Robotics 110405442

Ahmad AL-Jarrah, Ph.D, P.Eng
Faculty of Engineering
Department of Mechatronics Engineering
Office: E3141
Email: Jarrah@hu.edu.jo

Textbooks:

- John Craig, “Introduction to Robotics mechanics and control”, 4th Ed., Pearson Education Inc.
- Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, “Robot Modeling and Control”, 1st Ed., John Wiley & Sons, Inc.

Chapter 1

Introduction

- The word “Robot” was first used by the Czechoslovakian play writer Karel Capek who wrote a play entitled “Rossum’s Universal Robots” back in 1921.
- “Robot” word comes from a Czech word “Robotnik” which means workers who performed manual labor for human beings.
- Mostly, the word “Robot” today means any man-made machine that can perform work or other actions normally performed by humans.

What is a Robot?

- Many different definitions for robots exist !
- “A robot is a **re-programmable**, **multifunctional** machine designed to manipulate materials, parts, tools, or specialized devices through variable programmed motions for the performance of a **variety of tasks**”.

(*Robot Industries Association*)

Multitask robot - YouTube

Automation vs. robots

Automation: Machinery designed to carry out a specific task.

- Bottling machine
- Transfer line
- Dishwasher

Robots: machinery designed to carry out variety of tasks.

- Pick and place arms (Manipulator)
- Assembling (Manipulator)

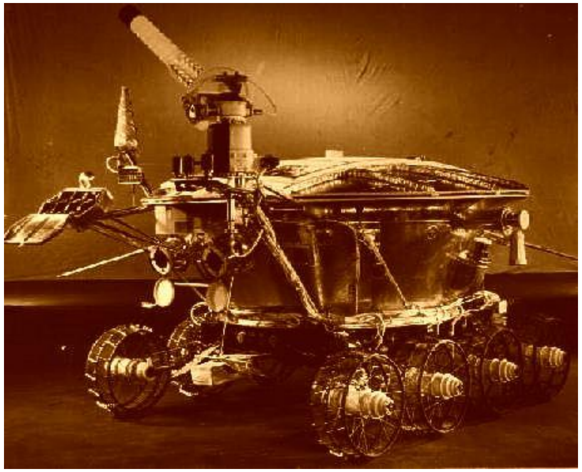
Robots Classifications

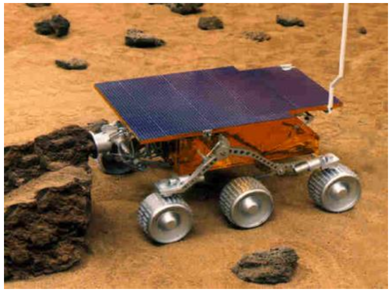
- *Manipulators*: robotic arms. These are most commonly found in industrial settings.

- *Mobile Robots*: unmanned vehicles

- *Hybrid Robots*: mobile robots with manipulators

- Humanoid robot





Robotic Manipulator Classifications

Robot manipulators can be classified by several criteria such as

1. Their **power source**, or way in which the joints are actuated,
    - ➢ Electrical Manipulators
    - ➢ Hydraulic Manipulators
    - ➢ Pneumatic Manipulators
2. Their **geometry**, or kinematic structure
    - ➢ Cartesian manipulator
    - ➢ Cylindrical manipulator…..etc
3. Their **method of control**.
    - ➢ Servo manipulators
    - ➢ Non-servo manipulators

Applications

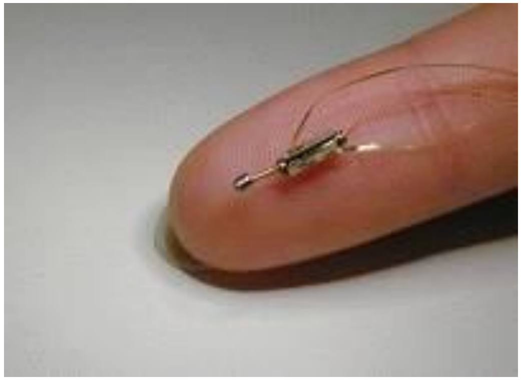
- Dangerous:
  - -Space exploration
  - -chemical spill cleanup
  - -disarming bombs
  - -disaster cleanup

- Repetitive:
  - -Welding car frames
  - -Part pick and place
  - -Assembling operations.

- High precision/High speed:
  - -Electronics chips
  - -Surgery
  - -Precision machining

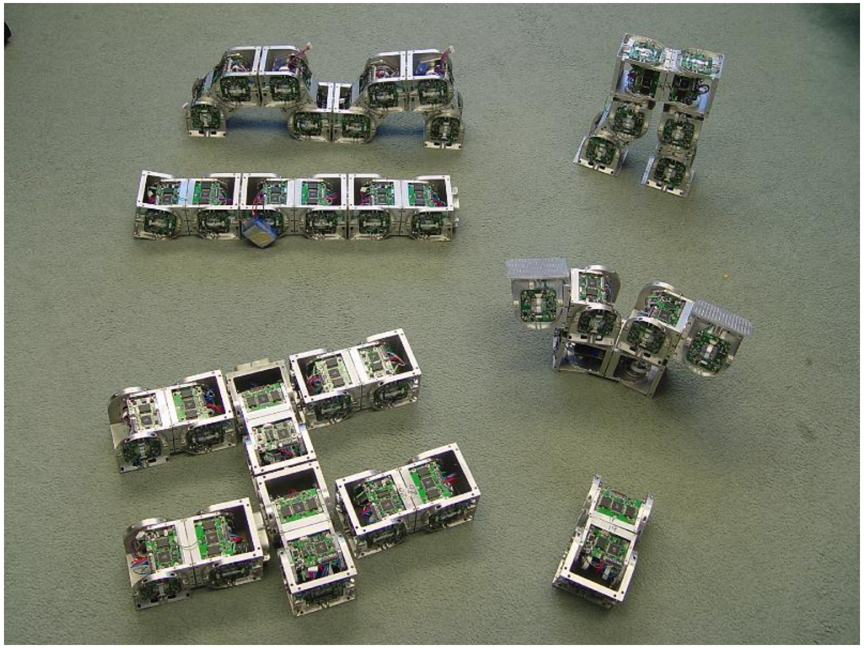
New direction

Nanobots



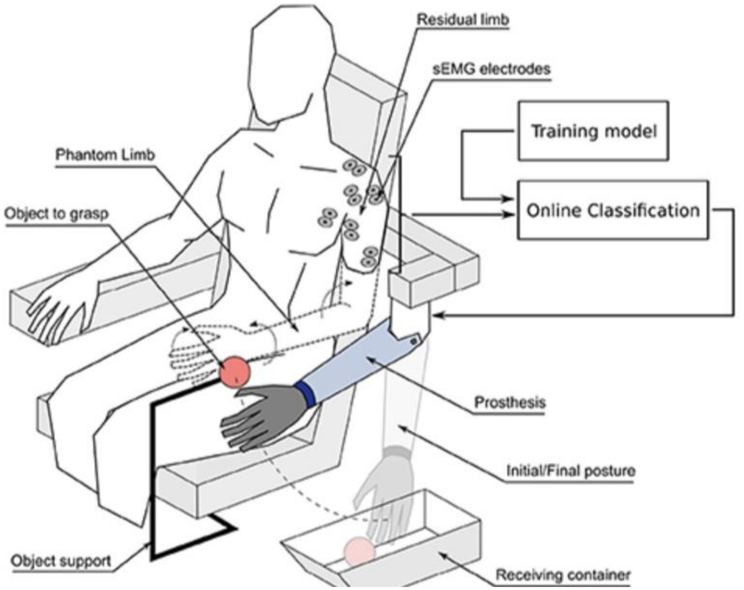
Nanobots could be small solution to big problems (cnn.com)

Reconfigurable Robot



Superbot (wevolver.com)

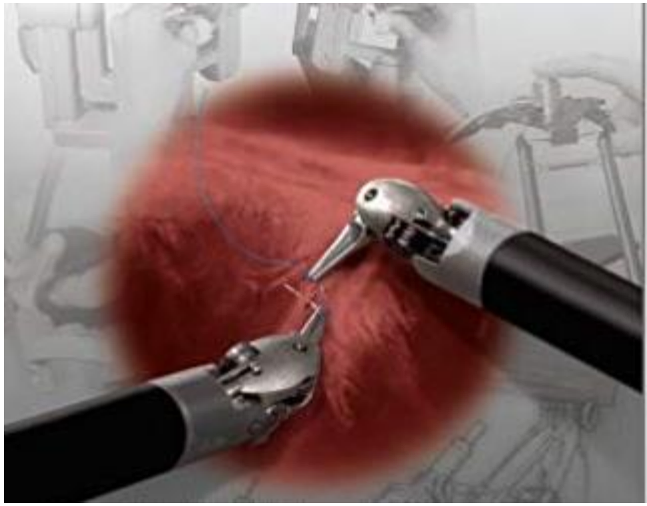
Robotic Limb



[This new robotic limb can be controlled with your mind | World Economic Forum (weforum.org)]

Powered Exoskeleton (Robot Suit)

[Cyberdyne's robot suit HAL to keep people walking - YouTube]



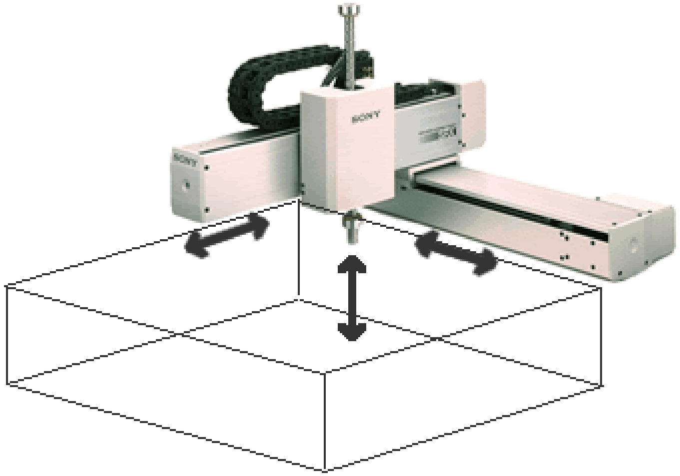
Haptic Interaction

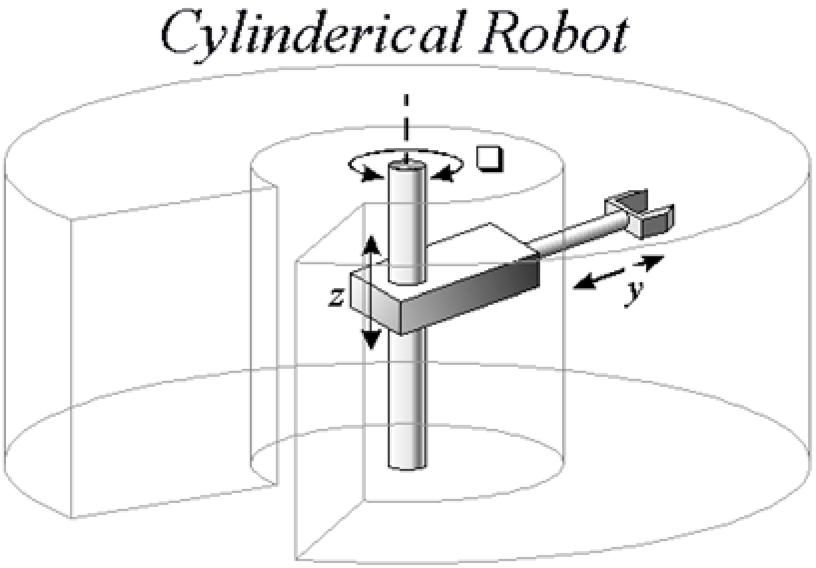
- In surgery
- In remote environments

Measures of performance

- Workspace
  - The space within which the robot operates.
  - Larger volume costs more but can increase the capabilities of a robot

Cartesian Robot





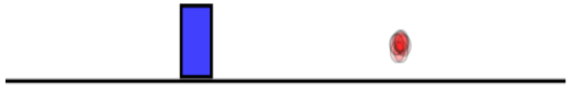
Measures of performance

- Speed and acceleration
  - Faster speed often reduces resolution or increases cost
  - Varies depending on position, load.
  - Speed can be limited by the task the robot performs (welding, cutting)

Measures of performance

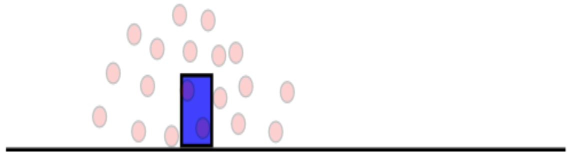
- Accuracy

The difference between the actual position of the robot and the programmed position

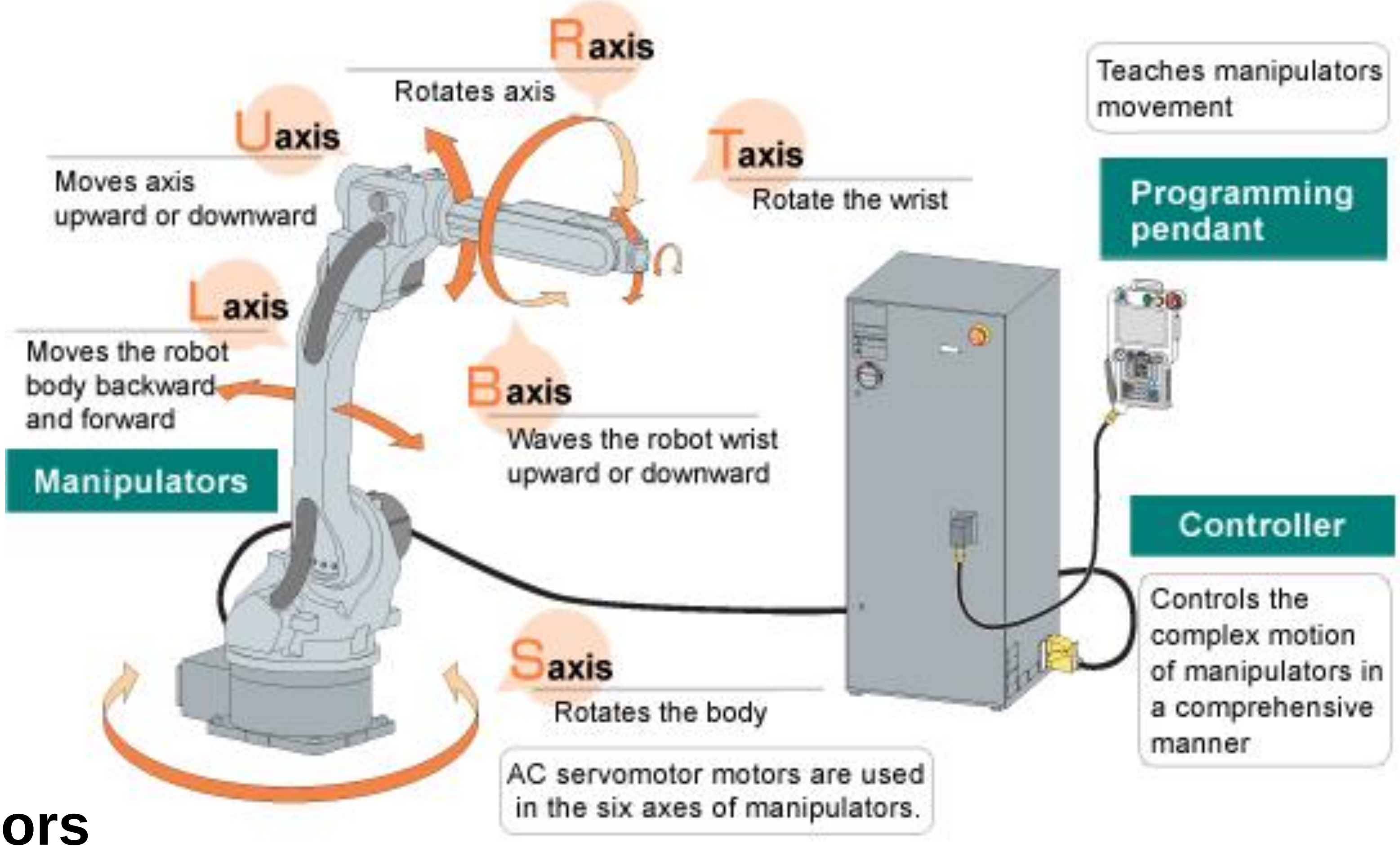


- Repeatability

Will the robot always return to the same point under the same conditions?



Robot Components



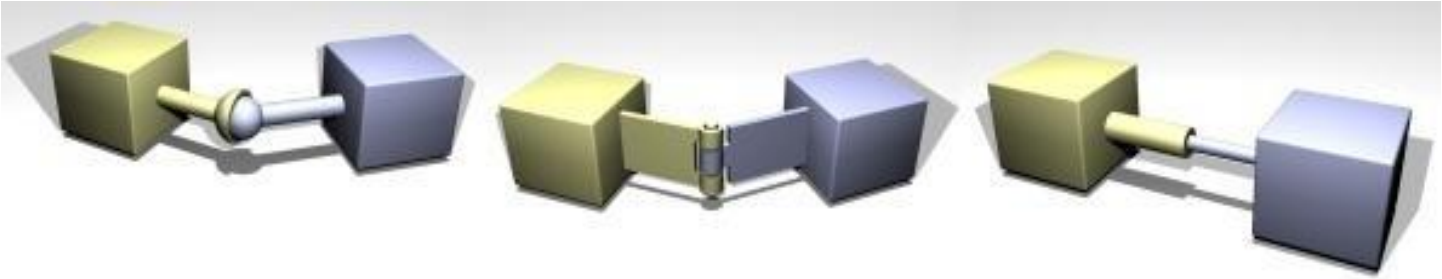
- **Body**
- **End Effectors**
- **Actuators**
- **Sensors**
- **Controller**
- **Software**

Robot: Body

The body consists of links and joints

- A link is a part, a shape with physical properties.
- A joint is a constraint on the spatial relations of two or more links.

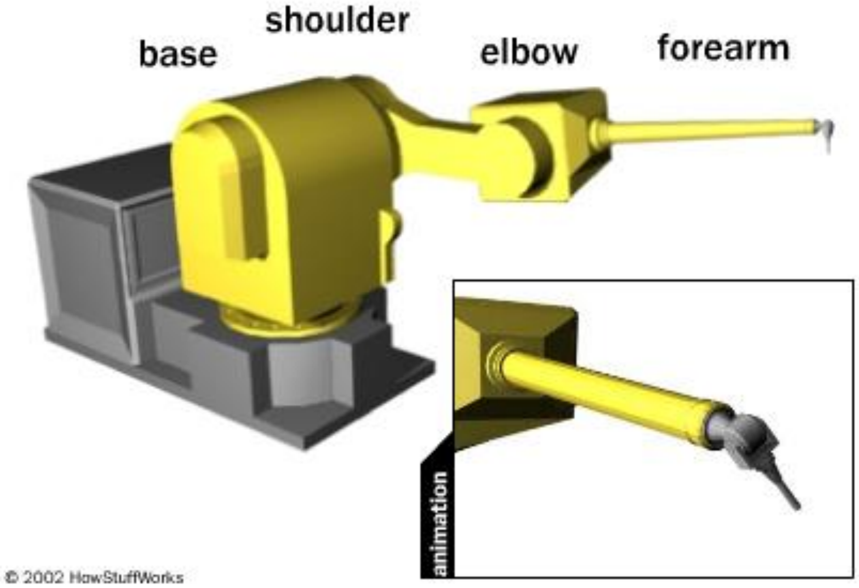
These are just a few examples…



ball joint ***Revolute (hinge) joint*** ***Prismatic (slider) joint***

Continue…Degrees of Freedom

- Joints constraint free movement, measured in “Degrees of Freedom” (DOFs).
- Number of DOF is the number of independent position variables
- Joints reduce the number of DOFs by constraining some translations or rotations.
- Robots classified by total number of DOFs



How many DOFs can you identify?

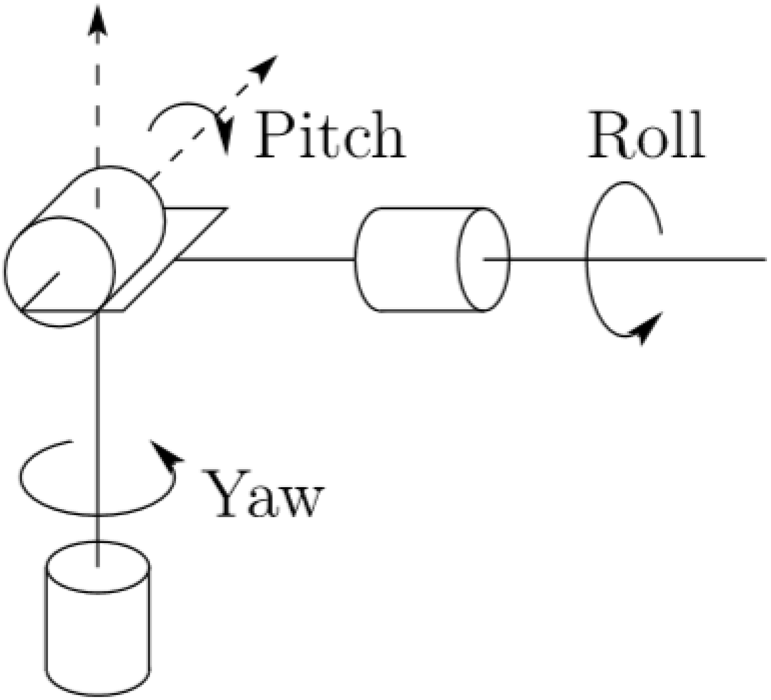
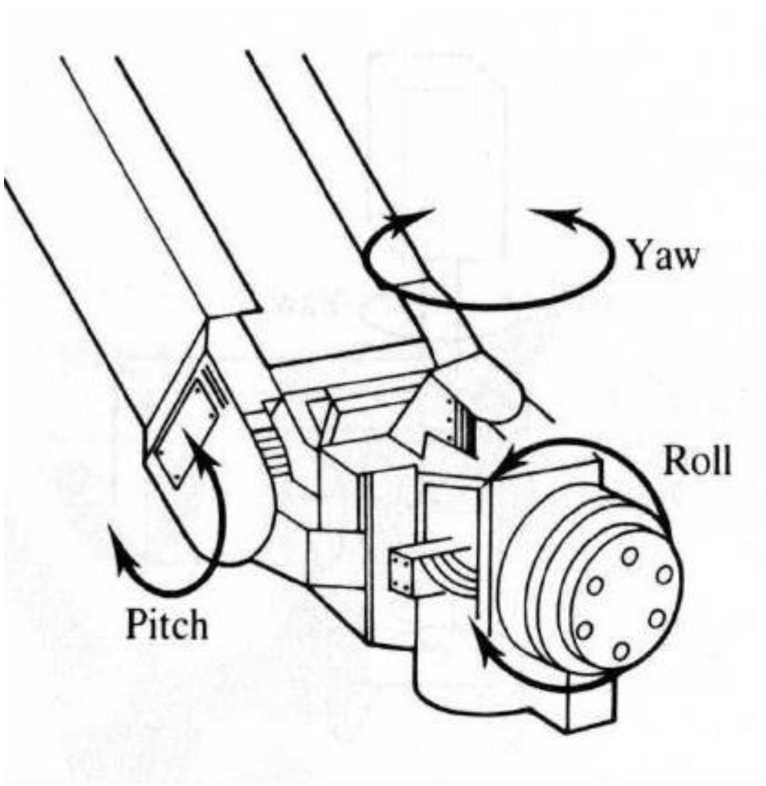
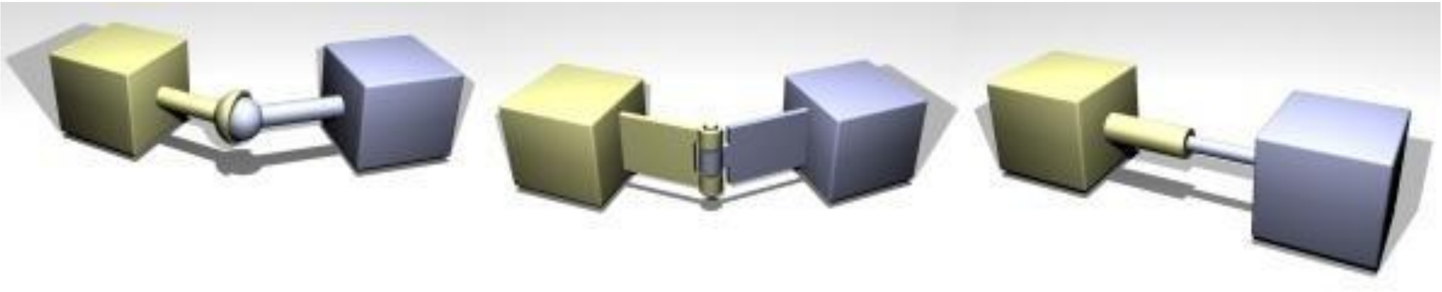


Fig. 1.5 Structure of a spherical wrist.

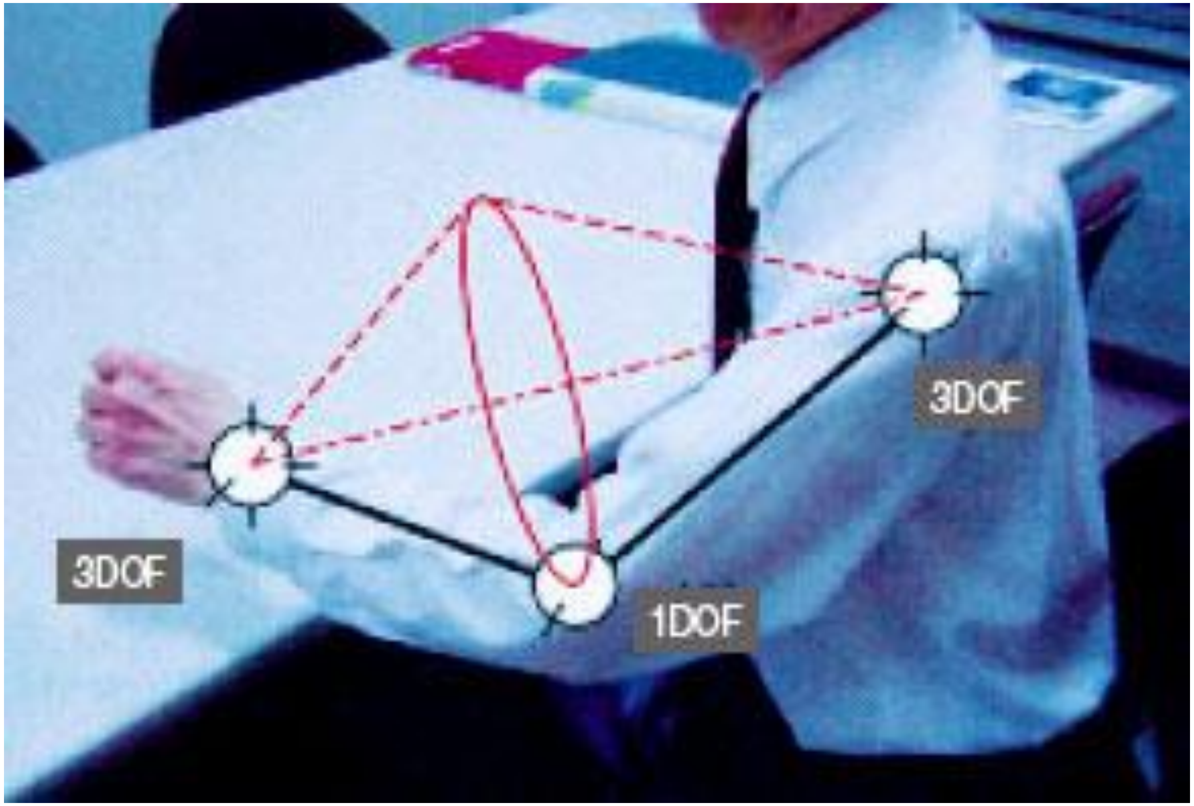


Robot wrist



How many DOFs can you identify ?

Degrees of Freedom



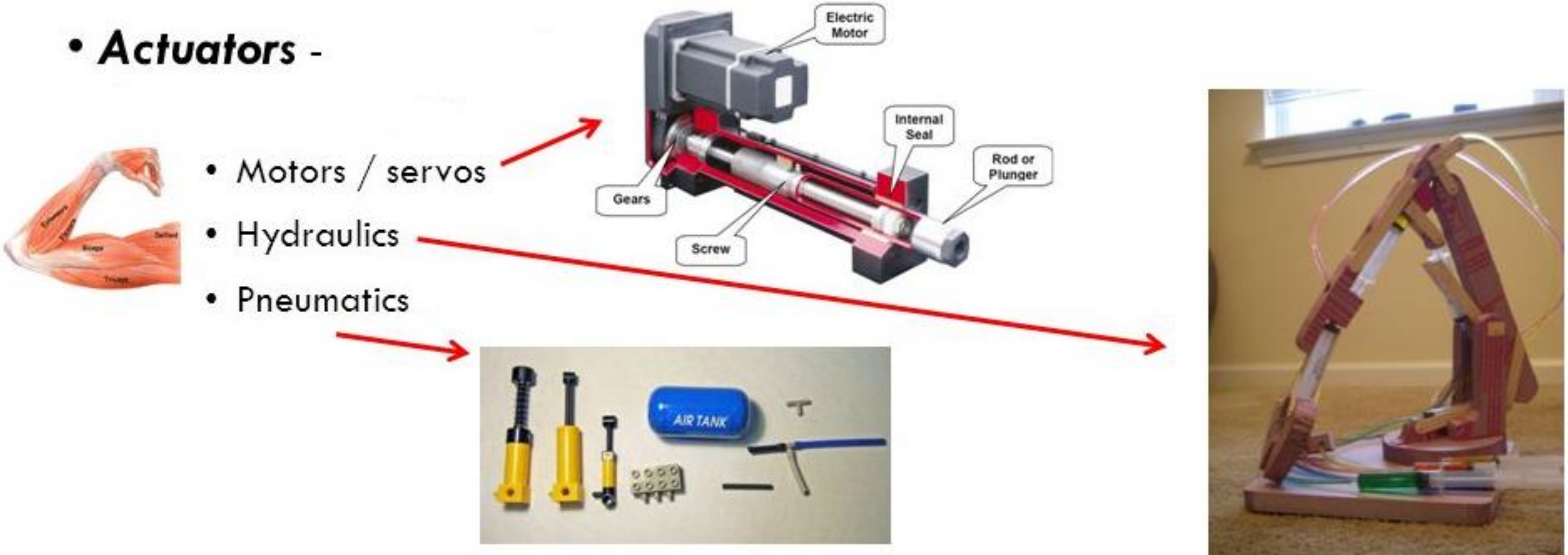
How many DOFs can you identify in your arm?

Robot: End Effectors

- Component to accomplish some desired physical function
- Examples:
    - Hands
    - Tools
    - Torch

Robot: Actuators

- Actuators are the “muscles” of the robot.
- These can be electric motors, hydraulic systems, pneumatic systems, or any other system that can produce forces or torques to the system.



- **Actuators** -
  - Motors / servos
  - Hydraulics
  - Pneumatics

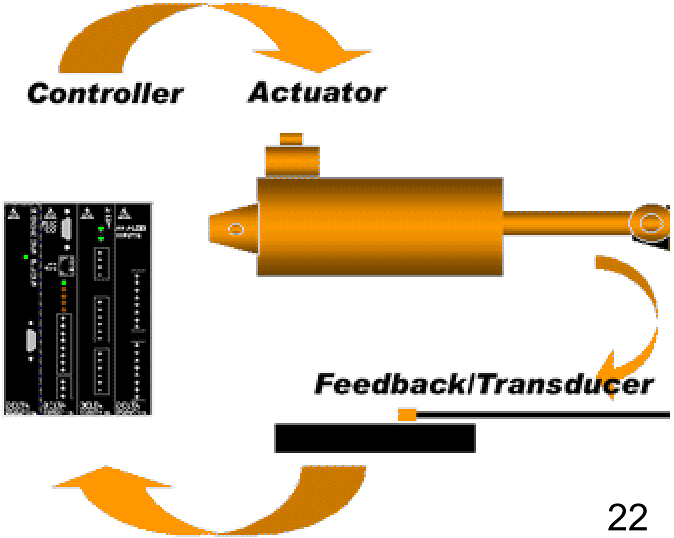
Robot: Sensors

- Rotation encoders
- Cameras
- Pressure sensors
- Limit switches
- Optical sensors
- Sonar

Robot: Controller

- Controllers direct a robot how to move.
  - ⇨ Position Control
  - ⇨ Force Control

  } Hybrid Control
- There are two controller paradigms
  - Open-loop controllers execute robot movement without feedback.
  - Closed-loop controllers execute robot movement and judge progress with sensors. They can thus compensate for errors.



Position Control

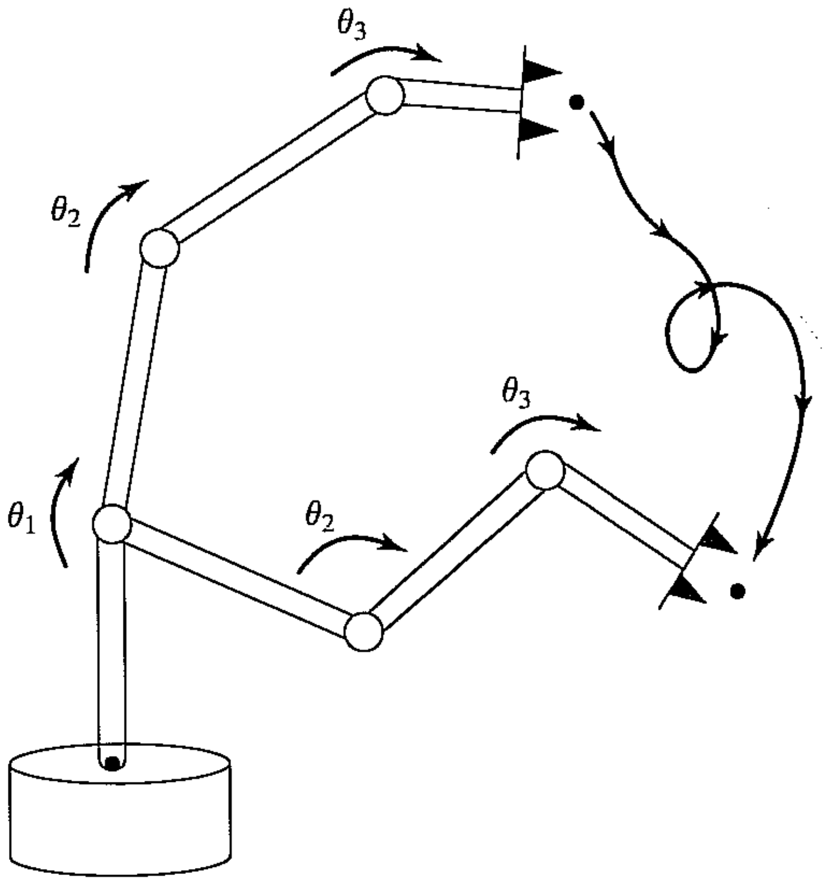


FIGURE 1.13: In order to cause the manipulator to follow the desired trajectory, a position-control system must be implemented. Such a system uses feedback from joint sensors to keep the manipulator on course.

Force Control

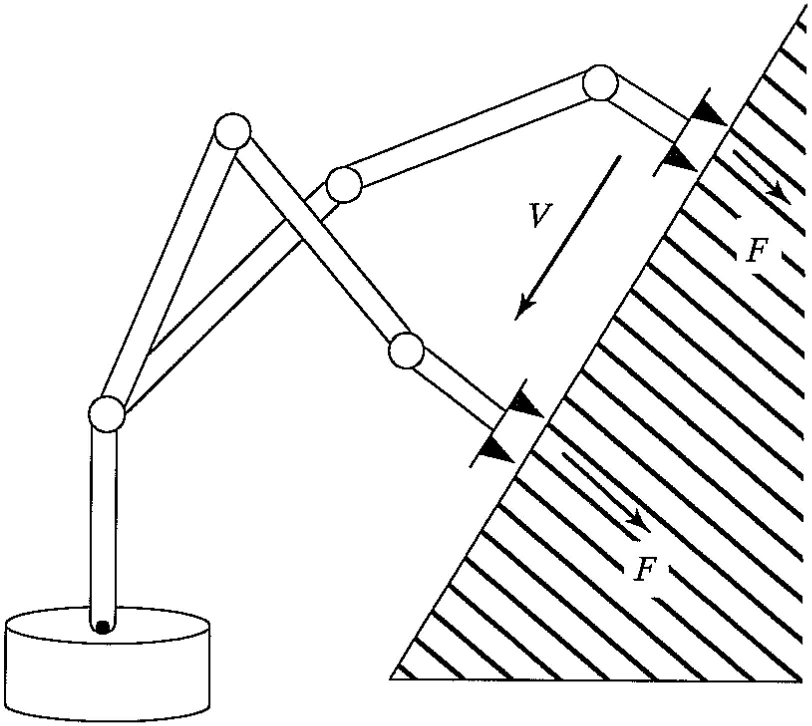


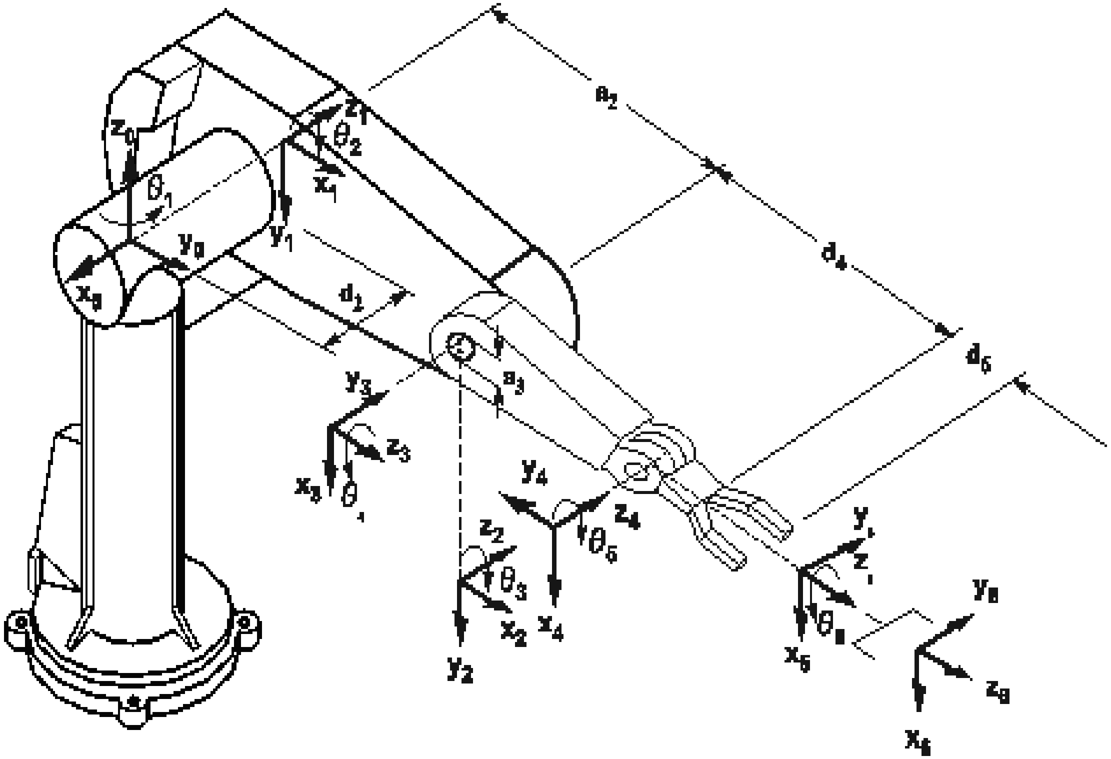
FIGURE 1.14: In order for a manipulator to slide across a surface while applying a constant force, a hybrid position–force control system must be used.

Kinematics

- Kinematics is the study of motion without regard for the forces that cause it.
- It refers to all time-based and geometrical properties of motion.
- It ignores concepts such as torque, force, mass, energy, and inertia.

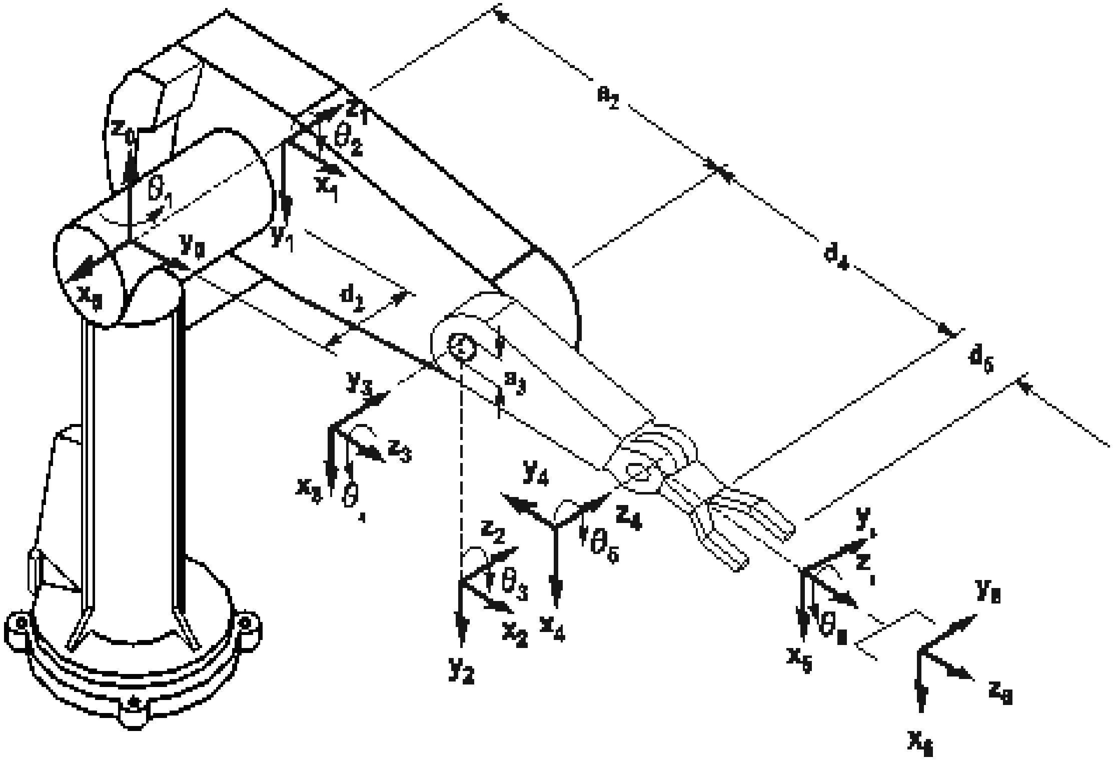
Forward Kinematics

- For a robotic arm, this would mean calculating the position and orientation of the end effector given all the joint variables.

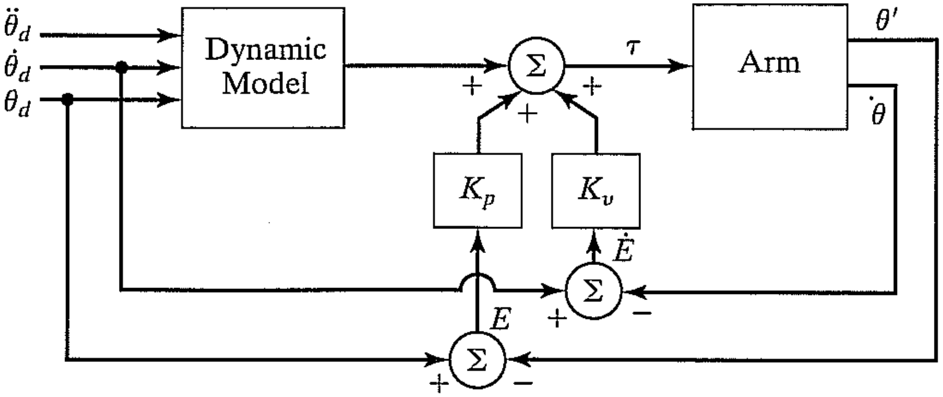


Inverse Kinematics

- Inverse Kinematics is the reverse of Forward Kinematics.
- It is the calculation of joint values given the positions, orientations, and geometries of mechanism’s parts.
- It is useful for planning how to move a robot in a certain way.



Dynamics



Dynamics is the study of forces/torques required to cause a motion

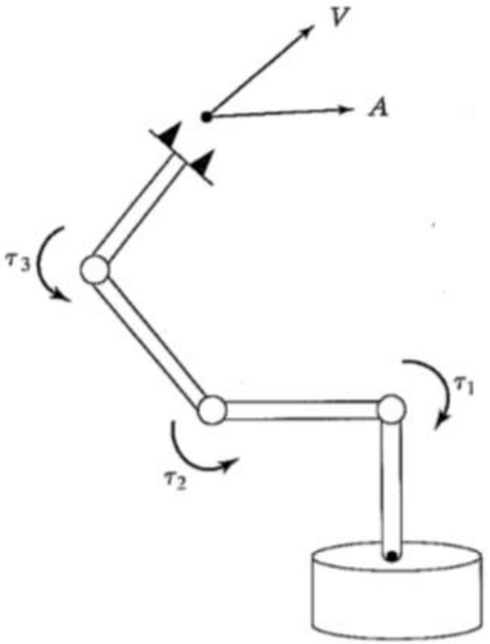
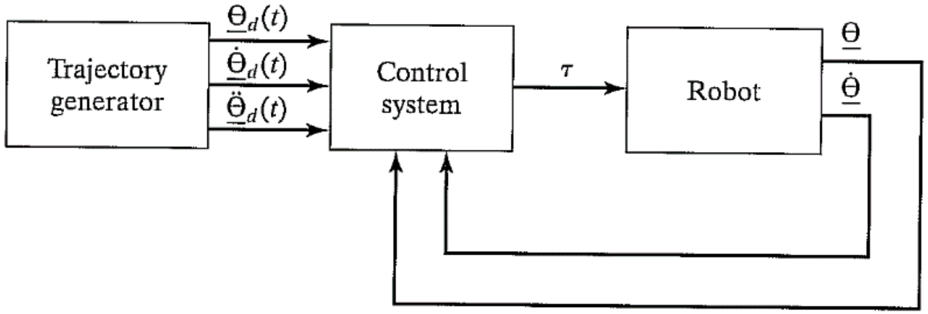


FIGURE 1.10: The relationship between the torques applied by the actuators and the resulting motion of the manipulator is embodied in the dynamic equations of motion.

