MEGN544 - Project Part 4 Report

Student: Thong Quoc (Bill) Huynh

November 18, 2023

For Project Part 4, **the approach** is to implement and operational space control system. This means that the error is calculated directly in the operational space (in the linear velocity and the angular velocity of the end effector). Such error is used as negative feedback to the forward ("open-loop") linear velocity and angular velocity so that the robot arm not only traces CSM but also traces at the desired location. Generally speaking, this approach is similar the dhInvKine() function implemented in Project Part 3, in which the Pose Error Gain K was set at exactly 1.

In details, one of the first major steps is to create way points in the trajectory using vectors combining the target 3D positions and the target rotations (encoded in unit quaternions). Constant acceleration interpolation is still used to interpolate between the way points to generate "position" vectors that include interpolated 3D positions (p) and interpolated quaternions (Q), as well as "velocity" vectors that include the interpolated rates - linear velocities (v) and quaternion rates ($\dot{Q}$). An angular velocity (ω) can be calculated from each quaternion rate ($\dot{Q}$) using the appropriate conversion formula. With the interpolated results, the desired transformations (T_{des}) and desired rates ($twist_{des}$) can be generated. All of this is carried out in the Trajectory Generator block, as seen in the Simulink block diagram Figure 1 for reference.

The desired transformations (T_{des}) is used in the Transform Error block, together with the current actual transformation, to calculate the error in the pose (both position pose error and rotation pose error). This pose error is then multiplied with the chosen Pose Error Gain to generate the negative feedback to the rates ($twist_{des}$) needed to trace the CSM trajectory as accurately as possible. The current transform can be calculated in two ways, either by forward kinematics of the current joint angles or by laser tracking.

Finally, a multiplication of the inverse of the Velocity Jacobian to the operational rates can give the joint rates ($\dot{\theta}$) needed to feed to the Theta Controller. The target joint angles θ's themselves are calculated from integrating the joint rates, inside the Theta Controller block. The Theta Controller's block diagram is shown in Figure 2 for reference.

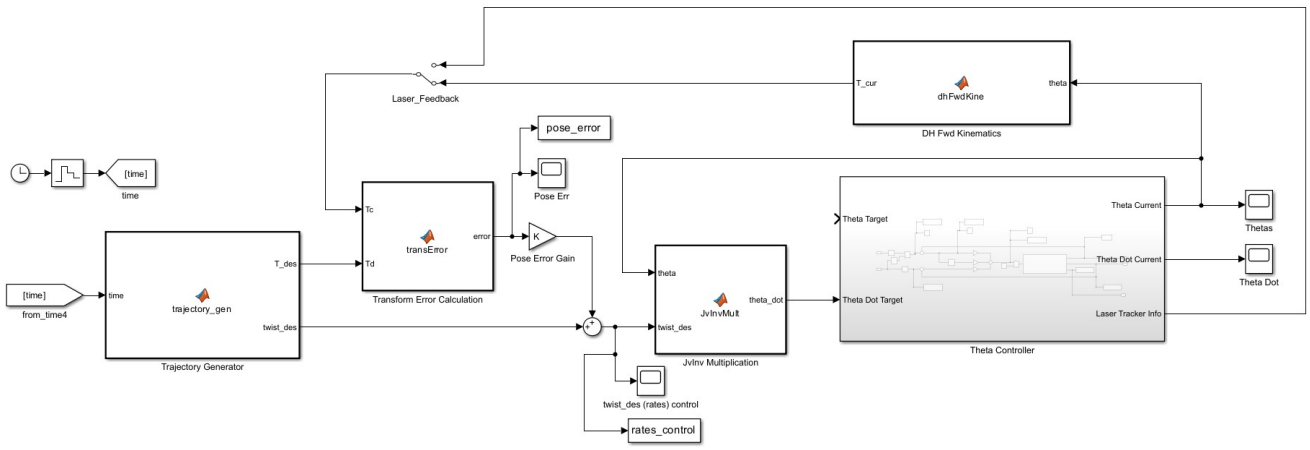

Figure 1: Screenshot of top-level block diagram used in MATLAB Simulink.

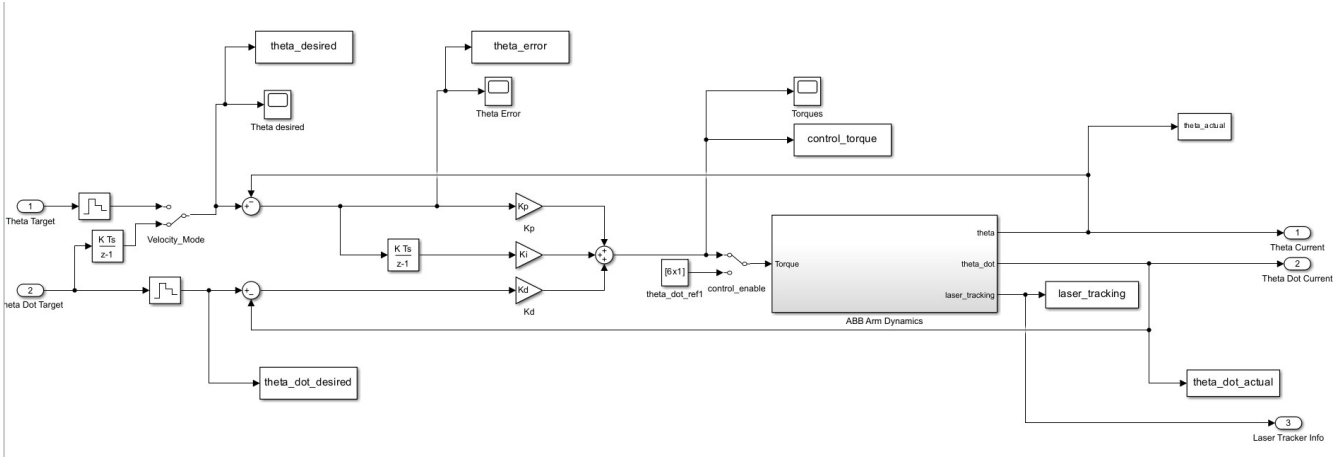

Figure 2: Screenshot of Theta Controller block diagram used in MATLAB Simulink.

In terms of the **control gains**, the proportional gain ($K_p = [500; 500; 500; 300; 10; 10]$) and derivative gain ($K_d = [75; 75; 75; 40; 1; 1]$) internal to the Theta Controller are not tuned in this part of the project, compared to what is initially given in the simulation model. The internal integral gain (K_i) is set to all 0 because the control law of the system is such that the Pose Error Gain (K) constitutes a system that acts like a pure integrator in the plant already, making the internal inegral gain (K_i) no longer necessary. The Pose Error Gain (K) is chosen experimentally. A gain of 1 essentially still helps the robot arm trace CSM at the approximately correct location, but the error tracking is not aggressive enough to generate straight movements and the arm makes some unnecessarily wide movements. I then gradually increase the gain and can see the actual trajectory show fewer unnecessary traversals. From the gain value of 35 on, I do not see significant improvements with increased gains. The $twist_{des}$ control signals even show more control chatter above the gain of 35. Therefore, I have chosen to use the value of $[35; 35; 35; 35; 35; 35]$ for the Pose Error Gain (K).

The **result (without laser tracking** with the robot arm successfully tracing the CSM letters with no significant unnecessary or inaccurate movement is shown below, from Figure 3 to Figure 13, including the result CSM trajectory traced, plots of joint angles, joint angle rates, control operational space rates, pose error, and control torques. This result uses pose error feedback from forward kinematics of actual theta signals, without laser tracking.

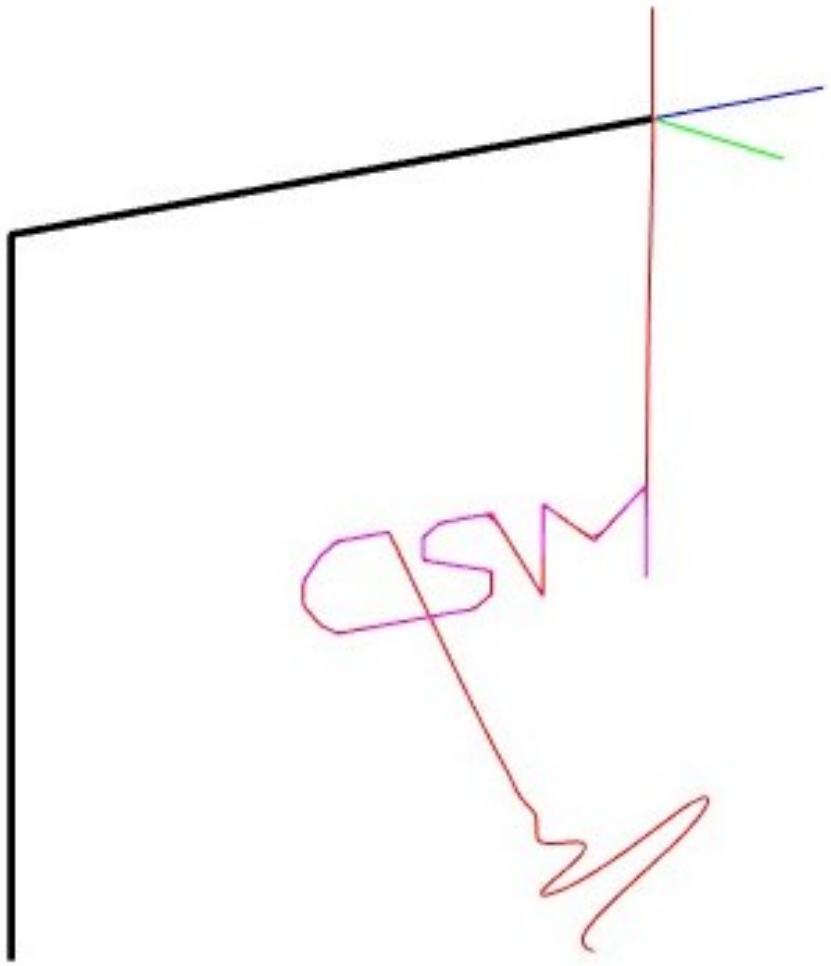

Figure 3: Graph of result trajectory traced by robot arm, with no laser tracking.

