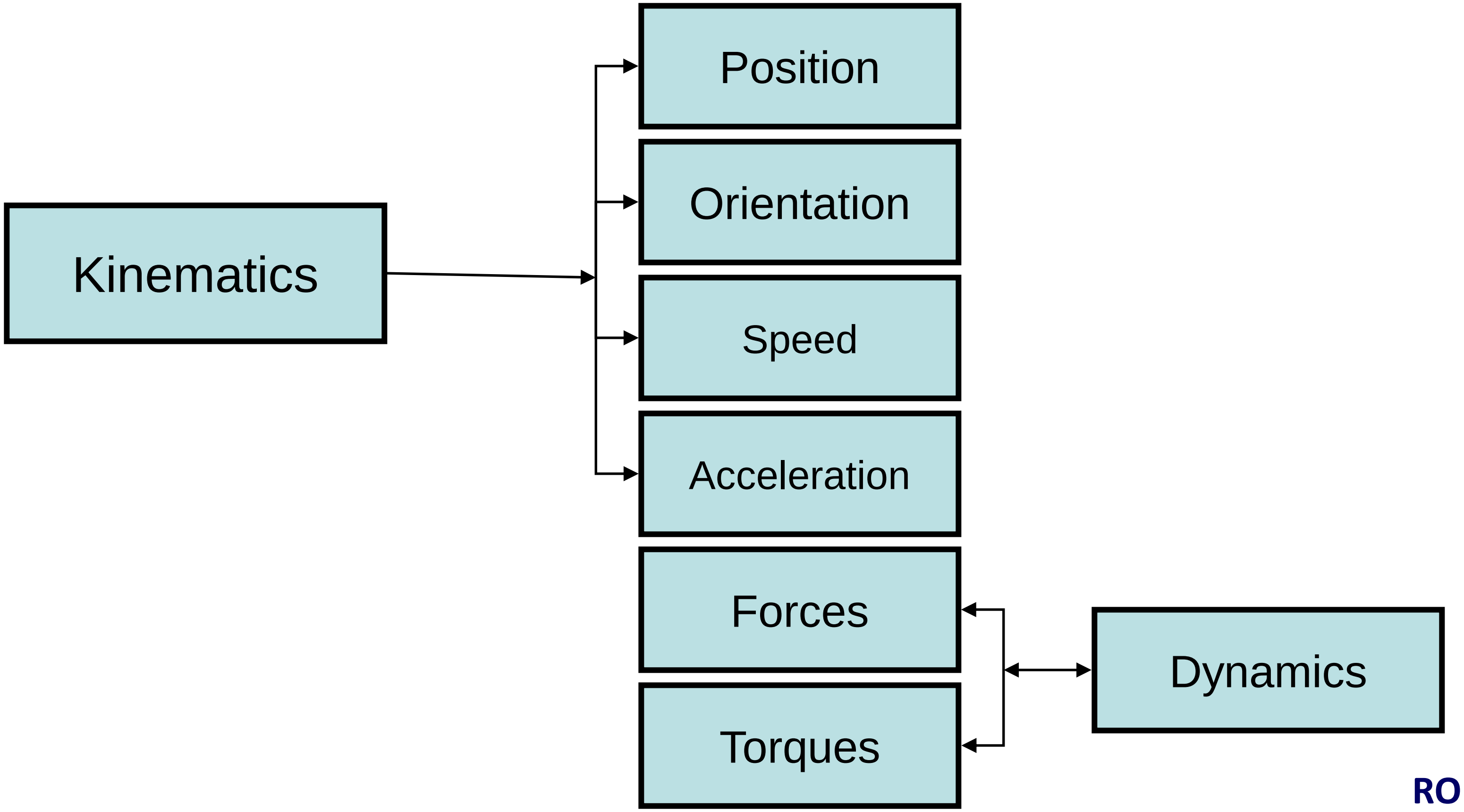
ROBOTICS KINEMATICS

Forward Kinematics

By Mustafa Shiple

- Task of kinematics is to describe the location of systems in space.



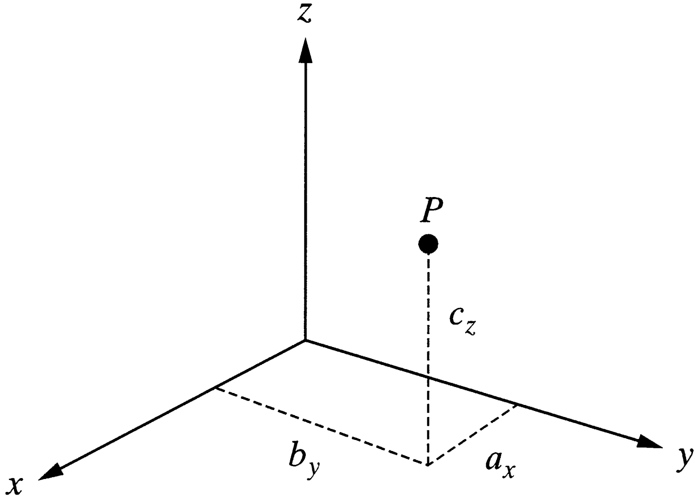
Kinematics

- Forward Kinematics:
  to determine where the robot's hand is?
  (If all joint variables are known)

- Inverse Kinematics:
  to calculate what each joint variable is?
  (If we desire that the hand be located at a particular point)

Matrix Representation(point)

- A point *P* in space :
  3 coordinates relative to a reference frame



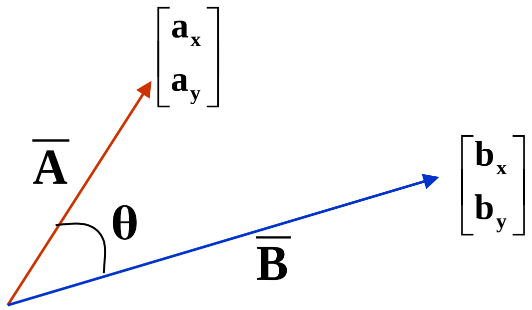
$$P = a_x \hat{i} + b_y \hat{j} + c_z \hat{k}$$

$$P = \begin{bmatrix} a_x \\ b_y \\ c_z \end{bmatrix}$$

Quick Math Review (Dot Product)

Geometric Representation:

$$\bar{A} \bullet \bar{B} = \|A\| \ \|B\| \ \cos\theta$$



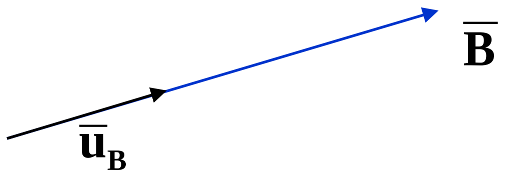
Matrix Representation:

$$\bar{A} \bullet \bar{B} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} \bullet \begin{bmatrix} b_x \\ b_y \end{bmatrix} = a_x b_x + a_y b_y$$

Unit Vector

Vector in the direction of a chosen vector but whose magnitude is 1.

$$\bar{u}_B = \frac{\bar{B}}{\|B\|}$$



Quick Matrix Review (Matrix Multiplication)

An (m x n) matrix A and an (n x p) matrix B, can be multiplied since the number of columns of A is equal to the number of rows of B.