Trajectory generating



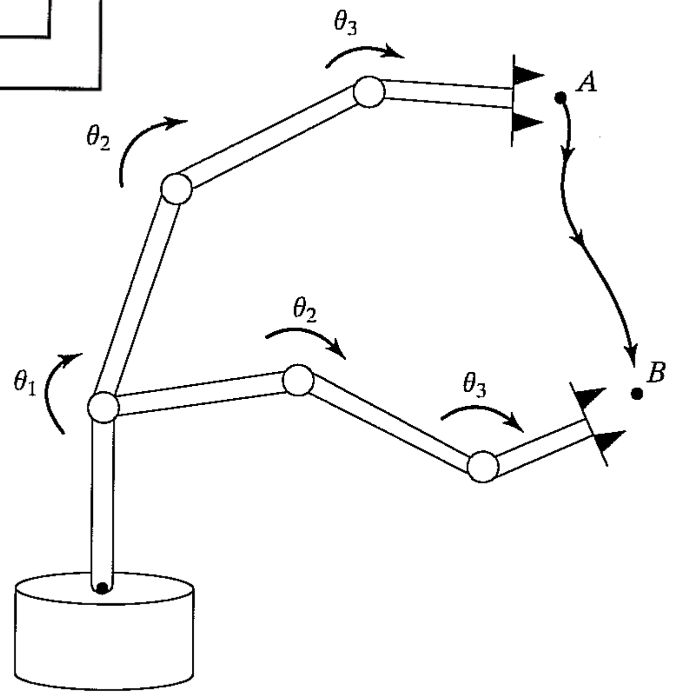
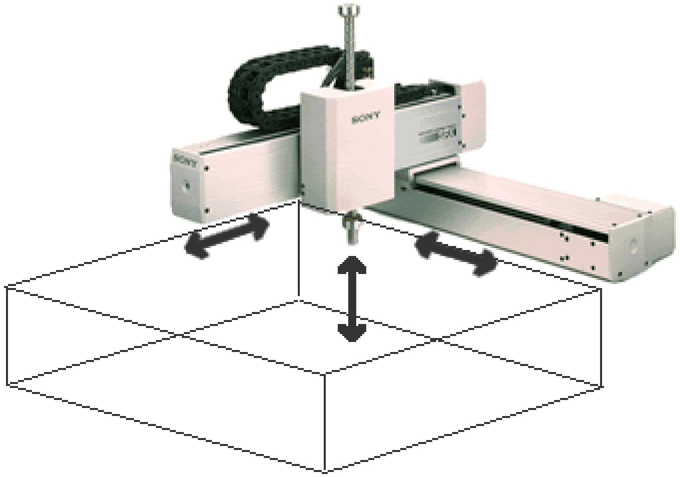
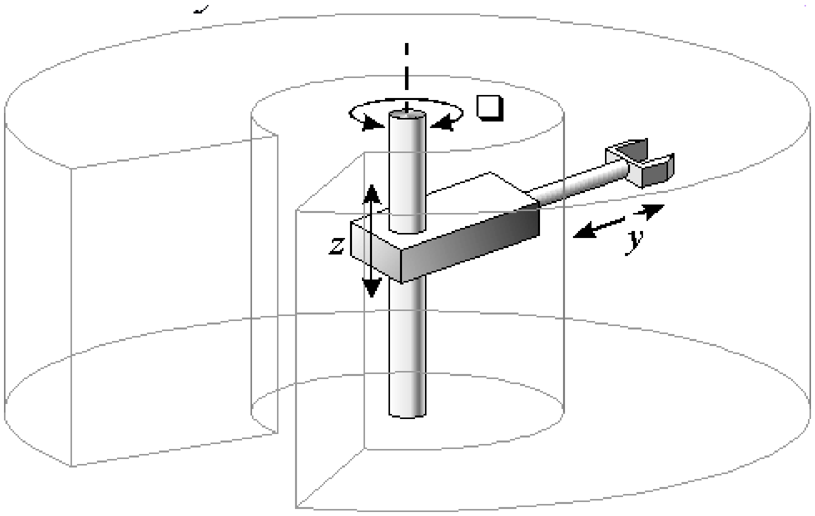


FIGURE 1.11: In order to move the end-effector through space from point A to point B, we must compute a trajectory for each joint to follow.

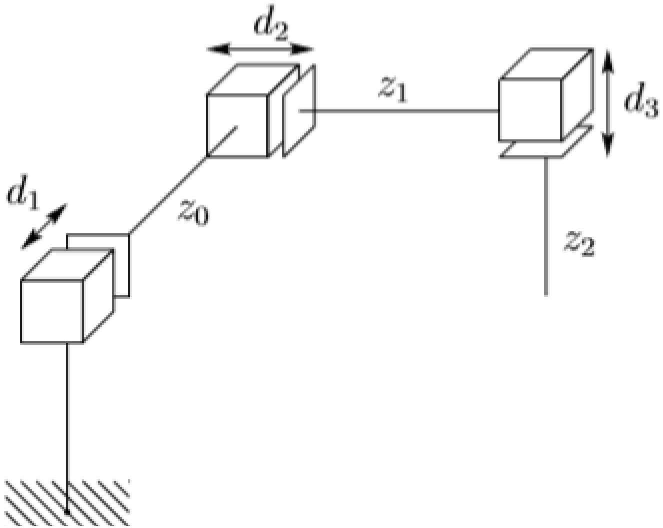
Common Kinematic arrangements of Manipulators

1. Cartesian manipulator (PPP)
2. Cylindrical manipulator (RPP)
3. Spherical manipulator (RRP)
4. Articulated (Revolute) manipulator (RRR)
5. SCARA (**S**elective **C**ompliant **A**rticulated **R**obot for **A**ssembly) manipulator (RRP)

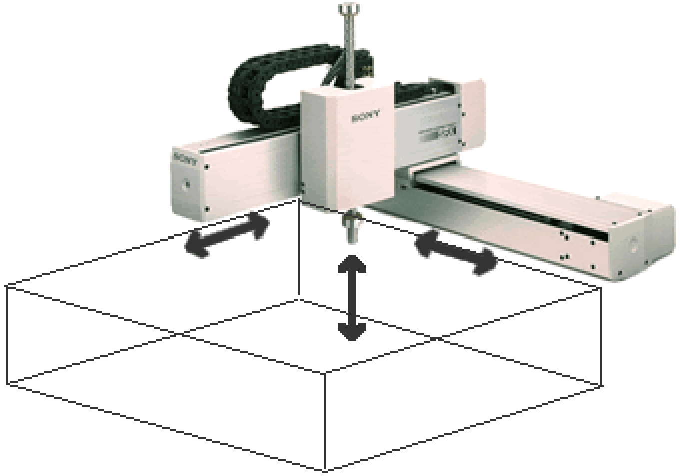




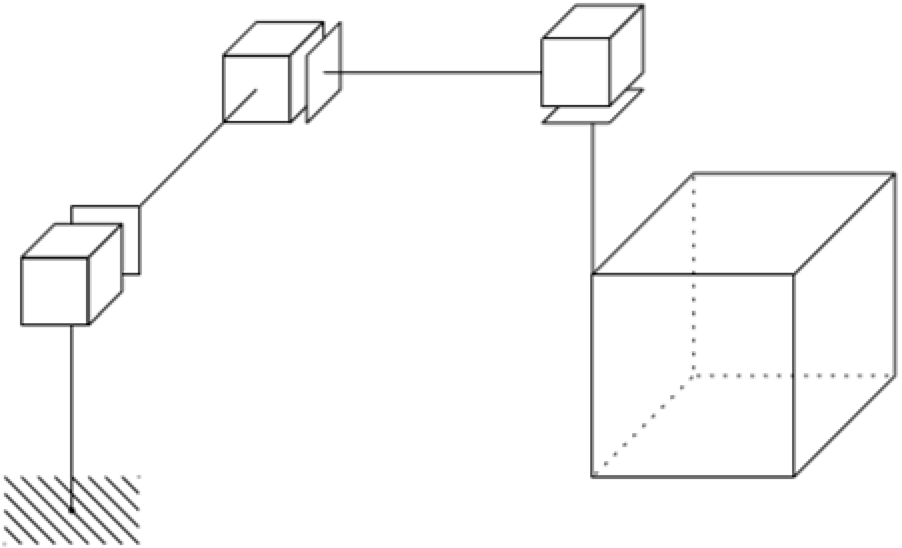
Cartesian Manipulators (PPP)



- A manipulator whose first three joints are prismatic
- Simple kinematics to be used in assembly applications and material transfer
- Rigid structure, pneumatic actuators can be used for pick and place operations.
- can only reach in front of itself and rails are hard seal (exposed to dirt)

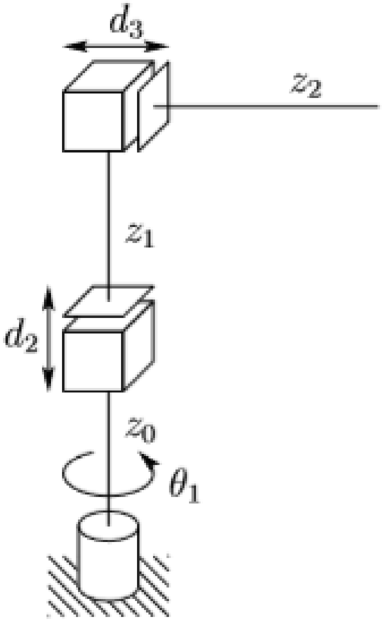


Epson Cartesian Robot



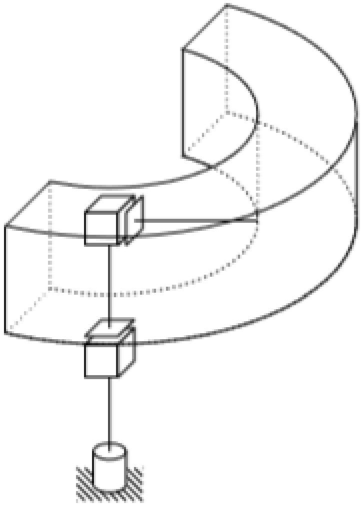
Workspace of the Cartesian manipulator.

Cylindrical Manipulators (RPP)



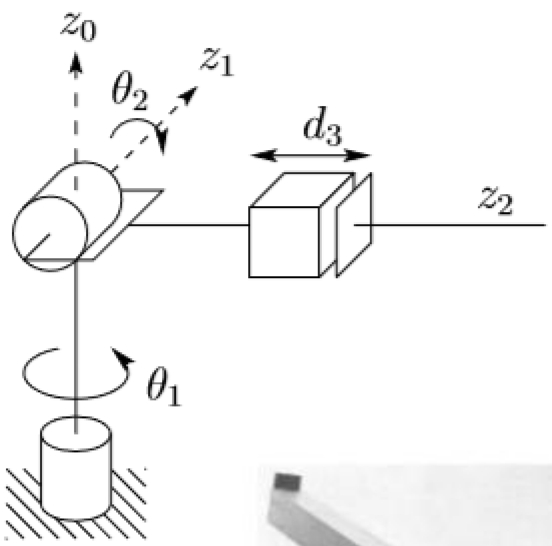
- Revolute joint then two prismatic joints
- The joint variables are the cylindrical coordinates of the end-effector with respect to the base.
- Can reach all around itself and powerful if hydraulic actuators are used.
- Will not reach around obstacles and above itself

Seiko RT3300 Robot

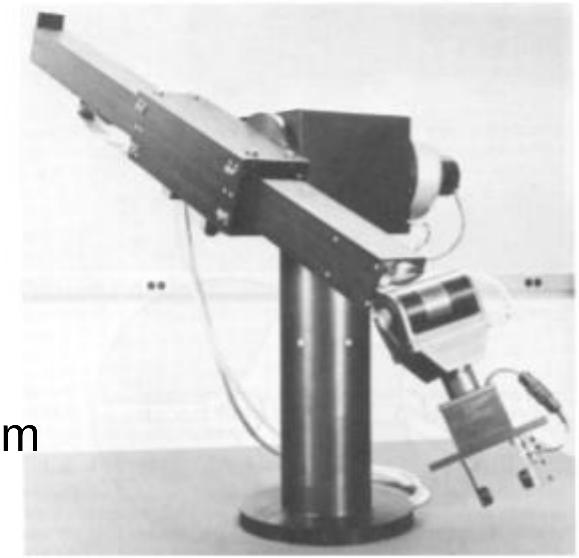


Workspace of the cylindrical manipulator

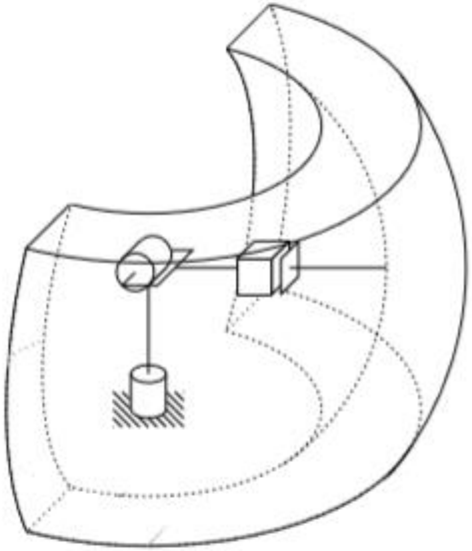
Spherical Manipulators (RRP)



- Two revolute joints then one prismatic joint
- The joint variables are the spherical coordinates defining the position of the end-effector with respect to base.
- Can bend down to pick objects up off the floor.

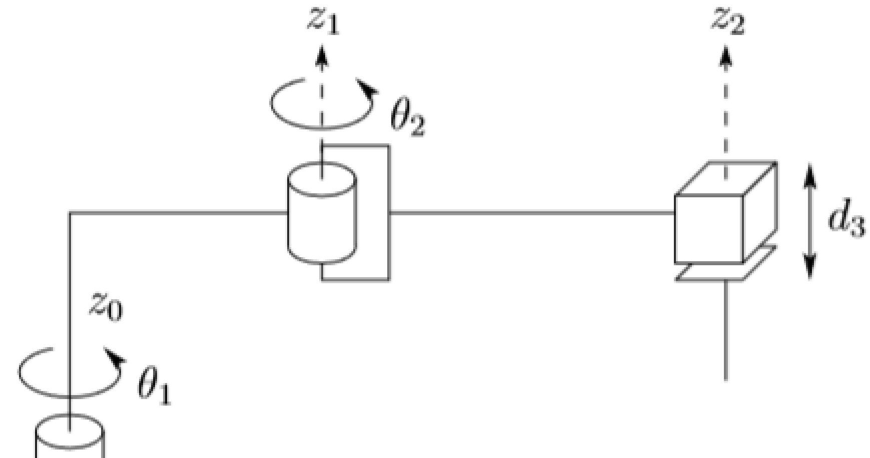


The Stanford Arm Robot



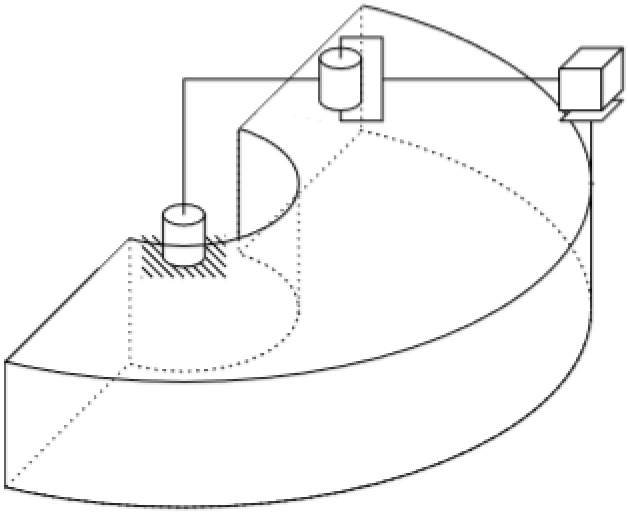
Workspace of the spherical manipulator.

SCARA Manipulators (RRP)



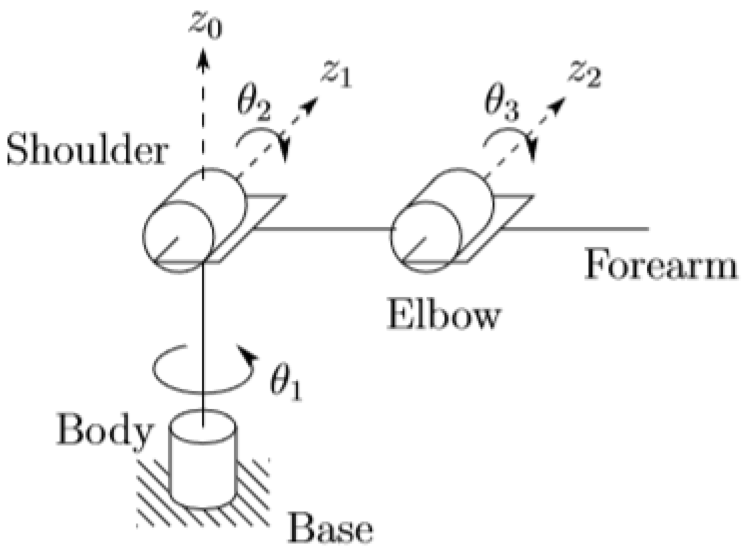
- Two revolute joints then one prismatic joint
- Unlike the spherical design, which has z0 perpendicular to z1, and z1 perpendicular to z2, the SCARA has z0,z1, and z2 mutually parallel.
- Can reach around obstacles with large horizontal reach.

The Epson E2L653S SCARA Robot



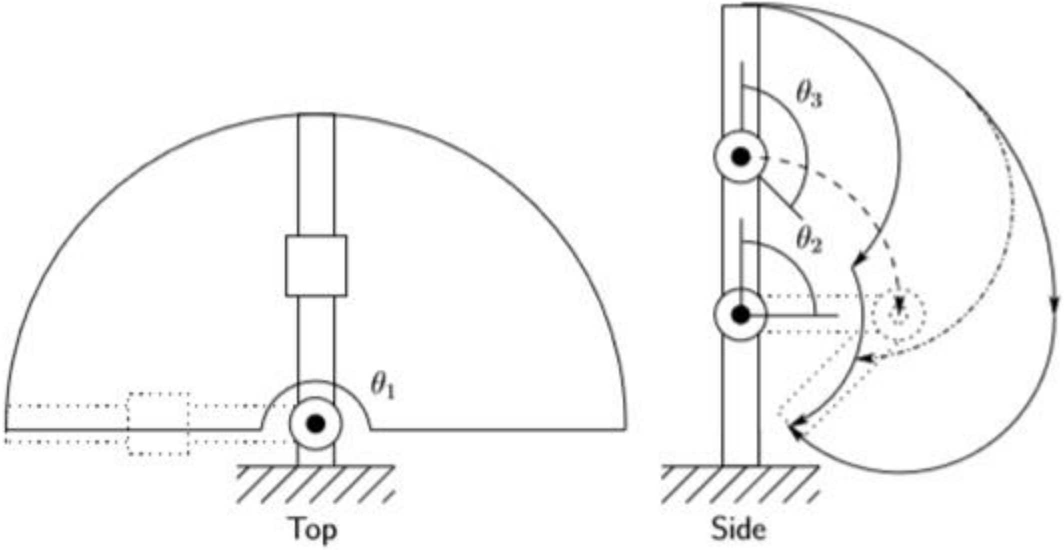
Workspace of the SCARA manipulator.

Articulated Manipulators (RRR)



- Revolute joints
- Very complex kinematics and dynamics
- Joint axis z2 is parallel to z1 and both z1 and z2 are perpendicular to z0. This kind of manipulator is known as an elbow manipulator.
- Provides large freedom of movement in a compact space and can reach above or below obstcles
- More than one way to reach a point in space.

The ABB IRB1400 Robot



Workspace of the elbow manipulator.

Robotics 110405442

Chapter 2

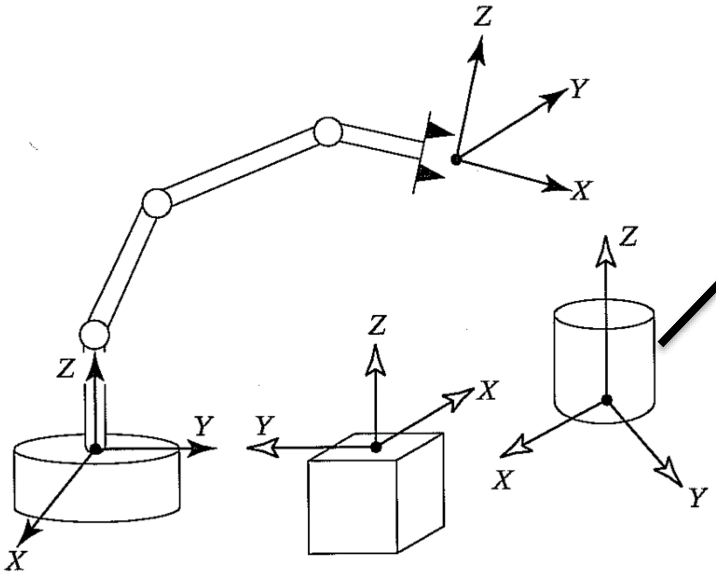
Spatial descriptions

Text book:
John Craig, “Introduction to Robotics mechanics and control”, 3rd Ed., Pearson Education Inc.

Ahmad AL-Jarrah

Introduction:

In the study of robotics, we are constantly concerned with the location of objects in three-dimensional space. These objects are the links of the manipulator, the parts and tools with which it deals, and other objects in the manipulator's environment.



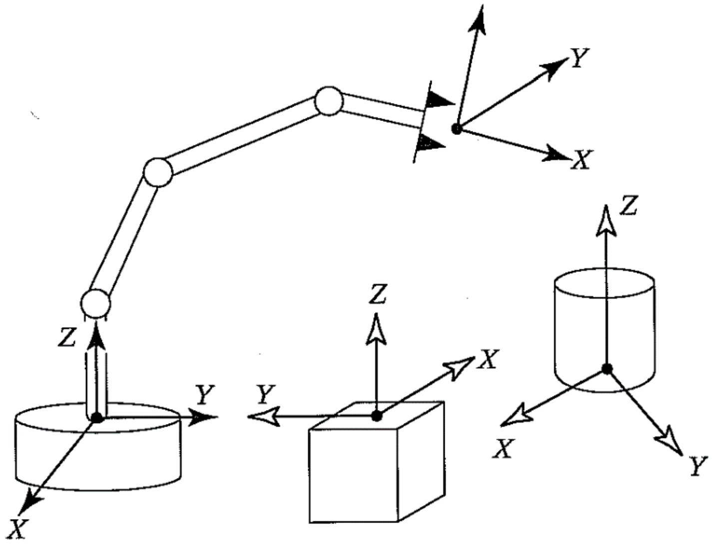
position and orientation.

FIGURE 1.5: Coordinate systems or "frames" are attached to the manipulator and to objects in the environment.

Introduction: cont.

In order to describe the position and orientation of a body in space, we will always attach a coordinate system, or **frame**, rigidly to the object. We then proceed to describe the position and orientation of this frame with respect to some reference coordinate system. (See Fig. 1.5.)

A **description** is used to specify attributes of various objects with which a manipulation system deals. These objects are parts, tools, and the manipulator itself. In this section, we discuss the description of positions, of orientations, and of an entity that contains both of these descriptions: the frame.

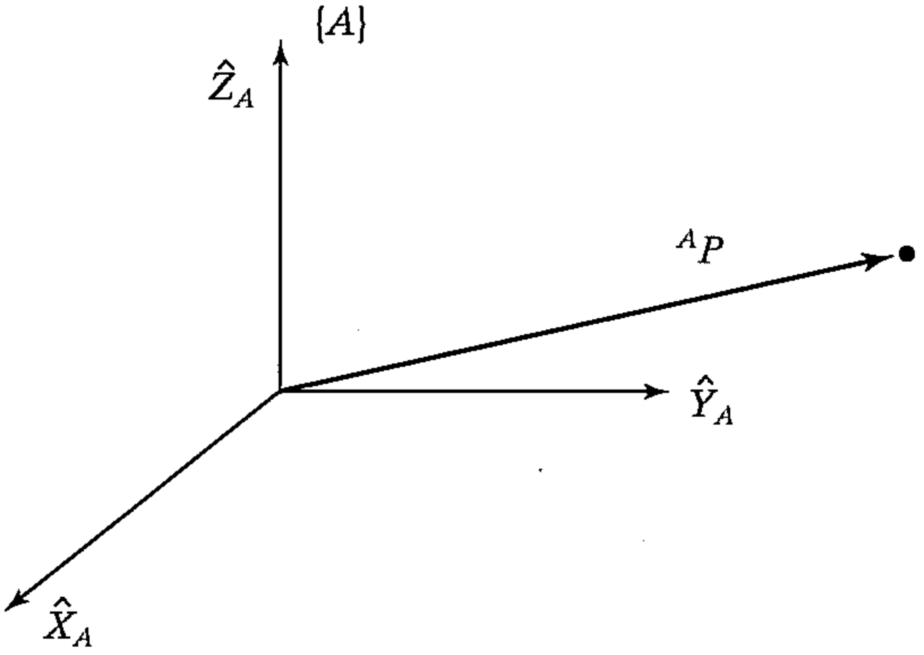


position and orientation.

FIGURE 1.5: Coordinate systems or "frames" are attached to the manipulator and to objects in the environment.

Description of a position

Once a coordinate system is established, we can locate any point in the universe with a 3×1 **position vector**. Because we will often define many coordinate systems in



$$|\hat{X}| = 1,$$

$$|\hat{Y}| = 1,$$

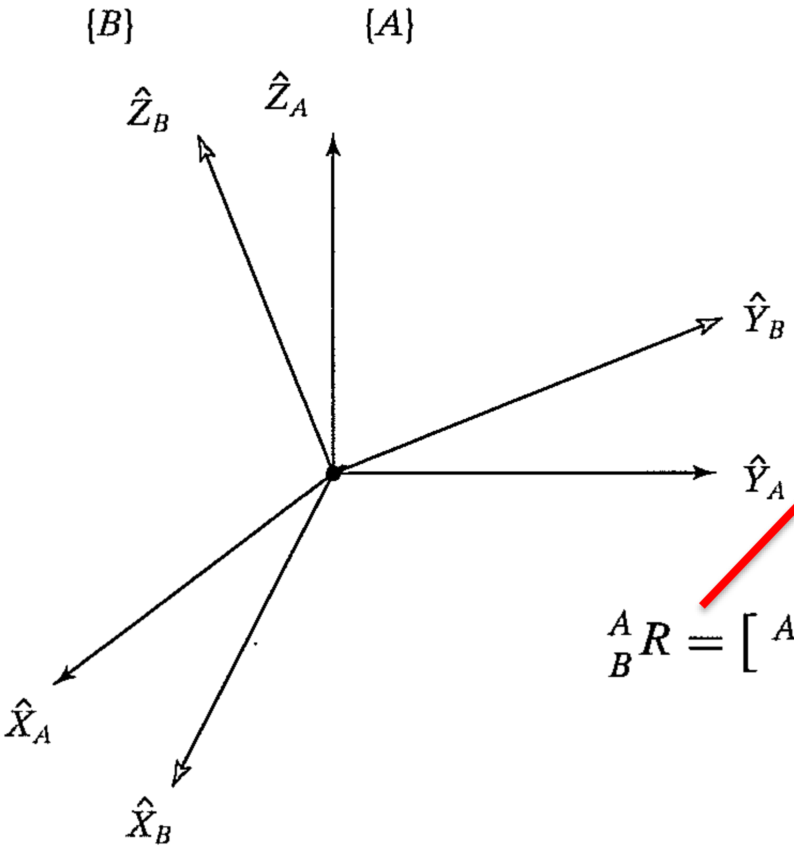
$$|\hat{Z}| = 1,$$

$${}^A P = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}.$$

FIGURE 2.1: Vector relative to frame (example).

Description of an orientation

Often, we will find it necessary not only to represent a point in space but also to describe the **orientation** of a body in space.

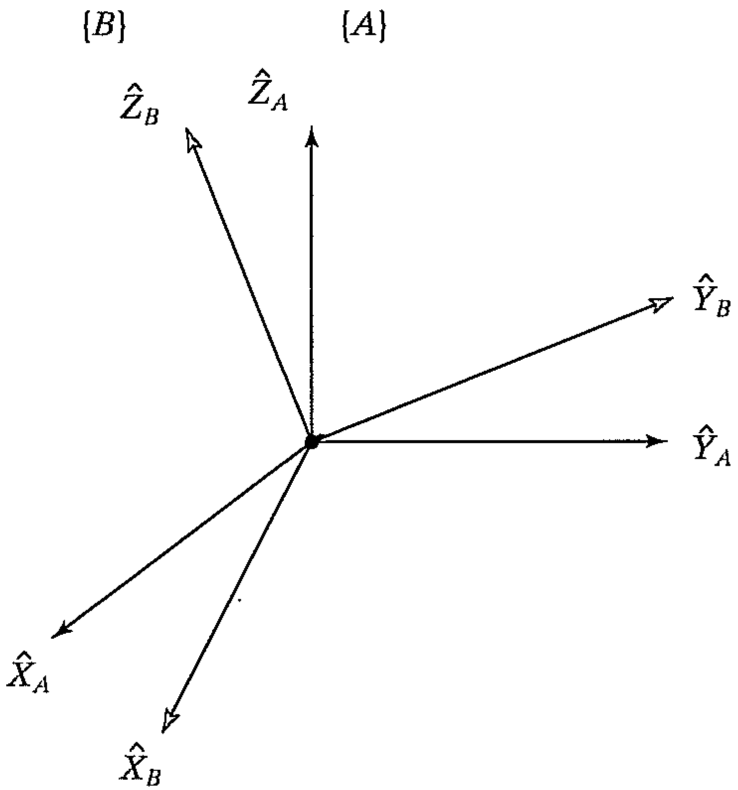


rotation matrix : $\{B\}$ relative to $\{A\}$

$$ {}^{A}_{B}R = \left[{}^{A}\hat{X}_B \;\; {}^{A}\hat{Y}_B \;\; {}^{A}\hat{Z}_B \right] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} . \quad \textbf{(2.2)} $$

We can give expressions for the scalars r_{ij} in (2.2) by noting that the components of any vector are simply the projections of that vector onto the unit directions of its reference frame. Hence, each component of ${}^{A}_{B}R$ in (2.2) can be written as the dot product of a pair of unit vectors:

$$
{}^{A}_{B}R = \left[\, {}^{A}\hat{X}_B \quad {}^{A}\hat{Y}_B \quad {}^{A}\hat{Z}_B \, \right] = \begin{bmatrix} \hat{X}_B \cdot \hat{X}_A & \hat{Y}_B \cdot \hat{X}_A & \hat{Z}_B \cdot \hat{X}_A \\ \hat{X}_B \cdot \hat{Y}_A & \hat{Y}_B \cdot \hat{Y}_A & \hat{Z}_B \cdot \hat{Y}_A \\ \hat{X}_B \cdot \hat{Z}_A & \hat{Y}_B \cdot \hat{Z}_A & \hat{Z}_B \cdot \hat{Z}_A \end{bmatrix}. \qquad (2.3)
$$



Further inspection of (2.3) shows that the rows of the matrix are the unit vectors of $\{A\}$ expressed in $\{B\}$; that is,

$$ {}^{A}_{B}R = \left[\, {}^{A}\hat{X}_{B} \quad {}^{A}\hat{Y}_{B} \quad {}^{A}\hat{Z}_{B} \,\right] = \begin{bmatrix} {}^{B}\hat{X}_{A}^{T} \\ {}^{B}\hat{Y}_{A}^{T} \\ {}^{B}\hat{Z}_{A}^{T} \end{bmatrix}. \tag{2.4} $$

Hence, ${}^{B}_{A}R$, the description of frame $\{A\}$ relative to $\{B\}$, is given by the transpose of (2.3); that is,

$$ {}^{B}_{A}R = {}^{A}_{B}R^{T}. \tag{2.5} $$

This suggests that the inverse of a rotation matrix is equal to its transpose, a fact that can be easily verified as

$$ {}^{A}_{B}R^{T}\,{}^{A}_{B}R = \begin{bmatrix} {}^{A}\hat{X}_{B}^{T} \\ {}^{A}\hat{Y}_{B}^{T} \\ {}^{A}\hat{Z}_{B}^{T} \end{bmatrix} \left[\, {}^{A}\hat{X}_{B} \quad {}^{A}\hat{Y}_{B} \quad {}^{A}\hat{Z}_{B} \,\right] = I_3, \tag{2.6} $$

where I_3 is the 3×3 identity matrix. Hence,

$$ {}^{A}_{B}R = {}^{B}_{A}R^{-1} = {}^{B}_{A}R^{T}. \tag{2.7} $$

2.3 MAPPINGS: CHANGING DESCRIPTIONS FROM FRAME TO FRAME

In a great many of the problems in robotics, we are concerned with expressing the same quantity in terms of various reference coordinate systems. The previous section introduced descriptions of positions, orientations, and frames; we now consider the mathematics of **mapping** in order to change descriptions from frame to frame.

Mappings involving translated frames

$$^{A}P = {}^{B}P + {}^{A}P_{BORG}.$$

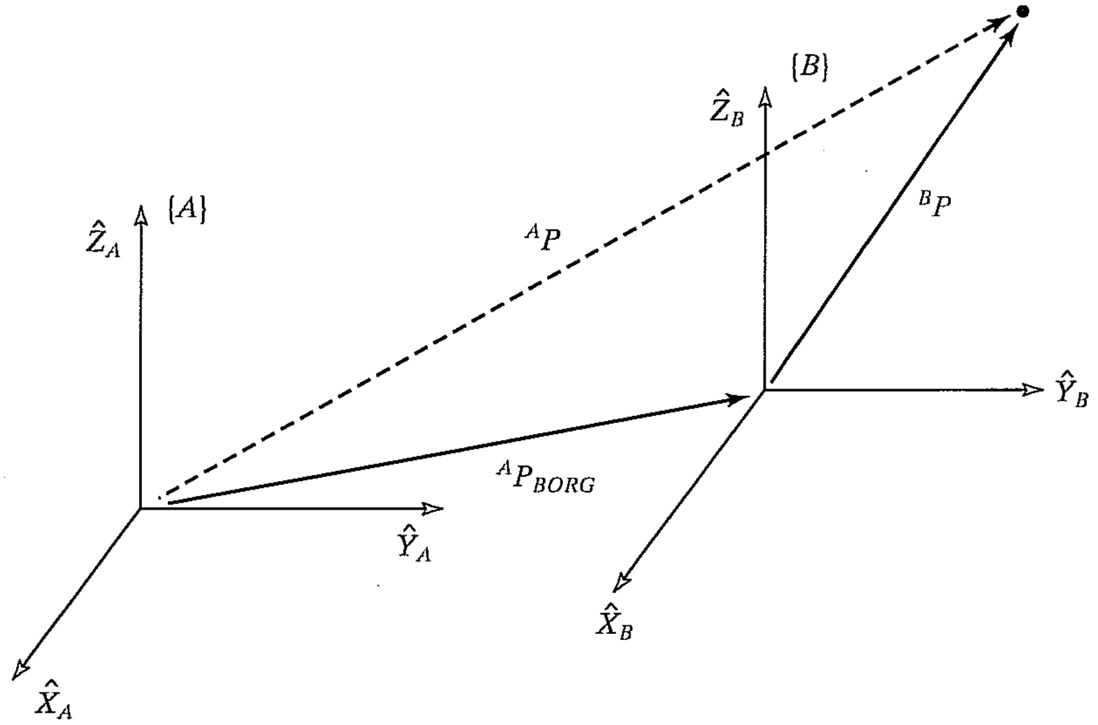
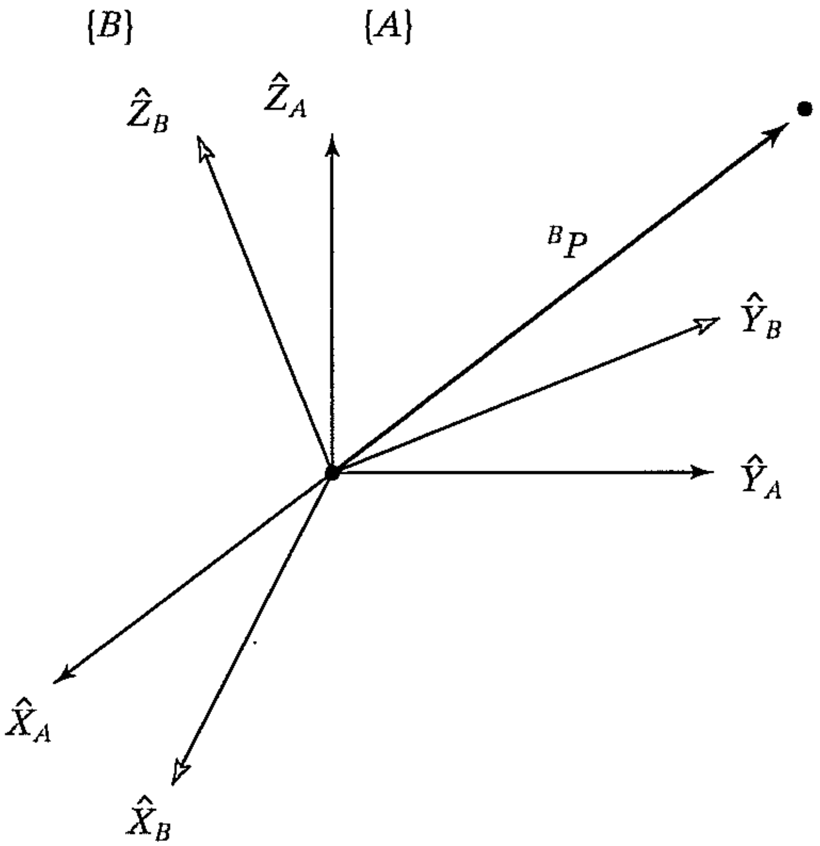


FIGURE 2.4: Translational mapping.

Mappings involving rotated frames



$$ {}^{A}P = {}^{A}_{B}R \, {}^{B}P. $$

FIGURE 2.5: Rotating the description of a vector.

Mappings involving rotated frames

EXAMPLE 2.1

Figure 2.6 shows a frame $\{B\}$ that is rotated relative to frame $\{A\}$ about $\hat{Z}$ by 30 degrees. Here, $\hat{Z}$ is pointing out of the page.

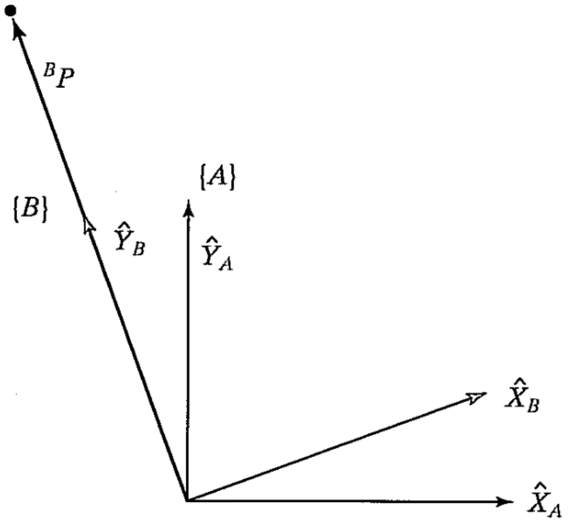


FIGURE 2.6: $\{B\}$ rotated 30 degrees about $\hat{Z}$.

$${}^{B}P = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}$$

$${}^{A}P = {}^{A}_{B}R\,{}^{B}P = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}.$$

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

$${}^{A}_{B}R = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}.$$

APPENDIX A

Formulas for rotation about the principal axes by θ:

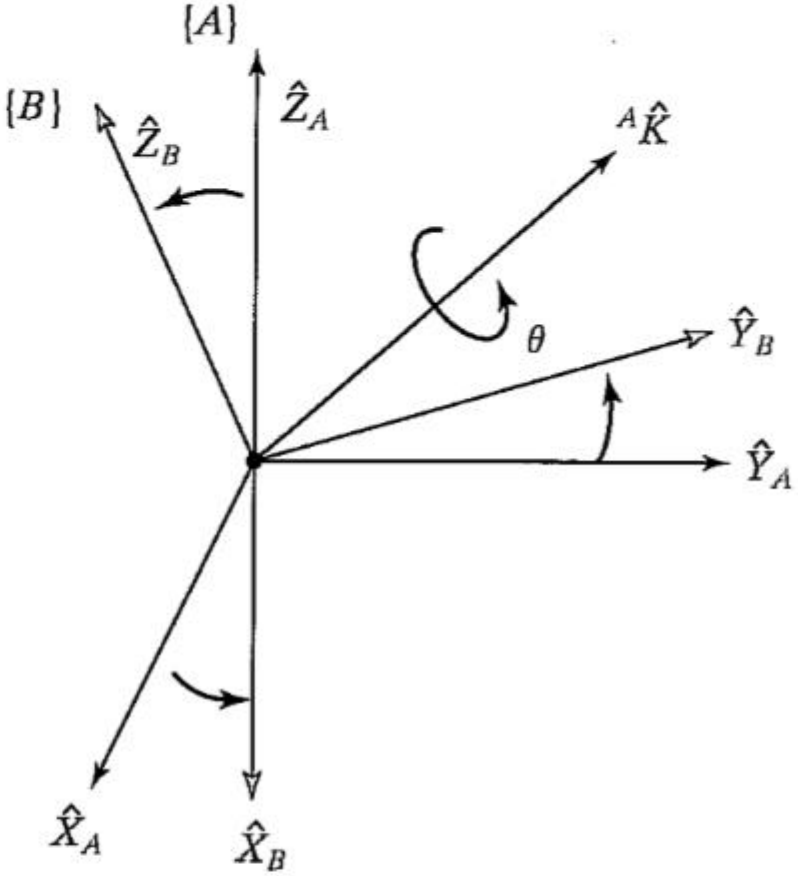
$$R_X(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}, \tag{A.1}$$

$$R_Y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \tag{A.2}$$

$$R_Z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.3}$$

Rotation about an arbitrary axis

Start with the frame coincident with a known frame $\{A\}$; then rotate $\{B\}$ about the vector ${}^{A}\hat{K}$ by an angle θ according to the right-hand rule.



$$R_K(\theta) = \begin{bmatrix} k_x k_x v\theta + c\theta & k_x k_y v\theta - k_z s\theta & k_x k_z v\theta + k_y s\theta \\ k_x k_y v\theta + k_z s\theta & k_y k_y v\theta + c\theta & k_y k_z v\theta - k_x s\theta \\ k_x k_z v\theta - k_y s\theta & k_y k_z v\theta + k_x s\theta & k_z k_z v\theta + c\theta \end{bmatrix},$$

where $c\theta = \cos\theta$, $s\theta = \sin\theta$, $v\theta = 1 - \cos\theta$, and ${}^A\hat{K} = [k_x k_y k_z]^T$

$${}^A_B R_K(\theta) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix},$$

**Given the orientation,
How to compute the angle
and vector?**

$$\theta = \text{Acos}\left(\frac{r_{11} + r_{22} + r_{33} - 1}{2}\right)$$

$$\hat{K} = \frac{1}{2\sin\theta} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}.$$

EXAMPLE 2.8

A frame $\{B\}$ is described as initially coincident with $\{A\}$. We then rotate $\{B\}$ about the vector ${}^{A}\hat{K} = [0.7070\ 7070\ 0]^T$ (passing through the origin) by an amount $\theta = 30$ degrees. Give the frame description of $\{B\}$.

Substituting into (2.80) yields the rotation-matrix part of the frame description. There was no translation of the origin, so the position vector is $[0, 0, 0]^T$. Hence,

$$ {}^{A}_{B}T = \begin{bmatrix} 0.933 & 0.067 & 0.354 & 0.0 \\ 0.067 & 0.933 & -0.354 & 0.0 \\ -0.354 & 0.354 & 0.866 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}. \qquad (2.83) $$

Homogeneous Transformation

Mappings involving general frames

Very often, we know the description of a vector with respect to some frame $\{B\}$, and we would like to know its description with respect to another frame, $\{A\}$. We now consider the general case of mapping. Here, the origin of frame $\{B\}$ is not coincident with that of frame $\{A\}$ but has a general vector offset. The vector that locates $\{B\}$'s origin is called ${}^{A}P_{BORG}$. Also $\{B\}$ is rotated with respect to $\{A\}$, as described by ${}^{A}_{B}R$. Given ${}^{B}P$, we wish to compute ${}^{A}P$, as in Fig. 2.7.

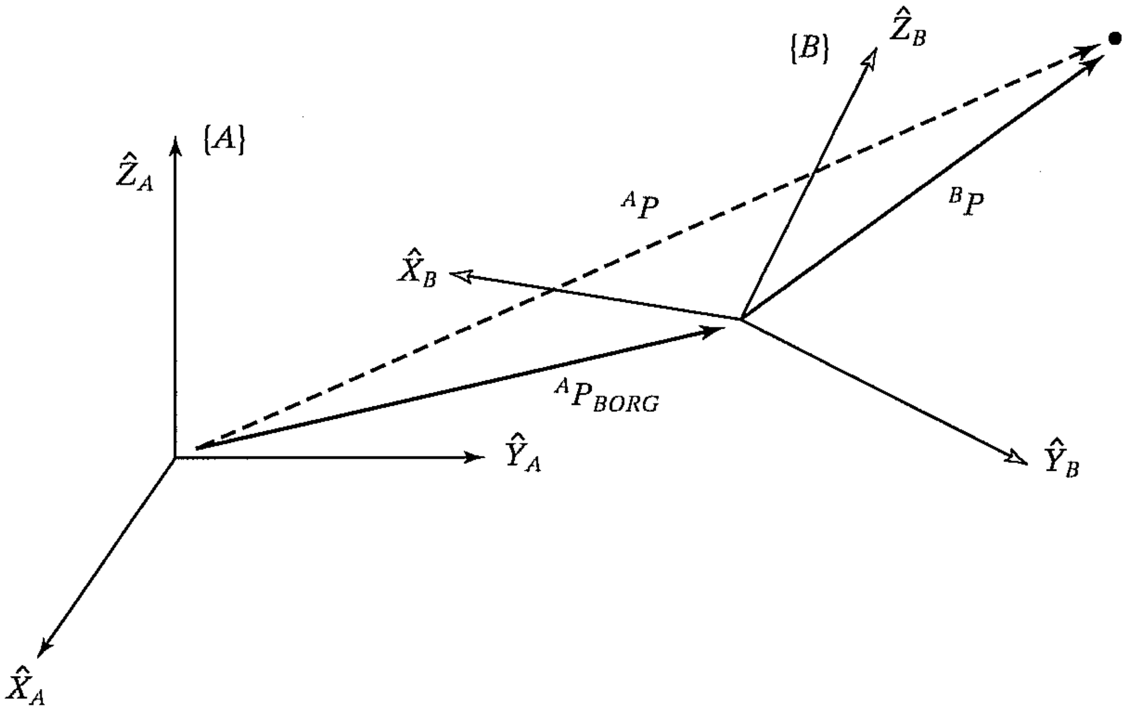
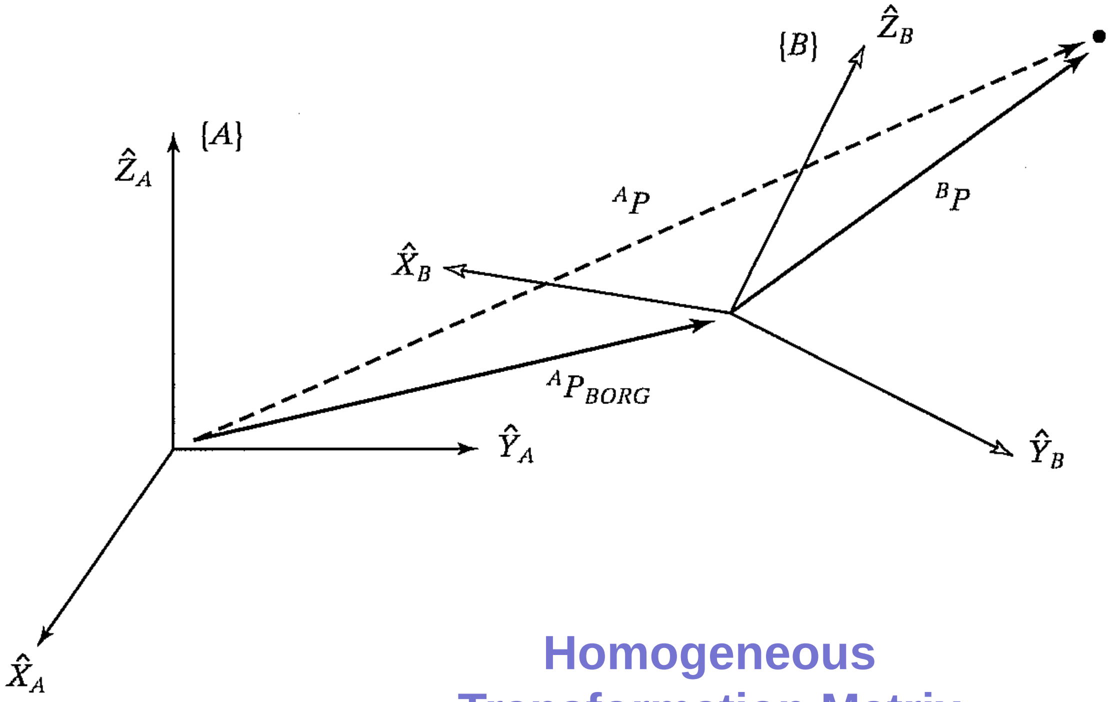


FIGURE 2.7: General transform of a vector.



$${}^{A}P = {}^{A}_{B}R\,{}^{B}P + {}^{A}P_{BORG}.$$

Homogeneous Transformation Matrix

$${}^{A}P = {}^{A}_{B}T\,{}^{B}P.$$

$$\begin{bmatrix} {}^{A}P \\ 1 \end{bmatrix} = \overbrace{\left[\begin{array}{ccc|c} & {}^{A}_{B}R & & {}^{A}P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]}^{{}^{A}_{B}T} \begin{bmatrix} {}^{B}P \\ 1 \end{bmatrix}.$$

EXAMPLE 2.2

Figure 2.8 shows a frame $\{B\}$, which is rotated relative to frame $\{A\}$ about $\hat{Z}$ by 30 degrees, translated 10 units in $\hat{X}_A$, and translated 5 units in $\hat{Y}_A$. Find ${}^{A}P$, where ${}^{B}P = [3.07.00.0]^T$.

