CS65K Robotics

Modelling, Planning and Control

Chapter 2: Kinematics

Section 2.8-2.11

LECTURE 5: DANEVIT-HARTENBERG TABLE AND MATRIX

DR. ERIC CHOU

IEEE SENIOR MEMBER

Objectives

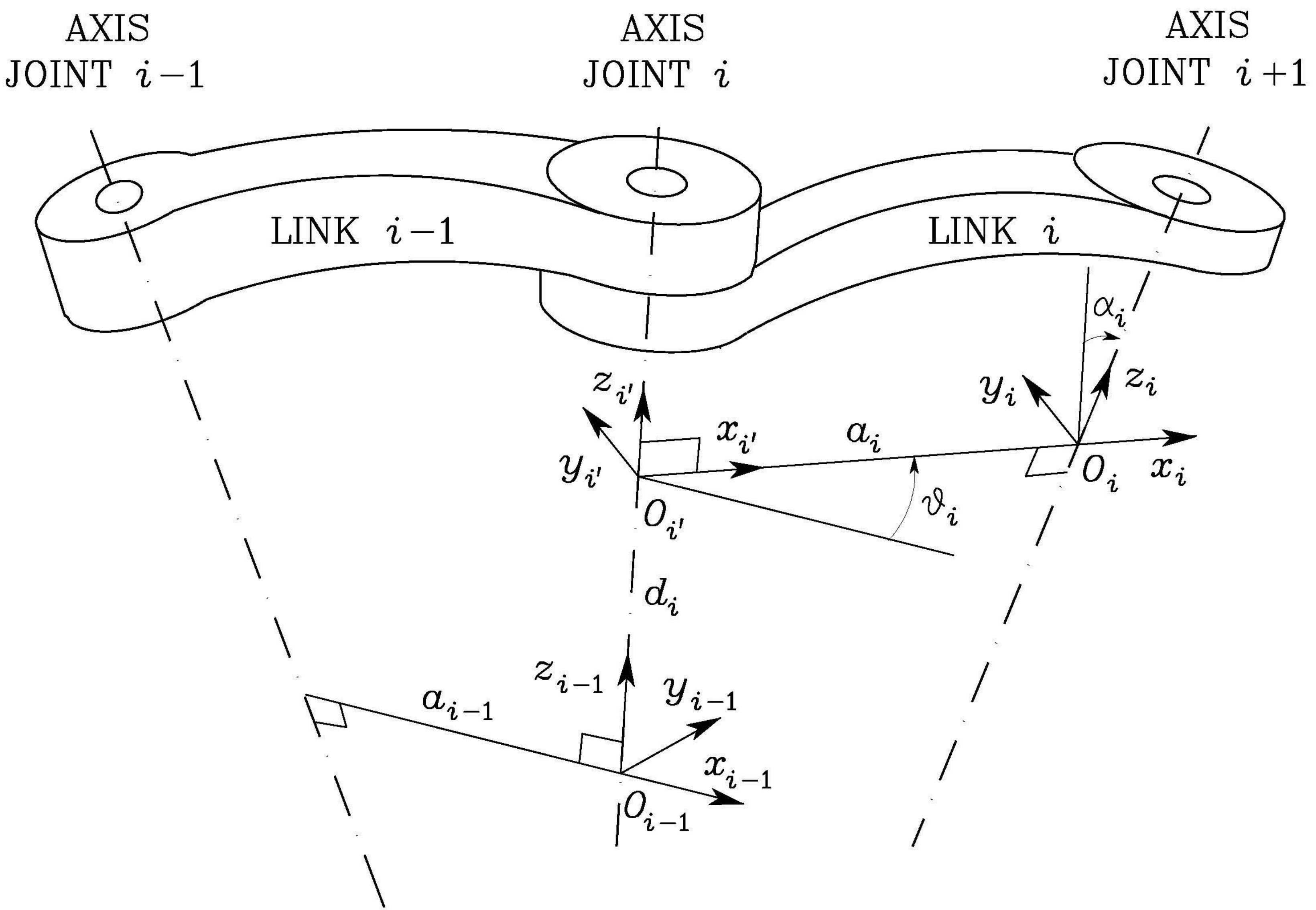
- Denavit-Hartenberg parameters are introduced
- A formula is derived to compute the transformation matrix from one link to the next one in a kinematic chain
- A computationally recursive operating procedure is illustrated
- The direct kinematics equation is computed for a number of typical manipulator structures
- Composition of the kinematics of the arm with the kinematics of the wrist is presented
- The joint space and operational space concepts are illustrated

Denavit-Hartenberg Convention

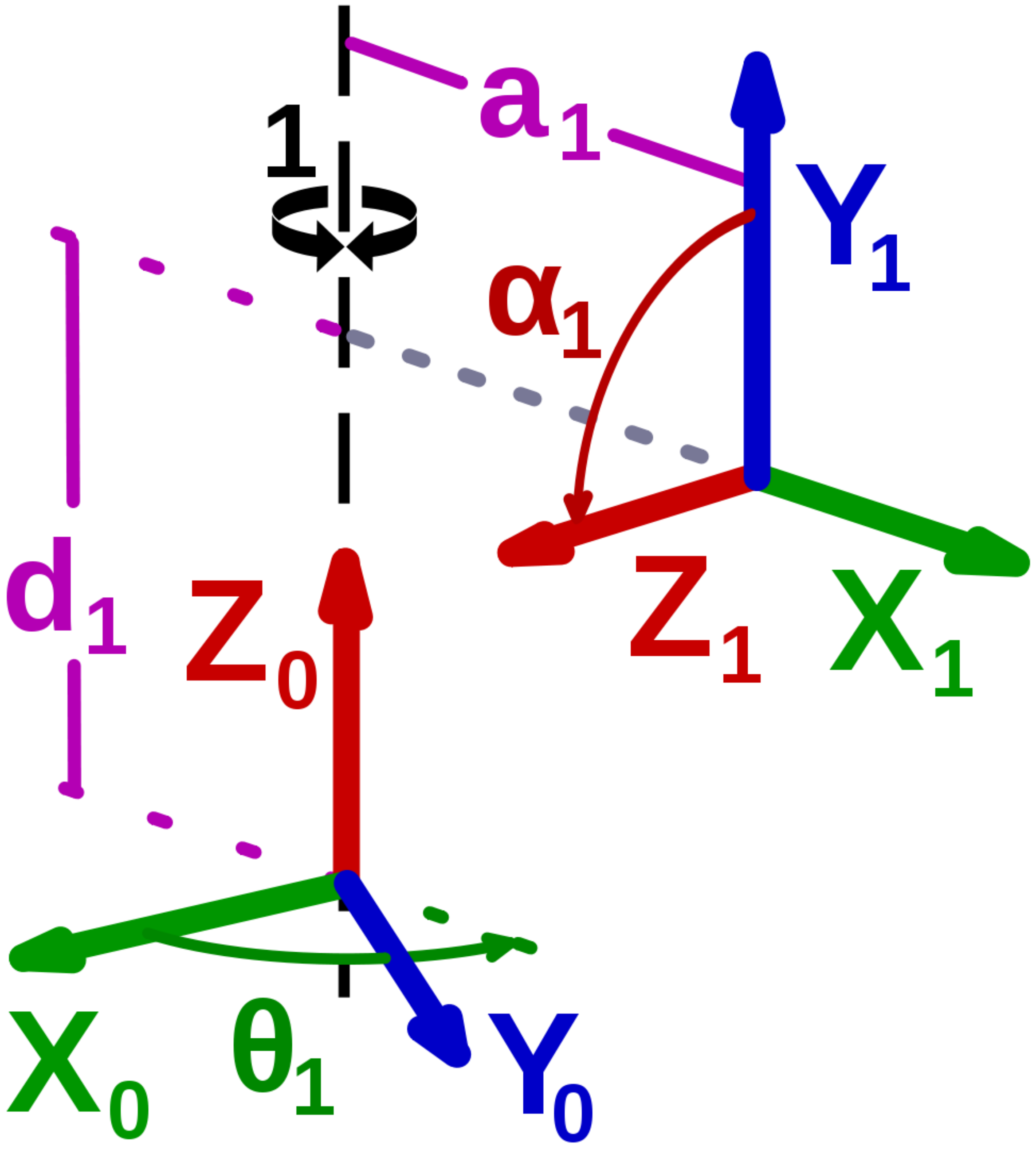
SECTION 1

Denavit-Hartenberg Convention

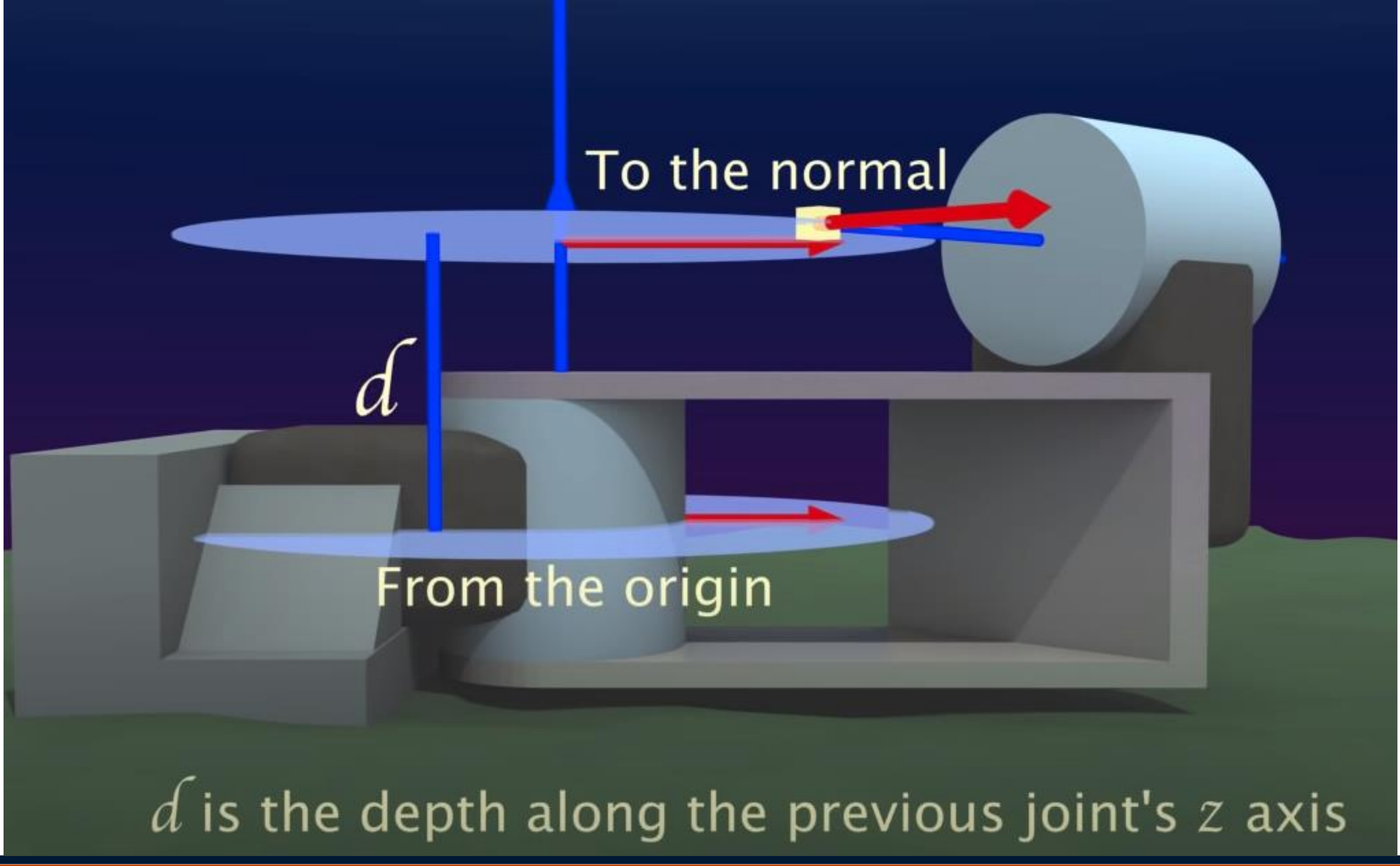
- Choose axis z_i along the axis of Joint $i+1$
- Locate the origin O_i at the intersection of axis z_i with the common normal to axes z_{i-1} and z_i. Also, locate $O_{i'}$ at the intersection of the common normal with axis z_{i-1}
- Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from Joint i to Joint $i+1$
- Choose axis y_i so as to complete a right-handed frame

