Individual Lab Report #1

Sensors and Motors Lab

February 08, 2025

Yufan (Lance) Liu

Team B

Teammates: Gweneth Ge, Joshua Pen, Yi Wu, Jet Situ

Table of Contents

Contents

1 Individual Progress

1.1 Sensors and Motor Lab

I did two parts in this Sensors Lab, from the usage of ultrasonic sensor, DC motor, control of DC motor to code integration. The ultrasonic sensor has a range of 10 to 645 cm, then mapped to DC motor's 360 degree rotation range. For ultrasonic sensor, I applied an unweighted 5-step moving average filter.

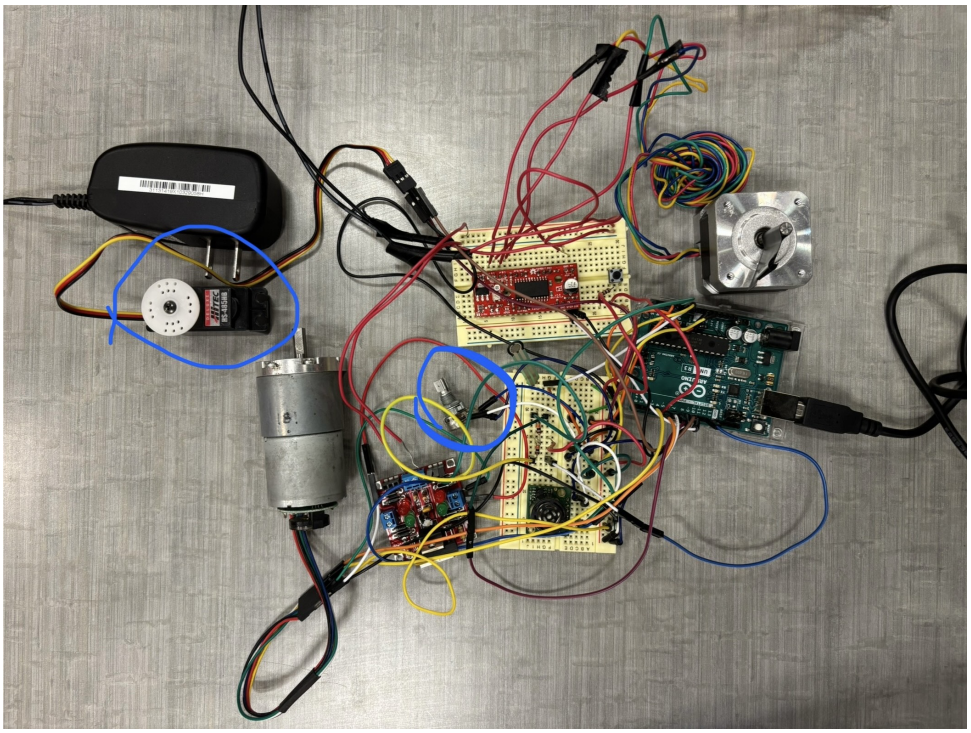

Figure 1: The Entire Circuit

Since the picture only shows a final integrated circuit, I will try illustrating the wiring with language. The H-bridge driver has 12 Voltage power source in its power input vin and gnd. DC motor's power pin connects to H-briged dirver's power output. DC motor's hall A and hall B pin connect to a 10k pull-up resistor respectively, then to arduino's pin 2 and 5. The other pin of each resistor is connected to arduino's 5V power. Finally, the hall vcc and hall gnd goes to arduino's 5v and gnd.

The ultrasonic sensor reads a distance input, which then gets translated to a target position in the motor's encoder unit. Since each full turn, i.e., 360 degree corresponds to roughly 5000 units on encoder position, we are able to get a linear mapping from detected distance to target truning angle on the motor.

