Lecture 3 - Revision

https://kahoot.it/

✓ Pose

$^{A}\boldsymbol{\xi}_{B}$

- Transformation Operator
- $^{A}\boldsymbol{p} = {}^{A}R_{B}\, {}^{B}\boldsymbol{p} + {}^{A}\boldsymbol{p}_{Bo}$

✓ Rotation

- Euler angle, Roll-Pitch-Yaw
- Rotation about current axes and fixed axes
- rotx(),roty(),rotz() in RVC Toolbox

✓ Homogeneous Transformations

$$^{A}T_{B} = \begin{bmatrix} ^{A}R_{B} & ^{A}\boldsymbol{p}_{Bo} \\ 0\ 0\ 0 & 1 \end{bmatrix}$$

$$SE(3) = \mathbb{R}^{3} \times SO(3)$$

MTRN4230
Robotics

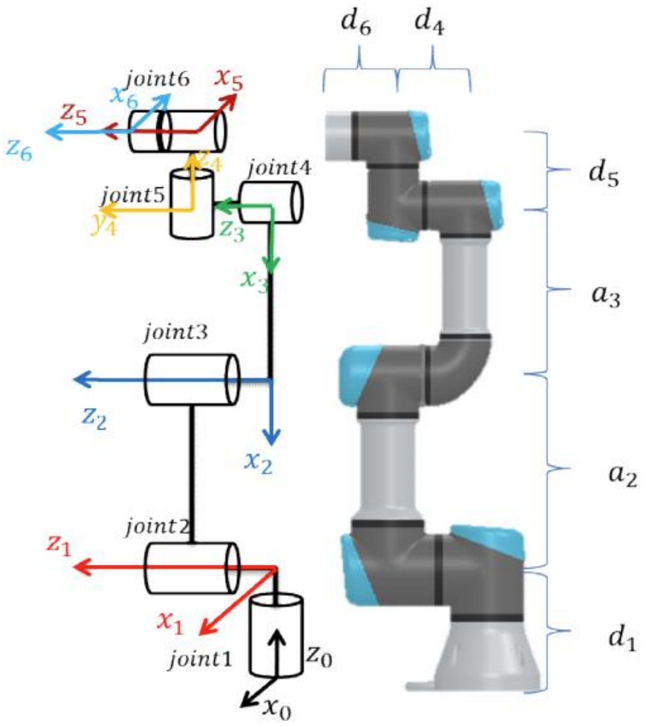
Lecture 4
Denavit-Hartenberg Convention

Hoang-Phuong **Phan** – T2 2023

UNSW SYDNEY

Learning objectives

- ❑ Understand the DH Convention
- ❑ Define the DH parameters
- ❑ Solve the forward kinematics of serial manipulators

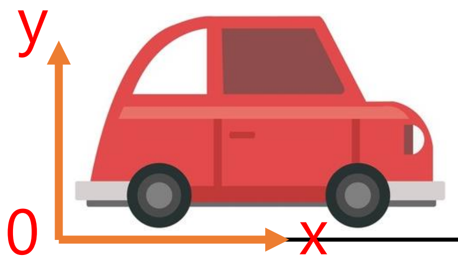


UR5e				
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]
Joint 1	0	0	0.1625	π/2
Joint 2	0	-0.425	0	0
Joint 3	0	-0.3922	0	0
Joint 4	0	0	0.1333	π/2
Joint 5	0	0	0.0997	-π/2
Joint 6	0	0	0.0996	0

Why D-H Convention?

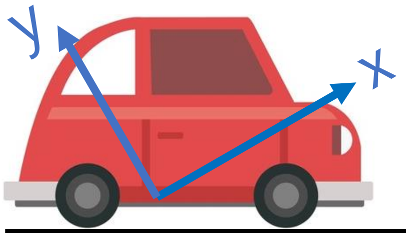
- In many kinematic problems, by properly defining coordinate frames, we can significantly simplify the solutions

Constant acceleration



$$s_x = v_o t + \frac{at^2}{2}$$

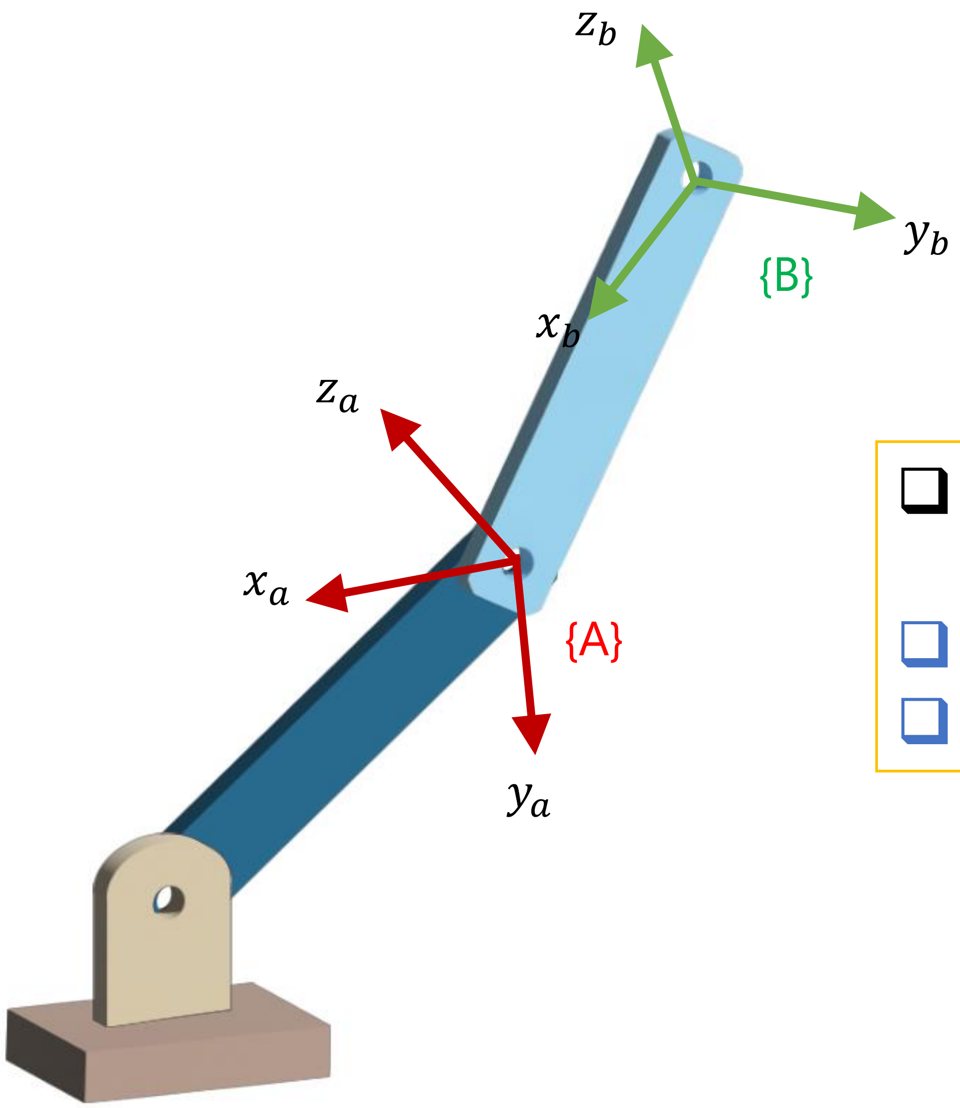
$$s_y = 0$$



$$s_x = ?$$

$$s_y = ?$$

Why D-H Convention?



- Homogeneous Transformation

$$\begin{bmatrix} {}^{A}\boldsymbol{p} \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} {}^{A}R_{B} & {}^{A}\boldsymbol{p}_{Bo} \\ 0\ 0\ 0 & 1 \end{bmatrix}}_{{}^{A}T_{B}} \begin{bmatrix} {}^{B}\boldsymbol{p} \\ 1 \end{bmatrix}$$

- If we choose the pose of {A} and {B} randomly, there will be Six variables required to express matrix ${}^{A}T_{B}$
- Three for ${}^{A}R_{B}$
- Three for the translation vector ${}^{A}\boldsymbol{p}_{Bo}$

Complicated calculation!

Use D-H Convention

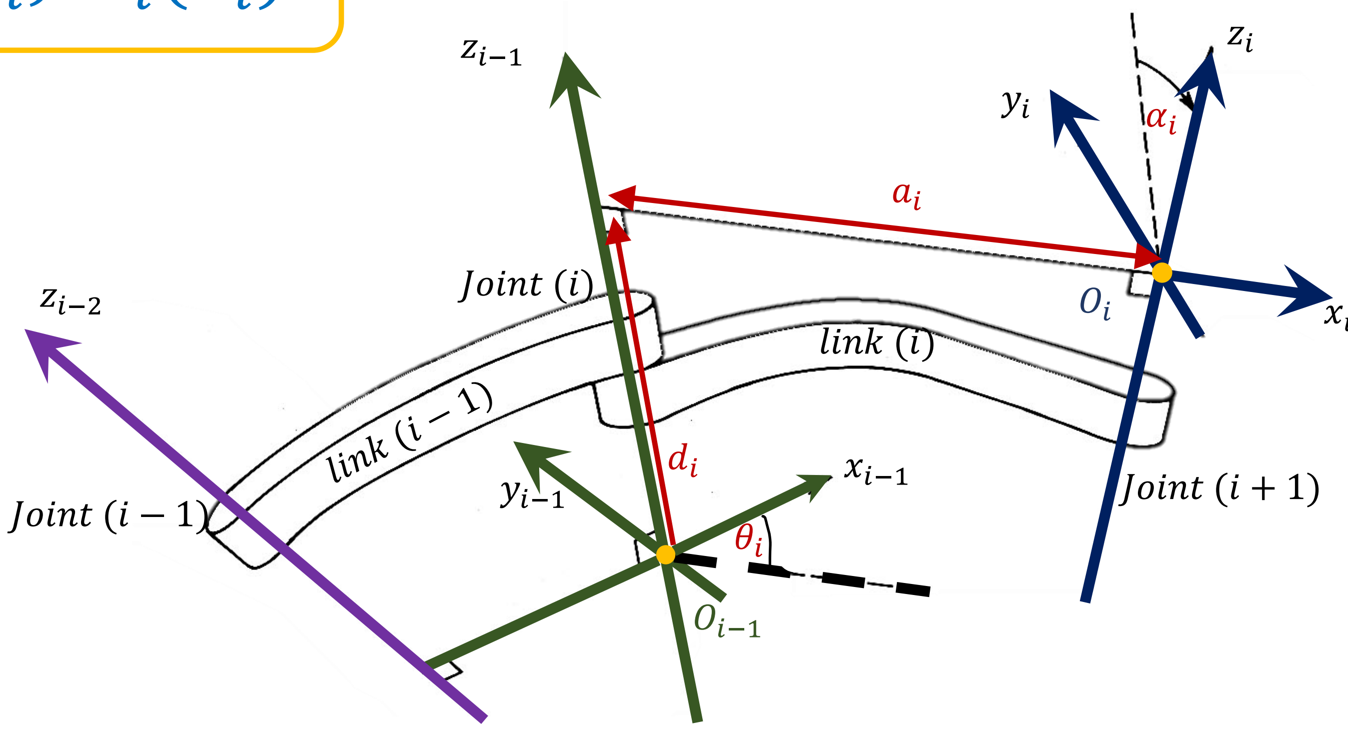
D-H Convention

- A common convention used to attach coordinate frames to the links of a robot manipulator
- Its aim was to standardize coordinate frames for spatial linkages
- Parameters from the DH Convention can be used to transform one coordinate frame to another in a robot arm manipulator

D-H Convention

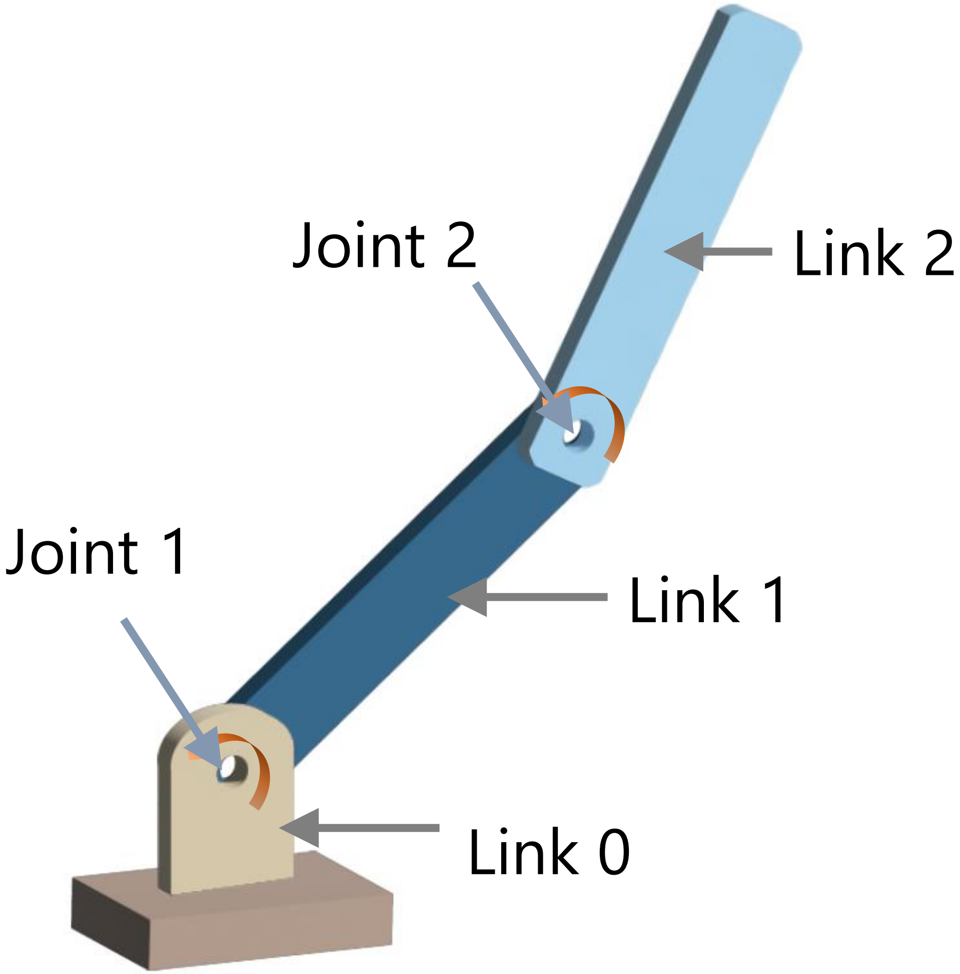
$$ {}^{i-1}T_i = R_{(i-1)}(\theta_i).\,Q_{(i-1)}(d_i).\,Q_i(a_i).\,R_i(\alpha_i) $$

- ❑ First two terms: the previous axis
- ❑ Last two terms: the new axis



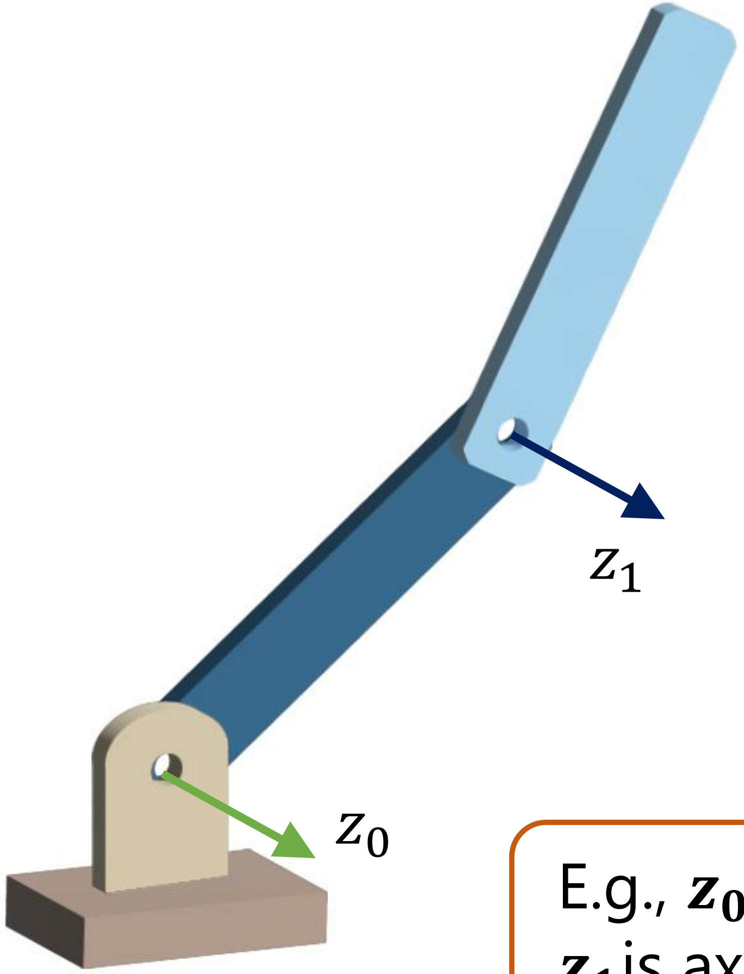
Source: Spong M., Hutchinson S. and Vidyasagar M., *Robot Modeling and Control*, 2006, John Wiley & Sons