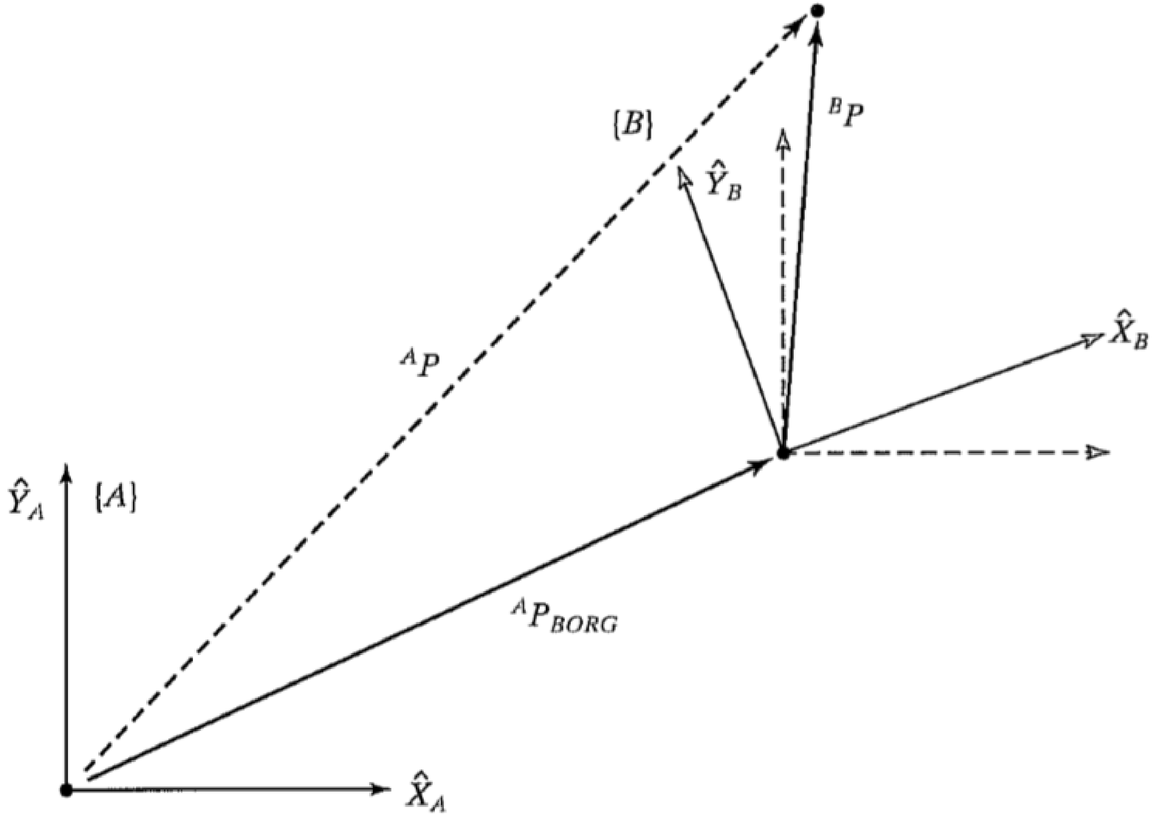


FIGURE 2.8: Frame $\{B\}$ rotated and translated.

The definition of frame $\{B\}$ is

$$ {}^{A}_{B}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 5.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.21} $$

Given

$$ {}^{B}P = \begin{bmatrix} 3.0 \\ 7.0 \\ 0.0 \end{bmatrix}, \tag{2.22} $$

we use the definition of $\{B\}$ just given as a transformation:

$$ {}^{A}P = {}^{A}_{B}T\ {}^{B}P = \begin{bmatrix} 9.098 \\ 12.562 \\ 0.000 \end{bmatrix}. \tag{2.23} $$

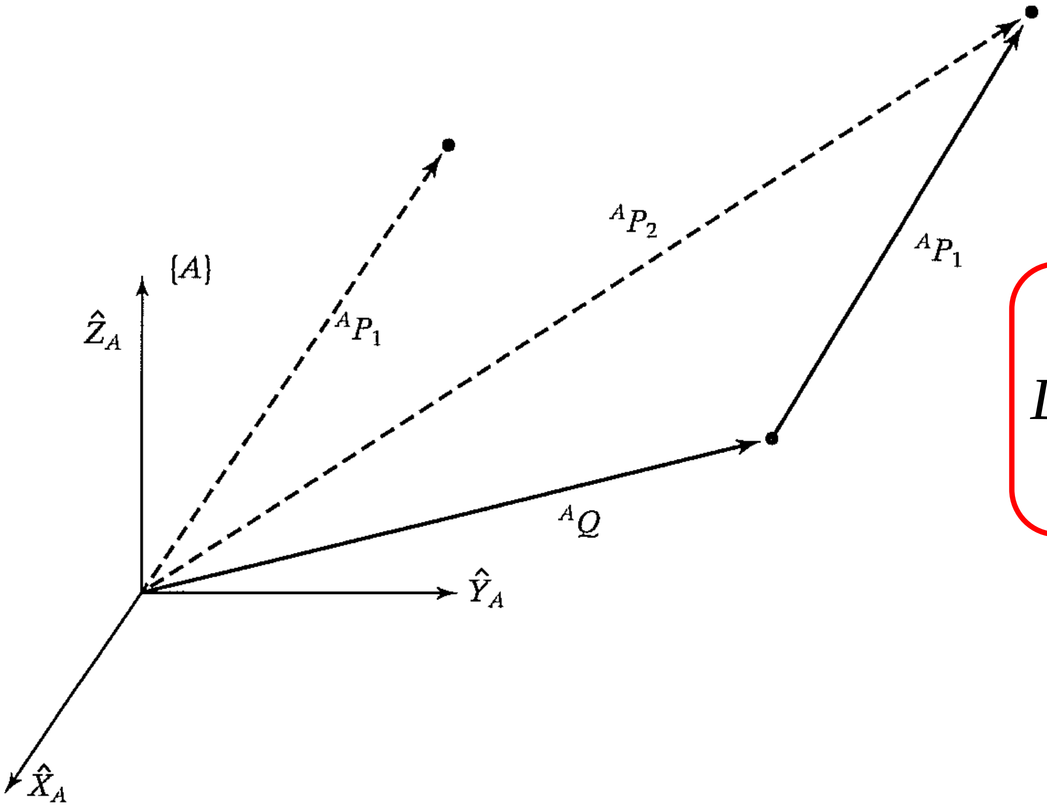
2.4 OPERATORS: TRANSLATIONS, ROTATIONS, AND TRANSFORMATIONS

The same mathematical forms used to map points between frames can also be interpreted as operators that translate points, rotate vectors, or do both. This section illustrates this interpretation of the mathematics we have already developed.

Translational operators

$$^{A}P_2 = {}^{A}P_1 + {}^{A}Q.$$

$$^{A}P_2 = D_Q(q)\,{}^{A}P_1,$$



^{A}Q **: is the translational vector**

$D_Q(q)$ **: is the operator equivalent to Homogeneous transform**

$$D_Q(q) = \begin{bmatrix} 1 & 0 & 0 & q_x \\ 0 & 1 & 0 & q_y \\ 0 & 0 & 1 & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotational operators

$$^AP_2 = R\ ^AP_1.$$

$$^AP_2 = \underbrace{R_K(\theta)}\ ^AP_1.$$

Rotational operator that performs a rotation about the axis direction $\vec{K}$ by angle θ

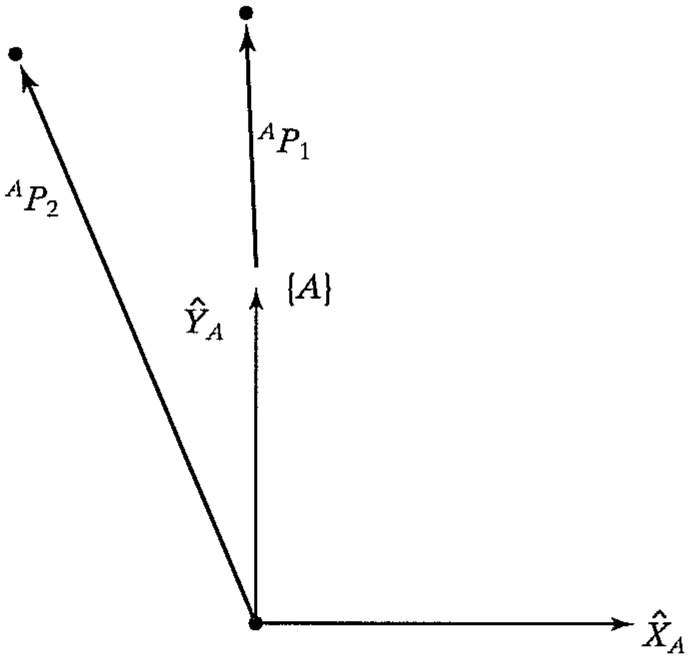


FIGURE 2.10: The vector $^AP_1$ rotated 30 degrees about $\hat{Z}$. ⇨ $R_z(\Theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$

Figure 2.10 shows a vector ${}^{A}P_1$. We wish to compute the vector obtained by rotating this vector about $\hat{Z}$ by 30 degrees. Call the new vector ${}^{A}P_2$.

The rotation matrix that rotates vectors by 30 degrees about $\hat{Z}$ is the same as the rotation matrix that describes a frame rotated 30 degrees about $\hat{Z}$ relative to the reference frame. Thus, the correct rotational operator is

$$R_z(30.0) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}. \tag{2.30}$$

Given

$${}^{A}P_1 = \begin{bmatrix} 0.0 \\ 2.0 \\ 0.0 \end{bmatrix}, \tag{2.31}$$

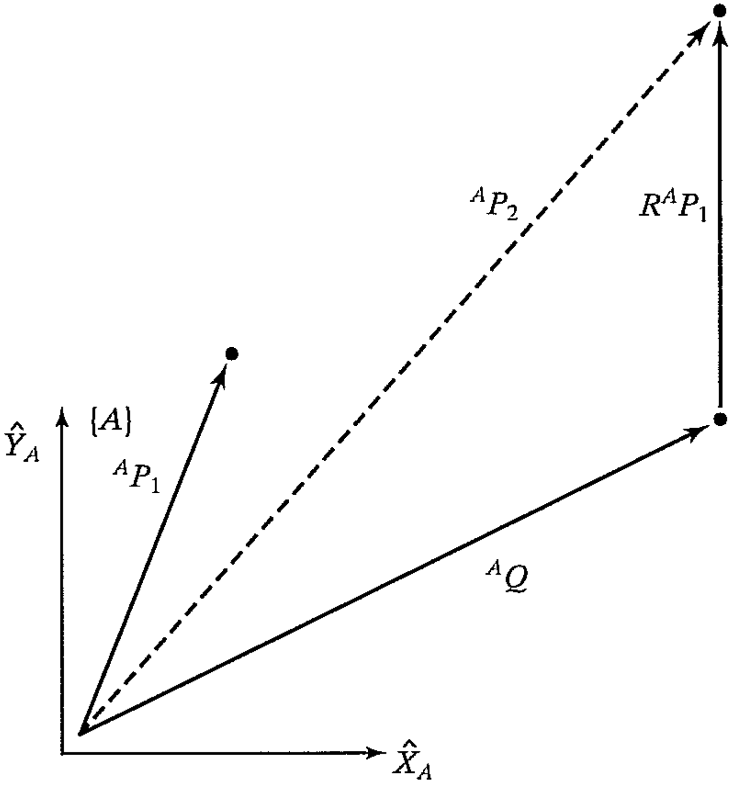
we calculate ${}^{A}P_2$ as

$${}^{A}P_2 = R_z(30.0)\,{}^{A}P_1 = \begin{bmatrix} -1.000 \\ 1.732 \\ 0.000 \end{bmatrix}. \tag{2.32}$$

Transformation operators

EXAMPLE 2.4

Figure 2.11 shows a vector ${}^{A}P_1$. We wish to rotate it about $\hat{Z}$ by 30 degrees and translate it 10 units in $\hat{X}_A$ and 5 units in $\hat{Y}_A$. Find ${}^{A}P_2$, where ${}^{A}P_1 = [3.0\ 7.0\ 0.0]^T$.



$$T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 10.0 \\ 0.500 & 0.866 & 0.000 & 5.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$${}^{A}P_2 = T\ {}^{A}P_1.$$

$${}^{A}P_2 = T\ {}^{A}P_1 = \begin{bmatrix} 9.098 \\ 12.562 \\ 0.000 \end{bmatrix}.$$

FIGURE 2.11: The vector ${}^{A}P_1$ rotated and translated to form ${}^{A}P_2$.

2.5 SUMMARY OF INTERPRETATIONS

As a general tool to represent frames, we have introduced the *homogeneous transform*, a 4 × 4 matrix containing orientation and position information.

We have introduced three interpretations of this homogeneous transform:

1. It is a *description of a frame.* ${}^A_BT$ describes the frame $\{B\}$ relative to the frame $\{A\}$. Specifically, the columns of ${}^A_BR$ are unit vectors defining the directions of the principal axes of $\{B\}$, and ${}^AP_{BORG}$ locates the position of the origin of $\{B\}$.
2. It is a *transform mapping.* ${}^A_BT$ maps ${}^BP \rightarrow {}^AP$.
3. It is a *transform operator*. T operates on ${}^AP_1$ to create ${}^AP_2$.

Compound transformations

we have ${}^{C}P$ and wish to find ${}^{A}P$.

$${}^{B}P = {}^{B}_{C}T\,{}^{C}P$$

$${}^{A}P = {}^{A}_{B}T\,{}^{B}P.$$

$${}^{A}P = {}^{A}_{B}T\,{}^{B}_{C}T\,{}^{C}P$$

$${}^{A}_{C}T = {}^{A}_{B}T\,{}^{B}_{C}T$$

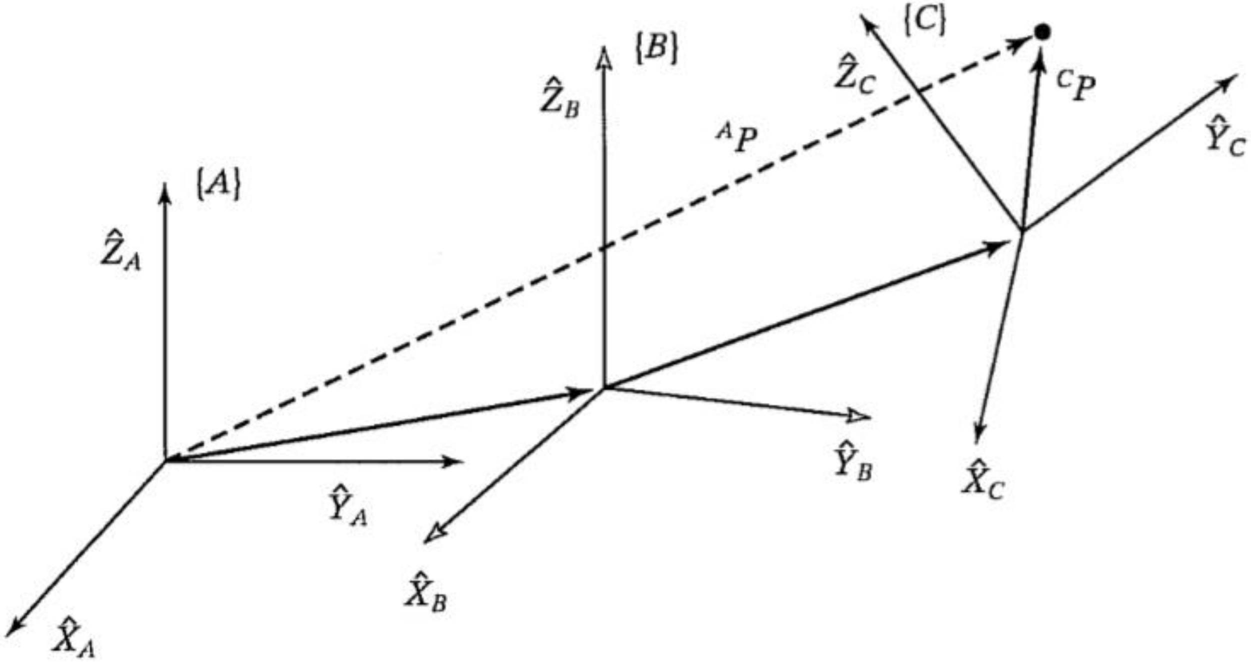


FIGURE 2.12: Compound frames: Each is known relative to the previous one.

Inverting a transform

Consider a frame $\{B\}$ that is known with respect to a frame $\{A\}$—that is, we know the value of ${}^{A}_{B}T$.

Note that, with our notation,

$$ {}^{B}_{A}T = {}^{A}_{B}T^{-1}. $$

We have the option of finding the inverse of 4x4 matrix or using simpler process as follows:

$$ {}^{B}_{A}R = {}^{A}_{B}R^{T} $$

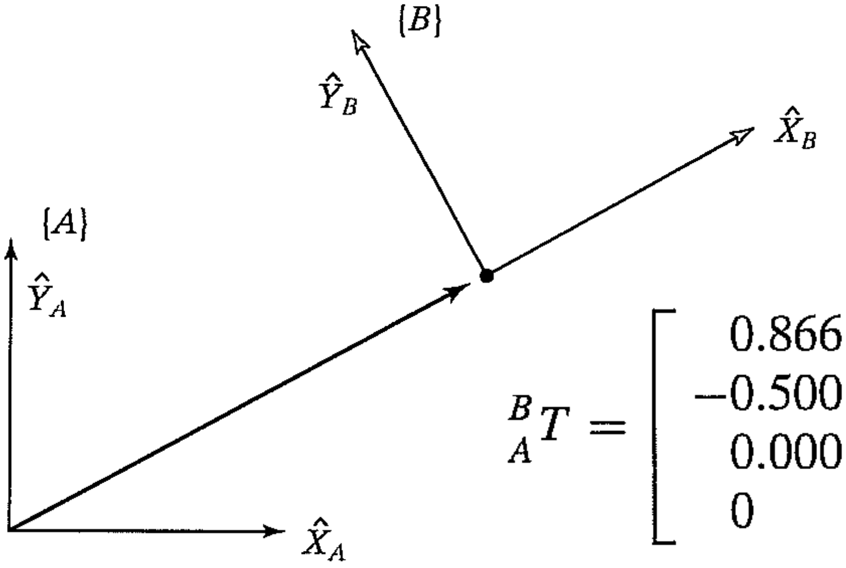
$$ \boxed{ {}^{B}_{A}T = \left[\begin{array}{ccc|c} & {}^{A}_{B}R^{T} & & -{}^{A}_{B}R^{T}\,{}^{A}P_{BORG} \\ \hline 0 & 0 & 0 & 1 \end{array} \right] } $$

EXAMPLE 2.5

Figure 2.13 shows a frame $\{B\}$ that is rotated relative to frame $\{A\}$ about $\hat{Z}$ by 30 degrees and translated four units in $\hat{X}_A$ and three units in $\hat{Y}_A$. Thus, we have a description of ${}^{A}_{B}T$. Find ${}^{B}_{A}T$.

The frame defining $\{B\}$ is

$$ {}^{A}_{B}T = \begin{bmatrix} 0.866 & -0.500 & 0.000 & 4.0 \\ 0.500 & 0.866 & 0.000 & 3.0 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \qquad (2.46) $$



$$ {}^{B}_{A}T = \begin{bmatrix} 0.866 & 0.500 & 0.000 & -4.964 \\ -0.500 & 0.866 & 0.000 & -0.598 \\ 0.000 & 0.000 & 1.000 & 0.0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

FIGURE 2.13: $\{B\}$ relative to $\{A\}$.

2.7 TRANSFORM EQUATIONS

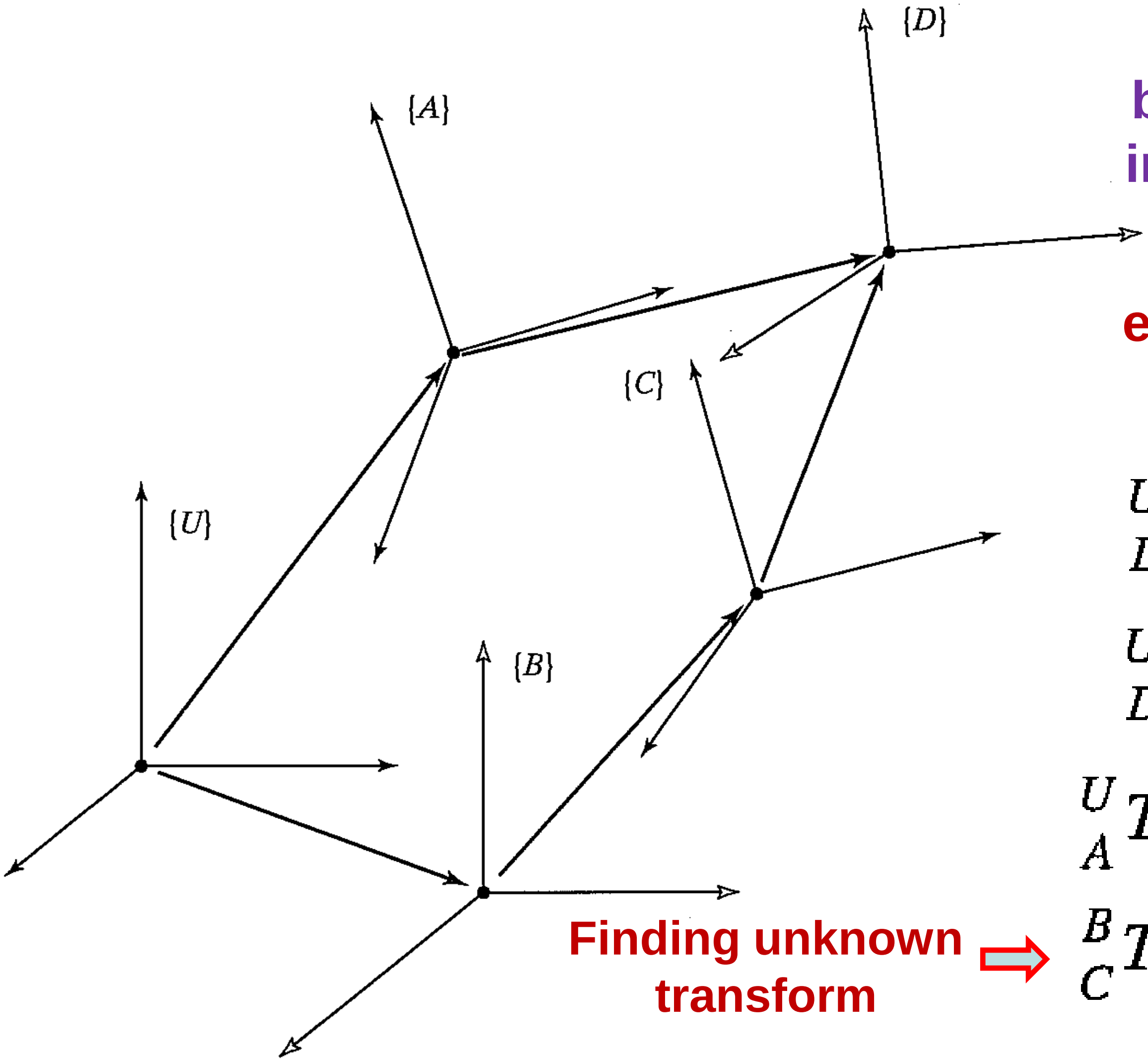


FIGURE 2.14: Set of transforms forming a loop.

The arrows directions between frames origins indicates which way the frames are defined !

e.g. Frame D is defined relative to frame A

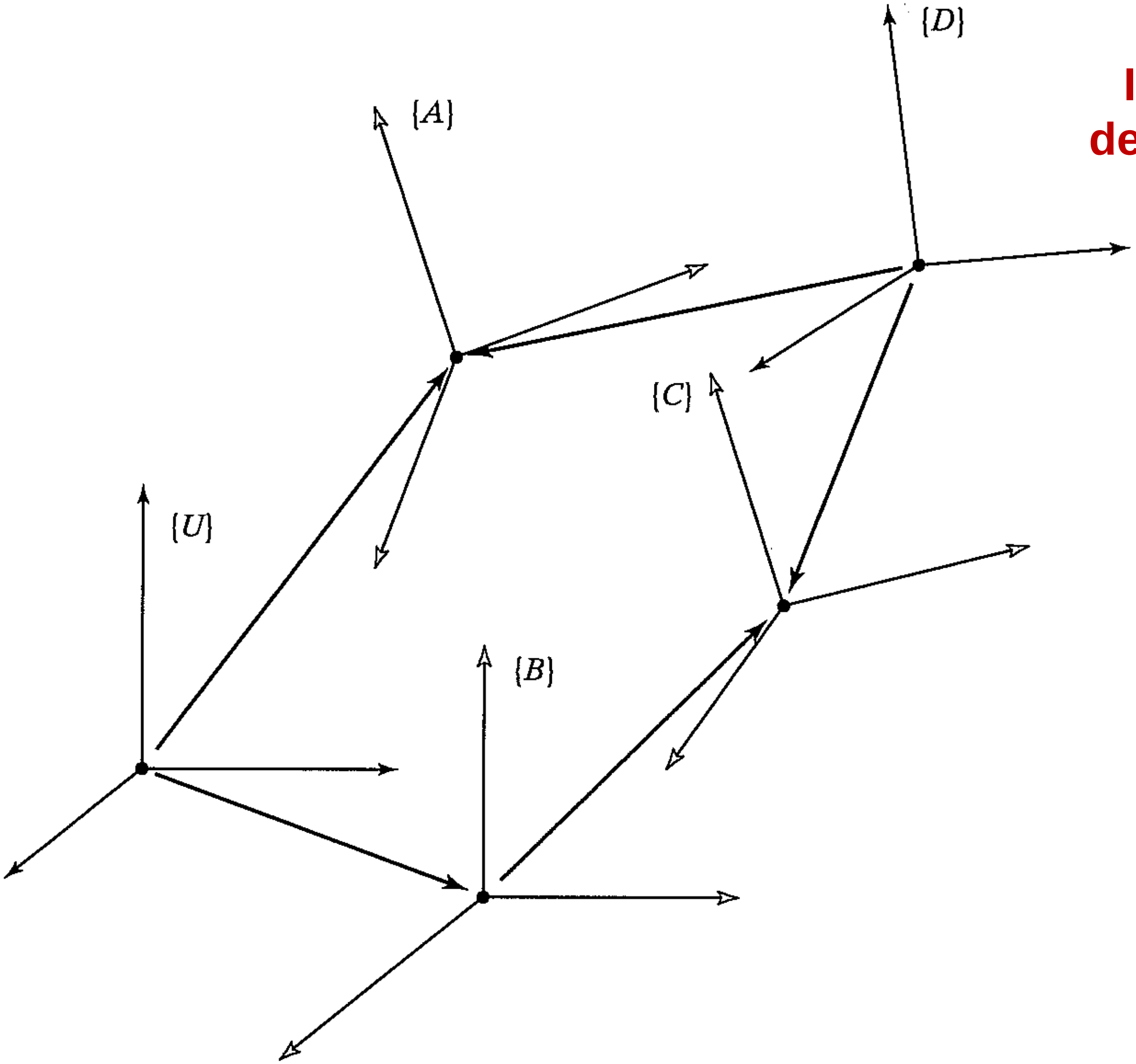
$$ {}^U_D T = {}^U_A T \, {}^A_D T; $$

$$ {}^U_D T = {}^U_B T \, {}^B_C T \, {}^C_D T. $$

$$ {}^U_A T \, {}^A_D T = {}^U_B T \, {}^B_C T \, {}^C_D T. $$

Finding unknown transform ⇨

$$ {}^B_C T = {}^U_B T^{-1} \, {}^U_A T \, {}^A_D T \, {}^C_D T^{-1}. $$



In this case, Frame A is defined relative to frame D

$$ {}^{U}_{C}T = {}^{U}_{A}T \, {}^{D}_{A}T^{-1} \, {}^{D}_{C}T $$

$$ {}^{U}_{C}T = {}^{U}_{B}T \, {}^{B}_{C}T. $$

Example 2.6:

Calculate the position and orientation of the bolt relative to the manipulator's hand.

$$ {}^{T}_{G}T = {}^{B}_{T}T^{-1}\, {}^{B}_{S}T\, {}^{S}_{G}T. $$

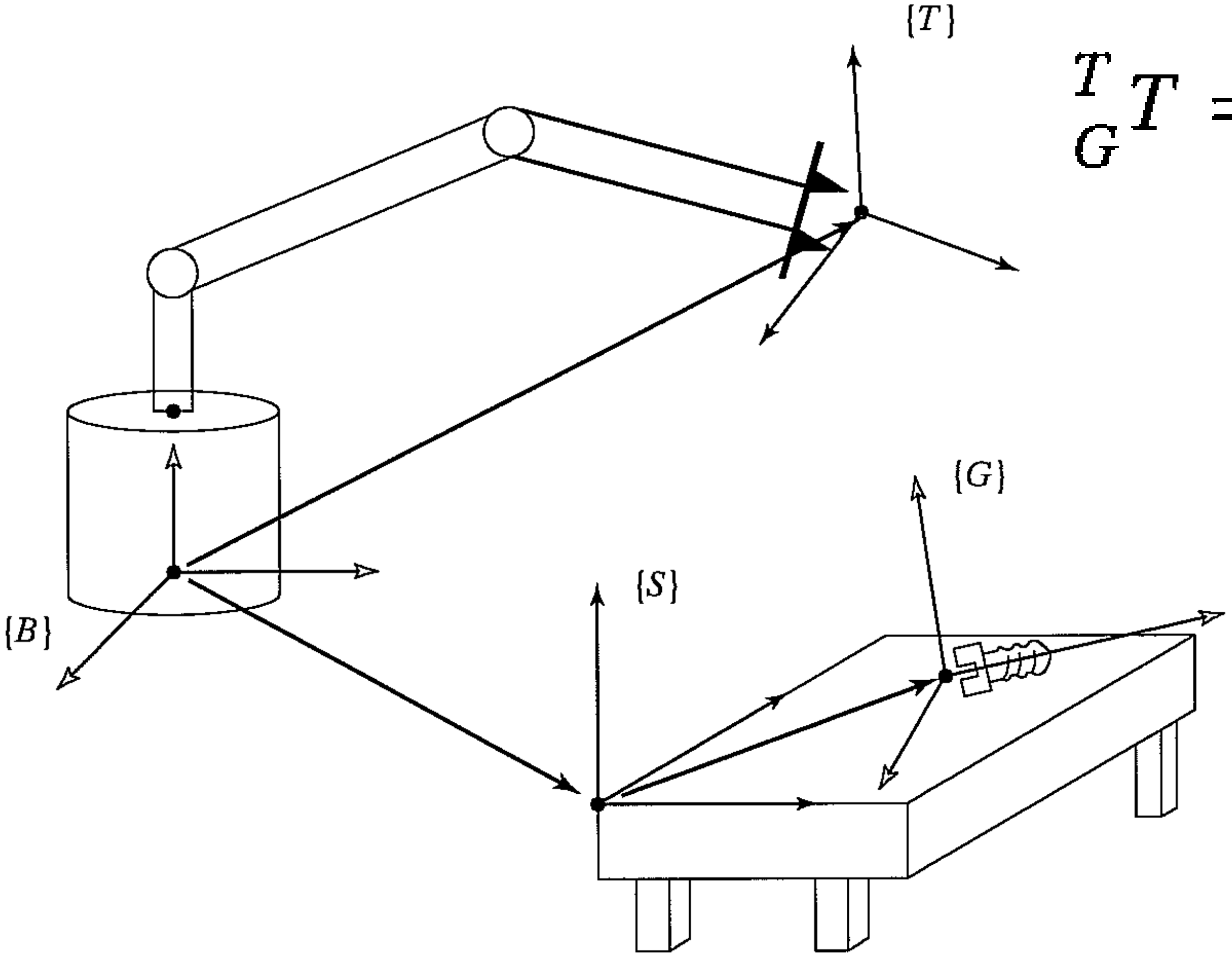


FIGURE 2.16: Manipulator reaching for a bolt.

2.8 More on Representation of Orientation

rotation matrix determinant is +1

$$R = [\hat{X}\ \hat{Y}\ \hat{Z}].$$

⇩

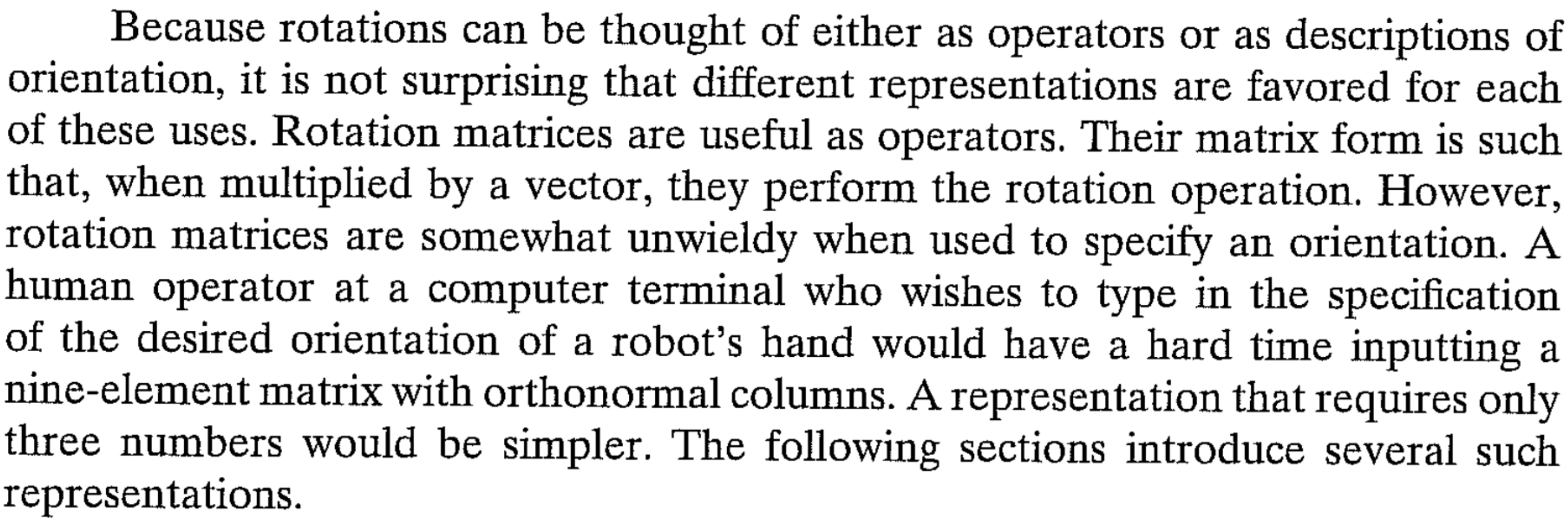
Can we describe the orientation with fewer than 9 elements?!

$$|\hat{X}| = 1,$$
$$|\hat{Y}| = 1,$$
$$|\hat{Z}| = 1,$$
$$\hat{X} \cdot \hat{Y} = 0,$$
$$\hat{X} \cdot \hat{Z} = 0,$$
$$\hat{Y} \cdot \hat{Z} = 0.$$

Clearly, the nine elements of a rotation matrix are not all independent. In fact, given a rotation matrix, R, it is easy to write down the six dependencies between the elements.

Therefore, rotation matrix can be specified by just three parameters.

More over…



Because rotations can be thought of either as operators or as descriptions of orientation, it is not surprising that different representations are favored for each of these uses. Rotation matrices are useful as operators. Their matrix form is such that, when multiplied by a vector, they perform the rotation operation. However, rotation matrices are somewhat unwieldy when used to specify an orientation. A human operator at a computer terminal who wishes to type in the specification of the desired orientation of a robot's hand would have a hard time inputting a nine-element matrix with orthonormal columns. A representation that requires only three numbers would be simpler. The following sections introduce several such representations.

X–Y–Z fixed angles

One method of describing the orientation of a frame $\{B\}$ is as follows:

> Start with the frame coincident with a known reference frame $\{A\}$. Rotate $\{B\}$ first about $\hat{X}_A$ by an angle γ, then about $\hat{Y}_A$ by an angle β, and, finally, about $\hat{Z}_A$ by an angle α.

Each of the three rotations takes place about an axis in the fixed reference frame $\{A\}$. We will call this convention for specifying an orientation **X–Y–Z fixed angles**. The word "fixed" refers to the fact that the rotations are specified about the fixed (i.e., nonmoving) reference frame (Fig. 2.17). Sometimes this convention is referred to as **roll, pitch, yaw angles**, but care must be used, as this name is often given to other related but different conventions.

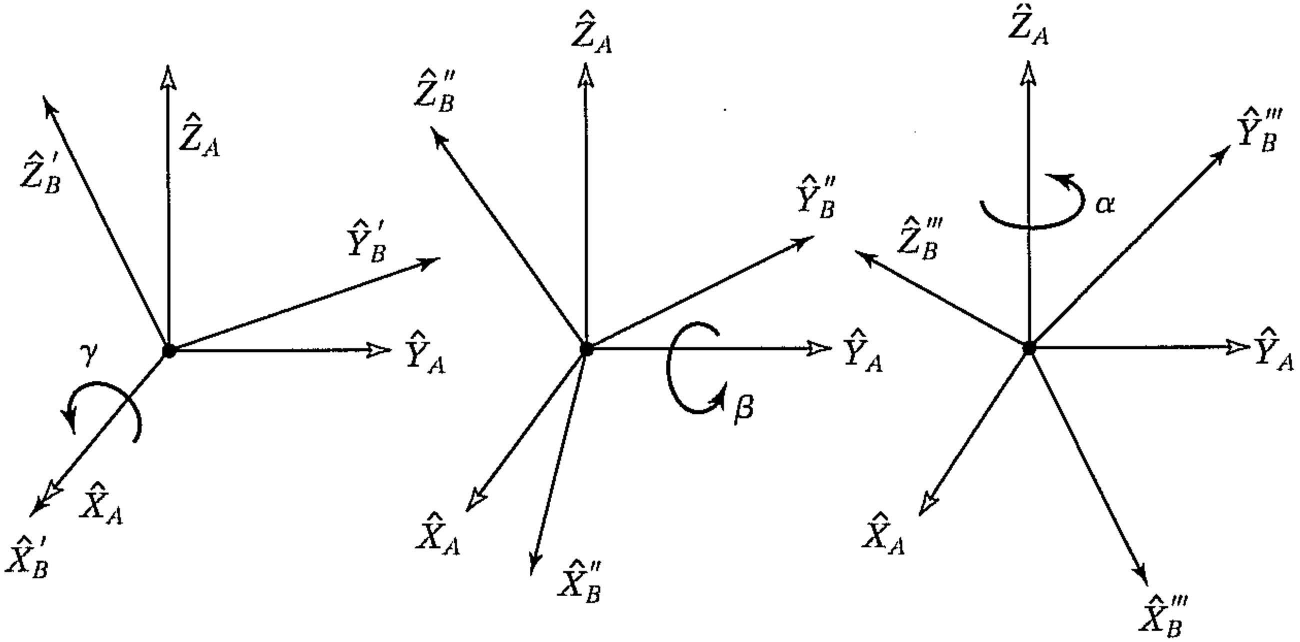


FIGURE 2.17: X–Y–Z fixed angles. Rotations are performed in the order $R_X(\gamma)$, $R_Y(\beta)$, $R_Z(\alpha)$.

$$
{}^A_BR_{XYZ}(\gamma,\beta,\alpha) = R_Z(\alpha)R_Y(\beta)R_X(\gamma)
$$

$$
= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix},
$$

$$ {}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}. $$

$$ {}^{A}_{B}R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}. $$

Given the orientation, How to compute the angles?

$$ \beta = \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) $$

$$ \alpha = \text{Atan2}(r_{21}/c\beta, r_{11}/c\beta) $$

$$ \gamma = \text{Atan2}(r_{32}/c\beta, r_{33}/c\beta) $$

EXAMPLE 2.7

Consider two rotations, one about $\hat{Z}$ by 30 degrees and one about $\hat{X}$ by 30 degrees:

$$R_z(30) = \begin{bmatrix} 0.866 & -0.500 & 0.000 \\ 0.500 & 0.866 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix}$$

$$R_x(30) = \begin{bmatrix} 1.000 & 0.000 & 0.000 \\ 0.000 & 0.866 & -0.500 \\ 0.000 & 0.500 & 0.866 \end{bmatrix}$$

$$R_z(30) R_x(30) = \begin{bmatrix} 0.87 & -0.43 & 0.25 \\ 0.50 & 0.75 & -0.43 \\ 0.00 & 0.50 & 0.87 \end{bmatrix}$$

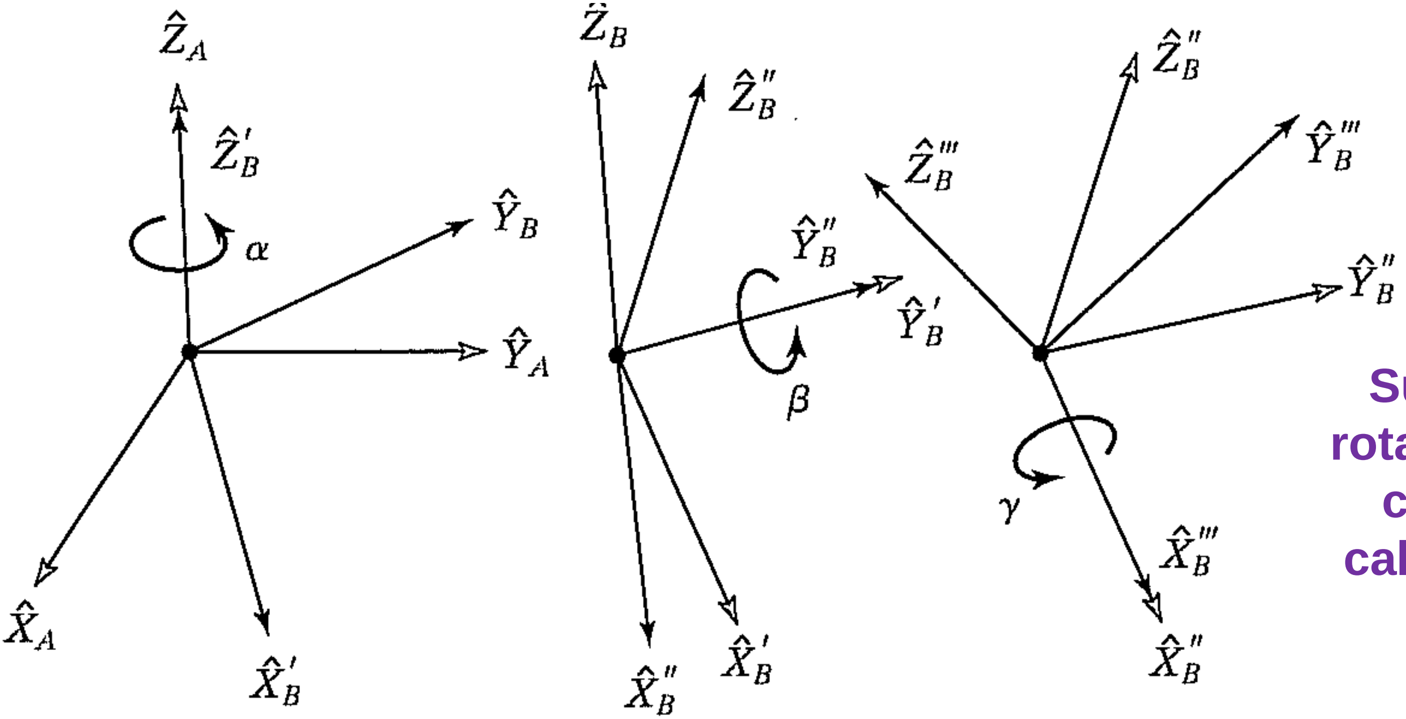
$$\neq R_x(30) R_z(30) = \begin{bmatrix} 0.87 & -0.50 & 0.00 \\ 0.43 & 0.75 & -0.50 \\ 0.25 & 0.43 & 0.87 \end{bmatrix}$$

Z–Y–X Euler angles

Another possible description of a frame $\{B\}$ is as follows:

> Start with the frame coincident with a known frame $\{A\}$. Rotate $\{B\}$ first about $\hat{Z}_B$ by an angle α, then about $\hat{Y}_B$ by an angle β, and, finally, about $\hat{X}_B$ by an angle γ.

In this representation, each rotation is performed about an axis of the moving system $\{B\}$ rather than one of the fixed reference $\{A\}$.



