EE-381 Robotics-1
UG ELECTIVE

Lecture 4

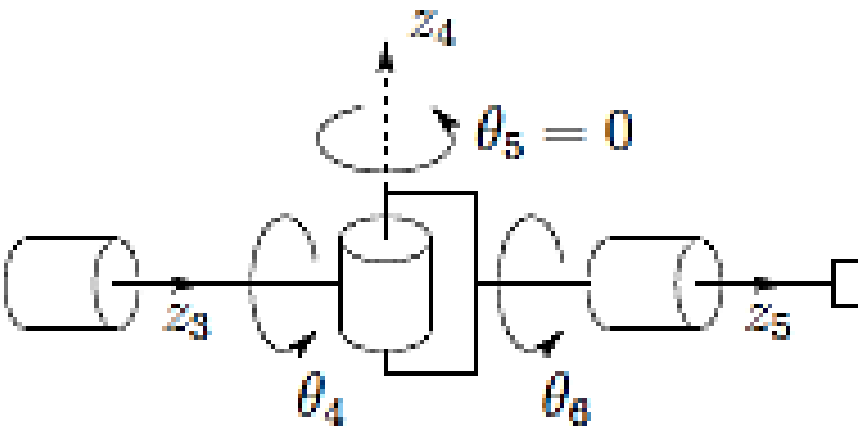
Dr. Hafsa Iqbal
Department of Electrical Engineering,
School of Electrical Engineering and Computer Science,
National University of Sciences and Technology,
Pakistan

Recap

- Pose
- Mappings
- Homogeneous transformation matrix: inverse, product
- Compositions of transformations
- Euler rotation theorem: conventions, singularity

Singularity

- When two of the axes become aligned, the system <u>loses a degree of freedom</u>

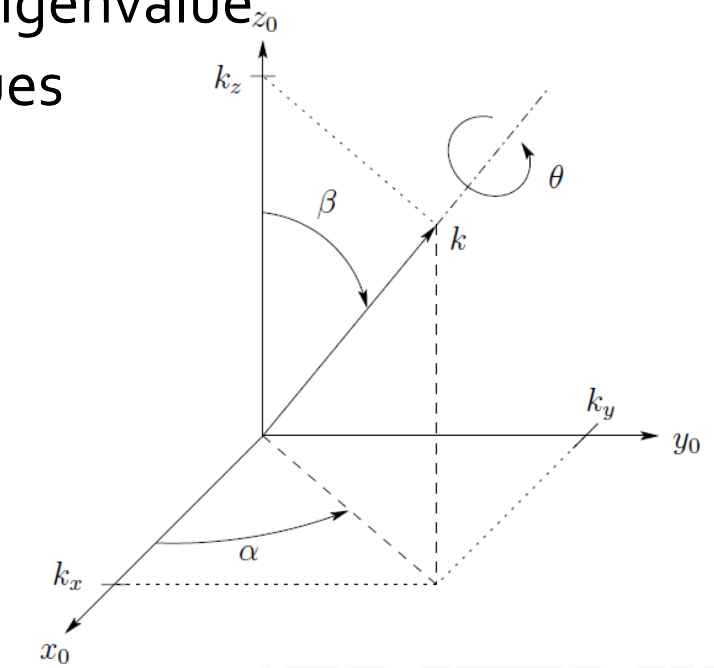


- when the axis of first and third joint aligned then their rotation makes the motion of second joint equal to zero.
- The singularity occurs when the axes of two of the joints are aligned, and the third joint loses its ability to rotate, resulting in a loss of one degree of freedom.

Axis/angle Representation

- Two coordinate frames of arbitrary orientation are related by a *single* rotation about some axis in space
- Rotation axis must be unchanged
  - Rotation matrix has three eigenvalues
  - Rotation axis is given by real value eigenvalue
  - Theta is given by complex eigenvalues

$$\boldsymbol{k} = \begin{bmatrix} k_x \\ k_y \\ k_z \end{bmatrix}$$

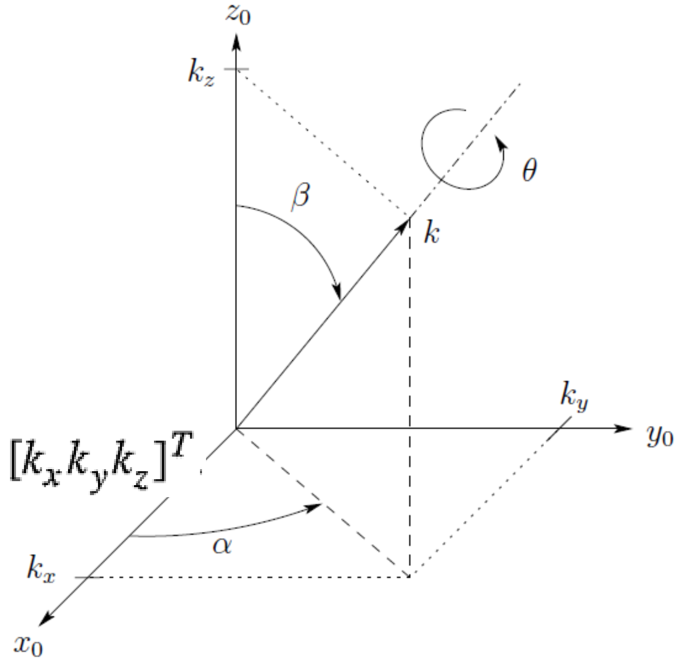


We wish to derive the rotation matrix representing a rotation of θ degrees about the arbitrary axis.

