Assignment 2 Path Tracking for a Robot Manipulator

Johan Bertil Sollenius (ID: **264180**)
Mohsin Iqbal (ID: **258626**)

19 November 2025

0.1 Implement the cost and constraint functions required for path tracking define running cost and dynamics(): define terminal cost and constraints(): Do not change the tasks' weights. Report and discuss the resulting trajectories.

After implementing the cost and constraint functions, The solver managed to find a solution. Thanks to imposing some of the hard constraints such that the end effector needs to stay on the path y(q) = p(s) and the terminal constraint sN = 1. The results in Figure 1 clearly show that the end-effector follows the infinity-shaped path.

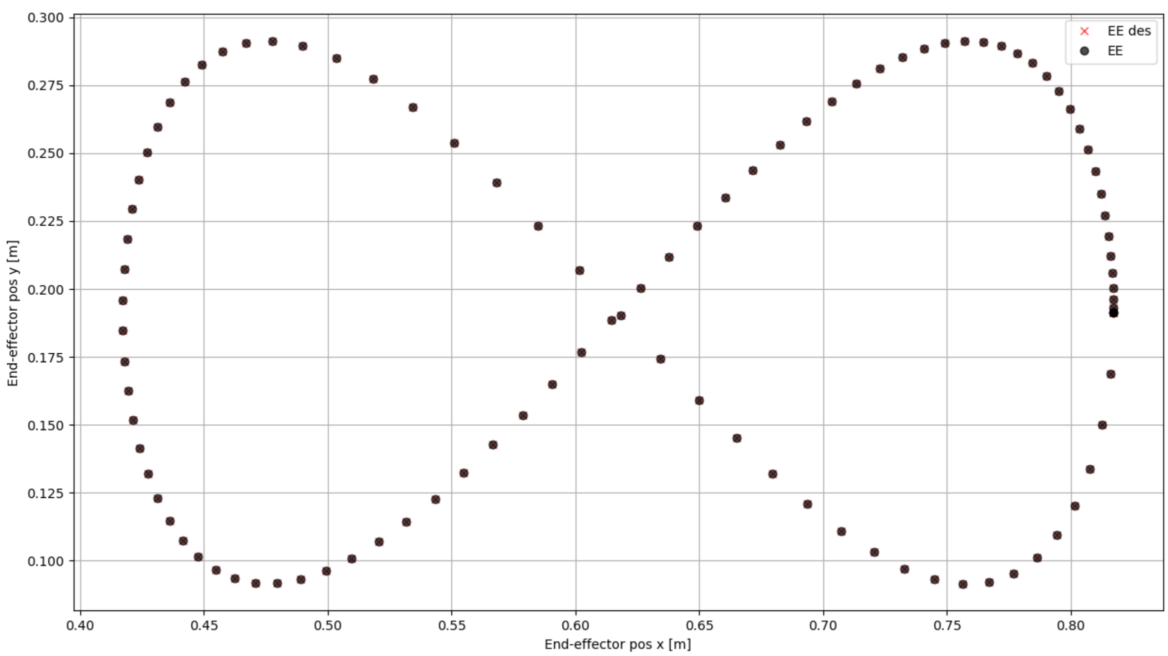

Figure 1: (Task 0.1) Position of end effector during Path Tracking

However, The movement is not "harmonic". For example, Looking at the joint torques in Figure 2 the torques applied for each joint, They are all within the reasonable range which is expected since they obey the limits we set. But, the joint 1 seem to carry most of the load followed by 2 and 0. While the other joints have almost no torque applied. This is expected, since the optimization only enforces path tracking and torque limits, without any cost encouraging a more uniform torque distribution or a preferred joint posture. Additionally, What is seen especially by the torque applied to joint 0 is that it keeps on adding force without stopping/reversing. Which suggests it continued to have a notable velocity after finishing the simulation.