Transformation Rules for Robot Arms



- A serial robot with $\boldsymbol{n}$ **joints** will has $\boldsymbol{n} + \mathbf{1}$ **links** since each joint connects 2 links
- Number the **link** from **0** to **n**, starting from the base (link 0)
- Number the joint from 1 to n
- Joint variable q_i:
  - If revolute, $\boldsymbol{q_i} = \boldsymbol{\theta_i}$ (angle of rotation)
  - If prismatic, $\boldsymbol{q_i} = \boldsymbol{d_i}$, (joint displacement)

Step 1 – Setup Z axis for all frames



- Identify all the joint axes, and label them as,

  $$z_0, z_1 \dots , z_{n-1}$$

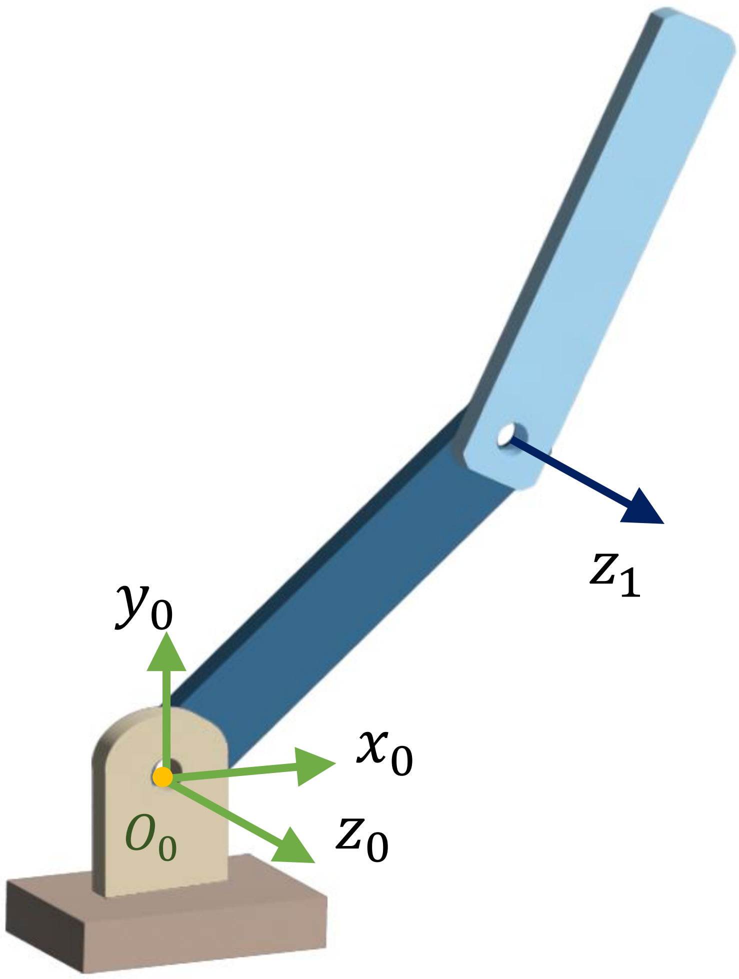
- $\boldsymbol{z_i}$ is the **actuation axis** of **joint $\boldsymbol{i+1}$**

- Rotation axis for revolute joint, or axis of translation for prismatic joint

E.g., $\boldsymbol{z_0}$is axis of actuation of joint 1, $\boldsymbol{z_1}$is axis of actuation for joint 2

Source: Spong M., Hutchinson S. and Vidyasagar M., *Robot Modeling and Control*, 2006, John Wiley & Sons

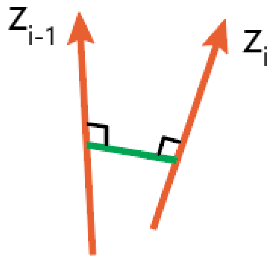
Step 2 – Define Frame Z_0



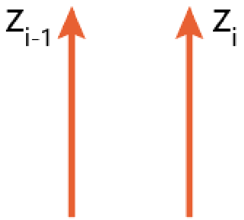
- The origin O_0 of the base frame can be anywhere along the z_0 axis
- Choose x_0 and y_0 to satisfy the right-hand rule

Step 3: Define frame {i} based-on frame {i-1}

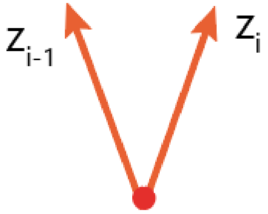
- z_i are defined in step 1 → only need to define x_i, y_i and the origin O_i
- Since y_i can be defined from x_i and z_i using the right-hand frame → only need to define x_i and O_i
- The construction of $\boldsymbol{x_i}$ and $\boldsymbol{O_i}$ depends on the relative position between $\boldsymbol{z_{i-1}}$ and $\boldsymbol{z_i}$



- **Case 1: z_{i-1} and z_i are not coplanar**



- **Case 2: z_{i-1} is parallel to z_i**

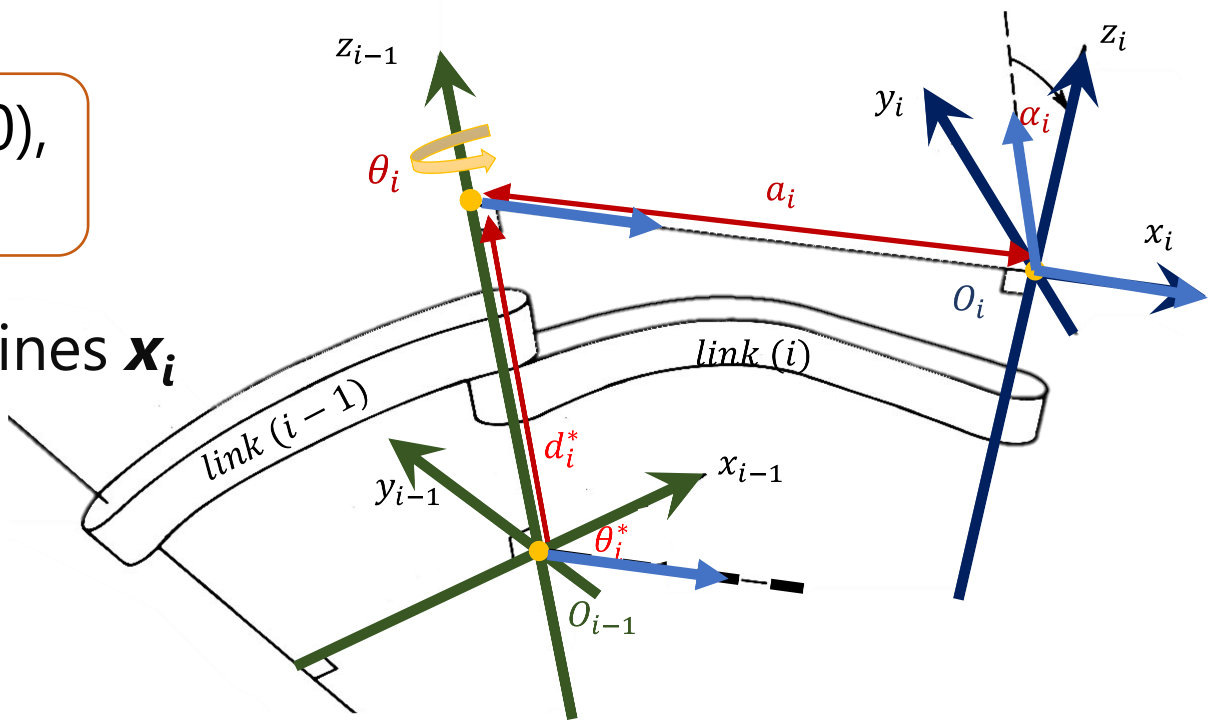


- **Case 3: z_{i-1} and z_i intersect**

Step 3: Define frame {i} based-on frame {i-1}

Case 1: z_{i-1} and z_i are not coplanar

Let consider the home position first (i.e., θ_i =0), then we will add joint variables (θ_i) later



- The common normal between $\boldsymbol{z_{i-1}}$ and $\boldsymbol{z_i}$ defines $\boldsymbol{x_i}$
- The point where $\boldsymbol{x_i}$ **and** $\boldsymbol{z_i}$ **intersect** is the origin $\boldsymbol{O_i}$ of frame {i}
- Transformation from {i-1} to {i} has 4 steps:

Rotate by θ_i^* about z_{i-1}; Translate by d_i along z_{i-1};
Translate by a_i along x_i; Rotate by α_i along x_i

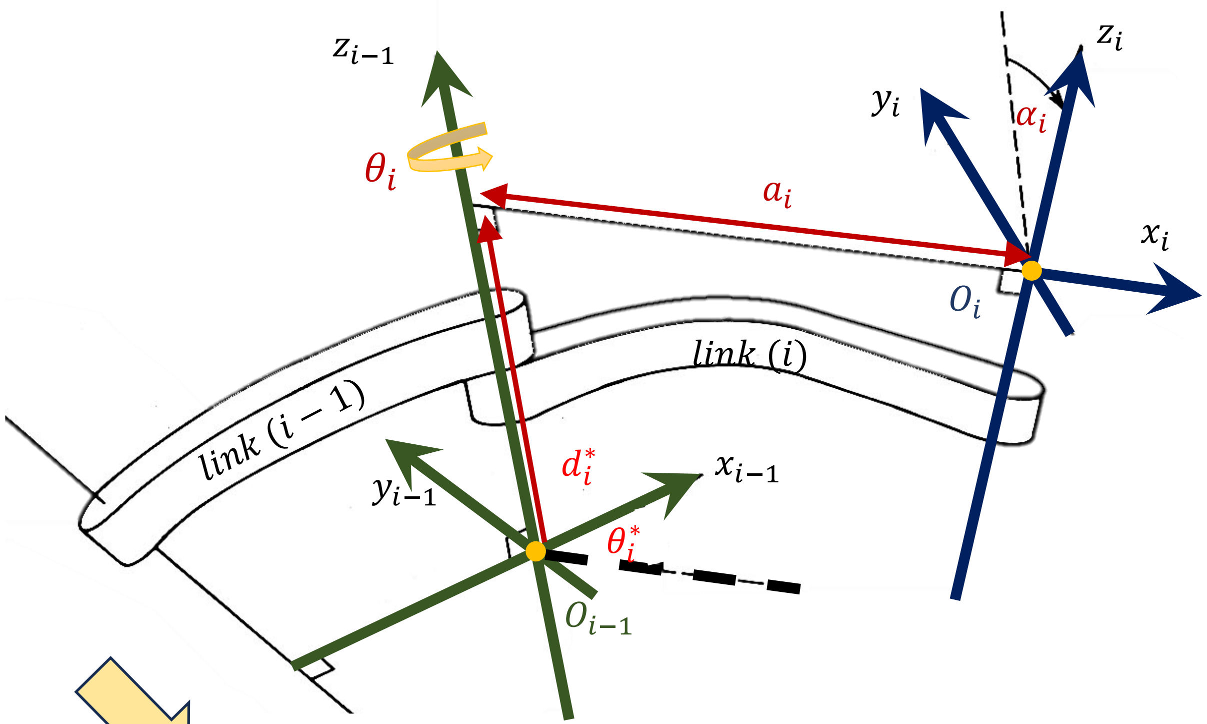
$\boldsymbol{a_i}$, $\boldsymbol{\alpha_i}$, d_i^*, and θ_i^* are generally given the names link length, link twist, link offset, and joint angle.

Step 3: Define frame {i} based-on frame {i-1}

Case 1: z_{i-1} and z_i are not coplanar

Link	Joint angle (degree)	Link offset (cm)	Link length (cm)	Link twist (degree)
i	θ_i^*	d_i^*	a_i	α_i

Robot actuation: Joint (i-1)th rotates an angle of θ_i (for revolute joint) or translate a distance of d_i (for prismatic joint). We need to add this joint variable to the DH table



i-1 is a revolute joint

Link	Joint angle (degree)	Link offset (cm)	Link length (cm)	Link twist (degree)
i	$\theta_i + \theta_i^*$	d_i^*	a_i	α_i

i-1 is a prismatic joint

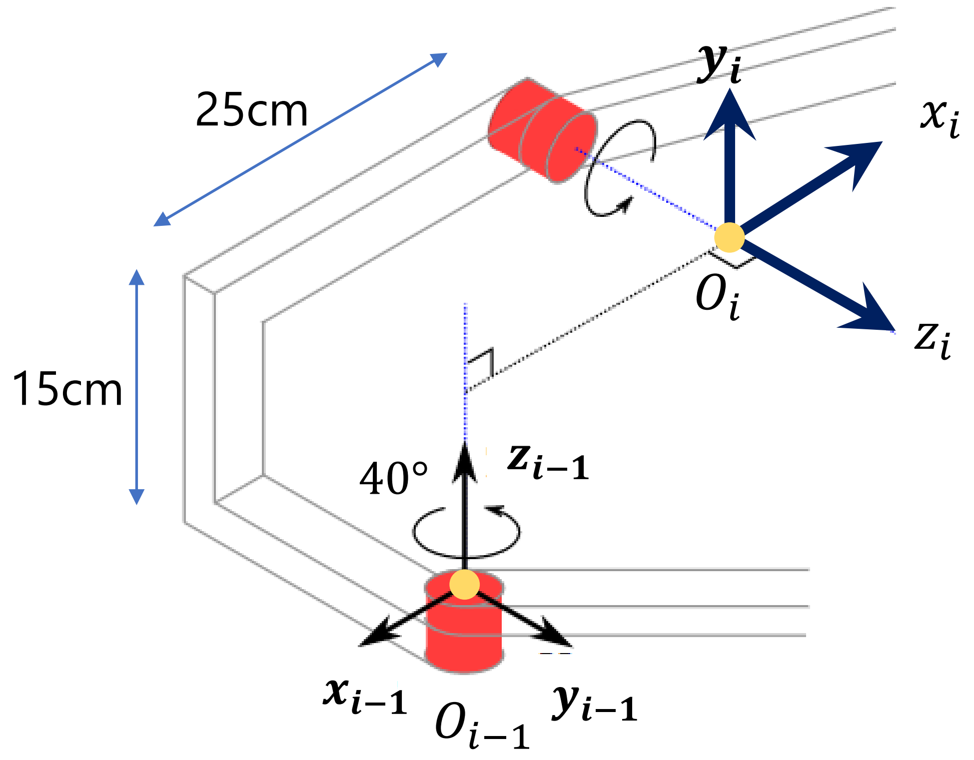
Link	Joint angle (degree)	Link offset (cm)	Link length (cm)	Link twist (degree)
i	θ_i^*	$d_i + d_i^*$	a_i	α_i

Step 3: Define frame {i} based-on frame {i-1}

❑ Case 1: z_{i-1} and z_i are not coplanar

- Example

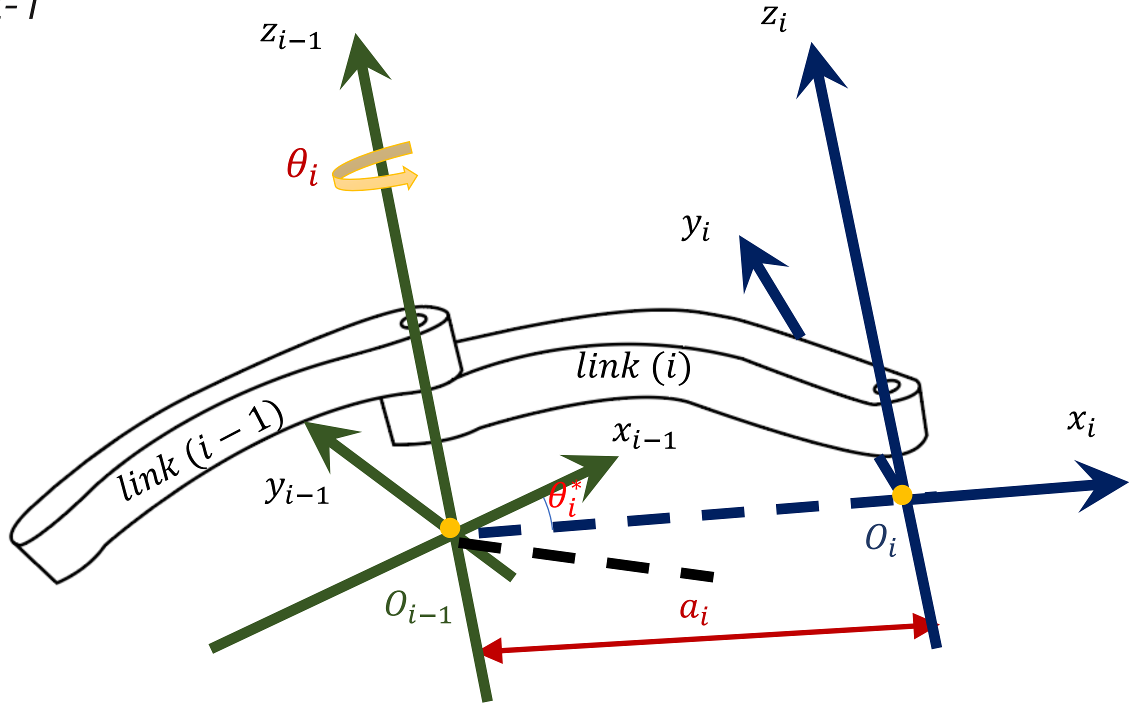
Link	θ_i (degree)	d_i (cm)	a_i (cm)	α_i (degree)
i	40+180	15	25	90



Step 3: Define frame {i} based-on frame {i-1}

❑ Case 2: z_{i-1} is parallel to z_i

- There are infinitive common normals between z_{i-1} and z_i
- A common method to choose $\boldsymbol{x_i}$ is the common normal that **pass through $\boldsymbol{O_{i-1}}$**
- O_i is the intersect between the selected x_i and z_i



