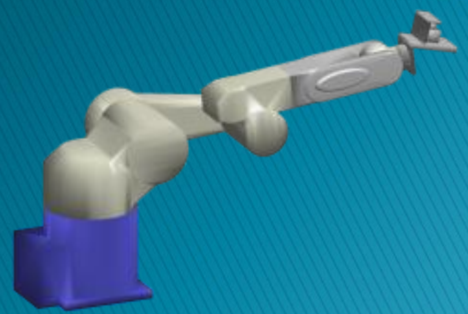
Denavit-Hartenberg (DH) Parameters

© Jon Woolfrey

Centre for Autonomous Systems
University of Technology Sydney



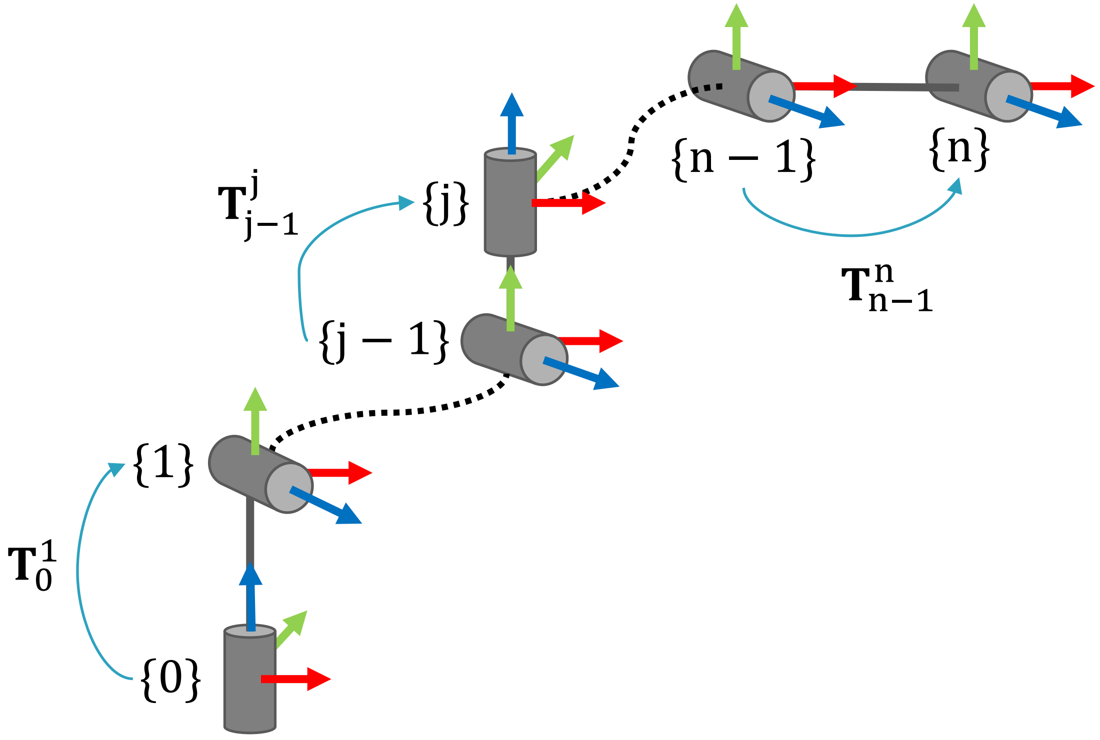
Forward Kinematics Using Transformation Matrices

We can concatenate transformation matrices between joint frames to determine the end-effector pose.

$$\mathbf{T}_0^n = \mathbf{T}_0^1 \times \mathbf{T}_1^2 \times \mathbf{T}_2^3 \times \cdots \times \mathbf{T}_{n-1}^n$$

$$= \prod_{j=1}^{n} \mathbf{T}_{j-1}^{j}$$

Need to describe $\mathbf{T}_{j-1}^{j}$ as a function of simple geometry.



