Mct/ROB/200 Robotics, Spring Term 12-13

Lecture 4 – Friday March 15, 2012

Forward Kinematics

Objectives

When you have finished this lecture you should be able to:

- Learn how to derive the forward kinematic equations of the robot using Denavit-Hartenberg (D-H) representation technique.

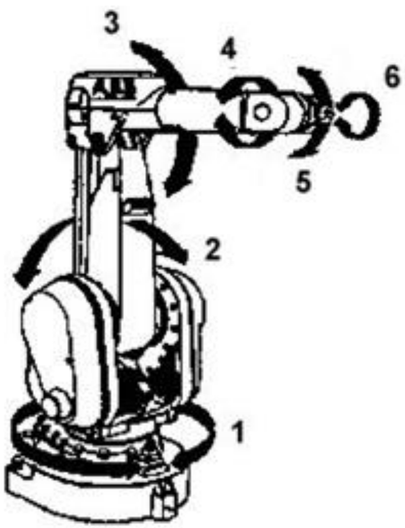
Outline

- Forward Kinematics
- Denavit-Hartenberg Algorithm
- Summary

Outline

- **Forward Kinematics**
- Denavit-Hartenberg Algorithm
- Summary

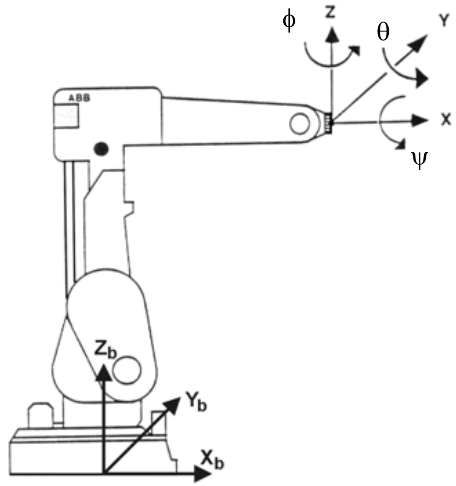
Forward Kinematics



Given: The angle of each joint

Required: The position of end-effector or any point (i.e. its coordinates) $(p_x, p_y, p_z, \phi, \theta, \psi)$

Forward Kinematics



Given:

θ_1
θ_2
θ_3
θ_4
θ_5
θ_6

Joint Space

Required:

$p_x = f_x(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$p_y = f_y(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$p_z = f_z(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$\phi = f_\phi(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$\theta = f_\theta(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$\psi = f_\psi(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$

Cartesian Space

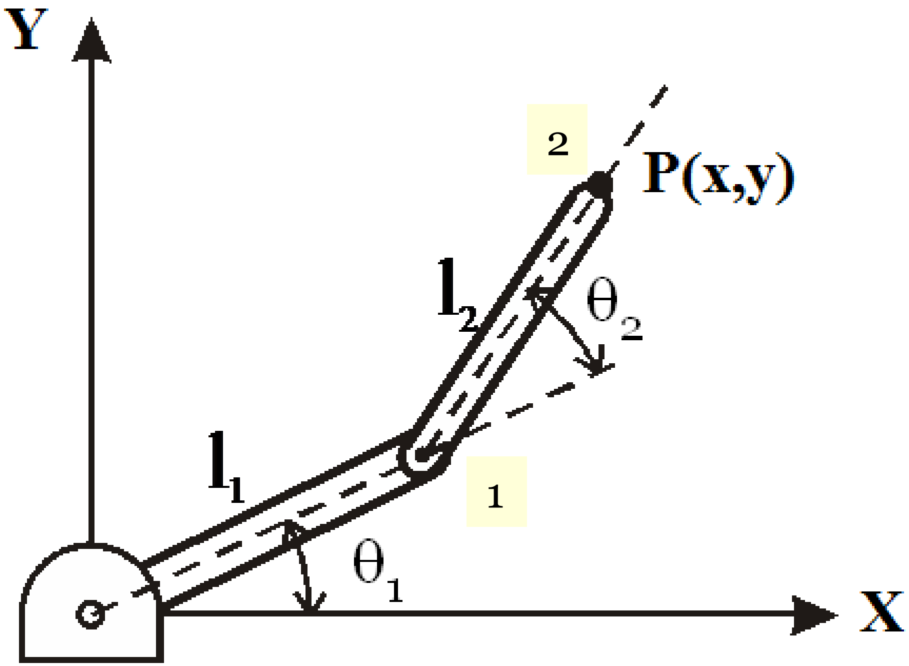
Forward Kinematics

- **A Plan Arm with 2-DOF**

Given: θ_1, θ_2

Required: x, y

Solution:



Trigonometric Solution

$$x = l_1 \cdot \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2)$$

$$y = l_1 \cdot \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2)$$

See Trigonometric Solution.xls

Algebraic Solution

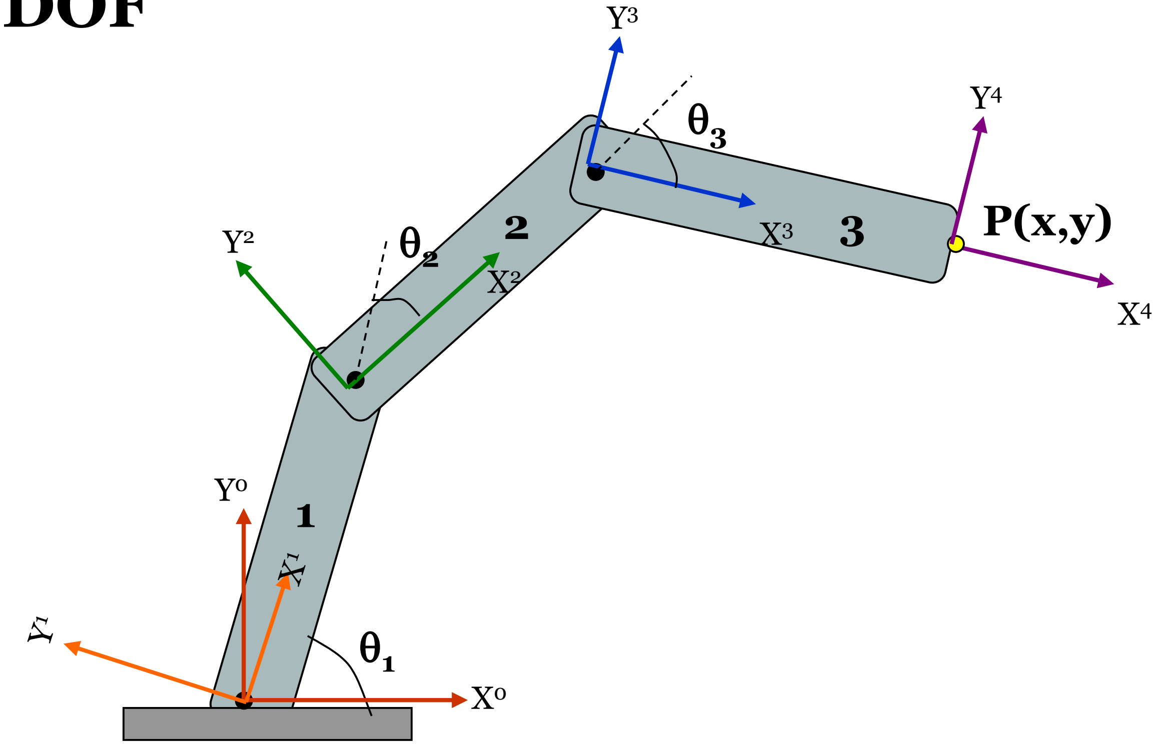
$${}^0T_2 = {}^0T_1\ {}^1T_2$$

Forward Kinematics

- **A Plan Robot with 3-DOF**

Given: $\theta_1, \theta_2, \theta_3, l_1, l_2, l_3$

Required: x, y



Solution:

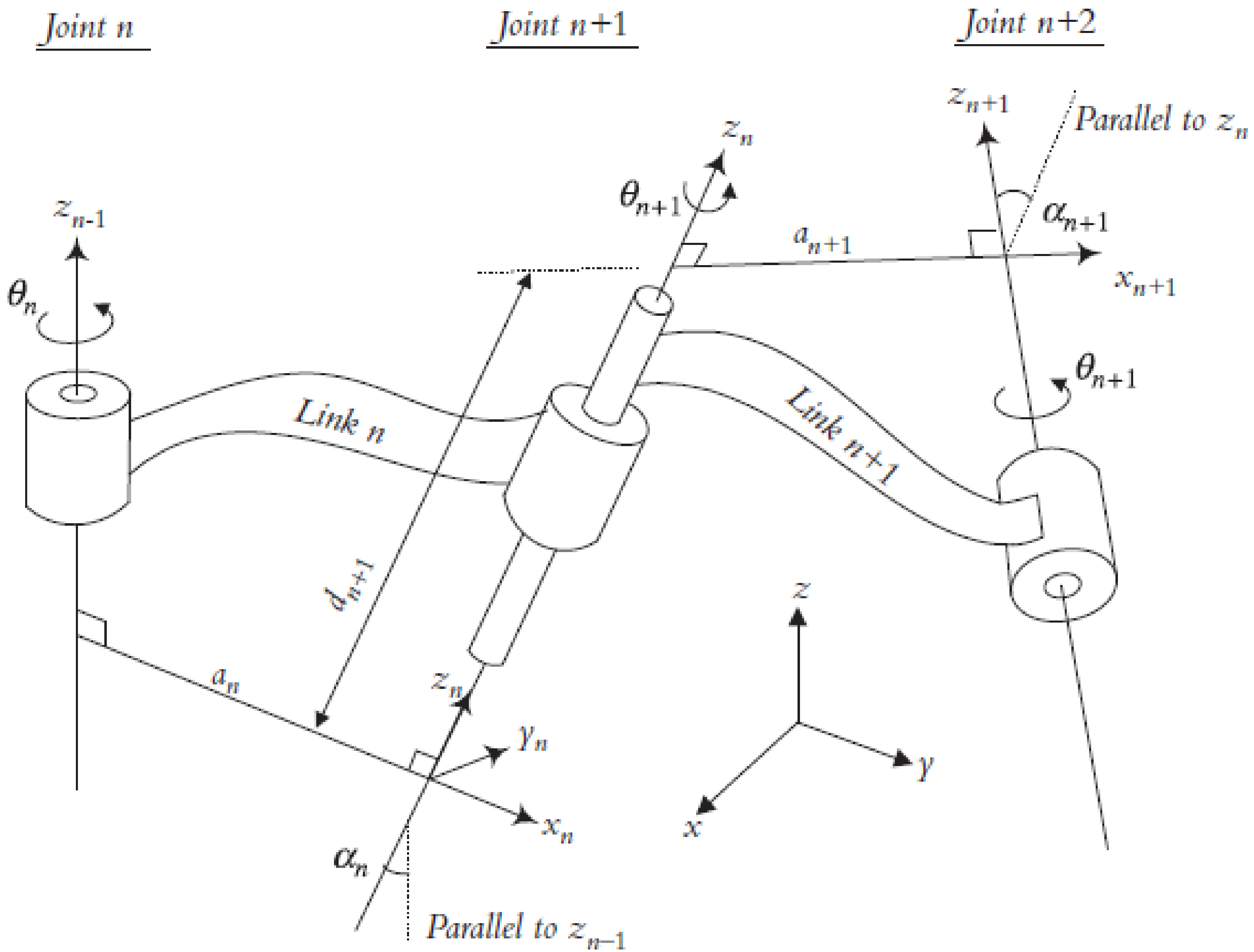
$$ {}^{0}T_4 = \quad {}^{0}T_1 \quad {}^{1}T_2 \quad {}^{2}T_3 \quad {}^{3}T_4 $$

${}^{0}T_1$	${}^{1}T_2$	${}^{2}T_3$	${}^{3}T_4$
Rotation (θ_1)	*Rotation (θ_2) followed by translation (l_1)*	*Rotation (θ_3) followed by translation (l_2)*	*translation (l_3)*

Outline

- Forward Kinematics
- **Denavit-Hartenberg Algorithm**
- Summary

Denavit-Hartenberg Algorithm



Denavit-Hartenberg Algorithm

Analyzing the robot morphology (Links and Joints)

↓

Establishing coordinate systems S_i for each joint

↓

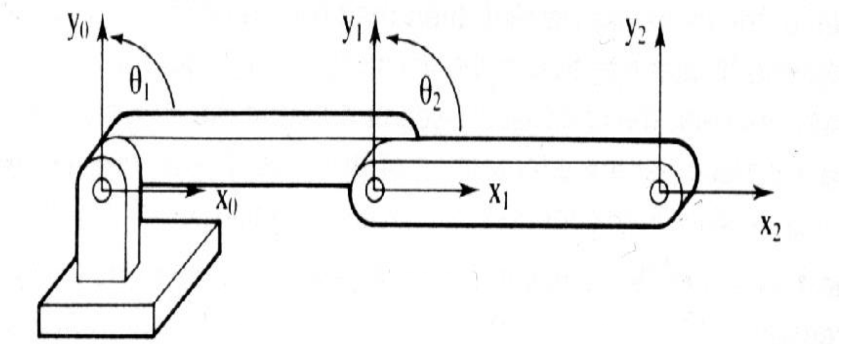
Calculate D-H Parameters: (θ_i, d_i, a_i, α_i)

↓

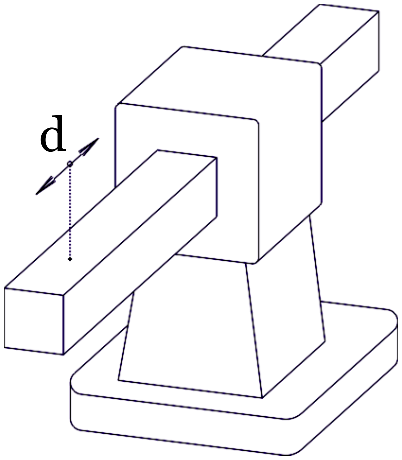
Obtaining D-H Transformation Matrix for adjacent coordinate frames

Denavit-Hartenberg Algorithm

- **Analyzing the robot morphology**



Revolute Joints
1 DOF (Variable - θ)



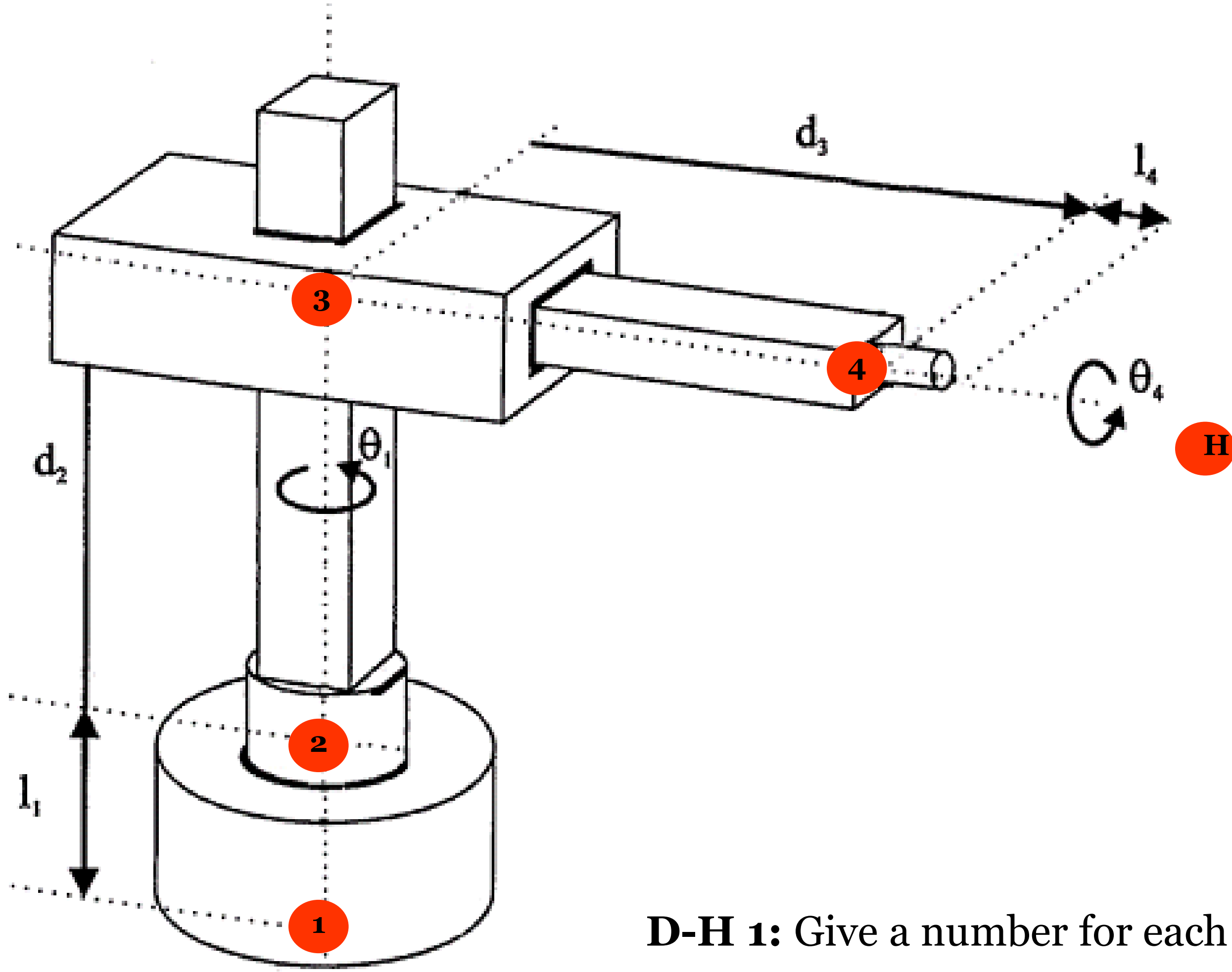
Prismatic Joint
1 DOF (linear) (Variable - d)

Axis of motion is in the direction of rotation as followed by the right-hand rule for rotations.

Axis of motion is along the direction of the linear movement.