Axis/angle Representation

1. We can rotate $\boldsymbol{K}$ into z_0 by first rotating about z_0 by $-\alpha$
2. Then rotating $\boldsymbol{K}$ about y_0 by rotating $-\beta$
3. All rotations will be performed relative to the fixed frame {0}, the resultant rotation matrix will be

$$R_{\boldsymbol{k},\theta} = R_{z,\alpha}R_{y,\beta}R_{z,\theta}R_{y,-\beta}R_{z,-\alpha}.$$



$$\boxed{R_{k,\theta}} = \left[\begin{array}{c|c|c} k_x^2 v_\theta + c_\theta & k_x k_y v_\theta - k_z s_\theta & k_x k_z v_\theta + k_y s_\theta \\ k_x k_y v_\theta + k_z s_\theta & k_y^2 v_\theta + c_\theta & k_y k_z v_\theta - k_x s_\theta \\ k_x k_z v_\theta - k_y s_\theta & k_y k_z v_\theta + k_x s_\theta & k_z^2 v_\theta + c_\theta \end{array}\right]$$

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

axis/angle representation of R

Axis/angle Representation: **Inverse** Problem

- An arbitrary rotation matrix with r_{ij}

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

Rotation matrix

- The equivalent angle θ

$$\begin{aligned} \theta &= \cos^{-1}\left(\frac{Tr(R)-1}{2}\right) \\ &= \cos^{-1}\left(\frac{r_{11}+r_{22}+r_{33}-1}{2}\right) \end{aligned}$$

Axis/angle Representation: **Inverse** Problem

- and equivalent axis $\boldsymbol{k}$ are given by the expressions

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

- **Limitation:** θ is always in between [0,180]. A rotation of $-\theta$ about $-\boldsymbol{k}$ is the same as a rotation of θ about $\boldsymbol{k}$.

$$R_{\boldsymbol{k},\theta} = R_{-\boldsymbol{k},-\theta}$$

- Fails if $\theta = 0$ or $\theta = 180$.

Axis/angle Representation:

- Example: Suppose R is generated by a rotation of 90 degree about z_0 followed by a rotation of 30 degree about y_0 followed by a rotation of 60 degree x_0. Then

$$R = R_{x,60}R_{y,30}R_{z,90}$$

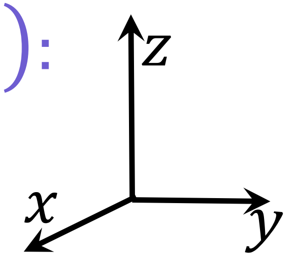
$$= \begin{bmatrix} 0 & -\frac{\sqrt{3}}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{\sqrt{3}}{4} & -\frac{3}{4} \\ \frac{\sqrt{3}}{2} & \frac{1}{4} & \frac{\sqrt{3}}{4} \end{bmatrix}$$

- Compute theta and arbitrary axis?

$$\theta = \cos^{-1}\left(-\frac{1}{2}\right) = 120^o \qquad \mathbf{k} = \left(\frac{1}{\sqrt{3}}, \frac{1}{2\sqrt{3}} - \frac{1}{2}, \frac{1}{2\sqrt{3}} + \frac{1}{2}\right)^T$$

Quaternions

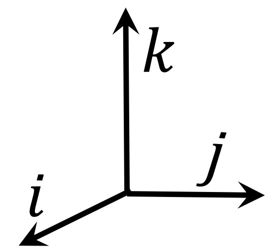
- It is tuple of 4 real elements $\left(w, \epsilon_x i + \epsilon_y j + \epsilon_z k\right)$:



$$\boxed{Q = w + \epsilon_x i + \epsilon_y j + \epsilon_z k}$$

$$\boxed{Q = w + \boldsymbol{\epsilon}}$$

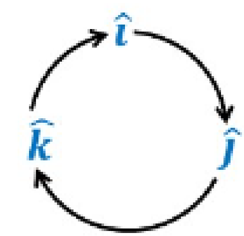
$$\boxed{w < \epsilon_x, \epsilon_y, \epsilon_z >}$$



Alternative representation

- Scalar part: w

- Vector part: $\boldsymbol{\epsilon} = \epsilon_x i + \epsilon_y j + \epsilon_z k$



- They were created by William R. Hamilton as a generalization of complex number (for 3D) such as:

$$i^2 = j^2 = k^2 = ijk = -1$$

$\hat{\imath} \times \hat{\jmath} = \hat{k}$	$\hat{\jmath} \times \hat{\imath} = -\hat{k}$
$\hat{\jmath} \times \hat{k} = \hat{\imath}$	$\hat{k} \times \hat{\jmath} = -\hat{\imath}$
$\hat{k} \times \hat{\imath} = \hat{\jmath}$	$\hat{\imath} \times \hat{k} = -\hat{\jmath}$

Unit Quaternions

- Quaternions with unit norm: $||Q|| = 1$

Relation with rotation:

- Unit quaternions represent a rotation of θ about a unit axis $\boldsymbol{r}$ (rotation axis)

$$\boxed{Q = (w, \boldsymbol{\epsilon}) = \left(cos\frac{\theta}{2}, \boldsymbol{r}\, sin\frac{\theta}{2} \right)}$$

- Some important properties of unit quaternions are:
  - $Q^{-1} = Q^*$
  - $||Q_1 \circ Q_2|| = 1$
- **Note**: in robotics we usually refer to unit quaternions simply as quaternions

Quaternions: Operations

- Consider the following quaternions:

$$Q_1 = (w_1, \epsilon_1) \quad , \quad Q_2 = (w_2, \epsilon_2) \quad , \quad Q = (w, \epsilon)$$

- Addition: $Q_1 + Q_2 = (w_1 + w_2, \epsilon_1 + \epsilon_2)$

(associative and commutative)

- Compounding:

$$Q_1 \mathrm{o} Q_2 = (\underbrace{w_1 w_2 - \epsilon_1^{\mathrm{T}} \epsilon_2,\ w_1 \epsilon_2 + w_2 \epsilon_1 + \epsilon_1 \times \epsilon_2}_{\text{Hamilton product}})$$

- Conjugate→ $Q^* = (w, -\epsilon)$
- Norm $||Q|| \rightarrow ||Q||^2 = w^2 + \epsilon_x^2 + \epsilon_y^2 + \epsilon_z^2 = Q \mathrm{o} Q^*$

Norm of a vector

Unit Quaternions

- **Example:**
  a) Find the quaternions that represents a rotation of 60^0 about $(1,0,0)$.

$$Q = \left(\cos\left(\frac{60^\circ}{2}\right), \sin\left(\frac{60^\circ}{2}\right)(1,0,0) \right) = (0.866,\ 0.5,\ 0,\ 0)$$

  b) Find the conjugate and inverse of the previous quaternions Q

$$Q^* = Q^{-1} = (0.866,\ -0.5,\ 0,\ 0)$$

Unit Quaternions

Application of a rotation Q to a vector $\boldsymbol{v}$:

1. Convert vector $\boldsymbol{v}$ to a quaternion (0 scalar component): $\widetilde{\boldsymbol{v}} = (\boldsymbol{0}, \boldsymbol{v})$

2. Apply the rotation Q: $\widetilde{\boldsymbol{v}}_{\boldsymbol{q_{rot}}} = \boldsymbol{Q} \circ \widetilde{\boldsymbol{v}} \circ \boldsymbol{Q}^*$