Jon Woolfrey

Denavit–Hartenberg (DH) Parameters

Minimum of 4 parameters, applied in sequence:

1. Rotate about z-axis by θ

$$\mathbf{T}_{\mathrm{Rz}}(\theta) = \begin{bmatrix} \mathbf{R}_z(\theta) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

2. Translate across z-axis by d

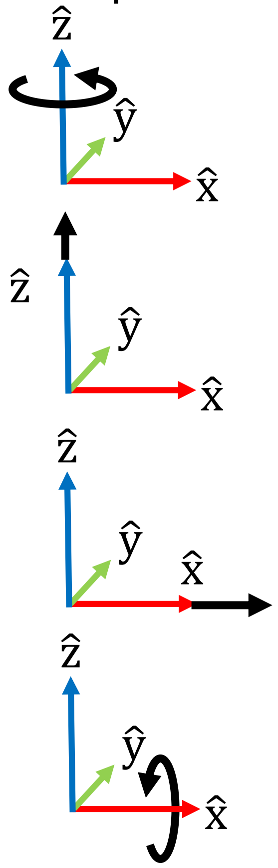
$$\mathbf{T}_z(d) = \begin{bmatrix} & & & 0 \\ & \mathbf{I} & & 0 \\ & & & d \\ & \mathbf{0}_{1\times3} & & 1 \end{bmatrix}$$

3. Translate across x-axis by a

$$\mathbf{T}_x(a) = \begin{bmatrix} & & & a \\ & \mathbf{I} & & 0 \\ & & & 0 \\ & \mathbf{0}_{1\times3} & & 1 \end{bmatrix}$$

4. Rotate about x-axis by α

$$\mathbf{T}_{\mathrm{Rx}}(\alpha) = \begin{bmatrix} \mathbf{R}_x(\alpha) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

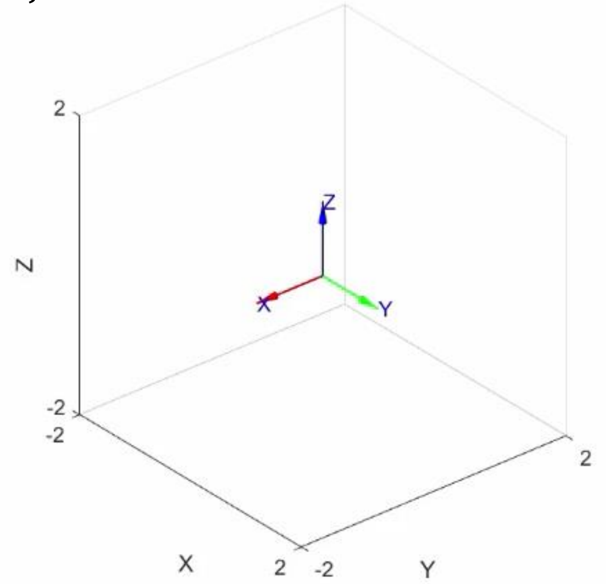


For joint {j−1} to {j}:

$$\mathbf{T}_{j-1}^{j} = \mathbf{T}_{\mathrm{Rz}}(\theta_j)\mathbf{T}_z(d_j)\mathbf{T}_x(a_j)\mathbf{T}_{\mathrm{Rx}}(\alpha_j)$$

Then for the end-effector frame {n}:

$$\mathbf{T}_0^n = \prod_{j=1}^{n} \mathbf{T}_{j-1}^{j}$$



Jon Woolfrey

Why DH Parameters?