Non-Commutative Multiplication

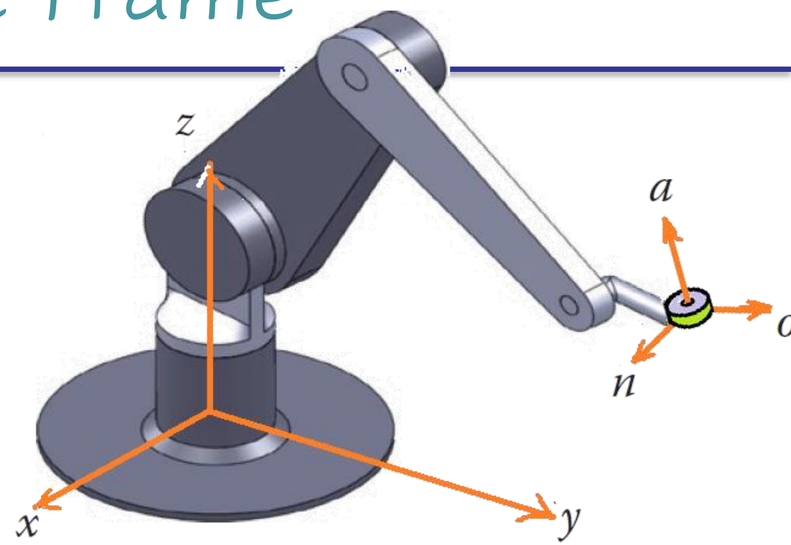
AB is NOT equal to BA

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} * \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} (ae+bg) & (af+bh) \\ (ce+dg) & (cf+dh) \end{bmatrix}$$

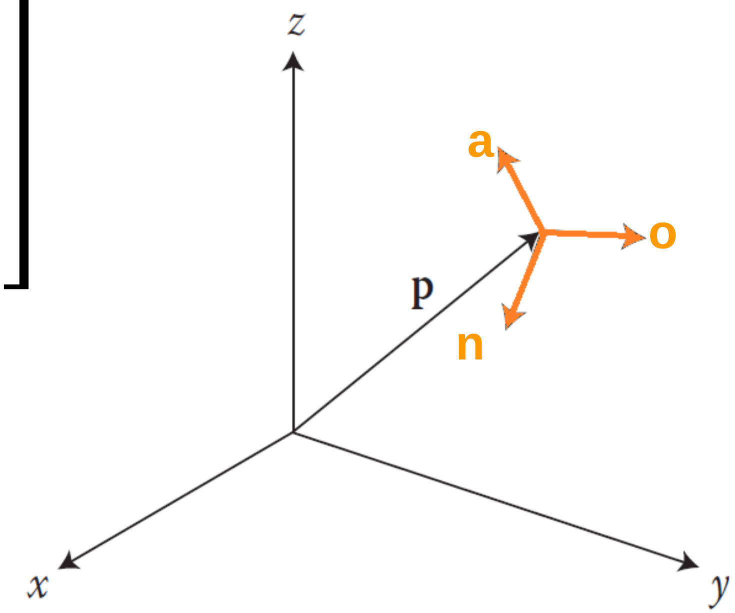
Matrix Addition:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} + \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} (a+e) & (b+f) \\ (c+g) & (d+h) \end{bmatrix}$$

Representation of a Frame Relative to a Fixed Reference Frame

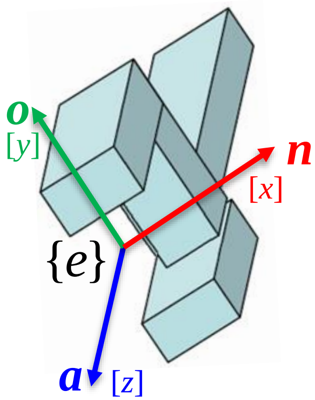


$$F = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



What's n, o, and a

- (almost) By convention:



$\{e\}$: end effector frame

  - $\boldsymbol{a}$: approach vector (aligned with the *roll* axis and pointing outwards)
  - $\boldsymbol{o}$: orientation vector (in the direction of motion of the gripper jaws)
  - $\boldsymbol{n}$: normal vector (orthogonal to the plane defined by o and a)

- Homogeneous transformation matrix:

Pose of the end effector with respect to the robot base

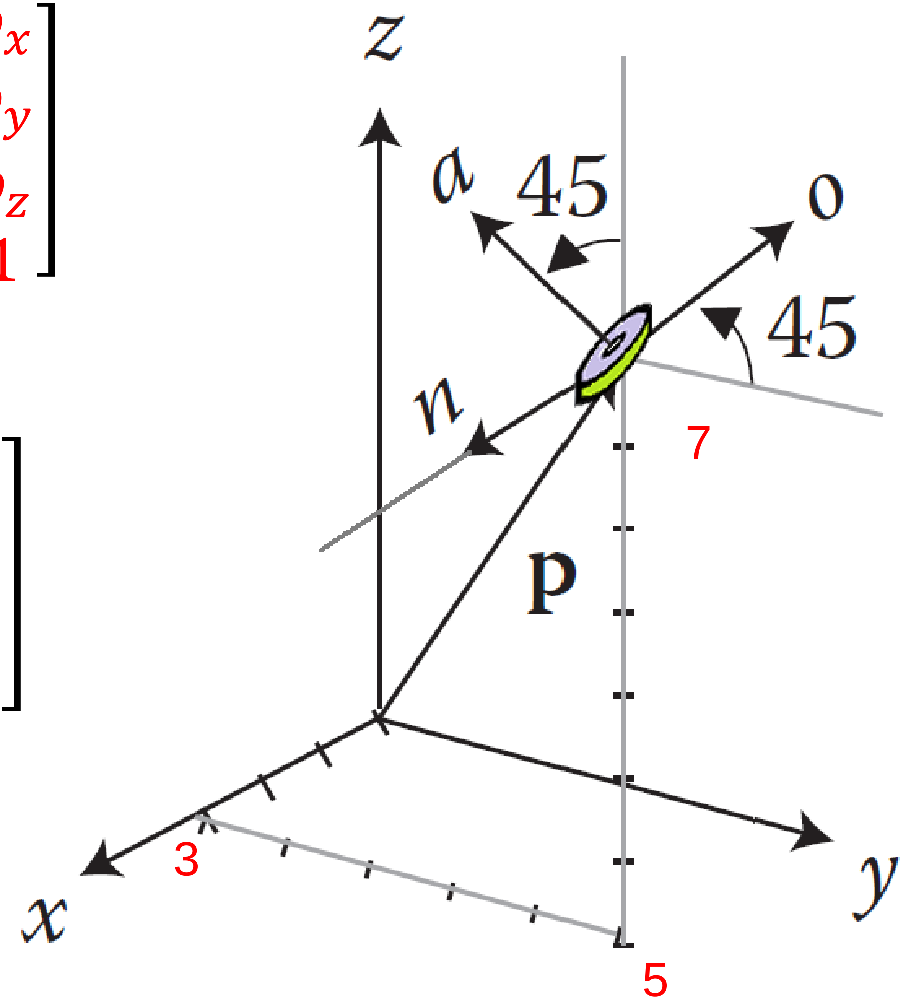
$$^{0}T_e = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(p_x, p_y, p_z): position of $\{e\}$ with respect to the base

Example

$$F = \begin{bmatrix} n_x = 1 & o_x & a_x & p_x \\ n_y = 0 & o_y & a_y & p_y \\ n_z = 0 & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

