Final Project Report

The Human Arm

<u>Members</u>

Venkata Sai Charan Paladugu - vpaladug

Jason Chen - 116605403 - jchen777

ENPM662 Intro to Robot Modeling

Fall 2023

Dr. Reza Monfaredi

Introduction

We designed a robotic system to imitate the traits and features of an actual human arm. The creation of an arm controller and a replication of the precise joint functions are the most fundamental needs. The arm has the same functionality as a real human barring the feedback capabilities. The arm however is designed from the elbow and up.

Application

This system has the potential to have several applications in the prosthetics industry, upon extending this system by enabling a feedback mechanism to the brain we can use it as a prosthetic arm for veterans who lose their arms in battle or for people who lose it due to an accident. The arm can also mimic a real human arm with the help of a remote-controlled device or even mirror the other arm in case creating neural feedback might not be feasible.

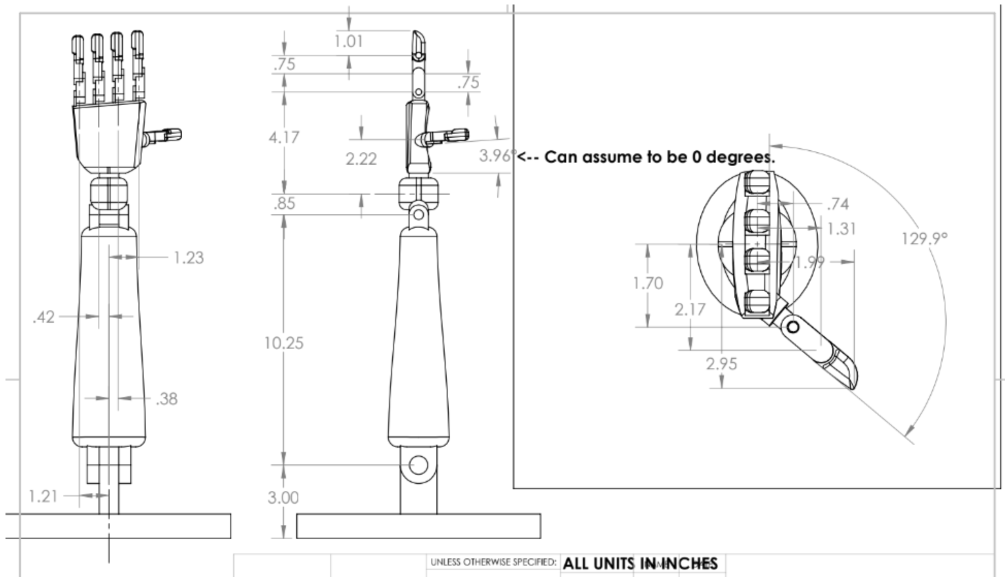
Robot Type

Our robot can be used in medical robotics as a prosthetic or as a manipulator to lift objects.