3. The rotation vector $\boldsymbol{v}_{\mathrm{rot}}$ is the vector component $\boldsymbol{v}_{\mathrm{rot}} = (\boldsymbol{0}, \boldsymbol{v_{rot}})$

Unit Quaternions

Example : Find the rotation of point p=(3,5,2) by an angle of 60^0 about (1,0,0) **(a)** using quaternions, **(b)** using a rotation matrix.

$$\tilde{p} = (0,\ 3,\ 5,\ 2)$$

Solution

a) Using quaternions

$$\tilde{p}_{rot} = Q \circ (0,\ 3,\ 5,\ 2) \circ Q^*$$

$$= (0.866,\ 0.5,\ 0,\ 0) \circ (0,\ 3,\ 5,\ 2) \circ (0.866,\ -0.5,\ 0,\ 0)$$

$$= (-1.5,\ 2.6,\ 3.33,\ 4.23) \circ (0.866,\ -0.5,\ 0,\ 0)$$

$$= (0,\ 3,\ 0.768,\ 5.33) \longrightarrow p_{rot} = (3,\ 0.768,\ 5.33)$$

b) Using a rotation matrix

$$p_{rot} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos 60^\circ & -\sin 60^\circ \\ 0 & \sin 60^\circ & \cos 60^\circ \end{bmatrix} \begin{bmatrix} 3 \\ 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 0.768 \\ 5.33 \end{bmatrix}$$

Unit Quaternions / Euler parameters

- Another representation of orientation is by means of four numbers called the **Euler parameters**.
- Sometimes, the Euler parameters are viewed as a 3×1 vector plus a scalar. However, as a 4×1 vector, the Euler parameters are known as a **unit quaternion**.
- In terms of equivalent axis $K = \begin{bmatrix} K_x & K_y & K_z \end{bmatrix}^T$ and equivalent angle theta

$$\epsilon_1 = k_x \sin\frac{\theta}{2},$$

$$\epsilon_2 = k_y \sin\frac{\theta}{2},$$

$$\epsilon_3 = k_z \sin\frac{\theta}{2},$$

$$\epsilon_4 = \cos\frac{\theta}{2}.$$

$$\epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 + \epsilon_4^2 = 1$$

Quaternions

- The rotation matrix R_ϵ that is equivalent to a set of Euler parameters is

$$R_\epsilon = \begin{bmatrix} 1-2\epsilon_2^2-2\epsilon_3^2 & 2(\epsilon_1\epsilon_2-\epsilon_3\epsilon_4) & 2(\epsilon_1\epsilon_3+\epsilon_2\epsilon_4) \\ 2(\epsilon_1\epsilon_2+\epsilon_3\epsilon_4) & 1-2\epsilon_1^2-2\epsilon_3^2 & 2(\epsilon_2\epsilon_3-\epsilon_1\epsilon_4) \\ 2(\epsilon_1\epsilon_3-\epsilon_2\epsilon_4) & 2(\epsilon_2\epsilon_3+\epsilon_1\epsilon_4) & 1-2\epsilon_1^2-2\epsilon_2^2 \end{bmatrix}$$

- Given a rotation matrix, the equivalent Euler parameters are

$$\epsilon_1 = \frac{r_{32}-r_{23}}{4\epsilon_4}, \qquad \epsilon_3 = \frac{r_{21}-r_{12}}{4\epsilon_4},$$

$$\epsilon_2 = \frac{r_{13}-r_{31}}{4\epsilon_4}, \qquad \epsilon_4 = \frac{1}{2}\sqrt{1+r_{11}+r_{22}+r_{33}}.$$

Summary: Pose in 3D

- Pose ➡ position and orientation

  ⬇ ⬇

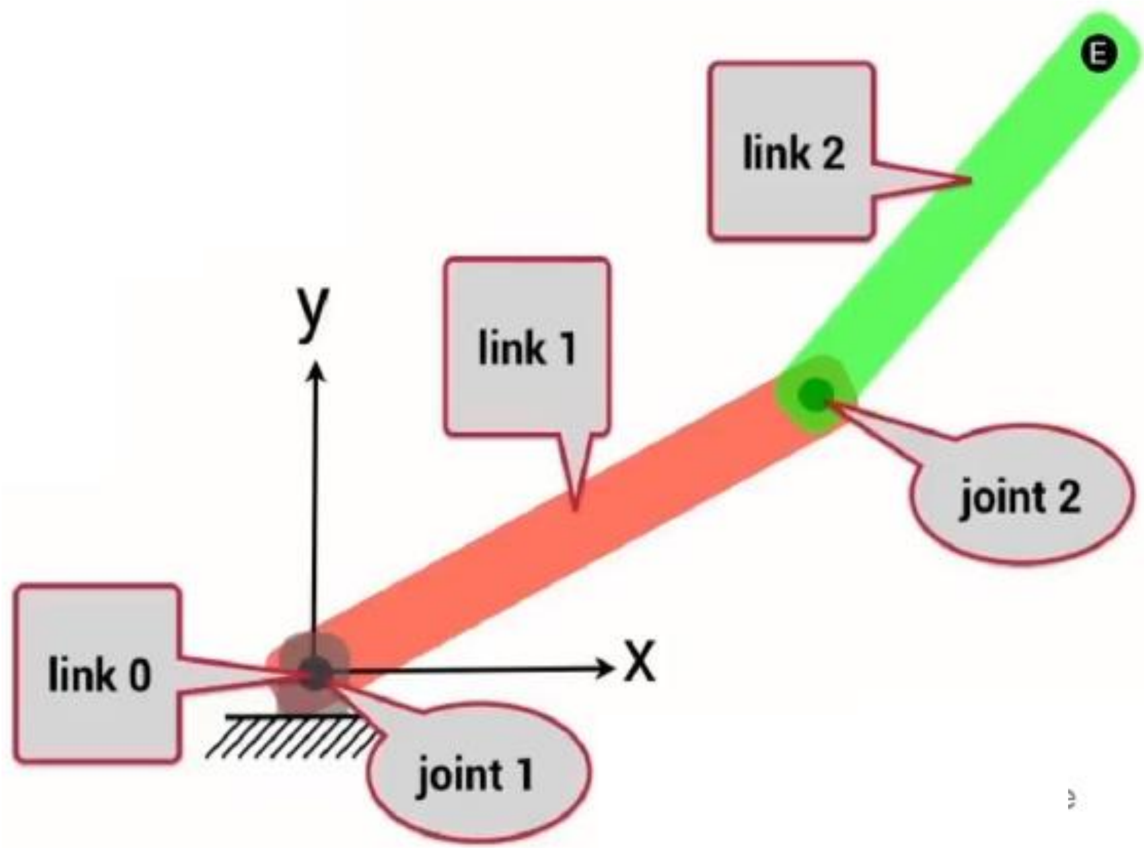
  Translation and rotation

- Homogeneous transformation
- Euler angles
- Cardanian angles
- Quaternion

Robotic Manipulators

- A robotic manipulator is a kinematic chain
  - i.e. an assembly of pairs of rigid bodies that can move with respect to one another via a mechanical constraint
- The rigid bodies are called *links*
- The mechanical constraints are called *joints*
- Each joint has 1 –DOF, either translational (a sliding or prismatic joint) or rotational (a revolute joint)

