Individual Lab Report #1

Sensors and Motors Lab

February 7, 2025

Gweneth Ge

Team B

Teammates: Joshua Pen, Lance Liu, Yi Wu, Jet Situ

Table of Contents

Contents

1 Individual Progress

1.1 Sensors and Motor Lab

My project consists of two main tasks. The first task involves controlling a Servo HS-485HB using a Potentiometer Sensor PDB181-A420K-503A2. The potentiometer sensor has a range of 300±5 degrees, which is mapped to the servo motor's 180-degree rotation range. When rotating the sensor clockwise, the motor rotates counterclockwise, and vice versa.

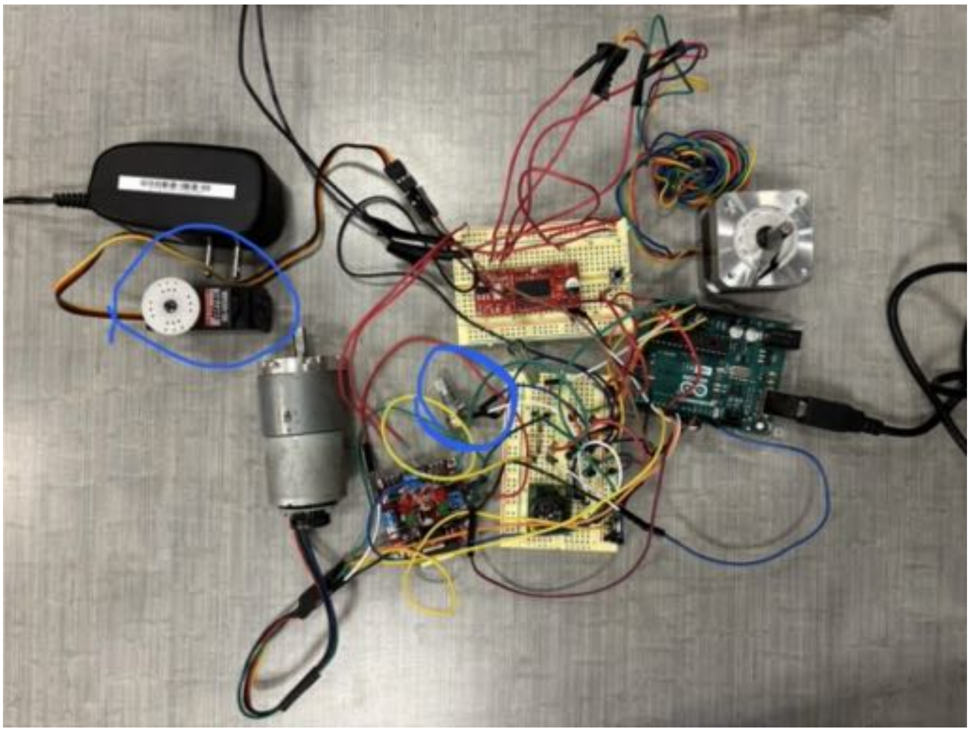

Figure 1: The Potentiometer Sensor and the Servo Motor

The second task focuses on determining the transfer function of the Maxbotix Ultrasonic Rangefinder (LV-EZ1) through physical measurements. I placed the sensor on a desk with no surrounding obstacles and used a measuring tape to measure the height of a plane, which served as the obstacle for the ultrasonic sensor. I conducted three experimental trials, measuring distances between 13 to 31 inches. The relationship between the output voltage (converting Arduino signal range 0-1023 to 0-5V) and the obstacle distance (in centimeters) is plotted in the figure.

The experimental results yielded a fitted model of:

$$V = 0.003722d - 0.018777$$

Based on the sensor datasheets, the distance can be calculated using:

$$Distance = (voltage/(5/512))2.54$$

which can be reformulated as:

$$V = 0.003845d + 0$$

Comparing this with the theoretical transfer function from the datasheet, our analysis revealed slight variations in the measurements. We observed a slope difference of

