MTRN4230 Assignment 3

Runkai Zhao, z5146927

1.a A vector pA is rotated about ZA axis by θ degrees and then rotated about XA axis by ϕ degrees. Give the rotation matrix considering the orders given.

Answer:

First the vector is rotated about z-axis by θ degrees, the rotational matrix from frame A to frame B is,

$$R_B^A = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Then, the vector is rotated about x-axis (rotation about fixed axes in frame A) by ϕ degrees, the rotational matrix from fame B to frame C is,

$$R_C^B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix}$$

Hence, the rotation matrix from frame A to frame B is,

$$R_C^A = R_C^B \times R_B^A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & -sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix} \times \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

1.b. Frame {B} initially coincident with frame {A}. Now rotate {B} about ZB axis by θ degrees and rotate the resulting frame about XB *axis by* ϕ degrees. Find rotation matrix for vectors pB *to* pA .

Answer:

Rotated about z-axis by θ degrees, the rotational matrix from frame A to frame C is,

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

Rotated about current x-axis by ϕ degrees, the rotational matrix from frame C to frame B is,

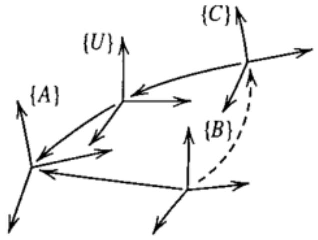
$$R_B^C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

Hence, the rotation matrix from frame A to frame B is,

$$R_B^A = R_C^A \times R_B^C = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

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

1.c. Given below frames



Calculate ${}^{B}_{C}T$ when ${}^{U}_{A}T$, ${}^{B}_{A}T$ and ${}^{C}_{U}T$ are given.

Answer:

Define some vectors in frame A, B and C, so we can have the following equations,

$$P_A = ({}^{B}_{A}T)^{-1} \times P_B$$

$$P_A = \left({}^{U}_{A}T\right)^{-1} \times \left({}^{C}_{U}T\right)^{-1} \times P_C$$

Equate the above equations,

$$({}^{B}_{A}T)^{-1} \times P_B = \left({}^{U}_{A}T\right)^{-1} \times \left({}^{C}_{U}T\right)^{-1} \times P_C$$

Due to,

$$P_B = {}^{B}_{C}T \times P_C$$

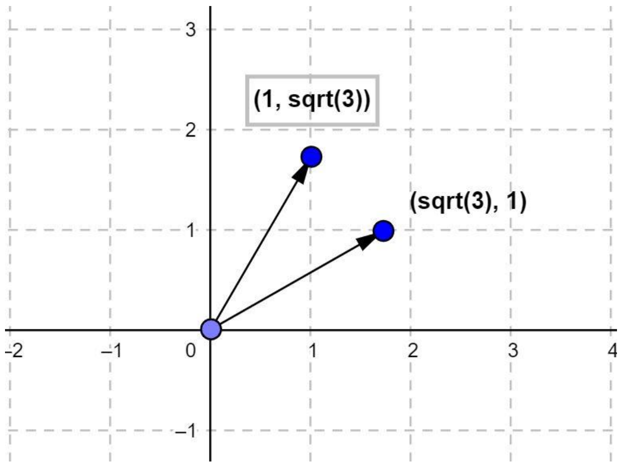
Inserting this into the general equation and rearranging,

$${}^{B}_{C}T = \left({}^{U}_{A}T\right)^{-1} \times \left({}^{C}_{U}T\right)^{-1} \times \ {}^{B}_{A}T$$

1.d. Proof that inverse of a rotation matrix must be equal to its transpose and rotation matrix is orthonormal. Show it with the help of two vectors embedded in a rigid body so no matter how the body rotates, the geometric angel between them (two vectors) preserve.

