Optimization and Learning for Robot Control: Assignment 1

Luca Sartore - 256154

Contents

1 Test on Steady Robot

1.1 Q1: Baseline

The baseline controller (the one with the default weights) shows decent tracking performance but is not perfect. Errors on the center of mass along the x and z axes are minor (at most 1 mm and 0.3 mm, respectively), but the error along the y axis (the one with the sine wave applied) is larger—about 3 cm at the peak, as shown in Figure 1.

The most likely cause of the tracking error is that there are model inaccuracies resulting in unknown forces that are not compensated for by the controller. Common sources of such errors include joint friction and damping. Other possible causes are imprecise motor controllers or inaccuracies in the robot model (e.g., joint mass and inertia matrices). However, since this experiment is conducted in simulation, these modeling errors should be minimal.

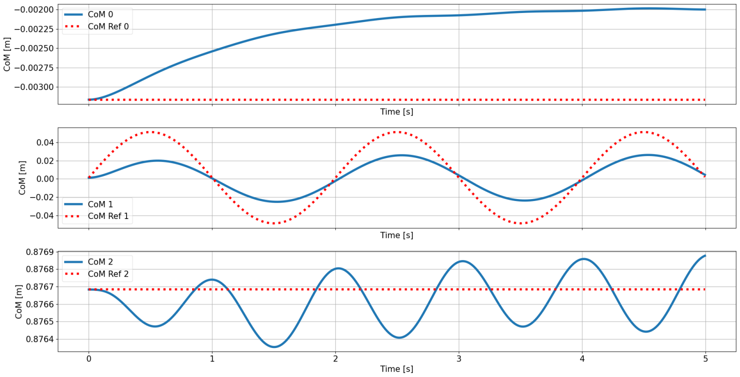

Figure 1: Baseline controller performance.

1.2 Q2: Tuning the Parameters

1.2.1 Increasing the Weight of the Center of Mass Task

In the first scenario (A), the weight of the center of mass task was increased from 1 to 10. The resulting performance is shown in Figure 2.

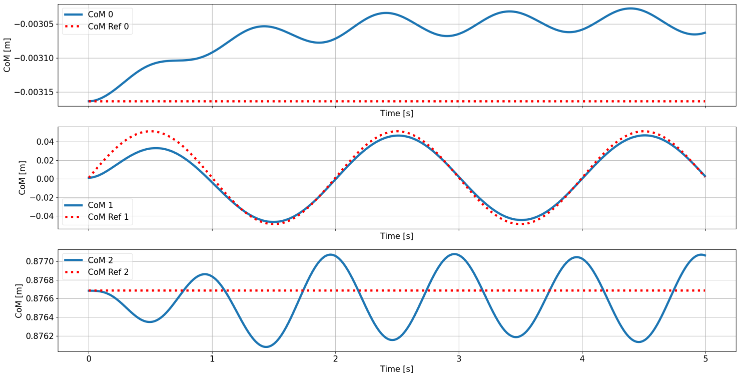

Figure 2: Effect of increasing the center of mass task weight.

1.2.2 Increasing the Proportional Coefficients of the Center of Mass Task

In the second scenario (B), the proportional coefficient of the center of mass task was increased from 10 to 100. Results are shown in Figure 3.

1.2.3 Differences Between the Two Scenarios

Initial Behavior. Scenario A performs worse than scenario B at the beginning, showing a larger gap between the desired and actual positions. We focus only on the y axis (CoM1), since there are no significant differences along the other two axes.

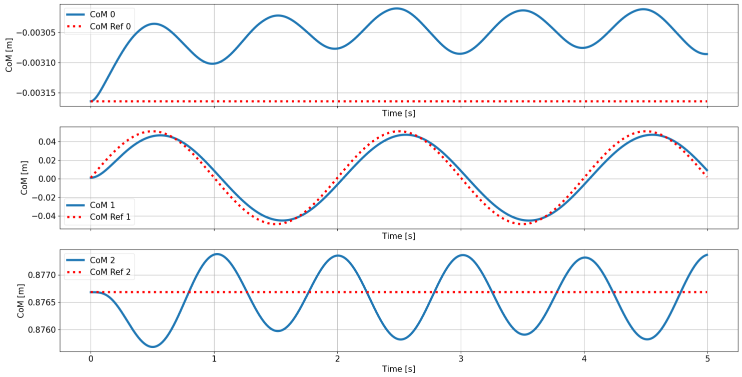

Figure 3: Effect of increasing the proportional gain of the center of mass task.