Minimum number of parameters to describe forward kinematics

Universal nomenclature

- Knowing DH parameters gives complete knowledge of kinematics
- Easily understood by anyone

Computationally efficient for differential kinematics, dynamics

- Need to minimize no. of calculations for high-frequency feedback control
- Electricity travels **3000 km** in 0.01s (100 Hz)

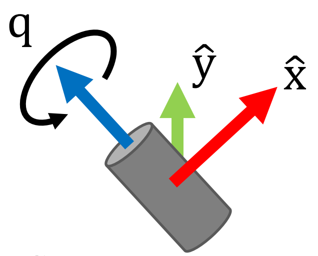
Rules for DH Parameters

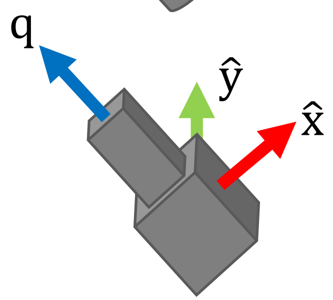
1. Actuate about z-axis
   - Rotate about z for revolute joints

$$\mathbf{T}_{Rz}(q) = \begin{bmatrix} \mathbf{R}_z(q) & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$$

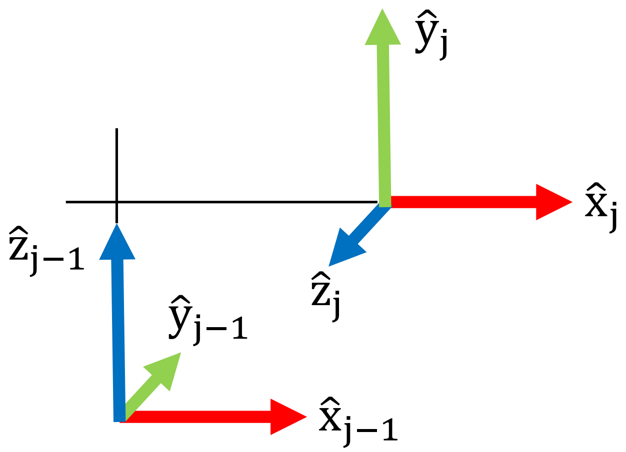
   - Translate about z for prismatic joints

$$\mathbf{T}_z(q) = \begin{bmatrix} & & & 0 \\ & \mathbf{I} & & 0 \\ & & & q \\ & \mathbf{0}_{1\times 3} & & 1 \end{bmatrix}$$





2. Axis $\hat{z}_{j-1}$ is perpendicular to, and intersects, $\hat{x}_j$
3. The y-axis is solved implicitly: $\hat{y}_j = \hat{z}_j \times \hat{x}_j$

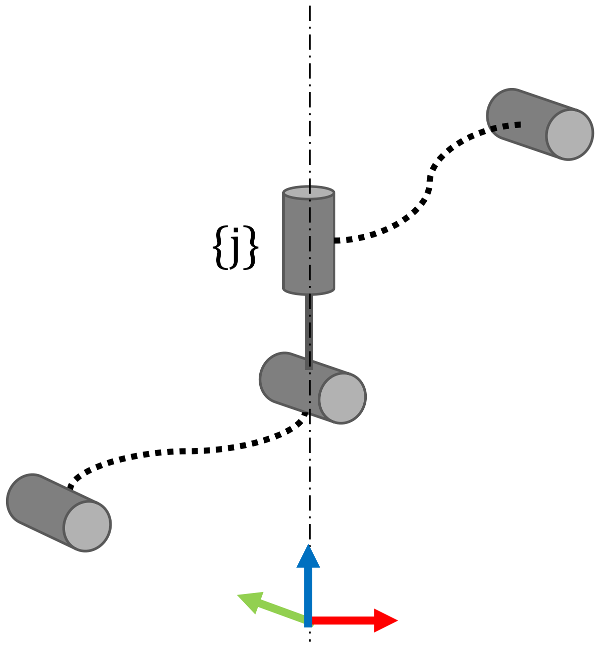


Tips for DH Parameters

- The joint frame does not need to physically coincide with the actual joint.

  It only needs to align with the axis of actuation.