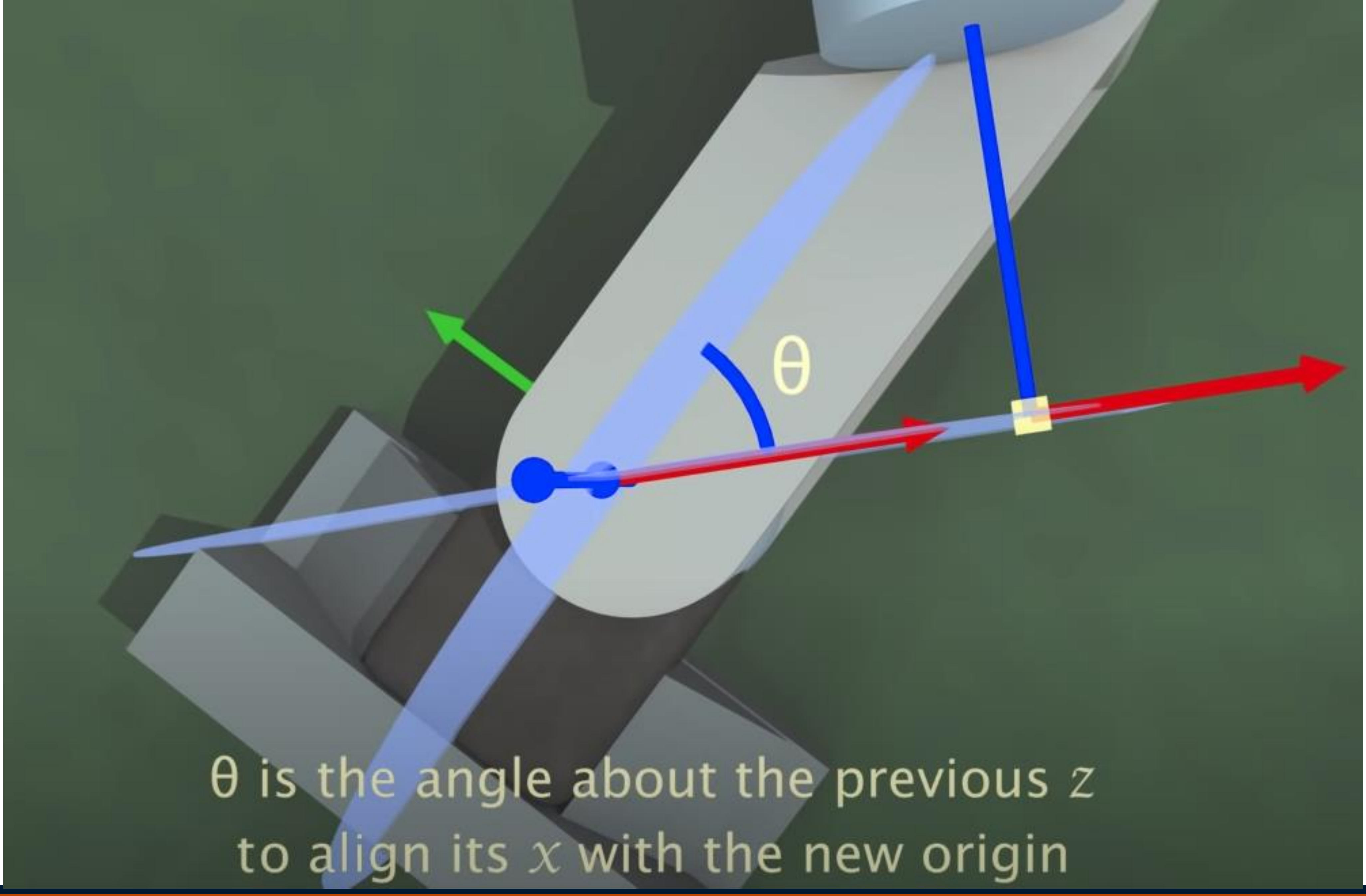


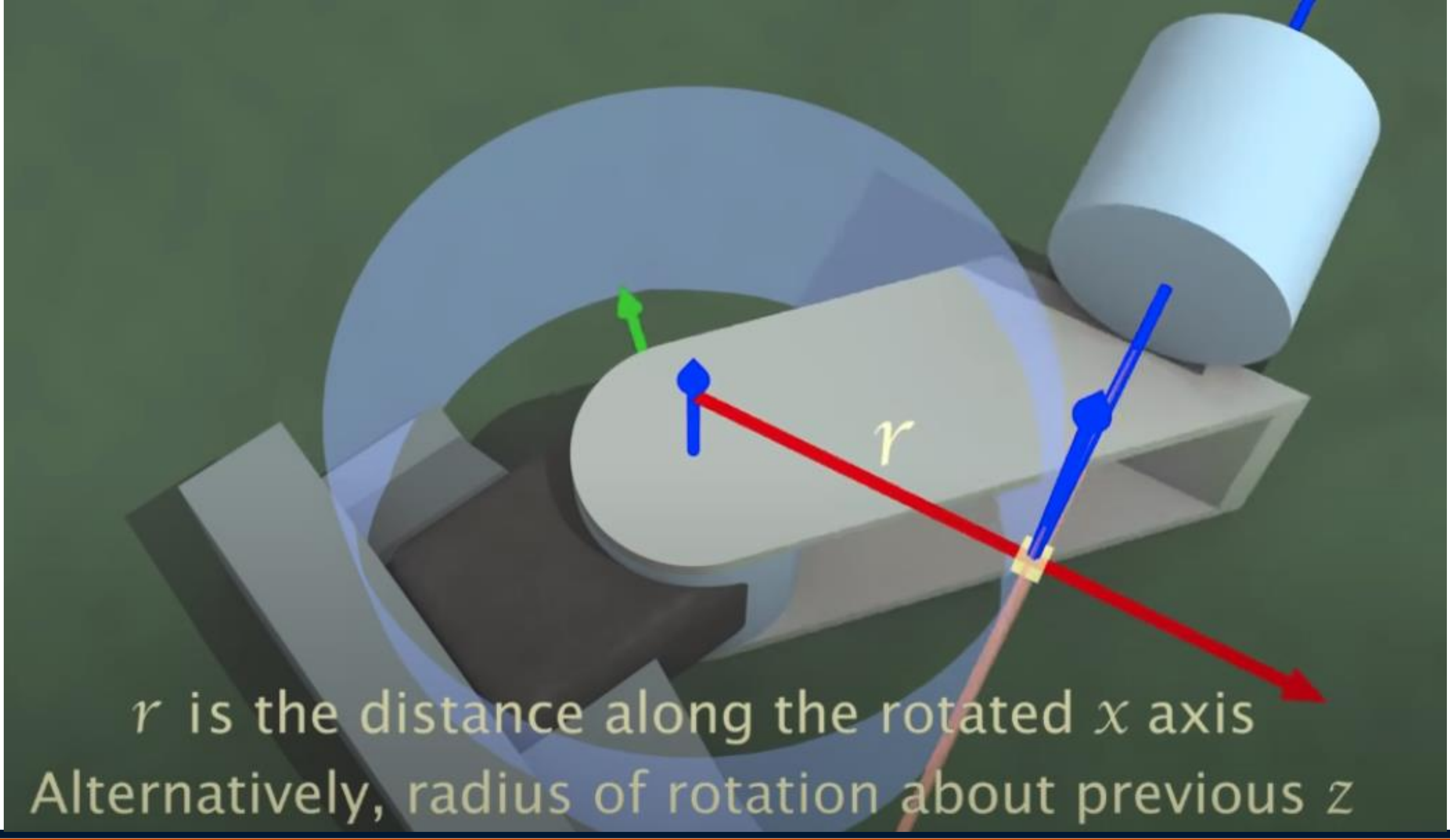
Denavit-Hartenberg kinematic parameters

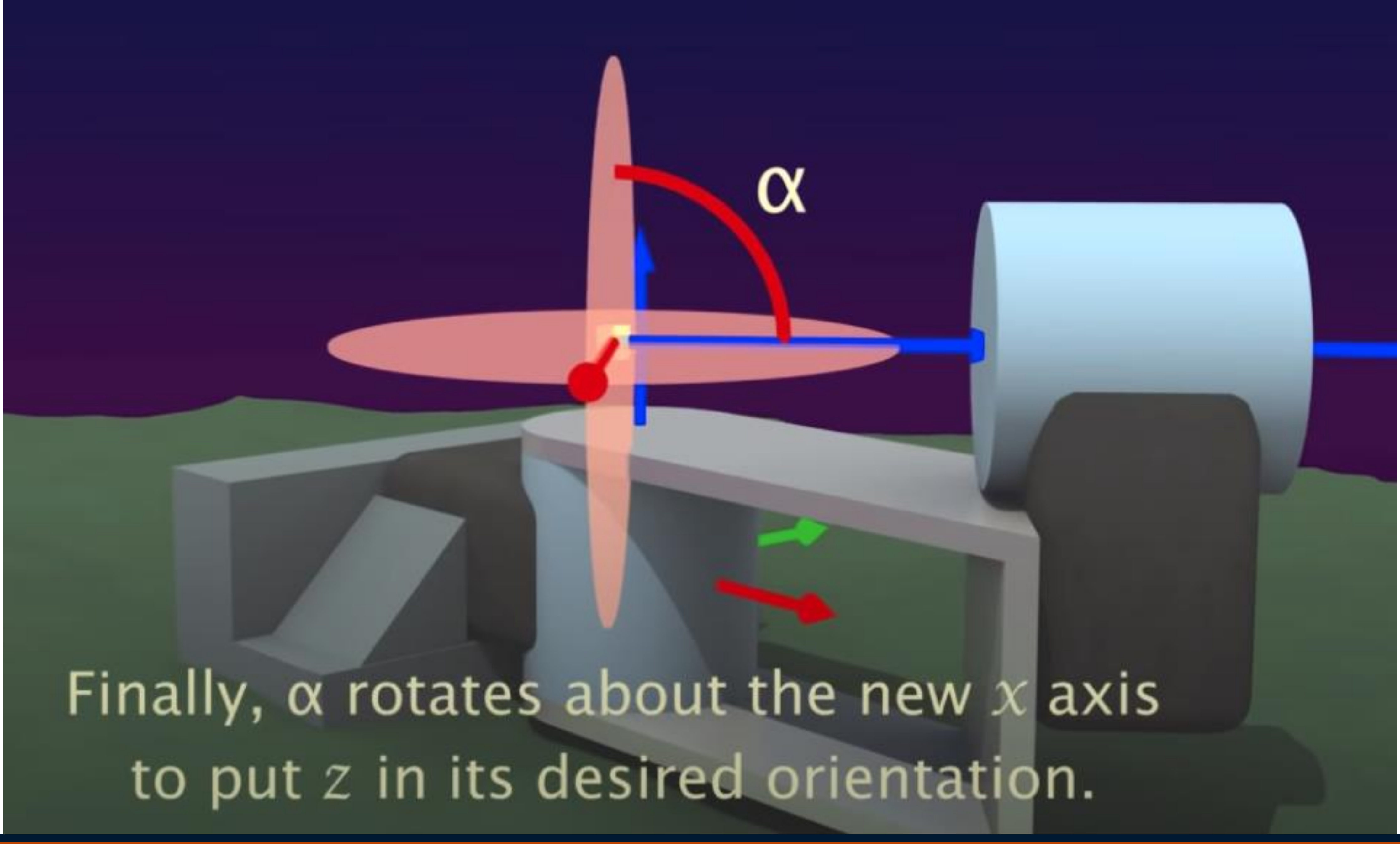


Denavit-Hartenberg Parameters


To the normal
d
From the origin
d is the depth along the previous joint's z axis


θ
θ is the angle about the previous z
to align its x with the new origin


r
r is the distance along the rotated x axis
Alternatively, radius of rotation about previous z


α
Finally, α rotates about the new x axis
to put z in its desired orientation.

Denavit-Hartenberg Convention II

Nonunique definition of the link frame in the following cases

- For Frame 0, only the direction of axis z_0 is specified; then O_0 and x_0 can be arbitrarily chosen
- For Frame n, since there is no Joint $n+1$, z_n is not uniquely defined while x_n has to be normal to axis z_{n-1}. Typically, Joint n is revolute, and thus z_n is to be aligned with the direction of z_{n-1}
- When two consecutive axes are parallel, the common normal between them is not uniquely defined
- When two consecutive axes intersect, the direction of x_i is arbitrary
- When Joint i is prismatic, the direction of z_{i-1} is arbitrary

Denavit-Hartenberg Parameters

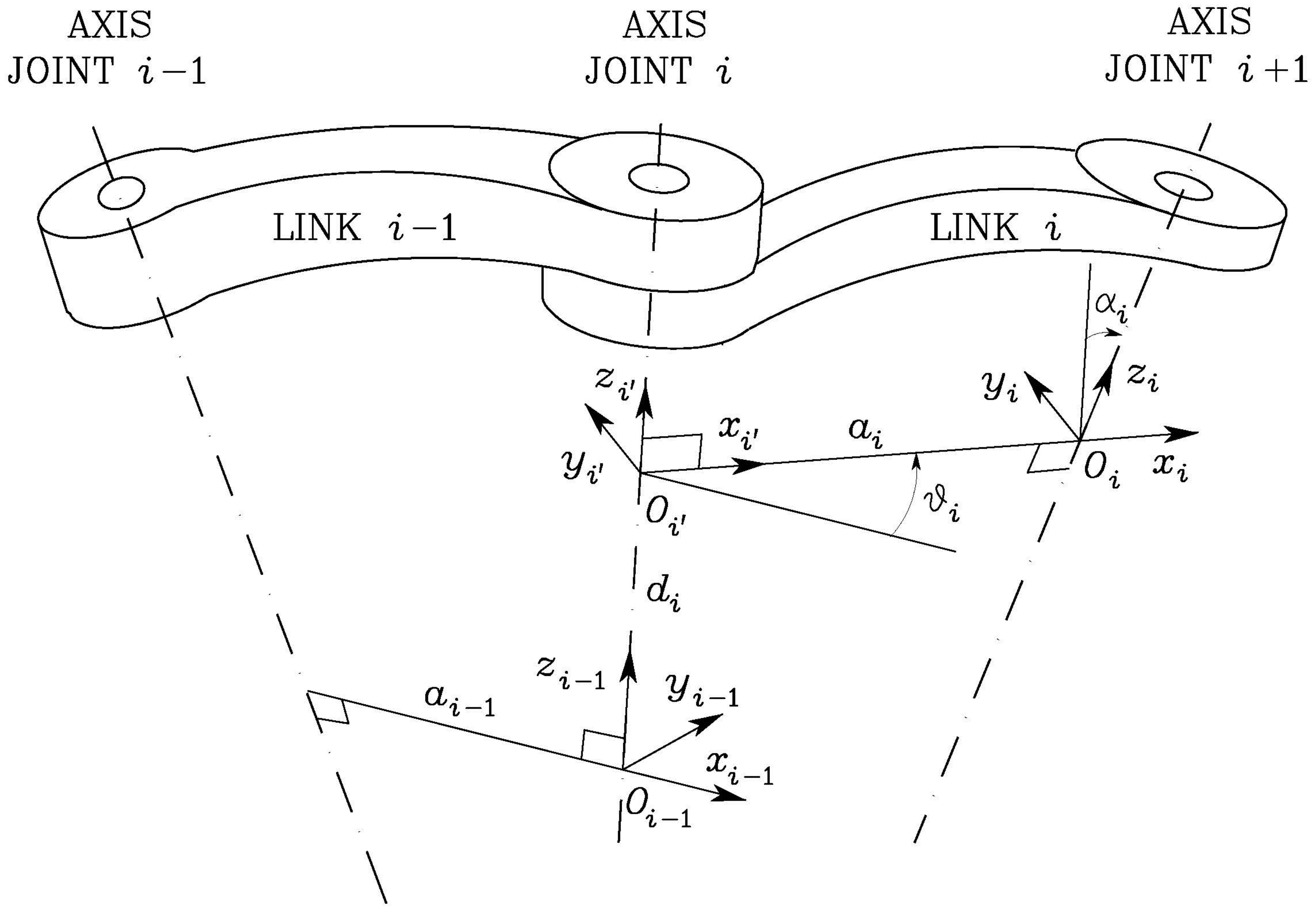
a_i: distance between O_i and $O_{i'}$

d_i: coordinate of $O_{i'}$ along z_{i-1}

α_i: angle between axes z_{i-1} and z_i about axis x_i to be taken positive when rotation is made counter-clockwise

ϑ_i: angle between axes x_{i-1} and x_i about axis z_{i-1} to be taken positive when rotation is made counter-clockwise

- a_i and α_i are always constant
- If Joint i is *revolute* the variable is ϑ_i
- If Joint i is *prismatic* the variable is d_i



Denavit-Hartenberg parameters

Coordinate Transformation

- Transformation from Frame *i - 1* to Frame *i'*

$$\boldsymbol{A}_{i'}^{i-1} = \begin{bmatrix} \cos\vartheta_i & -\sin\vartheta_i & 0 & 0 \\ \sin\vartheta_i & \cos\vartheta_i & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Transformation from Frame *i'* to Frame *i - 1*

$$\boldsymbol{A}_{i}^{i'} = \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & \cos\alpha_i & -\sin\alpha_i & 0 \\ 0 & \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Coordinate Transformation II