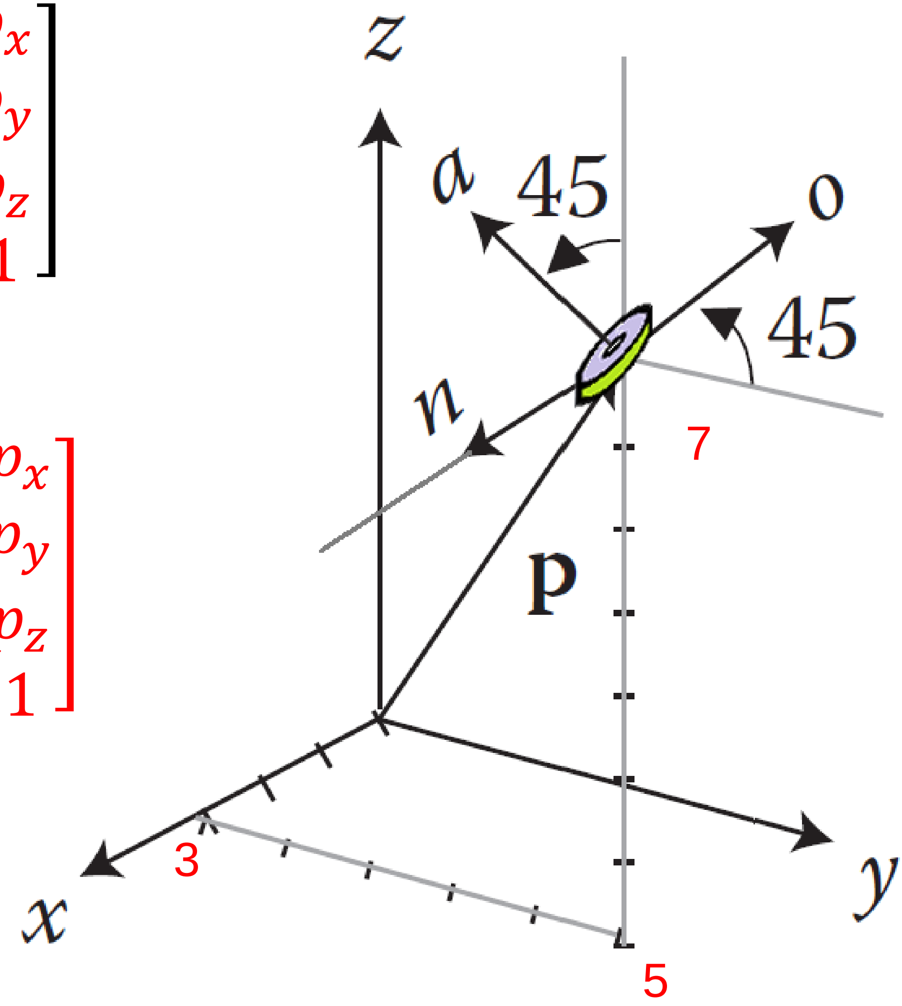
$$F = \begin{bmatrix} 1 & o_x = 0 & a_x & p_x \\ 0 & o_y = & a_y & p_y \\ 0 & o_z = & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Example

$$F = \begin{bmatrix} n_x = 1 & o_x & a_x & p_x \\ n_y = 0 & o_y & a_y & p_y \\ n_z = 0 & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

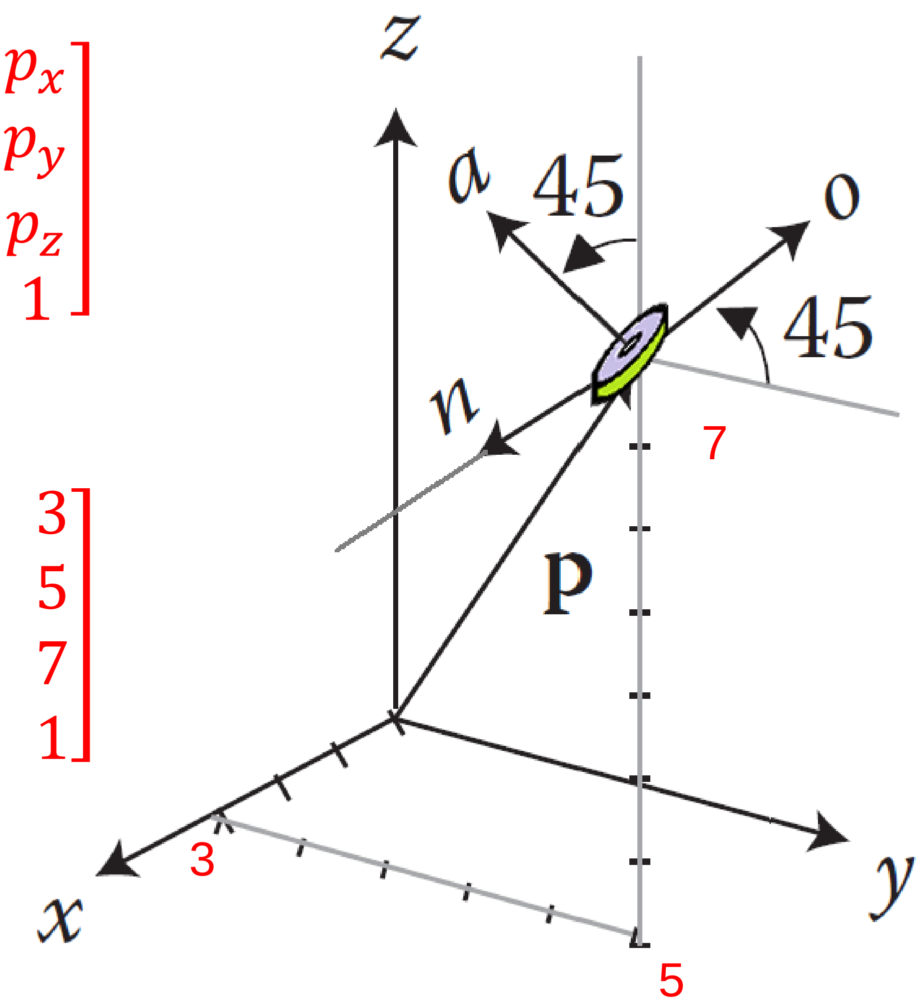
$$F = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 0.707 & -0.707 & p_y \\ 0 & 0.707 & 0.707 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



