MUHAMMAD HARRIZ BIN AZHAM (1718707)

**ROBOTIC PROJECT**

1. **INTRODUCTION**

SCARA is an acronym for Selective Compliance Articulated Robot Arm. It is also compliant in the X-Y axis and rigid in the Z-axis. Generally, SCARA’s structure consists of two links joined at the base and the intersection of link one. Based on Figure 1, two independent motors are used for inverse kinematics and interpolation at joints 1 and 2 to control X-Y motion. The final location at the end of two links is a factor of angle and length from both joint 1 and joint 2. For this project, it will design a SCARA Robot with 3 Degree-of-Freedom (DoF). It is Revolute, Revolute and Prismatic joints.



Figure 1: The standard model of SCARA from Mitsubishi Electric

1. **Objective**

* To apply the idea of Denavit – Hartenberg in analysing the transformation of robot
* To understand the concept of forward and inverse kinematics of robot
* To compare the result from the simulation of Matlab and numerical calculation

1. **Concept**
   1. **Denavit – Harternberg Analysis of SCARA Robot**

D-H analysis basically refers to the four parameters to be identified from the structure of robot. The parameters are:

|  |  |
| --- | --- |
| θ | A rotation angle between two links, about the z-axis (revolute). |
| d | The distance (offset) on the z-axis, between links (prismatic). |
| a | The length of each common normal (Joint offset). |
| α | The “twist” angle between two successive z-axes (Joint twist) |

*\*Only θ and d are joint variables.*

Table 1: The parameters involved in Denavit – Harternberg Analysis

First, identify the number of joints involved in the robot and classify them as prismatic or revolute. Then, initialize the frame at each joint. Bear in mind that D-H analysis will focus on axis of x and z at each joint.

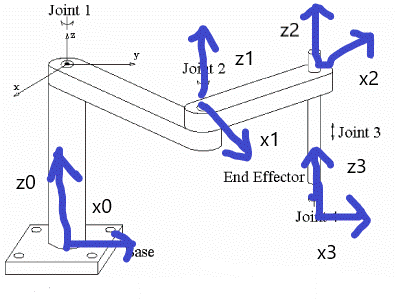


Figure 2: 3-DoF SCARA robot with the defined frames.

However, there some rules should be considered to analyse the frame and the rules are:

1. Numbering the joint from 1 to n and the frame from 0 to n-1. (n refers to the number of joints). This is important to assist in identify the four parameters.
2. For revolute joint, use right-hand rule to determine the z-axis. If the curls of the fingers are in the same direction with the rotation, then the thumb will represent as z direction
3. For prismatic, ensure that z-axis is parallel with the prismatic path taken.
4. At the first frame, x0 can be defined everywhere as long as it is perpendicular to z-axis.
5. To define the next x axis, the xn axis must intersect with the previous z axis z n-1
6. If zn-1 and zn are perpendicular, apply right hand rule to determine the direction of xn.

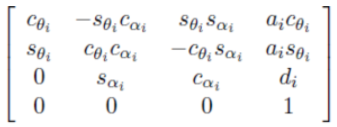
After considering theses rules, obtain the parameters.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Link (Frame) | θ | d | Α | a |
| 1 (0-1) | θ1\* | d1 | 0 | a1 |
| 2 (1-2) | θ2\* | 0 | 0 | a2 |
| 3 (2-3) | 0 | -d3 | 0 | 0 |

*‘ \* ‘ indicates the variable joint*

Table 2: D-H Table

From the table, insert the parameters into the formula for each row.

 **(1)**

A1 = **(2)**

A2 = **(3)**

A3 = **(4)**

Then, multiply the metrices of each joint to obtain final transformation matrix.

A1A2A3A4 = **(5)**

Referring to the final transformation, the first 3X3 represents the orientation of rotation and last column represents the translation or position of end effector. Remember that the last row always [0 0 0 1] because the representation of metrices should be 4X4 to ease the process of multiplication.