- The resulting coordinate transformation is obtained by post-multiplication of the single transformations

$$\boldsymbol{A}_i^{i-1}(q_i) = \boldsymbol{A}_{i'}^{i-1}\boldsymbol{A}_i^{i'} = \begin{bmatrix} c_{\vartheta_i} & -s_{\vartheta_i}c\alpha_i & s_{\vartheta_i}s_{\alpha_i} & a_i c_{\vartheta_i} \\ s_{\vartheta_i} & c_{\vartheta_i}c_{\alpha_i} & -c_{\vartheta_i}s_{\alpha_i} & a_i s_{\vartheta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Operating Procedure

1. Find and number consecutively the joint axes; set the directions of axes $z_0, \dots, z_n$

2. Choose Frame 0 by locating the origin on axis z_0; axes x_0 and y_0 are chosen so as to obtain aright-handed frame. If feasible, it is worth choosing Frame 0 to coincide with the base frame

Execute steps from 3. to 5. for $i = 1, \dots, n-1$

3. Locate the origin O_i at the intersection of z_i with the common normal to axes z_{i-1} and z_i. If axes z_{i-1} and z_i are parallel and Joint i is revolute, then locate O_i so that $d_i = 0$; if Joint i is prismatic, locate O_i at a reference position for the joint range (mechanical limit)

4. Choose axis x_i along the common normal to axes z_{i-1} and z_i with direction from Joint i to Joint $i+1$

5. Choose axis y_i so as to obtain a right-handed frame

Operating Procedure II

6. Choose Frame n; if Joint n is revolute, then align z_n with z_{n-1}, otherwise, if Joint n is prismatic, then choose z_n arbitrarily. Axis x_n is set according to step 4.

7. For $i = 1, \dots, n$, form the table of parameters $a_i, d_i, \alpha_i, \vartheta_i$

8. On the basis of the parameters in 7. compute the homogeneous transformation matrices $\boldsymbol{A}_i^{i-1}(q_i)$ for $i = 1, \dots, n$

9. Compute the homogeneous transformation $\boldsymbol{T}_n^0(\boldsymbol{q}) = \boldsymbol{A}_1^0 \dots \boldsymbol{A}_n^{n-1}$ that yields the position and orientation of Frame n with respect to Frame 0

10. Given $\boldsymbol{T}_0^b$ and $\boldsymbol{T}_e^n$, compute the direct kinematics function as $\boldsymbol{T}_e^b(\boldsymbol{q}) = \boldsymbol{T}_0^b \boldsymbol{T}_n^0(\boldsymbol{q}) \boldsymbol{T}_e^n$ that yields the position and orientation of the end-effector frame with respect to the base frame.

