SCARA ROBOT

ROBOTIK PROJECT

MCTE 4352

SEM 1, 2020/2021

NAME: MUHAMMAD ASYRAAF BIN NOOR AZAMAN

MATRIC CARD: 1716133

SECTION: 1

LECTURER: DR. TANVEER SALEH

DATE: 19 DECEMBER 2020[i]

TABLE OF CONTENT

1.0 INTRODUCTION

The SCARA (Selection Compliance Assembly Robot Arm) is also known as a horizontal articulated arm robot. SCARA robots are well used in factories to do operation like the pick and place motion, assembly, and other operations. The 4DOF SCARA which a well-known robot where the robot rotates about all three axes and has sliding motion along one axis in combination with rotation about another. In figure 1 below show the example of SCARA in real life.

Figure 1: Example of SCARA robot.

The workspace of SCARA robot is identical to those of the cylindrical, but SCARA can perform operations where the angle is close to the center of the workspace and cylindrical cannot perform it. Below figure 2 shows the workspace of SCARA robot.

Figure 2: workspace of the SCARA robot.

2.0 SCARA CONFIGURATION

This SCARA configuration is where to determine the joint angle of each joint for the future operation. Figure 3 shows the drawing of 4DOF SCARA robot.

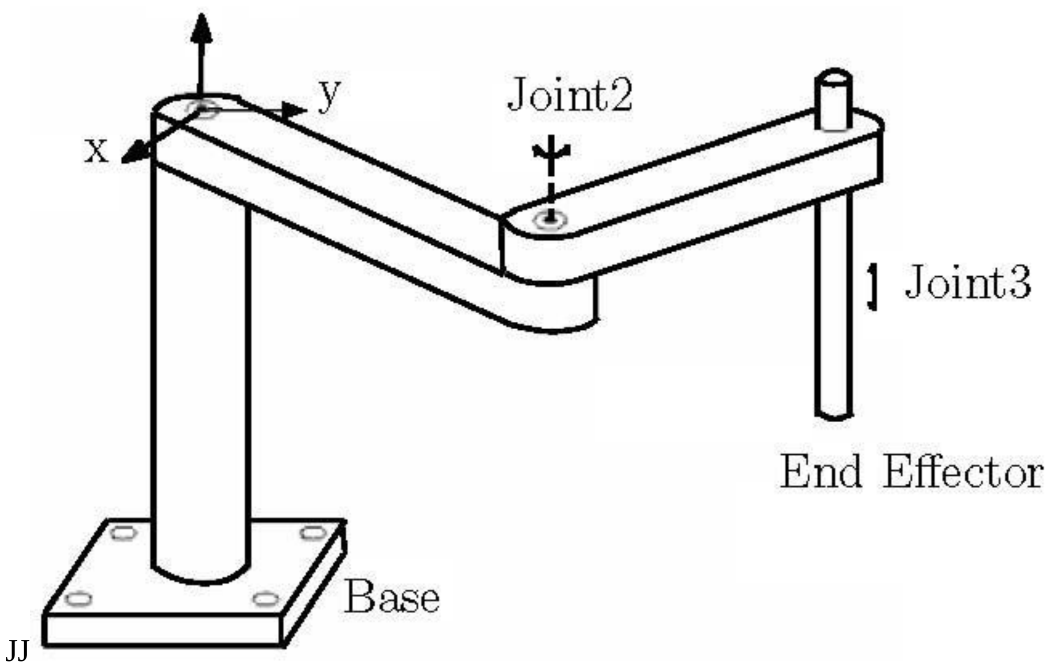


Figure 3: 4DOF SCARA robot

The DH parameters can be gotten from the set rules as labelled in figure 4. This is where the axis is determine based on the world reference. For example, the direction and angle of next axis respected to the past axis. Right hand rule is used to assist the determining the parameters.

Finding DH parameters

- a_i is distance from z_{i-1} to z_i measured along x_i.
- angle α_i is angle from z_{i-1} to z_i measured about x_i.
- d_i is distance from x_{i-1} to x_i measured along z_{i-1}.
- angle θ_i is from x_{i-1} to x_i measured about z_{i-1}.

Figure 4: Rules to find the DH parameters.

For figure 5 it is the individual transformations and are post multiplied to get the final transformations matrix like in the Figure 6. In Figure 7, x, and y used to find the theta_1 and theta_2 like in inverse kinematics where given x and y to find joint angles and for forward kinematics, joint angles are set up to determine the position of the ending factor.

$$
\begin{aligned}
A_i &= R_{z,\theta_i}\mathrm{Trans}_{z,d_i}\mathrm{Trans}_{x,a_i}R_{x,\alpha_i} \qquad (3.10)\\
&= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
$$

Figure 5: The individual of transformations matrix.

$$\begin{bmatrix} \cos(\theta_1+\theta_2+\theta_4) & -\sin(\theta_1+\theta_2+\theta_4) & 0 & a_2\cos(\theta_1+\theta_2)+a_1\cos(\theta_1) \\ \sin(\theta_1+\theta_2+\theta_4) & \cos(\theta_1+\theta_2+\theta_4) & 0 & a_2\sin(\theta_1+\theta_2)+a_1\sin(\theta_1) \\ 0 & 0 & 1 & d_1-d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 6: The Final Transformations Matrix of 4DOF SCARA robot.

