Jimit Gandhi
Progress Review 12
Individual Lab Report

Team G- Robographers

Team Members

Rohit Dashrathi

Sida Wang

Gauri Gandhi

Tiffany May

Introduction

In this progress review I worked on navigation subsystem, integrated detection subsystem, fixed the voice command problem. Also performed tests and fine tuned the flocking behavior

Individual Progress

In this Progress review my goals were the following

1. Demonstrate the flocking behavior.
2. Fix the voice command problem.
3. Test the integrated detection system.

I explained the flocking controller last progress review. This week I implemented it and made a Rocon rapp for it. The rapp as I mentioned in the previous ILRs contains 4 main files which are as follows

1. Parameters file - This contains all topic list and package names
2. Launch file - This contains the nodes that will be launched and topics that will be remapped
3. Interface file -This contains interactions with the user (left blank for my system)
4. Rapp file - This contains list of above files such that the hub knows about them.

I created this rapp file for flocking algorithm. I then tested it on 2 Turtlebots. It worked perfectly. The overhead localization setup gave accurate readings. Then the flocking was tested on 3 turtlebots by tele-operating one of them. Once that worked we tried to flock the 3 turtlebots by using our own navigate towards a person node[Figure 1]. So one turtlebot detects april tag on person's chest and moves towards it. The other turtlebots just flock with that turtlebot. The flocking was tested many times this way.