Answer:

First, there are two vectors embedded in a same coordinate frame or rigid body as shown below,



Based on this example, the rotation matrix can be found as,

$$R = \begin{bmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

The matrix can be inverted as,

$$R^{-1} = \frac{1}{\frac{\sqrt{3}}{2} \times \frac{\sqrt{3}}{2} - \frac{1}{2} \times \left(-\frac{1}{2}\right)} \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

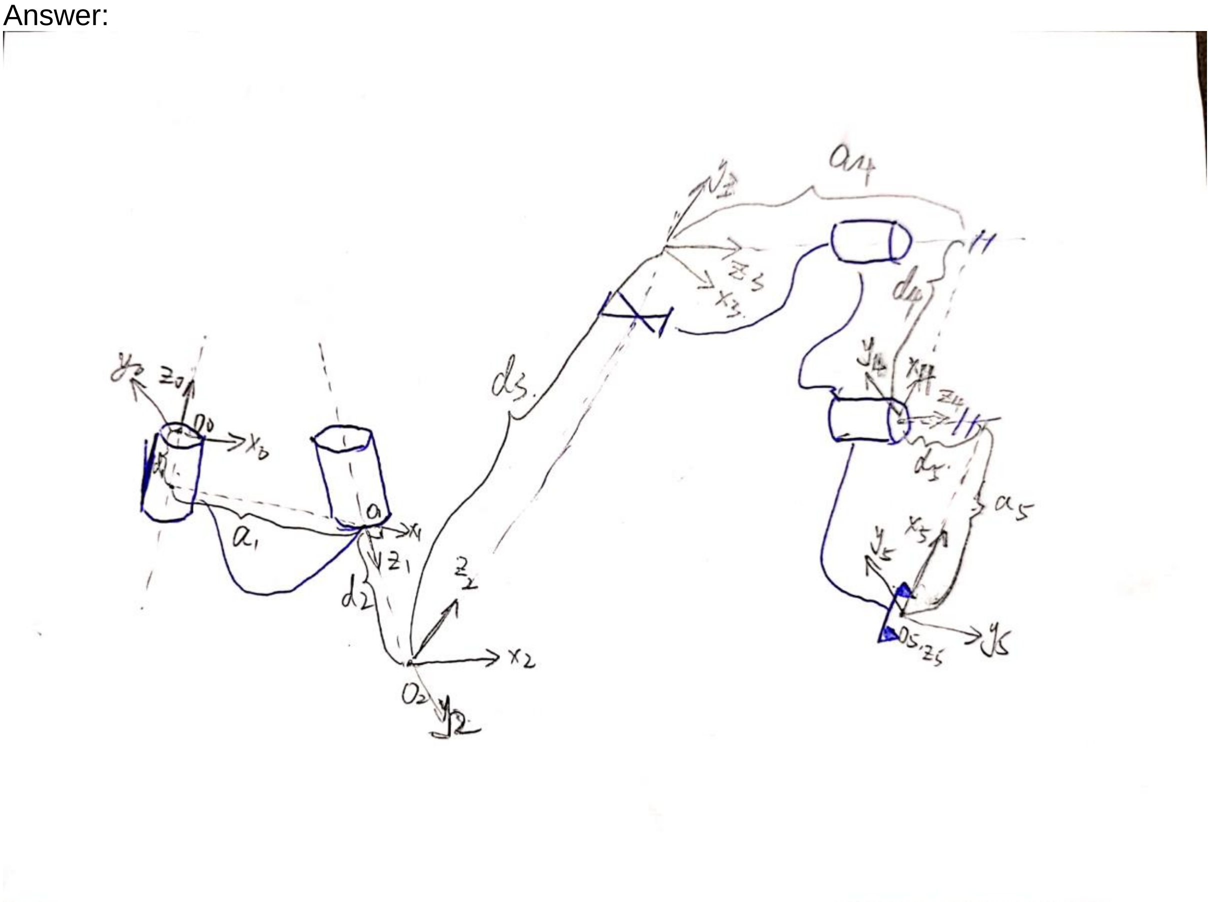
The transpose of the rotation matrix is,