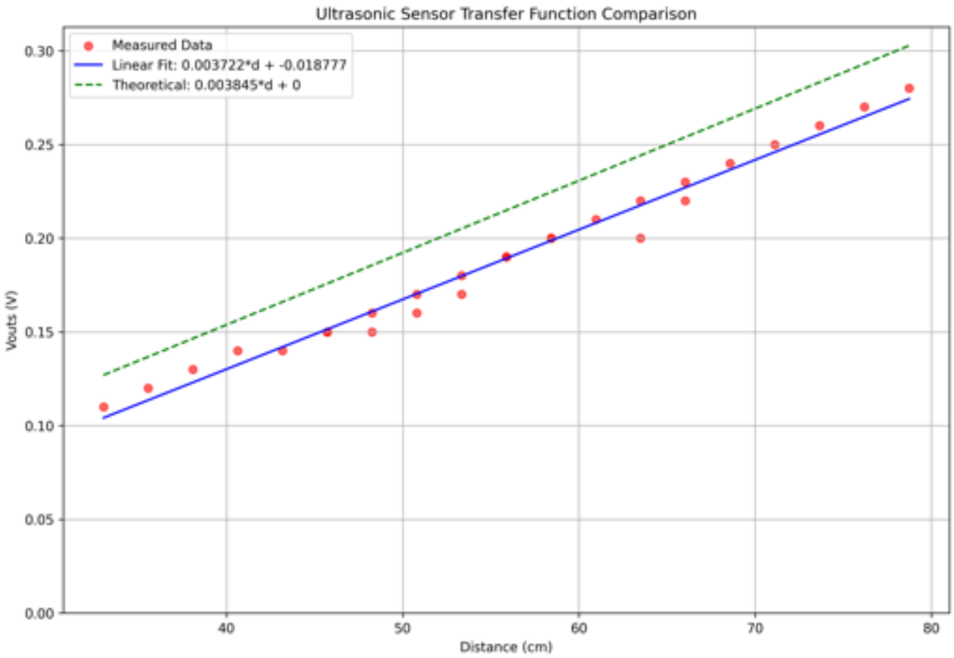

Figure 2: The transfer function of Ultrasonic Sensor

-0.000123, representing a -3.19% deviation from the theoretical value, and an intercept offset of -0.018777V. These discrepancies between the experimental and theoretical models can likely be attributed to measurement errors introduced when reading the ruler measurements by human eyes.

1.2 MRSD Project

I primarily contributed to the development of the first-version drone for DARPA submission. The submission contains videos of our drone's autonomous flight and E-stop with QGroundControl, both during daytime and nighttime. I assisted in the electrical and mechanical integration, including sourcing NDAA-compliant components, configuring Pixhawk2 ArduPilot and BlueCube on the DJI M100 frame, as well as powering and connecting the gimbal, cameras, Nvidia Jetson Orin and Doodle Lab radio system.

Following the initial DARPA demonstration, we rebuilt the drone system to enhance its robustness and stability. I helped with the reintegration of of NVIDIA Jetson Orin and camera gimbal, and setup radio communication. What's more, I supported the overall project management and logistics, such as drone FAA registration, scheduling tasks based on DARPA and the Airlab timeline, sending delicate parts to machine shop, and progress tracking.

2 Challenges

2.1 Sensors and Motor Lab

The primary technical challenge for me during the sensor and motor lab was building transfer functions for the ultrasonic sensor. It was difficult to tell whether my measurements were actually accurate, because the way I did was to hold a power tape with one hand, and another hand holding a plane on top of the sensor and then align to the tape. Although we tested 3 times, my arm sometimes shaken a bit and may have caused errors of measurements.

2.2 MRSD Project

One of our biggest challenges as a team arose during flight testing for the DARPA demo video. The test included autonomous daytime takeoff, landing, waypoint navigation to the target patient, nighttime takeoff and landing, and emergency stop (e-stop) capabilities with QGroundControl.

During testing, we broke two sets of propellers, requiring us to order replacements before resuming. Unfortunately, both replacements also broke, forcing us to order six additional sets to complete the demo. To mitigate this issue, we decided to install propeller guards to reduce the risk of propeller damage during crashes.

We also encountered challenges with the payload integration and configuration of the Hadron 640R with Cube Blue ArduPilot and the NVIDIA Orin NX due to NDAA compliance restrictions. Since we were not allowed to use the HereLink device provided by the company because of its Wi-Fi capabilities, we opted to connect the gimbal via Ethernet to the NVIDIA Orin NX, enabling video streaming from the gimbal.