Robotic Manipulators



Joints

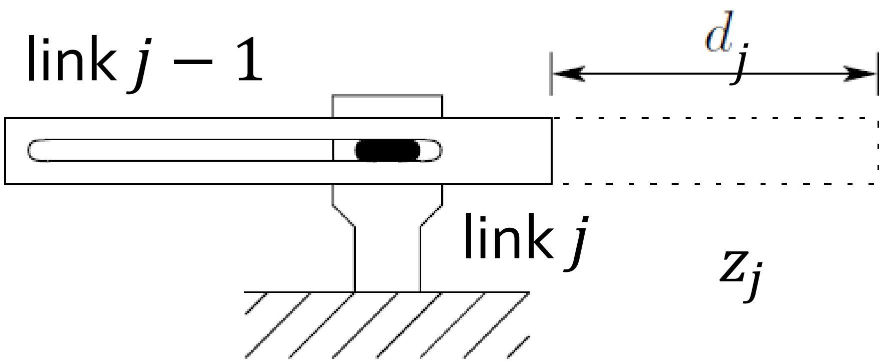
- Joints: Rotary (Revolute), Linear (Prismatic)

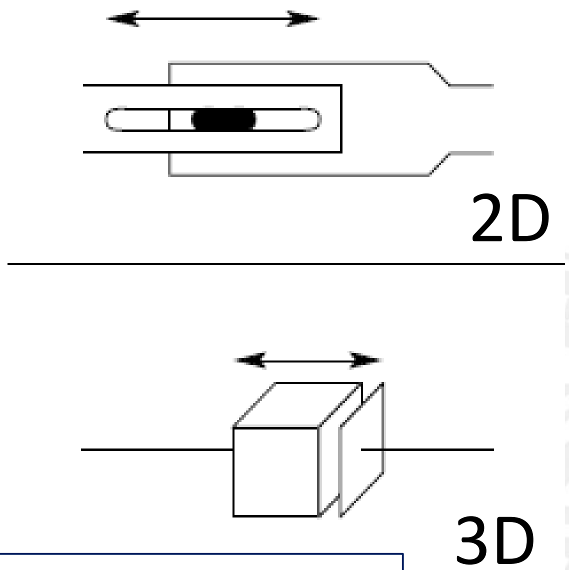
- Our convention: joint j connects link $j-1$ to link j
- When joint j is actuated, link j moves

Robotic Joints

Prismatic Joint: allows a **linear** relative motion along a fixed axis between two links.

- Represented with P
- Example: Piston



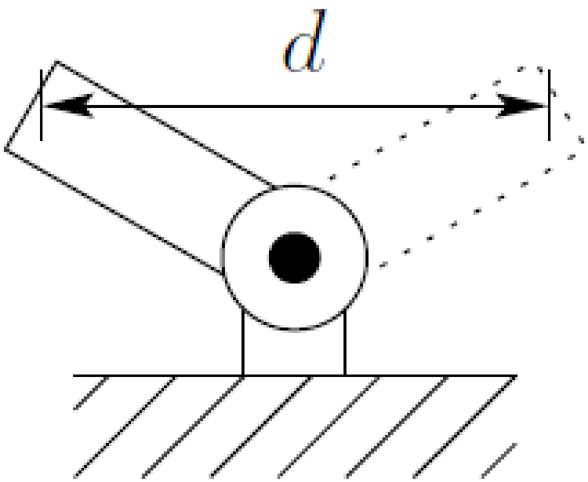


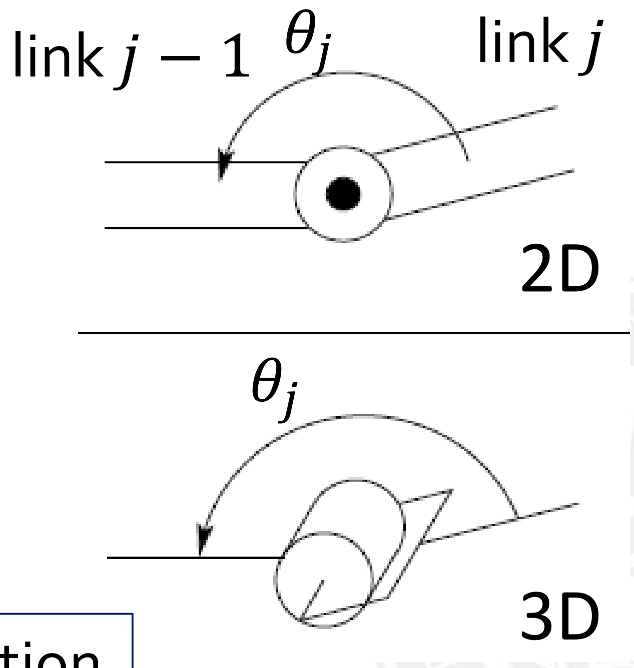
Axis of translation is z-axis by convention

Robotic Joints

Revolute Joint: is like a hinge and allows relative **rotation** about a fixed axis between two links

- Represented with R
- Like a Hinge





Axis of rotation is z-axis by convention

Joint variables

- Revolute and prismatic joints are one degree of freedom (DOF) joints; thus, they can be described using a single numeric value called a joint variable
- q_j → joint variable for joint j
- **Prismatic**
  - $q_j = d_j$ → displacement of link j relative to link $j - 1$
- **Revolute**
  - $q_j = \theta_j$ → angle of rotation of link j relative to link $j - 1$

Manipulator Arrangements

- Most industrial manipulators have six or fewer joints
  - First three joints are the arm
  - Remaining joints are the wrist
- It is common to describe such manipulators using the joints of the arm
  - R: revolute joint
  - P: prismatic joint

Robot Arm

- Manipulator
  - A serial-link manipulator, comprises a chain of rigid links and joints.
  - Each joint has one degree of freedom, either translational (a sliding or prismatic joint) or rotational (a revolute joint)
- One end of the chain, the base, is generally fixed and the other end is free to move in space and holds the tool or end-effector that does the useful work.