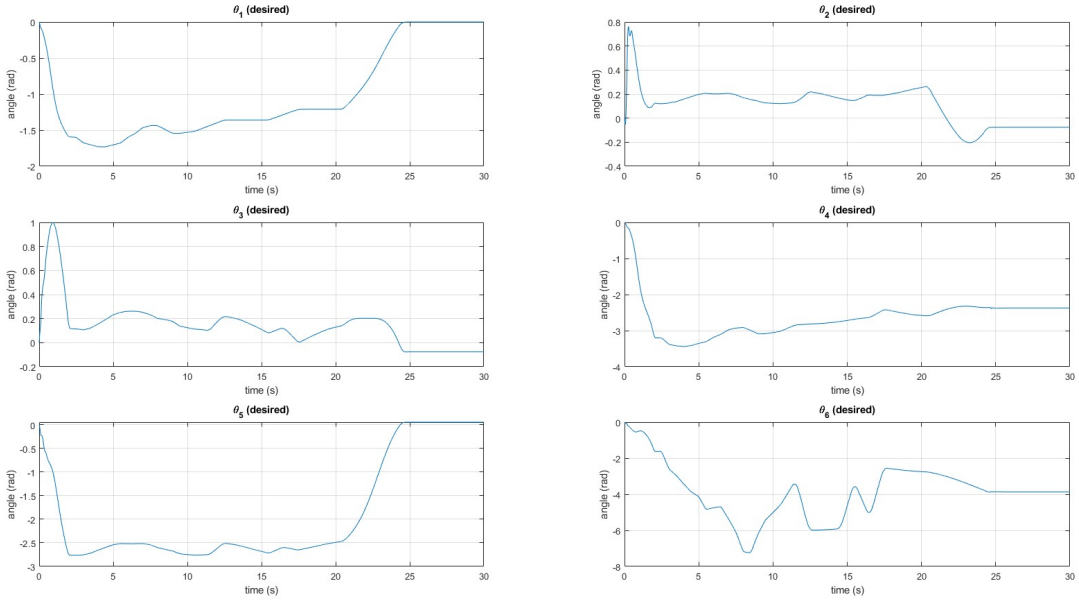

Figure 4: Graphs of desired θ's over time, with no laser tracking.

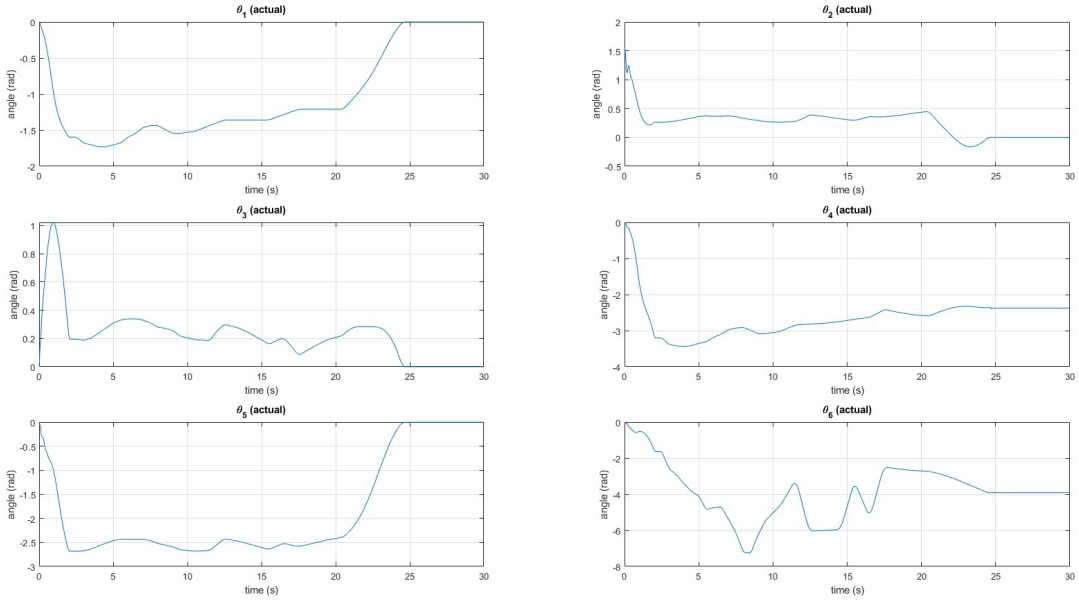

Figure 5: Graphs of actual θ's over time, with no laser tracking.

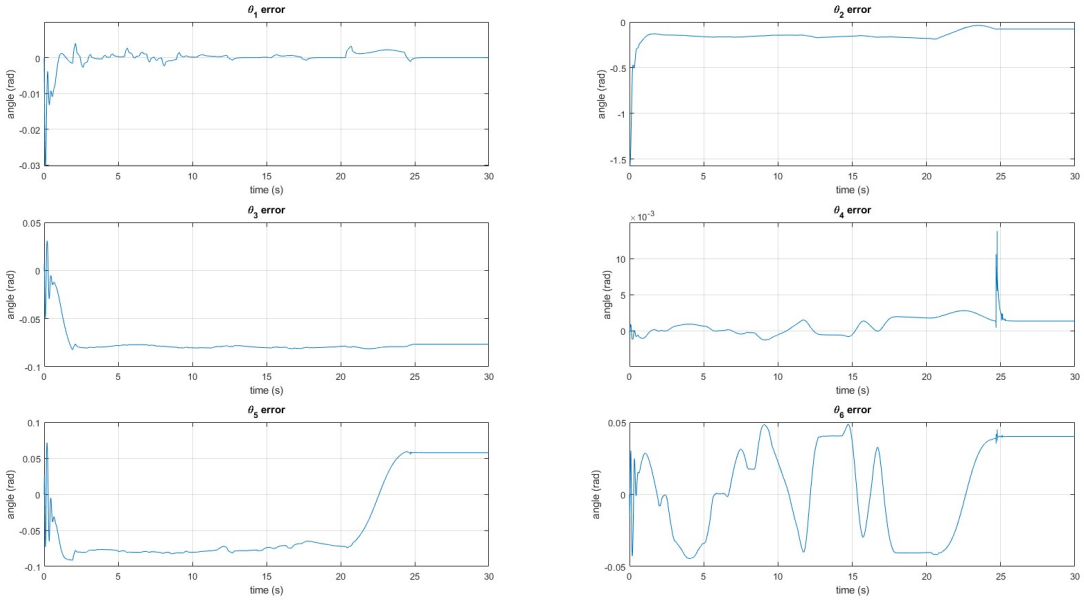

Figure 6: Graphs of θ errors over time, with no laser tracking.

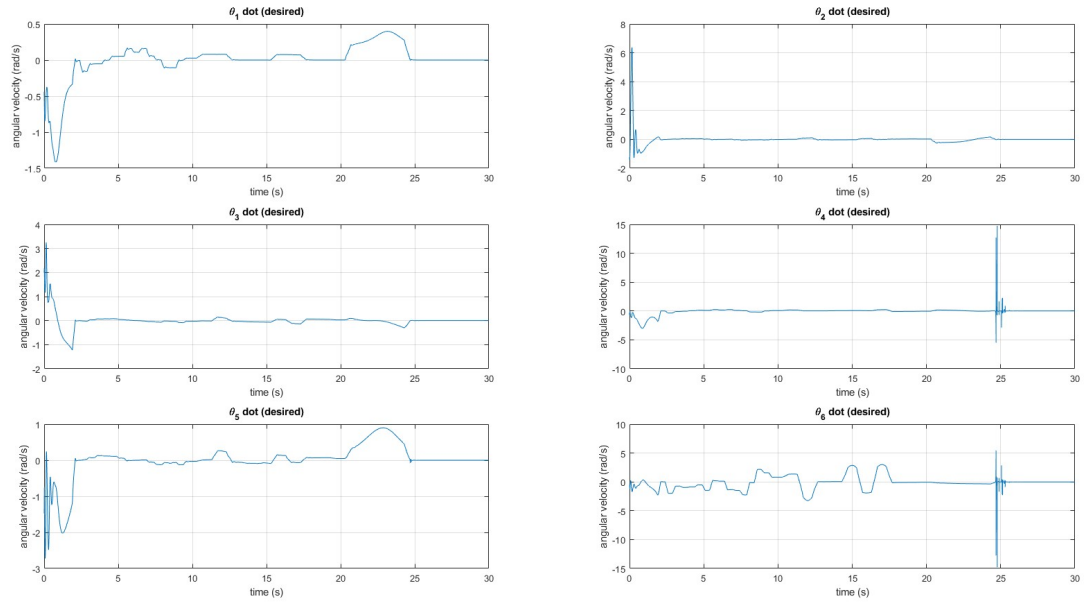

Figure 7: Graphs of desired $\dot{\theta}$'s over time, with no laser tracking.

