Individual Lab Report #3

Progress Review 2

February 28, 2025

Joshua Pen

Team B

Teammates: Gweneth Ge, Lance Liu, Yi Wu, Jet Situ

Table of Contents

Contents

1 Individual Progress

1.1 PCB Assignment

In our recent project, Jet led the PCB assignment, ensuring its successful completion. Meanwhile, Gweneth and I collaborated closely on several key deliverables. Together, we documented our team's progress on the MRSD website, created the project management slides, developed the Lessons Learned Retrospective slides, and assembled the Progress Review slides. These efforts facilitated a comprehensive evaluation of our project's outcomes and identified areas for future improvement.

1.2 MRSD Project

For the MRSD project, I designed, integrated, and tested the Path Planner for searching the Geofence Zone and conducting the Local Search of the Patient Planner using QGroundControl. For testing purposes, the Path Planner for searching the Geofence Zone generated a geofence near CMU. It then calculated the minimum and maximum longitude and latitude values to define a bounding box. This bounding box was populated with waypoints based on key parameters such as robot altitude, camera field of view (FOV), percent overlap, and margin from the geofence border to ensure appropriate waypoint density. Any waypoints outside the geofence were removed. The finalized waypoint list was then published as a mission to QGroundControl, enabling the drone to navigate to each waypoint sequentially, pausing for five seconds at each location before proceeding to the next. These waypoints were displayed on the QGroundControl interface.

The following equations describe how the waypoint density is calculated within a geofence. Given an altitude h and a camera field-of-view θ (in radians), the half-width of the camera footprint on the ground is:

$$w_{\text{half}} = h \cdot \tan\left(\frac{\theta}{2}\right)$$

Thus, the total footprint width is:

$$w = 2\,h \cdot \tan\left(\frac{\theta}{2}\right)$$

To convert the footprint from meters to degrees:
For latitude:

$$\Delta\text{lat} = \frac{w}{111320}$$

since 1° latitude is approximately $111320\,\text{m}$.
For longitude at latitude ϕ:

$$\Delta\text{lon} = \frac{w}{111320 \cdot \cos(\phi)}$$

If a desired image overlap O is specified (expressed as a fraction, e.g. 0.6 for 60% overlap), the effective step sizes become:

$$s_{\text{lat}} = \Delta\text{lat} \cdot (1 - O)$$

$$s_{\text{lon}} = \Delta\text{lon} \cdot (1 - O)$$

Let the geofence be bounded by:

$$\text{lat}_{\text{min}}, \quad \text{lat}_{\text{max}}, \quad \text{lon}_{\text{min}}, \quad \text{lon}_{\text{max}}$$

and let a minimum margin d_{min} (in meters) be converted as:

$$\Delta\text{lat}_{\text{min}} = \frac{d_{\text{min}}}{111320}, \quad \Delta\text{lon}_{\text{min}} = \frac{d_{\text{min}}}{111320 \cdot \cos(\phi)}$$

Then the number of passes (rows and columns) is:

$$N_{\text{lat}} = \max\left(1, \left\lfloor \frac{\text{lat}_{\text{max}} - \text{lat}_{\text{min}} - 2\,\Delta\text{lat}_{\text{min}}}{s_{\text{lat}}} \right\rfloor\right)$$

$$N_{\text{lon}} = \max\left(1, \left\lfloor \frac{\text{lon}_{\text{max}} - \text{lon}_{\text{min}} - 2\,\Delta\text{lon}_{\text{min}}}{s_{\text{lon}}} \right\rfloor\right)$$

For each row i (with $i = 0, \ldots, N_{\text{lat}} - 1$), the latitude is computed as:

$$\text{lat}_i = \text{lat}_{\text{min}} + \Delta\text{lat}_{\text{min}} + i \cdot \frac{\text{lat}_{\text{max}} - \text{lat}_{\text{min}} - 2\,\Delta\text{lat}_{\text{min}}}{N_{\text{lat}}}$$

For each row, the longitude for column j depends on the row's parity:

$$\text{lon}_{ij} = \begin{cases} \text{lon}_{\text{min}} + \Delta\text{lon}_{\text{min}} + j \cdot \frac{\text{lon}_{\text{max}} - \text{lon}_{\text{min}} - 2\,\Delta\text{lon}_{\text{min}}}{N_{\text{lon}}}, & \text{if } i \text{ is even}, \\ \text{lon}_{\text{max}} - \Delta\text{lon}_{\text{min}} - j \cdot \frac{\text{lon}_{\text{max}} - \text{lon}_{\text{min}} - 2\,\Delta\text{lon}_{\text{min}}}{N_{\text{lon}}}, & \text{if } i \text{ is odd}. \end{cases}$$

The resulting set of waypoints is:

$$\{(\text{lat}_i, \text{lon}_{ij}) \mid i = 0, \ldots, N_{\text{lat}} - 1,\ j = 0, \ldots, N_{\text{lon}} - 1\}$$

arranged in a zigzag (back-and-forth) pattern to minimize travel distance.