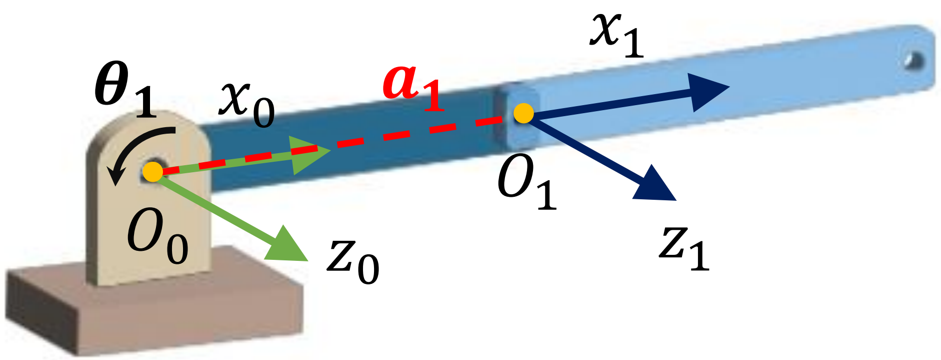
Link	θ_i (degree)	d_i (cm)	a_i (cm)	α_i (degree)
i	$\theta_i^* + \theta_i$	0	a_i	0

$$H = R_{z_{i-1}}(\theta_i^* + \theta_i).\, T_{x_i}(a_i)$$

(or 180 if you choose the opposite direction)

Step 3: Define frame {i} based-on frame {i-1}

- **Case 2: z_{i-1} is parallel to z_i**



- Example

Link	θ_i (degree)	d_i (cm)	a_i (cm)	α_i (degree)
1	θ_1	0	a_1	0

In this example (before the robot move an angle of θ_1), we have $\theta_1^* = 0$

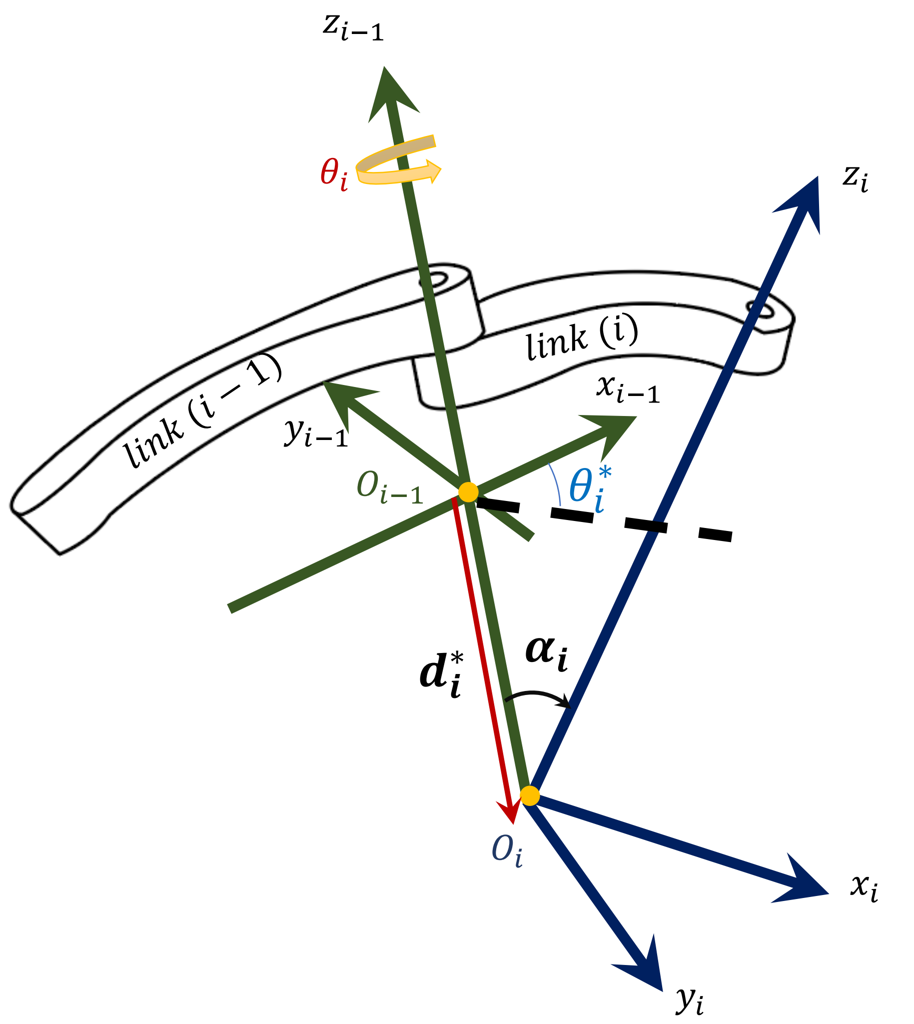
Step 3: Define frame {i} based-on frame {i-1}

❑ Case 3: z_{i-1} intersects z_i

- x_i is chosen normal to the plane formed by z_{i-1} and z_i . The direction of x_i is arbitrary

- Generally, O_i is the intersect between z_{i-1} and z_i

Link	θ_i (degree)	d_i (cm)	a_i (cm)	α_i (degree)
i	$\theta_i^* + \theta_i$	d_i^*	0	α_i

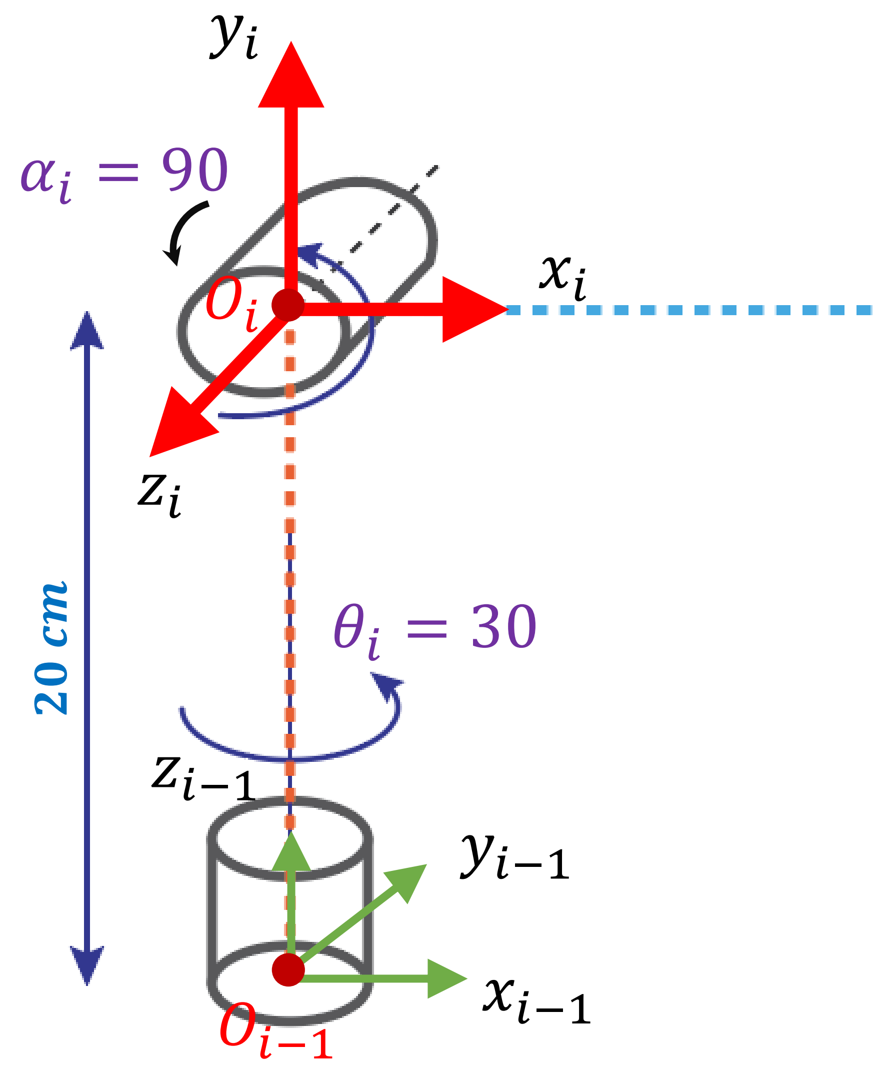


Step 3: Define frame {i} based-on frame {i-1}

- **Case 3: z_{i-1} intersects z_i**

- *Example*

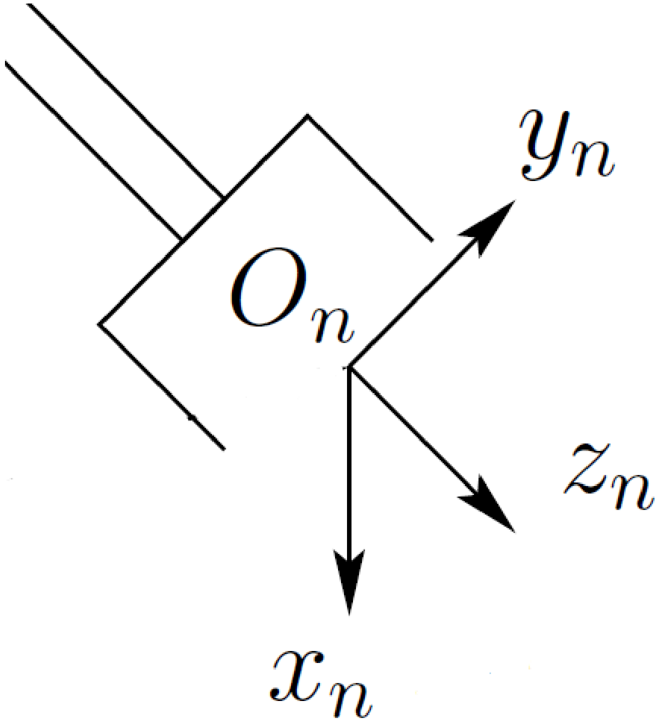
Link	θ_i (degree)	d_i (cm)	a_i (cm)	α_i (degree)
i	30	20	0	90



Step 4 –Construct the end-effector frame

- The final coordinate system {n} is the end-effector or tool frame

  - There are several approaches to define the tool frame (in some cases, can be similar to frame {n-1})
  - In Spong' book, z_n: approach direction (the gripper approaches an object)
  - y_n: sliding direction (open and close robot finger)
  - x_n: right-hand rule



Step 5 Complete the DH table & apply the chain rule

Link	$\boldsymbol{\theta_i}$	$\boldsymbol{d_i}$	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$
0	0	0	0	0
1	$\boldsymbol{\theta_1}$	$\boldsymbol{d_1}$	a_1	α_1
2	$\boldsymbol{\theta_2}$	$\boldsymbol{d_2}$	a_2	α_2
.....	....	....	....	....
n	$\boldsymbol{\theta_n}$	$\boldsymbol{d_n}$	a_n	α_n

$$^{i-1}T_i = \begin{pmatrix} cos\theta_i & -sin\theta_i\, cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- $^{i-1}T_i$ represents the product of four basic transformations.

$$^{i-1}T_i = \underbrace{R_{(i-1)}(\theta_i)}_{\text{angle}}.\underbrace{Q_{(i-1)}(d_i)}_{\text{offset}}.\underbrace{Q_i(a_i)}_{\text{length}}.\underbrace{R_i(\alpha_i)}_{\text{twist}}$$

Four Parameters of DH — angle offset length twist

Note: In many textbooks, for simplification, $\theta_i^* + \boldsymbol{\theta_i}$ is written as $\boldsymbol{\theta_i}$, which includes offset angle inside.

Please remember to check if there is any offset angle at the home position

Step 5 Complete the DH table & apply the chain rule

- Transformation matrix from Frame {0} to Frame {n}:

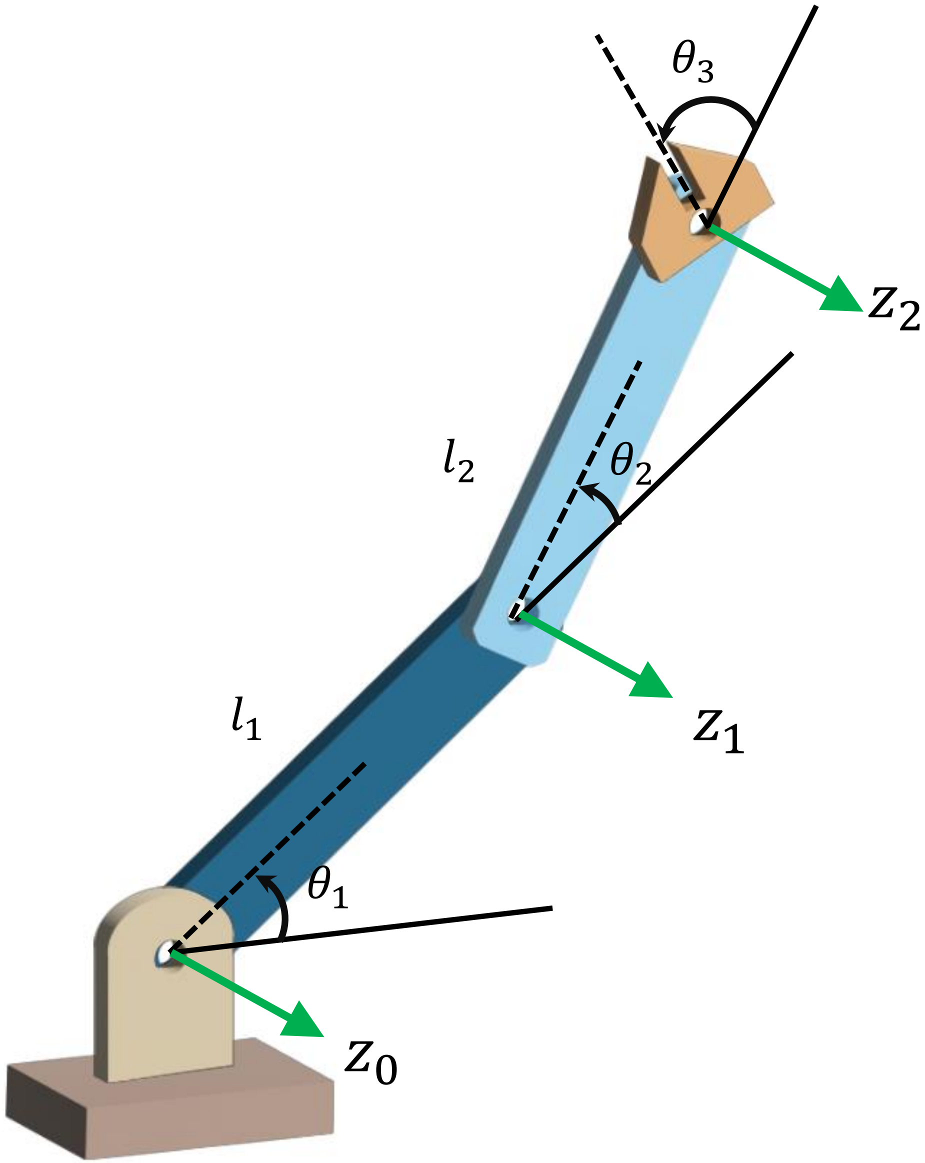
$$ {}^{0}\boldsymbol{T}_{n} = {}^{0}\boldsymbol{T}_{1}\ {}^{1}\boldsymbol{T}_{2} \ldots\ {}^{n-2}\boldsymbol{T}_{n-1}\ {}^{n-1}\boldsymbol{T}_{n} $$

as the position and orientation of the tool frame w.r.t the base frame.

- A vector relative to the tool frame can be expressed in base coordinates as:

$$ {}^{0}\boldsymbol{p} = {}^{0}T_{n}\, {}^{n}\boldsymbol{p} $$

Example 1: Revolute Manipulator

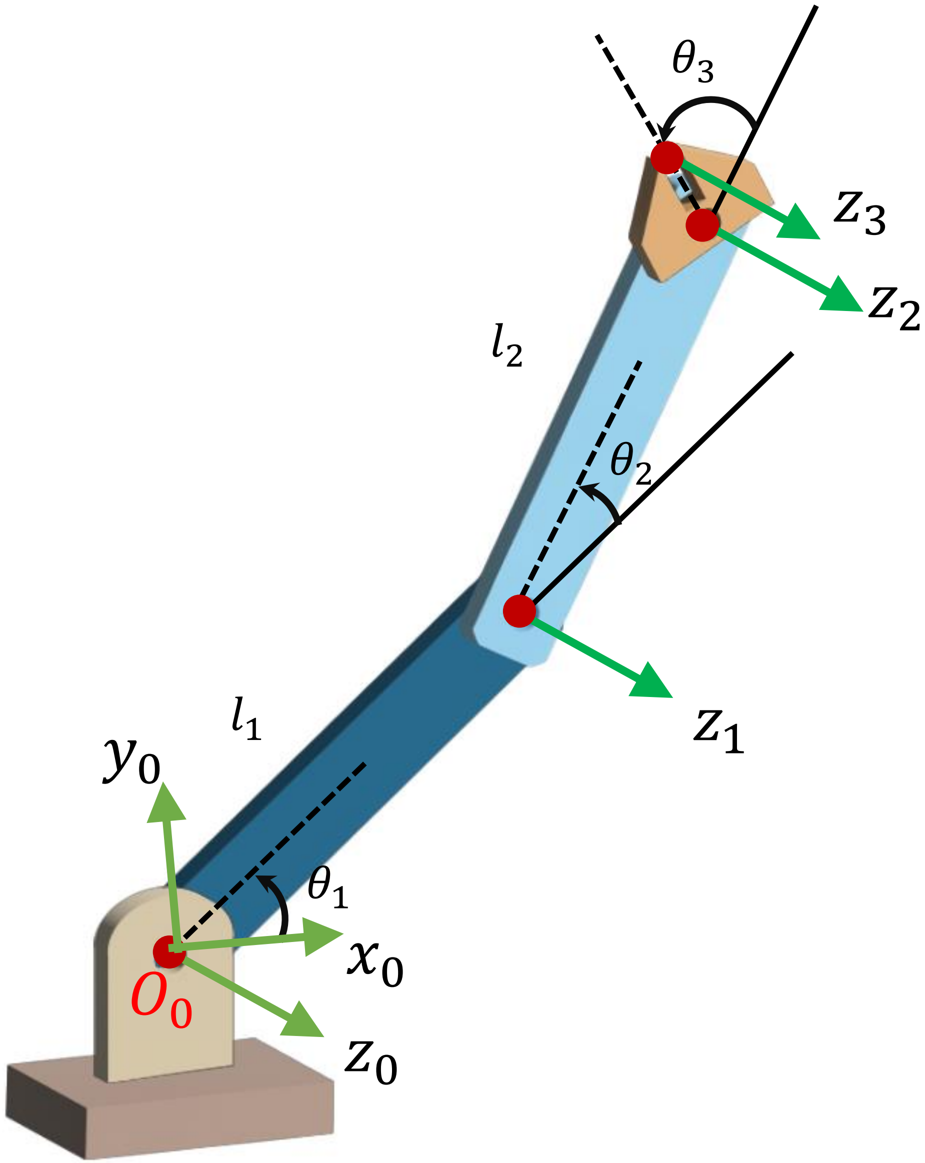


If $l_1 = 3, l_2 = 4, \theta_1 = 25, \theta_2 = 40,\ \theta_3 = 0$, and the end effector tool point is considered to be at (0,0,0) of frame 2 (i.e, $l_3 = 0$).