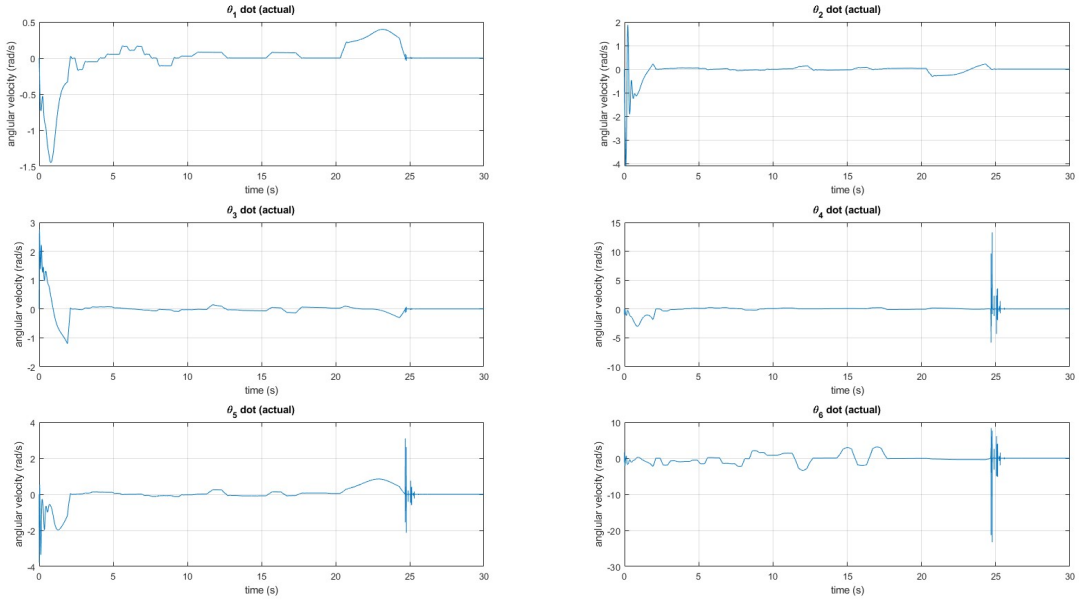

Figure 8: Graphs of actual $\dot{\theta}$'s over time, with no laser tracking.

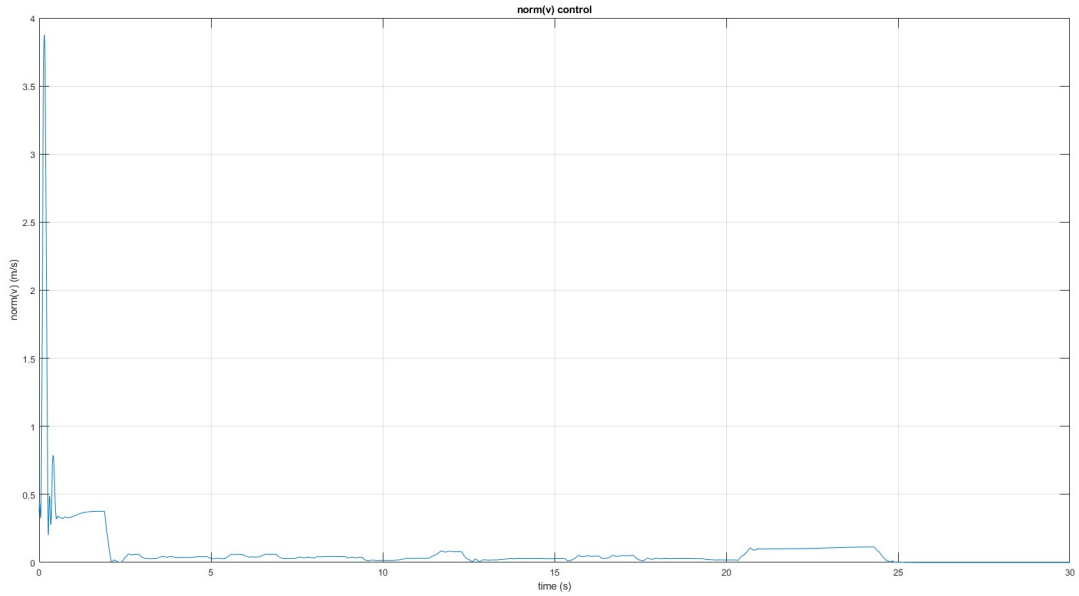

Figure 9: Graph of norm of control linear velocity over time, with no laser tracking.

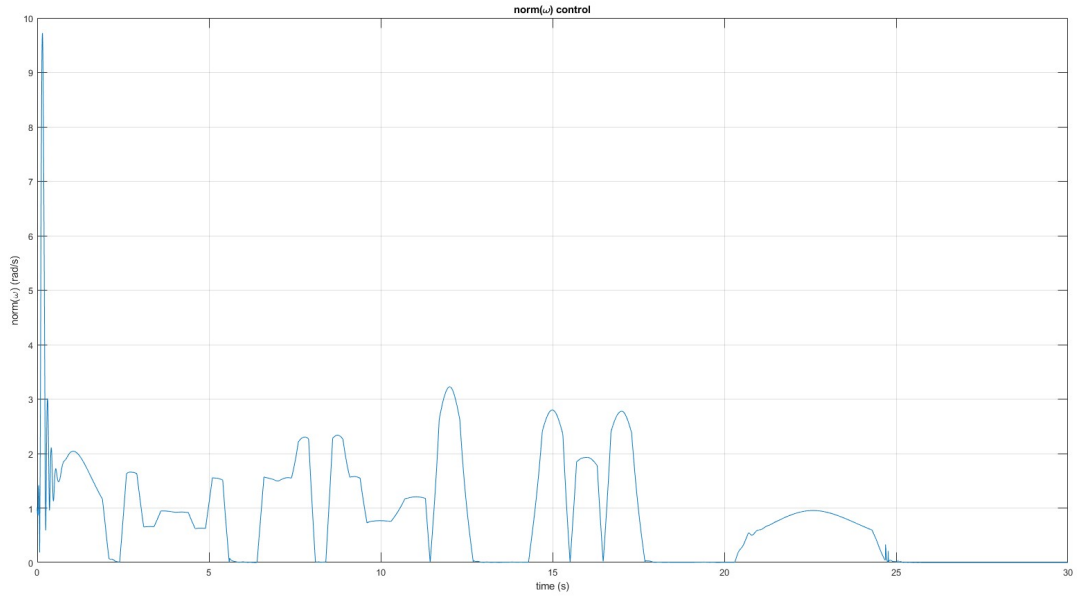

Figure 10: Graph of norm of control angular velocity over time, with no laser tracking.

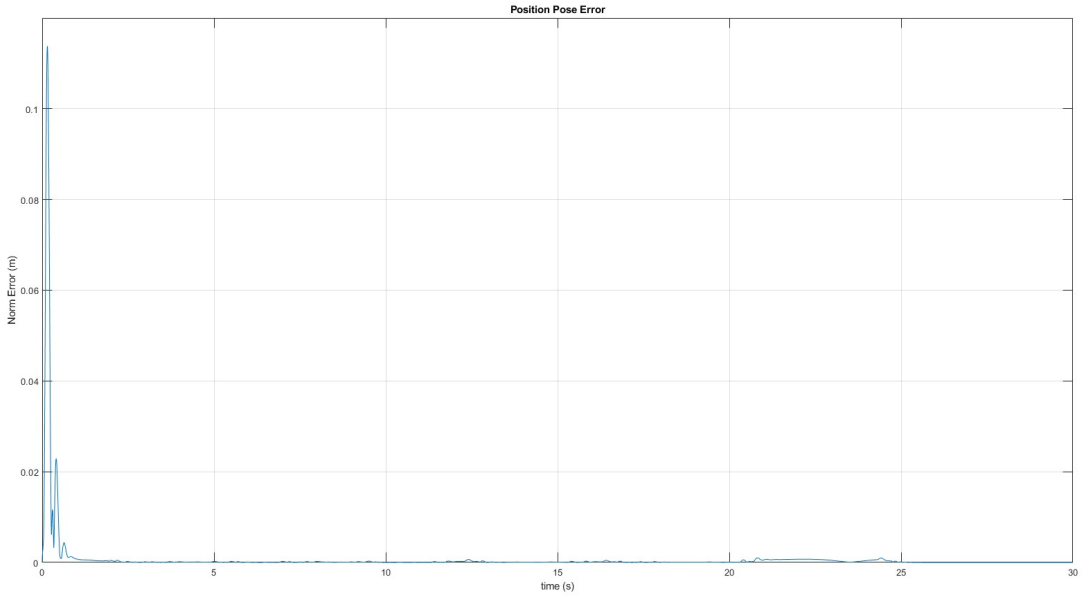

Figure 11: Graph of norm of position pose error over time, with no laser tracking.