For the Local Search of the Patient Planner, a random GPS location near CMU was assigned as a hypothetical patient location. A delay was implemented to simulate the drone traveling to that GPS location. Once the drone reached the designated location, a circular geofence was generated around the target GPS point. The same waypoint generation method used in the Geofence Zone Path Planner was then applied. The waypoint list was published as a mission to QGroundControl, allowing the drone to visit each waypoint sequentially, holding for five seconds at each location. These waypoints were also displayed on the QGroundControl interface.

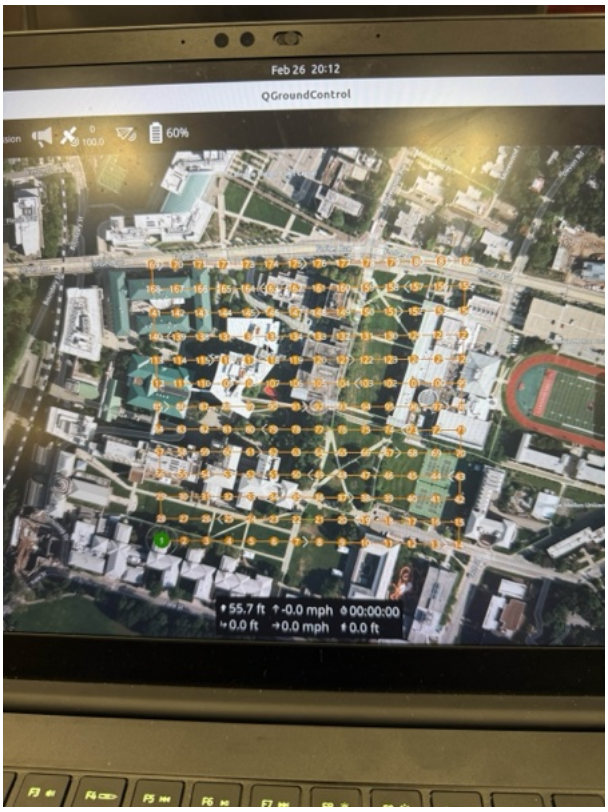

Figure 1: Path Planner for searching the Geofence Zone on QGroundControl

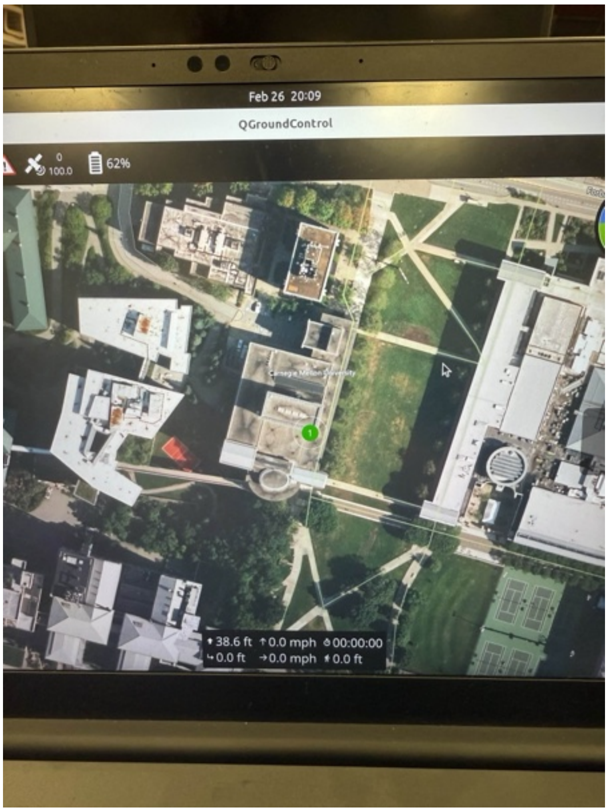

Figure 2: Local Search of the Patient Planner on QGroundControl(Initial Patient Target Waypoint)

Additionally, I worked on the initial integration of the Path Planner for the Geofence Zone search into the ROS2 network for flight testing, though it has not yet been integrated with the patient detection system. To facilitate controlled testing, the Path Planner was modified to generate a 5x5 meter geofence at the center of the NREC drone cage. This geofence was populated with waypoints using the same algorithm as before, with waypoints transmitted one by one. After publishing each waypoint, the node subscribed to the drone's location to verify it was within range before waiting five seconds and publishing the next waypoint. The five second delay is meant to simulate running the patient detection algorithm at each waypoint.

