ENPM662 Homework 3

Brendan Neal

October 28, 2023

Contents

1 Introduction

This homework has students practicing forming D-H tables for a set of manipulators. Additionally, this homework has students forming transformation matrices and geometric validation based on said D-H tables.

2 Problem 1

2.1 Coordinate Frame Definitions

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

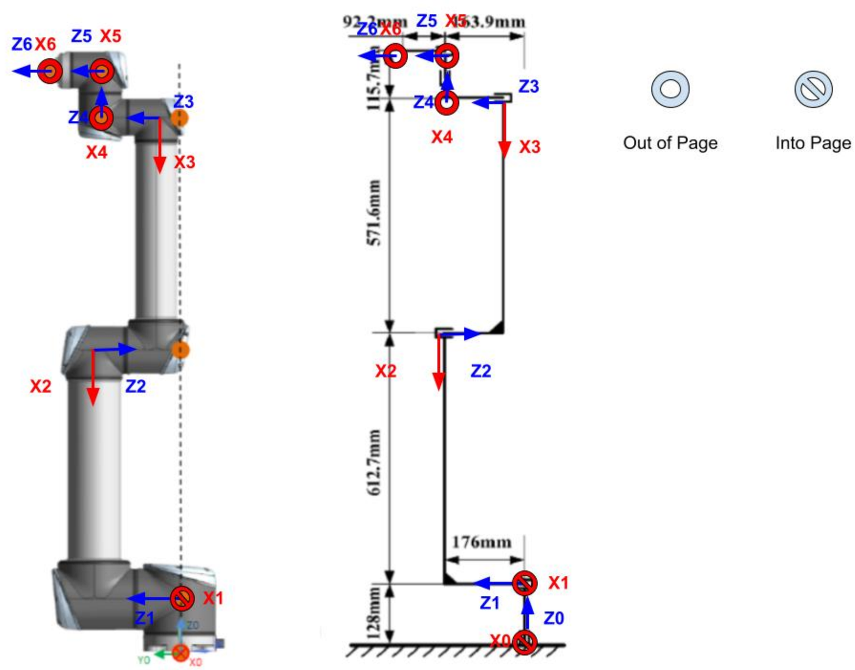

Figure 1: UR10 Robot Frame Definitions

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:

$$a_2 = -612.7mm$$

$$a_3 = -571.6mm$$

$$d_1 = 128mm$$

$$d_2 = 176mm$$

$$d_3 = 163.9mm$$

$$d_4 = 151.8mm$$

$$d_5 = 115.7mm$$

$$d_6 = 92.2mm$$

I calculated d_3 and d_4 using the given drawing and performing:

$$d_3 = 176 - (176 - 163.9) = 163.9$$

$$d_4 = 163.9 - (176 - 163.9) = 151.8$$

2.3 Transformation Matrices

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 transformation matrices for this problem are outlined below. Please note that when SymPy sees the +/- 90 within a trig equation, it automatically changes the trig equation accordingly. Hence, if some of the transformation matrices look different, just know that the +90 in link 2 and link 4 have been accounted for and transformed by SymPy.

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

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

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

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

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

$$T_6 = \begin{bmatrix} c(\theta_6^*) & -s(\theta_6^*) & 0 & 0 \\ s(\theta_6^*) & c(\theta_6^*) & 0 & 0 \\ 0 & 0 & 1 & 92.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.4 Final Transformation Matrix

Shown below in Figure 2 is the final transformation matrix after performing:

$$T_{final} = T_1 * T_2 * T_3 * T_4 * T_5 * T_6$$

I am inserting this as an image because it is way too tedious to type out. Additionally, even pprint() couldn't make it any easier to read, but based on my geometric validation, I believe this is correct. Please see my code to zoom in and inspect more.