Specifically, the ultrasonic sensor reading is divided by a voltage-to-distance conversion factor and gets added smoothed by a moving average filter to reduce noise. The filtered distance is then mapped from (0, 645) cm to (0, 5000) encoder positions to get a target encoder position.

```
if (state == 3) {
    // Read and filter ultrasonic sensor
    int adcValue = analogRead(SONAR_PIN);
    float voltage = (adcValue * VCC) / 1023.0;
    float rawDistance = (voltage / SCALE) * 2.54;
```

```
6
7       // Update moving average
8       total = total - readings[readIndex];
9       readings[readIndex] = rawDistance;
10      total = total + readings[readIndex];
11      readIndex = (readIndex + 1) % FILTER_SIZE;
12
13      // Calculate filtered distance
14      float filteredDistance = total / FILTER_SIZE;
15
16      if(counter > 200) {
17          Serial.print("ULTRA:");
18          Serial.println(rawDistance);
19          Serial.print("FILTERULTRA:");
20          Serial.println(filteredDistance);
21      }
22
23      // Map the filtered distance to a target position
24      int target = map(filteredDistance, 0, 645, 0, 50000);
25
26      if (counter % 50 == 0) {
27          motor1.set_target(target);
28      }
29
30      motor1.start_pos_pid(dc_pos_kp, dc_pos_kd, dc_pos_ki);
31      long current_pos = motor1.get_position();
32      long error = abs(current_pos - target);
33
34      if (error <= 100) {
35          motor1.turn_off();
36      }
37  }
```

Listing 1: Ultrasonic Sensor-Based PID Motor Control

A PID controller then adjusts the motor's position based on the error between the current position and the target position.

1.2 MRSD Project

I contributed to three major parts of rebuilding the NDAA-compliant drone, acounting for a big part of the total work. For embedded architecture design, I assisted Jet on structuring the system, I integrated all hardware-software interfaces, and debugged ensuring compatibility between components. Participated in all iterations testing and refinement, and execution of every test flight while fixing issues encountered during testing.

After the first qualification video, I deployed and continued developing the AirLab codebase on our platform. I adapted Docker containers for the ground control station, the onboard robot, and the simulator to fit our task dependencies and ARM computer platform architecture. Once the containers were running, I set up the MAVROS bridge to establish communication between the autopilot Cube, onboard computer (Orin), radio link, and ground control station. I validated the setup by ensuring proper sensor data transmission and control command execution.
With the communication framework in place, I proceeded to develop the behavior tree module, which is the decision-making hierarchy for autonomous flight. My next step is

to deploy the autonomous flight modules and conduct testing in IsaacSim.

For everyone:
On January 15th, we sent a series of flight videos of our drone to DARPA. The videos successfully showcased the drone's autonomous flight and E-stop capabilities with QGroundControl, both during daytime and nighttime. All system components we used are NDAA-compliant, and our team also obtained the FAA Part 107 certification to ensure safe outdoor operations.

Following the initial DARPA demonstration, we rebuilt the drone system to enhance its robustness and stability. This process involves the reintegration of the BlueCube with Pixhawk2 ardupilot, NVIDIA Jetson Orin computing unit, Doodle Labs radio system, GPS module, and camera gimbal. Mechanically, our team has 3D printed and mounted all key components mentioned above. In addition, we have mounted the extension legs,protective snow shield, props and propellers to ensure reliable operation.

We made a good progress on the software development. We containerized two main components using Docker: the Ground Control Station(GCS) and the Robot Container. Both components were built on Ubuntu 22.04 with ROS2 Humble. The GCS is responsible for real-time monitoning, mission management and remote control of the drone; The Robot Autonomy System manages onboard operations and composed of several critical modules, including MAVROS Interface, Robot bringup and Planner. Additionally, we successfully established radio communication, Orin-GCS into network with time server recognition. The launch file for both the Orin GCS interface and the autopilot system have also been completed.

2 Challenges

2.1 Sensors and Motor Lab

The primary challenge I encountered during the sensor and motor lab was getting stable encoder position signal from DC encoder. One of the issues arose due to integer overflow when storing the encoder's position with an int type. Since the encoder returns increasing/decreasing position coundts, the integer flipped the sign unpredictably – e.g., from 32000 to -32000 – which caused shaky motion behavior and confusion for debugging. The problem was difficult to detect because the sign flipped only when reaching a large count value, making it seem like a random glitch rather than a fundamental issue.

racies in my temperature measurements.

2.2 MRSD Project

System integration was the major challenge in our last phase, primarily because we were rebuilding the drone from scratch to meet NDAA requirements. We build the drone from the level of essential components, including motors, autopilot Cube, Pixhawk autopilot platform, and a complicated radio communication system. Unlike a pre-assembled system, every module had to be individually set up, configured, and tested, meaning nothing worked stably during the process. The system and components were not inherently stable

or functional. Every test flight introduced new problems, requiring countless iterations of debugging.