What is the position of the end effector in frame 0 coordinates?

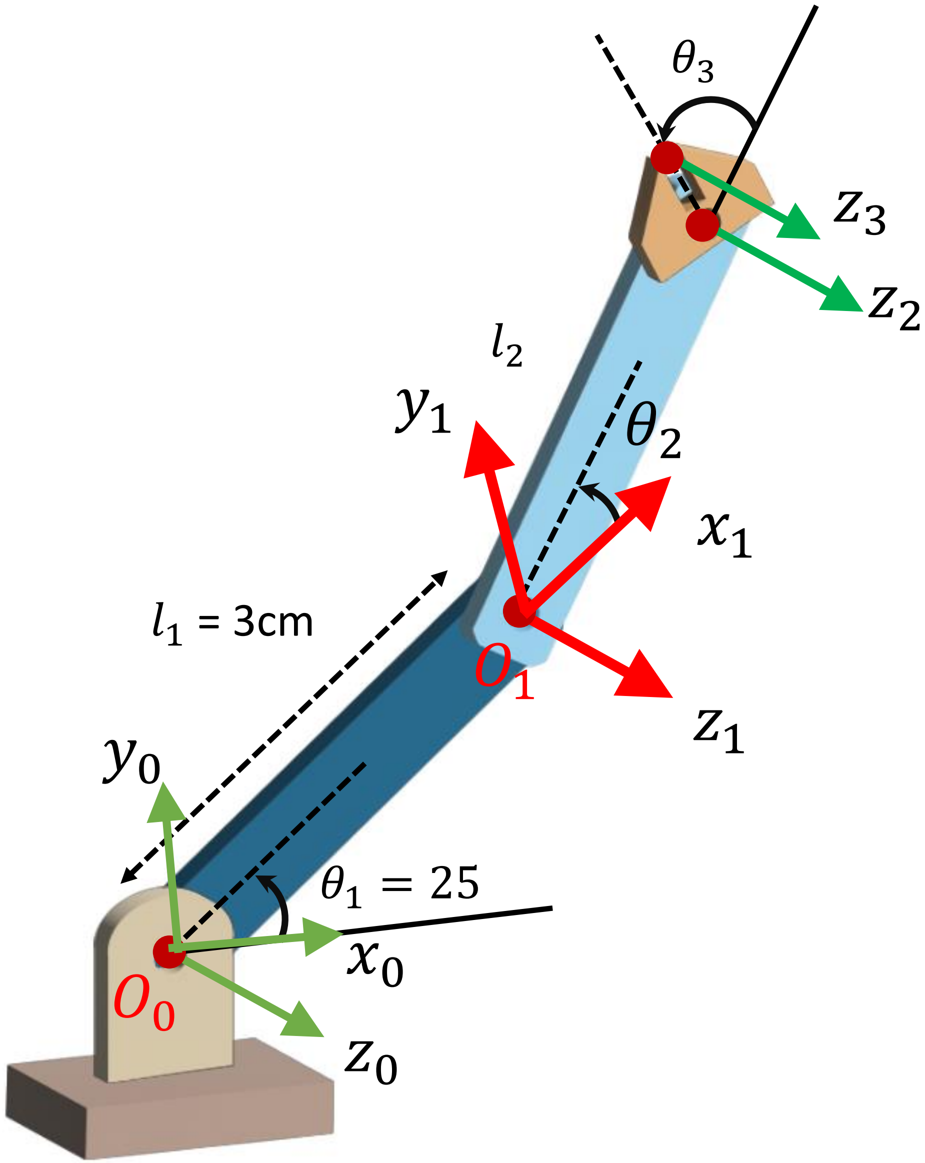
$\boldsymbol{{}^{0}p} = {}^{0}T_{n}{}^{n}\boldsymbol{p}$, where ${}^{n}\boldsymbol{p}$= (0 0 0)

Example 1: Revolute Manipulator



- ❑ Identify all the joint axes, z_0, z_1, z_2 (z_{i-1} is the axis of revolution or translation of joint i)
- ❑ Select the origin O_0 of the base frame
- ❑ x_0 and y_0 to satisfy the right-hand rule

Example 1: Revolute Manipulator

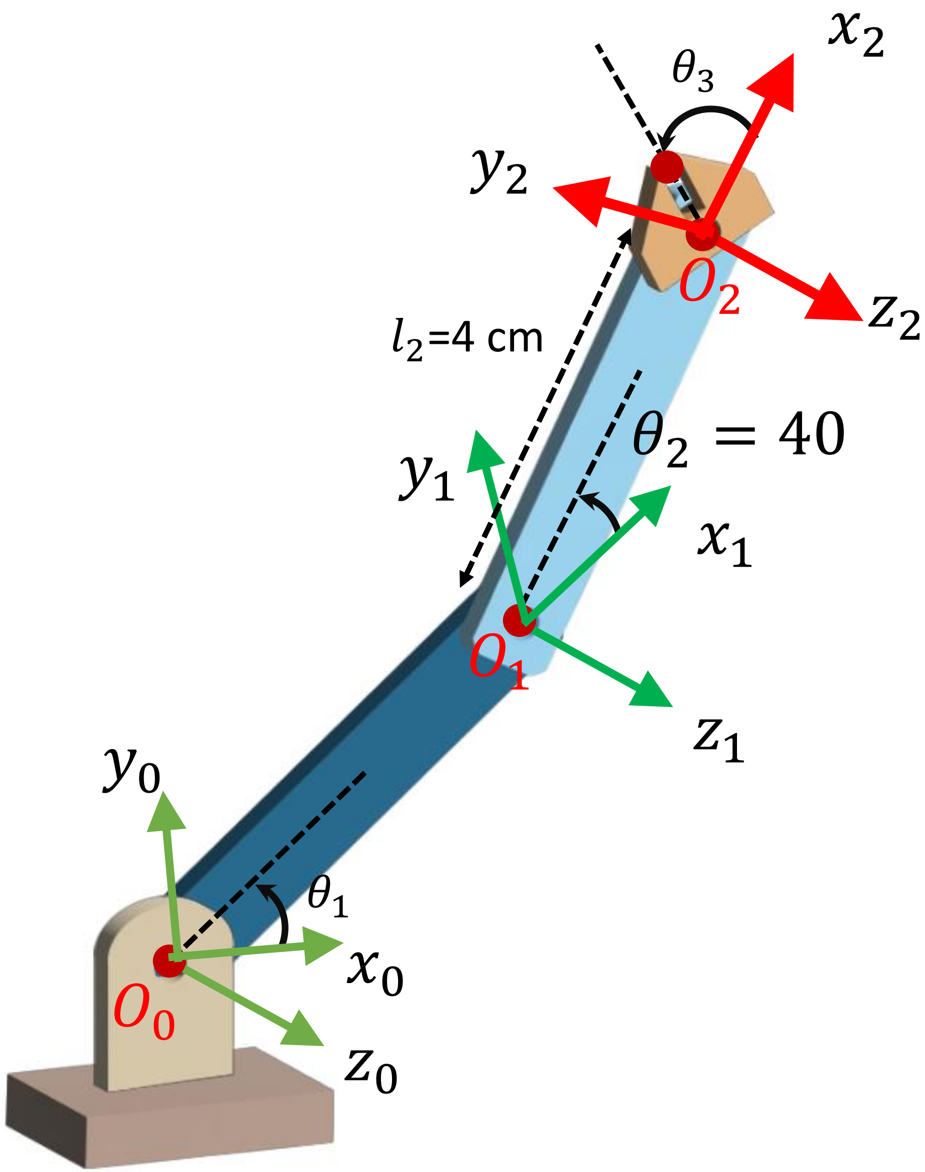


- <u>Frame {1}:</u> As z_1 is parallel to z_0, x_1 is a common normal between z_0 and z_1,

- x_1 in the direction from z_0 to z_1 that passthrough O_0
- O_1 is the intersect between x_1 and z_1

$\boldsymbol{i}$	$\boldsymbol{\theta_i}$	$\boldsymbol{d_i}$	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$
1	25	0	3	0

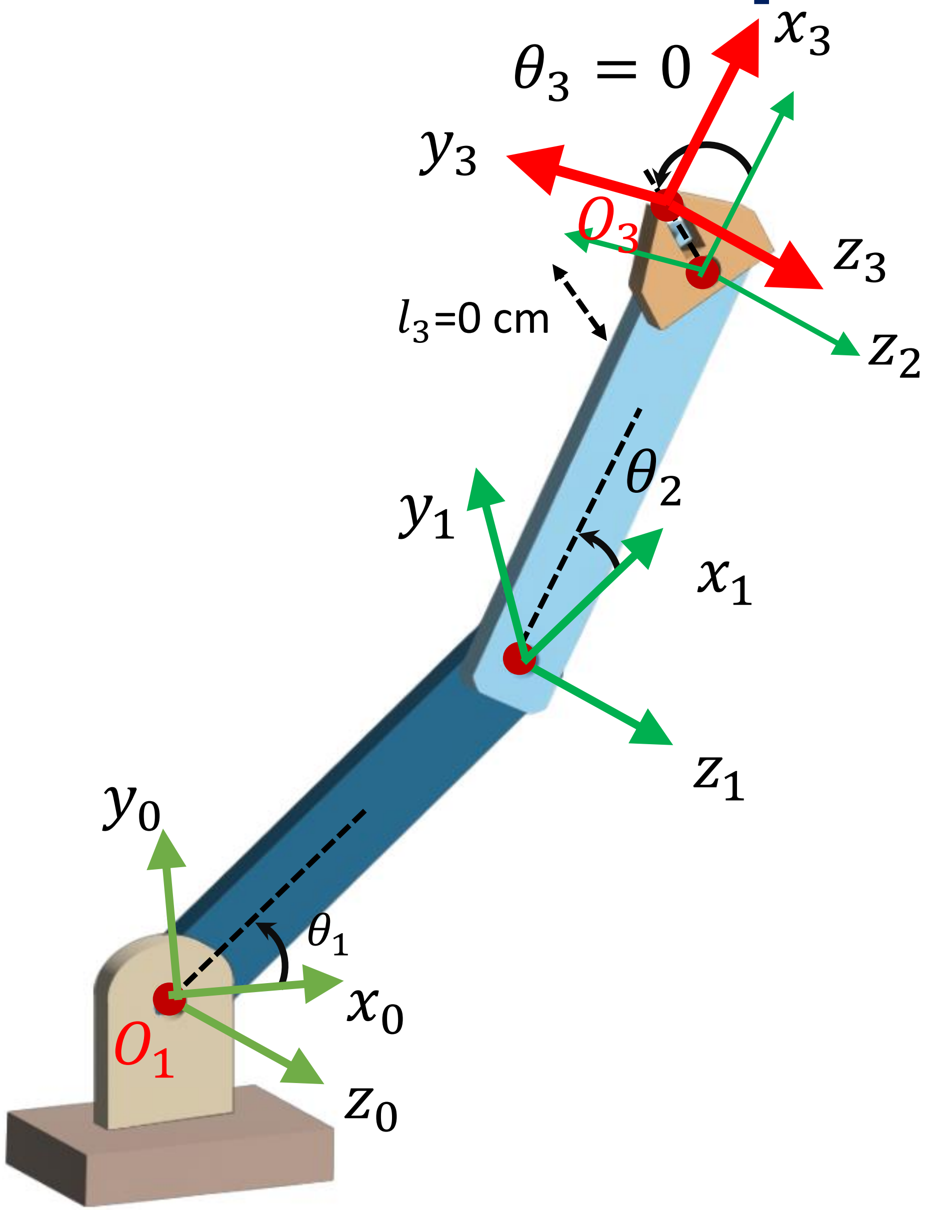
Example 1: 3-Revolute Manipulator



- Apply the same procedure for frame {2}
- z_1 is parallel to z_2; hence, $\boldsymbol{x_2}$ is a common normal between z_1 and z_2

$\boldsymbol{i}$	$\boldsymbol{\theta_i}$	$\boldsymbol{d_i}$	$\boldsymbol{a_i}$	$\boldsymbol{\alpha_i}$
2	40	0	4	0

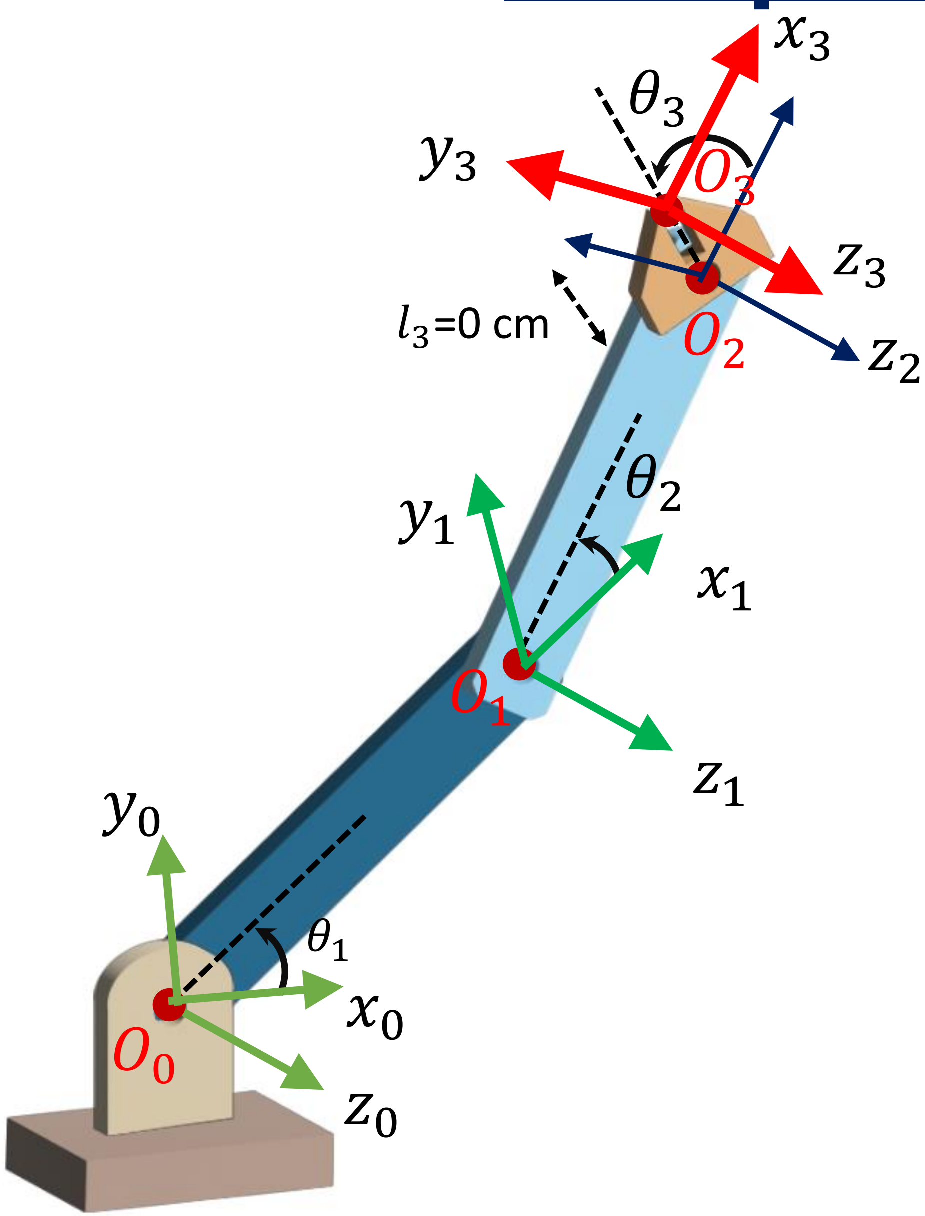
Example 1: Revolute Manipulator



- Establish the end-effector frame $O_3x_3y_3z_3$
- z_3 is parallel to z_2, and and x_3 is a common normal between z_2 and z_3.

