16682-A MRSD Project 2 | Individual Lab Report # 9 October 23, 2019

Paulo Camasmie | Team D – CuBi

Teammates: Jorge Anton, Nithin Meganathan, Changshen Bobby Shen, Laavanye Bahl

Individual Progress:

I integrated the grasping validation into our State Machine with Jorge. Then I switched my focus during the last two weeks to find ways to make CuBi achieve faster its overall task of collecting multiple toys in a room and delivering them to a location. I interpreted CuBi's mission by repeatedly watching and timing its video in action, Fig 1., through the lenses of the Pareto's law, and tried to identify major area for potential improvements. Considering that in the end, we have not built a trunk for CuBi to hold multiple objects concomitantly, it became obvious that CuBi will spend most of its time traversing the environment. My focus then became to cut the time that CuBi takes to move linearly.

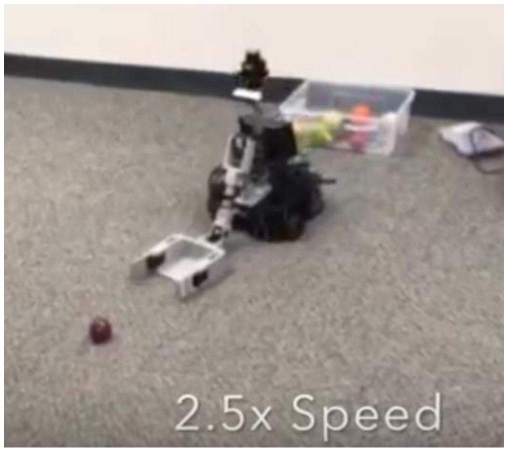

Figure 1 CuBi demo video, running 2.5X to be tolerable to watch

I queried the original specs for the Turtlebot Waffle Pi, which is our mobile base, and its nominal linear speed is 0.26 m/s [1] . I was not observing those speeds, and I knew that the low-level controller was done by the Dynamixels, who receives desired linear speed targets and translate that into motor angular velocities. We have our own PID control in place for the high-level desired velocity of CuBi, which is a function of the position error from our planning and odometry feedback.

Delving into the code, I noticed that we were using very small proportional gains that were inhibiting CuBi to develop full velocity. Worse, it was spending most of its time, very close to the target, just waiting for convergence. Reducing the threshold for convergence did not seem wise to me.

I have considered and tried different strategies and determined the best option to be adaptive PID gains for CuBi. I am very happy with the results. CuBi will now traverse 1 meter in 9.20 seconds, versus the previous 23.20 seconds, as measured in a one meter drag race, Fig 2.

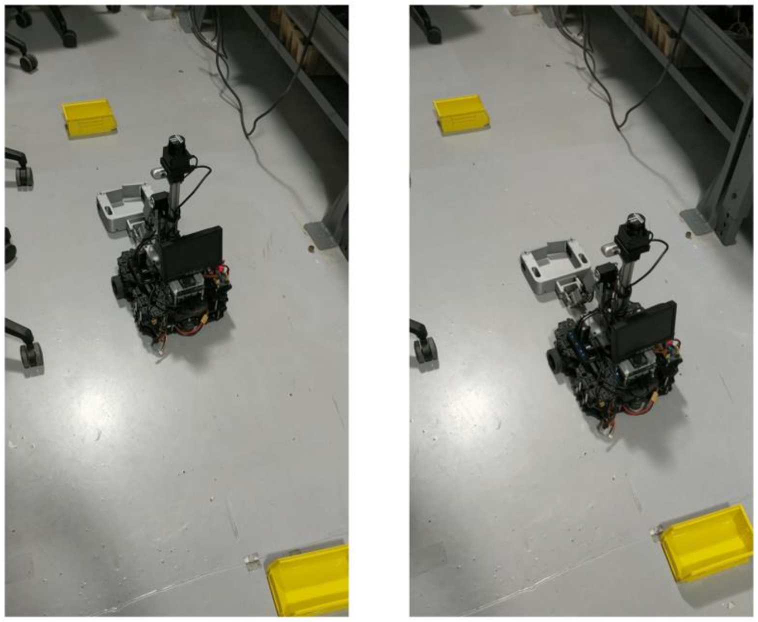

Figure 2 One meter drag race. CuBi arrives first to the target on the right picture

I achieved this by using an aggressive gain when CuBi is far from the object and switches automatically to a moderate gain as it gets closer. I found this to be the best option as it avoids overshooting the goal, not because of its physical constraints, but due to the low frequency of ROS state feedback updates. The results can be seen on video [2] . I have also implemented a similar strategy for the rotation of CuBi. I have forfeited the use of any derivative and integral gains, since I did not observe oscillatory behavior or steady state convergence short than our goal.