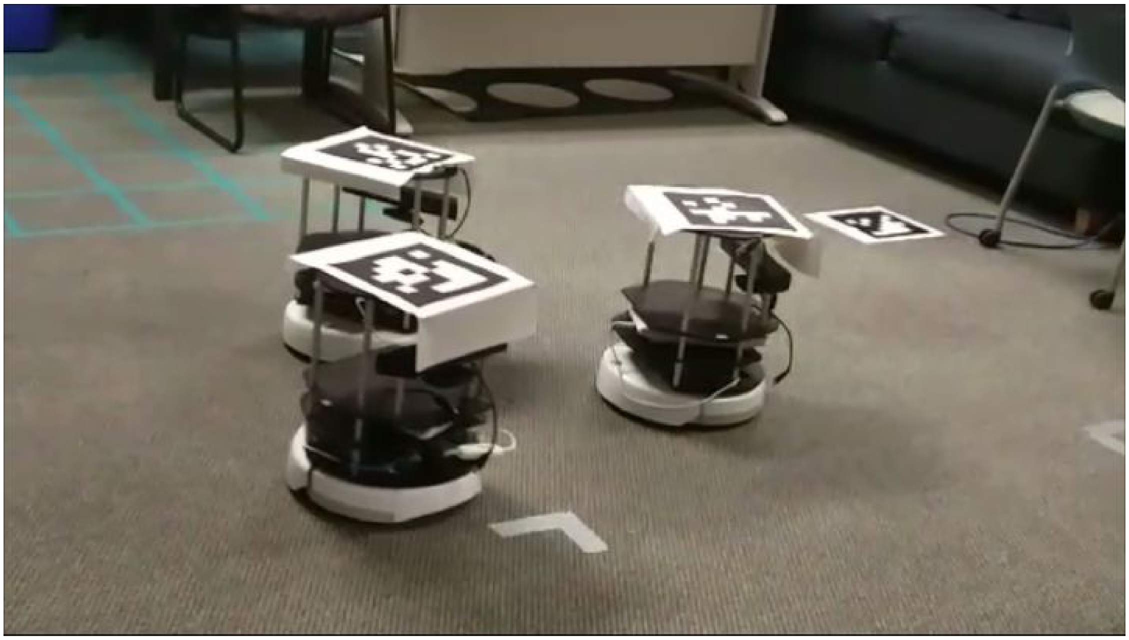

Figure 1: Turtlebots Flocking

In the last Progress review, Sida had integrated the voice command feature which says “say cheese” and once the picture is taken it says “ Nice smile”. But it does so in a repeated loop. In other words, it continuously says “Nice smile” and “say cheese” in repeated loop irrespective of the events that occurred. I fixed it by introducing a smile_flag which is 0 when no picture is being taken and says “Smile please”. Once picture is taken, the smile_flag is changed to 1 the “nice smile “ loop is run and again the smile_flag is set to 0. The “say cheese command also depends on the availability of the topic of smile percentage. If the smile percentage topic is not received than the loop of “smile please” does not run. If the topic is available with values than the value is checked to see how much is it less than the smile threshold which is 0.5. If it is less than 0.5 the system sends out the voice command “smile please” until the person smiles. Next task will be to do it just 3 times and then if smile is not detected the robot moves on. Once picture is taken it says “nice smile” only once.

The next task that I performed was testing the entire sub-system with Gauri and Rohit from start to end. Once launched it detects the person’s April tag, then each of the camera detect the person’s expression and head pose. They publish each of these values to the central server which compares the smile expression and head pose values are sent to pan tilt which move accordingly to track person’s face. Gauri had fixed the chromebook crashing problem we had last time where all chromebooks would crash. It was a kernel panic error which caused the error. The kernel was being somehow abused because multiple image_view windows would open at the same time. This would obviously cause an overload. So we did two things that could be possible solutions. Gauri upgraded the kernels and Sida removed the excess unrequired windows. Just one final output image_window was enough. The problem was solved. We ran the entire detection system and we got the final output of the person smiling as expected. We tried this a multiple times and it worked all the times.

Challenges faced

The testing of flocking is not challenging but requires some amount of preparation. One has to ssh into each of the Turtlebots and launch the Rocon Rapp and launch other files such as overhead localization and flocking controller.So basically one has to run 9 commands in 9 different windows and that too in particular order to get Rocon working properly. Again it is not challenging but to get the order right took multiple trials and two to three hours of tinkering.
The other challenges occurred right during the progress review. To test different subsystems we would launch Rocon for each of the subsystems in different laptops. When we did trial runs the day before and the morning before progress review, the subsystems worked perfectly fine as no two subsystems were launched at the same time. But during progress review when we tried to launch both the subsystems simultaneously on the same network, some interference occurred which led to none of the subsystems working correctly. This happens because Rocon uses a framework called ZeroConfig which I have mentioned in one of my previous ILRs. This framework is used when one has a multi-master setup to automatically detect the devices connected on the same network. This works perfectly when there is one hub but in our case during the PR we launched two hubs on the same network. The two different Rocon networks interfered as they would conflict over same devices to connect with.

Another challenge was the overhead localization problem. So during the Progress review, another issue occurred which would not have been detected if not debugged. So initially I thought that due to interference and launching of two hubs caused the system not to run. But then at the same time to debug I tried publishing the localization topics which is nothing but coordinates of the april tags. As I tried to print out the tag detections, nothing got printed on the output screen. This should have worked as localization works locally on central computer and has nothing to do with Rocon. Then debugging further I realised that due to excess lights one of the APril tag was not being detected in the camera. This april tag was responsible for extrinsic calibration of two of the four cameras. So if two cameras cannot find relative pose between them the entire global localization does not work. Hence it is pertinent to check the lights and the april tag detection all the time before testing or starting the system. The only condition is that april tag has to be detected only once in the entire run time.

Team Work

Gauri fixed chromebook issues. She proved to be an integral part of the team as a systems person. She helped Rohit debug, fix and test the pan tilt tracking of the face of the person. She also helped me implement and run the flocking algorithm. She was also involved with the testing of detection subsystem. She also handles the version control part of the project of merging all the codes.

Rohit worked on writing the pan tilt code from scratch and fine tuning it. He did it with ease and tested it. It was sheer bad luck that he was not aware of the issues that came up during the progress review.

Sida worked on changing the window size of the final video stream as well as removing the irrelevant windows which were causing the kernel to crash.

Tiffany was working on developing her own pan tilt tracking unit and merged the arduino code on ROS.

Future Work

Next is the SVE, so we will focus on integrating the navigation subsystem with the detection subsystem which will be the last step of the project. We will run it multiple times and produce a performance matrix.