i	θ_i	d_i	a_i	α_i
3	$\theta_3 = 0$	0	$l_3 = 0$	0

Example 1: Revolute Manipulator



DH Parameter Table

i	θ_i	d_i	a_i	α_i
1	25	0	3	0
2	40	0	4	0
3	0	0	0	0

Example 1: Revolute Manipulator

Recall

$$^0p = {}^0T_3\,{}^3p = {}^0T_1\,{}^1T_2\,{}^2T_3\,{}^3p; \quad {}^{i-1}\boldsymbol{T}_{\boldsymbol{i}} = \begin{pmatrix} cos\theta_i & -sin\theta_i\, cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^0p = \begin{pmatrix} 0.906 & -0.423 & 0 & 2.718 \\ 0.423 & 0.906 & 0 & 1.268 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0.766 & -0.643 & 0 & 3.064 \\ 0.643 & 0.766 & 0 & 2.571 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$^0p = \begin{pmatrix} 4.409 \\ 4.893 \\ 0 \\ 1 \end{pmatrix}$$

Using RVC toolbox for transformation matrix

$$^{i-1}T_i = R_{(i-1)}(\theta_i).\,Q_{(i-1)}(d_i).\,Q_i(a_i).\,R_i(\alpha_i)$$

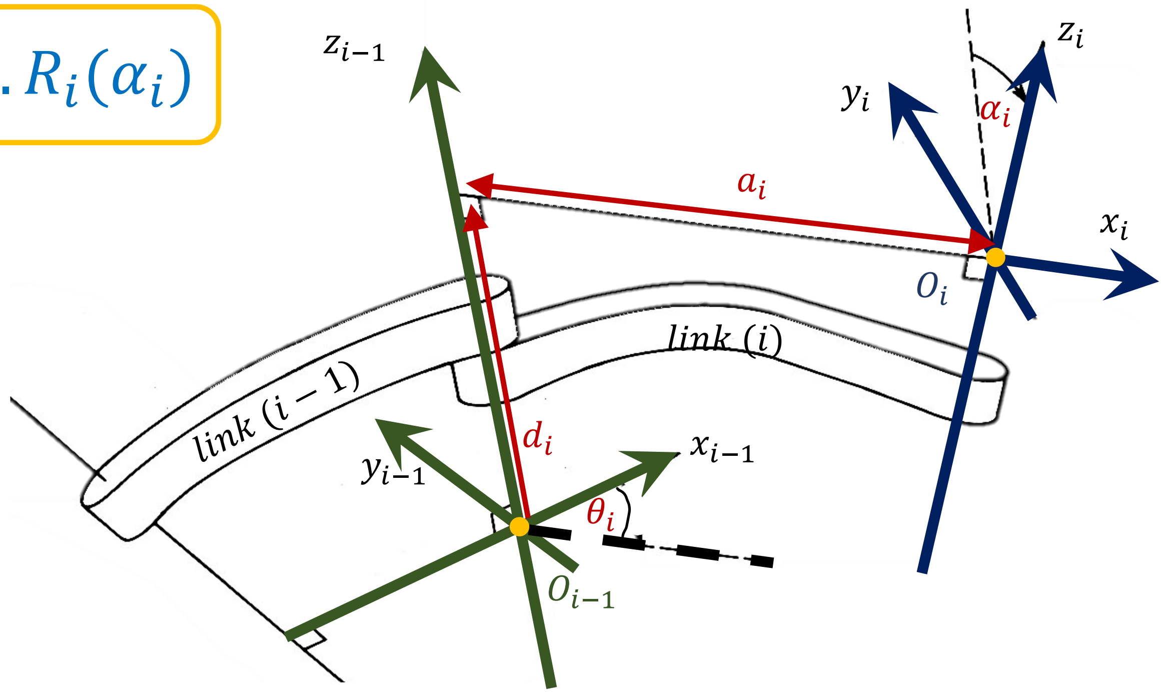
- Rotation matrix in 3D: rotx(), roty(), and rotz() for x, y z axis respectively
- Homogeneous transformation (HT) matrix for rotation: trotz() for z axis, and trotx() for x axis
- HT matrix for translation: transl([x y z])

Quiz: Using RVC toolbox for transformation matrix

What is the transformation matrix (${}^{i-1}T_i$) between frame {i-1} and {i}?

$$ {}^{i-1}T_i = R_{(i-1)}(\theta_i).Q_{(i-1)}(d_i).Q_i(a_i).R_i(\alpha_i) $$

Hint: Trotx, transl



Answer

```
startup_rvc
%%% Convert degree to radian %%%

theta_i = deg2rad(theta_degree);
alpha_i = deg2rad(alpha_degree);
```

${}^{i-1}T_i$ = **trotz(theta_i)*transl(d_i)*transl(a_i)*trotx(alpha_i)**

Using RVC toolbox in Example 1

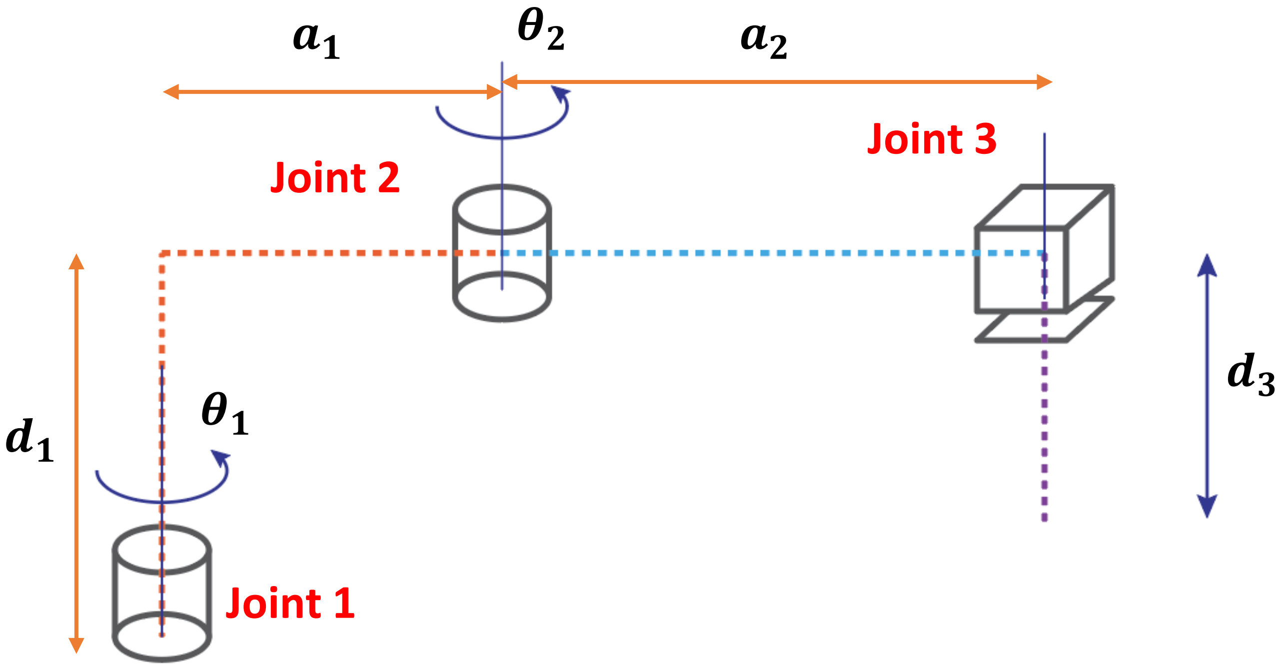
```
startup_rvc
%%%% Convert degree to radian %%%%
theta = deg2rad([25 40 0]);

%%%% Create three link of the robot  %%%

L(1) = Link('revolute', 'd', d1, 'a', a1, 'alpha', alpha1, 'offset', 0); % Link 1.
L(2) = Link('revolute', 'd', d2, 'a', a2, 'alpha', alpha2, 'offset', 0); % Link 2.
L(3) = Link('revolute', 'd', d3, 'a', a3, 'alpha', alpha3, 'offset', 0); % Link 3.

%%%%% Connect all links in a series %%%%

robot= SerialLink(L, 'name', ' three link');

%%%%% Forward kinematic %%%%

Matrix= robot.fkine([theta]);
```

i	θ_i	d_i	a_i	α_i
1	25	0	3	0
2	40	0	4	0
3	0	0	0	0

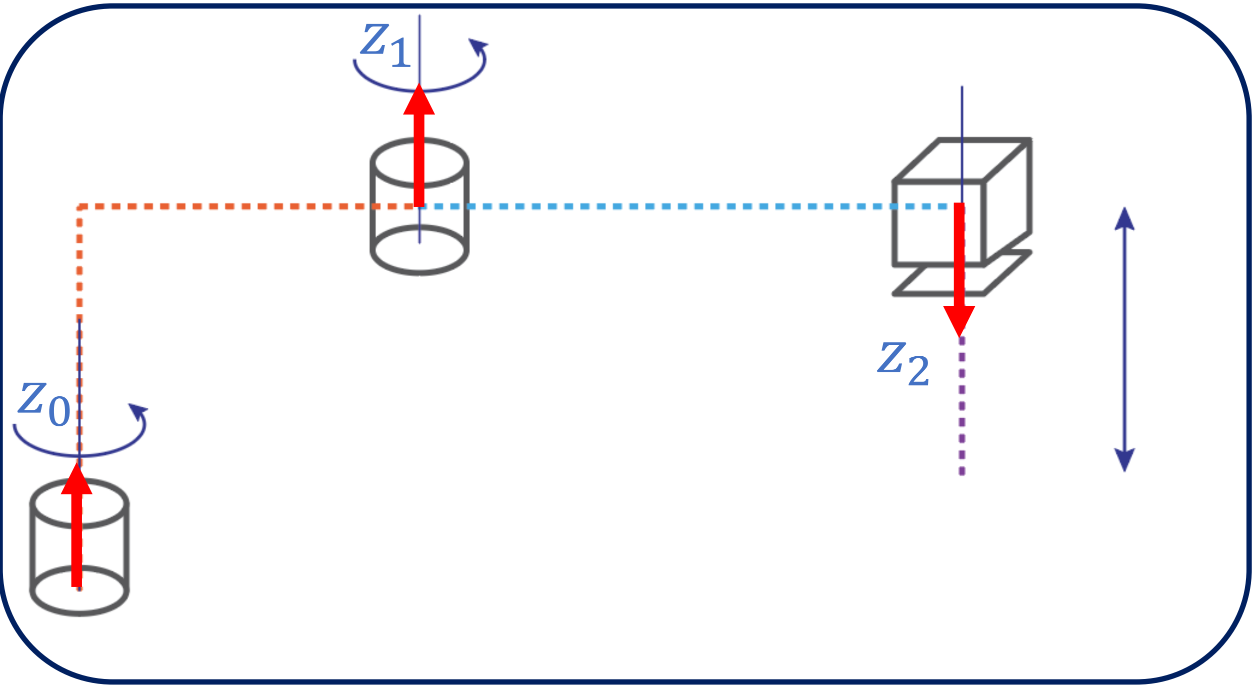
(set $\boldsymbol{\theta_i}$=0 if there is no offset angle at the home position)

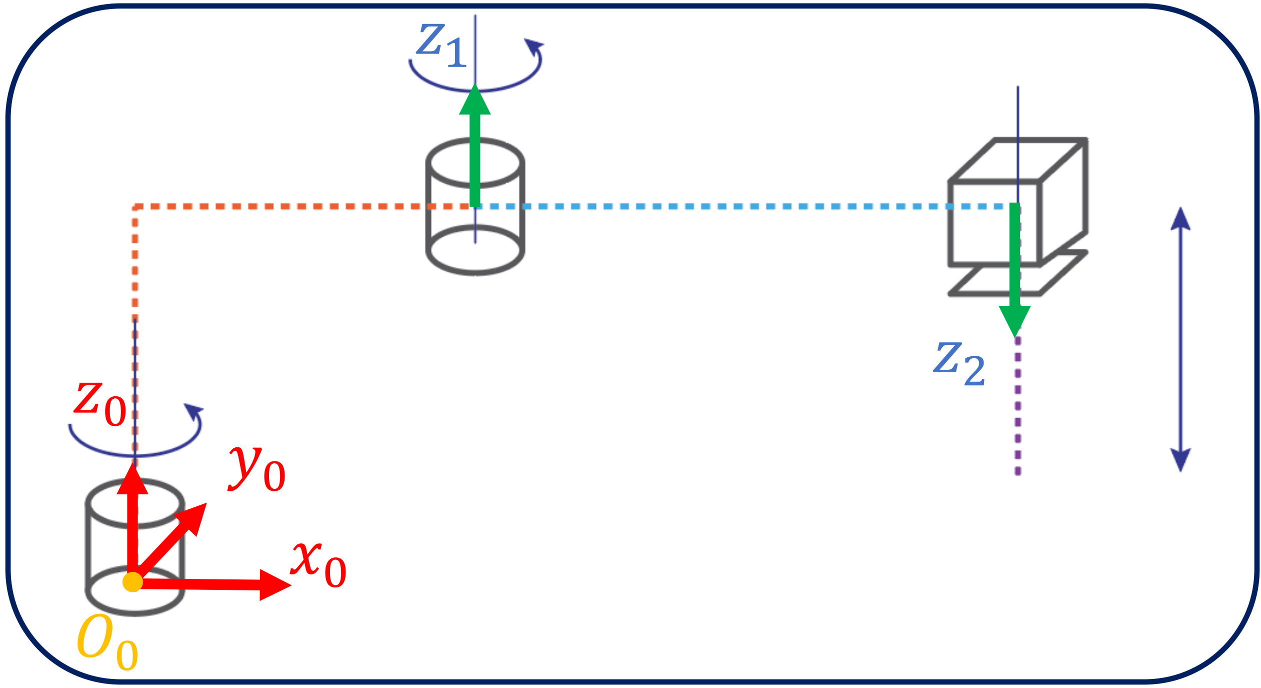
$$ {}^{0}T_3 = \begin{matrix} 0.4226 & -0.9063 & 0 & 4.409 \\ 0.9063 & 0.4226 & 0 & 4.893 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} $$

