ENPM662 Homework 4

Brendan Neal

November 11, 2023

Contents

1 Introduction

This homework assignment has students define the Jacobian matrix and define the forward velocity kinematics for a UR10 manipulator. Additionally, students must use inverse velocity kinematics in order to draw a circle.

2 Frame Definitions

Shown below in Figure 1 are the coordinate frame definitions I used for the UR10 Robot:

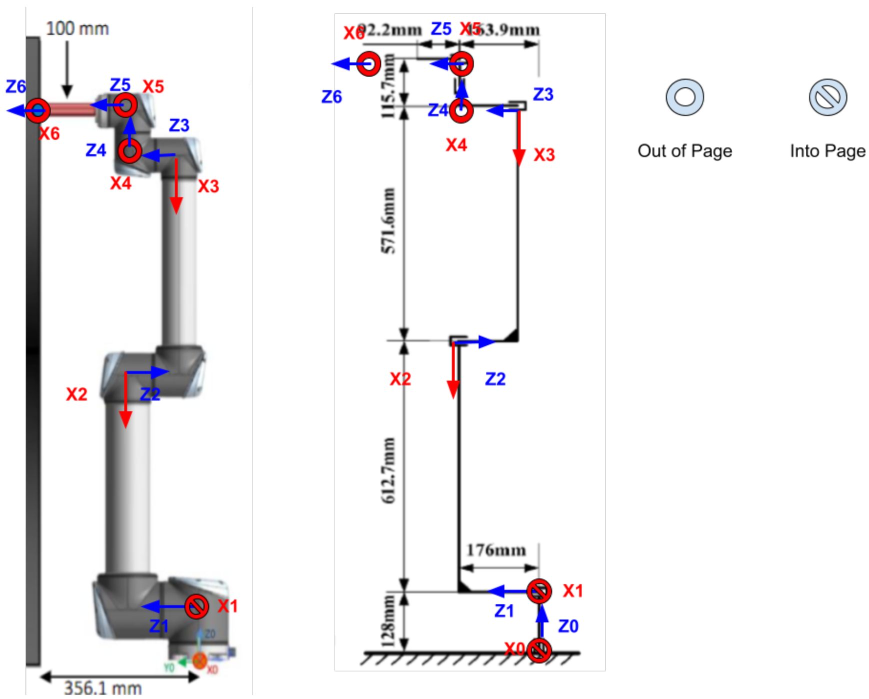


Figure 1: UR10 Robot Frame Definitions

3 D-H Table

Below is the D-H table I defined based on the coordinate frame definitions listed above:

D-H Table				
Link	α	a	d	θ
0 to 1	-90	0	d_1	θ_1^*
1 to 2	180	a_2	d_2	$90 + \theta_2^*$
2 to 3	180	a_3	d_3	θ_3^*
3 to 4	-90	0	d_4	$90 + \theta_4^*$
4 to 5	90	0	d_5	θ_5^*
5 to 6	0	0	d_6	θ_6^*

Where:

$$a_2 = -612.7mm$$

$$a_3 = -571.6mm$$

$$d_1 = 128mm$$

$$d_2 = 176mm$$

$$d_3 = 163.9mm$$

$$d_4 = 151.8mm$$

$$d_5 = 115.7mm$$

$$d_6 = 192.2mm$$

I wanted the end effector frame to be the tip of the pen, so I shifted the end effector frame another 100mm along the z axis (editing d_6 from the previous HW assignment).

4 Jacobian Matrix Setup

To set up the Jacobian matrix, I am using Method 2. First, each transformation matrix is calculated per row. The multiplication happens as follows:

$$T = R_z(\theta) * T_z(d) * T_x(a) * R_x(\alpha)$$

The homogeneous transformation matrices between each frame are given by:

$$T_1^0 = \begin{bmatrix} c(\theta_1^*) & 0 & -s(\theta_1^*) & 0 \\ s(\theta_1^*) & 0 & c(\theta_1^*) & 0 \\ 0 & -1 & 0 & 128 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2^1 = \begin{bmatrix} -s(\theta_2^*) & c(\theta_2^*) & 0 & 612.7 * s(\theta_2^*) \\ c(\theta_2^*) & s(\theta_2^*) & 0 & -612.7 * c(\theta_2^*) \\ 0 & 0 & -1 & 176 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_3^2 = \begin{bmatrix} c(\theta_3^*) & s(\theta_3^*) & 0 & -571.6 * c(\theta_3^*) \\ s(\theta_3^*) & -c(\theta_3^*) & 0 & -571.6 * s(\theta_3^*) \\ 0 & 0 & -1 & 163.9 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_4^3 = \begin{bmatrix} -s(\theta_4^*) & 0 & -c(\theta_4^*) & 0 \\ c(\theta_4^*) & 0 & -s(\theta_4^*) & 0 \\ 0 & -1 & 0 & 151.8 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_5^4 = \begin{bmatrix} c(\theta_5^*) & 0 & s(\theta_5^*) & 0 \\ s(\theta_5^*) & 0 & -c(\theta_5^*) & 0 \\ 0 & 1 & 0 & 115.7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_6^5 = \begin{bmatrix} c(\theta_6^*) & -s(\theta_6^*) & 0 & 0 \\ s(\theta_6^*) & c(\theta_6^*) & 0 & 0 \\ 0 & 0 & 1 & 192.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Now, to find the Jacobian matrix, I have to find each homogeneous transformation matrix of each joint with respect to the base frame. These are given by:

$$H_1^0 = T_1^0$$

$$H_2^0 = T_1^0 T_2^1$$

$$H_3^0 = T_1^0 T_2^1 T_3^2$$

$$H_4^0 = T_1^0 T_2^1 T_3^2 T_4^3$$

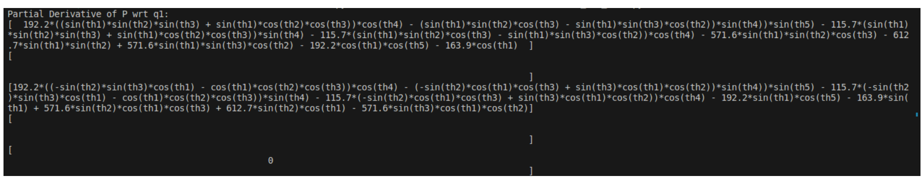
$$H_5^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4$$

$$H_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5$$

Now, to formulate the Jacobian matrix, I have to extract P and the Z components from each respective base-frame transformation matrices. P is the position of the end effector as a function of joint positions (last column, first three rows of H_6^0). Z is the third column, first three rows of each transformation matrix. Then, I can calculate each component of the Jacobian matrix in the following form:

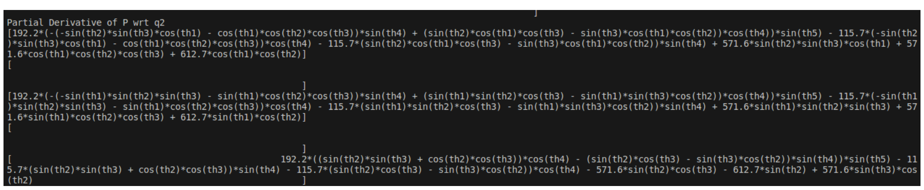
$$J_i = \begin{bmatrix} \frac{\partial P}{\partial q_i} \\ Z_i \end{bmatrix}$$

Using SymPy, I extract P from H_6^0 and calculate the general form of each partial derivative as:



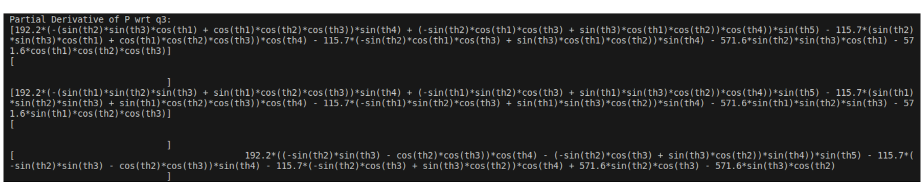
```
Partial Derivative of P wrt q1:
[  192.2*((sin(th1)*sin(th2)*sin(th3) + sin(th1)*cos(th2)*cos(th3))*cos(th4) - (sin(th1)*sin(th2)*cos(th3) - sin(th1)*sin(th3)*cos(th2))*sin(th4))*sin(th5) - 115.7*(sin(th1)
*sin(th2)*sin(th3) + sin(th1)*cos(th2)*cos(th3))*sin(th4) - 115.7*(sin(th1)*sin(th2)*cos(th3) - sin(th1)*sin(th3)*cos(th2))*cos(th4) - 571.6*sin(th1)*sin(th2)*cos(th3) - 612
.7*sin(th1)*sin(th2) + 571.6*sin(th1)*sin(th3)*cos(th2) - 192.2*cos(th1)*cos(th5) - 163.9*cos(th1)  ]
[
                                                                                                                                                         ]
[192.2*((-sin(th2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*cos(th4) - (-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*sin(th4))*sin(th5) - 115.7*(-sin(th2
)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*sin(th4) - 115.7*(-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*cos(th4) - 192.2*sin(th1)*cos(th5) - 163.9*sin(
th1) + 571.6*sin(th2)*cos(th1)*cos(th3) + 612.7*sin(th2)*cos(th1) - 571.6*sin(th3)*cos(th1)*cos(th2)]
[
                                                                                                                                                         ]
[
                                                                     0
                                                                                                                                                         ]
```

Figure 2: $\frac{\partial P}{\partial q_1}$



```
Partial Derivative of P wrt q2
[192.2*(-(-sin(th2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*sin(th4) + (sin(th2)*cos(th1)*cos(th3) - sin(th3)*cos(th1)*cos(th2))*cos(th4))*sin(th5) - 115.7*(-sin(th2
)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*cos(th4) - 115.7*(sin(th2)*cos(th1)*cos(th3) - sin(th3)*cos(th1)*cos(th2))*sin(th4) + 571.6*sin(th2)*sin(th3)*cos(th1) + 57
1.6*cos(th1)*cos(th2)*cos(th3) + 612.7*cos(th1)*cos(th2)]
[
                                                                                          ]
[192.2*(-(-sin(th1)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*sin(th4) + (sin(th1)*sin(th2)*cos(th3) - sin(th1)*sin(th3)*cos(th2))*cos(th4))*sin(th5) - 115.7*(-sin(th1
)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*cos(th4) - 115.7*(sin(th1)*sin(th2)*cos(th3) - sin(th1)*sin(th3)*cos(th2))*sin(th4) + 571.6*sin(th1)*sin(th2)*sin(th3) + 57
1.6*sin(th1)*cos(th2)*cos(th3) + 612.7*sin(th1)*cos(th2)]
[
                                                                                          ]
[                                                                     192.2*((sin(th2)*sin(th3) + cos(th2)*cos(th3))*cos(th4) - (sin(th2)*cos(th3) - sin(th3)*cos(th2))*sin(th4))*sin(th5) - 11
5.7*(sin(th2)*sin(th3) + cos(th2)*cos(th3))*sin(th4) - 115.7*(sin(th2)*cos(th3) - sin(th3)*cos(th2))*cos(th4) - 571.6*sin(th2)*cos(th3) - 612.7*sin(th2) + 571.6*sin(th3)*cos
(th2)                                                                                     ]
```