Such set of three rotations around the current frame is called Euler Angles

$$
{}^{A}_{B}R_{Z'Y'X'} = R_Z(\alpha)R_Y(\beta)R_X(\gamma)
$$

$$
= \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix},
$$

$$
{}^{A}_{B}R_{Z'Y'X'}(\alpha, \beta, \gamma) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix}.
$$

$$
\boxed{\begin{aligned} \beta &= \text{Atan2}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\ \alpha &= \text{Atan2}(r_{21}/c\beta, r_{11}/c\beta) \\ \gamma &= \text{Atan2}(r_{32}/c\beta, r_{33}/c\beta) \end{aligned}}
$$

Z–Y–Z Euler angles

Another possible description of a frame $\{B\}$ is

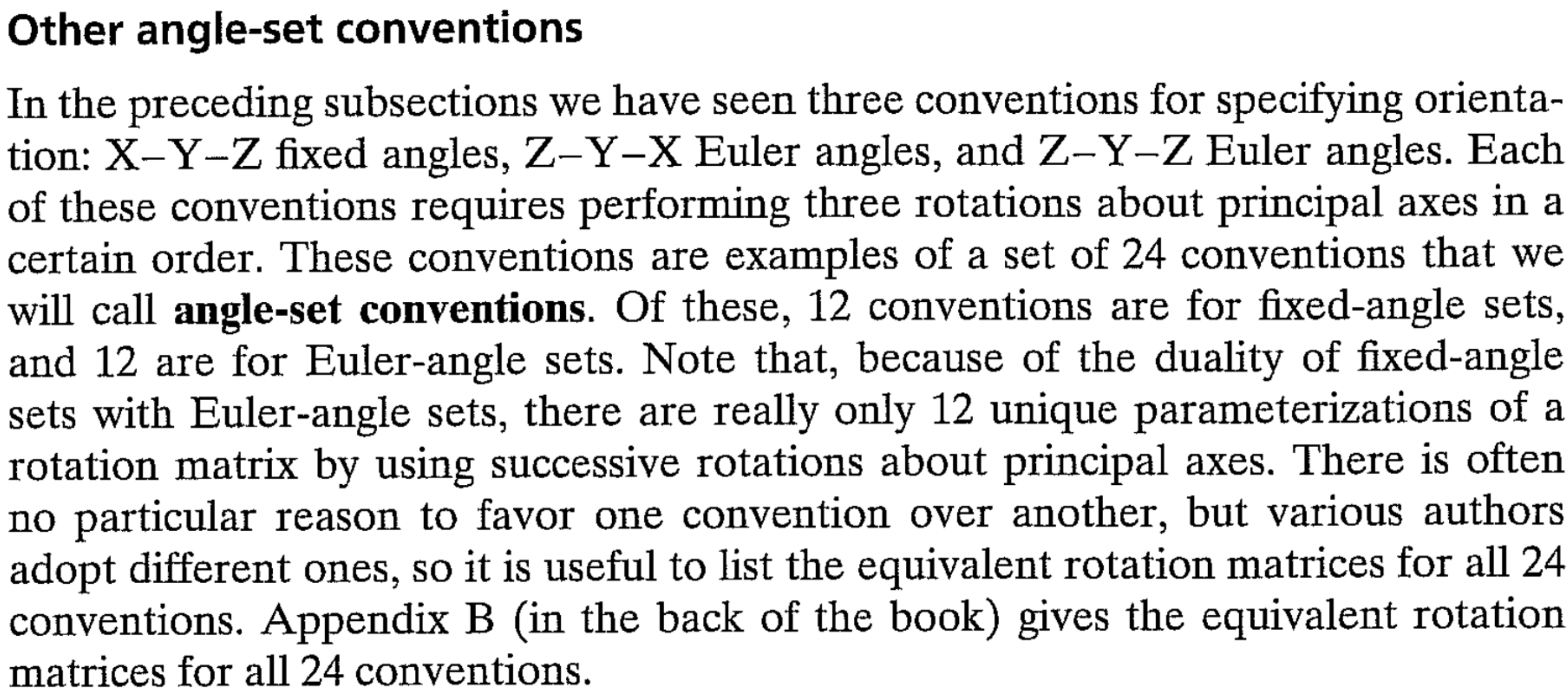
> Start with the frame coincident with a known frame $\{A\}$. Rotate $\{B\}$ first about $\hat{Z}_B$ by an angle α, then about $\hat{Y}_B$ by an angle β, and, finally, about Z_b by an angle γ.

$$ {}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}. $$

$$ {}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, $$

$$ \boxed{\begin{aligned} \beta &= \text{Atan2}\left(\sqrt{r_{31}^2 + r_{32}^2}, r_{33}\right) \\ \alpha &= \text{Atan2}(r_{23}/s\beta, r_{13}/s\beta) \\ \gamma &= \text{Atan2}(r_{32}/s\beta, -r_{31}/s\beta) \end{aligned}} $$

As a conclusion for other angle-sets for specifying orientation……

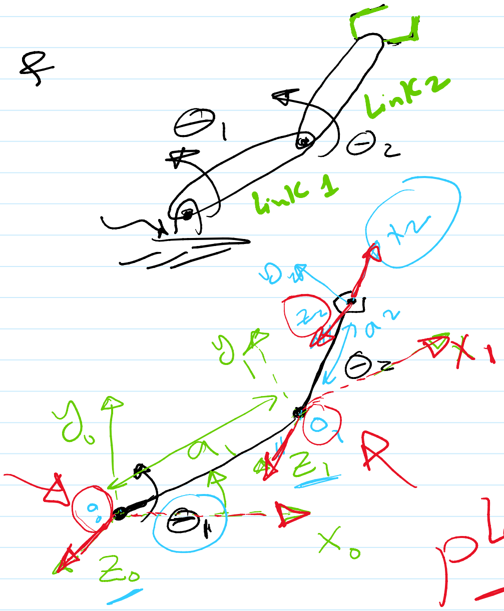


Other angle-set conventions

In the preceding subsections we have seen three conventions for specifying orientation: X–Y–Z fixed angles, Z–Y–X Euler angles, and Z–Y–Z Euler angles. Each of these conventions requires performing three rotations about principal axes in a certain order. These conventions are examples of a set of 24 conventions that we will call **angle-set conventions**. Of these, 12 conventions are for fixed-angle sets, and 12 are for Euler-angle sets. Note that, because of the duality of fixed-angle sets with Euler-angle sets, there are really only 12 unique parameterizations of a rotation matrix by using successive rotations about principal axes. There is often no particular reason to favor one convention over another, but various authors adopt different ones, so it is useful to list the equivalent rotation matrices for all 24 conventions. Appendix B (in the back of the book) gives the equivalent rotation matrices for all 24 conventions.

Ex)

{two moving links & one fixed



planar

DH1 & DH2 { $x_1 \perp z_0$
& x_1 intersects z_0 }

{ $x_2 \perp z_1$
x_2 intersects z_1 }

⇒ O_0 & O_1 could be in different planes

But try to simplify!

⇒ { if O_0 & O_1 in different planes,
we will have $d_1 \neq 0$. }

⇒ for frame 2, $x_2 y_2 z_2$ can be assigned somewhere else not @ the end effector which could be any point on link 2. or even could be @ joint 2

O_1 & O_2 are the same

⇒ { Always assign frames in a way to simplify the process and having zero DH parameters } what to do after assigning the frames ?!

⇒ DH-table

	Link	θ_i	d_i	α_i	a_i
${}^0T_1$	1	θ_1	0	0	a_1
${}^1T_2$	2	θ_2	0	0	a_2

relation between frames 0 & 1

relation between frames 1 & 2

What's next ?!

$$\Rightarrow \underbrace{{}^0T_1 \ \& \ {}^1T_2}_{\text{from the table}}$$

In general;

$$ {}^{i-1}T_i = \left[\begin{array}{ccc|c} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ \hline 0 & 0 & 0 & 1 \end{array}\right] $$

∴ subs. for θ, α, d, a from DH-table:

$$T_1^0 = \left[\begin{array}{ccc|c} C\theta_1 & -S\theta_1 & 0 & a_1 C\theta_1 \\ S\theta_1 & C\theta_1 & 0 & a_1 S\theta_1 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

rotation around z-axis by θ_1

& $$T_2^1 = \begin{bmatrix} C\theta_2 & -S\theta_2 & 0 & a_2 C\theta_2 \\ S\theta_2 & C\theta_2 & 0 & a_2 S\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

again, rotation around z-axis by θ_2

∴ $$T_2^0 = T_1^0 \, T_2^1$$

$$\Rightarrow T_2^0 = \left[\begin{array}{ccc|c} C_1C_2 - S_1S_2 & -C_1S_2 - S_1C_2 & 0 & a_2C_1C_2 - a_2S_1S_2 + a_1C_1 \\ S_1C_2 + C_1S_2 & -S_1S_2 + C_1C_2 & 0 & a_2S_1C_2 + a_2C_1S_2 + a_1S_1 \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

But,
$$\begin{cases} C_{12} = \overset{C_{12}}{C(\theta_1 \pm \theta_2)} = \overset{C_1C_2}{C\theta_1 C\theta_2} \mp \overset{S_1S_2}{S\theta_1 S\theta_2} \\ S_{12} = S(\theta_1 \pm \theta_2) = C\theta_1 S\theta_2 \pm C\theta_2 S\theta_1 \end{cases}$$

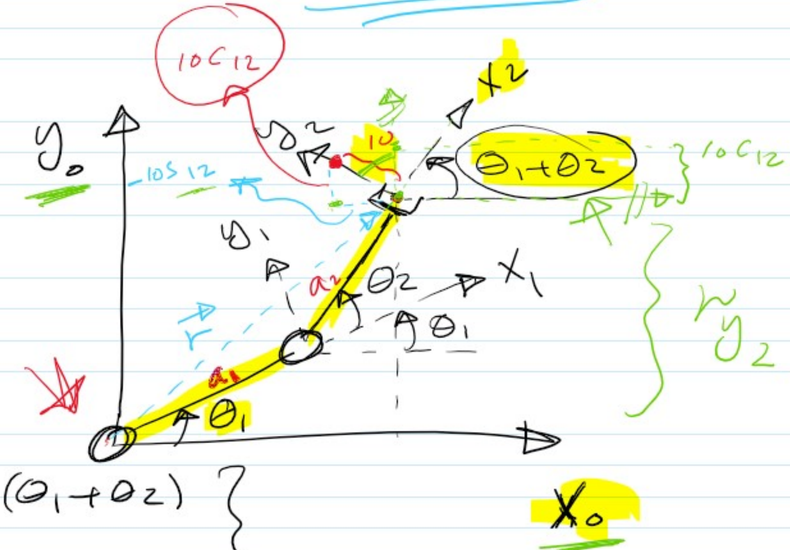
so $${}^{0}T_{2} = \left[\begin{array}{ccc|c} C(\theta_1+\theta_2) & -S(\theta_1+\theta_2) & 0 & \overset{r_x}{a_2 C_{12} + a_1 C_1} \\ S(\theta_1+\theta_2) & C(\theta_1+\theta_2) & 0 & \underset{r_y}{a_2 S_{12} + a_1 S_1} \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

rotation around z_0-axis base frame by $(\theta_1+\theta_2)$

translation

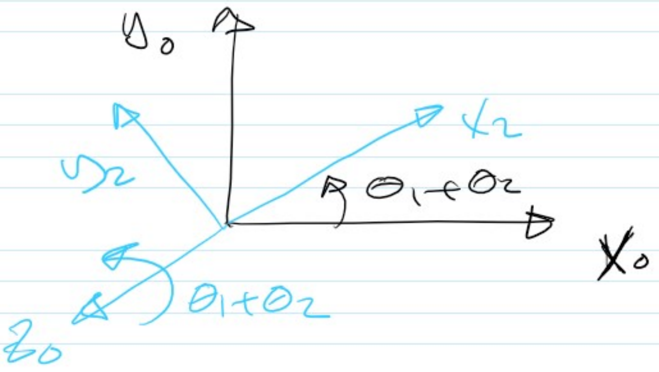
using Geometry ;

$$\vec{r} = \begin{bmatrix} r_x \\ r_y \\ 0 \end{bmatrix}$$



$$\left.\begin{aligned} r_x &= a_1 \cos\theta_1 + a_2 \cos(\theta_1+\theta_2) \\ r_y &= a_1 \sin\theta_1 + a_2 \sin(\theta_1+\theta_2) \end{aligned}\right\}$$

$R(z_0, \theta_1+\theta_2)$



$$R(z_0, \theta_1+\theta_2) = \begin{bmatrix} c_{12} & -s_{12} & 0 \\ s_{12} & c_{12} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

from Geometry ;

$$ {}^{0}T_{2} = \left[\begin{array}{ccc|c} & R(z_0, \theta_1+\theta_2) & & r_x \\ & & & r_y \\ & & & 0 \\ \hline 0 & 0 & 0 & \end{array}\right]$$

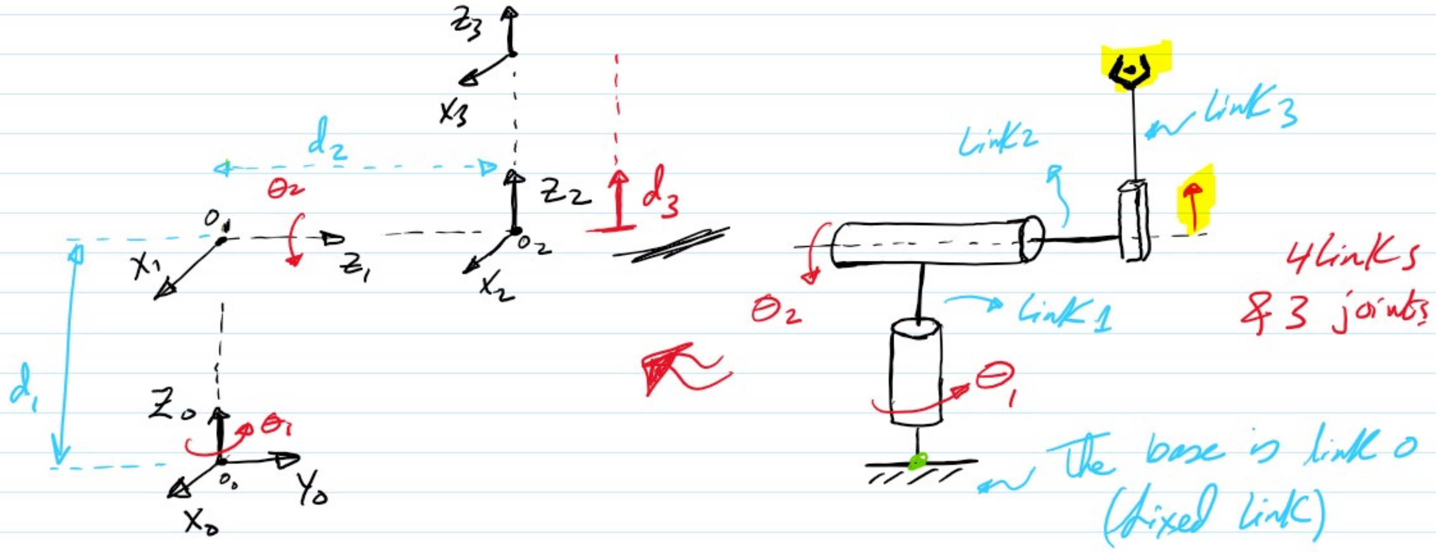
$$P_{x_0 y_0 z_0} = {}^{0}T_{2} \, \boxed{P_{x_2 y_2 z_2}}$$

$$= \boxed{{}^{0}T_{2}} \begin{bmatrix} 0 \\ 10 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \boxed{-10 s_{12}} + a_1 c_1 + a_2 c_{12} \\ 10 c_{12} + a_1 s_1 + a_2 s_{12} \\ 0 \\ 1 \end{bmatrix}$$

$\Downarrow$

$\underline{4 \times 4}$

EX.) "Spherical Robot (RRP)"



$\begin{cases} d_1: \text{distance from } X_0 \text{ to } X_1 \text{ along } Z_0 = d_1 \\ d_2: \quad " \quad " \quad X_1 \quad " \quad X_2 \quad " \quad Z_1 = d_2 \\ d_3: \quad " \quad " \quad X_2 \quad " \quad X_3 \quad " \quad Z_2 = d_3 \end{cases}$

$\begin{cases} a_1: \text{distance from } Z_0 \text{ to } Z_1 \text{ along } X_1 = 0 \\ a_2: \quad " \quad " \quad Z_1 \quad " \quad Z_2 \quad " \quad X_2 = 0 \\ a_3: \quad " \quad " \quad Z_2 \quad " \quad Z_3 \quad " \quad X_3 = 0 \end{cases}$

& $\begin{cases} \alpha_1: \text{angle around } X_1 \text{ to align } Z_0 \text{ with } Z_1 = -90 \\ \alpha_2: \quad " \quad " \quad X_2 \quad " \quad " \quad Z_1 \quad " \quad Z_2 = 90 \\ \alpha_3: \quad " \quad " \quad X_3 \quad " \quad " \quad Z_2 \quad " \quad Z_3 = 0 \end{cases}$

go

${}^0T_1$ ←

${}^1T_2$ ←

${}^2T_3$ ←

Link	a_i	α_i	d_i	θ_i
1	0	-90	d_1	θ_1
2	0	90	d_2	θ_2
3	0	0	d_3	0

→ d_3 is changing not fixed

go $${}^0T_3 = {}^0T_1 \, {}^1T_2 \, {}^2T_3$$

Note When all X's (X_0, X_2, X_2 & X_3) are parallel, this means the initial values of the manipulator joints are zeros.

& This is called zero-position manipulator

⇒ All Industrial Robots has zero-positions.

Ex.

{ d_1: distance from X_0 to X_1 along $Z_0 = d_1$

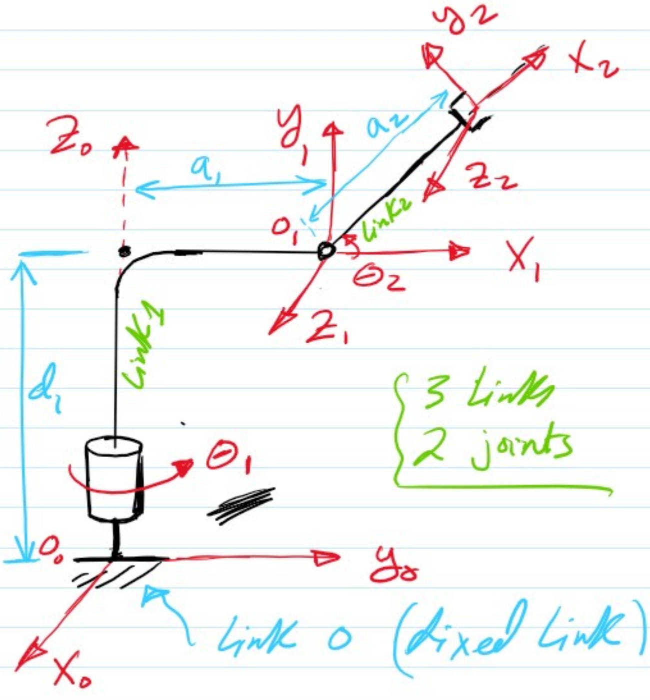
d_2: " " X_1 " X_2 " $Z_1 = 0$

{ a_1: distance from Z_0 to Z_1 along $X_1 = a_1$

a_2: " " Z_1 " Z_2 " $X_2 = a_2$

{ α_1: angle around X_1 to align Z_0 to $Z_1 = 90$

α_2: " " X_2 " " Z_1 " $Z_2 = 0$



θ_1 : 〃 〃 $Z_0 = \theta_1$

θ_2 : 〃 〃 $Z_1 = \theta_2$

	Link	a_i	α_i	d_i	θ_i
${}^0T_1 \rightarrow$	1	a_1	90	d_1	θ_1
${}^1T_2 \rightarrow$	2	a_2	0	0	θ_2

θ's are variable & d_1 is fixed

$${}^0T_2 = {}^0T_1 \; {}^1T_2$$

Ex.1

Three-Link Planar Arm

Consider the arm in Fig. 2.17, where the base frame and the link frames have been illustrated. Since the revolute axes are all parallel, the simplest choice was made for all axes x_i along the direction of the relative links (the direction of x_0 is arbitrary) and all lying in the plane (x_0, y_0). In this way, all the parameters d_i are null and the angles between the axes x_i directly provide the joint variables. The Denavit-Hartenberg parameters are specified in the table below:

Link	a_i	α_i	d_i	ϑ_i
1	a_1	0	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

$${}^{0}T_{3} = {}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}$$

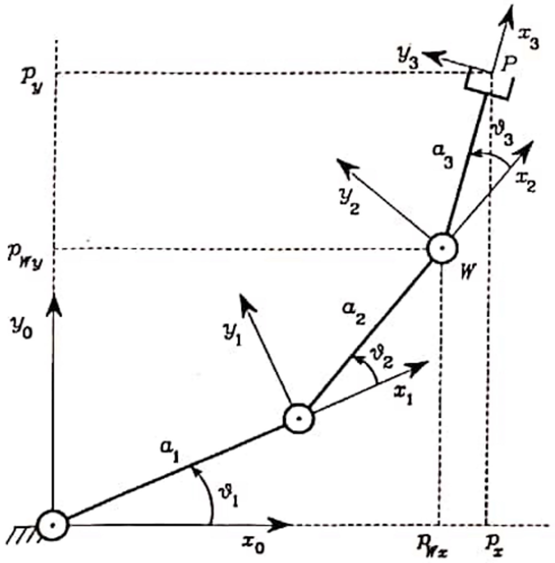


FIGURE 2.17
Three-link planar arm.

Robotics 110405442

Chapter 3

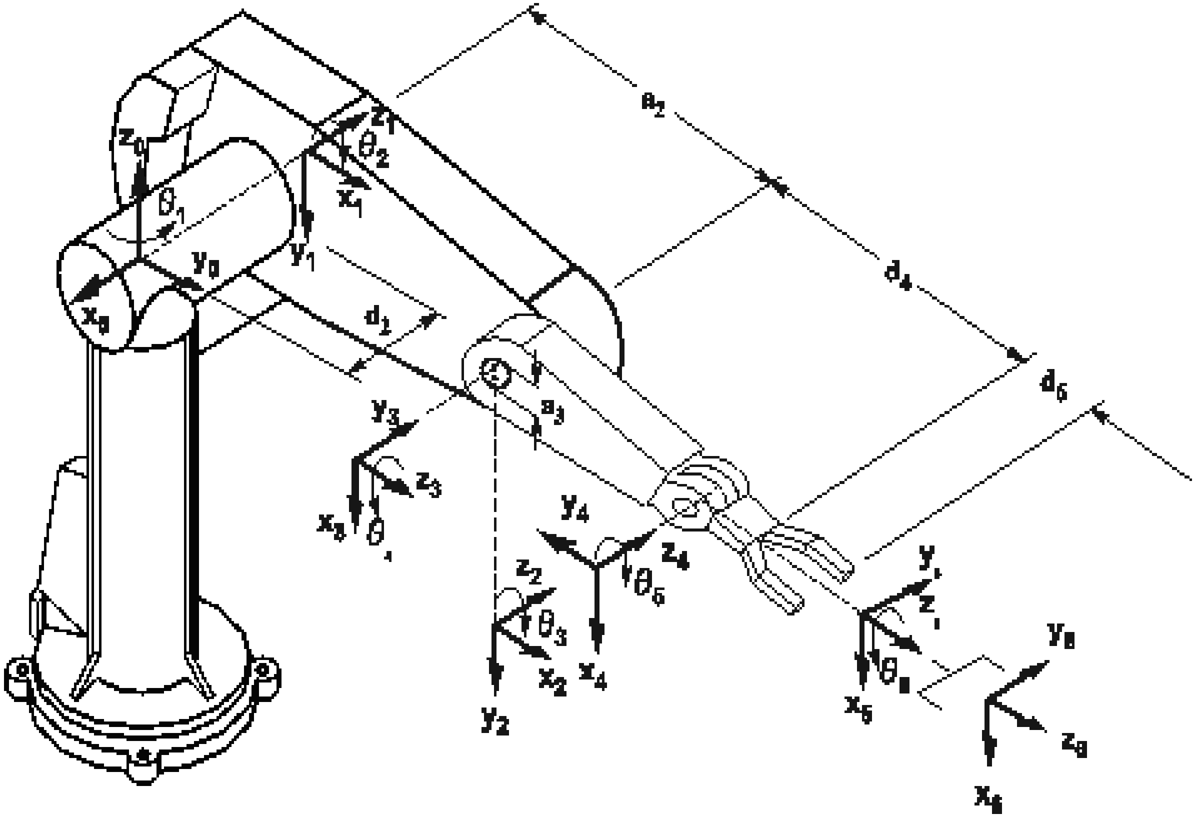
Manipulator Forward kinematics: Denavit-Hartenberg (DH) Convention

Textbook:

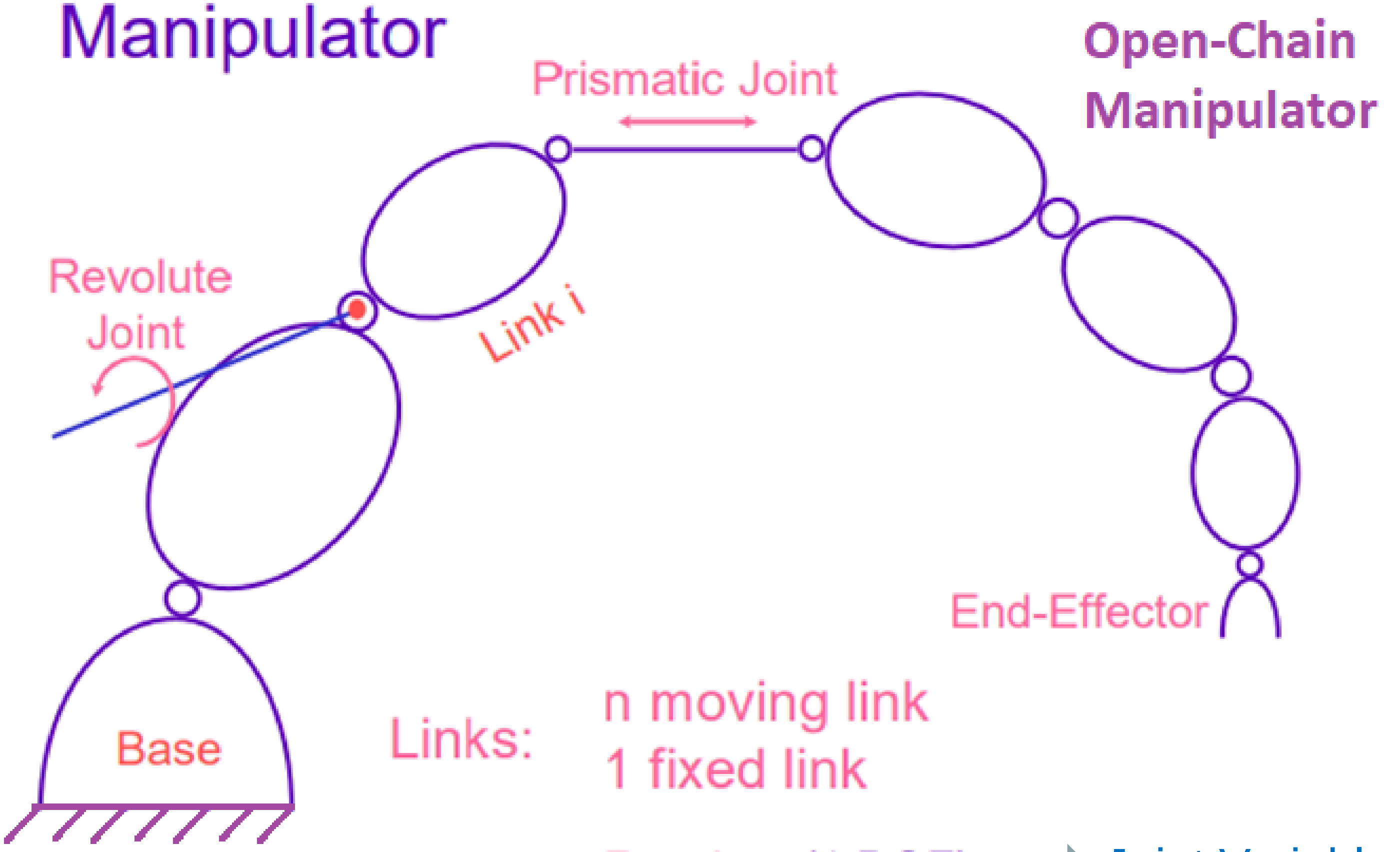
Robert J. Schilling, “Fundamentals of Robotics: Analysis & Control”, Pearson Education Com., 1990.

Ahmad AL-Jarrah

Forward Kinematics: means calculating the position and orientation of the end effector given all the joint variables.



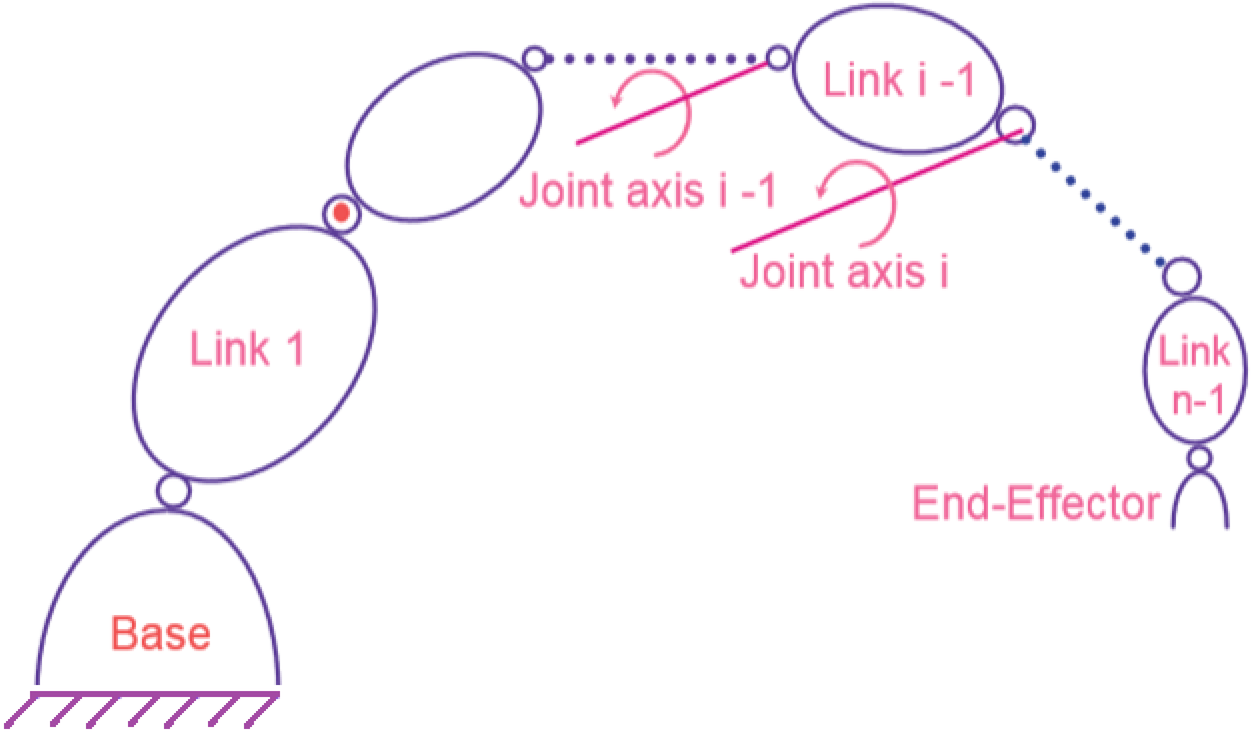
Manipulator



Links: n moving link
1 fixed link

Joints:
Revolute (1 DOF) ⟹ Joint Variable is θ_i
Prismatic (1 DOF) ⟹ Joint Variable is d_i

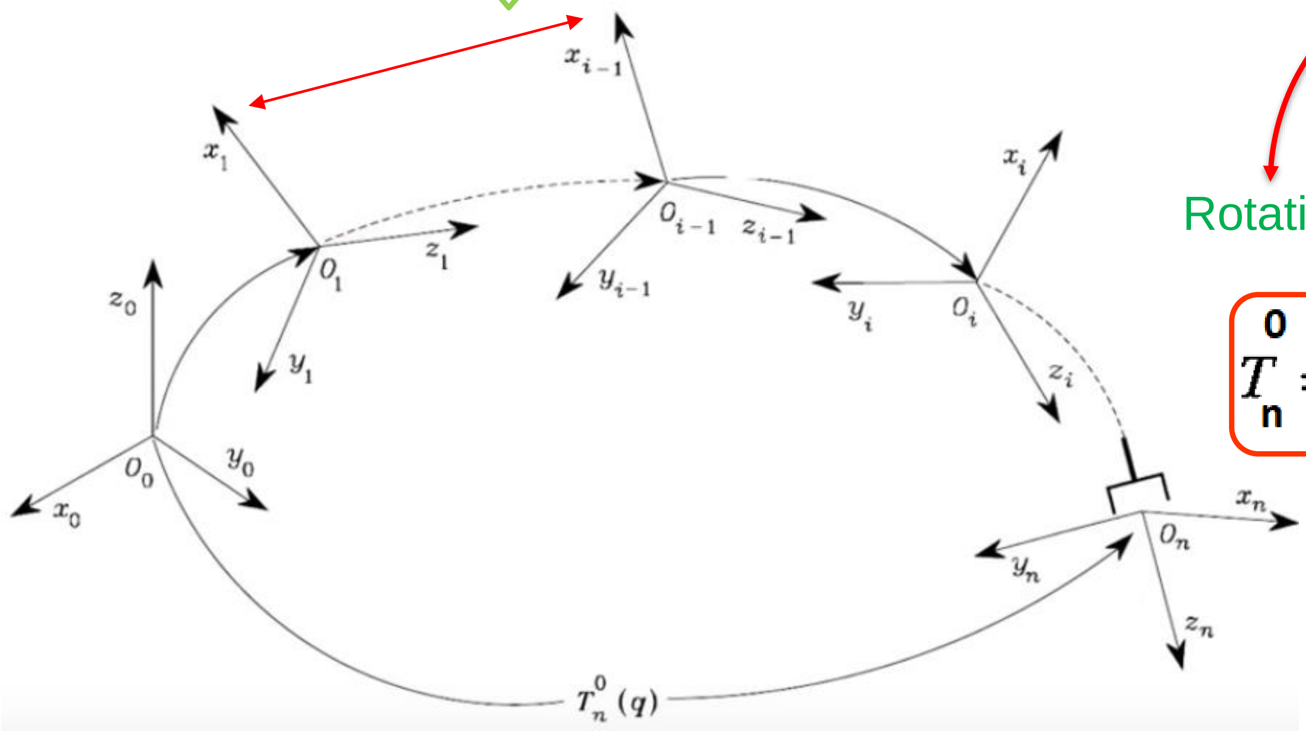
Manipulator



How to describe Forward Kinematics?

⇨ Homogeneous Transformation Matrix, T

There is rotation and transformation between each adjacent frames (i.e. T)



$${}^A_BT$$

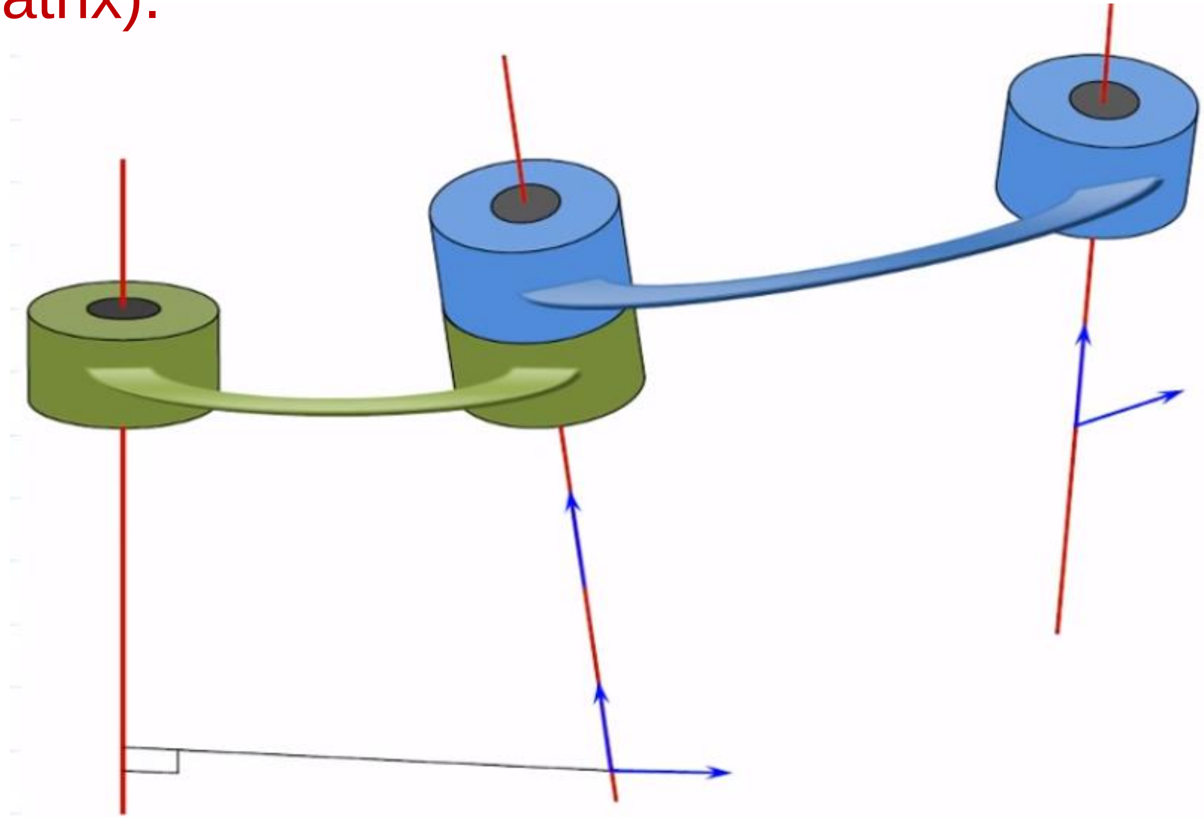
$$\left[\begin{array}{ccc|c} \multicolumn{3}{c|}{\boxed{{}^A_BR}} & \boxed{{}^AP_{BORG}} \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$

Rotation

Translation

$${}^0T_n = {}^0T_1\;{}^1T_2\;{}^2T_3 \dots {}^{n-1}T_n$$

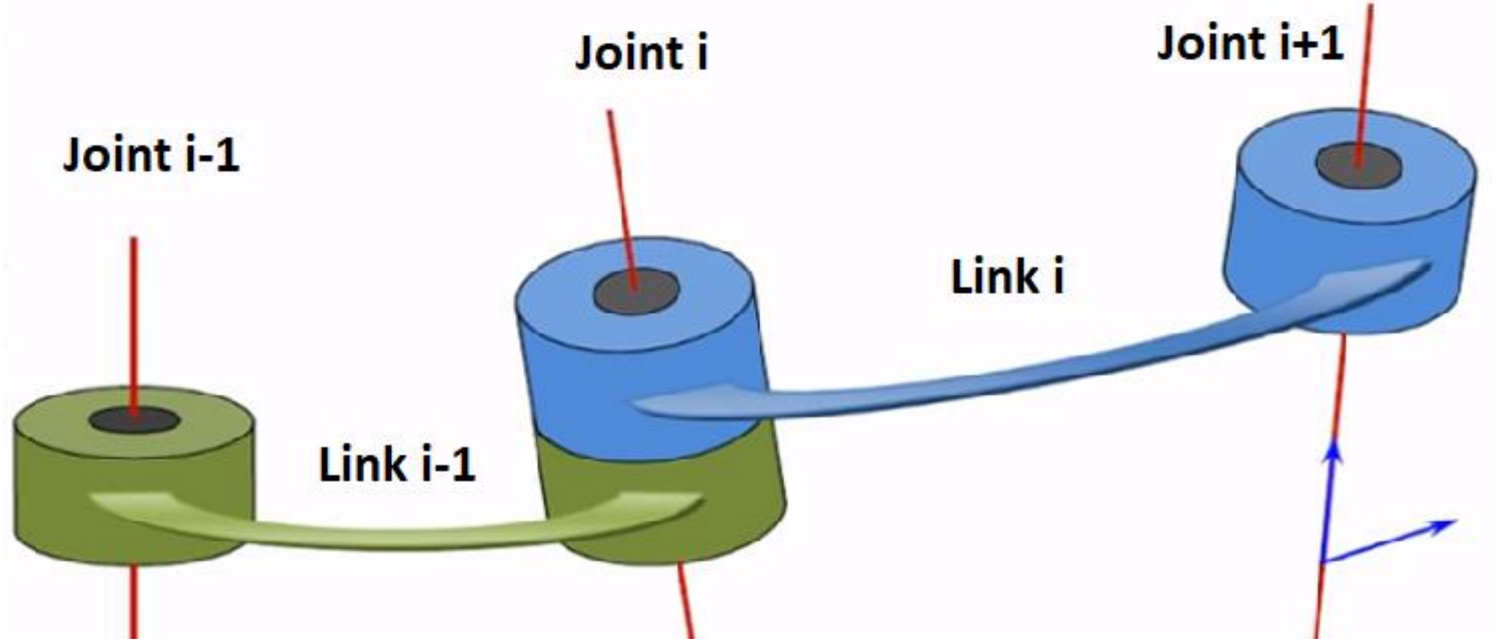
Since each joint connects only two consecutive links, it is reasonable to consider first the description of kinematic relationship between adjacent links and then obtain the overall description of the manipulator kinematics (i.e. the total Homogeneous transformation matrix).



How to find the kinematic description between two adjacent links?

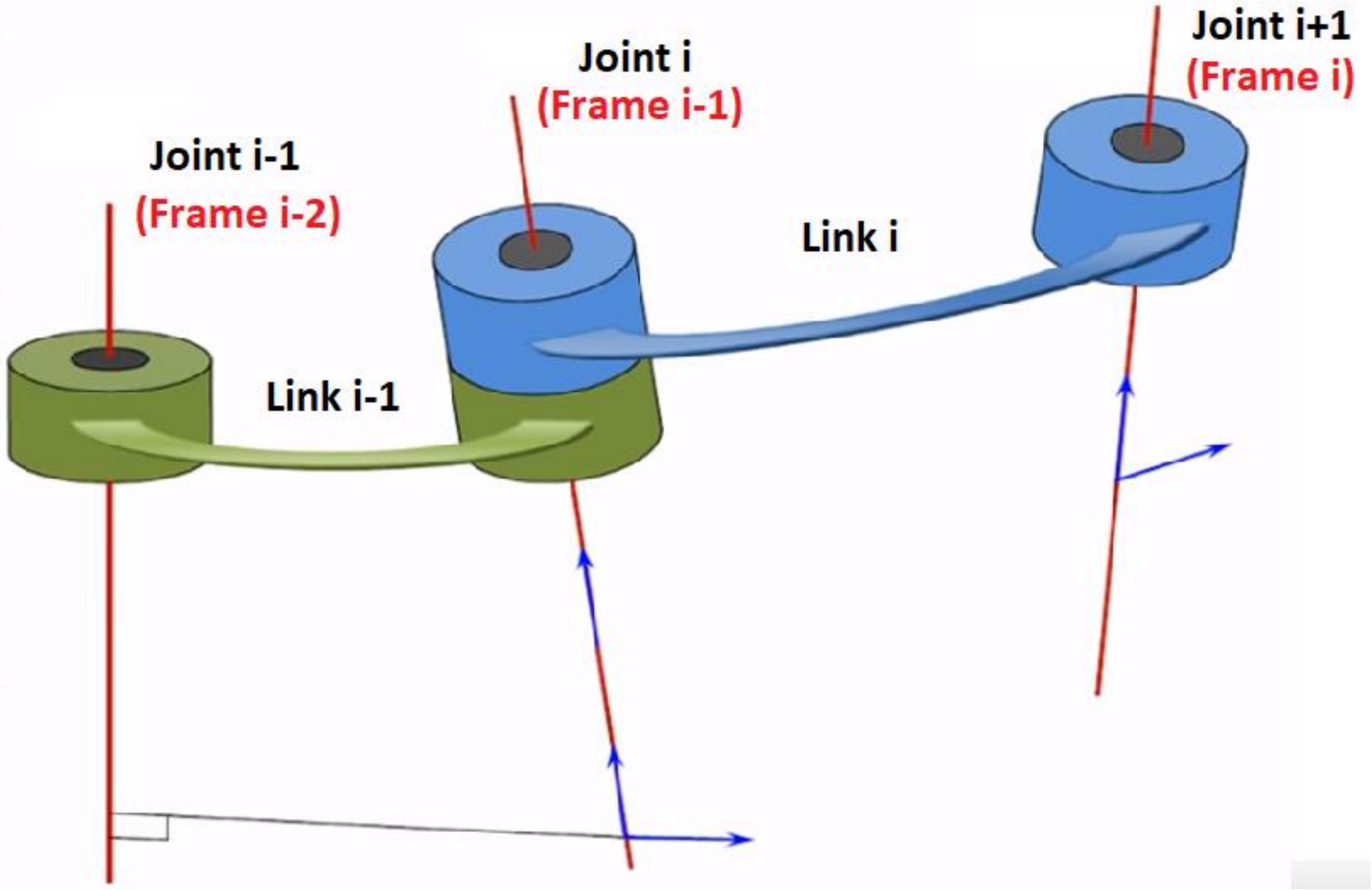
⇨ By assigning frames to each link

Frame *i* is rigidly attached to link *i* at joint *i+1*



How the link moves?

⇨ If a joint moves, which link will move?!