-

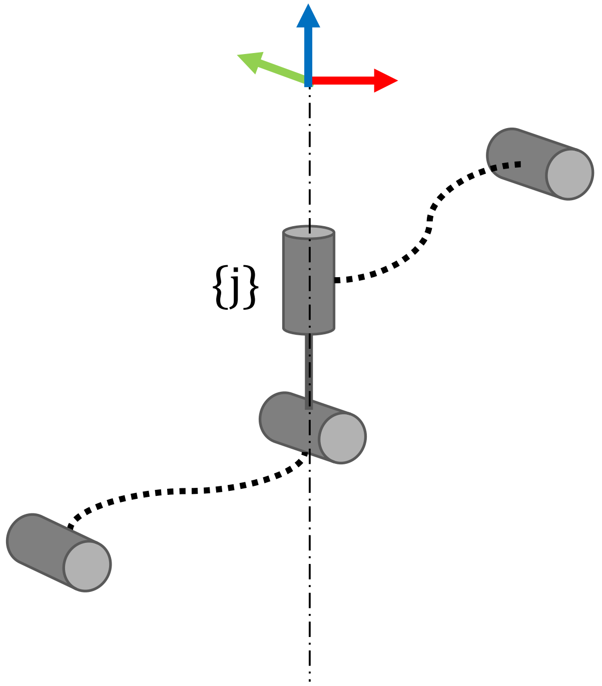


Tips for DH Parameters

- The joint frame does not need to physically coincide with the actual joint.

  It only needs to align with the axis of actuation.

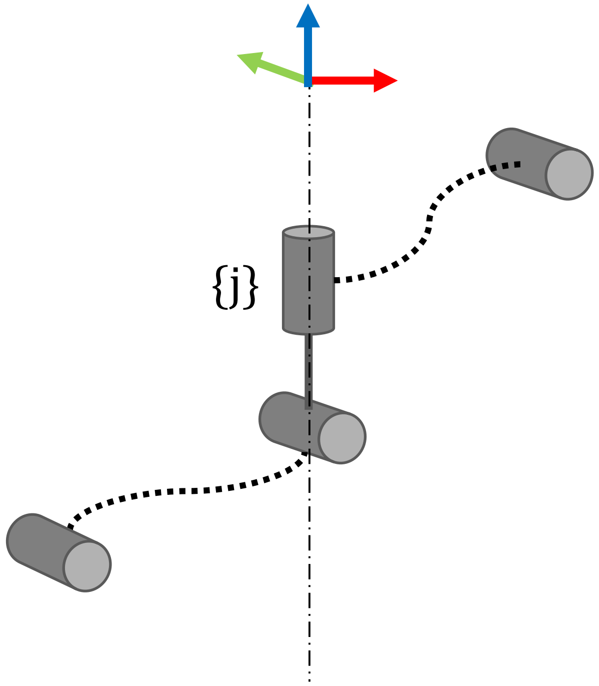
-



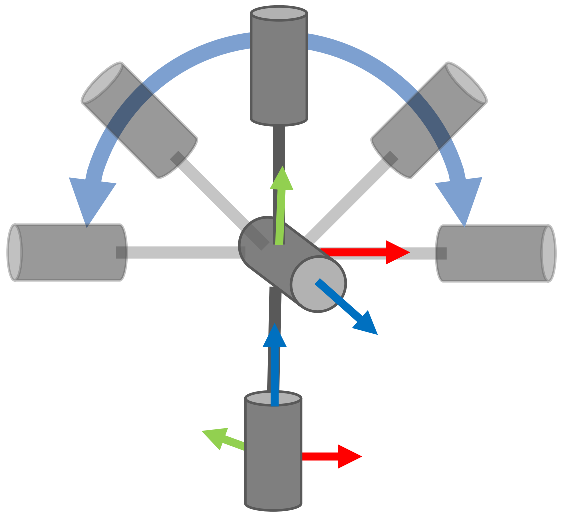
Tips for DH Parameters

- The joint frame does not need to physically coincide with the actual joint.

  It only needs to align with the axis of actuation.