Kinematics of Typical Manipulator Structures

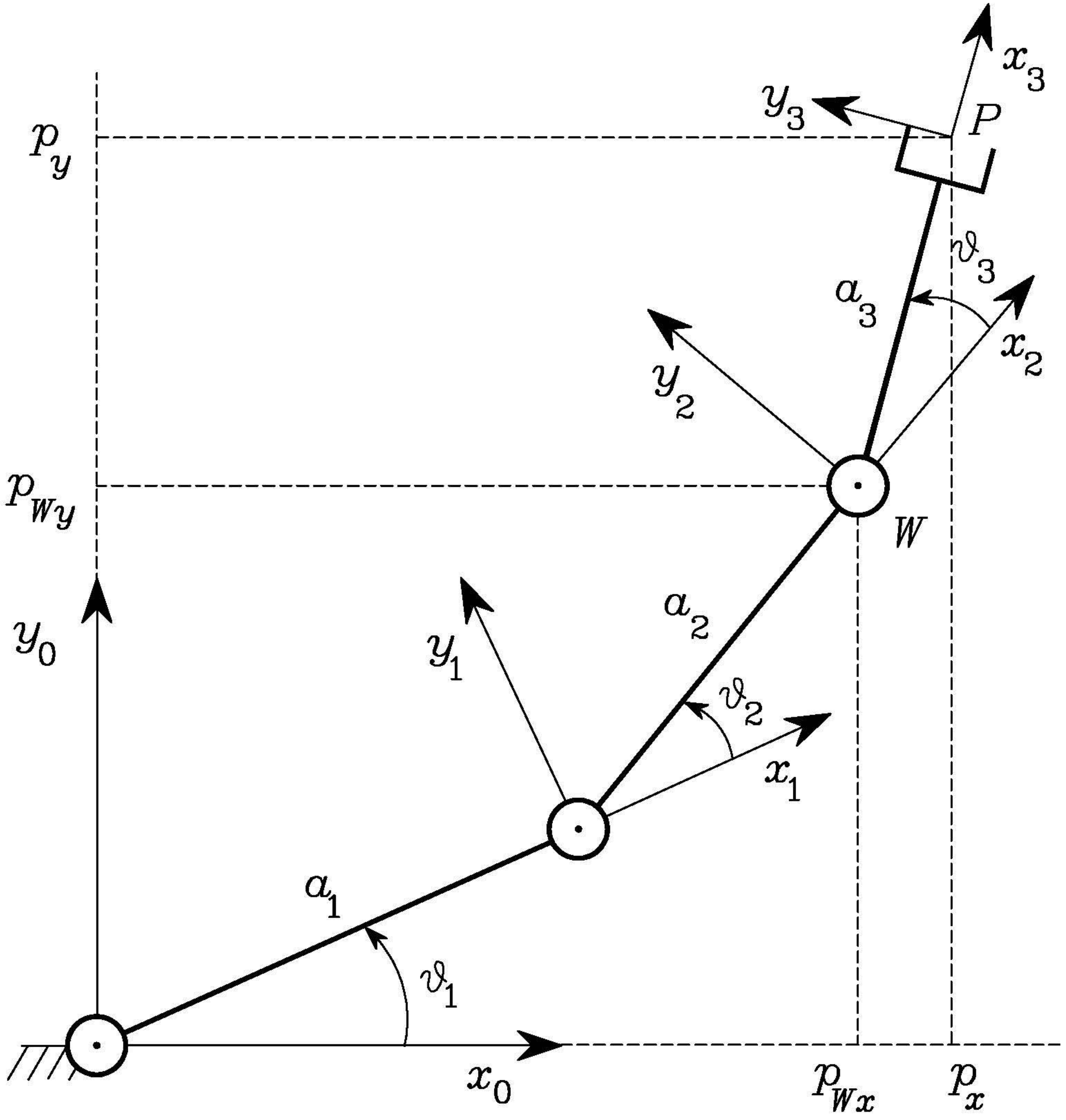
SECTION 2

Three-link Planar Arm

DH parameters

Link	a_i	α_i	d_i	ϑ_i
1	a_1	0	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

$$\boldsymbol{A}_i^{i-1}(\vartheta_i) = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Three-link planar arm with frame assignment

Three-link Planar Arm II

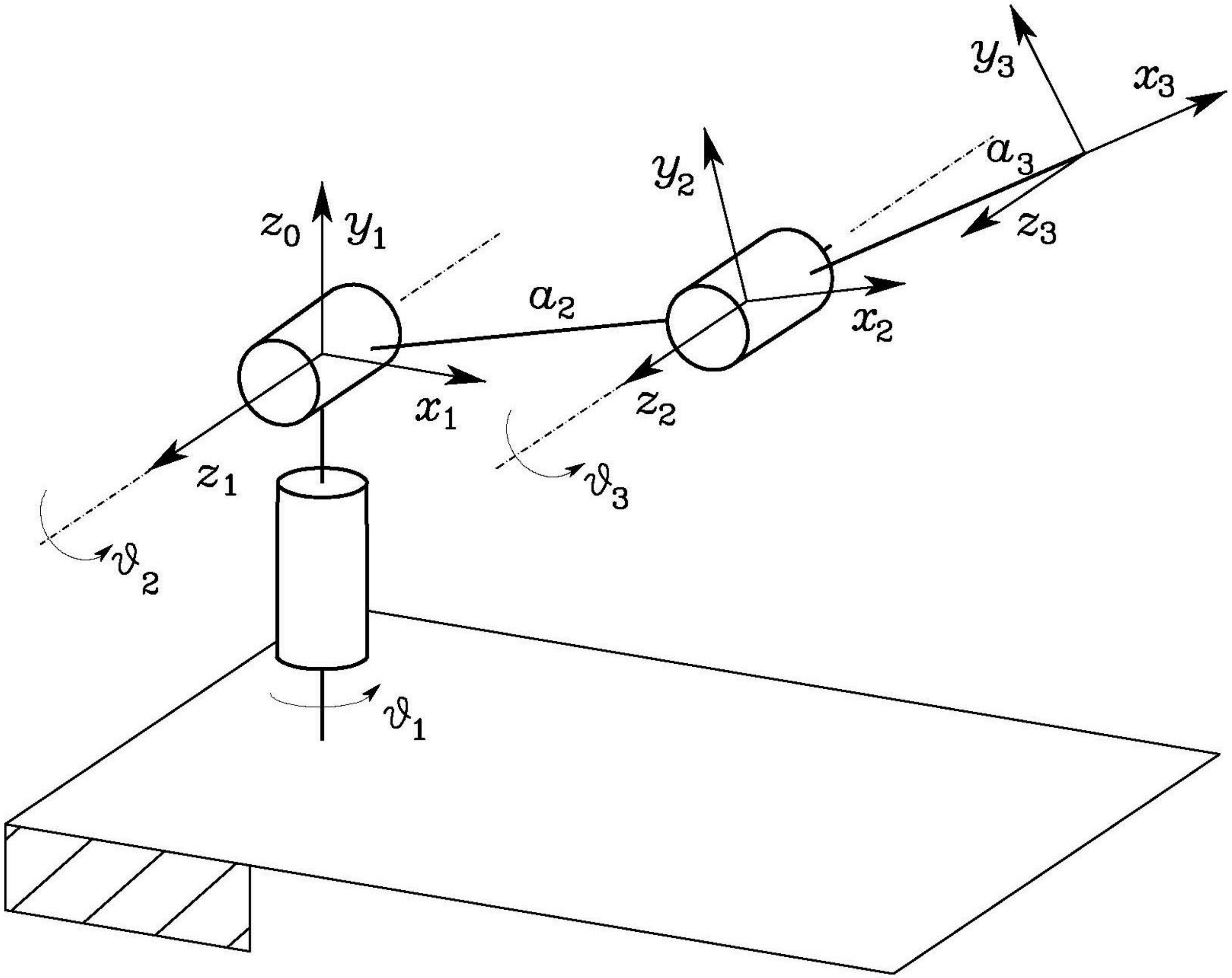
$$T_3^0(q) = A_1^0 A_2^1 A_3^2 = \begin{bmatrix} c_{123} & -s_{123} & 0 & a_1 c_1 + a_2 c_{12} + a_3 c_{123} \\ s_{123} & c_{123} & 0 & a_1 s_1 + a_2 s_{12} + a_3 s_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Anthropomorphic Arm

DH parameters

Link	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	a_3	0	0	ϑ_3

$$\boldsymbol{A}_1^0(\vartheta_1) = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{A}_i^{i-1}(\vartheta_i) = \begin{bmatrix} c_i & -s_i & 0 & a_i c_i \\ s_i & c_i & 0 & a_i s_i \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad i = 2, 3$$