$$x = a_2\cos(\theta_1+\theta_2) + a_1\cos(\theta_1)\text{..............}(1)$$

$$y = a_2\sin(\theta_1+\theta_2) + a_1\sin(\theta_1)\text{...............}(2)$$

Figure 7: Analytical solution to find the theta1 and theta2

3.0 DISCUSSION

3.1 SCARA DRAWING

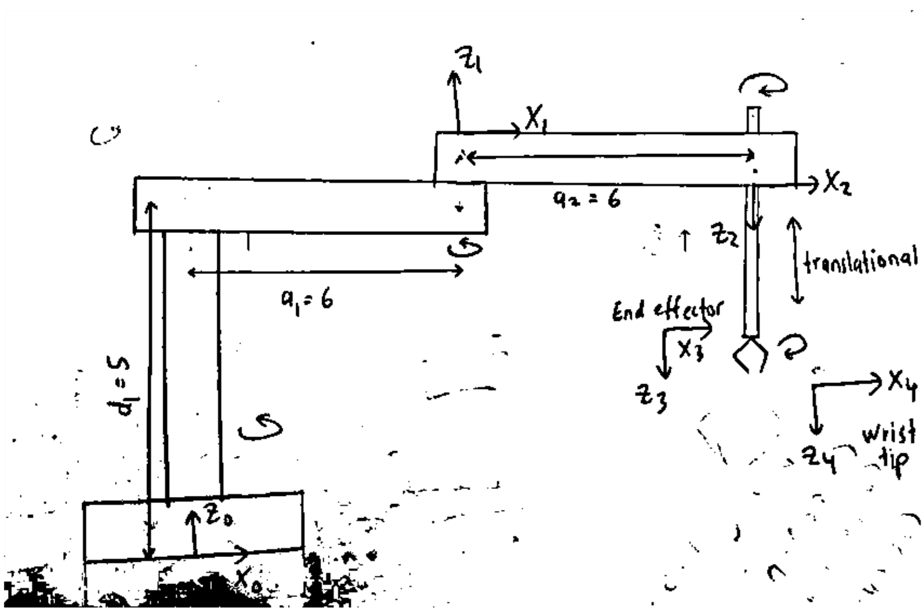


Figure 8: Axes Of SCARA

3.2 DH TABLE

LINK	a_i	α_i	d_i	θ_i
0 – 1	a_1	0	d_1	θ_1^*
1 – 2	a_2	180	0	θ_2^*
2 – 3	0	0	$-d_3$	0
3 – 4	0	0	0	θ_4^*

Figure 9: DH Parameters

3.3 ANALYTICAL SOLUTION

After got the DH table we use the final transformation matrix like in Figure 6, we take the 4th column where it represents the position the x, y, and z of the ending factor. Then using the mathematical knowledge, we derive the equation of x and y to find the theta1 and theta2 formula. The step is shown like in Figure 10 to Figure 12. Thus, the individual joint angles can be found.

$$y^2 = a_2^2 \cos^2(\theta_1+\theta_2) + a_1^2 \cos^2\theta_1 + 2a_2a_1\cos(\theta_1+\theta_2)\cos\theta_1$$

$$\rightarrow 2a_2a_1[(C_1C_2 - S_1S_2)]C_1 \rightarrow 2a_2a_1 C_1^2C_2 - \cancel{2}a_2a_1 S_1S_2C_1$$

$$y^2 = a_2^2\sin^2(\theta_1+\theta_2) + a_1^2\sin^2\theta_1 + 2a_2a_1\sin(\theta_1+\theta_2)\sin\theta_1$$

$$\rightarrow 2a_2a_1[(S_1C_2 + C_1S_2)]S_1 \rightarrow 2a_1a_2 S_1^2C_2 + \cancel{2}a_2a_1 C_1S_2S_1$$

$$x = a_2 C_{12} + a_1 C_1 \quad -①$$

$$y = a_2 S_{12} + a_1 S_1 \quad -②$$

$$x^2 + y^2 = a_2^2(1) + a_1^2(1) + 2a_1a_2C_2[\overset{1}{\cancel{C_1^2 + S_1^2}}] \quad -③$$

③

$$\theta_2 = \pm\cos^{-1}\left[\frac{x^2+y^2-a_2^2-a_1^2}{2a_1a_2}\right]$$

⑦

$$\frac{x}{C_1} = \frac{a_2C_{12} + a_1}{C_1} \qquad \frac{y}{C_1} = \frac{a_2S_{12} + a_1S_1}{C_1}$$

$$\frac{x}{C_1} = \frac{a_2\cos(\theta_1+\theta_2)}{C_1} + a_1 \qquad = \frac{a_2[S_1C_2 + C_1S_2] + a_1S_1}{C_1}$$

$$= a_1 + \frac{a_2}{C_1}(C_1C_2 - S_1S_2) \qquad y = a_2C_2t_1 + a_2S_2 + a_1t_1 \quad -⑤$$