Joint i-1
(Frame i-2)
Joint i
(Frame i-1)
Joint i+1
(Frame i)
Link i-1
Link i

Frames are assigned using Danevit-Hartenberg (DH) Convention as follows :

⇨ First, assign z-axis as the axis of motion

⇨ The motion always about/along the z-axis whether its rotation about z-axis or translation along z-axis

Revolute Joint

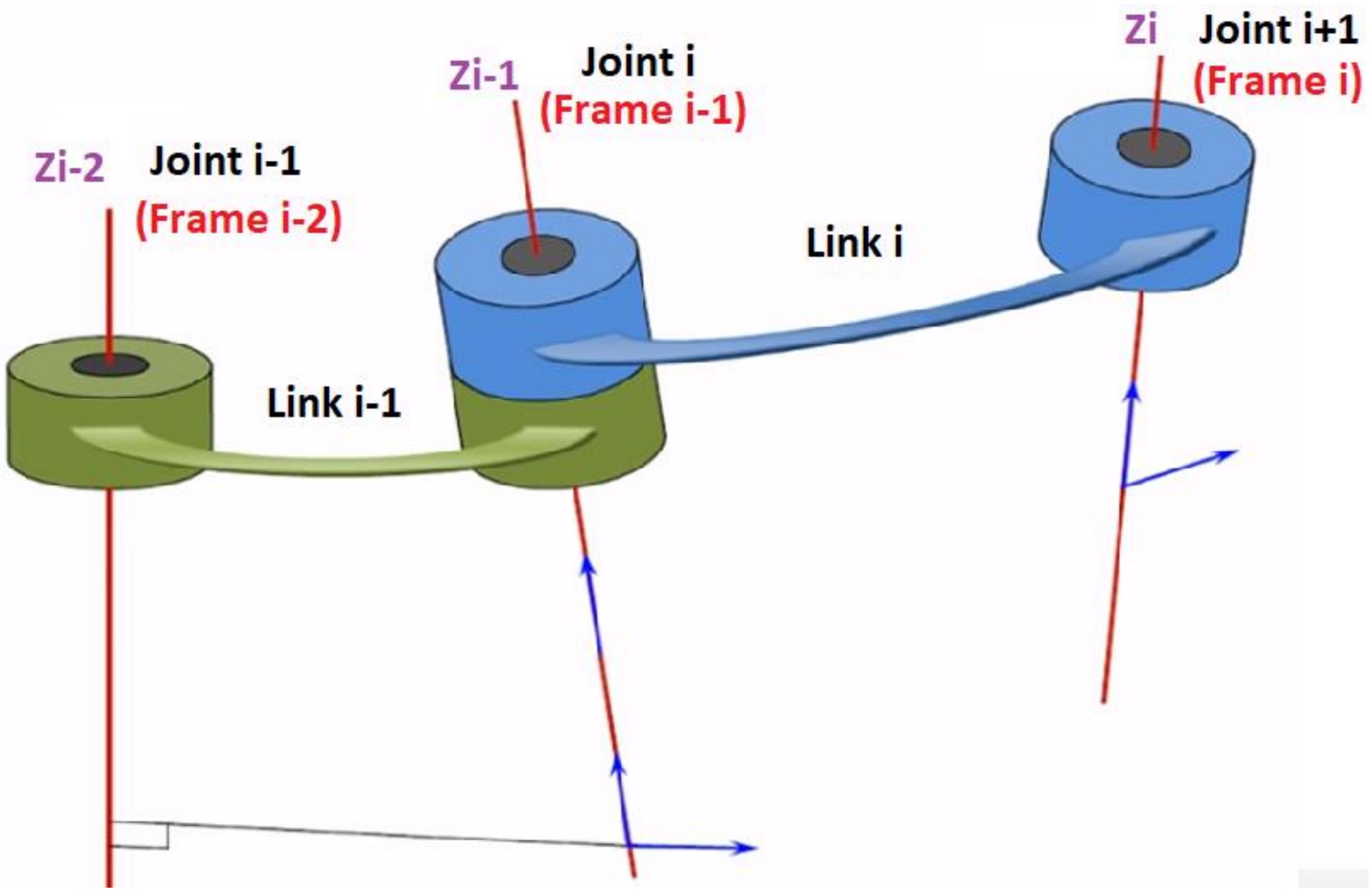
Joint Variable is θ_i

Prismatic Joint

Joint Variable is d_i

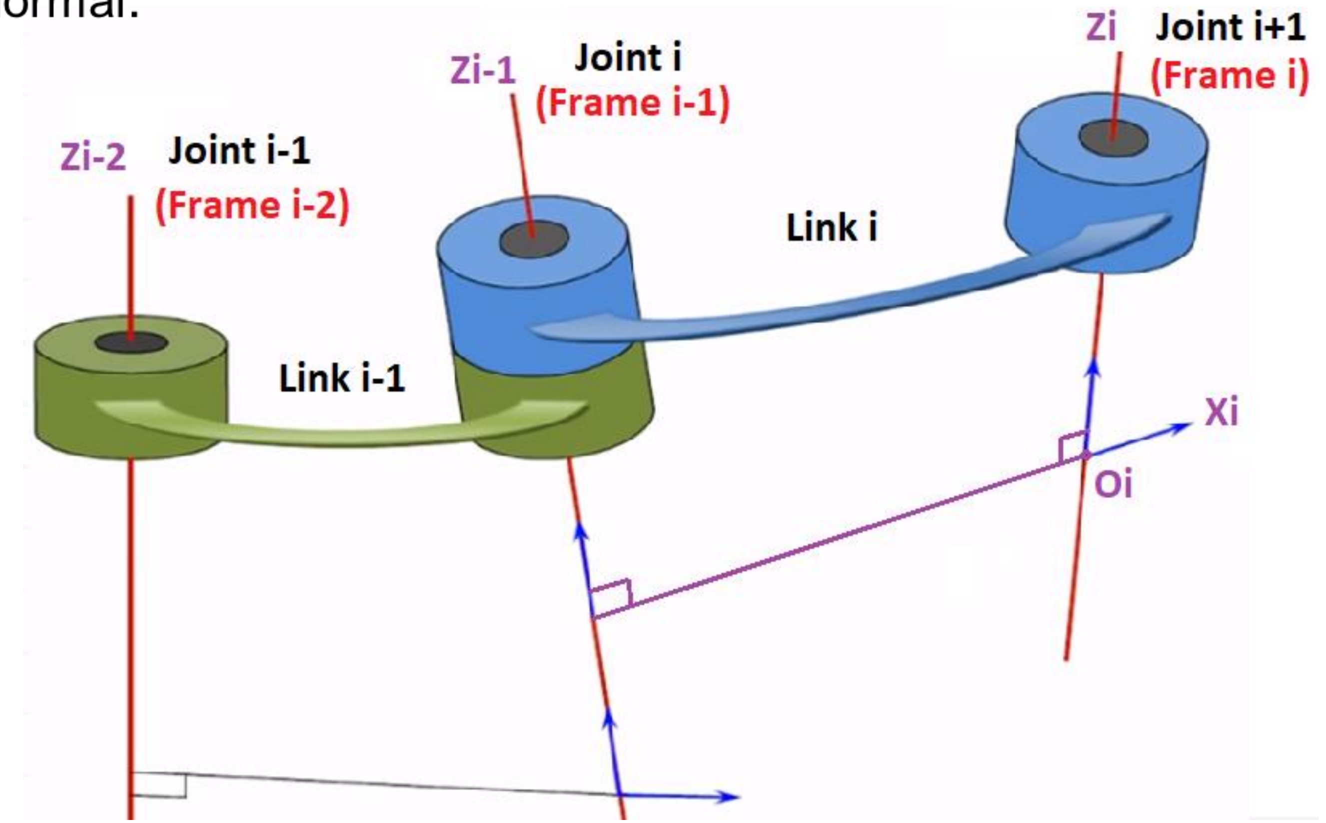
DH Constraints

- (DH1): The axis x_i is perpendicular to the axis z_{i-1}.
- (DH2): The axis x_i intersects the axis z_{i-1}.

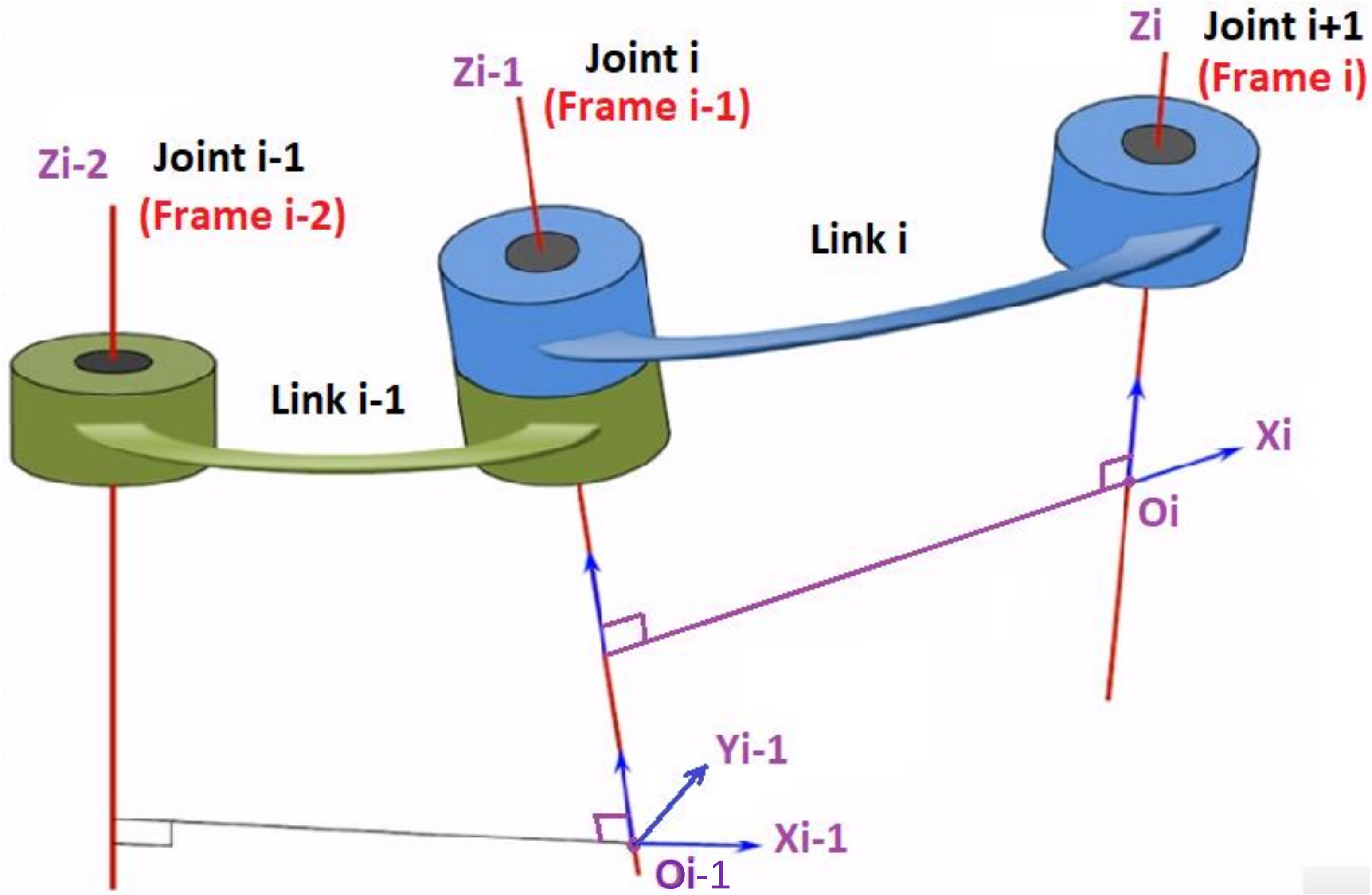


⇨ Then, assign the x-axis !! How?

- In order to assign x_i , find the common normal between z_i and z_{i-1}
- If z_i and z_{i-1} do not intersect and not parallel, then there is a unique common normal between them.
- x_i is along the common normal line and must fulfill DH1 and DH2. Then we locate the origin o_i at the intersection between z_i and the common normal.



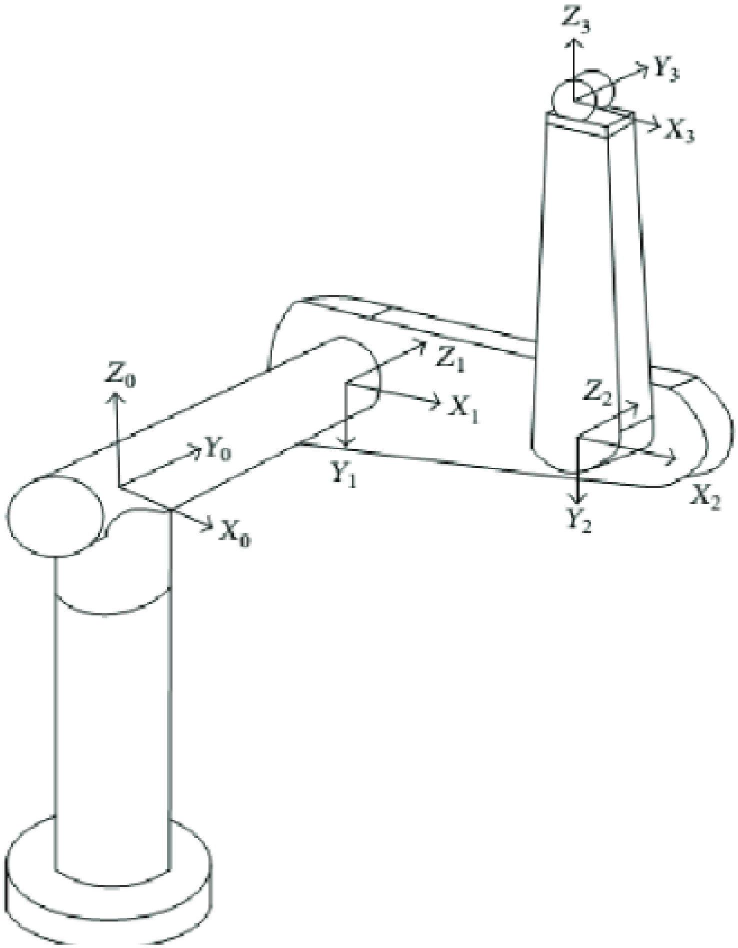
- Same way you determine for x_{i-1} based on the common normal between z_{i-1} and z_{i-2} .
- Then, you determine the Y-axis based on right hand rule.

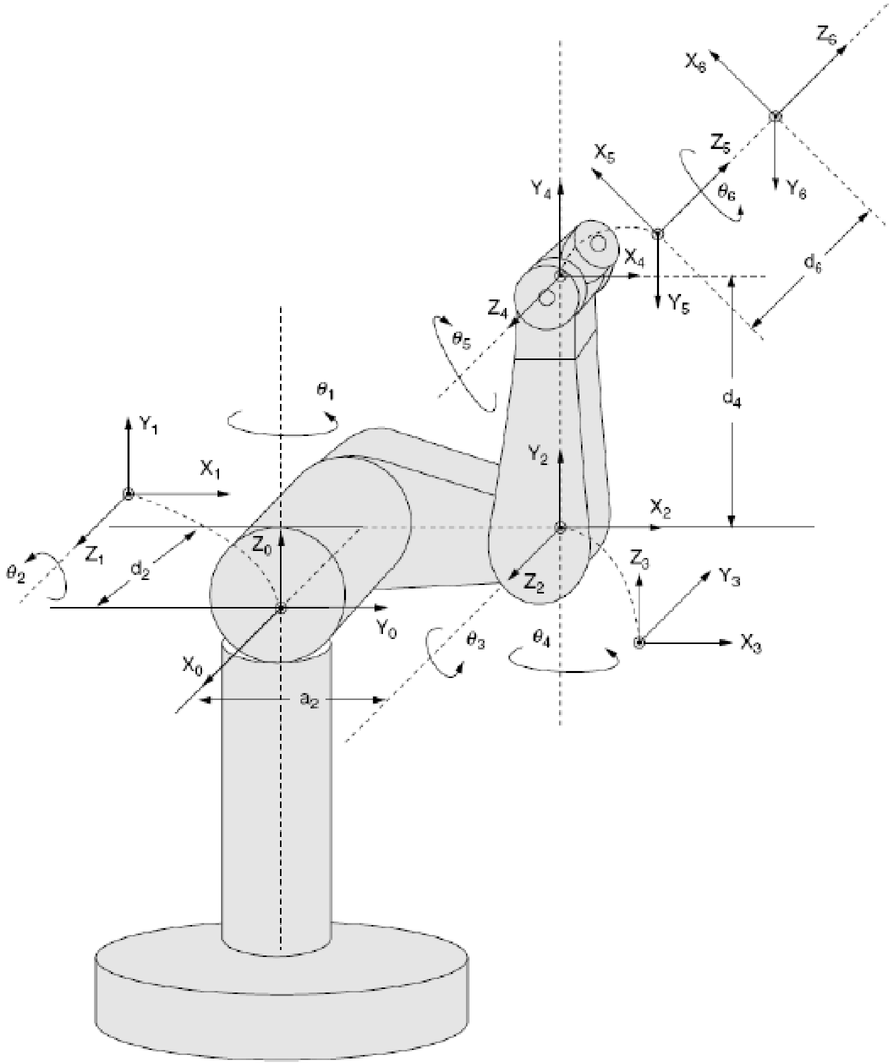


- If z_i and z_{i-1} are parallel and do not intersect, then there is a unlimited common normal lines between them.
  - In this case, you pick any common normal line and then locate the origin o_i along z_i then the intersection will be x_i along the common normal.
- For frame 0, only the direction of z-axis is specified, then o_0 and x_0 can be arbitrarily chosen.
- When joint i is prismatic, only the direction of z-axis is determined.
- If z_i and z_{i-1} intersect, then x_i axis can be assigned arbitrarily based on DH constraints. One way could be assigned using the following:

$$X_i = \pm Z_{i-1} \times Z_i$$

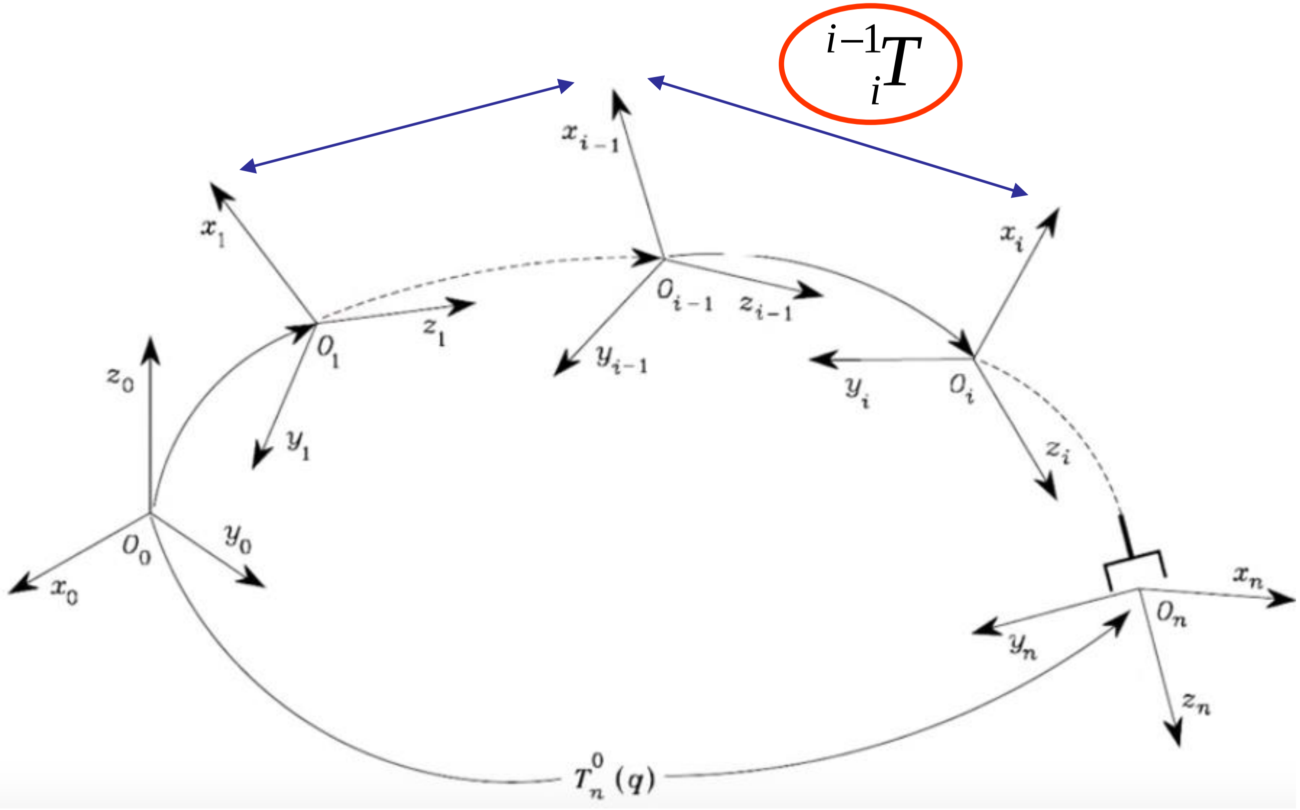
As a result, the origin Oi is the intersection of Zi and Zi-1

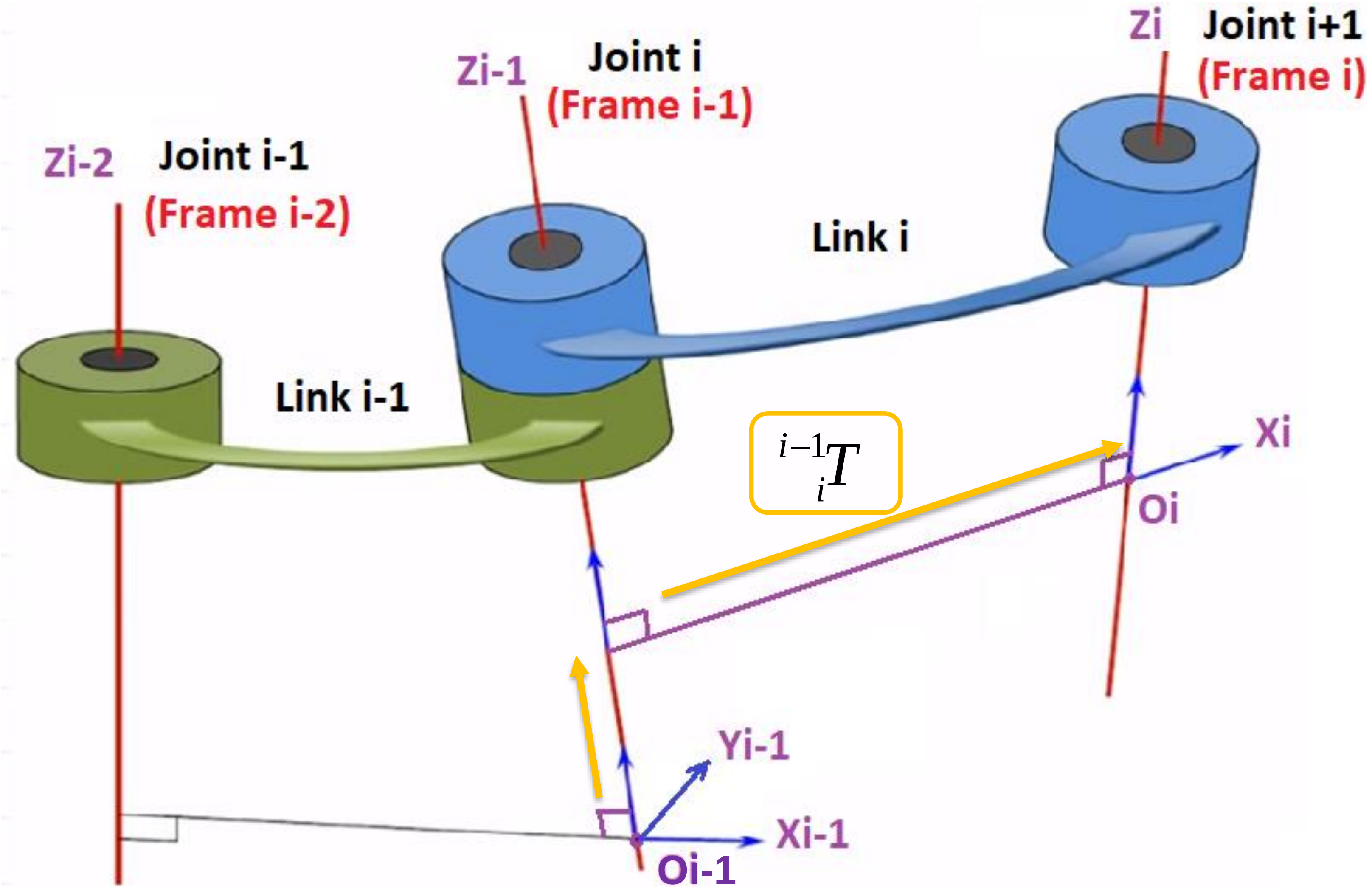

Z_3
Y_3
X_3
Z_0
Z_1
X_1
Z_2
Y_0
Y_1
X_0
Y_2
X_2

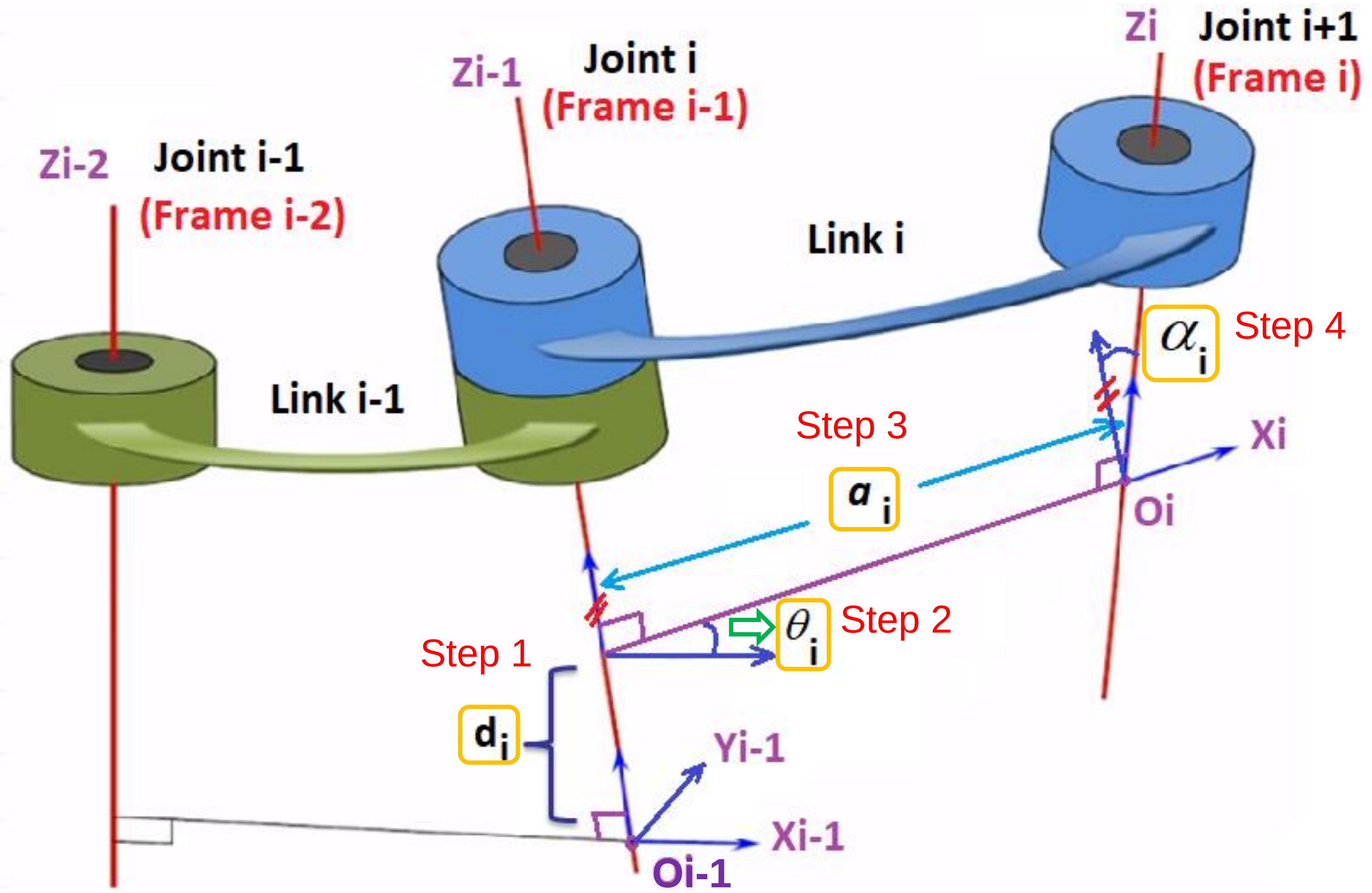

Z_6
X_6
X_5
Z_5
Y_4
θ_6
Y_6
X_4
d_6
Z_4
Y_5
θ_5
θ_1
d_4
Y_1
Y_2
X_1
X_2
Z_0
Z_1
θ_2
d_2
Z_3
Z_2
Y_3
Y_0
θ_3
θ_4
X_3
X_0
a_2

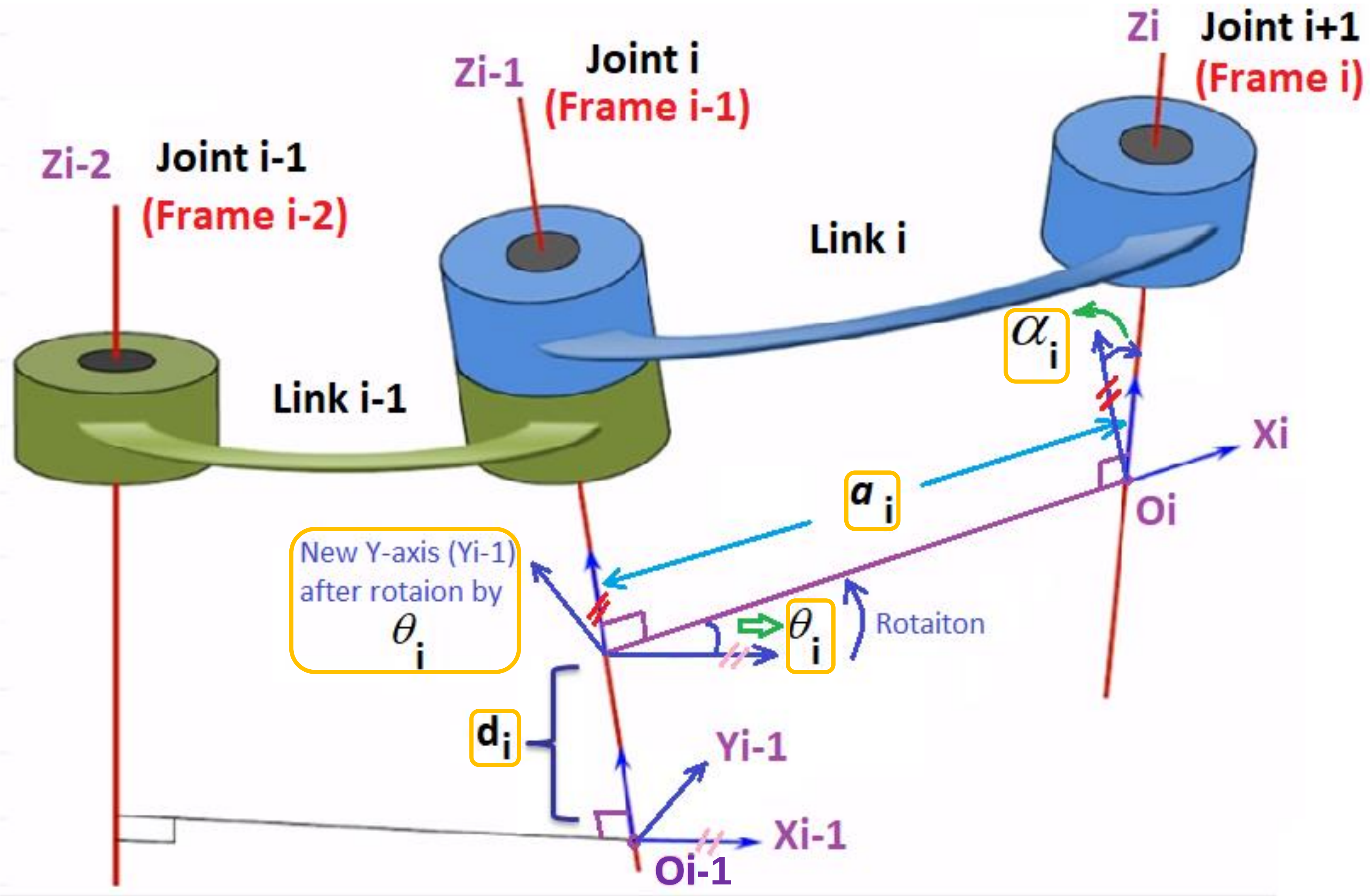
What to do after assigning all frames?

⇨ Looking for the transformation matrices




Zi
Joint i+1
(Frame i)
Zi-1
Joint i
(Frame i-1)
Zi-2
Joint i-1
(Frame i-2)
Link i
Link i-1
$^{i-1}_{i}T$
Xi
Oi
Yi-1
Xi-1
Oi-1


Zi
Joint i+1
(Frame i)
Zi-1
Joint i
(Frame i-1)
Zi-2
Joint i-1
(Frame i-2)
Link i
Link i-1
α_i
Step 4
Step 3
Xi
a_i
Oi
θ_i
Step 2
Step 1
d_i
Yi-1
Xi-1
Oi-1


Zi
Joint i+1
(Frame i)
Zi-1
Joint i
(Frame i-1)
Zi-2
Joint i-1
(Frame i-2)
Link i
Link i-1
α_i
Xi
a_i
Oi
New Y-axis (Yi-1) after rotaion by θ_i
θ_i
Rotaiton
d_i
Yi-1
Xi-1
Oi-1

Steps of finding the transformation matrix ${}^{i-1}_{i}T$ in order to get from frame i-1 to frame i of two adjacent frames (i.e. reading frame i with respect to frame i-1):

1. Translate along axis z_{i-1} by distance d_i
2. Rotate around the current z-axis (i.e. z_{i-1}) by angle θ_i
3. Translate along current x-axis (i.e. x_i) by distance a_i
4. Rotate around the current x-axis (i.e. x_i) by angle α_i

$$\boxed{d_i, \theta_i, a_i, \alpha_i \text{ are called DH parameters}}$$

$$\therefore {}^{i-1}_{i}T = \overbrace{T_z(\mathrm{d}_i)\mathrm{T}_z(\theta_i)\mathrm{T}_x(a_i)\mathrm{T}_x(\alpha_i)}^{\text{Basic Homogeneous transformation matrix with elementary rotations and translations}}$$

Post-Multiplications

$$\xrightarrow{\hspace{6cm}}$$

$$ {}^{i-1}_{i}T = T_z(\mathrm{d}_i)\,\mathrm{T}_z(\theta_i)\,\mathrm{T}_x(a_i)\,\mathrm{T}_x(\alpha_i) $$

$$ T_z(\mathrm{d}_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_z(\theta_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & 0 \\ \sin\theta_i & \cos\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ T_x(a_i) = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_x(\alpha_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{i-1}_{i}T = T_z(\mathrm{d}_i)\,\mathrm{T}_z(\theta_i)\,\mathrm{T}_x(a_i)\,\mathrm{T}_x(\alpha_i) $$

$$ \therefore\ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Summary:

- a_i = distance from Zi-1 to Zi along Xi (+ve if in the direction of Xi)
- α_i = angle from Zi-1 to Zi around Xi (+ve if counterclock wise)
- d_i = distance from Xi-1 to Xi along Zi-1 (+ve if in the direction of Zi-1)
- θ_i = angle from Xi-1 to Xi around Zi-1 (+ve if counterclock wise)

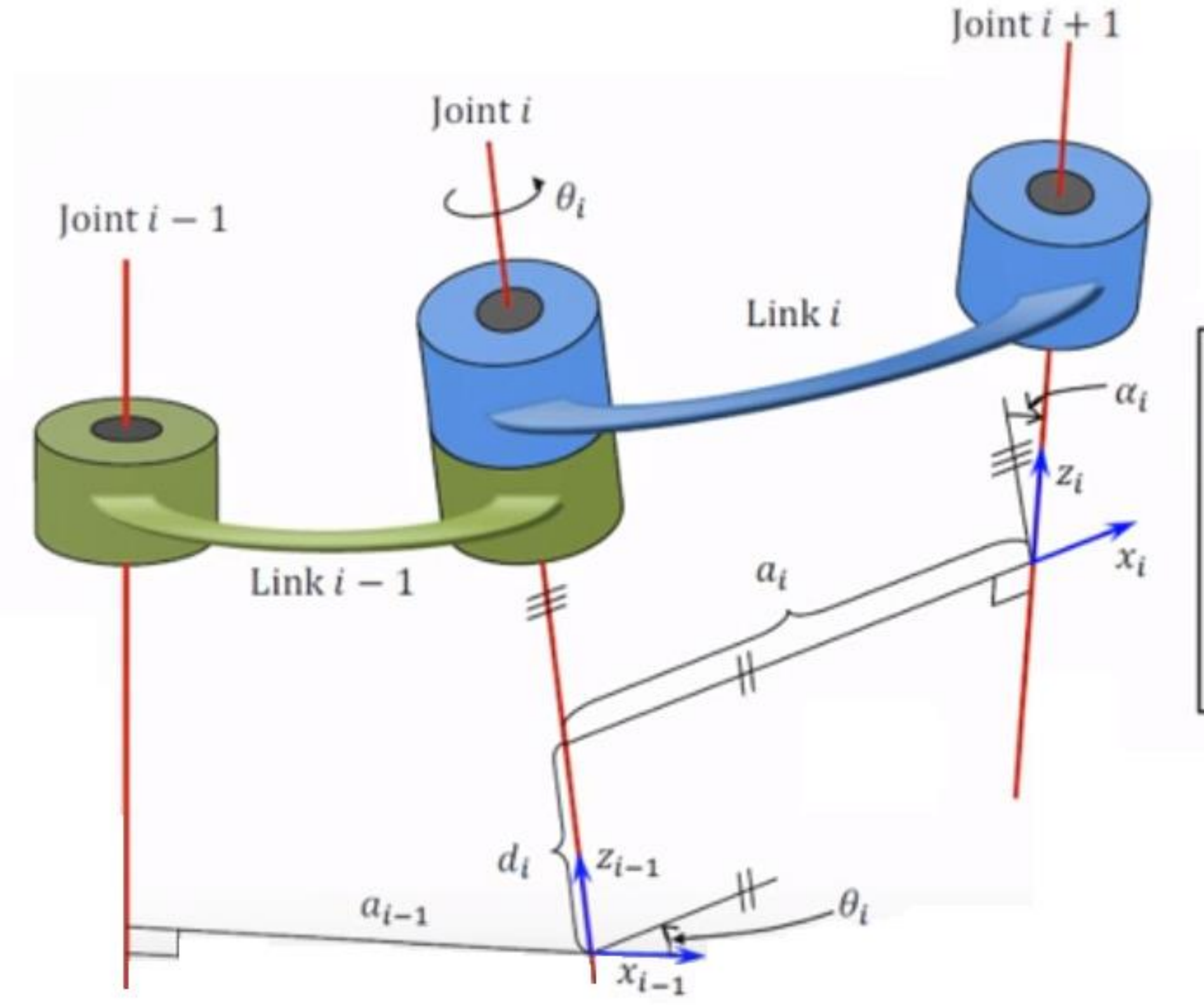
a_i is called link length (usually fixed!)

α_i is called link twist (usually fixed!)

d_i is called link offset

θ_i is called joint offset

- ❖ If the joint is prismatic then d_i is the variable along z-axis.
- ❖ If the joint is revolute then θ_i is the variable around z-axis



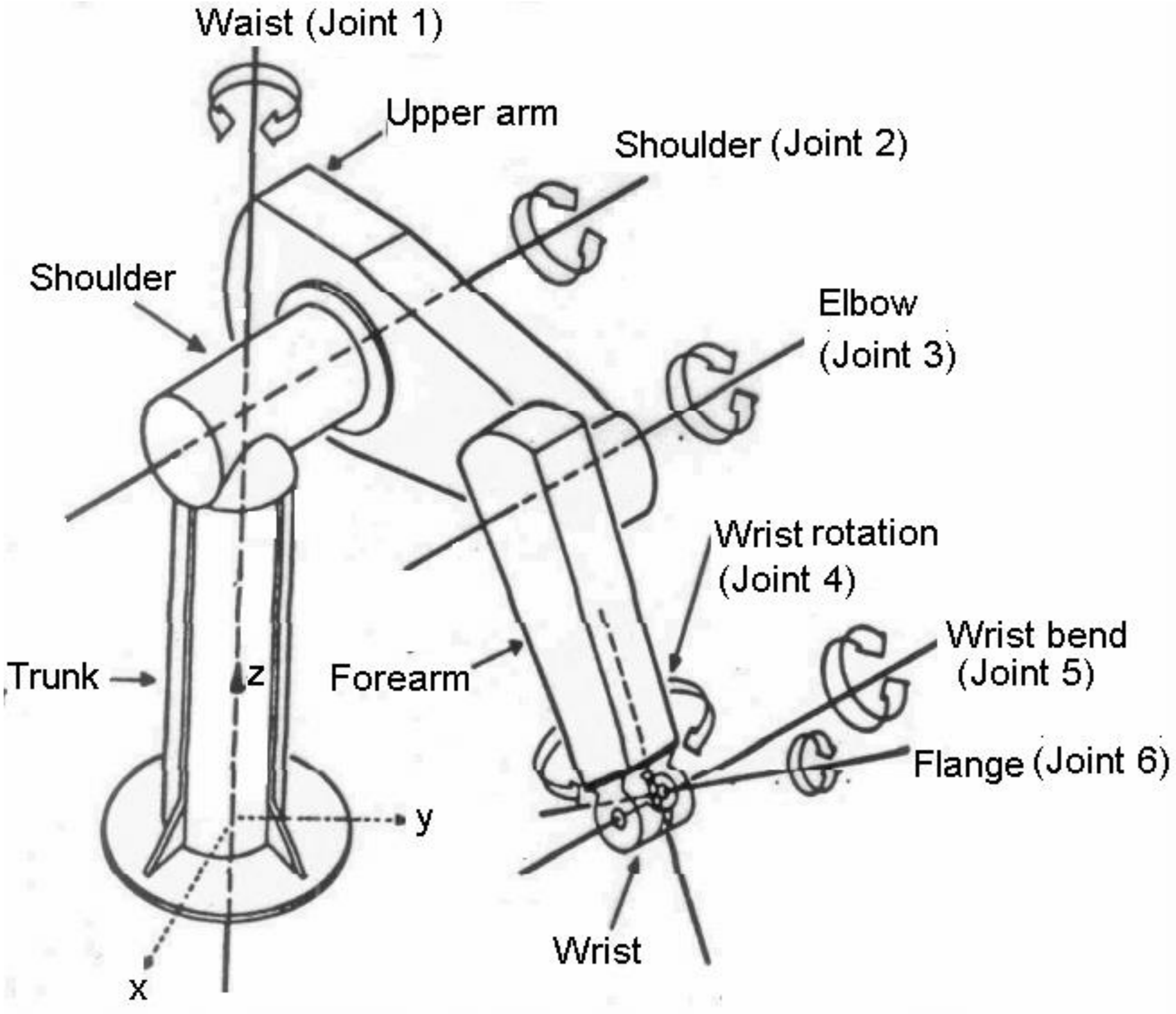
a_i = dist(z_{i-1}, z_i) along x_i

α_i = angle(z_{i-1}, z_i) about x_i

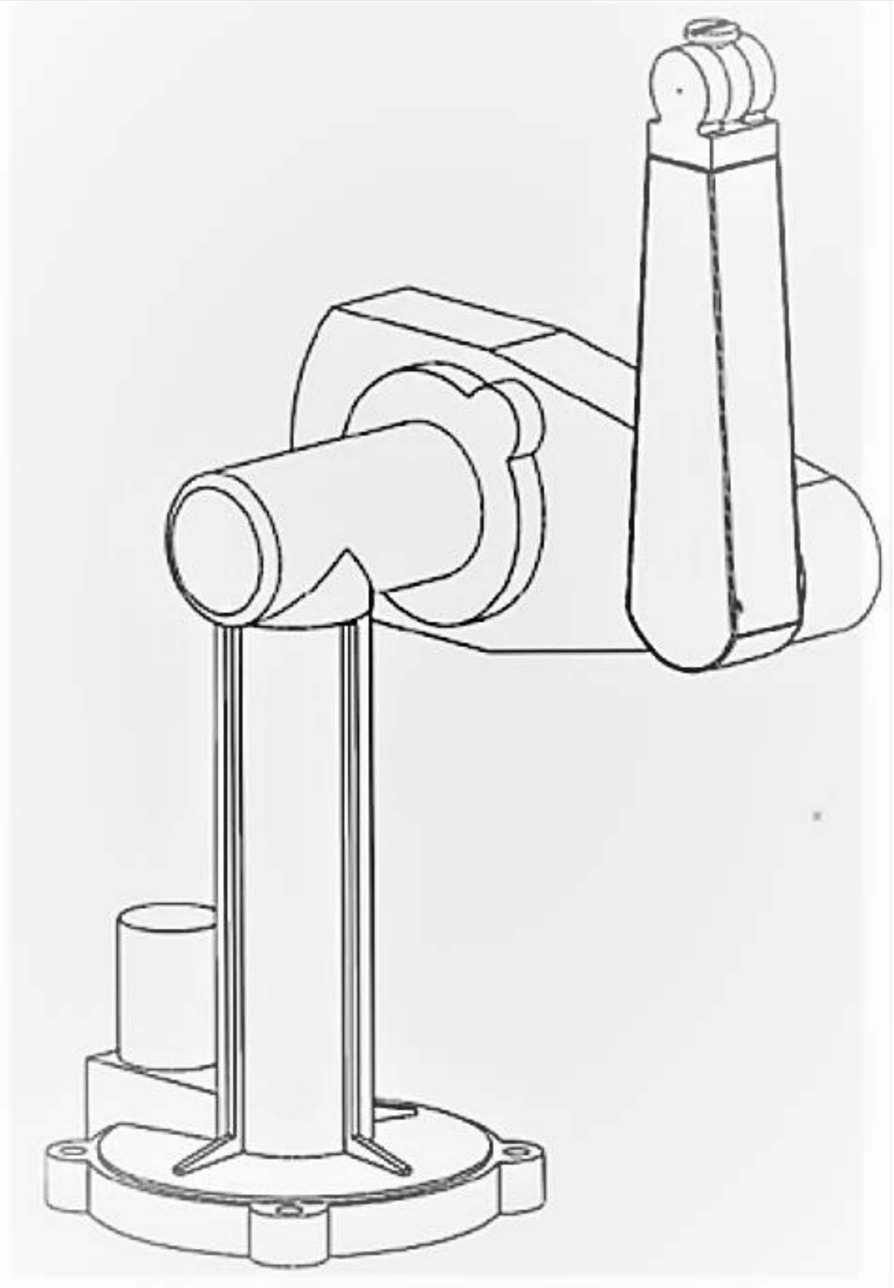
d_i = dist(x_{i-1}, x_i) along z_{i-1}

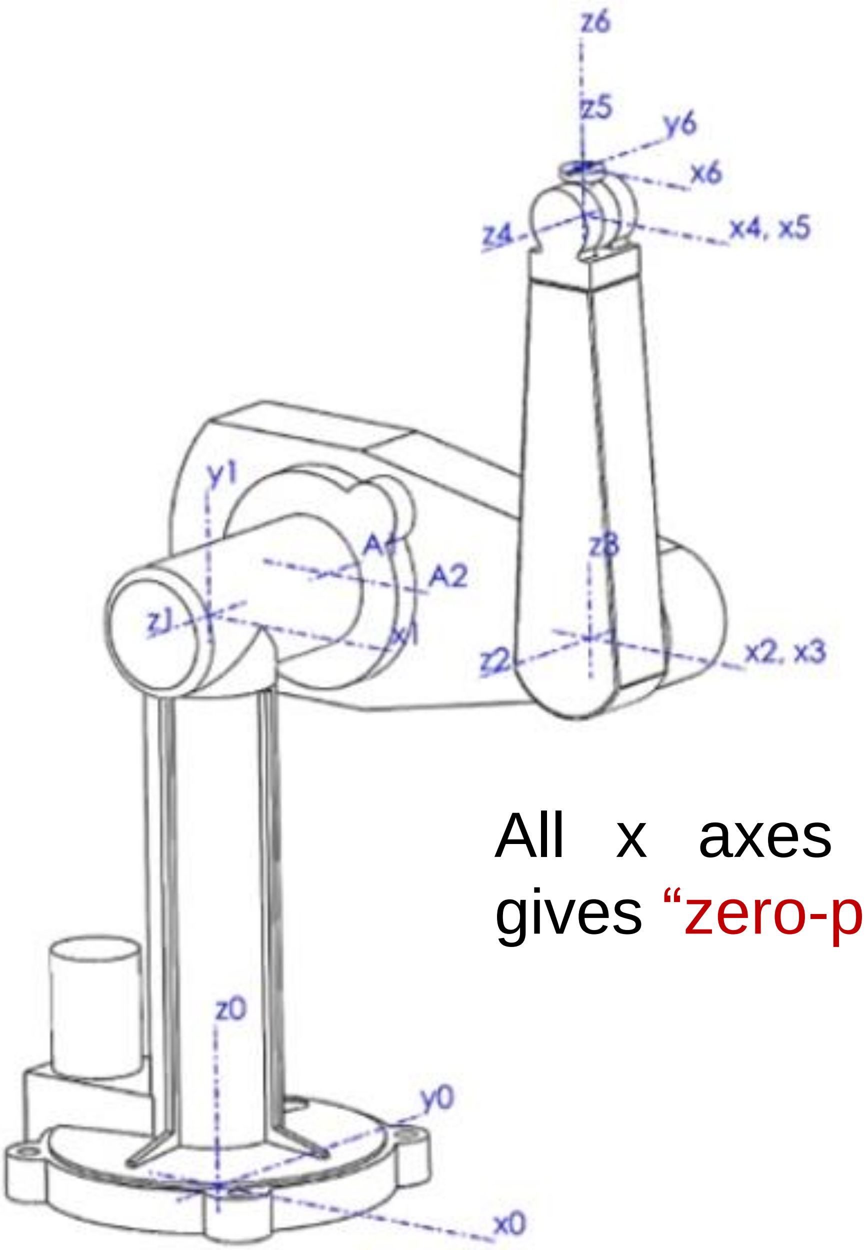
θ_i = angle(x_{i-1}, x_i) about z_{i-1}

Example: The Puma 560 Manipulator



Fill in the DH-table of the Puma 560 Manipulator.