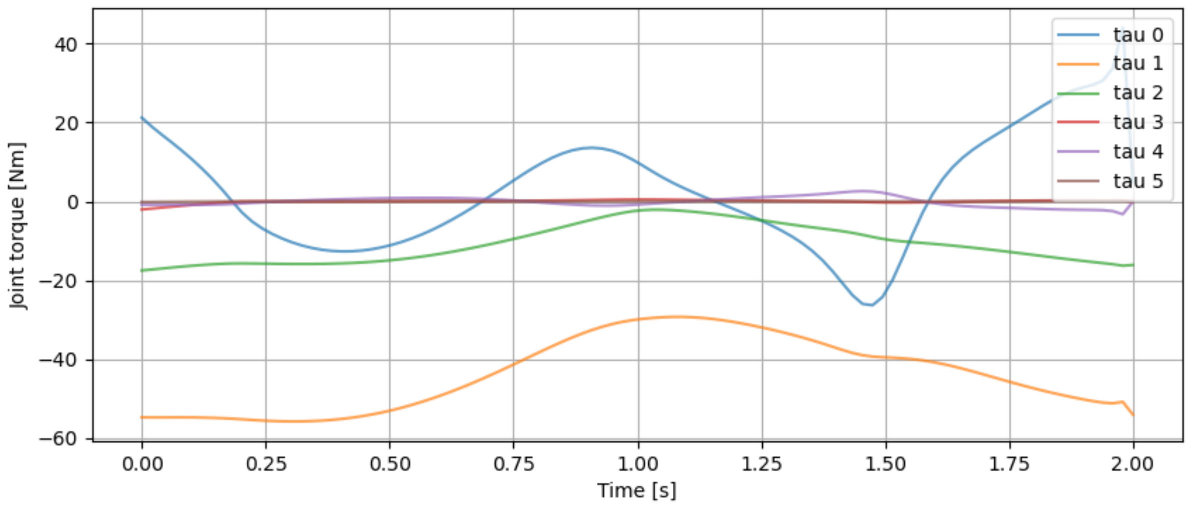

Figure 2: (Task 0.1) Torque applied to each joint Path Tracking

This unsymmetric movement becomes clearer by looking at Figure 3. At the end, the joints do not stop at zero velocity as they started from in the beginning. Since we are not enforcing a “nice” final state, such as a terminal velocity constraint, the optimizer has no incentive to end the motion smoothly. In a real application, following a path would require all joints to stop after completion.

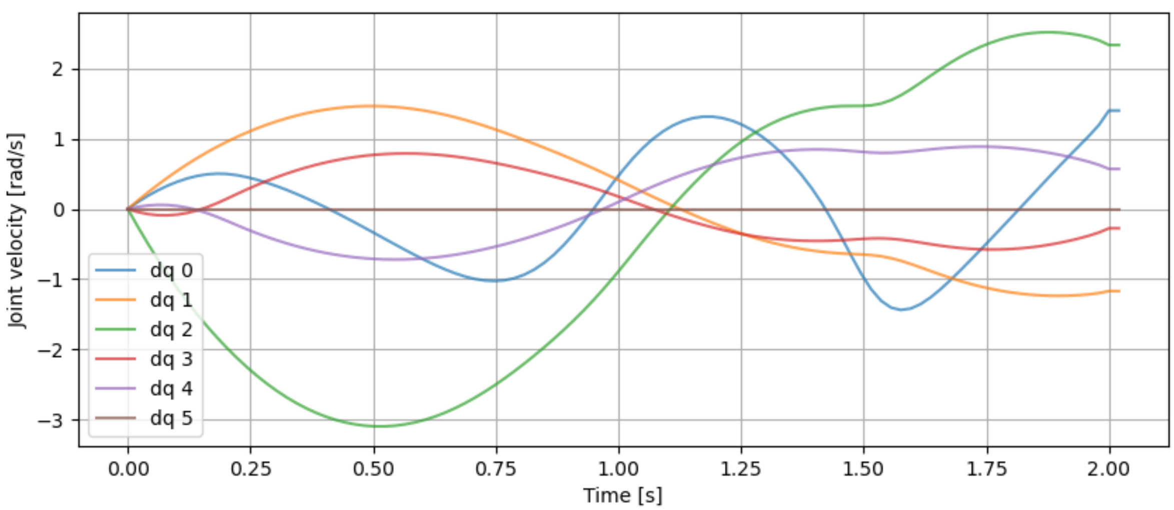

Figure 3: (Task 0.1) Joint velocities Path Tracking

The same is observed when looking at the speed at which the end-effector traverses the path, as it is clear that the speed varies and is non-constant, which can be problematic in applications where maintaining a constant or specific speed is important (see Figures 4 and 5).

