Robot Arm Kinematics

Prof. Gerardo Flores

Sep 10, 2024

Robotics and Automation course, TAMIU

Describing a robot arm

Kinematics is the branch of mechanics that studies the motion of a body, or a system of bodies, without consideration given to its mass or the forces acting on it.

link

joint

j^{th} character represents the type of joint

Puma robotic arm with revolute **joints**. RRRRRR

https://youtu.be/k86D_kEtFTU?si=aOnIauINe_b2YMcM

The Stanford arm with one prismatic **joint**. RRPRRR

https://youtu.be/O1oJzUSlTeY?si=G4c35PffXhkWqxGr

Describing a robot arm

What is the goal?

- The objective is to represent the position and orientation of any part of the robot, typically the gripper (or end effector), relative to a reference frame, usually the inertial or fixed frame.
- To achieve this, we must:
    1. Define multiple coordinate frames along various parts of the robot.
    2. Establish the homogeneous transformations between these frames.
    3. Calculate the homogeneous transformation that relates the inertial frame to the end effector's tool frame.

Denavit-Hartenberg notation

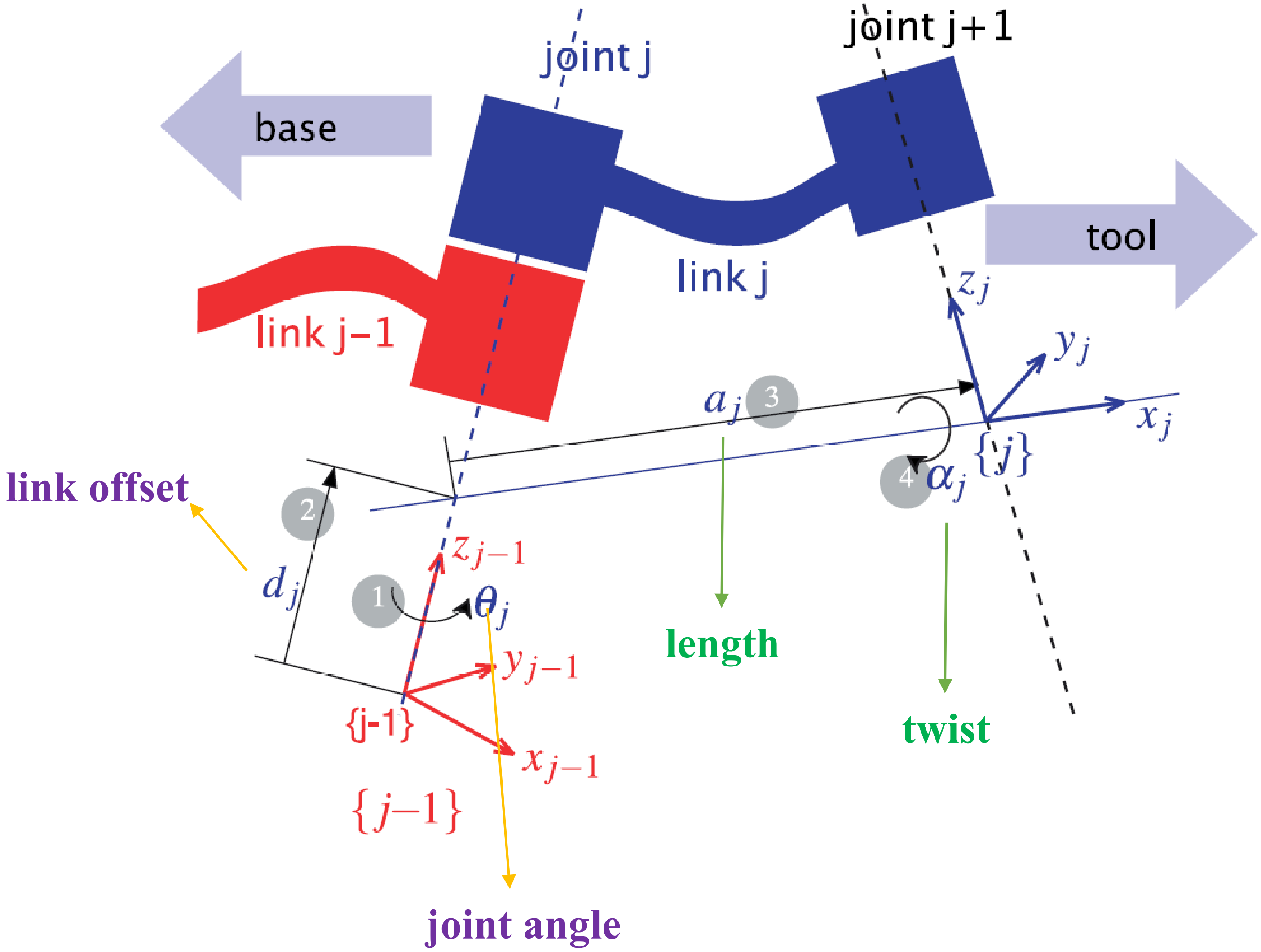
A systematic way of describing **the geometry** of a serial chain of links and joints was proposed by Denavit and Hartenberg in 1955 and is known today as *Denavit-Hartenberg notation*.

For a manipulator with ***N* joints** numbered from **1 to *N***, there are $N+1$ links, numbered from 0 to *N*.

Joints → 1 to N
Links → 0 to N

Link 0 is the base of the manipulator and link *N* carries the end effector.

Definition of standard Denavit and Hartenberg link parameters

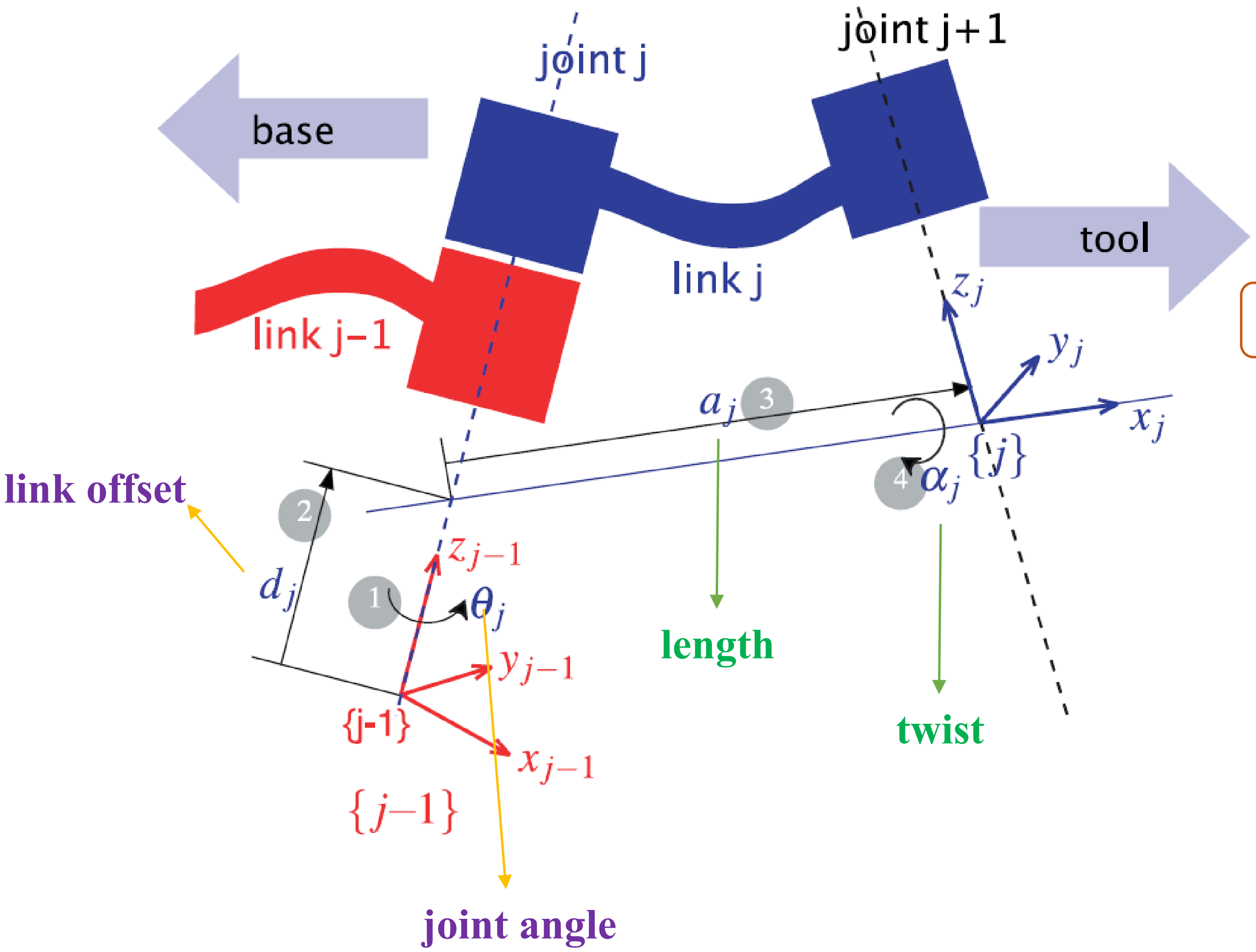


- The numbers in circles represent the **order** in which the elementary transforms are applied.

- Joint j connects link $j-1$ to link j and therefore joint j moves link j
- A **link** can be specified by two parameters, its **length** $\boldsymbol{a_j}$ and **its twist** $\boldsymbol{\alpha_j}$.

- The **link offset** d_j is the distance from one link coordinate frame to the next along the axis of the joint.
- The **joint angle** θ_j is the rotation of one link with respect to the next about the joint axis.

Definition of standard Denavit and Hartenberg link parameters



Joint angle	θ_j	the angle between the x_{j-1} and x_j axes about the z_{j-1} axis	revolute joint variable
Link offset	d_j	the distance from the origin of frame $j-1$ to the x_j axis along the z_{j-1} axis	prismatic joint variable
Link length	a_j	the distance between the z_{j-1} and z_j axes along the x_j axis; for intersecting axes is parallel to $\hat{z}_{j-1} \times \hat{z}_j$	constant
Link twist	α_j	the angle from the z_{j-1} axis to the z_j axis about the x_j axis	constant
Joint type	σ_j	$\sigma = 0$ for a revolute joint, $\sigma = 1$ for a prismatic joint	constant

$$
\begin{aligned}
A_i &= Rot_{z,\theta_i} \mathrm{Trans}_{z,d_i} \mathrm{Trans}_{x,a_i} Rot_{x,\alpha_i} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&\quad \times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i} c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Definition of standard Denavit and Hartenberg link parameters

DH Coordinate Frame Assumptions

(DH1) The axis x_1 is perpendicular to the axis z_0.

(DH2) The axis x_1 intersects the axis z_0.

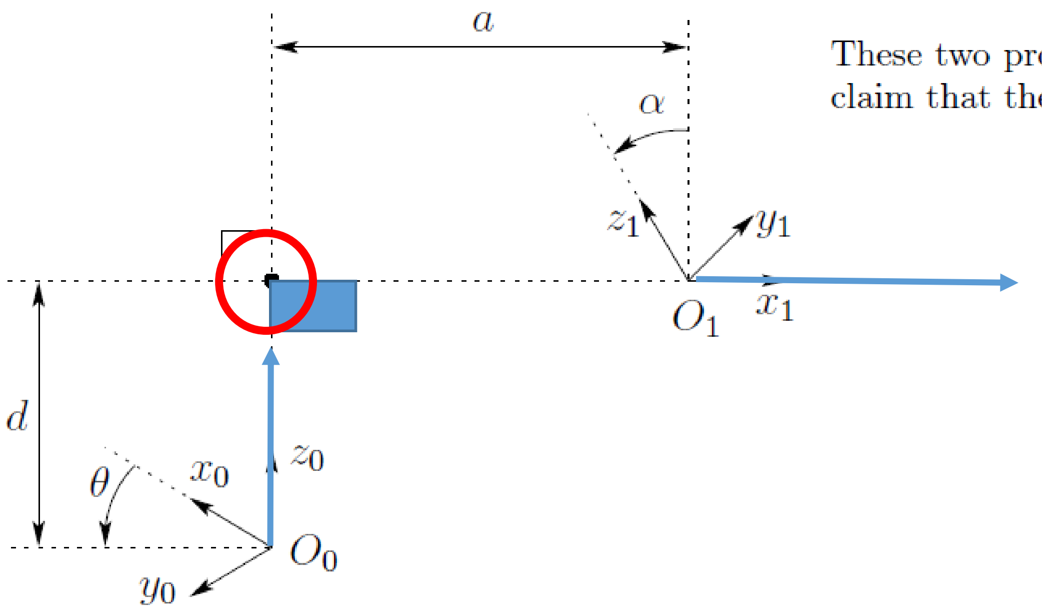


Fig. 3.2 Coordinate frames satisfying assumptions DH1 and DH2

These two properties are illustrated in Figure 3.2. Under these conditions, we claim that there exist unique numbers a, d, θ, α such that

$$A = Rot_{z,\theta}\text{Trans}_{z,d}\text{Trans}_{x,a}Rot_{x,\alpha} \tag{3.11}$$

DH Algorithm

Step 1. Locate and label the joint axes $z_0, \ldots, z_{n-1}$

Step 2. Establish the base frame. Set the origin anywhere on the z_0 axis. The x_0, y_0 axes are chosen conveniently to form a right-handed frame.