To improve hardware reliability, I designed and installed a new Orin and GPS mount. This modification was necessary after the decision to transition away from the Doodle Labs radio due to poor connectivity. Additionally, the GPS mounting location was adjusted to the center-top of the drone to enhance GPS signal accuracy, as previous weak signals had prevented the drone from arming. I also replaced broken extension legs (3D printed), the gimbal attachment mount (3D printed), and a damaged motor (sourced from a retired DJI drone in storage) to ensure the drone was flight-ready.

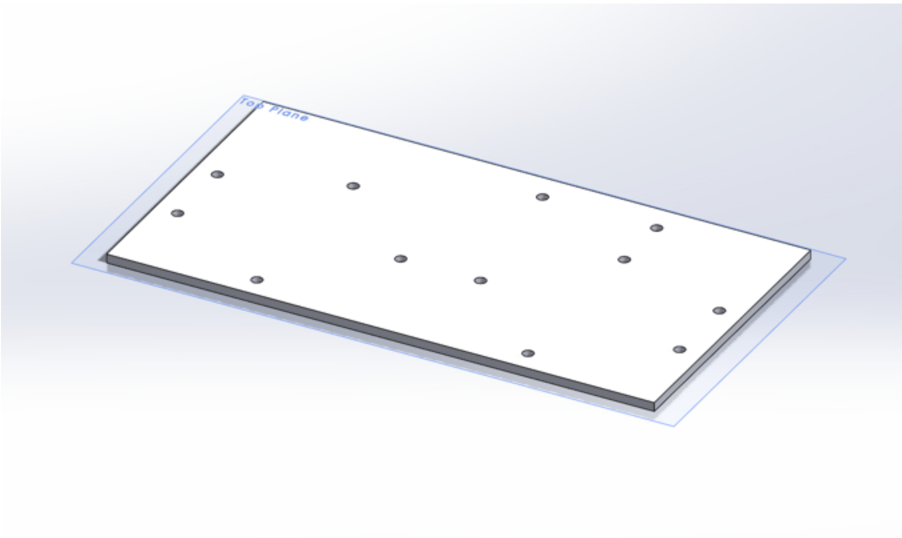

Figure 3: Orin and GPS Attachment Mount

Following these preparations, I set up and conducted an autonomous flight test at NREC. We successfully demonstrated autonomous takeoff and landing, manual override during flight, communication loss/E-stop behavior, geofence breach response (which should hold the flight midair), and the autonomous takeoff, waypoint navigation, and landing sequence. Additionally, I initiated the development of a new Foxglove GUI to enhance data visualization and debugging for future missions.

2 Challenges

2.1 PCB Assignment

In our recent project, I did not encounter any challenges with the PCB assignment, as Jet effectively managed this aspect. Regarding the MRSD website, Project Management slides, Lessons Learned Retrospective slides, and Progress Review slides, Gweneth and I collaborated closely and did not face any major challenges.

2.2 MRSD Project

One of the main challenges I faced while working on the MRSD project was software integration. Specifically, I needed to determine which ROS2 topics and services to subscribe to in order to obtain geofence information, robot GPS location, and status updates regarding patient detection. With so many available topics and services, I spent considerable time reviewing documentation and fine-tuning my code to correctly receive the required data through subscribers, as well as ensuring that the published waypoints were formatted correctly.

The most significant roadblock was subscribing to the correct topic or service for geofence coordinates. Our initial plan was to draw the geofence in QGroundControl and have my ROS2 node subscribe to receive the geofence data for testing. However, I encountered persistent issues with QoS mismatches and difficulties keeping the node spinning—my node would stop spinning after only one second. Although I verified using ROS2 topic information and echoing the geofence topic that valid geofence GPS coordinates were being published and that my node was subscribed, the node was not receiving the data as expected.

Additionally, we faced hardware challenges. For instance, we encountered a broken gimbal wire that we attempted to repair at a microfabrication lab under a microscope; unfortunately, the repair was unsuccessful and we had to reorder the gimbal. This delay impacted the development of our patient detection code and postponed our flight test. During the test flight at NREC, we experienced another issue where two motors were spinning slightly slower than expected. We later discovered that the remote controller had been inadvertently touched by others at AirLab, which disrupted our calibration. Furthermore, we encountered multiple instances where our GPS signal would cut out—possibly due to overhead aircraft or general signal degradation—resulting in the inability to arm the vehicle.