Anthropomorphic arm with frame assignment

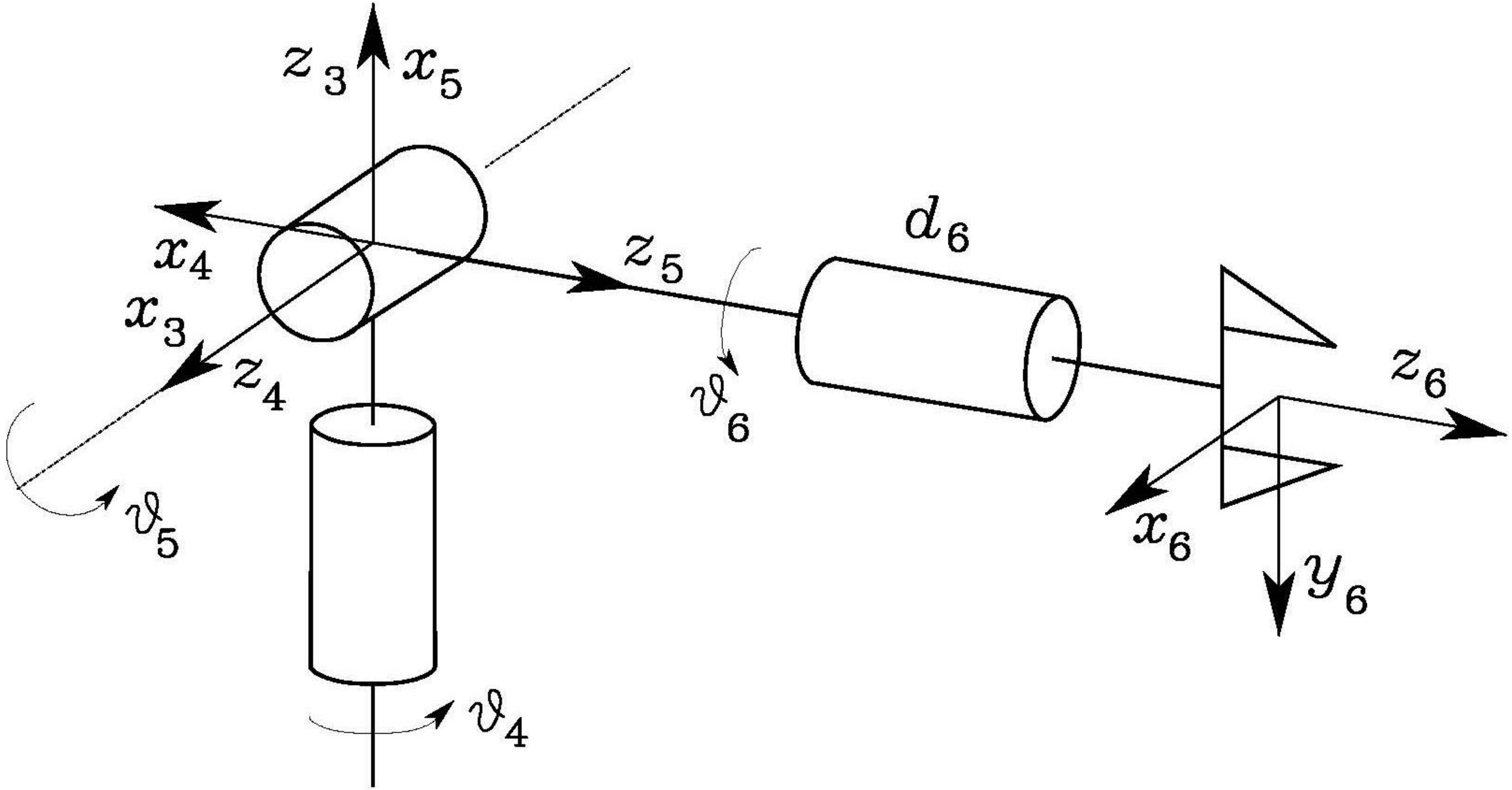
Anthropomorphic Arm II

$$\boldsymbol{T}_3^0(\boldsymbol{q}) = \boldsymbol{T}_1^0 \boldsymbol{T}_2^1 \boldsymbol{T}_3^2 = \begin{bmatrix} c_1 c_{23} & -c_1 s_{23} & s_1 & c_1(a_2 c_2 + a_3 c_{23}) \\ s_1 c_{23} & -s_1 s_{23} & -c_1 & s_1(a_2 c_2 + a_3 c_{23}) \\ s_{23} & c_{23} & 0 & a_2 s_2 + a_3 s_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Spherical Wrist

DH parameters

Link	a_i	α_i	d_i	ϑ_i
4	0	$-\pi/2$	0	ϑ_4
5	0	$\pi/2$	0	ϑ_5
6	0	0	d_6	ϑ_6



Spherical wrist with frame assignment

Spherical Wrist II

$$A_4^3(\vartheta_4) = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad A_5^4(\vartheta_5) = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

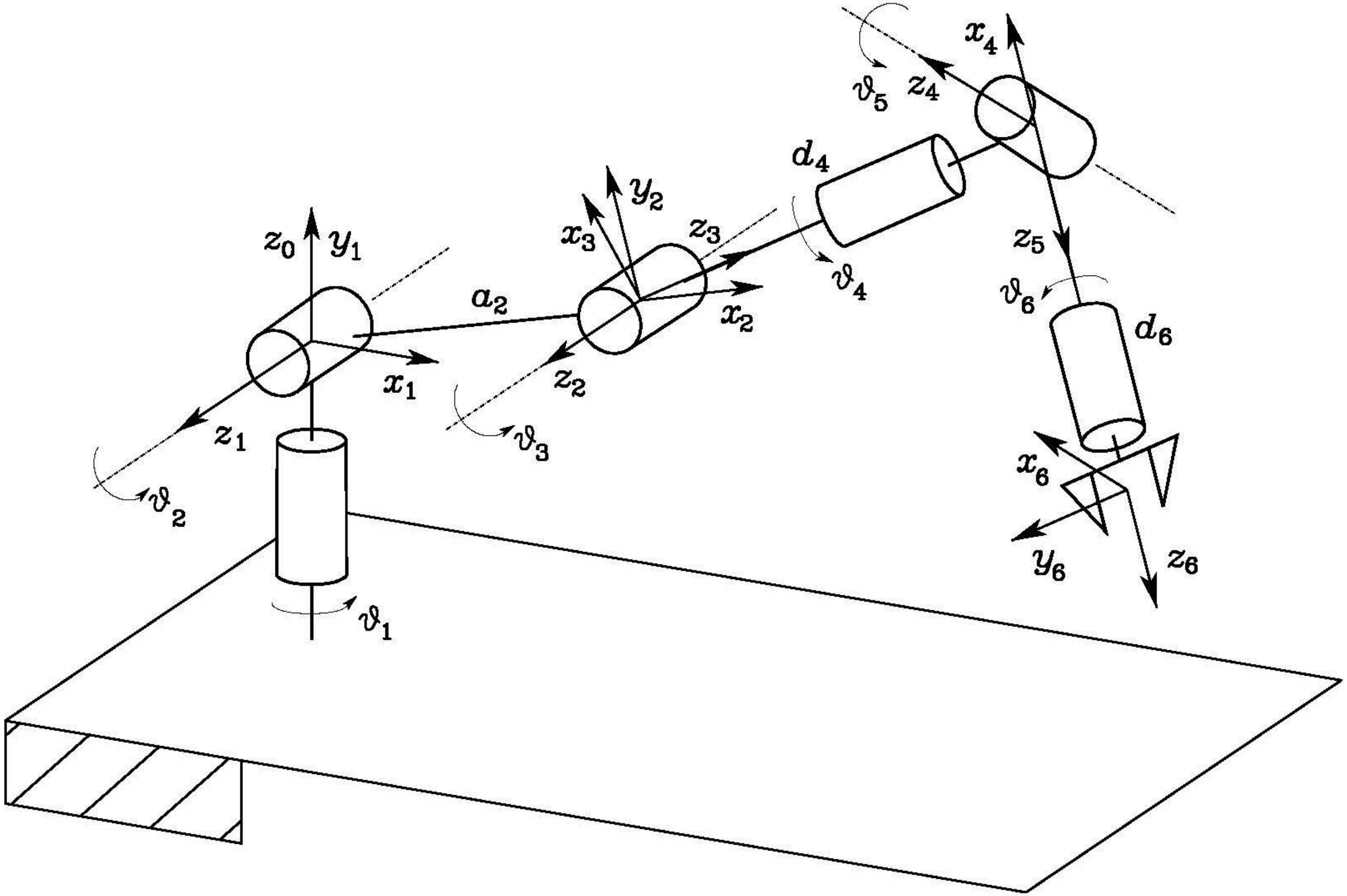
$$A_6^5(\vartheta_6) = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^3(q) = A_4^3 A_5^4 A_6^5 = \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Anthropomorphic Arm with Spherical Wrist

DH parameters

Link	a_i	α_i	d_i	ϑ_i
1	0	$\pi/2$	0	ϑ_1
2	a_2	0	0	ϑ_2
3	0	$\pi/2$	0	ϑ_3
4	0	$-\pi/2$	d_4	ϑ_4
5	0	$\pi/2$	0	ϑ_5
6	0	0	d_6	ϑ_6



Anthropomorphic arm with spherical wrist with frame assignment

Anthropomorphic Arm with Spherical Wrist II

$$\boldsymbol{A}_3^2(\vartheta_3) = \begin{bmatrix} c_3 & 0 & s_3 & 0 \\ s_3 & 0 & -c_3 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \boldsymbol{A}_4^3(\vartheta_4) = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Anthropomorphic Arm with Spherical Wrist III

$$\boldsymbol{n}_6^0 = \begin{bmatrix} c_1\big(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6\big) + s_1(s_4c_5c_6 + c_4s_6) \\ s_1\big(c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6\big) - c_1(s_4c_5c_6 + c_4s_6) \\ s_{23}(c_4c_5c_6 - s_4s_6) + c_{23}s_5c_6 \end{bmatrix}$$

$$\boldsymbol{s}_6^0 = \begin{bmatrix} c_1\big(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6\big) + s_1(-s_4c_5s_6 + c_4c_6) \\ s_1\big(-c_{23}(c_4c_5s_6 + s_4c_6) + s_{23}s_5s_6\big) - c_1(-s_4c_5s_6 + c_4c_6) \\ -s_{23}(c_4c_5s_6 + s_4c_6) - c_{23}s_5s_6 \end{bmatrix}$$

$$\boldsymbol{a}_6^0 = \begin{bmatrix} c_1(c_{23}c_4s_5 + s_{23}c_5) + s_1s_4s_5 \\ s_1(c_{23}c_4s_5 + s_{23}c_5) - c_1s_4s_5 \\ s_{23}c_4s_5 - c_{23}c_5 \end{bmatrix}$$

$$\boldsymbol{p}_6^0 = \begin{bmatrix} a_2c_1c_2 + d_4c_1s_{23} + d_6\big(c_1(c_{23}c_4s_5 + s_{23}c_5) + s_1s_4s_5\big) \\ a_2s_1c_2 + d_4s_1s_{23} + d_6\big(s_1(c_{23}c_4s_5 + s_{23}c_5) - c_1s_4s_5\big) \\ a_2s_2 - d_4c_{23} + d_6(s_{23}c_4s_5 - c_{23}c_5) \end{bmatrix}$$

Joint Space and Operational Space

Joint space

$$\boldsymbol{q} = \begin{bmatrix} q_1 & \dots & q_n \end{bmatrix}^T$$

- $q_i = \vartheta_i$ (revolute joint)
- $q_i = d_i$ (prismatic joint)

Operational space

$$\boldsymbol{x}_e = \begin{bmatrix} \boldsymbol{p}_e \\ \boldsymbol{\phi}_e \end{bmatrix} \qquad \boldsymbol{x}_e \in \mathrm{R}^m, m \le n$$

- $\boldsymbol{p}_e$ (position)
- $\boldsymbol{\phi}_e$ (orientation)

Direct kinematics equation

$$\boldsymbol{x}_e = \boldsymbol{k}(\boldsymbol{q})$$

$m < n$: Kinematically *redundant* manipulator

Joint Space and Operational Space

Examples

- Three-link planar arm

$$\boldsymbol{x}_e = \begin{bmatrix} p_x \\ p_y \\ \phi \end{bmatrix} = \boldsymbol{k}(\boldsymbol{q}) = \begin{bmatrix} a_1c_1 + a_2c_{12} + a_3c_{123} \\ a_1s_1 + a_2s_{12} + a_3s_{123} \\ \vartheta_1 + \vartheta_2 + \vartheta_3 \end{bmatrix}$$

- In the most general case of a six-dimensional operational space ($m = 6$), the computation of the three components of the function $\boldsymbol{\phi}_e(\boldsymbol{q})$ cannot be performed in closed form but goes through the computation of the elements of the rotation matrix $\boldsymbol{R}_e(\boldsymbol{q}) = [\boldsymbol{n}_e(\boldsymbol{q}) \quad \boldsymbol{s}_e(\boldsymbol{q}) \quad \boldsymbol{a}_e(\boldsymbol{q})]$ via inverse formulae

