# MEGN544 - Project Part 5 Report

Student: Thong Quoc (Bill) Huynh

December 12, 2023

## Introduction

The theme of the project in our class over the semester has been to implement a controller to trace a trajectory resembling the CSM letters in the most accurate way and robust to geometry imperfections. In Part 3, we have utilized a theta-only open-loop controller. Then in Part 4, we used operational space rates feedback and velocity Jacobian inverse to calculate joint rates control signals. In Part 5, we strive for a controller with even higher complexity and better performance.

## Approach

**The modeling approach** in Project Part 5 is to implement an operational space inverse dynamics controller. The errors (linear pose, angular pose, linear velocity, and angular velocity errors of the end effector) are calculated in the operational space. The operational-space errors provide feedback directly in the pose and velocity of the end effector. The inverse dynamics calculation using Newton Euler function serves to find the torque control signal.

Similar to the previous Part 4, way points in the trajectory are created using vectors combining the target 3D positions and the target rotations (encoded in unit quaternions), based on the 3D points of the CSM path. Constant acceleration interpolation is used to interpolate between the way points to generate "position" vectors that include interpolated 3D positions ($p$) and interpolated quaternions ($Q$), "velocity" vectors that include the interpolated rates - linear velocities ($v$) and quaternion rates ($\dot{Q}$), and similarly "acceleration" vectors - linear accelerations ($a$) and quaternion accelerations ($\ddot{Q}$). We set:

$$\begin{aligned}\begin{bmatrix} a \\ \alpha \end{bmatrix}_{control} &= \begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} \\ &\rightarrow \ddot{e} + K_p e + K_d \dot{e} = 0\end{aligned}$$

where $\ddot{e}$ is the error in operational-space acceleration vector (with $a$ and $\alpha$), $\dot{e}$ is the error in operational-space velocity vector (with $v$ and $\omega$), and $e$ is the error in operational-space position vector (with $p$ and $R$). Angular velocity ($\omega$) and angular acceleration ($\alpha$) can be calculated from, respectively, quaternion rate ($\dot{Q}$) and quaternion acceleration ($\ddot{Q}$) using the appropriate conversion equations.

It can be proven that the error goes to zero at steady state. We use the control law:

$$\begin{aligned}\tau_m &= M(\ddot{\theta}_c) + G + C + J_v^T W_e \\ &= M J_v^{-1}\left(\begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} - \dot{J}_v \dot{\theta}\right) + G + C + J_v^T W_e\end{aligned}$$

We also have the closed loop differential equation:

$$\begin{aligned}
\ddot{\theta} &= M^{-1}(-C - G - J_v^T W_e + \tau_m) \\
&= M^{-1}(-C - G - J_v^T W_e + M J_v^{-1}(\begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} - \dot{J}_v \dot{\theta}) + G + C + J_v^T W_e) \\
&= M^{-1}(\delta + M J_v^{-1}(\begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} - \dot{J}_v \dot{\theta}) \\
&= J_v^{-1}(\begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} - \dot{J}_v \dot{\theta}) \\
\rightarrow J_v \ddot{\theta} + J_v \dot{\theta} &= \begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} \\
\rightarrow \begin{bmatrix} a \\ \alpha \end{bmatrix}_{actual} &= \begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} \\
\xrightarrow{SS} 0 &= \ddot{e} + K_d \dot{e} + K_p e \\
\rightarrow 0 &= \ddot{x} + b\dot{x} + ax \rightarrow
\end{aligned}$$

It turns out:

$$\begin{aligned}
K_d &= 2\zeta\omega_n \quad (equation\ 1) \\
K_p &= \omega_n^2 \quad (equation\ 2) \\
\tau_m &= M J_v^{-1}(\begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} - \dot{J}_v \dot{\theta}) + G + C + J_v^T W_e \quad (equation\ 3) \\
\ddot{\theta}_c &= J_v^{-1}(\begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} - \dot{J}_v \dot{\theta}) \quad (equation\ 4) \\
\begin{bmatrix} a \\ \alpha \end{bmatrix}_c &= \begin{bmatrix} a \\ \alpha \end{bmatrix}_d + K_p e + K_d \dot{e} \quad (equation\ 5)
\end{aligned}$$

As seen in the Simulink block diagram Figure 1, the errors with $K_p$ and $K_d$ gains are summed with the accel-des vector to generator the control acceleration vector as in equation 5 above. This control acceleration vector is then subtracted by $\dot{J}_v \dot{\theta}$ and multiplied with $J_v^{-1}$ to generation the $\ddot{\theta}_c$ in equation 4 above. This control theta acceleration is then used together with theta actual and theta dot actual to feed into the Newton Euler function for inverse dynamics calculations, generating the control torque signal.

The separate block on the bottom right of Figure 1 shows that pose error to be shown in graphs is always calculated against laser pose (ground truth), whether theta feedback or laser feedback is used for pose trans-error calculation.

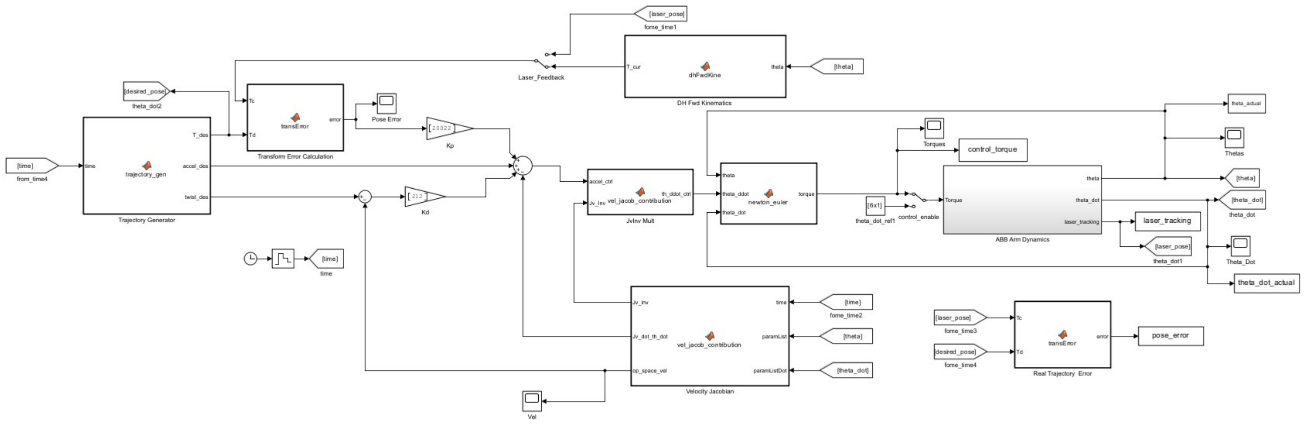

Figure 1: Screenshot of top-level block diagram used in MATLAB Simulink.

## Gain Calculation

For **control gains**, I choose a maximum overshoot (OS) of 5%, which gives me a minimum $\zeta$ of 0.6901 from the corresponding equation below. To have a safety factor, I choose specifically $\zeta = 0.75$. Then, I gradually reduce the rise time $t_r$ and setting time $t_s$ to reduce the error in the tracing of the CSM while making sure the torque control signal does not have significant chatter. Table 1 shows my process of tuning the rise time and settling time and the corresponding root-mean-square (RMS) of errors and all six joints' control torque signals.

$$\zeta \geq \sqrt{\frac{ln^2(OS)}{\pi^2 + ln^2(OS)}}$$
$$\omega_n \geq \frac{1.53 + 2.31\zeta^2}{t_r}$$
$$\omega_n \geq \frac{-ln(0.01)}{\zeta t_s}$$

Table 1: Table showing the process of tuning gains by changing rise time and settling times. For this process, sim-exact is set to false, sim-time is set to 10 seconds, and laser-feedback is set to false

| $\zeta$ | $t_r$ (s) | $t_s$ (s) | $K_d$ | $K_p$ | RMS pos. pose error | RMS ang error | RMS control torque (combined) (N) |
|---|---|---|---|---|---|---|---|
| 0.75 | 0.2 | 1 | 21 | 200 | 0.0169 | 0.5341 | 91.27 |
| 0.75 | 0.1 | 0.5 | 40 | 692 | 0.0068 | 0.2189 | 93.18 |
| 0.75 | 0.09 | 0.25 | 47 | 988 | 0.0053 | 0.1656 | 95.13 |
| 0.75 | 0.02 | 0.08 | **212** | **20022** | 0.0023 | 0.0164 | 451.2695 |

Eventually, I pick $\zeta$ of 0.75, $t_r$ of 0.02, $t_s$ of 0.08, resulting in $K_d$ of 212 and $K_p$ of 20022 because at these gain values, the errors are small error for a result CSM path that matches the goal CSM trajectory closely enough (visually), with reduced RMS errors compared to other table rows. I stop reducing the rise time and settling time any further because the RMS control torque is already higher than the other table rows. Although the RMS control torque seems high at the chosen gains, the control torque plot (as seen in graphs below) does not show significant chattering overall, except for initial ramp up period that contributes to the higher RMS control torque. The spikes could be because the gain $K_p$ of 20022 is technically higher than the control update rate of 1000 Hz.

## Result

The **result performances** of the controller in tracing the CSM trajectory are shown in plots below, from Figure 2 to Figure 17, with 4 cases stemming from 2 options - option of sim-exact (perfect plant) or not, and option of laser feedback or theta feedback. We can see the following points from the result performances.

Firstly, all CSM trajectories traced by 4 cases are smooth with no visible oscillation from the actual motion path. The paths are not as straight at the very beginning because the controller is trying hard to keep up with the ramp up and may try to correct the errors aggressively and the backtrack.