Denavit-Hartenberg Algorithm

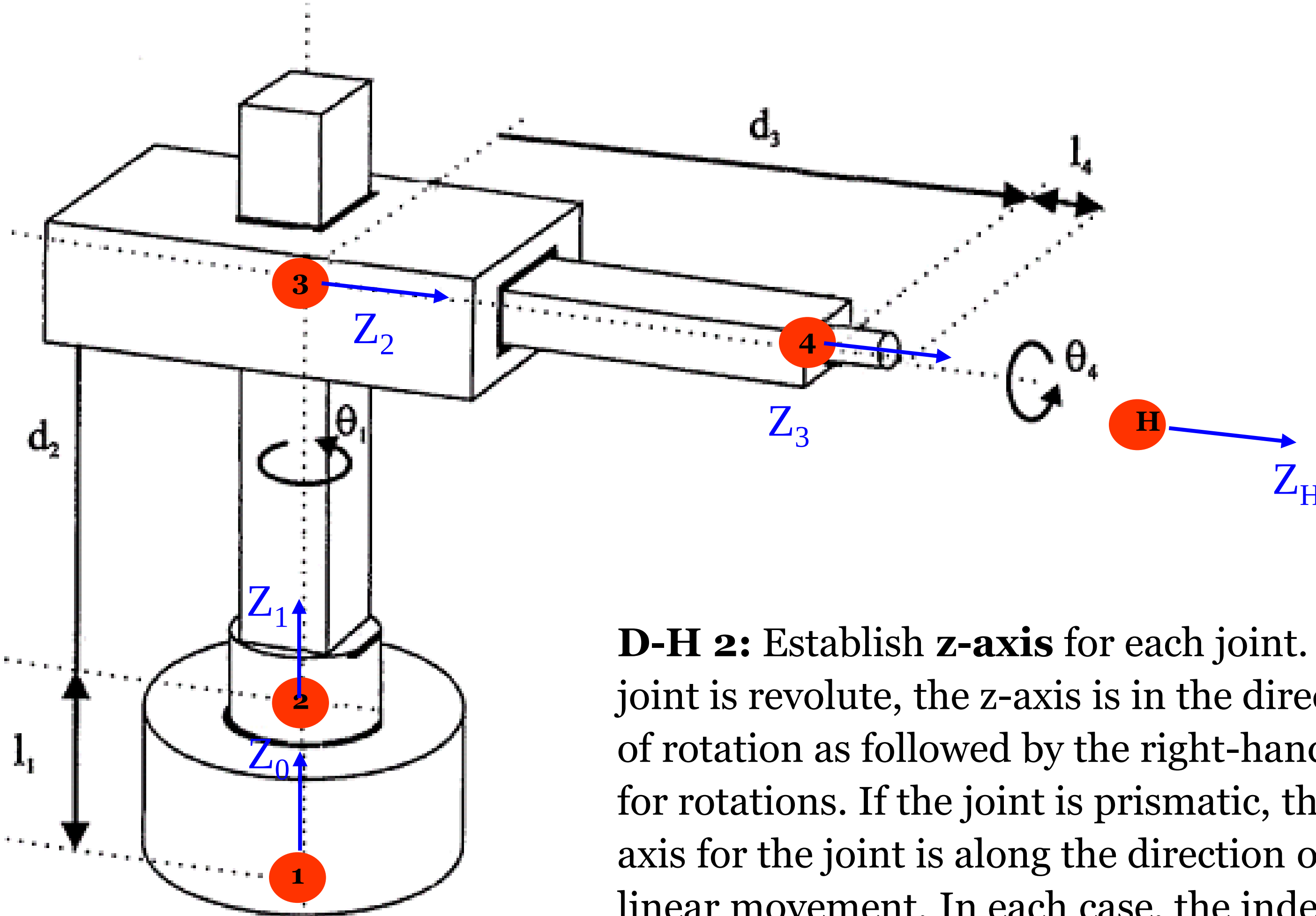
- **Analyzing the robot morphology**

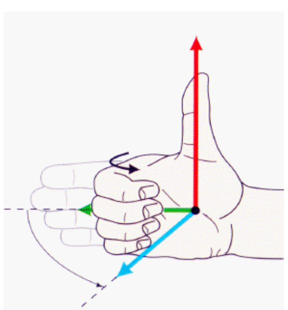


D-H 1: Give a number for each joint from 1 to N starting with the base and ending with the tool yaw, pitch, and roll, in that order

Denavit-Hartenberg Algorithm

- **Establishing Coordinate Systems**

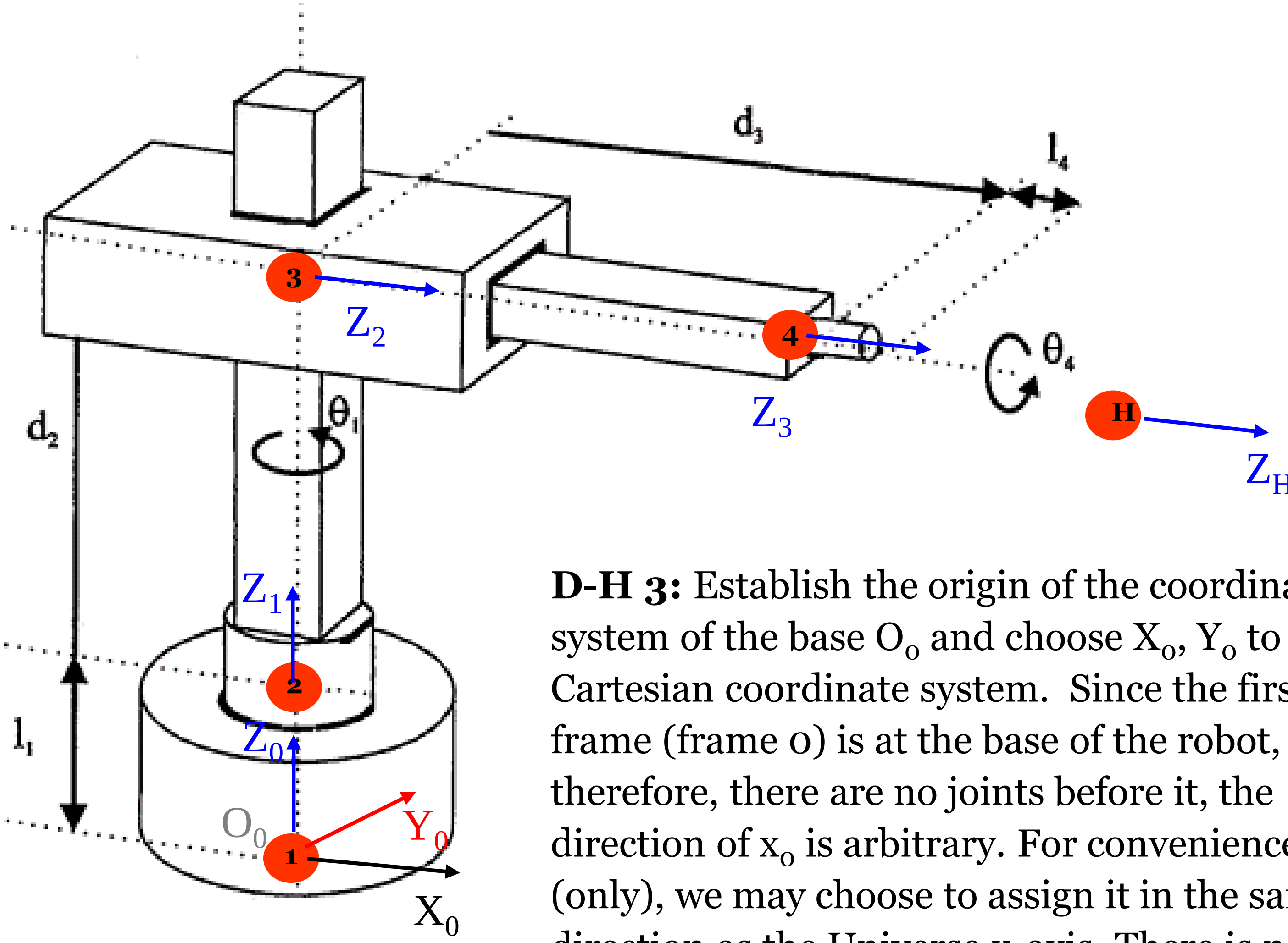


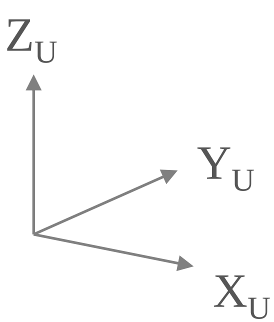


D-H 2: Establish **z-axis** for each joint. If the joint is revolute, the z-axis is in the direction of rotation as followed by the right-hand rule for rotations. If the joint is prismatic, the z-axis for the joint is along the direction of the linear movement. In each case, the index number for the z-axis of joint n (as well as the local reference frame for the joint) is n-1

Denavit-Hartenberg Algorithm

- **Establishing Coordinate Systems**

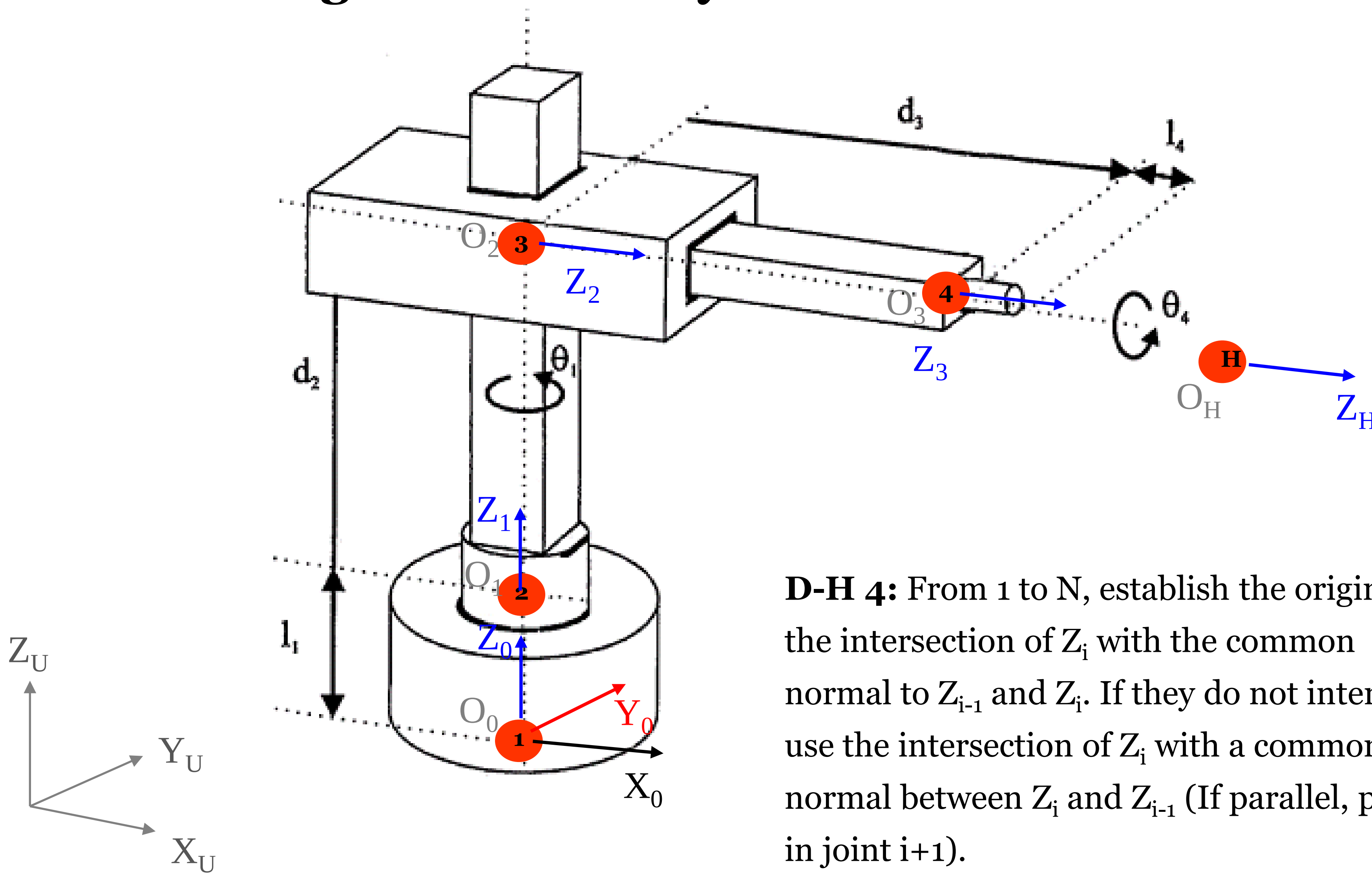




D-H 3: Establish the origin of the coordinate system of the base O_o and choose X_o, Y_o to form Cartesian coordinate system. Since the first frame (frame 0) is at the base of the robot, and therefore, there are no joints before it, the direction of x_o is arbitrary. For convenience (only), we may choose to assign it in the same direction as the Universe x-axis. There is no problem if another direction is chosen

Denavit-Hartenberg Algorithm

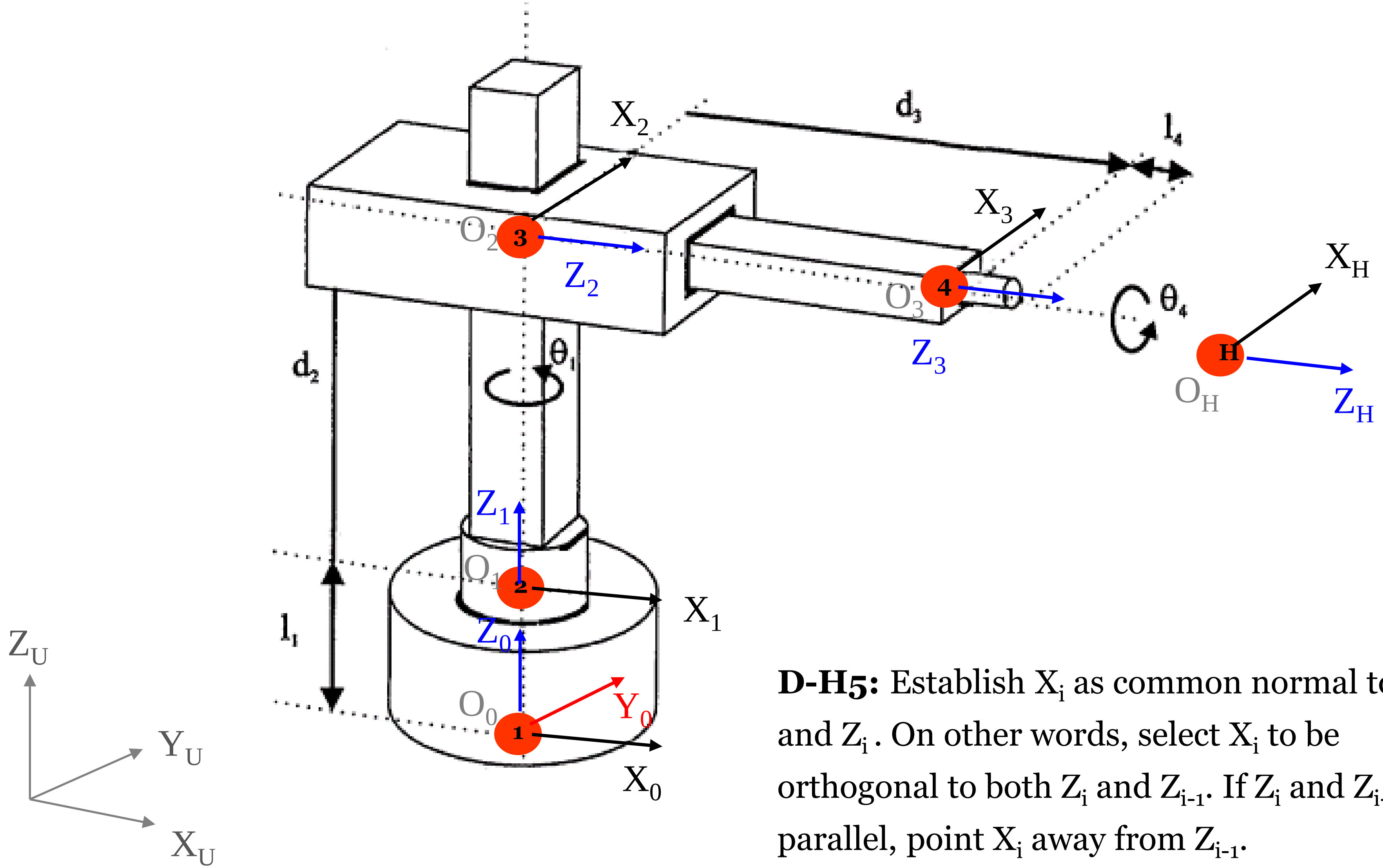
- **Establishing Coordinate Systems**



D-H 4: From 1 to N, establish the origin O_i in the intersection of Z_i with the common normal to Z_{i-1} and Z_i. If they do not intersect, use the intersection of Z_i with a common normal between Z_i and Z_{i-1} (If parallel, put O_i in joint i+1).