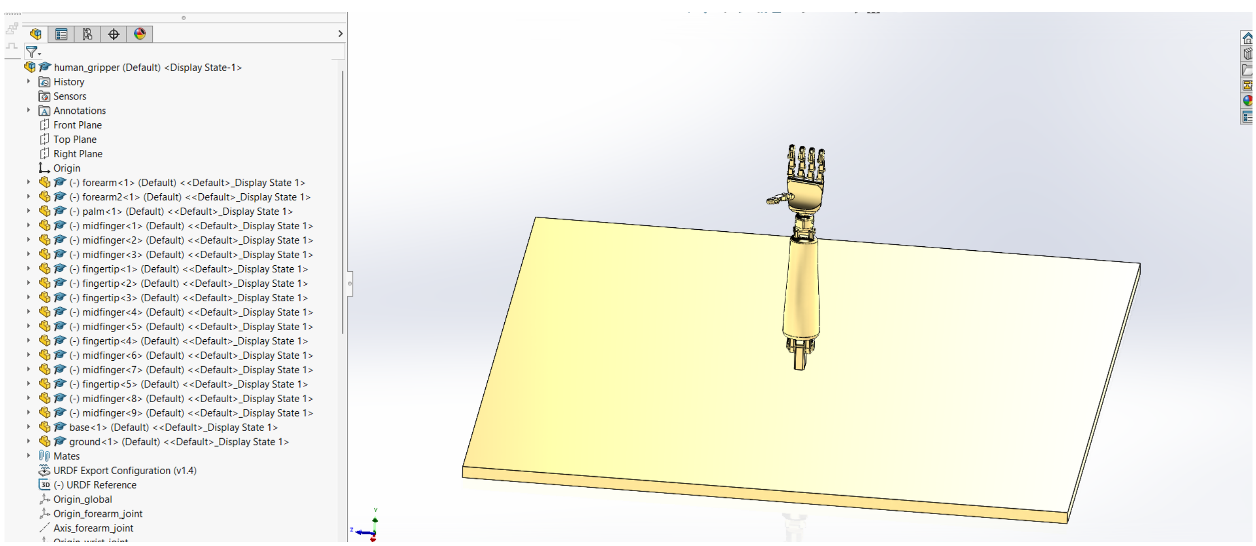
Robot Specifications

Our robot is a 17 Degree of Freedom robot with all the fingers in the arm being parallel robots by themselves. All the joints used in the robot system are revolute. However, considering the whole model from the base of the elbow joint to the topic of any of the bigger fingers on a human hand we get a system with 6 Degrees of Freedom. The elbow joint is built 3 cm from the ground frame placing a dummy link from the base. The elbow joint in reality can rotate about two axis, we however considered rotation about one axis because the rotation about the other axis is very minute. The wrist has a spherical joint in real life arm, to get a realistic model we used 2 revolute joints placed above each other which have their axis of rotation perpendicular to each other. There are 2 different axes for the whole robot, one for all the bigger fingers and one for the thumb finger because the bigger fingers are all oriented in the same direction as the wrist but the thumb however has a different orientation. For the bigger fingers, there are 3 revolute joints placed above each other at some distance. The thumb only has 2 revolute joints in it. The end effectors are the tip of the fingers.

CAD Models



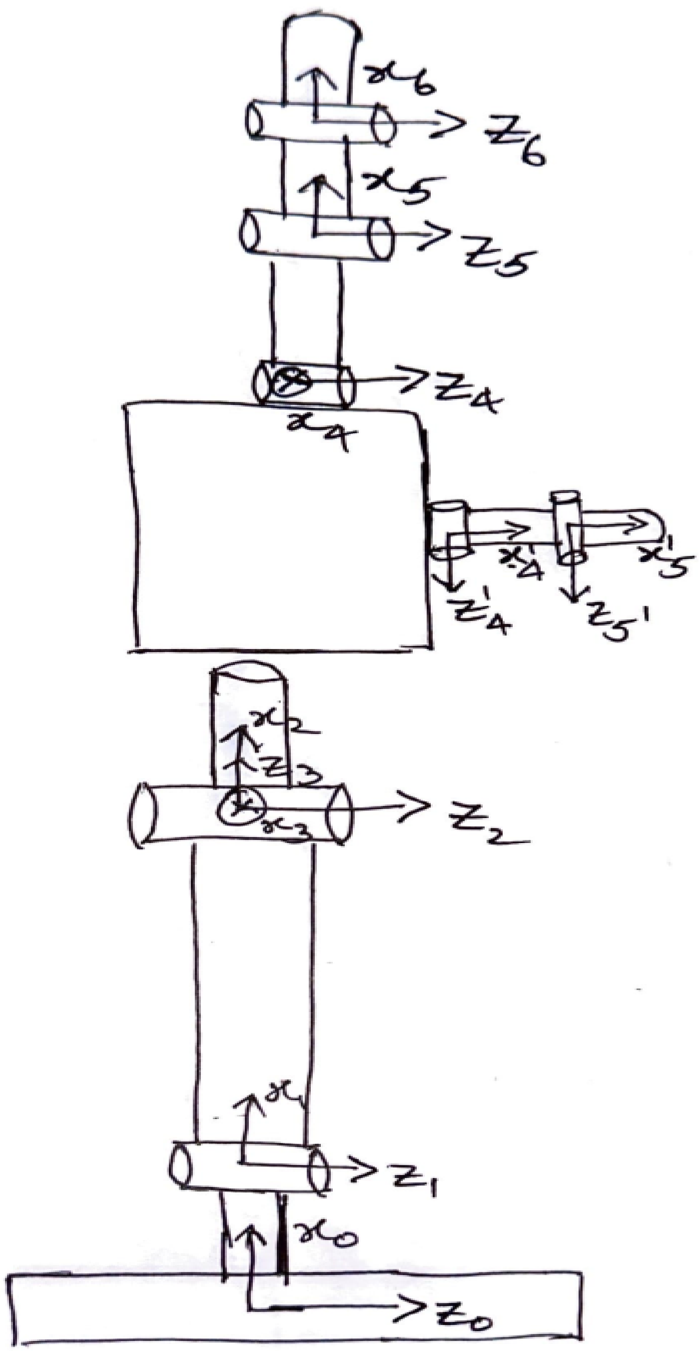
Drawing with dimensions for easier generation of DH table parameters. Some assumptions were made. Trade-offs have been made between the realism and feasibility aspects. A lot of thought was put into friendly design so that we can realistically see humans using it.



