sample, there is one path from every input leading to it. This is true for every output.

(c) $X(3) = x(0) + W_8^3 x(1) - W_8^2 x(2) + W_8^2 W_8^3 x(3) - W_8^0 x(4) - W_8^0 W_8^3 x(5) +$
$W_8^0 W_8^2 x(6) + W_8^0 W_8^2 W_8^3 x(7)$

8.16 $x = x_R + jx_I = (a + jb)(c + jd)$

$e = (a - b)d$ 1 add 1 mult

$x_R = e + (c - d)a$ 2 adds 1 mult

$x_I = e + (c + d)b$ 2 adds 1 mult

Total 5 adds 3 mult

8.19 $X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}$

Let $F(t)$, $t = 0, 1, \cdots, N-1$ be the DFT of the sequence on k $X(k)$.

$$F(t) = \sum_{k=0}^{N-1} X(k) W_N^{tk} = \{x(N-1), x(N-2), \cdots, x(1), x(0)\}$$

8.21 **(a)** $W(z) = \frac{1}{2}\frac{1 - z^{-N}}{1 - z^{-1}} - \frac{1}{4}\frac{1 - \left(z^{-1}e^{j\frac{2\pi}{N-1}}\right)^N}{1 - z^{-1}e^{j\frac{2\pi}{N-1}}}$

$$- \frac{1}{4}\frac{1 - \left(z^{-1}e^{-j\frac{2\pi}{N-1}}\right)^N}{1 - z^{-1}e^{-j\frac{2\pi}{N-1}}}$$

(b) $x_w(n) = w(n)x(n)$; $X_w(k) = W(k)NX(k)$

8.32 $X(k) = \sum_{m=0}^{N-1} x(m) W_N^{-k(N-m)}$

This can be viewed as the convolution of the N-length sequence $x(n)$ with the impulse response of a linear filter.

$h_k(n) \triangleq W_N^{kn} u(n)$, evaluated at time N

$H_k(z) = \sum_{n=0}^{\infty} W_N^{kn} z^{-n} = \frac{1}{1 - W_N^k z^{-1}} = \frac{Y_u(z)}{X(z)}$

Then, $y_k(n) = W_N^k y_k(n-1) + x(n)$, $y_k(-1) = 0$ and $y_k(N) = X(k)$

8.34 In the DIF case, the number of butterflies affecting a given output is $\frac{N}{2}$ in the first stage, $\frac{N}{4}$ in the second, $\cdots$. The total number is $N - 1$.

Every butterfly requires 4 real multiplies, and the error variance is $\frac{\delta^2}{12}$. Under the assumption that the errors are uncorrelated, the variance of the total output quantization error is

$$\sigma_q^2 = 4(N-1)\frac{\delta^2}{12} = \frac{N\delta^2}{3}$$

Chapter 9

9.3 $H(z) = \frac{8 - z^{-1}}{1 - 0.5z^{-1}}$

$h(n) = 8(0.5)^n u(n) - (0.5)^{n-1} u(n-1)$

9.6

$c_0 = 1$; $c_1 = -(b_1 + b_2)$; $d_1 = b_1$; $c_2 = b_2$

9.8 **(a)** $h(n) = \frac{1}{2}\left[\left(\frac{1}{2}\right)^n + \left(-\frac{1}{2}\right)\right]^n u(n)$

9.11 $H(z) = C\left(1 - \frac{53}{150}z^{-1} + 0.52z^{-2} - 0.74z^{-3} + \frac{1}{3}z^{-4}\right)$, where C is a constant

9.15 $k_2 = \frac{1}{3}$; $k_1 = \frac{3}{2}$

9.17 $h(n) = \delta(n) + 0.157\delta(n-1) + 0.0032\delta(n-2) + 0.8\delta(n-3)$

9.21 $H(z) = 1 + 1.38z^{-1} + 1.311z^{-2} + 1.337z^{-3} + 0.9z^{-4}$

9.30 **(a)** $|H(e^{jw})|^2 = \frac{a^2 - 2a\cos w + 1}{1 - 2a\cos w + a^2} = 1 \forall w$

9.32 $y(n) = 0.999y(n-1) + e(n)$
where $e(n)$ *is white noise, uniformly* distributed in the interval $\left[-\frac{1}{2^9}, \frac{1}{2^9}\right]$. Then $E\{y^2(n)\} = 0.999^2 E\{y^2(n-1)\} + E\{e^2(n)\} = 6.361x10^{-4}$

9.35 **(a)** $y(n) = Q[0.1\delta(n)] + Q[0.5y(n-1)]$; $y(0) = Q[0.1] = \frac{1}{8}$; $y(1) = Q\left[\frac{1}{16}\right] = 0$ and $y(2) = y(3) = y(4) = 0$; no limit cycle

9.37 Define $\rho_c = r\cos\theta$, $\rho_s = r\sin\theta$ for convenience, (a)

$$-\rho_s y(n-1) + e_1(n) + x(n) + \rho_c v(n-1) + e_2(n) = v(n)$$

$$\rho_s v(n-1) + e_3(n) + \rho_c y(n-1) + e_4(n) = y(n)$$

9.38 **(a)** $h(n) = \left[2\left(\frac{1}{2}\right)^n - \left(\frac{1}{4}\right)^n\right]u(n)$
$\sigma_q^2 = \frac{64}{35}\sigma_{e1}^2 + \frac{16}{15}\sigma_{e2}^2$

9.40 **(a)** $G_1 \frac{\left(1-0.8e^{j\frac{\pi}{4}}\right)\left(1-0.8e^{-j\frac{\pi}{4}}\right)}{(1-0.5)\left(1+\frac{1}{3}\right)} = 1$; $G_1 = 1.1381$

$G_2 \frac{(1+0.25)\left(1-\frac{5}{8}\right)}{\left(1-0.8e^{j\frac{\pi}{3}}\right)\left(1-0.8e^{-j\frac{\pi}{3}}\right)} = 1$; $G_2 = 1.7920$

9.41 **(a)** $k_1 = -\frac{4}{9}$; $k_2 = \frac{5}{32}$

(b) $k_2 = -\frac{1}{6}$; $k_1 = -\frac{1}{5}$

Chapter 10

10.1 $h_d(n) = \frac{\sin\frac{\pi}{6}(n-12)}{\pi(n-12)}$; $h(n) = h_d(n)w(n)$
where $w(n)$ is a rectangular window of length $N = 25$.