End-Effector will be a complex function of the state of each joint

Kinematics

- **Kinematics** is the branch of mechanics that studies the motion of a body or a system of bodies
- Concerned with position (and angles) and velocities (translational and angular)
- Not concerned with force or moments. Such a study is a part of Dynamics.

Robot Kinematics

Two kinematic problems in Robotics

- **Forward Kinematics**: Given joint angles, we can find the robot's tool tip.
- **Inverse Kinematics**: Given the pose of robot tool tip, we find the required joint angles.

https://www.youtube.com/watch?v=91q-__jTH1U

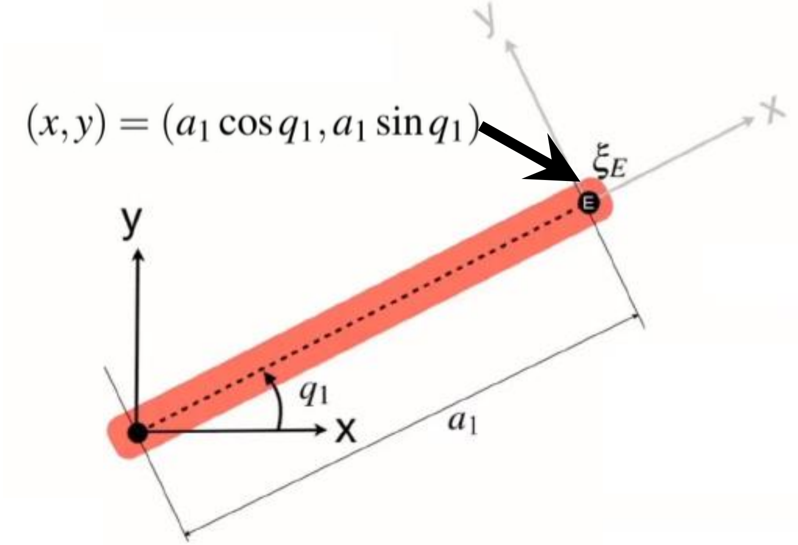
Goals

- **Forward Kinematics**: How to compute the pose of the end-effector?
- **Inverse Kinematics**: How to compute the position of each joint given the end-effector pose?
- How to generate smooth paths/trajectories for the end-effector?
- How to compute the Denavit Hartenberg (DH) parameters?

Forward Kinematics-Single Joint

- Consider the simple robot arm, which has a single rotational joint.

- We can describe the pose of its end-effector–frame $\{E\}$ – by a sequence of relative poses: a rotation about the joint axis and then a translation by a_1 along the rotated x-axis



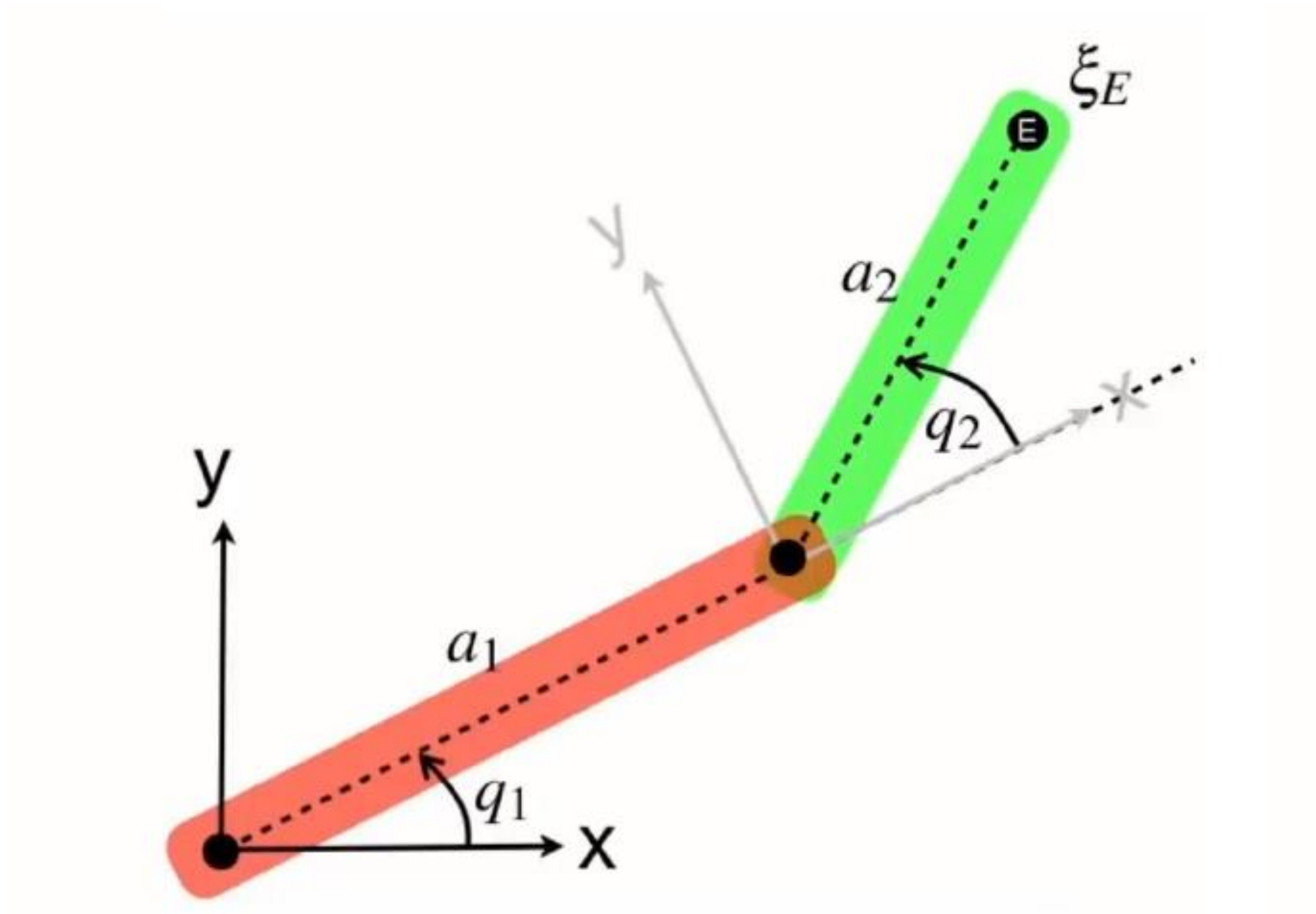
$$\mathbf{E} = \mathbf{R}(q_1)\,\mathbf{T}_x(a_1)$$

$$\mathbf{E} = \begin{pmatrix} \cos q_1 & -\sin q_1 & 0 \\ \sin q_1 & \cos q_1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & a_1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{E} = \begin{pmatrix} \cos q_1 & -\sin q_1 & a_1 \cos q_1 \\ \sin q_1 & \cos q_1 & a_1 \sin q_1 \\ 0 & 0 & 1 \end{pmatrix}$$

