EENG 428 Introduction to Robotics Laboratory

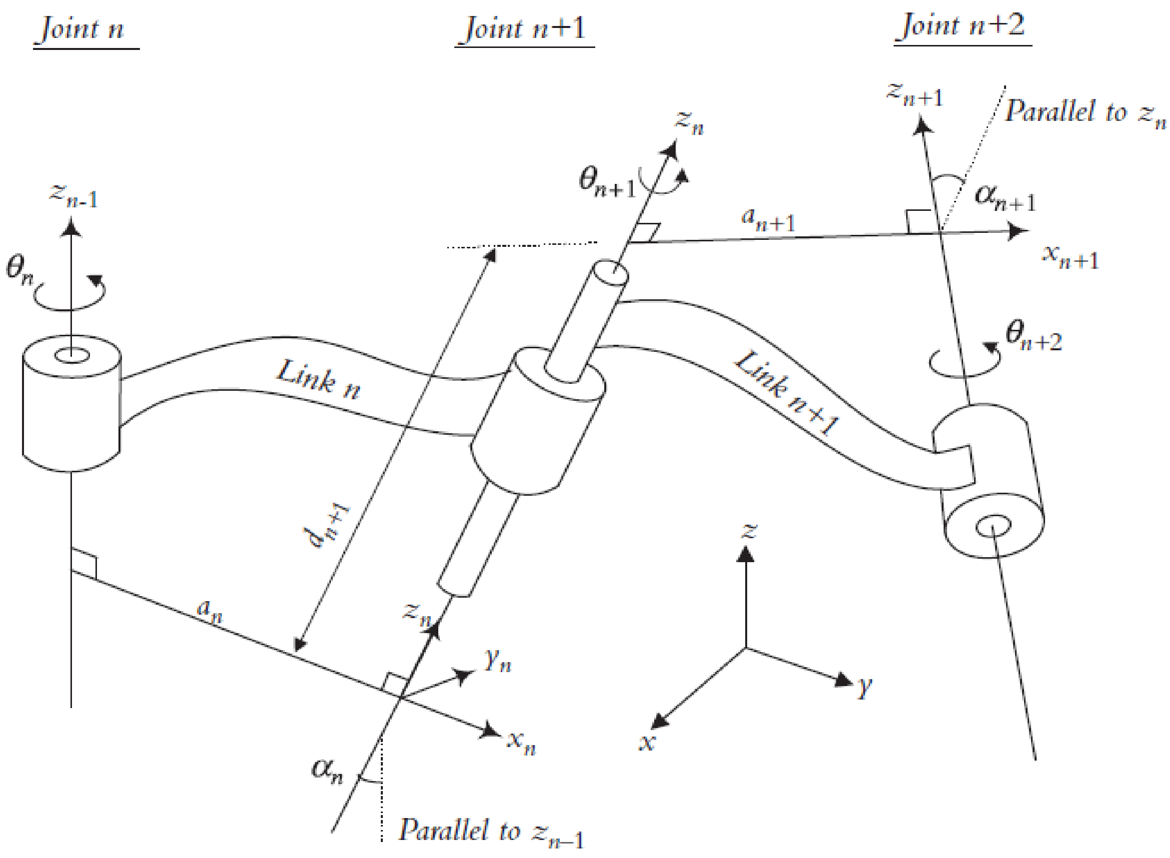
Lab Session 4

Objective

In this Lab session, Denavit-Hartenberg Representation of Forward Kinematic Equations for Robots is to be discussed with the assistance of the Matlab program to serve the purpose of this session.

Denavit-Hartenberg Representation

Denavit-Hartenberg convention is used to facilitate and standardize the way of representing two or more sequential frames of serial robotic manipulators. To model the robot with the D-H representation, we need to define the general procedure to assign reference frames to each joint. Then we will define how a transformation between any two successive frames may be accomplished. Finally, we will write the total transformation matrix for the robot. The following figure shows three joints. Each joint may be rotate or translate.



As shown in figure, we assign joint number n to the first joint, $n+1$ to the second joint, and $n+2$ to the third joint. There may be other joints before or after these. Each link is also assigned a link number as shown. Link n will be between joints n and $n+1$, and link $n+1$ is between joints $n+1$ and $n+2$.

To model the robot with the D-H representation, the first thing we need to do is assign a local reference frame for each and every joint. Therefore, for each joint, we will have to assign a z-axis and an x-axis. We normally do not need to assign a y-axis, since we always know that y-axes are mutually perpendicular to both x- and z-axes.

