Individual Lab Report- 5

By Danendra Singh

Team F: Falcon Eye

Danendra Singh

Yuchi Wang

Pulkit Goyal

Pratibha Tripathi

Rahul Ramakrishnan

22nd November 2017

Individual Progress:

As my contribution to the MRSD Project-1 since the past progress review, I have successfully implemented higher level flight control for bebop2 drone's autonomous GPS based waypoint navigation. I have also performed preliminary recorded dry runs for our fall validation experiments. Additionally, I have also started working to implement a PID controller for smoothing our waypoint based navigation.

Bebop2 Drone GPS navigation:

We initially decided to use the MAVlink files stored in Bebop2 to autonomously navigate the drone, but after intensive testing and trials it was concluded that neither the MAVlink files can't be dynamically updated for waypoint navigation nor are they easily accessible using ROS. Hence we decided to develop our own control code for waypoint navigation.

In the last PR, we were able to accurately calculate the Euclidean distance between a target GPS location and the Bebop's current location. To autonomously navigate we needed both the distance and heading of the bebop. Using the 'Bebop_autonomy' package, we could receive the current roll, pitch, yaw angles of the bebop. The yaw angle of bebop is defined as the radian angle difference between the absolute north and bebop's current heading. This is standard for most other drones. So, if we know two GPS locations, we could calculate the relative angle between them. One of these GPS location is the Bebop's current GPS location and the other is the target location. We then match the yaw heading of the Bebop to the relative angle between the current location and target location. Hence by this way we were successfully able to orient the Bebop towards the target location.

After matching the heading of the Bebop to the target location, we gave a simple command to move the bebop towards the target location while continuously matching its heading to target location.

For incorporating the multi waypoint navigation, we recorded the GPS latitude and longitude of all the target locations and stored them as a vector in the launch file of 'bebop_cmd_server'. When the end of the vector is encountered, land command is passed to the Bebop and the drone automatically lands.

To incorporate the GPS inaccuracy, in our program we set that as soon as the Bebop reaches within 0.5m of the 1^{st} target location, it moves to the next target location and so on.

Figure 1: Bebop autonomously navigating to the target location (person in the picture)

<u>FVE dry run and setup:</u>

As all our tests for FVE are highly dependent on outdoor weather conditions, we decided to dry run the tests we have already completed and record our results. So, we setup a simulated environment of our FVE and successfully performed 3 of our FVE goals within the stipulated time frame.

The first test we simulated was to detect 10 april tags with 80% accuracy. We setup the 10 april tags on the ground such that they all are inside the video frame of the drone's camera at a height of 5m from the ground. We were successfully able to recognize 10/10 april tags.

Figure 2: April Tag detection using drone video feed

The second test was to accurately localize two april tags. We placed two april tags on the ground separated by a distance of 135 cms. We then launched the drone and after reaching a height of 5m, calculated the distance between the 2 detected april tags. We were successfully able to calculate the distance as 142cms which lies within the 30cm range we specified as the error parameter.