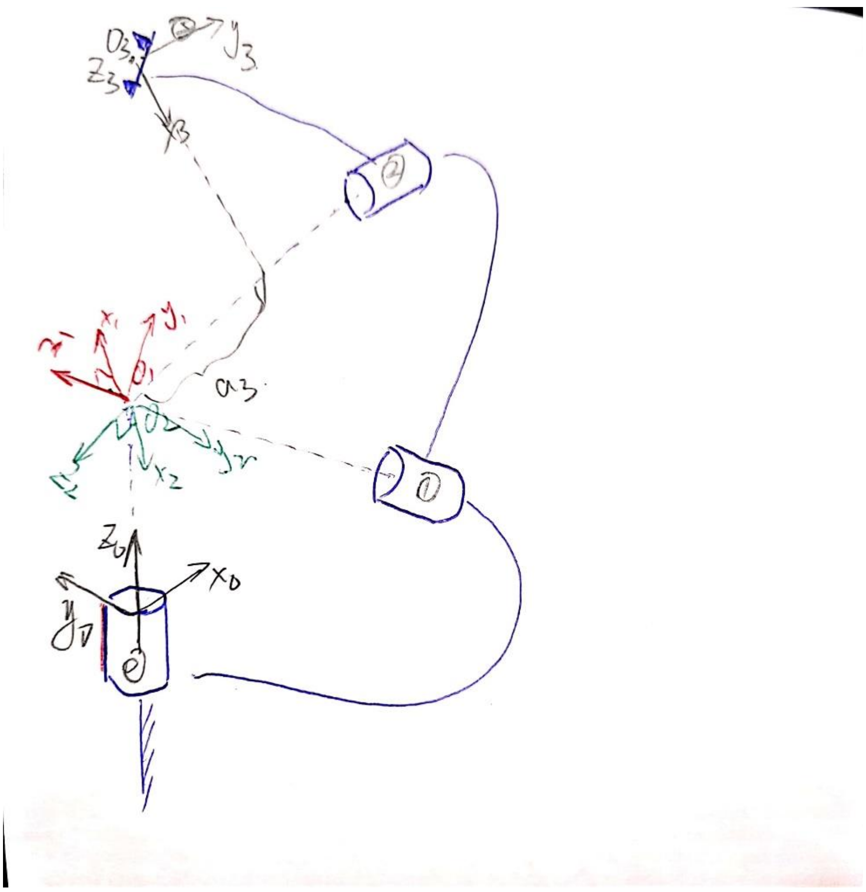
$$R^T = \begin{bmatrix} \frac{\sqrt{3}}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}$$

Obviously, the inverse of the rotation matrix is equal to the transpose of it. Hence, it can be deduced that the rotation matrix is orthonormal by the mathematical rules.

1.e. Show the link frames for the below manipulators schematically

Answer:





1.f. A 2DOF positioning table is used to help welding (two rotary joints $\theta 1$, $\theta 2$). The forward kinematics from based (link 1) to the bed of the table (link 2) is

$$
{}^{0}_{2}T = \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 & l_2s_1 + l_1 \\ s_2 & c_2 & 0 & 0 \\ -s_1c_2 & s_1s_2 & c_1 & l_2c_1 + h_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

Unit vector fixed in frame of link 2 is 2V. $Find\ inverse - kinematic\ solution\ for\ \theta 1, \theta 2)$ **when this unit vector is aligned with** $Z\ 0axis$**. Are there multiple solutions and is there a singular condition?**

Answer:
When the unit vector is aligned with z-axis in frame 0, so we can have the equations as,

$$
\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 \\ s_2 & c_2 & 0 \\ -s_1c_2 & s_1s_2 & c_1 \end{bmatrix} \begin{bmatrix} X^2 \\ Y^2 \\ Z^2 \end{bmatrix}
$$

Then, we can derive the three sub-equations,

$$c_1c_2X^2 - c_1s_2Y^2 + s_1Z^2 = 0\ (1)$$

$$s_2X^2 + c_2Y^2 = 0\ (2)$$

$$-s_1c_2X^2 + s_1s_2Y^2 + c_1Z^2 = 1\ (3)$$

From equation 2, we can get the θ_2 value,

$$\theta_2 = \tan^{-1}(\frac{-Y}{X})$$

Inserting this value into equation 1 and equation 2, we can obtain,

$$\theta_1 = \tan^{-1}(\frac{-\sqrt{X^2+Y^2}}{Z})$$

If both X and Y are equal to zero, then singular and θ_2 is arbitrary.

2. A manipulator shown below that is known as SCARA when d4= 0.1, a1 = 0.4 and a2 = 0.3.

2.a Use DH convention find the forward kinematics.