Modeling the robot with the D-H representation summarized by the following three steps

1- Assigning a local reference frame to each joint

All joints are represented by a z-axis. 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 is $n-1$. For example, the z-axis representing motions about joint number 1 is z_0.

In general, joints may not necessarily be parallel or intersecting. As a result, the z-axes may be *skew* lines. There is always one line *mutually perpendicular* to any two *skew* lines, called *common normal*, which is the shortest distance between them. We always assign the x-axis of the local reference frame in the direction of the common normal. Therefore, if a_n represents the common normal between z_{n-1}. and z_n, the direction of x_n. will be along a_n.

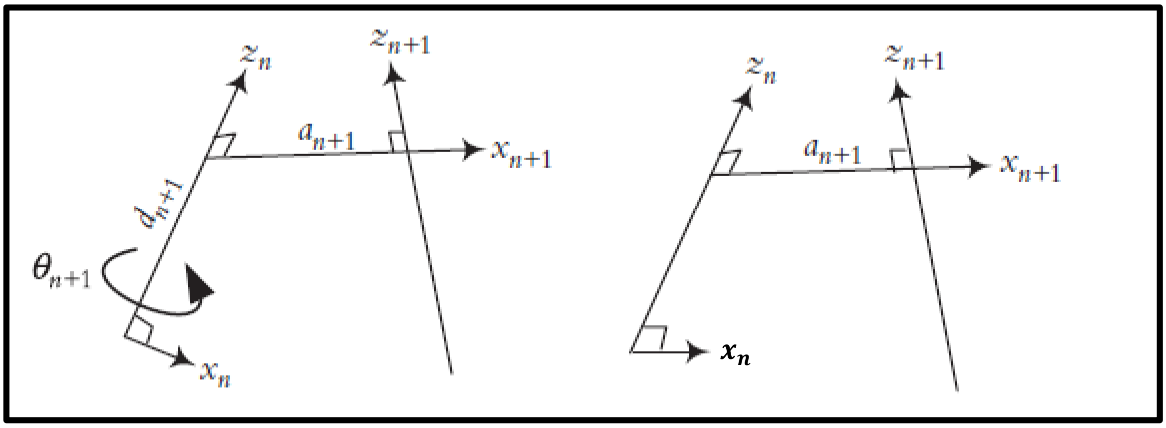
- If two z-axes are parallel, there are an infinite number of common normal between them. We will pick the common normal that is collinear with the common normal of the previous joint.

- If the z-axes of two successive joints are intersecting, there is no common normal between them. We will assign the x-axis along a line perpendicular to the plane formed by the two z-axes.

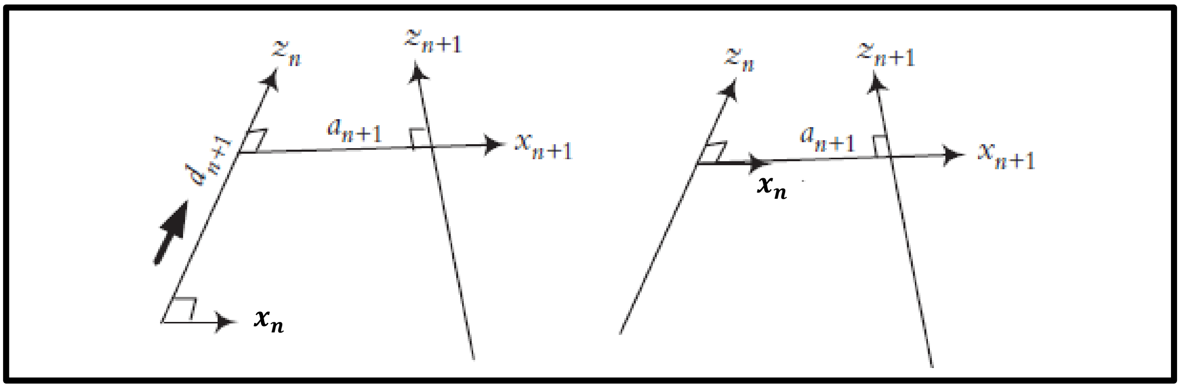
2- Define Transformation between any two successive frames

Assuming we are at the local reference frame $x_n - z_n$, we will do the following four standard motions to get to the next local reference frame $x_{n+1} - z_{n+1}$

