Introduction to Robotics

Manipulation and Programming

Unit 2: Kinematics

DENAVIT-HARTENBERG - HTM FOR DYNAMICS

DR. ERIC CHOU

IEEE SENIOR MEMBER

Objectives

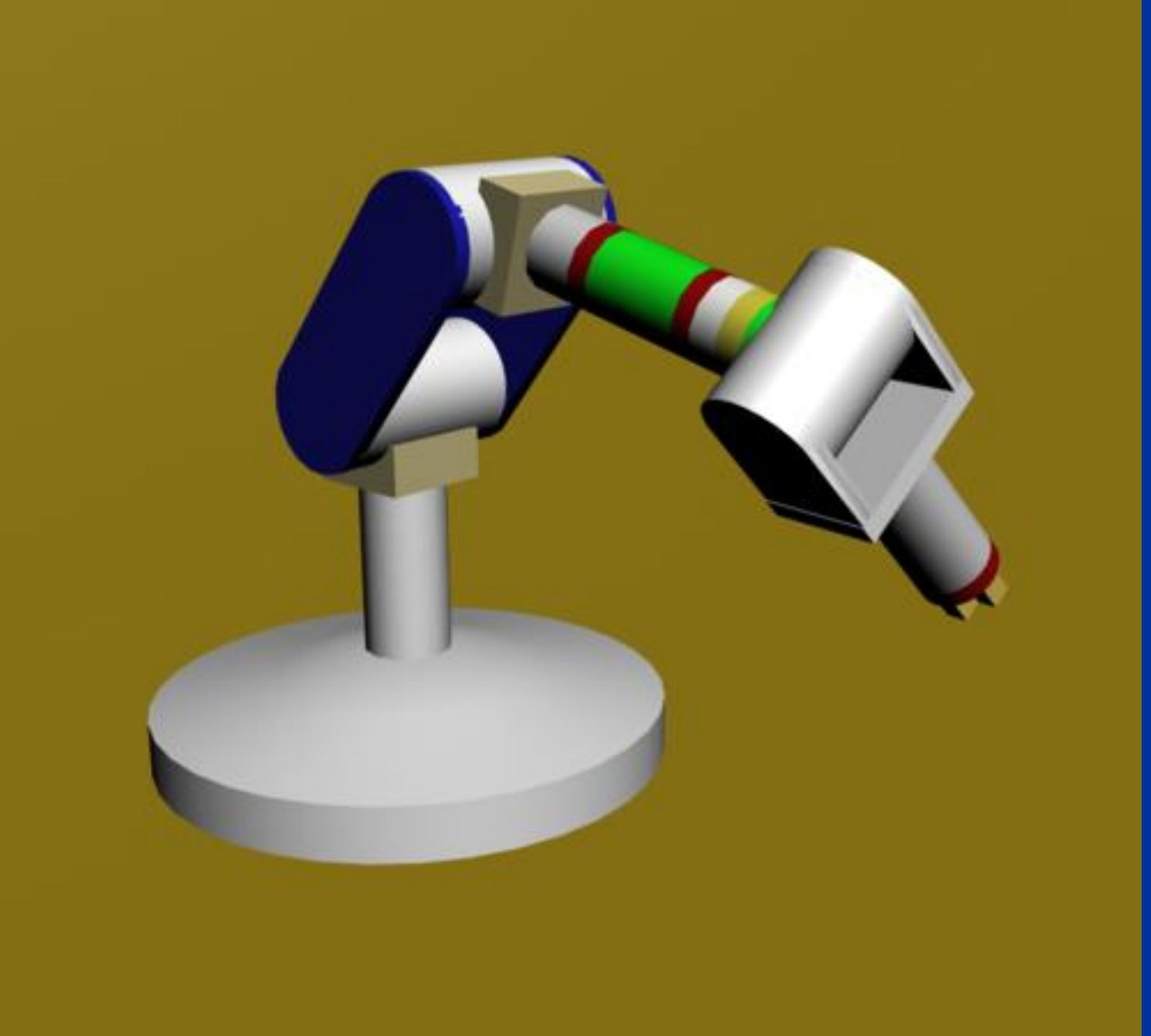
- Robot Serial Manipulator
- Singularity
- Denavit-Hartenberg Matrix
- Design of DH Rules
- Creation of DH Table
- Creation of DH Matrix
- Calculation of Point Position using DH_HTM matrix

Robot Serial Manipulator

SECTION 1

Robot Serial Manipulator

- **Serial manipulators** are the most common industrial robots and they are designed as a series of links connected by motor-actuated joints that extend from a base to an end-effector. Often they have an anthropomorphic arm structure described as having a "shoulder", an "elbow", and a "wrist".
- Serial robots usually have six joints, because it requires at least six degrees of freedom to place a manipulated object in an arbitrary position and orientation in the workspace of the robot.
- A popular application for serial robots in today's industry is the pick-and-place assembly robot, called a SCARA robot, which has four degrees of freedom.