Answer:

Link	Angle (θ_i)	Offset (d_i)	length (a_i)	Twist (α_i)
1	θ_1	0	a_1	0
2	θ_2	0	a_2	180
3	0	d_3	0	0
4	θ_4	d_4	0	0

By referring the lecture slides, the homogeneous matrices can be generated,

$$T_1^0 = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1 cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1 sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} cos(\theta_2) & sin(\theta_2) & 0 & a_2 cos(\theta_2) \\ sin(\theta_2) & -cos(\theta_2) & 0 & a_2 sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} cos(\theta_4) & -sin(\theta_4) & 0 & 0 \\ sin(\theta_4) & cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Form a general matrix with respect to the base frame,

$$T_4^0 = T_1^0 \times T_2^1 \times T_3^2 \times T_4^3 = \begin{bmatrix} c_{12}c_4 + s_{12}s_4 & -c_{12}s_4 + s_{12}c_4 & 0 & a_2c_{12} + a_1c_1 \\ s_{12}c_4 - c_{12}s_4 & -s_{12}s_4 - c_{12}c_4 & 0 & a_2s_{12} + a_1c_1 \\ 0 & 0 & -1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Where $c_{ij} = \cos(\theta_i + \theta_j)$, $s_{ij} = \sin(\theta_i + \theta_j)$, $c_i = \cos(\theta_i)$ and $s_i = \sin(\theta_i)$

Then, the MATLAB codes and results are presented as following,

```
syms theta1 alpha1 a1 d1
T_0_1 = [cos(theta1), -sin(theta1), 0, a1*cos(theta1);
         sin(theta1), cos(theta1), 0, a1*sin(theta1);
         0, 0, 1, 0;
         0, 0, 0, 1]

syms theta2 alpha2 a2 d2
T_1_2 = [cos(theta2), sin(theta2), 0, a2*cos(theta2);
         sin(theta2), -cos(theta2), 0, a2*sin(theta2);
         0, 0, -1, 0;
         0, 0, 0, 1]


syms theta3 alpha3 a3 d3
T_2_3 = [1, 0, 0, 0;
         0, 1, 0, 0;
         0, 0, 1, d3;
         0, 0, 0, 1]

syms theta4 alpha4 a4 d4
T_3_4 = [cos(theta4), -sin(theta4), 0, 0;
         sin(theta4), cos(theta4), 0, 0;
         0, 0, 1, d4;
         0, 0, 0, 1]

T_0_4 = T_0_1*T_1_2*T_2_3*T_3_4;
simplify(T_0_4)
```

T_0_1 =

$$\begin{pmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T_1_2 =

$$\begin{pmatrix} \cos(\theta_2) & \sin(\theta_2) & 0 & a_2\cos(\theta_2) \\ \sin(\theta_2) & -\cos(\theta_2) & 0 & a_2\sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T_2_3 =

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T_3_4 =

$$\begin{pmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

T_0_4 =

$$\begin{pmatrix} \cos(\theta_1+\theta_2-\theta_4) & \sin(\theta_1+\theta_2-\theta_4) & 0 & a_2\cos(\theta_1+\theta_2)+a_1\cos(\theta_1) \\ \sin(\theta_1+\theta_2-\theta_4) & -\cos(\theta_1+\theta_2-\theta_4) & 0 & a_2\sin(\theta_1+\theta_2)+a_1\sin(\theta_1) \\ 0 & 0 & -1 & -d_3-d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

where d4= 0.1, a1 = 0.4 and a2 = 0.3.

2.b Find inverse kinematics.

Answer: The corresponding homogeneous transform take the form as,

$$T_w^b = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & x \\ \sin(\beta) & \cos(\beta) & 0 & y \\ 0 & 0 & -1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Compared T_4^0 with T_w^b, we can see

$$\begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 & x \\ \sin(\beta) & \cos(\beta) & 0 & y \\ 0 & 0 & -1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_{12}c_4 + s_{12}s_4 & -c_{12}s_4 + s_{12}c_4 & 0 & a_2c_{12} + a_1c_1 \\ s_{12}c_4 - c_{12}s_4 & -s_{12}s_4 - c_{12}c_4 & 0 & a_2s_{12} + a_1c_1 \\ 0 & 0 & -1 & -d_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