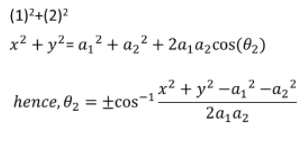
* 1. **Inverse and Forward Kinematics**

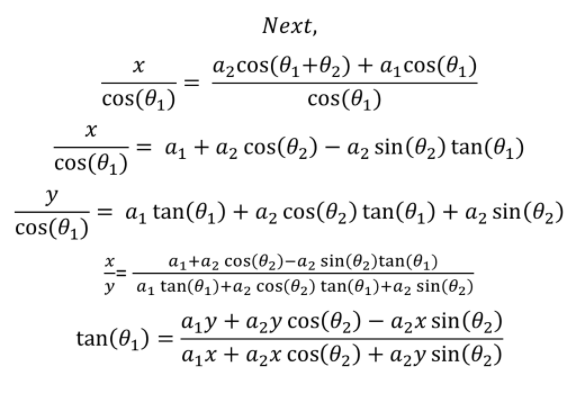
For inverse kinematics, it needs to calculate the value of each joint variable with the respect of the position of end effector. But, forward kinematics act another way as it will identify the exact position of end effector with the respect of then given value of joint variables.

For analysis of forward and inverse kinematics, just focus the last column as it represents the location with the coordinates of x, y and z.

**(6)**

For inverse kinematics, express the above equations with the respect of

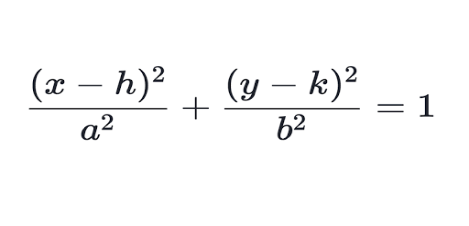
 **(7)**

 **(8)**

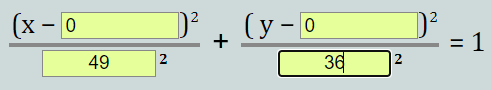
d3 = d1 – z  **(9)**

* 1. **Simulation between manual calculation and Matlab**

Design the SCARA robot with the given parameters d1=60; a1 = 40; a2 = 30. The boundary is [-100,100] for x, y and z axis. All the codes are tested on Matlab version 9. Also, design the path of robot which is ellipse for 2D and ellipsoid for 3D to show the z height varies discretely and continuously. The equation of ellipse is:

 **(10)**

Design the ellipse with h,k = 0, a= 49 and b=36.