$$x = a_1 + a_2C_2 - a_2S_2t_1 \quad -④$$

④:⑤

$$\frac{x}{y} = \frac{a_1 + a_2C_2 - a_2S_2t_1}{a_2C_2t_1 + a_2S_2 + a_1t_1}$$

$$\rightarrow a_2xC_2t_1 + a_2S_2x + a_1t_1x = a_1y + a_2yC_2 - a_2yS_2t_1$$

$$t_1(a_2xC_2 + a_1x + a_2yS_2) = a_1y + a_2y - a_2S_2x$$

$$t_1 = \frac{a_1y + a_2yC_2 - a_2S_2x}{a_2xC_2 + a_1x + a_2yS_2} \quad -⑥$$

$a_1 = 0.4$

$a_2 = 0.4$

① & ⑦ to find θ_1 & θ_2

Figure 10: Analytical solution to find x and y or theta1 and theta2 (Equation 3 and 6)

$$x = a_2\cos(\theta_1+\theta_2) + a_1\cos(\theta_1) \text{.............}(1)$$

$$y = a_2\sin(\theta_1+\theta_2) + a_2\cos(\theta_1) \text{...............}(2)$$

$$z = d_1 - d_3 \text{...}(3)$$

$(1)^2$+$(2)^2$

$$x^2 + y^2 = {a_1}^2 + {a_2}^2 + 2a_1a_2\cos(\theta_2)$$

$$hence, \theta_2 = \pm\cos^{-1}\frac{x^2 + y^2 - {a_1}^2 - {a_2}^2}{2a_1a_2}$$

Figure 11: Analytical solution to find x and y or theta1 and theta2

$$\frac{x}{\cos(\theta_1)} = \frac{a_2\cos(\theta_1+\theta_2) + a_1\cos(\theta_1)}{\cos(\theta_1)}$$

$$\frac{x}{\cos(\theta_1)} = a_1 + a_2\cos(\theta_2) - a_2\sin(\theta_2)\tan(\theta_1)$$

$$\frac{y}{\cos(\theta_1)} = a_1\tan(\theta_1) + a_2\cos(\theta_2)\tan(\theta_1) + a_2\sin(\theta_2)$$

$$\frac{x}{y} = \frac{a_1 + a_2\cos(\theta_2) - a_2\sin(\theta_2)\tan(\theta_1)}{a_1\tan(\theta_1) + a_2\cos(\theta_2)\tan(\theta_1) + a_2\sin(\theta_2)}$$

$$\tan(\theta_1) = \frac{a_1y + a_2y\cos(\theta_2) - a_2x\sin(\theta_2)}{a_1x + a_2x\cos(\theta_2) + a_2y\sin(\theta_2)}$$

Figure 12: Analytical solution to find x and y or theta1 and theta2

3.4 HAND CALCULATION

For T1) ($x = 7$, $y = 7$) $\rightarrow$ for continuous.

$a_1 = a_2 = 6$

$\boxed{Eq(3)}$

$$\theta_2 = \pm \cos^{-1}\left[\frac{(7)^2 + (7)^2 - (6)^2 - (6)^2}{2(6)(6)}\right]$$

$$= 1.201337$$

$\boxed{Eq(6)}$

$$\theta_1 = \tan^{-1}\left[\frac{(6)(7) + (6)(7)\cos(\theta_2) - (6)\sin(\theta_2)(7)}{(6)(7)\cos(\theta_2) + (6)(7) + (6)(7)\sin(\theta_2)}\right]$$

$$= 0.18477$$

Figure 13: Hand calculation for the first transformation (7,7)

3.5 EXCEL CALCULATION

Continuous path			EXCEL_CALCULATION	
x	y	z	theta1_radians	theta2_radians
7	7	7	0.184729565	1.201337197
7	-7	1.5	-1.386066762	1.201337197
-7	-7	-3	0.184729565	1.201337197
-7	7	6	-1.386066762	1.201337197

Table 1: Continuous table

Discrete path			EXCEL_CALCULATION	
x	y	z	theta1_radians	theta2_radians
4	4	5	-0.294515485	2.159827297
4	-4	5	1.276280842	2.159827297
-4	-4	5	-0.294515485	2.159827297
-4	4	5	1.276280842	2.159827297

Table 2: Discrete table

3.6 ROBOTIC TOOLBOX CALCULATION

	x	y	z	theta1_radians	theta2_radians
T1	7	7	7	0.1847	1.2013
T2	7	-7	1.5000	0.3395	2.1630
T3	-7	-7	-3	-1.7555	-1.2013
T4	-7	7	6	-0.6391	-2.1630

