Robotics: Dynamics and Control

Assignment 2 - Report

Sreeja Guduri (2021102007)

Q1.EULER ANGLES AND ROTATION MATRICES

1.1 -

For this part of the question, we had to make a function that takes in Euler angles and outputs the corresponding rotation matrix. We use the ZYX convention to form the rotation matrix

The formula used for this:

Rotation about z0 of angle α + Rotation about y1 of angle β + Rotation about x2 of angle γ

$$T_{0,3} = T_{0,1}T_{1,2}T_{2,3} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{bmatrix} =$$

$$\begin{bmatrix} \cos(\alpha)\cos(\beta) & \cos(\alpha)\sin(\beta)\sin(\gamma)-\sin(\alpha)\cos(\gamma) & \cos(\alpha)\sin(\beta)\cos(\gamma)+\sin(\alpha)\sin(\gamma) \\ \sin(\alpha)\cos(\beta) & \sin(\alpha)\sin(\beta)\sin(\gamma)+\cos(\alpha)\cos(\gamma) & \sin(\alpha)\sin(\beta)\cos(\gamma)-\cos(\alpha)\sin(\gamma) \\ -\sin(\beta) & \cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{bmatrix}$$

1.2 -

In the second part of the question, we are asked to make a function that outputs the Euler angles given the rotation matrix.

The formula used for this part is:

$$\text{If } (r_{11} = r_{21} = 0 \Leftrightarrow \cos(\beta) = 0)\text{, then } \begin{cases} \beta = \frac{\pi}{2}, \\ \alpha = 0, \\ \gamma = \tan_2^{-1}(r_{12}, r_{22}) \end{cases} \quad \text{Else, then } \begin{cases} \beta = \tan_2^{-1}(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}) \\ \alpha = \tan_2^{-1}(r_{21}, r_{11}) \\ \gamma = \tan_2^{-1}(r_{32}, r_{33}) \end{cases}$$

1.3 -

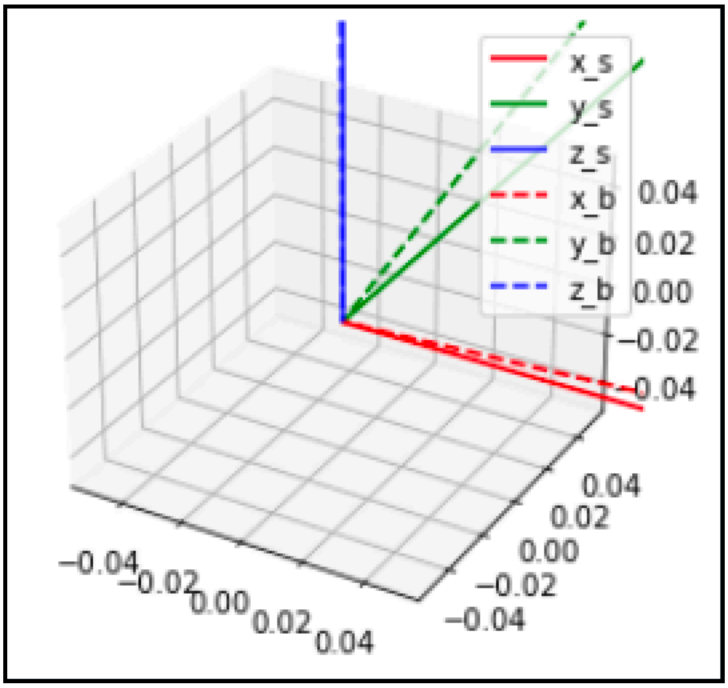
The functions are tested by giving an arbitrary rotation matrix and getting the corresponding Euler angles. Then, the resulting Euler angles are given as input to the second function which gives a rotation matrix.

We notice that the original rotation matrix that we started with is the same as the one we get after we pass the Euler angles through the function. Thus, the functions work as expected.

1.4 -

The two frames are graphically plotted on a 3D plot using matplotlib. I assume that they have the same origin and then plot the original {s} frame and the transformed {b} frame.

The dashed lines is the {b} frame and the solid lines represent the {s} frame.



Q2. 3R MANIPULATOR FORWARD KINEMATICS

2.1 -

We write a function to find the end-effector position (x,y) based on the given joint angles and link lengths. This is called forward kinematics.