 **(11)**

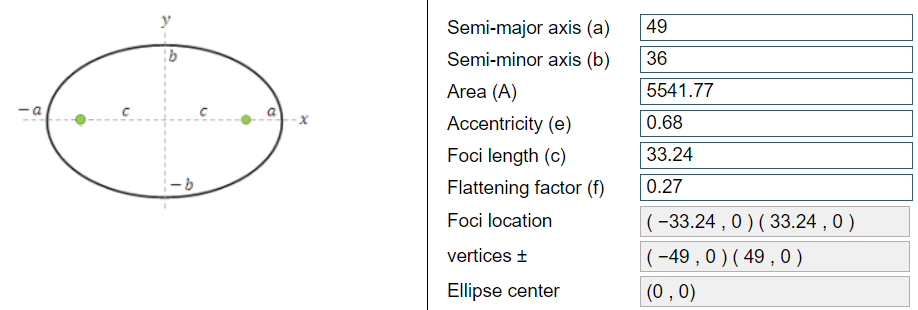


Figure 3: The information of ellipse

**(12)**

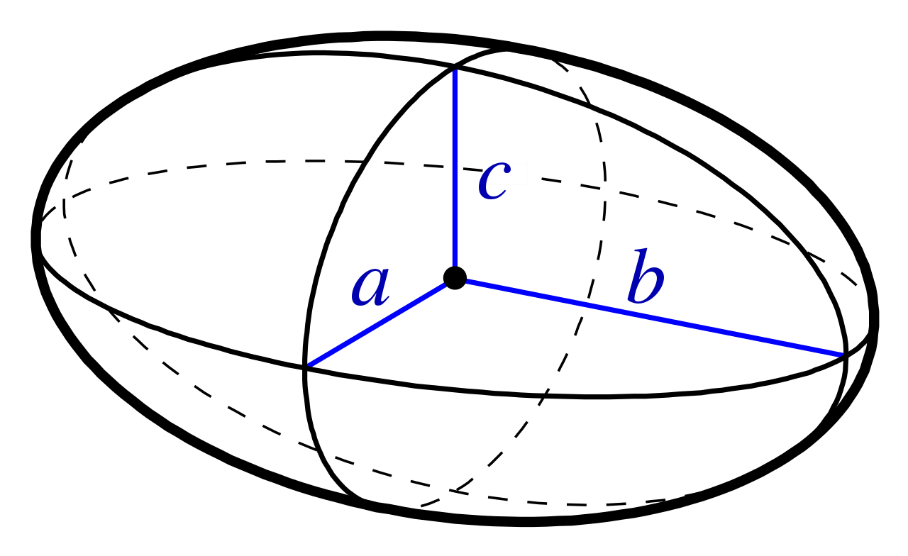


Figure 4: The ellipsoid

Thus, set the point for ellipse and ellipsoid.



Table 3: Path coordinates for Ellipse and Ellipsoid

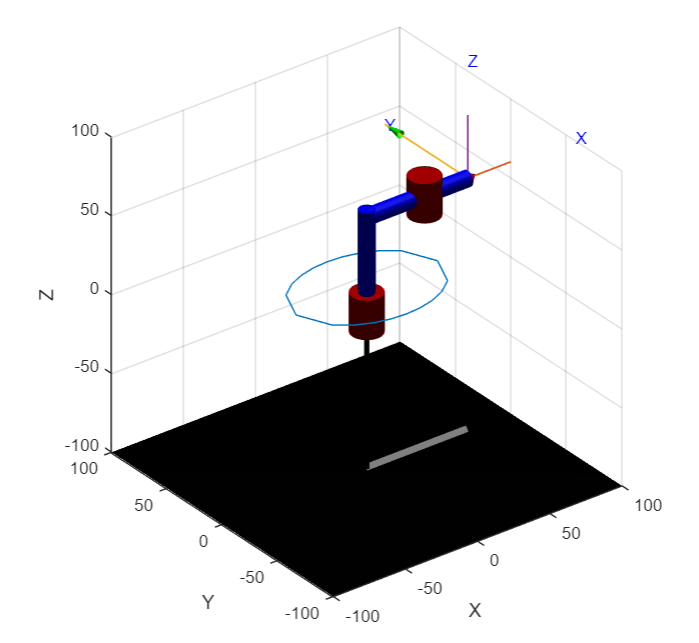


Figure 5: Ellipse path taken by SCARA robot (Z Discrete)

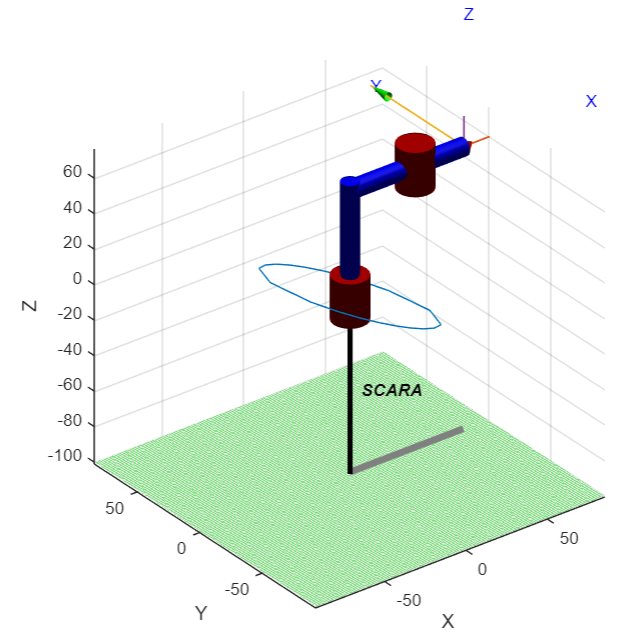


Figure 6: Ellipsoid path taken by SCARA robot (Z Continuous)