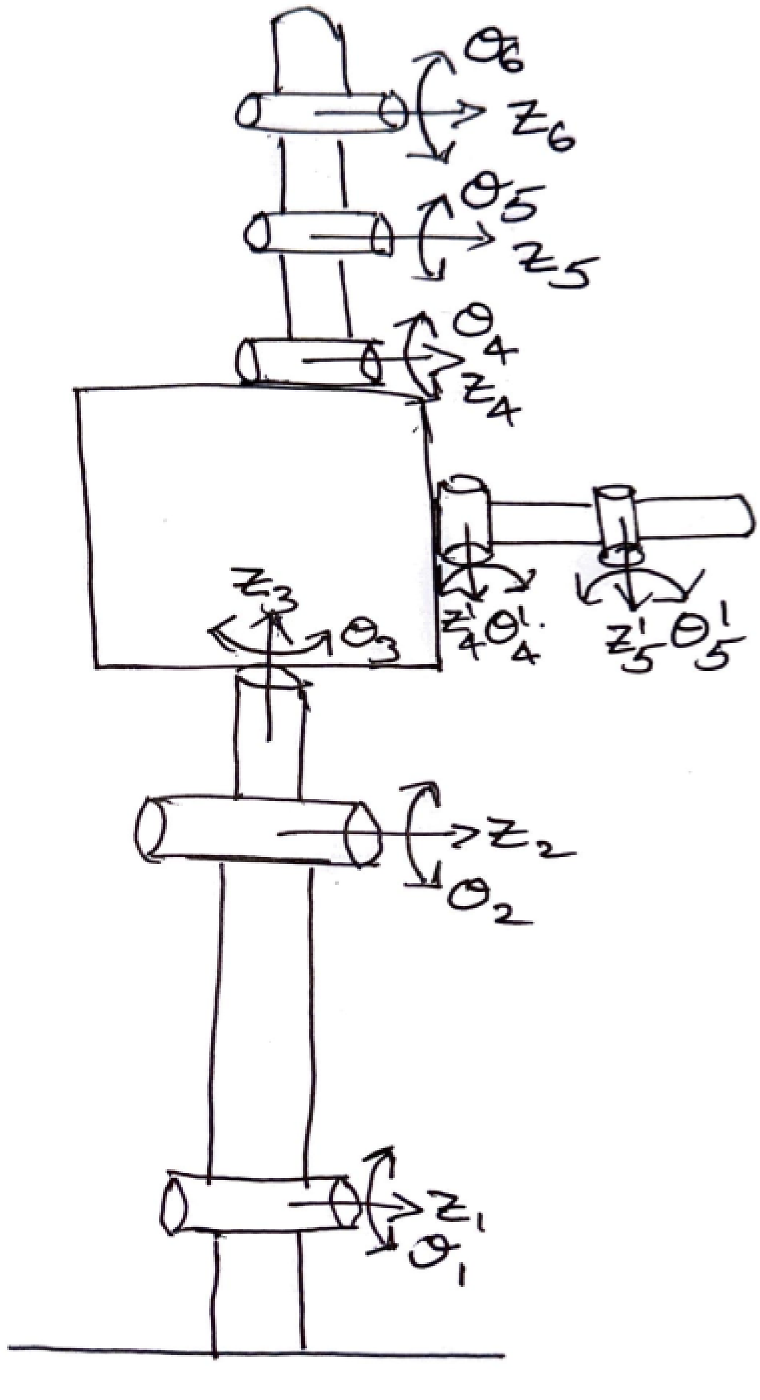
The material was recently changed into gold to give the model more weight and was part of the process of troubleshooting issues with model physics in Gazebo. Otherwise, the same models were used for the finger segments which explains why they look so similar to each other. The model was fixed on the ground, three inches above the origin to simplify the initial troubleshooting process. The goal was to attach this to a human model one day. The large platforms provide stability, which is crucial for a moving arm, especially if it is interacting with other things.

DH Parameters

The frame assignment is as follows:



The rotation of each joint is given by



Dimensions are given in the Solidworks CAD drawing !

For the bigger Fingers:

Links	a_i	α_i	d_i	θ_i
0-1	3	0	0	0
1-2	10.25	0	0	θ_1
2-3	0	-90	0	θ_2-90
3-4	0	90	5.02	θ_3
4-5	0.75	0	0.38(This changes for each finger)	θ_4+90
5-6	0.75	0	0	θ_5
6-7	1.01	0	0	θ_6

For Thumb:

Links	a_i	α_i	d_i	θ_i
0-1	3	0	0	0
1-2	10.25	0	0	θ_1
2-3	0.85	90	0	θ_2
3-4	2.22	-90	1.7	θ_3-90
4-5	0.47	90	0	θ_4+90
5-6	0.78	90	0	θ_5

Forward Kinematics

The Final transformation matrix for the system is shown below:

```
Transformation Matrix for System 1 (T1):
⎡((-(-sin(th1)·sin(th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th3)·cos(th4))·sin(th5) + ((-sin(th1)·sin(th2) + cos(th1)·cos(th2))·cos
⎢(-(-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th4))·sin(th5) + (-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·cos
⎢
⎣

(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3))·cos(th5))·cos(th6) + ((-(-sin(th1)·sin(th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(th2
(th3)·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4))·cos(th5))·sin(th6) + ((-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2) +
                  (-sin(th3)·sin(th4)·sin(th5) + sin(th3)·cos(th4)·cos(th5))·sin(th6) + (sin(th3)·sin(th4)·cos(th5) + sin(th3)·sin(th5)·cos(th4))·cos(th6)
                                                                                                  0
)·cos(th1))·cos(th3)·cos(th4))·cos(th5) - ((-sin(th1)·sin(th2) + cos(th1)·cos(th2))·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3))·sin(th5))·sin(th6
 sin(th2)·cos(th1))·cos(th4))·cos(th5) + (-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·cos(th3)·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4))·sin(th5))·cos(th6)



)  -((-(-sin(th1)·sin(th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th3)·cos(th4))·sin(th5) + ((-sin(th1)·sin(th2) + cos(th1)·cos(th2))·
    (-(-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th4))·sin(th5) + (-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·



cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3))·cos(th5))·sin(th6) + ((-(-sin(th1)·sin(th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(
cos(th3)·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4))·cos(th5))·cos(th6) - ((-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2
                     (-sin(th3)·sin(th4)·sin(th5) + sin(th3)·cos(th4)·cos(th5))·cos(th6) - (sin(th3)·sin(th4)·cos(th5) + sin(th3)·sin(th5)·cos(th4))·sin(th6)
                                                                                                    0
th2)·cos(th1))·cos(th3)·cos(th4))·cos(th5) - ((-sin(th1)·sin(th2) + cos(th1)·cos(th2))·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3))·sin(th5))·cos(
) + sin(th2)·cos(th1))·cos(th4))·cos(th5) + (-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·cos(th3)·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4))·sin(th5))·sin(t



th6)  (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th3)  1.01·((-(-sin(th1)·sin(th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th3)·cos(th
h6)   (sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th3)   1.01·(-(-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(t
                          cos(th3)
                             0
```

