Carnegie Mellon University

16-682

MRSD Project II

Task 07 Progress Review 11
Team C - COBORG

Jonathan Lord-Fonda
Teammates: Husam Wadi, Feng Xiang, Yuqing Qin, Gerry D’ascoli

November 10, 2021

Table of Contents

1. Individual Progress

My primary tasks for this progress review were centered on integrating the software stack and running the use case. For this process I worked with the whole team and spent many hours in the lab making sure all of the parts of the system came together correctly and operated with one another. This involved lots of debugging my code, as well as helping others debug and uncover problems in their code. Additionally, I helped adjust our testing structure so that we could run overhead use cases, something that we'd ignored up until that point. Running the use case involved at least three people. Someone to run the code and test their changes, someone to wear the backpack and use it, and someone to assist both the code runner and Coborg user. While running our use case we realized there was a calibration problem and I helped fix that issue. Additionally many of our use case runs were targeted at tuning the system variables to function as intended for the user, such as how hard the robot should push against the panel to maintain its position, without pushing the user over. I also worked on other aspects of the project, such as creating custom ethernet cables with Gerry so that we could implement a professional-looking hardware design. Beyond those other tasks I also worked with Husam to develop a system feedback structure that would allow our main state machine to monitor the status of the individual subsystems and help us recover from subsystem failures automatically during demonstrations.

Finally, I also spent more time working on the path stitching feature that would be great to incorporate into the final system, if we can get it to work. The problem I spent time resolving this week was that I needed to manually interpolate and send trajectories to the motors because MoveIt, despite having an asynchronous execute function, refuses to handle execution of multiple trajectories and simply crashes.

2. Challenges

We faced a lot of different challenges this week as we pushed everything through integration so that we could have our full demo working for PR 11. Some of these challenges were existential in nature for our solution. For example, as shown in Figure 1 below, the Coborg can get awfully close to its human operator, particularly when operating in the vertical use case scenario. Having the robot this close to the user's face makes it really difficult for our system to work well and still provide acceptable safety to the user. Ideally, we would redesign our system to have some sort of softness, both programmatically and in its physical components but we don't have the time to rebuild our system based on these principles. From the outset we knew this might be a problem and so implemented multiple safety features to help the user mitigate these scenarios, but adding safety stops is not nearly as effective as designing out the problem from the beginning. Thankfully, this issue shouldn't occur during the trajectory-planned stage of the movement, only during resolved rate. This is because resolved rate doesn't have obstacle avoidance and only makes small changes based on the Jacobian at its current time step. This lack of forward thinking can also cause resolved rate to fail after some time because it can't plan to move in a way to maintain a Jacobian of full rank.

Figure 1 - The Problems of a Hard Cobot

This figure shows a screenshot from one of our recent test runs. The carbon fiber arm is mere inches from Gerry's face. The arm wasn't moving quickly, and he could easily pull the e-stop if necessary, but there's a big difference between us running the system in a controlled test and us selling the Coborg to users who expect to use it daily in a work setting.

Other problems that cropped up while we were testing integration include the fact that the t265 has some drift and that we're working at a centimeter-scale. Initially I had assumed that the robot would push into the board about 15-25 centimeters to achieve the desired forces. Our initial testing showed that this actually made it difficult for the user to resist the Coborg, so we reduced the distance to 5-10 centimeters. This was better for the user, but gave our system a smaller threshold of force to work with, below which it would drop the panel. The t265 is reliable for tracking, but can easily drift a couple of centimeters. Moving the target 2 centimeters out of 15 or 25 doesn't make much of a difference, but moving the target 2 centimeters out of 5 can cause significant issues. We're currently working on adding force thresholds and adjusting our parameters so that our system will function in all desired scenarios.

Additionally, while we tested the arm's ability to reach all of the locations in our task space, we didn't check the jacobian at each of those points. Ideally, the arm would be shorter for the horizontal use case because as of right now the arm is a little too scrunched up, taking up the user's space. However, in the vertical use case, the arm is fully extended to reach the points, meaning that it has limited stability capability. If we had time to redesign the arm a little, turning the last link into a linear actuator would dramatically increase the capability of our system to handle points at many different locations in the task space.

3. Teamwork

Jason helped the whole team integrate the actuated manipulation subsystem and then the entire system as a whole so that we could run our full pipeline/use case. Additionally, Jason fine tuned and calibrated the camera mount and robot arm with Yuqing. Beyond this Jason debugged through edge cases in actuated manipulation during use case testing.

Gerry helped the whole team integrate the entire software pipeline. He also assisted with testing and tuning of the system, more specifically by adjusting the obstacle avoidance to provide sufficient capability while optimizing for speed. Additionally, Gerry implemented a first attempt at integrating orientation into resolved rate stabilization and built custom network cables for our professional hardware design. Beyond these tasks, Gerry also updated the voice_recog node with additional functionality and keyword tuning as requested by the main state machine integration tasks.

Yuqing helped the whole team integrate the vision subsystem with the other components of the software pipeline. She also switched the vision model to test alternatives and tuned the window size of the goal_getter. Additionally she added distance thresholding to resolve corner cases and worked with Jason to calibrate the cameras.

Husam's work during this previous cycle primarily focused on creating Coborg's V2 hardware. He created completely new wiring on the rear Coborg and re-attached the frame in a different orientation, as requested by the Vision team. Additionally, he helped Gerry and Jason with wiring up the robot arm and updated the main state machine node to handle feedback from the various subsystems. He also implemented a voice-activated emergency stop toggle.

4. Plans

Before the Fall Validation Demonstration we need to make sure that our system can consistently perform well. To that end I will be ensuring that it can automatically detect and mitigate errors so that it can run without interference from the team. Additionally we'll perform better tuning of the Coborg system so that it can reliably work alongside a human operator. Additionally I will be adding a couple of adjustments to resolved rate to improve its performance and mitigate some of the problems that we've run into this past week.