For $i = 1, \ldots, n-1$ perform steps 3 to 5

Step 3. Locate the origin o_i where the common normal to z_i and z_{i-1} intersects z_i.

- If z_i intersects z_{i-1} locate o_i at this intersection.
- If z_i and z_{i-1} are parallel, locate o_i in any convenient position along z_i.

Step 4. If z_{i-1} and z_i intersect, establish x_i in the direction normal to the plane formed by z_{i-1} and z_i. In other case, establish x_i along the common normal between z_{i-1} and z_i through o_i.

Step 5. Establish y_i to complete a right-handed frame.

Step 6. Establish the end-effector frame $o_n x_n y_n z_n$.

- Assuming the $n-th$ joint is revolute, set $z_n = a$ parallel to z_{n-1}. Establish the origin o_n conveniently along z_n, preferably at the center of the gripper or at the tip on any tool that the manipulator may be carrying. Set $y_n = s$ in the direction of the gripper closure and the set $x_n = n$ as $s \times a$.
- If the tool is not a simple gripper set x_n and y_n conveniently to form right-handed frame.

DH Algorithm

Step 7. Create a table of DH parameters $a_i, d_i, \alpha_i, \theta_i$.

a_i = distance along x_i from the intersection of the x_i and the z_{i-1} axes to o_i.

d_i = distance along z_{i-1} from o_{i-1} to the intersection of the x_i and z_{i-1} axes. If joint i is prismatic, d_i is variable.

α_i = the angle from z_{i-1} to z_i measured about x_i.

θ_i = the angle from x_{i-1} to x_i measured about z_{i-1}. If joint i is revolute, θ_i is variable.

Step 8. From the homogenoeus transformation matrices A_i by substituting the above parameters into

$$A_i = Rot_{z,\theta_i} \mathrm{Trans}_{z,d_i} \mathrm{Trans}_{x,a_i} Rot_{x,\alpha_i}$$

Step 9. Form $T_n^0 = A_1 \cdots A_n$. This then gives the position and orientation of the tool frame expressed in base coordinates.

Example: Planar Elbow Manipulator

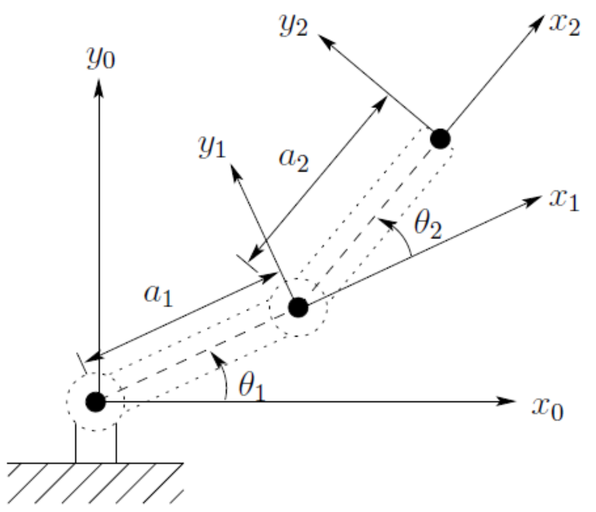


Fig. 3.6 Two-link planar manipulator. The z-axes all point out of the page, and are not shown in the figure

Table 3.1 Link parameters for 2-link planar manipulator

Link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1^*
2	a_2	0	0	θ_2^*

* variable

Step 1. Locate and label the joint axes $z_0, \ldots, z_{n-1}$

Step 2. Establish the base frame. Set the origin anywhere on the z_0 axis. The x_0, y_0 axes are chosen conveniently to form a right-handed frame.

For $i = 1, \ldots, n-1$ perform steps 3 to 5

Step 3. Locate the origin o_i where the common normal to z_i and z_{i-1} intersects z_i.

- If z_i intersects z_{i-1} locate o_i at this intersection.
- If z_i and z_{i-1} are parallel, locate o_i in any convenient position along z_i.

Step 4. Establish x_i along the common normal between z_{i-1} and z_i through o_i, or in the direction normal to the $z_{i-1} - z_i$ plane if z_{i-1} and z_i intersect.

Step 5. Establish y_i to complete a right-handed frame.

Step 6. Establish the end-effector frame $o_n x_n y_n z_n$.

- Assuming the $n-th$ joint is revolute, set $z_n = a$ parallel to z_{n-1}. Establish the origin o_n conveniently along z_n, preferably at the center of the gripper or at the tip on any tool that the manipulator may be carrying. Set $y_n = s$ in the direction of the gripper closure and the set $x_n = n$ as $s \times a$.
- If the tool is not a simple gripper set x_n and y_n conveniently to form right-handed frame.

Step 7. Create a table of DH parameters $a_i, d_i, \alpha_i, \theta_i$.

a_i = distance along x_i from the intersection of the x_i and the z_{i-1} axes to o_i.

d_i = distance along z_{i-1} from o_{i-1} to the intersection of the x_i and z_{i-1} axes. If joint i is prismatic, d_i is variable.

α_i = the angle from z_{i-1} to z_i measured about x_i.

θ_i = the angle from x_{i-1} to x_i measured about z_{i-1}. If joint i is revolute, θ_i is variable.

Step 8. From the homogeneous transformation matrices A_i by substituting the above parameters into

$$A_i = Rot_{z,\theta_i} \text{Trans}_{z,d_i} \text{Trans}_{x,a_i} Rot_{x,\alpha_i}$$

Step 9. Form $T_n^0 = A_1 \cdots A_n$. This then gives the position and orientation of the tool frame expressed in base coordinates

Example: Planar Elbow Manipulator

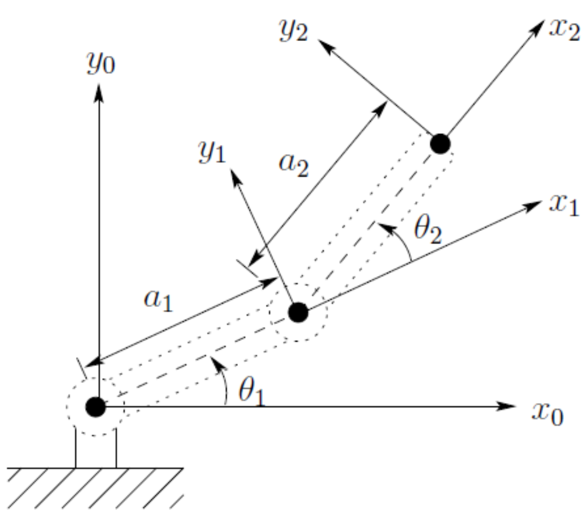


Fig. 3.6 Two-link planar manipulator. The z-axes all point out of the page, and are not shown in the figure

Table 3.1 Link parameters for 2-link planar manipulator

Link	a_i	α_i	d_i	θ_i
1	a_1	0	0	θ_1^*
2	a_2	0	0	θ_2^*

* variable

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & a_1c_1 \\ s_1 & c_1 & 0 & a_1s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The T-matrices are thus given by

$$T_1^0 = A_1$$

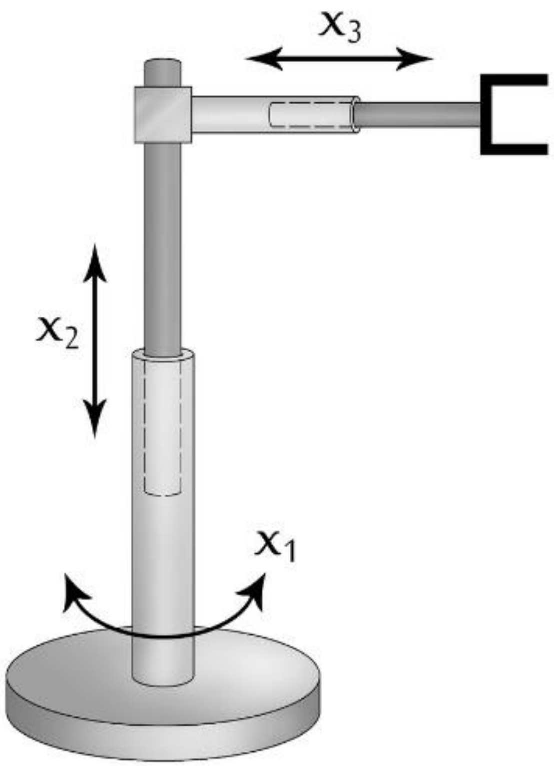
$$T_2^0 = A_1A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Notice that the first two entries of the last column of T_2^0 are the x and y components of the origin o_2 in the base frame; that is,

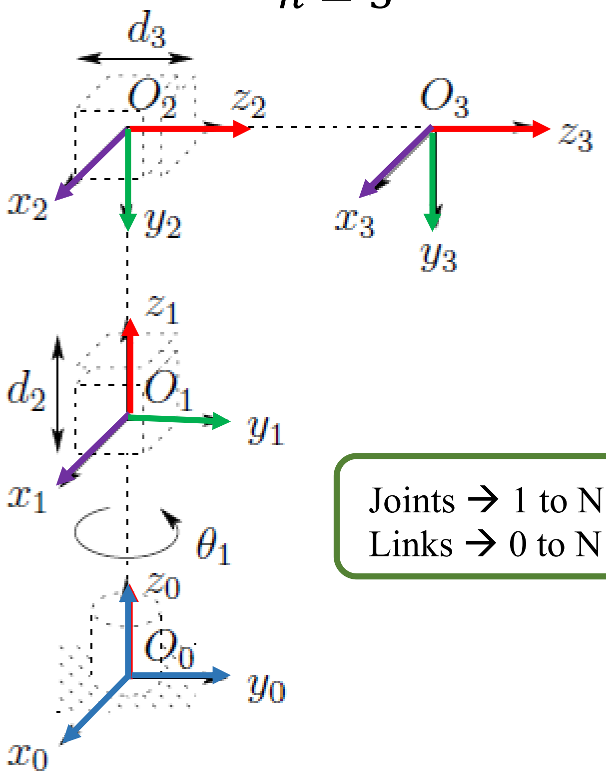
$$x = a_1c_1 + a_2c_{12}$$
$$y = a_1s_1 + a_2s_{12}$$

are the coordinates of the end-effector in the base frame. The rotational part of T_2^0 gives the orientation of the frame $o_2x_2y_2z_2$ relative to the base frame.
◇

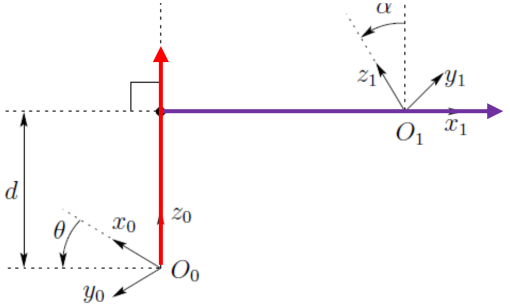
Three-Link Cylindrical Robot



$n = 3$



Joints → 1 to N
Links → 0 to N



Remember:

- Joint i is fixed w.r.t. frame i.
- When joint i is actuated, link i and its attached frame experience a motion.