Figure 3: $\frac{\partial P}{\partial q_2}$



```
Partial Derivative of P wrt q3:
[192.2*(-(sin(th2)*sin(th3)*cos(th1) + cos(th1)*cos(th2)*cos(th3))*sin(th4) + (-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*cos(th4))*sin(th5) - 115.7*(sin(th2)
*sin(th3)*cos(th1) + cos(th1)*cos(th2)*cos(th3))*cos(th4) - 115.7*(-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*sin(th4) - 571.6*sin(th2)*sin(th3)*cos(th1) - 57
1.6*cos(th1)*cos(th2)*cos(th3)]
[
                                          ]
[192.2*(-(sin(th1)*sin(th2)*sin(th3) + sin(th1)*cos(th2)*cos(th3))*sin(th4) + (-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*cos(th4))*sin(th5) - 115.7*(sin(th1)
*sin(th2)*sin(th3) + sin(th1)*cos(th2)*cos(th3))*cos(th4) - 115.7*(-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*sin(th4) - 571.6*sin(th1)*sin(th2)*sin(th3) - 57
1.6*sin(th1)*cos(th2)*cos(th3)]
[
                                          ]
[                                                             192.2*((-sin(th2)*sin(th3) - cos(th2)*cos(th3))*cos(th4) - (-sin(th2)*cos(th3) + sin(th3)*cos(th2))*sin(th4))*sin(th5) - 115.7*(
-sin(th2)*sin(th3) - cos(th2)*cos(th3))*sin(th4) - 115.7*(-sin(th2)*cos(th3) + sin(th3)*cos(th2))*cos(th4) + 571.6*sin(th2)*cos(th3) - 571.6*sin(th3)*cos(th2)
                                          ]
```

Figure 4: $\frac{\partial P}{\partial q_3}$

```
Partial Derivative of P wrt q4:
[192.2*(-(-sin(th2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*sin(th4) - (-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*cos(th4))*sin(th5) - 115.7*(-sin(th
2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*cos(th4) + 115.7*(-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*sin(th4)]
[
                                                                                                                                            ]
[192.2*(-(-sin(th1)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*sin(th4) - (-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*cos(th4))*sin(th5) - 115.7*(-sin(th
1)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*cos(th4) + 115.7*(-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*sin(th4)]
[
                                                                                                                                            ]
[                           192.2*(-(-sin(th2)*sin(th3) - cos(th2)*cos(th3))*cos(th4) - (sin(th2)*cos(th3) - sin(th3)*cos(th2))*sin(th4))*sin(th5) + 115.7*(-sin(th
2)*sin(th3) - cos(th2)*cos(th3))*sin(th4) - 115.7*(sin(th2)*cos(th3) - sin(th3)*cos(th2))*cos(th4)                                          ]
```

Figure 5: $\frac{\partial P}{\partial q_4}$

```
Partial Derivative of P wrt q5:
[192.2*((-sin(th2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*cos(th4) - (-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*sin(th4))*cos(th5) + 192.2*sin(th1)*
sin(th5)]
[
       ]
[192.2*((-sin(th1)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*cos(th4) - (-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*sin(th4))*cos(th5) - 192.2*sin(th5)*
cos(th1)]
[
       ]
[                       192.2*(-(-sin(th2)*sin(th3) - cos(th2)*cos(th3))*sin(th4) + (sin(th2)*cos(th3) - sin(th3)*cos(th2))*cos(th4))*cos(th5)
       ]
```

Figure 6: $\frac{\partial P}{\partial q_5}$

$$\frac{\partial P}{\partial q_6} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Additionally, from each transformation matrix H, I have to extract the Z component of the Jacobian. Depicted in Figure 3 are the extracted Z components:

```
Z0_1:
[-sin(th1)]
[         ]
[cos(th1) ]
[         ]
[    0    ]
Z0_2:
[sin(th1) ]
[         ]
[-cos(th1)]
[         ]
[    0    ]
Z0_3:
[-sin(th1)]
[         ]
[cos(th1) ]
[         ]
[    0    ]
Z0_4:
[-(-sin(th2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*sin(th4) - (-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*cos(th4)]
[                                                                                                                                          ]
[-(-sin(th1)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*sin(th4) - (-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*cos(th4)]
[                                                                                                                                          ]
[                 -(-sin(th2)*sin(th3) - cos(th2)*cos(th3))*cos(th4) - (sin(th2)*cos(th3) - sin(th3)*cos(th2))*sin(th4)                    ]
Z0_5:
[((-sin(th2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*cos(th4) - (-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*sin(th4))*sin(th5) - sin(th1)*cos(th5)]
[                                                                                                                                                                         ]
[((-sin(th1)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*cos(th4) - (-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*sin(th4))*sin(th5) + cos(th1)*cos(th5)]
[                                                                                                                                                                         ]
[                          (-(-sin(th2)*sin(th3) - cos(th2)*cos(th3))*sin(th4) + (sin(th2)*cos(th3) - sin(th3)*cos(th2))*cos(th4))*sin(th5)                               ]
Z0_6:
[((-sin(th2)*sin(th3)*cos(th1) - cos(th1)*cos(th2)*cos(th3))*cos(th4) - (-sin(th2)*cos(th1)*cos(th3) + sin(th3)*cos(th1)*cos(th2))*sin(th4))*sin(th5) - sin(th1)*cos(th5)]
[                                                                                                                                                                         ]
[((-sin(th1)*sin(th2)*sin(th3) - sin(th1)*cos(th2)*cos(th3))*cos(th4) - (-sin(th1)*sin(th2)*cos(th3) + sin(th1)*sin(th3)*cos(th2))*sin(th4))*sin(th5) + cos(th1)*cos(th5)]
[                                                                                                                                                                         ]
[                          (-(-sin(th2)*sin(th3) - cos(th2)*cos(th3))*sin(th4) + (sin(th2)*cos(th3) - sin(th3)*cos(th2))*cos(th4))*sin(th5)                               ]
```