Secondly, the actual theta plots for all 4 cases have no visible difference. They all show generally smooth theta paths with no discontinuity, except for $\theta_6$ path which shows some significant variations in angle value over time due to it being the wrist angle which needs to change drastically as the local coordinate frame continuously orients the x-axis towards each next point in the CSM trajectory.

While the angular pose error plots for all 4 cases do not have significant differences, the position pose error plot for the case with theta (no laser) feedback and non-perfect plant is the most different from all the other cases. It shows slower changes over wider periods of time. This could be due to the lack of accurate tracking resulting from imperfect knowledge of the arm geometry and imperfect feedback (theta feedback instead of laser pose feedback), making it more difficult for the arm to keep up quick enough with the trajectory waypoints. The non-perfect case with laser tracking still shows a different position pose error from those for perfect plant cases, but the differences are not as large as the theta feeback non-perfect plant case. Possibly, the laser tracking gives the controller more **robustness** in geometry inaccuracies. Even if a link length is in accurate, the laser feedback can provide the controller with the necessary error to correct and compensate for the inaccuracy.

The graphs of control torque are similar for all 4 cases, showing the requested torques staying below the saturation values, except for the initial control chatter period of roughly 0.5 seconds. The ramp-up period to get from initial pose to the initial point of the letter ”C” has a large position difference to cover, which is probably why the controller shows a lot of chatter (trying to keep up) at the beginning. There are small spikes of torque signal values throughout the simulation time. This may be because the control torque signal coming into the theta controller has a direct effect, a direct value for the theta controller to get to, making the actual torque control signal looking like little step functions in small periods of time.

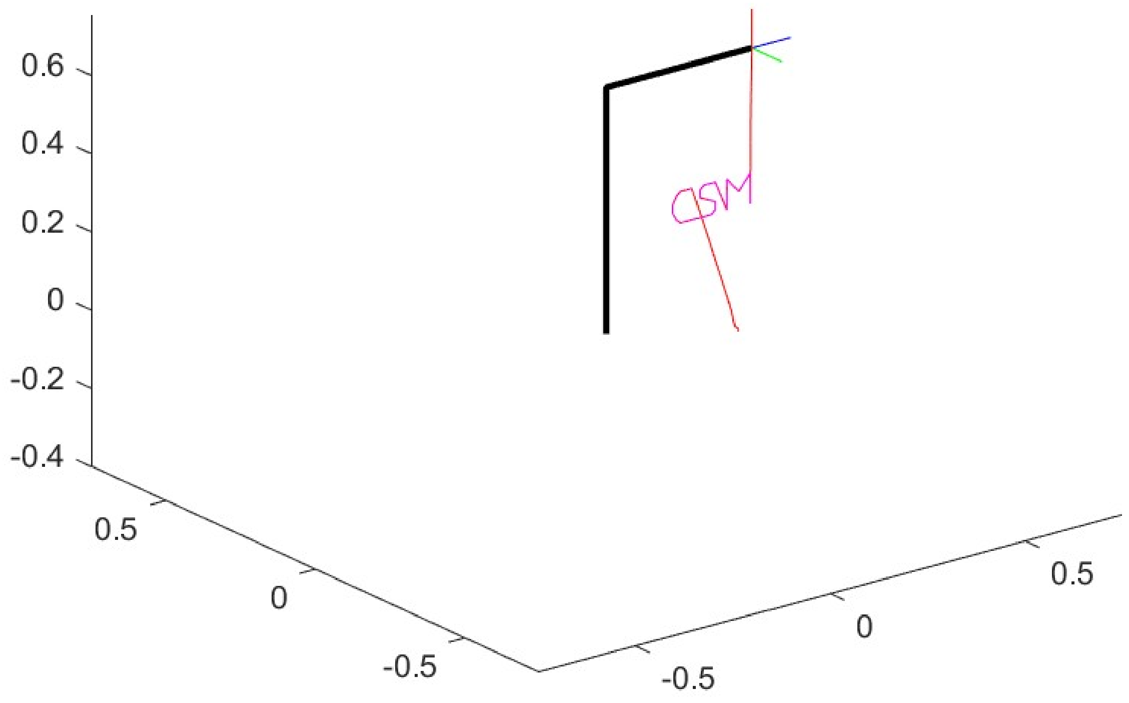

Figure 2: Result trajectory traced by robot arm, with no laser tracking, perfect plant (sim-exact = true).

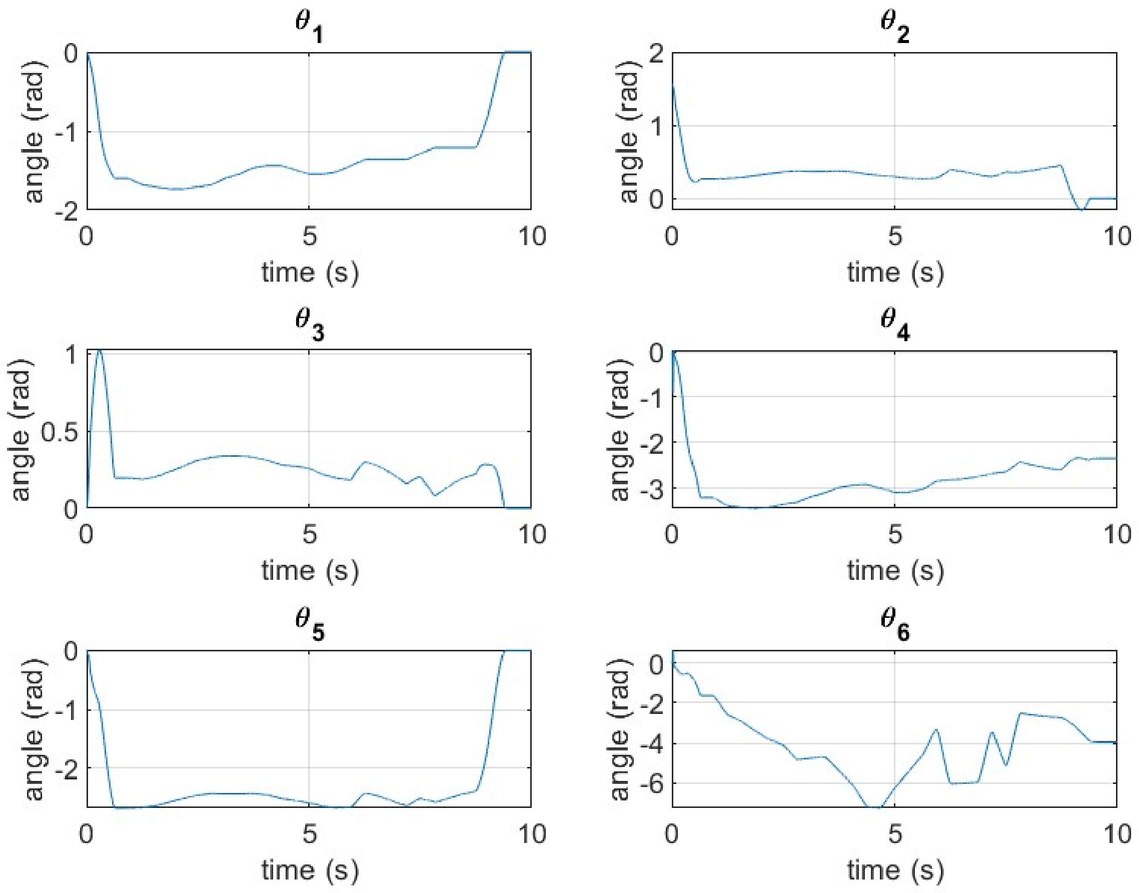

Figure 3: Graphs of actual $\theta$'s over time, with no laser tracking, perfect plant (sim-exact = true).

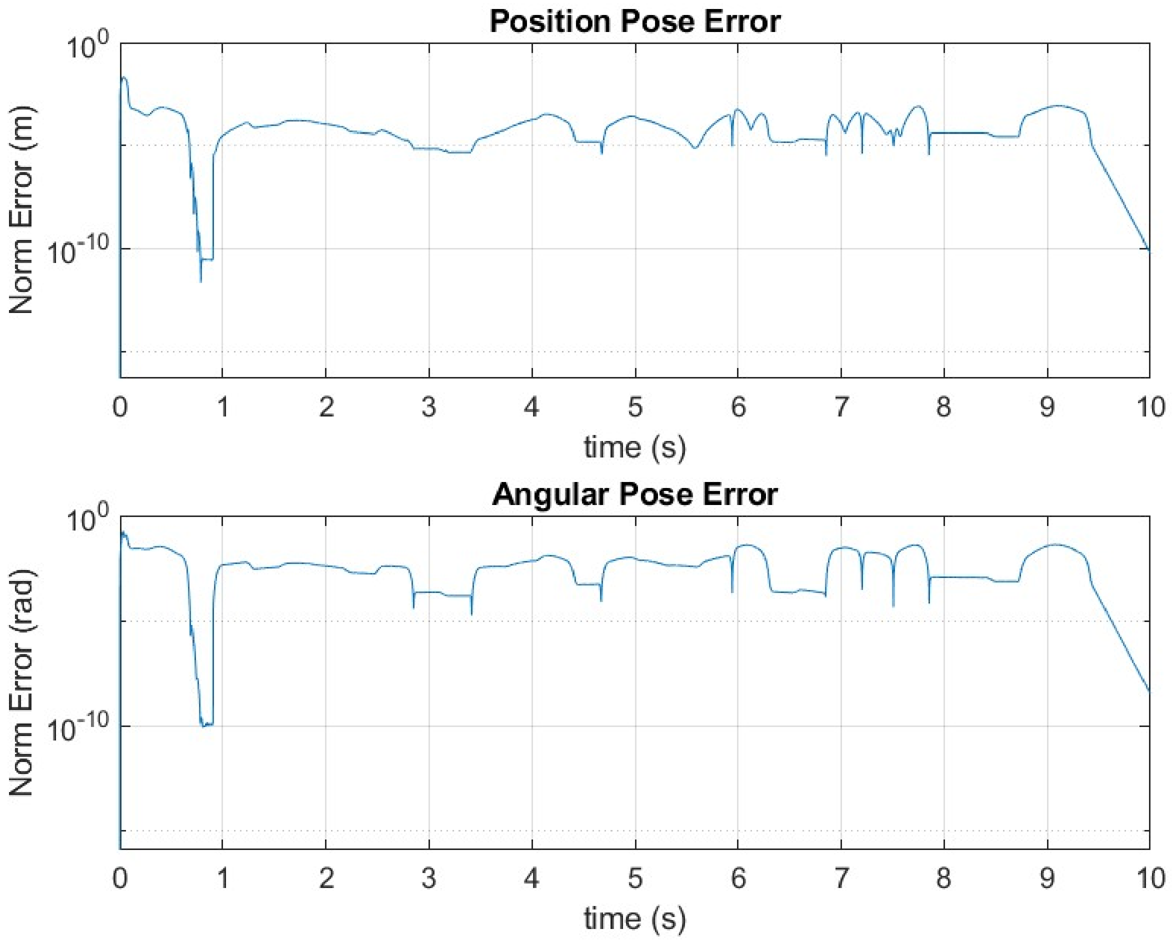

Figure 4: Graphs of pose errors over time, with no laser tracking, perfect plant (sim-exact = true).