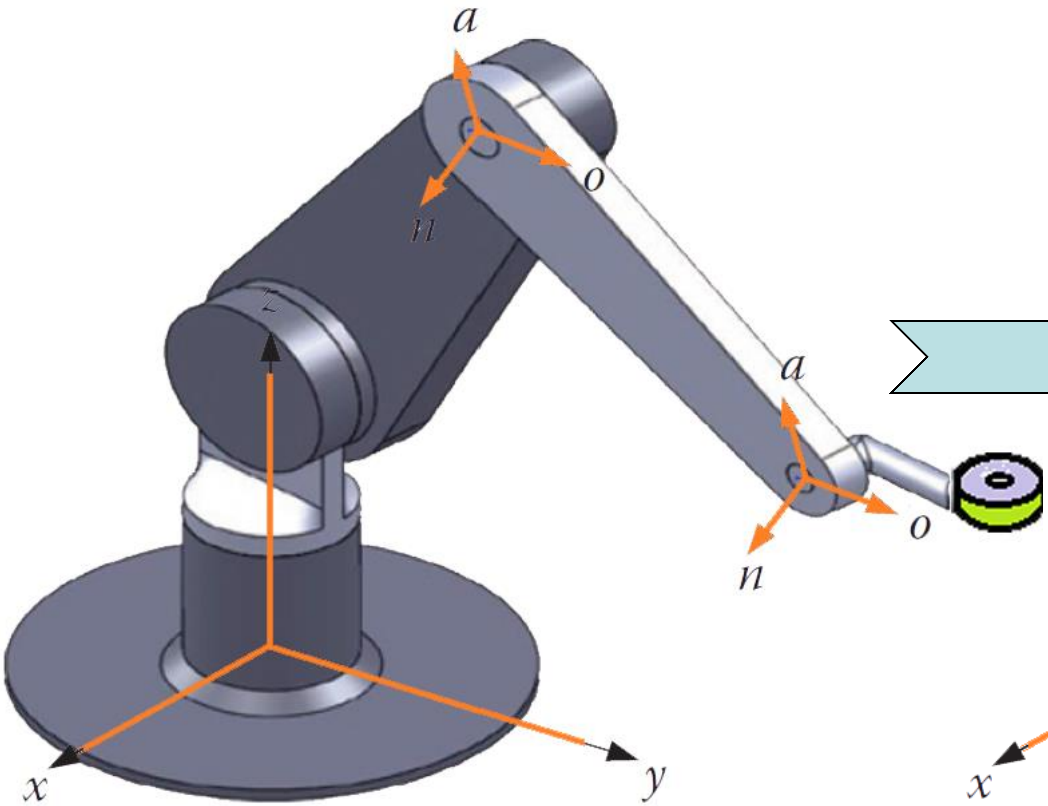
Example

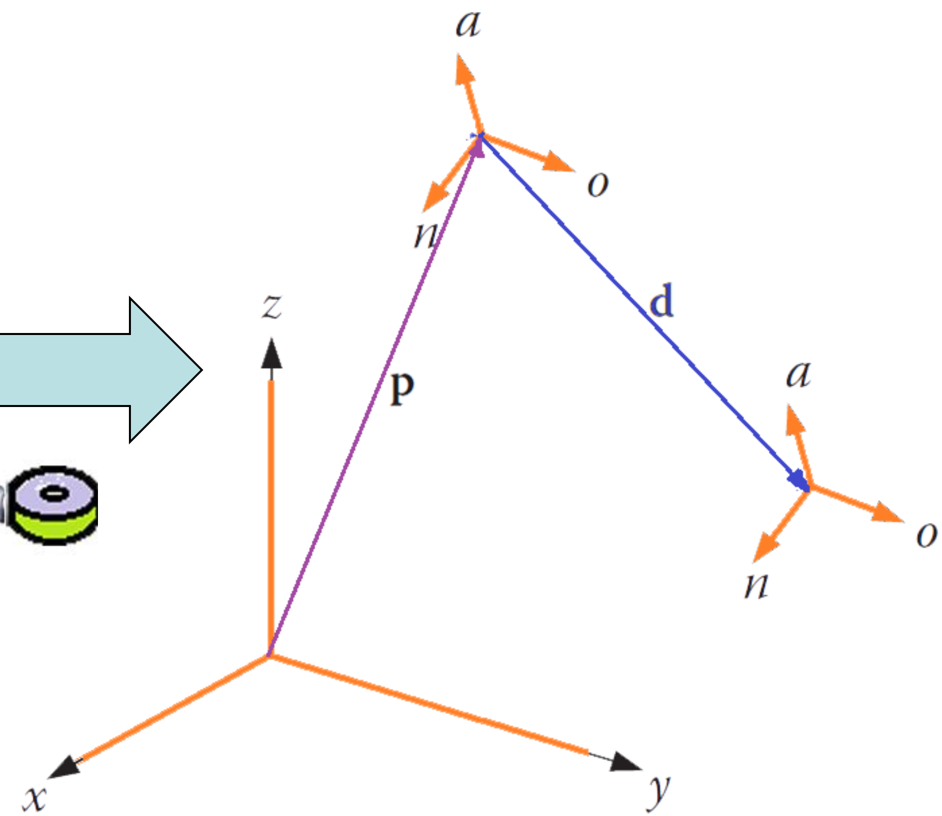
$$F = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 0.707 & -0.707 & p_y \\ 0 & 0.707 & 0.707 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F = \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 0.707 & -0.707 & 5 \\ 0 & 0.707 & 0.707 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

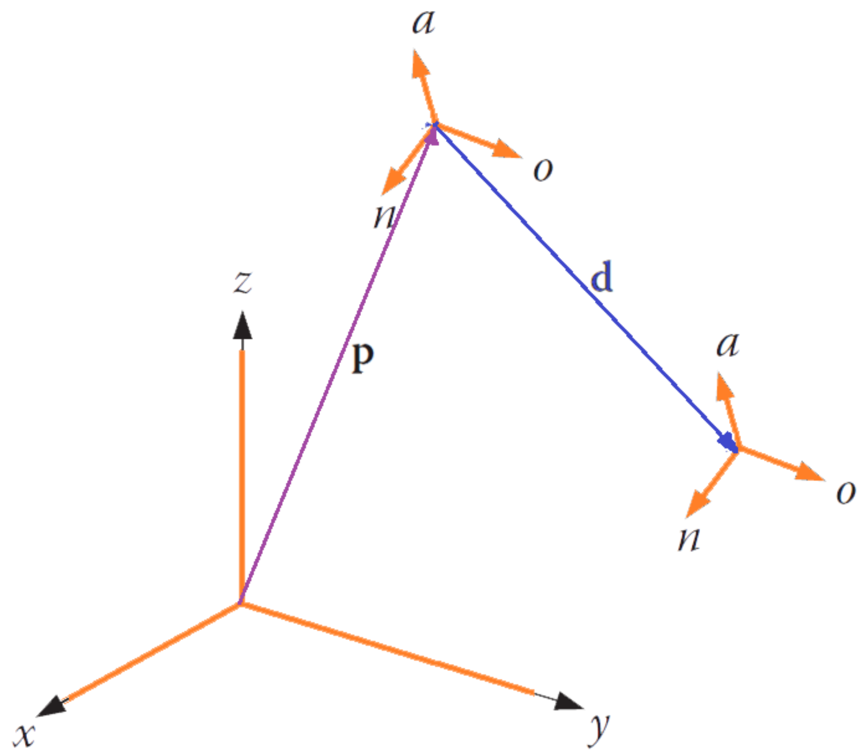


Representation of a Pure Translation





Representation of a Pure Translation

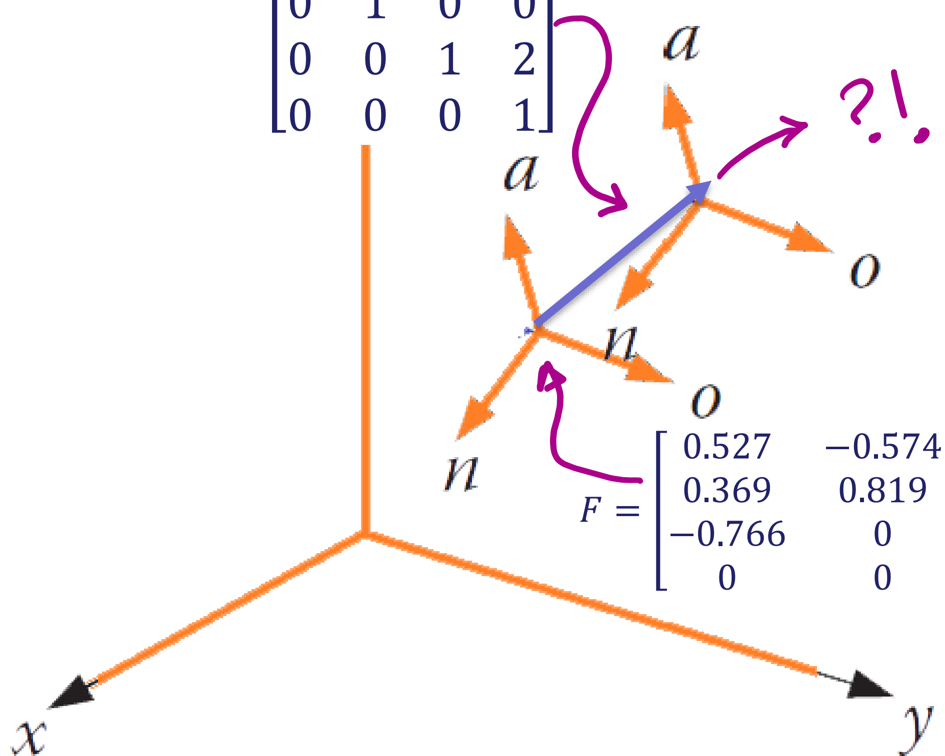


$$F = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x + d_x \\ n_y & o_y & a_y & p_y + d_y \\ n_z & o_z & a_z & p_z + d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

A frame F is moved 3 units along the x-axis and 2 units along the z-axis of the reference frame. Find the new location of the frame.