Long-Term / Steady-State Behavior. After a few seconds, scenario A begins to perform better. The difference is minor, but scenario A keeps the actual trajectory closer to the desired one, while scenario B shows a small delay between the desired and actual trajectories.

The reason scenario A performs poorly initially is likely due to a weak control signal. As shown in Figure 4, the robot starts with zero velocity, yet an immediate "high" velocity is requested. Controller B experiences a large initial error, which is multiplied by a high proportional gain, allowing it to reach the target more quickly. Controller A, with a lower proportional gain, reacts more slowly.

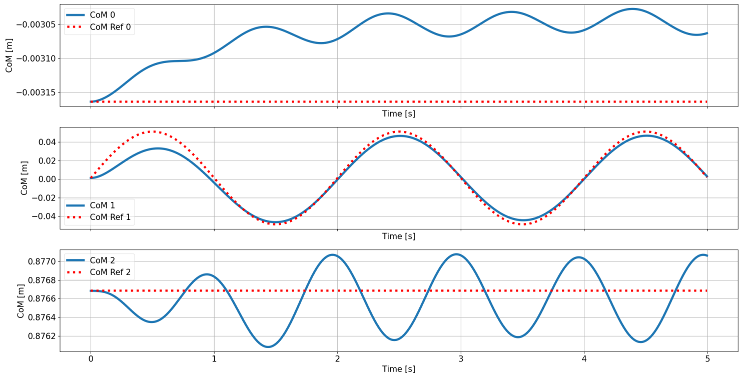

Figure 4: Velocity profiles

In a real-world scenario, we would aim to provide a smoother control signal and start from zero velocity, in which case controller A would be preferred. However, if sudden changes in velocity were common, controller B might perform better.

1.3 Q3: Increasing Sine Wave Frequency

In this third scenario, we increase the sine wave frequency to 1 Hz and compare the two controllers (A and B).

1.3.1 Controller A Performance

Figure 5 shows the center of mass tracking for controller A. Tracking performance is poor, and divergence between the desired and actual trajectories increases over time.

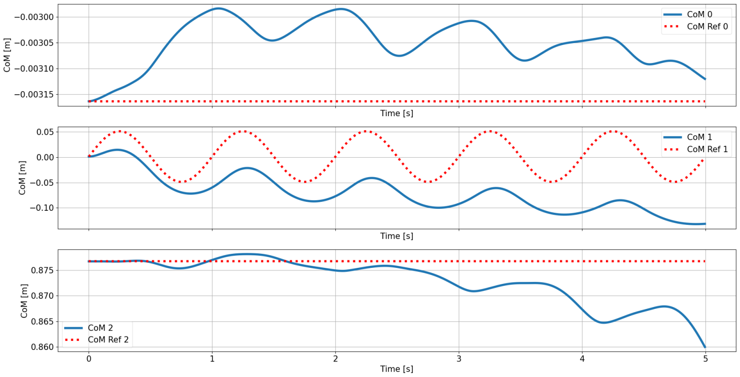

Figure 5: Center of mass tracking for controller A at 1 Hz.

Examining the robot's position at the final step (Figure 6), we see that it is tilted sideways. This occurs because increasing the center of mass task weight decreases the relative importance of the postural task. The controller thus ignores maintaining an upright stance, leading to instability and poor tracking. Over time, the robot eventually collapses.

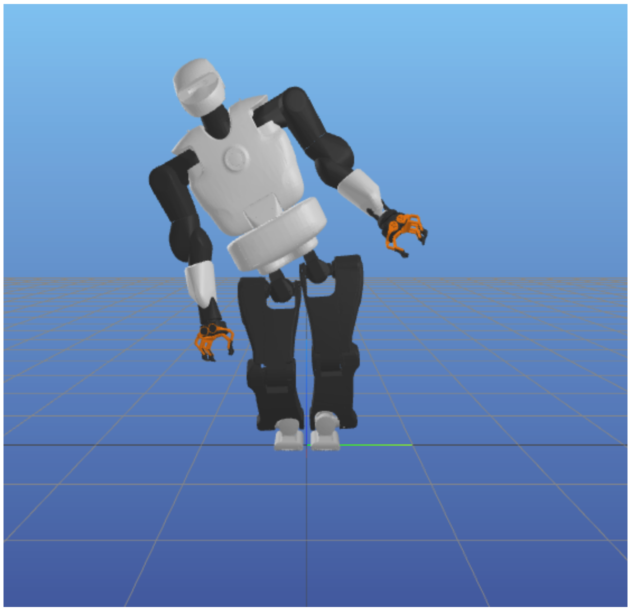

Figure 6: Robot posture for controller A at the final step.

