# HW5

Will Spaeth

April 17, 2024

## Problem 1-1

1. Both controllers use computational torque control in Cartesian space. The Cartesian force is mapped to joint torque with Jacobian transpose control.

   The servo control law $F'$ is a combination of velocity error, position error, and integral error. I initially included reference acceleration, but the trajectories were not smooth and generated huge acceleration errors, creating control instability.

$$
\begin{aligned}
F' &= K_v\dot{e} + K_p e + K_i \int e dt \\
\tau &= J^\top F \\
&= J^\top M_X(q)F' + J^\top h_X(q,\dot{q}) + J^\top g_X(q) \\
&= J^\top J^{-\top} M(q) J^{-1} F' + J^\top J^{-\top}(h(q,\dot{q}) - M(q)J^{-1}\dot{J}\dot{q}) + J^\top J^{-\top} g(q) \\
&= M(q)J^{-1}F' + h(q,\dot{q}) - M(q)J^{-1}\dot{J}\dot{q} + g(q) \\
&= M(q)J^{-1}(K_v \odot \dot{e} + K_p \odot e + K_i \odot \int e dt) + h(q,\dot{q}) - M(q)J^{-1}\dot{J}\dot{q} + g(q)
\end{aligned}
$$

   The values $M(q)$, $h(q,\dot{q})$, and $g(q)$ were all found using the symbolic equation script from HW4. I modified the parameters to work for the 3R robot, then copied the output equations into functions that could be called within Simulink. This script is called dynamics_3Rrobot.m. The equations for each term are too large to put in this assignment, but can be found within the scripts CalculateMassTerm.m, CalculateCoriolisTerm.m, and CalculateGravityTerm.m.

   As mentioned, the reference trajectories are not smooth and the velocities and accelerations are discontinuous. I ignored the reference acceleration, but clipped the error velocities to be within [-10, 10] to avoid massive velocity error spikes.

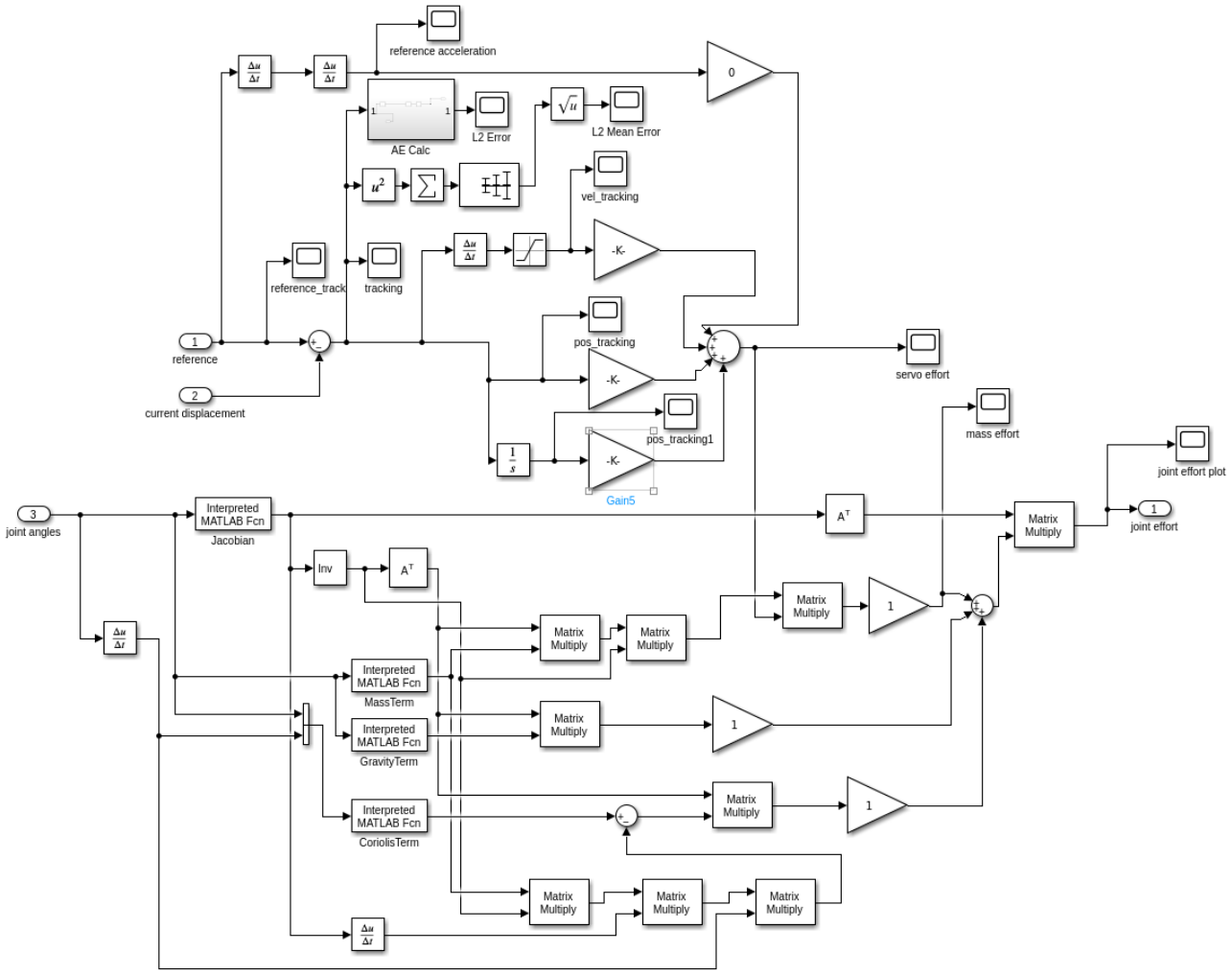
Values of gains for problem 1:

$$K_v = \begin{bmatrix} 60 \\ 60 \\ -60 \end{bmatrix}$$

$$K_p = \begin{bmatrix} 13000 \\ 13000 \\ -13000 \end{bmatrix}$$

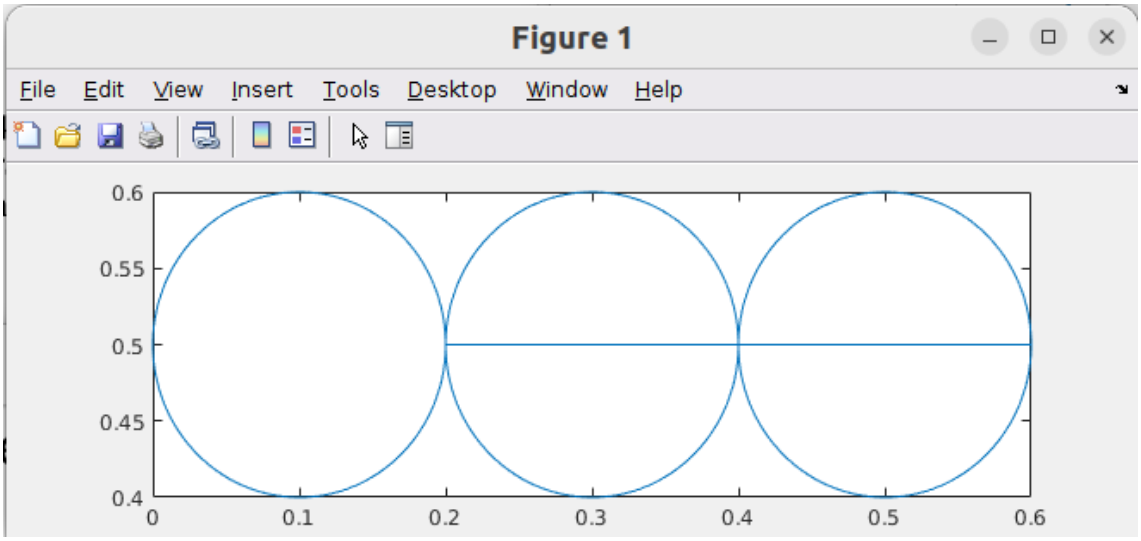
$$K_i = \begin{bmatrix} 130 \\ 130 \\ -130 \end{bmatrix}$$

The block diagram is below. The top portion of the diagram is the servo law, while the bottom portion is the computational torque control.