- The robot arm can be arranged in any configuration that suits the DH parameters.



Jon Woolfrey

Denavit–Hartenberg (DH) Parameters

To get forward kinematics at a particular joint configuration $\mathbf{q}$, substitute the joint value in to the **z–component** of the transform chain:

$$\mathbf{T}_{j-1}^{j} = \begin{cases} \mathbf{T}_{Rz}(q_j)\mathbf{T}_{z}(d_j)\mathbf{T}_{x}(a_j)\mathbf{T}_{Rx}(\alpha_j) & \text{for Revolute} \\ \mathbf{T}_{Rz}(\theta_j)\mathbf{T}_{z}(q_j)\mathbf{T}_{x}(a_j)\mathbf{T}_{Rx}(\alpha_j) & \text{for Prismatic} \end{cases}$$

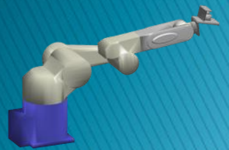
Alternative sequences can be used, for example:

$$\mathbf{T}_{j-1}^{j} = \mathbf{T}_{z}(d_j)\mathbf{T}_{Rz}(\theta_j)\mathbf{T}_{Rx}(\alpha_j)\mathbf{T}_{x}(a_j)$$

However, the order is important.

Matrices are **not** commutative; $\mathbf{AB} \neq \mathbf{BA}$.

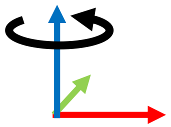
Must use the **same** sequence for **all** joints.



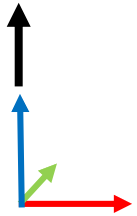
Forward Kinematics of a 3DOF Manipulator

DH Parameters from {0} to {1}:

$$\mathbf{T}_{Rz}(\theta_1) = \begin{bmatrix} \mathbf{R}_z(q_1) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

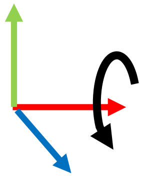


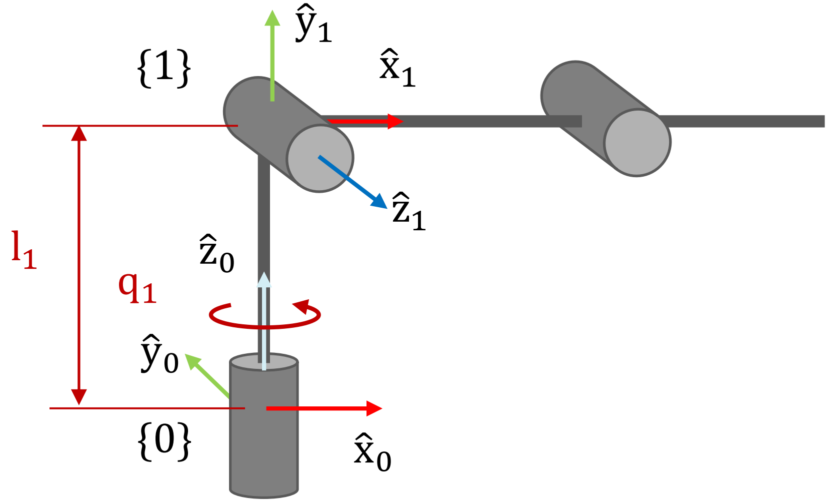
$$\mathbf{T}_z(d_1) = \begin{bmatrix} \mathbf{I}_3 & \begin{matrix} 0 \\ 0 \\ l_1 \end{matrix} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$



$$\mathbf{T}_x(a_1) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \mathbf{I}_4$$

$$\mathbf{T}_{Rx}(\alpha_1) = \begin{bmatrix} \mathbf{R}_x(\pi/2) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