Denavit-Hartenberg Algorithm

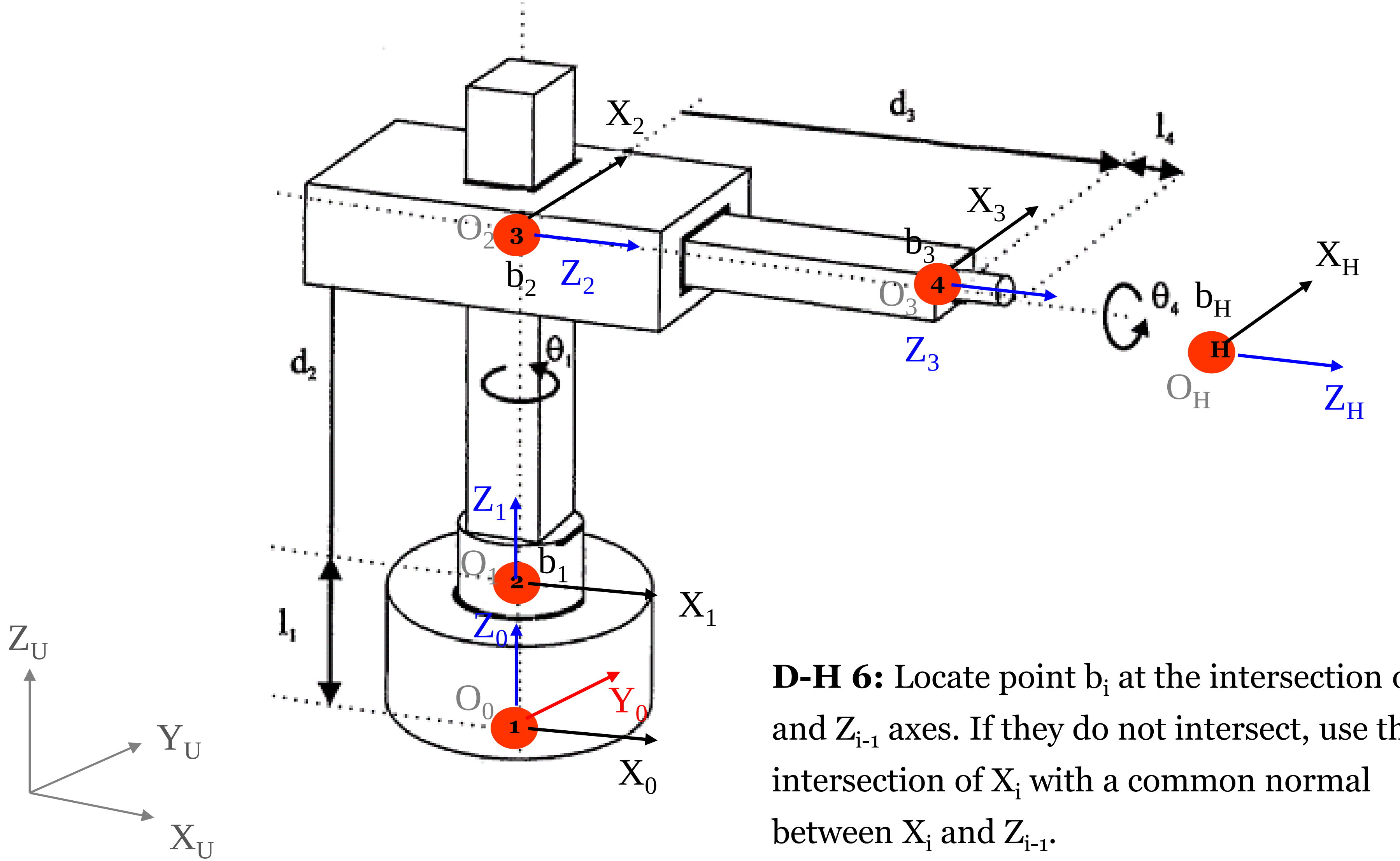
- **Establishing Coordinate Systems**



D-H5: Establish X_i as common normal to Z_{i-1} and Z_i . On other words, select X_i to be orthogonal to both Z_i and Z_{i-1}. If Z_i and Z_{i-1} are parallel, point X_i away from Z_{i-1}.

Denavit-Hartenberg Algorithm

- **Establishing Coordinate Systems**



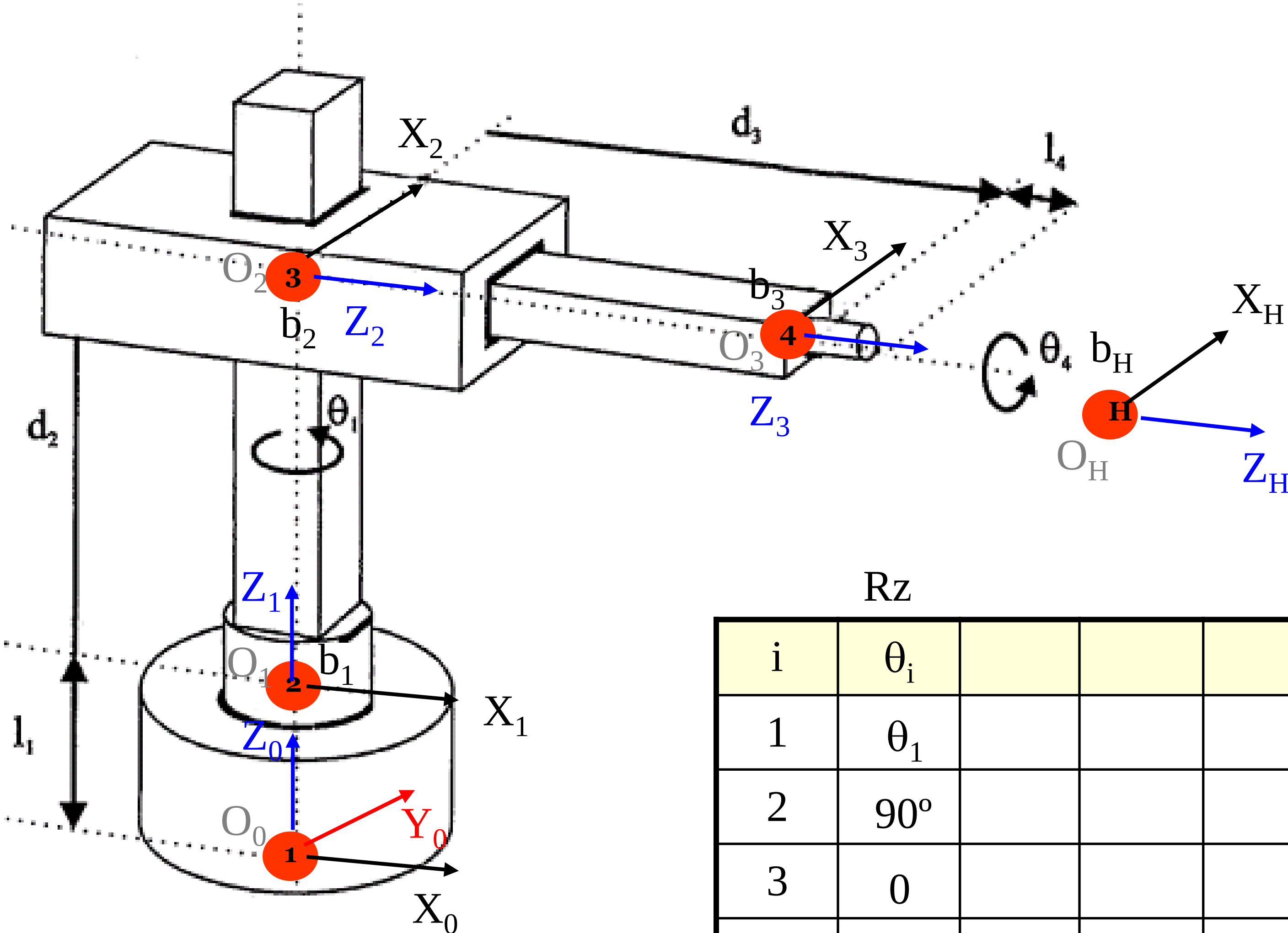
D-H 6: Locate point b_i at the intersection of X_i and Z_{i-1} axes. If they do not intersect, use the intersection of X_i with a common normal between X_i and Z_{i-1}.

Denavit-Hartenberg Algorithm

- **Calculate D-H Parameters**

θ_i

Is the rotation over Z_{i-1} required to make X_{i-1} parallel to X_i.



Rz

i	θ_i			
1	θ_1			
2	90º			
3	0			
H	θ_4			

Table of D-H Parameters

Denavit-Hartenberg Algorithm

- **Calculate D-H Parameters**

d_i

The distance from the origin of the frame O_{i-1} to the point b_i measured along Z_{i-1}