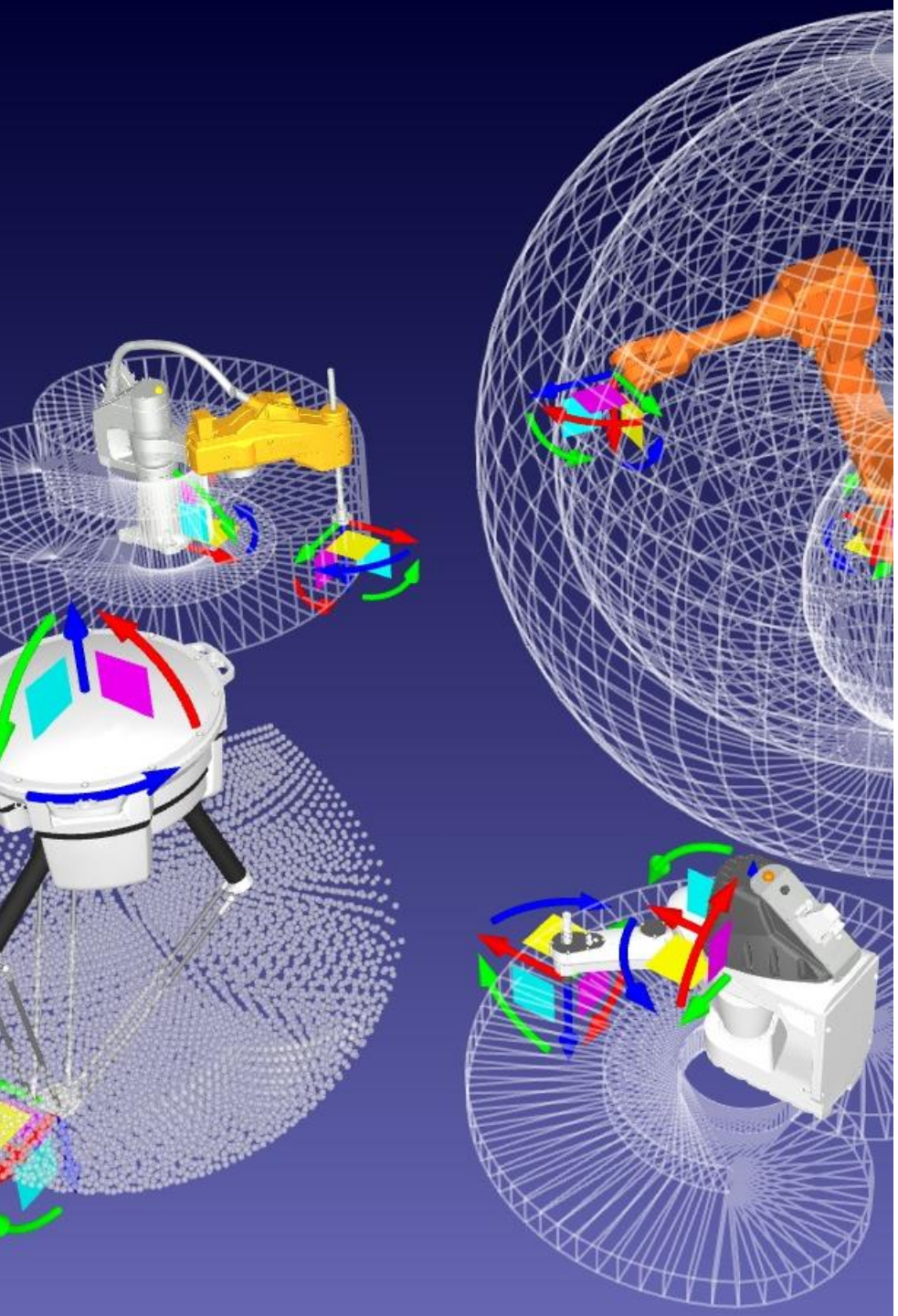
Robot Serial Manipulator 6 DOF

Kinematics

- The position and orientation of a robot's end effector are derived from the joint positions by means of a geometric model of the robot arm.
- For serial robots, the mapping from joint positions to end-effector pose is easy, the inverse mapping is more difficult. Therefore, most industrial robots have special designs that reduce the complexity of the inverse mapping.

Workspace

- The reachable workspace of a robot's end-effector is the manifold of reachable frames.
- The dextrous workspace consists of the points of the reachable workspace where the robot can generate velocities that span the complete tangent space at that point, i.e., it can translate the manipulated object with three degrees of freedom, and rotate the object with three degrees of rotation freedom.



Workspace

- The relationships between joint space and Cartesian space coordinates of the object held by the robot are in general multiple-valued: the same pose can be reached by the serial arm in different ways, each with a different set of joint coordinates.
- Hence the reachable workspace of the robot is divided in configurations (also called assembly modes), in which the kinematic relationships are locally one-to-one.

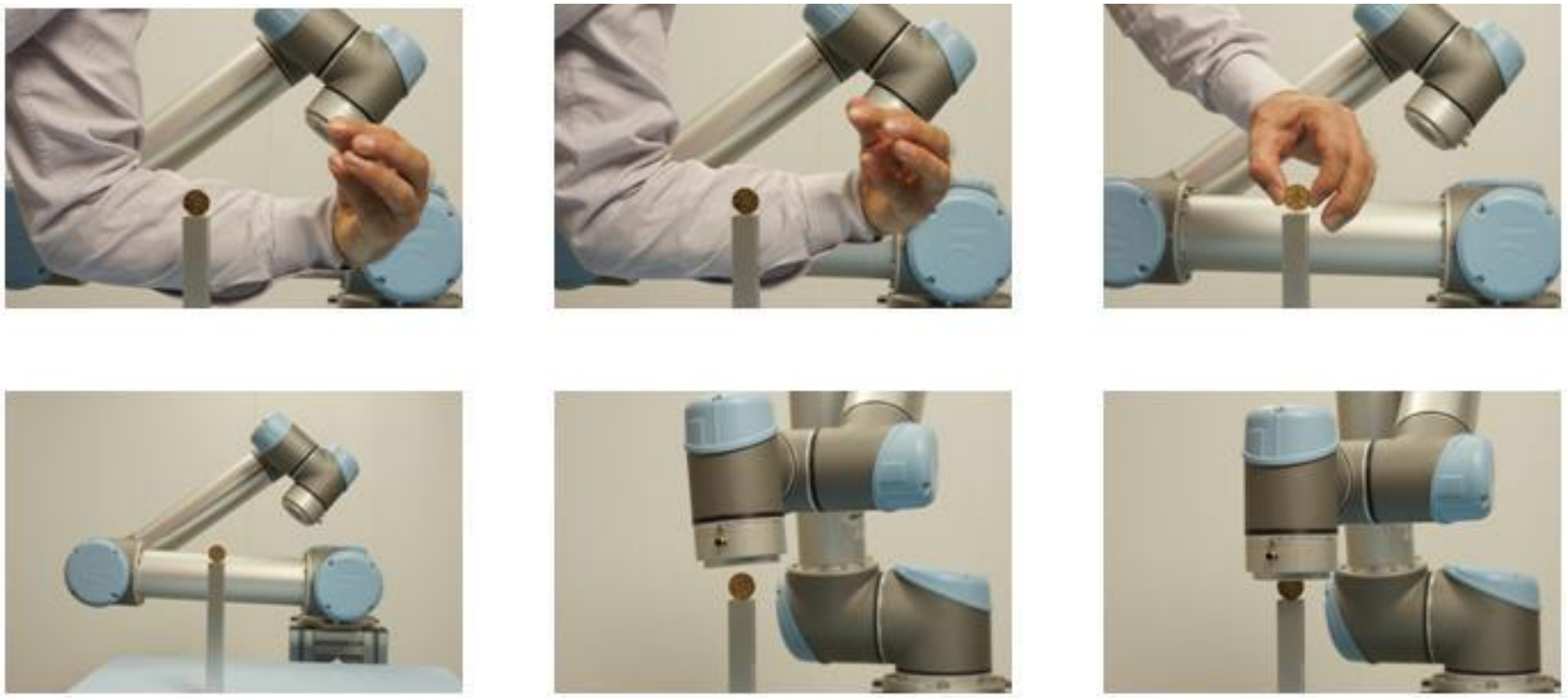
Singularity

- A **singularity** is a configuration of a serial manipulator in which the joint parameters no longer completely define the position and orientation of the end-effector. Singularities occur in configurations when joint axes align in a way that reduces the ability of the arm to position the end-effector.
- For example when a serial manipulator is fully extended it is in what is known as the boundary singularity.