Comparison between Matlab and Excel calculation

Ellipse (Discrete)

Full Coding for Matlab:

L1 = Link ([0,d1,a1,0,0]); %revolute

L2 = Link ([0,0,a2,0,0]); %revolute

L3 = Link ([0,0,0,0,1]); %prismatic

L(3).qlim = [0,10]; %limit for prismatic path

Rob = SerialLink ([L1 L2 L3]); %integrate all links

Rob.name = 'SCARA'

input = readmatrix('Ellipsoid.csv') % read the data of x, y and z as shown in Table 3

plot3(input(:,1), input(:,2), input(:,3)) %draw the path in graph

T = transl(input) %represents the translation of end effector

qa = Rob.ikine(T,[2.48 1.61 -45],[1,1,1,0,0,0]); %inverse kinematics

T\_Temp = qa;

Q = fkine(Rob,T\_Temp); &forward kinematics

for i = 1:(length(qa)-1) % movement of robot within path

Rob.plot(qa,'workspace',[-100,100,-100,100,-100,100]);

pause(0.02)

end

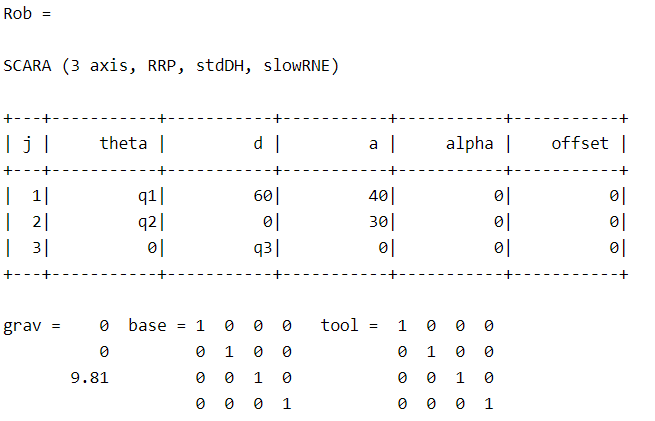


Table 4 shows the link established in SCARA robot.

Finding the inverse kinematics:

From Matlab:

qa = Rob.ikine(T,[2.48 1.61 -45],[1,1,1,0,0,0]);

* qa will carry the value of θ1, θ2 and d3 as it applies inverse kinematics
* The first column is θ1, second column is θ2 and followed by the value of d3.
* To verify the values of each joint, compare with the manual calculation using excel. Use equation (7), (8) and (9) to find θ1, θ2 and d3

|  |  |
| --- | --- |
| Matlab | Excel |
|  | |  |  |  | | --- | --- | --- | | theta1 | theta2 | d3 | | 2.483355 | 1.612058 | 45 | | 1.951058 | 1.766901 | 45 | | 1.627819 | 1.906128 | 45 | | 1.333827 | 2.009717 | 45 | | 1.04737 | 2.074292 | 45 | | 0.765719 | 2.096321 | 45 | | 0.494405 | 2.074292 | 45 | | 0.240454 | 2.009717 | 45 | | 0.004491 | 1.906128 | 45 | | -0.23176 | 1.766901 | 45 | | -0.65824 | 1.612058 | 45 | | -1.19053 | 1.766901 | 45 | | -1.51377 | 1.906128 | 45 | | -1.80777 | 2.009717 | 45 | | -2.09422 | 2.074292 | 45 | | -2.37587 | 2.096321 | 45 | | -2.64719 | 2.074292 | 45 | | -2.90114 | 2.009717 | 45 | | -3.1371 | 1.906128 | 45 | | -3.37335 | 1.766901 | 45 | | -3.79983 | 1.612058 | 45 | |
| * For d3, it carries the same value but different signs. This is because the D-H convention from Table 1, q3 is not negative. Supposedly, | |