Procedure Denavit-Hartenberg

SECTION 3

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

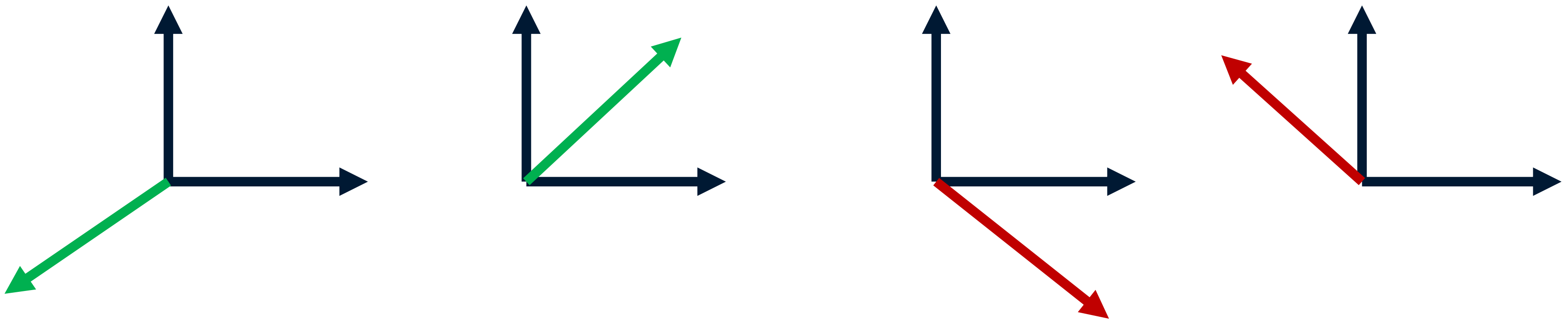
1. Assign frames according to the 4 Denavit-Hartenberg Rules
2. Create the Denavit-Hartenberg parameter table
3. Plug the table values into the Homogeneous Transformation Matrix
4. Multiply the matrices together

Denavit-Hartenberg Frames

SECTION 4

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.



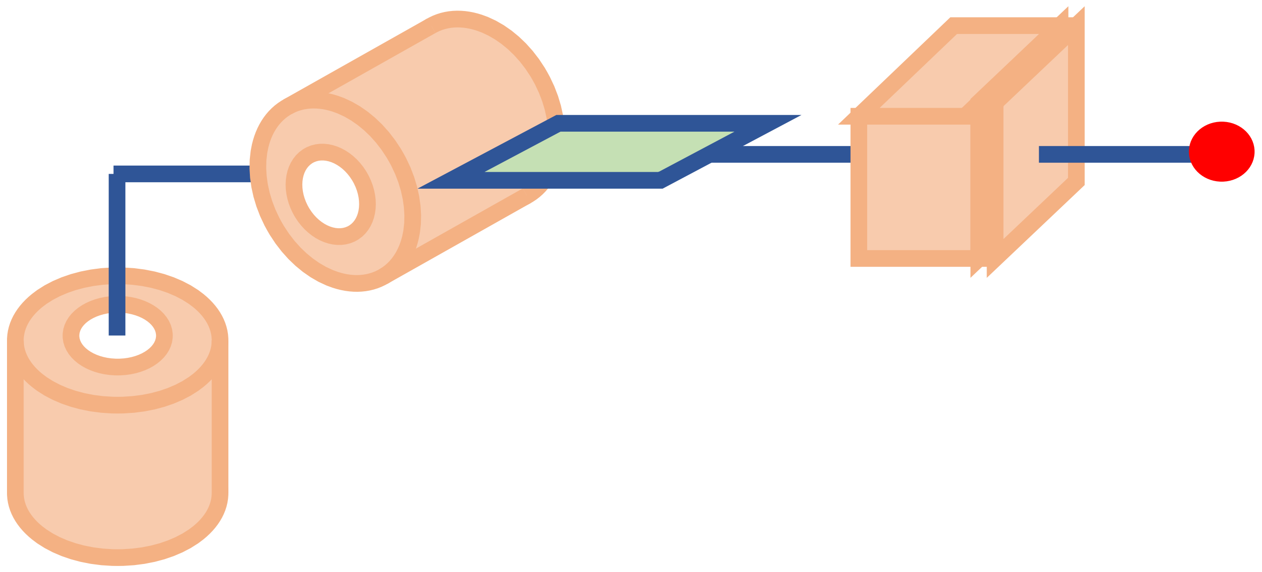
4 Denavit-Hartenberg Frame Rules

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.