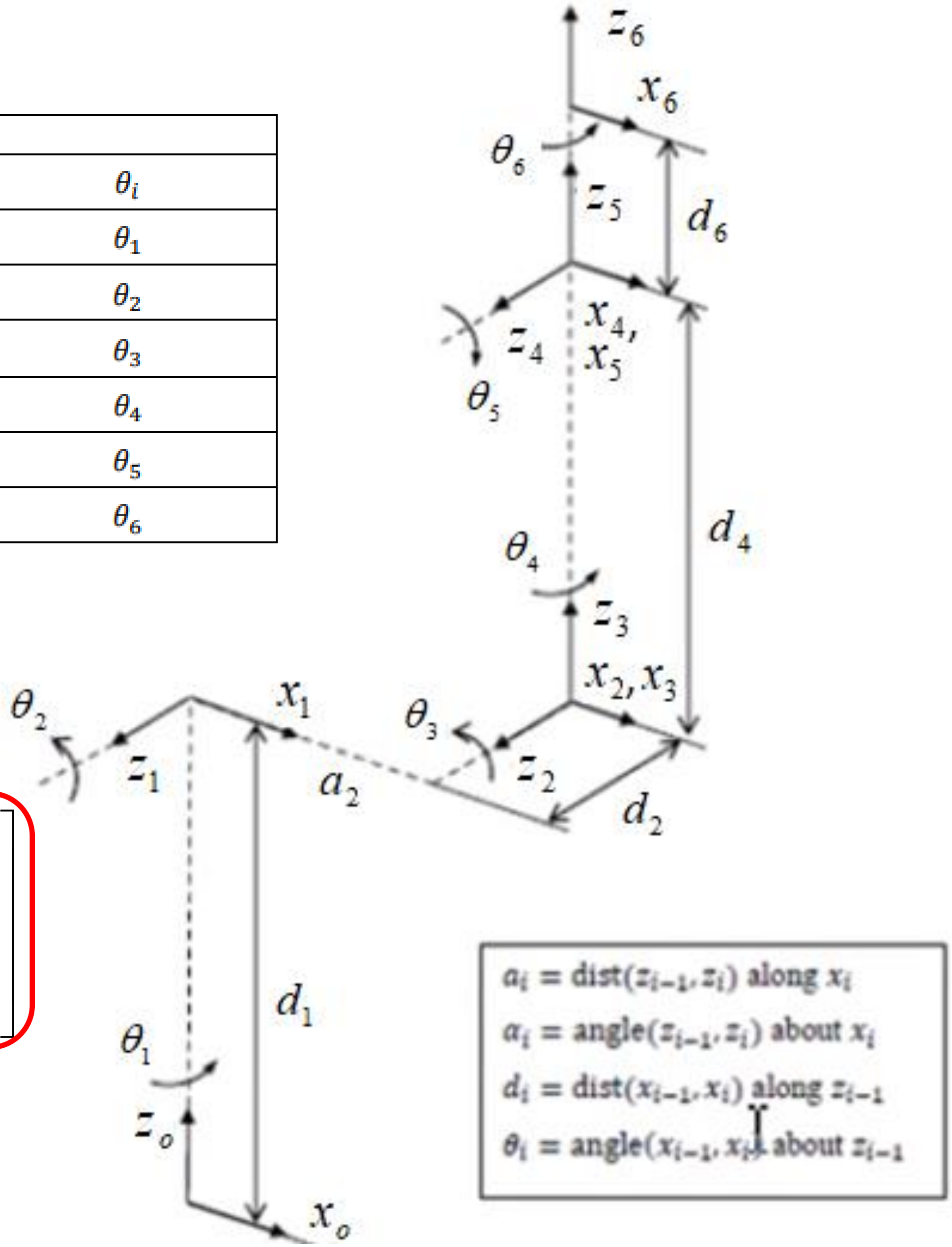
All x axes are parallel which gives “zero-position” manipulator

PUMA robot arm link coordinate parameters				
Link	α_i	a_i	d_i	θ_i
1	90	0	d_1	θ_1
2	0	a_2	$-d_2$	θ_2
3	−90	0	0	θ_3
4	90	0	d_4	θ_4
5	−90	0	0	θ_5
6	0	0	d_6	θ_6

$$\therefore\ {}^{0}_{6}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T\,{}^{5}_{6}T$$

Where

$${}^{i-1}_{\ \ i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



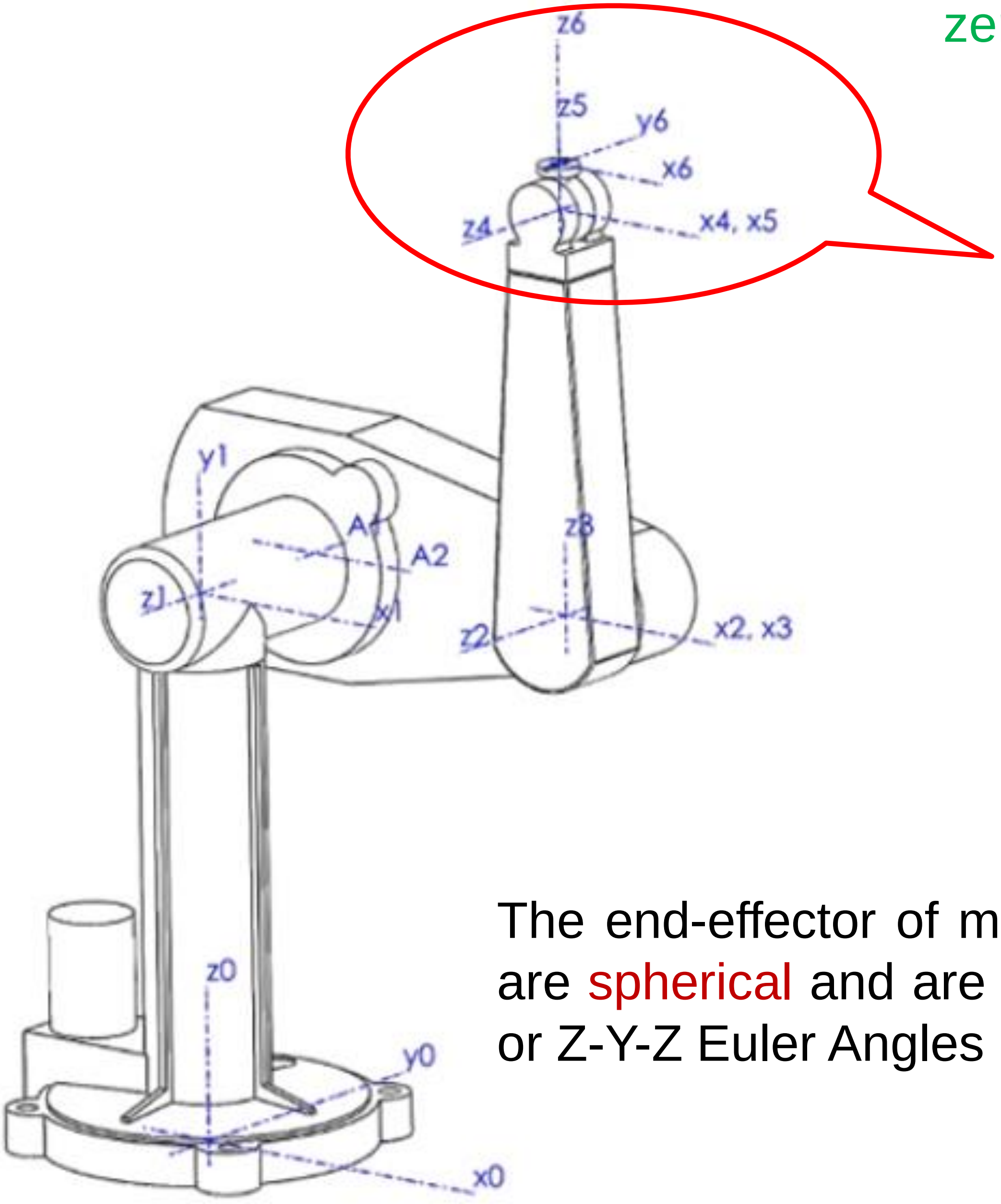
$a_i = \text{dist}(z_{i-1}, z_i)$ along x_i

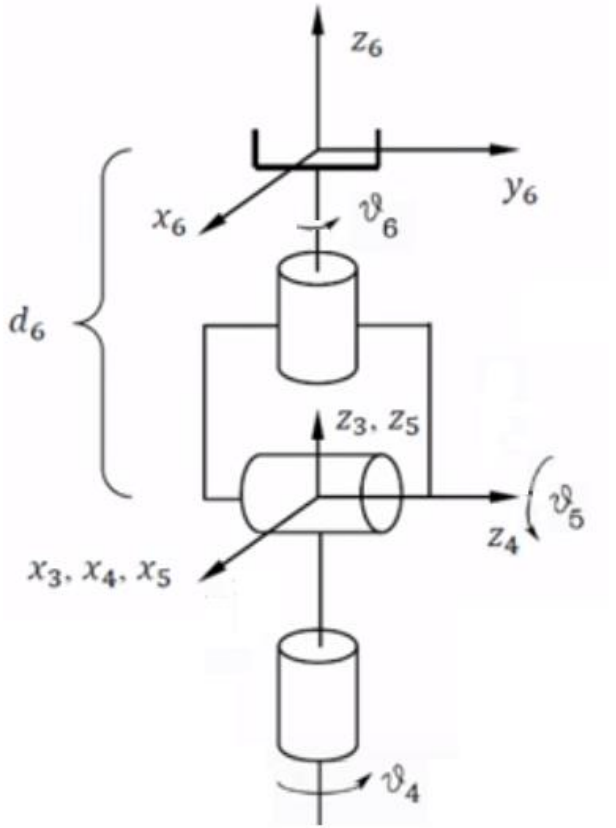
$\alpha_i = \text{angle}(z_{i-1}, z_i)$ about x_i

$d_i = \text{dist}(x_{i-1}, x_i)$ along z_{i-1}

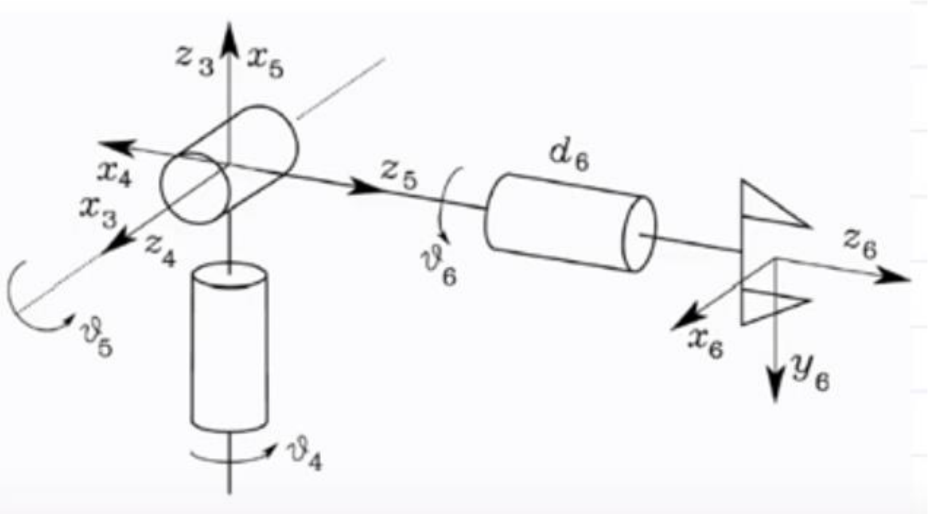
$\theta_i = \text{angle}(x_{i-1}, x_i)$ about z_{i-1}

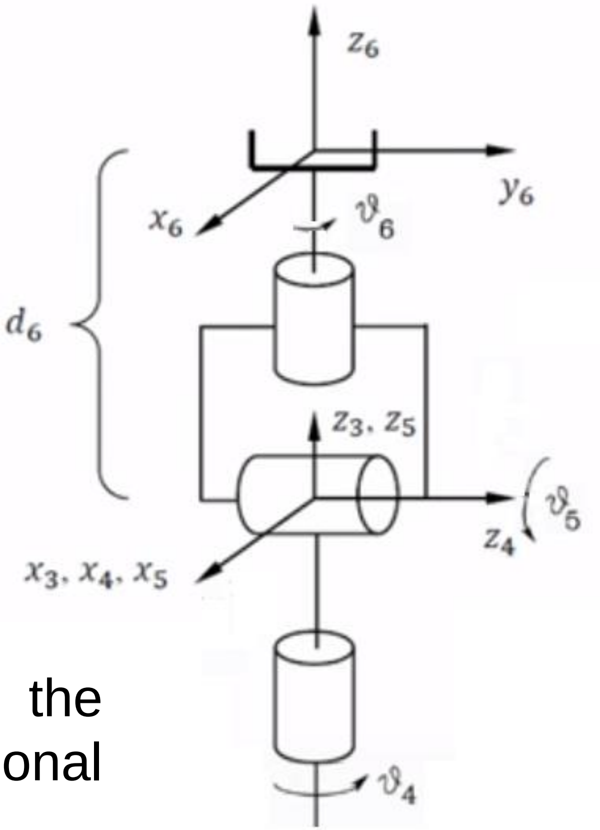
zero-position spherical wrist





The end-effector of most industrial manipulators are spherical and are expressed by roll-pitch-roll or Z-Y-Z Euler Angles





Two different positions of the spherical wrist but same rotational matrix (i.e. Z-Y-Z Euler Angles)

$$
{}^{A}_{B}R_{Z'Y'Z'}(\alpha,\beta,\gamma)=\begin{bmatrix} c\alpha c\beta c\gamma - s\alpha s\gamma & -c\alpha c\beta s\gamma - s\alpha c\gamma & c\alpha s\beta \\ s\alpha c\beta c\gamma + c\alpha s\gamma & -s\alpha c\beta s\gamma + c\alpha c\gamma & s\alpha s\beta \\ -s\beta c\gamma & s\beta s\gamma & c\beta \end{bmatrix}
$$

Robotics

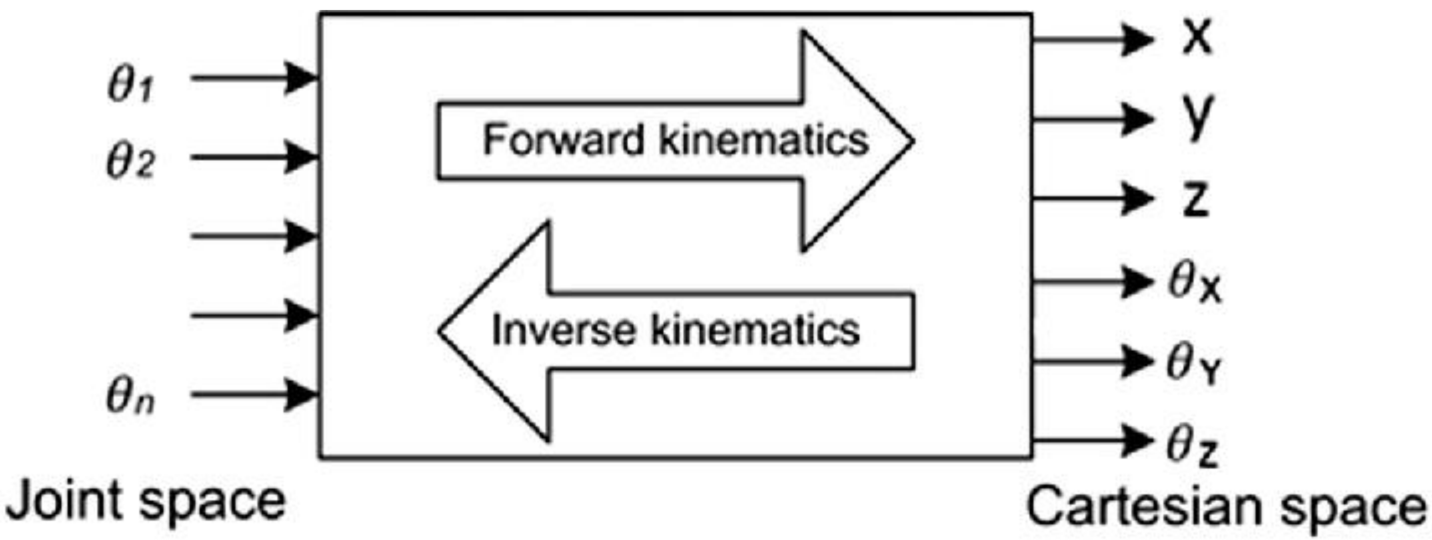
Chapter 4

Inverse kinematics

Ahmad AL-Jarrah

Introduction:

- Inverse Kinematics is the reverse of Forward Kinematics.
- It is the calculation of joint values given the positions, orientations, and geometries of mechanism's parts



i.e. Given the numerical values of ${}^{0}_{n}T$, we attempt to find the values of the joints angles $\theta_1, \theta_2, \theta_3, \ldots \theta_n$.

Assume the following Homogeneous Transformation matrix of a 6DOF manipulator:

Rotation Matrix or orientation of end-effector (e.g. Yaw, Pitch, Roll)

$$\begin{aligned} T_6^0 &= A_1 \cdots A_6 \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Position vector of end-effector

Where

$$\begin{aligned}
r_{11} &= c_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] - d_2(s_4c_5c_6 + c_4s_6) \\
r_{21} &= s_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] + c_1(s_4c_5c_6 + c_4s_6) \\
r_{31} &= -s_2(c_4c_5c_6 - s_4s_6) - c_2s_5c_6 \\
r_{12} &= c_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] - s_1(-s_4c_5s_6 + c_4c_6) \\
r_{22} &= -s_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] + c_1(-s_4c_5s_6 + c_4c_6) \\
r_{32} &= s_2(c_4c_5s_6 + s_4c_6) + c_2s_5s_6 \\
r_{13} &= c_1(c_2c_4s_5 + s_2c_5) - s_1s_4s_5 \\
r_{23} &= s_1(c_2c_4s_5 + s_2c_5) + c_1s_4s_5 \\
r_{33} &= -s_2c_4s_5 + c_2c_5 \\
d_x &= c_1s_2d_3 - s_1d_2 + +d_6(c_1c_2c_4s_5 + c_1c_5s_2 - s_1s_4s_5) \\
d_y &= s_1s_2d_3 + c_1d_2 + d_6(c_1s_4s_5 + c_2c_4s_1s_5 + c_5s_1s_2) \\
d_z &= c_2d_3 + d_6(c_2c_5 - c_4s_2s_5)
\end{aligned}$$

Assume the desired orientation and position of the end-effecter frame is

$$H = \begin{bmatrix} 0 & 1 & 0 & -0.154 \\ 0 & 0 & 1 & 0.763 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To find the corresponding joint variables θ_1, θ_2, d_3, θ_4, θ_5, and θ_6 we must solve the following simultaneous set of nonlinear trigonometric equations:

$$\left.\begin{aligned}
c_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] - s_1(s_4c_5c_6 + c_4s_6) &= 0 \\
s_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] + c_1(s_4c_5c_6 + c_4s_6) &= 0 \\
-s_2(c_4c_5c_6 - s_4s_6) - c_2s_5c_6 &= 1 \\
c_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] - s_1(-s_4c_5s_6 + c_4c_6) &= 1 \\
s_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] + c_1(-s_4c_5s_6 + c_4c_6) &= 0 \\
s_2(c_4c_5s_6 + s_4c_6) + c_2s_5s_6 &= 0 \\
c_1(c_2c_4s_5 + s_2c_5) - s_1s_4s_5 &= 0 \\
s_1(c_2c_4s_5 + s_2c_5) + c_1s_4s_5 &= 1 \\
-s_2c_4s_5 + c_2c_5 &= 0 \\
c_1s_2d_3 - s_1d_2 + d_6(c_1c_2c_4s_5 + c_1c_5s_2 - s_1s_4s_5) &= -0.154 \\
s_1s_2d_3 + c_1d_2 + d_6(c_1s_4s_5 + c_2c_4s_1s_5 + c_5s_1s_2) &= 0.763 \\
c_2d_3 + d_6(c_2c_5 - c_4s_2s_5) &= 0
\end{aligned}\right\}$$

The equations are much too difficult to solve directly in closed form !!

This is the case for most robot arms

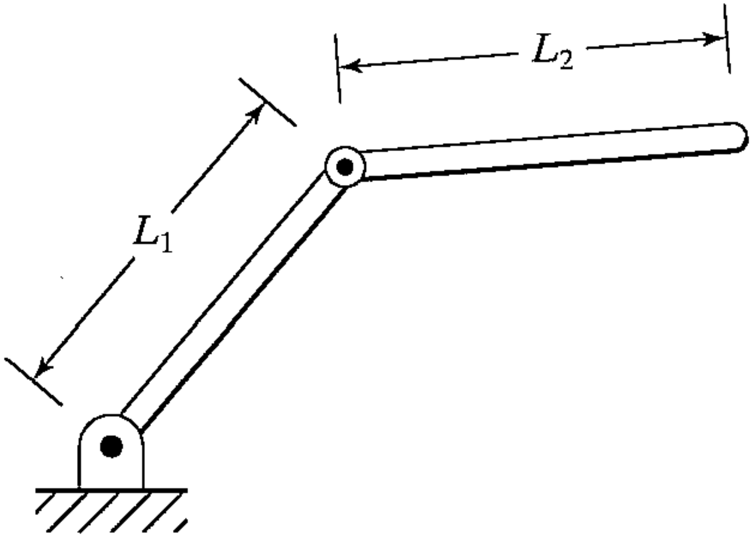
Solvability:

- Solving inverse kinematics is more complex than the forward kinematics.
  - The equations to solve are non-linear / transcendental.
  - Closed-form solution (explicit relations) can not always be found.
  - Try to use Numerical methods to solve them.
- Solving inverse kinematics raises the question of the **manipulator's workspace**.
- **Workspace** is the volume of space that the end-effector [E] of the manipulator can reach. (reaching the point in at least one orientation is called reachable workspace).
  - For a solution to exist, the specified goal point MUST lie within the workspace.

Consider the two-link manipulator

If $l_1 = l_2$ ➤ The reachable workspace is a disc of radius $2l_1$

If $l_1 \neq l_2$ ➤ The reachable workspace is a ring of outer radius $l_1 + l_2$ and inner radius $|l_1 - l_2|$



- Inside the reachable workspace there are two possible orientations of the end-effector.
- On the boundaries of the workspace, there is only one possible orientation.

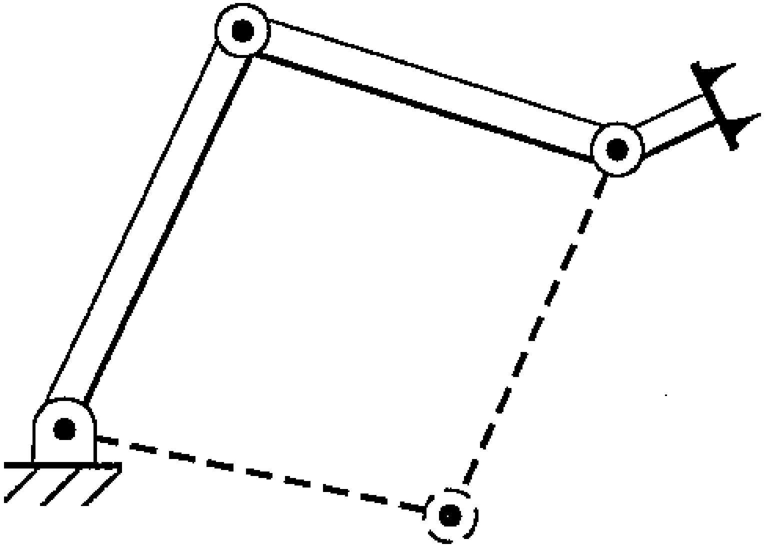
- In general, to attain a goal positions and orientations in a 3-space, the manipulator is required to have 6 DOF.
  - Manipulators less than 6 DOF can not reach general goals in the 3-space.
- The set of reachable goal frames for a given manipulator constitutes its reachable workspace.
- For a manipulator with n-DOF (n<6), the reachable workspace can be thought of as a portion of n-DOF **subspace.**
- In the same manner in which the workspace of a 6-DOF manipulator is a subset of space, the workspace of a simpler manipulator is a subset of its subspace.

- For example, Two-links planar manipulator is a 2DOF and hence is restricted to attain any goal in space
  - The subspace in this case is a plane and the workspace is subset of this plane; namely a circle of radius $l_1 + l_2$ for the case that $l_1 = l_2$

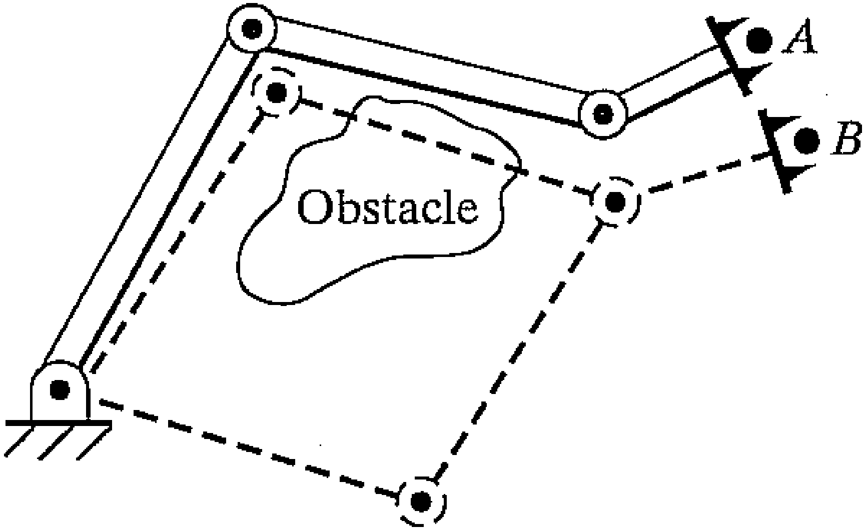
- The Workspace of a manipulator is a subset of a subspace that can be associated with any manipulator.

(many physical limits can add restrictions to reach a goal in the robot workspace!)

Multiple Solutions:



Dashed lines indicates a second solution



-Closest solution

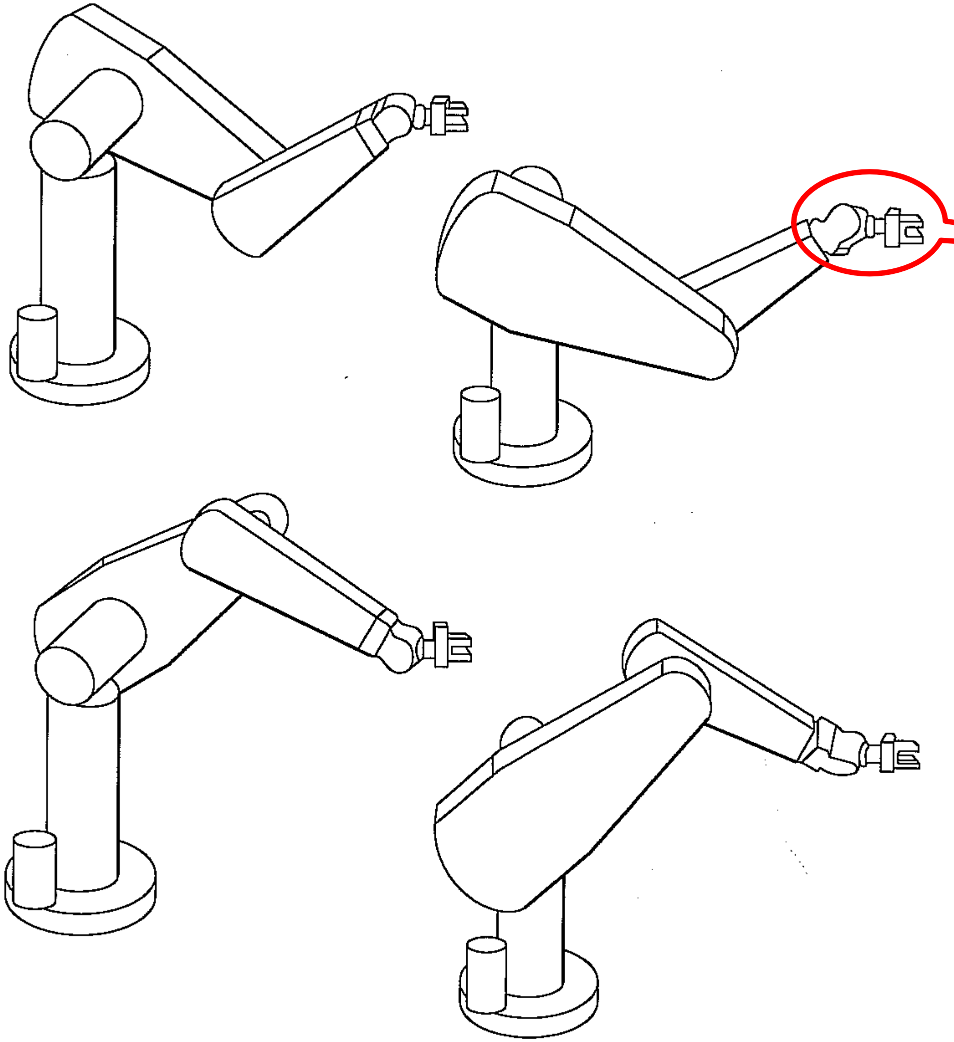
-The presence of obstacles!

- Moving smaller joints

The number of solutions depends not only on the number of joints but also on the link parameters (i.e. DH-parameters) and the allowable ranges of motion of the joints.

Puma560

Flipping the end-effector angels

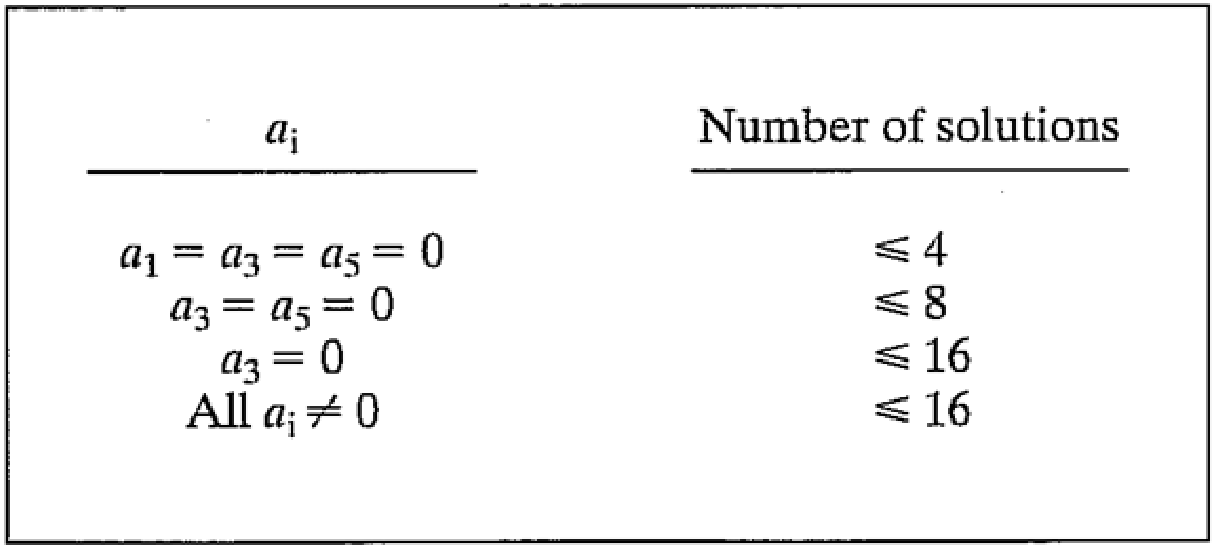


$$\theta_4' = \theta_4 + 180°$$

$$\theta_5' = -\theta_5$$

$$\theta_6' = \theta_6 + 180°$$

- More non-zero parameters give more solutions.
- The table shows number of solutions for general rotary-joints manipulator with 6 DOF.



a_i	Number of solutions
$a_1 = a_3 = a_5 = 0$	$\leqslant 4$
$a_3 = a_5 = 0$	$\leqslant 8$
$a_3 = 0$	$\leqslant 16$
All $a_i \neq 0$	$\leqslant 16$

Number of solutions vs. nonzero a_i

Method of Solution:

Closed-form solutions and numerical solutions.

Closed-form solutions are solutions based on analytical expressions (like the quadratic formula in algebra).

Numerical solutions use approximations and iterations to converge to a solution.

We will restrict our attention to closed-form solution methods.

Within the class of closed-form solutions, we distinguish two methods of obtaining the solution: **algebraic** and **geometric**.

- Numerical solutions are in general time consuming and expensive; hence, it is considered very important to design a manipulator so that a closed-form solution exists.

- A major result in kinematics is that all systems with revolute and prismatic joints having a total of 6 DOF in a single series chain are solvable (Mostly numerical solutions)

- Very special cases can robots with 6 DOF be solved analytically (closed-form) like the Puma 560

  - Such robots are characterized by having several intersecting joint axes or many zero or 90 degrees twist angles.

Examples!

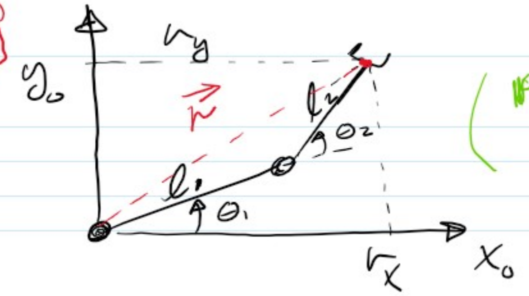
Examples

Analytical solution

Ex.1

Given r_x, r_y

find θ_1 & θ_2.



(planar Robot)

cosine law: $r^2 = l_1^2 + l_2^2 + 2l_1l_2C_2 = r_y^2 + r_x^2$

$$\begin{cases} r_x = l_1C\theta_1 + l_2C(\theta_1+\theta_2) \\ r_y = l_1S\theta_1 + l_2S(\theta_1+\theta_2) \end{cases}$$

from forward Kinematics

$$r_x^2 + r_y^2 = l_1^2C_1^2 + l_2^2C_{12}^2 + 2l_1l_2C_1\underline{C_{12}} + l_1^2S_1^2 + l_2^2S_{12}^2 + 2l_1l_2S_1\underline{S_{12}}$$

But,
$$\begin{cases} C(\theta_1 \mp \theta_2) = C\theta_1C\theta_2 \pm S\theta_1S\theta_2 \\ \& \ S(\theta_1 \mp \theta_2) = S\theta_1C\theta_2 \mp S\theta_2C\theta_1 \end{cases}$$

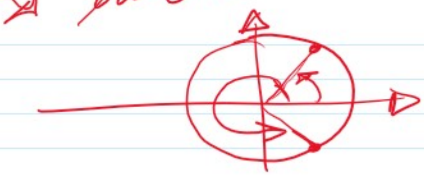
$$\therefore r_x^2 + r_y^2 = l_1^2 + l_2^2 + 2l_1l_2(C_1C_{12} + S_1S_{12})$$
$$= l_1^2 + l_2^2 + 2l_1l_2(C_1(C_1C_2 - S_1S_2) + S_1(S_1C_2 + S_2C_1))$$
$$= l_1^2 + l_2^2 + 2l_1l_2[C_2C_1^2 - \cancel{C_1S_1S_2} + S_1^2C_2 + \cancel{S_1S_2C_1}]$$
$$= l_1^2 + l_2^2 + 2l_1l_2[C_2\underbrace{(C_1^2 + S_1^2)}_{1}]$$

$$r_x^2 + r_y^2 = l_1^2 + l_2^2 + 2l_1l_2C_2$$

re-arrange;

$$\boxed{\cos\theta_2 = \frac{r_x^2 + r_y^2 - l_1^2 - l_2^2}{2l_1l_2}} \quad \dots (1)$$

→ two solutions



But,

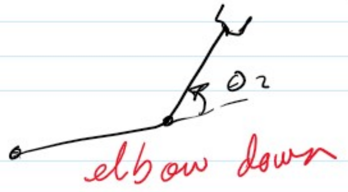
$$\boxed{\sin^2\theta_2 + \cos^2\theta_2 = 1} \quad \dots (2)$$

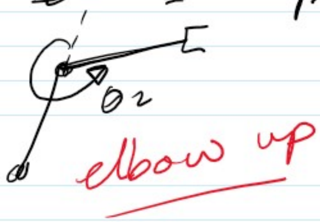
⇒ a solution of eqn (1) exists if $\boxed{-1 \le \cos\theta_2 \le 1}$, otherwise, the point given by r_x & r_y is out of reach (not within the robot workspace)

from eqn (2), ⇒ $\boxed{\sin\theta_2 = \pm\sqrt{1-\cos^2\theta_2}}$

if (+) sign gives ⇒ elbow down configuration (0 – 180°)

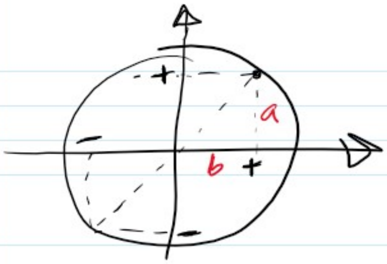
(−) sign gives ⇒ " up. " (180°–360°)





⇒ tangent is more accurate than sin & cos.

Now, $\theta_2 = \text{atan2}\,(\sin(\theta_2), \cos(\theta_2))$



atan 2 uses the magnitude & signs and it's not like $\tan^{-1}$

atan2(a, b) ⇒ using magnitudes & signs of a & b.

⇒ atan 2 (−2, 2) ≠ atan 2 (2, −2)

where $\tan^{-1}\frac{-2}{2} = \tan^{-1}\frac{2}{-2}$

Once we find θ_2, we need to find θ_1

⇒ Subs. for θ_2 into r_x & r_y eqns. as follows:-

$$\begin{cases} r_x = l_1 C\theta_1 + l_2 \boxed{C(\theta_1+\theta_2)} \\ r_y = l_1 S\theta_1 + l_2 \boxed{S(\theta_1+\theta_2)} \end{cases}$$

using:

$C_{12} = C_1 C_2 - S_1 S_2$

$S_{12} = S_1 C_2 + S_2 C_1$

$\therefore \; l_1 C_1 + l_2 C_1 C_2 - l_2 S_1 S_2 = r_x$ ... (1)

$l_1 S_1 + l_2 S_2 C_1 + l_2 S_1 C_2 = r_y$ ... (2)

re-arrange based on θ_2 ,

$$\left\{ \begin{array}{l} (l_1 + l_2 C_2) C_1 - l_2 S_2 S_1 = r_x \\ l_2 S_2 C_1 + (l_1 + l_2 C_2) S_1 = r_y \end{array} \right\}$$

let

$K_1 = l_1 + l_2 C_2$

$K_2 = l_2 S_2$

$$\therefore \left\{ \begin{array}{ll} K_1 C\theta_1 - K_2 S\theta_1 = r_x & \text{... (a)} \\ K_2 C\theta_1 + K_1 S\theta_1 = r_y & \text{... (b)} \end{array} \right. \Rightarrow$$

Multiply by $\frac{K_2}{K_1}$ then add to $\underline{r_x}$

$$\begin{array}{rl} & K_1 C_1 - \cancel{K_2 S_1} = r_x \\ + & \frac{K_2^2}{K_1} C_1 + \cancel{K_2 S_1} = \frac{K_2}{K_1} r_y \\ \hline \end{array}$$

$$K_1 C_1 + \frac{K_2^2}{K_1} C_1 = r_x + \frac{K_2}{K_1} r_y$$

$$C_1 \left[\frac{K_1^2 + K_2^2}{\cancel{K_1}} \right] = \frac{K_1 r_x + K_2 r_y}{\cancel{K_1}}$$

$$\boxed{\therefore \; \cos\theta_1 = \frac{K_1 r_x + K_2 r_y}{K_1^2 + K_2^2}}$$

subs. for K_1 & K_2 :

$$\cos\theta_1 = \frac{(l_1 + l_2 C_2) r_x + l_2 S_2 r_y}{l_1^2 + l_2^2 C_2^2 + 2 l_1 l_2 C_2 + l_2^2 S_2^2}$$

$$\boxed{\cos\theta_1 = \frac{(l_1 + l_2 C_2)\, r_x + l_2 S_2\, r_y}{l_1^2 + l_2^2 + 2 l_1 l_2 C_2}}$$

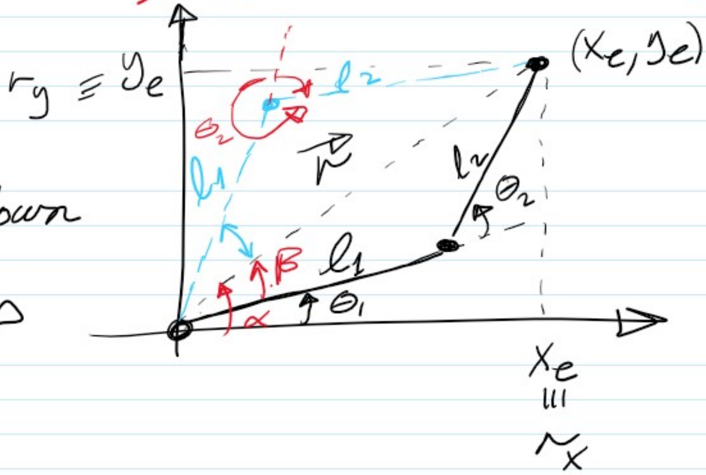
$r_x^2 + r_y^2$ using cosine law

Similarly, using eqns (a) & (b) for $S\theta_1$ instead;

$$\boxed{\sin\theta_1 = \frac{(l_1 + l_2 C_2)\, r_y - l_2 S_2\, r_x}{r_x^2 + r_y^2}}$$

Then, $\theta_1 = \text{atan2}(S_1, C_1)$

If we use Geometry for same robot;



from Geometry;

$\theta_1 = \alpha - \beta$ elbow down

$\theta_1 = \alpha + \beta$ elbow up

→ $180 > \theta_2 > 0$

→ $360 > \theta_2 > 180$ or $\underbrace{-180 < \theta_2 < 0}_{\text{clock wise}}$

Find α & β to find θ_1.

$$\rightarrow \alpha = \tan^{-1} \frac{y_e}{x_e}$$

$$r = \sqrt{y_e^2 + x_e^2}$$

using the cosine law:

$$l_2^2 = r^2 + l_1^2 - 2 r l_1 \cos\beta$$

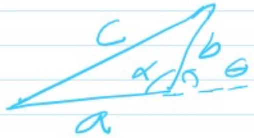
subs. for $r = \sqrt{y_e^2 + x_e^2}$

$$\rightarrow l_2^2 = x_e^2 + y_e^2 + l_1^2 - 2 l_1 \sqrt{y_e^2 + x_e^2}\,\cos\beta$$

$$\therefore \cos\beta = \frac{x_e^2 + y_e^2 + l_1^2 - l_2^2}{2 l_1 \sqrt{y_e^2 + x_e^2}} \qquad 0 < \beta < 180$$

∴ we know α & β ⇒ find θ_1.

Note:



$$c^2 = a^2 + b^2 + 2ab\cos\theta$$

$$\text{or } = a^2 + b^2 - 2ab\cos\alpha$$

For θ_2: use cosine law;

$$r^2 = l_1^2 + l_2^2 + 2 l_1 l_2 \cos\theta_2$$

where, $r^2 = x_e^2 + y_e^2$

$$\boxed{\therefore \cos\theta_2 = \frac{x_e^2 + y_e^2 - l_1^2 - l_2^2}{2 l_1 l_2}}$$

$180 > \theta_2 > 0 \Rightarrow$ elbow down

$360 > \theta_2 > 180 \Rightarrow$ " up ⇒ or $\theta_2 = -\theta_2$

$$\boxed{-180 < \theta_2 < 0}$$

$$\boxed{-180 < \theta_2 < 0}$$

OR, we might be given forward Kinematics through ($^{0}T_e$) & we try to solve for the angles.

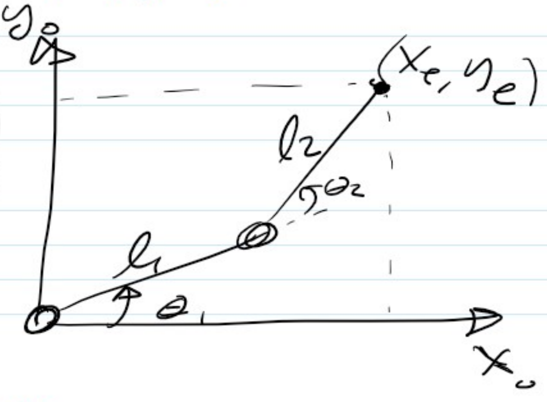
$$^{0}T_e = \left[\begin{array}{c|c} \text{orientation} & x_e \\ R & x_y \\ & x_z \\ \hline - & 1 \end{array}\right]$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & x_e \\ r_{21} & r_{22} & r_{23} & y_e \\ r_{31} & r_{32} & r_{33} & z_e \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

given in numbers & we need to find joints variables. ($\theta_1, \theta_2 \cdots$).

compare

$$^{0}T_e = \left[\begin{array}{ccc|c} c_{12} & -s_{12} & 0 & l_1 c_1 + l_2 c_{12} \\ s_{12} & c_{12} & 0 & l_1 s_1 + l_2 s_{12} \\ 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 1 \end{array}\right]$$



use
$$\left.\begin{array}{l} S_{12} = r_{21} \\ C_{12} = r_{11} \end{array}\right] \text{numbers}$$

$$\boxed{\therefore\ \theta_1+\theta_2 = \operatorname{atan2}(r_{21}, r_{11})} \text{ --- (1)}$$

also,

$$x_e = l_1 C_1 + l_2 \underbrace{C_{12}}_{r_{11}} \Rightarrow C\theta_1 = \frac{x_e - l_2 r_{11}}{l_1}$$

&

$$y_e = l_1 S_1 + l_2 \underbrace{S_{12}}_{r_{21}} \Rightarrow S\theta_1 = \frac{y_e - l_2 r_{21}}{l_1}$$

$$\boxed{\therefore\ \theta_1 = \operatorname{atan2}(S_1, C_1)} \text{ --- ②}$$

one solution.

from eqn (1), we can find θ_2.