Forward Kinematics—2 Joint

$$E = R(q_1)T_x(a_1)R(q_2)T_x(a_2)$$



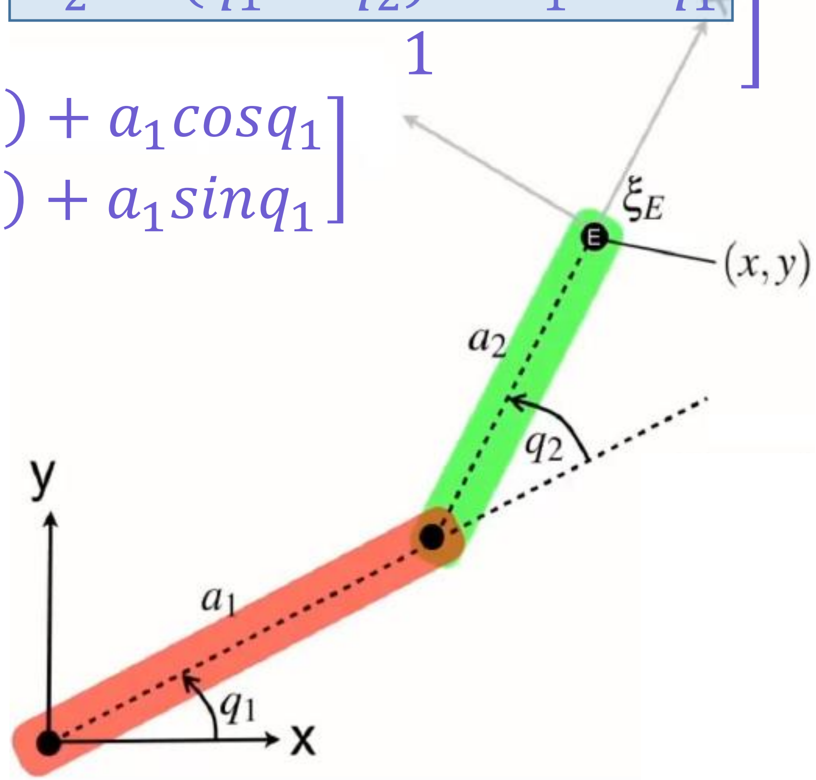
Forward Kinematics—2 Joint

- The gray frame is result of

$$\boldsymbol{E} = \boldsymbol{R}(q_1)\boldsymbol{T}_x(a_1)\boldsymbol{R}(q_2)\boldsymbol{T}_x(a_2)$$

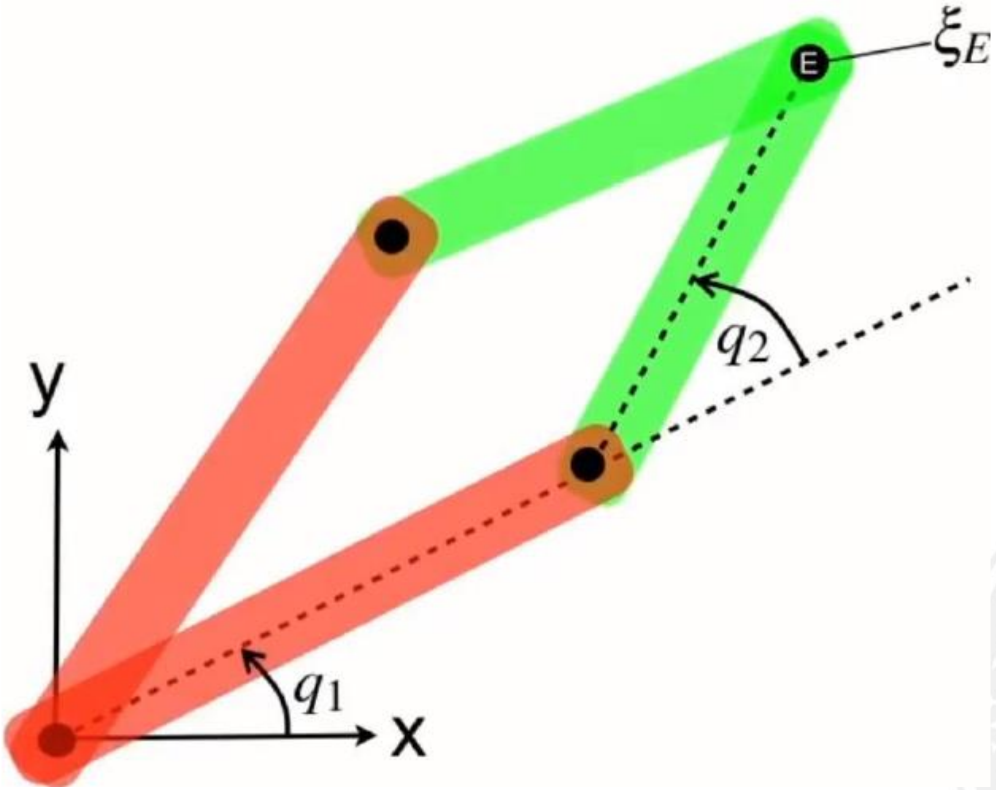
$$\boldsymbol{E} = \begin{bmatrix} \cos(q_1+q_2) & -\sin(q_1+q_2) & \boxed{\begin{matrix} a_2\cos(q_1+q_2)+a_1 cos q_1 \\ a_2\sin(q_1+q_2)+a_1 sin q_1 \end{matrix}} \\ \sin(q_1+q_2) & \cos(q_1+q_2) & \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \xi_x \\ \xi_y \end{bmatrix} = \begin{bmatrix} a_2\cos(q_1+q_2)+a_1 cos q_1 \\ a_2\sin(q_1+q_2)+a_1 sin q_1 \end{bmatrix}$$