Simplified steps:

1. Define n according to the degrees of freedom of robot.
2. Assign the z axes such that z_i to be the axis of actuation for joint $i + 1$.
3. Define the base frame in a convenient manner.
4. Choose the x_i with Step 4 of the algorithm:
   - If z_{i-1} and z_i intersect, establish x_i in the direction normal to the plane formed by z_{i-1} and z_i. In other case, establish x_i along the common normal between z_{i-1} and z_i through o_i.
5. Corroborate the 2 assumptions for all frames.

   DH Coordinate Frame Assumptions

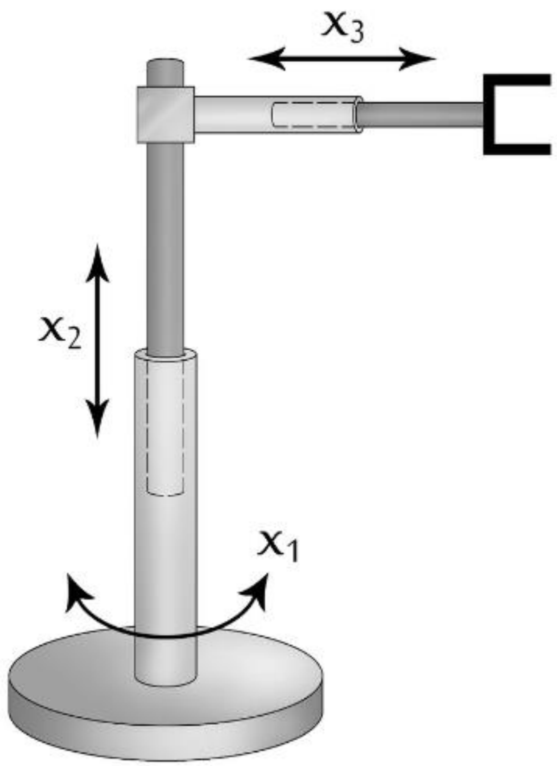
   (DH1) The axis x_1 is perpendicular to the axis z_0.

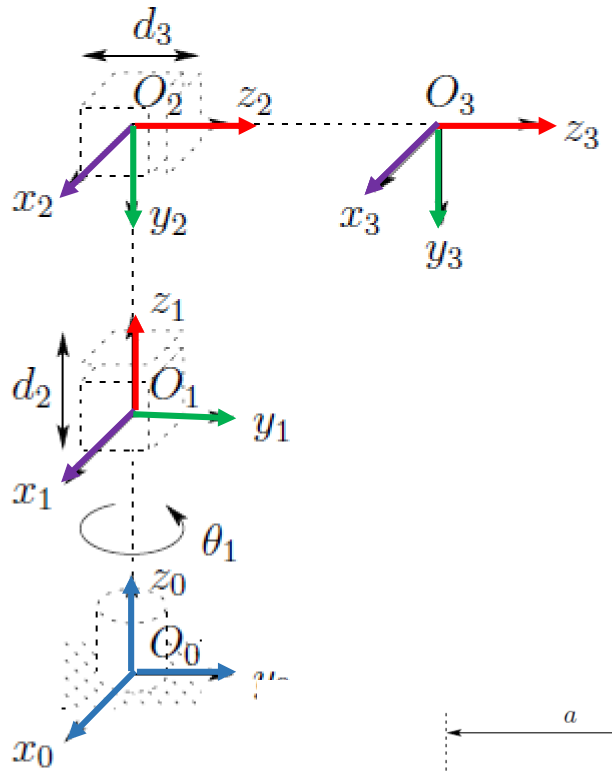
   (DH2) The axis x_1 intersects the axis z_0.
6. Establish y_i to complete a right-handed frame

Three-Link Cylindrical Robot

Simplified steps:

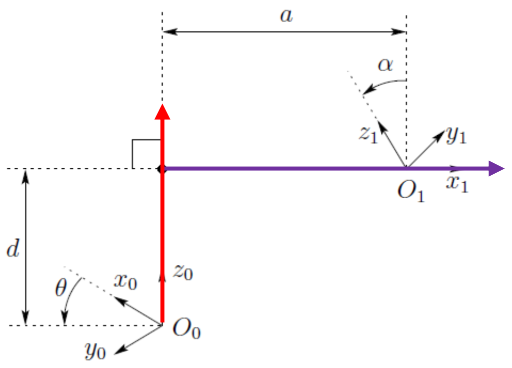
8. We create the table of DH parameters.

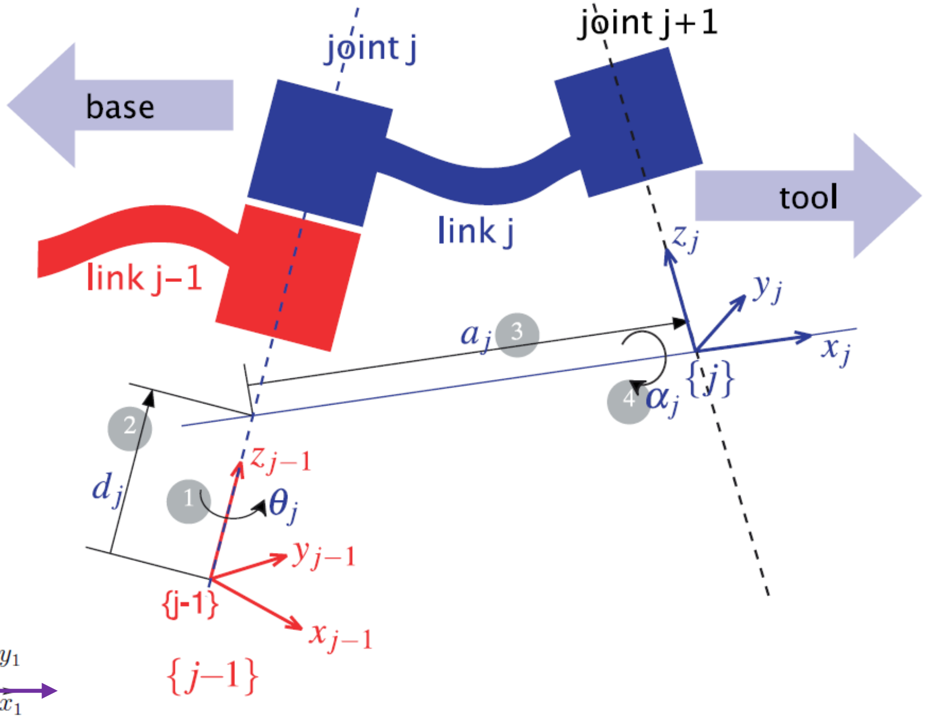




Link	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90	d_2^*	0
3	0	0	d_3^*	0

* variable

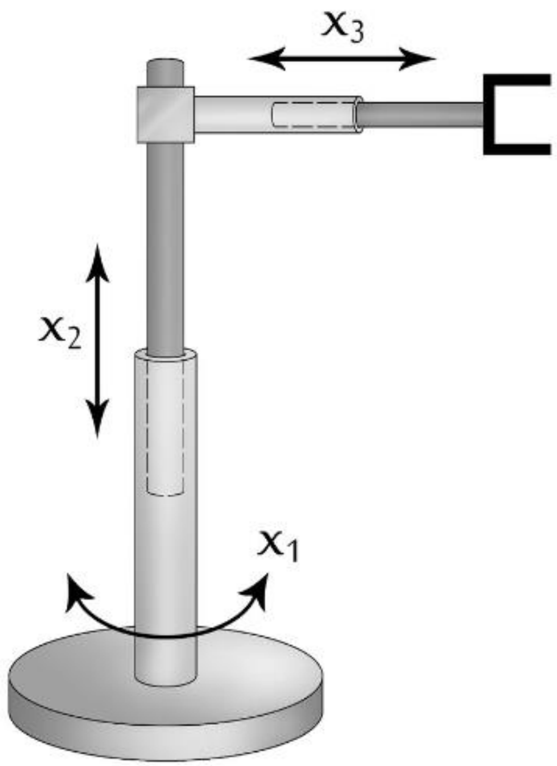


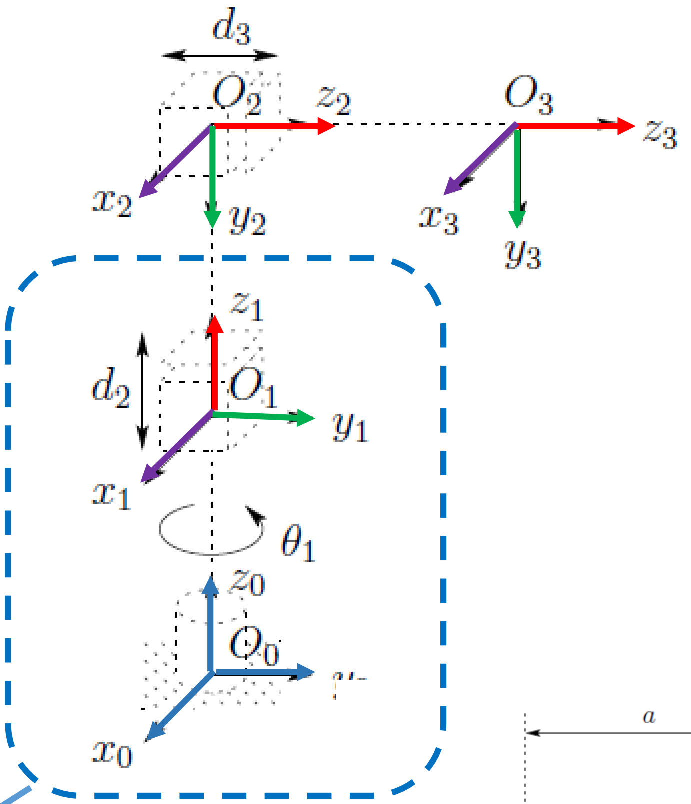


Three-Link Cylindrical Robot

Simplified steps:

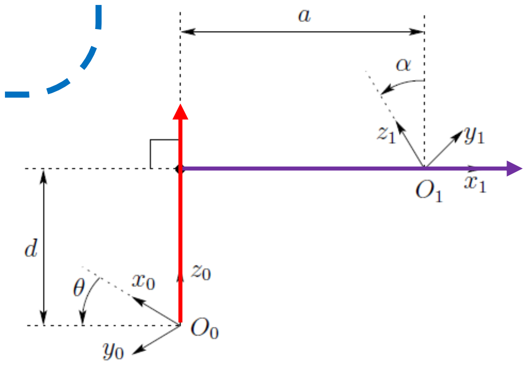
8. We create the table of DH parameters.