Table 5: Comparison of values of θ1, θ2 and d3 from Matlab and Excel

Forward Kinematics

From Matlab:

T\_Temp = qa;

* qa carries the value of θ1, θ2 and d3

Q = fkine(Rob,T\_Temp);

Q

Result:  
Q(:,:,1) =  
  
 -0.5786 0.8156 0 -49.0000  
 -0.8156 -0.5786 0 0.0000  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,2) =  
  
 -0.8384 0.5450 0 -40.0000  
 -0.5450 -0.8384 0 20.7933  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,3) =  
  
 -0.9240 0.3824 0 -30.0000  
 -0.3824 -0.9240 0 28.4641  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,4) =  
  
 -0.9797 0.2006 0 -20.0000  
 -0.2006 -0.9797 0 32.8647  
 0 0 1.0000 15.0000  
 0 0 0 1.0000

Q(:,:,5) =  
  
 -0.9998 -0.0199 0 -10.0000  
 0.0199 -0.9998 0 35.2423  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,6) =  
  
 -0.9612 -0.2759 0 -0.0000  
 0.2759 -0.9612 0 36.0000  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,7) =  
  
 -0.8403 -0.5421 0 10.0000  
 0.5421 -0.8403 0 35.2423  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,8) =  
  
 -0.6283 -0.7780 0 20.0000  
 0.7780 -0.6283 0 32.8647  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,9) =  
  
 -0.3333 -0.9428 0 30.0000  
 0.9428 -0.3333 0 28.4641  
 0 0 1.0000 15.0000  
 0 0 0 1.0000

Q(:,:,10) =  
  
 0.0356 -0.9994 0 40.0000  
 0.9994 0.0356 0 20.7933  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,11) =  
  
 0.5786 -0.8156 0 49.0000  
 0.8156 0.5786 0 -0.0000  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,12) =  
  
 0.8384 -0.5450 0 40.0000  
 0.5450 0.8384 0 -20.7933  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,13) =  
  
 0.9240 -0.3824 0 30.0000  
 0.3824 0.9240 0 -28.4641  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,14) =  
  
 0.9797 -0.2006 0 20.0000  
 0.2006 0.9797 0 -32.8647  
 0 0 1.0000 15.0000  
 0 0 0 1.0000

Q(:,:,15) =  
  
 0.9998 0.0199 0 10.0000  
 -0.0199 0.9998 0 -35.2423  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,16) =  
  
 0.9612 0.2759 0 0.0000  
 -0.2759 0.9612 0 -36.0000  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,17) =  
  
 0.8403 0.5421 0 -10.0000  
 -0.5421 0.8403 0 -35.2423  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,18) =  
  
 0.6283 0.7780 0 -20.0000  
 -0.7780 0.6283 0 -32.8647  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,19) =  
  
 0.3333 0.9428 0 -30.0000  
 -0.9428 0.3333 0 -28.4641  
 0 0 1.0000 15.0000  
 0 0 0 1.0000

Q(:,:,20) =  
  
 -0.0356 0.9994 0 -40.0000  
 -0.9994 -0.0356 0 -20.7933  
 0 0 1.0000 15.0000  
 0 0 0 1.0000  
  
  
Q(:,:,21) =  
  
 -0.5786 0.8156 0 -49.0000  
 -0.8156 -0.5786 0 0.0000  
 0 0 1.0000 15.0000  
 0 0 0 1.0000

