Week 3 Lecture 1

FOUNDATIONS OF ROBOT MOTION: KINEMATICS AND TRANSFORMATIONS

Chek-in: 845589

Content

- 3D Transformations
- Rotation Matrices
- Homogeneous Transformations
- Forward Kinematics Using Homogeneous Transformations
- D-H Parameters
- Manipulator Jacobian

3D Translation Concept

- Translation shifts objects from one location to another in three-dimensional space, altering their position without rotation (Craig, 2005).

$$p' = p + t$$

where

p is the original position vector

t_x is the displacement in x

t_y is the displacement in y

t_z is the displacement in z

t=t_x, t_y, t_z is the translation vector

p' is the new position.

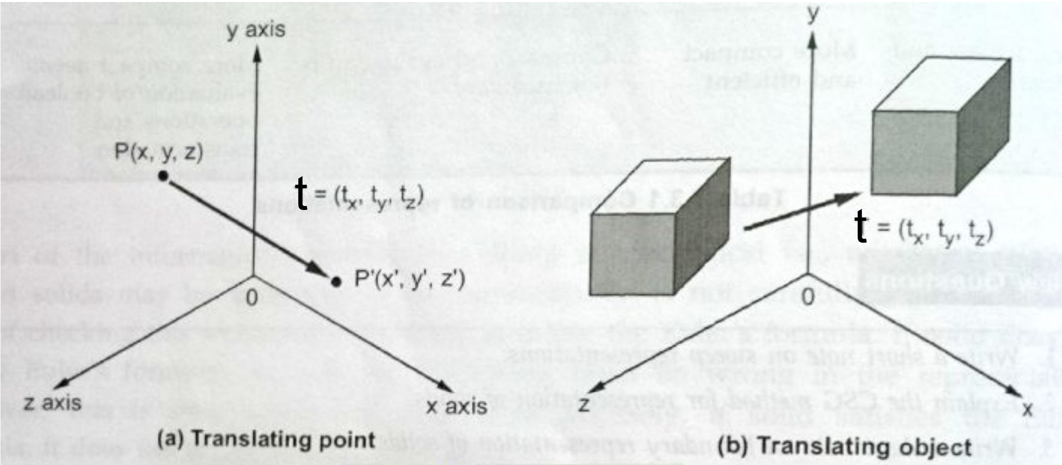


Fig. (1) 3D translation

Rotation Matrices

Mathematically represent orientation changes, essential for kinematic modelling in robotics (Siciliano, Sciavicco, Villani, & Oriolo, 2010).

$$R \in SO(3) = \{R \in \mathbb{R}^{3\times 3} \mid R^T R = I,\ \det(R) = 1\}$$

This defines the group of all valid 3D rotation matrices.

Rotation Matrices

Rotation matrices are a special class of **orthogonal matrices**, which means:

- **Transpose equals inverse**:
  $R^T = R^{-1}$ This ensures that applying a rotation and then its transpose (inverse) returns the original vector.
- **Preserves vector length and angles**:
  Rotation matrices maintain the **Euclidean norm** of vectors, meaning: $\|R\mathbf{v}\|=\|\mathbf{v}\|$ and the angle between vectors remains unchanged.

3D Rotation Principles

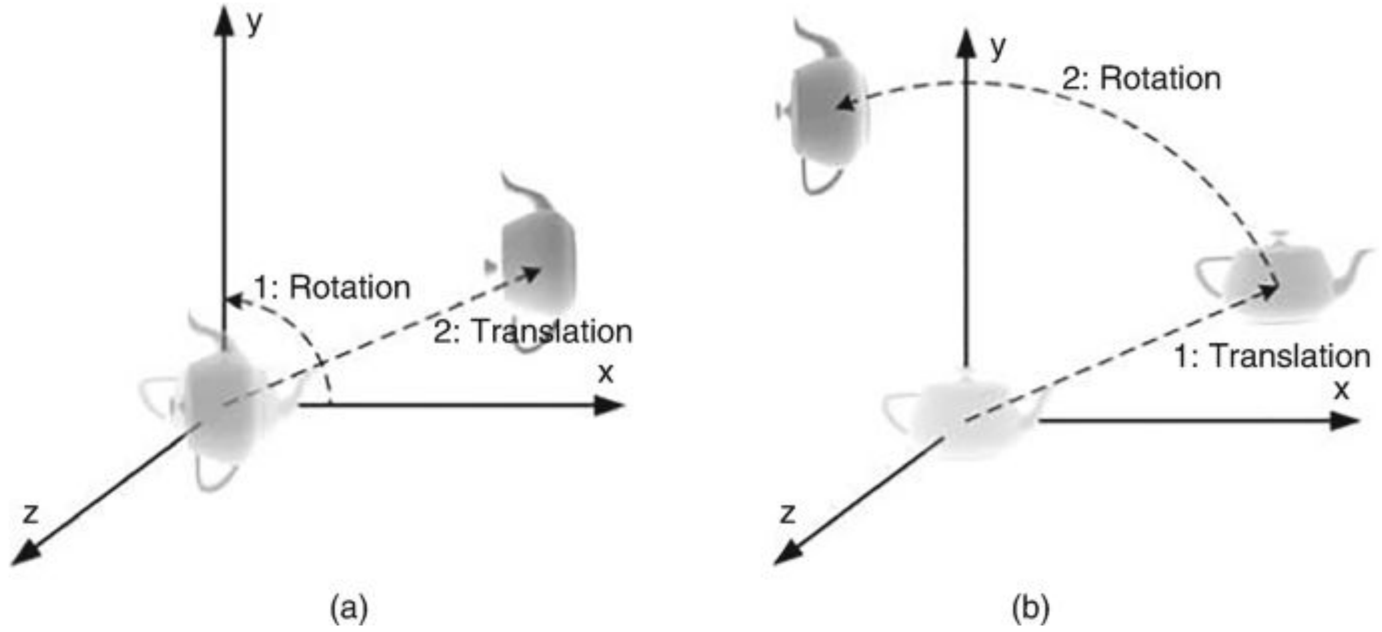
Rotation turns objects around principal axes using rotation matrices to change orientation in 3D space (Murray, Li, & Sastry, 1994).

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

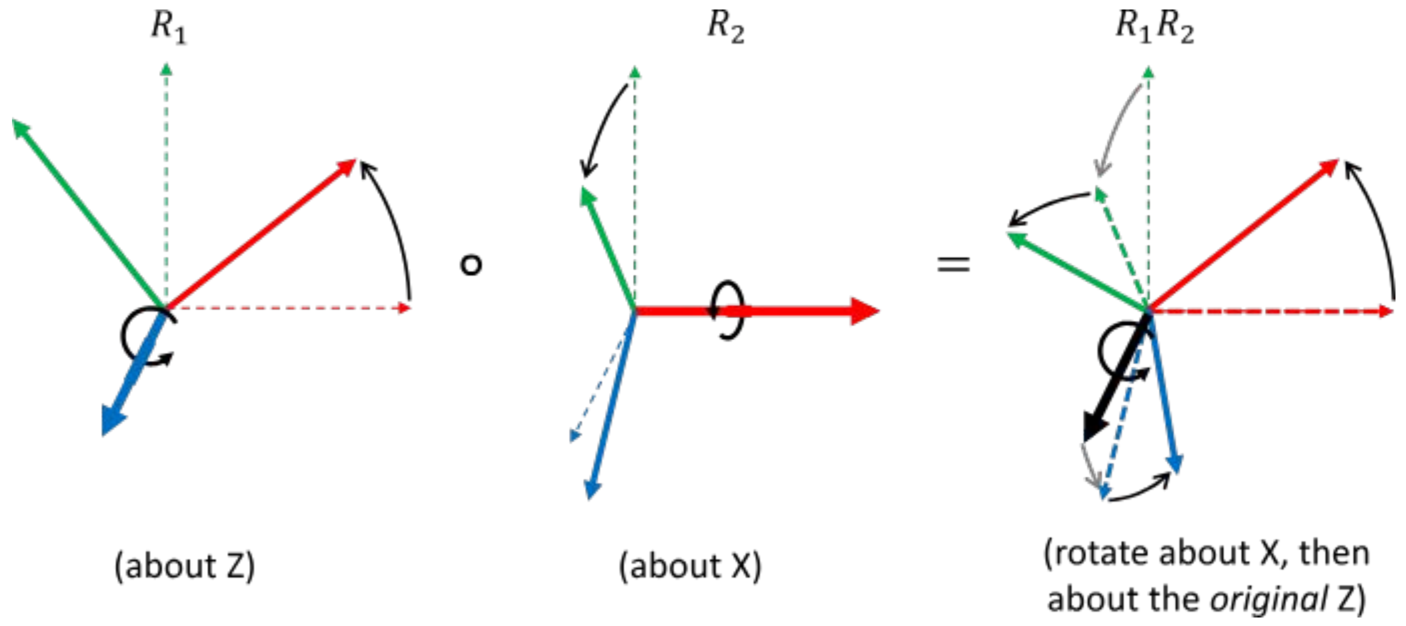
$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