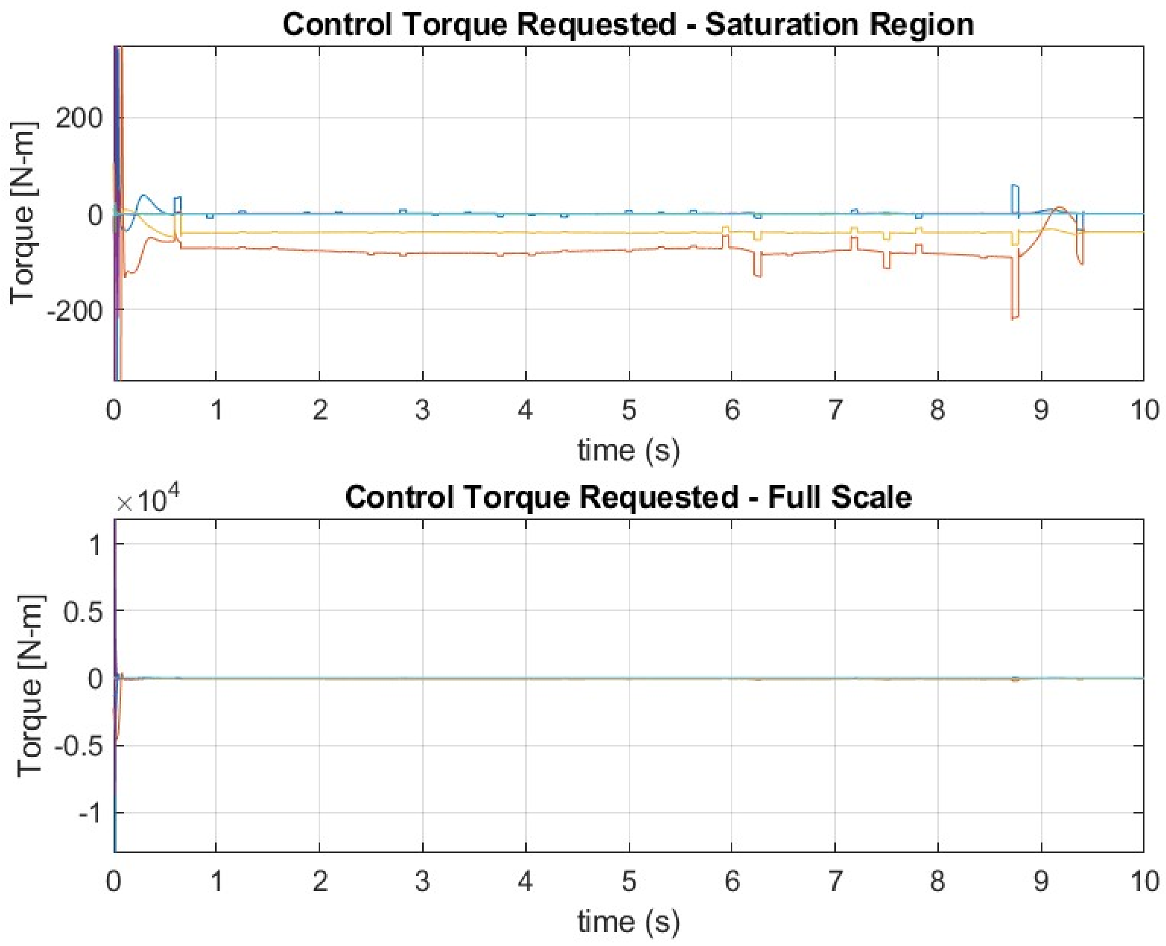

Figure 5: Graph of control torques over time, with no laser tracking, perfect plant (sim-exact = true).

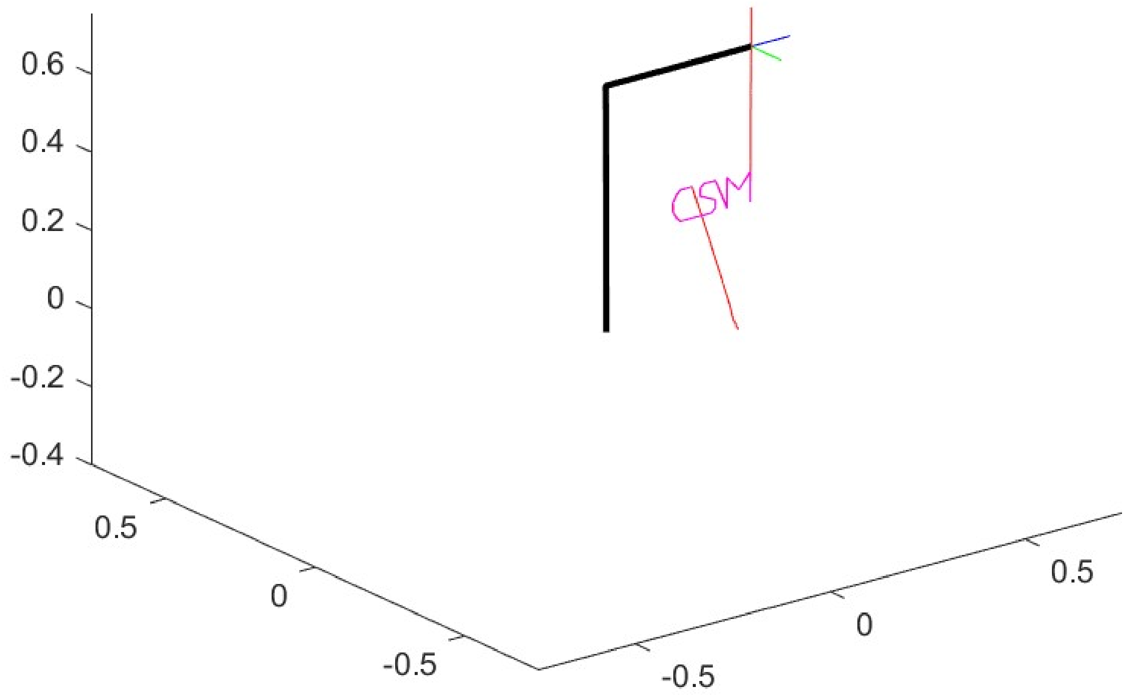

Figure 6: Result trajectory traced by robot arm, with no laser tracking, non-perfect plant (sim-exact = false).

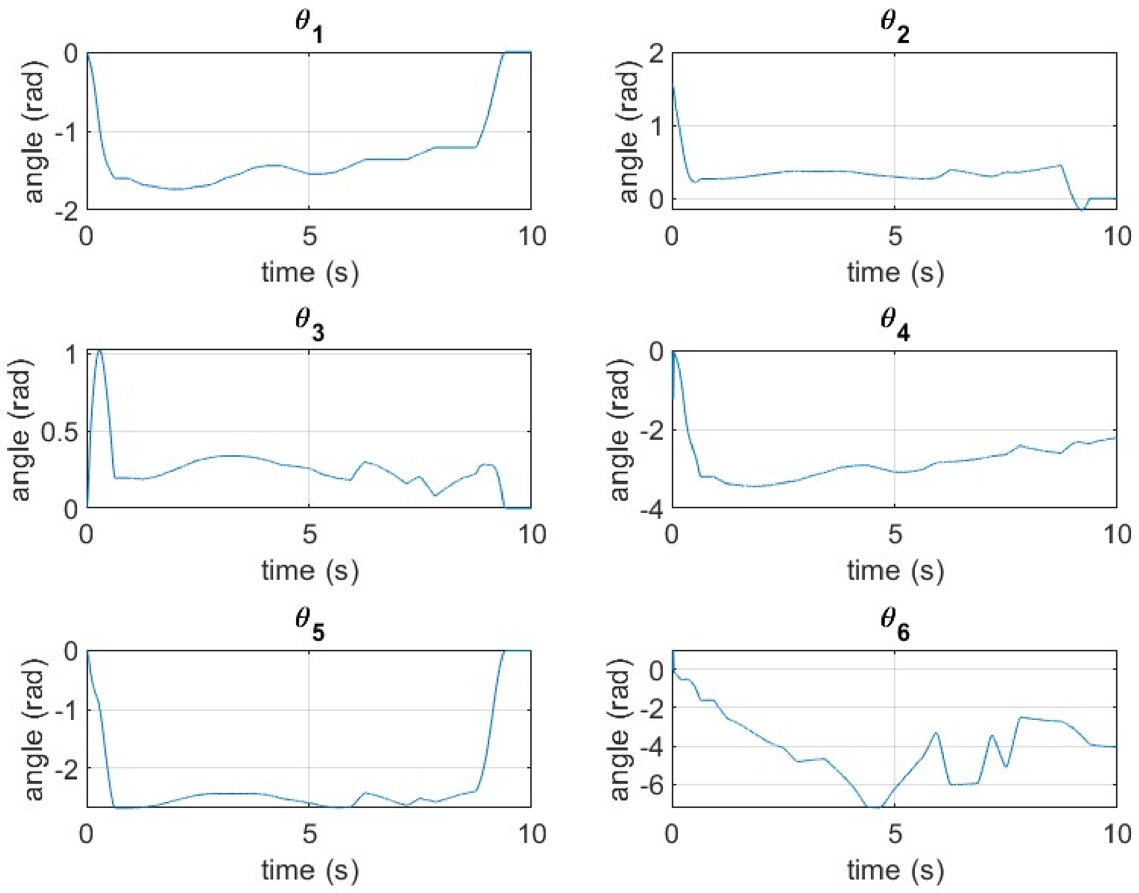

Figure 7: Graphs of actual $\theta$'s over time, with no laser tracking, non-perfect plant (sim-exact = false).