From Q1 to Q21, it represents the translation of end effector with the assigned value of θ1, θ2 and d3

|  |  |  |
| --- | --- | --- |
| theta1 | theta2 | d3 |
| 2.483355 | 1.612058 | 45 |
| 1.951058 | 1.766901 | 45 |
| 1.627819 | 1.906128 | 45 |
| 1.333827 | 2.009717 | 45 |
| 1.04737 | 2.074292 | 45 |
| 0.765719 | 2.096321 | 45 |
| 0.494405 | 2.074292 | 45 |
| 0.240454 | 2.009717 | 45 |
| 0.004491 | 1.906128 | 45 |
| -0.23176 | 1.766901 | 45 |
| -0.65824 | 1.612058 | 45 |
| -1.19053 | 1.766901 | 45 |
| -1.51377 | 1.906128 | 45 |
| -1.80777 | 2.009717 | 45 |
| -2.09422 | 2.074292 | 45 |
| -2.37587 | 2.096321 | 45 |
| -2.64719 | 2.074292 | 45 |
| -2.90114 | 2.009717 | 45 |
| -3.1371 | 1.906128 | 45 |
| -3.37335 | 1.766901 | 45 |
| -3.79983 | 1.612058 | 45 |

From excel, use equation (6) to find x, y and z.

|  |  |  |
| --- | --- | --- |
| X.forward | Y.forward | Z.forward |
| -49 | 0 | 15 |
| -40 | 20.793265 | 15 |
| -30 | 28.464055 | 15 |
| -20 | 32.864722 | 15 |
| -10 | 35.242339 | 15 |
| 0 | 36 | 15 |
| 10 | 35.242339 | 15 |
| 20 | 32.864722 | 15 |
| 30 | 28.464055 | 15 |
| 40 | 20.793265 | 15 |
| 49 | 0 | 15 |
| 40 | -20.79326 | 15 |
| 30 | -28.46405 | 15 |
| 20 | -32.86472 | 15 |
| 10 | -35.24234 | 15 |
| 0 | -36 | 15 |
| -10 | -35.24234 | 15 |
| -20 | -32.86472 | 15 |
| -30 | -28.46405 | 15 |
| -40 | -20.79326 | 15 |
| -49 | 0 | 15 |

Table 6: Forward Kinematic through Excel calculations

From above table, each row can be compared to Q1 – Q21 to validate the result of forward kinematics from Matlab and Excel calculation.

Ellipsoid (Continuous)

Full Coding for Matlab:

d1=60; a1 = 40; a2 = 30;

L1 = Link ([0,d1,a1,0,0]); %revolute

L2 = Link ([0,0,a2,0,0]); %revolute

L3 = Link ([0,0,0,0,1]); %prismatic

L(3).qlim = [0,10]; %limit path for prismatic

Rob = SerialLink ([L1 L2 L3]); %integrate all links

Rob.name = 'SCARA'

input = readmatrix('Ellipsoid.csv') %read the value of x, y and z as shown in Table 3

T = transl(input) %represents the location of end effector

qa = Rob.ikine(T,[2.48 1.61 35],[1,1,1,0,0,0]); %inverse kinematics

qa = Rob.ikine(T,[2.48 1.61 -45],[1,1,1,0,0,0]); %inverse kinematics

T\_Temp = qa;

for i = 1:(length(qa)-1) %movement of robot

Rob.plot(qa,'workspace',[-100,100,-100,100,-100,100]);

pause(0.02)

end

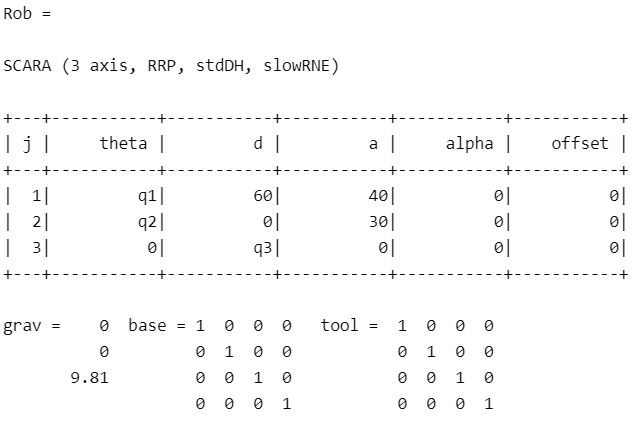


Table 7 shows the links established in the SCARA robot.