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

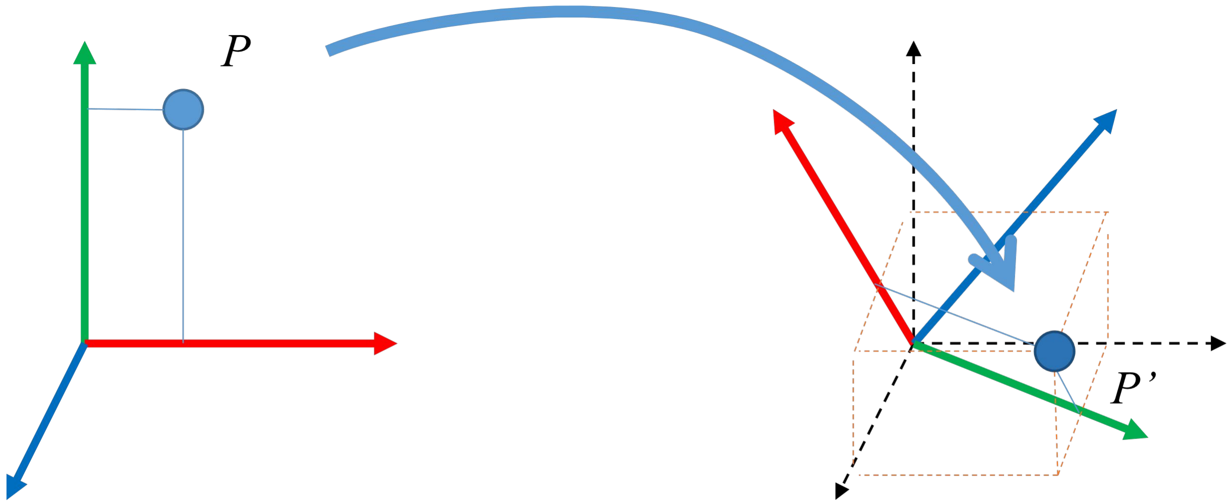
3D Rotation Principles



3D Rotation Principles



3D Rotation Principles



$$\boldsymbol{p} = (x, y, z)$$

$$\boldsymbol{p}' = R\boldsymbol{p} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Homogeneous Transformations

A homogeneous transformation matrix is a 4×4 matrix that combines:

- A **3×3 rotation matrix** (describing orientation)
- A **3×1 translation vector** (describing position)
- This matrix allows for the representation of both **rotation and translation** in a single operation, enabling efficient computation of spatial transformations.

$$T = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}$$

Homogeneous Transformations

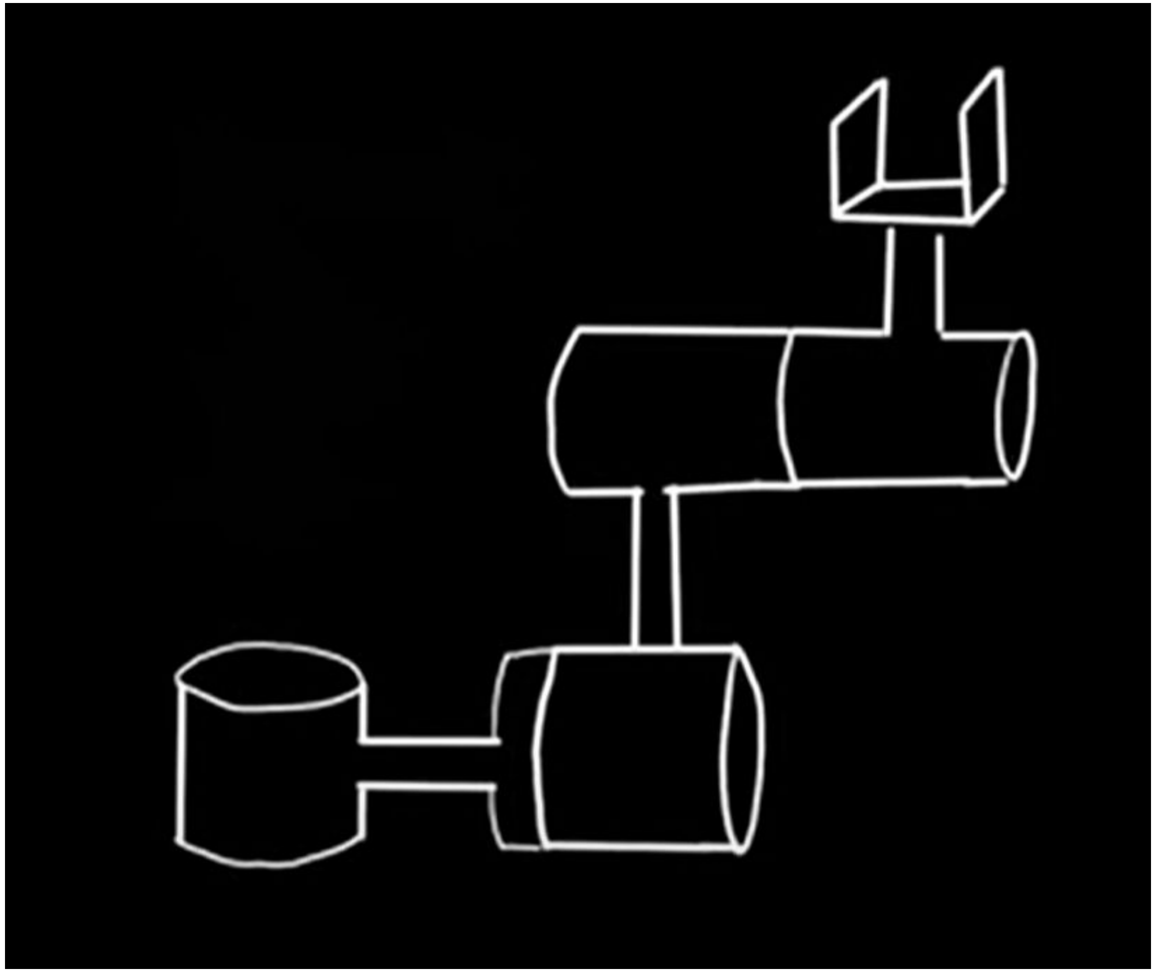
- $$R \in SO(3) = \{R \in R^{3x3} \mid R^T R = I, det(R) = 1\}$$
- This represents a **3D rotation matrix**, ensuring orthogonality and unit determinant.

$$p \in R^3$$

- This is the **translation vector**, representing position shift in 3D space.