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

```
  if (state == 1) {
    // Read potentiometer value (0-1023)
    int potValue = analogRead(potPin);

    // Map potentiometer value (0-1023) to servo angle (0-180)
    int servoAngle = map(potValue, POT_MIN, POT_MAX, SERVO_MIN, SERVO_MAX);

    // Control servo to move to specified angle
    myservo.write(servoAngle);

    // Output debug information
    Serial.print("Potentiometer Value: ");
    Serial.print(potValue);
    Serial.print(" | Servo Angle: ");
    Serial.println(servoAngle);

    delay(15);  // Short delay to allow servo to respond
```

```
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;

  // Update moving average
  total = total - readings[readIndex];
  readings[readIndex] = rawDistance;
  total = total + readings[readIndex];
  readIndex = (readIndex + 1) % FILTER_SIZE;

  // Calculate filtered distance
  float filteredDistance = total / FILTER_SIZE;

  // Map the filtered distance to a target position
  // Assuming distance range of 0-100cm maps to 0-1000 encoder ticks
  // Adjust these values based on your needs
  int target = map(filteredDistance, 0, 645, 0, 3500);
```

Sensors and Motor Control Lab Quiz

1.

- What is the sensor's range?
    - Minimum is -3g to 3g, but typically could measure -3.6g to 3.6g.

Parameter	Conditions	Min	Typ	Max	Unit
SENSOR INPUT	Each axis				
Measurement Range		±3	±3.6		*g*

- What is the sensor's dynamic range?
    - Dynamic range is $3.6g \times 2 = 7.2g$ (typically).
- What is the purpose of the capacitor C_{DC} on the LHS of the functional block diagram on p. 1? How does it achieve this?

FUNCTIONAL BLOCK DIAGRAM

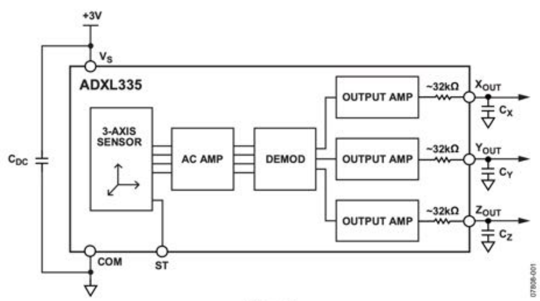

Figure 1.

POWER SUPPLY DECOUPLING

For most applications, a single 0.1 μF capacitor, C_{DC}, placed close to the ADXL335 supply pins adequately decouples the accelerometer from noise on the power supply. However, in applications where noise is present at the 50 kHz internal clock frequency (or any harmonic thereof), additional care in power supply bypassing is required because this noise can cause errors in acceleration measurement.

- The capacitor is used for power supply decoupling, which can suppress high-frequency noise in power supply signals. Such capacitors work as a temporary battery that stores a small amount of energy and releases it to maintain the voltage if the voltage suddenly drops.
- Write an equation for the sensor's transfer function.
    - $V_{out} = 300a + 1.5$, where a is the measured acceleration

SENSITIVITY (RATIOMETRIC)[2]	Each axis				
Sensitivity at X_{OUT}, Y_{OUT}, Z_{OUT}	V_S = 3 V	270	300	330	mV/g
Sensitivity Change Due to Temperature[3]	V_S = 3 V		±0.01		%/°C
ZERO *g* BIAS LEVEL (RATIOMETRIC)					
0 *g* Voltage at X_{OUT}, Y_{OUT}	V_S = 3 V	1.35	1.5	1.65	V
0 *g* Voltage at Z_{OUT}	V_S = 3 V	1.2	1.5	1.8	V
0 *g* Offset vs. Temperature			±1		mg/°C

The slope for V_{out} is the sensor sensitivity (300mV/g), and the Voltage level at 0g represents the offset of the transfer function (1.5 V).

- What is the largest expected nonlinearity error in g?
  - The largest expected nonlinearity error in g is 0.3% of the full scale 7.2g, which is $0.3\% \times 7.2g = 0.0216g$.