3 Team Work

3.1 Sensors and Motor Lab

Name	**Sensor**	**Motor**	**Contribution**
Jet Situ	GUI		GUI implementation with Arduino Integration.
Joshua Pen	Temperature Sensor	DC Stepper	Implemented a user-operated switch with debouncing. Integrated Temperature Sensor with DC Stepper.
Lance Liu	Ultrasonic Range Finder	DC Motor	Integrated Ultrasonic Sensor with Mean Filter and DC Motor. Designed PID controller for DC Motor Postion Control.
Gweneth Ge	Potentiometer	RC Servo	Integrated Potentiometer with RC Servo.
Yi Wu	Force Sensor	DC Motor	Designed PID controller for DC Motor Velocity Control.

Table 1: Team Members and Their Components

3.2 MRSD Project

Name	Contribution
Jet Situ	Assisted in rebuilding the drone for NDAA compliance during the final assembly phase, including mounting components to the provided attachment mounts. Soldered connection wires between the Orin and the Cube Blue, and connected the gimbal's UART control system to the Cube Blue. Validated and connected the gimbal's data connection to onboard networked protocol. Assisted in the development of deployed software on the Orin, and the design of the overall software architecture protocol. Contributed to setting up the ROS2 architecture for inter-system communication in the DTC. Obtained a Part 107 license for purposes of outdoor flight evaluation and testing.
Joshua Pen	Assisted in rebuilding the drone for NDAA compliance, replacing the DJI controller with Cube Blue ArduPilot. Soldered wires to connect various components during the drone rebuild. Designed attachment mounts for the Doodle Labs Radio antennas, Intel Realsense D435, and a separate mount for the NVIDIA Orin NX and Doodle Labs Smart Radio. Developed a mounting solution for the Hadron 640R Gimbal and designed extension legs for the drone. Procured propellers and shields. Contributed to the payload integration and configuration of the Hadron 640R with Cube Blue ArduPilot and NVIDIA Orin NX.
Lance Liu	Contributed to the rebuilding of the NDAA compliant drone. Established communication between the ground control station (GCS) and the drone via a sophisticated radio system. Executed iterative refining and testing on the embedded system. Integrated and validated the interaction pipeline between the GCS and the onboard Orin using MAVROS. Designed a behavior tree to manage decision during the challenge operation. Migrated the AirLab Docker environment to support the ARM architecture on the NVIDIA Orin, and integrated—while continuing to adapt—AirLab's codebase within our platform.
Gweneth Ge	Primarily contributed to the development of the first-version drone for submitting video documentation to DARPA, focusing on sourcing and integration of NDAA-compliant components. Helped the electrical integration, including configuring the Pixhawk ArduPilot and BlackCube on the DJI M100 frame, as well as powering and connecting the gimbal, cameras and radio systems. Additionally, supported overall project management and logistics.
Yi Wu	Collected human detection datasets specifically for drone applications and conducted a literature review of

4 Plans

4.1 MRSD Project

Name	Contribution
Jet Situ	Polygon covering waypoints generator, including sending entire waypoints in one go and flying to designated position at a certain altitude. Gimbal control protocol and sensor nodes development. Take off/landing planner, patients searching logic, task allocation planner, and visualization.
Joshua Pen	In my future role, I will focus on developing gimbal control protocols and sensor nodes, along with additional mechanical modifications. I will assist in sensor nodes development, including detection launch, visualization, and clicking interaction. Furthermore, I will handle project management and logistics.
Lance Liu	ROS2 network refinement. Data transmission of sensors including RGB, Thermal and gimbal. Behavior tree executive and management implementation including auto takeoff, land, safe landing, RTB, mapping, searching, and inspecting. Overall robot system bringing up.
Gweneth Ge	Primarily work on Inter-UAV collision logic and planner launch. Assist in sensor nodes development, detection launch, visualization and clicking interaction. Continue supporting on project management and logistics.
Yi Wu	Integrate the human pose estimation algorithm with the upstream person Re-Identification (ReID) algorithm. Test the algorithm performance on DARPA datasets, and prepare new annotated datasets if necessary for retraining purposes. Develop solutions for pose estimation algorithms using thermal camera data during nighttime conditions.

Table 3: Team Members and Their Contributions