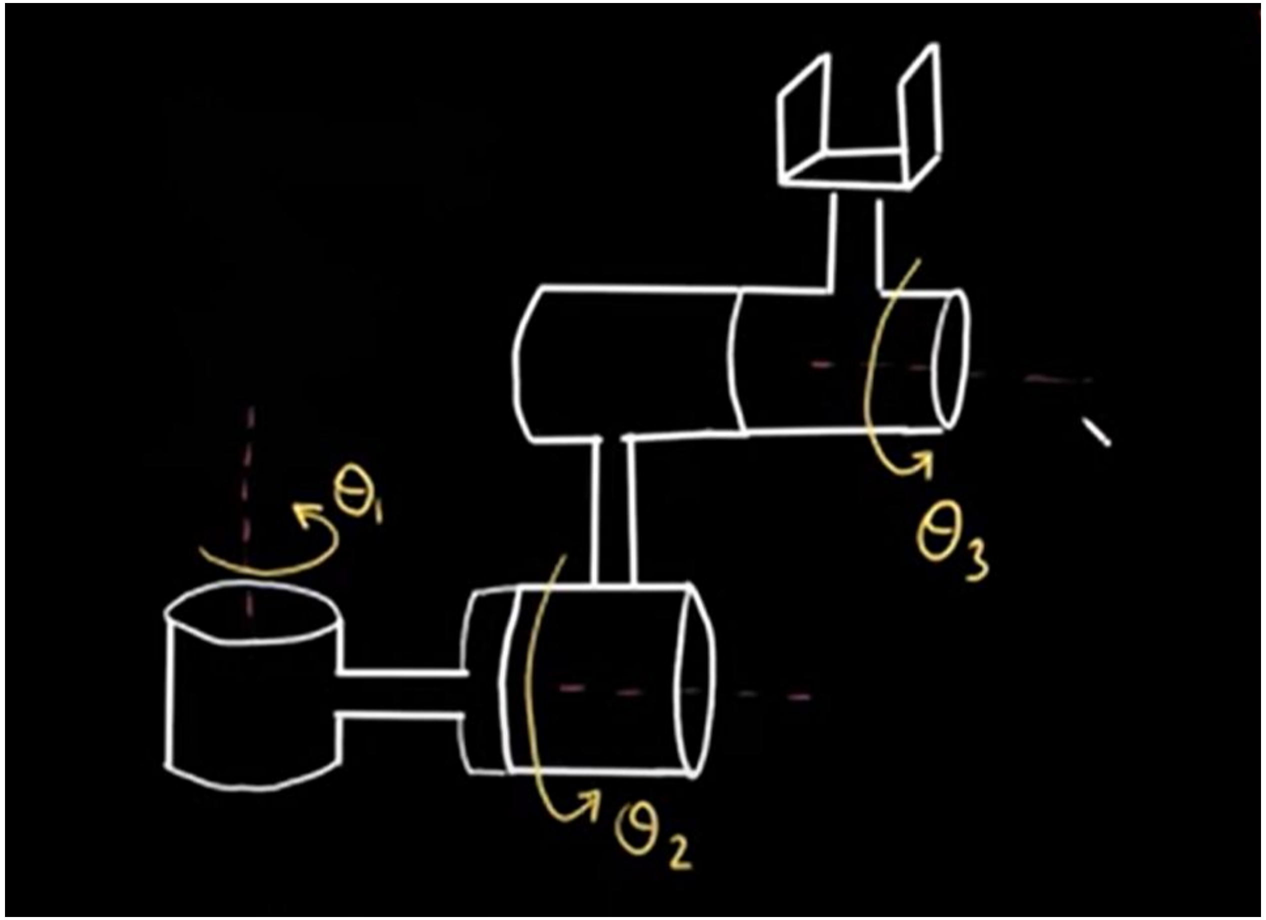
$$T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Forward Kinematics using Homogeneous Transformation



Step1: Asses the direction of the motion based on the joint type (rotation or extension)

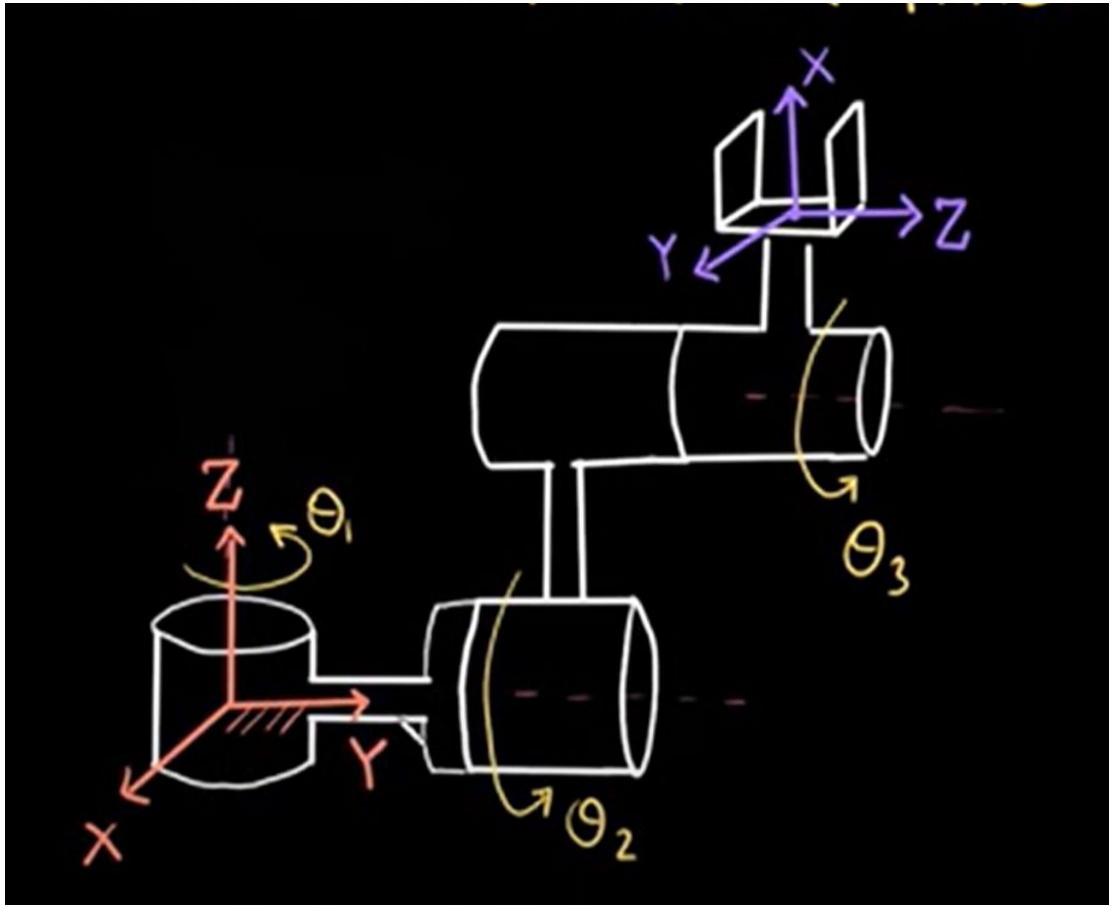
Forward Kinematics using Homogeneous Transformation