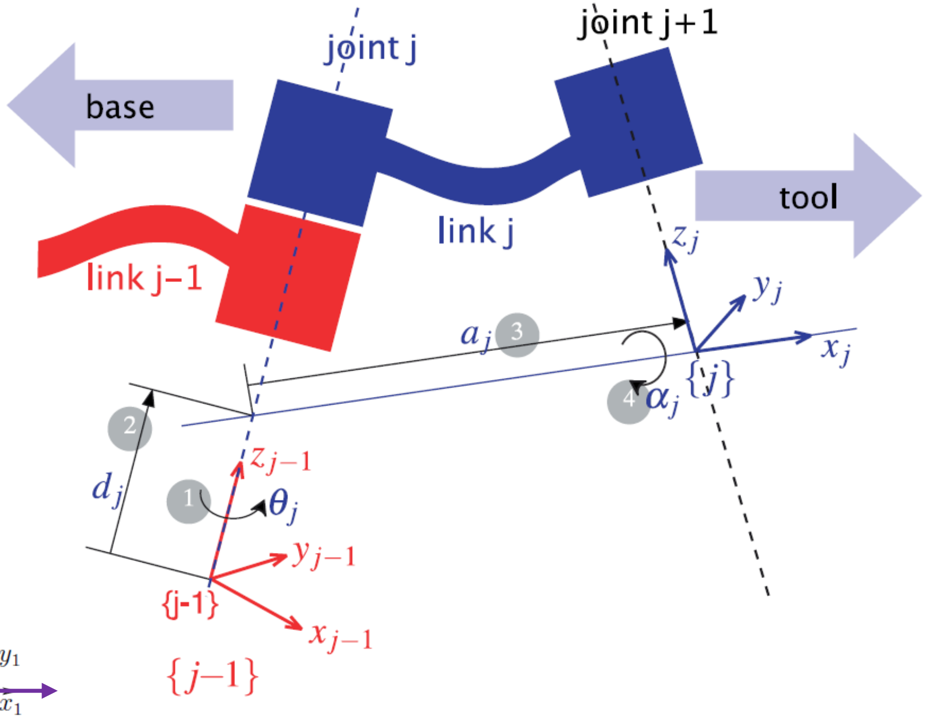




Link	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90	d_2^*	0
3	0	0	d_3^*	0

* variable

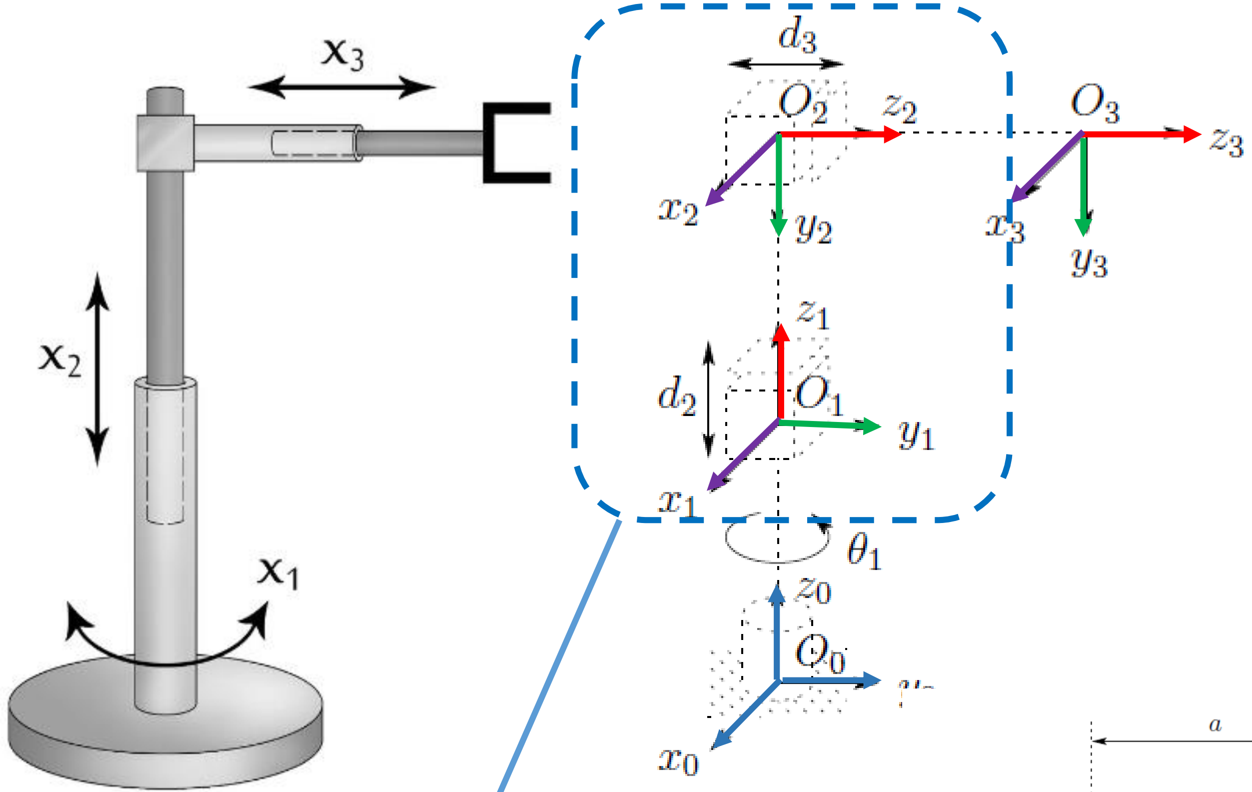




Three-Link Cylindrical Robot

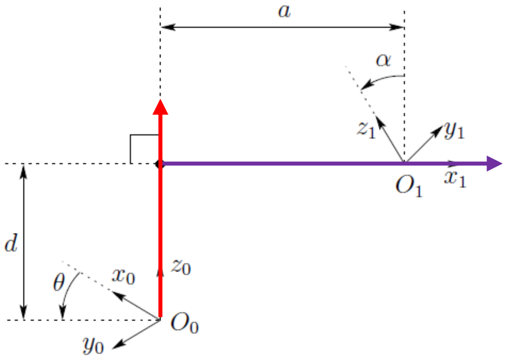
Simplified steps:

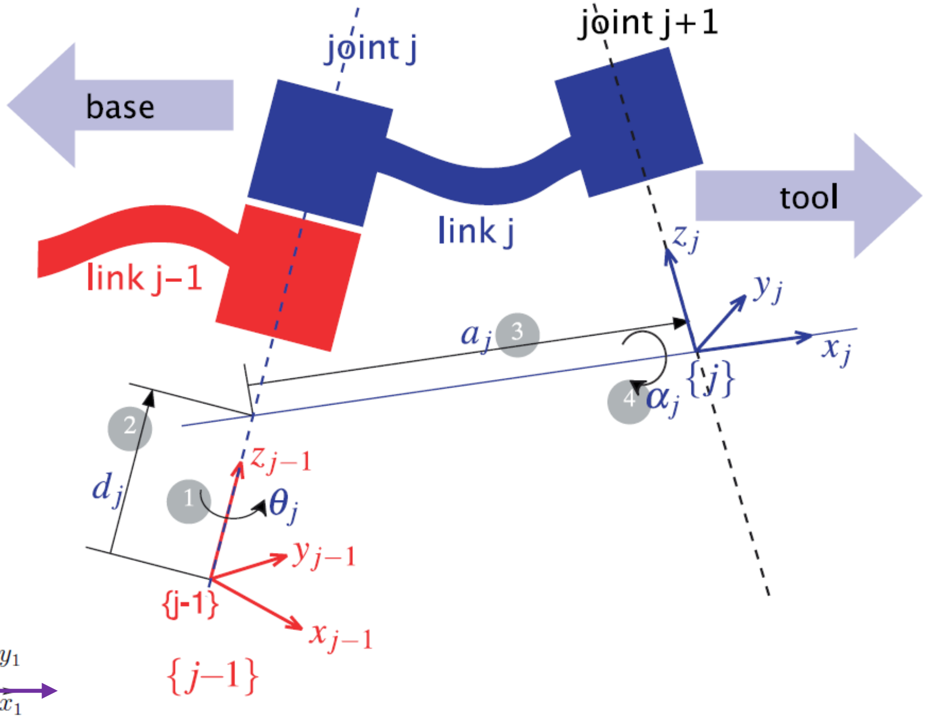
8. We create the table of DH parameters.



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

* variable

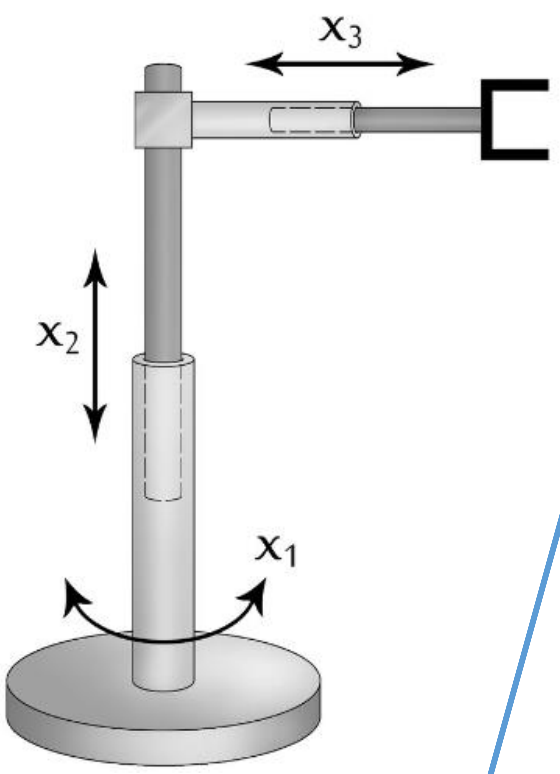


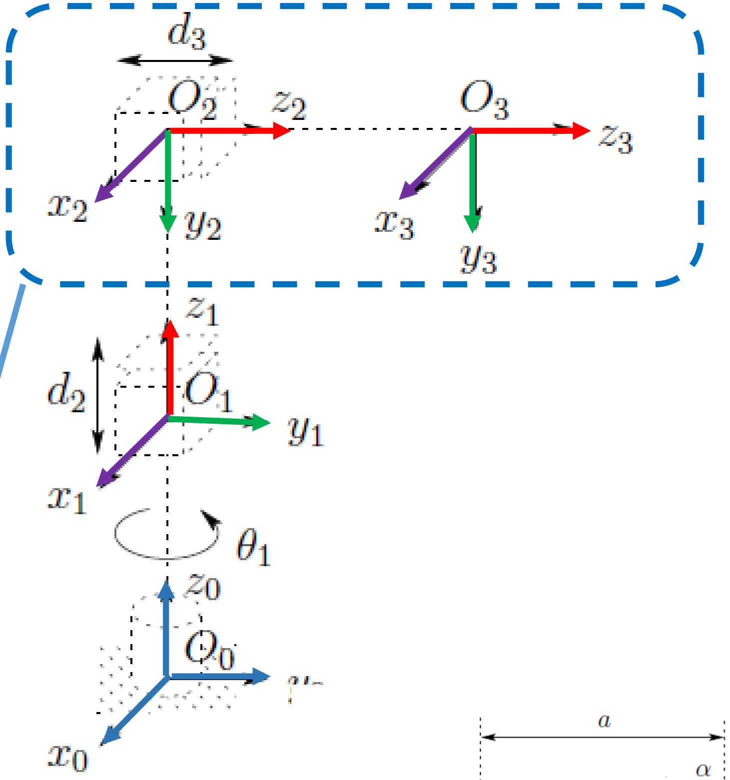


Three-Link Cylindrical Robot

Simplified steps:

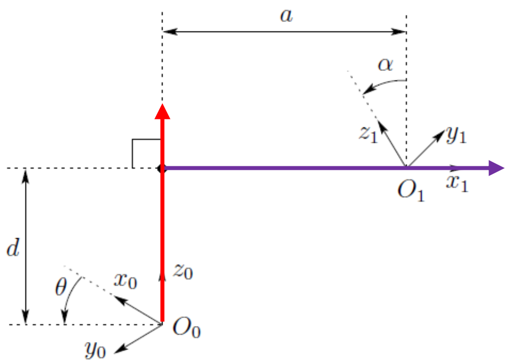
8. We create the table of DH parameters.

