Jorge Anton Garcia
Team D – CuBi
Teammates: Laavanye Bahl, Paulo Camasmie, Changsheng Shen (Bobby), Nithin Subbiah Meganathan

ILR08
Oct. 09, 2019

Individual progress

This week I worked on planning exploration with Bobby and on repeating the encore with Nithin. I also created multiple testing suites, refactored the entire code base, and integrated everyone's work.

CuBi Technical

For planning the global exploration module, I read several papers to determine how to go about performing the path planning of the coverage region task.

Our constraints are as follows:

- We know the position of the walls
- Don't know position of obstacles
- Don't know position of toys
- Robot must move through all the points in the target area covering it completely
- Robot must fill the region without overlapping paths
- Robot must avoid all obstacles

I divided the task into three parts: decomposing the occupancy grid, determining in what order to traverse the cells, and how to avoid obstacles. We need to research the ROS navigation stack to see what I can provide.

I also looked into a genetic path planning algorithm. It optimized for path length, number of turns, and the revisited areas and does not seem complicated to implement. It also takes care of obstacle avoidance making it a very interesting option. It only requires the boundary of the room and perform the exploration in real-time with a dynamic environment. The general idea of this algorithm is that it creates local trajectories based on what it can see and designs fitness levels to those mini trajectories. It then combines different mini trajectories through mutations and crossovers to add some randomness to the exploration.

To get the SVD running again, I had our state machine, position follower, and absolute to relative pose converter modules. I reduced the lines of code by over 33% and made the interfaces much more clear. I also created abstraction layers, so other team members could run only the sections of the code they need. For example, I wrote a function in which you give it a series of manipulator movements and it will perform all the movements in series. It also provided a timing analysis. Therefore, Paulo or Laavanye could just input MANIPULATOR.FINGER_TOGGLE or MANIPULATOR.VALIDATION_POSTION and the manipulator will perform that action. Paulo can also use the timing analysis to having metrics of how much improvements he has done when tuning the gains.

I also added timing analysis for every state. Cubi will perform the SVD and when it ends, it will output how much time is spent in each state (search for toy, picking up toy, etc…). In this way, we know where we spend the most time. Furthermore, I added PID control for angular movements and cleaned up the logic of the low-level controls. Finally, I added the ability to debug Cubi without moving it. We can assume Cubi's low-level mobility code works, so we can

send it fake commands as if it were moving. In this way, we can test cubi's logic higher-level logic independently.

CuBi Project Management

This was our best week as a team. We all met our objectives. As every week, I kept Notion up to date, but as opposed to other weeks, I told everyone to bring their higher-level to very specific tasks. In this way, I could keep track of progress much more granularly and in our team meetings we could discuss the issues we were having and the status of the project. Great week for team Cubi!

We did a learning retrospection after our PR. The following are our action items on the PM-side for next week:

- Everyone needs to test their code when they push it to Dev branch. If Dev breaks, then we meet as a group to debug it.
- No work is considered complete until it's integrated.
- We will try to finish our work the day before our PR, so we can discuss it before then.

Challenges

Technical

We faced issues with Cubi not moving suddenly. The OpenCR board we had had been accidentally had its settings changed. It took a long time to debug, but now this issue is documented for whenever it happens again.

Cubi also crashed into the box with the manipulator, so I had to raise the manipulator before it rotated.

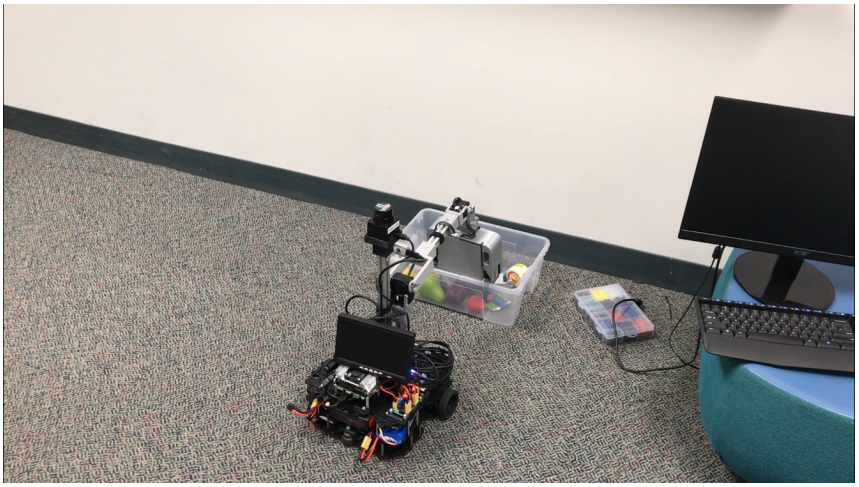

Figure 1. Example where Cubi crashed into the box.

I also faced unreliability issues when performing the SVD. I was able to narrow down the issue to having it being caused by a failed callback. However, I did not have enough time to fix the entire issue. This will be pushed back to next week.

Team Work

Nithin: I worked with him directly with to help design the SLAM portion of our project. He then helped me debug the issues I was finding with re-running the SVD.

Paulo: I add extra abstraction layers with timing analysis, so that Paulo can improve the PID controls. I worked with him to get a sense of what functions could help him.

Laavanye: He wrote a function to validate grasping. I wrote the function to move the manipulator to the necessary position to validate grasping and integrated his work with the state machine.

Bobby: We worked together to design the exploration module of the project.

Plans

- Improvements to SVD:
    - Create an approach point, so we always go to the box head on.
    - Add the ability to exit go_to_point in the case of a failure
    - Fix the offset when going to a toy
    - Fix the positions where the position follower does not directly go to the position.
    - Debug Dynamixel failing issue
- First implementation for exploration policy with Bobby
- Plan the obstacle detection with Laavanye and Nithin
- Help Paulo tune gains