Singularity

- At a singularity the end-effector loses one or more degrees of twist freedom (instantaneously, the end-effector cannot move in these directions).
- Serial robots with less than six independent joints are always singular in the sense that they can never span a six-dimensional twist space. This is often called an architectural singularity. A singularity is usually not an isolated point in the workspace of the robot, but a sub-manifold.

Singularity



The solution is simple enough, just reposition the robot by giving it new waypoints, thus the robot should be able to reach its target.

Denavit-Hartenberg

SECTION 2

Denavit-Hartenberg Method is a short cut to HTM

$$H_n^0 = \begin{bmatrix} & R_n^0 & & d_n^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Denavit-Hartenberg Method

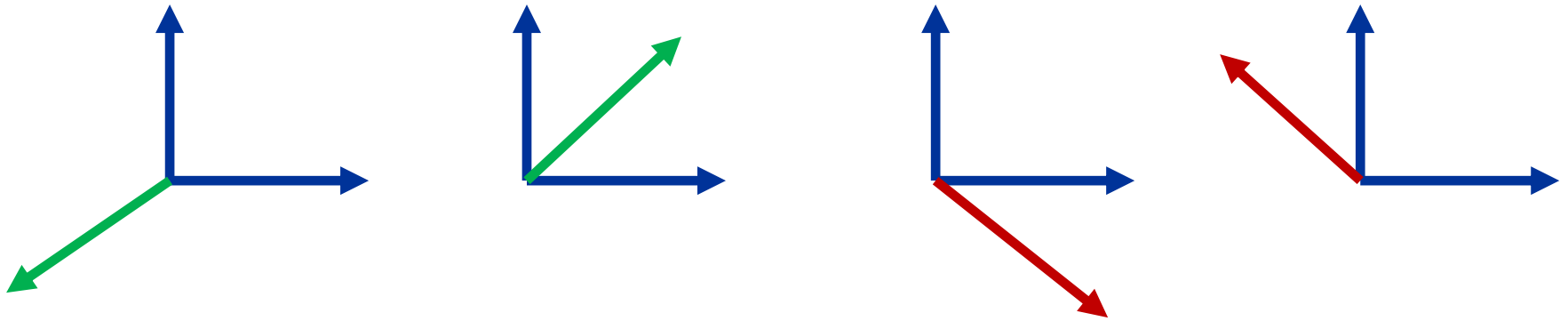
1. Assign frames according to the 4 Denavit-Hartenberg Rules
2. Create the Denavit-Hartenberg parameter table
3. Plug the table values into the Homogeneous Transformation Matrix
4. Multiply the matrices together

Denavit-Hartenberg Frames

SECTION 3

Preliminary Rules

1. You must have at least one **more** frame than there are **joints** – one frame **must** be on the **end-effector**.
2. All axes must be drawn either up, down, right, left, or in the first or third quadrant.



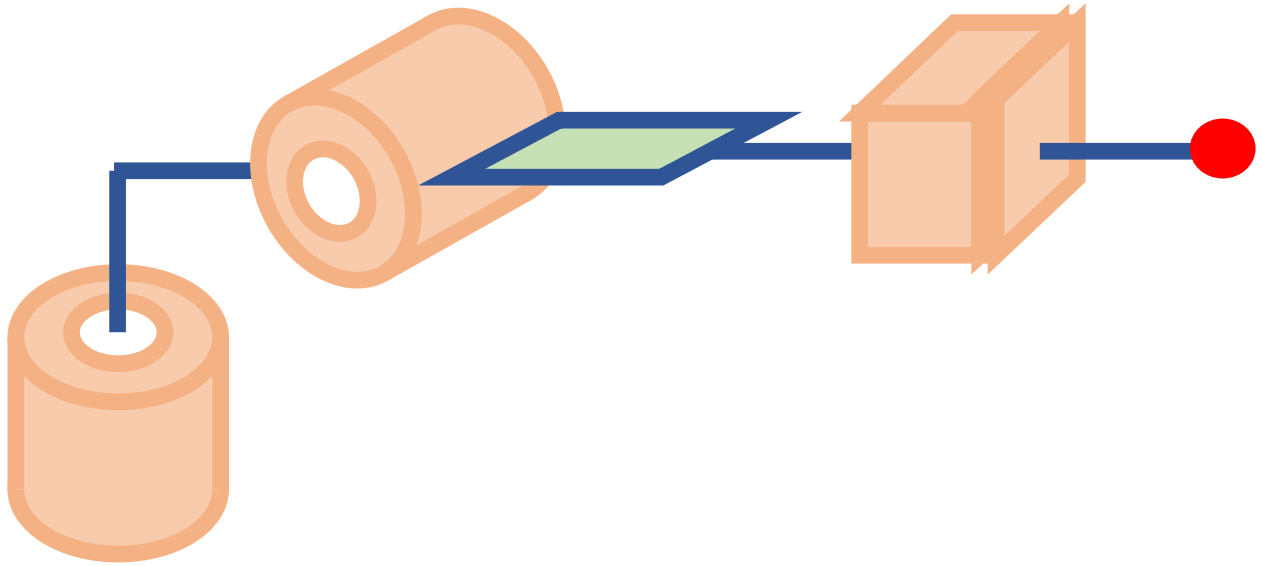
4 Denavit-Hartenberg Frame Rules

1. The **Z** axis must be the axis of revolution or the direction of motion.
2. The **X** axis must be perpendicular to the **Z** axis of the frame before it.
3. The **X** axis must intersect the **Z** axis of the frame before it.
4. The **Y** axis must be drawn so that the whole frame follows the right hand-rule.