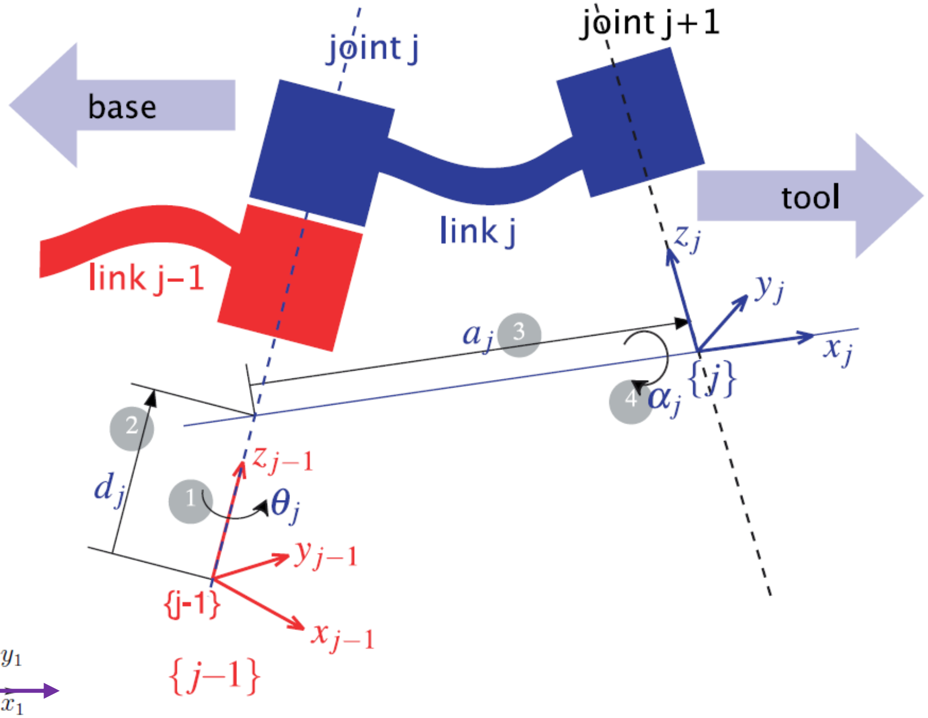




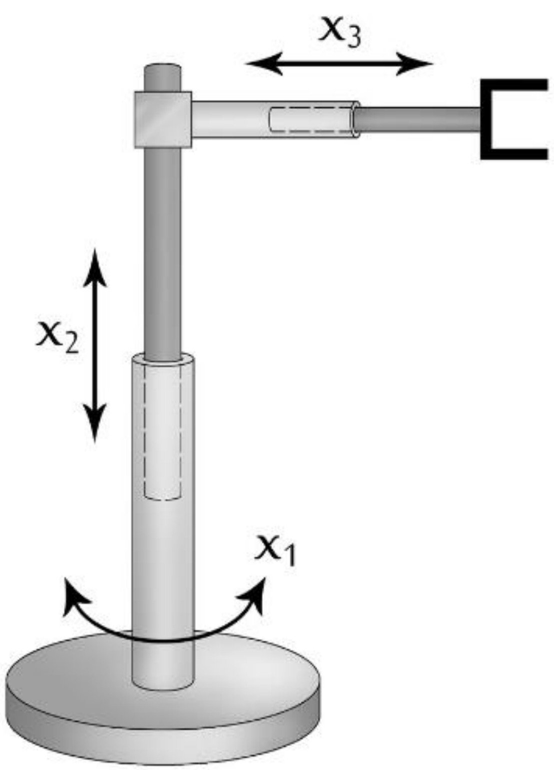
Link	a_i	α_i	d_i	θ_i
1	0	0	d_1	θ_1^*
2	0	-90	d_2^*	0
3	0	0	d_3^*	0

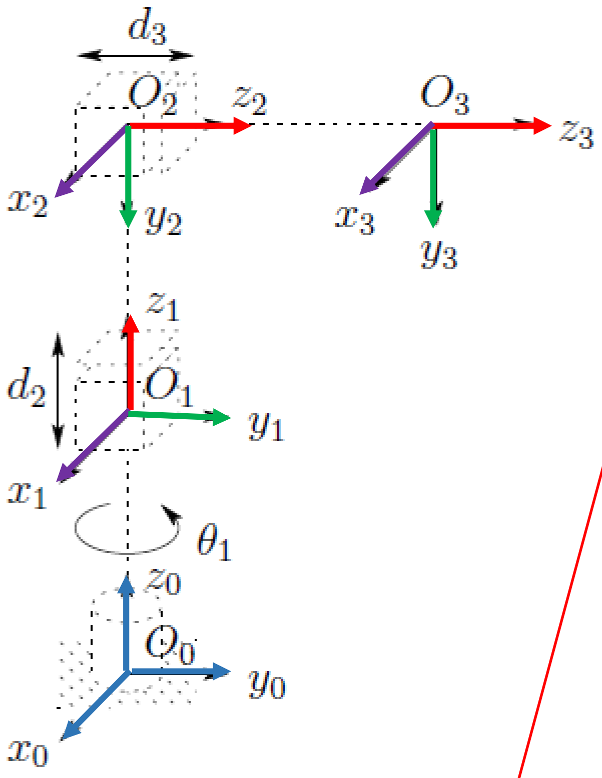
* variable





Three-Link Cylindrical Robot





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

* variable

Simplified steps:

9. Build the matrices

$$A_1 = \begin{bmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

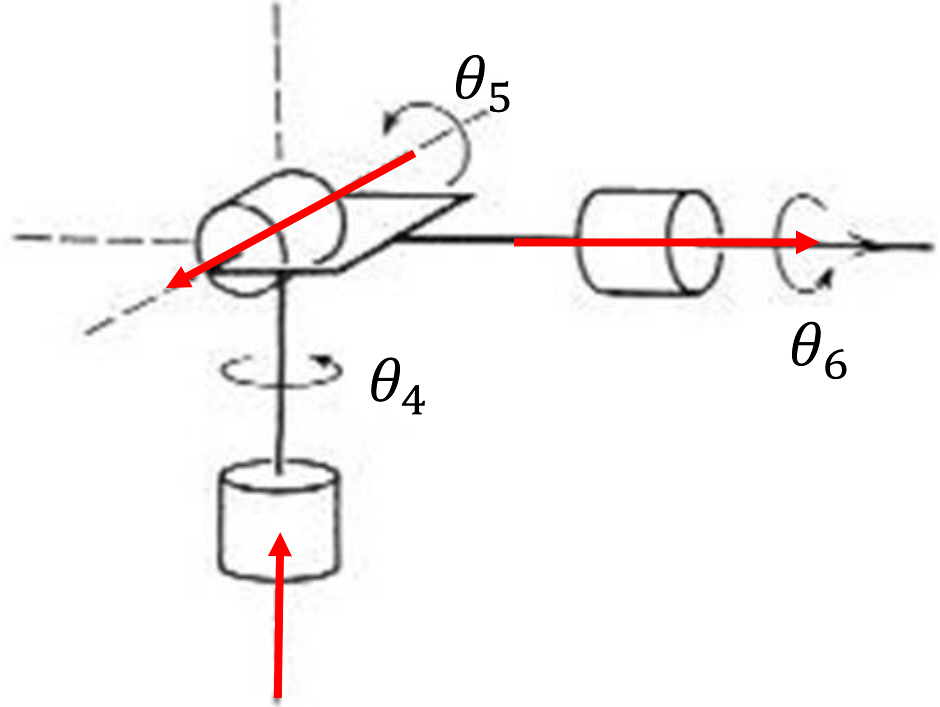
$$A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

10. Build the T_n^0

$$T_3^0 = A_1 A_2 A_3 = \begin{bmatrix} c_1 & 0 & -s_1 & -s_1 d_3 \\ s_1 & 0 & c_1 & c_1 d_3 \\ 0 & -1 & 0 & d_1 + d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Spherical wrist



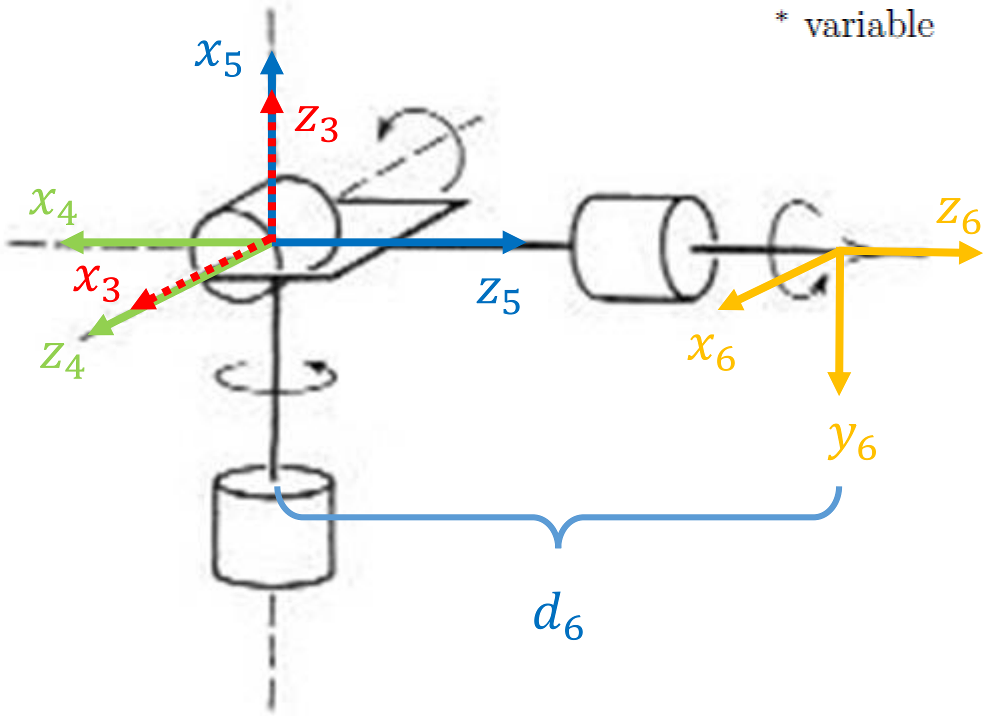
Notice the direction of each arrow.

Use the right-hand rule where your thumb is x_4

Table 3.3 DH parameters for spherical wrist

Link	a_i	α_i	d_i	θ_i
4	0	−90	0	θ_4^*
5	0	90	0	θ_5^*
6	0	0	d_6	θ_6^*

* variable



We have intentionally omitted the y-axis for clarity. Please try to identify the relevant values.

Spherical wrist

$$A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

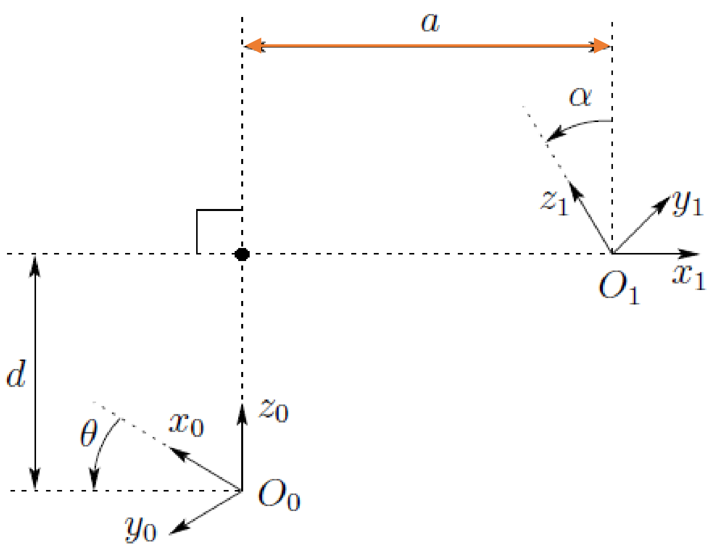
$$A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

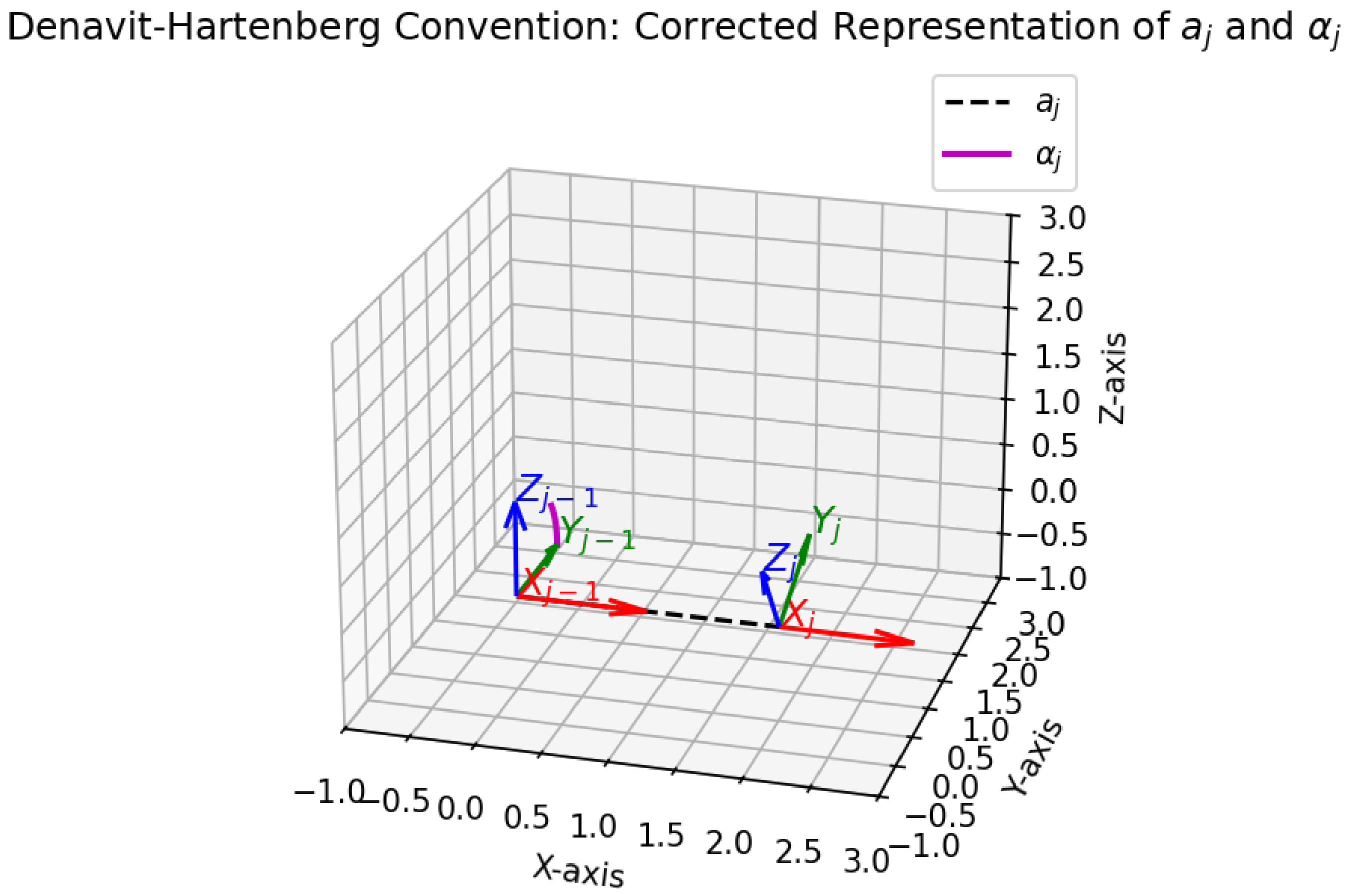
$$A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Multiplying these together yields