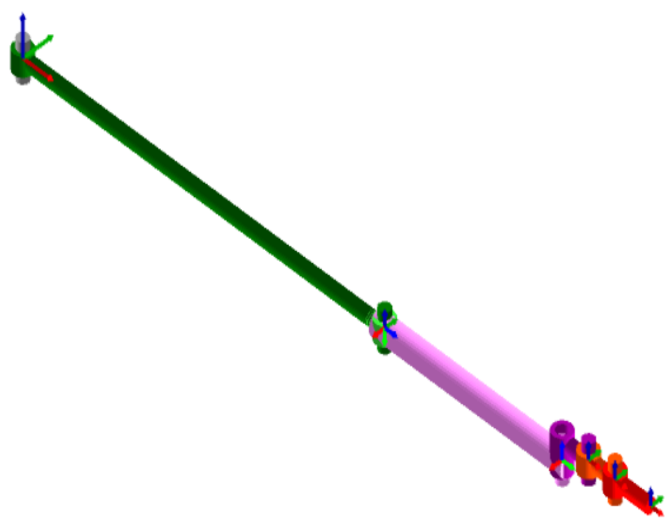

```
th6)  (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th3)  1.01·((-(-sin(th1)·sin(th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th3)·cos(th
h6)   (sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th3)   1.01·(-(-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(t
                          cos(th3)
                             0
```


```
4))·sin(th5) + ((-sin(th1)·sin(th2) + cos(th1)·cos(th2))·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3))·cos(th5))·cos(th6) + 1.01·((-(-sin(th1)·sin(
h4))·sin(th5) + (-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·cos(th3)·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4))·cos(th5))·sin(th6) + 1.01·((-(sin(th1)·sin(



th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th3)·cos(th4))·cos(th5) - ((-sin(th1)·sin(th2) + cos(th1)·cos(th2))·cos(th4) - (sin(th1)·c
th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th4))·cos(th5) + (-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·cos(th3)·cos(th4) - (s
                                                                                          1.01·(-sin(th3)·sin(th4)·sin(th5) + sin(th3)·cos(th4)·cos(th5))·sin(th6)

os(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3))·sin(th5))·sin(th6) + 0.75·(-(-sin(th1)·sin(th2) + cos(th1)·cos(th2))·sin(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos
in(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4))·sin(th5))·cos(th6) + 0.38·(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th3) - 0.75·(sin(th1)·sin(th2) - cos(th1)·cos(th2))·s
 + 1.01·(sin(th3)·sin(th4)·cos(th5) + sin(th3)·sin(th5)·cos(th4))·cos(th6) + 0.75·sin(th3)·sin(th4)·cos(th5) + 0.75·sin(th3)·sin(th4) + 0.75·sin(th3)·sin(th5)·cos(th4) + 0
                                                         1
(th3)·cos(th4))·sin(th5) + 0.75·((-sin(th1)·sin(th2) + cos(th1)·cos(th2))·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3))·cos(th5) + 0.75·(-sin(th1)·
in(th4)·cos(th3) + 0.75·(sin(th1)·cos(th2) + sin(th2)·cos(th1))·cos(th4) + 0.75·(-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·sin(th4)·cos(th3) + (sin(th1)·cos(th2) + sin(th2)
.38·cos(th3)

sin(th2) + cos(th1)·cos(th2))·cos(th4) + 0.38·(sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th3) - 0.75·(sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4)·cos(th3) - 5.02·sin(
·cos(th1))·cos(th4))·cos(th5) + 0.75·(-(sin(th1)·sin(th2) - cos(th1)·cos(th2))·cos(th3)·cos(th4) - (sin(th1)·cos(th2) + sin(th2)·cos(th1))·sin(th4))·sin(th5) + 5.02·sin(th



th1)·sin(th2) + 5.02·cos(th1)·cos(th2) + 10.25·cos(th1)⎤
1)·cos(th2) + 10.25·sin(th1) + 5.02·sin(th2)·cos(th1)  ⎥
                                                       ⎥
                                                       ⎦
```

Forward Kinematics Validation

We used RoboAnalyzer to validate our forward kinematics for the system for 4 configurations. However, RoboAnalyser doesn't account for dummy frames so we implemented the system from the first joint rather than the base. However, the length of the base link is considered in the code. Thus, you can always see there is a difference of 3 units in the x coordinates in the matrix and the obtained image. The blue line represents the z-axis, the red line represents the x-axis and the red line represents the y-axis of each joint. The wrist joint has 2 joints on top of each other and perpendicular to each other. Hence the axes overlap with each other and might be difficult to understand. The validation is done for the middle finger of the hand.

