1 Method

1.1 DOF

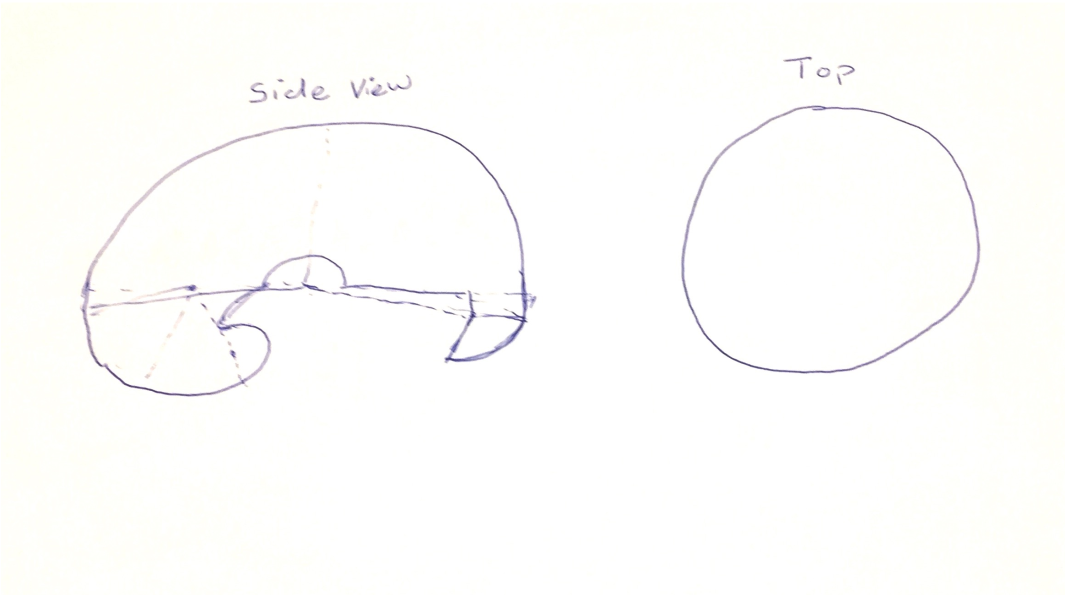
There are 4+1 DOF (4 joints and 1 gripper with prismatic and rotational components). Depending on how you see this you could see this as 6 DOF too.

- Rotational base
- Rotational shoulder
- Rotational elbow
- Rotational Wrist
- Gripper
  - Rotational Component
  - Prismatic Component