$$\begin{aligned} T_6^3 &= A_4 A_5 A_6 \\ &= \begin{bmatrix} R_6^3 & o_6^3 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} c_4c_5c_6 - s_4s_6 & -c_4c_5s_6 - s_4c_6 & c_4s_5 & c_4s_5d_6 \\ s_4c_5c_6 + c_4s_6 & -s_4c_5s_6 + c_4c_6 & s_4s_5 & s_4s_5d_6 \\ -s_5c_6 & s_5s_6 & c_5 & c_5d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

Recap





See code!

Recap

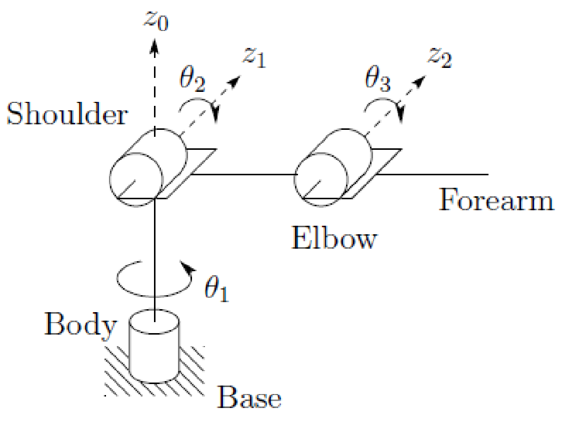
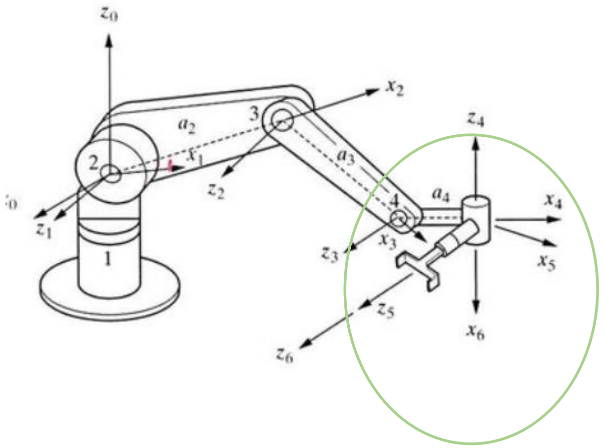


Fig. 1.8 Structure of the elbow manipulator.



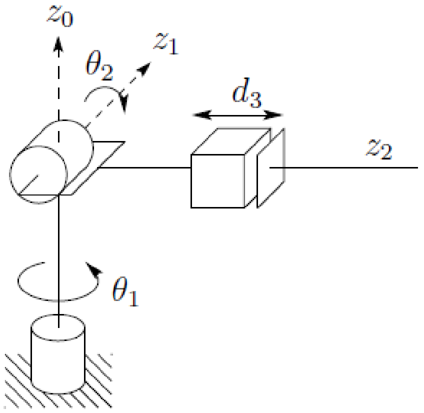
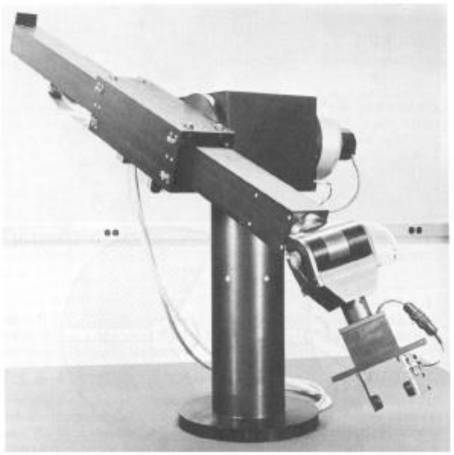


Fig. 1.10 The spherical manipulator.



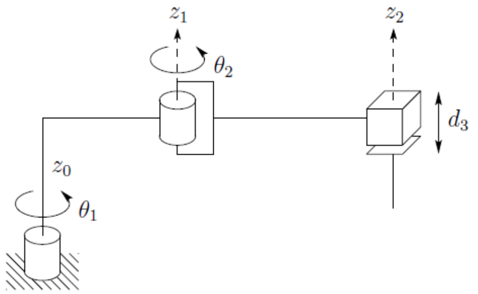


Fig. 1.13 The SCARA (Selective Compliant Articulated Robot for Assembly).

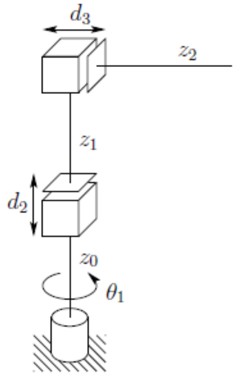


Fig. 1.16 The cylindrical manipulator.

Visual Studio Code

Install Visual Studio Code

Follow the steps:

- https://code.visualstudio.com/docs/python/python-tutorial
- Install Peter Corke's Robotics library:
- https://github.com/petercorke/robotics-toolbox-python

Exercise 1

- <u>Problem</u>:

1. Following the previous analysis compute the T_n^0 for the planar robot, and give the coordinates of the tip.
2. Verify your results on Python using the Corke's libraries. Study the codes:
    1. robotarmDH.py and robotarmAnalytics-3DCylindrical.py

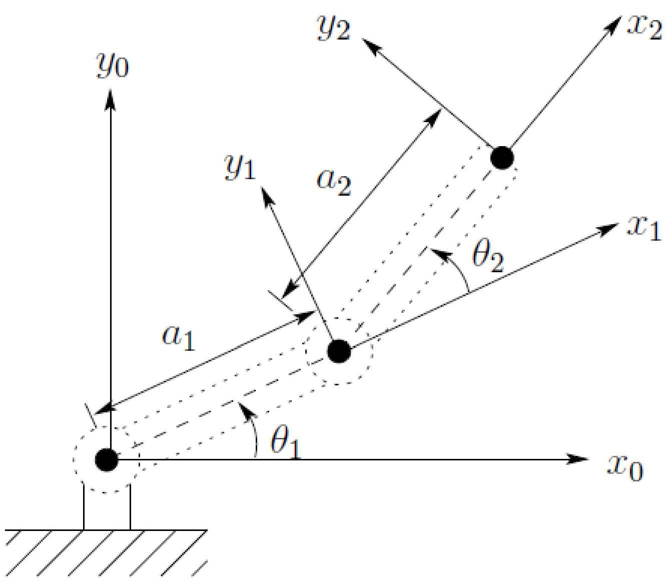
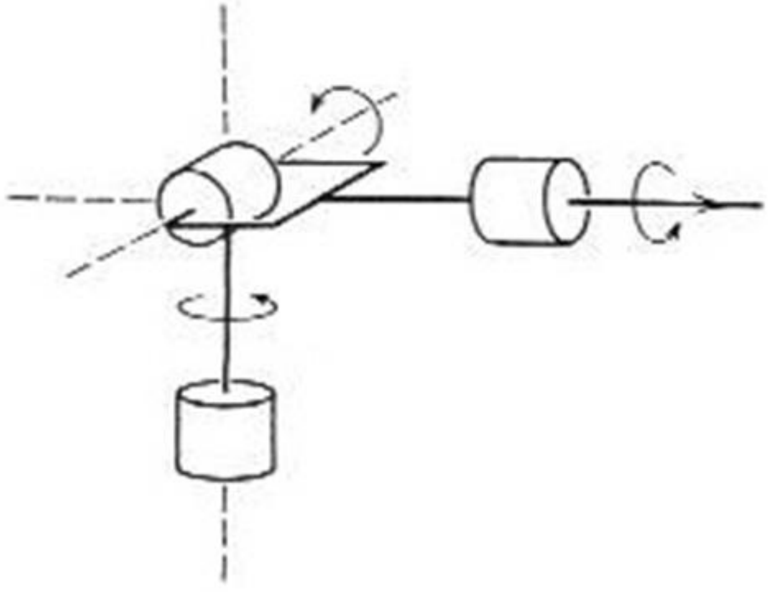


Fig. 3.6 Two-link planar manipulator. The z-axes all point out of the page, and are not shown in the figure

Exercise 2

- Problem:

1. Following the previous analysis compute the T_n^0 for the robot below, and give the coordinates of the tip.
2. Verify your results on Python using the Corke's libraries.



Exercise 3

- Problem:

1. Following the previous analysis compute the T_n^0 for the robot below, and give the coordinates of the tip.
2. Verify your results on Python using the Corke's libraries.

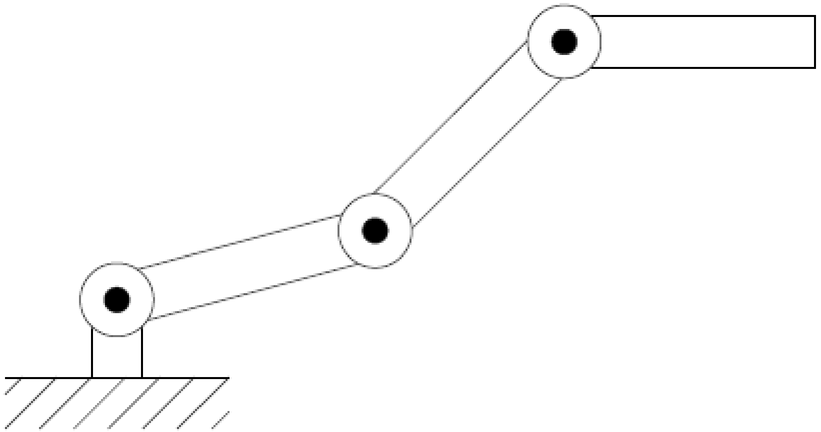


Fig. 3.23 Three-link planar arm of Problem 3-2

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

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

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

Exercise 4

- <u>Problem</u>:

1. Following the previous analysis compute the T_n^0 for the robot below, and give the coordinates of the tip.
2. Verify your results on Python using the Corke's libraries.v