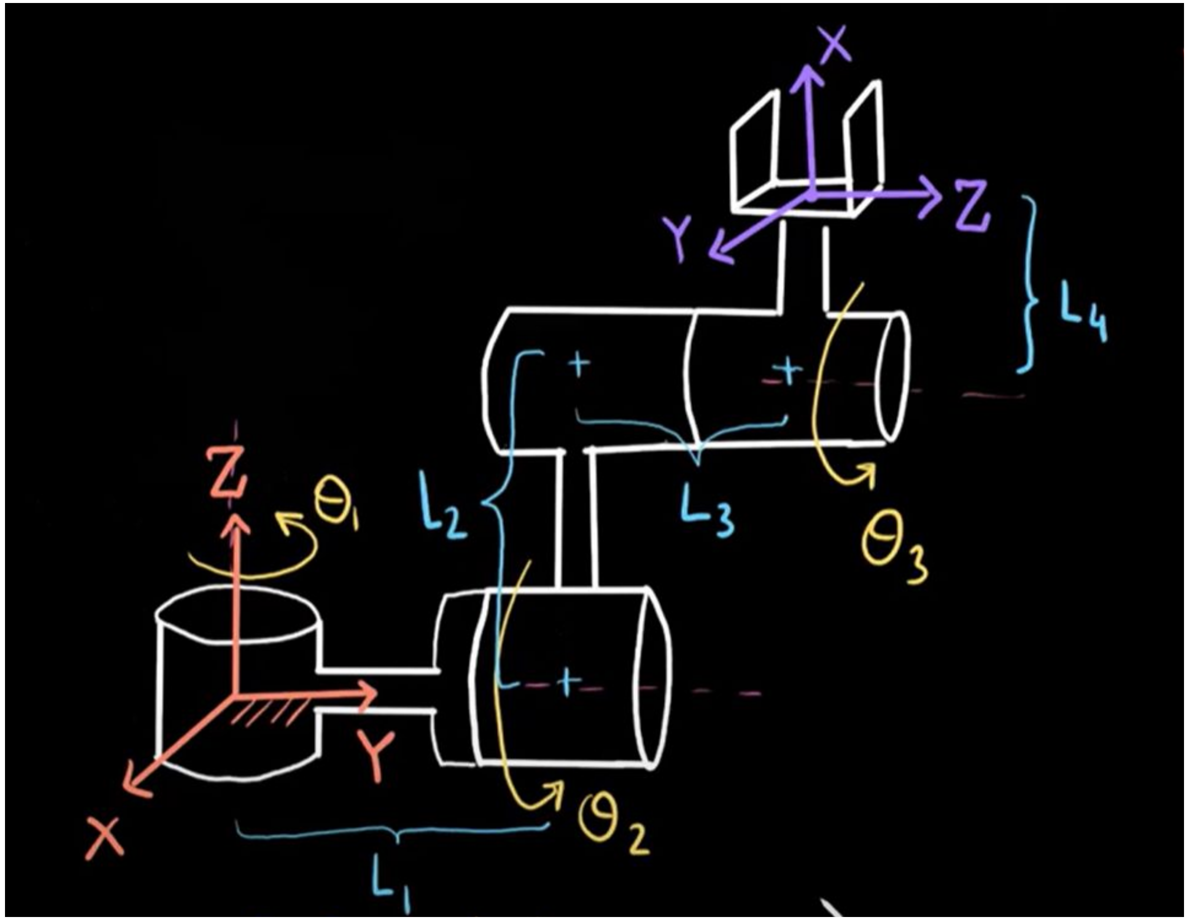
Step2: Assign coordinate frames

Forward Kinematics using Homogeneous Transformation

Step3: Define displacement

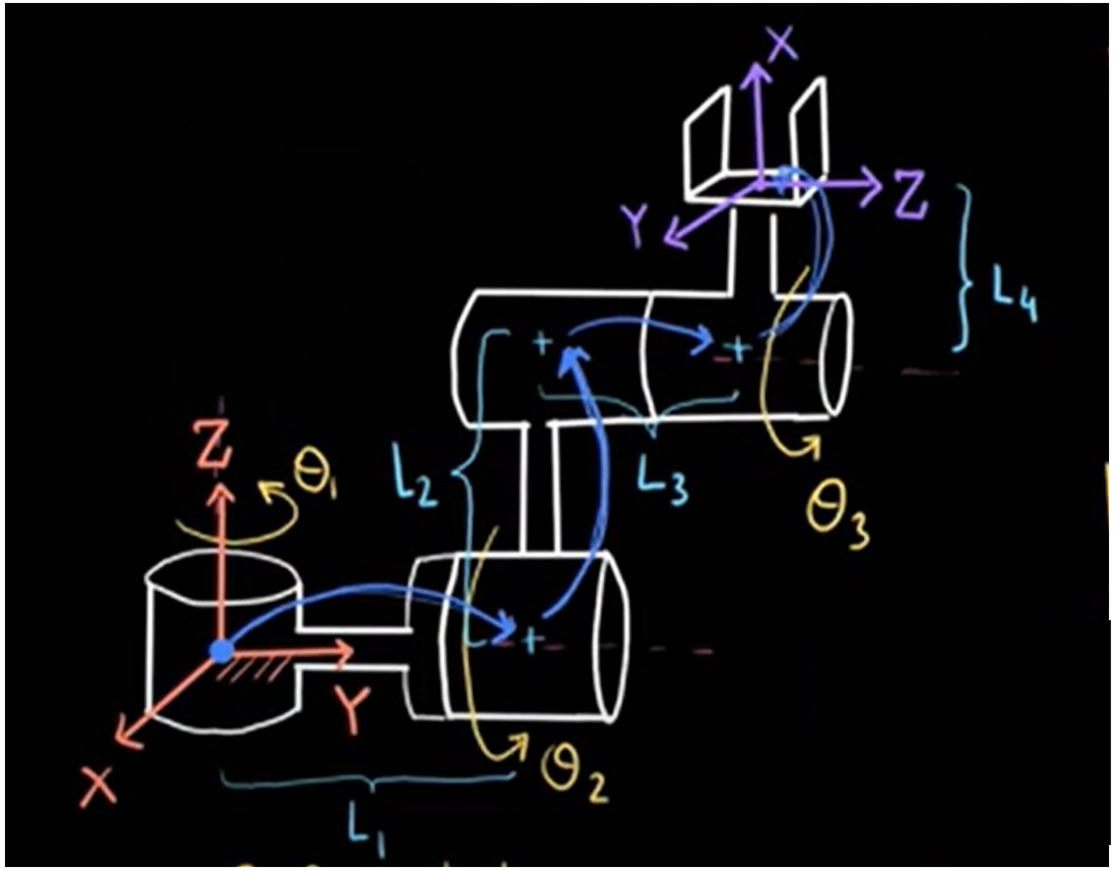


Forward Kinematics using Homogeneous Transformation



Step4: Define a strategy for how to capture translation and rotation when both exist.

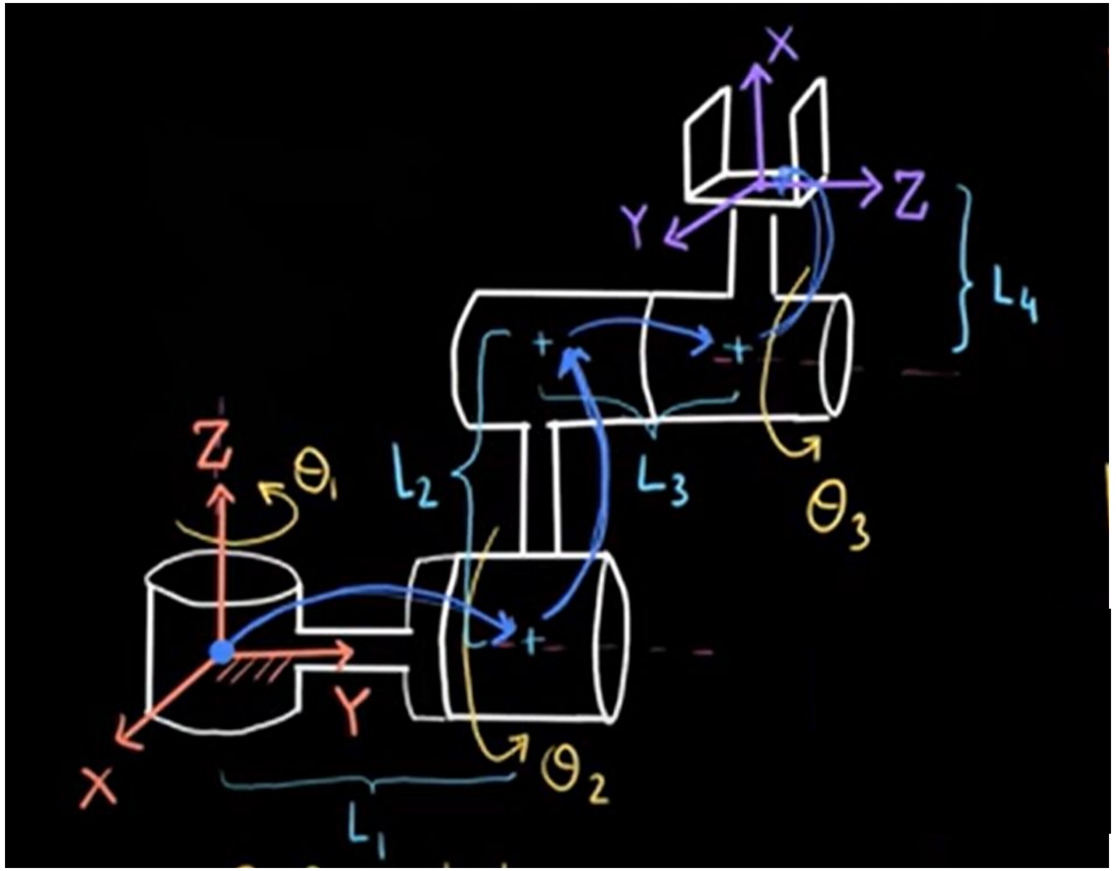
Forward Kinematics using Homogeneous Transformation



