ENPM662 Homework 5

Brendan Neal

December 2, 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. Thirdly, students will use Lagrangian Dynamics to ensure the pen is pushing with a force of 5 N during the circle drawing process.

2 Setup from Previous Homework

2.1 Frame Definitions

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

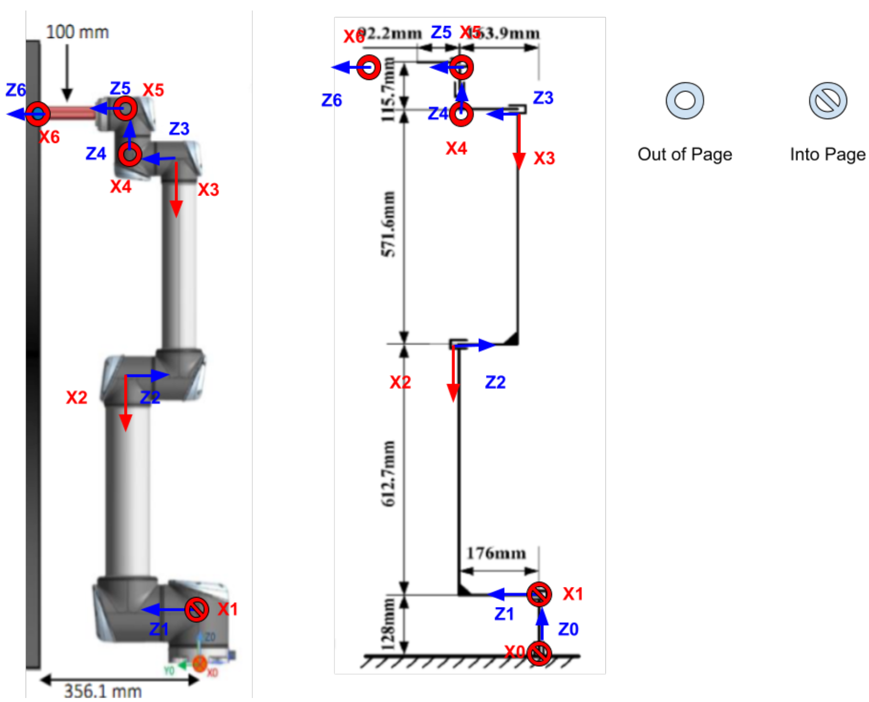

Figure 1: UR10 Robot Frame Definitions

2.2 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:

$$a_2 = -612.7mm$$

$$a_3 = -571.6mm$$

$$d_1 = 128mm$$

$$d_2 = 176mm$$

$$d_3 = 163.9mm$$

$$d_4 = 151.8mm$$

$$d_5 = 115.7mm$$

$$d_6 = 192.2mm$$

2.3 Jacobian

I am reusing the same process/code outputs in order to compute the Jacobian. Please see my HW4 report. From here onward, assume that I have an already computed Jacobian matrix, J.

2.4 Circle Drawing Process

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

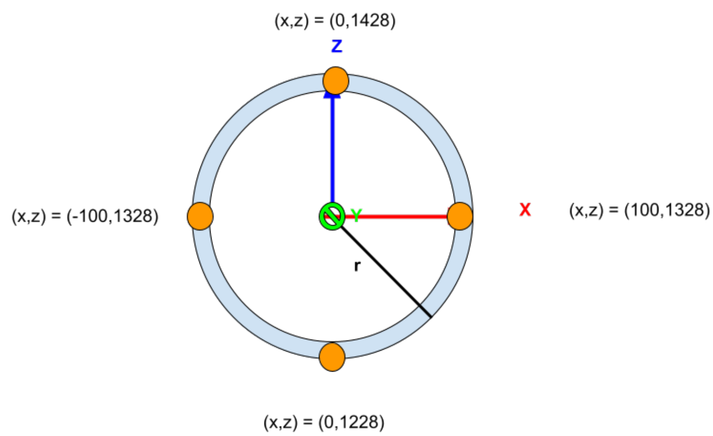

Figure 2: Circle Definition

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 $\frac{2*\pi}{200}$ because we want to draw one circle in 200 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:

$$\theta(t) = \frac{2 * \pi * t}{200}$$

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:

$$\dot{\theta(t)} = \frac{2 * \pi}{200}$$

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 (200s) and N is a selected number of data points (400 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.

