# Individual Lab Report #3

Progress Review 2

February 28, 2025

**Yufan (Lance) Liu**

Team B

Teammates: Gweneth Ge, Joshua Pen, 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

### 1.2.1 Autonomy Stack– BehaviorExecutive Node

This week, I made a significant progress for the team in developing and fine-tuning the BehaviorExecutive ROS node, a critical component of the drone autonomy stack. The node uses behavior tree (BT) architectures to orchestrate complex autonomous drone operations reliably and robustly.

Core autonomy functionalities have been implemented: The BehaviorExecutive node comprehensively manages key drone operations, including:

- Automated Flight Operations:
  - Auto-Takeoff: Reliable transition from ground idle to airborne state with altitude verification.
  - Auto-Landing: Controlled descent and touchdown sequence ensuring positional accuracy.
  - Arming/Disarming: Safe motor engagement and disengagement following operational checks.
- Emergency Handling and Interrupt Priority:
  - Integrated *Emergency Hold (E-stop)* feature with priority interrupts, enabling immediate response to critical conditions and instantaneous system hold.
- Navigation and Coordinate Management:
  - Precise altitude adjustments (*altitude climb/descent*) commands.
  - Accurate execution of *go-to-position* commands involving global-to-local coordinate transformations, ensuring precise navigation and positional accuracy.

ROS Communication & Integration: Comprehensive ROS interfaces were established for seamless integration:

- ROS Topic Subscribers:
  - GPS telemetry (`/mavros/global_position/global`)
  - Altitude telemetry (`/mavros/altitude`)
  - Drone state telemetry (`/mavros/state`)

- ROS Service Clients for MAVROS Integration:
  - Mode management
  - Waypoint navigation requests and position setpoints
- Asynchronous Handlers:
  - Implemented non-blocking service clients enabling concurrent execution of drone commands without task interruptions.

# 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

During the integration and testing phases, several challenges emerged, primarily due to unforeseen behaviors in low-level firmware and compatibility issues in the deployment environment.

### 2.2.1 Individual: Unexpected Auto-Takeoff Mode Errors

A recurring issue was observed where the drone's automated takeoff sequence would unexpectedly abort mid-operation, triggering an immediate auto-landing failsafe response. A thorough analysis of the PX4 source code revealed that this failsafe behavior is activated whenever the `local_position` estimator reports an invalid or unstable position estimate. Upon inspection of PX4 log files, the yaw estimator status parameter (`yaw_estimator_status/innov_ve`) exhibited sharp spikes at the exact moments when the failsafe was triggered, suggesting that the velocity innovation check within the Extended Kalman Filter (EKF) responsible for yaw estimation was likely failing due to transient sensor discrepancies or noisy input data. Further investigation into sensor fusion algorithms and EKF tuning parameters is currently ongoing to resolve this intermittent behavior.

### 2.2.2 Individual: Docker Build Complexity on ARM Architecture

Deploying the ROS2 gimbal-control Docker container onto an ARM64 (NVIDIA Jetson AGX Orin) architecture posed significant challenges. The main difficulty stemmed from incompatibility and dependency issues related to the ARM-specific build environment, causing persistent "file not found" errors despite seemingly correct package installations. Extensive manual checks were necessary to verify each installed package's compatibility and correct architecture linkage, resulting in approximately seven hours of troubleshooting. Eventually, these issues were mitigated by meticulously specifying library paths, environment variables, and employing a carefully configured multi-stage Docker build process explicitly tailored for ARM64 architecture.

### 2.2.3 Teamwise

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

| 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. |
| 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. |
| 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. |



Table 1: Team Members and Their Contributions

# 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