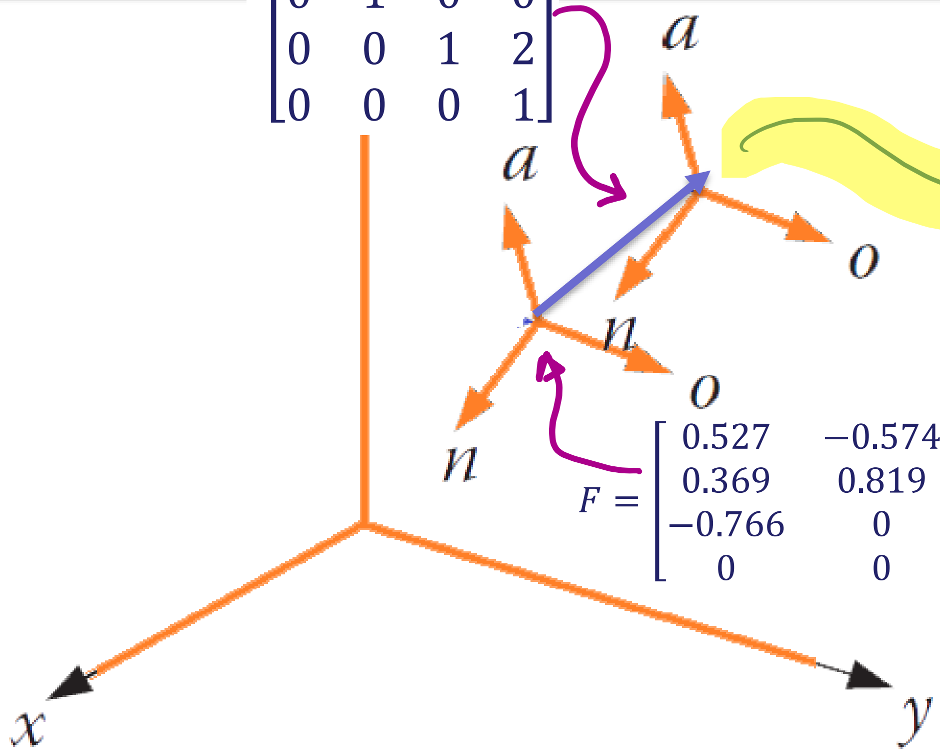
$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$F = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 8 \\ 0.369 & 0.819 & 0.439 & 10 \\ -0.766 & 0 & 0.643 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

$$\begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0.527 & -0.574 & 0.628 & 11 \\ 0.369 & 0.819 & 0.439 & 10 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$F = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 8 \\ 0.369 & 0.819 & 0.439 & 10 \\ -0.766 & 0 & 0.643 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0.527 & -0.574 & 0.628 & 8 \\ 0.369 & 0.819 & 0.439 & 10 \\ -0.766 & 0 & 0.643 & 6 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.527 & -0.574 & 0.628 & 11 \\ 0.369 & 0.819 & 0.439 & 10 \\ -0.766 & 0 & 0.643 & 8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

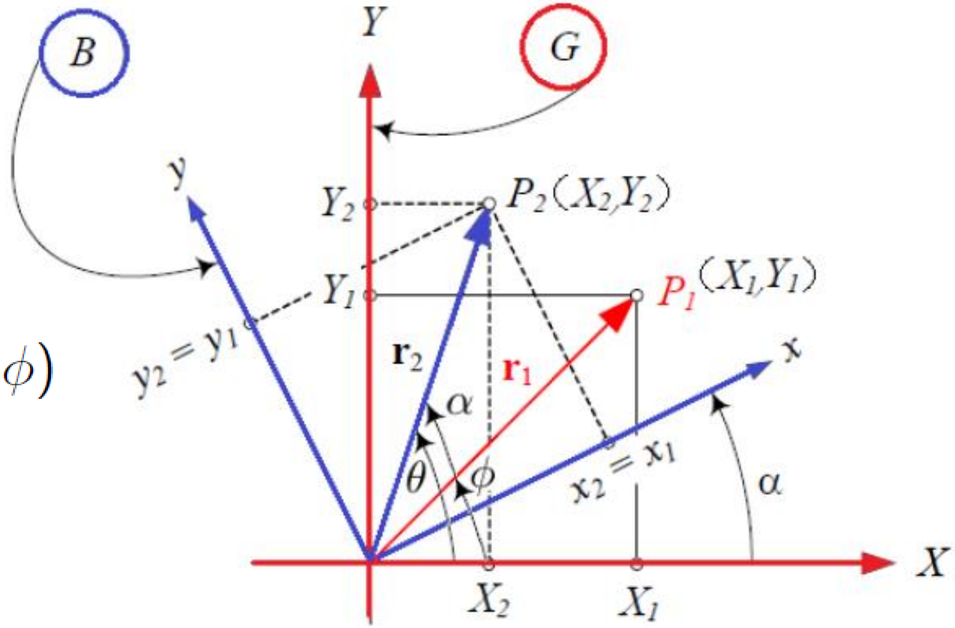
Rotation around axis

Rotation about x-axis

$$Y_1 = r_1 \sin \emptyset$$
$$Y_2 = r_1 \sin \theta \Leftrightarrow Y_2 = r_1 \sin(\alpha + \phi)$$

$$X_1 = r_1 \cos \emptyset$$
$$X_2 = r_1 \cos \theta \Leftrightarrow X_2 = r_1 \cos(\alpha + \phi)$$

$$X_2 = r_1 cos(\alpha)cos(\phi) - r_1 sin(\alpha)sin(\phi)$$



$$X_2 = cos(\alpha)X_1 - sin(\alpha)Y_1$$
$$Y_2 = sin(\alpha)X_1 + cos(\alpha)Y_1$$

$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

Recall

$$cos(\alpha + \phi) = cos(\alpha)cos(\phi) - sin(\alpha)sin(\phi)$$
$$sin(\alpha + \phi) = sin(\alpha)cos(\phi) + cos(\alpha)sin(\phi)$$

Rotation about x-axis (General form)

- As same as x-axis

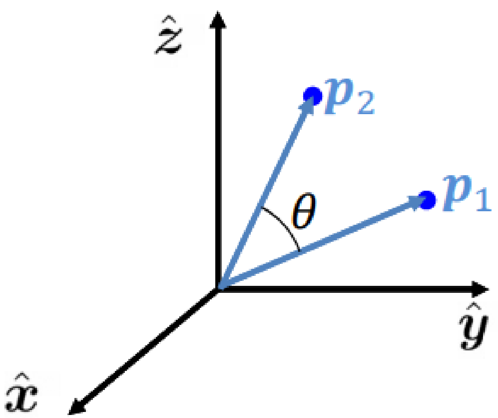
$$\begin{bmatrix} X_2 \\ Y_2 \end{bmatrix} = \begin{bmatrix} cos(\alpha) & -sin(\alpha) \\ sin(\alpha) & cos(\alpha) \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \end{bmatrix}$$

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} p_n \\ p_o \\ p_a \end{bmatrix}$$

$$Rot(x,\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\theta & -S\theta \\ 0 & S\theta & C\theta \end{bmatrix}$$

- Example:

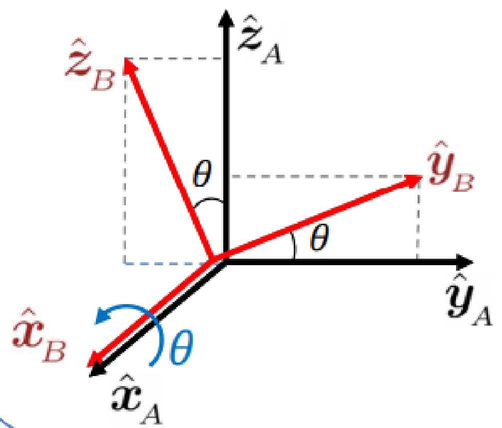
Find the resulting vector after rotating vector $\boldsymbol{p}_1 = (0, \sqrt{3}, 1)$ an angle 30° about axis $\boldsymbol{x}$



$$R(30°, \mathbf{x}) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(30°) & -\sin(30°) \\ 0 & \sin(30°) & \cos(30°) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \sqrt{3}/2 & -1/2 \\ 0 & 1/2 & \sqrt{3}/2 \end{bmatrix}$$

$$\mathbf{p}_2 = R(30°, \mathbf{x})\mathbf{p}_1 = \begin{bmatrix} 0 \\ 1 \\ \sqrt{3} \end{bmatrix}$$