Table 3: Continuous table

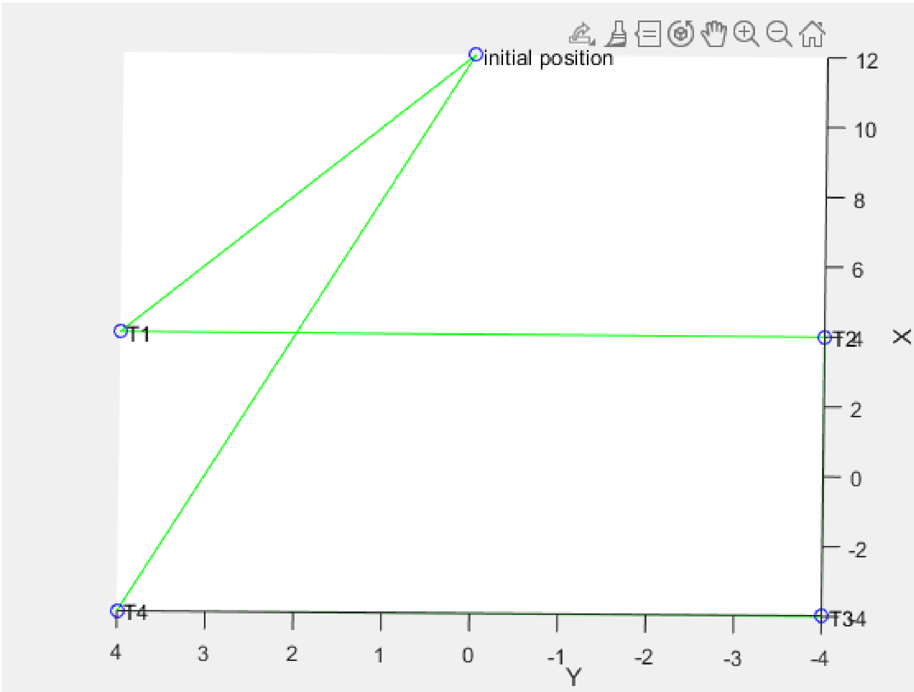
	x	y	z	theta1_radians	theta2_radians
T1	4	4	5	-0.2945	2.1598
T2	4	-4	5	-1.8653	2.1598
T3	-4	-4	5	2.8471	2.1598
T4	-4	4	5	1.2763	2.1598

Table 4: Discrete table

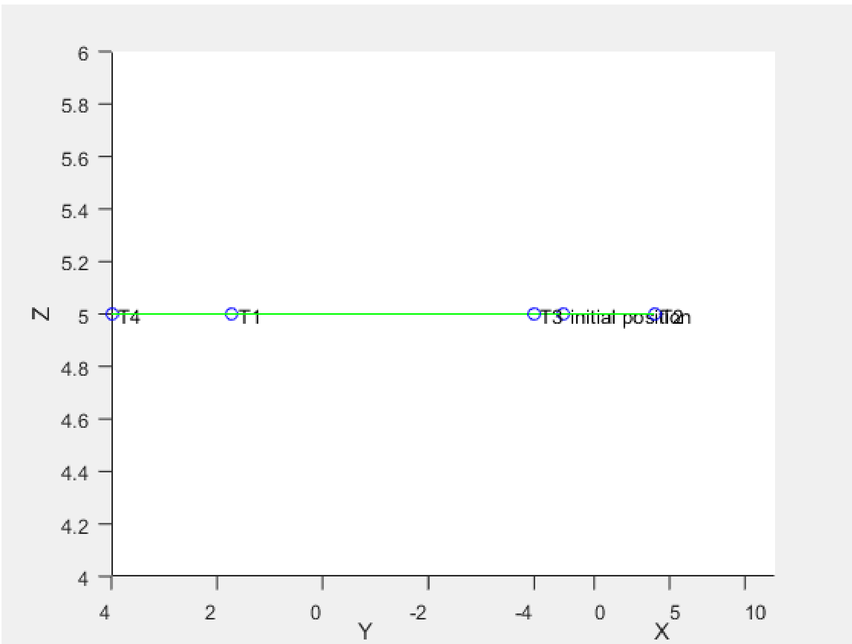
As you can see in the Table (1-4) of Continuous and Discrete of both by Excel calculation and Robotic Toolbox calculation, the theta 2 of the discrete from both calculations got the same answer however the theta 2 in continuous path, there is some differ in the value in excel to the Robotic toolbox. But the difference is at the value of theta 1 where the value in excel is differed from the Robotic calculation.

The reason is because of the theta 2 in excel are calculated from the base reference angle but in Matlab the theta 2 are being respect to the previous joint angle. Thus because of this, the theta 1 of both continuous and discrete in excel will be different to those in robotic toolbox where theta 2 is the one controlled the angle of theta 1.

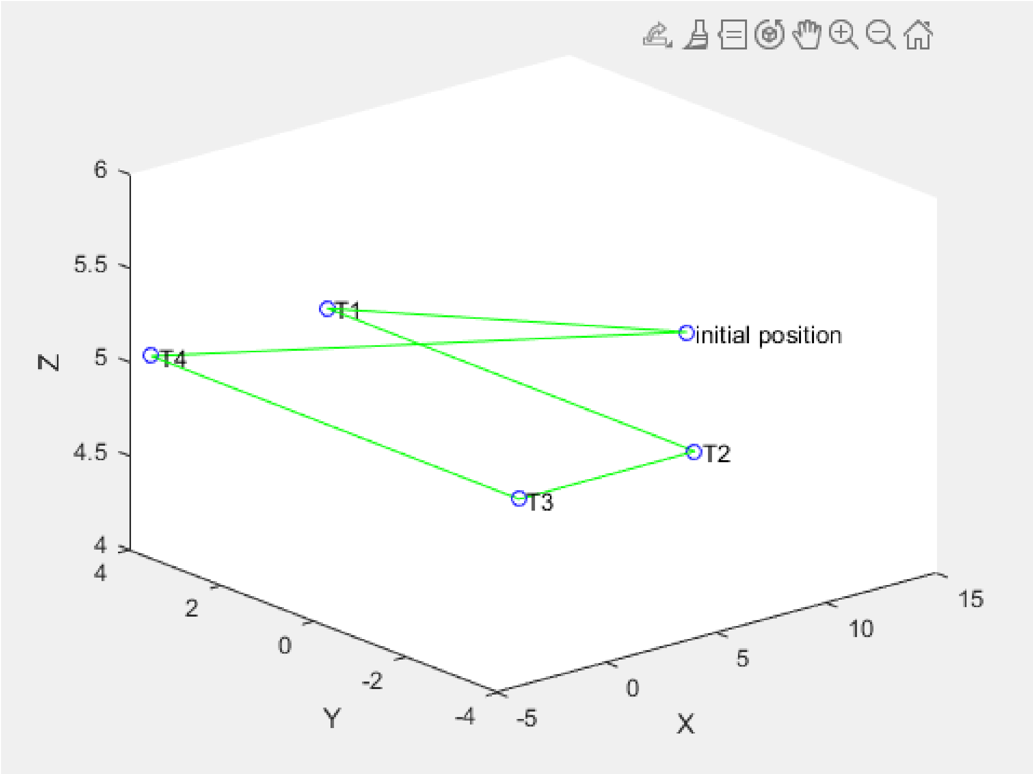
For the trajectory, due to my low skills and knowledge, I cannot trace the actual path or display the actual trajectory path on how the SCARA robot in closed loop. However, I did an alternative way where the position for each transformation are plotted in 3d view like in Graph (1-6).