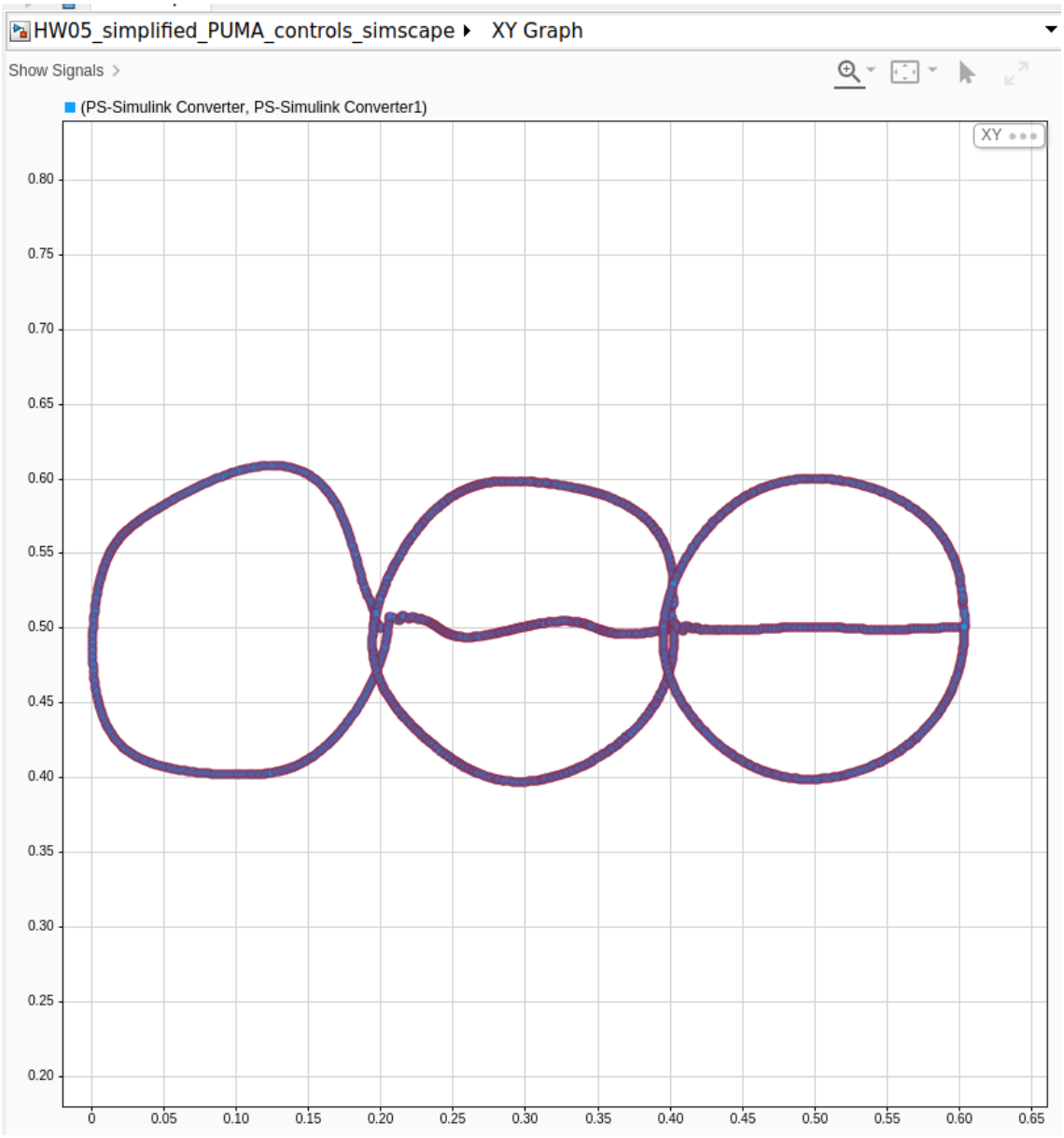



Computational torque controller block diagram

2. Trajectory Plots

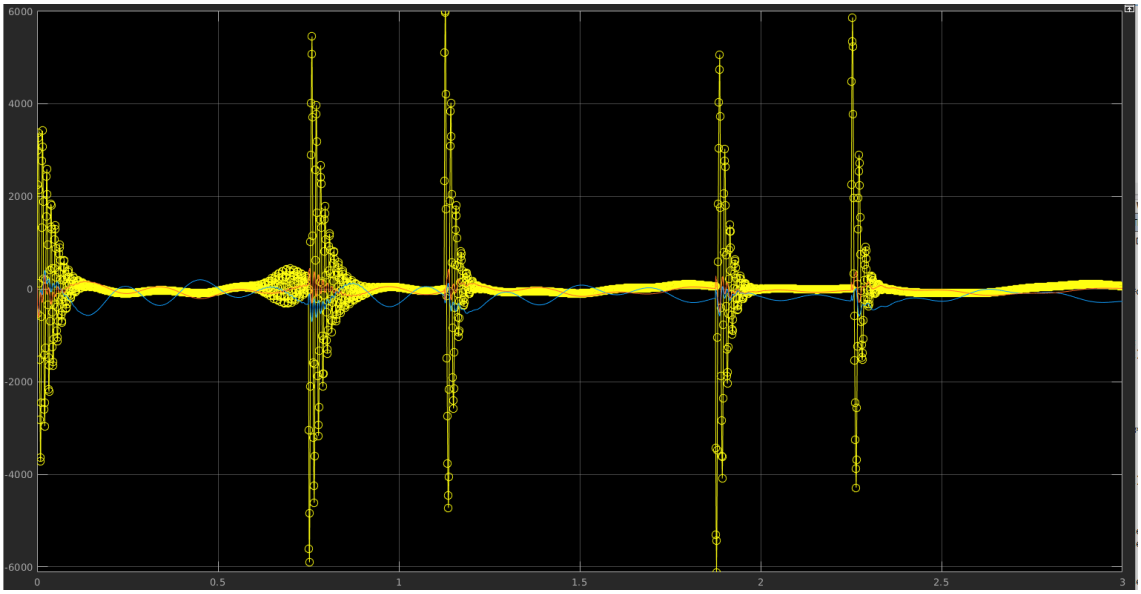



Reference plot



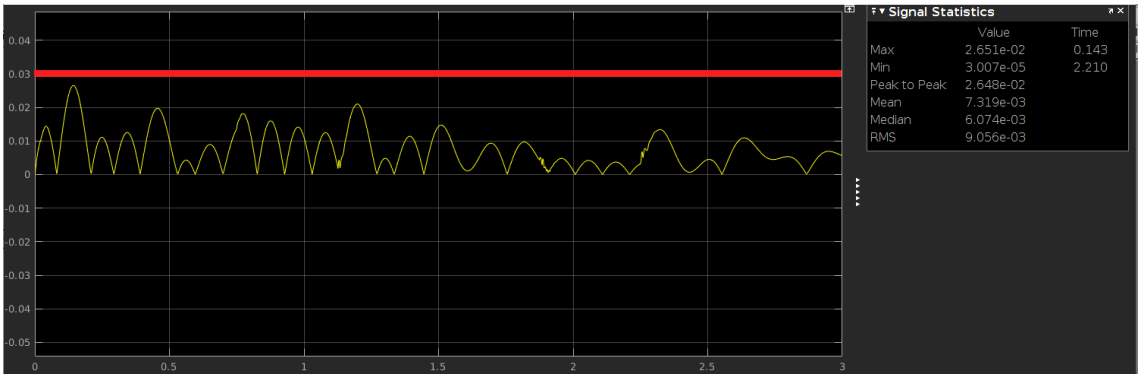

Actual plot

3. Joint Torques

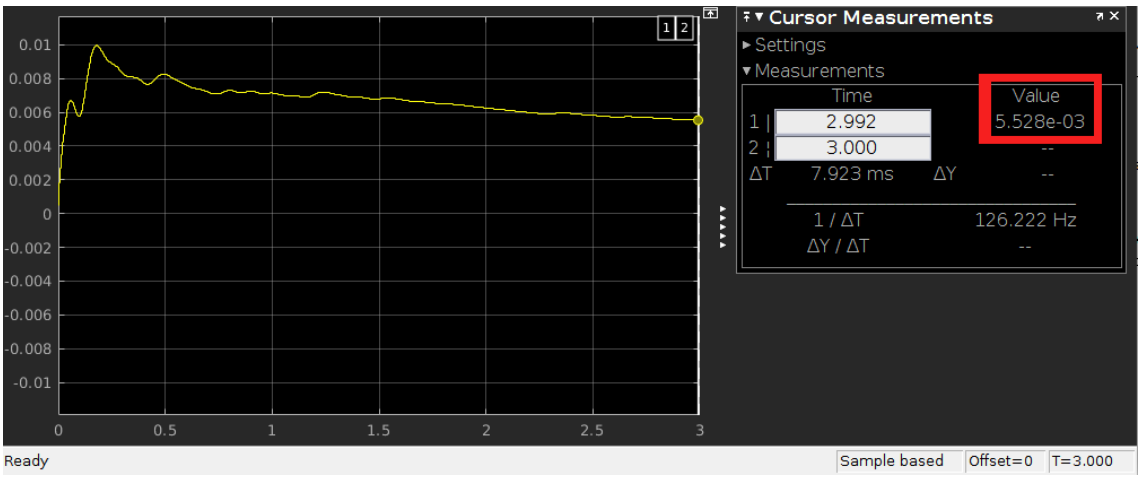



Joint torques

4. End Effector Error Plots




Instantaneous error




RMSE

5. Video found within submission as "P1-1 Motion Video".

# Problem 1-2

1. This controller is the same as the previous controller, except with clipped error velocities within [-0.5, 0.5] and the following gains,