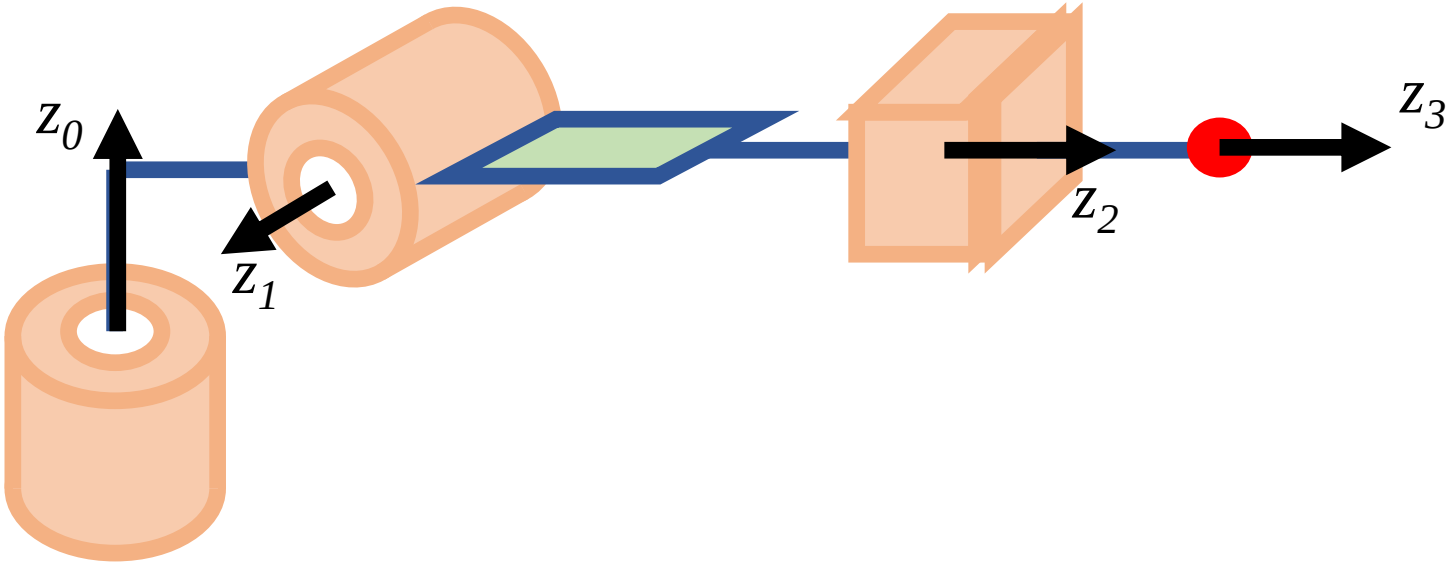
3 Joints

4 Frames at least

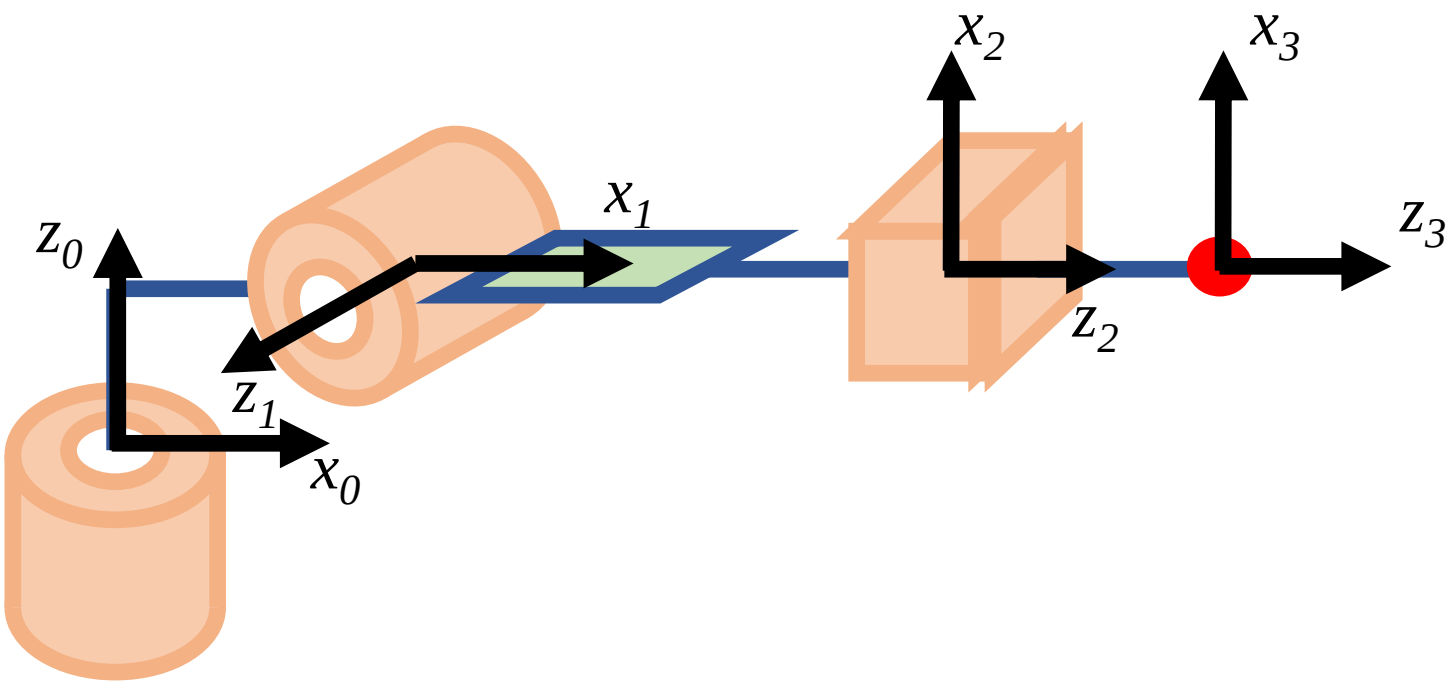
3 DOF



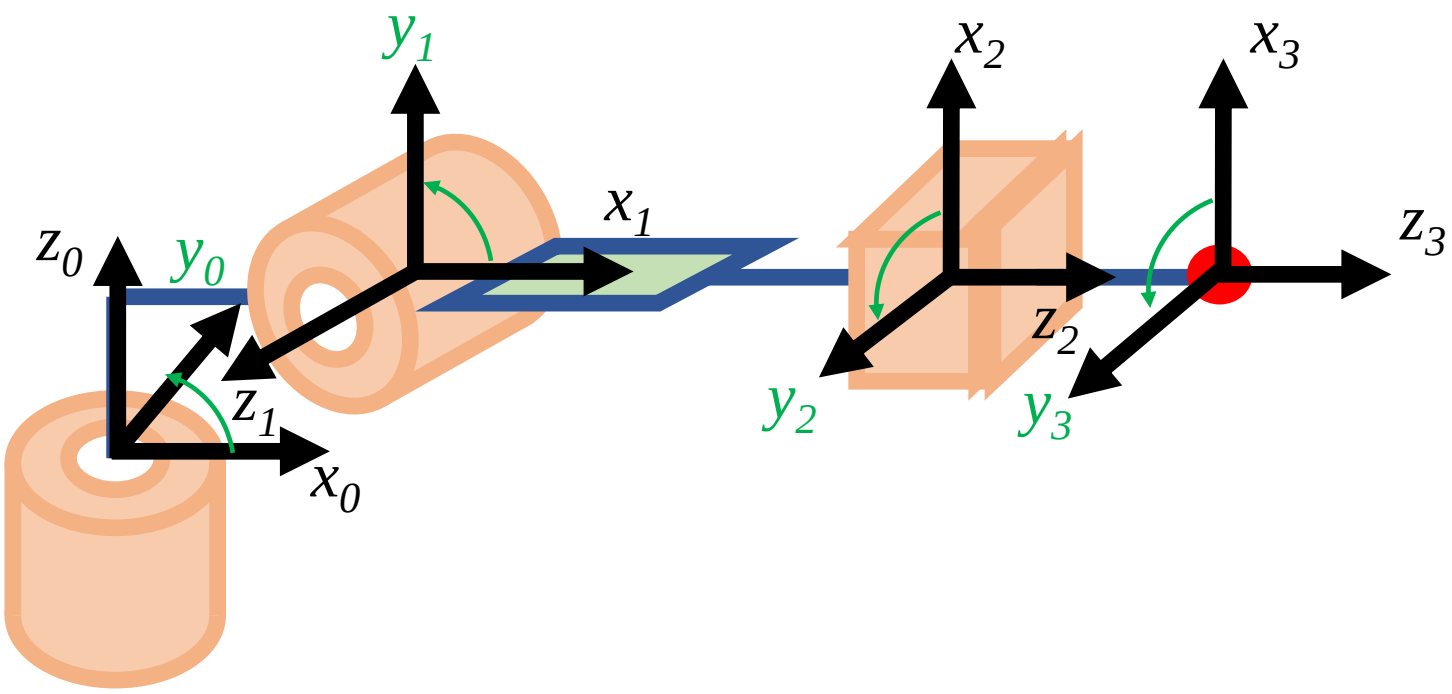
Rule 1: Z axis