The formula used for this is:

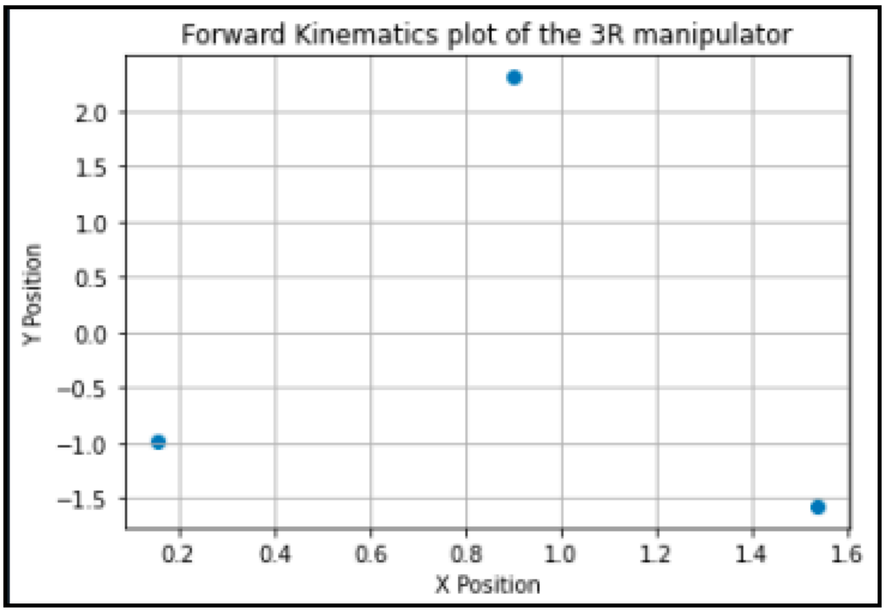
$$P = \begin{bmatrix} x_e \\ y_e \end{bmatrix} = \begin{bmatrix} L_{12}c_1 + L_{23}c_{1+2} + L_{34}c_{1+2+3} \\ L_{12}s_1 + L_{23}s_{1+2} + L_{34}s_{1+2+3} \end{bmatrix}$$

$$\gamma = \theta_1 + \theta_2 + \theta_3$$

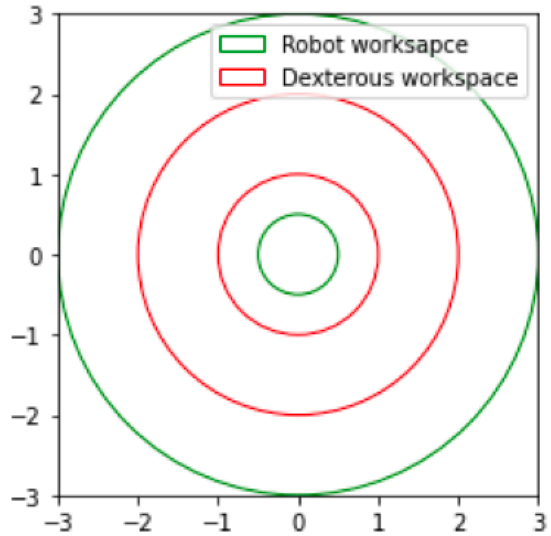
Where P is the final position of the end effector and 'gamma' is the final angle.

2.2 -

The forward kinematics is represented graphically by plotting the positions of the end effector as well as the two joints connecting the three links.



The dextrous workspace of the 3R manipulator is shown by a set of concentric circles whose radius depends on the link lengths.



Q3. AXIS-ANGLE REPRESENTATION

3.1 -

The vector n is a (1x3) vector that is used to find the rotation matrix. The elements of this matrix can be found using this formula:

$$\begin{aligned} a_x &= (r_{32} - r_{23})/2 \sin\theta \\ a_y &= (r_{13} - r_{31})/2 \sin\theta \\ a_z &= (r_{21} - r_{12})/2 \sin\theta. \end{aligned}$$

3.2 -

A program that takes in the rotation matrix and outputs the n vector and angle and vice-versa is written in code. It is then tested by checking if initial rotation matrix and final rotation matrix output from the second function is the same.

Q3. 7R MANIPULATOR DH REPRESENTATION

4.1 -

The DH parameters of the given 7R robot have been calculated and tabulated as shown below:

i	a_i (m)	α_i (rad)	d_i (m)	θ_i (rad)
1	0	$\frac{\pi}{2}$	0.310	θ_1
2	0	$-\frac{\pi}{2}$	0	θ_2
3	0	$\frac{\pi}{2}$	0.40	θ_3
4	0	$-\frac{\pi}{2}$	0	θ_4
5	0	$\frac{\pi}{2}$	0.39	θ_5
6	0	$-\frac{\pi}{2}$	0	θ_6
7	0	0	0.078	θ_7

Where the values given for 'd' are arbitrarily chosen.

4.2 -

Based on the above DH table, we write a function to compute the transformation matrix for each of the 7 joints and then find the final transformation matrix for the end effector by post-multiplying.

The formula to find the intermediate transformation matrices based on the DH parameters is given below:

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

Thus, we will have 7 such matrices for each joint in the robot and we will post multiply them to get the end-effector matrix.

4.3 -

To validate with the home configuration, we set the joint angles of all of the joints to be 0 and then find the final end-effector pose based on that.

4.4 -

To graphically show the robot configuration, I made frames at every point of the transformation of the robot, to show how the robot goes from its initial configuration.