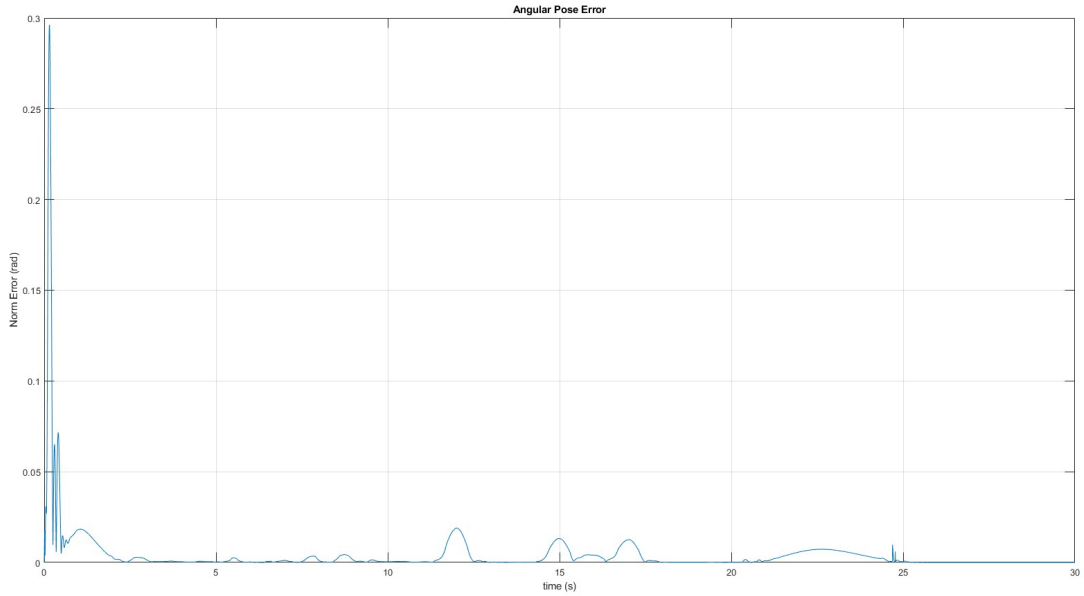

Figure 12: Graph of norm of angular pose error over time, with no laser tracking.

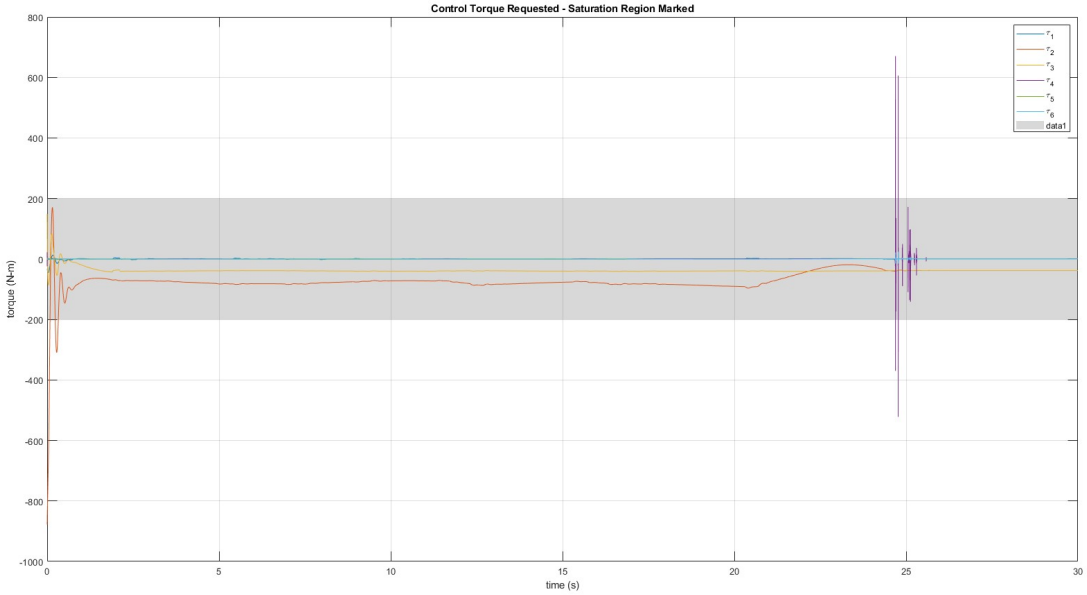

Figure 13: Graph of control torques over time, with saturation region marked, with no laser tracking.

As seen from Figure 3, **the result CSM trajectory with this DH inverse control approach and no laser tracking** shows a smooth path closely following the goal CSM points, with some minimal oscillating motion or chatter at the beginning. In the saved trajectory before interpolation, the starting point is the DH zero configuration and the final point is the encoder zero configuration. Because the actual starting point is the DH zero configuration and the theta signal inside the Theta Controller (which is integrated from the theta dot signal) starts from numerically zero, there is a difference at the starting time of 0. This explains why the robot arm tries hard to

account for the error and oscillates at the beginning.

Judging from the theta dot actual plots in Figure 8, the initial control chatter goes away by 0.8 seconds. In this initial period, one high theta dot overshoot absolute value is in $\dot{\theta}_5$, -3.788 rad/s at 0.004 seconds. Another group of chatter occurs at the end of the CSM trajectory right before all the theta dots go to zero (around 25 seconds) and this chatter period lasts for 0.86 seconds on average. This is seemingly because the controller realizes that the next way point (encoder zero configuration) is quite far from the end the M letter and it tries to generate a high joint angular velocity to correct this error. This causes big changes and overshoots and backtrackings.

The actual theta plots in Figure 5 show generally smooth theta paths with no significant chatter nor discontinuity. The θ_6 path 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. Still, the θ_6 path does not show chatter.

The norm control linear velocity in Figure 9 and norm control angular velocity in Figure 10 also show the initial control chatter going away by 0.8 seconds. After 2 seconds, the control linear velocity stays mostly near zero with some periods of higher values but not exceeding 0.25 m/s, while the control angular velocity shows frequent changes, exceeding π at one point (around 12 seconds).

The position pose error graph in Figure 12 and angular pose error graph in Figure 12 show the pose errors go to near zero after the initial period of control chatter of 0.8 seconds, except some small occasional changes over time. This means that the error feedback manages to bring the end effector to near target location fast with the gain values chosen and the error is maintained generally low afterwards.

The graph of control torque in Figure 13 show the requested torques staying below the saturation values, except for the initial control chatter period of 0.8 seconds and the overcorrection at the end of the CSM trajectory to get to encoder zero configuration. The highest torque value in the initial control chatter period is in τ_2 with -307.34 N-m at 0.282 seconds.

The **result with laser tracking** is shown below, from Figure 14 to Figure 24. This result uses pose error feedback from laser tracking in the ABB arm.

From the theta dot actual plots in Figure 19, the initial control chatter also goes away by 0.8 seconds. In this initial period, the theta dot overshoot absolute value is in $\dot{\theta}_5$, is -3.790 rad/s at 0.006 seconds, similar to the DH inverse result (only 0.06 % difference). Another group of chatter occurs at the end of the CSM trajectory right before all the theta dots go to zero (around 25 seconds), and it lasts for 0.23 seconds on average. **Compared to 0.86 seconds from the DH inverse result above, the final control chatter period with laser tracking is reduced by roughly 73%. Such reduction in control chatter is desirable.**

The actual theta plots in Figure 16 is **not different from the DH inverse without laser tracking result** in Figure 5, with also smooth theta paths and no significant chatter nor discontinuity. The θ_6 path 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.

The norm control linear velocity in Figure 20 and norm control angular velocity in Figure 21 **are not very different from the DH inverse without laser tracking result**, without any considerable improvement. The graphs also show the initial control chatter going away by 0.8 seconds. After 2 seconds, the control linear velocity stays mostly near zero with some periods of higher values but not exceeding 0.25 m/s, while the control angular velocity shows frequent changes, exceeding π at one point (around 12 seconds).

The position pose error graph in Figure 23 and angular pose error graph in Figure 23 **are similar the DH inverse without laser tracking result**, showing the pose errors go to near zero after the initial period of control chatter of 0.8 seconds, except some small occasional changes over time. This means that the error feedback manages to bring the end effector to near target location fast with the gain values chosen and the error is maintained generally low afterwards.

The graph of control torque in Figure 24 show the requested torques staying below the saturation values, except for the initial control chatter period of 0.8 seconds and the overcorrection at the end of the CSM trajectory to get to encoder zero configuration. The highest torque value in the initial control chatter period is in τ_2 with -309.88 N-m at 0.276 seconds, **only a 0.83% difference compared to -307.34 N-m in the DH inverse without laser tracking result**.

Overall, the result with laser tracking is similar to DH inverse result without laser tracking, with an improvement in shortening the length of the final control chatter period when the robot arm tries to move to encoder zero configuration. Theoretically, the result with laser tracking should show improvement compared to the result without laser tracking. Without laser tracking, the control system uses the current theta signals and DH

forward kinematics to calculate the current pose, which could result in inaccuracy due to the imperfect knowledge of the geometry (length measurements, etc.). With laser tracking, the system can provide current pose signal directly as feedback to the error calculation block transError. However, the improvements are not very clear, possibly due to the Pose Error Gain value K correcting the errors too fast for the differences to show (between laser tracking and no laser tracking). The small differences lie in the peak values in the periods with control chatter. With laser tracking, the peaks seem to be even higher than the case without laser tracking. This could be because the laser tracking provides more accurate feedback on the actual pose of the robot arm, showing a larger error, prompting the system to correct that larger error in a more aggressive manner.