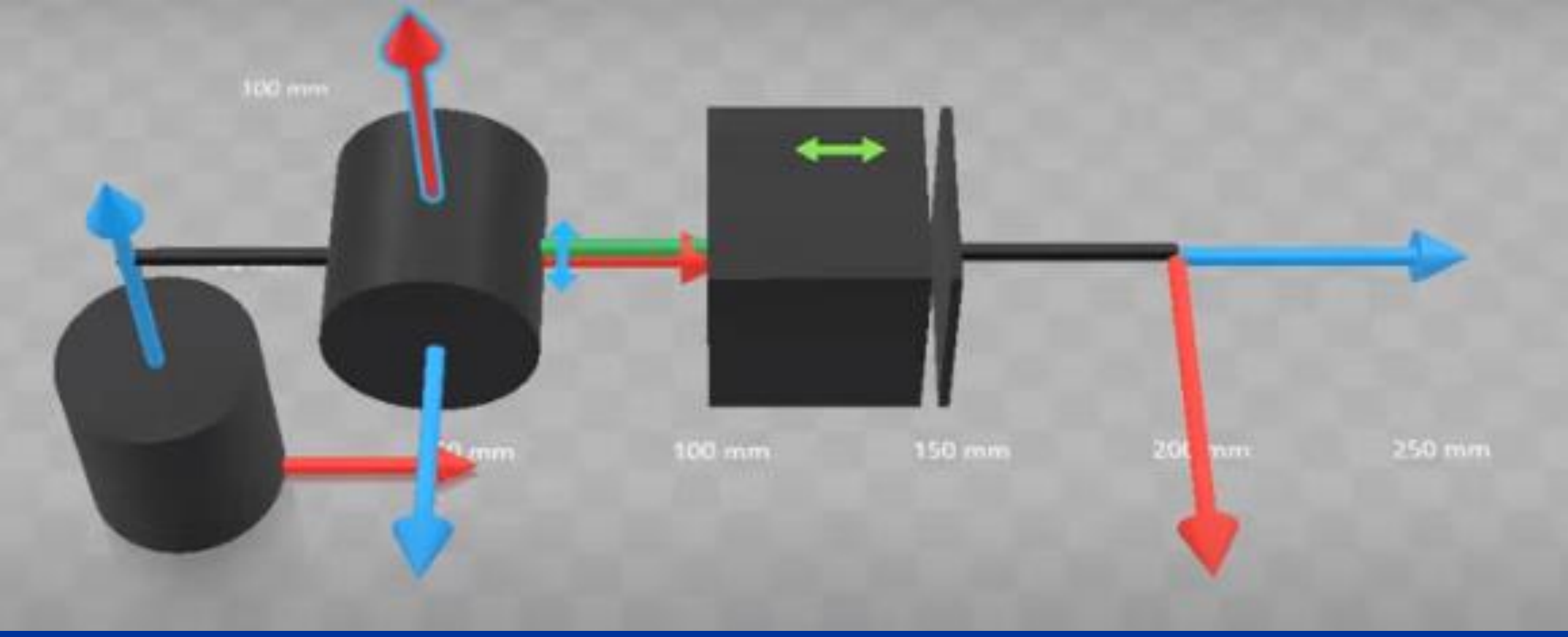


Rule 2/3: **X** axis



Rule 4: **Y** axis





Joints and Axes

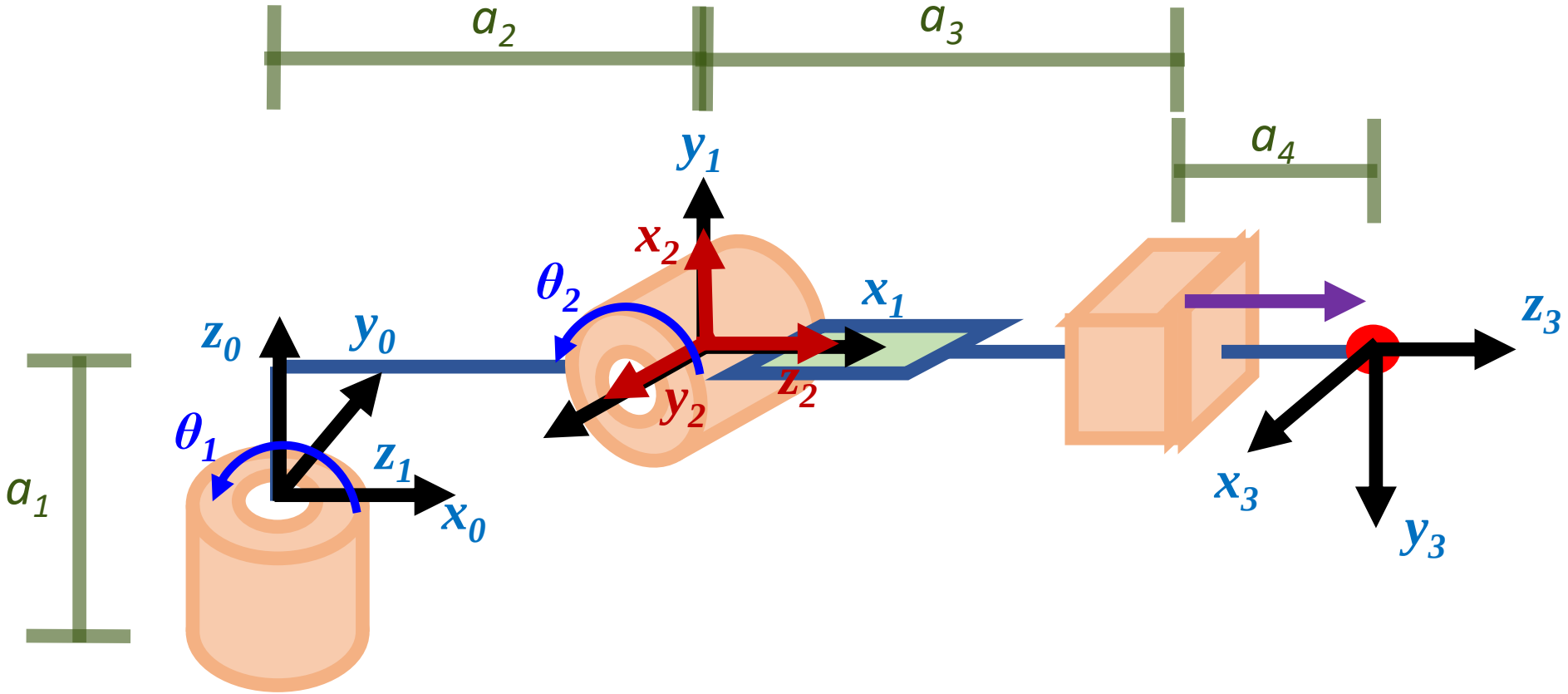
Denavit-Hartenberg Parameter Table

SECTION 4

Denavit-Hartenberg Parameter Table

	Rotation		Displacement	
Parameter	x-revolute angle θ	Z-revolute angle α	frame x-movement **r**	frame z-movement **d**
1				
2				