So,

$$x^2 + y^2 = (a_1c_1 + a_2c_{12})^2 + (a_1c_1 + a_2s_{12})^2$$

Simplifying the above equation in,

$$cos\theta_2 = \frac{X^2 + Y^2 - a_2^2 - a_1^2}{2a_1a_2}$$

Base the triangle theorems, we can get,

$$\theta_2 = \cos^{-1}(\frac{X^2 + Y^2 - a_2^2 - a_1^2}{2a_1a_2})$$

Then, θ_1 can be represented as,

$$\theta_1 = \tan^{-1}\left(\frac{Y}{X}\right) - \tan^{-1}\left(\frac{a_2sin\theta_2}{a_1 + a_2cos\theta_2}\right)$$

Since,

$$\theta_1 + \theta_2 - \theta_4 = \beta$$
$$\theta_4 = \theta_1 + \theta_2 - \beta$$

And,

$$d_3 = d_4 - Z$$

In summary, the inverse kinematics can be expressed as

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_4 \\ d_3 \end{bmatrix} = \begin{bmatrix} tan^{-1}\left(\frac{Y}{X}\right) - tan^{-1}\left(\frac{a_2sin\theta_2}{a_1 + a_2cos\theta_2}\right) \\ cos^{-1}\left(\frac{X^2 + Y^2 - a_2^2 - a_1^2}{2a_1a_2}\right) \\ \theta_1 + \theta_2 - \beta \\ -d_4 - Z \end{bmatrix}$$

where d4= 0.1, a1 = 0.4 and a2 = 0.3.

The MATLAB codes and results are presented as below,

```
syms x y z beta k1 k2
sin_theta2 = (x^2+y^2-a2^2-a1^2)/(2*a1*a2)
cos_tehta2 = sqrt(1-sin_theta2^2);
theta2 = atan(sin_theta2/cos_tehta2);
k1 = a1 + a2*cos(theta2);
k2 = a2*sin(theta2);
theta1 = atan(y/x) - atan(k2/k1)
theta4 = theta1 + theta2 - beta
d3 = -z - d4
inverseKResult = [theta1;theta2;theta4;d3]
```

```
inverseKResult =
```

$$\begin{pmatrix} \operatorname{atan}\left(\frac{y}{x}\right) + \sigma_2 \\ -\sigma_1 \\ \operatorname{atan}\left(\frac{y}{x}\right) - \beta + \sigma_2 - \sigma_1 \\ -d_4 - z \end{pmatrix}$$

where

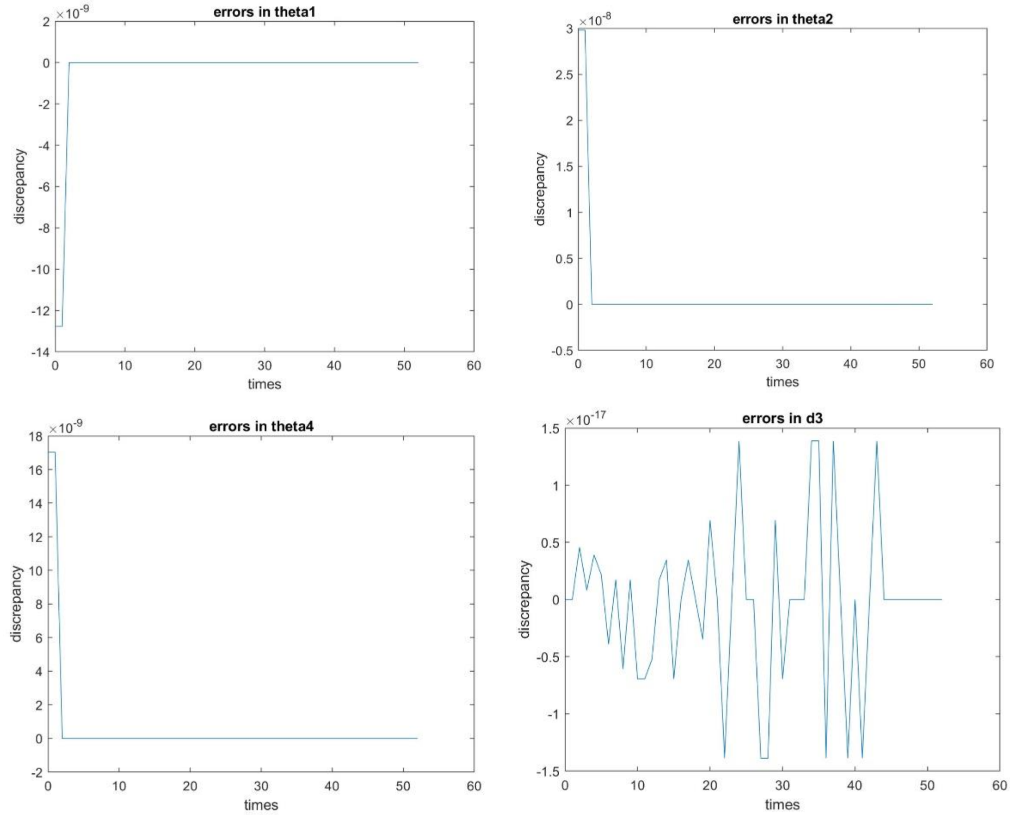
$$\sigma_1 = \operatorname{atan}\left(\frac{{a_1}^2 + {a_2}^2 - x^2 - y^2}{2\, a_1\, a_2\, \sqrt{1 - \frac{\left({a_1}^2 + {a_2}^2 - x^2 - y^2\right)^2}{4\, {a_1}^2\, {a_2}^2}}}\right)$$