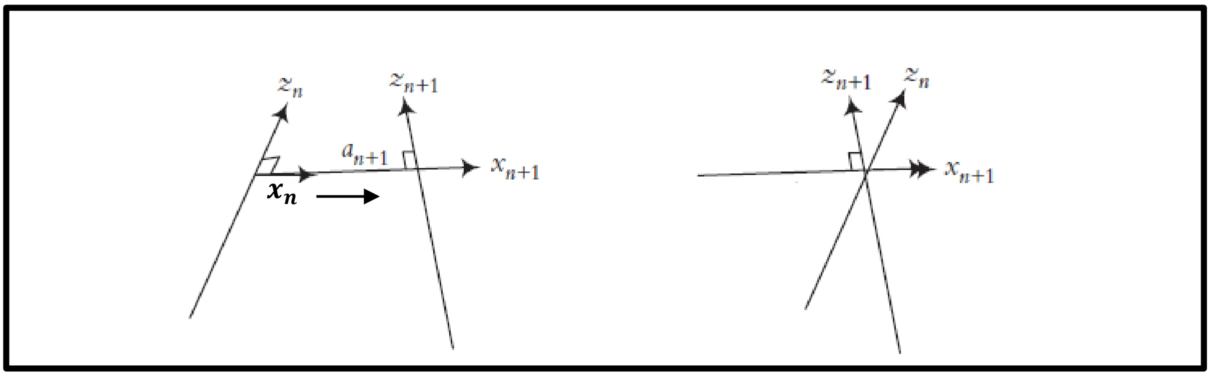
- Rotate about the z_n axis an angle θ_{n+1} this will make x_n and x_{n+1} parallel to each other.



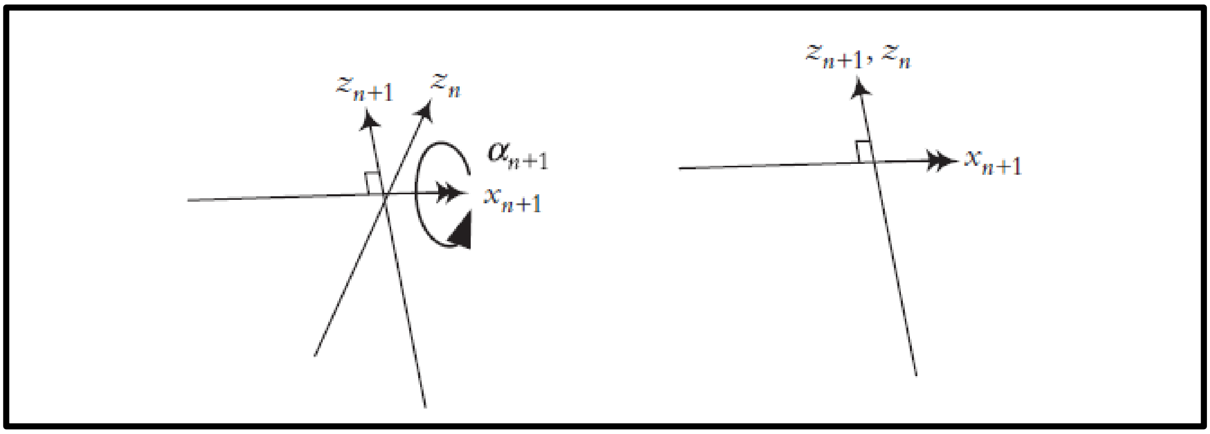
- Translate along the z_n axis a distance of d_{n+1} to make x_n and x_{n+1} collinear.



- Translate along the (already rotated) x_n axis a distance of a_{n+1} to bring the origins of the two reference coordinates together



- Rotate about x_{n+1} axis an angle of α_{n+1} to align z_n axis with z_{n+1} axis. At this point, frames n and $n+1$ will be exactly the same and we have transformed from one to the next.



The transformation ${}^{n}T_{n+1}$ called(A_{n+1}) between two successive frames representing the preceding four movements is the product of the four matrices representing them. Since all transformations are relative to the current frame, all matrices are post-multiplied.

$$A_{n+1} = \begin{bmatrix} C\theta_{n+1} & -S\theta_{n+1}C\alpha_{n+1} & S\theta_{n+1}S\alpha_{n+1} & a_{n+1}C\theta_{n+1} \\ S\theta_{n+1} & C\theta_{n+1}C\alpha_{n+1} & -C\theta_{n+1}S\alpha_{n+1} & a_{n+1}S\theta_{n+1} \\ 0 & S\alpha_{n+1} & C\alpha_{n+1} & d_{n+1} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Using Matlab, the D-H matrix (the A matrix) can be written as

```
syms t al a d
A= trotz(t)*transl(a,0,d)*trotx(al)

A =

[ cos(t), -cos(al)*sin(t),  sin(al)*sin(t), a*cos(t)]
[ sin(t),  cos(al)*cos(t), -sin(al)*cos(t), a*sin(t)]
[      0,         sin(al),         cos(al),        d]
[      0,               0,               0,        1]
```