Nonlinearity	% of full scale	±0.3	%

- What is the sensor's bandwidth for the X- and Y-axes?
  - The sensor's bandwidth for the X- and Y-axes is 1600 Hz.

FREQUENCY RESPONSE[4]			
Bandwidth X_{OUT}, Y_{OUT}[5]	No external filter	1600	Hz
Bandwidth Z_{OUT}[5]	No external filter	550	Hz

- How much noise do you expect in the X- and Y-axis sensor signals when your measurement bandwidth is 25 Hz?
  - The noise would be

$$150\ \mu g/\sqrt{Hz} \times \sqrt{25\ Hz} \times 10^{-6} = 0.75 \times 10^{-3}\ g$$

- If you didn't have the datasheet, how would you determine the RMS noise experimentally? State any assumptions and list the steps you would take.

2. Signal conditioning
   - Filtering
     - Name at least two problems you might have in using a moving average filter.
       1. Loss of real-time information
       2. Affected by extreme values
       3. latency
     - Name at least two problems you might have in using a median filter.
       1. Heaving computing
       2. Lose important information if window size is too big
       3. Loss of signals which may be quite important

- Opamps
  - In the following questions, you want to calibrate a linear sensor using the circuit in Fig. 1 so that its output range is 0 to 5V. Identify in each case: 1) which of V1 and V2 will be the input voltage and which the reference voltage; 2) the values of the ratio Rf/Ri and the reference voltage. If the calibration can't be done with this circuit, explain why.

$$\frac{V_1 - V_2}{R_i} = \frac{V_2 - V_{out}}{R_f} \quad (1)$$

    - Your uncalibrated sensor has a range of -1.5 to 1.0V (-1.5V should give a 0V output and 1.0V should give a 5V output).

      V_2 is the input, and V_1 is the reference voltage -3V. $Rf/Ri = 1$.
    - Your uncalibrated sensor has a range of -2.5 to 2.5V (-2.5V should give a 0V output and 2.5V should give a 5V output).

      $Rf/Ri = -1$ when V_1 is the input voltage, and $Rf/Ri = 0$ when V_2 is the input voltage. Thus it cannot be calibrated.

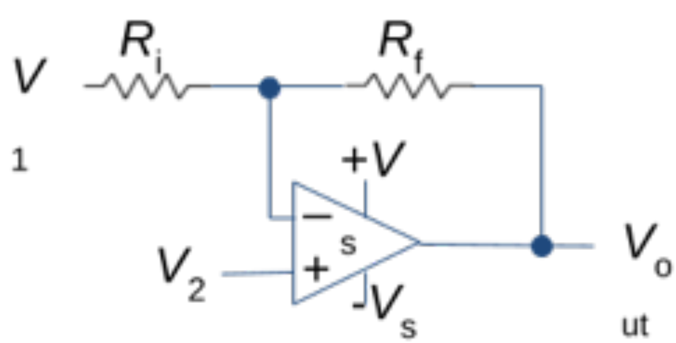

Fig. 1. Opamp gain and offset circuit

3. Control
   - If you want to control a DC motor to go to a desired position, describe how to form a digital input for each of the PID (Proportional, Integral, Derivative) terms.

- Let $e = desired\ position - actual\ position$ and T_s as the testing period. Then the input to K_p is e, and the input to K_D is $(e_k - e_{k-1})/T_s$, and the input to K_I is the integral of e.

$$u = K_p e + K_I \int e\, dt + K_D \frac{de}{dt}$$

- If the system you want to control is sluggish, which PID term(s) will you use and why?
    - Use proportional control K_P if the system is sluggish, because higher K_p will produce a larger control signal for a given error so that it could decrease the rise time and make the motor move more aggressively. K_D could also be used since it reduces overshooting, and thus makes the system response faster.
- After applying the control in the previous question, if the system still has significant steady-state error, which PID term(s) will you use and why?
    - Use integral control K_P because this term accumulates the error over time and thus could adjust the control signal until the steady-state error is close to zero.
- After applying the control in the previous question, if the system still has overshoot, which PID term(s) will you apply and why?
    - Use K_D to reduce the overshoot, because it reacts to the speed of error change. Increasing K_D will add a damping effect to the system, which slows down the system when it is near to the desired position.