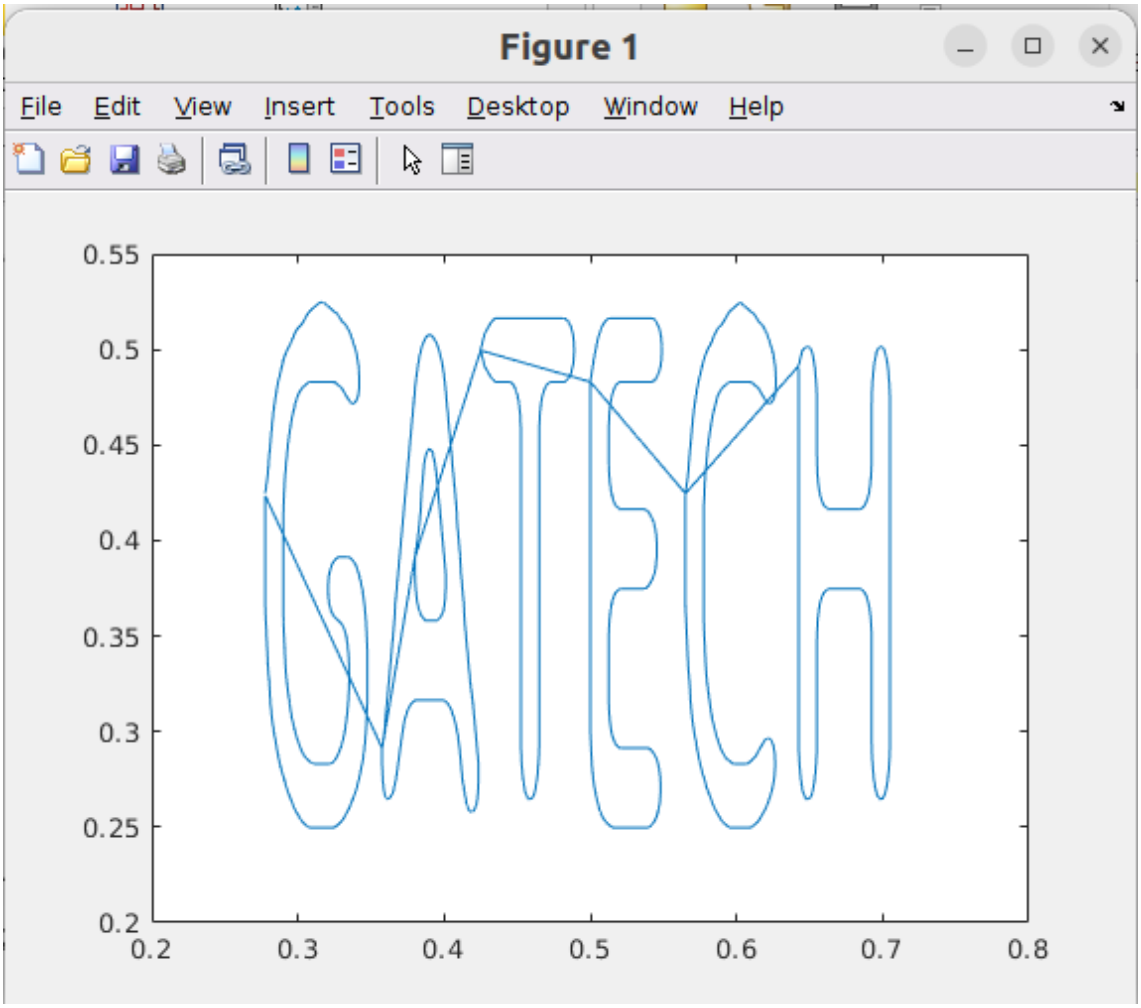
   Values of gains for problem 2:

$$K_v = \begin{bmatrix} 250 \\ 250 \\ -250 \end{bmatrix}$$

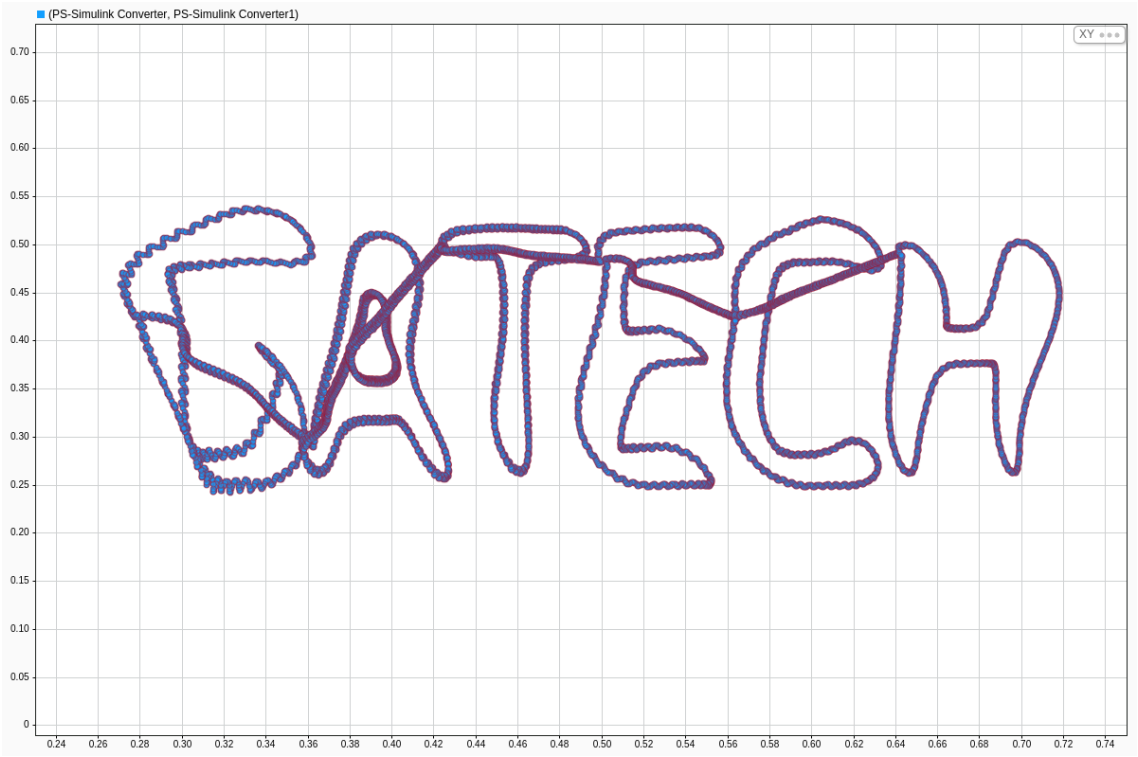
$$K_p = \begin{bmatrix} 8000 \\ 8000 \\ -8000 \end{bmatrix}$$