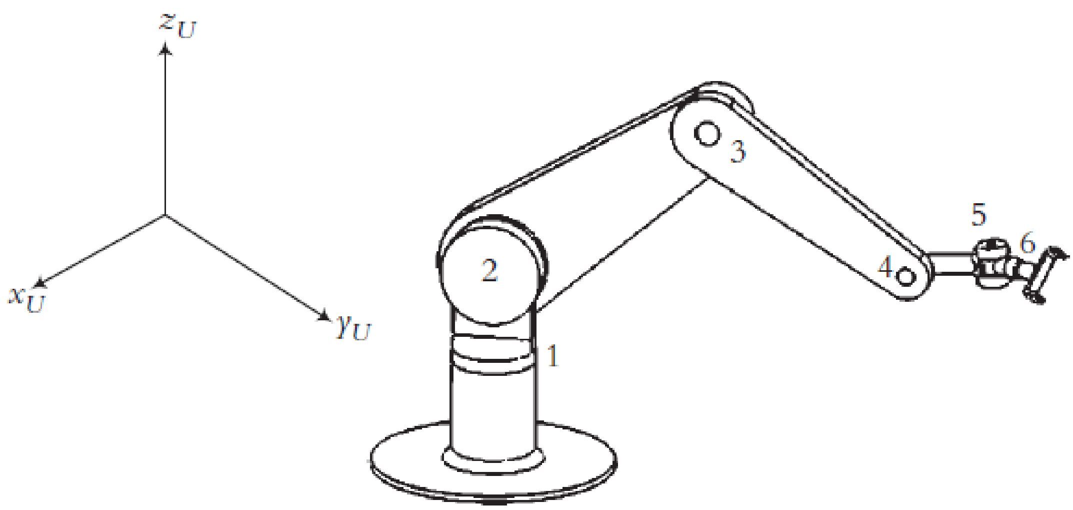
3- Total transformation matrix for the robot

The complete transformation matrix that represent the relation between the base of the robot and the hand effector calculated by multiplying the transformations between any two successive frames starting with the first joint and transform to the second joint, and so on, until the hand of the robot and eventually the end effector. The total transformation between the base of the robot and the hand given below

$$^{R}T_{H} = {}^{R}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}\ldots{}^{n-1}T_{n} = A_1A_2A_3\ldots A_n$$

Example:

For the simple 6-DOF robot assign the necessary coordinate frames based on the D-H representation, fill out the accompanying parameters table, and derive the forward kinematic equation of the robot.



D-H Parameters Table for Example

#	θ	d	a	α
0–1	θ_1	***d*1**	0	90
1–2	θ_2	0	a_2	0
2–3	θ_3	0	a_3	0
3–4	θ_4	0	a_4	−90
4–5	θ_5 +90	0	0	90
5–6	θ_6	0	0	0

The following matlab code aims to find (Forward Kinematics of the robot) total transformation between the base of the robot and the hand using D-H Representation.

```
syms t al a d
A= trotz(t)*transl(a,0,d)*trotx(al);
syms t1 t2 t3 t4 t5 t6 a2 a3 a4 d1
A1=subs(A,[t d a al],[t1 d1 0 pi/2])
A2=subs(A,[t d a al],[t2 0 a2 0])
A3=subs(A,[t d a al],[t3 0 a3 0])
A4=subs(A,[t d a al],[t4 0 a4 -pi/2])
A5=subs(A,[t d a al],[(t5+pi/2) 0 0 pi/2])
A6=subs(A,[t d a al],[t6 0 0 0])
Atotal=A1*A2*A3*A4*A5*A6;
Atotal=simplify(Atotal)
```