Figure 7: Z Jacobian Components

Once I have each Jacobian component, I can formulate the Jacobian matrix as follows:

$$J = \begin{bmatrix} J_1 & J_2 & J_3 & J_4 & J_5 & J_6 \end{bmatrix}$$

For some reason, pprint() is not making my matrix any easier to read. I wanted to include a screenshot from my terminal, but even then, it was too large to include in the report. Please see my code to view the full Jacobian Matrix.

5 Circle-Drawing Process

Shown below in Figure 3 is the circle setup drawing with coordinate Frame 0 transposed onto the center of the circle.

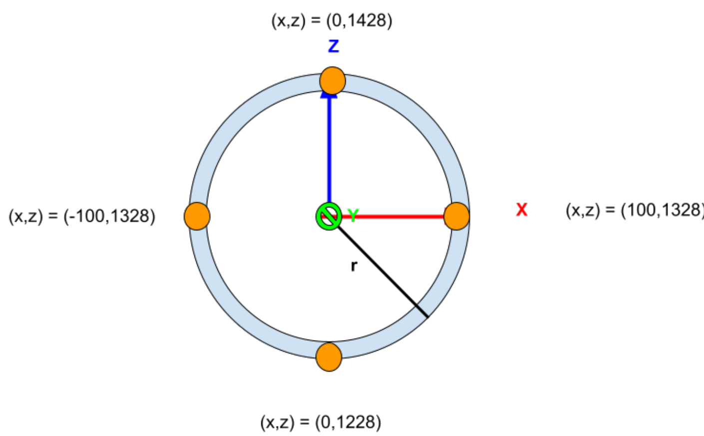


Figure 8: Circle Definition

I needed to define the position of the circle relative to the base frame of the robot. To do so, I used the figure in the problem statement to measure where Point S is relative to the base frame in the base frame's coordinate frame. Once I got the (X,Y,Z) positions of Point S in the base frame, I used the radius r to extrapolate where the center of the circle is in order to parametrically define the trajectory. Below are these definitions:

- r = 100mm
- Coordinates of point S relative to Frame 0: (0,356.1,1428).
- Coordinates of center relative to Frame 0: (0,356.1,1328).

Now I need to define the circle's trajectory parametrically. Looking at Figure 2, I can see that the x coordinate of the end effector has to move from $0 \rightarrow 100 \rightarrow 0 \rightarrow -100 \rightarrow 0$, therefore I can define it's trajectory as a sine function. Similarly, the z coordinate moves from $1428 \rightarrow 1328 \rightarrow 1228 \rightarrow 1328 \rightarrow 1428$, so I can define it's trajectory as a cosine function. Finally, y is a constant 356.1 mm. We set the period for x and z to be $\frac{2*\pi}{20}$ because we want to draw one circle in 20 seconds. Therefore, the circle's trajectory in parametric form relative to the base frame is given by:

$$x = r * sin(\theta(t))$$

$$y = 356.1$$

$$z = r * cos(\theta(t)) + 1328$$

Where:

$$\theta(t) = \frac{2 * \pi * t}{20}$$

If we differentiate the trajectory with respect to time, we can get the x, y, and z velocities of the circle in parametric form:

$$V_x = \dot{x} = r * cos(\theta(t)) * \dot{\theta(t)}$$

$$V_y = \dot{y} = 0$$

$$V_z = \dot{z} = -r * sin(\theta(t)) * \dot{\theta(t)}$$

Where:

$$\dot{\theta(t)} = \frac{2 * \pi}{20}$$

Next, we define q (the joint position matrix) to be

$$q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{bmatrix}$$

and $\dot{q}$ (the joint velocity matrix) to be

$$\dot{q} = \begin{bmatrix} \dot{q_1} \\ \dot{q_2} \\ \dot{q_3} \\ \dot{q_4} \\ \dot{q_5} \\ \dot{q_6} \end{bmatrix}$$

We also define $\dot{X}$ to be

$$\dot{X} = \begin{bmatrix} V_x \\ V_y \\ V_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} -r * sin(\theta(t)) * \dot{\theta(t)} \\ 0 \\ r * cos(\theta(t)) * \dot{\theta(t)} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

ω_x, ω_y, and ω_z are all set to 0 because the end effector (the pen) is fixed and has no angular velocity. With $\dot{X}$, J, and $\dot{q}$, we can write the following equations:

$$\dot{X} = J\dot{q}$$

$$\dot{q} = J^{-1}\dot{X}$$

In order to obtain the new joint angles, we use numerical integration:

$$q_i = q_{i-1} + \dot{q}_i \Delta t$$

Where $\Delta t = \frac{T}{N}$ and T is the total time (20s) and N is a selected number of data points (200 for a faster visualization). Finally, to plot the circle, we iterate through this procedure in a loop and extract/plot the end effector position from H_6^0 in 3-D space.

Some important notes I made for the completion of this assignment: I had to change the initial joint angles to be small, but nonzero because if they were zero, the robot would follow the velocity trajectory, but in the opposite direction.

Hence, my initial joint state looks as follows:

$$q_{init} = \begin{bmatrix} 0.0002 \\ 0.0001 \\ -0.0001 \\ 0.0001 \\ 0.0004 \\ 0.00001 \end{bmatrix}$$

6 Results - Circle Plot

Using the procedure outlined before this, I created a Python script that will perform the operations above. Below in Figures 9-11 is the outputted drawing of the circle:

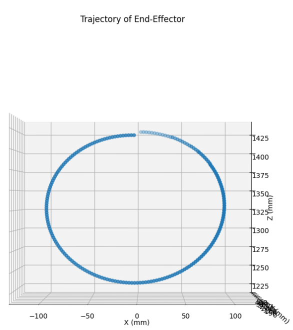


Figure 9: Trajectory of EE - XZ Plane

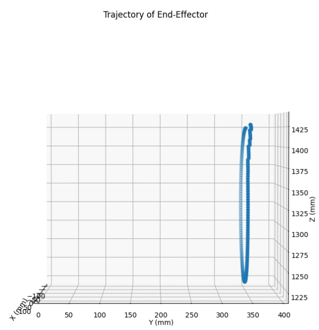


Figure 10: Trajectory of EE - YZ Plane

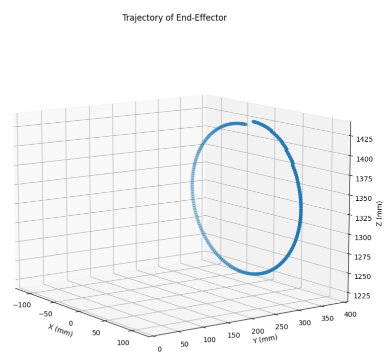


Figure 11: Trajectory of EE - XYZ Plane

The results are not exactly perfect due to using numerical integration and setting non-perfect zero home position, however, the goal of drawing a circle was achieved.