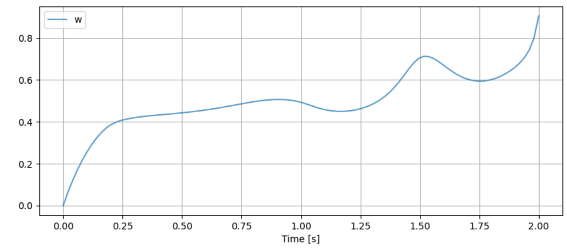

Figure 4: (Task 0.1) Speed variable w(t) Path Tracking

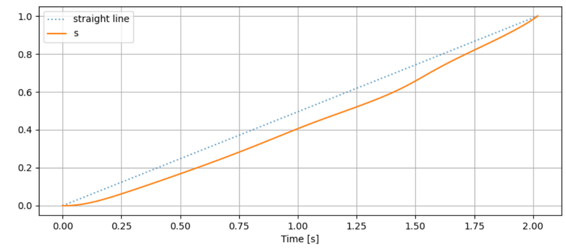

Figure 5: (Task 0.1) Progress variable s(t) Path Tracking

To summarize, the path is tracked successfully according to the defined constraints, however the motion itself does not appear "harmonic" or symmetric, since there is no cost encouraging the joints to share the effort or to end where they started.

0.2 Implement a terminal cost that penalizes the distance between the initial and final state with a weight wfinal = 1. Report and discuss the resulting trajectories.

After implementing a terminal cost that penalizes the distance between initial and final state, The joint velocities were more close to stopping at 0. (initial state). Same for the joint angles, trending to go back to its original state. (see Figures 6 and 7).

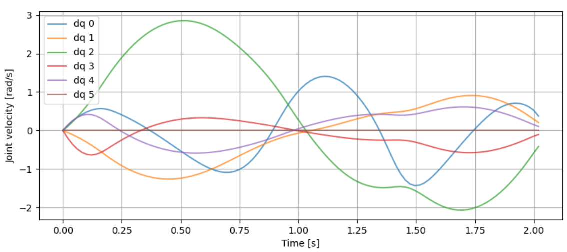

Figure 6: (Task 0.2) Joint velocities Path Tracking with Terminal Cost

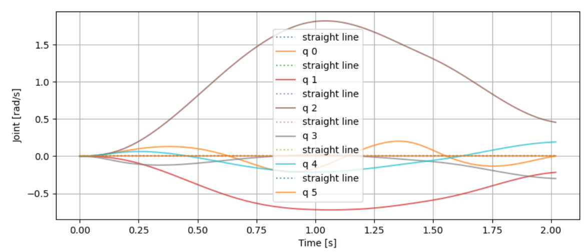

Figure 7: (Task 0.2) Joint angles Path Tracking with Terminal Cost

What is interesting, however, is that the progress speed variable $w(t)$ slows down toward the end of the motion instead of speeding up. Our conclusion is that the interaction between the terminal cost, which encourages the final state $x_N \approx x_0$, and the terminal constraint $s_N = 1$. These two objectives are partially conflicting: the robot must both complete the path and return close to its initial configuration. As a result, the solver chooses to complete most of the path earlier by increasing $w(t)$ in the middle of the motion and then slows down near the end to satisfy the terminal cost.

