ROB-UY 2004 Final Project

Asfaq Fahim

May 4 2025

Methodology

Before I got to the final controller code, I tried solving the problem at hand by writing a purely task-space controller. From the given information, I was able to determine the position of the red cube, blue cube, and green tower with respect to the robot's base. From these positions, I determined phases that the robot should follow, which are basically 10 positions where the robot moves above or near the objects. I decided to move the blue cube on top of the green tower first because it seemed more difficult than moving the red cube to the bowl. To program the gripper, I made the phases a dictionary where the values is a list of the positions and a boolean value. This boolean value would determined whether the gripper should open or close based on the current phase it is in. Using the same logic as Lab 5's task space controller, I used the `compute_trajectory()` to determine the end-effector positions and velocities for the 10 phases and used the Jacobian transpose to map the end-effector velocities to joint velocities. To compensate for the gravity, I used the robot's `get_gravity()` function, which returns the joint gravity torques, and added it to the total joint torques. Finally, I sent the total torques to the robot. Below are the position and velocity versus time of the end-effector.

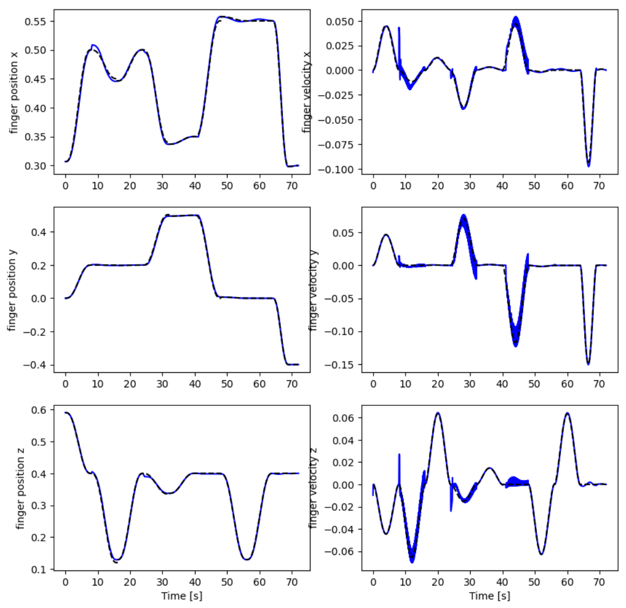

Figure 1: End-effector Position and Velocity vs. Time

The measured positions and velocities follow the desired positions and velocities as shown in Figure 1. The PD gains I chose were an array of 7 where $P = 120$ and $D = 100$. Only 3 of the values were used in the task-space while the full array was utilized for the joint-space controller.

From the initial task-space controller, I noticed a major problem where the gripper arm would not orient or align with the blue cube, which would result in the cube slipping off or the gripper unable to grasp the cube at all. This is where I implemented the joint-space controller. I chose three positions: grabbing the blue cube, placing it on top of the green tower, and grabbing the red cube, where I would choose specific joint values for joint 5, 6, and 7 to ensure that the gripper is aligned with the cube. Since I knew the phases where I would need to orient the gripper arm, I coded the loop such that at certain phases, (namely phase 1, 3, and 5) if the difference in the desired and measured position is close to 0, which means the robot is approaching the next phase, the specific joints would have different joint velocities while all other joint velocities were zero. For these phases, I needed to use inverse geometry to ensure that all the joint angles fit to the end-effector trajectory. Lo and behold, this made grabbing the blue cube smooth and placing it in the same orientation as the green tower.

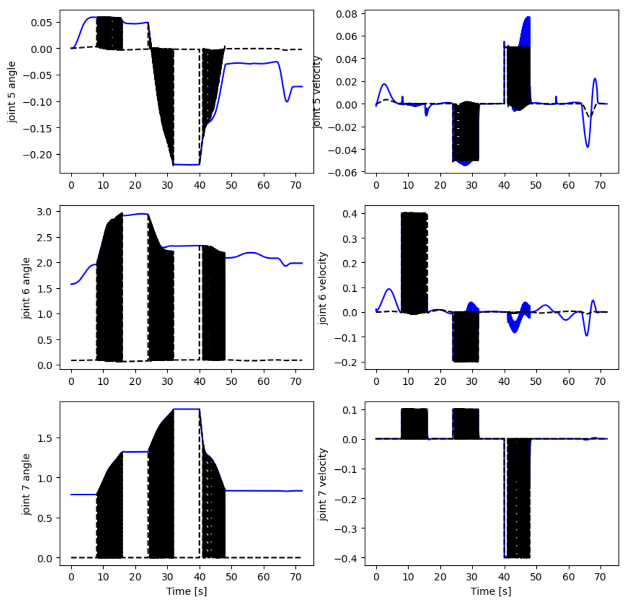

Figure 2: Joint 5, 6, 7 Angle and Velocity vs. Time

Figure 2 shows the angles and velocities of Joint 5, 6, and 7. Based on the plots, the measured values are closely following the desired values. However, the dark spots, which I am unsure about, may be due to scaling issues with the plot which cannot show the rapid change in the values.