3 Team Work

3.1 MRSD Project

Table 1: Team Members and Their Contributions

Name	Contribution
Jet Situ	Designed and implemented inter-subsystem electrical and data interfaces, providing internal data transfer infrastructure for drone sensor fusion. Implemented new radio interface for long-range communication over both MAVLink and a LAN-linked signal. Provided software architecture design and debugging assistance, as well as software QA prior to test flight. Assisted in debugging hardware and electrical issues prior to flight. Developed and executed pre-flight plans, test plans, integration plans, and safety tests. Coordinated with the FAA to schedule flights and demos, and practiced drone piloting for future tests. Designed the PCB schematic.
Joshua Pen	Designed, integrated, and tested Path Planner for searching the Geofence Zone and Local Search of Patient Planner with QGroundControl. Initial integration of Path Planner for searching the Geofence Zone to ROS2 network for flight test (not integrated with Patient Detection yet). Designed and installed new Orin and GPS mount. Repaired or replaced broken extension legs, gimbal attachment mount, and motor before flight test. Setup and conducted autonomous flight test. Initialized development of the new Foxglove GUI. Contributed to project management and logistics. Collaborated on various presentations including progress reviews, project management updates, and retrospectives.
Lance Liu	Built and fine-tuned BehaviorExecutive – a ROS node that uses behavior trees to orchestrate drone operations. This node handles essential drone functions including auto-takeoff, auto-landing, arming/disarming, emergency hold (E-stop) with priority interrupt, navigation (altitude climbs and go-to-position commands with coordinate transforms), ROS plumbing (subscribers for GPS, altitude, and state telemetry; service clients for MAVROS integration; async handlers for non-blocking command execution), and a real-time execution control loop (10 Hz) to monitor active actions and service call status. Error handling was particularly improved by catching service timeouts, coordinate jumps, and FCU rejections with proper fallbacks.

Continued on next page

Table 1 – Continued

Name	Contribution
Gweneth Ge	Primarily worked on project management and safety features required by DARPA for the March 7th workshop. Specifically, integrated Remote ID for continuous flight broadcasting, implemented a battery level monitor in QGroundControl to trigger a return-to-home when low, and fixed compass position issues to improve accuracy. Collaborated on various presentations including progress reviews, project management updates, and retrospectives.
Yi Wu	Modified human pose estimation code to conform to the standard base class in AirLab's HumanFlow repository. Designed ROS2 packages for gimbal control on both x86 and ARM systems (including angle control, zooming, and video streaming). Developed a patient lock-on algorithm featuring person detection, tracking, and visualization components. Initiated the Isaac Sim pipeline and established communication between Isaac Sim and ROS2.

4 Plans

4.1 MRSD Project

Name	Contribution
Jet Situ	Will assist all software personnel with debugging and software QA and integration across several different stacks. Will coordinate with DARPA and other authorities to ensure proper protocols are followed prior to test. Will assist in coordinating and running all flight tests, and robustify existing electrical components. Finally, will represent MRSD Team at the DARPA Workshop in Georgia from 03-07 to 03-15
Joshua Pen	Fully integrate and test Path Planner for searching the Geofence Zone and Local Search of Patient Planner with QGroundControl with ROS2 network (also integrate with Patient Detection), Design, integrate and test Triage Planner. Design new mount for Rajant DX2 radio. Contribute to wire management, project management, and logistics.
Lance Liu	Implement dynamic geofence boundary subscription (replace hardcoded coordinates). Develop waypoint progress monitoring. Gimbal control integration: pixel-to-geodetic conversion, gimbal/FCU Sync, streaming, gimbal tracking. Robust autonomous flight: bugs fixing, performance fine-tuning and optimization.
Gweneth Ge	Integrate and test Foxglove as the GUI for easier commands to drone by Mar. 3rd, which will be used in workshop starting on Mar. 7th. Continue working on overall project management and meetings arrangement with the sponsor Airlab, as well as NREC and Mill19 to align upcoming flight tests and the DARPA Triage Challenge.
Yi Wu	Upgrade the gimbal control code, including enhancing the tracking logic design, improving the speed of camera tracking, and adding human pose estimation into the gimbal control. Integrate the gimbal control with other subsystems, like controlling the gimbal based on upstream path planning behavior designs.

Table 2: Team Members and Their Plans