Individual Lab Report- 3

By Danendra Singh

Team F: Falcon Eye

Danendra Singh

Yuchi Wang

Pulkit Goyal

Pratibha Tripathi

Rahul Ramakrishnan

27th October 2017

Individual Progress:

As my contribution to the MRSD Project-1 since the past progress review, I have worked on bebop2 drone and controlled it via ROS interface using a game controller. I have also designed the layout of the power distribution board for the Husky. I have been successfully able to control the drone with commands from the ROS package. Additionally, I have performed multiple flight testing with the drone to check the robustness of the Wi-Fi connection between the drone and the ground PC.

Bebop2 Drone Connectivity:

The drone comes with a IEEE standard 802.11a/b/g/n/ac Wi-Fi having a MIMO Dual band connectivity. The added advantage is that the drone has 2.4 and 5 GHz dual dipole antennas with a maximum output power of 21dB

Although, the technical specifications claim a range of 300m, we have received significant signal lag and instability even within a 20m vertical distance between the laptop and the drone. We have found out that without using a Wi-Fi booster, although we have continuous connection up to 15-20m but there is a significant lag in the video stream.

One possibility is that, the range claimed is only with the specified controller supplied by the AR drone company, the manufacturers of Bebop2.

We have also procured a USB Wi-Fi adaptor with 2.4GHz, 802.11ngb specifications and manufactured by Bolse. We can attach a required high gain antenna to the Female RP-SMA connector provided in the Wi-Fi adaptor.

Figure 1: Bolse Wi-Fi adaptor

ROS Joystick control of Bebop2 Drone:

I used ‘bebop_autonomy’ ROS package to connect to bebop2. It is based on Parrot’s official ARDroneSDK3 and is supported by the online ROS community. The steps I followed to interface Bebop2 with ROS joystick control are as follows:

1. Identify the controller USB ID using
    a. lsusb
2. Edit the controller ID in the ‘gedit joy_teleop.launch’ launch file
3. In the bebop_ws, go to bebop_tools/config folder
4. Launch the ‘joy_teleop’ package using
    a. roslaunch bebop_tools joy_teleop.launch
5. Check whether the controller is functioning before launching the drone by
    a. rostopic echo/bebop/joy
6. Control the drone by sending the controller inputs from the joystick
7. The output of the joystick is from -1 to +1
8. To get the video frame from the drone use
    a. rqt_image_view

Below is a snapshot of the test flight using game controller

Figure 2: Drone Tele-Op using a Game Controller with ROS

Power Distribution Board:

With the inputs we received from Luis regarding the previous power distribution board, I incorporated his suggestions to design a new power distribution board to be used with the Husky. Although, we already have stable power supply output from the husky's internal DC converters but to provide more robust operation, we decided to separate the power inputs of the sensors from the power inputs to the motors. Hence, we will be using two 24V batteries, one for powering the motors and the other for the sensors and Mini-PC.

After rectifying the design faults from previous Power Distribution Board task, I performed the design calculations for the DC-DC converters. For Velodyne Lidar, GPS and IMU we used COTS power DC-DC converters of 12V and 5V in the schematics. However, since the operating range of the Mini PC is 18-24 V, we couldn't find any cost effective and robust power converters in that range. Hence, we decided to power up the PC directly from the 24V battery.

I calculated the thermal efficiencies of the DC-Converters and also the power loss in each converter. These are necessary to avoid excessive heating of the converters which can be minimized using a Heat Sink of appropriate size and material. Below is the schematic layout of our Power Distribution Board.