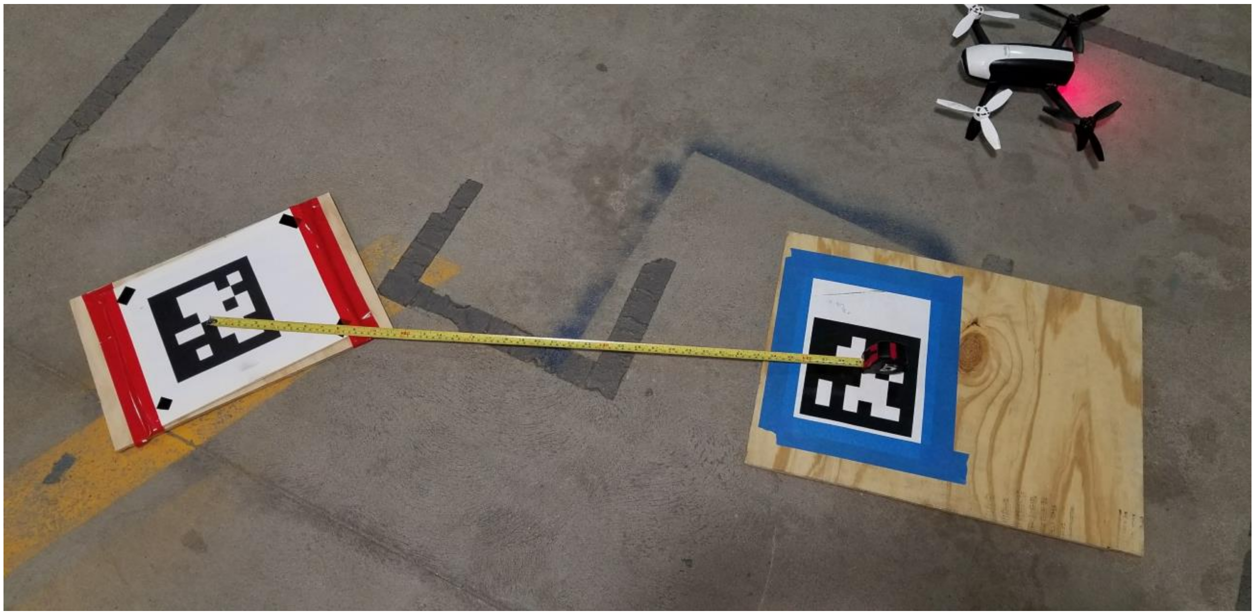

Figure 3: Calculating the distance between two april tags.

The third test was to autonomously navigate the drone between multiple waypoints. To achieve this, we selected 4 locations and saved their latitude and longitude values in the Bebop launch file. We then started the ROS master node and the drone moved to each specified location within the 5m error radius and finally landed on reaching the last location.

Figure 4: Drone landing after reaching final location (Indicated by the bottle)

Challenges

For this PR, the biggest challenge we faced was the inaccuracy in the GPS waypoint navigation and the unavailability of an open test location. To prevent the drone straying away to unknown location and causing damage, we reduced its flying height to 3m but we weren't able to test our control code during the daytime in an open location due to the presence of people around our testing locations. Hence, we decided to test during the night time as for GPS based waypoint navigation, visual odometry is not essential (which is unavailable in low light conditions).

With the recently added FVE goal we have to work in a much faster pace and with accuracy. We are facing interfacing problems with the Mini PC controlling the Husky and this has resulted in significant delay in our timelines. We are keeping an option open to shift to a laptop as a backup and focus on making the husky autonomously navigate based on GPS waypoints.

For testing the GPS waypoint navigation, we have to go in the open and face the cold weather which is a challenge in itself. Also, low temperatures tend to discharge the battery of Bebop faster, leaving us with lesser flight time.

Another big challenge the team is facing is the increased pressure of assignments to be submitted around FVE and FVE encore dates. We are trying to plan our schedule accordingly to incorporate these assignments so that they don't affect our project at the last moment.

Teamwork

Yuchi worked with me on developing the control code for waypoint navigation. We carried out multiple flight tests to validate our control codes and also performed dry runs for FVE experiment validation.

Pratibha worked to read IMU and GPS data simultaneously and tested them for accuracy for our system. She also made common launch files for the IMU and GPS data for Husky. She also helped me to decide components for our PDB and order them.

Rahul and Pulkit solved the WiFi connection problem for the Bebop and were able to connect it as a client to our home network. They also worked on solving the network issue with Husky and changed the dev rules for plugging in different serial components on the Husky. They also worked to make Husky's communication stable with all the sensors connected.They also tested the network bandwidth and range after connecting the Husky and Bebop to the same network.

Thus, by defining each member's goal successfully and working together as a strong team, we could achieve all the tasks for the PR-4.

Future plans

Since the FVE is approaching fast, I have decided to improve the waypoint navigation capabilities of the drone by using PID control and also work with other team members to swiftly implement GPS based waypoint navigation for the Husky. We are positive that we would be able to achieve some significant development in the Husky autonomous navigation for our FVE.