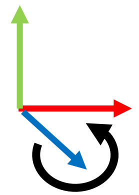




Forward Kinematics of a 3DOF Manipulator

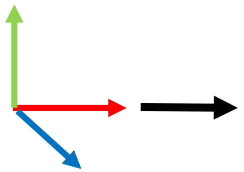
DH Parameters from {1} to {2}:

$$\mathbf{T}_{Rz}(\theta_2) = \begin{bmatrix} \mathbf{R}_z(q_2) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

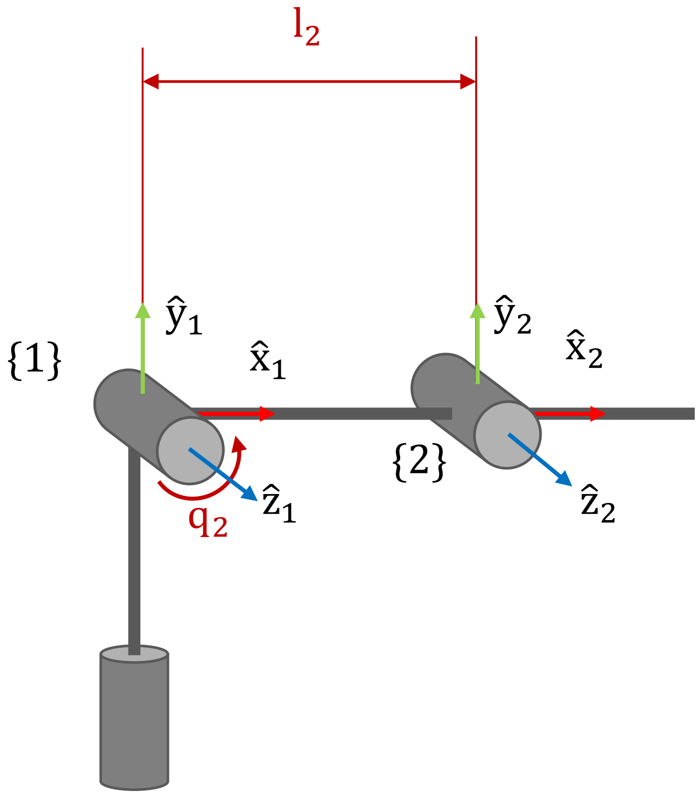


$$\mathbf{T}_z(d_2) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \mathbf{I}_4$$

$$\mathbf{T}_x(a_2) = \begin{bmatrix} \mathbf{I}_3 & \begin{matrix} l_2 \\ 0 \\ 0 \end{matrix} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$



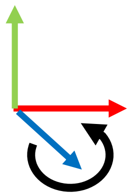
$$\mathbf{T}_{Rx}(\alpha_2) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \mathbf{I}_4$$



Forward Kinematics of a 3DOF Manipulator

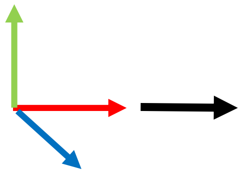
DH Parameters from {2} to {3}:

$$\mathbf{T}_{Rz}(\theta_3) = \begin{bmatrix} \mathbf{R}_z(q_3) & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