Rotation Summary

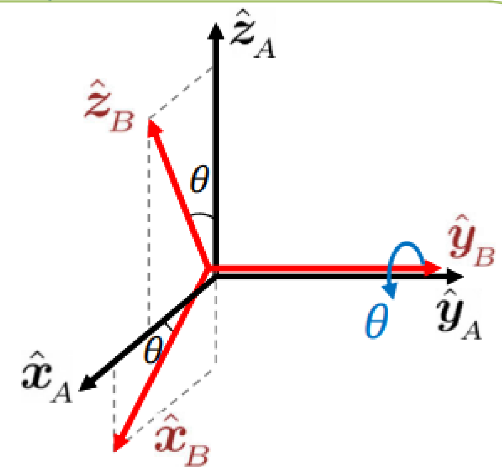


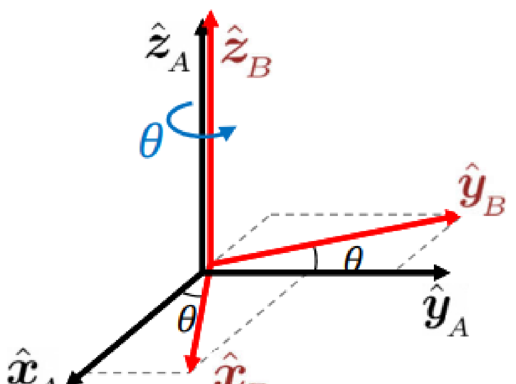
Rotation about the *x* axis

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

Rotation about the *y* axis

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

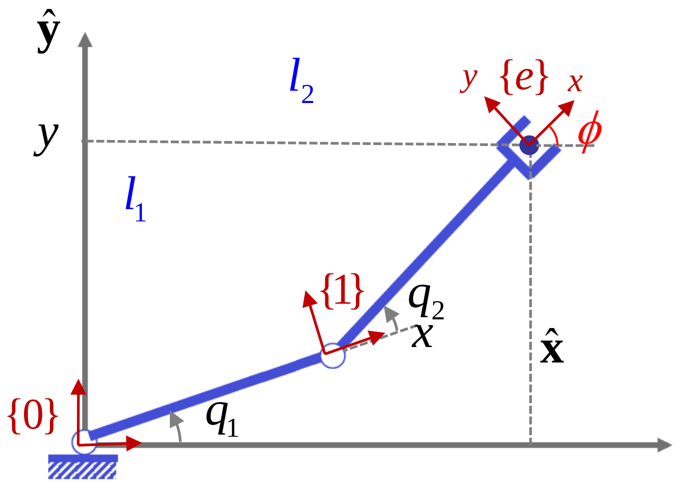




Rotation about the *z* axis

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

2D Example: R-R Robot



Procedure:

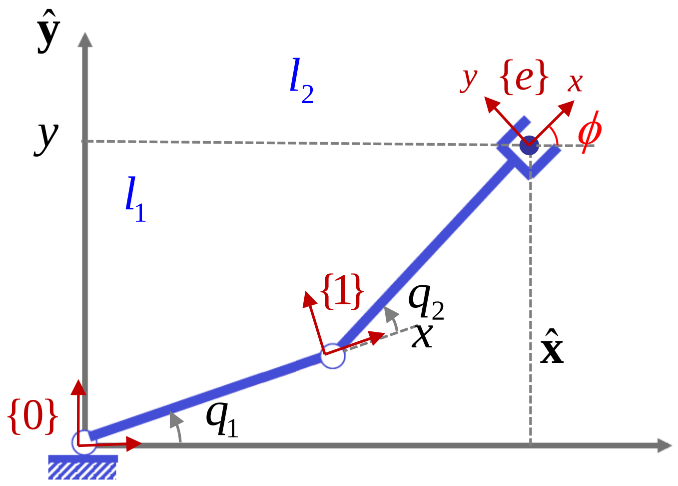
1. Assign frames (that move with each link)
2. Relate a frame to the previous one ($^0T_1$ and ^1T_e)

- $^0T_1$: system {1} with respect to {0} = "take" {0} to {1}
  - Rotate q_1 about z: $Rot_z(q_1)$
  - Translate l_1 along the resulting x (current frame): $Tr_x(l_1)$

$$^0T_1 = Rot_z(q_1)Tr_x(l_1)$$

$$^0T_1 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & l_1\cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & l_1\sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2D Example: R-R Robot



Procedure:

1. Assign frames (that move with each link)
2. Relate a frame to the previous one ($^0T_1$ and ^1T_e)

- ^1T_e: system $\{e\}$ with respect to $\{1\}$ = "take" $\{1\}$ to $\{e\}$
  - Rotate q_2 about z: $Rot_z(q_2)$
  - Translate l_2 along the resulting x (current frame): $Tr_x(l_2)$

$$^1T_e = Rot_z(q_2)Tr_x(l_2)$$

$$^1T_e = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0 \\ \sin(q_2) & \cos(q_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_2\cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & l_2\sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2D Example: R-R Robot

Procedure:

3. Multiply to obtain the final transformation matrix ${}^{0}T_{e}$

$$ {}^{0}T_{e} = \left({}^{0}T_{1}\right)\left({}^{1}T_{e}\right) $$

$$ {}^{0}T_{e} = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & l_1\cos(q_1) \\ \sin(q_1) & \cos(q_1) & 0 & l_1\sin(q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & l_2\cos(q_2) \\ \sin(q_2) & \cos(q_2) & 0 & l_2\sin(q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{0}T_{e} = \begin{bmatrix} \cos(q_1+q_2) & -\sin(q_1+q_2) & 0 & l_1\cos(q_1)+l_2\cos(q_1+q_2) \\ \sin(q_1+q_2) & \cos(q_1+q_2) & 0 & l_1\sin(q_1)+l_2\sin(q_1+q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Position and orientation of the end effector with respect to the base frame

Cobra robot

https://www.youtube.com/watch?v=IRDJnwFDq88

Example Combine rotation

$$\text{End effector} = \begin{bmatrix} 5 \\ 30 \\ 10 \end{bmatrix} \rightarrow 30^\circ \text{ @ Z-axis} \rightarrow 30^\circ \text{ @ X-axis} \rightarrow 90^\circ \text{ @ Y-axis.}$$

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \end{bmatrix} = \begin{bmatrix} cos(30) & -sin(30) & 0 \\ sin(30) & cos(30) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 30 \\ 10 \end{bmatrix} = \begin{bmatrix} -10.68 \\ 28.48 \\ 10.0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(30) & -sin(30) \\ 0 & sin(30) & cos(30) \end{bmatrix} \begin{bmatrix} -10.68 \\ 28.48 \\ 10.0 \end{bmatrix} = \begin{bmatrix} -10.68 \\ 19.66 \\ 22.9 \end{bmatrix}$$

$$= \begin{bmatrix} cos(90) & 0 & sin(90) \\ 0 & 1 & 0 \\ -sin(90) & 0 & cos(90) \end{bmatrix} \begin{bmatrix} -10.68 \\ 19.66 \\ 22.9 \end{bmatrix} = \begin{bmatrix} 22.90 \\ 19.66 \\ 10.68 \end{bmatrix}$$

Example

- A point $p[7, 3, 1]^T$ is attached to a frame F_{noa} and is subjected to the following
- transformations:
- 1) Rotation of 90° about the z-axis
- 2) Followed by a rotation of 90° about the y-axis
- 3) Followed by a translation of [4, −3, 7]

$$Rot(z,\theta) = \begin{bmatrix} C\theta & -S\theta & 0 \\ S\theta & C\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Rot(y,\theta) = \begin{bmatrix} C\theta & 0 & S\theta \\ 0 & 1 & 0 \\ -S\theta & 0 & C\theta \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- A point $p[7, 3, 1]^T$ is attached to a frame F_{noa} and is subjected to the following
- transformations:
- 1) Rotation of 90° about the z-axis
- 2) Followed by a rotation of 90° about the y-axis
- 3) Followed by a translation of [4, −3, 7]

$$\text{translation of } [4, -3, 7] = \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Example

- A point $p[7, 3, 1]^T$ is attached to a frame F_{noa} and is subjected to the following transformations:
- 1) Rotation of 90° about the z-axis
- 2) Followed by a rotation of 90° about the y-axis
- 3) Followed by a translation of [4, −3, 7] x,y,z axes

Pre-multiplied

$$p_{xyz} = Trans(4, -3, 7) Rot(y, 90) Rot(z, 90) p_{noa} =$$

$$\begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ 10 \\ 1 \end{bmatrix}$$

Transformations Relative to the Current (Moving) Frame

Moving frame

- A point p[7, 3, 1]T is attached to a frame Fnoa and is subjected to the following transformation relative to the current moving frame as:
  1) A rotation of 90° about the a-axis
  2) translation of 4, −3,7 along n-, o-, a-axes
  3) Followed by a rotation of 90° about the o-axis

post-multiplied

$$p_{xyz} = Rot(a,90)Trans(4,-3,7)Rot(o,90)p_{noa} =$$

$$p_{xyz} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & 7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 7 \\ 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 5 \\ 0 \\ 1 \end{bmatrix}$$

Combined about frame

- A frame B was rotated about the x-axis 90° followed by a translation about the current o-axis of 2 inches, followed by a rotation about the a-axis of 90° and a translation along the current y-axis of 3 inches.
- a) Write an equation that describes the motions.
- b) Find the final location of a point Bp = 1,3,2 T relative to the reference frame.