Configuration-1: The home position of the hand is given when θ_2=-90 and θ_4=90. This is because of the offset values given in the DH Table. The figure shown below shows the front view of the system.



This is the Validation matrix for the system at home position:

```
Validation Matrix for System 1:
⎡1  0  0  20.78⎤
⎢              ⎥
⎢0  1  0    0  ⎥
⎢              ⎥
⎢0  0  1  0.38 ⎥
⎢              ⎥
⎣0  0  0    1  ⎦
```

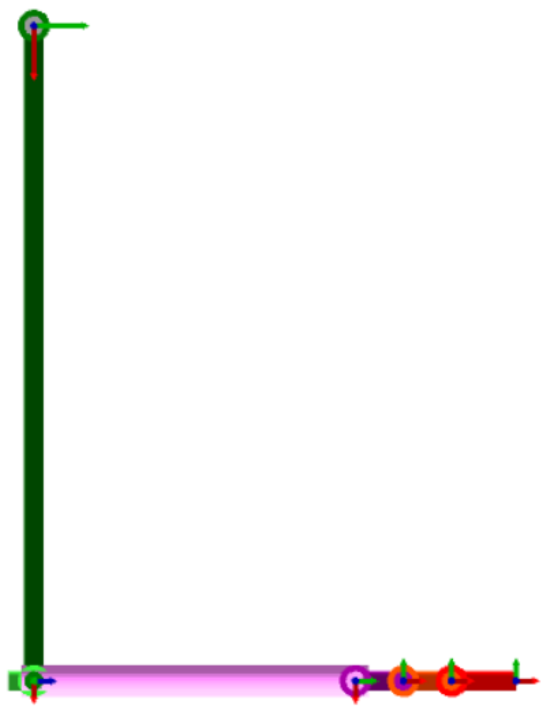
The x,y,z coordinates from the matrices show that the length of the system along the x-axis should be 20.78 which can be verified from the cad model. This is the total length of the arm. The z coordinate signifies the offset of the finger and the axis of the rest of the arm from elbow to wrist.

<u>Configuration-2:</u> The second configuration taken is when the middle joint of the finger(θ_5) is rotated by an angle of 90^o. This must result in the decrease of the x coordinate by the length of the last 2 links(1.01+0.75) and increase in the y coordinate by the same length. The z-coordinate remains the same. This can clearly be seen in the validation matrix below and the system visualization.

This is the validation matrix for the second configuration:

```
Validation Matrix for System 1:
⎡0  -1  0  19.02⎤
⎢               ⎥
⎢1  0   0  1.76 ⎥
⎢               ⎥
⎢0  0   1  0.38 ⎥
⎢               ⎥
⎣0  0   0    1  ⎦
```

Configuration-3: The third configuration is when the wrist is rotated about the first joint in the wrist(θ_2) is rotated by 90° and the rest of the hand is in the home position. This should result in the x coordinate having length of the system till the wrist and the rest being added to the y coordinate.The visualization and validation matrix for this configuration are given below which confirm the same.



The validation matrix for configuration 3 is given below:

Validation Matrix for System 1:

$$\begin{bmatrix} 0 & -1 & 0 & 13.25 \\ 1 & 0 & 0 & 7.53 \\ 0 & 0 & 1 & 0.38 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

<u>Configuration-4:</u> This configuration depicts when the elbow is rotated by 90°. This should result in rotating the whole system from the xz plane to the yz plane except for the base link which lies along the x axis. This can be seen clearly in the validation matrix but since we aren't considering the base link in the visualization the system doesn't show the base link but shows the change in

axis. The image might look like it is in its home position but if you look at the axis there is a change of planes. The visualization and validation matrix are given below:

This is the Validation matrix.

Validation Matrix for System 1:

$$\begin{bmatrix} 0 & -1 & 0 & 3 \\ 1 & 0 & 0 & 17.78 \\ 0 & 0 & 1 & 0.38 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Inverse Kinematics

The jacobian for inverse kinematics of the system is given below:

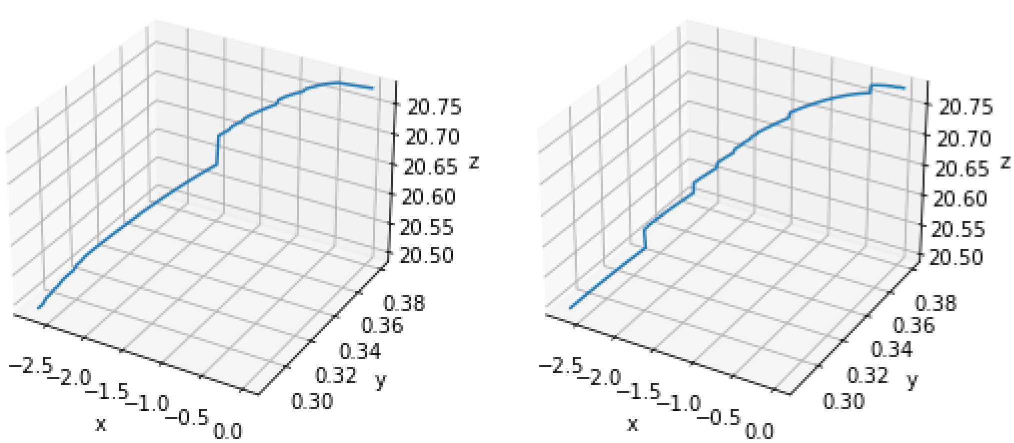
```
In [5]: pprint(trigsimp(JacobianMat))
⎡
⎢
⎢-10.25⋅sin(θ_0) + 0.38⋅sin(θ_2)⋅cos(θ_0 + θ_1) - 0.75⋅sin(θ_3)⋅cos(θ_2)⋅cos(θ_0 + θ_1
⎢
⎢
⎢
⎢
⎢
⎢
⎢
⎣

                 -0.75⋅sin(θ_2)⋅sin(θ_3) - 0.75⋅sin(θ_2)⋅sin(θ_3 + θ_4) - 1.01⋅sin(θ

) - 0.75⋅sin(θ_0 + θ_1)⋅cos(θ_3) - 0.75⋅sin(θ_0 + θ_1)⋅cos(θ_3 + θ_4) - 1.01⋅sin(θ_0 +

                                                                  0

                                                                  0

                                                                  0

                                                                  1

_2)⋅sin(θ_3 + θ_4 + θ_5) - 0.38⋅cos(θ_2)

 θ_1)⋅cos(θ_3 + θ_4 + θ_5) - 5.02⋅sin(θ_0 + θ_1) - 0.75⋅sin(θ_3 + θ_4)⋅cos(θ_2)⋅cos(θ_0





                                                      0.38⋅sin(θ_2)⋅sin(θ_0 + θ_1)

+ θ_1) - 1.01⋅sin(θ_3 + θ_4 + θ_5)⋅cos(θ_2)⋅cos(θ_0 + θ_1)

                                                      0.38⋅sin(θ_2)⋅cos(θ_0 + θ_1)




 - 0.75⋅sin(θ_3)⋅sin(θ_0 + θ_1)⋅cos(θ_2) - 0.75⋅sin(θ_0 + θ_1)⋅sin(θ_3 + θ_4)⋅cos(θ_2)


 - 0.75⋅sin(θ_3)⋅cos(θ_2)⋅cos(θ_0 + θ_1) - 0.75⋅sin(θ_0 + θ_1)⋅cos(θ_3) - 0.75⋅sin(θ_0
```

```
- 1.01⋅sin(θ_0 + θ_1)⋅sin(θ_3 + θ_4 + θ_5)⋅cos(θ_2) + 0.75⋅cos(θ_3)⋅cos(θ_0 + θ_1) + 0.

                              0

 + θ_1)⋅cos(θ_3 + θ_4) - 1.01⋅sin(θ_0 + θ_1)⋅cos(θ_3 + θ_4 + θ_5) - 5.02⋅sin(θ_0 + θ_1)

                              0

                              1

                              0

75⋅cos(θ_0 + θ_1)⋅cos(θ_3 + θ_4) + 1.01⋅cos(θ_0 + θ_1)⋅cos(θ_3 + θ_4 + θ_5) + 5.02⋅cos(


- 0.75⋅sin(θ_3 + θ_4)⋅cos(θ_2)⋅cos(θ_0 + θ_1) - 1.01⋅sin(θ_3 + θ_4 + θ_5)⋅cos(θ_2)⋅cos(




θ_0 + θ_1)                            -(0.75⋅sin(θ_2)⋅sin(θ_3) + 0.75⋅sin(θ_2)⋅s

         0.38⋅sin(θ_2) + 0.375⋅sin(θ_2 - θ_3) - 0.375⋅sin(θ_2 + θ_3) - 0.375⋅sin(-

θ_0 + θ_1)                            (0.75⋅sin(θ_2)⋅sin(θ_3) + 0.75⋅sin(θ_2)⋅si




in(θ_3 + θ_4) + 1.01⋅sin(θ_2)⋅sin(θ_3 + θ_4 + θ_5) + 0.38⋅cos(θ_2))⋅cos(θ_0 + θ_1)

θ_2 + θ_3 + θ_4) - 0.375⋅sin(θ_2 + θ_3 + θ_4) - 0.505⋅sin(-θ_2 + θ_3 + θ_4 + θ_5) - 0.50

n(θ_3 + θ_4) + 1.01⋅sin(θ_2)⋅sin(θ_3 + θ_4 + θ_5) + 0.38⋅cos(θ_2))⋅sin(θ_0 + θ_1)