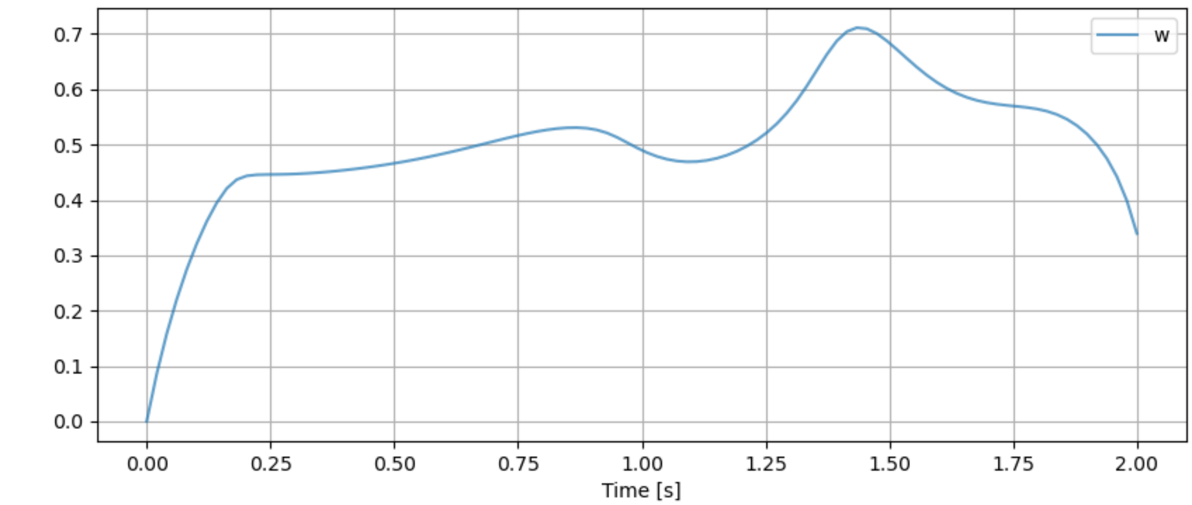

Figure 8: (Task 0.2) Speed variable w(t) Path Tracking with Terminal Cost

After adding the terminal cost, the torque profiles become noticeably larger and more "mountain-like" compared to the smooth, low-amplitude torques in the prior tracking case without the terminal cost. We believe we can contribute this due to the fact that adding a terminal cost tightens the problem, we are forcing the optimizer to work harder to satisfy more goals. So torques increase in magnitude and develop clear peaks.

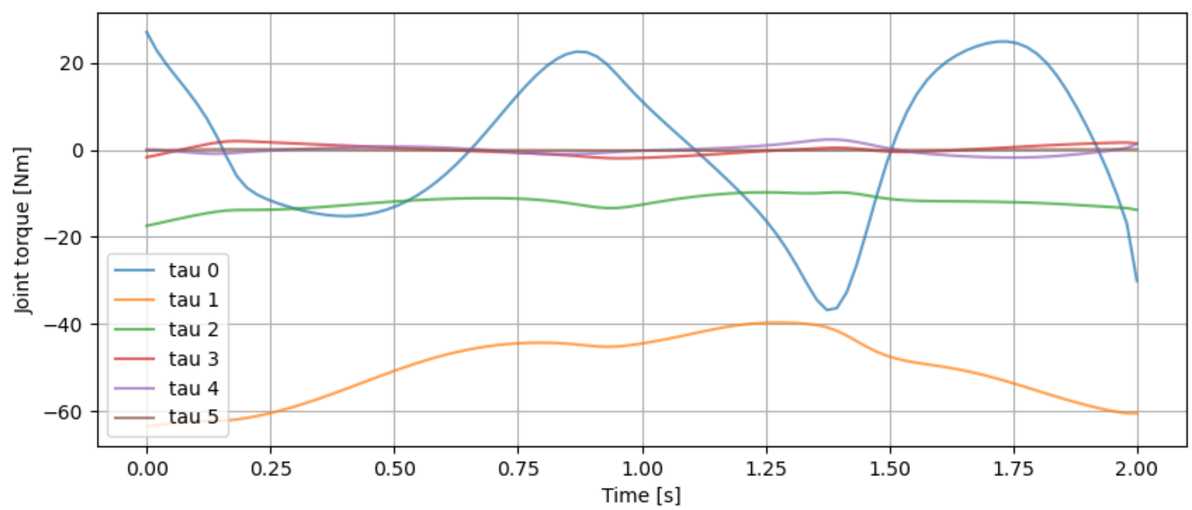

Figure 9: (Task 0.2) Torque applied to each joint Path Tracking with Terminal Cost

Also, Since the optimal control problem becomes more constrained after adding the terminal cost, the solver time increased as expected, from approximately $\sim 20s$ to $\sim 100s$ in our setup.