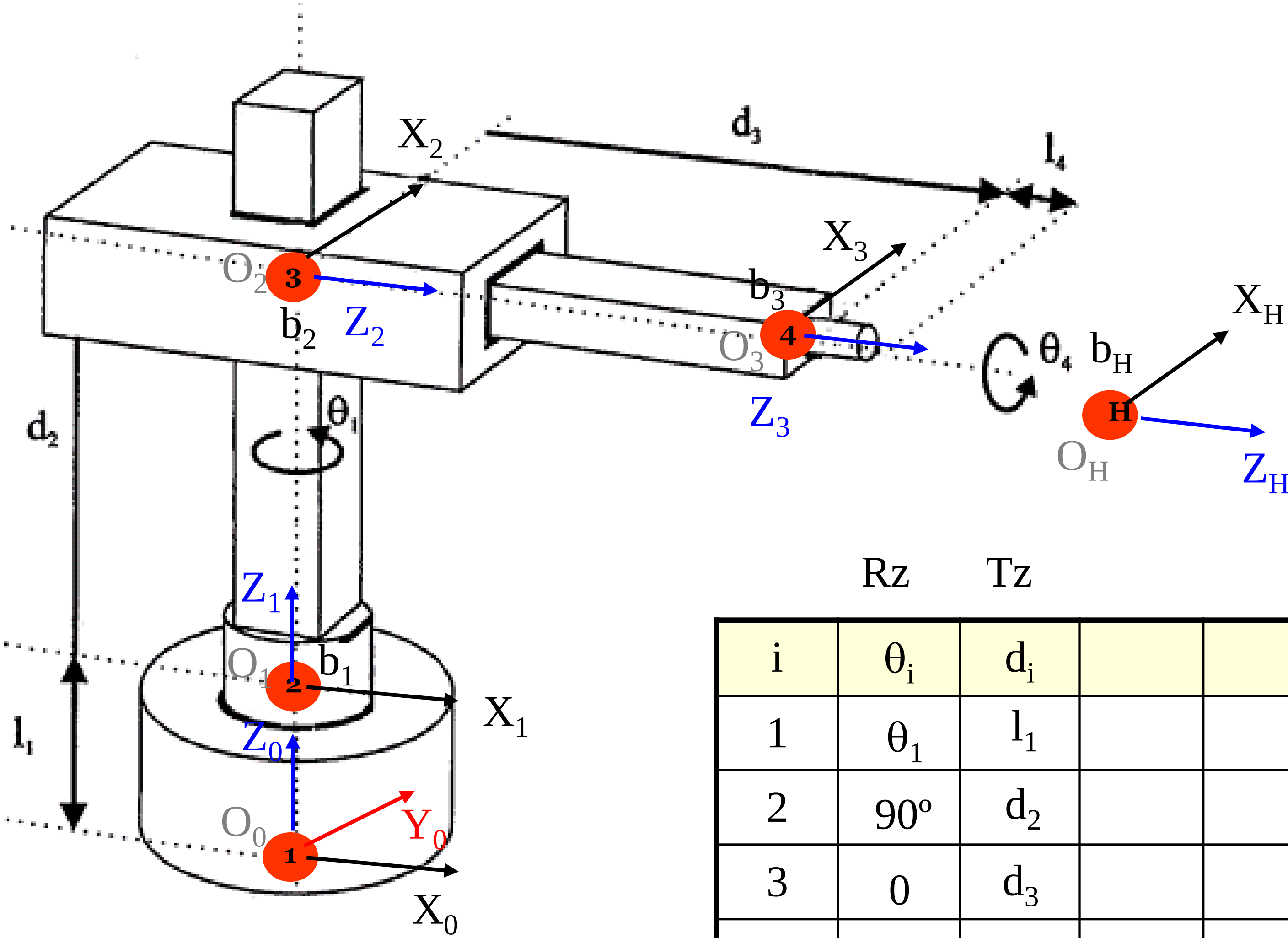


Table of D-H Parameters

	Rz	Tz		
i	θ_i	d_i		
1	θ_1	l_1		
2	90^o	d_2		
3	0	d_3		
H	θ_4	l_4		

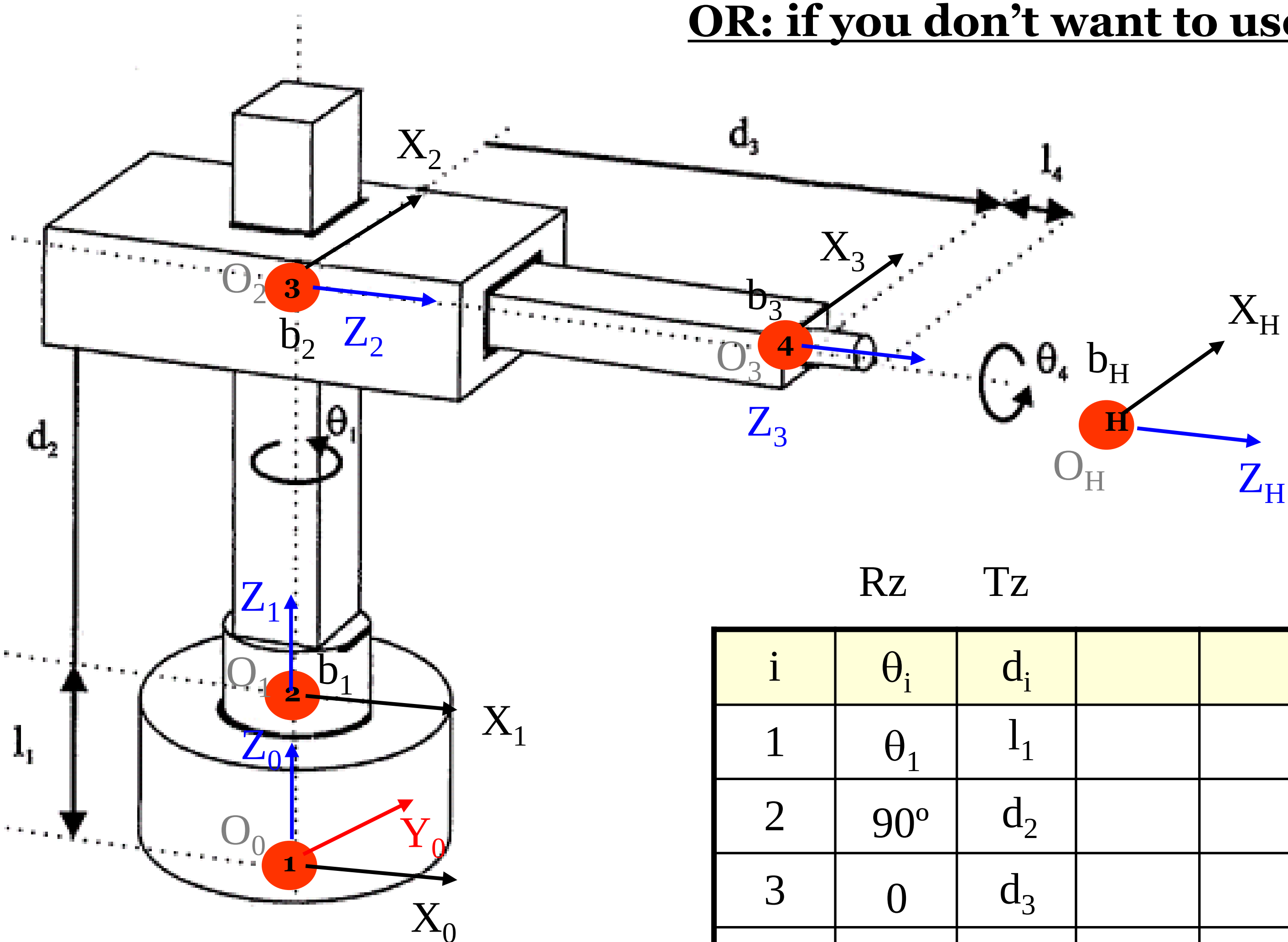
Denavit-Hartenberg Algorithm

- **Calculate D-H Parameters**

OR: if you don't want to use b_i

d_i

The translation over Z_{i-1} required to make X_{i-1} and X_i aligned



Rz Tz

i	θ_i	d_i		
1	θ_1	l_1		
2	90^o	d_2		
3	0	d_3		
H	θ_4	l_4		

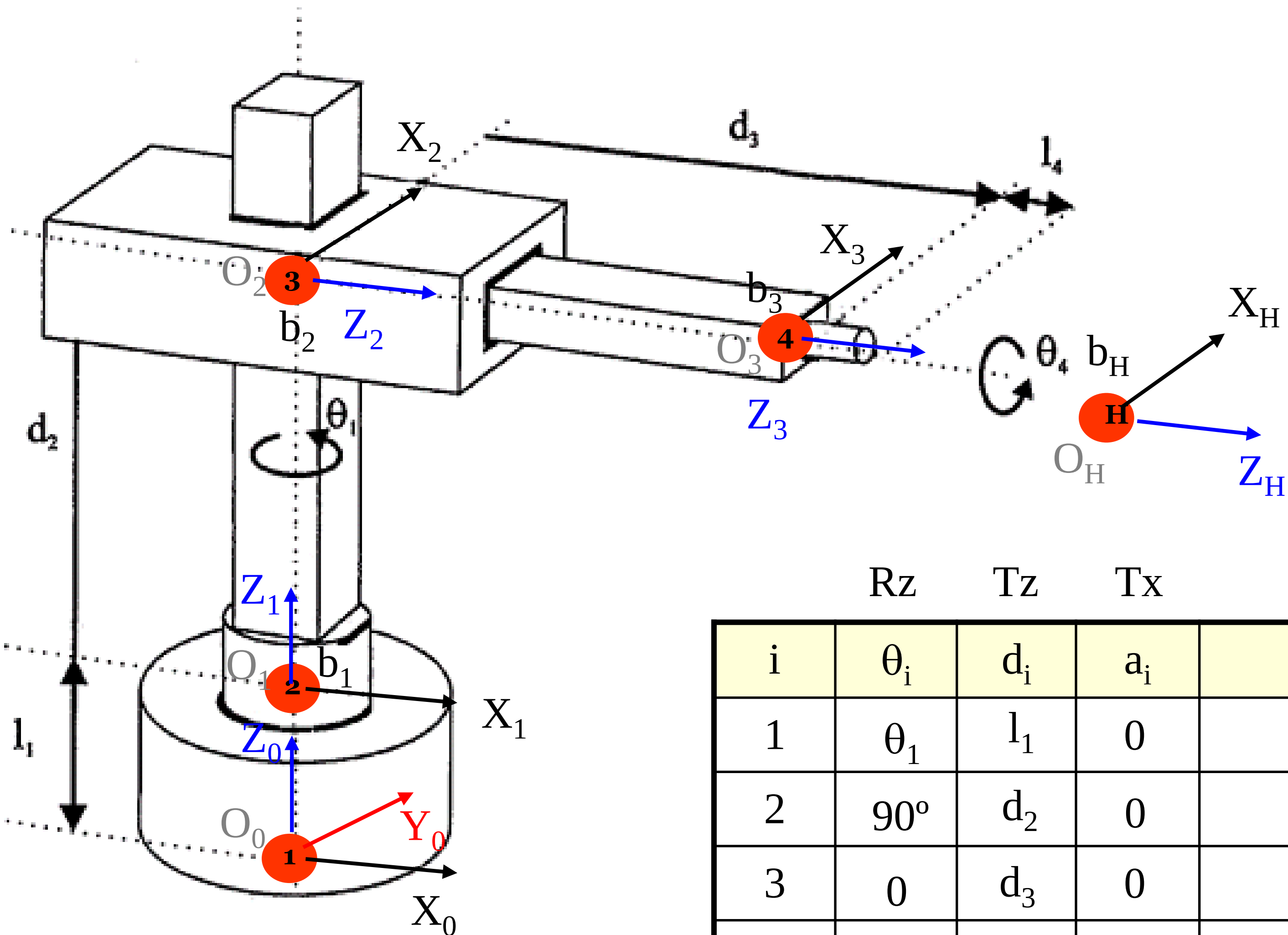
Table of D-H Parameters

Denavit-Hartenberg Algorithm

- **Calculate D-H Parameters**

a_i

The distance from point b_i to the origin of frame O_i measured along X_i



	Rz	Tz	Tx	
i	θ_i	d_i	a_i	
1	θ_1	l_1	0	
2	90º	d_2	0	
3	0	d_3	0	
H	θ_4	l_4	0	

Table of D-H Parameters

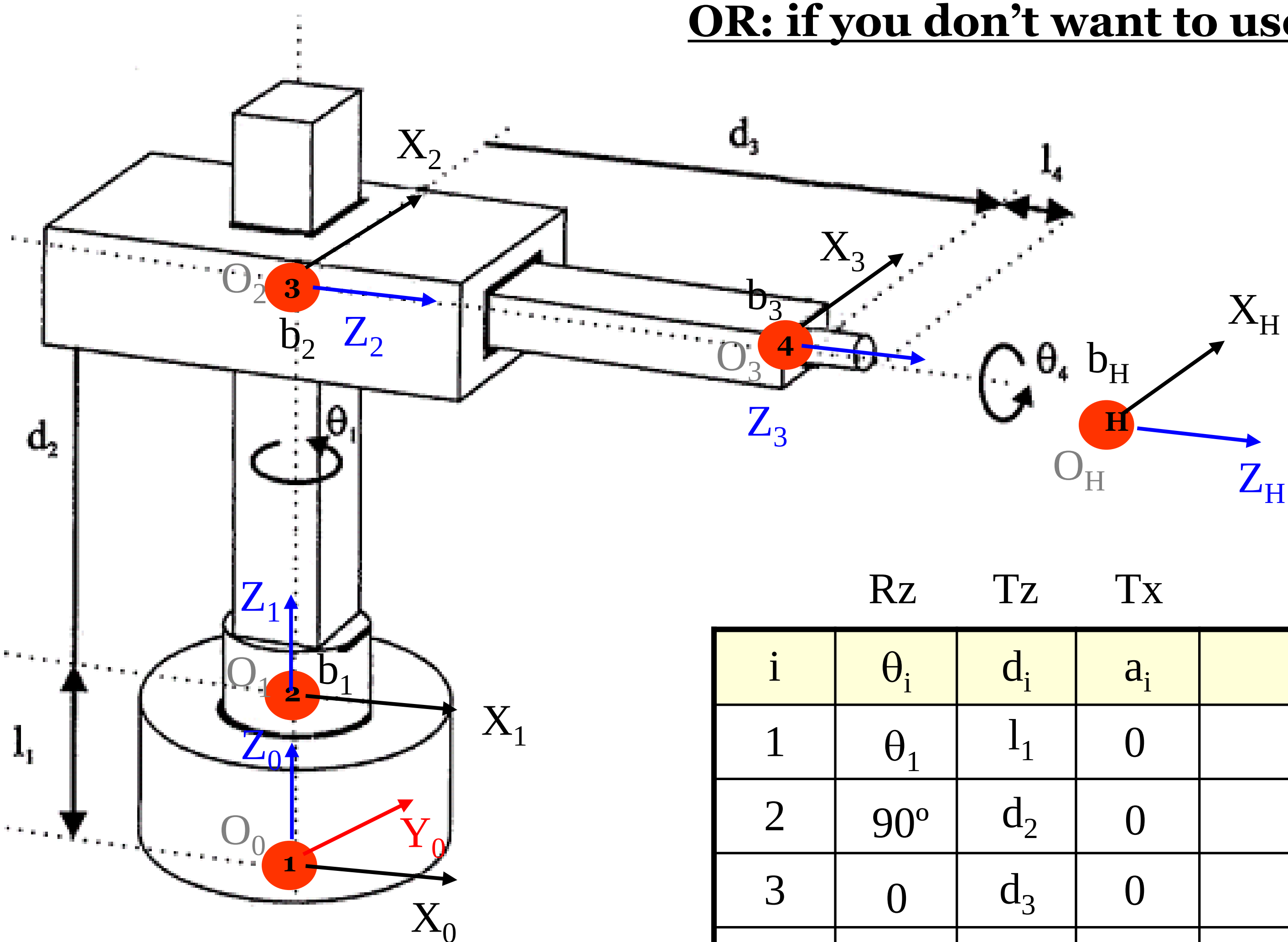
Denavit-Hartenberg Algorithm

- **Calculate D-H Parameters**

a_i

Is the translation over X_i required to make O_{i-1} and O_i coincided

OR: if you don't want to use b_i



	Rz	Tz	Tx	
i	θ_i	d_i	a_i	
1	θ_1	l_1	0	
2	90°	d_2	0	
3	0	d_3	0	
H	θ_4	l_4	0	

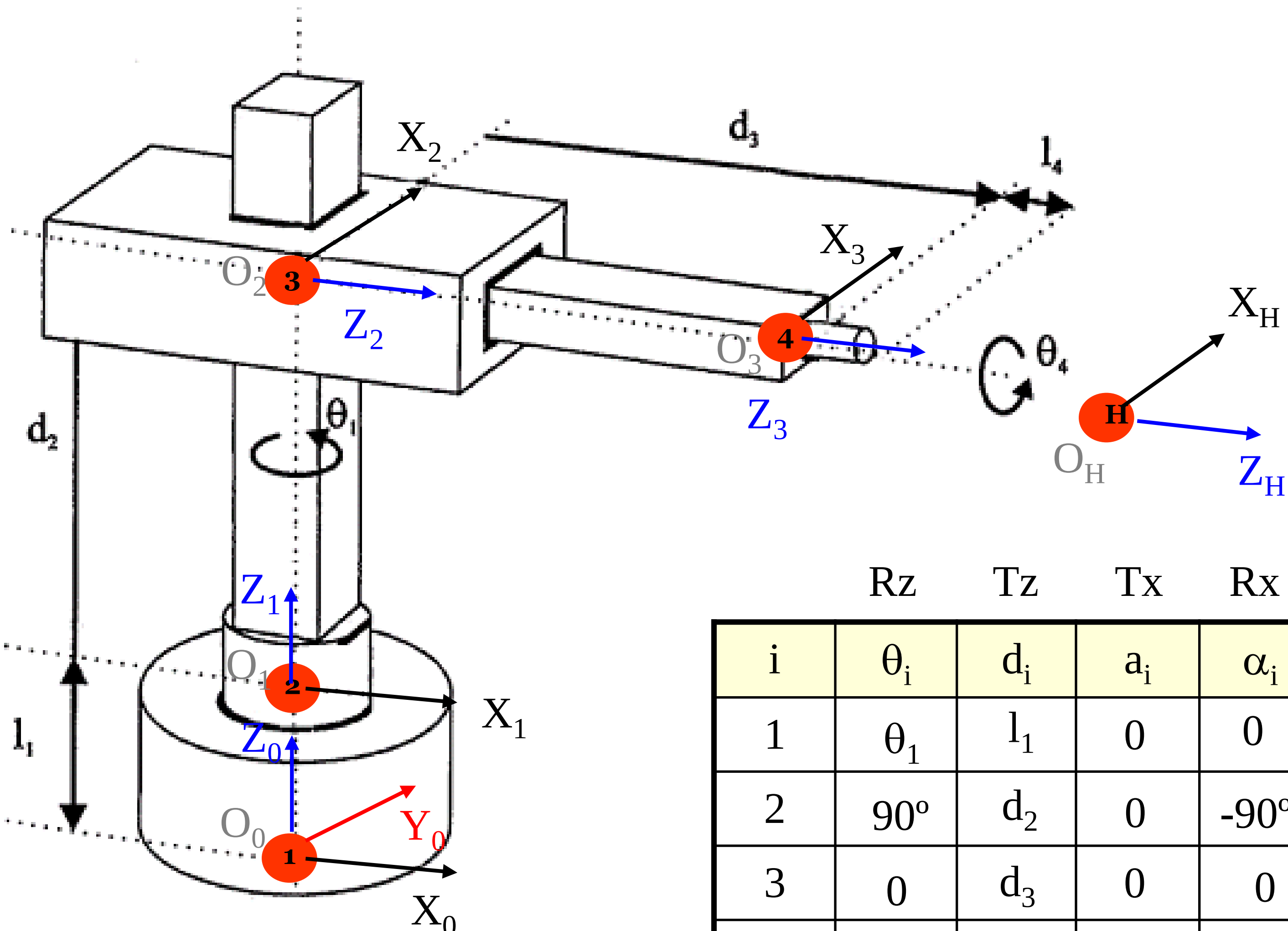
Table of D-H Parameters

Denavit-Hartenberg Algorithm

- **Calculate D-H Parameters**

α_i

Is the rotation over X_i required to align Z_{i-1} with Z_i



	Rz	Tz	Tx	Rx
i	θ_i	d_i	a_i	α_i
1	θ_1	l_1	0	0
2	90º	d_2	0	-90º
3	0	d_3	0	0
H	θ_4	l_4	0	0

Table of D-H Parameters

Note: Rotate z_{i-1}-axis about x_i-axis an angle of α_i to align z_{i-1}-axis with z_i-axis

Denavit-Hartenberg Algorithm

- **Calculate D-H Parameters**

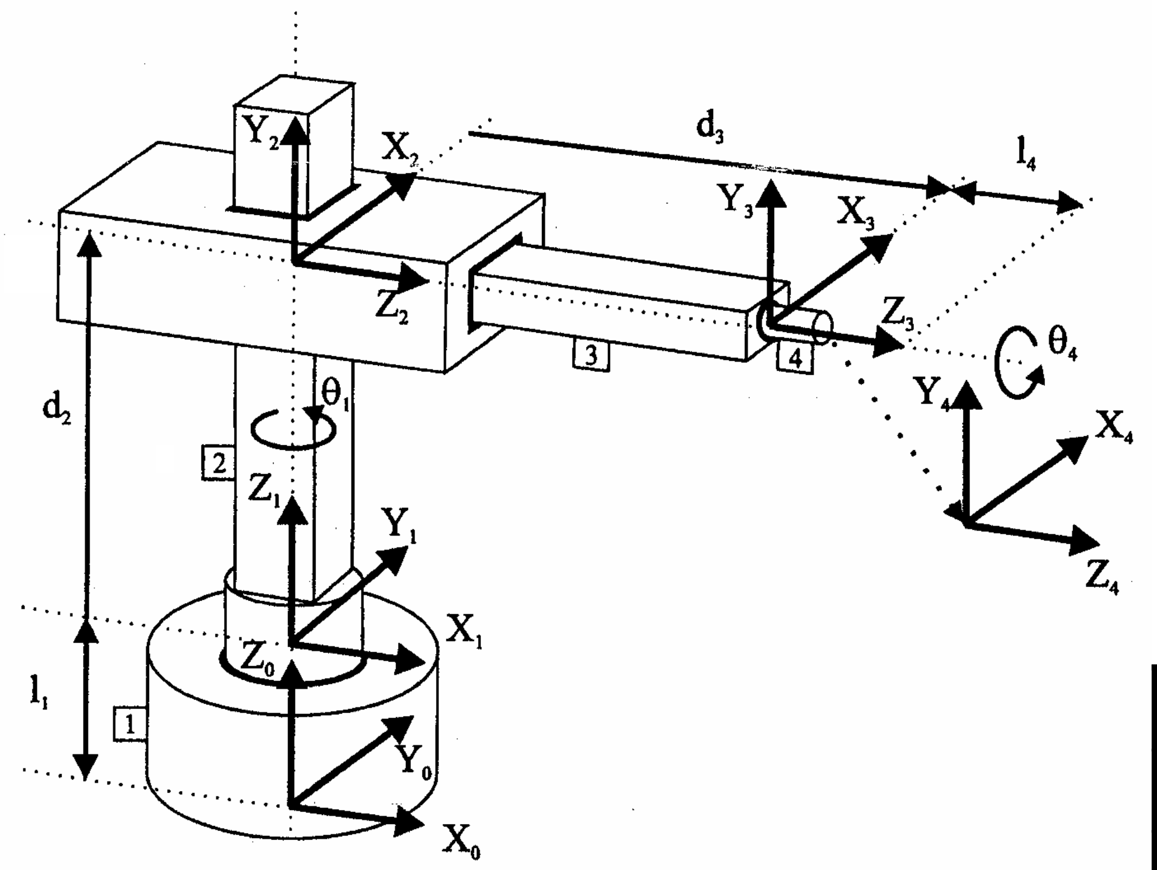


Table of D-H Parameters

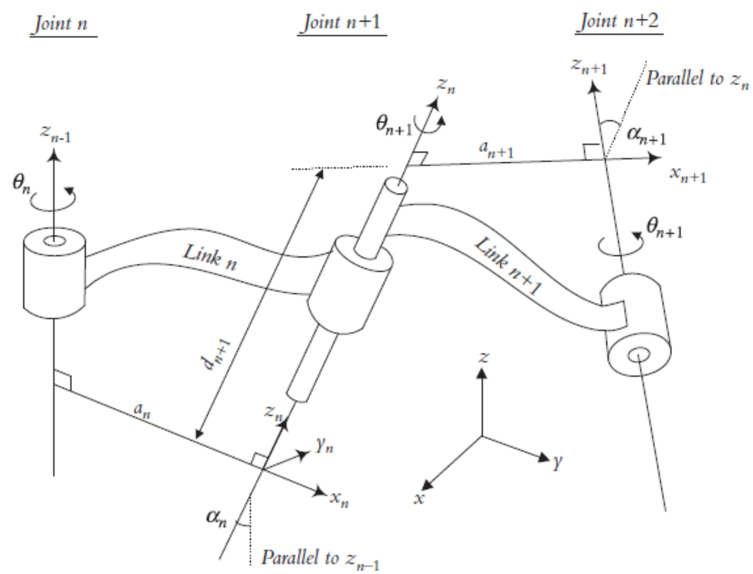
	Rz	Tz	Tx	Rx
i	θ_i	d_i	a_i	α_i
1	θ_1	l_1	0	0
2	90º	d_2	0	-90º
3	0	d_3	0	0
H	θ_4	l_4	0	0

Denavit-Hartenberg Algorithm

- **D-H Transformation Matrix**

D-H Transformation Matrix for adjacent coordinates frames, i and i-1

$$ {}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} $$



Total Transformation between the base of the robot and the hand is:

$$ {}^{R}T_{H} = {}^{0}A_{n} = {}^{0}A_{1} \cdot {}^{1}A_{2} \cdot {}^{2}A_{3} \cdot \ldots \cdot {}^{(n-1)}A_{n} $$

Denavit-Hartenberg Algorithm

- **D-H Transformation Matrix**

Partial Matrices

$$
{}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Table of D-H Parameters

	Rz	Tz	Tx	Rx
i	θ_i	d_i	a_i	α_i
1	θ_1	l_1	0	0
2	90º	d_2	0	-90º
3	0	d_3	0	0
H	θ_4	l_4	0	0

$$
{}^{0}A_1 = \begin{bmatrix} C\theta_1 & -S\theta_1 & 0 & 0 \\ S\theta_1 & C\theta_1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\qquad
{}^{1}A_2 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Denavit-Hartenberg Algorithm

- **D-H Transformation Matrix**

Partial Matrices

$$^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Table of D-H Parameters

	Rz	Tz	Tx	Rx
i	θ_i	d_i	a_i	α_i
1	θ_1	l_1	0	0
2	90º	d_2	0	-90º
3	0	d_3	0	0
H	θ_4	l_4	0	0

$$^{2}A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^{3}A_H = \begin{bmatrix} C\theta_4 & -S\theta_4 & 0 & 0 \\ S\theta_4 & C\theta_4 & 0 & 0 \\ 0 & 0 & 1 & l_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Denavit-Hartenberg Algorithm