In this case, controller B would clearly be preferred.

1.3.2 Controller B Performance

Controller B also performs poorly, as shown in Figure 7. The difference between desired and actual trajectories is larger compared to the lower-frequency test. This is expected, since increasing the sine wave frequency increases the required velocity, which in turn demands higher torques. If the proportional coefficient remains constant, a higher error is required to produce the necessary torques.

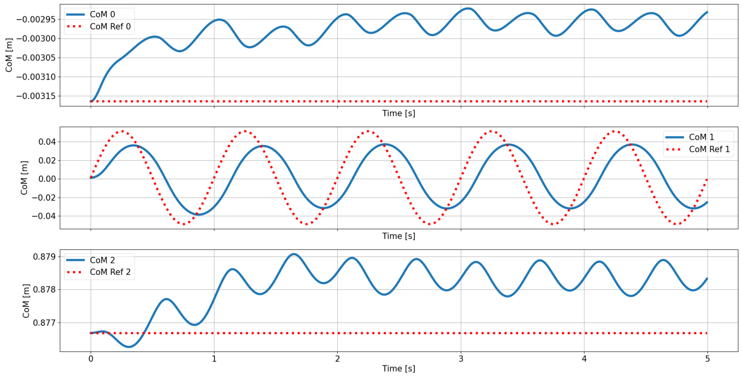

Figure 7: Controller B tracking performance at 1 Hz.

2 Test on Walking Robot

2.1 Q1: Experiment on 15 cm Steps

2.1.1 Tuning the Postural Task Weight

Initially, the robot was unable to stay upright and tilted sideways before falling—similar to controller A in the previous tests. Increasing the postural task weight from zero to somewhere between 10^{-2} and 10^{-3} stabilized the robot. A final value of 10^{-3} produced the tracking shown in Figure 8, which demonstrates excellent performance.

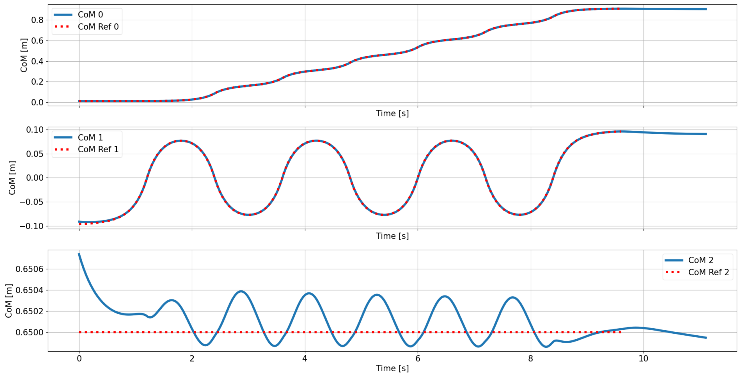

Figure 8: Tracking performance for walking robot with 15 cm steps.

2.1.2 Effect of Different Tasks on Robot Behavior

To evaluate the effect of each task on walking performance, each task weight was set to zero individually, and the resulting behavior was observed.

Center of Mass Task. When the center of mass task weight is set to zero, the robot tilts forward and eventually falls. This is expected—if the center of mass moves outside the foot's projection area, the system becomes unstable.

Angular Momentum Task. Disabling the angular momentum task caused no significant change in behavior. This task ideally keeps the robot's angular momentum near zero (e.g., moving the left arm forward when the right leg moves forward). At low walking speeds, however, this effect is negligible. Increasing the task weight excessively led to exaggerated arm movements and reduced stability.

Foot Motion Task. This task is essential: without it, the robot receives no command to move its feet and therefore cannot walk.

Foot Contact Task. When this task is disabled, the robot sinks into the ground at the start of the simulation. This task ensures realistic ground contact, preventing foot penetration into the terrain—hence its high default weight.

Joint Posture Task. This task encourages the robot to maintain a natural posture. Disabling it results in unnatural limb movements and eventual falling. Maintaining a natural posture helps keep joints away from singularities and improves overall stability.

2.2 Q2: Experiment on 30 cm Steps

In the first run with 30 cm step size, the robot tilted backward (as if doing the limbo) and fell. To address this, the postural task weight was gradually increased from 10^{-3} up to 5×10^{-2}. This adjustment prevented backward tilting, but the robot began falling forward during the last step. Finally, increasing the center of mass task weight from 1 to 10 resolved the issue.