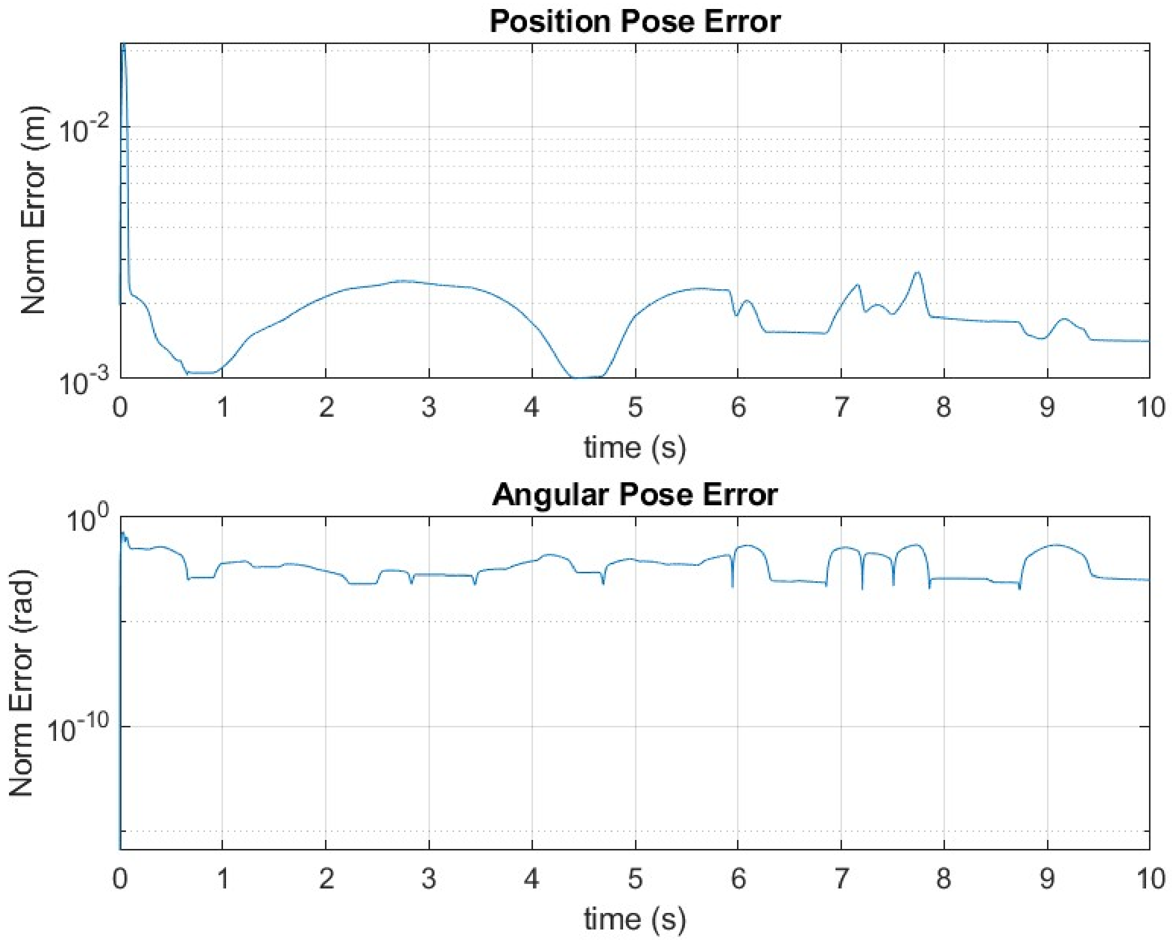

Figure 8: Graphs of pose errors over time, with no laser tracking, non-perfect plant (sim-exact = false).

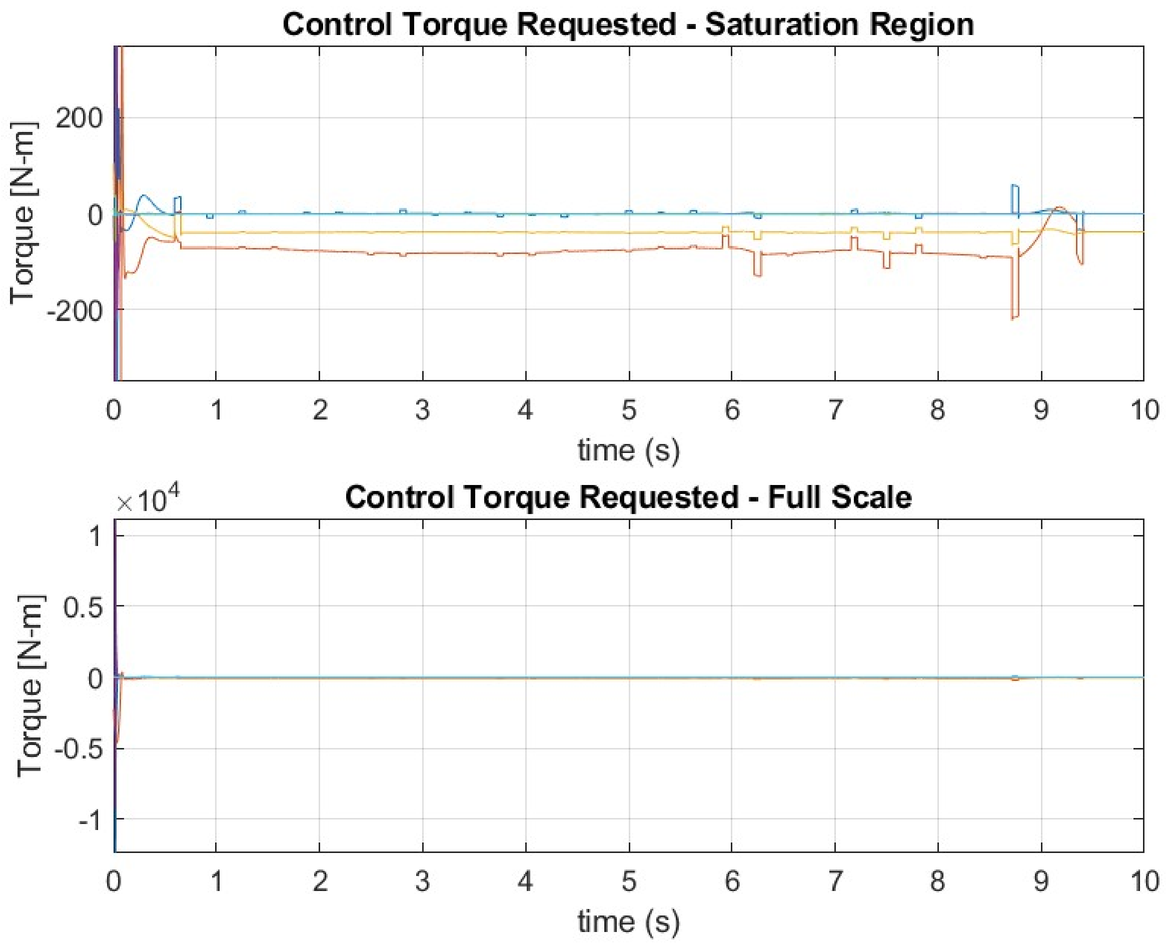

Figure 9: Graph of control torques over time, with no laser tracking, non-perfect plant (sim-exact = false).

