Individual Lab Report- 4

By Danendra Singh

Team F: Falcon Eye

Danendra Singh

Yuchi Wang

Pulkit Goyal

Pratibha Tripathi

Rahul Ramakrishnan

10th November 2017

Individual Progress:

As my contribution to the MRSD Project-1 since the past progress review, I have worked on bebop2 drone autonomous flight based on GPS waypoints navigation. I have also completed the power distribution board CAD for the Husky. I have been successfully able to test the GPS accuracy of the drone and found it to be well within our system requirement. Additionally, I have performed few flight testing to test the waypoint navigation capabilities of the drone.

Bebop2 Drone GPS testing:

One of the challenges in GPS based autonomous navigation of drones is the accuracy in reaching the final goal location. This is partly because of the localization inaccuracy of the GPS receivers used in the drones. Our mission's requirement is to reach the final goal location within an accuracy of 5m radius hence we wanted to test the capabilities of our aerial system for meeting this requirement.

After connecting the drone to our system and launching the bebop2 driver core, we used the following command to receive the GPS information of the Bebop2 in NavSat format by echoing to the topic 'fix':

rostopic echo /bebop/bebeop/fix

Below is a sample output of GPS data:

```
latitude: 40.4435528333
longitude: -79.9467903333
altitude: 270.0
position_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
position_covariance_type: 0
---
header:
  seq: 31
  stamp:
    secs: 1510352952
    nsecs: 406833995
  frame_id: /gps
status:
  status: 0
  service: 3
latitude: 40.443555
longitude: -79.9467918333
altitude: 270.0
position_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
position_covariance_type: 0
---
```

Figure 1: GPS data received from Bebop2

Bebop2 Drone Waypoint Navigation:

The drone comes with a proprietary software with which we can make Bebop2 to fly autonomously, create waypoints, and record videos with unparalleled clarity and stability. This feature however is not available in developer mode and needs to be purchased as an in-app purchase on the Parrot's official FreeFlight app. Another problem with the flight plan feature is that even after downloading the waypoints .xml file, it cannot be dynamically updated while running Bebop2 on a ROS node.

We have performed few flight tests to test the 'Flight Plan' capabilities of the drone but unfortunately, they are not as accurate as we expected them to be. There is an inaccuracy of 3m radius which seriously affects the waypoints we can set on the app. The downside is that the waypoint selected does not incorporate the drift of the drone and wind conditions and hence is not accurate.

Below is a snapshot of the flight plan we generated for a test flight using Parrot's proprietary software, Free Flight3