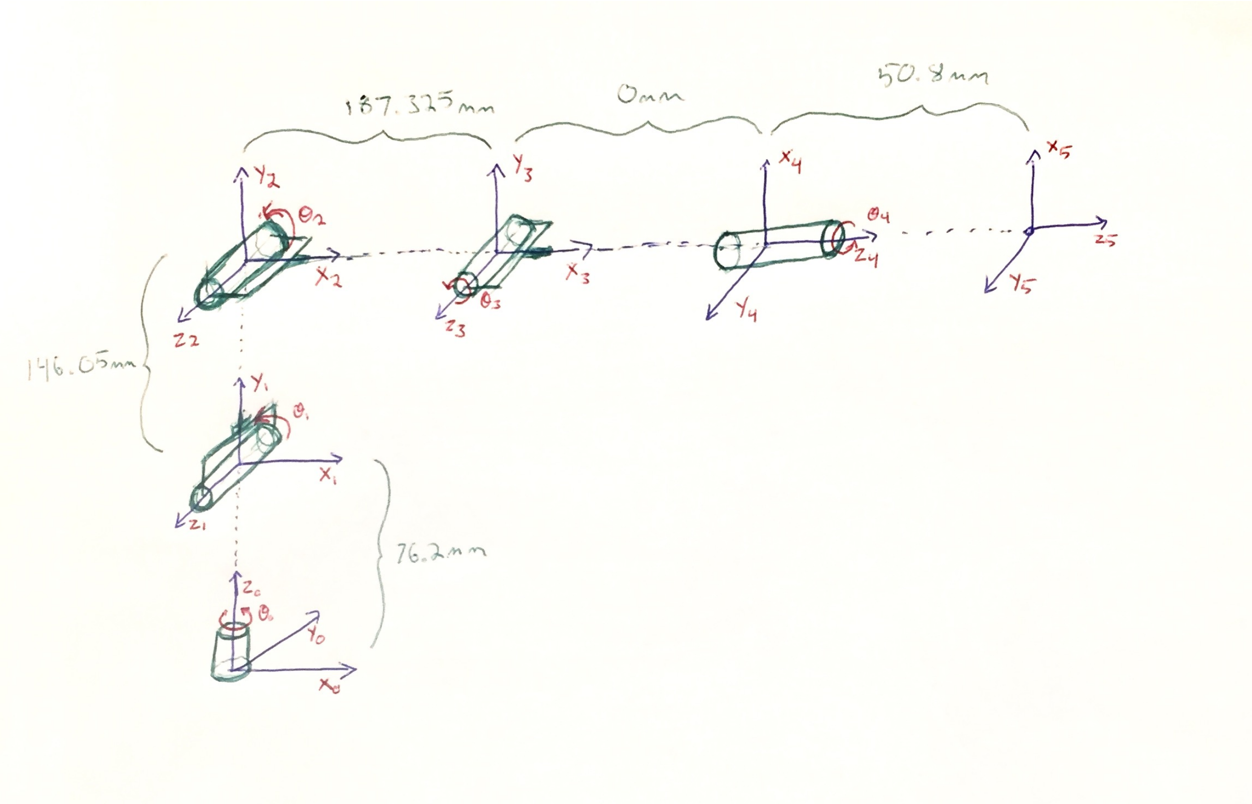
1.2 Kinematic Arrangement

The kinematic arrangement is R R R R R P

1.3 Approximate Workspace



1.4 Drawing



1.5 Homogeneous Transformation Matrices

Using the DH process, we came up with the following table: $DHparameters =$

Link	a_i	α_i	distance	θ_i
Link 1	0	$\pi/2$	76.2	$-\theta_0$
Link 2	146.05mm	0	0	$-\theta_1 + \pi/2$
Link 3	187.325mm	0	0	$-\theta_2 - \pi/2$
Link 4	0	$\pi/2$	0	$-\theta_3 + \pi/2$
Link 5	0	0	50.8mm	θ_4

Note that the reason that there are negative thetas and offsets of $\pi/2$ and $-\pi/2$ in the joint angels is because when we were testing the the robot (Legend), it seemed to be the case that the default position of the robot was wrong. Since we are setting the θs ourselves, this should not affect our final answers.
Then we entered all this data into the matrix for A_i:

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} c_{\alpha_i} & s_{\theta_i} s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c - \theta_i c_{\alpha_i} & -c_{\theta_i} s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

For example for the matrix A_1^0 we derived the following matrix

$$A_1^0 = \begin{bmatrix} c_\theta & 0 & -s_{\theta_1} & 0 \\ -s_{\theta_1} & 0 & -c_{\theta_1} & 0 \\ 0 & 1 & 0 & 76.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Note that respective matrix (for example A_3^2 or A_5^4) would be in the exact same format; all we did was entered the parameters from our DH table into the matrix. For the sake of saving space and paper, we're not being redundant in showing all the tables.

1.6 Position of center of Gripper

The position of the center of gripper in terms of the base frame is:

$$T_e^0 = A_1^0 A_2^1 A_3^2 A_4^3 A_5^4 A_e^5$$

2 Evaluation

2.1 Expectation

In the zero position, we can expect that the x coordinate will be positive, the y coordinate will be zero, and the z coordinate will be positive.