10.2 $h_d(n) = \delta(n) - \frac{\sin\frac{\pi}{3}(n-2)}{\pi(n-12)} + \frac{\sin\frac{\pi}{6}(n-12)}{\pi(n-12)}$; $h(n) = h_d(n)w(n)$
where $w(n)$ is a rectangular window of length 25.

10.5 $H_r(\omega) = 2\sum_{n=0}^{1} h(n)\cos\left[\omega\left(\frac{3}{2} - n\right)\right]$
From $H_r(0) = 1$ and $H_r\left(\frac{\pi}{2}\right) = 1/2$, we obtain $h(0) = 0.073$, $h(1) = 0.427$, $h(2) = h(1)$ and $h(3) = h(0)$

10.7 $h(n) = \{0.3133, -0.0181, -0.0914, 0.0122, 0.0400, -0.0019, -0.0141, 0.52, 0.52, -0.0141, -0.0019, 0.0400, 0.0122, -0.0914, -0.0181, 0.3133\}$

10.9 $h_d(n) = \frac{\cos\pi(n-10)}{(n-10)}, \qquad 0 \le n \le 20,\ n \ne 10$

$= 0, \qquad n = 10$

Then $h(n) = h_d(n)w(n)$, where $w(n)$ is the Hamming window of length 21.

10.12 **(a)** Let $T = 2$. Then $H(z) = \frac{1+z^{-1}}{1-z^{-1}} \Rightarrow y(n) = y(n-1) + x(n) + x(n-1)$

10.13 $H(z) = A\frac{\left(1+z^{-1}\right)\left(1+2z^{-1}+z^{-2}\right)}{\left(1-\frac{1}{2}z^{-1}\right)\left(1-\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}\right)}$
$H(z)|_{z=1} = 1$; $A = \frac{3}{64}$, $b_1 = 2$, $b_2 = 1$, $a_1 = 1$, $c_1 = -\frac{1}{2}$, $d_1 = -\frac{1}{2}$, $d_2 = \frac{1}{4}$

10.15 From the design specifications we obtain $\varepsilon = 0.509$, $\delta = 99.995$, $f_p = \frac{1}{6}$, $f_s = \frac{1}{4}$

$$\text{Butterworth filter:} \quad N_{min} \geq \frac{\log \eta}{\log k} = 9.613 \Rightarrow N = 10$$

$$\text{Chebyshev filter:} \quad N_{min} \geq \frac{\cosh^{-1}\eta}{\cosh^{-1}k} = 5.212 \Rightarrow N = 6$$

$$\text{Elliptic filter:} \quad N_{min} \geq \frac{k(\sin\alpha)}{k(\cos\alpha)}, \frac{k(\cos\beta)}{k(\sin\beta)} = 3.78 \Rightarrow N = 4$$

10.19 **(a)** MATLAB is used to design the FIR filter using the Remez algorithm. We find that a filter of length $M = 37$ meets the specifications. We note that in MATLAB, the frequency scale is normalized to $\frac{1}{2}$ of the sampling frequency.

(b) $\delta_1 = 0.02$, $\delta_2 = 0.01$, $\Delta f = \frac{20}{100} - \frac{15}{100} = 0.05$

With equation (10.2.94) we obtain $\hat{M} = \frac{-20\log_{10}\left(\sqrt{\delta_1\delta_2}\right)-13}{14.6\Delta f} + 1 \approx 34$

With equation (10.2.95) we obtain $D_\infty(\delta_1\delta_2) = 1.7371$; $f(\delta_1\delta_2) = 11.166$

and $\hat{M} = \frac{D_\infty(\delta_1\delta_2) - f(\delta_1\delta_2)(\Delta f)^2}{\Delta f} + 1 \approx 36$

Note (10.2.95) is a better approximation of M.

10.21 **(a)** dc gain: $H_a(0) = 1$; 3 dB frequency: $\Omega_c = \alpha$

For all Ω, only $H(j\infty) = 0$; $h_a(\tau) = \frac{1}{e}h_a(0) = \frac{1}{e}$; $r = \frac{1}{\alpha}$

10.24 $H(z) = \frac{1}{6}(1 - z^{-6})(1 - z^{-1})\left(2 + z^{-1} + \frac{3}{2}z^{-\alpha} + \frac{1}{2}z^{-3} + z^{-4}\right)$

This filter is FIR with zeros at $z = 1$, $e^{\pm j\frac{\pi}{6}}$, $e^{\pm j\frac{\pi}{2}}$, $e^{\pm j5\frac{\pi}{6}}$, $-0.55528 \pm j0.6823$ and $0.3028 \pm j0.7462$

10.25 **(a)** $f_L = \frac{900}{2500} = 0.36$; $f_H = \frac{1100}{2500} = 0.44$

$h_d(n) = \frac{2\sin 0.08\pi(n-15)}{(n-15)}$; $h(n) = h_d(n)w_H(n)$

$w_H(n) = 0.54 - 0.46\cos\frac{2\pi(n-15)}{30}$

Index

E

F

G

H

I

Q

R

S

T

U

V

W

Y

Z