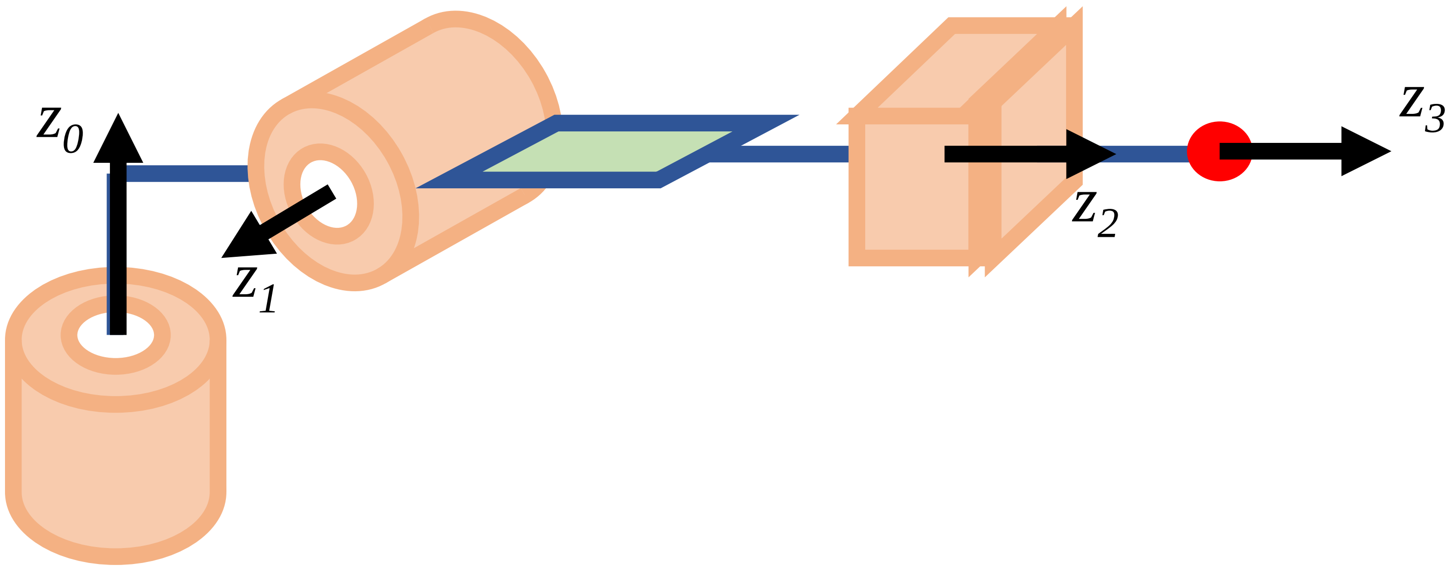
3 Joints

4 Frames at least

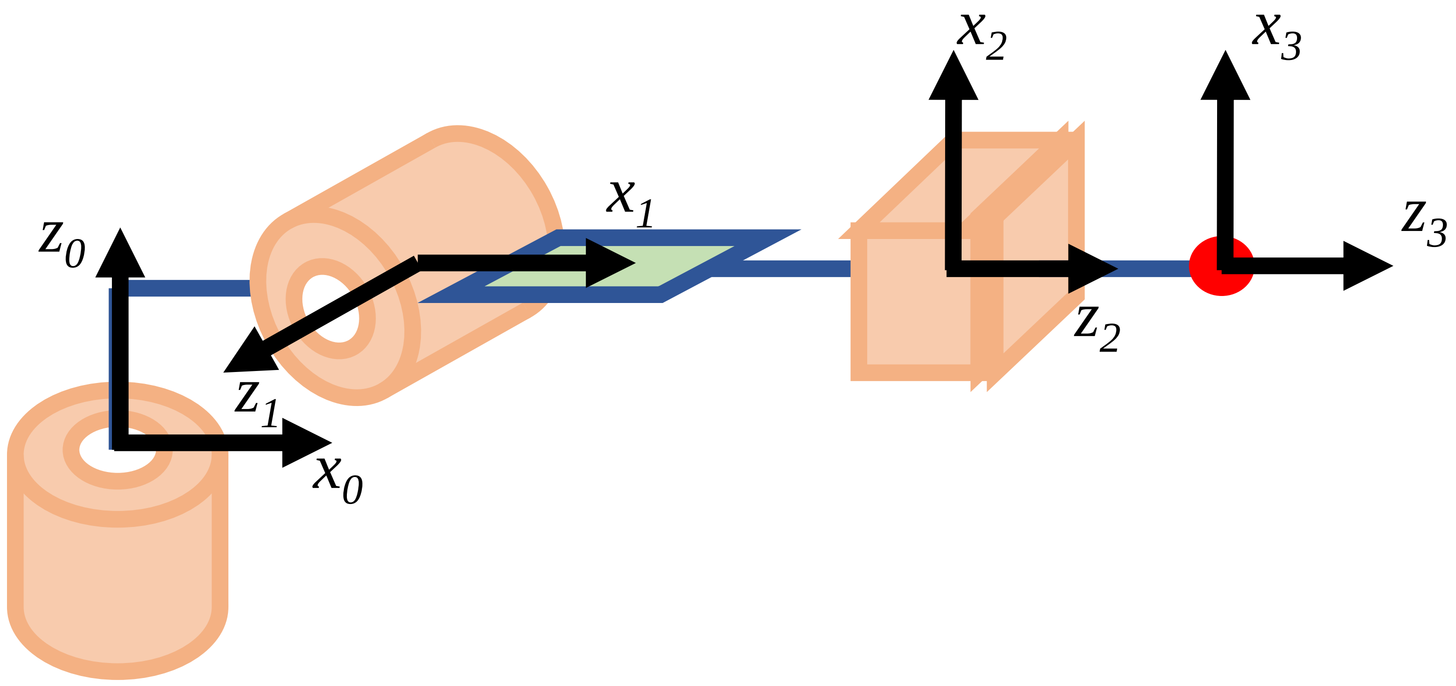
3 DOF



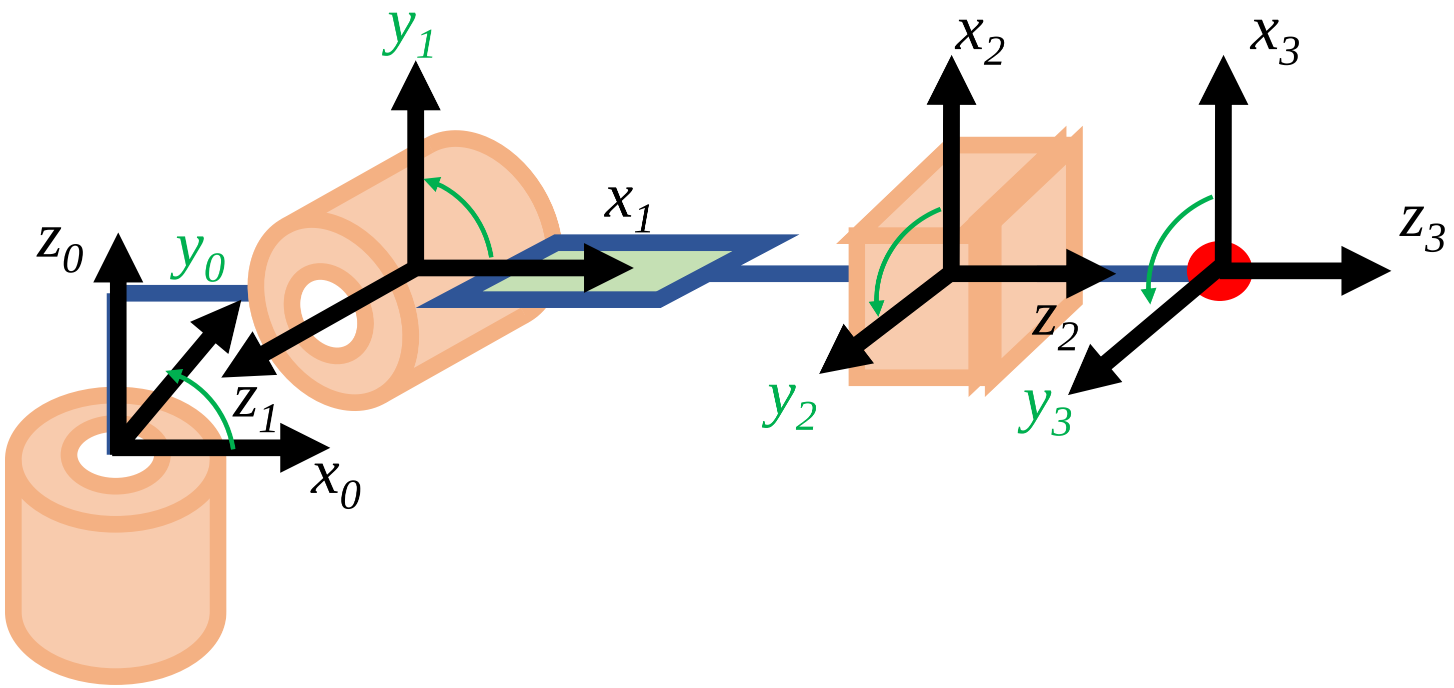
Rule 1: Z axis

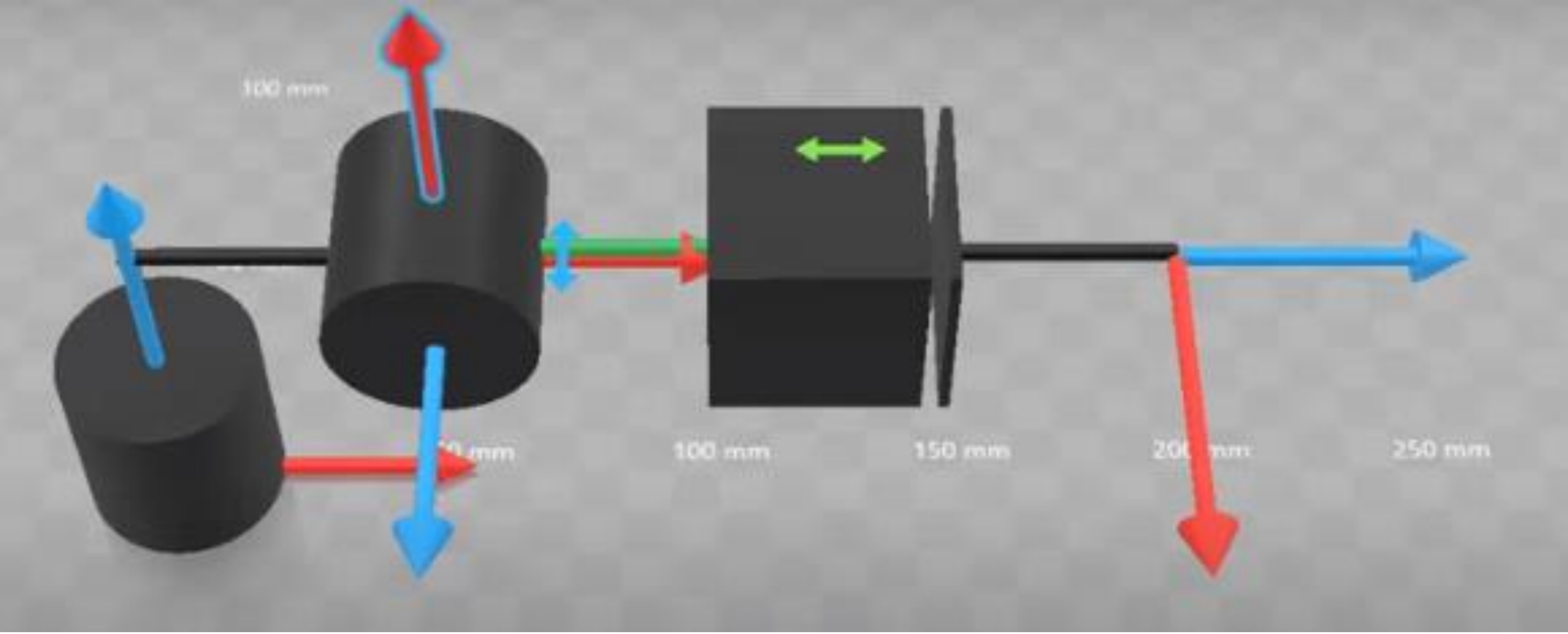


Rule 2/3: X axis



Rule 4: Y axis




100 mm
100 mm
150 mm
250 mm

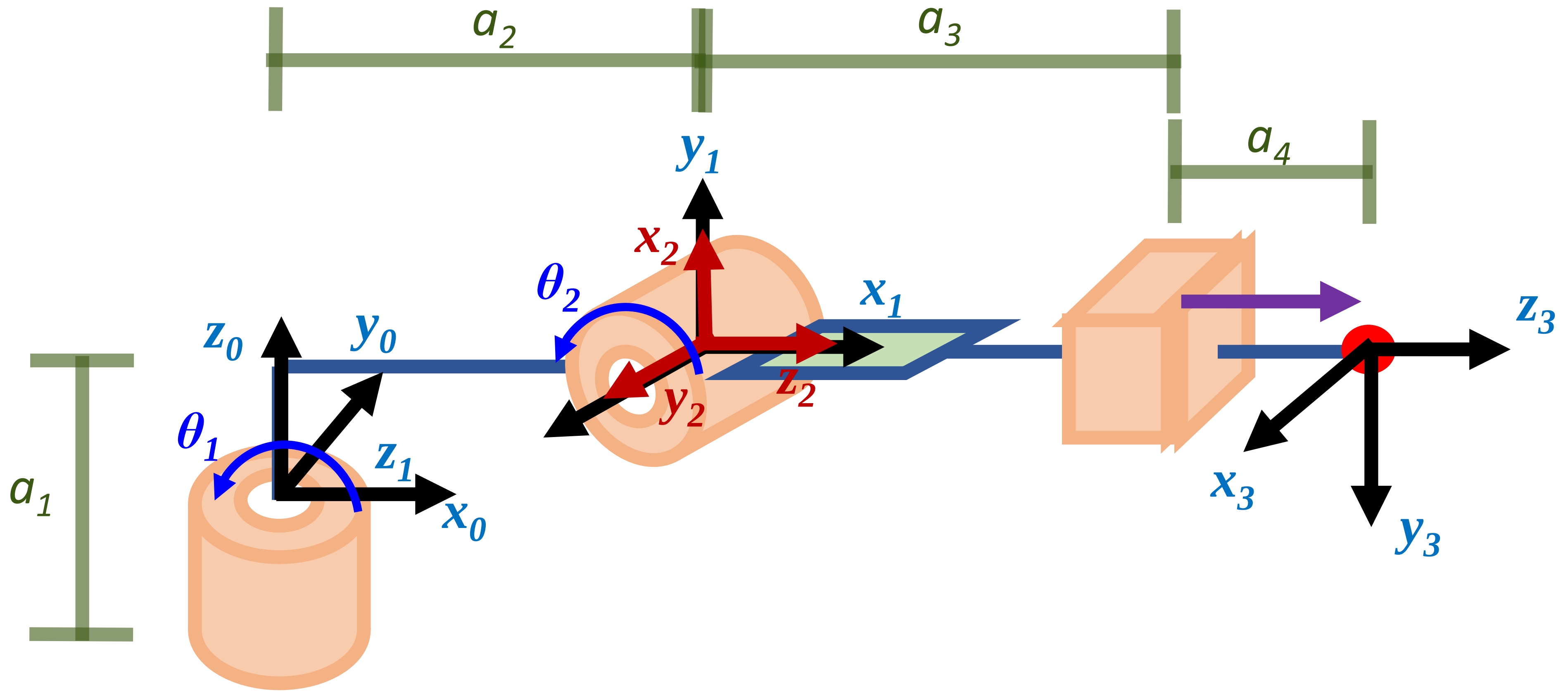
Denavit-Hartenberg Parameter Table

SECTION 5

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

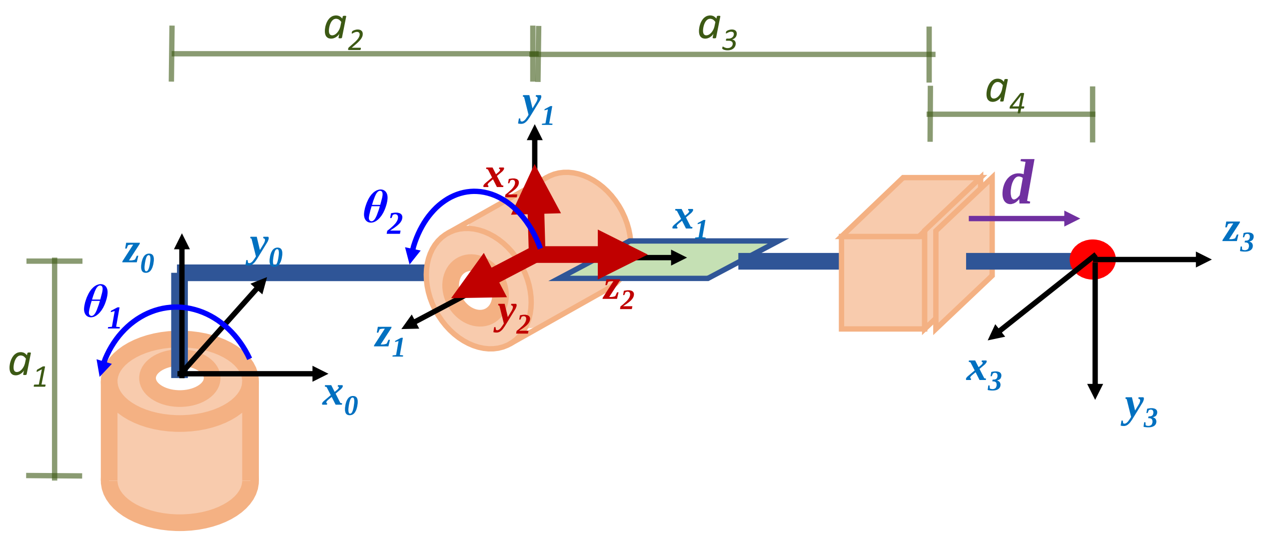


a_2
a_3
a_4
a_1
y_1
x_2
θ_2
x_1
z_0
y_0
θ_1
z_1
x_0
y_2
z_2
z_3
x_3
y_3

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

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

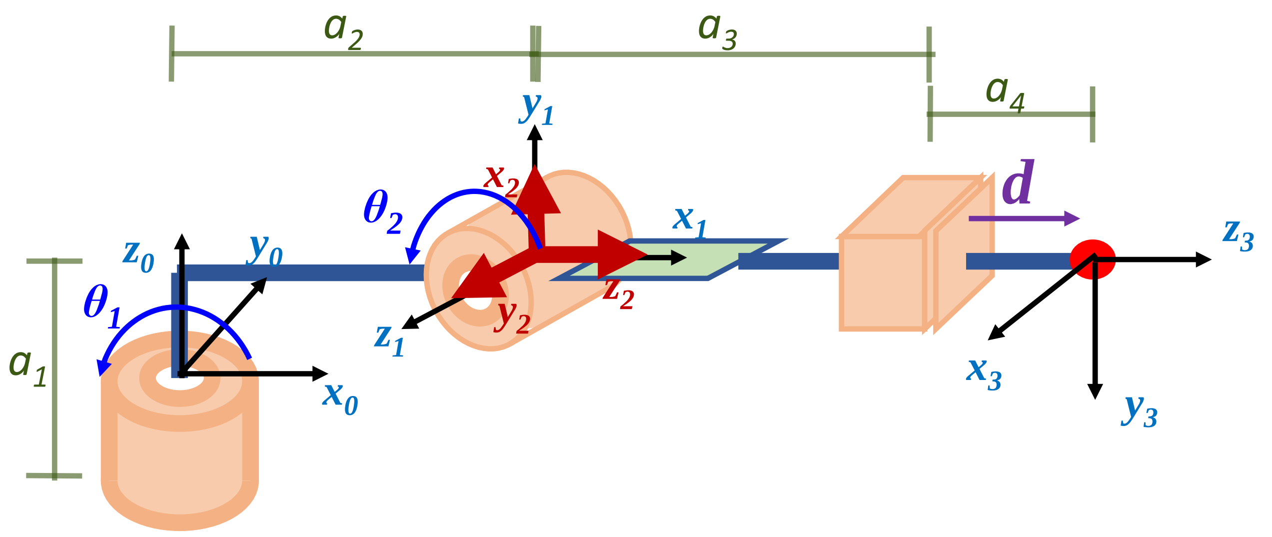
	θ	α	r	d
1	$0+\theta_1$			
2	$90+\theta_2$			
3	90			