Pre-multiplied (reference) | post-multiplied (moving)

$${}^{U}T_B = Trans(0,3,0)Rot(x,90)[B]Trans(0,2,0)Rot(a,90)$$

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

Rule of thumb

- Transformation around fixed frame (pre-multiplication)
- Transformation around moving frame (post-multiplication)

THE END

EXTRA EXAMPLES

Forward Kinematics: Geometric Method

3D Example: SCARA Robot

Adept's SCARA Robot

Schematic model of a SCARA robot

3D Example: SCARA Robot

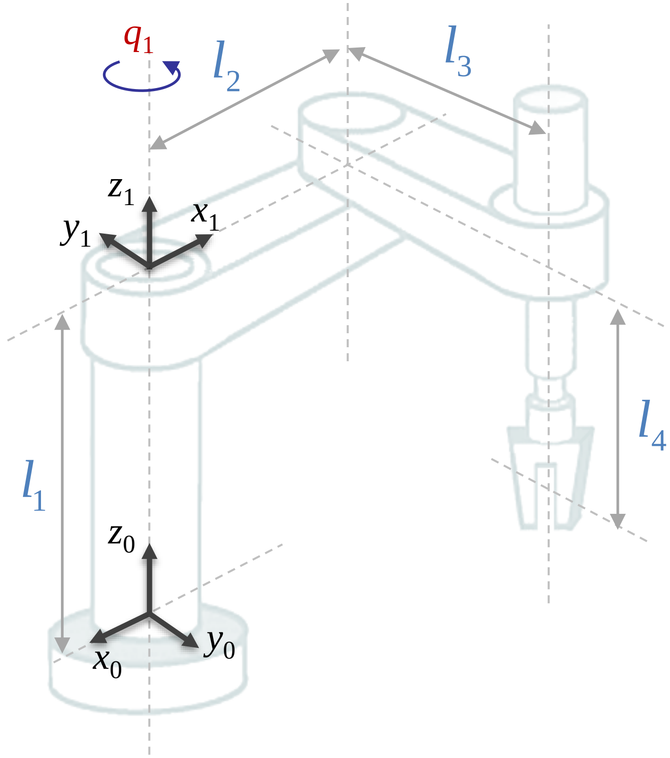
1. Take {0} to {1}

- Translate {0} a distance l_1 along $\boldsymbol{z}_0$.
- Rotate 180°+q_1 about $\boldsymbol{z}_0$ to get to {1}

$$ {}^0T_1 = \underbrace{Tr_z(l_1)}_{} \; \underbrace{Rot_z(180^\circ + q_1)}_{} $$

$$ \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \begin{bmatrix} -c_1 & s_1 & 0 & 0 \\ -s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^0T_1 = \begin{bmatrix} -c_1 & s_1 & 0 & 0 \\ -s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



Arbitrarily assign frames

Since translation and rotation are with respect to the same axis, we have: $Tr_z(l_1)Rot_z(q_1) = Rot_z(q_1)Tr_z(l_1)$

3D Example: SCARA Robot

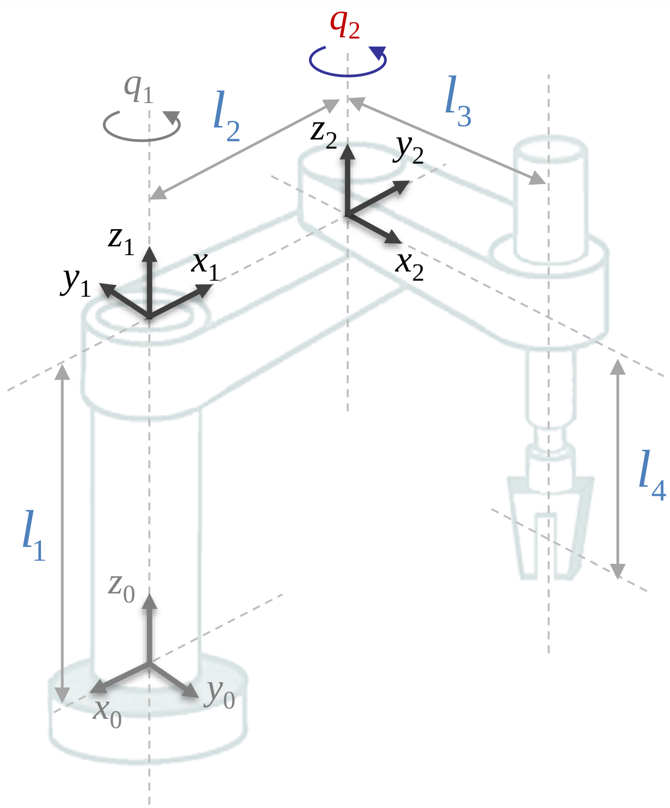
2. Take {1} to {2}

- Translate {1} a distance l_2 along $\boldsymbol{x}_1$
- Then rotate (-90°+q_2) about the new $\boldsymbol{z}$ to get to {2}

$$^{1}T_2 = Tr_x(l_2)Rot_z(-90^\circ + q_2)$$

$$Tr_x(l_2) = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad Rot_z(-90^\circ + q_2) = \begin{bmatrix} s_2 & c_2 & 0 & 0 \\ -c_2 & s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{1}T_2 = \begin{bmatrix} s_2 & c_2 & 0 & l_2 \\ -c_2 & s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Arbitrarily assign frames

In this case (as in general) the product is not commutative

3D Example: SCARA Robot

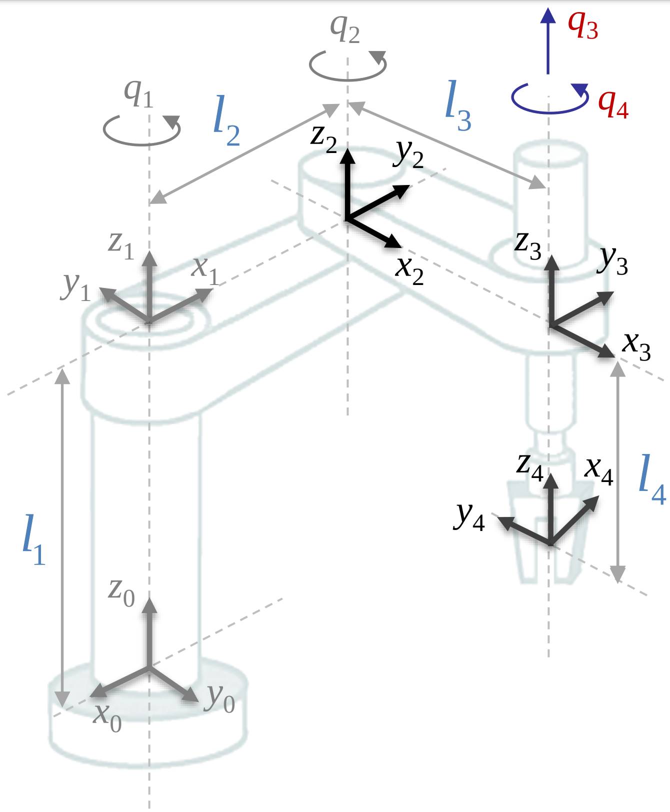
3. Take {2} to {3}

Translate {2} a distance l_3 along $\boldsymbol{x}_2$ to get to {3}

$$^{2}T_3 = Tr_x(l_3) = \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. Take {3} to {4}