Finding the inverse kinematics:

In Matlab

qa = Rob.ikine(T,[2.48 1.61 35],[1,1,1,0,0,0]);

* qa will carry the value of θ1, θ2 and d3 as it applies inverse kinematics
* The first column is θ1, second column is θ2 and followed by the value of d3.
* To verify the values of each joint, compare with the manual calculation using excel. Use equation (7), (8) and (9) to find θ1, θ2 and d3

|  |  |
| --- | --- |
| Matlab | Excel |
|  | |  |  |  | | --- | --- | --- | | Theta1 | Theta2 | d3 | | 2.483355 | 1.612058 | 35 | | 1.951058 | 1.766901 | 39.59184 | | 1.627819 | 1.906128 | 44.69388 | | 1.333827 | 2.009717 | 49.79592 | | 1.04737 | 2.074292 | 54.89796 | | 0.765719 | 2.096321 | 60 | | 0.494405 | 2.074292 | 65.10204 | | 0.240454 | 2.009717 | 70.20408 | | 0.004491 | 1.906128 | 75.30612 | | -0.23176 | 1.766901 | 80.40816 | | -0.65824 | 1.612058 | 85 | | -1.19053 | 1.766901 | 80.40816 | | -1.51377 | 1.906128 | 75.30612 | | -1.80777 | 2.009717 | 70.20408 | | -2.09422 | 2.074292 | 65.10204 | | -2.37587 | 2.096321 | 60 | | -2.64719 | 2.074292 | 54.89796 | | -2.90114 | 2.009717 | 49.79592 | | -3.1371 | 1.906128 | 44.69388 | | -3.37335 | 1.766901 | 39.59184 | | -3.79983 | 1.612058 | 35 | |

Table 8: Comparison of values of θ1, θ2 and d3 from Matlab and Excel

Calculate forward kinematics

In Matlab

T\_Temp = qa;

* qa carries the value of θ1, θ2 and d3

Q = fkine(Rob,T\_Temp);

Q(:,:,1) =  
  
 -0.5786 0.8156 0 -49.0000  
 -0.8156 -0.5786 0 0.0000  
 0 0 1.0000 25.0000  
 0 0 0 1.0000  
  
  
Q(:,:,2) =  
  
 -0.8384 0.5450 0 -40.0000  
 -0.5450 -0.8384 0 20.7933  
 0 0 1.0000 20.4082  
 0 0 0 1.0000  
  
  
Q(:,:,3) =  
  
 -0.9240 0.3824 0 -30.0000  
 -0.3824 -0.9240 0 28.4641  
 0 0 1.0000 15.3061  
 0 0 0 1.0000  
  
  
Q(:,:,4) =  
  
 -0.9797 0.2006 0 -20.0000  
 -0.2006 -0.9797 0 32.8647  
 0 0 1.0000 10.2041  
 0 0 0 1.0000

Q(:,:,5) =  
  
 -0.9998 -0.0199 0 -10.0000  
 0.0199 -0.9998 0 35.2423  
 0 0 1.0000 5.1020  
 0 0 0 1.0000  
  
Q(:,:,6) =  
  
 -0.9612 -0.2759 0 -0.0000  
 0.2759 -0.9612 0 36.0000  
 0 0 1.0000 0.0000  
 0 0 0 1.0000  
  
  
Q(:,:,7) =  
  
 -0.8403 -0.5421 0 10.0000  
 0.5421 -0.8403 0 35.2423  
 0 0 1.0000 -5.1020  
 0 0 0 1.0000  
  
  
Q(:,:,8) =  
  
 -0.6283 -0.7780 0 20.0000  
 0.7780 -0.6283 0 32.8647  
 0 0 1.0000 -10.2041  
 0 0 0 1.0000  
  
  
Q(:,:,9) =  
  
 -0.3333 -0.9428 0 30.0000  
 0.9428 -0.3333 0 28.4641  
 0 0 1.0000 -15.3061  
 0 0 0 1.0000  
  