- **D-H Transformation Matrix**

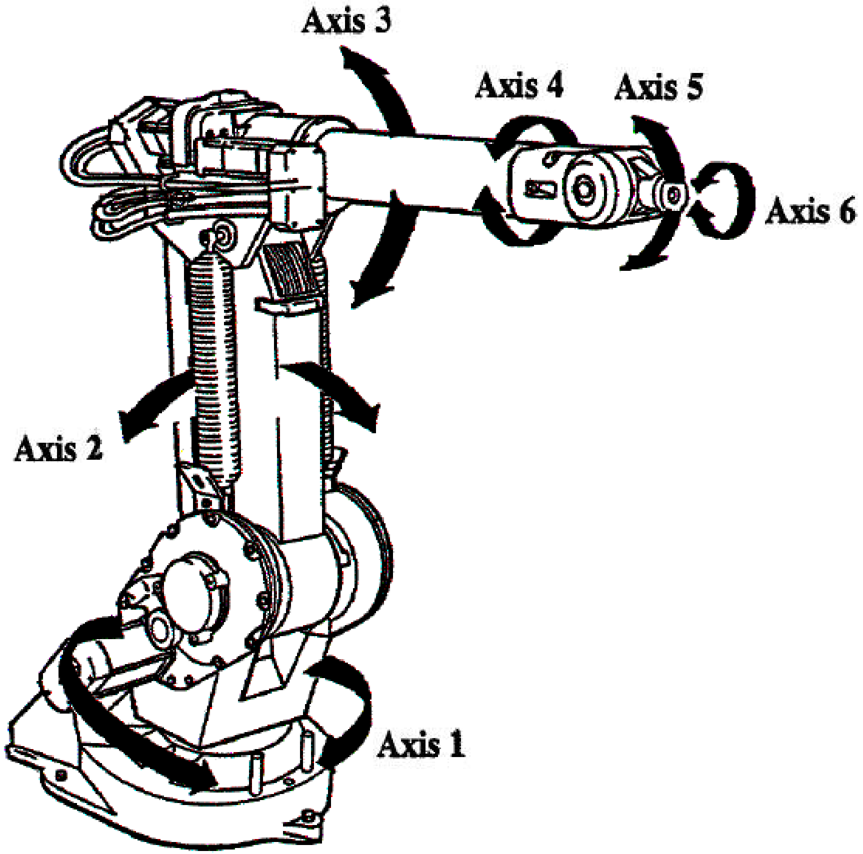
Total Transformation Matrix

$^{R}T_{H} = {}^{0}A_{4} = {}^{0}A_{1} \cdot {}^{1}A_{2} \cdot {}^{2}A_{3} \cdot {}^{3}A_{4}$

$$^{R}T_{H} = {}^{0}A_{H} = \begin{bmatrix} -S_1C_4 & S_1S_4 & C_1 & C_1(d_3+l_4) \\ C_1C_4 & -C_1S_4 & S_1 & S_1(d_3+l_4) \\ S_4 & C_4 & 0 & d_2+l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Denavit-Hartenberg Algorithm

- **ABB IRB 1400**



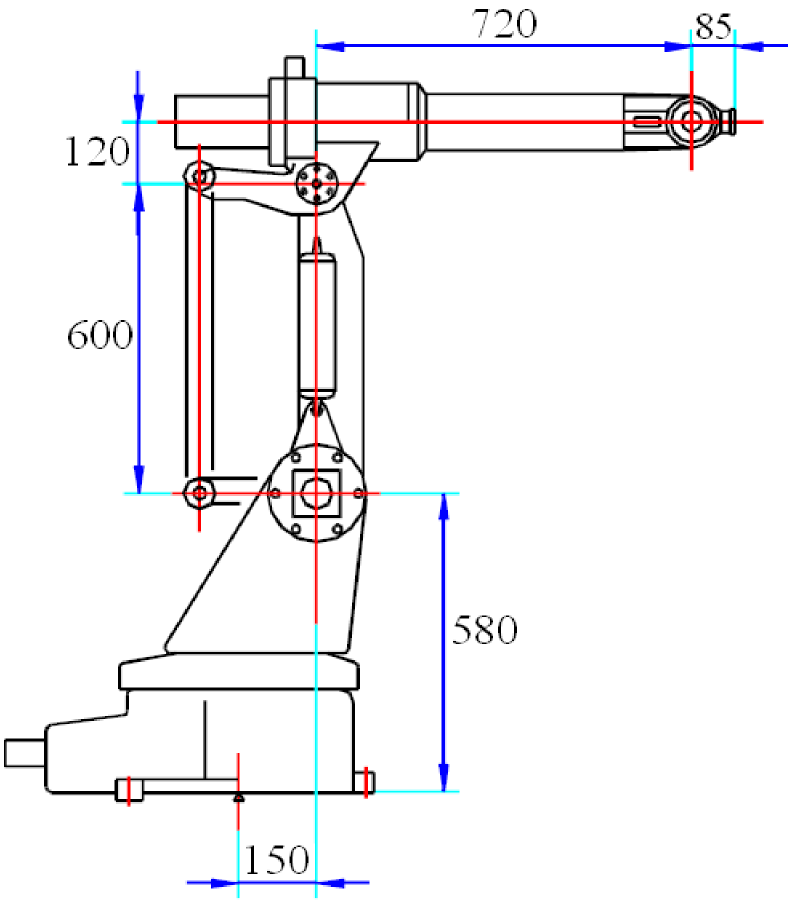


ABB IRB 1400

Denavit-Hartenberg Algorithm

- **ABB IRB 1400**

For this ABB IRB 1400 robot:

- Assign the coordinate frames based on the D-H representation;
- Fill out the parameters table;
- Write all the A matrices;
- Write the total transformation matrix in terms of the A matrices.

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

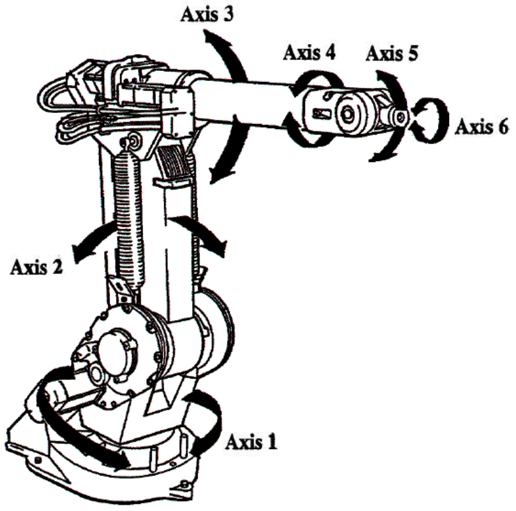
Assigning Frame-0:

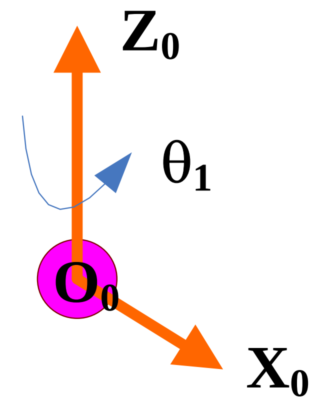
The given first frame will be considered as the base frame.

Z_0 is the axis of actuation for joint 1. Its direction will be as shown in the figure.

We may choose the origin O_0 of the base frame to be any point on Z_0.

We then choose X_0 (and Y_0 if you want) in any convenient manner as long as the resulting frame is right-handed. In this problem, the direction of X_0 is already given. This sets up frame 0.





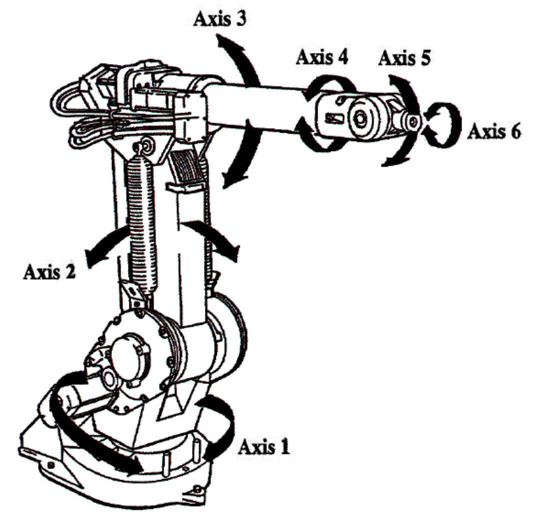
Denavit-Hartenberg Algorithm

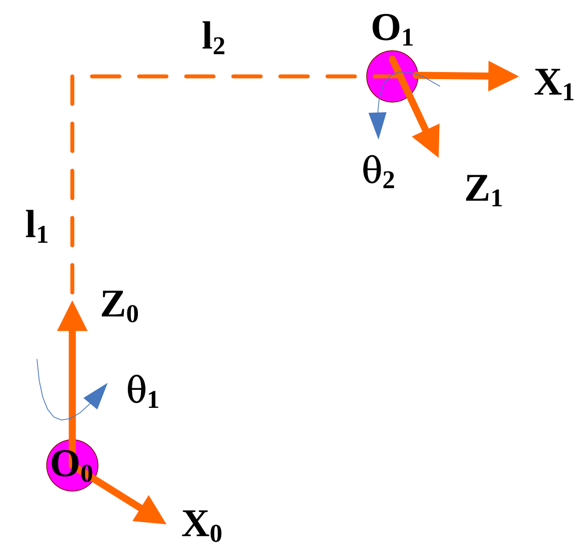
- **ABB IRB 1400: Frame Assignment**

Assigning Frame-1:

Z_1 is the axis of actuation for joint 2. Its direction will be as shown in the figure.

Z_1 and Z_0 are parallel. In this case, there are infinitely many common normals between them. In this case we are free to choose the **origin O_1 anywhere along Z_1**. One often chooses O_1 to simplify the resulting equations.



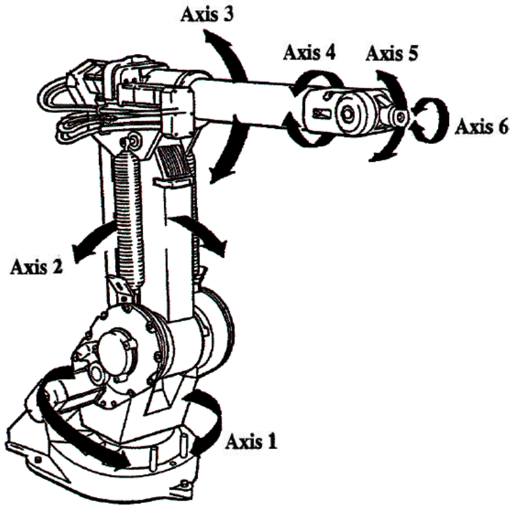


Denavit-Hartenberg Algorithm

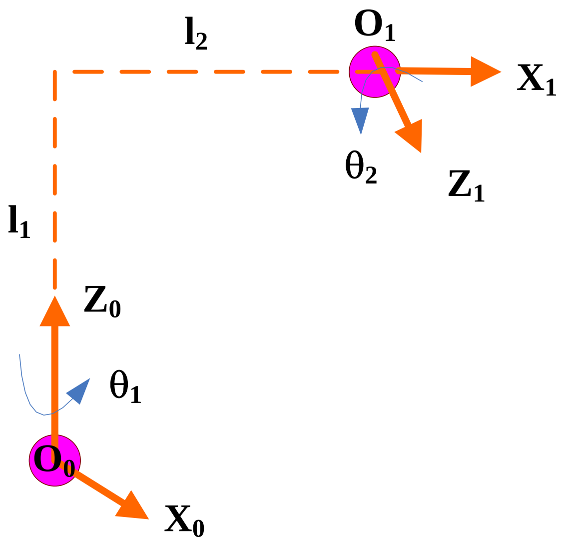
- **ABB IRB 1400: Frame Assignment**

Assigning Frame-1:

The **axis X_1** is then chosen either to be **directed from O_1 toward Z_0**, along the common normal, or as the **opposite** of this vector.



A common method for choosing O_1 is to choose the normal that passes through O_0 as the X_1 axis; O_1 is then the point at which this normal intersects Z_1. Following these rules, O_1 and X_1 are chosen as shown in the figure. This sets up frame 1.



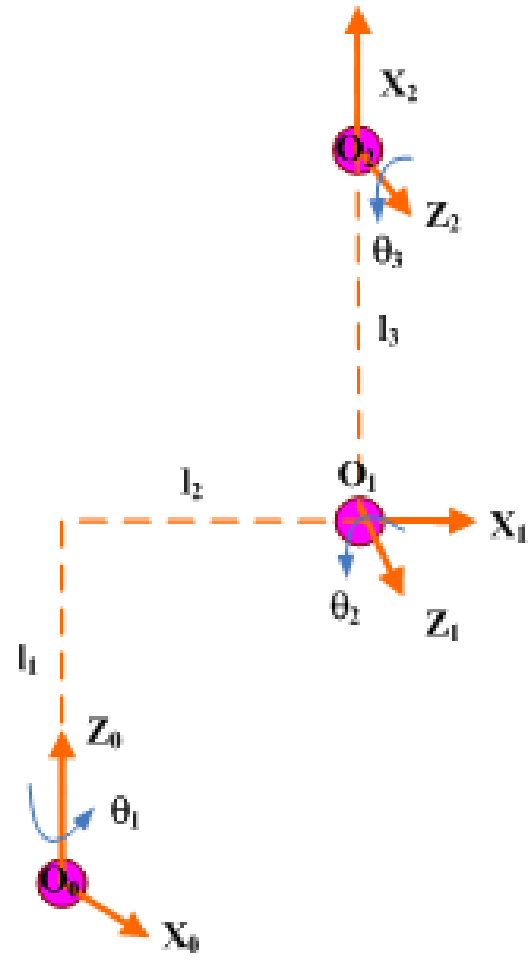
Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

Assigning Frame-2:

Z_2 is the axis of actuation for joint 3. Its direction will be as shown in the figure.

Z_2 and Z_1 are parallel. In this case, there are infinitely many common normals between them. In this case we are free to choose the origin O_2 anywhere along Z_2. One often chooses O_2 to simplify the resulting equations.



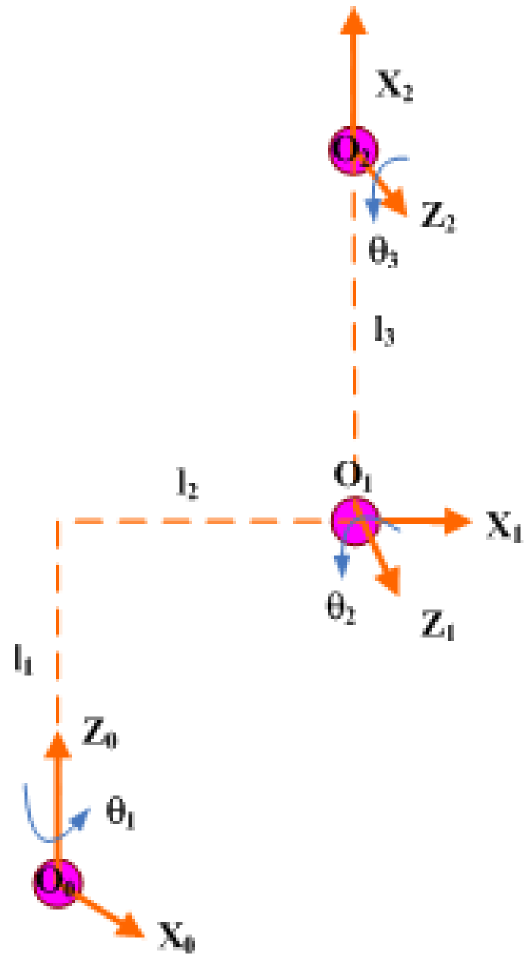
Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

Assigning Frame-2:

The axis X_2 is then chosen either to be directed from O_2 toward Z_1, along the common normal, or as the opposite of this vector.

A common method for choosing O_2 is to choose the normal that passes through O_1 as the X_2 axis; O_2 is then the point at which this normal intersects Z_2. Following these rules, O_2 and X_2 are chosen as shown in the figure. This sets up frame 2.



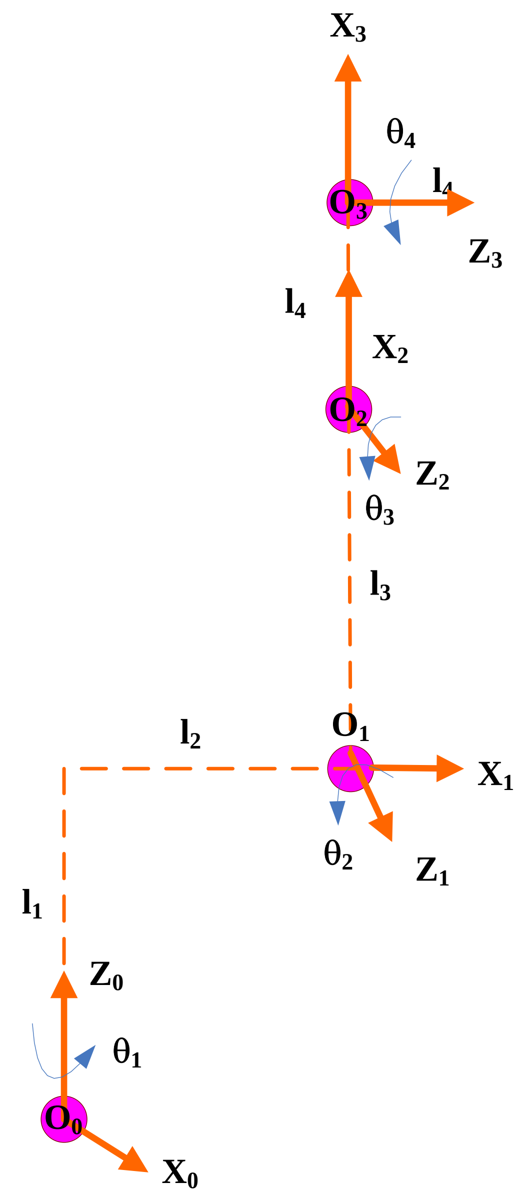
Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

Assigning Frame-3:

Z_3 is the axis of actuation for joint 4. Its direction will be as shown in the figure.

Z_3 and Z_2 are parallel. In this case, there are infinitely many common normals between them. In this case we are free to choose the **origin O_3 anywhere along Z_3**. One often chooses O_3 to simplify the resulting equations.