          sin(θ_0 + θ_1)

               0

          cos(θ_0 + θ_1)

                         -0.75⋅sin(θ_3)⋅sin(θ_0 + θ_1) - 0.75⋅sin(θ_0 + θ_1)⋅sin(θ

5⋅sin(θ_2 + θ_3 + θ_4 + θ_5)                              -0.375⋅sin(θ_2 - θ_3)

                         -0.75⋅sin(θ_3)⋅cos(θ_0 + θ_1) - 0.75⋅sin(θ_0 + θ_1)⋅cos(θ
```

```
₃ + θ4) - 1.01·sin(θ0 + θ1)·sin(θ3 + θ4 + θ5) + 0.75·cos(θ2)·cos(θ3)·cos(θ0 +
- 0.375·sin(θ2 + θ3) + 0.375·sin(-θ2 + θ3 + θ4) - 0.375·sin(θ2 + θ3 + θ4) + 0.
2)·cos(θ3) - 0.75·sin(θ0 + θ1)·cos(θ2)·cos(θ3 + θ4) - 1.01·sin(θ0 + θ1)·cos(θ2
                                                  -sin(θ2)·cos(θ0 + θ1)
                                                         cos(θ2)
                                                   sin(θ2)·sin(θ0 + θ1)
θ1) + 0.75·cos(θ2)·cos(θ0 + θ1)·cos(θ3 + θ4) + 1.01·cos(θ2)·cos(θ0 + θ1)·cos(θ
505·sin(-θ2 + θ3 + θ4 + θ5) - 0.505·sin(θ2 + θ3 + θ4 + θ5)
)·cos(θ3 + θ4 + θ5) - 0.75·sin(θ3 + θ4)·cos(θ0 + θ1) - 1.01·sin(θ3 + θ4 + θ5)·



3 + θ4 + θ5)  -0.75·sin(θ0 + θ1)·sin(θ3 + θ4) - 1.01·sin(θ0 + θ1)·sin(θ3 + θ4
                                      0.375·sin(-θ2 + θ3 + θ4) - 0.375·sin(θ2 + θ
cos(θ0 + θ1)  -0.75·sin(θ0 + θ1)·cos(θ2)·cos(θ3 + θ4) - 1.01·sin(θ0 + θ1)·cos(



+ θ5) + 0.75·cos(θ2)·cos(θ0 + θ1)·cos(θ3 + θ4) + 1.01·cos(θ2)·cos(θ0 + θ1)·cos
3 + θ4) + 0.505·sin(-θ2 + θ3 + θ4 + θ5) - 0.505·sin(θ2 + θ3 + θ4 + θ5)
θ2)·cos(θ3 + θ4 + θ5) - 0.75·sin(θ3 + θ4)·cos(θ0 + θ1) - 1.01·sin(θ3 + θ4 + θ5
   -sin(θ2)·cos(θ0 + θ1)
          cos(θ2)
    sin(θ2)·sin(θ0 + θ1)
(θ3 + θ4 + θ5)  1.01·(-sin(θ0 + θ1)·sin(θ3 + θ4 + θ5) + cos(θ2)·cos(θ0 + θ1)·c    os(θ3 + θ4 + θ5))|
                               0.505·(sin(-θ2 + θ3 + θ4 + θ5) - sin(θ2 + θ3 + θ4 +    θ5))             |
)·cos(θ0 + θ1)  -1.01·(sin(θ0 + θ1)·cos(θ2)·cos(θ3 + θ4 + θ5) + sin(θ3 + θ4 +     θ5)·cos(θ0 + θ1))|
                                                -sin(θ2)·cos(θ0 + θ1)                               |
                                                       cos(θ2)                                      |
                                                sin(θ2)·sin(θ0 + θ1)                                |
```