Q(:,:,10) =  
  
 0.0356 -0.9994 0 40.0000  
 0.9994 0.0356 0 20.7933  
 0 0 1.0000 -20.4082  
 0 0 0 1.0000  
  
Q(:,:,11) =  
  
 0.5786 -0.8156 0 49.0000  
 0.8156 0.5786 0 -0.0000  
 0 0 1.0000 -25.0000  
 0 0 0 1.0000  
  
  
Q(:,:,12) =  
  
 0.8384 -0.5450 0 40.0000  
 0.5450 0.8384 0 -20.7933  
 0 0 1.0000 -20.4082  
 0 0 0 1.0000  
  
  
Q(:,:,13) =  
  
 0.9240 -0.3824 0 30.0000  
 0.3824 0.9240 0 -28.4641  
 0 0 1.0000 -15.3061  
 0 0 0 1.0000  
  
  
Q(:,:,14) =  
  
 0.9797 -0.2006 0 20.0000  
 0.2006 0.9797 0 -32.8647  
 0 0 1.0000 -10.2041  
 0 0 0 1.0000  
  
  
Q(:,:,15) =  
  
 0.9998 0.0199 0 10.0000  
 -0.0199 0.9998 0 -35.2423  
 0 0 1.0000 -5.1020  
 0 0 0 1.0000

Q(:,:,16) =  
  
 0.9612 0.2759 0 0.0000  
 -0.2759 0.9612 0 -36.0000  
 0 0 1.0000 -0.0000  
 0 0 0 1.0000  
  
  
Q(:,:,17) =  
  
 0.8403 0.5421 0 -10.0000  
 -0.5421 0.8403 0 -35.2423  
 0 0 1.0000 5.1020  
 0 0 0 1.0000  
  
  
Q(:,:,18) =  
  
 0.6283 0.7780 0 -20.0000  
 -0.7780 0.6283 0 -32.8647  
 0 0 1.0000 10.2041  
 0 0 0 1.0000  
  
  
Q(:,:,19) =  
  
 0.3333 0.9428 0 -30.0000  
 -0.9428 0.3333 0 -28.4641  
 0 0 1.0000 15.3061  
 0 0 0 1.0000  
  
  
Q(:,:,20) =  
  
 -0.0356 0.9994 0 -40.0000  
 -0.9994 -0.0356 0 -20.7933  
 0 0 1.0000 20.4082  
 0 0 0 1.0000

Q(:,:,21) =  
  
 -0.5786 0.8156 0 -49.0000  
 -0.8156 -0.5786 0 0.0000  
 0 0 1.0000 25.0000  
 0 0 0 1.0000

From Q1 to Q21, it represents the translation of end effector with the assigned value of θ1, θ2 and d3. From excel, use equation (6) to find x, y and z.

|  |  |  |
| --- | --- | --- |
| x | y | z |
| -49 | 0 | 25 |
| -40 | 20.79326 | 20.40816 |
| -30 | 28.46405 | 15.30612 |
| -20 | 32.86472 | 10.20408 |
| -10 | 35.24234 | 5.102041 |
| 0 | 36 | 0 |
| 10 | 35.24234 | -5.10204 |
| 20 | 32.86472 | -10.2041 |
| 30 | 28.46405 | -15.3061 |
| 40 | 20.79326 | -20.4082 |
| 49 | 0 | -25 |
| 40 | -20.7933 | -20.4082 |
| 30 | -28.4641 | -15.3061 |
| 20 | -32.8647 | -10.2041 |
| 10 | -35.2423 | -5.10204 |
| 0 | -36 | 0 |
| -10 | -35.2423 | 5.102041 |
| -20 | -32.8647 | 10.20408 |
| -30 | -28.4641 | 15.30612 |
| -40 | -20.7933 | 20.40816 |
| -49 | 0 | 25 |

Table 9: Forward kinematics from Excel calculations

From above table, each row can be compared to Q1 – Q21 to validate the result of foraward kinematics from Matlab and Excel calculation.