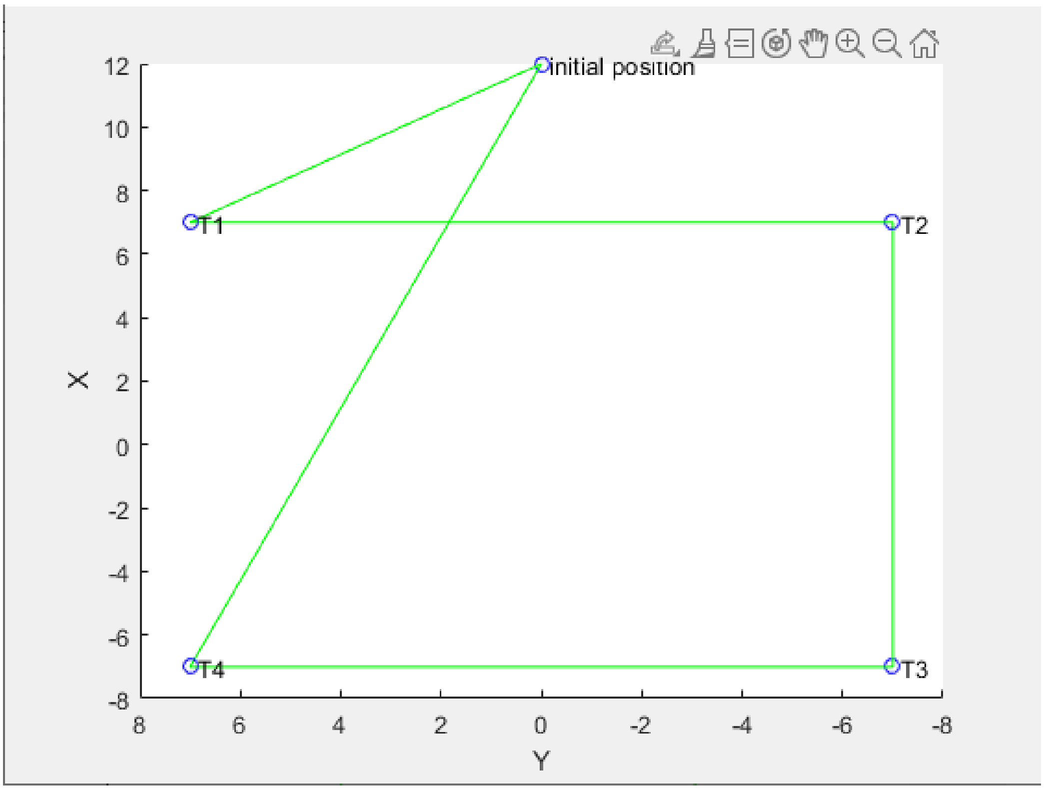
Graph 1: Discrete Trajectory path of 2D view (X and Y axis)