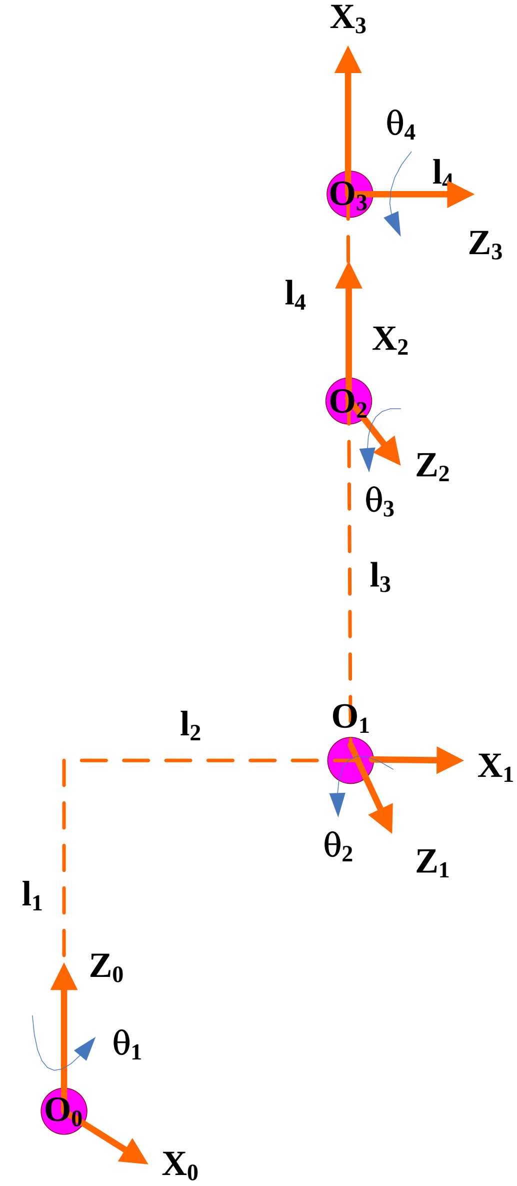
Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

Assigning Frame-3:

The axis X_3 is then chosen either to be directed from O_3 toward Z_2, along the common normal, or as the opposite of this vector.

A common method for choosing O_3 is to choose the normal that passes through O_2 as the X_3 axis; O_3 is then the point at which this normal intersects Z_3. Following these rules, O_3 and X_3 are chosen as shown in the figure. This sets up frame 3.



Denavit-Hartenberg Algorithm

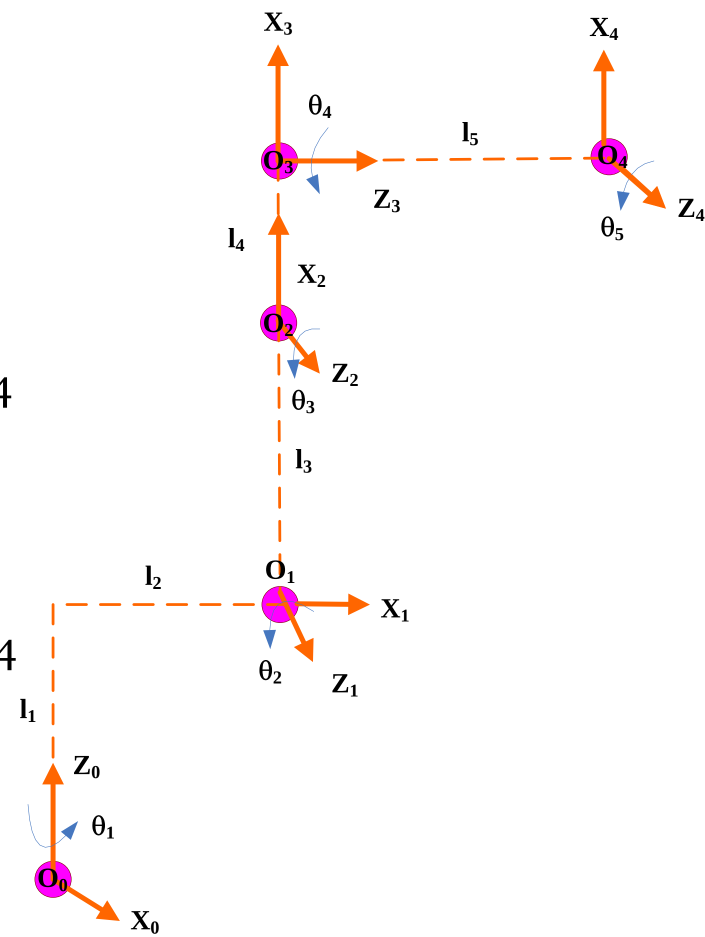
- **ABB IRB 1400: Frame Assignment**

Assigning Frame-4:

Z_4 is the axis of actuation for joint 5. Its direction will be as shown in the figure.

Z_4 and Z_3 intersect. In this case, X_4 is chosen normal to the plane formed by Z_4 and Z_3. The positive direction of X_4 is arbitrary.

The most natural choice for the origin O_4 in this case is at the point of intersection of Z_4 and Z_3. However, any convenient point along the axis Z_4 suffices. This sets up frame 4.



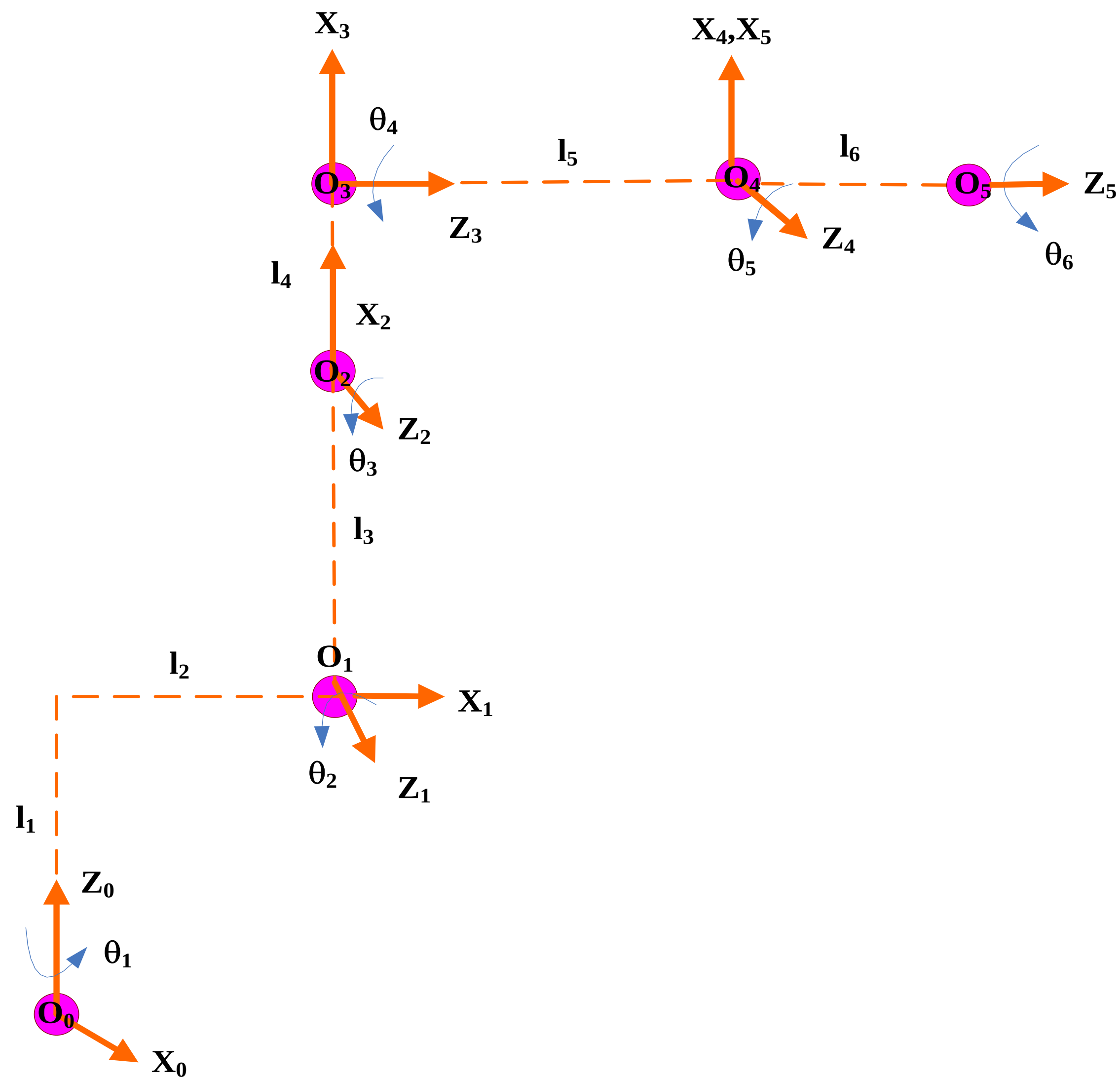
Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

Assigning Frame-5:

Z_5 is the axis of actuation for joint 6. Its direction will be as shown in the figure.

Z_5 and Z_4 intersect. In this case, X_5 is chosen normal to the plane formed by Z_5 and Z_4. The positive direction of X_5 is arbitrary.

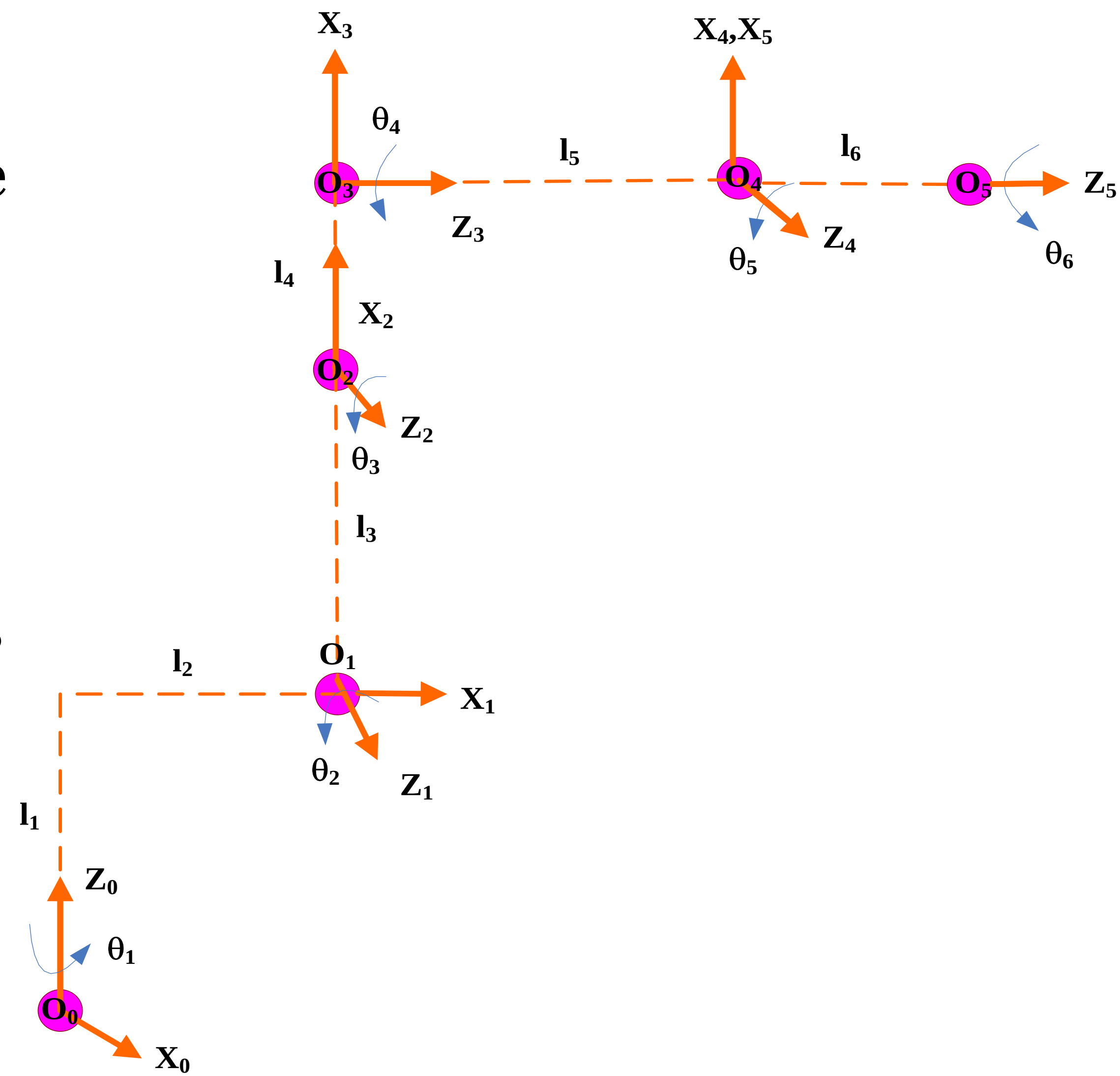


Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

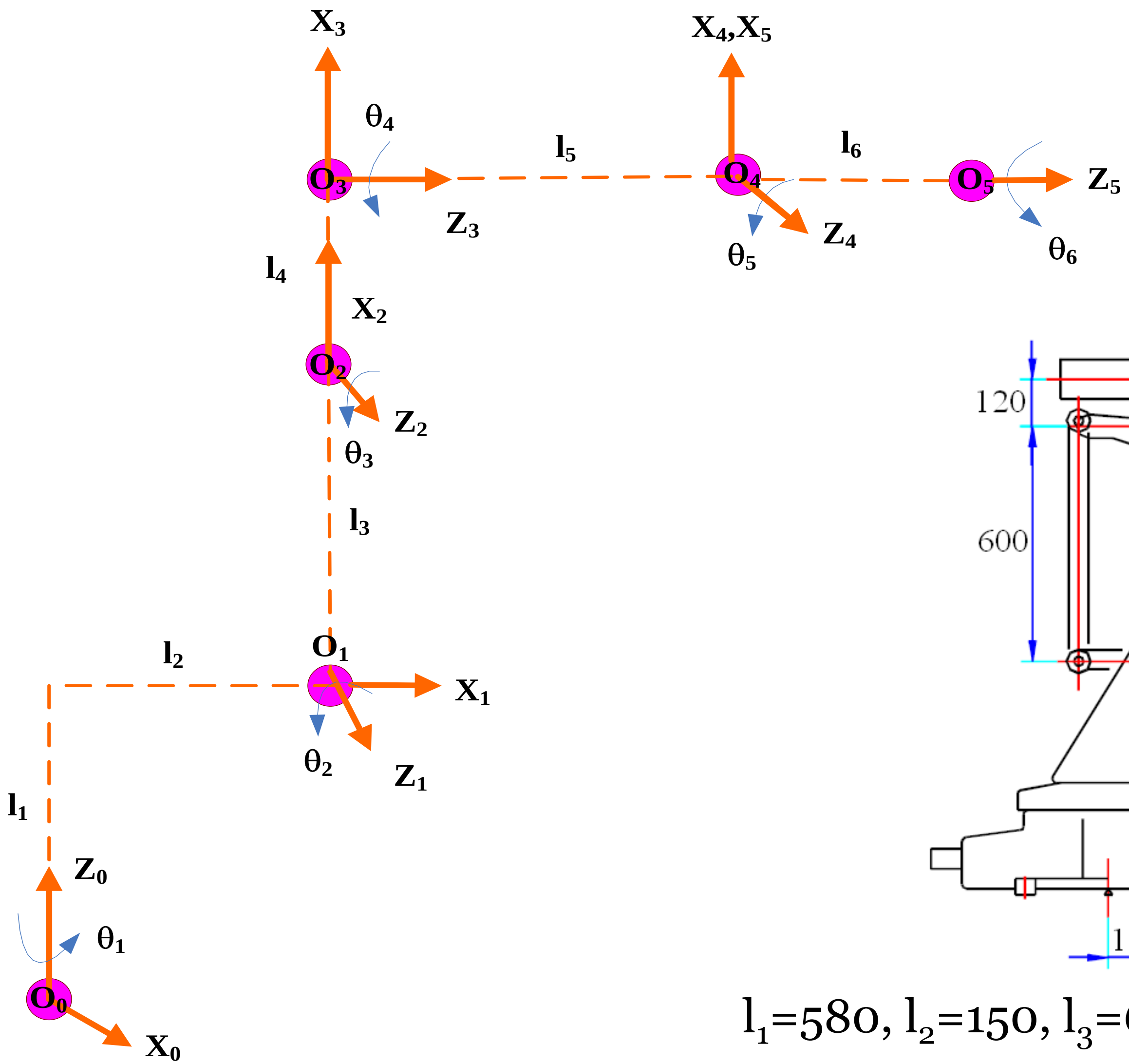
Assigning Frame-5:

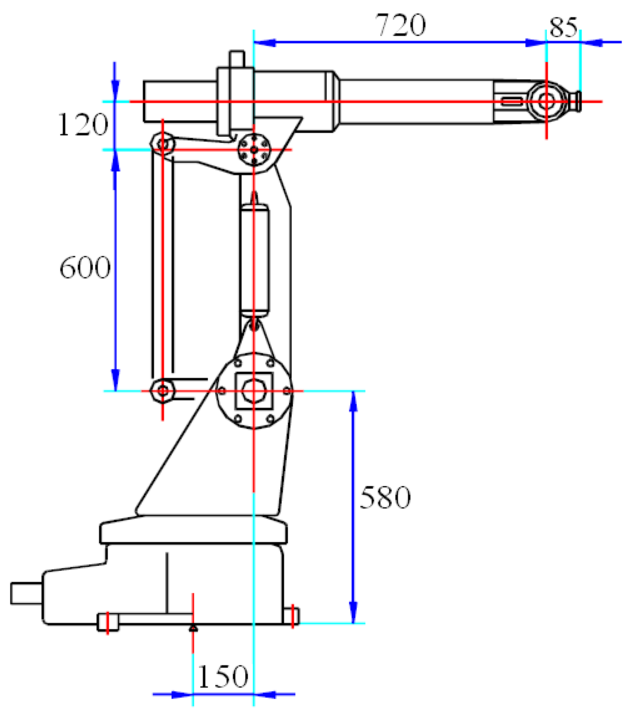
The most natural choice for the origin O_5 in this case is at the point of intersection of Z_5 and Z_4. However, any convenient point along the axis Z_5 suffices. This sets up frame 5.



Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Frame Assignment**

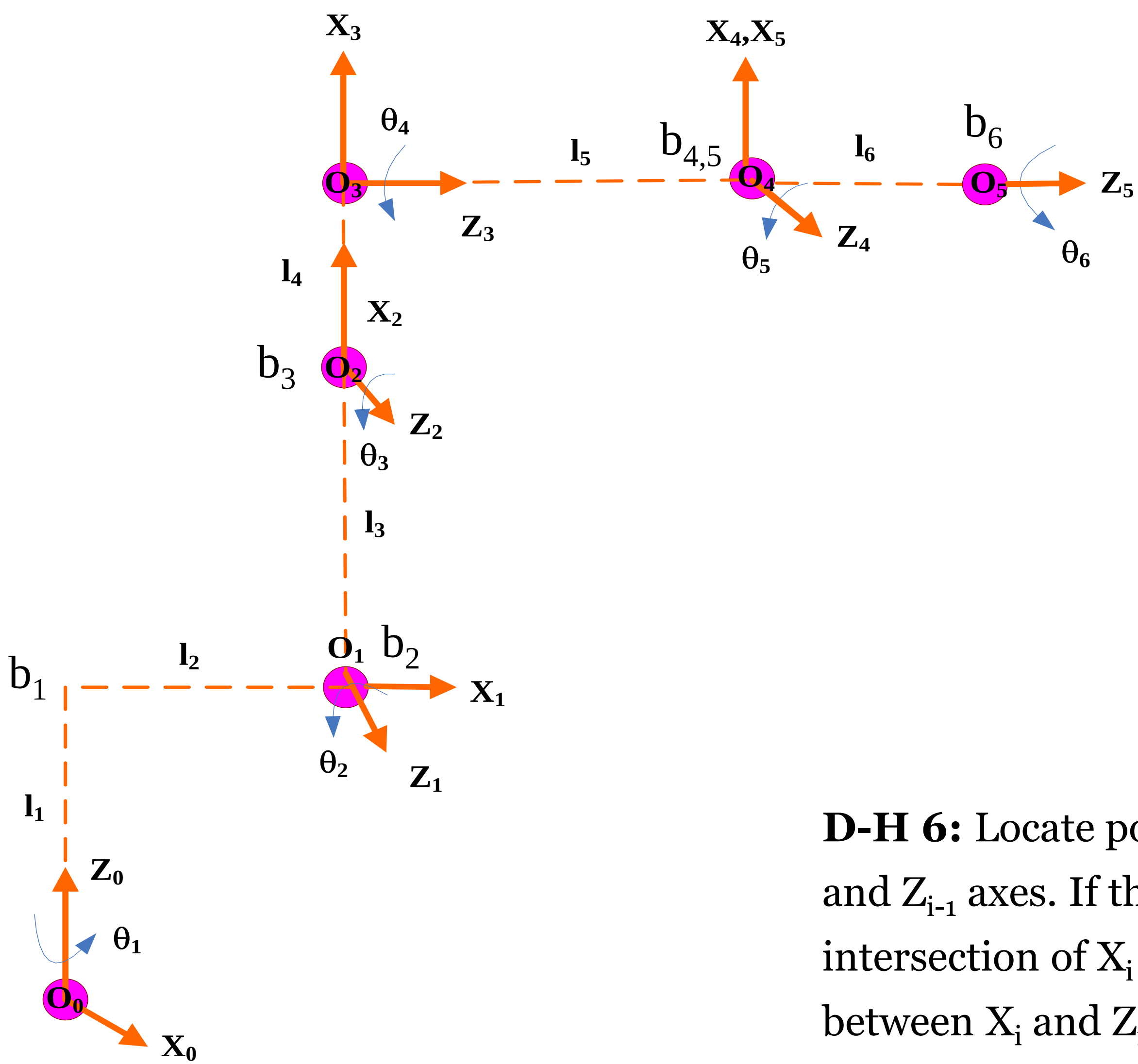




$l_1=580,\ l_2=150,\ l_3=600,\ l_4=120,\ l_5=720,\ l_6=85$

Denavit-Hartenberg Algorithm

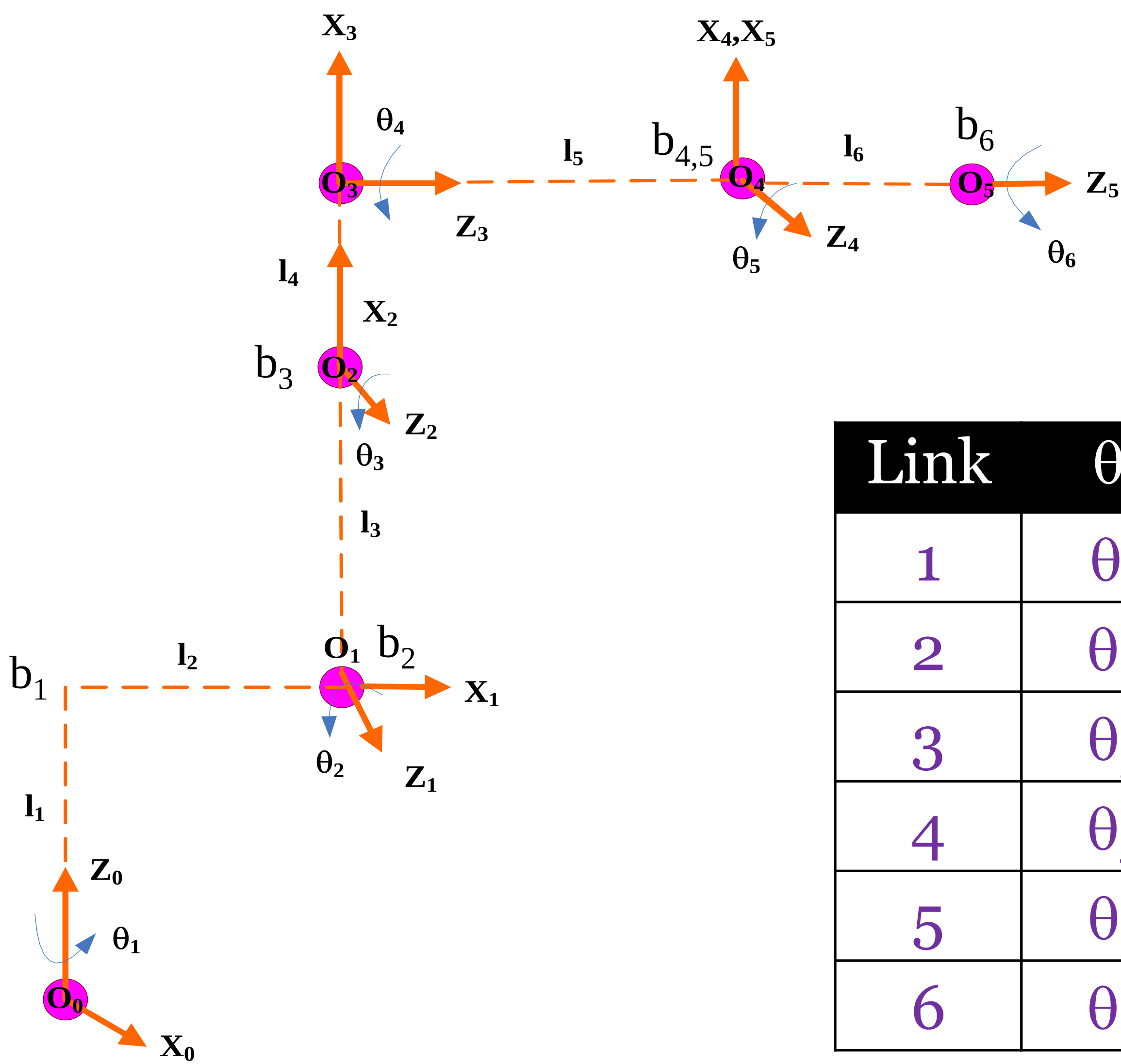
- **ABB IRB 1400: Frame Assignment**



D-H 6: Locate point b_i at the intersection of X_i and Z_{i-1} axes. If they do not intersect, use the intersection of X_i with a common normal between X_i and Z_{i-1}.

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: D-H Parameter Table**



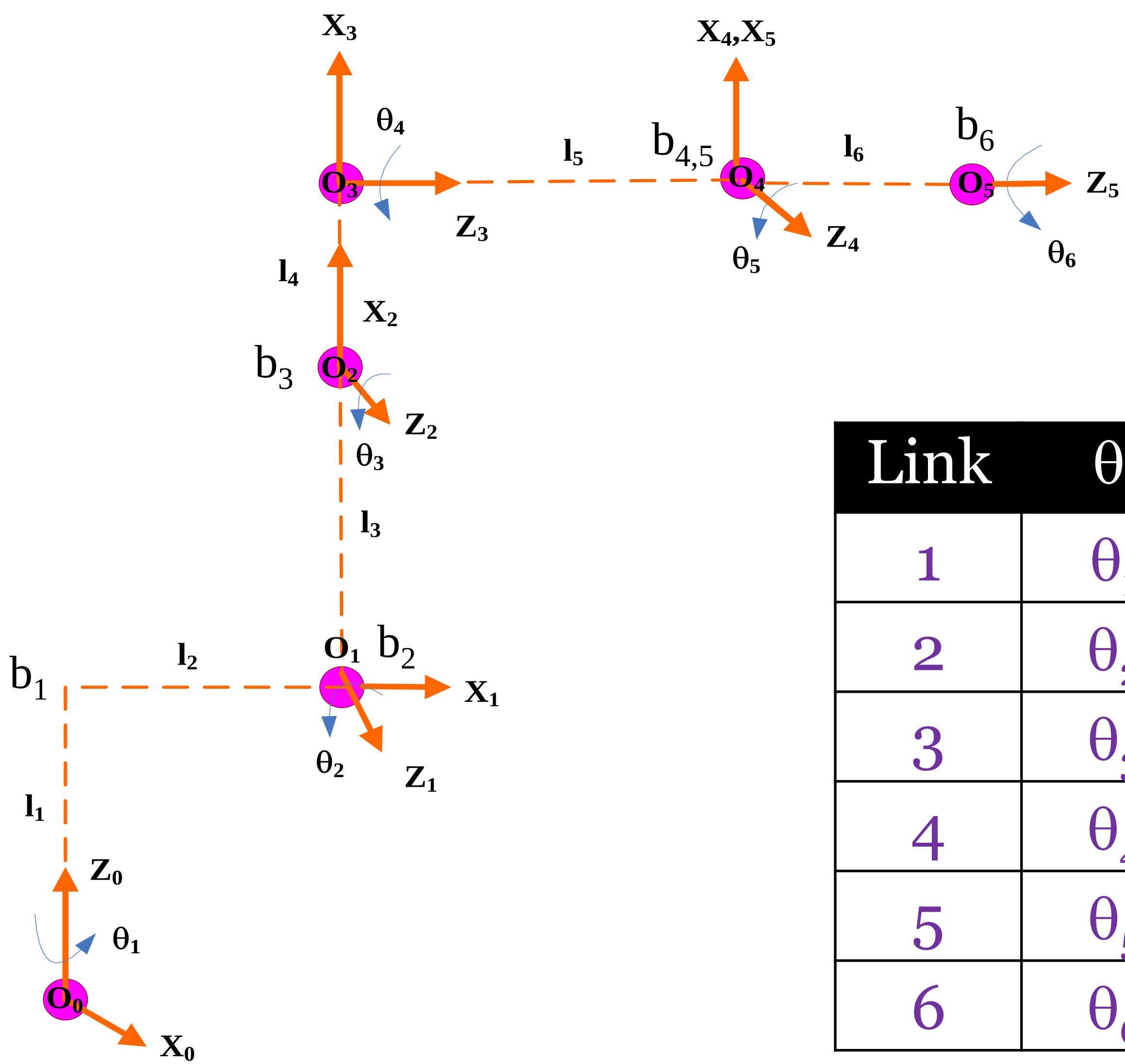
θ_i

Is the rotation over Z_{i-1} required to make X_{i-1} parallel to X_i.

Link	θ_i	d_i	a_i	α_i
1	θ_1			
2	θ_2			
3	θ_3			
4	θ_4			
5	θ_5			
6	θ_6			

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: D-H Parameter Table**



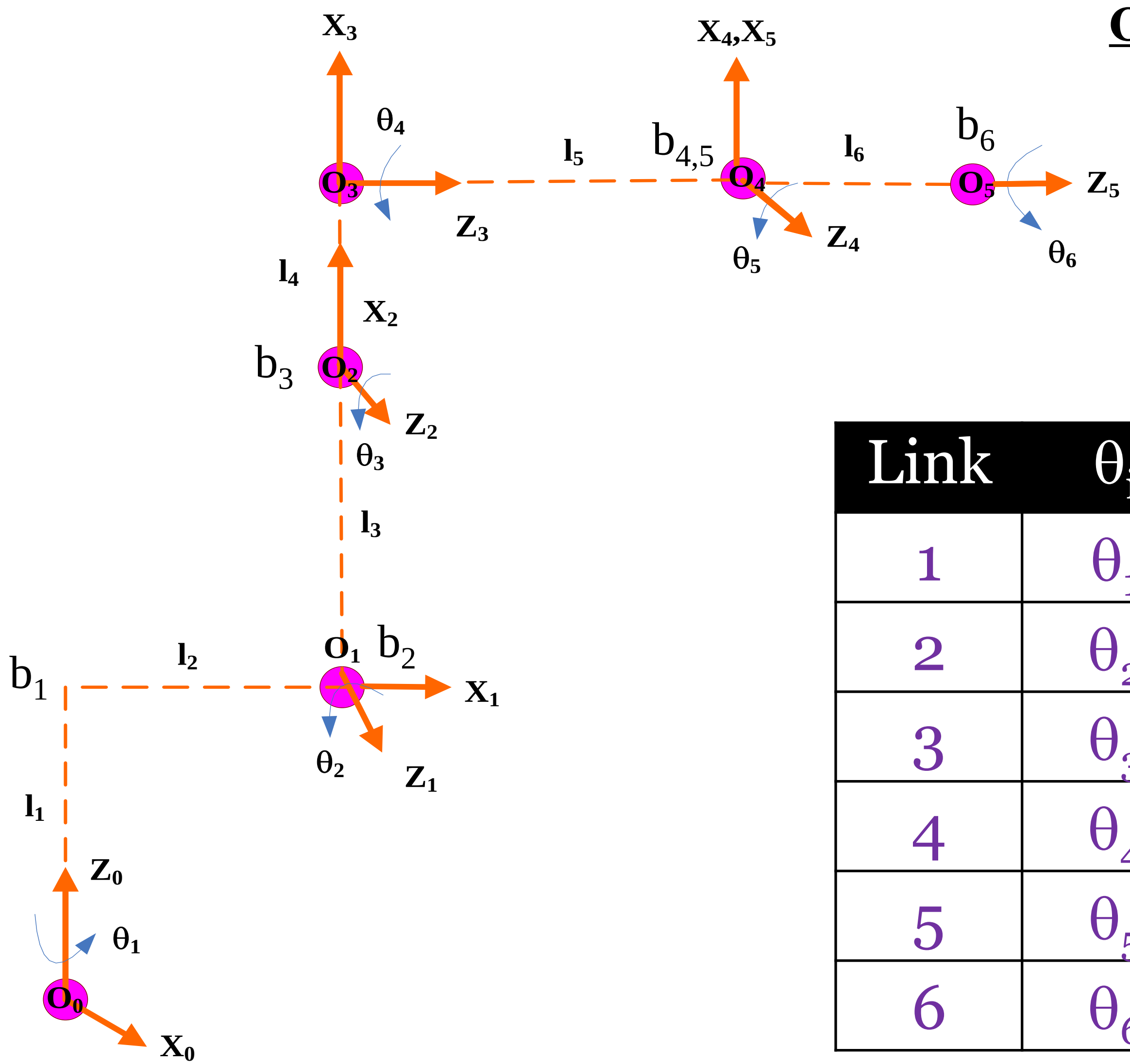
d_i

The distance from the origin of the frame O_{i-1} to the point b_i measured along Z_{i-1}

Link	θ_i	d_i	a_i	α_i
1	θ_1	l_1		
2	θ_2	0		
3	θ_3	0		
4	θ_4	l_5		
5	θ_5	0		
6	θ_6	l_6		

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: D-H Parameter Table**



<u>OR: if you don't want to use b_i</u>

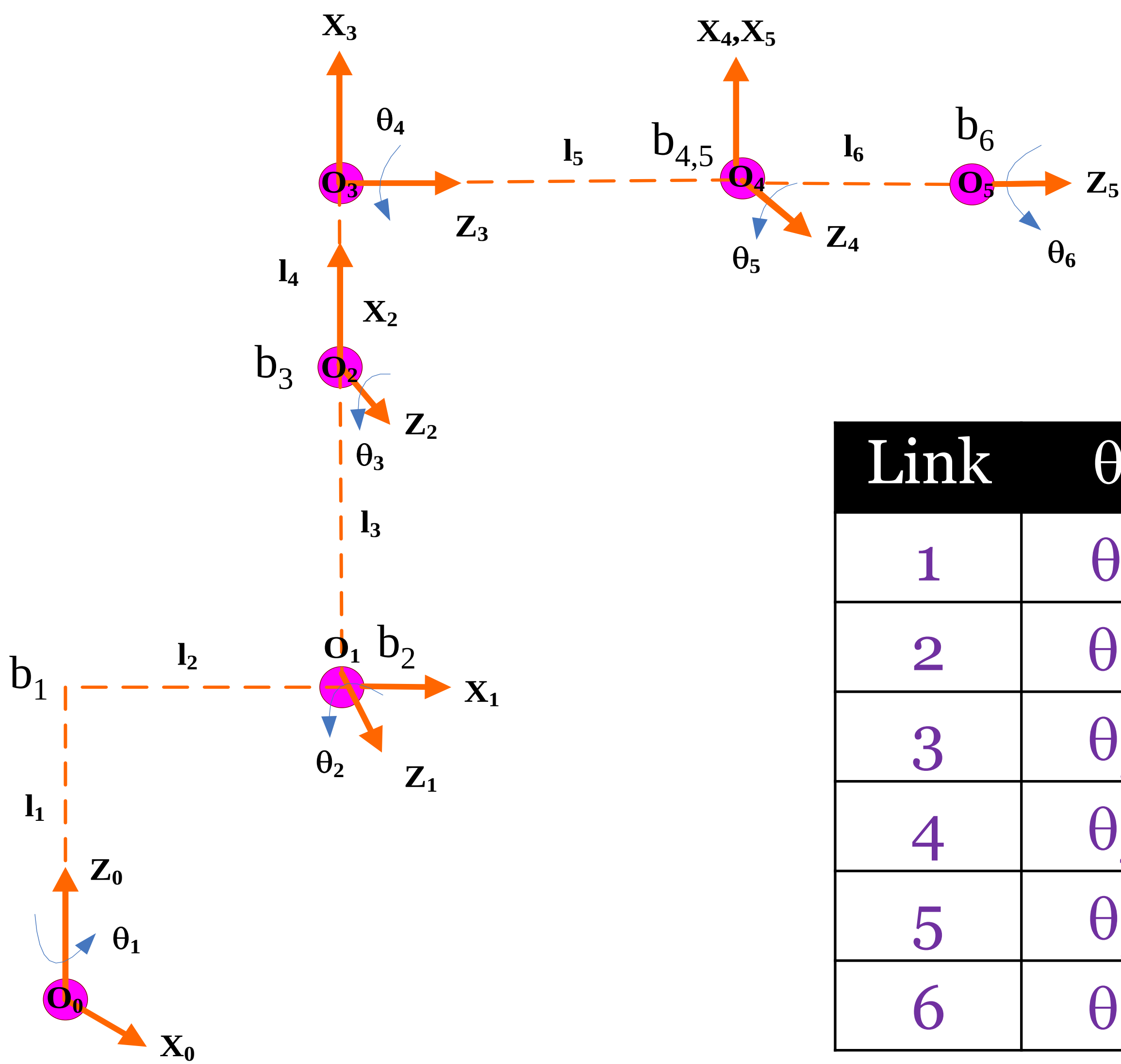
d_i

The translation over Z_{i-1} required to make X_{i-1} and X_i aligned

Link	θ_i	d_i	a_i	α_i
1	θ_1	l_1		
2	θ_2	0		
3	θ_3	0		
4	θ_4	l_5		
5	θ_5	0		
6	θ_6	l_6		