⇒ Most of times, while solving the inverse-Kinematics, you end up by solving a Transcendental eqn.

↳ can't be solved directly like Quadratic eqn to have a closed-form solution.

⇒ Usually, we use numerical methods to solve Transcendental eqn.

$\Rightarrow$ one method is trying to convert it into algebric eqn.

$$a \cos\theta + b \sin\theta = c \quad \text{--- (1)}$$

$\hookrightarrow$ There is no way to say $\theta = ..$

$\Rightarrow$ to convert eqn (1) into algebric eqn (polynomial);

$$\text{let} \quad u = \tan\frac{\theta}{2} \quad \text{--- (2)}$$

$$u = \frac{\sin\theta/2}{\cos\theta/2}$$

$\sin^2\theta + \cos^2\theta = 1$

$$u^2 = \frac{1 - \cos^2\theta/2}{\cos^2\theta/2} \quad \text{--- (3)}$$

$$\text{But,} \quad \cos^2\frac{\theta}{2} = \frac{1+\cos\theta}{2} \quad \text{---- (4)}$$

$\hookrightarrow$ using;

$$\cos 2\theta = \cos\theta\cos\theta - \sin\theta\sin\theta$$
$$= \cos^2\theta - (1-\cos^2\theta)$$
$$\cos 2\theta = 2\cos^2\theta - 1$$

Subs. eqn. (4) into (3);

$$u^2 = \frac{1 - \left(\frac{1+\cos\theta}{2}\right)}{\frac{1+\cos\theta}{2}} = \frac{1-\cos\theta}{1+\cos\theta}$$

$$\text{re-arrange} \Rightarrow \quad \cos\theta = \frac{1-u^2}{1+u^2} \quad \text{--- (5)}$$

$$\text{But,} \quad \sin^2\theta = 1 - \cos^2\theta$$

$$\sin\theta = \frac{2u}{(1+u^2)^2} \quad -- \quad (6)$$

subs. eqns (5) & (6) in to (1) & re-arrange;

$$(a+c)u^2 - 2bu + (c-a) = 0$$

$$\therefore \quad u = \frac{2b \pm \sqrt{4b^2 - 4(a+c)(c-a)}}{2(a+c)}$$

$$\therefore \quad \theta = 2\tan^{-1}\left(\frac{b \pm \sqrt{b^2 - 2(a+c)(c-a)}}{a+c}\right)$$

Another method is by converting in to Quadratic eqn;

$$a\cos\theta + b\sin\theta = c \quad \cdots \quad (1)$$

use $\cos\theta = \sqrt{1-\sin^2\theta} \quad --- \quad (2)$

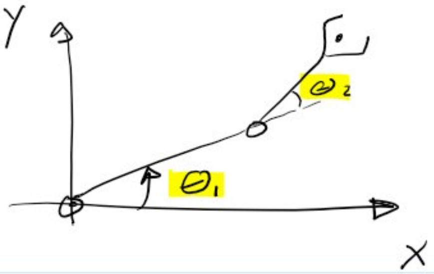
Subs. (2) into (1) & re-arrange;

$$(a^2+b^2)\sin^2\theta - 2bc\sin\theta + (c^2-a^2) = 0$$

$$\hookrightarrow \quad \sin\theta = \frac{2bc \pm \sqrt{4b^2c^2 - 4(a^2+b^2)(c^2-a^2)}}{2(a^2+b^2)}$$

Singularity

$$\underbrace{V}_{ee} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = J \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$



$$\begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \end{bmatrix}$$

2-links
planar

$\phi = \theta_1 + \theta_2$

orientation of end-effector (ee) w.r.t fixed frame @ base.

$\rightarrow \dot{\phi} = \dot{\theta}_1 + \dot{\theta}_2 = \omega_1 + \omega_2$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

* Singularity :

let **V** is the velocity of the end-effector of a manipulator & θ is the vector representing Joints variables

$\rightarrow$ given $\underline{\theta}$ & $\underline{\dot{\theta}}$, we can find **V**

$$V = J(\theta)\,\dot{\theta} \quad --- (*)$$

Now, if we have V & the configuration of the robot $(\theta \Rightarrow J(\theta))$, can we find $\dot{\theta}$?

$$\dot{\theta} = \underbrace{J^{-1}(\theta)}_{\text{inverse of Jacobian}} V$$

so to find $\dot{\theta}$, $J(\theta)$ MUST be invertible.

It is not always possible to find $\dot{\theta}$

& if the inverse doesn't exist, the matrix $J(\theta)$ is singular.

⇒ Most of manipulators have configurations where $J(\theta)$ is sigular

⇒ These configurations are called singularities of the Robot.

Back to the previous example of 2-links planar manipulator;

$$J(\theta) = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \end{bmatrix}$$

finding singularities

$$\Rightarrow \det(J(\theta)) = (-l_1 S_1 - l_2 S_{12})(l_2 C_{12}) + l_2 S_{12}(l_1 C_1 + l_2 C_{12}) = 0$$

$$\vdots$$

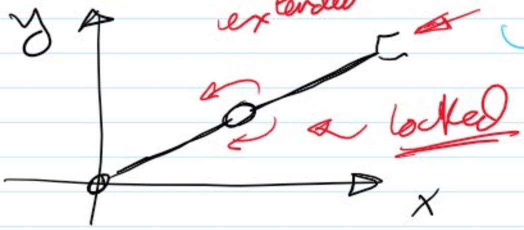
$$\det(J(\theta)) = \underbrace{l_1 l_2}\, \sin\theta_2 = 0$$

↳ To have $\det(J(\theta)) = 0 \Rightarrow \underbrace{\sin\theta_2 = 0}$

$\Downarrow$ when $\theta_2 = \pm \overset{\text{integer}}{k}\pi$

i.e (0 or ±π)

if $\theta_2 = 0$ → Link 2 is fully extended.

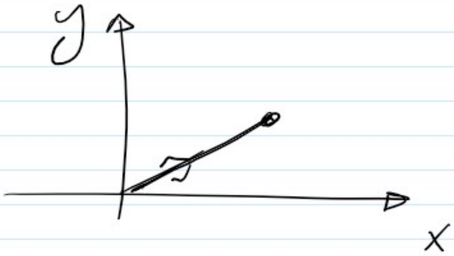


applying a load @ the tip of ee will not result in a motion since the 2nd joint is locked

((i.e the robot lost one of it's DOF))

↳ means, losing the ability to move in or around a certain axis (axis of rotation)

if $\theta_2 = \pm 180°$



- The singularities are also @ the boarders of minipulator workspace.

(e.g. fully extended)

- Also, @ singularity configuration, the speeds of joints will be very high close to ∞.

Try to find $J^{-1}(\theta)$;

$$J^{-1} = \frac{1}{l_1 l_2 \sin\theta_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix}$$

if $\underbrace{\dot{x} = 1\ m/s}$ & $\underbrace{\dot{y} = 0}$, find $\dot{\theta}_1$ & $\dot{\theta}_2$.

$$\therefore \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = J^{-1}(\theta) \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix}$$

$$= \frac{1}{l_1 l_2 \sin\theta_2} \begin{bmatrix} l_2 c_{12} & l_2 s_{12} \\ -l_1 c_1 - l_2 c_{12} & -l_1 s_1 - l_2 s_{12} \end{bmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$\therefore \dot{\theta}_1 = \frac{\cancel{l_1} c_{12}}{\cancel{l_1} l_2 s_2} = \frac{c_{12}}{l_2 \boxed{s_2}}$$

$$\dot{\theta}_2 = \frac{-l_1 c_1 - l_2 c_{12}}{l_1 l_2 s_2} = -\frac{c_1}{l_2 \boxed{s_2}} - \frac{c_{12}}{l_1 \boxed{s_2}}$$

if we assume $\theta_2 = 0$ (singularity position)

$$\Downarrow$$

$$\det(J(\theta)) = 0$$

$\rightarrow$ $\dot{\theta}_1 \rightarrow \infty$

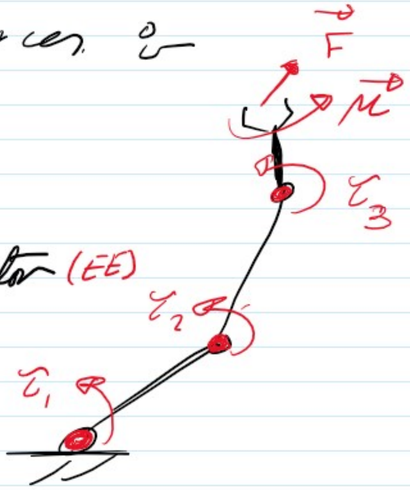
& $\dot{\theta}_2 \rightarrow \infty$

$\text{rank}(J) < \underline{2}$ $\Leftarrow$ { J is not in full rank

∴ We always try to avoid singularity in Design & Simulation.

* Jacobian: Static forces &

- The robot sometimes is pushing on something in the environment with the end-effector (EE) or perhaps is supporting a load at the EE without motion



$\Downarrow$

means static equilibrium.

What torques/forces are required at the joints in order to maintain static equ.?

Given $\underbrace{F_x, F_y, F_z}_{\vec{F}}, \underbrace{M_x, M_y \; \& \; M_z}_{\vec{M}}$

find the torques/forces at the joints to keep the system at static equilibrium.

$\Rightarrow$ let $\vec{F}$: (6x1) vector of forces & Moments acting @ EE.

$\delta\vec{x}$: (6x1) vector of infinitesmal displacement of the EE.

cartesian space

joints space

$\delta\vec{\theta}$: (6×1) vector of infintesmal joint displacement.

$\underline{\underline{\theta}}$ or $\underline{\underline{d}}$

(angular) (Linear)

& $\vec{\tau}$: (6×1) vector representing the joint torques/forces to balance $\vec{F}$ & keep static. equ.

Using vistual work;

$$\underbrace{\vec{F}\cdot\delta\vec{X}}_{\text{work done @ EE.}} = \underbrace{\vec{\tau}\cdot\delta\vec{\theta}}_{\text{total work done @ joints}}$$

scalar (energy)

→ virtual work of applied forces/moments is zero for all virtual movements from static equilibrium.

But;

$$\begin{bmatrix} F_x \\ F_y \\ F_z \\ M_x \\ M_y \\ M_z \end{bmatrix} \cdot \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\theta_x \\ \delta\theta_y \\ \delta\theta_z \end{bmatrix} = \underbrace{\begin{bmatrix} F_x & F_y & \cdots & M_z \end{bmatrix}}_{\text{transpose}} \begin{bmatrix} \delta x \\ \vdots \\ \delta\theta_z \end{bmatrix}$$

$\therefore \vec{F}\,\delta\vec{X} = \vec{\tau}\cdot\delta\vec{\theta} \Rightarrow \boxed{F^T\delta x = \tau^T\delta\theta}$

recall from Jacobian $\Rightarrow$ $\delta X = J\,\delta\theta$ 5

$$\therefore \quad F^T J(\theta) \cancel{\delta\theta} = \tau^T \cancel{\delta\theta}$$

$$\therefore \tau^T = F^T J(\theta)$$

Note)

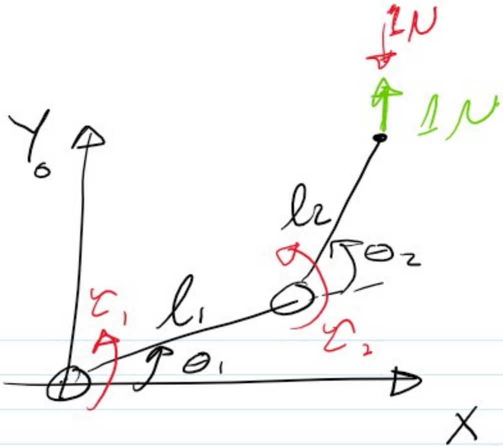
$$A^T = B^T C$$

$$(A^T)^T = (B^T C)^T = C^T B$$

$$\therefore \tau = J^T(\theta) F$$

Ex.)

$$J(\theta) = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & -l_2 S_{12} \\ l_1 C_1 + l_2 C_{12} & l_2 C_{12} \end{bmatrix}$$



$$\therefore J^T(\theta) = \begin{bmatrix} -l_1 S_1 - l_2 S_{12} & l_1 C_1 + l_2 C_{12} \\ -l_2 S_{12} & l_2 C_{12} \end{bmatrix}$$

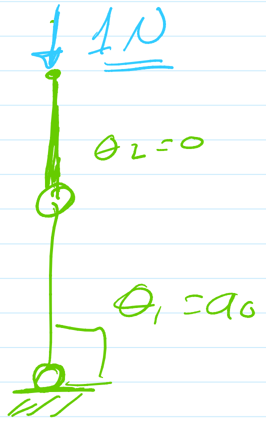
if $l_1 = l_2 = 1$ m & $\theta_1 = 0$, $\theta_2 = 60^\circ$

find τ_1 & τ_2 that should support 1N from environment @ EE as shown.

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = J^T(\theta) \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{\sqrt{3}}{2} & \frac{3}{2} \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} \\ \frac{1}{2} \end{bmatrix} [N.m]$$

assume $\theta_1 = 90$ & $\theta_2 = 0$ (singularity)



$$\underline{\underline{\tau_1 = \tau_2 = 0 \ N.m}}$$

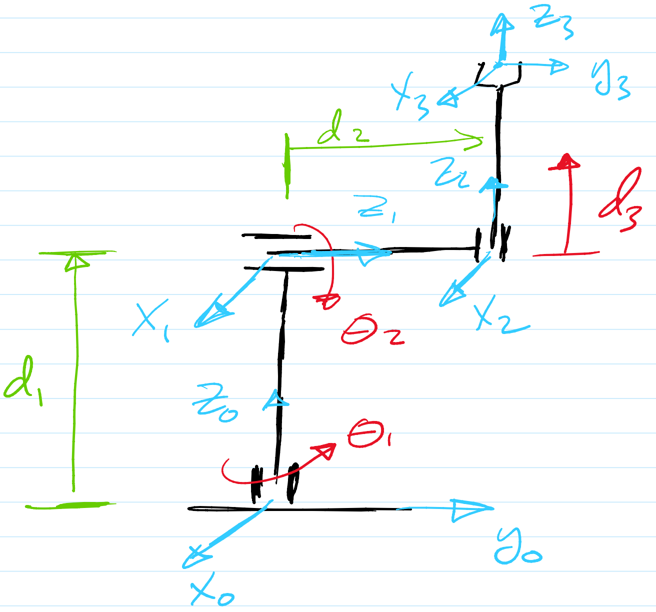
even if the acting external force is 1000 N.

- @ singularity, the system (robot) can sustain a lot of forces even the robot is not moving in that direction.

- near singular configuration, mechanical advantange leads toward infinity (i.e small torques @ joints can produce large forces @ EE).

Ex.

i	θ_i	α_i	a_i	d_i
1	θ_1	$-\frac{\pi}{2}$	0	d_1
2	θ_2	$\pi/2$	0	d_2
3	0	0	0	d_3



$$= {}^0T_3 = \left[\begin{array}{ccc|c} & & & x \\ & & & y \\ & & & z \\ 0 & 0 & 0 & 1 \end{array}\right]$$

$$\| $$

$$\underbrace{{}^0T_1}\,\underbrace{{}^1T_2}\,\underbrace{{}^2T_3}$$

$$= {}^{0}T_{3} = \left[\begin{array}{ccc|c} & & & \vdots \\ & & & y \\ & & & z \\ \hline 0 & 0 & 0 & 1 \end{array}\right] \qquad x_0$$

$$\| $$

$$\underbrace{{}^{0}T_{1}}\,\underbrace{{}^{1}T_{2}}\,\underbrace{{}^{2}T_{3}}$$

$$\begin{cases} x = c_1 s_2 d_3 - s_1 d_2 = f_1(\theta_1, \theta_2, d_3) \\ y = s_1 s_2 d_3 + c_1 d_2 = f_2(\theta_1, \theta_2, d_3) \\ z = c_2 d_3 = f_3(\theta_1, \theta_2, d_3) \end{cases}$$

So $J = \begin{pmatrix} \frac{\partial f_1}{\partial \theta_1} & \frac{\partial f_1}{\partial \theta_2} & \frac{\partial f_1}{\partial d_3} \\ \frac{\partial f_2}{\partial \theta_1} & \frac{\partial f_2}{\partial \theta_2} & \frac{\partial f_2}{\partial d_3} \\ \frac{\partial f_3}{\partial \theta_1} & \frac{\partial f_3}{\partial \theta_2} & \frac{\partial f_3}{\partial d_3} \end{pmatrix}$

So $\frac{\partial f_1}{\partial \theta_1} = -s_1 s_2 d_3 - c_1 d_2$

$\frac{\partial f_1}{\partial \theta_2} = c_1 c_2 d_3$ & $\frac{\partial f_1}{\partial d_3} = c_1 s_2$

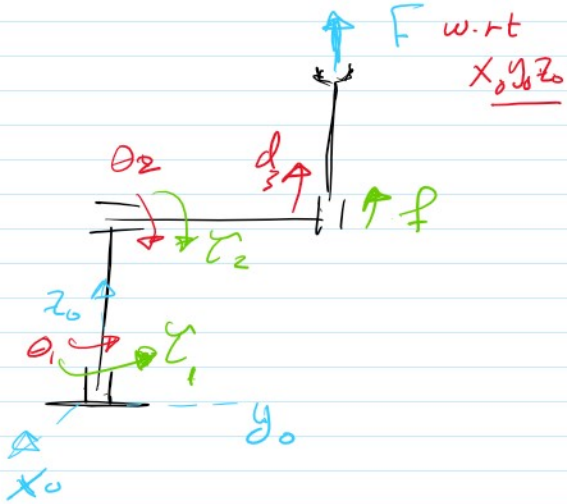
& $\frac{\partial f_2}{\partial \theta_1} = c_1 s_2 d_3 - s_1 d_2$

$\frac{\partial f_2}{\partial \theta_2} = s_1 c_2 d_3$ & $\frac{\partial f_2}{\partial d_3} = s_1 s_2$

finally, $\frac{\partial f_3}{\partial \theta_1} = 0$, $\frac{\partial f_3}{\partial \theta_2} = -s_2 d_3$ & $\frac{\partial f_3}{\partial d_3} = c_2$

$$\therefore J = \begin{bmatrix} -S_1S_2d_3 - C_1d_2 & C_1C_2d_3 & C_1S_2 \\ C_1S_2d_3 - S_1d_2 & S_1C_2d_3 & S_1S_2 \\ 0 & -S_2d_3 & C_2 \end{bmatrix}$$

Now, assume there is a force acting upon the tip (F) & we want to find τ_1, τ_2 & f to maintain static equ.



$$\begin{bmatrix} \tau_1 \\ \tau_2 \\ f \end{bmatrix} = \underbrace{J^T(\theta)} \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}$$

$$\therefore \begin{bmatrix} \tau_1 \\ \tau_2 \\ f \end{bmatrix} = \begin{bmatrix} -S_1S_2d_3 - C_1d_2 & C_1S_2d_3 - S_1d_2 & 0 \\ C_1C_2d_3 & S_1C_2d_3 & -S_2d_3 \\ C_1S_2 & S_1S_2 & C_2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ F \end{bmatrix}$$

$\therefore \tau_1 = 0$, $\tau_2 = -S_2 d_3 F$ & $f = F C_2$

no rotation of 1st joint

i.e. we left by only moment at the 2nd joint

$\left\{ f = F \text{ if } \theta_2 = 0 \right.$

$\left\{ \& \ f = 0 \text{ if } \theta_2 = 90^\circ \right.$

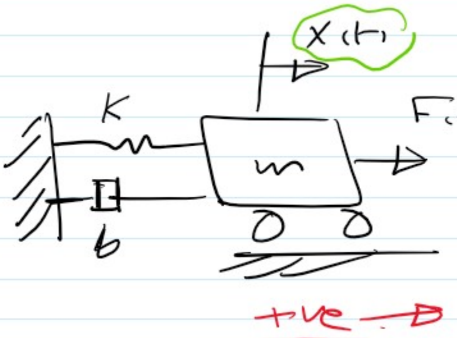
* Robot Dynamics :-

- So far, we studied static forces without motion.
- In this section, what are the forces/torques required to cause a motion ?

or what are the torques/forces of robot joints necessary to overcome dynamic effects ?

like inertia, gravity, friction ...

- The dynamic model of a manipulator provides a description of the relationship between joints torques/forces & the motion of joints given by positions, speeds & accelerations.
- Dynamic model is necessary for controllers design.



e.g.

$$\sum \vec{F} = m\vec{a}$$

$$F(t) - \underbrace{f_s}_{Kx} - \underbrace{f_d}_{b\dot{x}} = m\ddot{x}$$

$$m\ddot{x} + b\dot{x} + Kx = F(t)$$ eqn of motion

1 DOF

if multiple DOF ;

$$\underbrace{M\ddot{X}} + \underbrace{B\dot{X}} + \underbrace{KX} = F$$ on matrix form

How to find eqn of motion ?

(i) Newton's 2nd laws → $\sum \vec{F} = m\vec{a}$
→ $\sum \vec{M} = J\vec{\alpha}$

(ii) momentum-impulse method.

(iii) Lagrange approach (Energy Method)
→ a systematic method uses the Kinetic & potential energies of the system to generate the dynamic eqns.

⇒ The Lagrangian of a system (L) is given by :

$$\boxed{L = K - P}$$

where,
K: total Kinetic energy of the sys.
P: " potential " " "

are the joints variables

Now, let q_i ($i=1,\dots,n$) is the ith independant variable describing the dynamic of a system ;

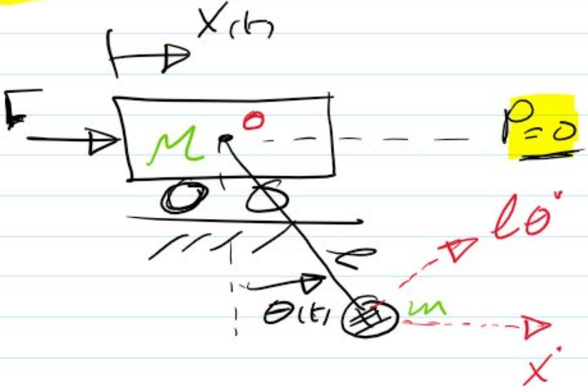
∴ The n differential equs are :

$$\boxed{\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = Q_i} \quad (i = 1, 2, \ldots n)$$

where Q_i is the external $\overset{\text{revolute}}{\text{torque}}$ / $\underset{\text{prismatic}}{\text{force}}$ applied to joint i

Ex.) { 2 independent variable x & θ.

→ 2 D.O.F



∴ we need two differential equs to describe the system Dynamics.

$$K = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m V^2$$

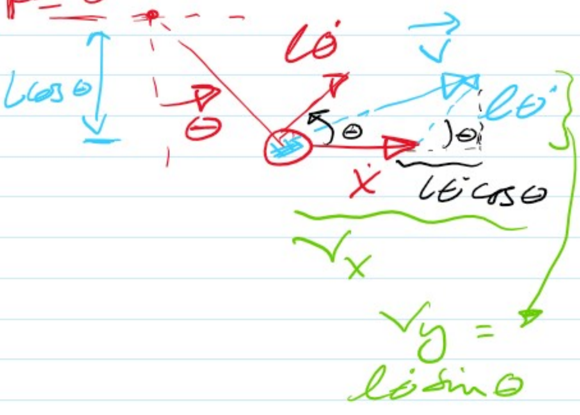
⇒ m will rotate around (O) & move with the cart;

for m;

$$\vec{V} = V_x \hat{i} + V_y \hat{j}$$

$$V = \sqrt{V_x^2 + V_y^2}$$

$$\therefore V^2 = V_x^2 + V_y^2$$



But, $$\begin{cases} v_x = \dot{x} + l\dot{\theta}\cos\theta \\ v_y = l\dot{\theta}\sin\theta \end{cases}$$

$$\text{so } v^2 = (\dot{x} + l\dot{\theta}\cos\theta)^2 + (l\dot{\theta}\sin\theta)^2$$

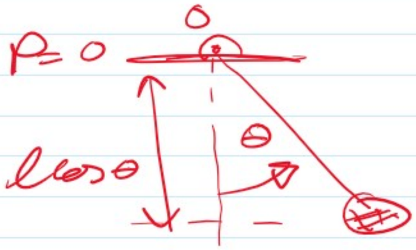
$$= \dot{x}^2 + 2l\dot{x}\dot{\theta}\cos\theta + (l\dot{\theta}\cos\theta)^2$$
$$+ (l\dot{\theta}\sin\theta)^2$$

$$v^2 = \underline{\dot{x}^2 + 2l\dot{x}\dot{\theta}\cos\theta + l^2\dot{\theta}^2}$$

$$\text{so } K = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m(\dot{x}^2 + 2l\dot{x}\dot{\theta}\cos\theta + l^2\dot{\theta}^2)$$

$$K = \underbrace{\frac{1}{2}(M+m)\dot{x}^2}_{\text{both masses are moving together with } \dot{x}} + \underbrace{\frac{1}{2}m(l\dot{\theta})^2}_{\text{only rotation of } m \text{ contribution}} + ml\dot{x}\dot{\theta}\cos\theta$$

& $P = -mgl\cos\theta$



Now,

$$L = K - P$$

$$L = \frac{1}{2}(M+m)\dot{x}^2 + \frac{1}{2}m(l\dot{\theta})^2 + ml\dot{\theta}\dot{x}\cos\theta$$
$$+ mgl\cos\theta$$

2 D.O.F ⇒ x & θ

for x ⇒ $$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F$$

$$\frac{\partial L}{\partial \dot{x}} = (M+m)\dot{x} + ml\,\underbrace{\dot{\theta}}\,\underbrace{\cos\theta}$$

so $$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) = (M+m)\ddot{x} + ml\ddot{\theta}\cos\theta - ml\dot{\theta}^2\sin\theta$$

& $$\frac{\partial L}{\partial x} = 0$$

--- (1)

$$\boxed{\text{so } (M+m)\ddot{x} + \underbrace{ml\cos\theta}\,\ddot{\theta} - \underbrace{ml\dot{\theta}^2\sin\theta}_{\text{normal accn.}} = F}$$

- coupling: effect of two masses on each other
- we must have similar term but with $\ddot{x}$ instead of $\ddot{\theta}$ when applying L in the θ-direction:

for θ ⇒ $$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0$$

$$\frac{\partial L}{\partial \dot{\theta}} = ml^2\dot{\theta} + ml\dot{x}\cos\theta$$

so $$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) = ml^2\ddot{\theta} + ml\ddot{x}\cos\theta - ml\dot{x}\dot{\theta}\sin\theta$$

$$\& \quad \frac{\partial L}{\partial \theta} = -ml\dot{\theta}\dot{x}\sin\theta - mlg\sin\theta$$

$$\therefore \; ml^2\ddot{\theta} + ml\cos\theta\,\ddot{x} - ml\dot{x}\cancel{\dot{\theta}}\sin\theta + ml\cancel{\dot{\theta}\dot{x}}\sin\theta + mlg\sin\theta = 0$$

$$\boxed{ml^2\ddot{\theta} + \underbrace{ml\cos\theta}_{\text{coupling: effect of } M \text{ on } m}\ddot{x} + mlg\sin\theta = 0} \quad --(2)$$

In matrix form [from eqns (1) & (2)];

$$\boxed{\begin{bmatrix} M+m & ml\cos\theta \\ ml\cos\theta & ml^2 \end{bmatrix}}\begin{pmatrix} \ddot{x} \\ \ddot{\theta} \end{pmatrix} + \begin{bmatrix} -ml\sin\theta\,\dot{\theta}^2 \\ mlg\sin\theta \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}$$

coupling forces between M & m.

Inertia-matrix

or Mass "

(M-matrix)

Notes) Manipulator robot has links (rigid bodies)

↳ To find Kinetic energy K, we need to find moment of inertia of each link around it's end (rotating Point)

each link around it's end (rotating Point)
using "Parallel-axis Theorem"

C: Center of mass of the link

O: rotating point

$$\therefore K = \frac{1}{2} I_o \dot{\theta}^2$$

$$\& \; I_o = \underbrace{I_c} + m\left(\frac{L}{2}\right)^2$$

$$I_o = \frac{1}{12} ml^2 + m\frac{l^2}{4}$$

$$I_o = \frac{1}{3} ml^2$$

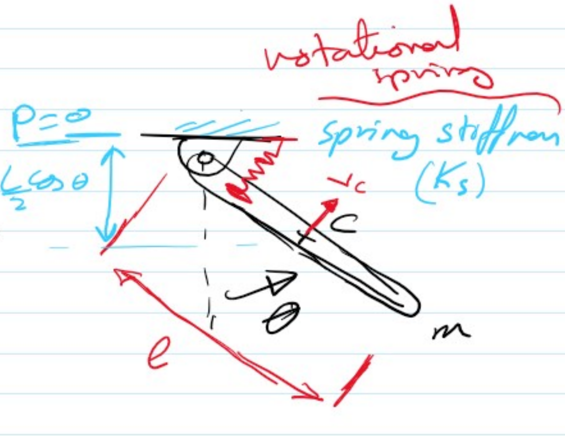
$$\therefore K = \frac{1}{6} ml^2 \dot{\theta}^2$$

→ Kinetic energy of rotating rod around it's end (O) is bigger by 4-times then when it is rotating around (C)

$$I_C = \frac{1}{12} ml^2$$

$$\left\{ K = \frac{1}{24} ml^2 \dot{\theta}^2 \right\}$$

Ex.1 1 DOF → θ



$$K = \frac{1}{2} m v_c^2 + \frac{1}{2} I_c \dot{\theta}^2$$

$$\left\{ \begin{array}{l} v_c = \frac{l \dot{\theta}}{2} \\ I_c = \frac{1}{12} ml^2 \end{array} \right.$$

$$= \frac{1}{6} ml^2 \dot{\theta}^2$$

$$P = \frac{1}{2} K_3 \theta^2 - mg \frac{L}{2} \cos\theta$$

or $\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0$ where

$\underline{L = K - P}$

Robotics

Chapter 6

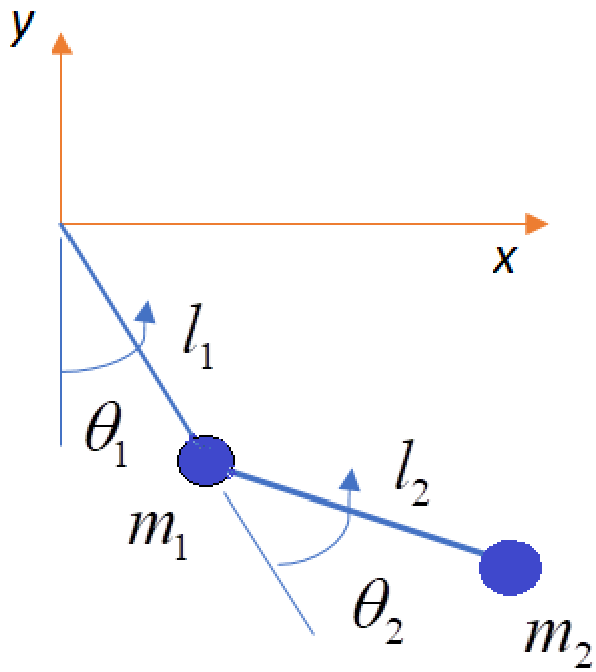
Manipulator Dynamics

Ahmad AL-Jarrah

Dynamics ….continue

Inertia Matrix and Jacobian

- For Complex systems, the kinetic energy is usually expressed in terms of Jacobians where the relation between the Cartesian space and the joint space of a manipulator is utilized.
- To explain the concept, assume the two-links manipulator shown where the mass for each link is assumed to be concentrated at the end of each link.



The position of the first mass with respect to the *xy*-coordinate is

$$x_1 = l_1 \sin\theta_1 \ \ and \ \ y_1 = -l_1 \cos\theta_1$$

Taking the first derivative results in the following velocities

$$\dot{x}_1 = l_1 \cos\theta_1 \;\; \dot{\theta}_1$$

$$\dot{y}_1 = l_1 \sin\theta_1 \;\; \dot{\theta}_1$$

In matrix format

$$\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix} = \underbrace{\begin{bmatrix} l_1 \cos\theta_1 & 0 \\ l_1 \sin\theta_1 & 0 \end{bmatrix}}_{J_1} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

J_1 : Jacobian of the first link with respect to the fixed frame

- Similarly, the position of the second mass with respect to the *xy*-coordinate is

$$x_2 = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

$$y_2 = -l_1 \cos\theta_1 - l_2 \cos(\theta_1 + \theta_2)$$

Taking the first derivative results in the following velocities

$$\dot{x}_2 = l_1 \cos\theta_1 \dot{\theta}_1 + l_2 \cos(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)$$

$$\dot{y}_2 = l_1 \sin\theta_1 \dot{\theta}_1 + l_2 \sin(\theta_1 + \theta_2)(\dot{\theta}_1 + \dot{\theta}_2)$$

In matrix format

$$\begin{bmatrix} \dot{x}_2 \\ \dot{y}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} l_1 c_1 + l_2 c_{12} & l_2 c_{12} \\ l_1 s_1 + l_2 s_{12} & l_2 s_{12} \end{bmatrix}}_{J_2} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$

J_2 : Jacobian of the second link with respect to the fixed frame

Noting that the Jacobian of the second link, J_2, is the Jacobian of the whole system since it is only two-links manipulator.

How to find the kinetic energy of the system?

- For m_1 :

$$k_1 = \frac{1}{2} m_1 v_1^2 \quad where \quad v_1 = \underbrace{\begin{bmatrix} \dot{x}_1 \\ \dot{y}_1 \end{bmatrix}}_{V_1 \text{ vector}} = J_1 \underbrace{\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}}_{\Theta \text{ vector}}$$

$$\underbrace{\therefore V_1}_{} = J_1 \dot{\Theta} \;\&\; \underbrace{v_1^2 = v_1^T v_1}_{}$$

Velocity vector corresponds to the first mass

Scalar quantity not vector

$$\therefore v_1^2 = \left[J_1 \dot{\Theta} \right]^T \left[J_1 \dot{\Theta} \right] = \dot{\Theta}^T J_1^T J_1 \dot{\Theta}$$

$$\& \quad k_1 = \frac{1}{2} m_1 \dot{\Theta}^T \left[J_1^T J_1 \right] \dot{\Theta}$$

$$Similarly \;\; for \; m_2 \Rightarrow k_2 = \frac{1}{2} m_2 \dot{\Theta}^T \left[J_2^T J_2 \right] \dot{\Theta}$$

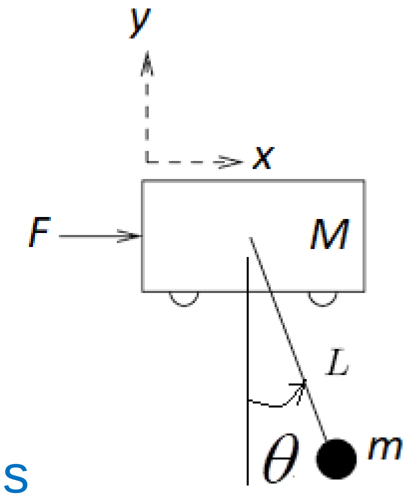
Hence, the total kinetic energy of the system is

$$k = k_1 + k_2 = \frac{1}{2} \dot{\Theta}^T \underbrace{\left[m_1 J_1^T J_1 + m_2 J_2^T J_2 \right]}_{\text{Scalar quantity}} \dot{\Theta}$$

As a conclusion, the inertia matrix can be given as

$$\boxed{M = m_1 J_1^T J_1 + m_2 J_2^T J_2 + \ldots + m_n J_n^T J_n}$$

In order to check the previous conclusion, let us find the inertia matrix of the previous example (shown) using Jacobian.



This system is 2 DOF ➤ Two motions of *M* and *m*

$$x, \theta$$

- The position of *M* with respect to the xy-fram is

$$P = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix}$$

$$and \;\; \dot{P} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} \dot{x} \\ 0 \end{bmatrix} \Rightarrow \;\; \therefore \;\; J_1 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$$

$$such\ that \begin{bmatrix} v_x \\ v_y \end{bmatrix} = J_1 \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix}$$

- The position of *m* with respect to the xy-frame is

$$P_m = \begin{bmatrix} p_{mx} \\ p_{my} \end{bmatrix} = \begin{bmatrix} x + l\sin\theta \\ -l\cos\theta \end{bmatrix}$$

$$and \;\; \dot{P}_m = \begin{bmatrix} v_{mx} \\ v_{my} \end{bmatrix} = \begin{bmatrix} \dot{x} + l\cos\theta\,\dot{\theta} \\ l\sin\theta\;\dot{\theta} \end{bmatrix} \Rightarrow \;\therefore\;\; J_2 = \begin{bmatrix} 1 & l\cos\theta \\ 0 & l\sin\theta \end{bmatrix}$$

$$such \; that \; \begin{bmatrix} v_{mx} \\ v_{my} \end{bmatrix} = J_2 \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix}$$

- Hence, the inertia matrix of the system is

$$\underbrace{M}_{\text{M-matrix}} = \underbrace{M}_{\text{Cart mass}} J_1^T J_1 + \underbrace{m}_{\text{pendulum mass}} J_2^T J_2$$

$$\therefore M = M\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix} + m\begin{bmatrix}1 & 0\\ l\cos\theta & l\sin\theta\end{bmatrix}\begin{bmatrix}1 & l\cos\theta\\ 0 & l\sin\theta\end{bmatrix}$$

$$= \begin{bmatrix}M & 0\\ 0 & 0\end{bmatrix} + \begin{bmatrix}m & ml\cos\theta\\ ml\cos\theta & ml^2\end{bmatrix}$$

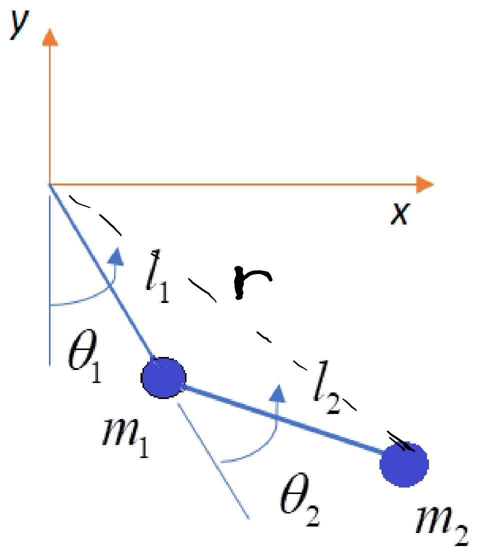
$$= \boxed{\begin{bmatrix}M+m & ml\cos\theta\\ ml\cos\theta & ml^2\end{bmatrix}}$$

Same M-matrix found in class last lecture using the Lagrangian of the system

$$= \boxed{\begin{bmatrix}m_{11} & m_{12}\\ m_{21} & m_{22}\end{bmatrix}}$$

One can notice the coupling forces between the two masses where $m_{12} = m_{21}$ Associated with the corresponding acceleration of each mass

Two-links example continue…



$$\therefore M = m_1 J_1^T J_1 + m_2 J_2^T J_2$$

$$= m_1 \begin{bmatrix} l_1c_1 & l_1s_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} l_1c_1 & 0 \\ l_1s_1 & 0 \end{bmatrix} +$$

$$m_2 \begin{bmatrix} l_1c_1+l_2c_{12} & l_1s_1+l_2s_{12} \\ l_2c_{12} & l_2s_{12} \end{bmatrix} \begin{bmatrix} l_1c_1+l_2c_{12} & l_2c_{12} \\ l_1s_1+l_2s_{12} & l_2s_{12} \end{bmatrix}$$

$$= \begin{bmatrix} m_1l_1^2+m_2r^2 & m_2(l_1l_2c_2+l_2^2) \\ m_2(l_1l_2c_2+l_2^2) & m_2l_2^2 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

Coupling forces/torques

Effective moment of inertia seen at each joint

Noting that ➤

$$r = l_1^2 + l_2^2 + 2l_1l_2c_2$$

$$m_{11} = m_1l_1^2 + m_2r^2$$

$$m_{12} = m_2(l_1l_2c_2 + l_2^2)$$

$$m_{21} = m_2(l_1l_2c_2 + l_2^2)$$

$$m_{22} = m_2l_2^2$$

And

$m_1l_1^2$: is the moment of inertia of m_1 seen at the fixed frame (i.e. 1st motor)

m_2r^2 : is the moment of inertia of m_2 seen at the fixed frame (i.e. 1st motor)

$m_2l_2^2$: is the moment of inertia of m_2 seen at m_1 (i.e. 2nd motor or joint)

Important Remarks:

m_{11} : is the moment of inertia of the manipulator seen at the 1st motor and it depends on the following joints configurations $\theta_2 \ldots \theta_n$.

m_{22} : is the moment of inertia of the manipulator seen at the 2nd motor and it depends on the following joints configurations $\theta_3 \ldots \theta_n$.

.

.

.

m_{nn} : is the moment of inertia of the end-effector and it doesn't depend on any joints configurations (i.e. fixed inertia).

m_{12} & m_{21} : are same terms and show the coupling forces/torques between m_1 and m_2 .

(i.e. effect of acceleration of joint$_2$ on joint$_1$)

Dynamic equation related to 1st joint

$$\left\{ m_{11}\ddot{\theta}_1 + m_{12}\ddot{\theta}_2 = \tau_1 \right.$$

The torque of the first joint (motor)

How to find the Kinetic energy of the system after finding the M-matrix?

The total kinetic energy for a manipulator after finding the M-matrix will be

$$k = \frac{1}{2}\begin{bmatrix}\dot{\theta}_1 & \dot{\theta}_2 \ldots\ldots \dot{\theta}_n\end{bmatrix} M \begin{bmatrix}\dot{\theta}_1 \\ \dot{\theta}_2 \\ . \\ . \\ \dot{\theta}_n\end{bmatrix}$$

In general, the dynamics of a manipulator will be in the following form

$$\underbrace{M(\Theta)}_{\substack{\text{M-matrix}\\ \text{depends on}\\ \text{the robot}\\ \text{configuration}}}\ddot{\Theta} + \underbrace{\underbrace{C(\dot{\theta}_i\dot{\theta}_j)}_{\substack{\text{Coriolis}\\ \text{term}}} + \underbrace{N(\dot{\theta}_i^2)}_{\substack{\text{Normal force}\\ \text{due to}\\ \text{normal}\\ \text{acceleration}}} + \underbrace{\mathrm{G}(\Theta)}_{\substack{\text{Gravity}\\ \text{term}}}}_{\text{Nonlinear terms that can be}} = \underbrace{\mathrm{T}}_{\substack{\text{torques/forces}\\ \text{of the}\\ \text{Motors for}\\ \text{each joint}}}$$

- Ignored in some cases while designing the control law
- Or can be compensated for
- Or can be overcomed using high gains of the control law as we will see later!

Important Note :

- In order to simulate the motion of a manipulator and be able to design a controller, it is very important to solve the dynamic equation of the manipulator for acceleration.

- Assuming the following manipulator dynamical model

$$M(\Theta)\ddot{\Theta} + \boxed{B} = \mathrm{T}$$

Equivalent to the nonlinear terms

Solving the dynamic equation for acceleration requires

$$\ddot{\Theta} = \underbrace{M^{-1}}_{}(\Theta)(\mathrm{T} - B)$$

Invertible Inertia matrix for the sake of controller design and simulation

Robotics

Chapter 7

Trajectory Generation

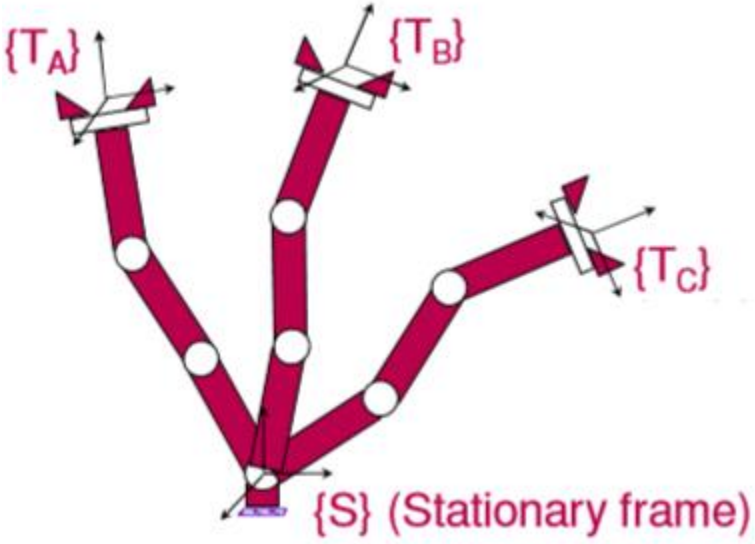
Ahmad AL-Jarrah

Trajectory Generation

- Trajectory is often viewed as a combination of a path, which is a purely geometric description of the sequence of configurations achieved by the robot, and a time scaling, which specifies the times when those configurations are reached.
- Trajectory Generation: Construct a trajectory (path + time scaling) so that the robot reaches a sequence of points in a given time.
- Also, trajectory generation refers to the time history of position, velocity, and acceleration for each joint (DOF).

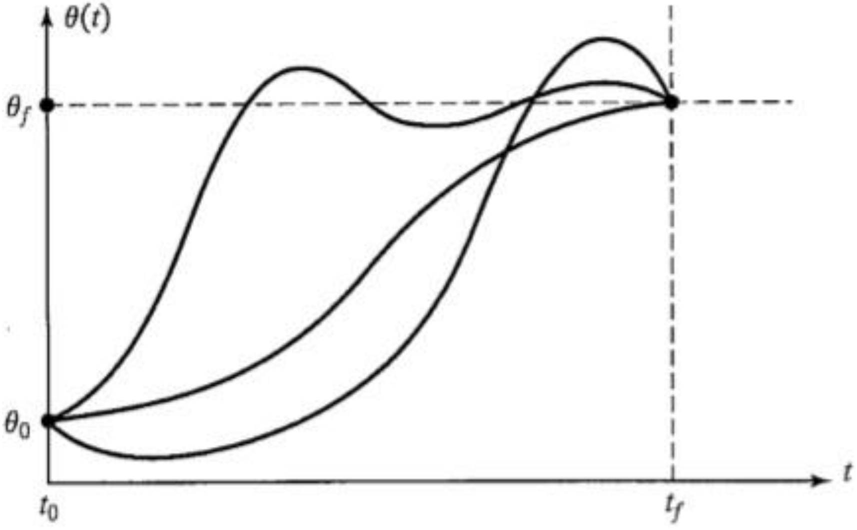
Basic Problem:

The figure shows moving the manipulator tool (end-effector) from position A to position C with respect to the stationary frame within a desired period of time.



Once the Cartesian poses are known (i.e. A and C positions), we solve the inverse kinematics problem to find the desired configuration for each joint required to reach the final desired pose at C.

After finding the desired initial and final configuration for each joint (i.e. Joint-space), a path planning is required to move between both configurations which can be achieved through different paths for each joint as shown in the figure.



It is very important to be consider that moving the manipulator tool, in the Cartesian-space, from A to C could go through some via points like the point B shown in the pervious slide.