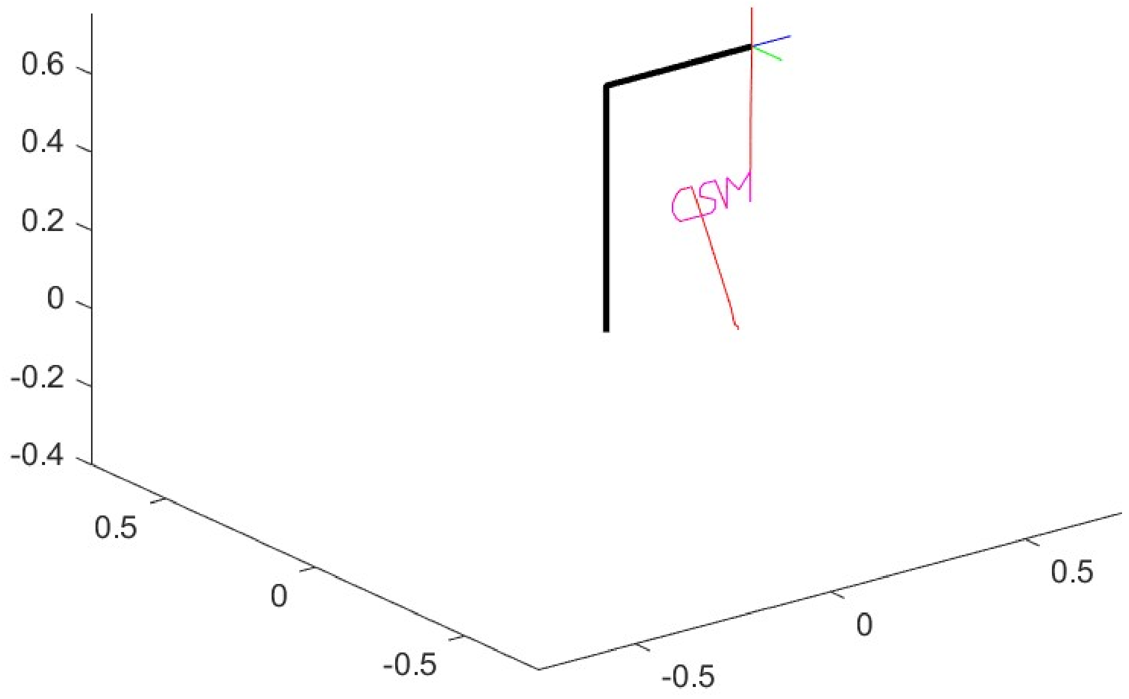

Figure 10: Result trajectory traced by robot arm, with laser tracking, perfect plant (sim-exact = true).

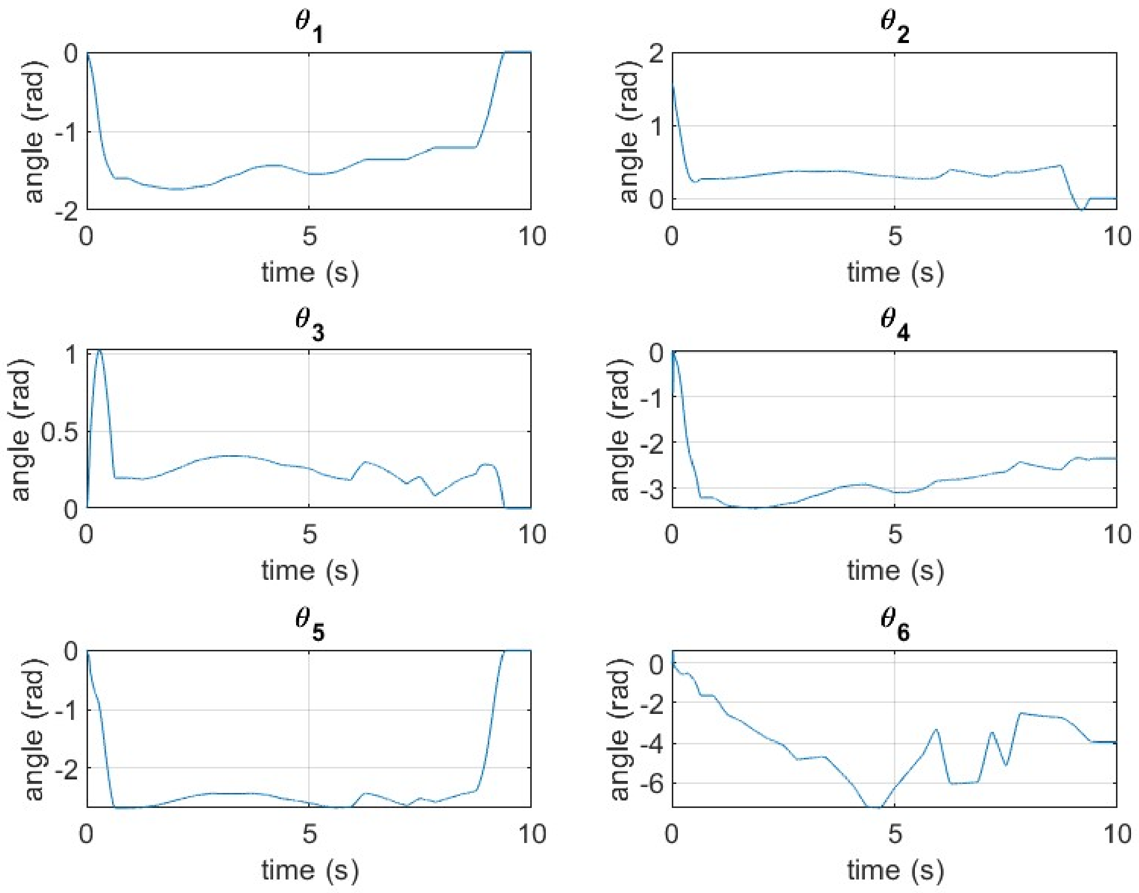

Figure 11: Graphs of actual $\theta$'s over time, with laser tracking, perfect plant (sim-exact = true).

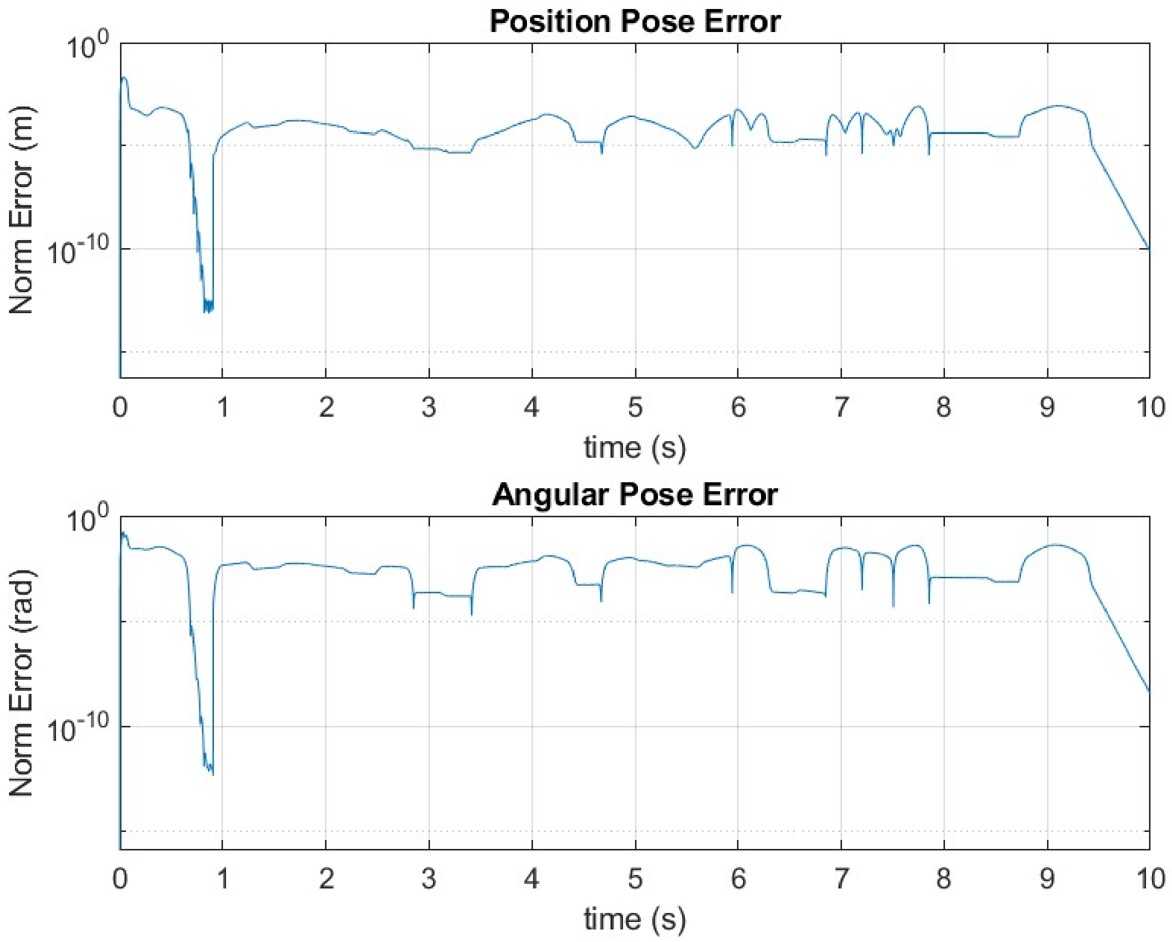

Figure 12: Graphs of pose errors over time, with laser tracking, perfect plant (sim-exact = true).