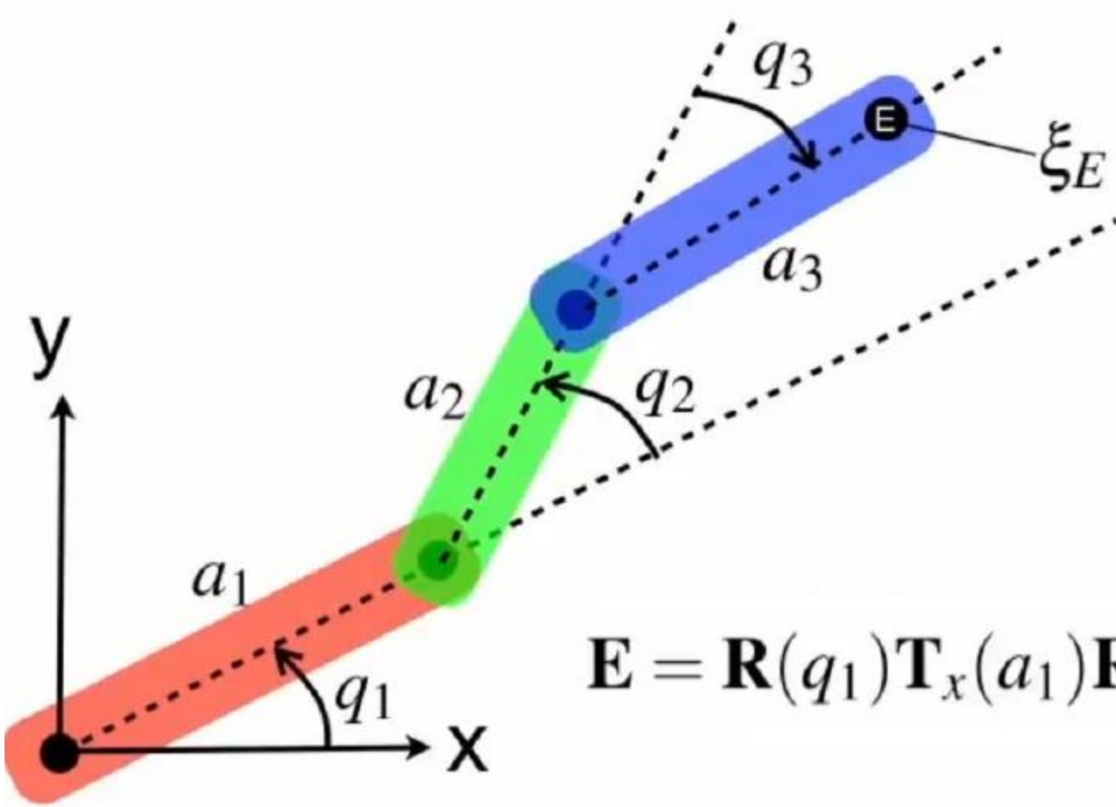


Forward Kinematics—2 Joint

- Configurations are not unique



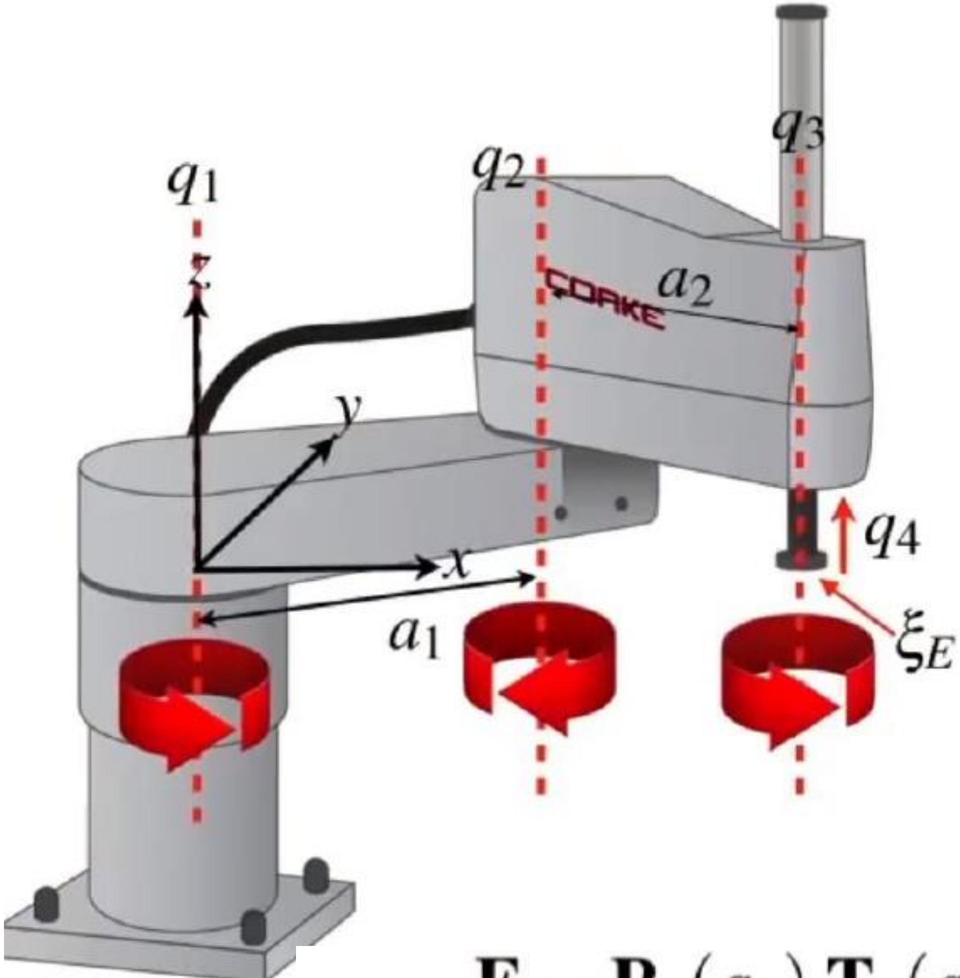
Forward Kinematics—3 Joint

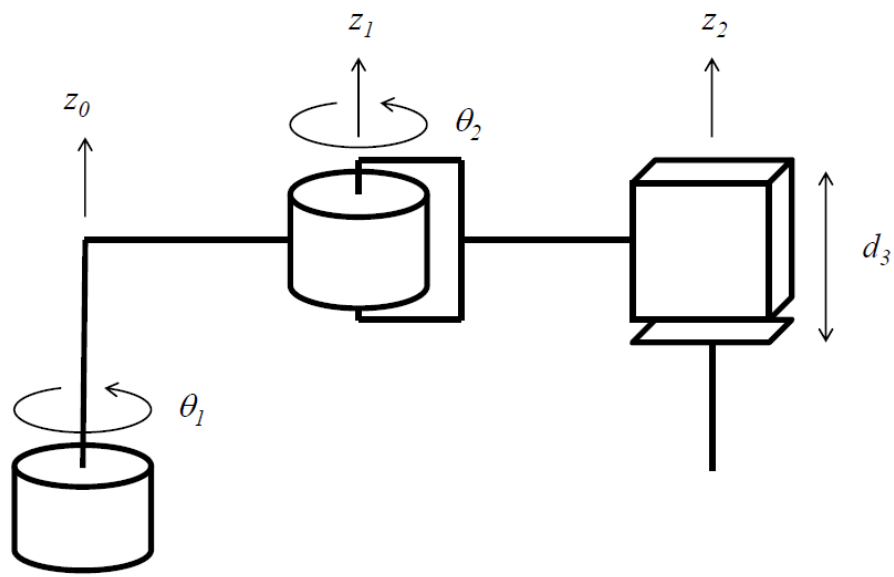


$$\mathbf{E} = \mathbf{R}(q_1)\mathbf{T}_x(a_1)\mathbf{R}(q_2)\mathbf{T}_x(a_2)\mathbf{R}(q_3)\mathbf{T}_x(a_3)$$

Forward Kinematics—$3D$ (SCARA Robot)

https://www.youtube.com/watch?v=-m1oKuFkSTE

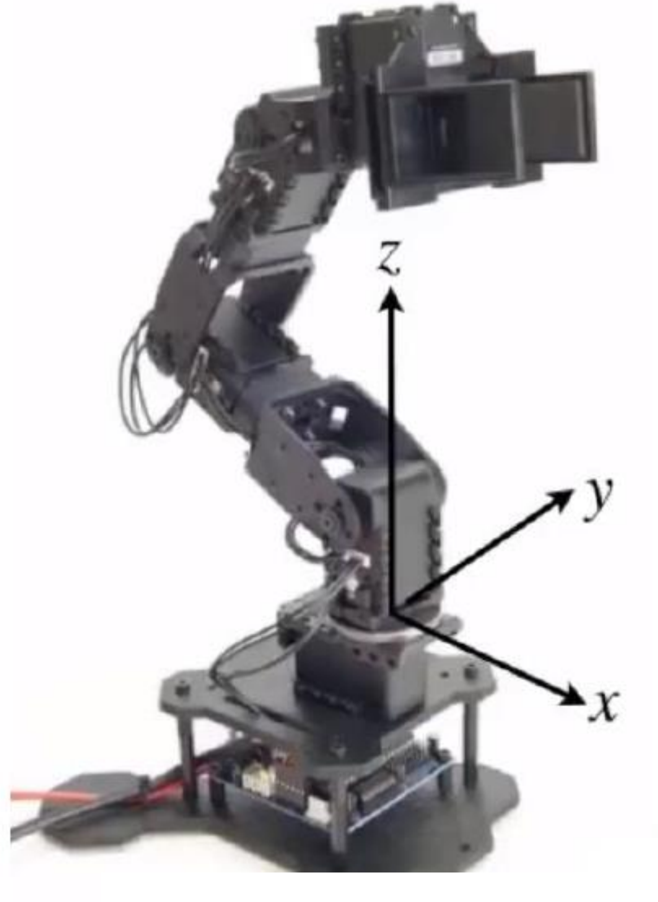