Example 2: SCARA Robot

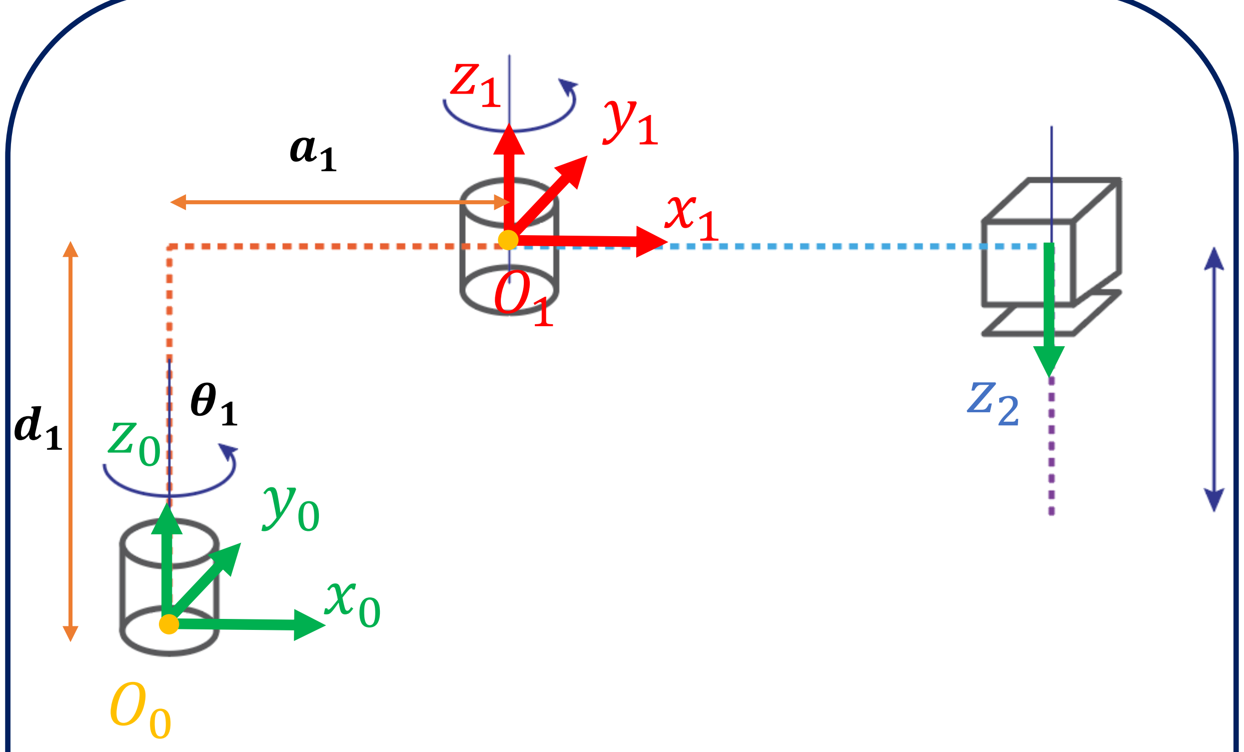


Example 2: SCARA Robot

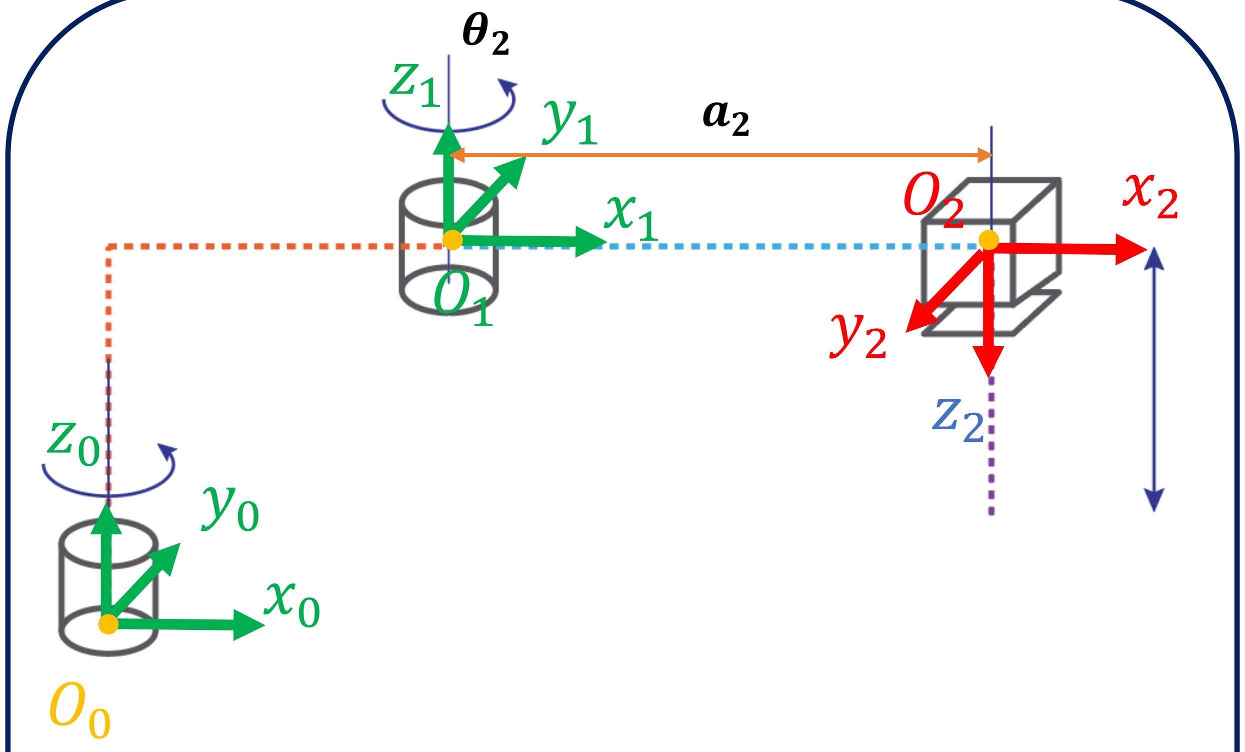




Example 2: SCARA Robot

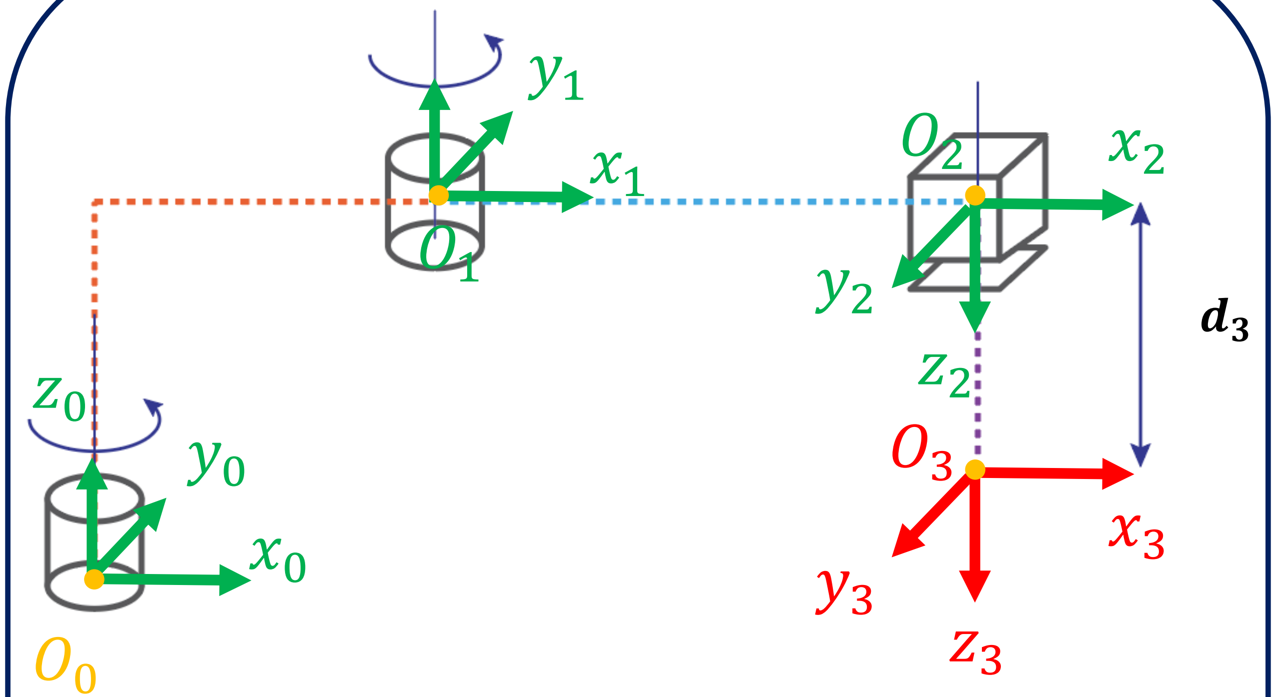


i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	0



i	θ_i	d_i	a_i	α_i
2	θ_2	0	a_2	180

Example 2: SCARA Robot



i	θ_i	d_i	a_i	α_i
3	$\mathbf{0}$	d_3	0	0

i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	0
2	θ_2	$\mathbf{0}$	a_2	180
3	$\mathbf{0}$	d_3	0	0

Example 2: SCARA Robot

i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	0
2	θ_2	$\mathbf{0}$	a_2	180
3	**0**	d_3	0	0

Self Practice

i	θ_i	d_i	a_i	α_i
1	15	10	15	0
2	45	0	20	180
3	0	20	0	0

$$ {}^{0}T_3 = \begin{matrix} 0.5000 & 0.8660 & 0 & 24.49 \\ 0.8660 & -0.5000 & 0 & 21.2 \\ 0 & 0 & -1 & -10 \\ 0 & 0 & 0 & 1 \end{matrix} $$

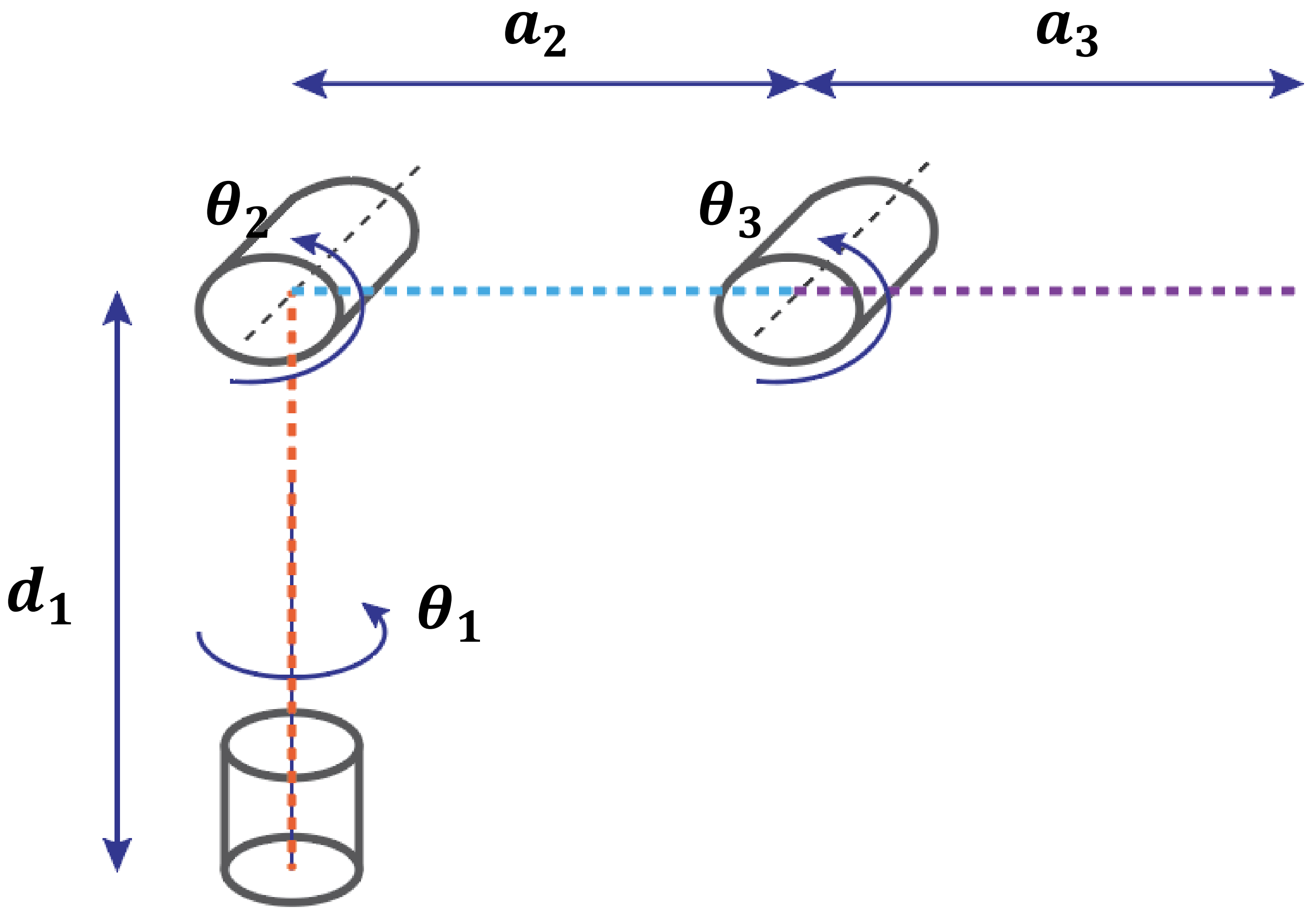
❑ **Hint using RVC**

(i) Use the ‘link’ function as the previous example

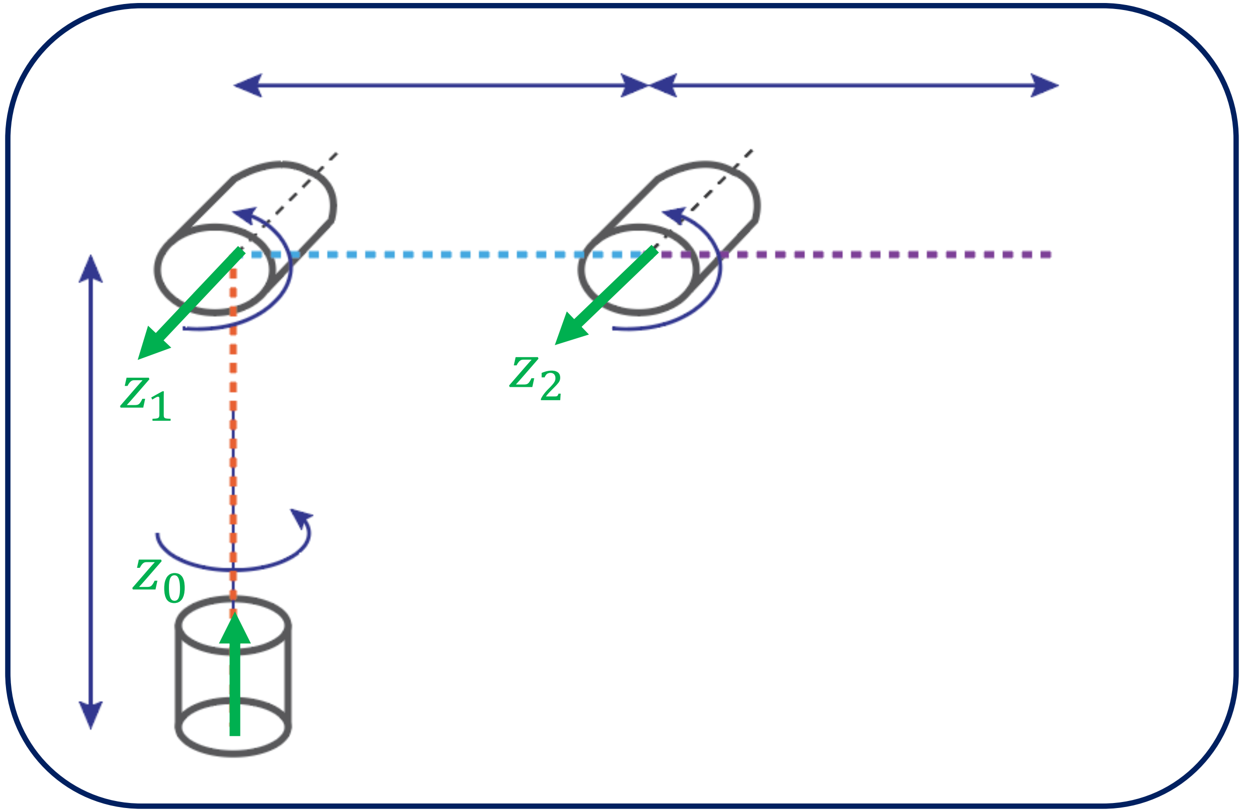
(ii) robot= SerialLink(L, 'name', ‘ SCARA’);

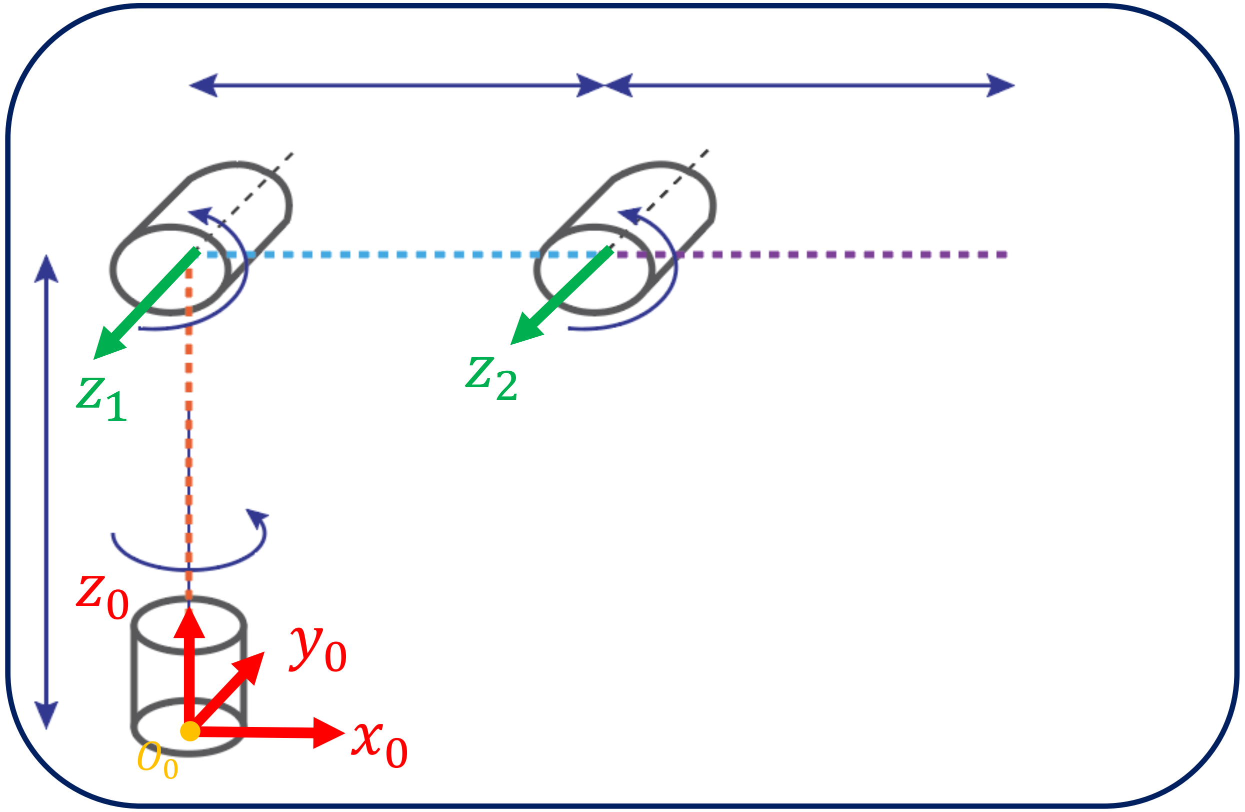
(iii) f= robot.fkine([theta1 theta2 **0**]); % theta3=0 for revolute joint