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: D-H Parameter Table**



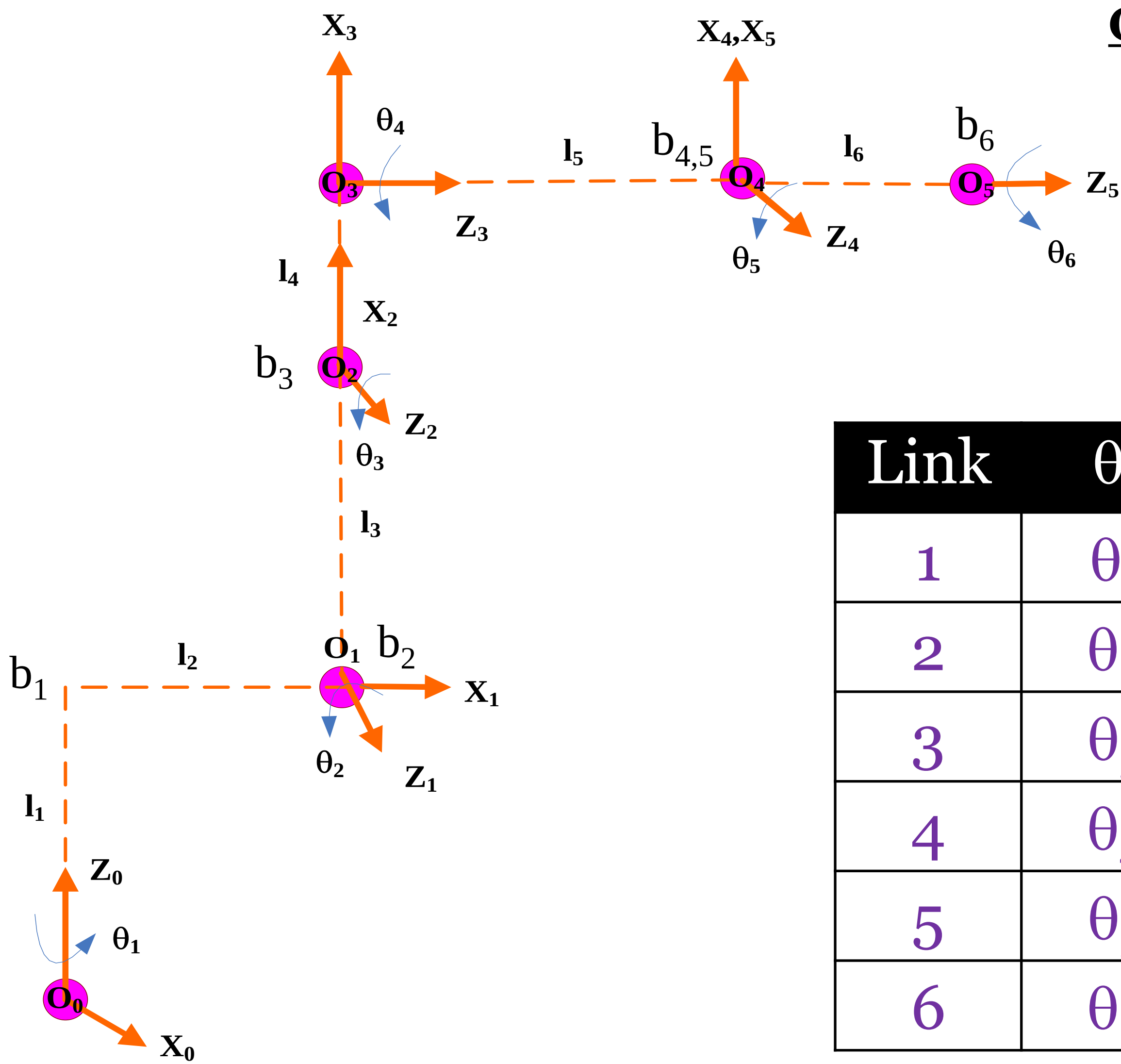
a_i

The distance from point b_i to the origin of frame O_i measured along X_i

Link	θ_i	d_i	a_i	α_i
1	θ_1	l_1	l_2	
2	θ_2	0	l_3	
3	θ_3	0	l_4	
4	θ_4	l_5	0	
5	θ_5	0	0	
6	θ_6	l_6	0	

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: D-H Parameter Table**



OR: if you don't want to use b_i

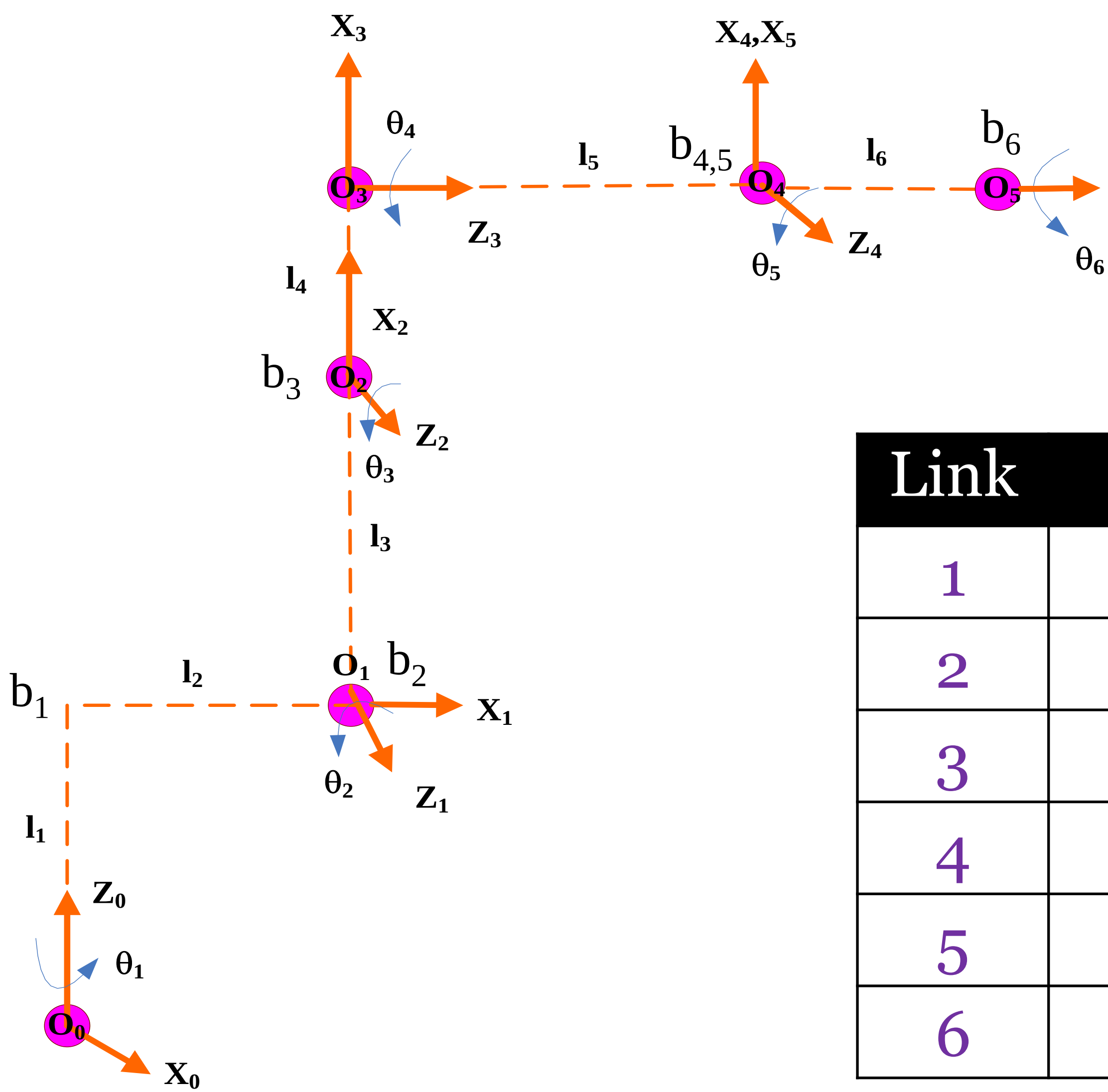
a_i

Is the translation over X_i required to make S_{i-1} and S_i coincided

Link	θ_i	d_i	a_i	α_i
1	θ_1	l_1	l_2	
2	θ_2	0	l_3	
3	θ_3	0	l_4	
4	θ_4	l_5	0	
5	θ_5	0	0	
6	θ_6	l_6	0	

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: D-H Parameter Table**



α_i

Is the rotation over X_i required to align Z_{i-1} with Z_i

Link	θ_i	d_i	a_i	α_i
1	θ_1	l_1	l_2	$+90^o$
2	θ_2	0	l_3	0
3	θ_3	0	l_4	$+90^o$
4	θ_4	l_5	0	-90^o
5	θ_5	0	0	$+90^o$
6	θ_6	l_6	0	0

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Partial/Adjacent Matrices**

$$
{}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{0}A_1 = \begin{bmatrix} C_1 & 0 & S_1 & l_2 C_1 \\ S_1 & 0 & -C_1 & l_2 S_1 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{1}A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_3 C_2 \\ S_2 & C_2 & 0 & l_3 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Denavit-Hartenberg Algorithm

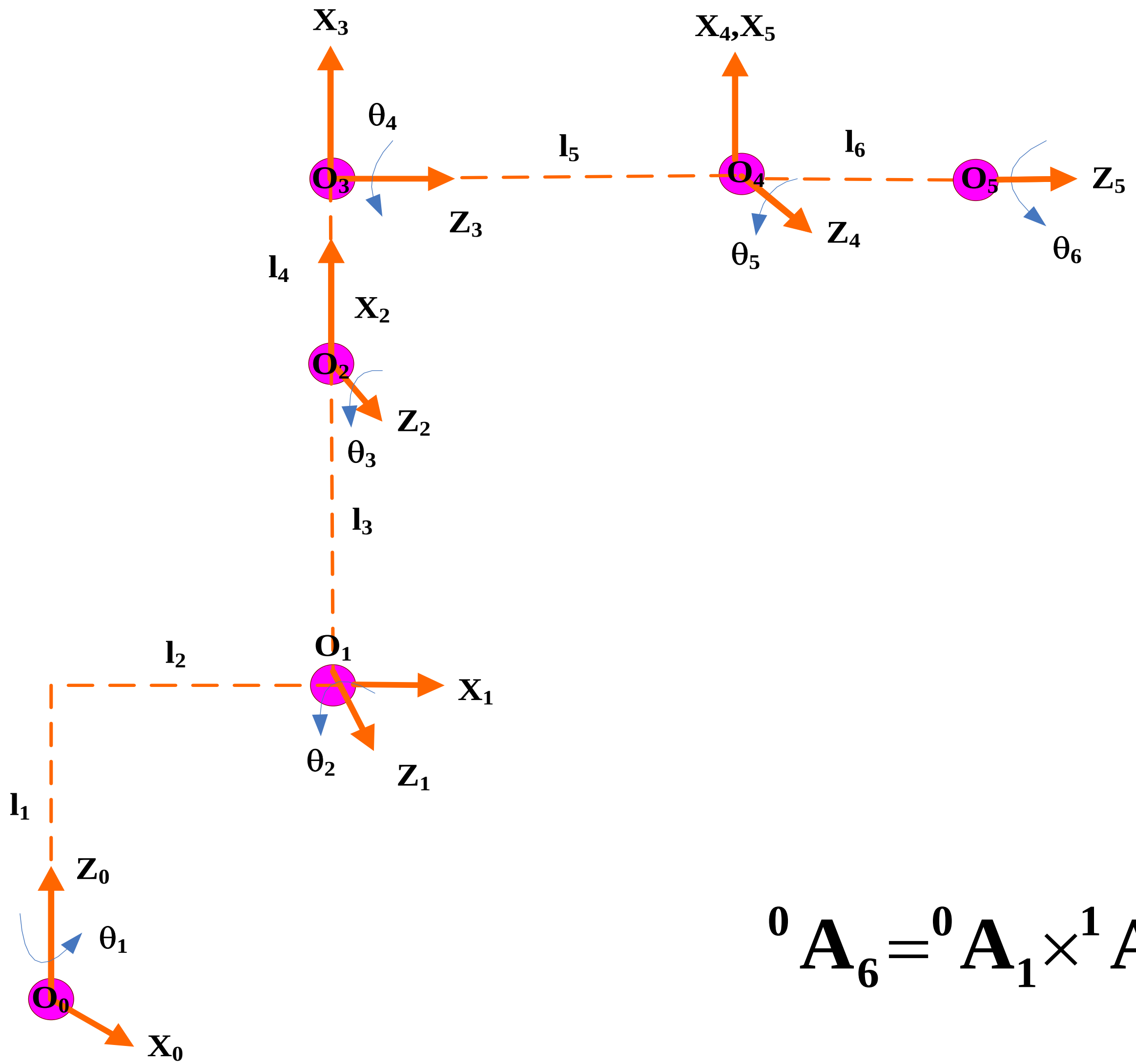
- **ABB IRB 1400: Partial/Adjacent Matrices**

$$^{4}A_{5} = \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^{2}A_{3} = \begin{bmatrix} C_3 & 0 & S_3 & l_4C_3 \\ S_3 & 0 & -C_3 & l_4S_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$^{3}A_{4} = \begin{bmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & 0 & C_4 & 0 \\ 0 & -1 & 0 & l_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad ^{5}A_{6} = \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & l_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Total Transformation Matrix**



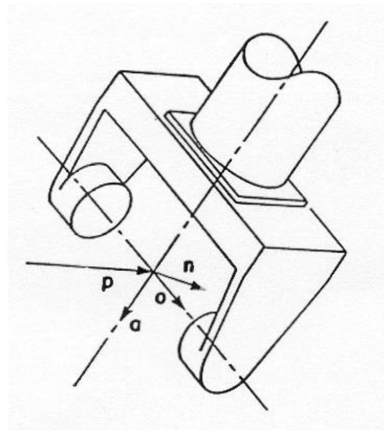
$$^{0}A_{6} = {}^{0}A_{1} \times {}^{1}A_{2} \times {}^{2}A_{3} \times {}^{3}A_{4} \times {}^{4}A_{5} \times {}^{5}A_{6}$$

Denavit-Hartenberg Algorithm

- **ABB IRB 1400: Forward Kinematics Solution**

Given:

θ_1
θ_2
θ_3
θ_4
θ_5
θ_6



Forward Kinematics

Required:

$p_x=f_x\ (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$p_y=f_y\ (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$p_z=f_z\ (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$n=f_n\ (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$o=f_o\ (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$
$a=f_a\ (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$

$$ {}^{0}A_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_x & a_x & p_y \\ n_z & o_x & a_x & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Outline

- Forward Kinematics
- Denavit-Hartenberg Algorithm
- **Summary**

Summary

- The Denavit-Hartenberg Representation has become the standard way of representing robots and modelling their motions. The method begins with a systematic approach to assigning and labelling an orthonormal (x,y,z) coordinate system to each robot joint. It is then possible to relate one joint to the next and ultimately to assemble a complete representation of a robot's geometry.
- In assigning the x- and z-axes, you may choose either direction along the chosen line of action. Ultimately, the result of the total transformation will be the same. However, your individual matrices and parameters are similarly affected.
- It is acceptable to use additional frames to make things easier to follow. However, you may not have any fewer or more unknown variables than you have joints.

Summary

- We can assign coordinate frames to all joints, with the following exceptions:
  - ◇ If two z-axes **are parallel**, there are an infinite number of common normals between them. We will pick the common normal that is colinear with the common normal of the previous joint. This will simplify the model.
  - ◇ If the z-axes of two successive joints **are intersecting**, there is no common normal between them (or it has a zero length). We will assign the x-axis along a line perpendicular to the plane formed by the two axes. This means that the common normal is a line perpendicular to the plane containing the two z-axes, which is the equivalent of picking the direction of the cross-product of the two z-axes. This also simplifies the model.

Summary

- The fundamental problem of D-H representation is that since all motions are about the x- and z-axes, the method cannot represent any motion about the y-axis.
- Therefore, if there is any motion about the y-axis, the method will fail. This occurs in a number of circumstances. For example, suppose two joint axes that are supposed to be parallel are assembled with a slight deviation. The small angle between the two axes will require a motion about the y-axis. Since all real industrial robots have some degree of inaccuracy in their manufacture, their inaccuracy cannot be modeled with the D-H representation.