ROB2004 Final Project

Pick and Place Objects

Akram Khairi

05.15.2023

TASK

Implement a controller using the code skeleton below in order to get the robot to go and pick up the red blocks and drop them in the green bowl. The bowl position is $(-0.3,0.55,0.65)$ (in spatial frame coordinates) and the blocks positions are $(0.35,0.58,0.65)$ and $(0.15,0.67,0.65)$ respectively.

METHODOLOGY

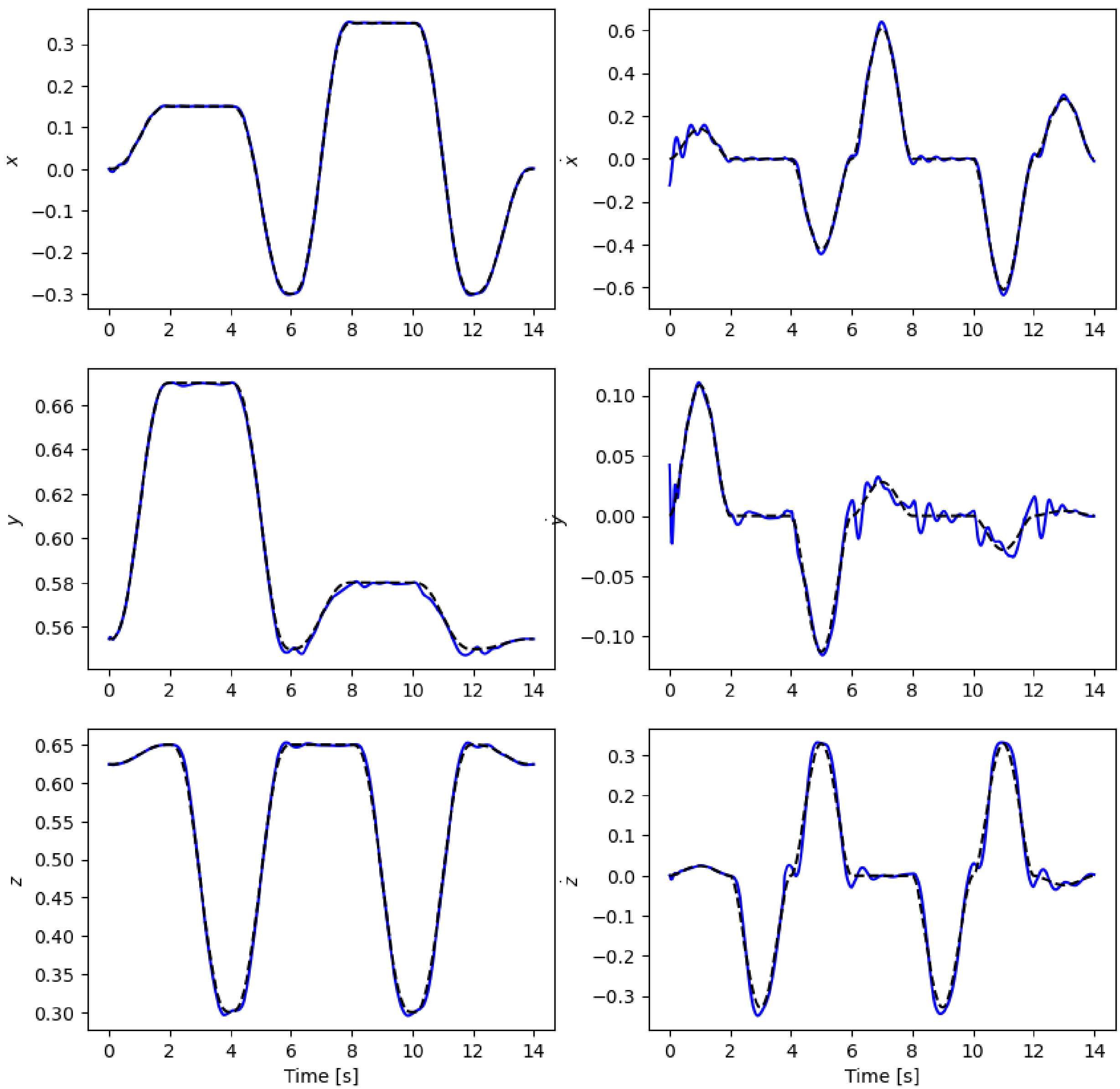
The controller used to achieve this task is a resolved rate controller utilizing the null space to maintain the robot's posture. The controller functions as described in the following steps:

1. The pose of the end-effector is calculated using the forwards kinematics function and the finger position is stored in an array.
2. The target positions the robot must navigate to are stored in an array.
3. 7 targets were selected with the last one being used to return the robot to its initial position.