$$\mathbf{E} = \mathbf{R}_z(q_1)\,\mathbf{T}_x(a_1)\,\mathbf{R}_z(q_2)\,\mathbf{T}_x(a_2)\,\mathbf{R}_z(q_3)\,\mathbf{T}_z(q_4)$$

Forward Kinematics$-3D$

https://www.youtube.com/watch?v=zwTRbiUEVPk



4 joints

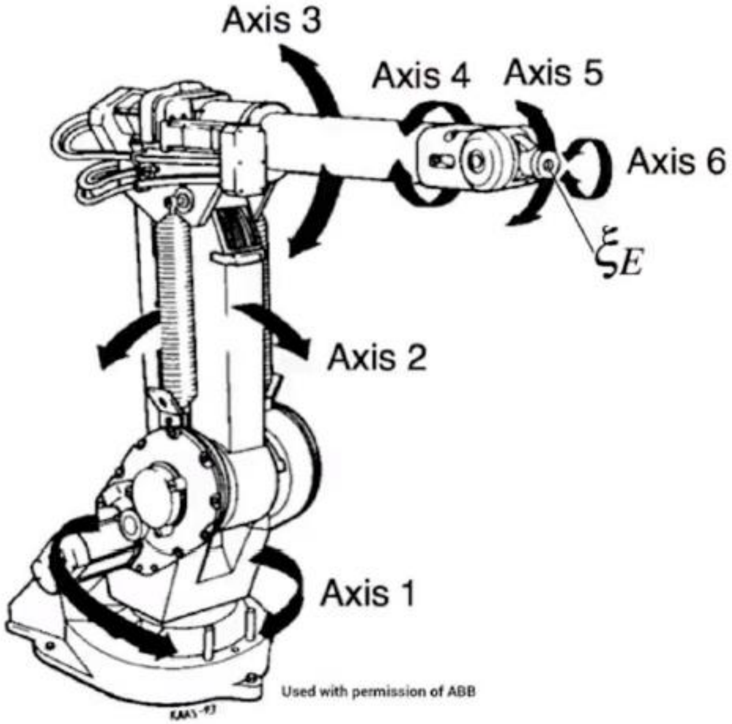
PhantomX Pincher Robot Arm 2014

$$\mathbf{E} = \mathbf{R}_z(q_1)\mathbf{T}_z(a_1)\mathbf{R}_y(q_2)\mathbf{T}_z(a_2)\mathbf{R}_y(q_3)\mathbf{T}_z(a_3)\mathbf{R}_y(q_4)\mathbf{T}_z(a_4)$$

Forward Kinematics-General Purpose 3D Robot

Home work

Unimate Puma 500
Oussama Khatib | Used with permission



Used with permission of ABB