- Translate {3} a distance $(-l_4+q_3)$ along $\boldsymbol{z}_3$
- Then, rotate $(90°+q_4)$ about $\boldsymbol{z}$ to get to {4}



$$^{3}T_4 = Tr_z(-l_4+q_3)Rot_z(90°+q_4) = \begin{bmatrix} -s_4 & -c_4 & 0 & 0 \\ c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & q_3-l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Arbitrarily assign frames

3D Example: SCARA Robot

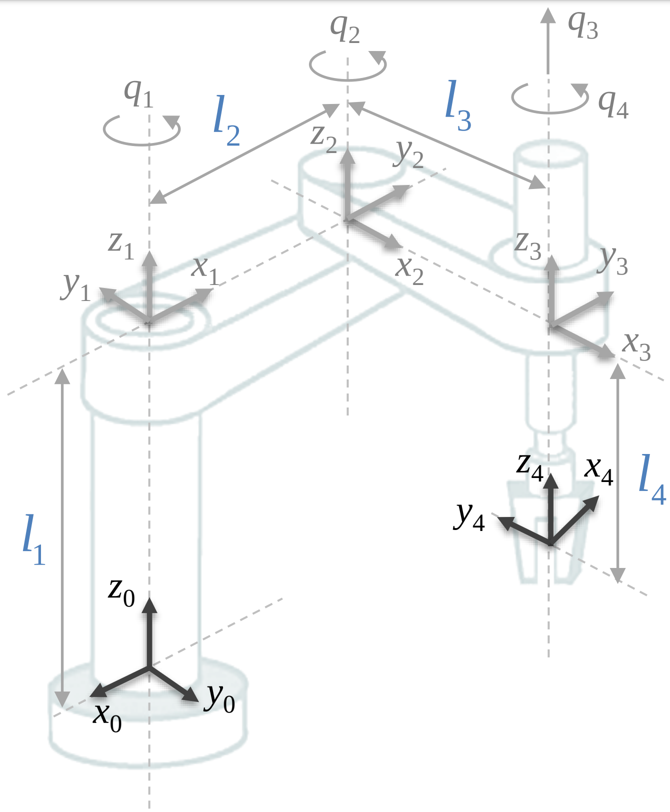
5. Multiply: take {0} to {4}

Write the end effector {4} in terms of the base {0} $\longrightarrow$ multiply the kinematic chain

$$^{0}T_4 = \left(^{0}T_1\right)\left(^{1}T_2\right)\left(^{2}T_3\right)\left(^{3}T_4\right)$$

$$^{0}T_4 = \begin{bmatrix} -c_1 & s_1 & 0 & 0 \\ -s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_2 & c_2 & 0 & l_2 \\ -c_2 & s_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -s_4 & -c_4 & 0 & 0 \\ c_4 & -s_4 & 0 & 0 \\ 0 & 0 & 1 & q_3 - l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{0}T_4 = \begin{bmatrix} -c_{124} & s_{124} & 0 & -l_3 s_{12} - l_2 c_1 \\ -s_{124} & -c_{124} & 0 & l_3 c_{12} - l_2 s_1 \\ 0 & 0 & 1 & l_1 - l_4 + q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Forward kinematics using arbitrary reference frames

3D Example: SCARA Robot

Optional:

The end effector convention can be used (axis $\boldsymbol{n} = x$, $\boldsymbol{o} = y$, $\boldsymbol{a} = z$) for frame {e}

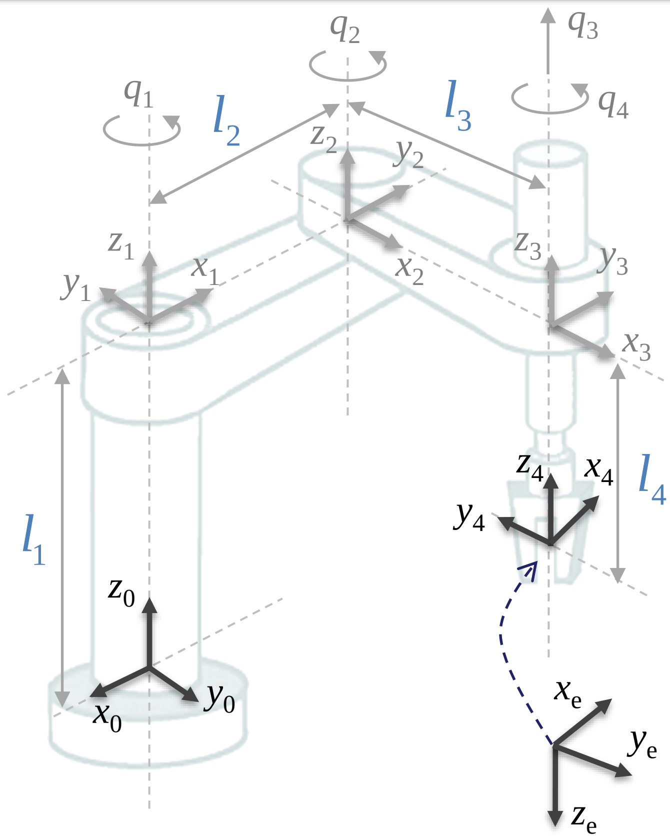
- Rotate {4} 180° about $\boldsymbol{x}_4$

$$
{}^4T_e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(180°) & -\sin(180°) & 0 \\ 0 & \sin(180°) & \cos(180°) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

- Forward kinematics:

$$
{}^0T_e = \left({}^0T_4\right)\left({}^4T_e\right)
$$

$$
{}^0T_e = \begin{bmatrix} -c_{124} & -s_{124} & 0 & -l_3 s_{12} - l_2 c_1 \\ -s_{124} & c_{124} & 0 & l_3 c_{12} - l_2 s_1 \\ 0 & 0 & -1 & l_1 - l_4 + q_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$



Verify the result when the joint configuration is null (zeros)

3D Example: SCARA Robot

Homogeneous Transformation Matrices using Python

Import sympy matrices

```
from sympy.matrices import Matrix
```

Function for translation

```
def transl(x, y, z):
    T = Matrix([[1,0,0,x],
                [0,1,0,y],
                [0,0,1,z],
                [0,0,0,1]])
    return T
```

Function for rotation about X

```
def trotx(ang):
    T = Matrix([[1,0,0,0],
                [0,cos(ang),-sin(ang),0],
                [0,sin(ang),cos(ang),0],
                [0,0,0,1]])
    return T
```

Function for rotation about Z

```
def trotz(ang):
    T = Matrix([[cos(ang),-sin(ang),0,0],
                [sin(ang), cos(ang),0,0],
                [0,0,1,0],
                [0,0,0,1]])
    return T
```

3D Example: SCARA Robot

Homogeneous Transformation Matrices using Python

Import sympy

```
from sympy import *
```

Homogeneous transformations and their products

```
q1, q2, q3, q4 = symbols("q1 q2 q3 q4")
l1, l2, l3, l4 = symbols("l1 l2 l3 l4")

T1 = transl(0,0,l1)*trotz(pi+q1)
T2 = transl(l2,0,0)*trotz(-pi/2+q2)
T3 = transl(l3,0,0)
T4 = transl(0,0,-l4+q3)*trotz(pi/2+q4)
T04 = simplify(T1*T2*T3*T4)
print(T04)
```

Using the convention for the end effector

```
Te = trotx(pi);
T0e = simplify(T04*Te);
print(T0e)
```

When all joint values are zero

```
T0e.subs([(q1,0),(q2,0),(q3,0),(q4,0)])
```