I am not planning any other immediate gains changing or tuning, since the traverse time will give us the biggest "bang for the buck". I am a bit concerned that if we aggressively change the Dynamixel gains for the manipulator, it could put unnecessary stress on the CuBi joints that are printed from PLA material—32Mpa tensile strength[3] —without giving us too much performance improvement.

Risk Mitigation

- I want to double check that the aggressive gains in linear speed will not cause unintended consequences on the overall behavior of CuBi, such as excessive drifting, or pushing through objects to be collected too aggressively. I am intentionally using a low gain as CuBi approaches the object for that reason. Still validating these changes for our complete spring validation is critical.

Individual Challenges:

- My biggest challenge was initially to interpret the initial control in place and its intentions or lessons learned without proper documentation from team. I then naively implemented a high gain control and noticed that the robot went crazy, with apparent oscillatory behavior from the wheels. I then had to dig deeper and follow many topics in ROS to realize that the source of the oscillatory behavior was that CuBi was over-shooting its goal since it was receiving sparse odometry message as feedback, and when the distance from target became negative, our code was in an edge case for which was not robust enough. I tried many tuning options before realizing that a high gain could always place CuBi in such an edge case, and simply adding a flag to override behavior in case CuBi passed the goal, felt too "hacky" to me. The best solution was the simplest after, all and it worked well.

Individual Next Steps:

- As I feel that I have fulfilled my promises to my team to date, and that CuBi is now on turbo mode, I will switch my focus to planning. I will be working with Jorge to implement local planning for CuBi to pickup toys and take it to destination. CuBi exploring strategy is one of cell decomposition, in which the cell size is encompassed by CuBi's Field of View. In my mind, a simple Minimum Spanning Tree algorithm or a Travelling Salesman one should be good choices. However, since CuBi has no trunk, we have to consider and modify each algorithm to add the object delivering and returning paths edges to the graph. Maybe we can create our own search algorithm to accommodate this idiosyncrasy.

Team Progress:

Laavanye further perfected the validation of successful grasping with a histogram of colors to be light invariant. He successfully showed the system during our PR.

Bobby implemented a localization policy for CuBi, using particle filter from the AMCL library. He successfully created the map of the new lab area that we have been using which CuBi can use to localize itself Fig 3.

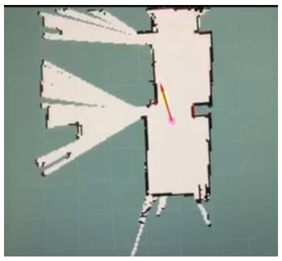

Figure 3 CuBi localization with particle filter

Nithin and Jorge implemented the global exploration policy to CuBi's state machine. It now includes mapping the area with LiDAR, exploring, avoid local obstacles, until achieving termination condition. They also compared methods between a classical approach and a deep learning one to build a future obstacle detection pipeline.

Team Challenges:

Laavanye had to put a considerable amount of effort to make the grasping validation robust and to integrate it in the state machine.

Bobby worked very hard to use two separate packages successfully to build a map and than to localize the robot.

Jorge had to dedicate considerable time to work with me and then with Laavanye to help integrate our codes into the state machine, since he was the most familiar with it. He also put a good effort into researching different exploration strategies.

Nithin put a good effort into researching different object detection pipelines, helping with map capture and preparing the presentation.

Team Next Steps:

Laavanye, Jorge and Nithin will research and plan for implementing potentially a semantic mapping, where dynamic objects can be ignored during localization. **Laavanye** will also work on robust obstacle detection.

Bobby, Nithin and Jorge will work on global planning and obstacle avoidance. They will also research data fusion to improve CuBi's state estimation.

References:

[1] http://emanual.robotis.com/docs/en/platform/turtlebot3/specifications/#hardware-specifications
[2] https://docs.google.com/presentation/d/12h0FB-ur3i-ZrxJrvUk6kZTh9ElsNGRaXtaI8uVpF2M/edit?pli=1#slide=id.g621a8d02d7_0_37

[3] https://3dprint.com/233916/tensile-properties-of-3d-printed-pla/