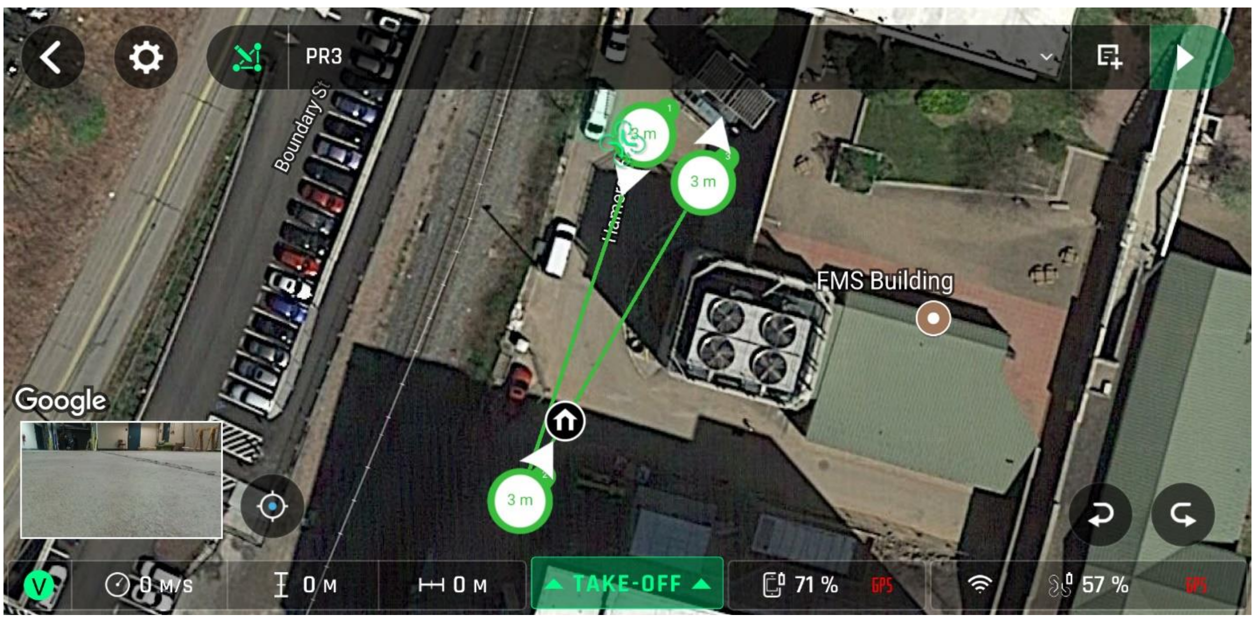

Figure 2: Waypoints generation and autonomous flight testing

The flight plan generated by the app is stored in bebop2 and hence can be downloaded to a system running a FTP client.

The Flight Plan is generated in the MAVlink file format which is shown below:

```
QGC WPL <VERSION>

<INDEX> <CURRENT WP> <COORD FRAME> <COMMAND> <PARAM1> <PARAM2> <PARAM3> <PARAM4>
<PARAM5/X/LONGITUDE> <PARAM6/Y/LATITUDE> <PARAM7/Z/ALTITUDE> <AUTOCONTINUE>
```

An example of the flight plan generated and saved on Bebop2 is shown below:

flightPlan.mavlink

```
QGC WPL 120
0	0	3	2500	0.000000	30.000000	2073600.000000	0.000000	0.000000	0.000000	0.000000	1
1	0	3	178	0.000000	6.000000	-1.000000	0.000000	0.000000	0.000000	0.000000	1
2	0	3	16	0.000000	5.000000	0.000000	360.000000	40.443367	-79.946922	3.000000	1
3	0	3	50000	0.000000	-1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1
4	0	3	21	0.000000	0.000000	0.000000	360.000000	40.443367	-79.946922	3.000000	1
5	0	3	2501	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1
```

Figure 3: Flight Plan data in MAVlink format

We decided to get rid of the app-based GPS waypoint generation and have started working to develop our own flight plan generation code by estimating the distance between the goal GPS position and the current Bebop location.

To test the preliminary structure of the code, we decide to make the bebop move towards a desired GPS location and land after reaching within 5m of the goal location (to account for the GPS data inaccuracy). However, we haven't got good results and the drone shot across the goal location.

<u>Power Distribution Board:</u>

Initially we decided to use linear switching regulators for DC power conversion in our power distribution board. However, as correctly pointed out by Luis, we were getting lot of power losses in those regulators. To avoid that, we decided to use power DC-DC buck converters. This significantly improved our efficiency by reducing the power losses in the converters.

Also, we had initially placed the decoupling capacitors far away from the regulators which essentially should be placed as close as to the IC as possible. We rectified this in our new design and incorporated Luis's suggestions to place the Inductors far away from the feedback pin.

After running the DRC checks, we visualized our design on gerber-viewer.com. Below is the visualization of our PCB board:

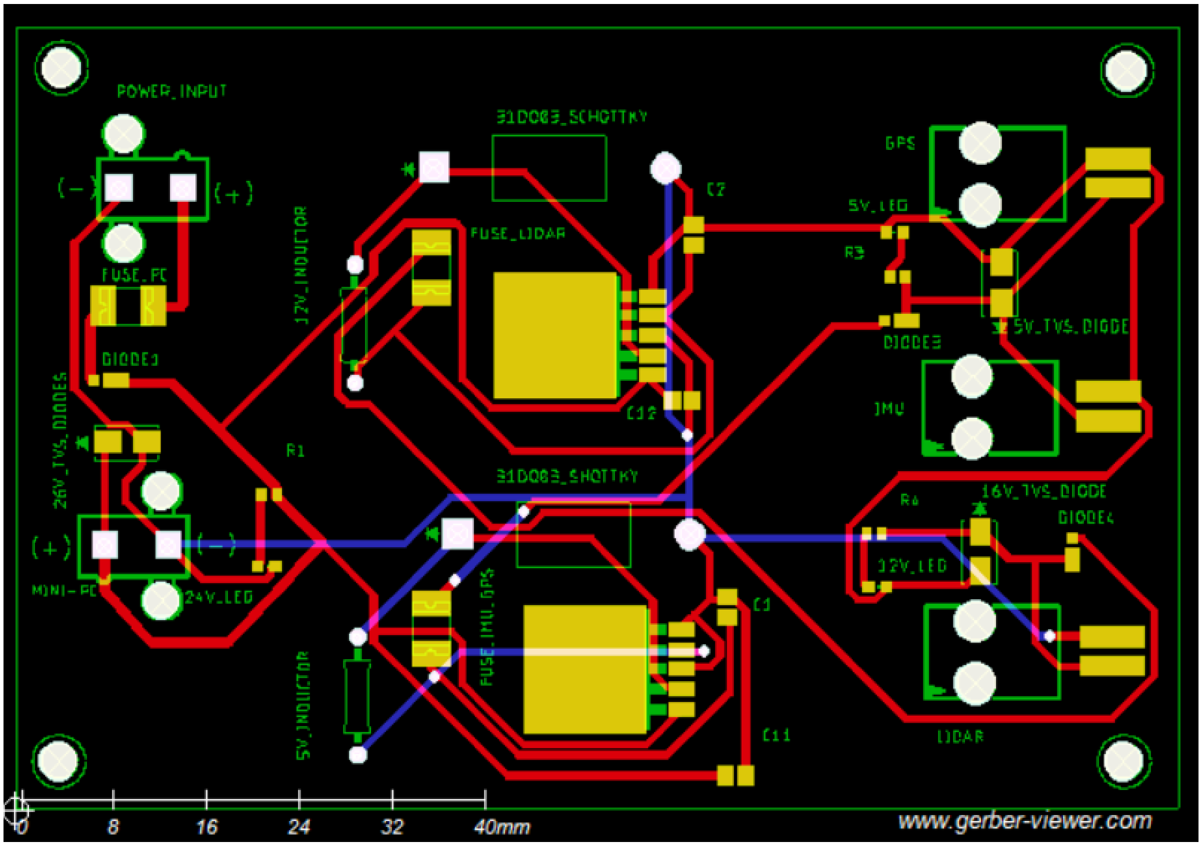

Figure 4: Final design of Power Distribution Board for Husky

Challenges

For this PR, the biggest challenge we faced was the unavailability of the waypoint generation feature in Bebop2 and insignificant online support for the same. We initially assumed waypoint navigation to be fairly easy task and didn't expect it to proprietary feature that is incompatible with ROS. We are currently trying to generate our own MAVlink files that can directly be read by the ROS core while at the same time generating our own control code for GPS based navigation.

Another difficulty we faced was the recent change in our work space. Due to many new people coming in to George's lab, we had to shift our entire system downstairs during which we lost a tool for changing the propellers of Bebop2. Fortunately, we had a spare tool which came in handy when one of our propeller blades chipped off.

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. To mitigate this, we have procured few spare batteries that can be replaced in no time.

Another big challenge the team is facing is the inaccuracy for April Tag detection. April Tag detection is a very essential aspect of our system and we are currently getting a lot of noise while finding the Aptil Tag transformations. We plan to improve this by using extended Kalman filter and will test out the outcomes soon.

Teamwork

Yuchi worked with me on testing the GPS accuracy for Bebop2 drone. We also worked together on the autonomous navigation of Bebop2. Additionally, he helped out Pulkit in setting up the Host connection and Bebop2 networking issue with Pulkit. Yuchi was able to successfully detect multiple April Tags and the relative distance between them from the Bebop2's camera feed.

Pratibha helped me in designing the final power distribution board design for the Husky. She worked with Pulkit to set up the IMU and Encoder data interface for the Husky. Pulkit and Pratibha tested the range of the WiFi Host network with the onboard Husky PC.

Rahul and Pratibha worked together to set up the mechanical mount for the LIDAR and camera on Husky. Rahul worked on Velodyne Puck LIDAR for implementing obstacle detection. He also worked for GPS NMEA data conversion and publishing it as a ROS topic.

Pulkit was responsible for the WiFi network setup and successfully accomplished it. He worked with Pratibha for getting IMU and Encoder data for Husky and with Yuchi to setup Bebop's network. He also tested the teleoperation of Husky on the Host network set up by him.

Thus, by defining each member's goal successfully and working together as a strong team, we could achieve all the tasks for the PR-3.

Future plans

As my next PR goal, I will continue my work on high level control of the drone. I am planning to explore how can we generate MAVLink files for Bebop's autonomous flight in ROS. Parallelly, I am planning to develop a controller that takes in GPS coordinates inputs and controls the flight parameters of the drones to reach the target location. Yuchi and Rahul will work to accurately localize the April Tag transforms. Pratibha and Pulkit plan to work towards setting up the WiFi network connection for Bebop2.