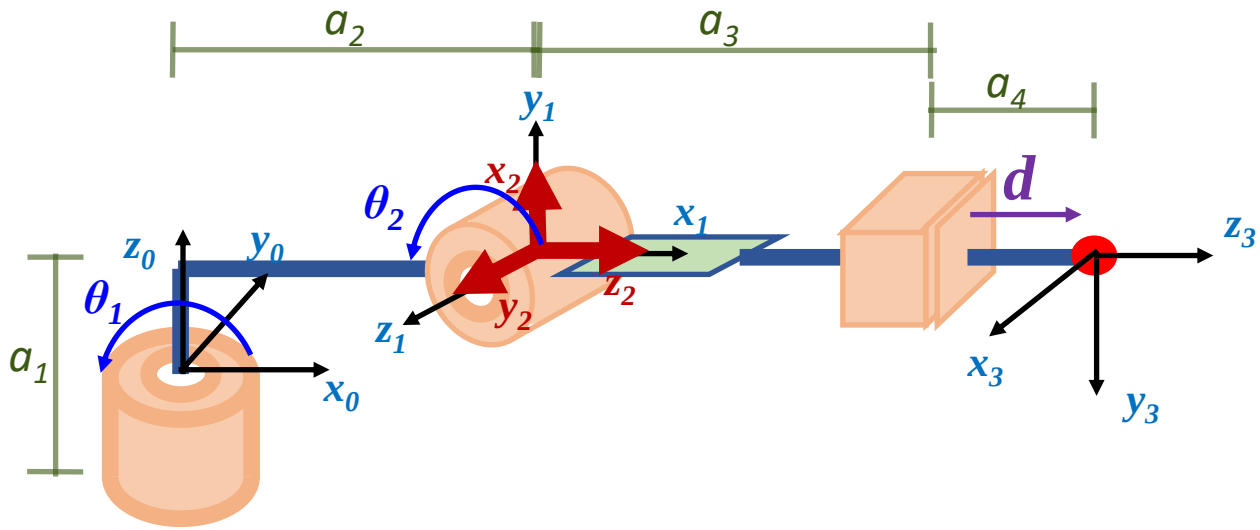
3 frames = 2 rows


a_2
a_3
a_4
a_1
y_1
x_2
θ_2
x_1
z_0
y_0
z_2
y_2
θ_1
z_1
x_0
z_3
x_3
y_3

X-Revolute Angle θ:
Angle of x axes in different frames

- Angle for rotation of X_{n-1} to X_n along Z_{n-1}

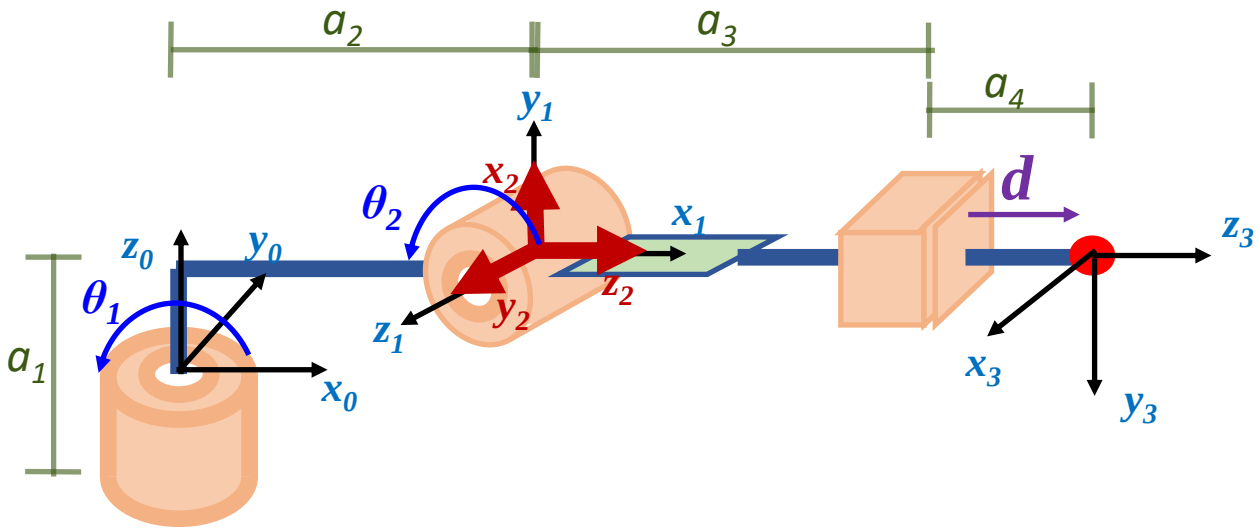
	θ	α	r	d
1	$0+\theta_1$			
2	$90+\theta_2$			
3	90			



Z-Revolute Angle α:
Angle of z axes in different frames

- Angle for rotation of Z_{n-1} to Z_n along X_n

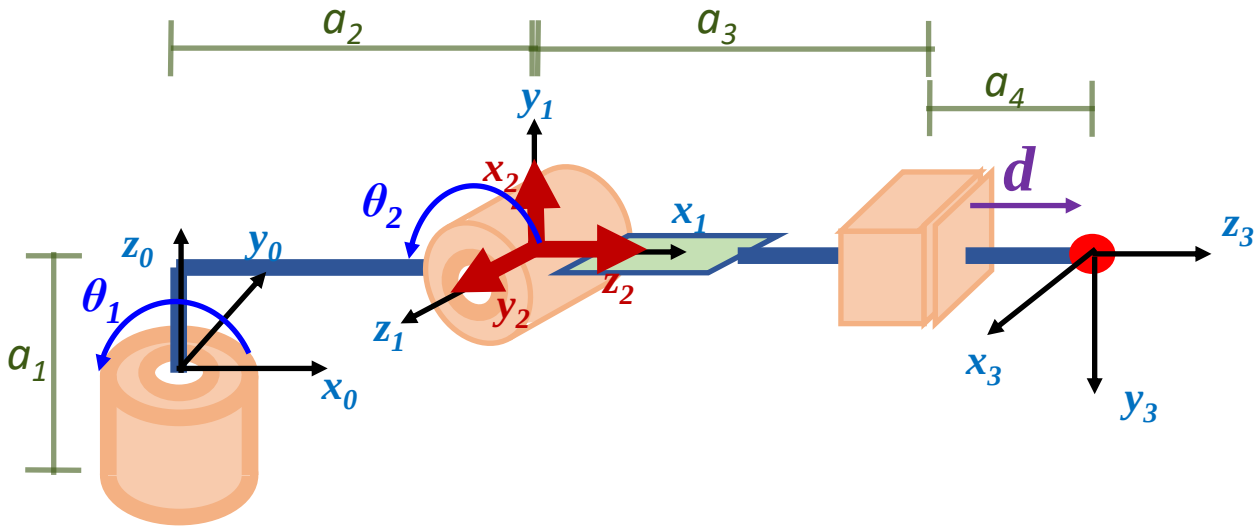
	θ	α	r	d
1	$0+\theta_1$	90		
2	$90+\theta_2$	90		
3	90	0		



X-Movement Displacement **r:** ***Displacement of frames in x axis***

- Displacement of frame f_{n-1} to frame f_n only on X_n

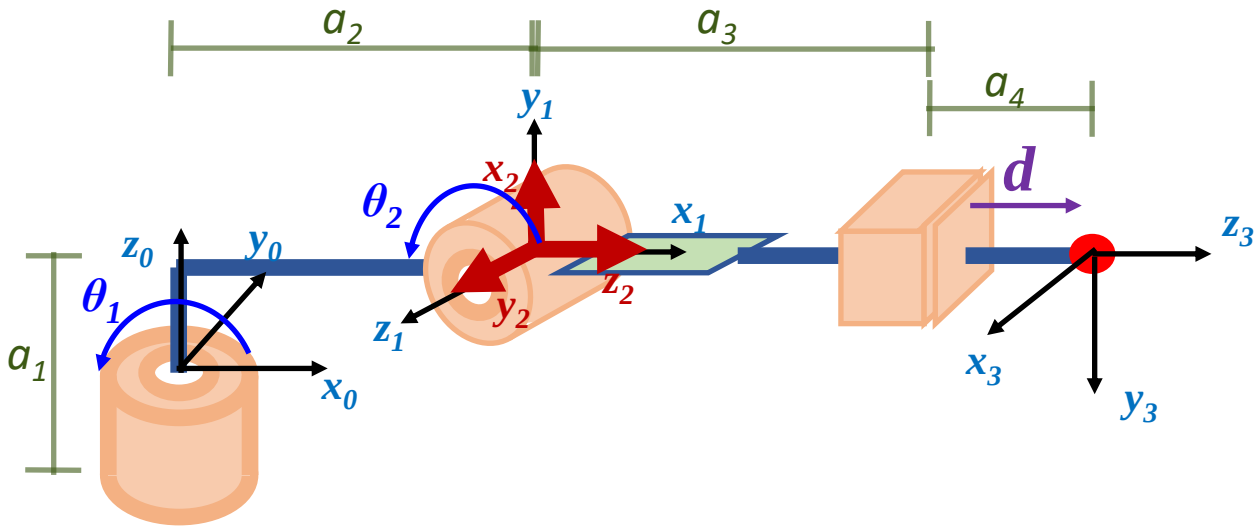
	θ	α	r	d
1	$0+\theta_1$	90	a_2	
2	$90+\theta_2$	90	0	
3	90	0	0	