Example 3: Articulated Robot

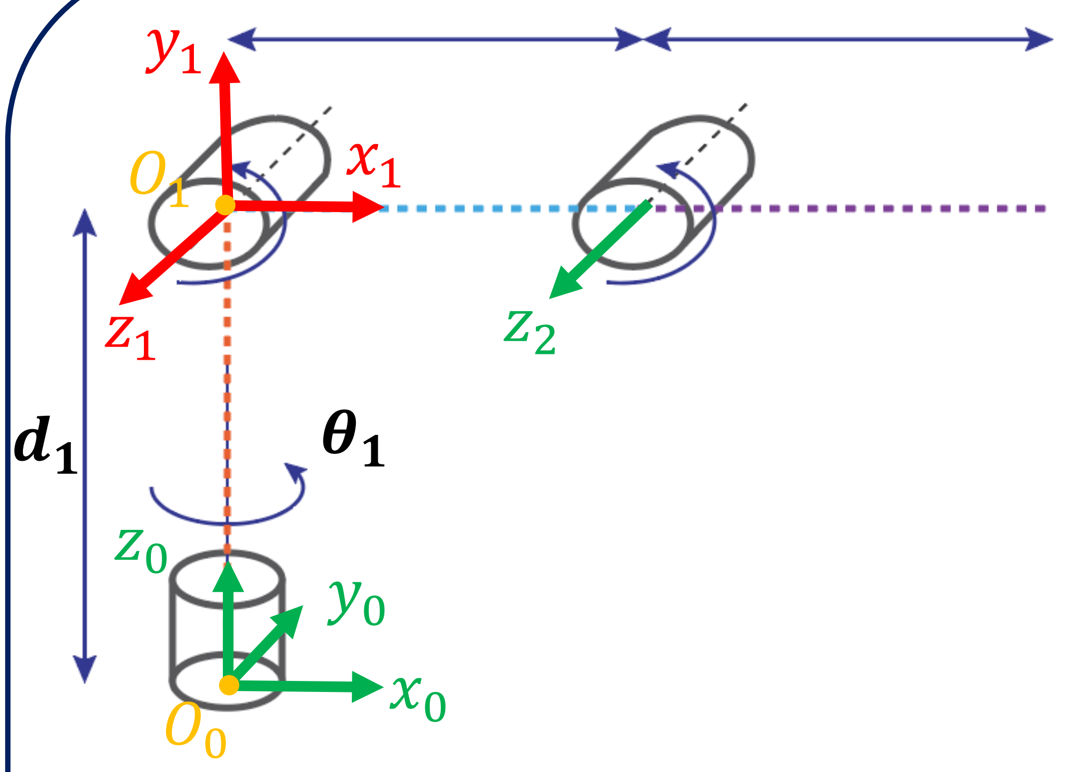


Example 3: Articulated Robot

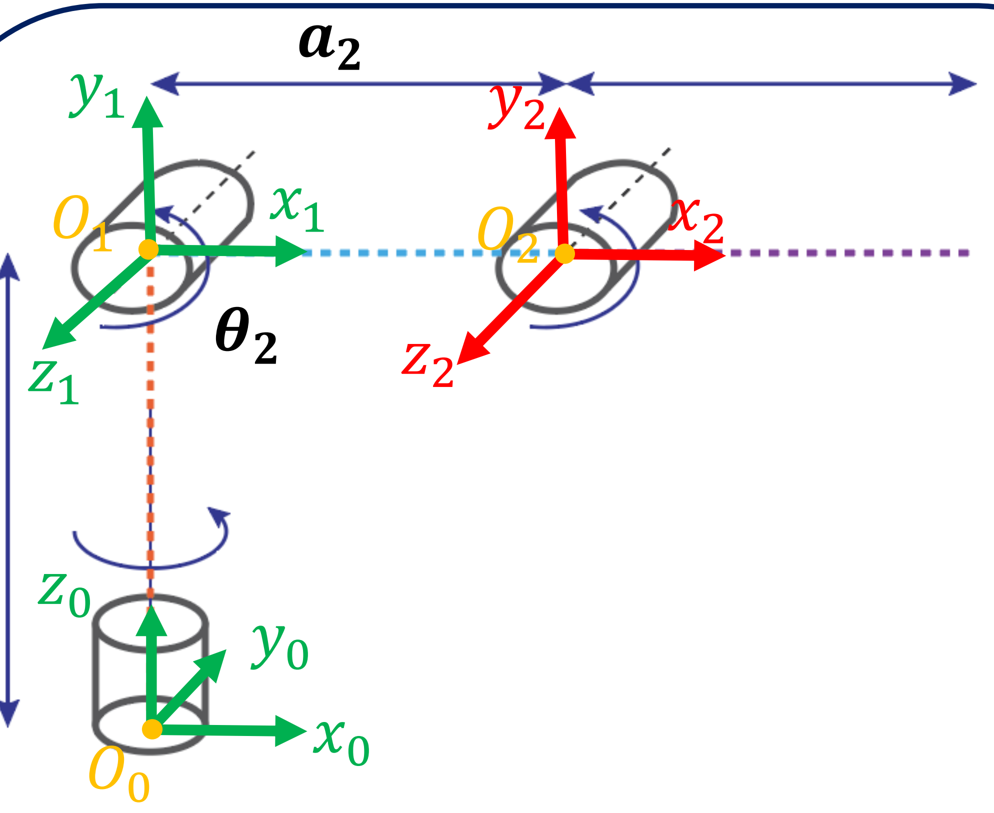




Example 3: Articulated Robot

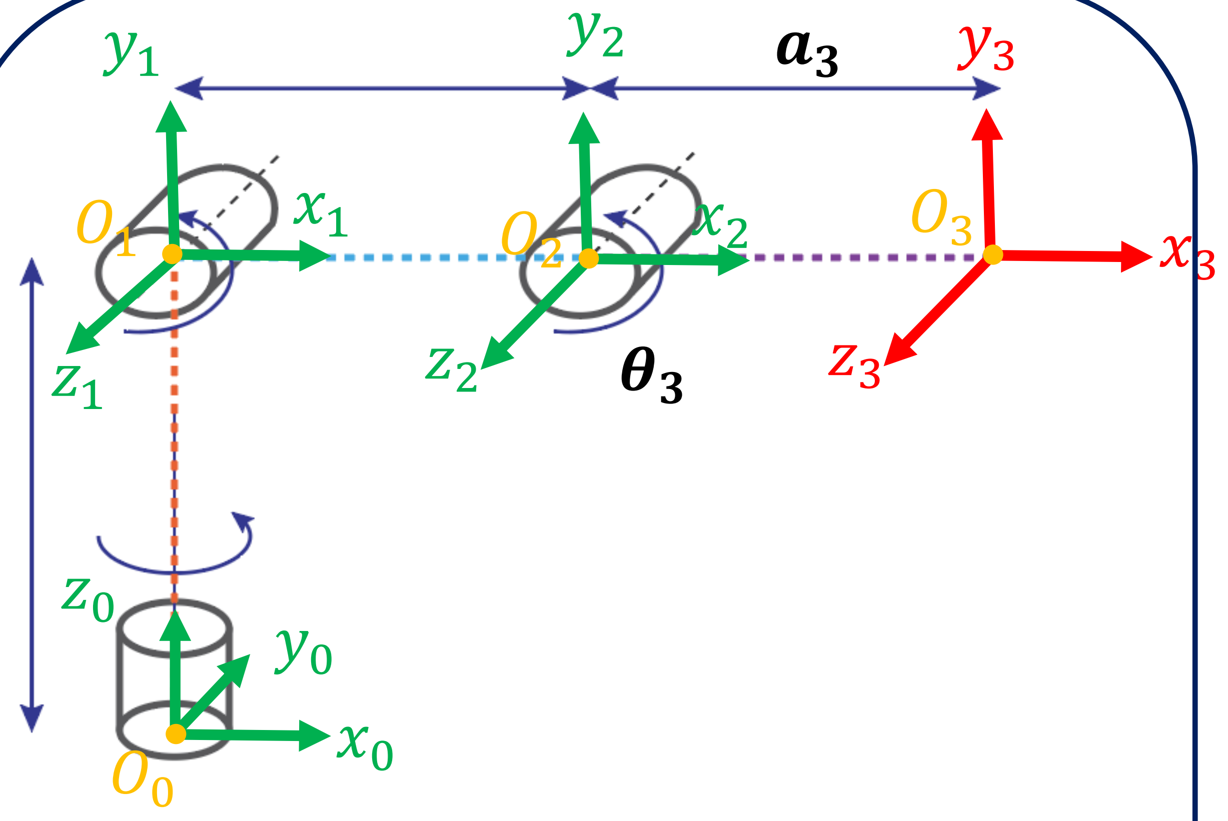


i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	0	90



i	θ_i	d_i	a_i	α_i
2	θ_2	0	a_2	0

Example 3: Articulated Robot



i	θ_i	d_i	a_i	α_i
3	θ_3	0	a_3	0

i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	0	90
2	θ_2	0	a_2	0
3	θ_3	0	a_3	0

Example 3: Articulated Robot

- **Hint using RVC**

(i) **Use the 'link' function as the previous example**

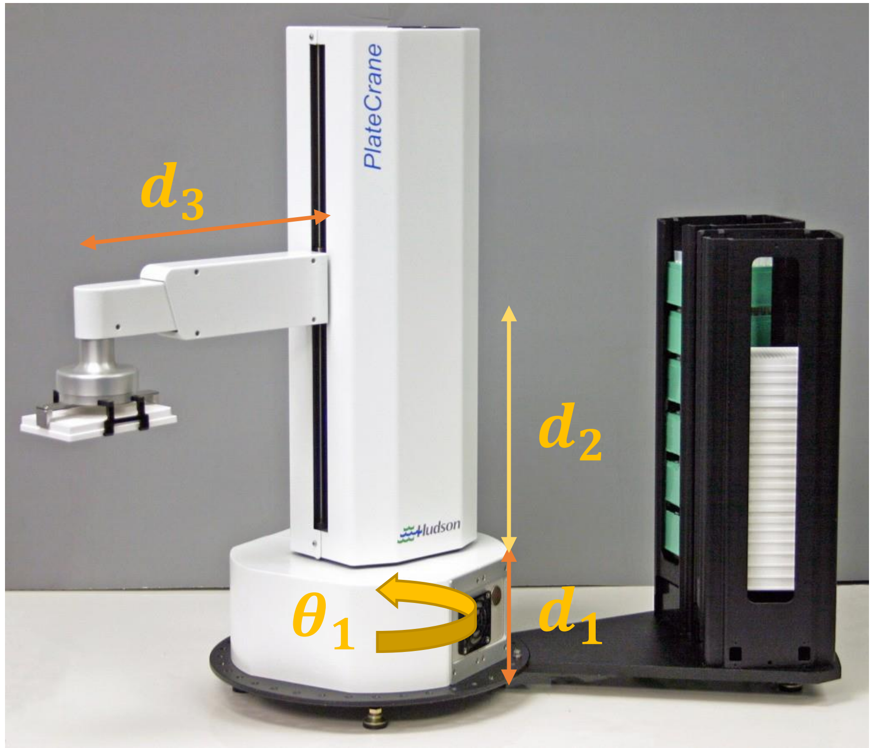
(ii) robot= SerialLink(L, 'name', ' Articulated');

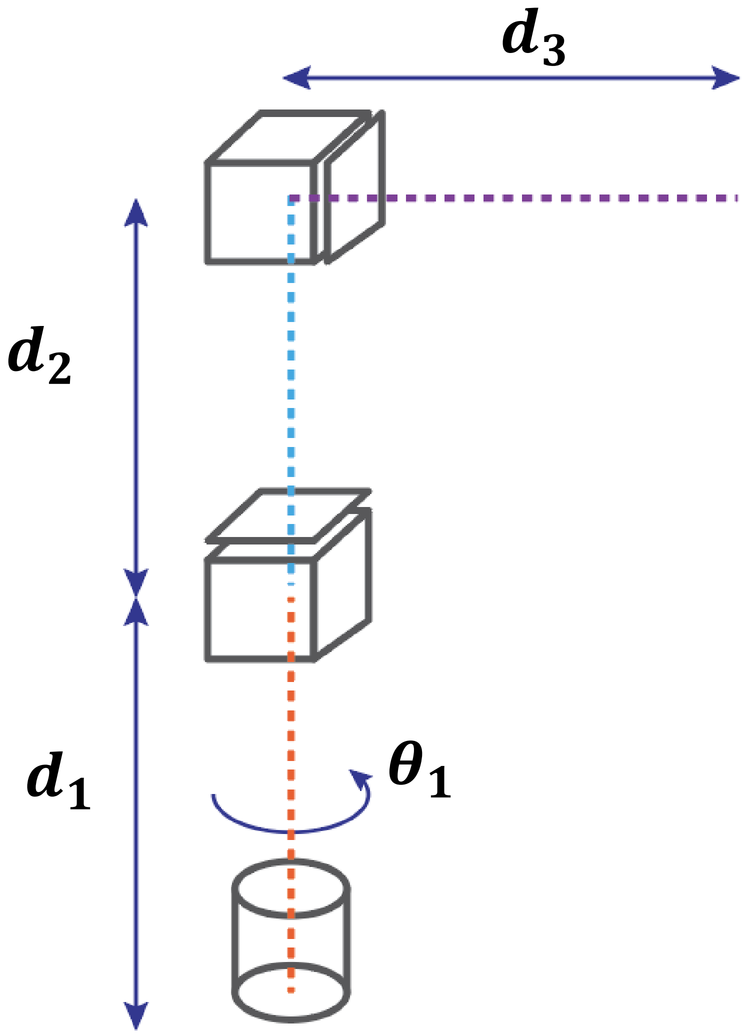
(iii) f= robot.fkine([theta1 theta2 theta3]);

i	θ_i	d_i	a_i	α_i
1	10	32	0	90
2	50	0	11	0
3	65	0	16	0

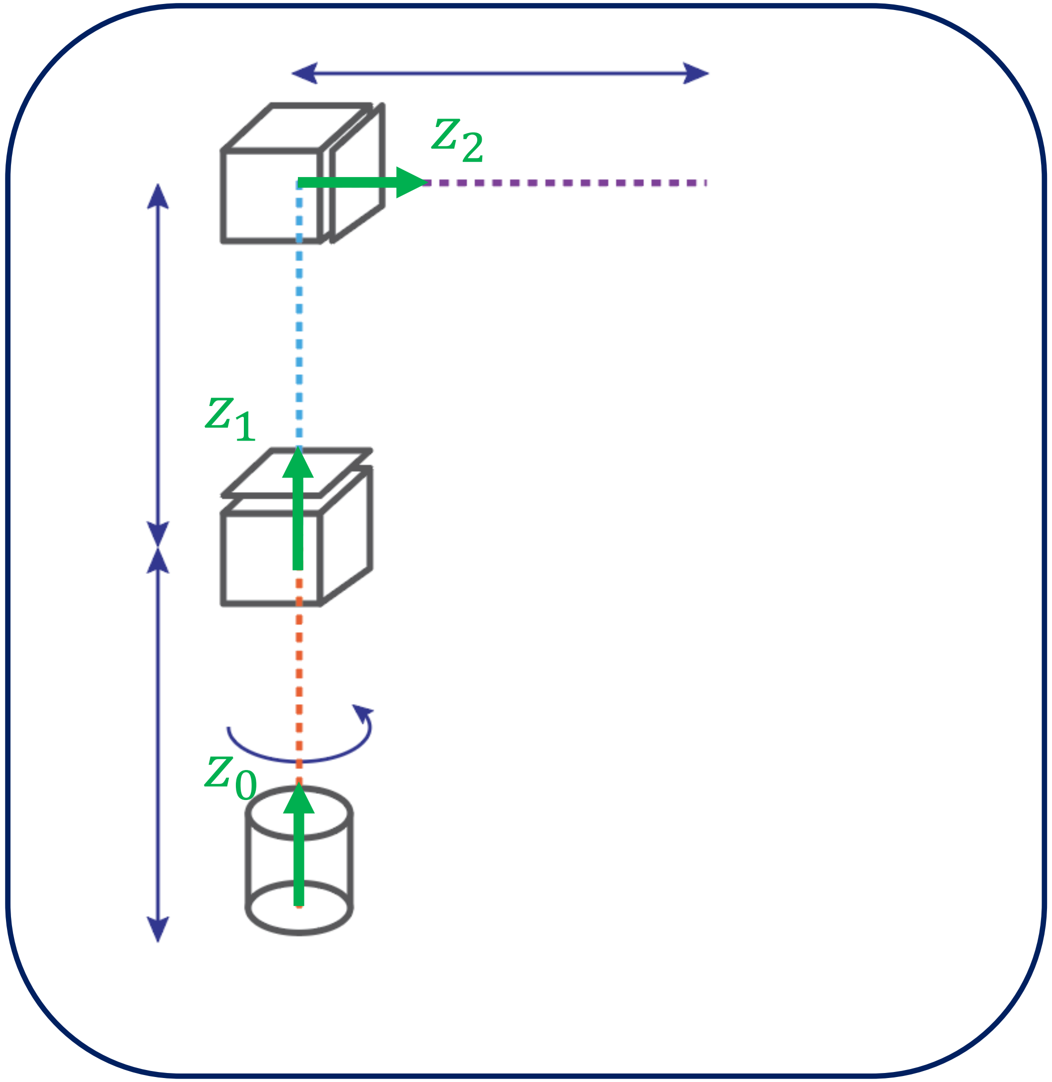
$$ {}^{0}T_3 = \begin{bmatrix} -0.4162 & -0.8925 & 0.1736 & 0.3041 \\ -0.0734 & -0.1574 & -0.9848 & 0.05362 \\ 0.9063 & -0.4226 & 0 & 54.93 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