Step4: Execution!

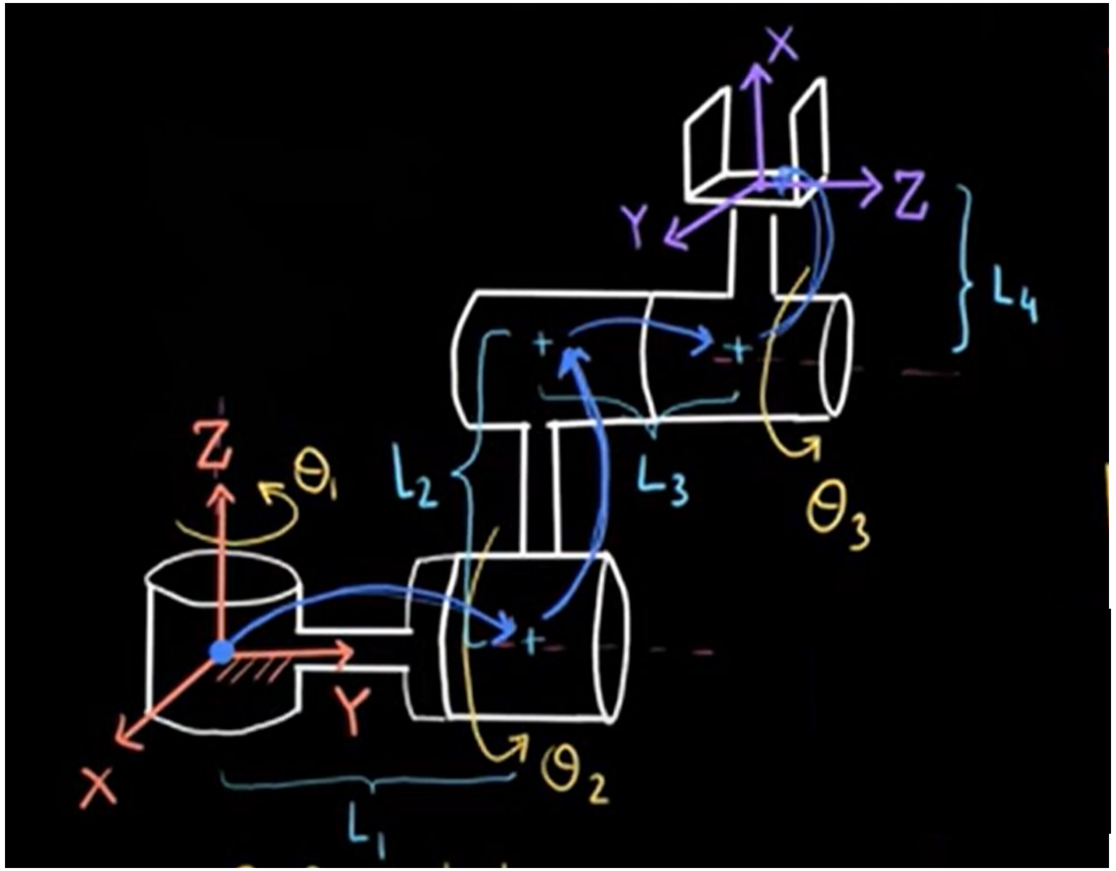
$$H \quad = H_1 H_2\ H_3\ H_4\ H_5$$

Forward Kinematics using Homogeneous Transformation



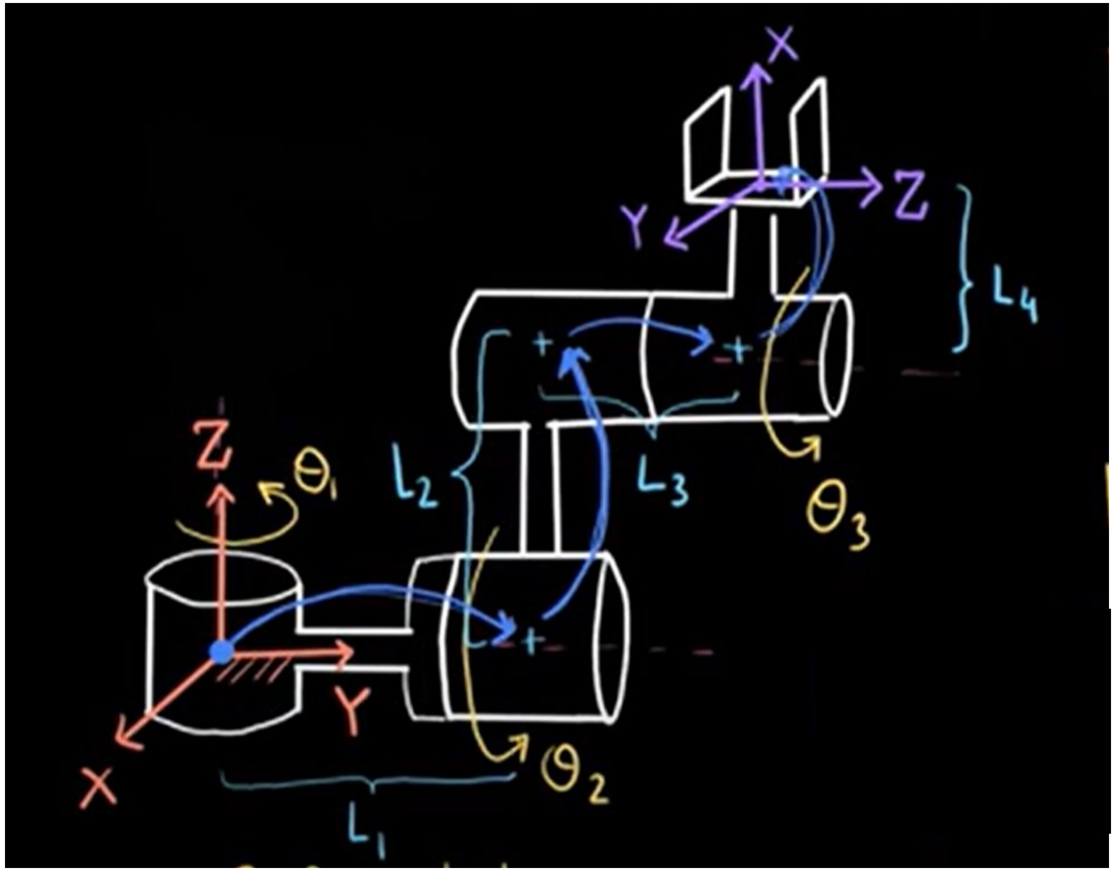
$$H = \begin{bmatrix} Z(\theta_1) & \vec{o} \\ \vec{o}^{T} & 1 \end{bmatrix}$$