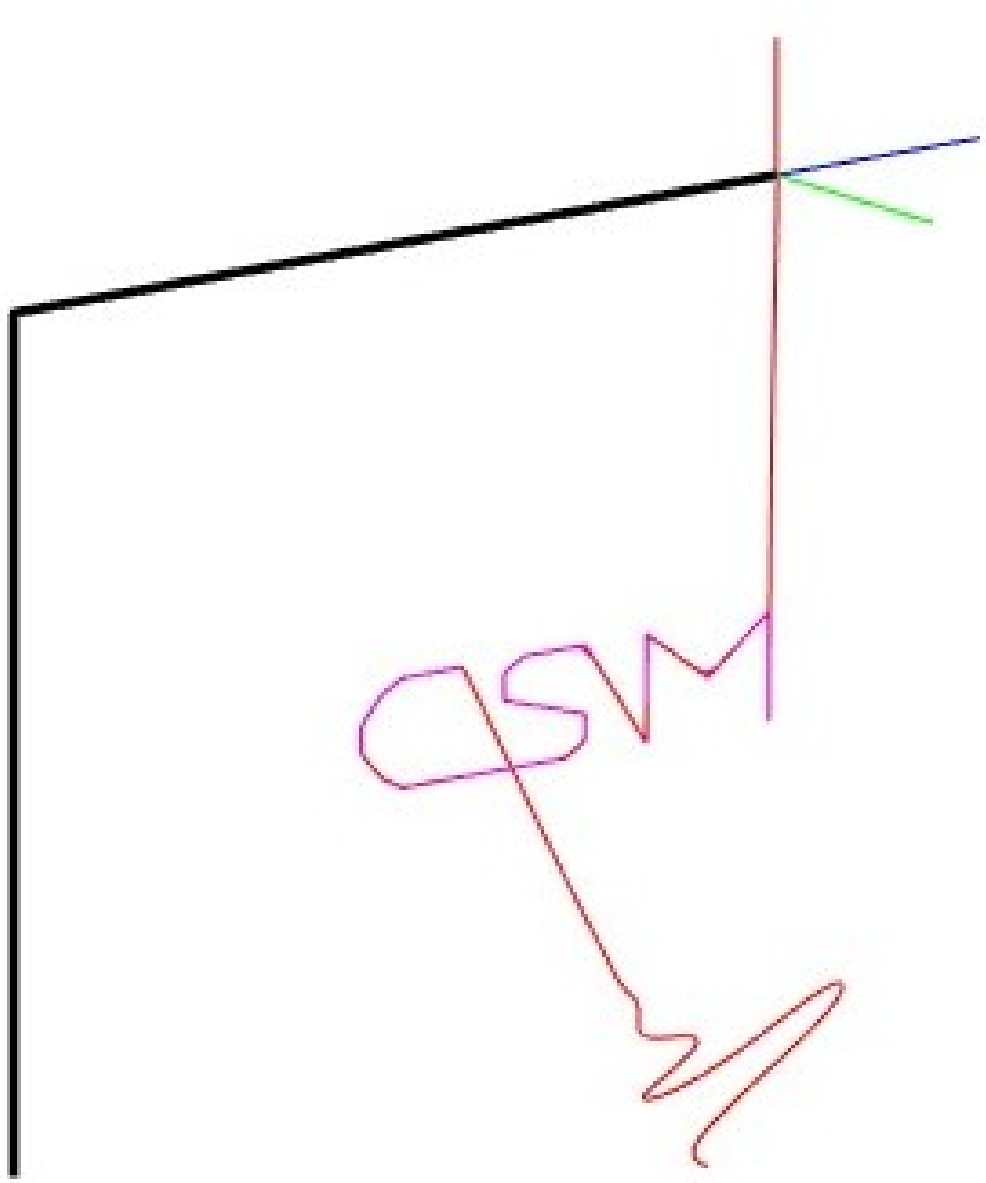

Figure 14: Graph of result trajectory traced by robot arm, with laser tracking.

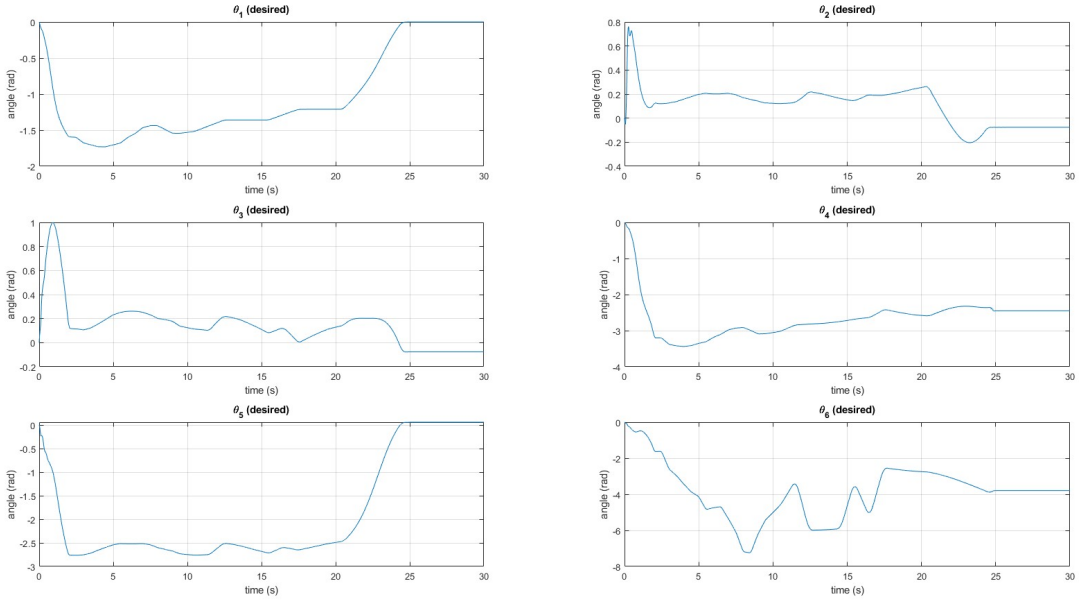

Figure 15: Graphs of desired θ's over time, with laser tracking.

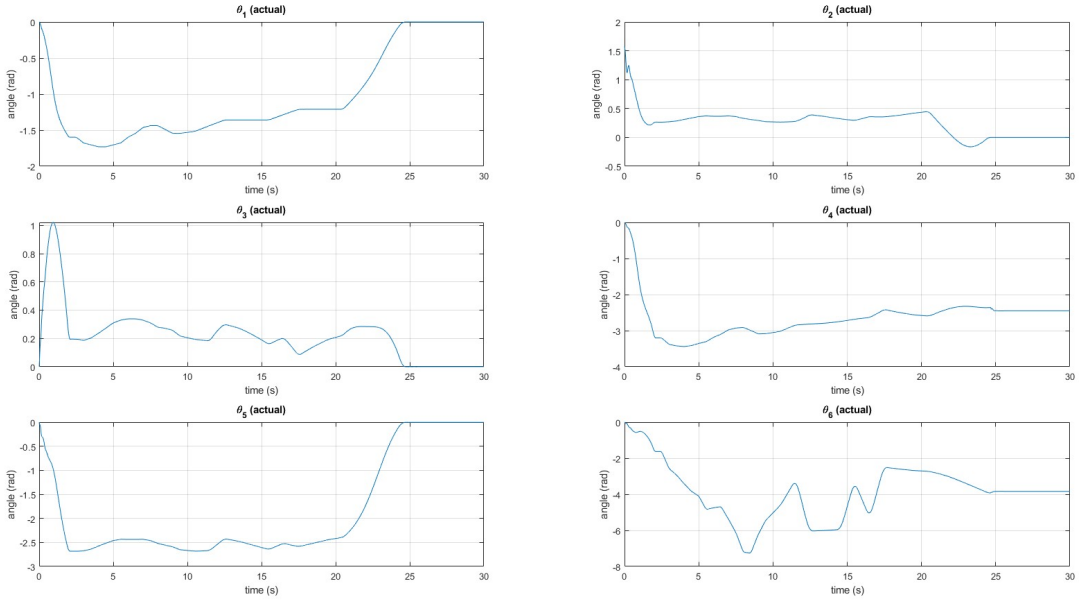

Figure 16: Graphs of actual θ's over time, with laser tracking.