Z-Movement Displacement **d:** ***Displacement of frames in z axis***

- Displacement of frame f_{n-1} to frame f_n only on Z_{n-1}

	θ	α	r	d
1	$0+\theta_1$	90	a_2	a_1
2	$90+\theta_2$	90	0	0
3	90	0	0	a_3+a_4 +d



DH HTM Matrix

SECTION 5

Denavit –Hartenberg Homogeneous Transformation Matrix

$$H_n^{n-1} = \begin{bmatrix} C(\theta_n) & -S(\theta_n)C(\alpha_n) & S(\theta_n)S(\alpha_n) & r_n C(\theta_n) \\ S(\theta_n) & C(\theta_n)C(\alpha_n) & -C(\theta_n)S(\alpha_n) & r_n S(\theta_n) \\ 0 & S(\alpha_n) & C(\alpha_n) & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

DH HTM Table Creation

	θ	α	r	d
1	$0+\theta_1$	90	a_2	a_1
2	$90+\theta_2$	90	0	0
3	90	0	0	a_3+a_4 +d

$$H_1^0 = \begin{bmatrix} C(\theta_1) & -S(\theta_1)C(90) & S(\theta_1)S(90) & a_2C(\theta_1) \\ S(\theta_1) & C(\theta_1)C(90) & -C(\theta_1)S(90) & a_2S(\theta_1) \\ 0 & S(90) & C(90) & a_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_2^1 = \begin{bmatrix} C(90+\theta_2) & -S(90+\theta_2)C(90) & S(90+\theta_2)S(90) & 0 \\ S(90+\theta_2) & C(90+\theta_2)C(90) & -C(90+\theta_2)S(90) & 0 \\ 0 & S(90) & C(90) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H_3^2 = \begin{bmatrix} C(90) & -S(90)C(0) & S(90)S(0) & 0 \\ S(90) & C(90)C(0) & -C(90)S(0) & 0 \\ 0 & S(0) & C(0) & d+a_3+a_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Python Coding

SECTION 6

DH_HTM(T, af, r, d)

```
D[0][0] = np.cos(T)
D[0][1] = -np.sin(T) * np.cos(af)
D[0][2] = np.sin(T) * np.sin(af)
D[0][3] = r * np.cos(T)

D[1][0] = np.sin(T)
D[1][1] = np.cos(T) * np.cos(af)
D[1][2] = - np.cos(T) * np.sin(af)
D[1][3] = r * np.sin(T)

D[2][0] = 0
D[2][1] = np.sin(af)
D[2][2] = np.cos(af)
D[2][3] = d

D[3][0] = 0
D[3][1] = 0
D[3][2] = 0
D[3][3] = 1
```

Parameters:

T: Theta

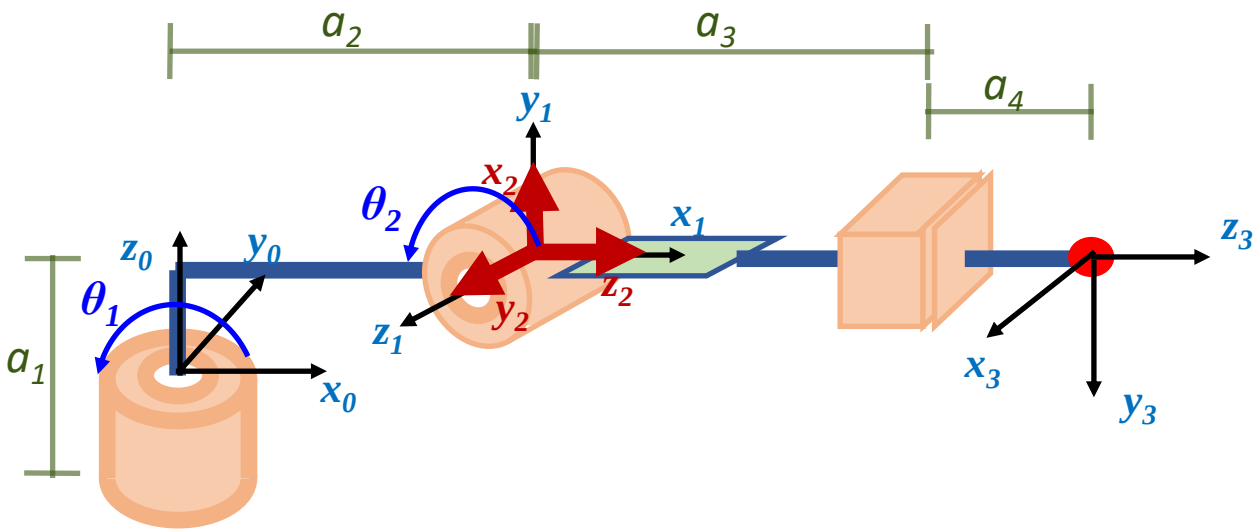
af: alpha

r: r

d: d

Calculation

```
Theta1 = deg90
Theta2 = 0
a1 = 1
a2 = 1
a3 = 1
a4 = 1
d  = 0
```



H0_1:
[[0. -0. 1. 0.]
 [1. 0. -0. 1.]
 [0. 1. 0. 1.]
 [0. 0. 0. 1.]]

H1_2:
[[0. -0. 1. 0.]
 [1. 0. -0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 0. 1.]]

H2_3:
[[0. -1. 0. 0.]
 [1. 0. -0. 0.]
 [0. 0. 1. 2.]
 [0. 0. 0. 1.]]

H0_3: T1=90, T2=0, arm=3
[[1. 0. 0. 0.]
 [0. 0. 1. 3.]
 [0. -1. 0. 1.]
 [0. 0. 0. 1.]]

Input: Point (0, 0, 0) with respect to frame 3

w:
[[0.]
 [0.]
 [0.]
 [1.]]

Each Frame (0, 0, 0)

W0_1:
[[0.]
 [1.]
 [1.]
 [1.]]

W0_2:
[[0.]
 [1.]
 [1.]
 [1.]]

W0_n:
[[0.]
 [3.]
 [1.]
 [1.]]