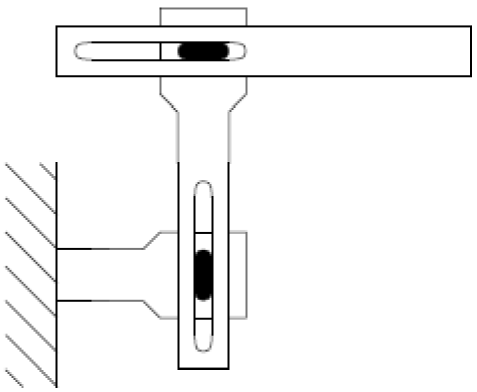
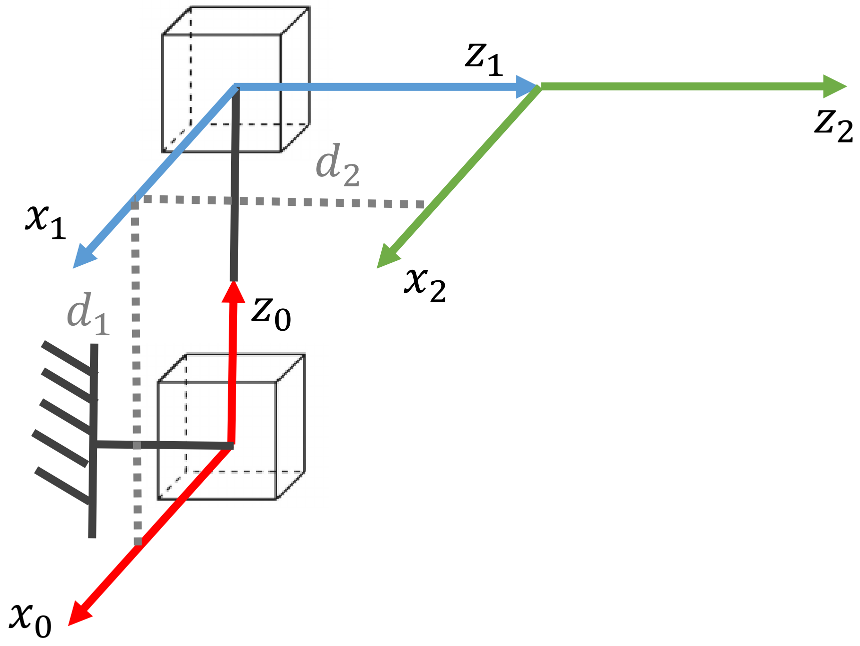


Fig. 3.24 Two-link cartesian robot of Problem 3-3

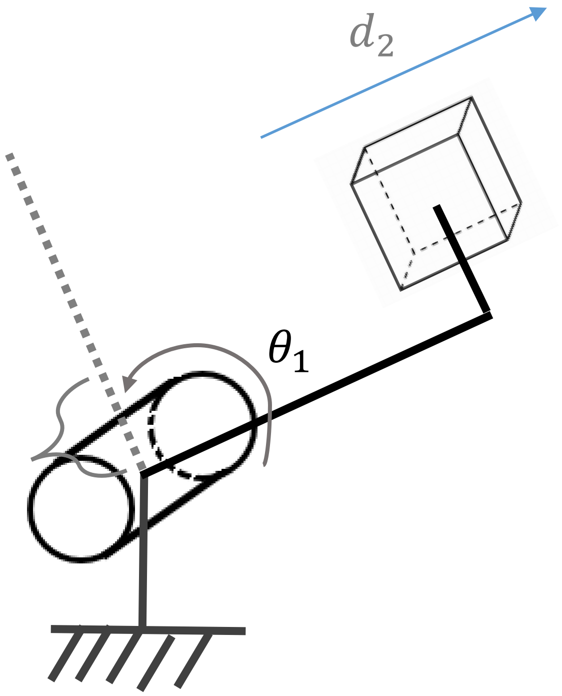


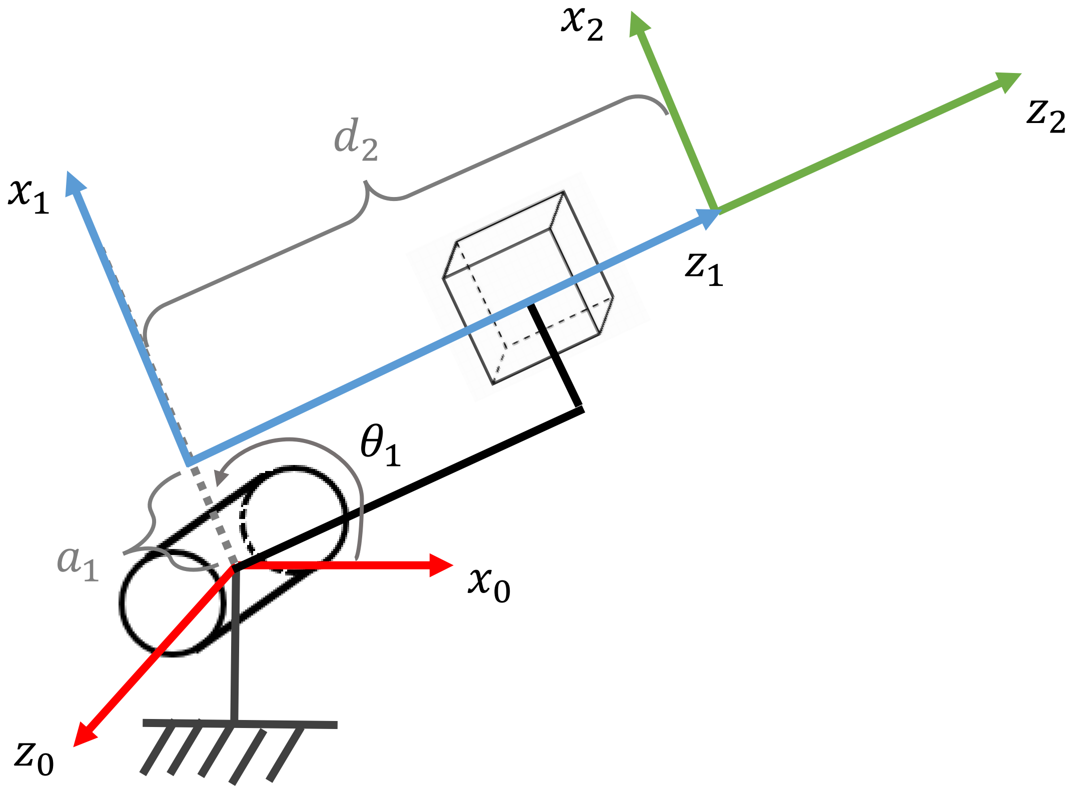
Exercise 5

link	a_i	α_i	d_i	θ_i
0	a_1	90	0	θ_1 *
1	0	0	d_2 *	0

- <u>Problem</u>:

1. Following the previous analysis compute the T_n^0 for the robot below, and give the coordinates of the tip.





Midterm

Midterm

Dear RAS,

The midterm exam will be held during our regular class time next **Wednesday, October 9th**. It will cover all material up to and including the topic of **forward kinematics**.

The exam will be taken individually, not as a team.

Please make sure to study thoroughly and come well-prepared!

Regards

Gerardo

Extra points:

- Please submit the printed work by **Wednesday, October 9th** at **12:00 PM**. Please submit the assignment printed, not handwritten. Each problem will award additional points.

Exercise 1

Compute the rotation matrix given by the product

$$R_{x,\theta}R_{y,\phi}R_{z,\pi}R_{y,-\phi}R_{x,-\theta}$$

Exercise 2

Consider the diagram of Figure 2.15. Find the homogeneous transformations H_1^0, H_2^0, H_2^1 representing the transformations among the three frames shown. Show that $H_2^0 = H_1^0, H_2^1$.

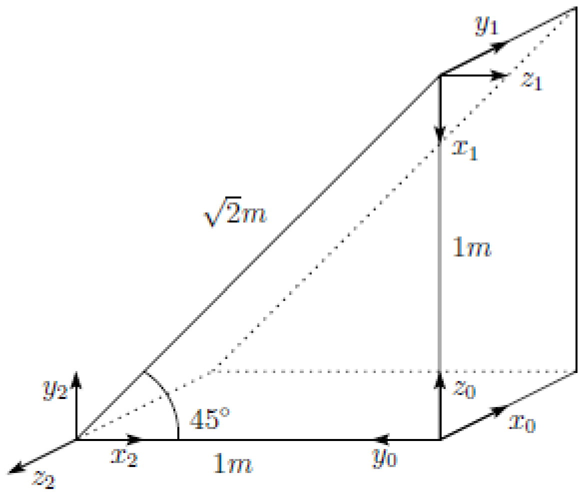


Fig. 2.15 Diagram for Problem 2-36.

Extra points:

Exercise 3

Write a Python code that computes the solution for Exercise 1.

Exercise 4

Write a Python code that computes H_2^0 of Exercise 2.

Exercise 5

Consider the diagram of Figure 2.15. Find the homogeneous transformations H_1^0, H_2^0, H_2^1 representing the transformations among the three frames shown. Show that $H_2^0 = H_1^0, H_2^1$.

Extra points:

Exercise 6

Consider the diagram of Figure 2.16. A robot is set up 1 meter from a table. The table top is 1 meter high and 1 meter square. A frame $o_1x_1y_1z_1$ is fixed to the edge of the table as shown. A cube measuring 20 cm on a side is placed in the center of the table with frame $o_2x_2y_2z_2$ established at the center of the cube as shown. A camera is situated directly above the center of the block 2m above the table top with frame $o_3x_3y_3z_3$ attached as shown. Find the homogeneous transformations relating each of these frames to the base frame $o_0x_0y_0z_0$. Find the homogeneous transformation relating the frame $o_2x_2y_2z_2$ to the camera frame $o_3x_3y_3z_3$.

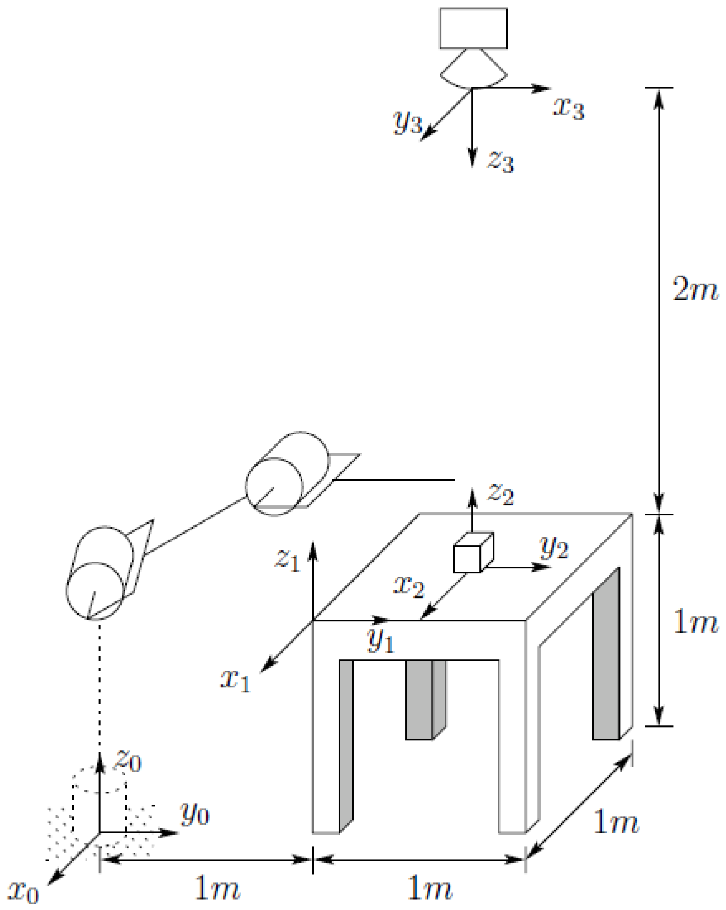


Fig. 2.16 Diagram for Problem 2.37.

Extra points:

Exercise 7

Consider the three-link articulated robot of Figure 3.27. Derive the forward kinematic equations using the DH-convention.