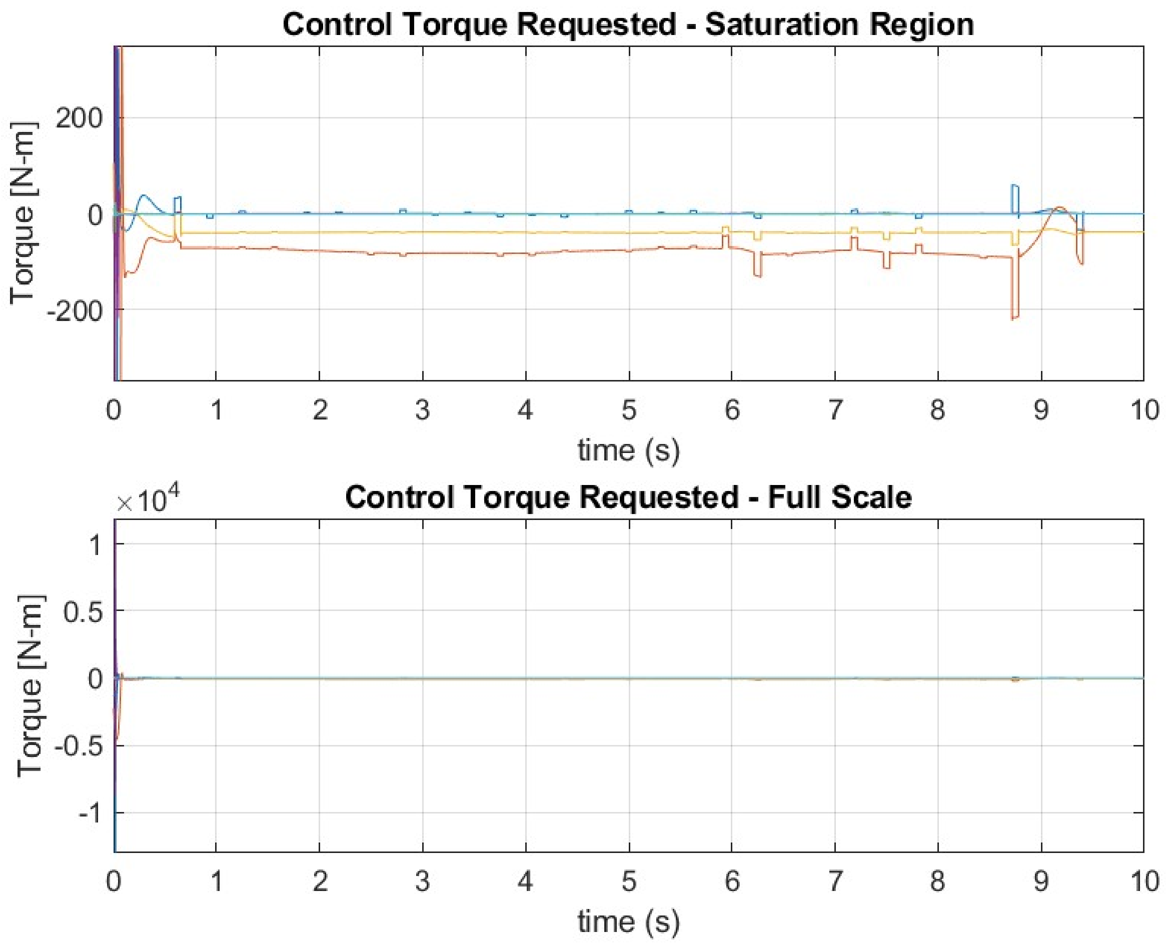

Figure 13: Graph of control torques over time, with laser tracking, perfect plant (sim-exact = true).

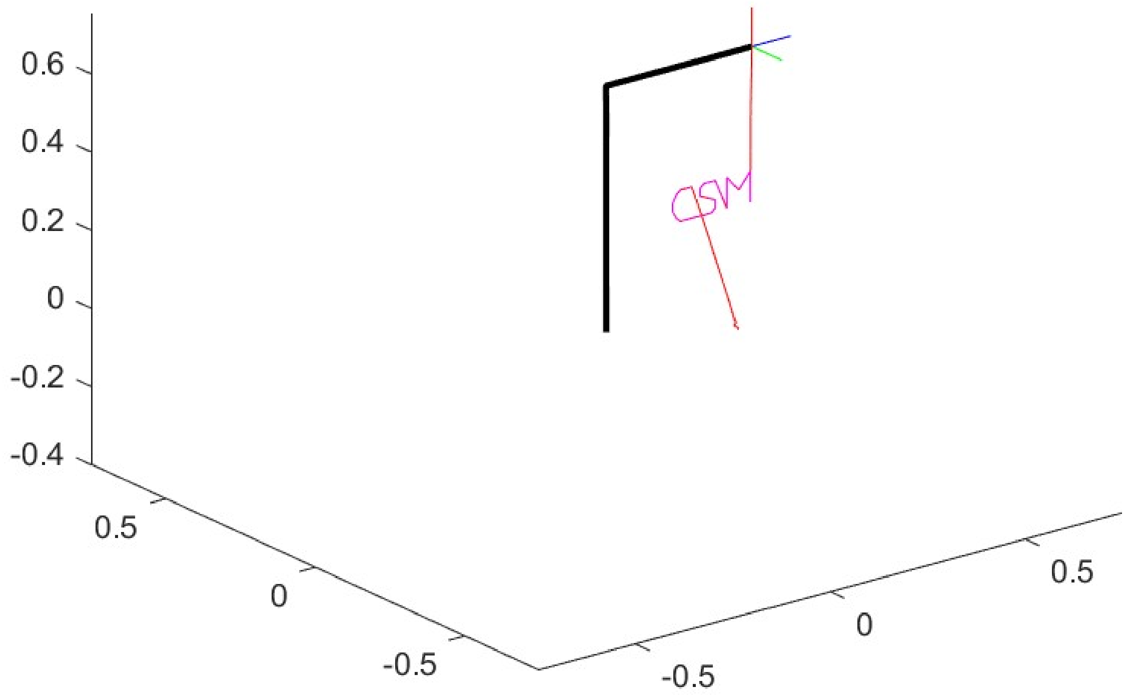

Figure 14: Result trajectory traced by robot arm, with laser tracking, non-perfect plant (sim-exact = false).

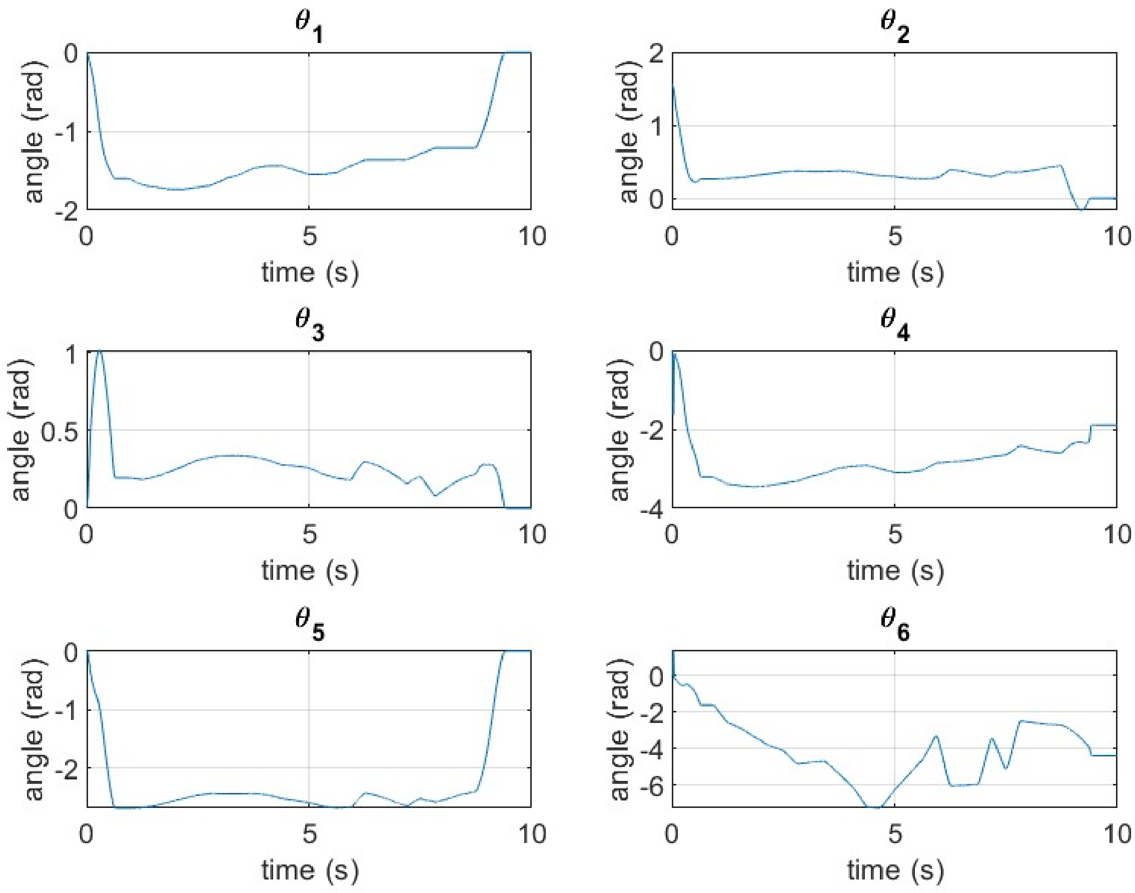

Figure 15: Graphs of actual $\theta$'s over time, with laser tracking, non-perfect plant (sim-exact = false).