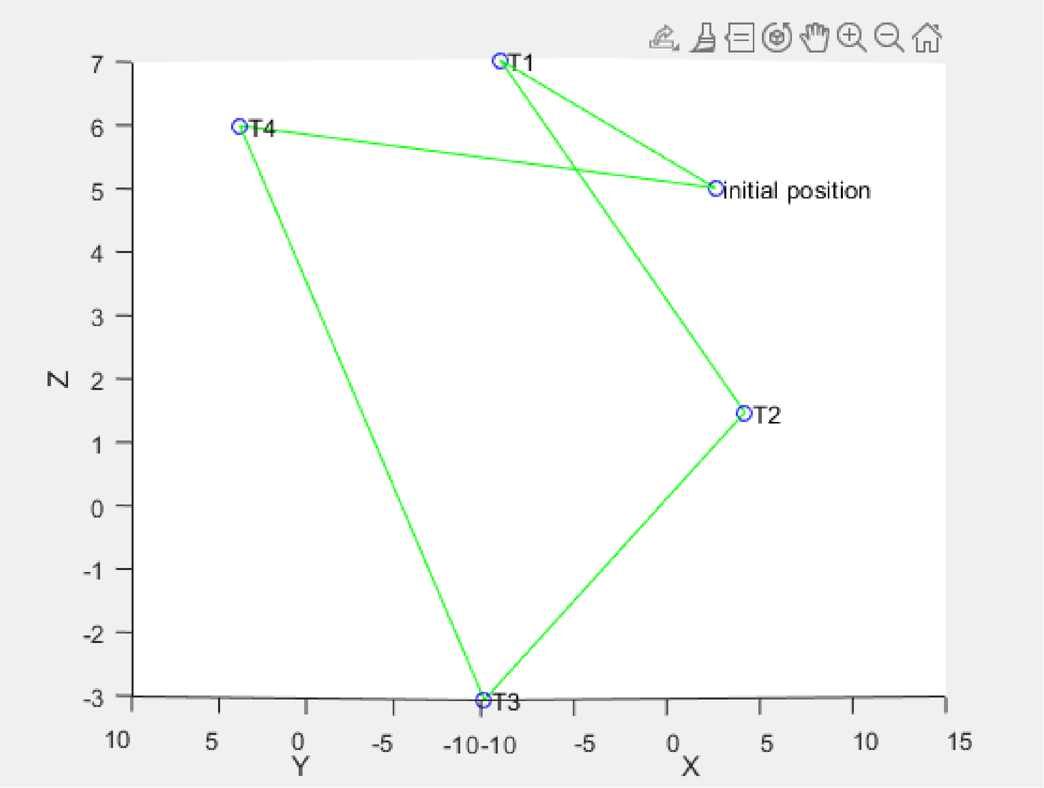
Graph 2: Discrete Trajectory path of 3D side view (X, Y, and Z axis)



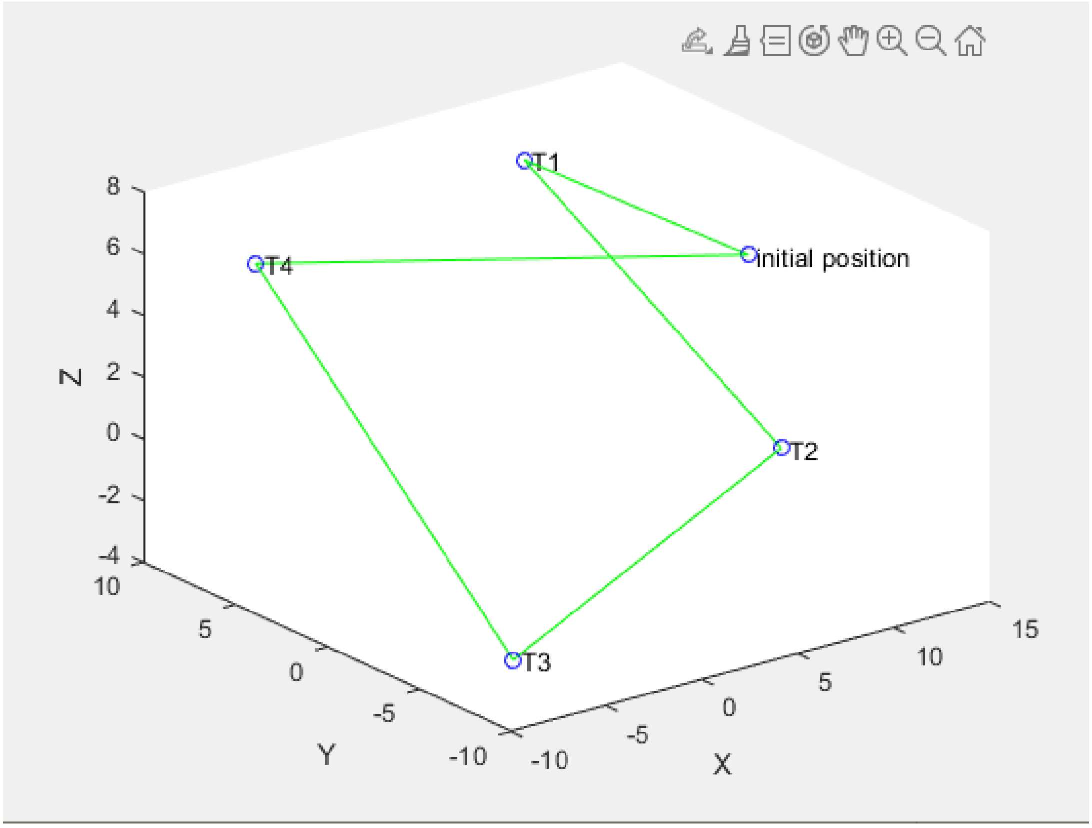
Graph 3: Discrete Trajectory path of 3D angle view (X, Y, and Z axis)



Graph 4: Continuous Trajectory path of 2D view (X, and Y axis)



Graph 5: Continuous Trajectory path of 3D side view (X, Y, and Z axis)



Graph 6: Continuous Trajectory path of 3D angle view (X, Y, and Z axis)