0.3 Why is the trajectory tracking achieved using a cost instead of a constraint? Would the same results be obtained if constraints (instead of costs) were used? – Report and discuss the resulting trajectories (highlighting the difference between these and the one obtained via path tracking)

Note: For Task 0.3, the terminal cost from Task 0.2 was kept

After implementing trajectory tracking it was clear that the end-effector followed the path closely but not exactly as seen in Fig 10. Which is expected since we are now using a cost instead of a constraint as in path tracking.

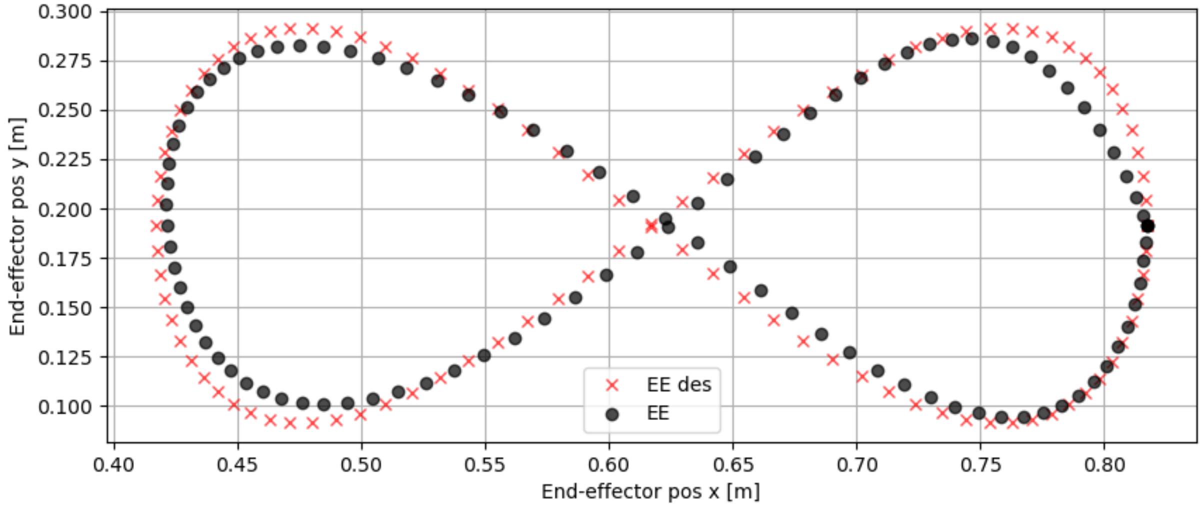

Figure 10: (Task 0.3) Position of end effector during Trajectory Tracking

What is also observed that the torque applied to the joints seem to be much smoother than path tracking. As seen in Figure 11. This is normal, since we are allowing the end-effector to deviate a little bit from the path by having a cost instead of a hard constraint. So it doesn't have to produce more agressive acceleration.