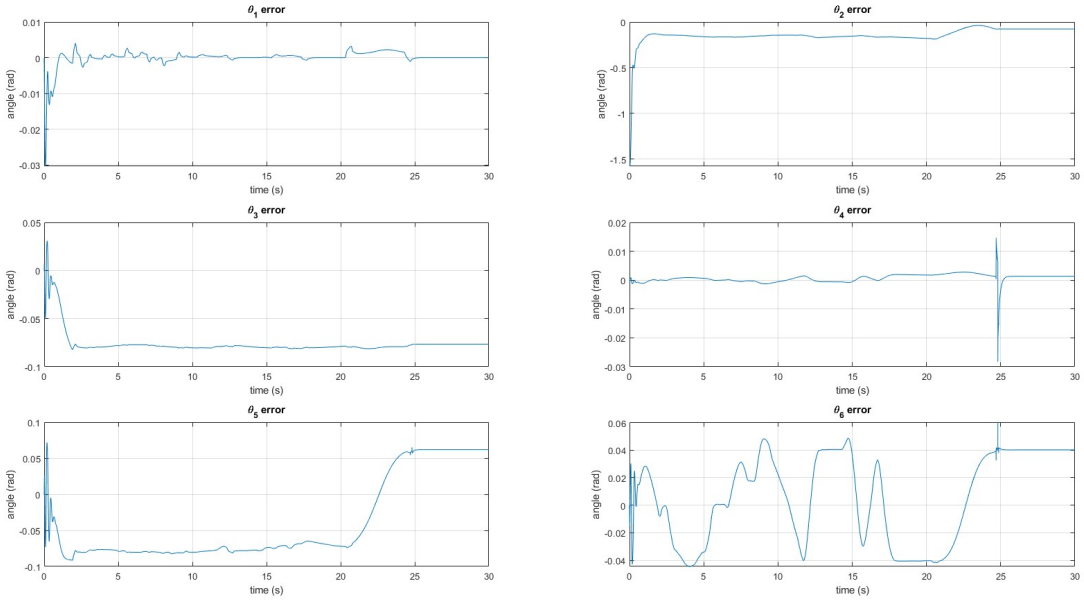

Figure 17: Graphs of θ errors over time, with laser tracking.

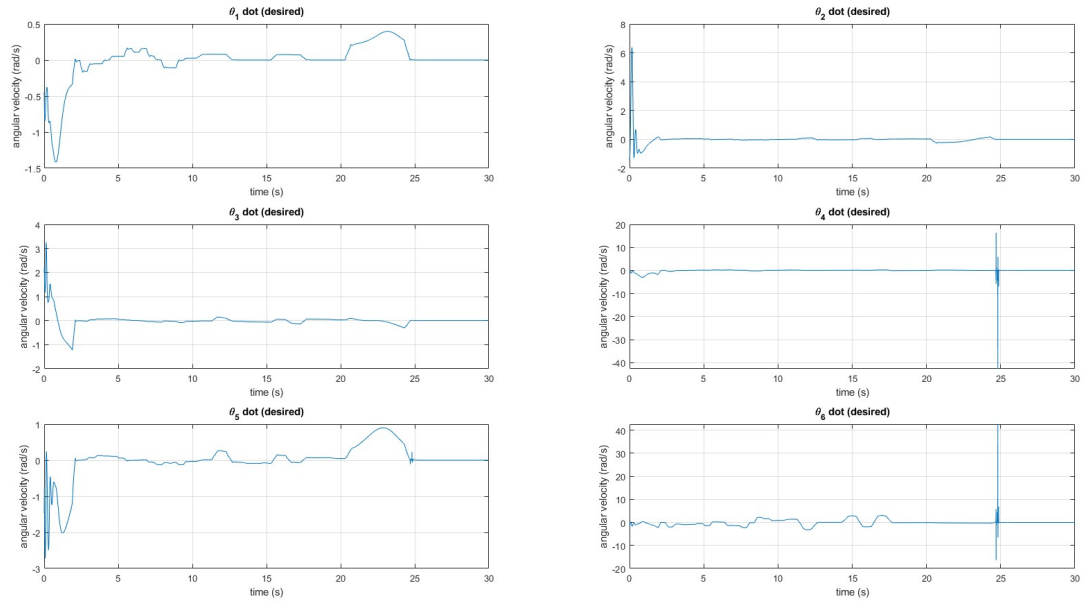

Figure 18: Graphs of desired $\dot{\theta}$'s over time, with laser tracking.

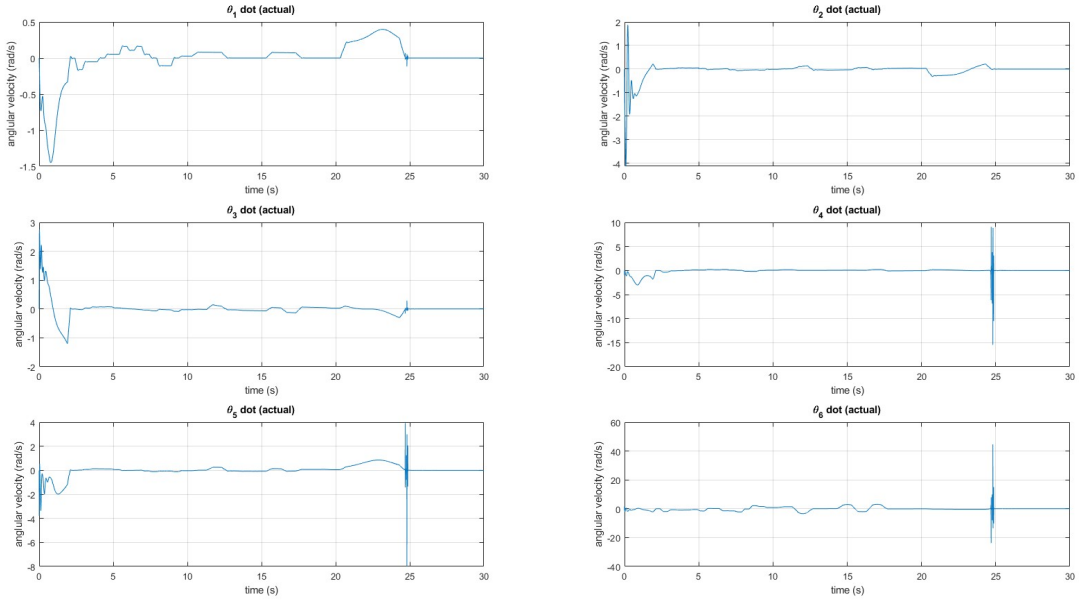

Figure 19: Graphs of actual $\dot{\theta}$'s over time, with laser tracking.

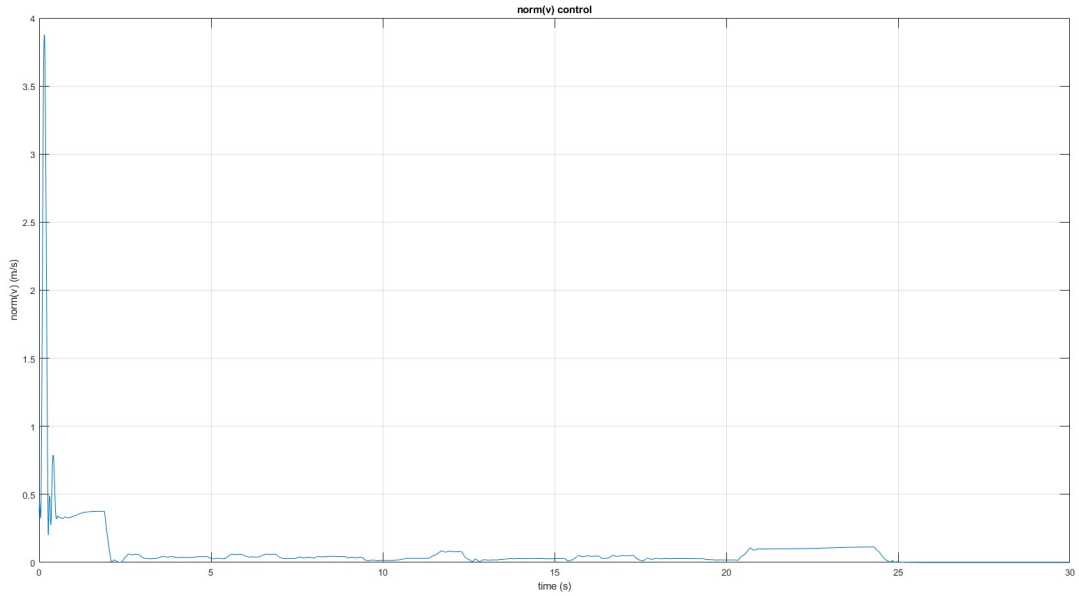

Figure 20: Graph of norm of control linear velocity over time, with laser tracking.

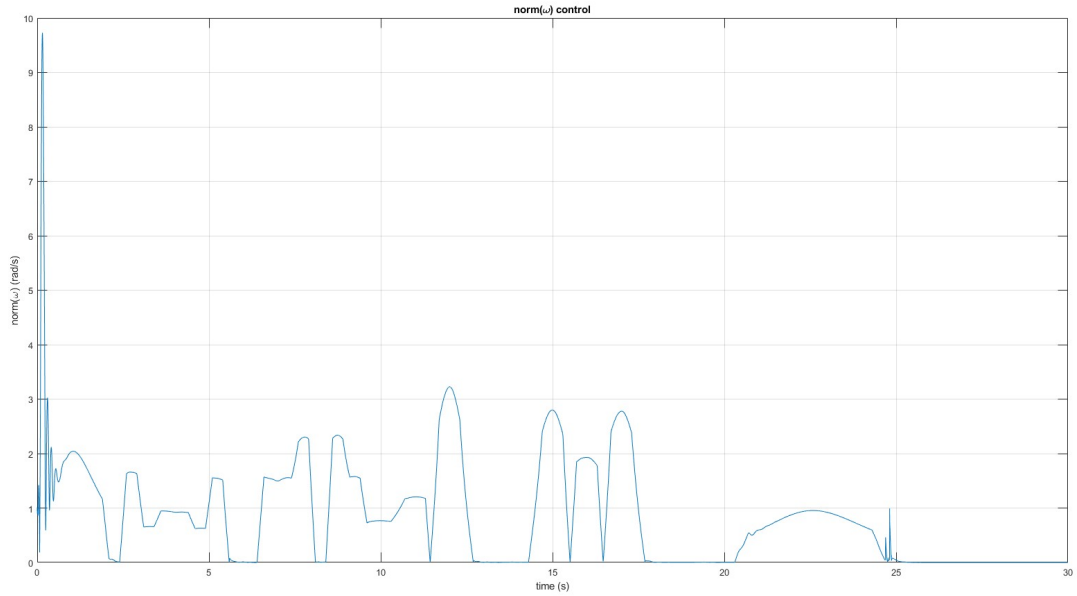

Figure 21: Graph of norm of control angular velocity over time, with laser tracking.