The via points are intermediate points which could be obstacles and that requires “Obstacle avoidance planning”

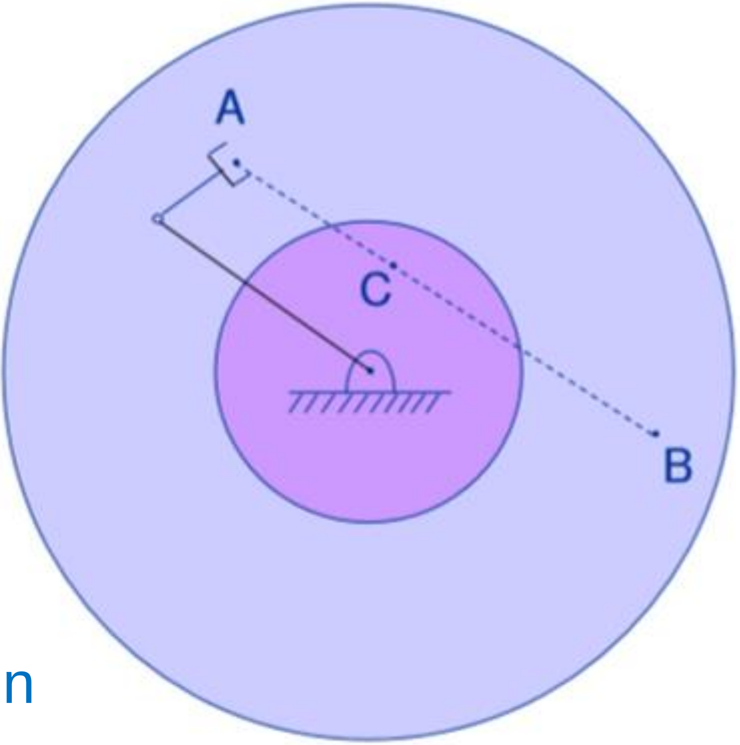
From the previous, the conclusion is:

Given the path points (Initial, final, and via points), we need to construct a trajectory for each joint so that the robot reaches a sequence of points in a given time.

Noting that: “it is required to move smoothly in the joint-space”.

Important Notes: Geometric Difficulties

- The reachable workspace of the robot (the blue zone).
- The initial point A and the final point B are reachable.
- The robot will not be able to move in a straight line since the intermediate point C is not reachable.
- Sometimes, for certain paths, it is impossible for the manipulator to perform.



- ➤ Near singularity configuration
- ➤ High joints rates close to infinity

Polynomials (splines) for path planning

- Knowing initial and final configurations leads to a straight-line path planning choice, as shown in the figure below, using the following equation:

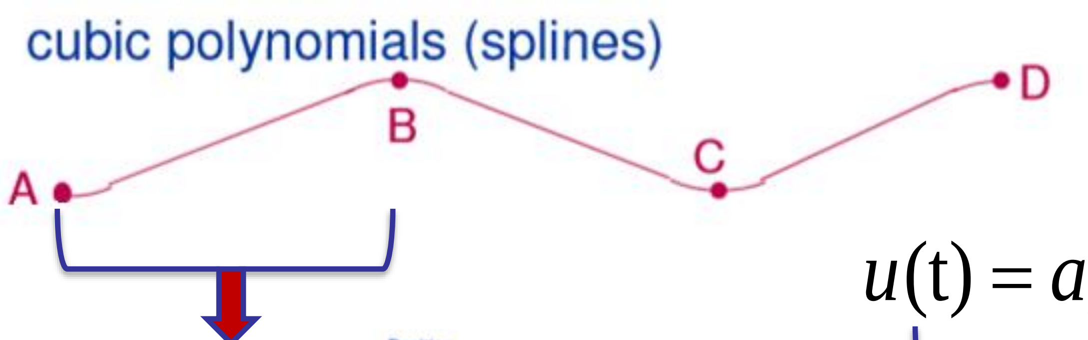
$$\underbrace{u(\mathrm{t}) = a_o + a_1 t}$$

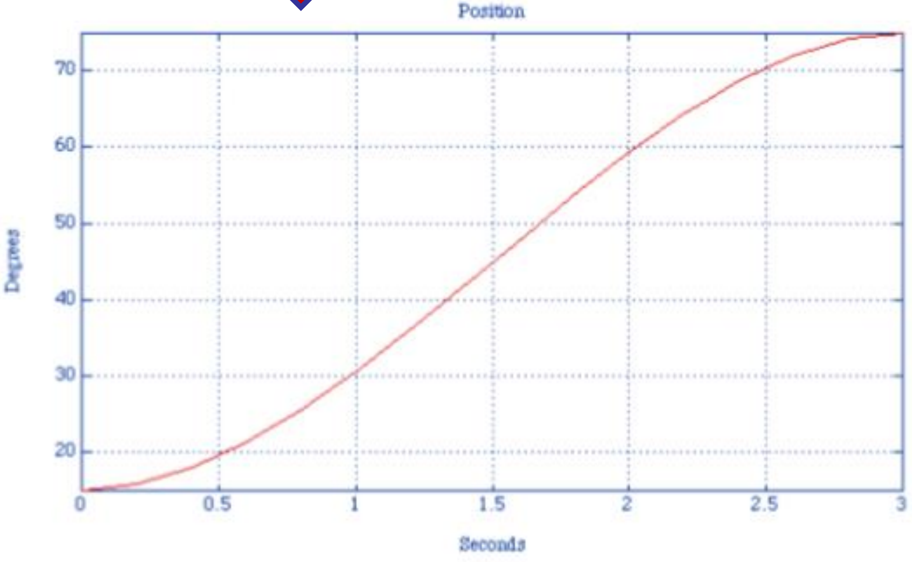
Where *u(t)* could be joint-space variable (i.e. joint position) or Cartesian-space variable like x-position.

Two unknown parameters that can be found using the known initial and final configurations $\theta(\mathrm{t}_o) = \theta_o$ & $\theta(\mathrm{t}_f) = \theta_f$ where the velocity is not involved (i.e. zero) and hence not controlled.

- The figure shows, moving and stopping at intermediate points to reach the final position at D using straight lines; this produces a discontinuous velocity (i.e. Jerk motion which causes system damages and more energy consumption)

- To avoid jerk motions and guarantee smoothness, one can use polynomials.
- Cubic polynomial will control the velocity at the beginning and at the end of each segment shown in the figure and hence guarantee smoot motion.





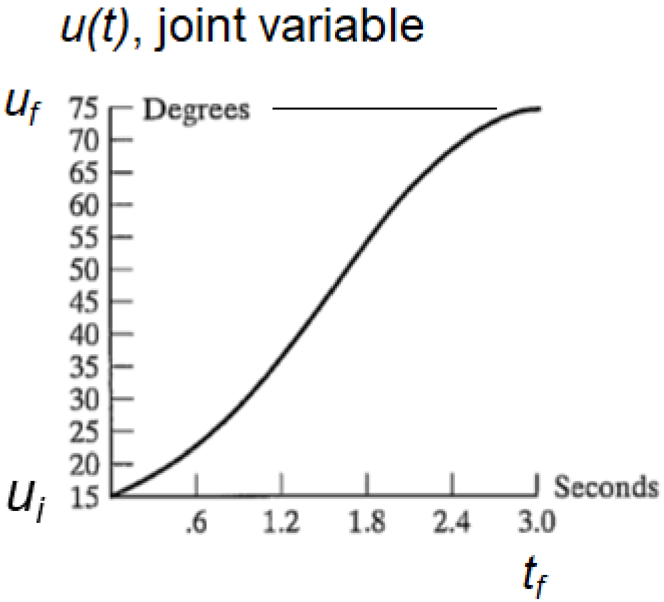
$$\underbrace{u(\mathrm{t}) = a_o + a_1 t + a_2 t^2 + a_3 t^3}$$

Cubic polynomial equation with Four unknown parameters that can be found using the known initial and final configurations and velocities (i.e. non zero velocities) .

∴ The following cubic polynomial path can be used for each joint position

$$u(\mathrm{t}) = a_o + a_1 t + a_2 t^2 + a_3 t^3$$

where the four known initial and final conditions for the configuration and velocity are:



$$u(0) = u_i \quad (\mathrm{i.e.}\ \theta_i)$$

$$u(\mathrm{t}_f) = u_f$$

$$\left.\begin{array}{l} \dot{u}(0) = \dot{u}_i \ \text{(could be zero)} \\ \dot{u}(\mathrm{t}_f) = \dot{u}_f \ \text{(could be zero)} \end{array}\right\}$$

Non zero velocities at t_f (or end of each segment) will result smooth motion with continuity.

The position and velocity equations after taking the first derivative of *u(t)* are:

$$u(\mathrm{t}) = a_o + a_1 t + a_2 t^2 + a_3 t^3$$

$$\dot{u}(\mathrm{t}) = a_1 + 2a_2 t + 3a_3 t^2$$

Substituting the previous given four conditions into $u(\mathrm{t})$ and $\dot{u}(\mathrm{t})$ results

$$u(0) = u_i \Rightarrow a_o = u_i$$

$$\dot{u}(0) = \dot{u}_i \Rightarrow a_1 = \dot{u}_i$$

and

$$\left.\begin{aligned} u(\mathrm{t}_f) &= u_f = u_i + \dot{u}_i t_f + a_2 t_f^2 + a_3 t_f^3 \\ \dot{u}(\mathrm{t}_f) &= \dot{u}_f = \dot{u}_i + 2a_2 t_f + 3a_3 t_f^2 \end{aligned}\right\}$$

Two equations with two unknowns to be solved for

a_2 and a_3

$$\therefore \quad a_o = u_i$$

$$a_1 = \dot{u}_i$$

$$a_2 = \frac{3}{t_f^2}(u_f - u_i) - \frac{2}{t_f}\dot{u}_i - \frac{1}{t_f}\dot{u}_f$$

$$a_3 = \frac{2}{t_f^3}(u_f - u_i) + \frac{1}{t_f^2}(\dot{u}_f + \dot{u}_i)$$

Notes:

1. Similar to the straight-line choice where no control over the velocity, the acceleration is not involved in the cubic polynomial choice and hence it is not controlled.

➡ Higher order polynomial with more parameters is required to control the acceleration

2. 5th order polynomial is a good candidate to get the acceleration conditions involved and hence controlled acceleration.

$$u(\mathrm{t}) = \underbrace{a_o + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5}$$

6 unknown parameters requires 6 conditions

i.e.

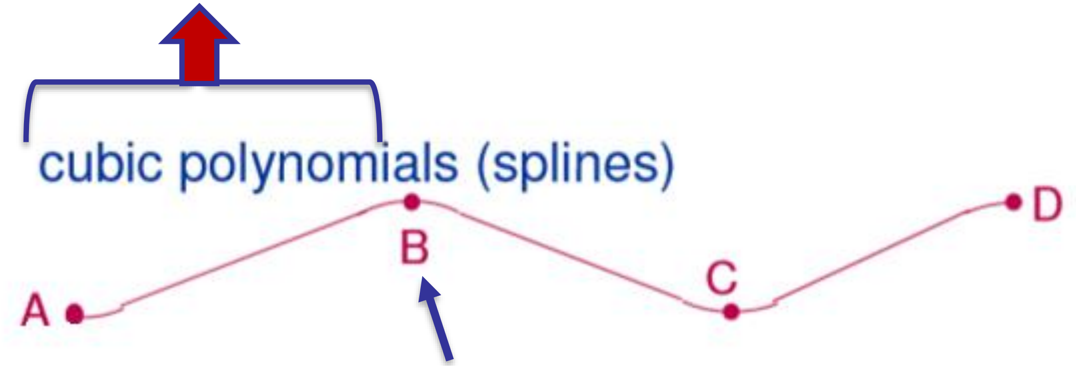
$$u(0) = u_i \, , \; u(\mathrm{t}_f) = u_f$$

$$\dot{u}(0) = \dot{u}_i \, , \; \dot{u}(\mathrm{t}_f) = \dot{u}_f$$

$$\ddot{u}(0) = \ddot{u}_i \, , \; \ddot{u}(\mathrm{t}_f) = \ddot{u}_f$$

3. To guarantee smooth motion when moving from one segment to another (refer to the figure in the next slide), the velocity at the end of the first segment can be selected to be the initial velocity of the coming segment and so on for all segments constructing the robot path.

From A to B is one segment



One way to get smooth motion when finishing the first segment and start the second one at point B is choosing same velocity values

i.e. $\boxed{\dot{u}_{1f}(t_{1f})} = \boxed{\dot{u}_{2i}(0)}$

Final velocity at the end of the first spline at B

Initial velocity at the beginning of the second spline at B

* Control of Manipulators :-

- There is no unique way to control amanipulater.

τ

- We are looking for the required torques of each joint to realize the desired trajectory.

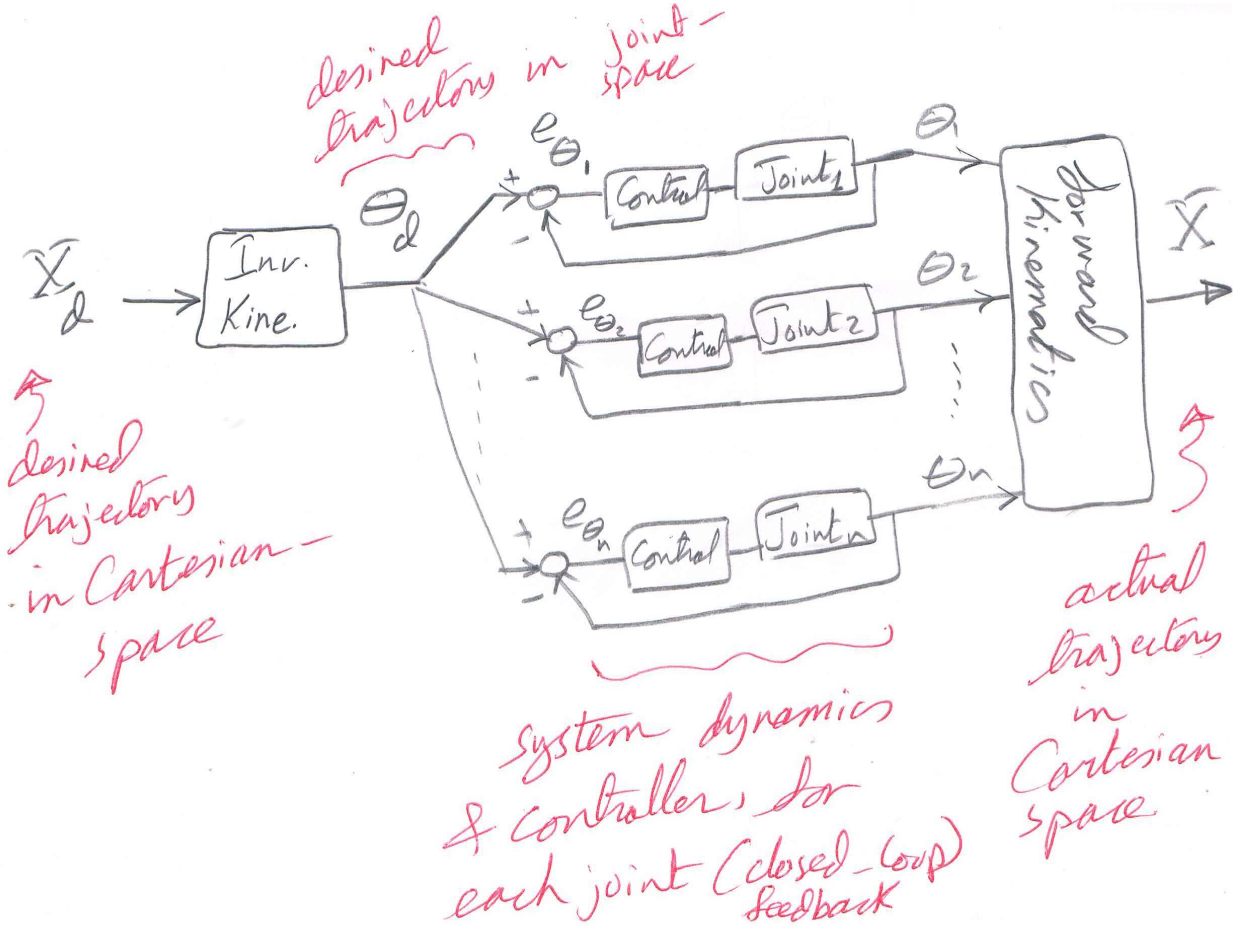
i.e open-loop control

⇒ we are given θ_d, $\dot{\theta}_d$ & $\ddot{\theta}_d$, by the trajectory generator & system dynimics, we calculate $\underline{\underline{\tau}}$.

$$\hookrightarrow \tau = M(\underset{\sim}{\theta_d})\underset{\sim}{\ddot{\theta}_d} + C(\underset{\sim}{\dot{\theta}_{i_d}}\underset{\sim}{\dot{\theta}_{j_d}}) + N(\underset{\sim}{\dot{\theta}^2_{i_d}}) + G(\underset{\sim}{\theta_d}$$

However, imperfection in the dynamics Model (nonlinearities, unmodeled elements, disturbances, uncertainities) make such scheme impractical in real applications

⇒ Closed-loop scheme !



desired trajectory in joint-space

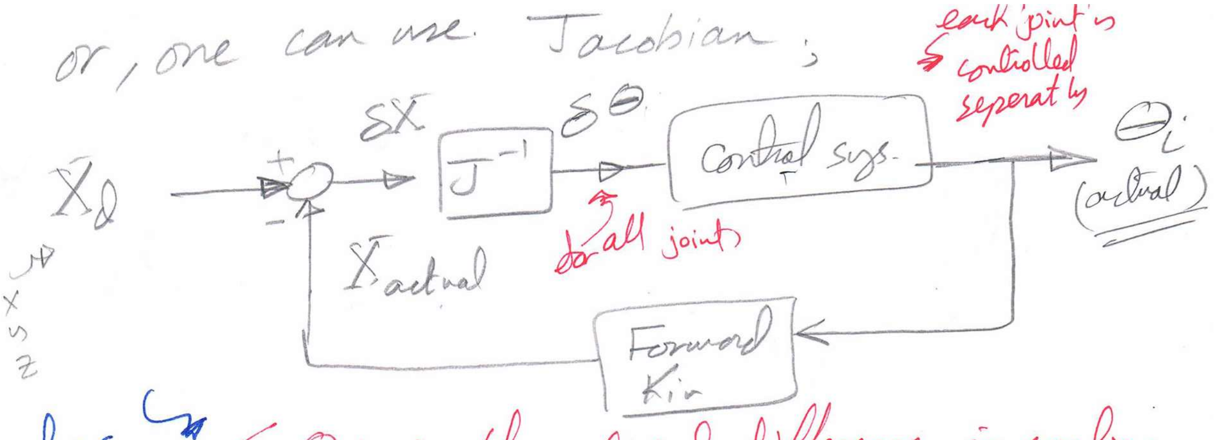
desired trajectory in Cartesian-space

system dynamics & controller, for each joint (closed-loop feedback

actual trajectory in Cartesian space

- The problem is in Inverse Kinematics & Singularities.

or, one can use Jacobian;



for all joints ↳ $\delta\Theta_i$ is the desired difference in config. to move to the new config. from the current one Θ_i

i.e. $\delta \bar{X} = X_d - X$ (error)

find $\delta \Theta = J^{-1} \delta X$

for all joints

Then ⇒ $$\underbrace{\Theta_{new}}_{\text{new desired config.}} = \underbrace{\Theta_{old}}_{\text{old config.}} + \delta\Theta$$

- This approach is called "Resolved motion rate control"

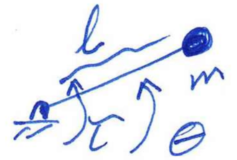
↳ it is appropriate for small motion since of J^{-1} ((singularities)) from dynamics eqn. of motion, the eqn of 1st joint for examples can be;

$$m_{11}\ddot{\theta}_1 + \underbrace{m_{12}\ddot{\theta}_2 + m_{13}\ddot{\theta}_3}_{\text{coupling}} + \underbrace{\text{non-linear terms}}_{\Downarrow} = \tau_1$$

- coriolus
- centerifugal
- gravity
- friction

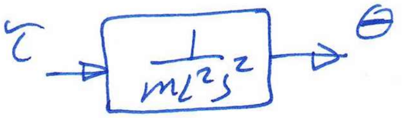
- if we take one link;

assume horizontal plane with no friction



i.e. ⇒ $\tau = J\ddot{\theta}$, $J = ml^2$

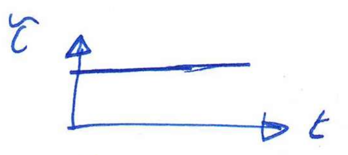
$\mathcal{L}$ @ 0 IC's;



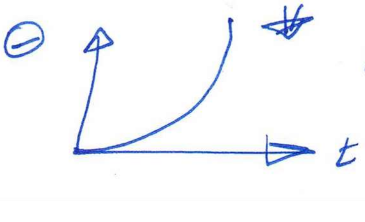
open-loop

$$\frac{\theta(s)}{\tau(s)} = \frac{1}{ml^2 s^2}$$

if $\tau(s) = \frac{A}{s}$ (step input)



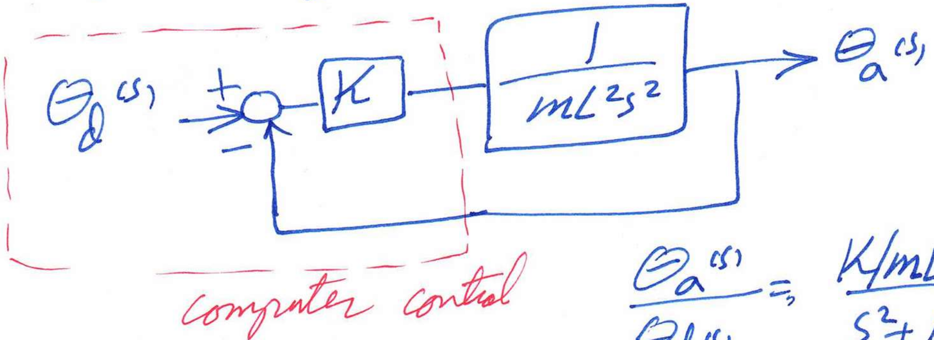
$\Downarrow$



$$\theta(t) = \mathcal{L}^{-1}\frac{A}{ml^2 s^3} = \frac{A}{ml^2}t^2$$

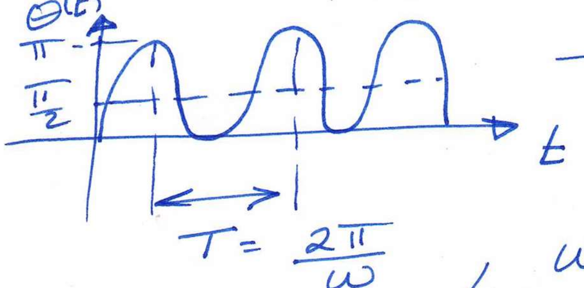
unstable
(Keep increasing θ to ∞.

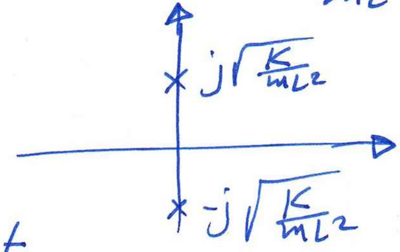
- we can use position feedback to Stablize the sys.



$$\frac{\Theta_a(s)}{\Theta_d(s)} = \frac{K/mL^2}{s^2 + \frac{K}{mL^2}}$$

if $\Theta_d(s)$ is step input $\frac{\pi}{2}$



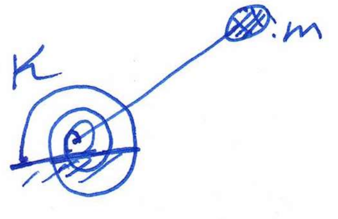


$T = \frac{2\pi}{\omega}$, $\omega = \sqrt{\frac{K}{mL^2}}$

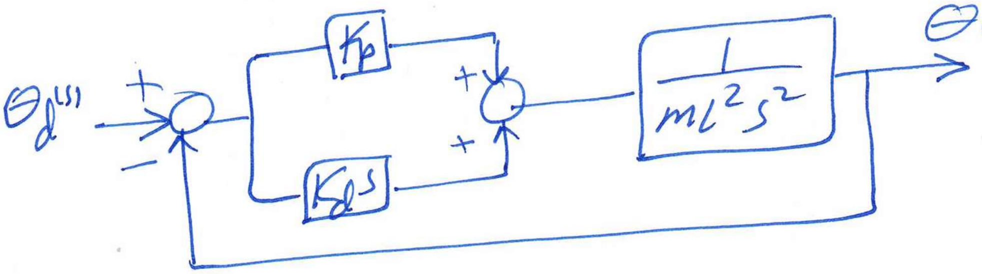
$K\uparrow \rightarrow \omega\uparrow \rightarrow T\downarrow$ (more freq.) & high osci.

using the gain K is equivalent to adding artificial spring to the system ⇒ more stiff.

proportional controller. ≡



- to eliminate oscillations, use PD-control action ∴



$$\frac{\Theta(s)}{\Theta_d(s)} = \frac{K_d s + K_p}{mL^2 s^2 + K_d s + K_p = \Delta} \text{ : characteristic eqn.}$$

$$s^2 + \frac{K_d}{mL^2} s + \frac{K_p}{mL^2} = 0 \equiv s^2 + 2\zeta\omega_n s + \omega_n^2$$

usually $\zeta = 1 \Rightarrow 2\zeta\omega_n = \frac{K_d}{mL^2}$

$$\omega_n^2 = \frac{K_p}{mL^2}$$

let $mL^2 = M$

$$\left.\begin{aligned} \therefore K_d &= 2M\omega_n \\ K_p &= M\omega_n^2 \end{aligned}\right\} \text{set } \omega_n \text{ like } \zeta, \text{ find } K_p \text{ \& } K_d$$

$$\omega_n \leq \frac{\omega_{resonant}}{2} \hookrightarrow$$

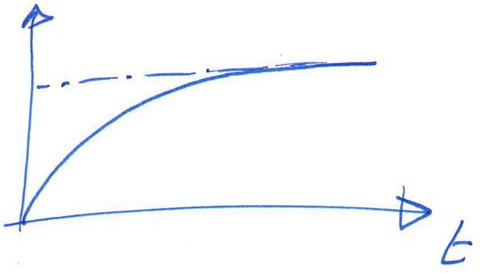
lowest structural flexibility

if $m=1$ Kg, $L=1$ m, design a PD-controller to have $\zeta=1$ & $\omega_n=10$.

$$2\zeta\omega_n = \frac{K_d}{mL^2} = K_d = 2*10=20$$

$$\& \quad \omega_n^2 = \frac{K_p}{mL^2} \Rightarrow K_p = \omega_n^2 = 100$$

⇒ once again, choosing ω_n depends on structural flexibities of the sys.



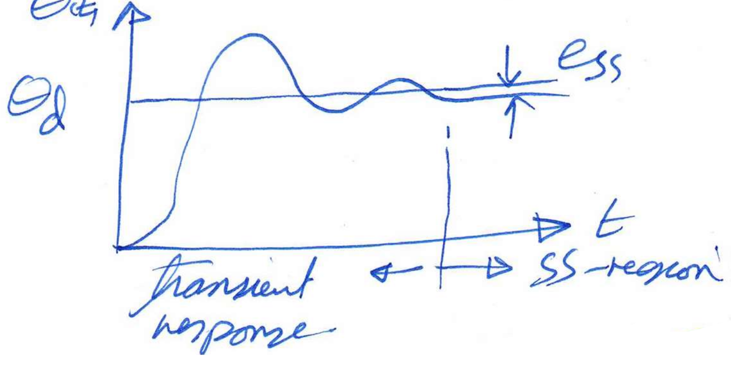
↳ very high K_p might lead to unstability by hiting the resonance.

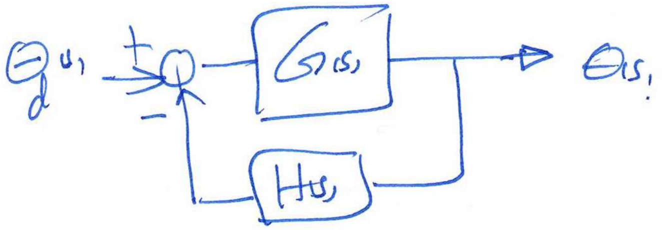
↳ i.e vibration modes. &

always stay away!

we always to choose K v.high

⇒ performance specifications:





$$E(s) = \theta_d(s) - \theta(s)$$

$$= \theta_d(s)\left[1 - \frac{\theta(s)}{\theta_d(s)}\right]$$

$$= \theta_d(s)\left[1 - \frac{G(s)}{1+G(s)H(s)}\right]$$

$$\frac{E(s)}{\theta_d(s)} = \frac{1}{1+G(s)} \quad \text{for } H(s)=1$$

$$e_{ss} = \lim_{s\to 0} sE(s)$$

$$\boxed{\text{if } K G'(s) = G(s) \qquad K\uparrow \to e_{ss}\downarrow}$$

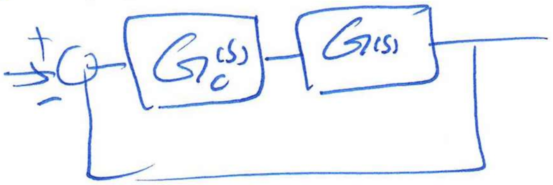
Sensitivity :

$$S^T_G = \frac{\boxed{\Delta T(s) / T(s)}}{\boxed{\Delta G(s) / G(s)}}$$

→ percentage change in sys-T.F

→ percentage change in process T.F

for small incremental changes → sensitivity of closed-loop T.F wrt small changes in $G(s)$



$$S^T_G = \frac{\partial T}{\partial G}\;\frac{G}{T}$$

$$T(s) = \frac{G_c G}{1+G_c G}$$

so

$$\boxed{S^T_G = \frac{1}{1+G_c(s)\,G(s)}}$$

$S^T_G \downarrow$ if $G_c(s) = K\uparrow$

in transient response;

$$t_s = \frac{4}{\zeta \omega_n}, \quad M_p\% = 100\, e^{\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}}\% \quad \text{for } 0 < \zeta < 1$$

could be used in design

for 2% critiria

- The error can be eliminated by adding an I-action to the PD;

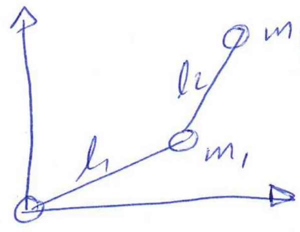
$$PID = K_p + K_d s + \frac{K_i}{s}$$

$$= \frac{K_d s^2 + K_p s + K_i}{s} \equiv K \frac{(s+z_1)(s+z_2)}{s}$$

⇒ So far, we have learnt how to design a controller for LTI system. If we know,

$$\underbrace{M(\theta)}_{\text{is changing!}} \ddot{\theta} + \underbrace{V(\theta, \dot{\theta}) + G(\theta)}_{\text{non-linear terms}} = \tau$$

$$M = \begin{bmatrix} m_1 L_1^2 + m_2(L_1^2 + l_2^2 + 2L_1 l_2 c_2) & - \\ - & m_2 L_2^2 \end{bmatrix}$$

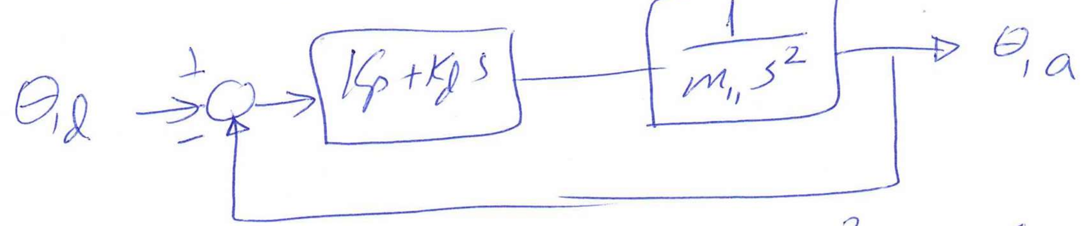


i.e. $M = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$

m_{11} is the inertia of robot seen @ joint 1 or motor 1. & it is time varient (non linear) because of θ_2.

one may
$\Rightarrow$ to design a controller (linear) of θ_1 is using the 1st entry of matrix M & ignor the non-Linear terms.

$$\tau_1 = m_{11}\ddot{\theta}_1$$



$$\Delta = m_{11} s^2 + K_d s + K_p \equiv s^2 + 2\zeta\omega_n s + \omega_n^2$$

$$\zeta = \frac{K_d}{2 m_{11}\omega_n} \quad \& \quad \omega_n = \sqrt{\frac{K_p}{m_{11}}}$$

In Robotics, we want the system to be critically damped or overdamped ($\zeta \geq 1$).

TO ensure that the system is overdamped, the selection of Kp & Kd should be based on ($m_{11\,max}$) which is when the robot is fully extendes ($\theta_2 = 0$).

$$\therefore \; m_{11\,max} = m_1 L_1^2 + m_2 (L_1 + L_2)^2 \;\Big|\; \text{linearized}$$

⇒ to decide for ω_n & select for Kp & then for Kd, we need a criteria that involves the closed loop desired natural freq. (ω_n) like t_s

Known (Desired) ⇝ $t_s = \frac{4}{\zeta \omega_n}$ & set $\zeta = 1$.

$$\therefore \; \omega_n = \frac{4}{t_s} = \sqrt{\frac{K_p}{m_{11(max)}}}$$

$$\boxed{K_p = \frac{16}{t_s^2} m_{11\,max}}$$

$$\& \; K_d = 2 m_{11\,max}\, \omega_n = \boxed{2 \sqrt{m_{1\,max} K_p}}$$

* Non-Linear Control of Robot :-

- since we will have different m's;
meaning, light link & heavy link
↳ the gain will be big.

back to ⇒ $\left\{\begin{array}{l} K_d = 2 m w_n \\ K_p = m w_n^2 \end{array}\right\}$ for $\zeta = 1$

{ for same desired closed-loop frequency w_n, we will have different gain values based on m-value of the link. }

i.e Scaling the gain by "m"

↳ so we update the gains as "m" changing based on configuration

so we can set the gains for Unit-mass system to be;

$$\boxed{\begin{array}{l} K_p' = w_n^2 \\ K_d' = 2 w_n \end{array}}$$

the prime is for unit-mass system which will not be f

∴ we update the gains by scaling the gains by the real m (as it changes)

$$\left.\begin{aligned} \therefore K_p &= m K_p' \\ K_d &= m K_d' \end{aligned}\right\} \text{ for } m\text{-mass system}$$

i.e m is going to change ✗ & by updating the gains (non-linear control), we force the whole system to have same desired closed loop freq. (ω_n) & damping ratio ζ ($\zeta = 1$)

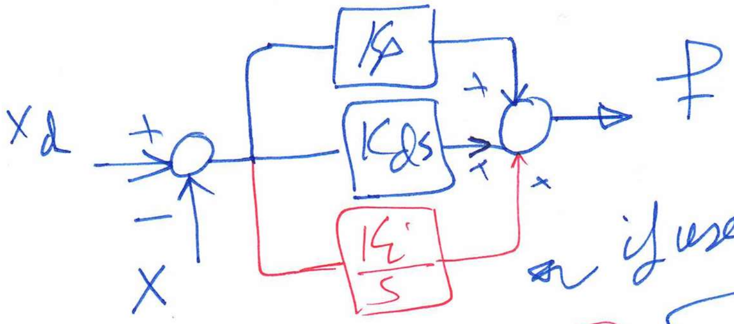
✗ This is important since same concept is applicable for M-matrix

i.e Non-Linear

we will be compensating for the variation in the mass.

∴ we design for unit-mass system and then we scale by M-matrix & (ζ, ω_n) will be same for the sys.

if $\underline{m\ddot{x} = f}$ ⇒ f is the control signal



if used!

$\underline{m}(1.\ddot{x}) = \underline{m}f'$

scaling

$\boxed{f' = K_p'(x_d - x) + K_d'(\underset{\text{could be zero}}{\dot{x}_d} - \dot{x})}$

which will be $\underline{\underline{M}}$_matrix

$f = m f'$,

$\boxed{1\ddot{x} = f' \text{ unit-mass system}}$

i.e. This system is controlled as a unit-mass scaled by m, hence, the dynamic behaviour is like $\boxed{\ddot{x} = f'}$ with $\omega_n = \sqrt{K_p'}$ & $\zeta = 1$

i.e $1\ddot{x} + \underset{2\zeta\omega_n}{K_d'}\dot{x} + \underset{\omega_n^2}{K_p'}(x - x_d) = 0$

if we introduce non-Linear terms like friction or gravity or ....;

let; $m\ddot{x} + \underbrace{b(x,\dot{x})}_{\text{non-Linear friction}} = f$

or assume the Known system of;

$$m\ddot{x} + \underbrace{b\dot{x} + Kx} = f$$

non-Linear terms that can be compensated for if we are able to anticipate it or estimate it & add to the control law to be non-Linear controller

$\downarrow$

i.e control partitioning (decoupling)

so control signal $\Rightarrow \boxed{f = \alpha f' + \beta}$

where; $\alpha = m$ (could be $\hat{m}$)

all are estimated if unknown $\Rightarrow$ $\left\{ \begin{array}{l} \beta = b(x,\dot{x}) \text{ (or } \hat{b}\text{)} \\ \text{or } = b\dot{x} + Kx \end{array} \right.$ (for simple spring-mass-damper syst i.e. we compensate for undesired or non-linear terms to force the system behave as a unit-mass sys. (Linearized)

v. close to each other

$\therefore\ m\ddot{x} + b\dot{x} + Kx = \alpha f' + \beta$ / $\begin{cases} \beta = b\dot{x} + Kx \\ \alpha = m \end{cases}$

$\cancel{m}\ddot{x} + b\cancel{\dot{x}} + \cancel{Kx} = \cancel{m}f' + b\cancel{\dot{x}} + \cancel{K}x$

$\boxed{\therefore 1\ \ddot{x} = f'}$

⇒ or, for;

$m\ddot{x} + b(x, \dot{x}) = f$

& $f = \alpha f' + \beta$

where, $\alpha = \hat{m}$

$\beta = \hat{b}(x, \dot{x})$

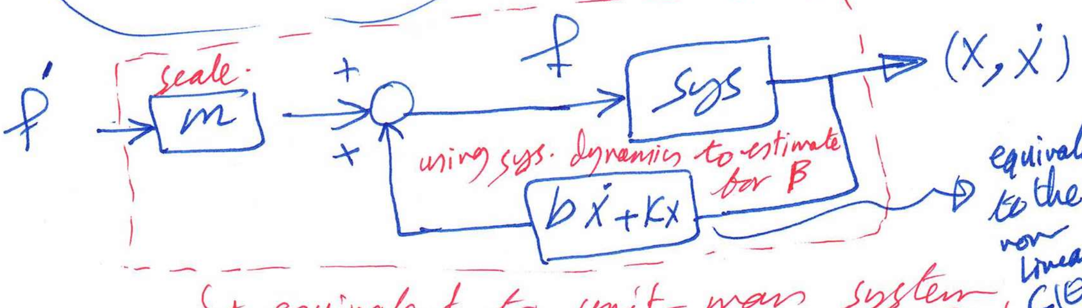
⇒

$m\ddot{x} + b(x, \dot{x}) = \hat{m}f' + \hat{b}(x, \dot{x})$

⇒ $1\ddot{x} = f'$

if $b \simeq \hat{b}$

$m \simeq \hat{m}$



⇒ equivalent to the non linearite $C(\Theta, \dot{\Theta})$ $G(\Theta)$

⇒ equivalent to unit-mass system behaviour.

i.e computed torque

∴ The controller includes the dynamics $(x, \dot{x})$ to estimate for non-Linearities & compensate for them & decouple M-matrix to have unit mass sys. behaviour.

⇒ if we track a trajectory i.e $\underline{(x_d, \dot{x}_d, \ddot{x}_d)}$

$$f' = \ddot{x}_d + K_d'(\dot{x}_d - \dot{x}) + K_p'(x_d - x)$$

"& $\ddot{x}_d$ could be zero"

if $e = x_d - x$

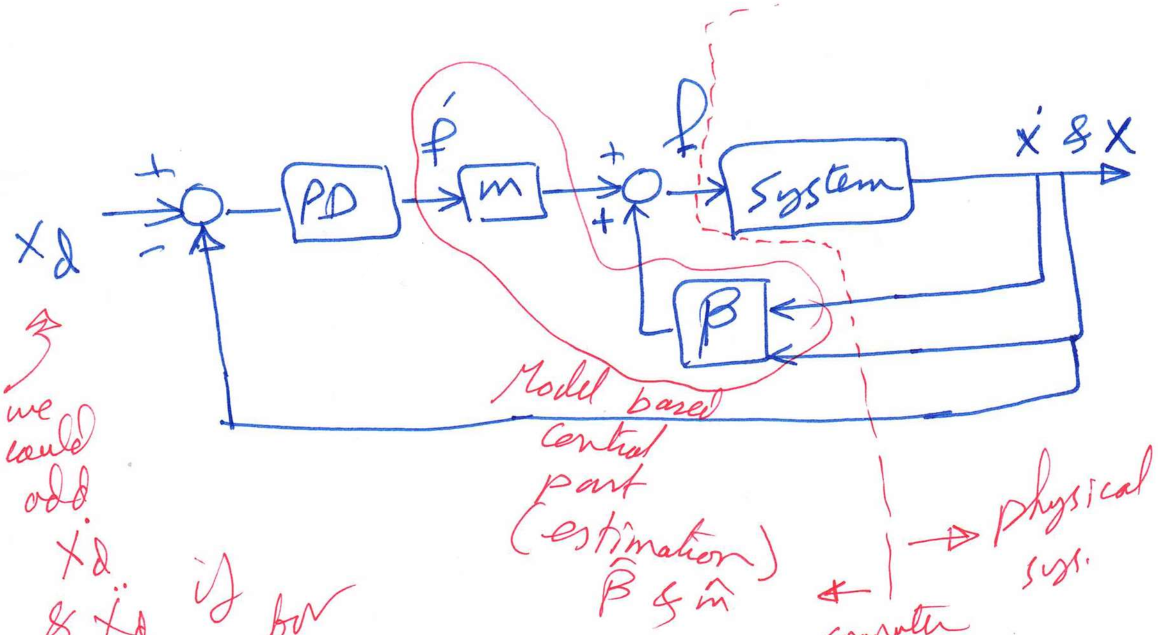
$$\therefore \; 1\ddot{x} = f' = \ddot{x}_d + K_d'(\dot{x}_d - \dot{x}) + K_p'(x_d - x)$$

$$\Rightarrow \ddot{e} + K_d'\dot{e} + K_p' e = 0$$

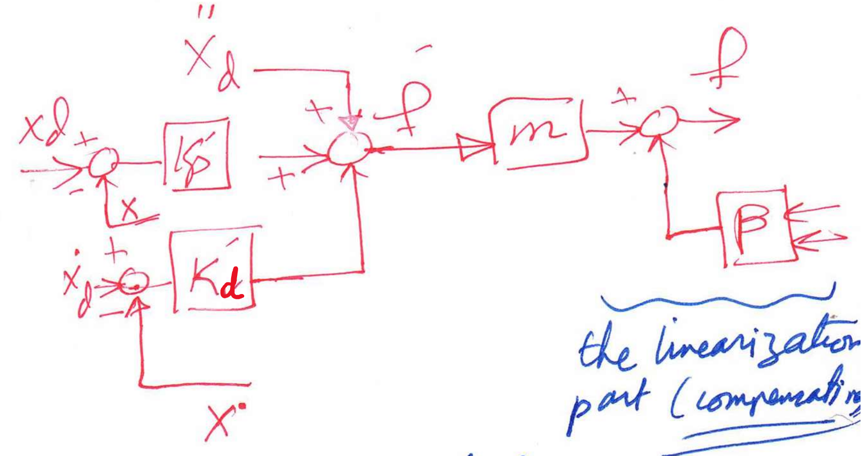
⇒

≡ to standard 2nd order sys.

that means, the error will converge to zero and we are controlling the error.



we could add $\dot{x}_d$ & $\ddot{x}_d$ is known for the desired trajectories



* computed torque Method:

- Here, we deal with matrices for the Robot manipulator;

$$\underbrace{M(\theta)}_{\text{inertia matrix}}\ddot{\theta}+\underbrace{V(\theta,\dot{\theta})+G(\theta)}_{\text{non-linear terms}}=\tau$$

vector torque

and the target is to convert the system into unit-inertia system

i.e $\tau = \ddot{\theta}$

or $$\begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & \cdots \\ 0 & 1 & 0\,0 & \cdots \\ 0 & 0 & 1 & \cdots \\ \vdots & \cdots & \cdots & 1 \end{bmatrix}}_{\text{Identity-matrix.}} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \vdots \\ \ddot{\theta}_n \end{bmatrix}$$

Similar to the previous system with m;

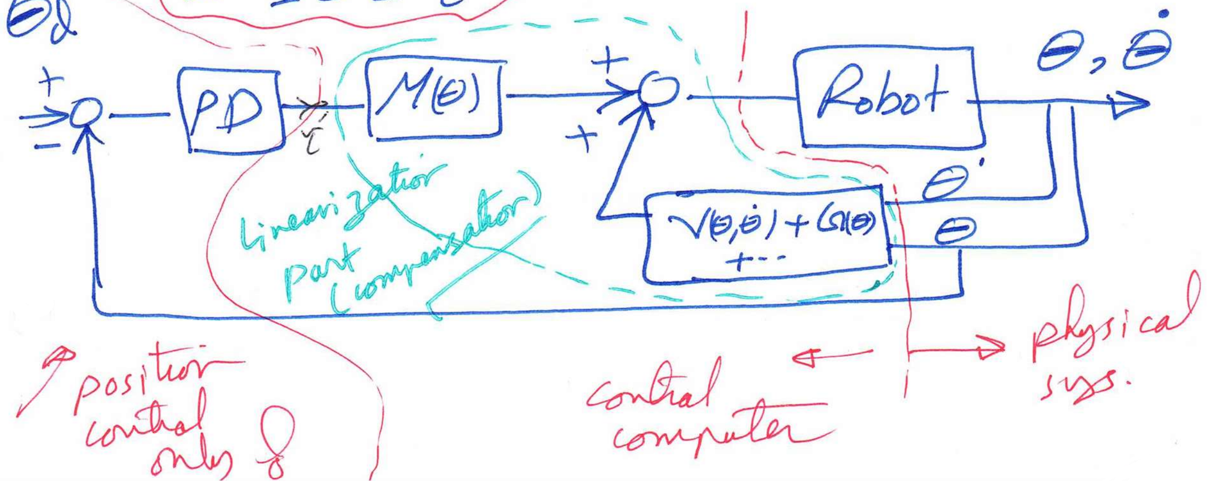
let $\tau = \alpha \tau' + \beta$ (matrix form)

$\therefore M(\theta)\ddot{\theta} + V(\theta,\dot{\theta}) + G(\theta) + \cdots = \alpha \tau' + \beta$

choose, $\alpha = M(\theta)$ & $\beta = \underbrace{V(\theta,\dot{\theta}) + G(\theta) + \cdots}_{\text{non Linearities}}$

Linear controller

$\therefore 1\ddot{\theta} = \tau'$



Example: PD-controller for a Two-links manipulator system