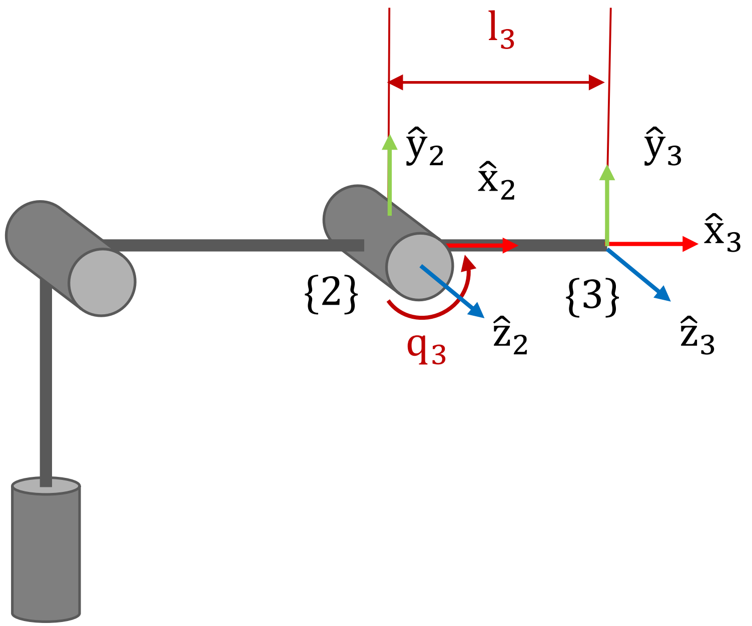


$$\mathbf{T}_z(d_3) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \mathbf{I}_4$$

$$\mathbf{T}_x(a_3) = \begin{bmatrix} & & & l_3 \\ & \mathbf{I}_3 & & 0 \\ & & & 0 \\ & \mathbf{0}_{1\times3} & & 1 \end{bmatrix}$$



$$\mathbf{T}_{Rx}(\alpha_3) = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} = \mathbf{I}_4$$



3DOF Manipulator With Prismatic Joint

Frame {0} to {1}:

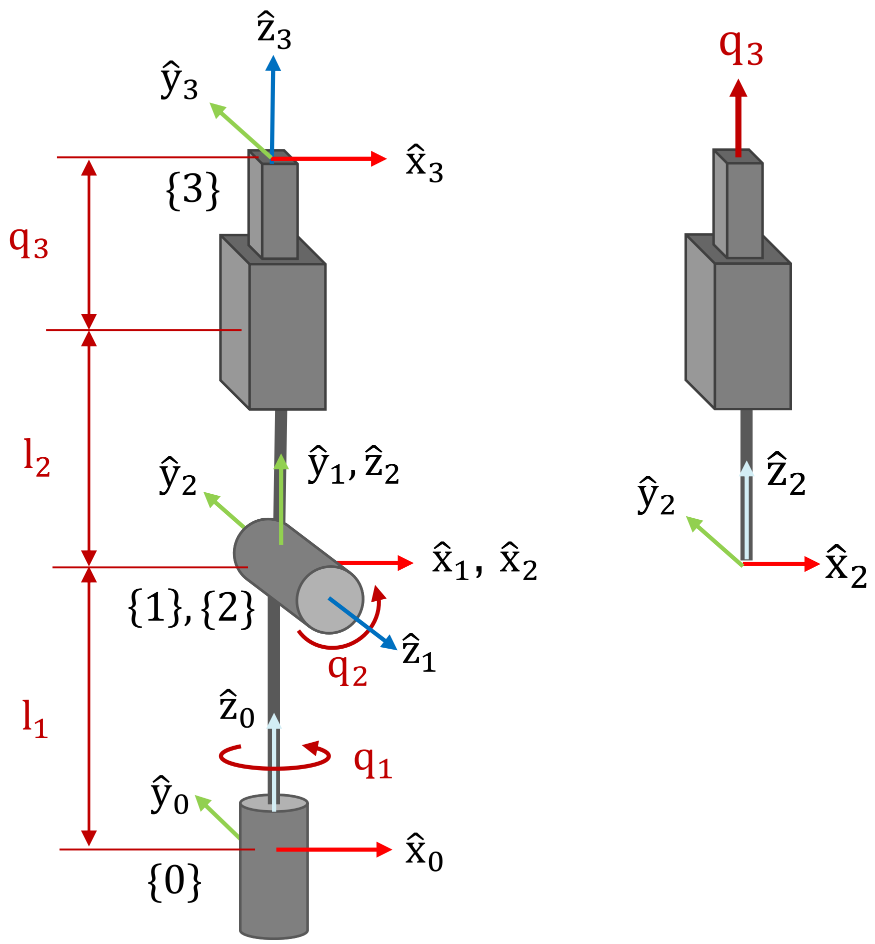
$$\mathbf{T}_0^1 = \mathbf{T}_{Rz}(q_1)\mathbf{T}_z(l_1)\mathbf{T}_x(0)\mathbf{T}_{Rx}(\pi/2)$$