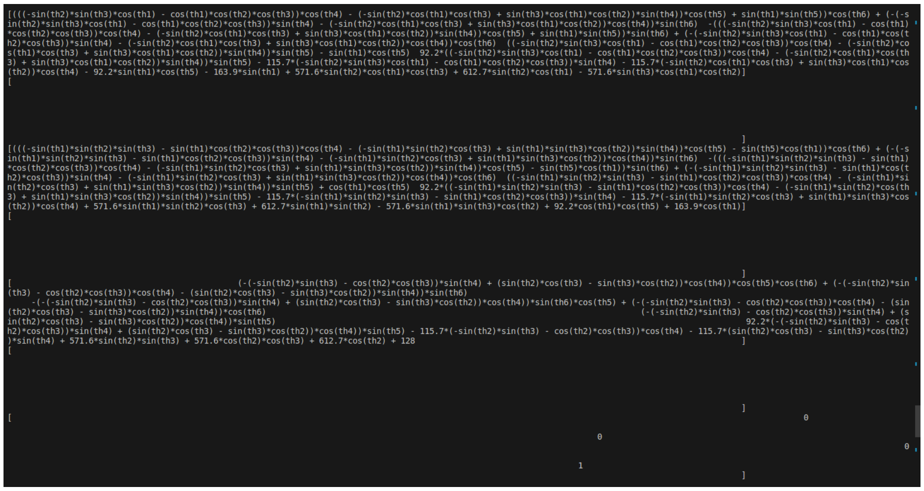

Figure 2: Final Transformation Matrix

2.5 Geometric Validation

2.5.1 Only θ_1^* Rotated by 90

$$\begin{bmatrix} 0 & 0 & -1 & -256.1 \\ -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1428.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5.2 Only θ_2^* Rotated by 90

$$\begin{bmatrix} 0 & 1 & 0 & 1300.0 \\ 0 & 0 & 1 & 256.1 \\ 1 & 0 & 0 & 128.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5.3 Only θ_3^* Rotated by 90

$$\begin{bmatrix} 0 & -1 & 0 & -687.3 \\ 0 & 0 & 1 & 256.1 \\ -1 & 0 & 0 & 740.7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5.4 Only θ_4^* Rotated by 90

$$\begin{bmatrix} 0 & 1 & 0 & 115.7 \\ 0 & 0 & 1 & 256.1 \\ 1 & 0 & 0 & 1312.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

2.5.5 Only θ_5^* Rotated by 90

$$\begin{bmatrix} 0 & 0 & -1 & -92.2 \\ -1 & 0 & 0 & 163.9 \\ 0 & 1 & 0 & 1428.0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3 Problem 2

3.1 Coordinate Frame Definitions

Figure 3 below, I have defined the coordinate frames for each of the joints in the Kuka robot. Please note, I performed a shift of origin from joint 3 and joint 2 (frame 2 and frame 1) in order to get the intersect/perpendicular condition for x_2 and z_1. I also performed a shift of origin from joint 5 and joint 4 (frame 4 and frame 3) in order to get the intersect/perpendicular condition for x_4 and z_3.

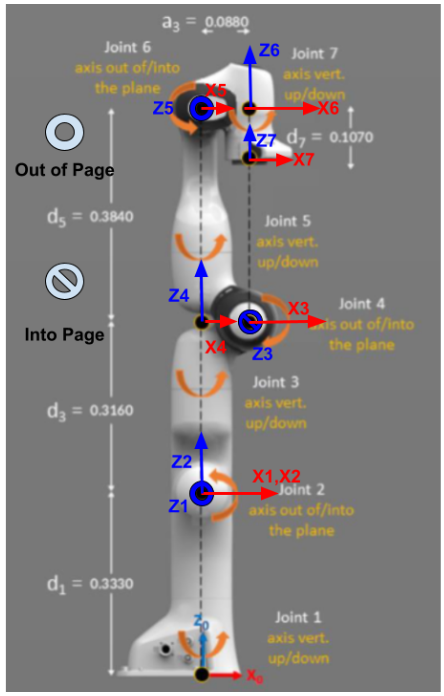

Figure 3: Kuka Robot Frame Definitions

3.2 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	-90	0	0	θ_2^*
2 to 3	-90	a_3	d_3	θ_3^*
3 to 4	90	$-a_3$	0	θ_4^*
4 to 5	90	0	d_5	θ_5^*
5 to 6	-90	a_3	0	θ_6^*
6 to 7	0	0	$-d_7$	θ_7^*