Forward Kinematics using Homogeneous Transformation



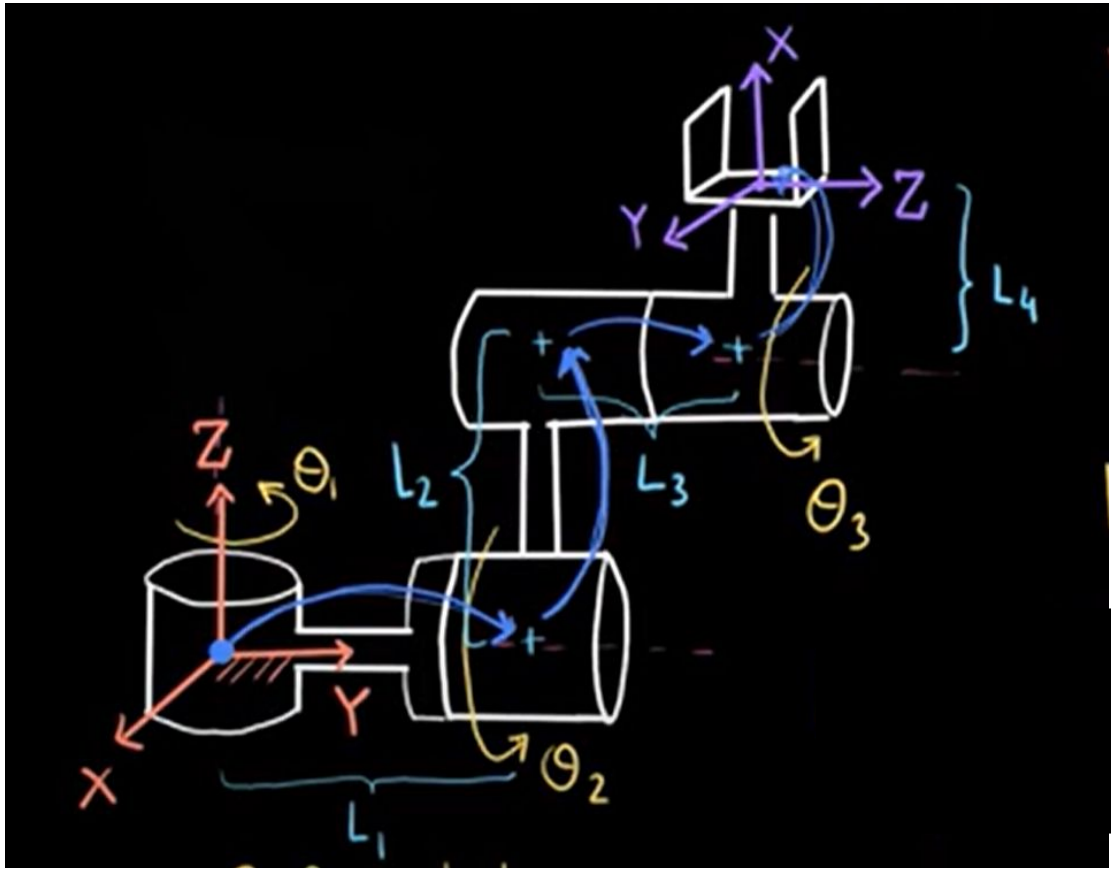
$$H = \begin{bmatrix} Z(\theta_1) & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(\theta_2) & \begin{matrix} 0 \\ L_1 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix}$$

Forward Kinematics using Homogeneous Transformation



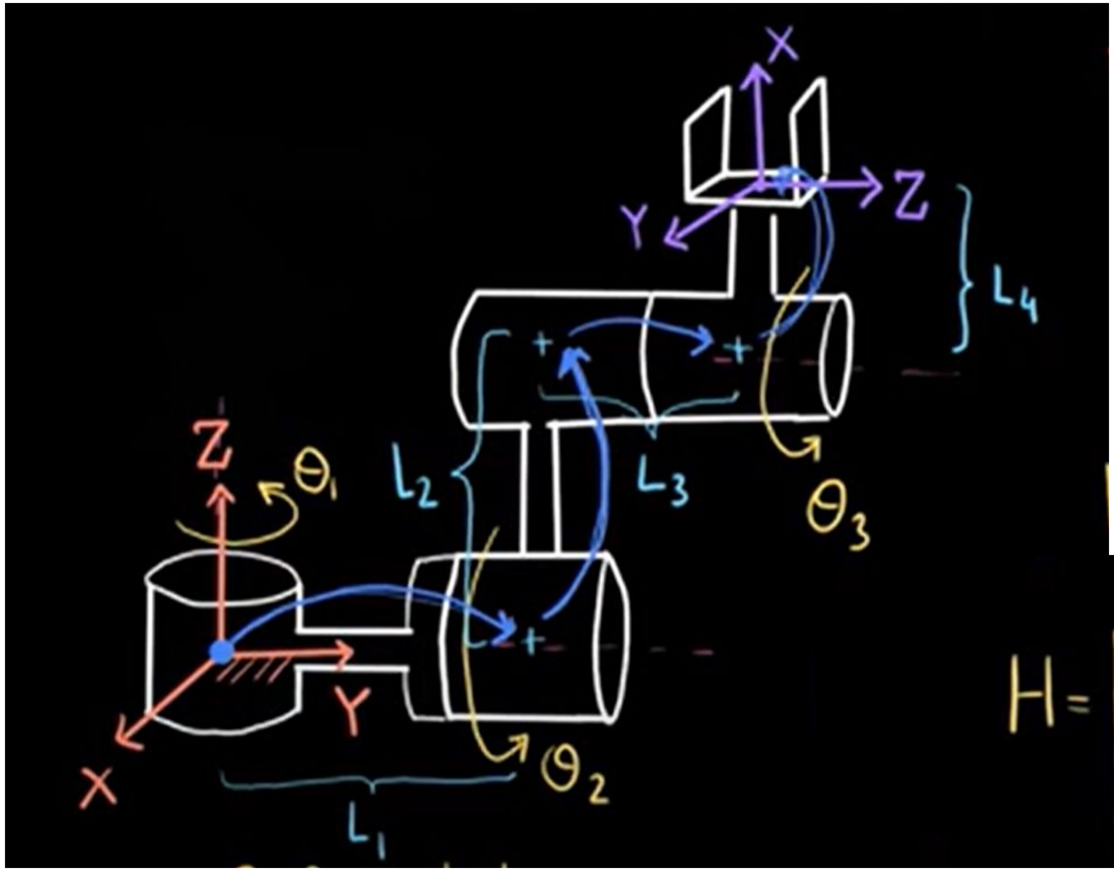
$$H = \begin{bmatrix} Z(\theta_1) & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(\theta_2) & \begin{matrix} 0 \\ L_1 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} I & \begin{matrix} 0 \\ 0 \\ L_2 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix}$$

Forward Kinematics using Homogeneous Transformation



$$H=\begin{bmatrix} Z(\theta_1) & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix}\begin{bmatrix} Y(\theta_2) & \begin{matrix} 0 \\ L_1 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix}\begin{bmatrix} I & \begin{matrix} 0 \\ 0 \\ L_2 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix}\begin{bmatrix} Y(\theta_3) & \begin{matrix} 0 \\ L_3 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix}$$

Forward Kinematics using Homogeneous Transformation



$$H = \begin{bmatrix} Z(\theta_1) & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(\theta_2) & \begin{matrix} 0 \\ L_1 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} I & \begin{matrix} 0 \\ 0 \\ L_2 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(\theta_3) & \begin{matrix} 0 \\ L_3 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(-90^\circ)X(-90^\circ) & \begin{matrix} 0 \\ 0 \\ L_4 \end{matrix} \\ \vec{0} & 1 \end{bmatrix}$$