$$K_i = \begin{bmatrix} 1000 \\ 1000 \\ -1000 \end{bmatrix}$$

2. Trajectory Plots

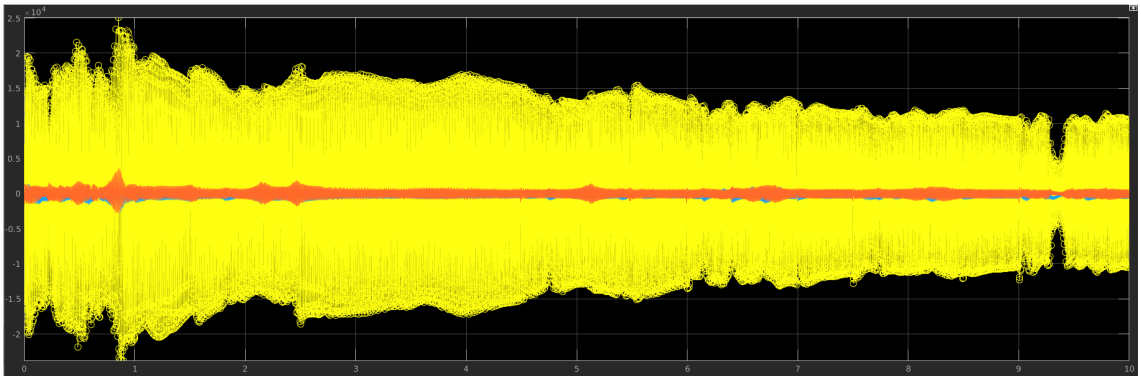



Reference plot



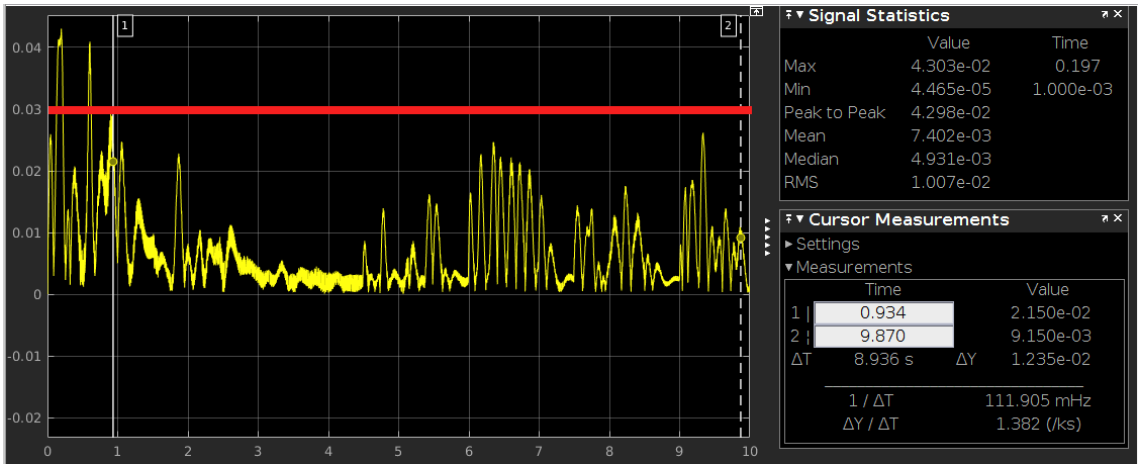

Actual plot

3. Joint Torques

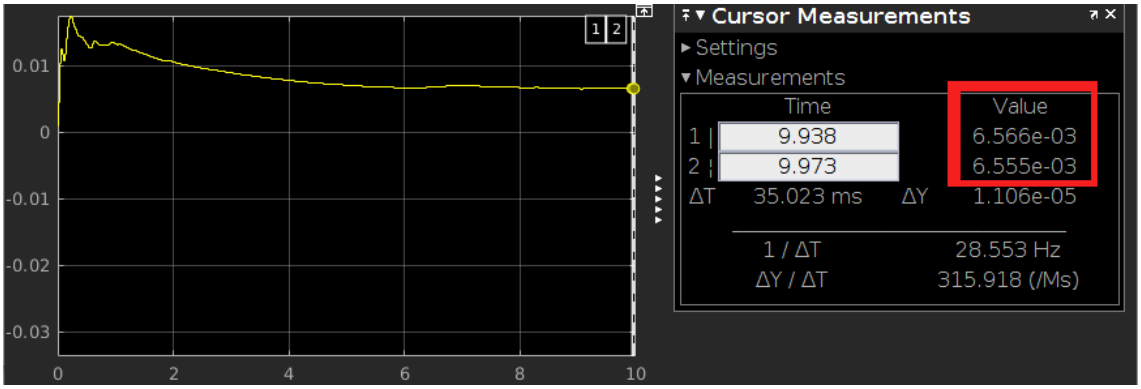


Joint torques

4. End Effector Error Plots




Instantaneous error




RMSE

5. Video found within submission as "P1-2 Motion Video".

# Problem 1-Bonus

(a) I implemented computational torque control for P1-1 and P1-2. This can be verified through the block diagram, as well as the files "dynamics_3Rrobot.m", "CalculateMassTerm.m", "CalculcateCoriolisTerm.m", and "CalculcateGravityTerm.m".

