Christopher Tsai

ME 449 Capstone Project README

This hands-on project was a great way to integrate all the material that was taught in ME 449. In summary, this piece of software creates a reference trajectory that a mobile manipulator follows and performs odometry as well as feedback control to move the robot as close as possible to the trajectory. During the trajectory, the robot picks-and-places a cube of known parameters in specified locations. The code can be separated into steps:

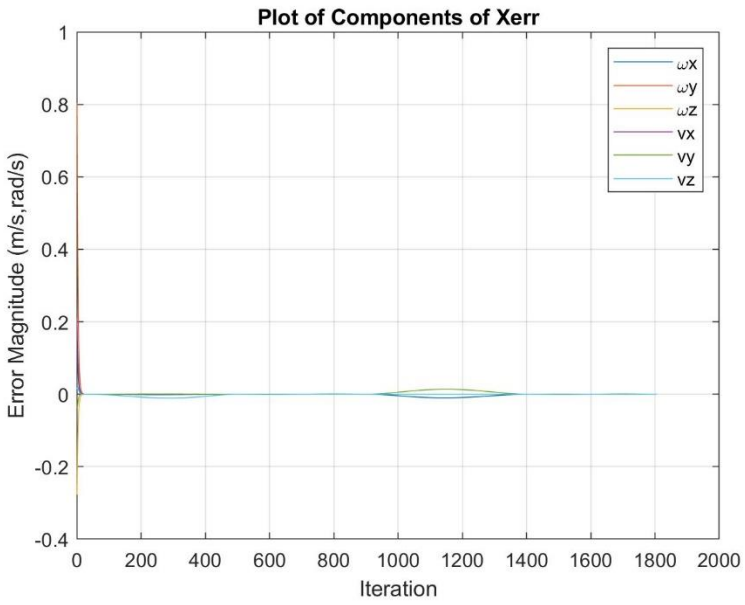
1. Creating a desired trajectory.
2. Initializing values and parameters that are used by functions, such as dimensions, max speed, and timesteps.
3. Iterating through each step of the desired trajectory and:
    a. Calculating control law to generate wheel and joint controls that move current configuration to desired configurations.
    b. Feeding these controls to a function that creates and builds the actual trajectory.
    c. Storing errors for later data visualization and analysis.
4. Plotting and exporting error data.
5. Exporting .csv file of actual trajectory.

After integrating the functions from the different milestones of the project, I found success by:

1. Tuning the gain parameters Kp and Ki.
2. Tuning the length and speed of the reference trajectory segments so that the robot wouldn't drop a cube once it grabbed it or take too long grabbing it.
3. Increasing the parameter k, the number of trajectory reference configurations per 0.01 seconds.

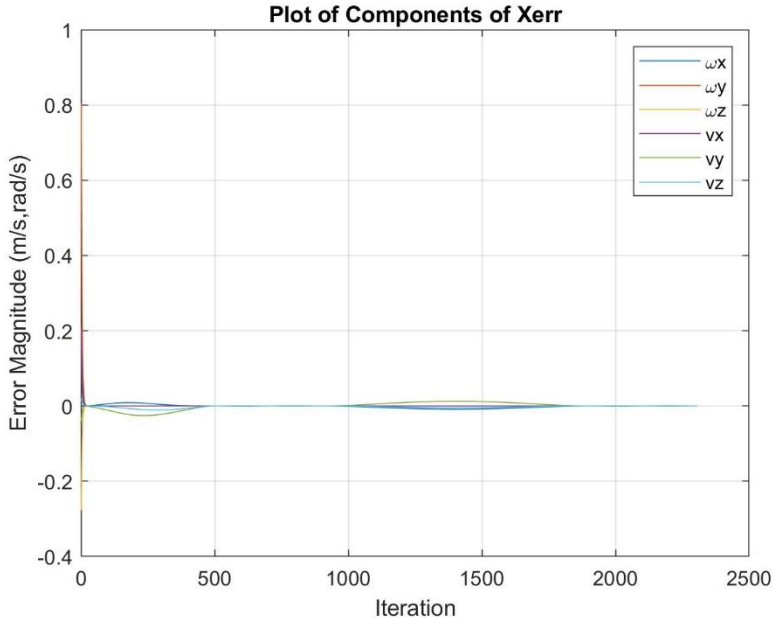
My main script file is called "runscript.m" and is in the Code directory of my project. It takes as inputs some of the variables I changed from working on the Best Run, Overshoot, and New Task subparts of the project. These input variables are specified in the README text files in subpart's directory inside the Results directory.

Results:



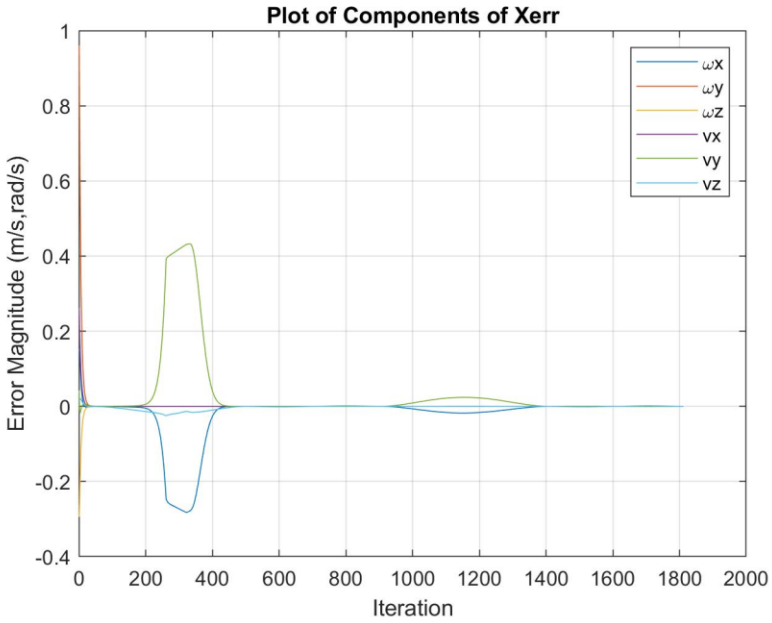

My Best Run exhibited very little error. The mobile manipulator was able to pick up, transport, and place down the cube very effectively.

This was further supported by the low errors in the New Task graph, since it involved a longer trajectory than Best Run:




As mentioned in the Overshoot README, increasing Kp too much (>2) led to unstable behavior (when keeping Ki at 0.8). This was an interesting way to observe how control gains can

negatively affect the performance of a system. Best overshoot performance came when Kp = 1.98:




Since video recordings are large files and V-REP consumes a lot of memory, they might lag when played with V-REP or MATLAB running. In case this happens, I have uploaded the videos to YouTube at:

- Best Run: https://www.youtube.com/watch?v=rEUm8NctBUc
- Overshoot: https://www.youtube.com/watch?v=EDnSSkjEJkQ
- New Task: https://www.youtube.com/watch?v=JDJmK_lgKG4