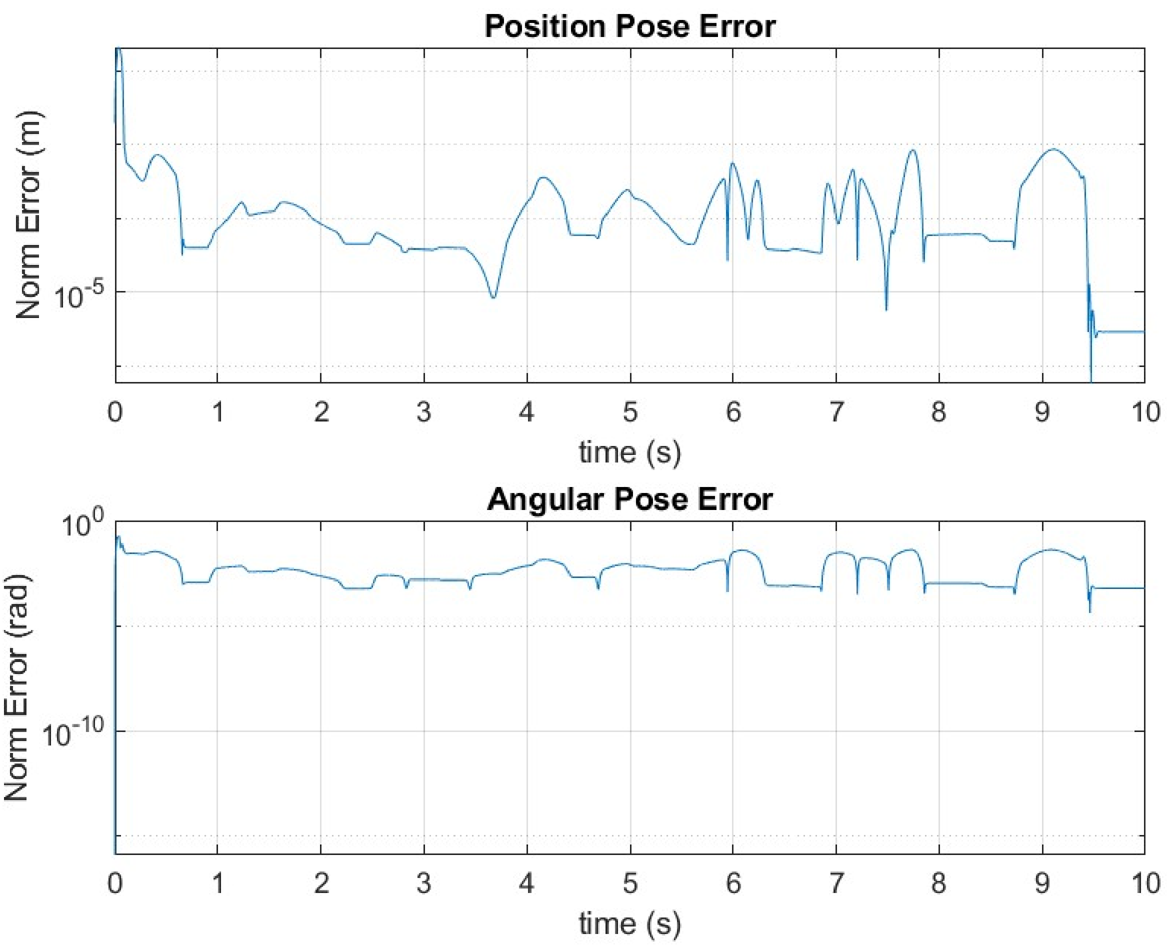

Figure 16: Graphs of pose errors over time, with laser tracking, non-perfect plant (sim-exact = false).

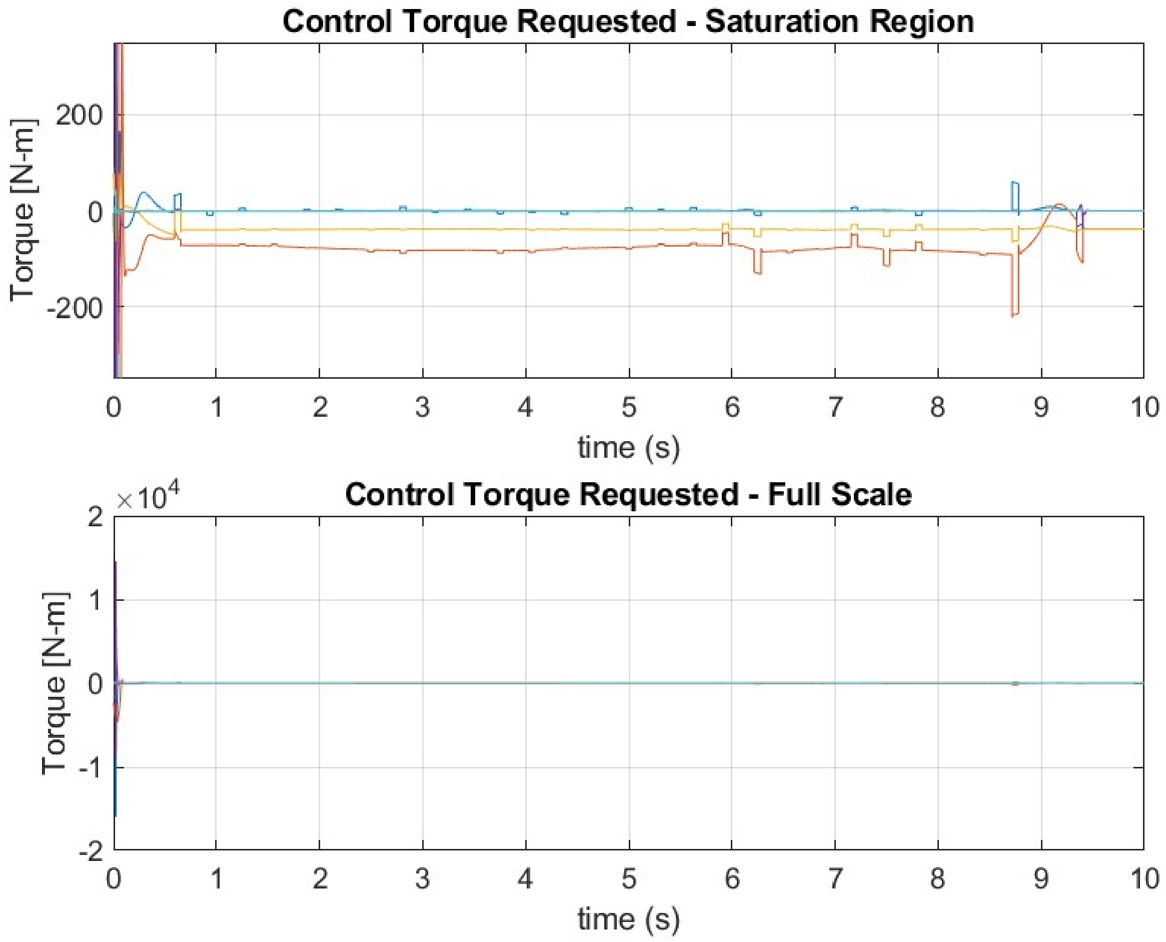

Figure 17: Graph of control torques over time, with laser tracking, non-perfect plant (sim-exact = false).

**The controller has its limits in the trajectory rate**. When the trajectory rate gets too high (little time to move through the trajectory), the controller cannot keep up and may have an undefined behavior, moving through a random trajectory while no longer being able to trace out "CSM" clearly. This could be due to over-correction and over-backtracking. Table 2 shows the progression from trajectory rates that the controller can manage to those that render the controller ineffective. We can see that, as the simulation time decreases (increasing trajectory rate), the RMS error increases. At simulation time of 6 seconds, the RMS errors increases roughtly 2 times compared to 8 seconds simulation time. It happens to be that 6 seconds is roughly the lowest simulation time that the controller can keep up, in my observation. Any faster rates (5 or 4 seconds simulation times), the arm cannot trace CSM well anymore (path no longer resembling CSM letters).

Table 2: Simulation time versus RMS errors. For this process, sim-exact is set to false, sim-time is set to 10 seconds, and laser-feedback is set to false. Gains are set at $K_d$ of 212 and $K_p$ of 20022.

| sim-time total (s) | RMS position pose error | RMS ang pose error |
|---|---|---|
| 20 | 0.0018 | 0.0071 |
| 18 | 0.0018 | 0.0081 |
| 16 | 0.0019 | 0.0092 |
| 14 | 0.0019 | 0.0107 |
| 12 | 0.0020 | 0.0129 |
| 10 | 0.0023 | 0.0164 |
| 8 | 0.0031 | 0.0234 |
| 6 | 0.0063 | 0.0449 |
| 5 | 0.0110 | 0.1023 |
| 4 | 0.0623 | 0.6360 |

## Conclusion

**Over the semester, the controllers have increased in complexity and accuracy has also increased as a result.**

In Part 3, the controller only implements theta-only (configuration space) open loop control. There is no feedback mechanism on the end effector location. We only feed the joint angles and joint rates as desired into the theta controller and hope that the geometry is accurate enough so that in the end the tool is actually at the location desired.

In Part 4, we have the feedback transform (actual) to find the transform error, which would eventually be multiplied with the gain to have an amount of error in the twist (operational space rates) vector to correct the arm trajectory. In doing so, we have a way to address the errors that we fail to anticipate beforehand that would put the end effector at a location other than desired. We also use the velocity Jacobian inverse to go from operational space velocities to joint space rates (theta dots). The joint space rates were the control signal to feed into the theta controller in Part 4. Laser feedback is also put into use in Part 4.

For Part 5, on top of the feedback mechanisms of seen in Part 4, it includes not only operational space velocities but also pose (position and angular). $\dot{J}_v$ is now used to calculate the actual operational space acceleration vector. The control acceleration vector is used with Jv inverse to generate joint accelerations. With joint accelerations, Newton Euler algorithm is user for inverse dynamics to calculate torque control signals.

**In conclusion**, we went from motions with wide curvatures and visible discrepancy with the goal trajectory (Part 3, Figure 18), to more straight motions with still some oscillation movements during initial ramp-up (Part 4, Figure 19), to straight motions with minimal unnecessary movements (Part 5, Figure 20). This controller in Part 5 proves to make more direct and accurate motions. With high enough gains, the controller performance is reasonably consistent across changes in feedback (theta or laser tracking) and robust to geometry imperfections. Possible **future extensions** to the work in Part 5 include addressing the short period of chattering and torque saturation during intial ramp-up to the letter C starting point, or making the controller capable of even higher trajectory rates, or reducing the gain needed (below 1000) to have the arm tracing CSM accurate to the human eye.