3.7 CODING

```
%% DH parameters
d1 = 5;
%d3 = 2.5;
a1 = 6;
a2 = 6;

%% Link Description

% Link([theta, d, a, alpha, R/P])
L(1) = Link([0,d1,a1,0,0],'standard');%theta 1
L(2) = Link([0,0,a2,pi,0],'standard');%theta 2
L(3) = Link([0,0,0,0,1],'standard');%d3%prismatic
L(3).qlim = [0,10];
L(4) = Link([0,0,0,0,0],'standard');%theta 4

%% SerialLink to make robot

robot = SerialLink(L,'name','SekaRA Robot');
% robot.plot([0 0 0 0],'workspace',[-15 15 -15 15 -15 15]);
% robot.teach

%% get path

prompt = 'Choose the path flow whether in Continous,C or Discrete,D
,(C=1 or D=0) \nuser input : ';
a = input(prompt);
matrix = getmatrix(a);
```

```
%% Transformation Matrix

qinit = [0 0 0 0];%initial position x = 12, y = 0, z = 5

T1 = transl(matrix(1,1),matrix(1,2),matrix(1,3));
T2 = transl(matrix(2,1),matrix(2,2),matrix(2,3));
T3 = transl(matrix(3,1),matrix(3,2),matrix(3,3));
T4 = transl(matrix(4,1),matrix(4,2),matrix(4,3));


%% use inverse kinematics to get joint angle

q1 = robot.ikine(T1,qinit,[1,1,1,0,0,0]);
q2 = robot.ikine(T2,q1,[1,1,1,0,0,0]);
q3 = robot.ikine(T3,q2,[1,1,1,0,0,0]);
q4 = robot.ikine(T4,q3,[1,1,1,0,0,0]);

%% use forwardkinematics to find the x,y,z ( to verify the joinnt angle that got from the inverese kinematics )

q1_T1 = robot.fkine(q1);
q2_T2 = robot.fkine(q2);
q3_T3 = robot.fkine(q3);
q4_T4 = robot.fkine(q4);

%% Animation

```

```
t =(0: .05:3)'; %time vector for where one transformation took 3 seconds + pause(0.02) before undergo next transformations;

% The trajectory of this scara is :
% qinit -> q1 -> q2 -> q3 -> q4 -> q1 -> qinit


% First Transformation = initial position to 1st position
transformation1 = jtraj(qinit,q1,t);
robot.plot(transformation1,'workspace',[-15 15 -15 15 -15 15]);
pause(0.02);

% Second Transformation = 1st position to 2nd position
transformation2 = jtraj(q1,q2,t);
robot.plot(transformation2,'workspace',[-15 15 -15 15 -15 15]);
pause(0.02);

% Third Transformation = 2nd position to 3rd position
transformation3 = jtraj(q2,q3,t);
robot.plot(transformation3,'workspace',[-15 15 -15 15 -15 15]);
pause(0.02);

% Fourth Transformation = 3rd position to 4th position
transformation4 = jtraj(q3,q4,t);
robot.plot(transformation4,'workspace',[-15 15 -15 15 -15 15]);
pause(0.02);

% Fifth Transformation = 4th position to 1st position
```

```
transformation5 = jtraj(q4,q1,t);
robot.plot(transformation5,'workspace',[-15 15 -15 15 -15 15]);
pause(0.02);

% Sixth Transformation = 1st position to initial position
transformation6 = jtraj(q1,qinit,t);
robot.plot(transformation6,'workspace',[-15 15 -15 15 -15 15]);
pause(0.02);

%% to plot the trajectory path

trajectory = [12 0 5;
            matrix(1,:);
            matrix(2,:);
            matrix(3,:);
            matrix(4,:);
                12 0 5];
    [nx,ny] = size(trajectory) ;
    figure
    hold on

    for i = 1:nx-1
        v=[trajectory(i,:);trajectory(i+1,:)];
        plot3(v(:,1),v(:,2),v(:,3),'g');
        plot3(v(:,1),v(:,2),v(:,3),'bo')
    end

```

```
text(trajectory(1,1),trajectory(1,2),trajectory(1,3), ' initial position');
text(trajectory(2,1),trajectory(2,2),trajectory(2,3), ' T1');
text(trajectory(3,1),trajectory(3,2),trajectory(3,3), ' T2');
text(trajectory(4,1),trajectory(4,2),trajectory(4,3), ' T3');
text(trajectory(5,1),trajectory(5,2),trajectory(5,3), ' T4');


 xlabel('X');
 ylabel('Y');
 zlabel('Z');

view(3);

%% if condition to show data

pause(2);

prompt1 = 'Permission to show data,(Yes=1 or No=0) \nuser input : ';
a = input(prompt1);

if a==1
    showdata(matrix,q1,q2,q3,q4);
end
if a==0
    disp('\ntable will not shown');
end
```

```matlab


%% show data table contains of x, y, z, theta1_radians and theta2_radians

function showdata(m1,Q1,Q2,Q3,Q4)

close all;
col = {'x','y','z','theta1_radians','theta2_radians'};
row = {'T1','T2','T3','T4'};
datatable = [ m1(1,:) Q1(1,1:2); m1(2,:) Q2(1,1:2); m1(3,:) Q3(1,1:2); m1(4,:) Q4(1,1:2)];
figure;
DATAtable = uitable('columnname',col,'rowname',row,'position',[50 50 700 200],'columnwidth',{100},'data',datatable);

end

%% get continuous or discrete transformation matrix

function [g] = getmatrix(b)
if b==0
    g = [4 4 5;
            4 -4 5;
            -4 -4 5;
            -4 4 5;];
    return;
```

```
161.0     end
162.0      if b==1
163.0            g = [7 7 7;
164.0                7 -7 1.5;
165.0                -7 -7 -3;
166.0                -7 7 6;];
167.0            return;
168.0      end
169.0
170.0
171.0
172.0     end
173.0
174.0
175.0
176.0
```

3.8 EXPLANATION OF CODING

In this project code, from line 4.0 to 24.0 is the Link command to build a link or joint after completing our DH table and then the SerialLink is for to build the robot or to combine the link.