2.2 Homogeneous transformation

When our $q = [0, 0, 0, 0, 0, 0]$, we get the following matrix in the zero position:

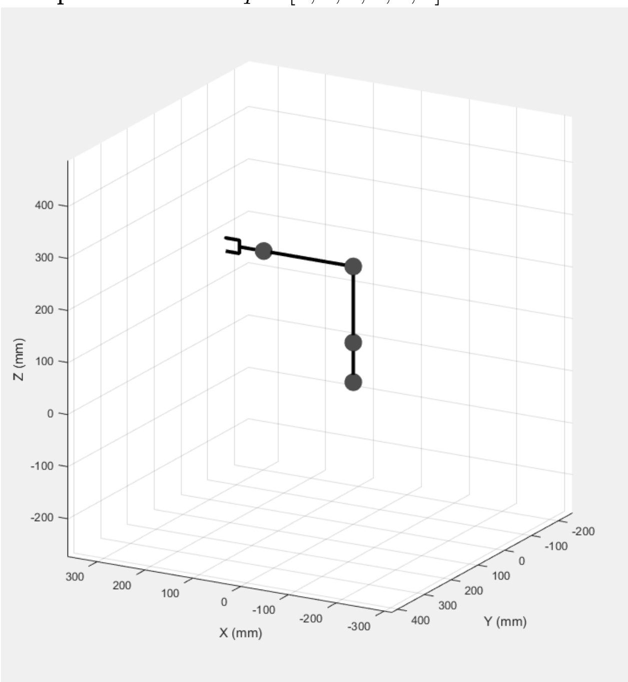
$$A_1^0 = \begin{bmatrix} 0 & 0 & 1 & 238.125 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 222.25 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

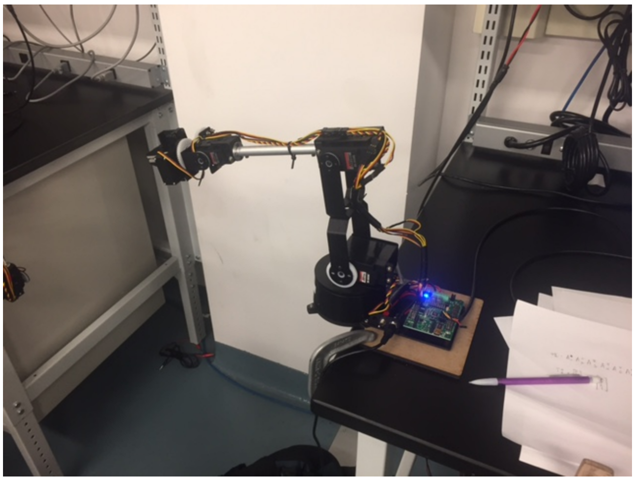
2.3 MATLAB Function and Analysis

We worked extensively to ensure that the simulation looked similar to the physical robot; this is partially the reason we had the offsets that were mentioned in section 1.4.
Nevertheless, here are three example poses:

2.3.1 Pose 1

This pose was when $q = [0, 0, 0, 0, 0, 0]$

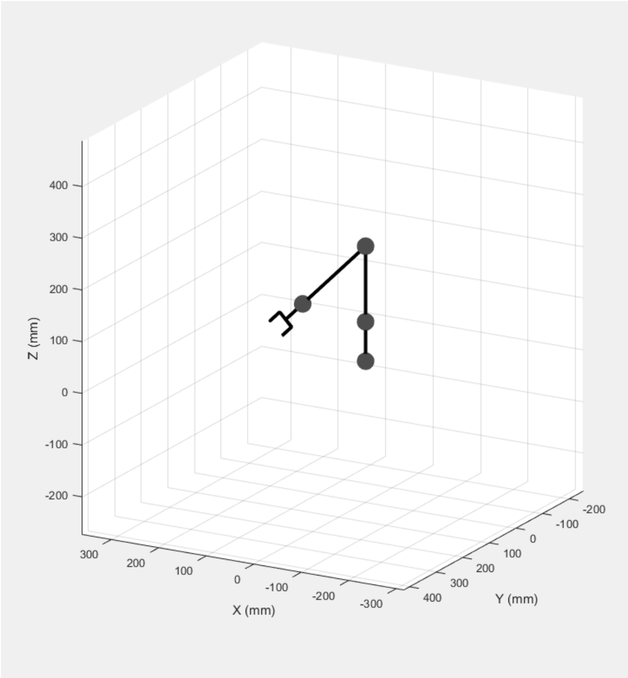


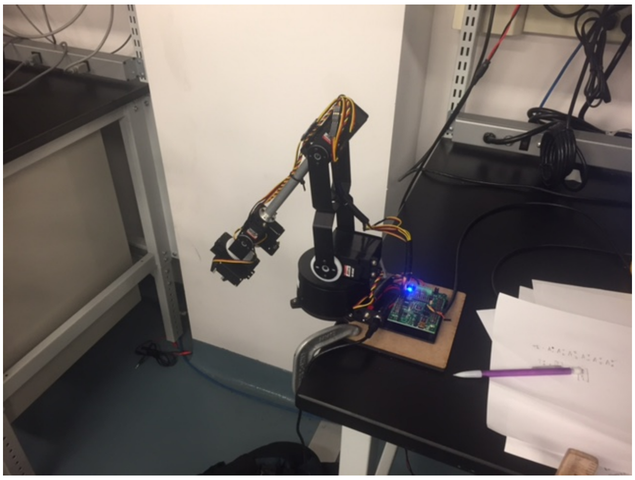


This pose seems fairly similar between the robot and the simulation. There is some difference as the robot seems to be drooping under its own weight.