(b) I completed P1-1 in 3 seconds under the constraints, which should qualify for this question.

# Problem 2

1. I used a grid search optimization to find the redundant joint angles. I used the following loss function $L$ to evaluate each joint position,

$$
\begin{aligned}
q &:= \text{joint angles} \\
p(q) &:= \text{forward kinematics} \\
k &= 1 \\
L_p &= \sum (p_{ref} - p(q))^2 \\
L_w &= \sqrt{\det(JJ^\top)} \\
L &= L_p - k \cdot L_w
\end{aligned}
$$

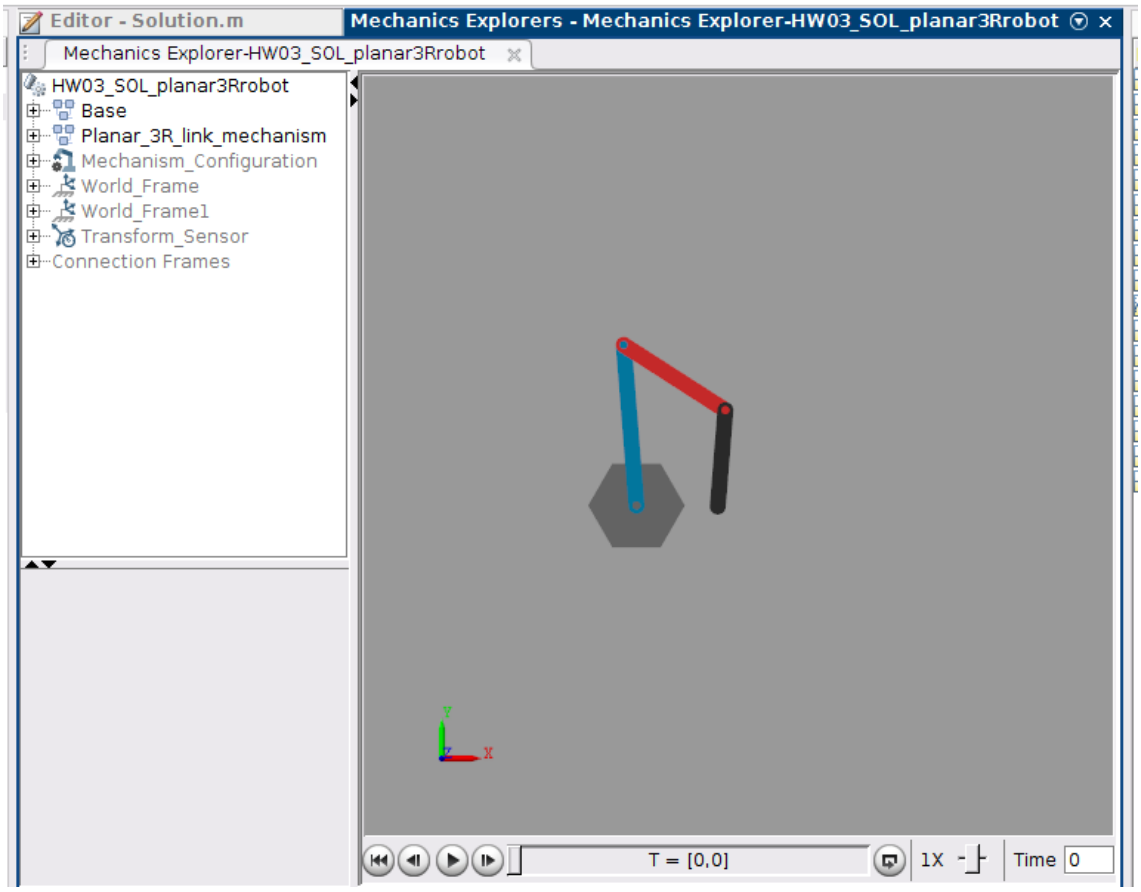
The goal is to minimize this loss function with respect to the joint angles, which will both minimize the end effector error and maximize manipulability.

For the grid, I sampled each joint angle 100 times between 0 and $2\pi$ and evaluated the full combination of $100^3 = 1000000$ joint angles. I took the minimum from all of these options. The parameter space is small enough that it is solvable within 2 seconds.

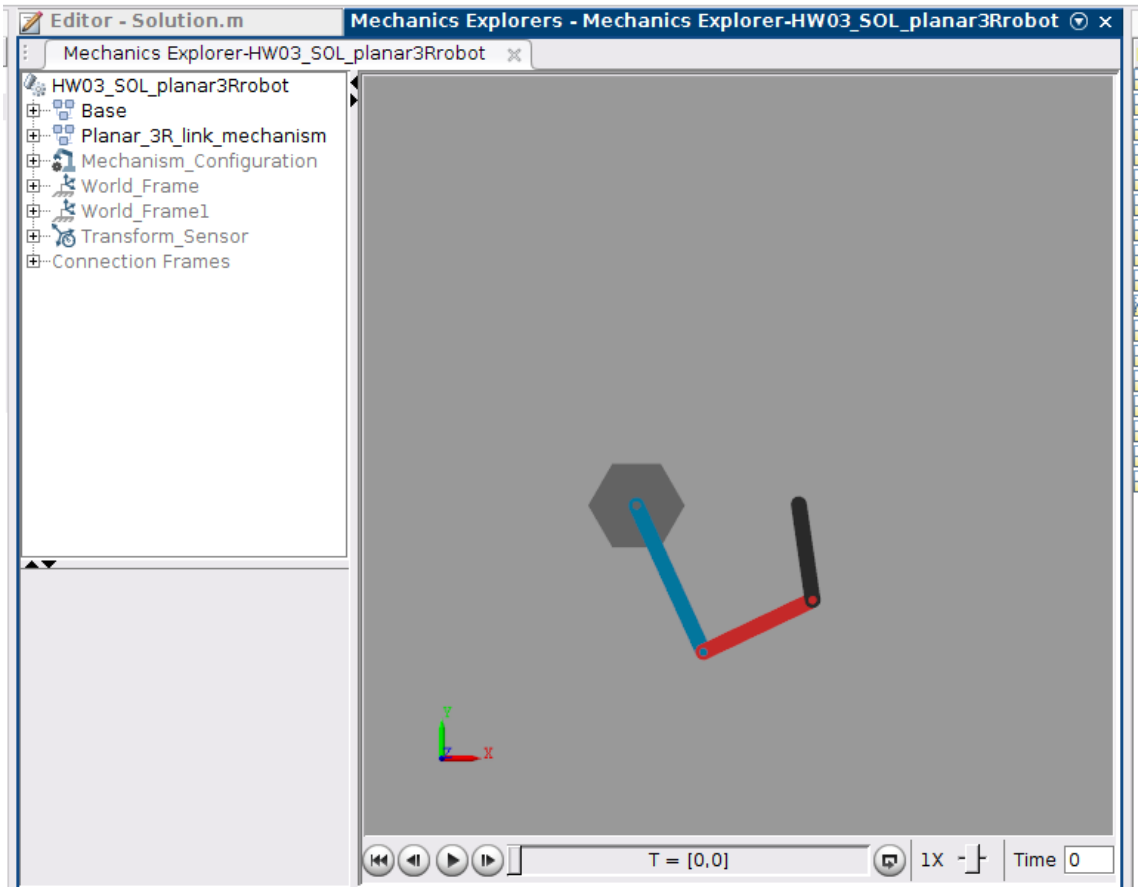
2.