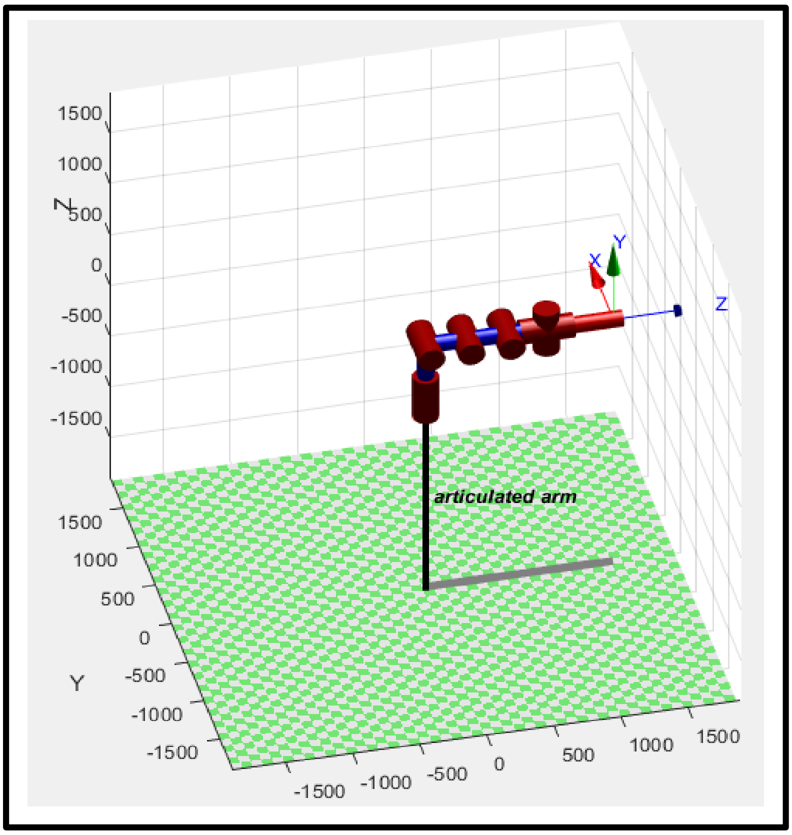
```
clc
clear all;
%% 6 Dof articulated arm manipulator
d1=500
a2=300;
a3=300;
a4=300;
L(1)=Link([0      d1      0      pi/2   ]);
L(2)=Link([0      0       a2      0     ]);
L(3)=Link([0      0       a3      0     ]);
L(4)=Link([0      0       a4    -pi/2   ]);
L(5)=Link([0      0       0      pi/2    0    pi/2 ]);
L(6)=Link([0     500      0       0     ]);
robot=SerialLink(L);
T = fkine(robot, [0 0 0 0 0 0])
```

```
robot.name='articulated arm';
%% modeling and simulation of the robot
robot.plot([pi pi 0 0 pi/2 0]);
for th1=pi:-pi/20:0
robot.plot([th1 pi 0 0 pi/2 0]);
pause(0.05)
end

robot.plot([0 pi 0 0 pi/2 0]);
for th2=pi:-pi/20:0
robot.plot([0 th2 0 0 pi/2 0]);
pause(0.05)
end

for th3=pi/2:-pi/20:0
robot.plot([0 0 0 0 th3 0]);
pause(0.05)
end

for th5=pi/2:pi/20:0
robot.plot([0 0 0 0 th5 0]);
pause(0.05)
end
robot.plot([0 0 0 0 0 0]);
```



Example:

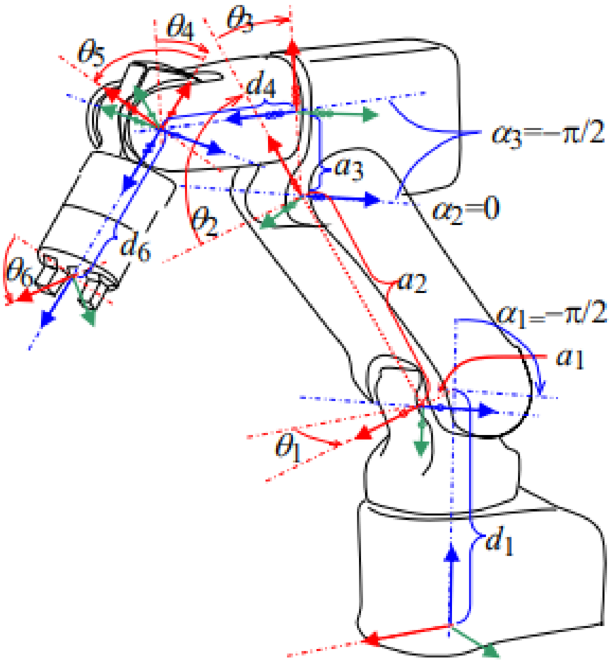
The link parameters (in mm and radians) of the Mitsubishi Robot are

a1 =85, a2 = 280, a3 =100, d1 = 350, d4 =315, d6 =85.