4. The overall trajectory is as follows:
    - From the initial position to the first block's position ((0.15,0.67,0.65) in spatial frame coordinates).
    - The end-effector is then lowered (to a z-value of 0.3 in spatial frame coordinates) and the gripper position is closed in order to grab the first block.
    - The end-effector moves to the bowl's position, (−0.3,0.55,0.65), and the gripper position is opened to drop the first block.
    - From above the bowl to the second block's position (0.35,0.58,0.65)
    - The end-effector is lowered to grab the second block.
    - The end-effector moves to the bowl's position, and the gripper position is opened to drop the second block.
    - The robot returns to its initial position.
5. The trajectory of the robot is computed using a trajectory function with velocity and acceleration constraints using a 5th order polynomial.
6. If the target position the robot reaches at the current time is a block then the gripper position is set to fully closed.
7. The oriented jacobian is computed using the function and the desired linear velocity of the end-effector is calculated using $v_o = \dot{x}_{ref} + P(x_{ref} - x_{measured})$ from which the desired velocities $\dot{q}$ are calculated
8. The null space is used by calculating $\dot{q} = J^+ v_o + (I - J^+ J)\,\dot{\bar{\theta}}$ where $\dot{\bar{\theta}} = P_{posture}(q_{desiredpose} - q)$ and $q_{desiredpose}$ is the initial position values of the robot in order to maintain the posture of the robot.
9. Lastly, the joint torques, $\tau = D(\dot{q}_{des} - \dot{q}) + G$ are calculated with gravity compensation, where G is the gravity force seen by the joints, and sent to the robot along with the gripper positions.

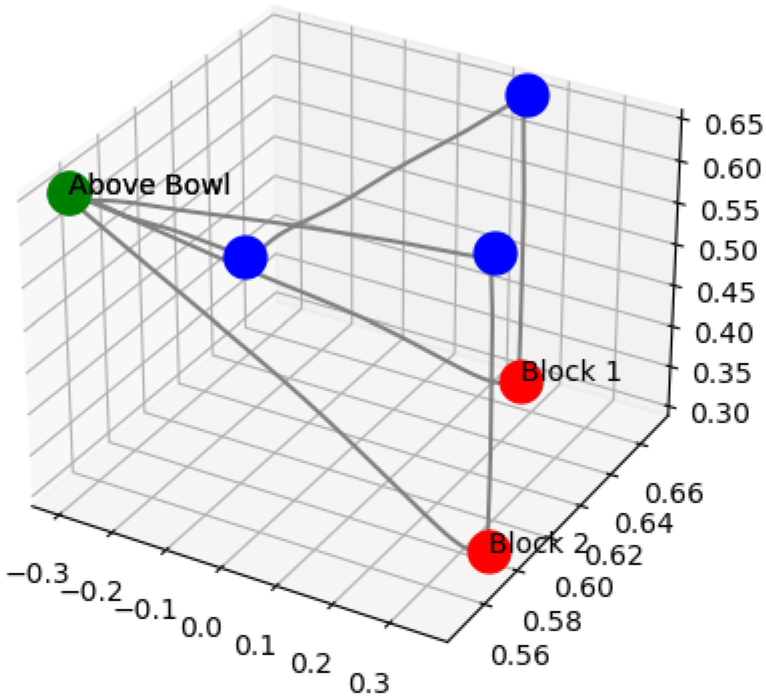
SYSTEM BEHAVIOR AND ANALYSIS

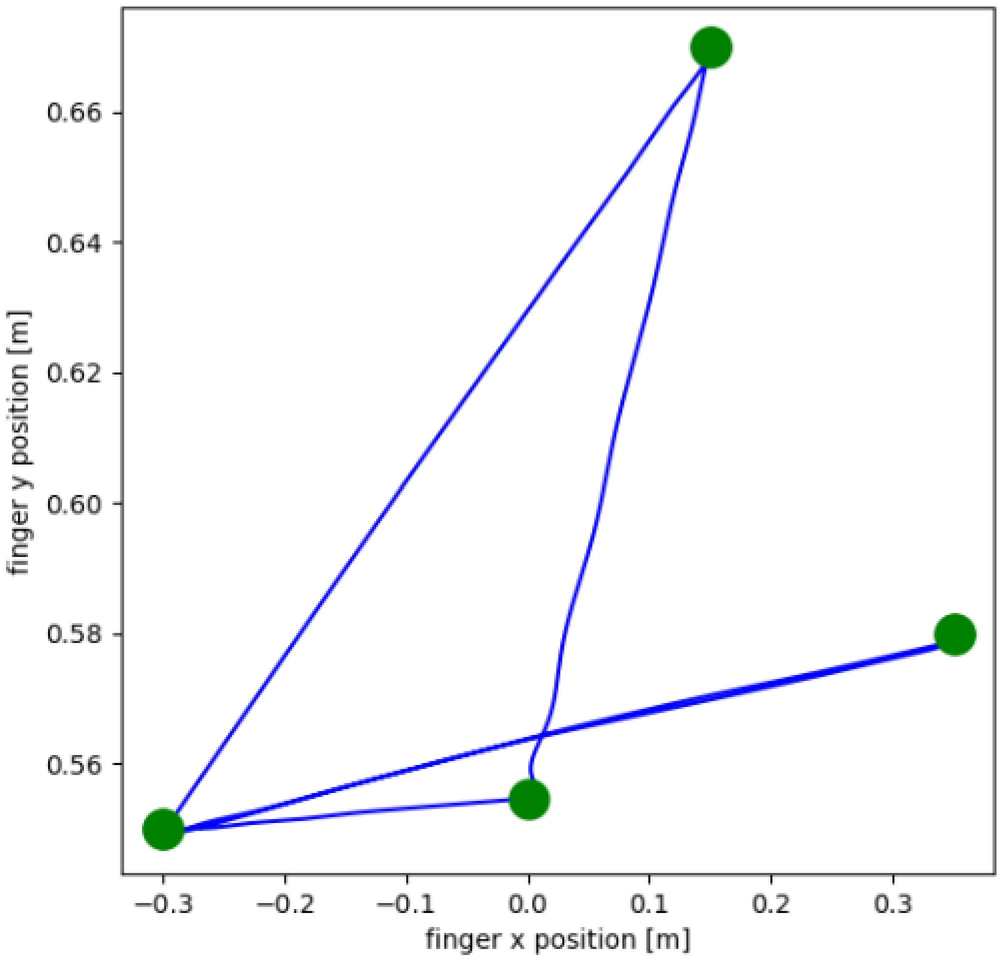
Position-Velocity Plots:

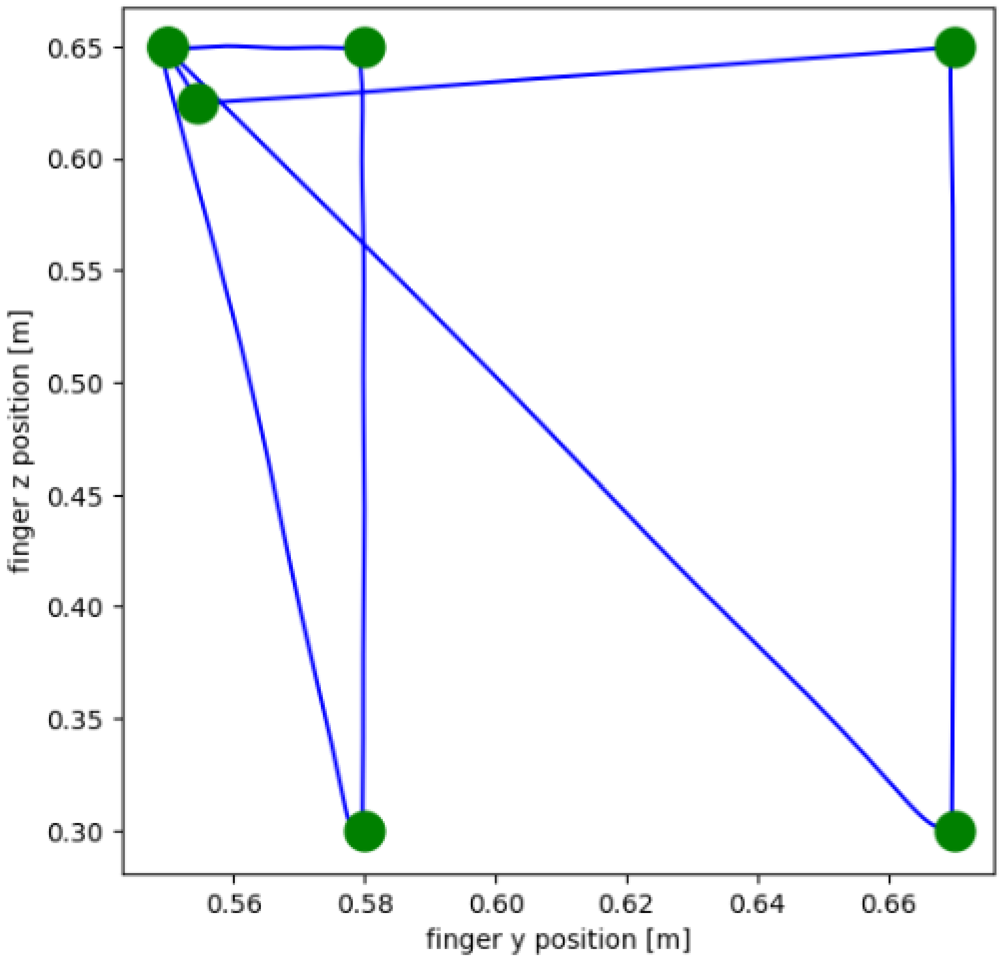


End-Effector Trajectory:

Trajectory Plots:




0.66
0.64
0.62
0.60
0.58
0.56
finger y position [m]
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
finger x position [m]



As can be seen in the plots above, the robot reaches its target positions as evident on the end-effector trajectory plot in the x-y plane and the y-z plane and the end-effector position and velocity plots. The measured end-effector positions and desired end effector positions are very similar with little discrepancies. Furthermore, the measured end-effector velocities and desired end-effector velocities have little discrepancy as well, but noticeably more than the position plots. Therefore, perhaps the time to each goal should be increased to allow the robot to stabilize before going to the next target.