

Mobile Manipulate Capstone: README

Quickstart Guide:

- After extracting the zip file, there should be 4 python files in your code directory, main.py, Milestone1.py, Milestone2.py and Milestone3.py.
- In your results directory, there should be three folders, “best”, “newTask” and “overshoot”. In each of those folders you’ll find 5 files that correspond to this project’s results.
 - A log file of the code, an mp4 of the traj.csv file working in CopelliaSim, a picture of the erro graph, the trajectory csv file and an Xerr csv file with all the error data.
- To run the full program, navigate to the code directory and run “python3 main.py”.
 - This will take a little while to run, be patient.
- Depending on what you want your results to be, you can comment and uncomment certain parts of the main.py code to manipulate the input controller gains
 - If you want the highest quality results, choose the “best” options
 - For an example of overshoot, choose the “overshoot” options
 - To see an example of overshoot and chattering, along with different box positions, choose “newTask” options.
 - According to your desired results, change the log file at line 22, the error csv file at line 151, and the trajectory csv file at line 183
 - You’ll also need to change the gains and box configurations starting at line 192.
- Once the program is done running a graph of 6 error plots should pop up on your screen.
- You should also see log and csv files in your code directory based on your controller choice.

Software Description:

- Milestone1, NextState function:
 - Given A 12-vector representing the current configuration of the robot, a 9-vector of controls indicating the arm and wheel twist, a timestep Δt and a speed limit, output a 12-vector representing the configuration of the robot time Δt later.
- Milestone2, TrajectoryGenerator function:
 - Given an initial configuration of end effector, the cube's initial configuration, the cube's desired final configuration, the end-effector's configuration relative to the cube when it is grasping the cube, the end-effector's standoff configuration above the cube, before and after grasping, relative to the cube, and the number of trajectory reference configurations per 0.01 seconds, generate the reference trajectory for the end-effector frame {e} for eight concatenated trajectory segments
- Milestone3, FeedbackControl function:
 - Given the current actual end-effector configuration, the current end-effector reference configuration, the end-effector reference configuration at the next timestep in the reference trajectory at a time Δt later, the PI gain matrices K_p and K_i and the timestep Δt between reference trajectory configurations, calculate the commanded end-effector twist expressed in the end-effector frame {e}.

Results:

- In each folder in the “results” directory, you can see how the youBot moves to a cube’s location, grasps the cube, moves the cube to its final destination and releases the cube. One thing to note is that the more time you give the robot to move in the first segment, the better your error will smooth out.
- One the next page are the graphs for each controller option. To see how the controllers work on the robot, load the appropriate csv file in Scene 6 in your VREP scenes in CopelliaSim.

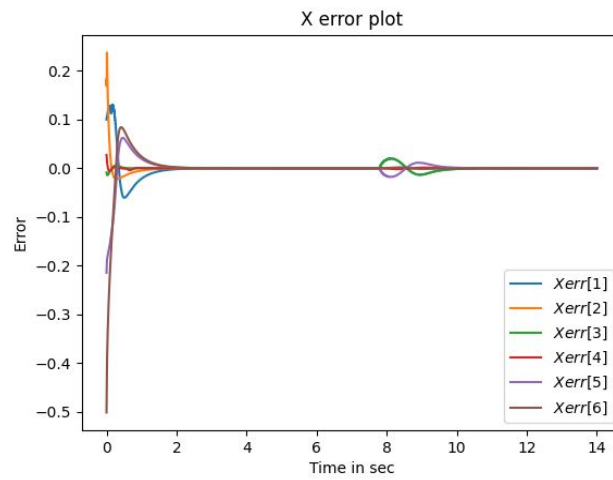


Figure 1: Xerror plot for the “best” controller gains

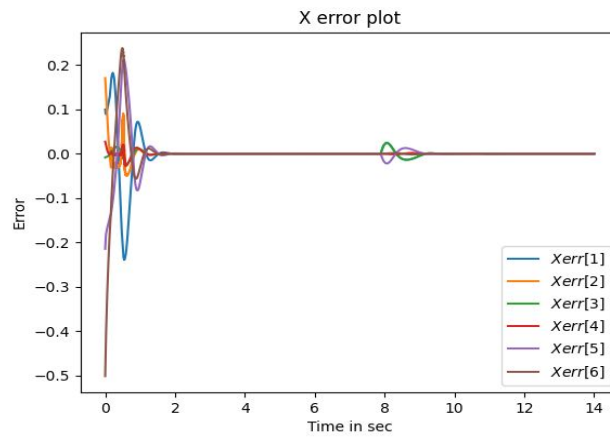


Figure 2: Xerror plot for an “overshoot” controller

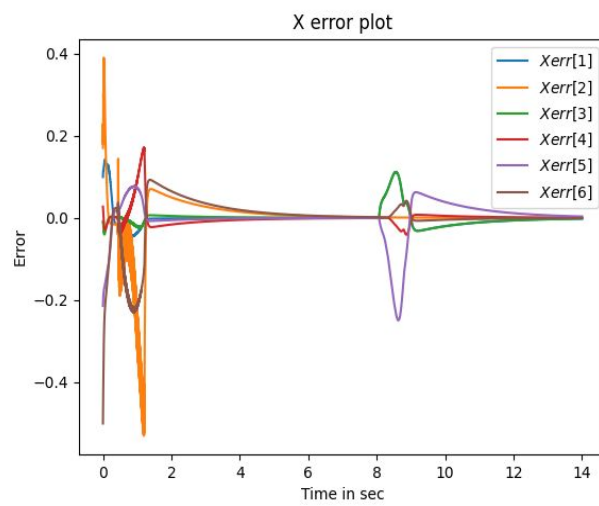


Figure 3: Xerror for the “newTask” controller