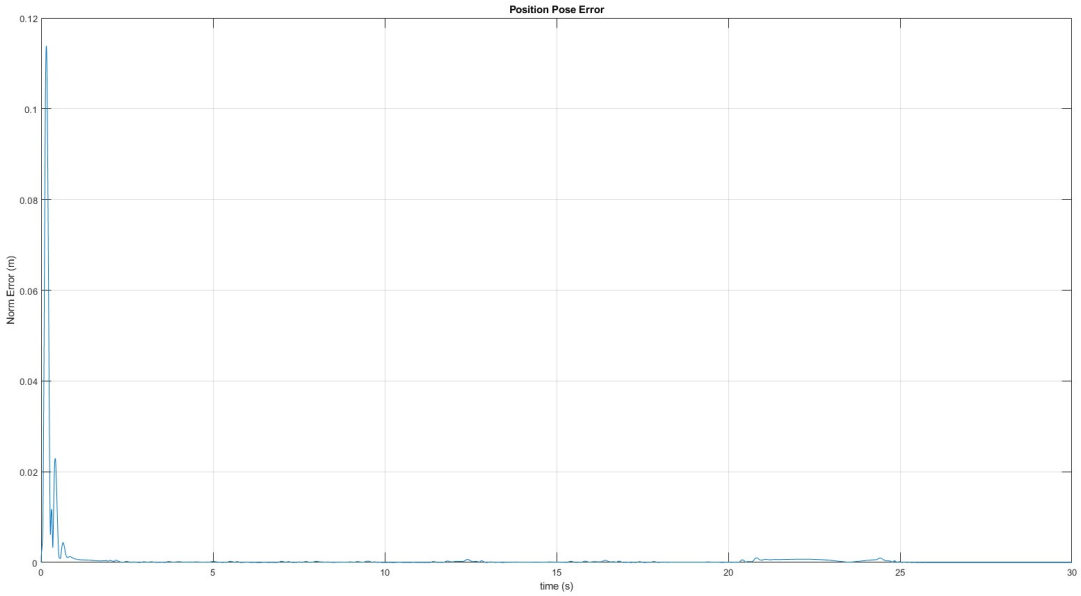

Figure 22: Graph of norm of position pose error over time, with laser tracking.

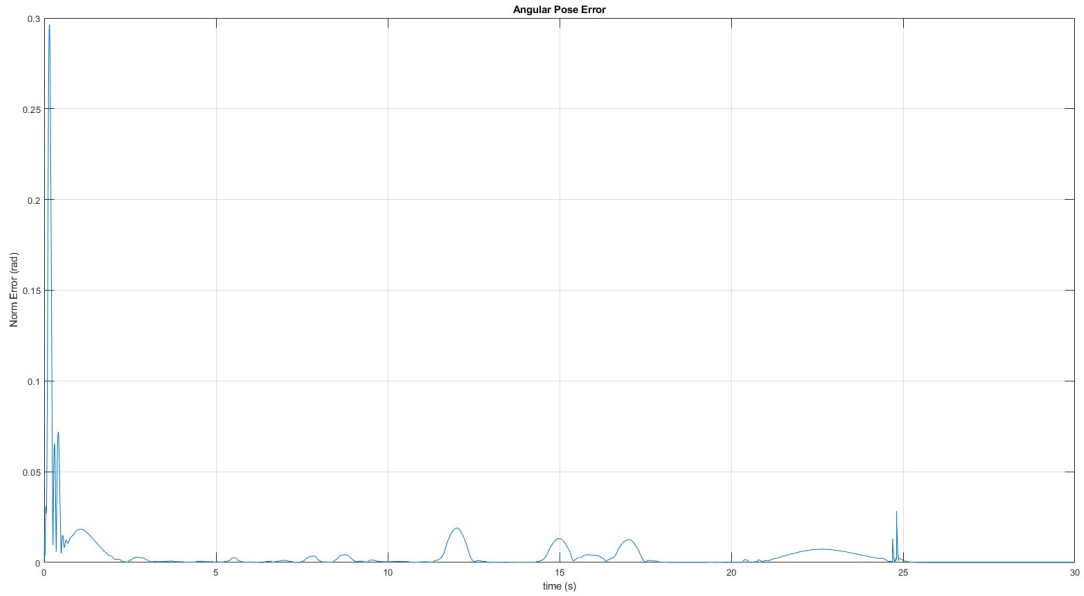

Figure 23: Graph of norm of angular pose error over time, with laser tracking.

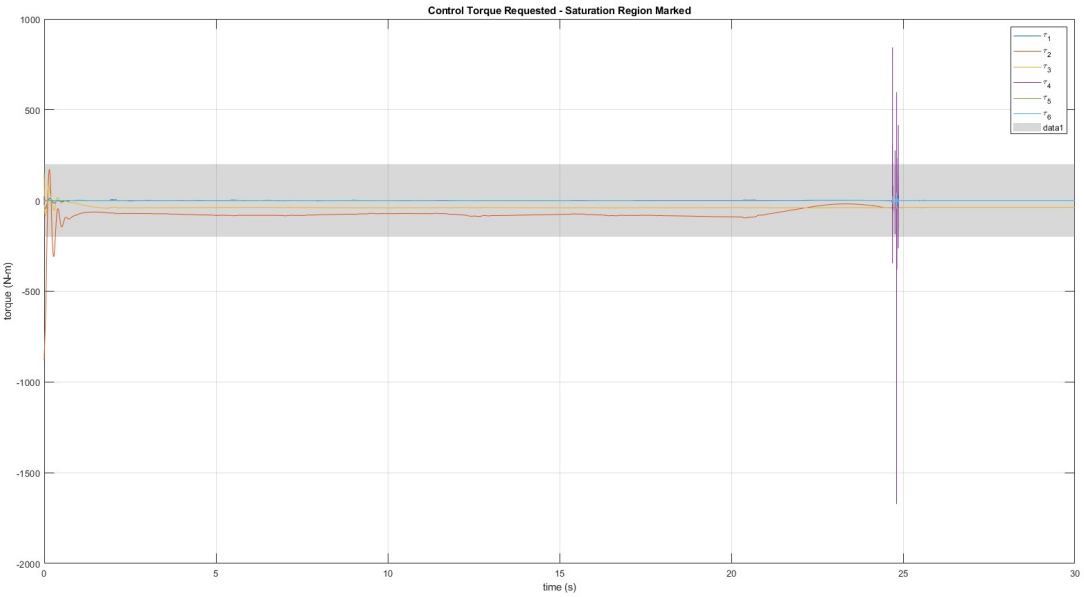

Figure 24: Graph of control torques over time, with saturation region marked, with laser tracking.

Compared to theta-only open-loop trajectory in Project Part 3, the approach in Project Part 4 (with and without laser tracking) shows improvement.

A qualitative improvement is in how smooth and accurate (close to the CSM target path) the result trajectories in Part 4 look, as seen in Figure 3 and Figure 14. Except for the initial chattering, the tool path from the starting configuration to the start of letter C and from the end of letter M to the ending configuration are also pretty straight paths. Back in Part 3, as shown in Figure 25, the tool paths to get into and out of drawing configurations had wide

curvatures, and the CSM trajectory did have discrepancies with respect to the target CSM plot.

One quantitative difference is in how the joint angles are decided. In Project Part 3, the desired theta angles were interpolated from a trajectory of theta angle values calculated from inverse kinematics, which impose angle wrapping from $-pi$ to pi. This could explain why the joint angles in Project Part 3 showed discontinuities, as in Figure 26 possibly from reaching joint limits imposed and having to wrap around. For Project Part 4, the joint angles are not necessarily confined to the range of $-pi$ to pi. For example, in Figure 16, θ_6 goes to -7.22 rad (roughly -2.3π at around 8.2 seconds). This theta is calculated from an integration operation inside the given Theta Controller block. This range of joint angle could be achieved if the wrist (joint 6) is a rotation joint with only the end-effector at the end instead of another link. In that case, the joint angle can beyond a full revolution in each direction with little risk of the links colliding with one another, hardware permitting.

Another quantitative difference is in the theta errors, such as in Figure 17 of Part 4 compared to Figure 27 of Part 3. In Part 3, except for joint 6 with a lot of rotations along the CSM trajectory, the theta errors of all the other joints stay near zero, for the most part, after 1 second. In Part 4, the theta errors fluctuate and can even stay at a non-zero value for a period of time (θ_3 staying at around -0.08 rad from 2.4 seconds to the end in Figure 17). In other words, the joint angle errors can stay at a non-zero value without being corrected to zero for significant periods of time in Project Part 4. This is because the error in Part 4 is calculated in the operational space, instead of the configuration (joint) space. The Theta Targets going into the ABB Arm Dynamics can be requested a one value, but the operational space tracking system (error calculation) may show that a slightly different value is what is actually needed for the end effector to be at the true target location. This discrepancy is due to the imperfect knowledge of the arm geometry.