2.3.2 Pose 2

This pose was when $q = [0, \pi/4, 0, 0, 0, 0]$

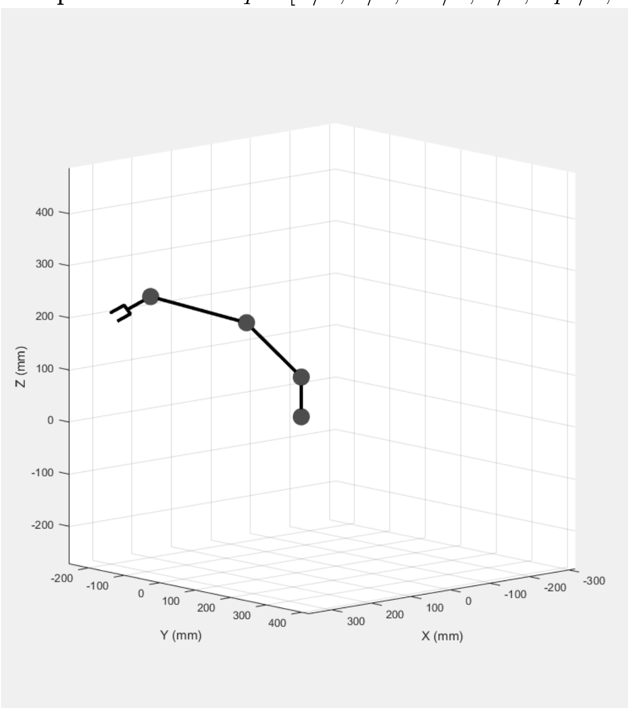


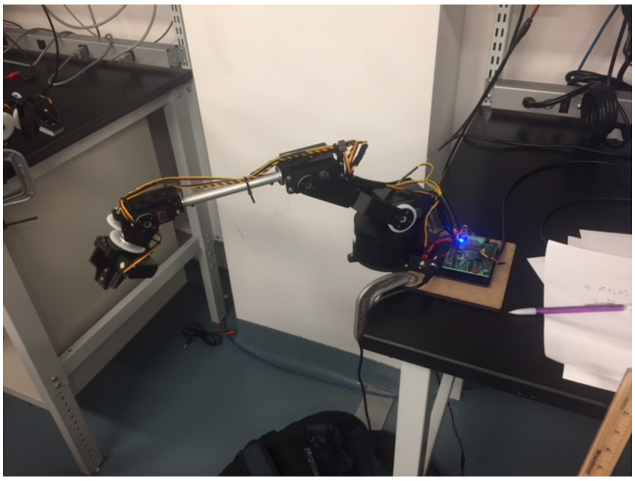


This pose is perhaps the most similar between the simulation and the robot.

2.3.3 Pose 3

This pose was when $q = [\pi/4, \pi/4, -\pi/3, \pi/4, -pi/6, 0]$





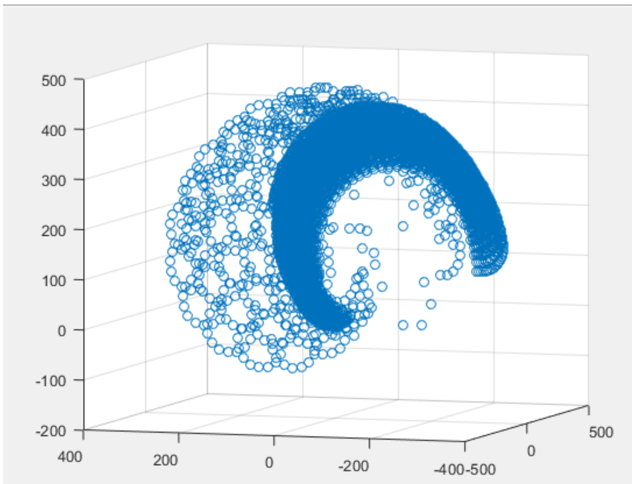
This pose is perhaps the most different between the robot and the simulation. We found this to be because of two main reasons: 1) the robot was drooping under it's weight even more than it was in Pose 1 which made its z positions lower than the simulation and 2) because there was some tension in the wires that seemed to prevent it from going it's full extension.

3 Reachable Workspace

3.1 Method

Here are the limits for each joint we found:

- Joint 1: -1.4:1.4
- Joint 2: -1.2:1.4
- Joint 3: -1.8:1.7
- Joint 4: -1.9:1.7
- Joint 5: -2:1.5
- Joint 6(Gripper): -15:30



Our general strategy was to use our UpdateQ function and give it every possible parameter possible (using the limits above). Then we used the X matrix that updateQ returns and we plot the x,y,z positions onto a 3D Scatterplot.
In particular, we used a series of 6 nested for loops, one for each joint; each joint went from the lower limit to the upper limit of each particular joint. In the inner-most loop, we set these parameters into q and sent it to updateQ which returned an X matrix which then we plotted.
We realize that this isn't the most space nor time efficient solution to this problem however, it provided a reasonable solution to have a ballpark estimate in a short amount of time for prototyping.

3.2 Evaluate

Yes there points that the simulation could reach but the actual robot could not. This is for a few reasons:

1. Wire entanglement
2. Wire tension and lack of wire length
3. Table obstruction preventing the robot from exploring most of the -x side of the Cartesian plane.
4. Clamp obstruction
5. Wires getting unplugged

3.3 Analysis

The main information we would need to incorporate into our model would be the physical table, the fact that there are physical wires that are a limiting factor, and the fact that the robot is at a height and so there are clamps to prevent it from falling down (that also obstruct it's full range of motion).
We would also need to incorporate some kind of basic physics into our simulation since right now we have assumed that each of the joints are massless objects that are not being forced down by gravity; this however is not true and as we saw earlier, our robots were consistently sagging in space under its own weight. We might also want to incorporate some $\mu_{staticfriction}$ because we saw that the robot base was moving as the joints moved (and a higher coefficient of friction could help model this).