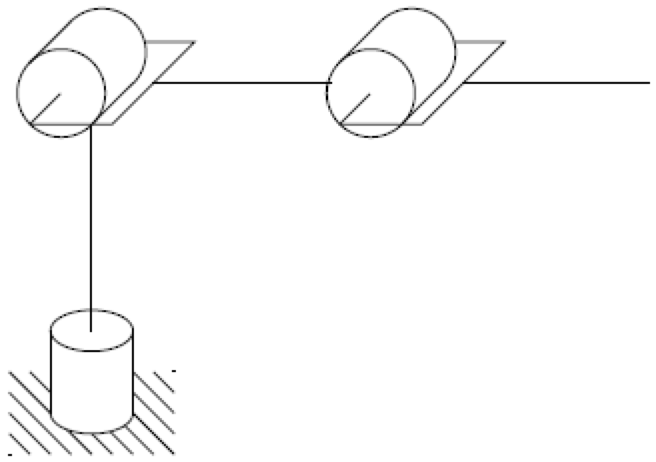


Fig. 3.27 Three-link articulated robot

Extra points:

Exercise 8

Consider the three-link cartesian manipulator of Figure 3.28. Derive the forward kinematic equations using the DH-convention.

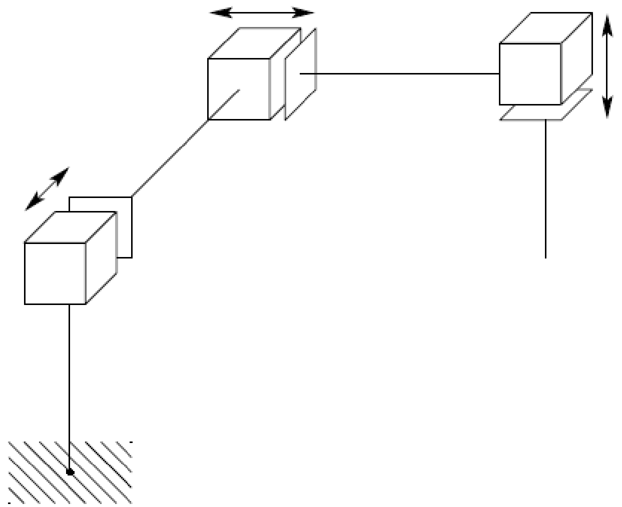


Fig. 3.28 Three-link cartesian robot

Forward kinematics (recap)

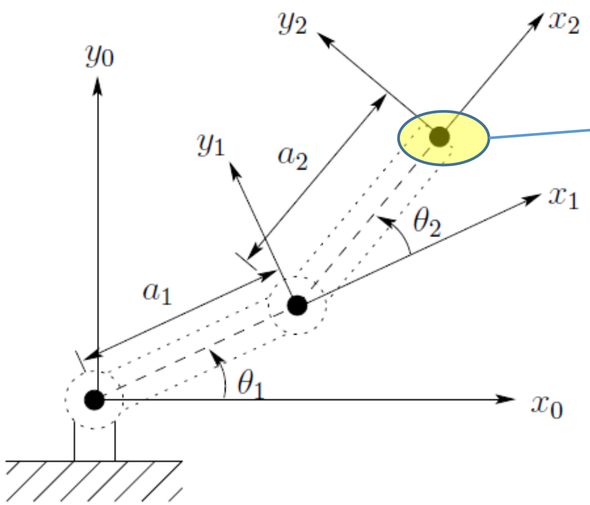


Fig. 3.6 Two-link planar manipulator. The z-axes all point out of the page, and are not shown in the figure

Orientation (R) Position (p)

$$T_2^0 = A_1 A_2 = \begin{bmatrix} c_{12} & -s_{12} & 0 & a_1c_1 + a_2c_{12} \\ s_{12} & c_{12} & 0 & a_1s_1 + a_2s_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Notice that the first two entries of the last column of T_2^0 are the x and y components of the origin o_2 in the base frame; that is,

$$\begin{aligned} x &= a_1c_1 + a_2c_{12} \\ y &= a_1s_1 + a_2s_{12} \end{aligned}$$

are the coordinates of the end-effector in the base frame. The rotational part of T_2^0 gives the orientation of the frame $o_2x_2y_2z_2$ relative to the base frame.

$\diamond$

With the forward kinematics process we find the position (vector p) and orientation (matrix R) of the tip of the robot.

Inverse Kinematics

The general problem of inverse kinematics can be stated as follows. Given a 4×4 homogeneous transformation

$$\boxed{H = \begin{bmatrix} R & o \\ 0 & 1 \end{bmatrix} \in SE(3)} \qquad (3.31)$$

with $R \in SO(3)$, find (one or all) solutions of the equation

$$T_n^0(q_1, \ldots, q_n) = H \qquad (3.32)$$

where

$$T_n^0(q_1, \ldots, q_n) = A_1(q_1) \cdots A_n(q_n) \qquad (3.33)$$

Here, H represents the desired position and orientation of the end-effector, and our task is to find the values for the joint variables $q_1, \ldots, q_n$ so that $T_n^0(q_1, \ldots, q_n) = H$.

So, R_d and p_d are given, then we have:

$$\begin{pmatrix} R & p \\ 0,0,0 & 1 \end{pmatrix} = \begin{pmatrix} R_d & p_d \\ 0,0,0 & 1 \end{pmatrix}.$$

And we need to find the values of each element of (R, p).

Inverse Kinematics Example

Stanford Manipulator

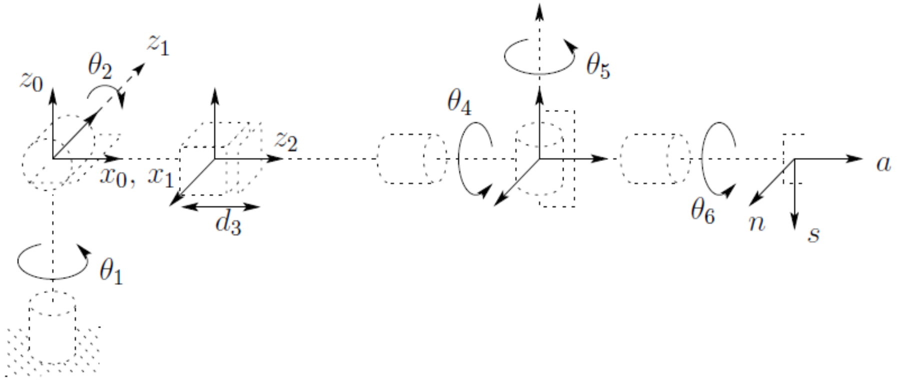


Fig. 3.10 DH coordinate frame assignment for the Stanford manipulator

Table 3.4 DH parameters for Stanford Manipulator

Link	d_i	a_i	α_i	θ_i
1	0	0	−90	$\theta^\star$
2	d_2	0	+90	$\theta^\star$
3	$d^\star$	0	0	0
4	0	0	−90	$\theta^\star$
5	0	0	+90	$\theta^\star$
6	d_6	0	0	$\theta^\star$

* joint variable

$$A_1 = \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_2 & 0 & s_2 & 0 \\ s_2 & 0 & -c_2 & 0 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

T_6^0 is then given as

$$T_6^0 = A_1 \cdots A_6 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_x \\ r_{21} & r_{22} & r_{23} & d_y \\ r_{31} & r_{32} & r_{33} & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\left\{ \begin{aligned}
r_{11} &= c_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] - d_2(s_4c_5c_6 + c_4s_6) \\
r_{21} &= s_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] + c_1(s_4c_5c_6 + c_4s_6) \\
r_{31} &= -s_2(c_4c_5c_6 - s_4s_6) - c_2s_5c_6 \\
r_{12} &= c_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] - s_1(-s_4c_5s_6 + c_4c_6) \\
r_{22} &= -s_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] + c_1(-s_4c_5s_6 + c_4c_6) \\
r_{32} &= s_2(c_4c_5s_6 + s_4c_6) + c_2s_5s_6 \\
r_{13} &= c_1(c_2c_4s_5 + s_2c_5) - s_1s_4s_5 \\
r_{23} &= s_1(c_2c_4s_5 + s_2c_5) + c_1s_4s_5 \\
r_{33} &= -s_2c_4s_5 + c_2c_5 \\
d_x &= c_1s_2d_3 - s_1d_2 + +d_6(c_1c_2c_4s_5 + c_1c_5s_2 - s_1s_4s_5) \\
d_y &= s_1s_2d_3 + c_1d_2 + d_6(c_1s_4s_5 + c_2c_4s_1s_5 + c_5s_1s_2) \\
d_z &= c_2d_3 + d_6(c_2c_5 - c_4s_2s_5)
\end{aligned} \right.$$

Inverse Kinematics Example

Stanford Manipulator

Suppose that the desired position and orientation of the final frame are given by:

$$H = \begin{bmatrix} 0 & 1 & 0 & -0.154 \\ 0 & 0 & 1 & 0.763 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

To find the corresponding joint variables $\theta_1, \theta_2, d_3, \theta_4, \theta_5$ and θ_6 we must solve the following simultaneous set of nonlinear trigonometric equations:

$$\begin{aligned}
c_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] - s_1(s_4c_5c_6 + c_4s_6) &= 0 \\
s_1[c_2(c_4c_5c_6 - s_4s_6) - s_2s_5c_6] + c_1(s_4c_5c_6 + c_4s_6) &= 0 \\
-s_2(c_4c_5c_6 - s_4s_6) - c_2s_5c_6 &= 1 \\
c_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] - s_1(-s_4c_5s_6 + c_4c_6) &= 1 \\
s_1[-c_2(c_4c_5s_6 + s_4c_6) + s_2s_5s_6] + c_1(-s_4c_5s_6 + c_4c_6) &= 0 \\
s_2(c_4c_5s_6 + s_4c_6) + c_2s_5s_6 &= 0 \\
c_1(c_2c_4s_5 + s_2c_5) - s_1s_4s_5 &= 0 \\
s_1(c_2c_4s_5 + s_2c_5) + c_1s_4s_5 &= 1 \\
-s_2c_4s_5 + c_2c_5 &= 0 \\
c_1s_2d_3 - s_1d_2 + d_6(c_1c_2c_4s_5 + c_1c_5s_2 - s_1s_4s_5) &= -0.154 \\
s_1s_2d_3 + c_1d_2 + d_6(c_1s_4s_5 + c_2c_4s_1s_5 + c_5s_1s_2) &= 0.763 \\
c_2d_3 + d_6(c_2c_5 - c_4s_2s_5) &= 0
\end{aligned}$$

Inverse Kinematics Example

If the values of the nonzero DH parameters are $d_2 = 0.154$ *and* $d_6 = 0.263$*, one solution to this set of equations is given by:*

$$\theta_1 = \pi/2, \quad \theta_2 = \pi/2, \quad d_3 = 0.5, \quad \theta_4 = \pi/2, \quad \theta_5 = 0, \quad \theta_6 = \pi/2.$$

Even though we have not yet seen how one might derive this solution, it is not difficult to verify that it satisfies the forward kinematics equations for the Stanford arm.

Problem:

Verify the solution on Python.

How to solve the Inverse Kinematics problem?

- Numerically! Using the Levenberg–Marquardt algorithm.
- Simple code:

```python
import roboticstoolbox as rtb
from spatialmath import SE3
import matplotlib.pyplot as plt

# Define the robot's DH parameters (theta, d, a, alpha)
# Example of a robot with 3 links
L1 = rtb.RevoluteDH(d=0.5, a=1, alpha=0) # Link 1
L2 = rtb.RevoluteDH(d=0, a=1, alpha=0) # Link 2
L3 = rtb.RevoluteDH(d=0, a=1, alpha=0) # Link 3

# Create the robot from the links
robot = rtb.DHRobot([L1, L2, L3], name='3-DOF Robot')

# Display the DH table (this prints the DH parameters)
print(robot)

# Joint configuration (angles in radians for each link)
q = [0, 0, 0] # You can modify the values to change the robot's posture

# Calculate forward kinematics (the final transformation of the end-effector)
T = robot.fkine(q)
print(f"End-effector position (forward kinematics): \n{T}")
```

```python
# Inverse kinematics
# Define the desired pose for the end-effector
Tep = SE3.Trans(0, 0, 3.5) # Target pose (x=1, y=0.8, z=0.5)

# Solve inverse kinematics to find the joint angles for the desired pose
q_inv = robot.ikine_LM(Tep) # Levenberg-Marquardt IK solver
# Extract the joint configuration that achieves the desired pose
q_solution = q_inv.q
print(f"Joint angles (inverse kinematics solution): {q_solution}")

# Plot the robot in the configuration obtained from inverse kinematics
robot.plot(q_solution, block=True)

# Interactive tool to visualize the robot's joint states and DH parameters
robot.teach(jointlabels=1) # Opens an interactive window with DH axes
```