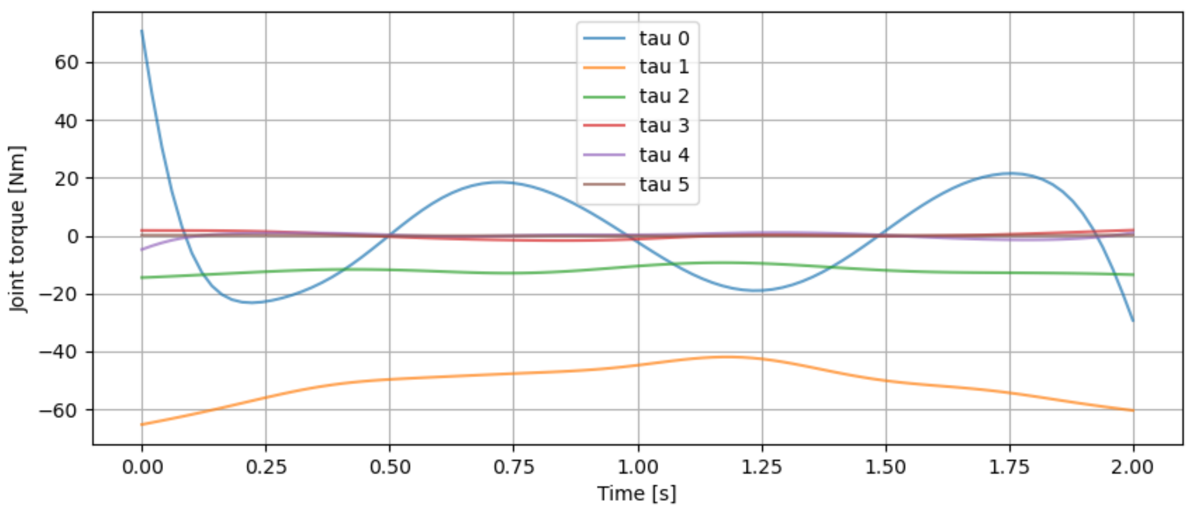

Figure 11: (Task 0.3) Torque applied to each joint Trajectory Tracking

A clear difference observed was that the speed variable w as seen in Figure 12 was more or less constant during the trajectory tracking, compared to path tracking. This behavior was expected. When path tracking the progress variable S(k) was a free decision variable where only the geometric constraint eepos(k) = p(S(k)) was enforced. Which let the optimizer adjust the motion along the path, resulting in a varying speed. In contrast, with trajectory tracking we fix the path parameter to s(k) = k/N where N is the total number of time steps. This defined a time paramterized reference, meaning that the robot is expected to reach a specific point on the path at each time step (uniformily). Which explains the observed behaviour of w and s.

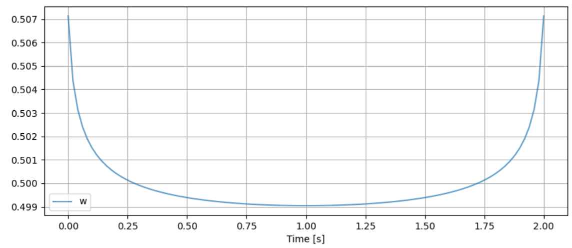

Figure 12: (Task 0.3) Speed variable w(t) Trajectory Tracking, shows a constant speed

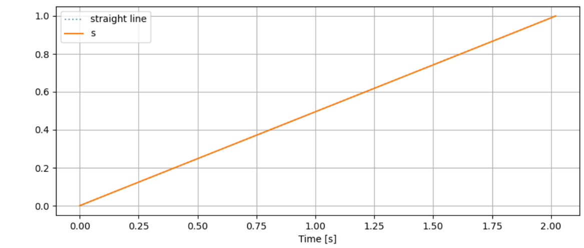

Figure 13: (Task 0.3) Progress variable s(t) Trajectory Tracking, shows a linear progress.

When it comes to the joint velocities & angles we could not notice any noteworthy difference. Between path tracking and trajectory tracking.

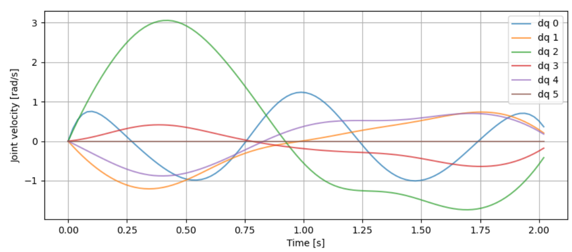

Figure 14: (Task 0.3), Joint velocities Trajectory Tracking

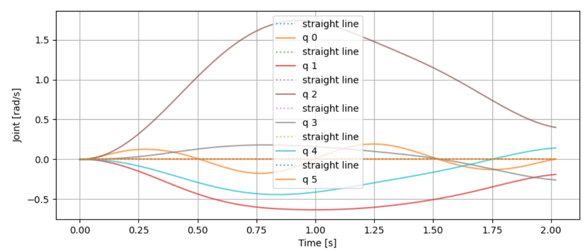

Figure 15: (Task 0.3), Joint angles Trajectory Tracking

We tried a hard constraint, where the end-effector had to stay on the path precisely. This resulted in an infeasible solution for the solver. Our conclusion is that a hard constraint in trajectory tracking eliminates most feasible motions, since the robot must simultaneously satisfy dynamics, torque limits, joint limits, and exact tracking. In practice, this typically leads to an infeasible problem, which is what we observed when attempting to impose trajectory tracking directly as a constraint.