$$\sigma_2 = \operatorname{atan}\left(\frac{{a_1}^2 + {a_2}^2 - x^2 - y^2}{2\, a_1\, \left(a_1 + \frac{a_2}{\sigma_3}\right)\, \sigma_3\, \sqrt{1 - \frac{\left({a_1}^2 + {a_2}^2 - x^2 - y^2\right)^2}{4\, {a_1}^2\, {a_2}^2}}}\right)$$

$$\sigma_3 = \sqrt{1 - \frac{\left({a_1}^2 + {a_2}^2 - x^2 - y^2\right)^2}{4\, {a_1}^2\, {a_2}^2\, \left(\frac{\left({a_1}^2 + {a_2}^2 - x^2 - y^2\right)^2}{4\, {a_1}^2\, {a_2}^2} - 1\right)}}$$

2.c Create kinematics verification mechanism as below and show the error as defined below in SIMULINK (use the trajectory provided).

Answer: The verification mechanism is to consider the provided trajectory data as inputs, then insert these inputs into forward kinematic model. After getting the forward kinematic matrix, implementing the inverse kinematic model. Compared the final outputs of $\theta_1, \theta_2, \theta_4, d_3$ with the given trajectory, the errors are calculated by (outputs - inputs). The error plots are shown as below.



As the figures shown, all the error values are much closer to zero. So, the forward and inverse kinematic matrix are correct and reliable.

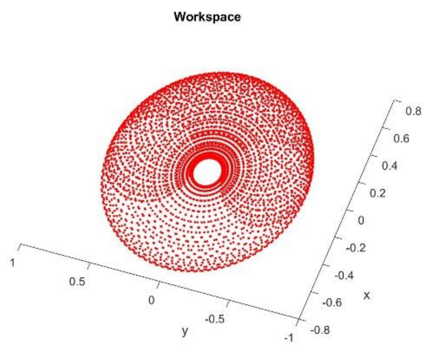
The MATLAB codes are presented as below,