Simulate the perspective view to perform following movements in Matlab, using Robot-Toolbox along the following path.

a) Start from q0 = (θ1 , θ2 , θ3 , θ4 , θ5 , θ6) = [-π/2 0 0 π/2 0 0]
b) move to q1=[-π 0 0 π/2 0 0] in 10 steps
c) then, continue to move to q2 = [-π -π/2 0 π/2 0 0] in 10 steps c) then, continue to move to q0 = [-π/2 0 0 π/2 0 0].

The configuration and the D-H table are given below

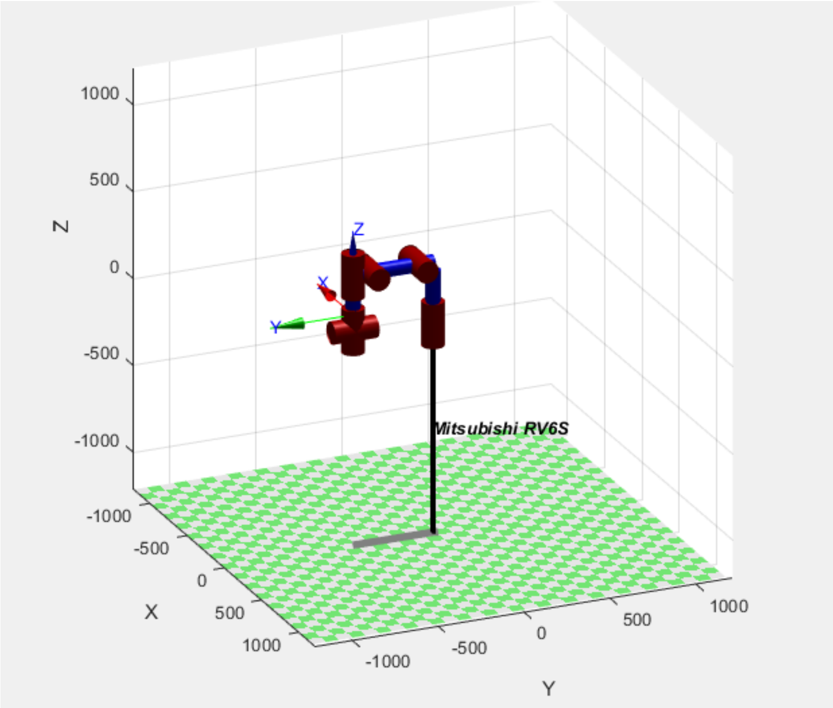


#	α	a	θ	d	σ
1	−π/2	a_1	θ_1	d_1	0
2	0	a_2	θ_2	0	0
3	−π/2	a_3	θ_3	0	0
4	−π/2	0	θ_4	d_4	0
5	−π/2	0	θ_5	0	0
6	0	0	θ_6	d_6	0

```
%% Mitsubishi RV6S manipulator
a1=85;
a2=280;
a3=100;
d1=350;
d4=315;
d6=85;
%          alfa     d       a       theta
L(1)=Link([0       d1      a1      -pi/2     ]);
L(2)=Link([0       0       a2        0       ]);
L(3)=Link([0       0       a3      -pi/2     ]);
L(4)=Link([0       d4      0       -pi/2     ]);
L(5)=Link([0       0       0       -pi/2     ]);
L(6)=Link([0       d6      0         0       ]);

robot=SerialLink(L);
robot.name='Mitsubishi RV6S';

%% modeling and simulation of the robot
robot.plot([-pi/2 0 0 pi/2 0 0]);
for th1=-pi/2:-pi/20:-pi
robot.plot([th1 0 0 pi/2 0 0]);
pause(0.1)
end
for th2=0:-pi/20:-pi/2
robot.plot([-pi th2 0 pi/2 0 0]);
pause(0.1);
end
th1=-pi:pi/20:-pi/2;
th2=-pi/2:pi/20:0;
for k=1:numel(th1)
TH1=th1(k);
TH2=th2(k);
robot.plot([TH1 TH2 0 pi/2 0 0])
pause(0.1);
end
```



Homework

For the SCARA-type robot shown in the next figure,

1- Assign coordinate frames for its links based on D-H representation.

2- Fill out the D-H parameters table.

3- Write all the Matrices separately.

4- Find the transformation matrix for the tool-frame coordinates w.r.t the base (the product of the A matrices).