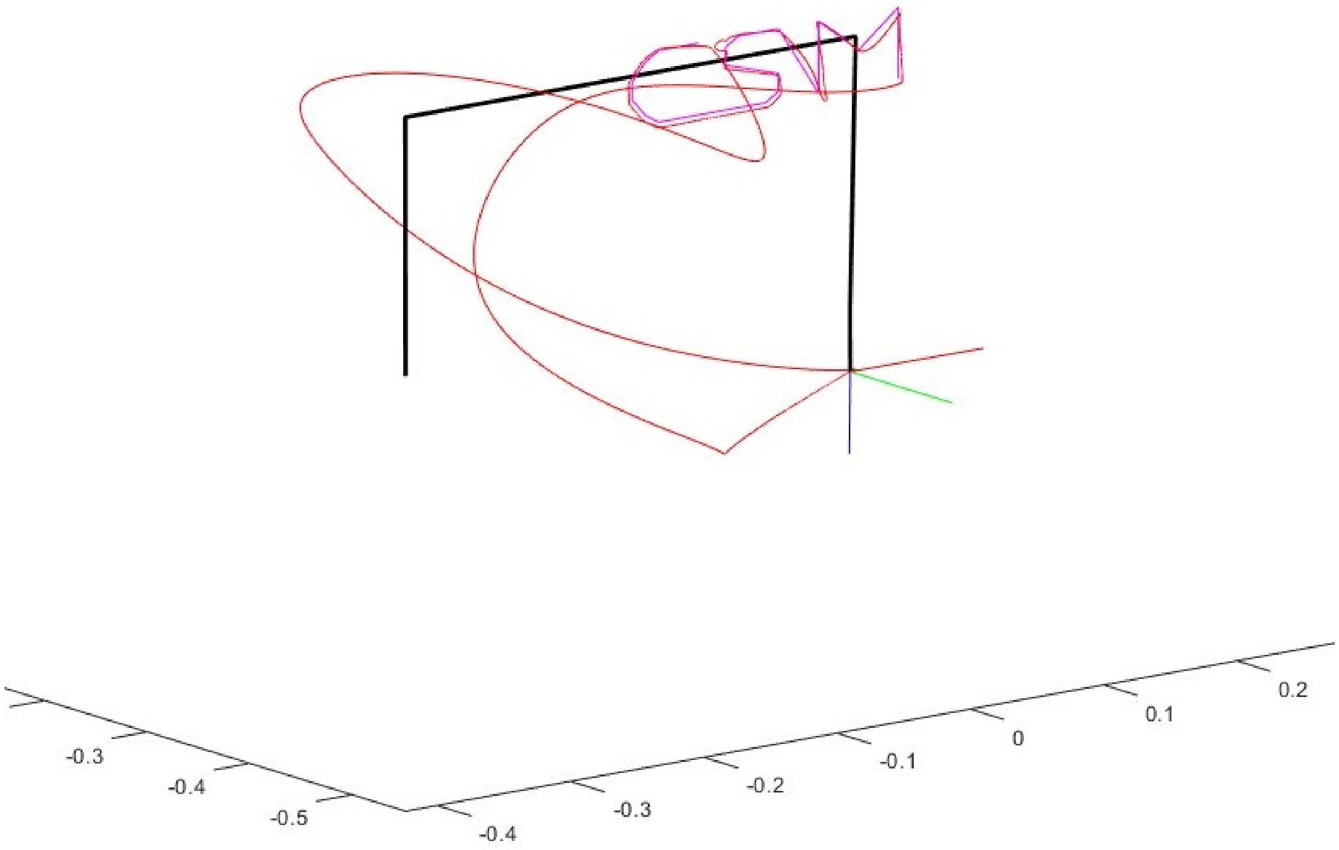

Figure 18: Result trajectory traced by robot arm in Part 3.

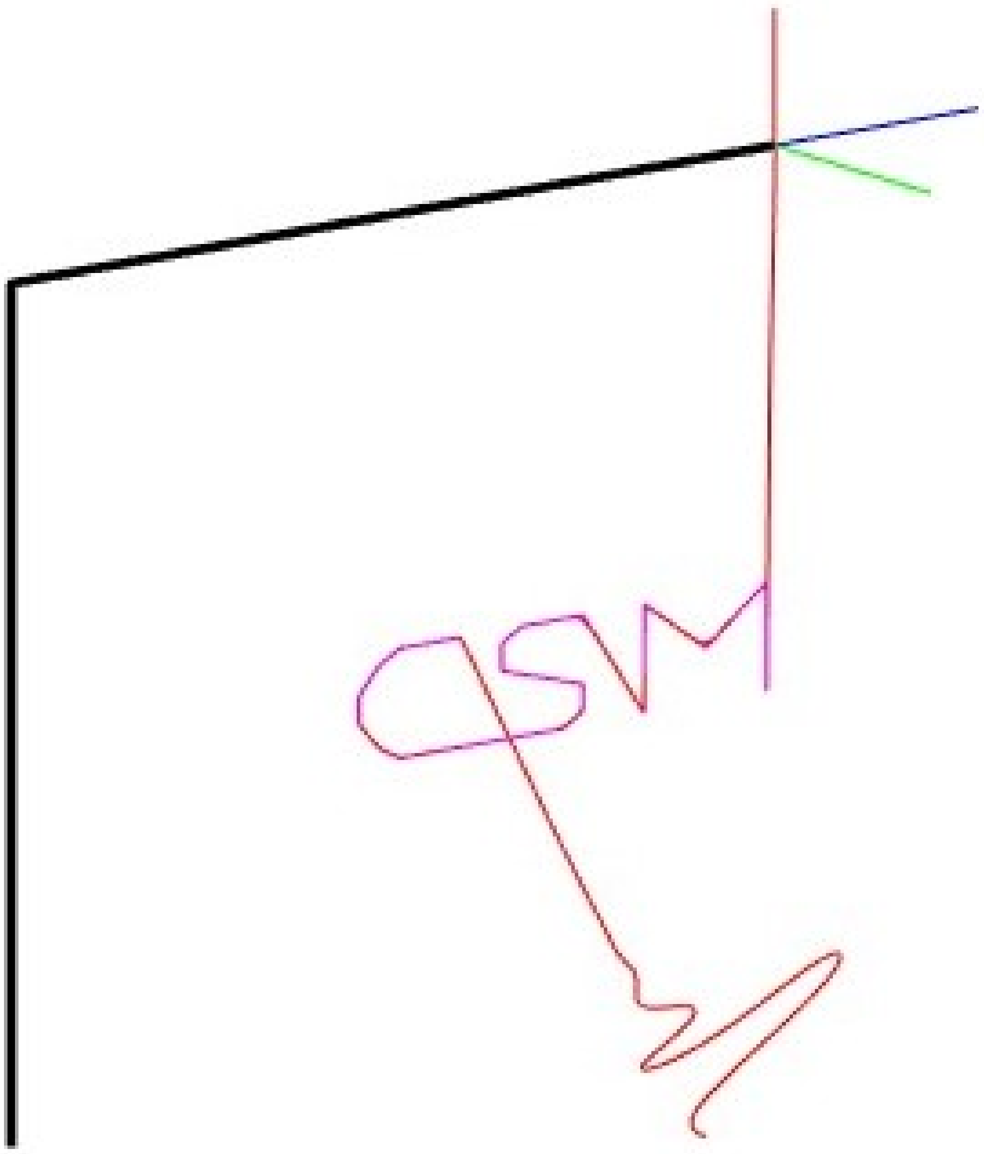

Figure 19: Result trajectory traced by robot arm in Part 4..

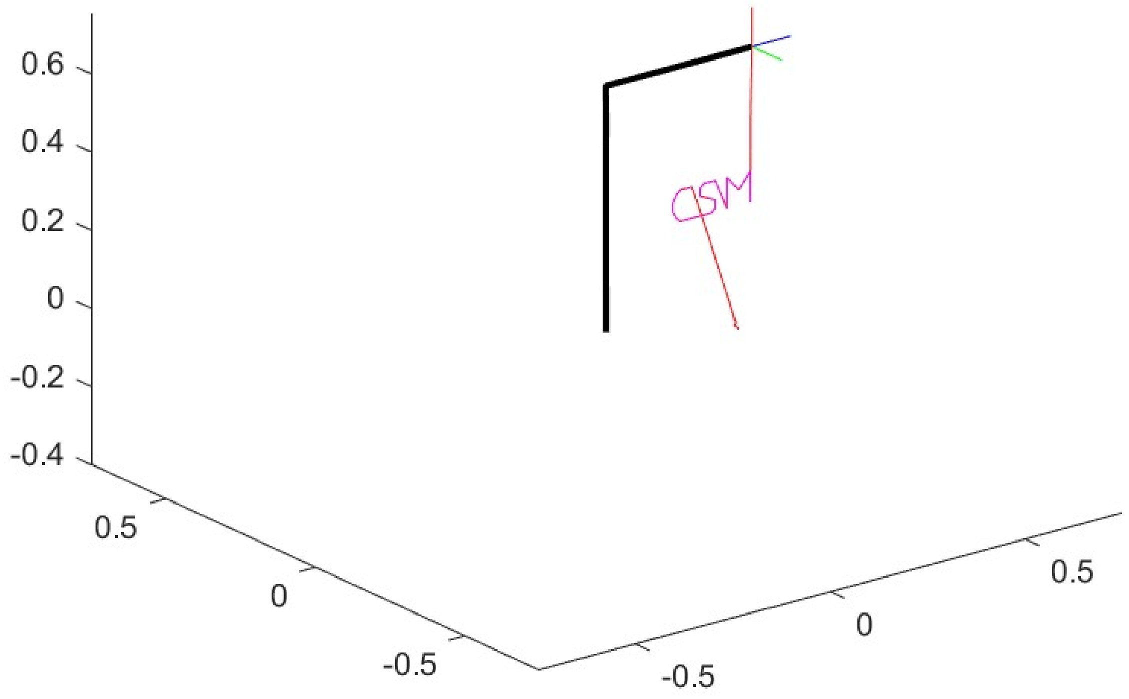

Figure 20: Result trajectory traced by robot arm in Part 5, with laser tracking, non-perfect plant (sim-exact = false).

**The videos of the robot arm in action are included in the submission package as mp4 files.**