Inverse Kinematics Validation

We used similar velocity controls as we did in homework 5. However, due to the extremely limited range of motion of our model, which resulted



The above plots draw a part of a circle (i.e. an arc) as our arm has a limited range of motion due to the goal of replicating anatomical accuracy.

Assumptions

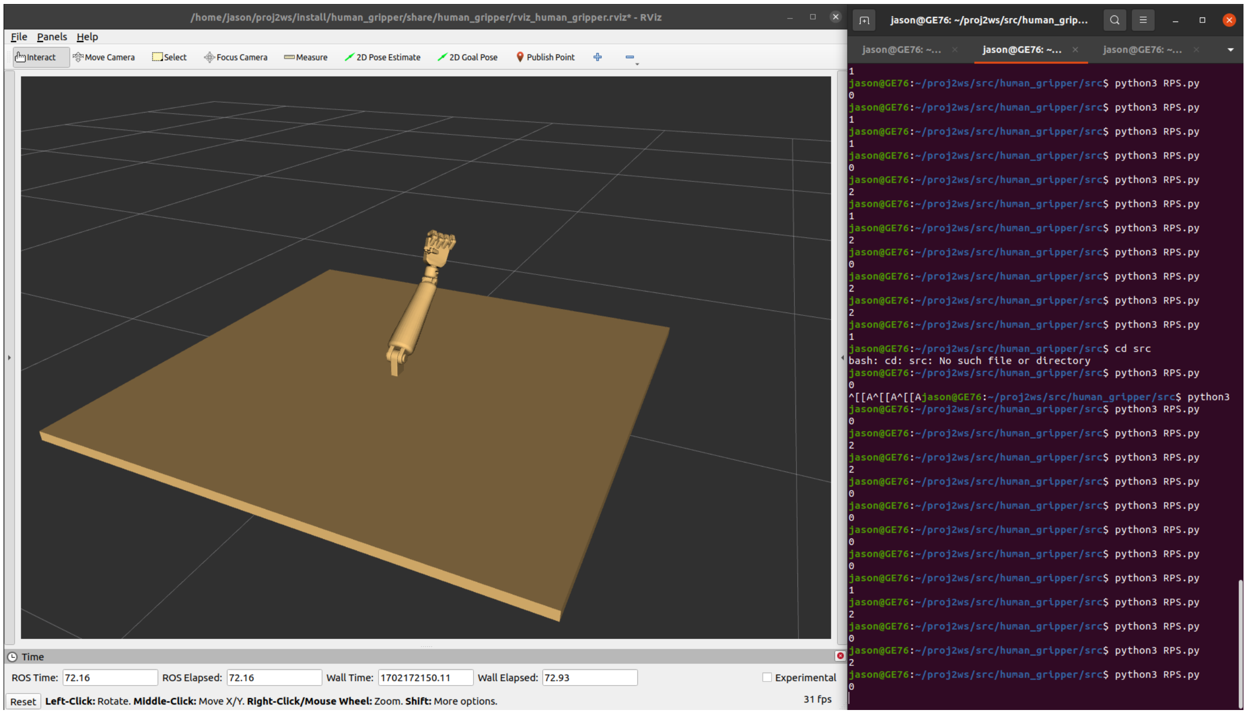
- The rotation of the elbow is just in one direction which is in the same direction as the rotation of finger joints.
- The wrist can be rotated in just 2 directions. This is a fair assumption to be taken since the movement in another direction is negligible in a real human arm.
- While writing the DH table and validating the kinematics we consider all the bigger fingers lie on the same axis. We do consider the offset values while validating the kinematics though.

Control Method

We used a simple modified teleoperation code adapted from the prior projects. Overall, it was simple for us to remap the exact key controls to ones that fit our need to verify that our model could move the way it was intended. We used the simple control code to test the limits of our joints and to make sure every joint moved like how they are supposed to, according to human anatomy (e.g. no finger is going backward). One unique decision that we made was to have the grip relax passively if no signal is sent to it otherwise. This is to mimic the need for us to focus on flexing or tensing up to do so under normal circumstances. Another choice was to operate all the fingers together (in our tele-op controller, at least) and to operate one of the wrist joints along with the elbow joint.

Gazebo and RVIZ Visualization

Here is a screen recording of our code and model in action in the Gazebo environment. Apologies for the static noise in the background. I forgot to select the mute mic option when recording. Regardless, the video shows our arm playing “rock, paper, scissors, shoot!” where it uses a random integer generator from numpy to choose which weapon to throw out (e.g. rock, paper, or scissors). Then it will return to “base” form and is ready for multiple rounds of intense battle! The Gazebo is environment proof as shown in the video link. Here is a second link in case the first one does not work. This one has slightly worse quality. Our project did not require the need for RVIZ visualization as we did not have any sensors to implement. However, we still included an image of RVIZ below.



Problems Faced

The bulleted problems correspond numerically to their respective lessons learned in the next section.

1. We faced significant challenges with the physics properties of our model in Gazebo. Our model was exported more than once after we made supposedly modifications to some of the parts. Even though we took great care to update all the code (there were no code errors) and made sure joints were not affected by our changes, the model developed an issue where it floated in place, clipped into other parts of the model, and was non-responsive to joint controllers.
2. The choice of our joints limited the possible orientations (more specifically, the workspace) that our model can obtain. This resulted in a tough time generating evidence for our inverse kinematics. Our arm does not translate between positions very smoothly.
3. There may have been some issues with our part being too small and light as well, not too sure about this. The inertia values were extremely small in the Xacro file and were a cause for concern when we faced a problem with the physics of our model.
4. We had issues with the Xacro file because we ended up deleting a line or two and the environment was unable to detect the issue when writing the code and only came up during running/compiling.

Lessons Learned

These bullet points correspond to the problems stated in the previous section.

1. We will aim to make sure all changes are finalized before exporting via SW2URDF. The software is finicky after you make changes to a part and re-upload it or if you make it reload with new parts after the initial loading and export.
2. We sacrificed workspace and movement range for the sake of realism. Next time, we need to more deeply weigh the pros and cons and think of how our joints will limit the range of motion. It is rough out here.
3. Add material to our parts within Solidworks before exporting. Maybe consider scaling up if the exact size is not required for calculations.
4. Expat errors usually result from some issue with the Xacro file.

Conclusions

The robot system proposed was successfully implemented. The robot has all the necessary functionalities as a real human hand.

Future Work

- Several extensions can be made for this project depending on the purpose which needs to be satisfied.
- By adding a proper feedback control this can be used as a prosthetic for people who lose their arm/s due to accidents.
- This can also be used as a pick and place robot by increasing the base link and adding some more revolute joints to the base.

References

1. Special thanks to TA Saksham Verma for helping us troubleshoot many small annoying issues.
2. Class homework and lectures were greatly relied upon.
3. Sympy documentation: https://docs.sympy.org/latest/index.html
4. Code adapted from homework and previous projects.