| Position (x,y) | Joint 1 (deg) | Joint 2 (deg) | Joint 3 (deg) | Manipulability w |
|---|---|---|---|---|
| (0.1,0) | 1.6501 | 4.0619 | 5.2043 | 0.0283 |
| (0.2,0) | 5.1408 | 1.5867 | 1.2693 | 0.0471 |
| (0.3,0) | 0.8885 | 5.1408 | 5.2043 | 0.0602 |
| (0.4,0) | 0.6347 | 5.3312 | 5.9659 | 0.0556 |
| (0.45,0) | 0.3173 | 5.9024 | 5.9024 | 0.0368 |

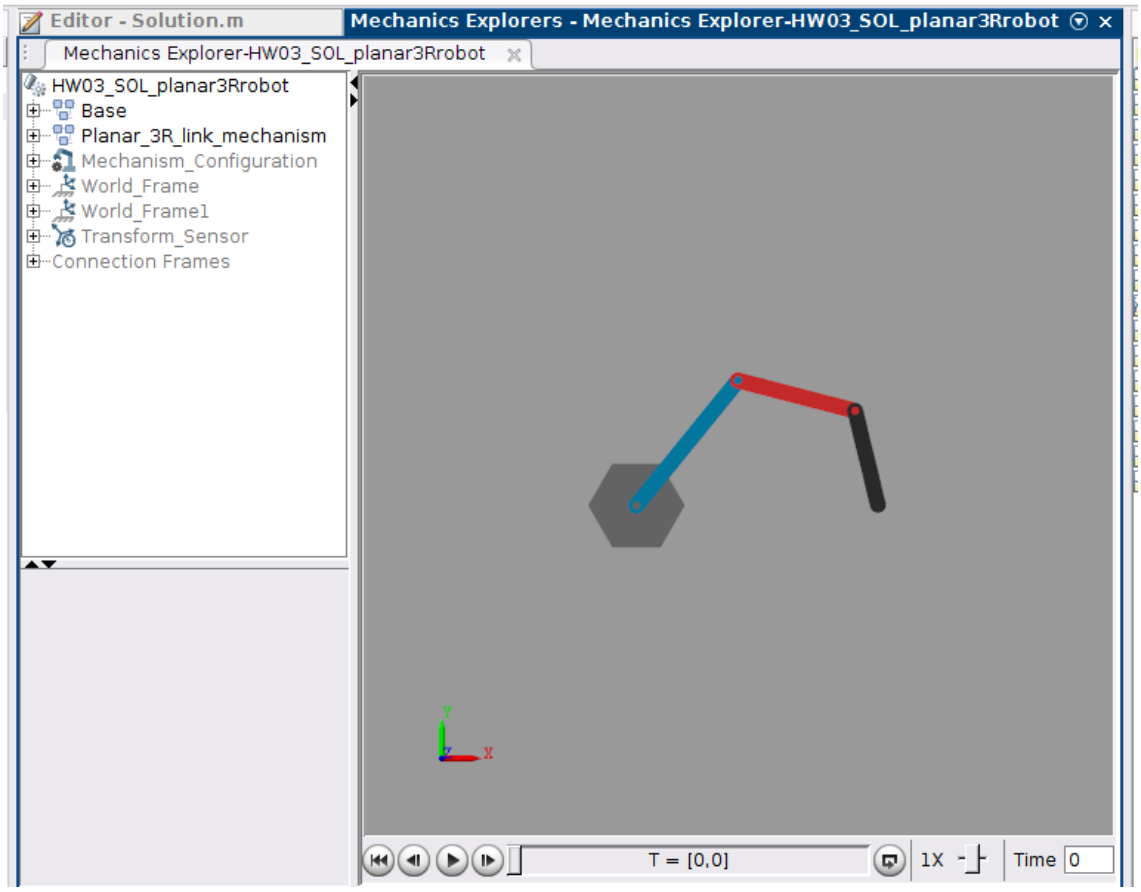
3.