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:

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

3 Computing Joint Torques

To calculate the joint torques for the UR10, we first have to examine the robot's dynamic equation. This is given by:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau + J^T(q)F$$

Assuming the robot is semi-static ($\dot{q} \approx 0$ and $\ddot{q} \approx 0$), the equation now becomes:

$$g(q) = \tau + J^T(q)F$$

To find torque, I can rearrange the equation as follows:

$$\tau = g(q) - J^T(q)F$$

Where:

$$\tau = \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \\ \tau_5 \\ \tau_6 \end{bmatrix}$$

and

$$F = \begin{bmatrix} F_x \\ F_y \\ F_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} 0 \\ -5 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

J(q) and F are known, but now to complete the equation we need to calculate g(q), which is explained in the next section.

4 Computing Gravity Matrix: $g(q)$

To start, we can calculate the potential energy for each link using:

$$P_i = m_i * g * z_{ci}$$

Where:

- P_i is the potential energy of the ith link.
- g is the gravity value: 9.81.
- z_{ci} is the center of mass z coordinate for link i.
- m_i is the mass of the ith link.

We define the total potential energy P as:

$$P = \sum_{i=1}^{n} P_i$$

And differentiating P with respect to q_k gives us the gravity matrix:

$$g(q) = \begin{bmatrix} \frac{\partial P}{\partial q_1} \\ \frac{\partial P}{\partial q_2} \\ \frac{\partial P}{\partial q_3} \\ \frac{\partial P}{\partial q_4} \\ \frac{\partial P}{\partial q_5} \\ \frac{\partial P}{\partial q_6} \end{bmatrix}$$

4.1 Mass Information

Using information garnered from this link, I was able to back out the mass information for the UR10. This is displayed in the table below:

UR10 Mass Information	
Link	Mass (kg)
0 to 1	7.1
1 to 2	12.7
2 to 3	4.27
3 to 4	2
4 to 5	2
5 to 6	0.365

4.2 Calculating Center of Mass

Unfortunately, I could not open the CAD file directly in order to discern the center of mass information. As a result, I have to calculate the center of mass information directly. To do so, I had to individually shift each of the joint frames to the center of masses of the links that they influence. I assume that the center of masses for each link are at the centroid of each link (even though in reality they could deviate a bit). Depicted below in Figure 3 is all of the shifts I performed:

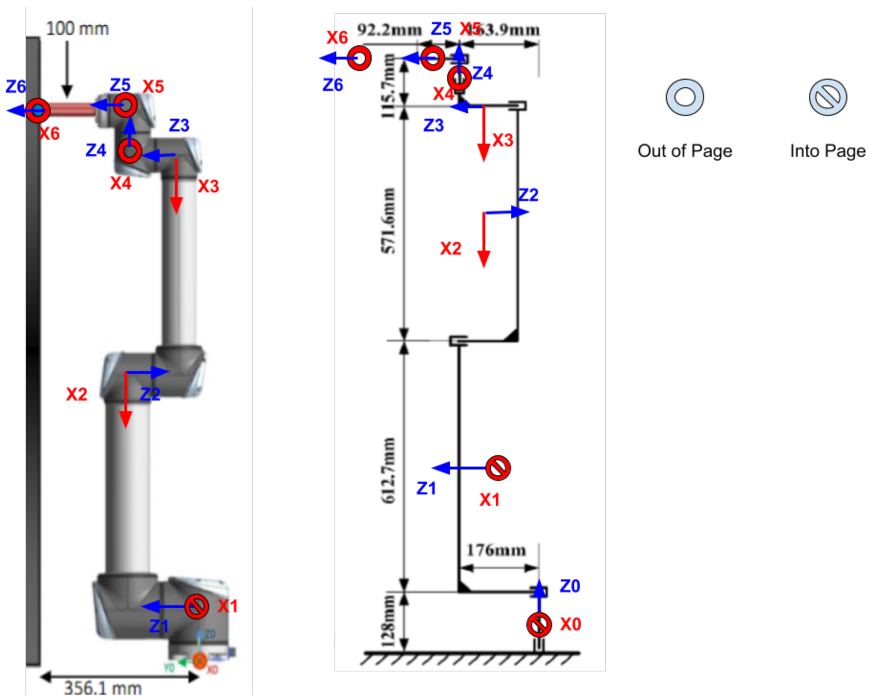

Figure 3: UR10 COM Frame Shifts

This figure is slightly deceiving because the calculations of center of mass were not performed all at once. The figure shows where each of the frame shifts occur when I calculate center of mass. This makes more sense below when I describe my process.

I will use frame 3 as an example of my process, then note that I repeated the process for the rest of the joints. I first had to redefine the D-H parameters for frame 3 alone (i.e. keeping the DH parameters the same for frames 0 through 2). I then calculated the individual transformation matrices for each of the rows, denoting the new parameters from frame 3 to be $T^0_{3_{new}}$. I then performed:

$$H^0_{3_{new}} = T^0_1 T^1_2 T^2_{3_{new}}$$

And extracted the z component (last column, third row) as the z component of the center of mass for the link.

I repeated the process from above and pulled my z components from each of these transformation matrices:

$$H^0_{1_{new}} = T^0_{1_{new}}$$

$$H^0_{2_{new}} = T^0_1 T^1_{2_{new}}$$

$$H^0_{3_{new}} = T^0_1 T^1_2 T^2_{3_{new}}$$
$$H^0_{4_{new}} = T^0_1 T^1_2 T^2_3 T^3_{4_{new}}$$
$$H^0_{5_{new}} = T^0_1 T^1_2 T^2_3 T^3_4 T^4_{5_{new}}$$
$$H^0_{6_{new}} = T^0_1 T^1_2 T^2_3 T^3_4 T^4_5 T^5_{6_{new}}$$

Using the COM components extracted above, I can then use the formulas and iterative procedure depicted in sections 3 and 4 to compute the gravity matrix and plot joint torques. The numeric changes in D-H Parameters can be found in my code.

5 Results

Using the torque equation found in Section 3, we can iteratively calculate the joint torques in a loop while plotting the circle. We can then save this data and plot them with respect to time. Below in Figure 4 is a plot of all 6 joint torques as a function of time:

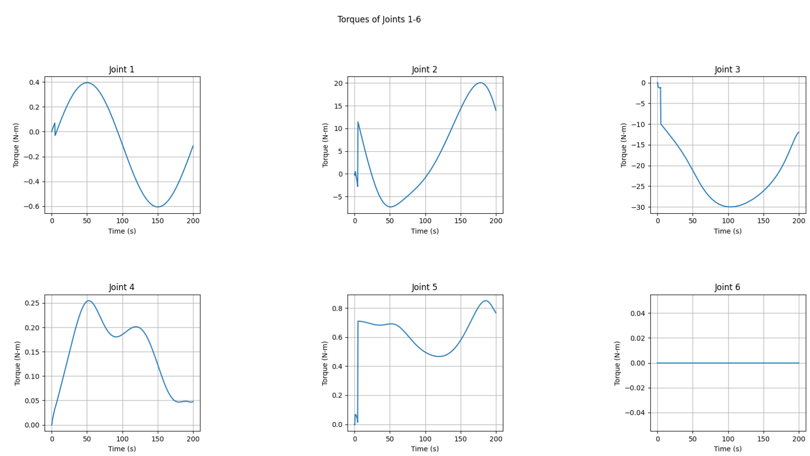

Figure 4: Joint Torques vs. Time