For transl command is where it perform the transformation matrix. To ease this project, we set our own workspace and our closed path means that make a matrix of our path like in line 155.0 to 172.0 where it will ask us to choose the continuous path or discrete path.

Then, use the ikine command which is the inverse kinematics method to get theta1 and theta2 given the x and y position. Line 41.0 to 47.0 shows how the code looks like. The qinit in line 33.0 is where I set that my individual joint angle is 0 for all.

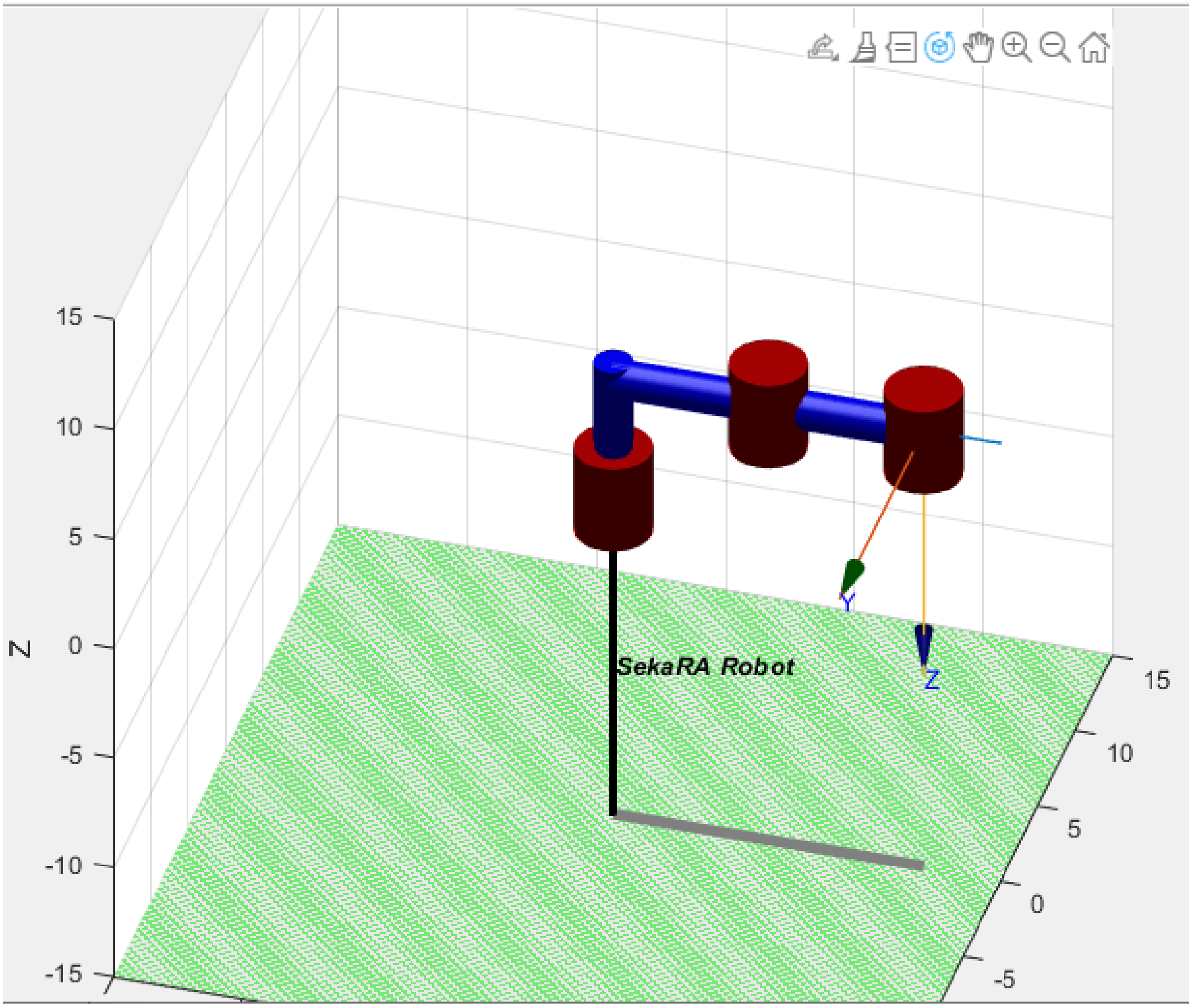


Figure 14: The each of individual joint angle of the SCARA robot is zero.

To verify or validate the data which is to know if the joint angle calculated by the ikine command wil give the same output we desired. In the code, it is from line 50.0 to 53.0 where the fkine command or forward kinematics command to get the x, y, and z position.

For animation, it can be operated by using jtraj and plot command along with time vector series. Jtraj is the junction trajectory where its parameters are the previous joint angle, next joint angle, and time series. It can be seen from 55.0 to 91.0.

Other coding line is for additional operation like showing trajectory path and table is for the ease for user.

4.0 CONCLUSION

This project objectives have been achieved where make the robot to go to the location that are set up. Furthermore, it has been proven that SCARA robot can operate better when joint angle is closer to the centre of the workspace where it easier to do the assembly and pick and place operation. It can also operate both in continuous and discrete path.