Forward Kinematics using Homogeneous Transformation

$$H = \begin{bmatrix} Z(\theta_1) & \vec{0} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(\theta_2) & \begin{matrix} 0 \\ L_1 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} I & \begin{matrix} 0 \\ 0 \\ L_2 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(\theta_3) & \begin{matrix} 0 \\ L_3 \\ 0 \end{matrix} \\ \vec{0}^T & 1 \end{bmatrix} \begin{bmatrix} Y(-90^\circ)X(-90^\circ) & \begin{matrix} 0 \\ 0 \\ L_4 \end{matrix} \\ \vec{0} & 1 \end{bmatrix}$$

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

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

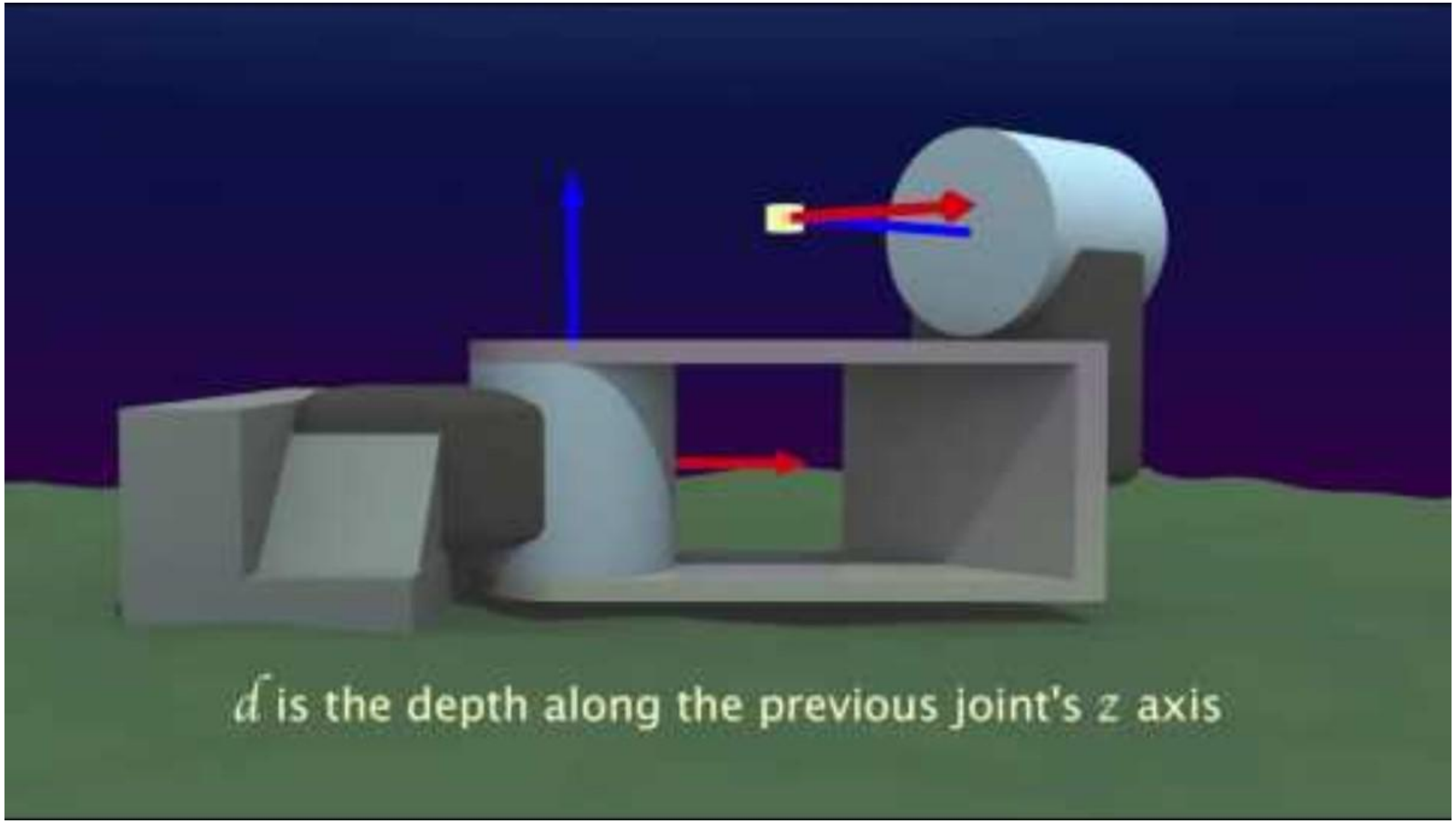
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

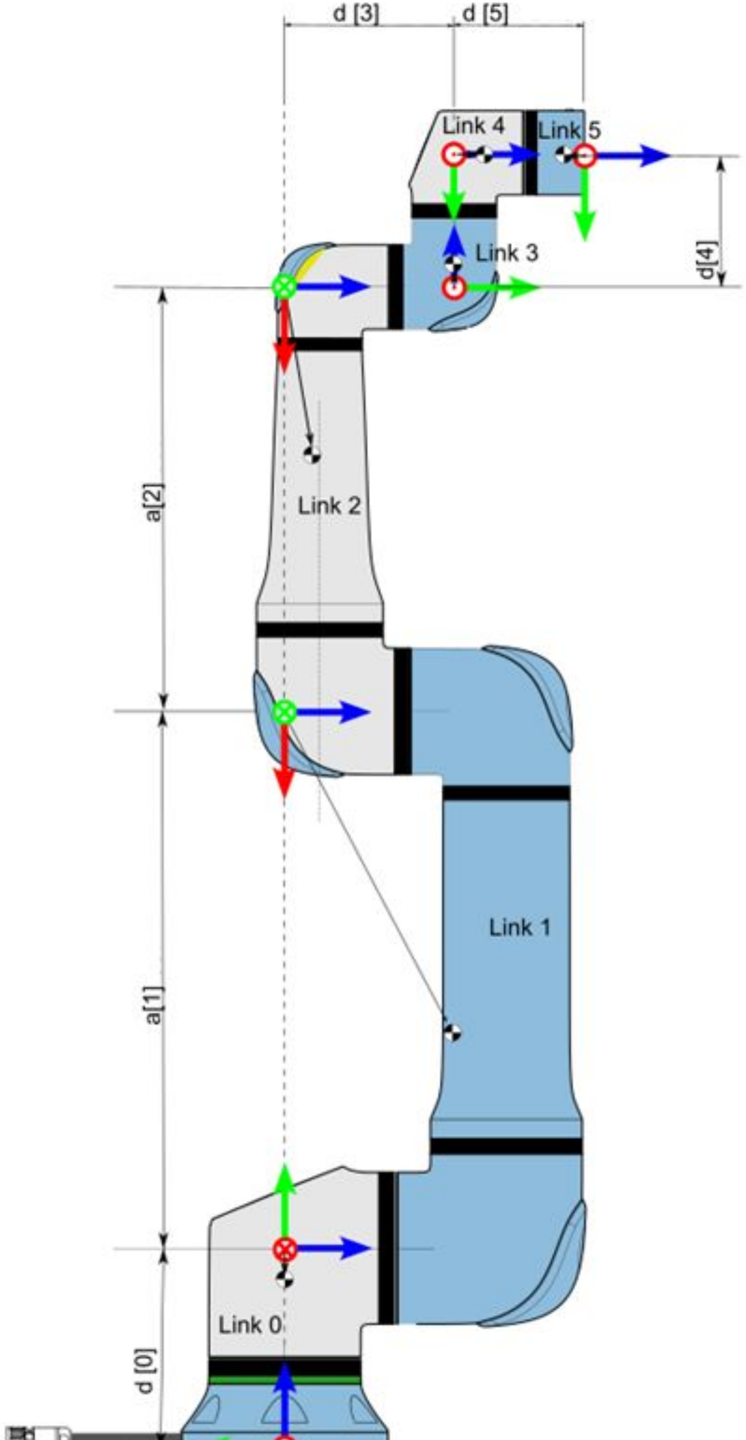
Denavit–Hartenberg (D-H) Parameters

The **D-H convention** is a standardised method to assign coordinate frames to the links and joints of a robot manipulator, simplifying the representation of kinematic chains (Craig, 2005). Each joint is described by **four parameters**:

- θ_i– Joint angle: rotation about the z_{i-1} axis.
- d_i– Joint offset: translation along z_{i-1} axis.
- a_i– Link length: distance between z_{i-1} and z_i along the x_i axis.
- α_i– Link twist: rotation about the x_i , measuring the angle between z_{i-1} and z_i

Denavit–Hartenberg (D-H) Parameters





UR10e/UR12e							
Kinematics	a [m]	d [m]	alpha [rad]	**Dynamics**	Mass [kg]	Center of Mass [m]	Inertia Matrix
Joint 1	0	0.1807	π/2	Link 1	7.369	[0.021, 0.000, 0.027]	[0.0341, 0.0000, -0.0043, 0.0000, 0.0353, 0.0001, -0.0043, 0.0001, 0.0216]
Joint 2	-0.6127	0	0	Link 2	13.051	[0.38, 0.000, 0.158]	[0.0281, 0.0001, -0.0156, 0.0001, 0.7707, 0.0000, -0.0156, 0.0000, 0.7694]
Joint 3	-0.57155	0	0	Link 3	3.989	[0.24, 0.000, 0.068]	[0.0101, 0.0001, 0.0092, 0.0001, 0.3093, 0.0000, 0.0092, 0.0000, 0.3065]
Joint 4	0	0.17415	π/2	Link 4	2.1	[0.000, 0.007, 0.018]	[0.0030, -0.0000, -0.0000, -0.0000, 0.0022, -0.0002, -0.0000, -0.0002, 0.0026]
Joint 5	0	0.11985	-π/2	Link 5	1.98	[0.000, 0.007, 0.018]	[0.0030, -0.0000, -0.0000, -0.0000, 0.0022, -0.0002, -0.0000, -0.0002, 0.0026]
Joint 6	0	0.11655	0	Link 6	0.615	[0, 0, -0.026]	[0.0000, 0.0000, -0.0000, 0.0000, 0.0004, 0.0000, -0.0000, 0.0000, 0.0003]

Denavit–Hartenberg (D-H) Parameters

$$ {}^{i-1}_{\ \ i}T = \begin{pmatrix} \mathrm{Cos}[\theta_i] & -\mathrm{Sin}[\theta_i] & 0 & a_{i-1} \\ \mathrm{Cos}[\alpha_{i-1}]\,\mathrm{Sin}[\theta_i] & \mathrm{Cos}[\alpha_{i-1}]\,\mathrm{Cos}[\theta_i] & -\mathrm{Sin}[\alpha_{i-1}] & -\mathrm{Sin}[\alpha_{i-1}]\,d_i \\ \mathrm{Sin}[\alpha_{i-1}]\,\mathrm{Sin}[\theta_i] & \mathrm{Cos}[\theta_i]\,\mathrm{Sin}[\alpha_{i-1}] & \mathrm{Cos}[\alpha_{i-1}] & \mathrm{Cos}[\alpha_{i-1}]\,d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

Denavit–Hartenberg (D-H) Parameters

1) Revolute joint (numeric example)

Given

$\theta = 45^\circ, d = 0.20\,\mathrm{m}, a = 0.50\,\mathrm{m}, \alpha = 90^\circ$

$(c\theta = s\theta = \frac{\sqrt{2}}{2} \approx 0.7071;\ c\alpha = 0,\ s\alpha = 1)$

$$T = \begin{bmatrix} c\theta & -s\theta\, c\alpha & s\theta\, s\alpha & a\, c\theta \\ s\theta & c\theta\, c\alpha & -c\theta\, s\alpha & a\, s\theta \\ 0 & s\alpha & c\alpha & d \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.7071 & 0 & 0.7071 & 0.3536 \\ 0.7071 & 0 & -0.7071 & 0.3536 \\ 0 & 1 & 0 & 0.2000 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Manipulator Jacobian

- The Jacobian matrix relates joint velocities to end-effector velocities in robotic manipulators.

$$x^{\cdot} = J(q)q^{\cdot}$$

Where

- $x^{\cdot}$: End-effector velocity (linear + angular)
- J(q): Jacobian matrix, depends on joint configuration q
- q ˙ : joint velocity vector

Manipulator Jacobian

- For an n-DOF manipulator in 3D space:

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix}$$

- $Jv(q)$: maps joint velocities to linear velocity of the end-effector.
- $J_\omega(q)$: maps joint velocities to angular velocity.

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \\ \Delta q_i \\ \Delta q_j \\ \Delta q_k \end{bmatrix} = \begin{bmatrix} \frac{\partial X}{\partial \theta_1} & \frac{\partial X}{\partial \theta_2} & \frac{\partial X}{\partial \theta_3} & \frac{\partial X}{\partial \theta_4} & \frac{\partial X}{\partial \theta_5} & \frac{\partial X}{\partial \theta_6} \\ \frac{\partial Y}{\partial \theta_1} & \frac{\partial Y}{\partial \theta_2} & \frac{\partial Y}{\partial \theta_3} & \frac{\partial Y}{\partial \theta_4} & \frac{\partial Y}{\partial \theta_5} & \frac{\partial Y}{\partial \theta_6} \\ \frac{\partial Z}{\partial \theta_1} & \frac{\partial Z}{\partial \theta_2} & \frac{\partial Z}{\partial \theta_3} & \frac{\partial Z}{\partial \theta_4} & \frac{\partial Z}{\partial \theta_5} & \frac{\partial Z}{\partial \theta_6} \\ \frac{\partial q_i}{\partial \theta_1} & \frac{\partial q_i}{\partial \theta_2} & \frac{\partial q_i}{\partial \theta_3} & \frac{\partial q_i}{\partial \theta_4} & \frac{\partial q_i}{\partial \theta_5} & \frac{\partial q_i}{\partial \theta_6} \\ \frac{\partial q_j}{\partial \theta_1} & \frac{\partial q_j}{\partial \theta_2} & \frac{\partial q_j}{\partial \theta_3} & \frac{\partial q_j}{\partial \theta_4} & \frac{\partial q_j}{\partial \theta_5} & \frac{\partial q_j}{\partial \theta_6} \\ \frac{\partial q_k}{\partial \theta_1} & \frac{\partial q_k}{\partial \theta_2} & \frac{\partial q_k}{\partial \theta_3} & \frac{\partial q_k}{\partial \theta_4} & \frac{\partial q_k}{\partial \theta_5} & \frac{\partial q_k}{\partial \theta_6} \end{bmatrix} \cdot \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5 \\ \theta_6 \end{bmatrix}$$

Thus:

$$\begin{bmatrix} \dot{p} \\ \omega \end{bmatrix} = J(q)\,\dot{q}$$

Manipulator Jacobian

- Each **column of the Jacobian** represents the contribution of one joint's motion to the end-effector's velocity.
- For **revolute joints**, the column is:

$$J_i = \begin{bmatrix} z_i \times (p - o_i) \\ z_i \end{bmatrix}$$

- For **prismatic joints**, the column is:

$$J_i = \begin{bmatrix} z_i \\ 0 \end{bmatrix}$$

Where:

- Z_i is the joint axis
- o_i is the joint origin
- p is the end-effector position

Uses of Jacobians in Robotics

- **Kinematics** → compute end-effector velocity from joint velocities.
- **Differential motion** → small changes in joint space vs task space.
- **Statics** → Joint space force to tip force mapping using transpose (J^T). Relates to safety and haptics.
- **Singularity analysis** → determinant of Jacobian indicates loss of mobility.

Key References

- Craig, J. J. (2005). *Introduction to robotics: Mechanics and control* (3rd ed.). Pearson Prentice Hall.
- Murray, R. M., Li, Z., & Sastry, S. S. (1994). *A mathematical introduction to robotic manipulation*. CRC Press.
- Spong, M. W., Hutchinson, S., & Vidyasagar, M. (2006). *Robot modeling and control*. John Wiley & Sons.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics: Modelling, planning and control*. Springer.