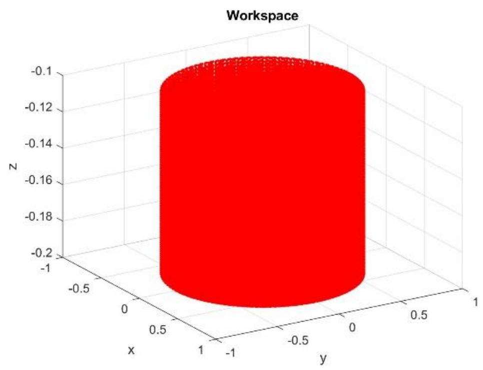
```
originalM = zeros(4,53);
inverseKResult = zeros(4,53);
T = zeros(4,4);
temp1 = zeros(1,53);
temp2 = zeros(1,53);
i = 1;
while i < 53
theta1 = qd(i,1);
theta2 = qd(i,2);
theta4 = qd(i,4);
d3 = qd(i,3);
d4 = 0.1;
a1 = 0.4;
a2 = 0.3;
T = [ cos(theta1 + theta2 - theta4),  sin(theta1 + theta2 - theta4),  0, a2*cos(theta1 + theta2) + a1*cos(theta1);
      sin(theta1 + theta2 - theta4), -cos(theta1 + theta2 - theta4), 0, a2*sin(theta1 + theta2) + a1*sin(theta1);
      0, 0, -1,-d3-d4;
      0, 0,  0, 1];

originalM(1,i) = theta1;
originalM(2,i) = theta2;
originalM(3,i) = theta4;
originalM(4,i) = d3;

x = T(1,4); y = T(2,4); z = T(3,4); beta = theta1 + theta2 - theta4;
T(1,1);
cos_theta2 = (x^2+y^2-a2^2-a1^2)/(2*a1*a2);
sin_theta2 = sqrt(1-cos_theta2^2);
theta2 = atan(sin_theta2/cos_theta2);
k1 = a1 + a2*cos(theta2);
k2 = a2*sin(theta2);

temp1(1,i) = y;
temp2(1,i) = beta;
if y/x < 0
    theta1 = pi + atan(y/x) - atan(k2/k1);
else
    theta1 = atan(y/x) - atan(k2/k1);
end
```

```
theta4 = theta1 + theta2 - beta ;
d3 = -z - 0.1;
inverseKResult(1,i) = theta1;
inverseKResult(2,i) = theta2;
inverseKResult(3,i) = theta4;
inverseKResult(4,i) = d3;

i = i+1;
end

error = inverseKResult - originalM;
figure(1);
x = 0:1:52;
plot(x,error(1,:))
title("errors in theta1");
xlabel("times");
ylabel("discrepancy");

figure(2);
plot(x,error(2,:))
title("errors in theta2");
xlabel("times");
ylabel("discrepancy");

figure(3);
plot(x,error(3,:))
title("errors in theta4");
xlabel("times");
ylabel("discrepancy");

figure(4);
plot(x,error(4,:))
title("errors in d3");
xlabel("times");
ylabel("discrepancy");
```

2.d Plot workspace when $-180^o \le \theta_1, \theta_2 \le 180^o$ and $0 \le d_3 \le 0.1$ in MATLAB.

The workspace is a cylinder shape as shown below,





2.e Calculate Jacobian and provide a function in MATLAB to calculate it.

The calculation process of Jacobian matrix is shown as below,

```
zero = zeros(3,1);
syms theta1 alpha1 a1 d1 theta2 alpha2 a2 d2 theta3 alpha3 a3 d3 theta4 alpha4 a4 d4
T = [ cos(theta1 + theta2 - theta4),  sin(theta1 + theta2 - theta4),  0, a2*cos(theta1 + theta2) + a1*cos(theta1);
     sin(theta1 + theta2 - theta4), -cos(theta1 + theta2 - theta4), 0, a2*sin(theta1 + theta2) + a1*sin(theta1);
     0, 0, -1,-d3-d4;
     0, 0,  0, 1];
Jacob1 = [cross(T(1:3,3), T(1:3,4));T(1:3,3)];
Jacob2 = [cross(T(1:3,3), T(1:3,4));T(1:3,3)];
Jacob3 = [T(1:3,3);zero];
Jacob4 = [cross(T(1:3,3), T(1:3,4));T(1:3,3)];
J = [Jacob1 Jacob2 Jacob3 Jacob4];
```

```
J =

[   a2*sin(theta1 + theta2) + a1*sin(theta1),   a2*sin(theta1 + theta2) + a1*sin(theta1),  0,   a2*sin(theta1 + theta2) + a1*sin(theta1)]
[ - a2*cos(theta1 + theta2) - a1*cos(theta1), - a2*cos(theta1 + theta2) - a1*cos(theta1),  0, - a2*cos(theta1 + theta2) - a1*cos(theta1)]
[                                          0,                                          0, -1,                                          0]
[                                          0,                                          0,  0,                                          0]
[                                          0,                                          0,  0,                                          0]
[                                         -1,                                         -1,  0,                                         -1]
```