These improvements compared to the Project Part 3 could be explained by the way the error is calculated and whether it is used. In Part 3, there was no error feedback. It was only an open-loop system, with the desired theta angles being fed to the Theta Controller to move the arm. Whether the end effector was at the right location was not fed back. With only the theta angles an input, the arm would make motions with wide curvatures to get to a desired configuration, instead of straight paths. For Part 4, error is fed back and is calculated directly in the operational space. This means that how far the end effector is from the target location is known and used with a gain value (Pose Error Gain) to tell the control system what velocity and direction to correct the end effector location. That is why the trajectory has more direct straight paths and follows the target CSM plot closely. This is even more accurate with laser tracking, so the system knows the more accurate location of the end effector in the operational space without relying on geometry measurements done beforehand (with uncertainties).

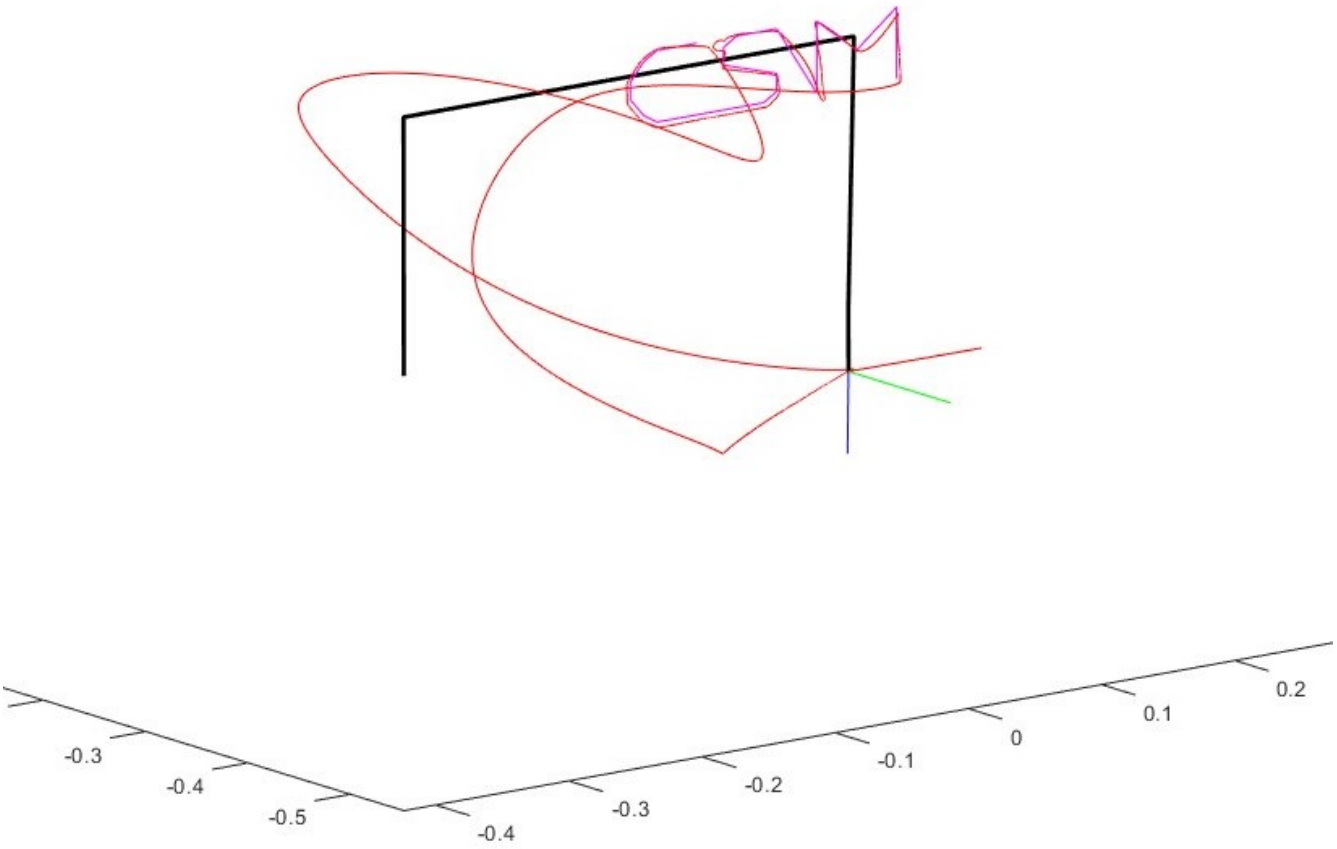

Figure 25: Graph of result trajectory traced by robot arm with theta-only open-loop control in Project Part 3.

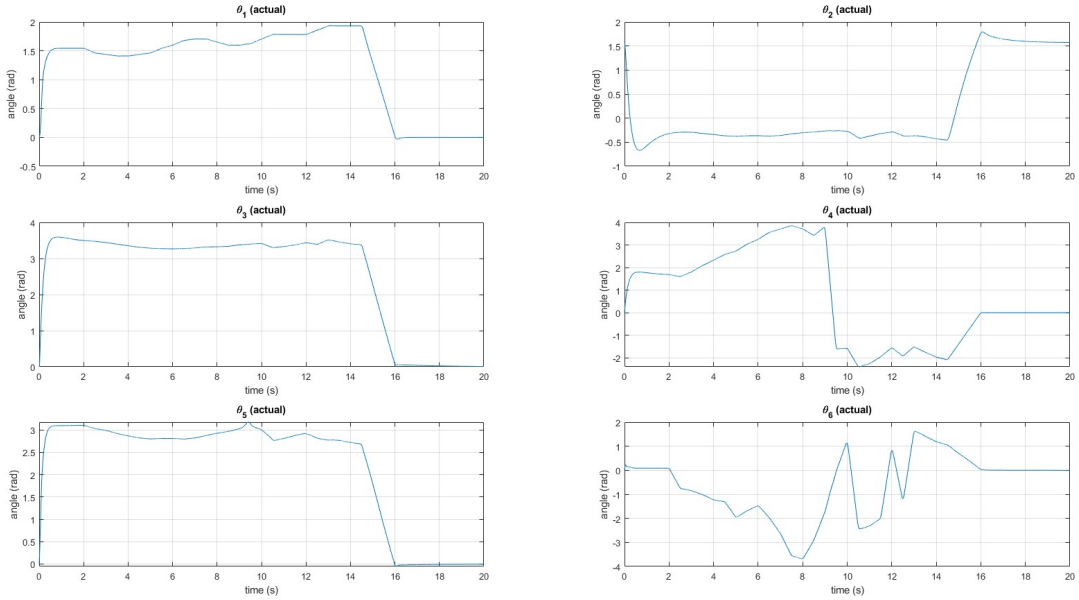

Figure 26: Graphs of actual θ's over time, with theta-only open-loop control in Project Part 3.

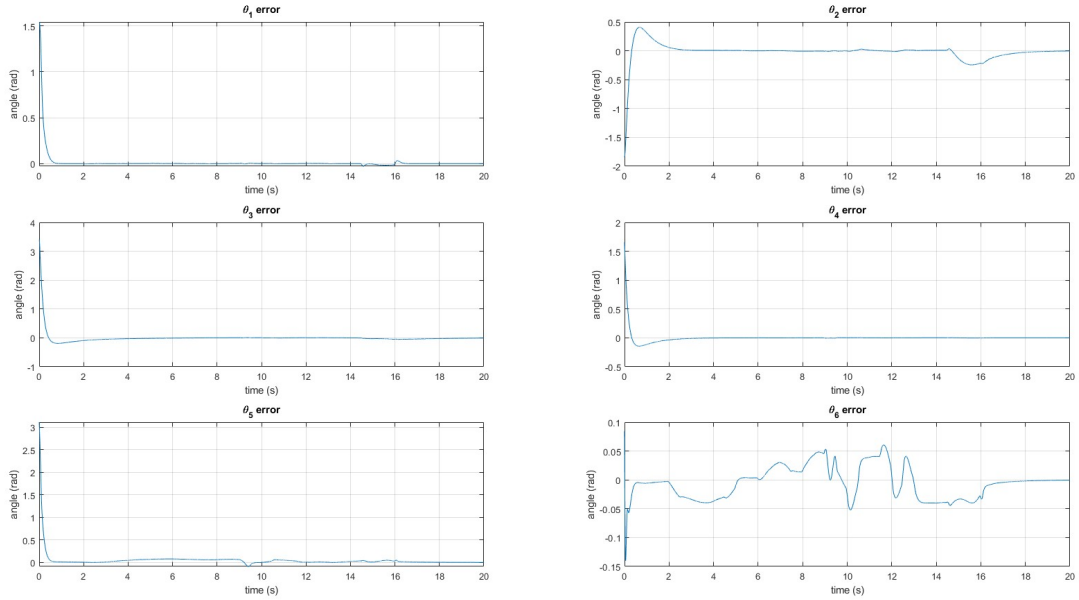

Figure 27: Graphs of θ errors over time, with theta-only open-loop control in Project Part 3.

The videos of the robot arm in action are included in the submission package as files "part4_nolaser.mp4" and "part4_withlaser.mp4".