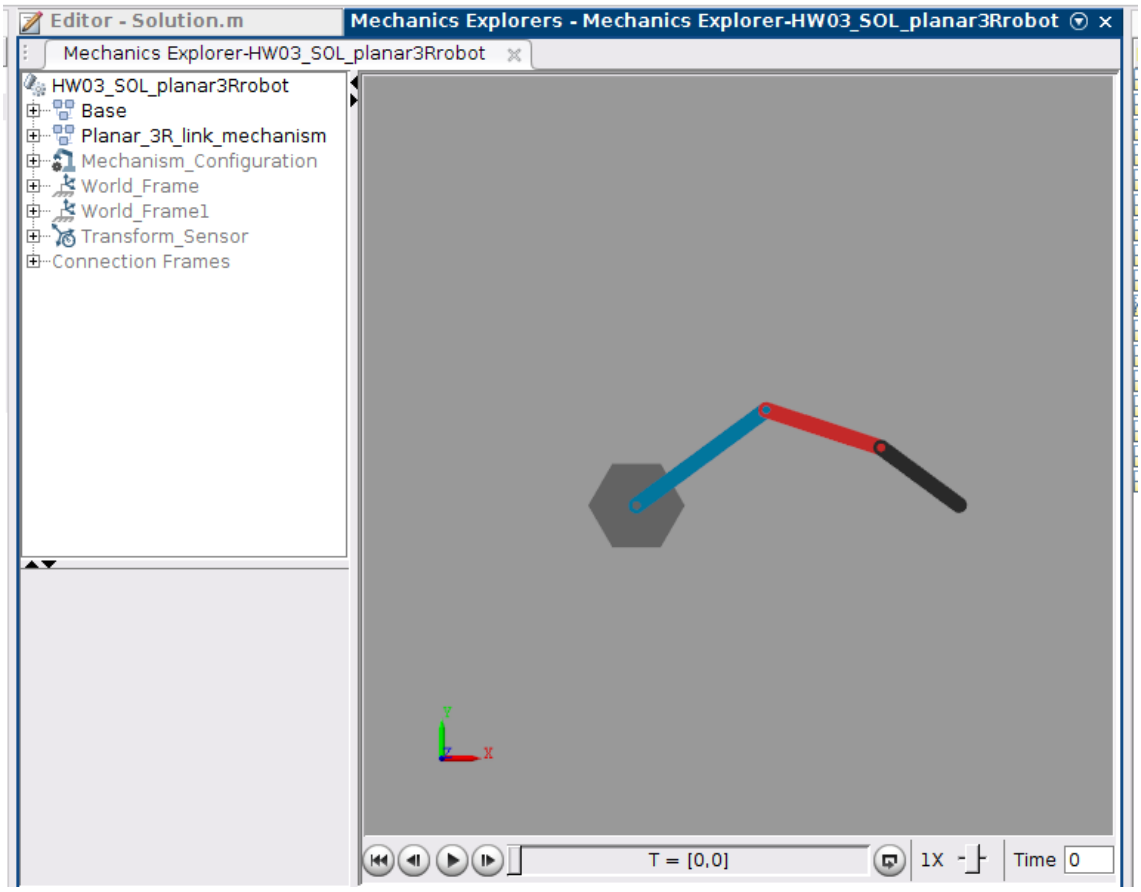

Position (0.1, 0)



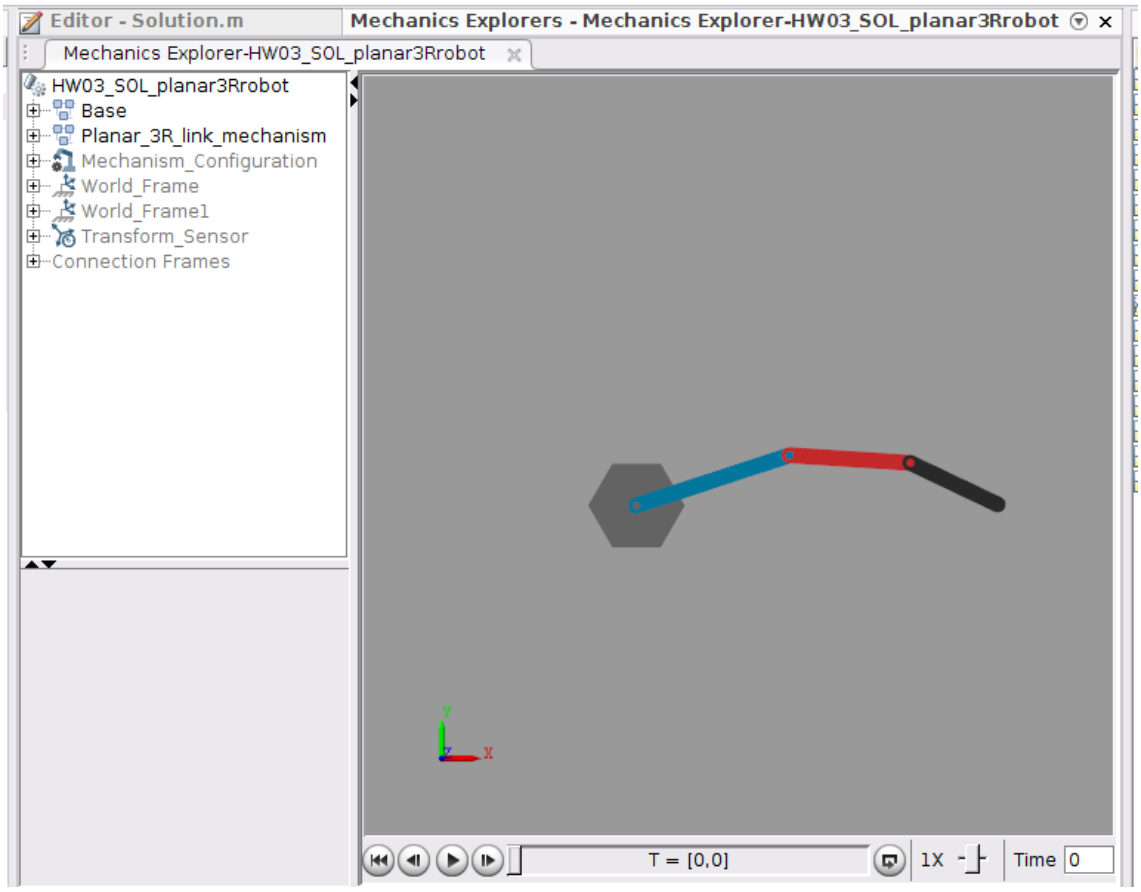

Position (0.2, 0)




Position (0.3, 0)




Position (0.4, 0)




Position (0.45, 0)