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

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

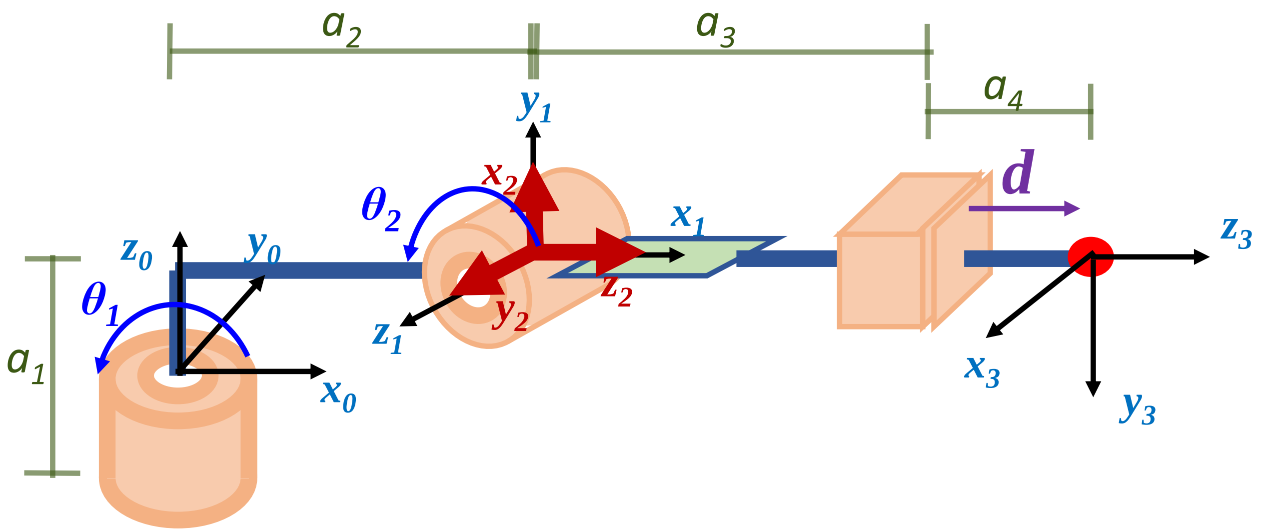
	θ	α	r	d
1	$0+\theta_1$	90		
2	$90+\theta_2$	90		
3	90	0		



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

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

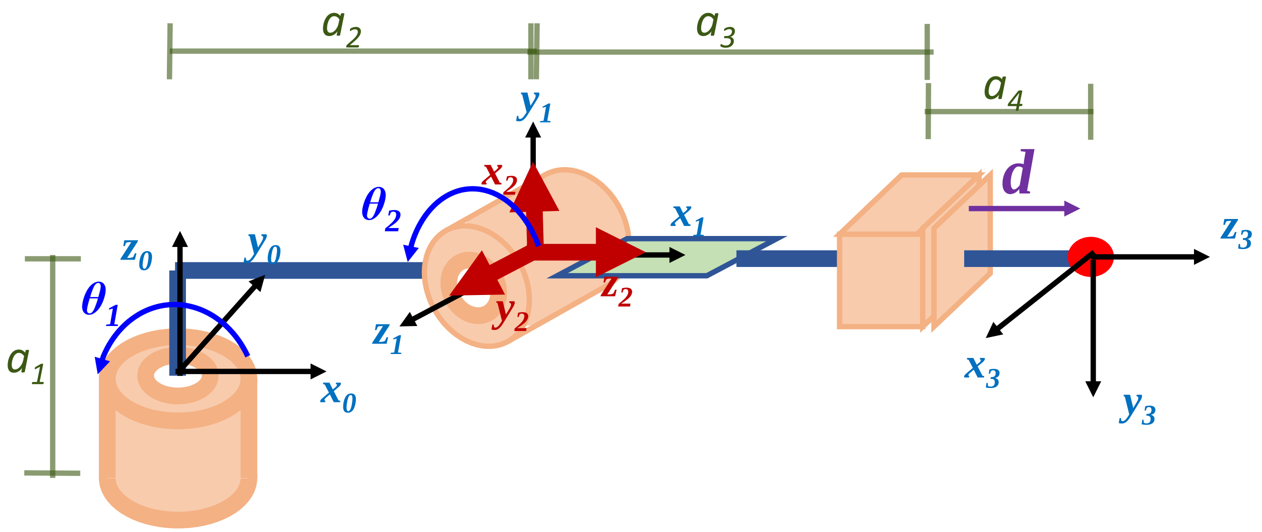
	θ	α	r	d
1	0+θ$_1$	90	a_2	
2	90+θ$_2$	90	0	
3	90	0	0	



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 6

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 7

DH_HTM(T, af, r, d)

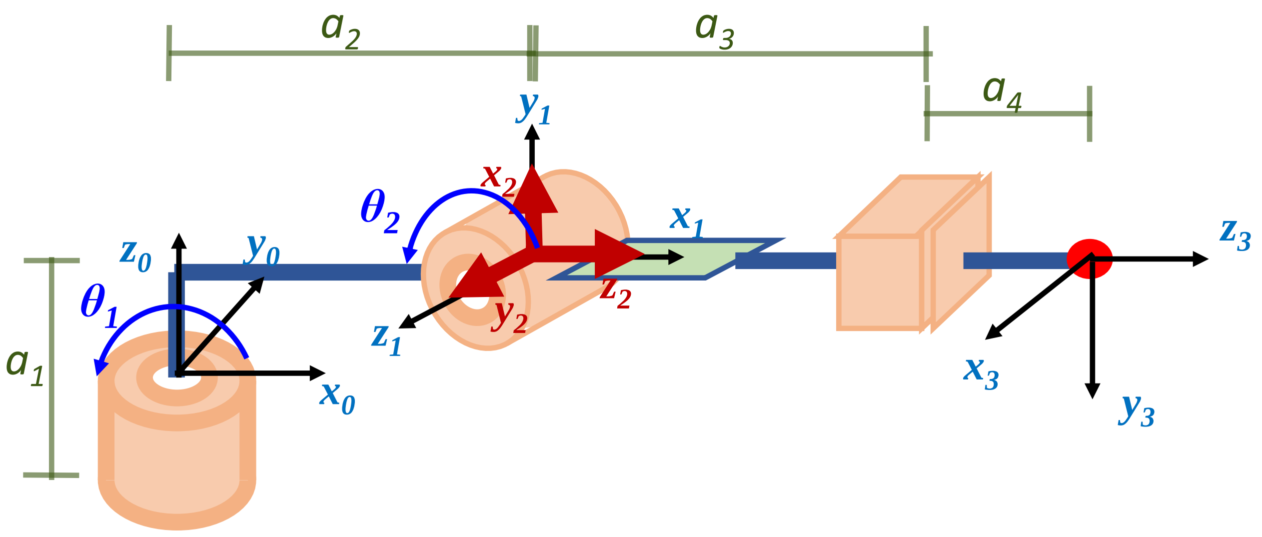
```
D[0][0] = np.cos(T)
D[0][1] = -np.sin(T) * np.cos(af)
D[0][2] = np.sin(T) * np.sin(af)
D[0][3] = r * np.cos(T)

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.]]
```