Note that for this case, frame {2} is **coincident** with frame {1}:

$$\mathbf{T}_1^2 = \mathbf{T}_{Rz}(q_2)\mathbf{T}_z(0)\mathbf{T}_x(0)\mathbf{T}_{Rx}(-\pi/2)$$

Since frame {2} is not located on joint 3, we need to add the link offset to the z-translation:

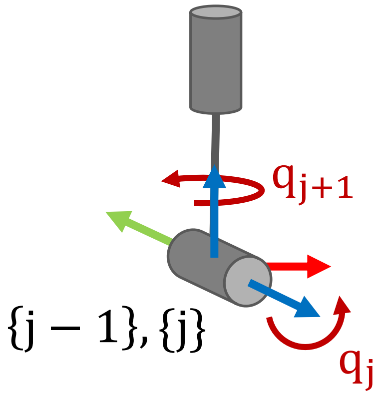
$$\mathbf{T}_2^3 = \mathbf{T}_{Rz}(0)\mathbf{T}_z(l_2 + q_3)\mathbf{T}_x(0)\mathbf{T}_{Rx}(0)$$



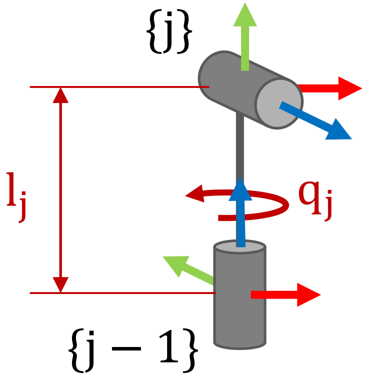
Common Joint-to-Joint Transforms

θ_j	d_j	a_j	α_j
q_j	0	0	$-\frac{\pi}{2}$

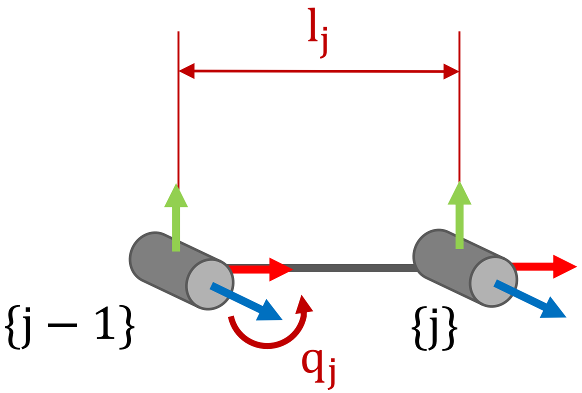
Joint frames are coincident.



θ_j	d_j	a_j	α_j
q_j	l_j	0	$\frac{\pi}{2}$



θ_j	d_j	a_j	α_j
q_j	0	l_j	0



Summary of DH Parameters

Forward Kinematics (FK) can be calculated by multiplying transforms between joint frames:

$$\mathbf{T}_0^n = \prod_{j=1}^{n} \mathbf{T}_{j-1}^{j}$$

DH Parameters use 4 variables to get the transform from frame {j−1} to frame {j}

$$\mathbf{T}_{j-1}^{j} = \begin{cases} \mathbf{T}_{Rz}(q_j)\mathbf{T}_z(d_j)\mathbf{T}_x(a_j)\mathbf{T}_{Rx}(\alpha_j) & \text{for Revolute} \\ \mathbf{T}_{Rz}(\theta_j)\mathbf{T}_z(q_j)\mathbf{T}_x(a_j)\mathbf{T}_{Rx}(\alpha_j) & \text{for Prismatic} \end{cases}$$

θ_j	A rotation about the z-axis
d_j	A translation about the z-axis
a_j	A translation about the x-axis
α_j	A rotation about the x-axis