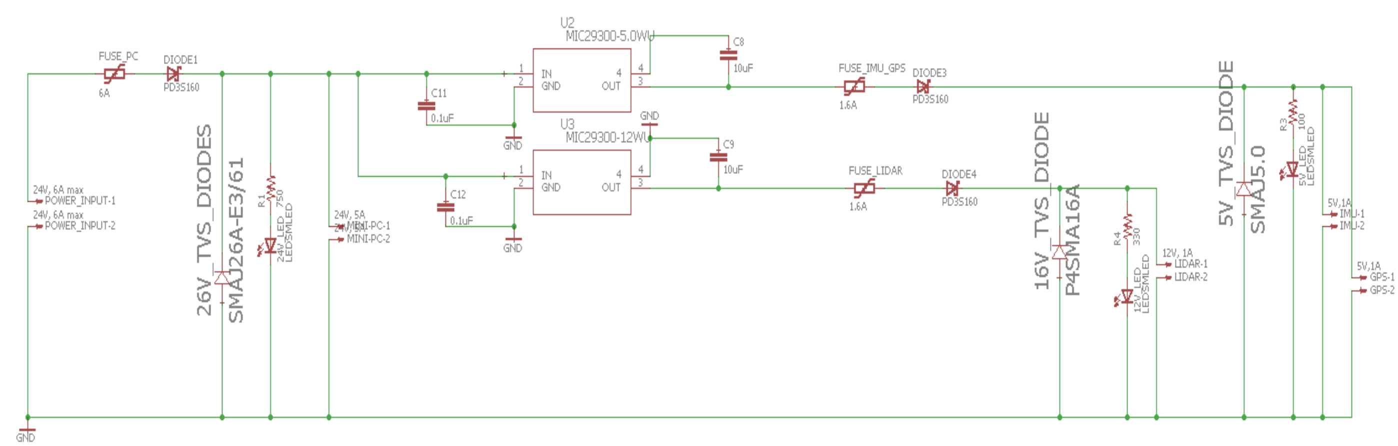

Figure 3: Power Distribution Board Schematic

Challenges

For this PR, the biggest challenge we faced was the lack of time to work together for the project. Out of the 5 members, 4 of us had 5 submissions this week which made it difficult to collaborate for the project. So, we decided to sit together over the previous weekend and discuss the action plan for the next week. This was an extremely important and necessary task as it allowed us to figure out a feasible plan which incorporates all the tasks and allows proper division of work. As a result, we worked extremely efficiently and delivered upon all the planned tasks for the Progress Review (PR)-2.

Another difficulty I faced was interfacing the game controller with Virtual Box Linux, The USB ID of the game controller is randomly generated in the virtual box and hence to fly each time, we have to configure the new ID and update it in the bebop ROS launch file. To get rid of this problem I removed all USB connections that were already setup in the Virtual Box and only allowed one unnamed connection. This solved the problem of finding the unique USB ID for the joystick.

During the flight testing of the Bebop2 drone, as we picked up altitude, we encountered lag in controller response and missed some video frames from the drone. We figured out that this could be possibly because of the small range of the laptop Wi-Fi. Hence, to mitigate this problem we have decided to use a Wi-Fi module which can be used with a high gain antenna. We will be testing further with the new setup and are hopeful in increasing the range of operation.

Teamwork

Yuchi worked with me on controlling the Bebop2 drone with a game controller. Additionally, he interfaced Radio Link GPS module with Arduino and was able to receive GPS data in NMEA format.

Pratibha worked in designing the PCB schematic and board design for the Husky power distribution board. She worked with me in incorporating the power distribution calculations and selection of appropriate devices. She also worked on the Husky Tele-op with a game controller.

Pulkit and Rahul worked on designing the mechanical CAD for the sensor placement on the Husky. They worked extensively to finalize sensor modules and visualize their placement in CAD. Additionally, Pulkit integrated the game controller with ROS for the movement control of the husky. Rahul also worked on Velodyne driver bag files and was able to visualize the output in rviz for calibration of the parameters.

Thus, by defining each member's goal successfully and working together as a strong team, we could achieve all the tasks for the PR-2.

Future plans

As my next PR goal, I will continue my work on high level control of the drone. I am planning to implement autonomous GPS based navigation for the drone. Using bebop2's ROS package, I aim to send target GPS coordinates to the drone and check for the accuracy of the drone performing the task successfully.

Yuchi plan on continuing his work on the GPS module and also localize the drone using April Tags. Pratibha and Pulkit plan to work to autonomously move the Husky based on the encoder data. In addition, Pulkit will also be working to further develop the mechanical CAD design with Rahul. Rahul will be mainly working with the Velodyne data calibration and continue with the testing phase of the Lidar.