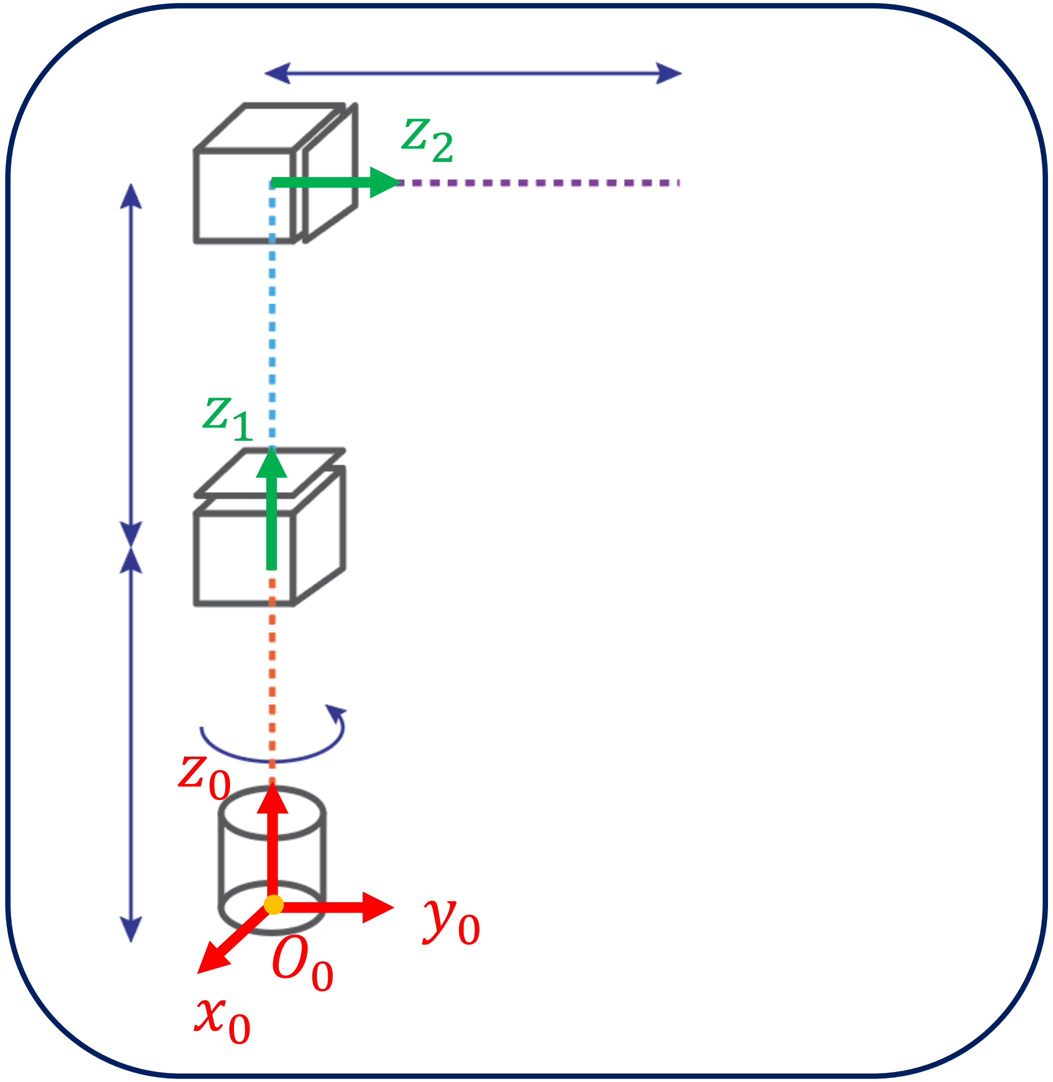
Example 4: Cylindrical robot



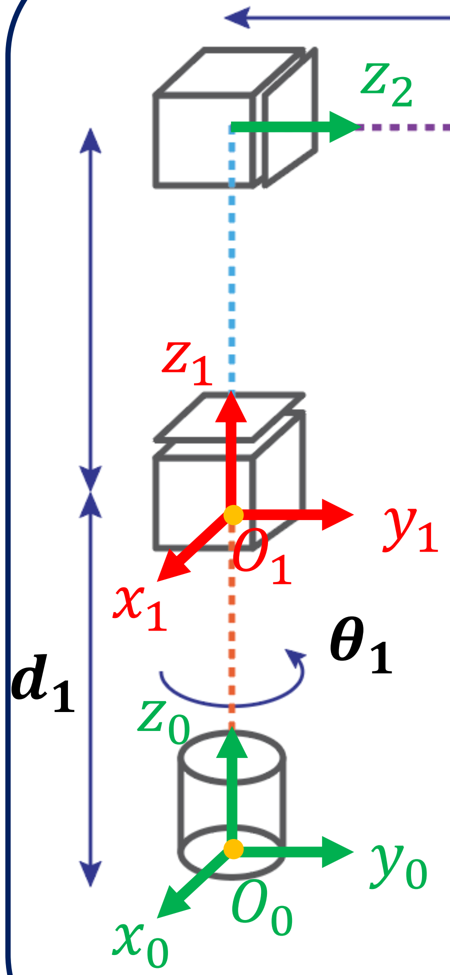


Example 4: Cylindrical robot

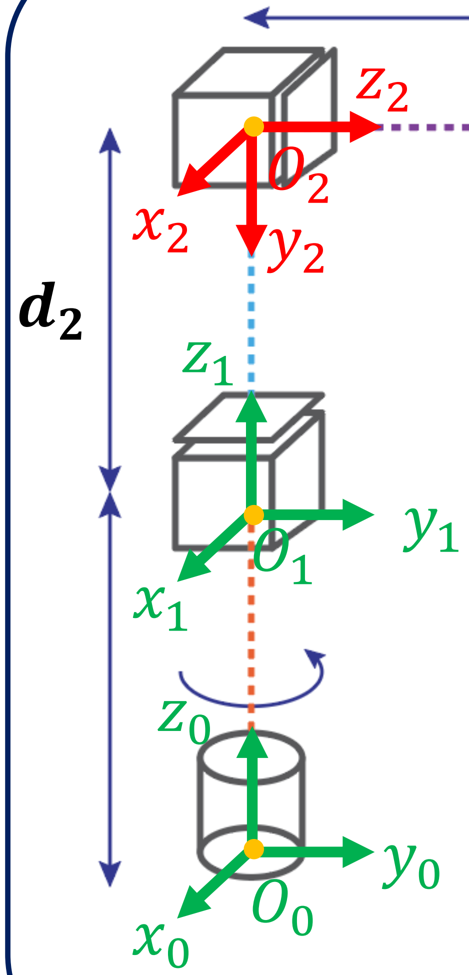




Example 4: Cylindrical robot

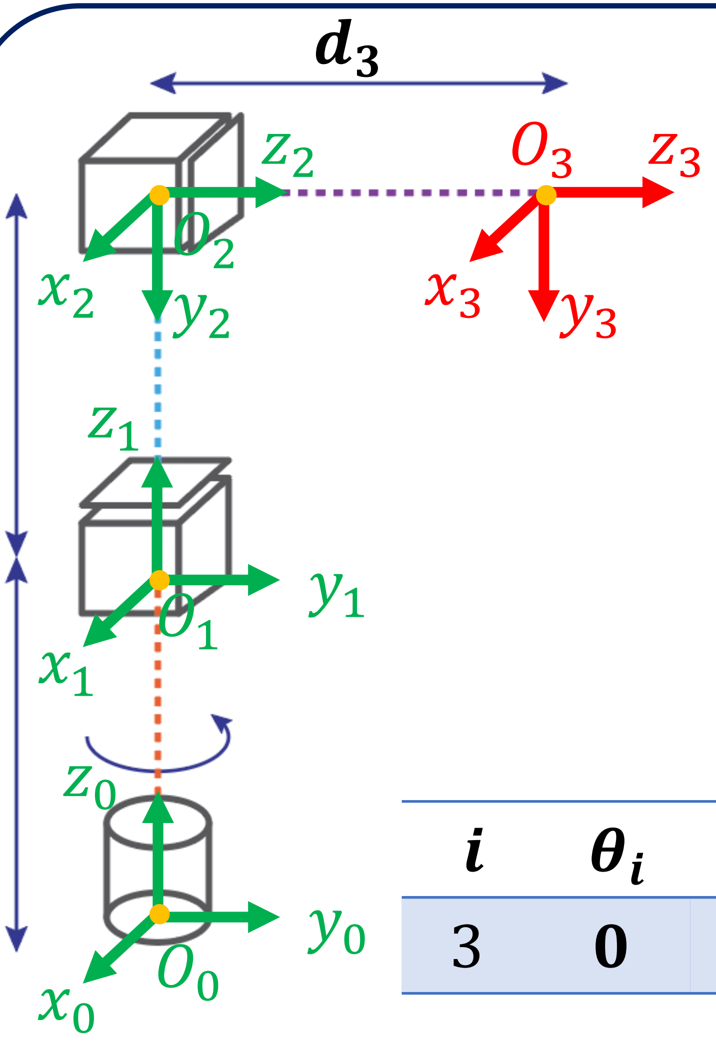


i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	0	0



i	θ_i	d_i	a_i	α_i
2	**0**	d_2	**0**	-90

Example 4: Cylindrical robot



i	θ_i	d_i	a_i	α_i
3	**0**	d_3	**0**	0

All combined

i	θ_i	d_i	a_i	α_i
1	θ_1	d_1	**0**	0
2	**0**	d_2	**0**	-90
3	**0**	d_3	**0**	0

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

Example 4: Cylindrical robot

❑ Hint using RVC

(i) Use the 'link' function as the previous example

(ii) robot= SerialLink(L, 'name', ' cylindrical');

(iii) f= robot.fkine([theta1 theta2 theta3]);

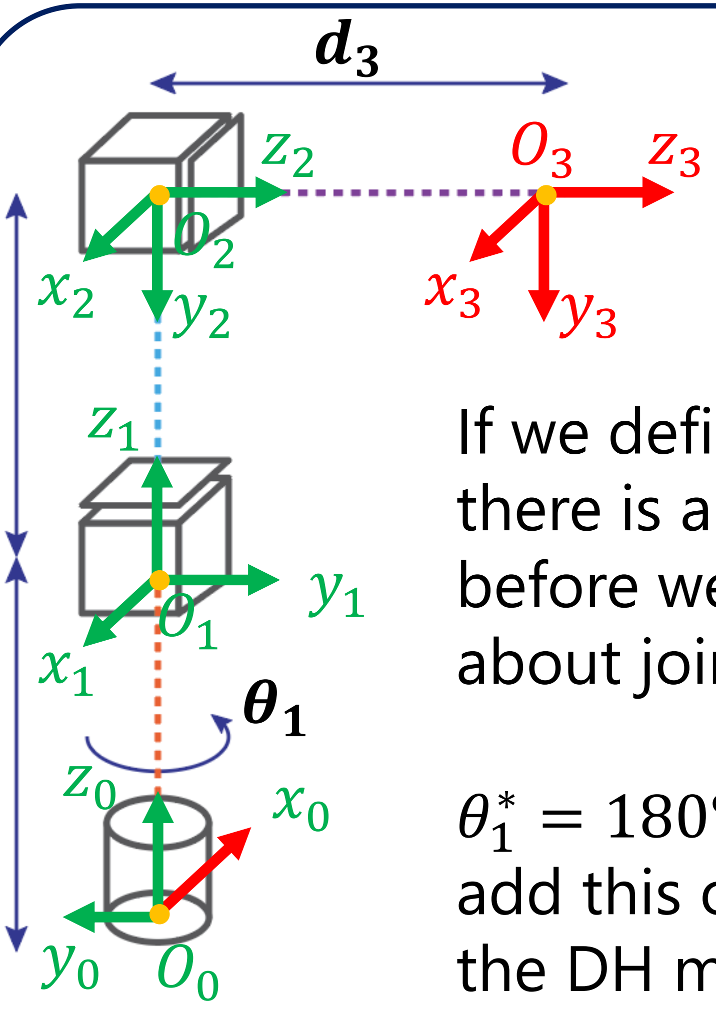
Example $\theta_1 = 68°$

i	θ_i	d_i	a_i	α_i
1	68	8	0	0
2	0	6	0	-90
3	0	8	0	0

$^0T_3$ =

0.3746	0	-0.9272	-7.417
0.9272	0	0.3746	2.997
0	-1	0	14
0	0	0	1

Example 5: Cylindrical robot-offset angle



If we define x_o like this, there is an offset angle before we rotate $\boldsymbol{\theta_1}$ about joint 1.

$\theta_1^* = 180°$. We need to add this offset angle to the DH matrix

i	θ_i	d_i	a_i	α_i
1	$\boldsymbol{\theta_1 + 180}$	d_1	**0**	0
2	**0**	d_2	**0**	-90
3	**0**	d_3	**0**	0

Example $\theta_1 = 68°$

i	θ_i	d_i	a_i	α_i
1	248	8	0	0
2	0	6	0	-90
3	0	8	0	0

$$^0T_3 = \begin{matrix} -0.3746 & 0 & 0.9272 & 7.417 \\ -0.9272 & 0 & -0.3746 & -2.997 \\ 0 & -1 & 0 & 14 \\ 0 & 0 & 0 & 1 \end{matrix}$$

Lecture 3: Summary

i	θ_i	d_i	a_i	α_i
0	0	0	0	0
1	θ_1	d_1	a_1	α_1
2	θ_2	d_2	a_2	α_2
3	θ_3	d_3	a_3	α_3

RVC Toolbox is very useful to build the robot & calculate the matrix!

$$^{i-1}T_i = R_{(i-1)}(\theta_i).\,Q_{(i-1)}(d_i).\,Q_i(a_i).\,R_i(\alpha_i)$$

Four Parameters of DH

angle offset length twist

Next week

- ❑ Lecture 5: Inverse Kinematics and The Jacobian

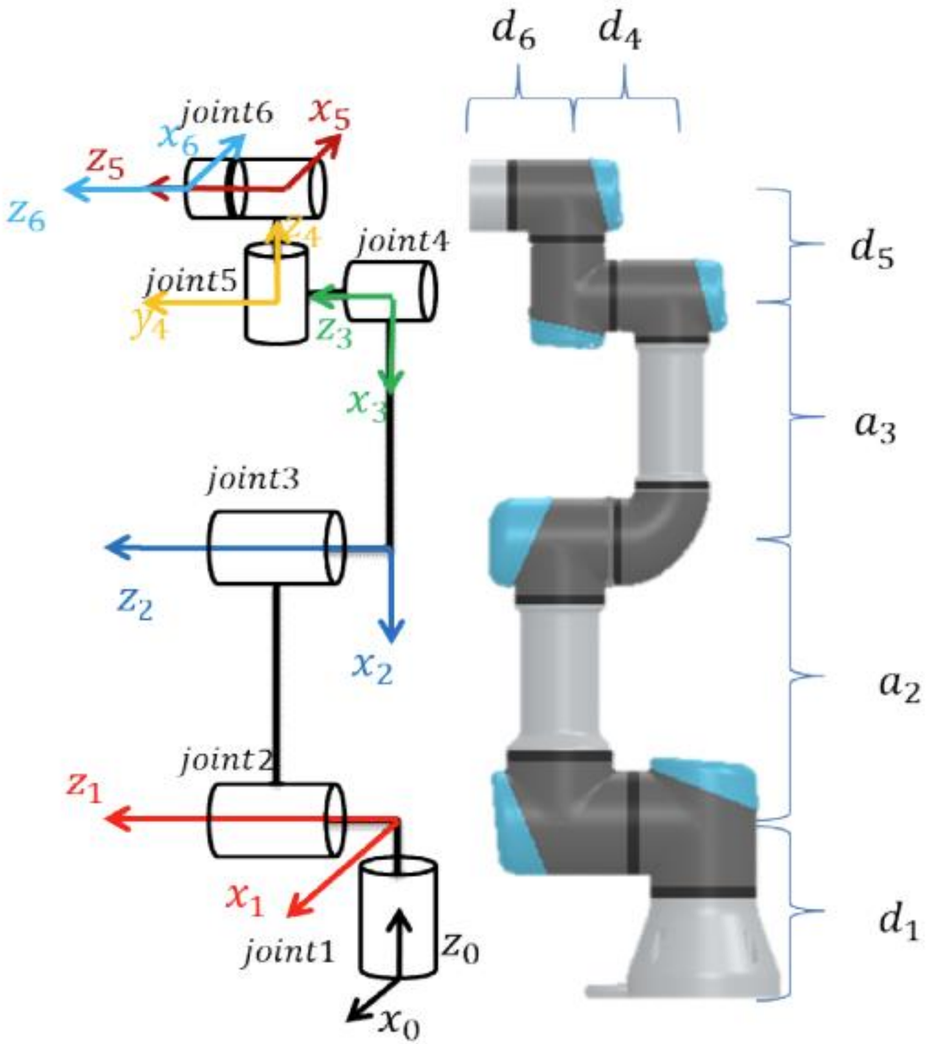
- ❑ Marking ROBOT2 during lab sessions

- ❑ Quiz1 (70 minutes on Moodle – Monday 26th from 16pm)

    - Time: 70 minutes on Moodle. Please start before 16:15pm
    - From weeks 1 to 4 (except slides 52-59 in Lecture 3)
    - Seven questions, including multiple choice and calculation questions
    - Can use Matlab for your calculation (e.g., RVC Toolbox)
    - Write your answer with 4 decimal places

Self practice

Find the DH parameters for UR5e (forward kinematics of MTRN4230's robots)



UR5e				
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]
Joint 1	0	0	0.1625	π/2
Joint 2	0	-0.425	0	0
Joint 3	0	-0.3922	0	0
Joint 4	0	0	0.1333	π/2
Joint 5	0	0	0.0997	-π/2
Joint 6	0	0	0.0996	0