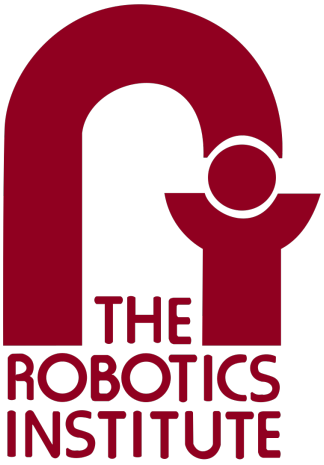
Sensors and Motor Control Lab ILR01



Lunar ROADSTER

Team I

Author: **Deepam Ameria**
Andrew ID: dameria
E-mail: *dameria@andrew.cmu.edu*

Teammate: **Ankit Aggarwal**
ID: ankitagg
E-mail: *ankitagg@andrew.cmu.edu*

Teammate: **Bhaswanth Ayapilla**
ID: bayapill
E-mail: *bayapill@andrew.cmu.edu*

Teammate: **Simson D'Souza**
ID: sjdsouza
E-mail: *sjdsouza@andrew.cmu.edu*

Teammate: **Boxiang (William) Fu**
ID: boxiangf
E-mail: *boxiangf@andrew.cmu.edu*

Supervisor: **Dr. William "Red" Whittaker**
Department: Field Robotics Center
E-mail: *red@cmu.edu*

February 17, 2025

Carnegie Mellon University
Robotics Institute

Contents

1 Individual Progress

1.1 Sensors and Motor Control Lab

My tasks for the Sensors and Motor Controls lab were to interface a sensor and an RC Servo Motor. I decided to proceed with an Inertial Measurement Unit (IMU) sensor and the MG996R Servo Motor. In addition, I assisted Simson in soldering and testing the DC Motor Controller, interfacing the temperature sensor, and assisted William with the hardware integration for this lab task. Outlined below are the steps I took to complete my tasks

1.1.1 Inertial Measurement Unit Sensor (IMU)

From the wide array of sensors that one can choose from for this task, I chose the IMU. I specifically interfaced an MPU6050 IMU, which is a 6-axis IMU (3-axis accelerometer, 3-axis gyroscope). Interfacing involved connecting the IMU to the SCL and SDA pins of the Arduino Due Board to establish I2C communication. I used the popular Adafruit MPU6050 library to get sensor readings from the IMU. Primarily, the acceleration values were read and processed to determine the pitch angle. The following formula was used to convert the acceleration values to physical angles:

$$\text{Pitch Angle} = \tan^{-1} \frac{Acceleration_x}{Acceleration_z} \times \frac{180}{\pi}$$

1.1.2 RC Servo Motor

My second contribution to this task was interfacing the Servo Motor with the microcontroller. The signal pin was connected to a PWM pin on the board. I utilized the in-built Servo library in Arduino IDE to attach the servo to the board and send (or receive) information to (or from) the motor for its position. The interfacing with the GUI was done such that if the user specifies any angle between 0 to 180 degrees (using a slider on the GUI), the servo will rotate correspondingly to that angle from its home position.
Additionally, I also worked with Bhaswanth to make sure that when the push-button in the circuit is pressed, the servo rotates 30 degrees incrementally.

The code for both these tasks was integrated with the complete code base in their specific callback functions, to be accessed by the GUI. (See Appendix for code)

1.1.3 Other Contributions

Apart from the two main tasks, I also assisted my teammates in this lab task. I worked with Simson to test and interface the L298 Motor Driver with the DC Motor, and later with the Roboclaw Motor Driver as well. I also worked with Simson and Bhaswanth in interfacing the TMP36 temperature sensor. Finally, William and I collaborated to integrate the sensor callbacks and the hardware. I assisted Ankit and William in setting up the circuit, depicted in Figure 1. I was actively involved in the unit and system-level testing of the entire setup.

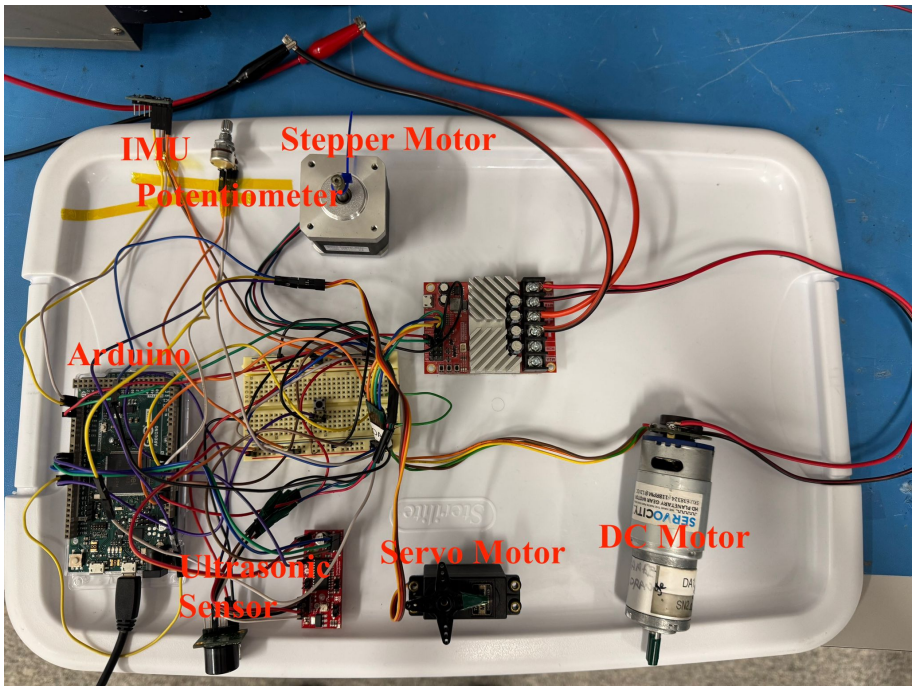


Figure 1: Sensors and Motors Lab Setup

1.2 MRSD Project

As part of the the MRSD Project, for the last couple of months (December - January), I have been working on the following things

- **Preliminary Teleoperation Tests**: We are using CraterGrader's Rover and building up our hardware and software over it. During the winter break, William and Bhaswanth got the rover running in teleoperation mode. Once the rover was up and running, I carried out the first preliminary tests of the rover in the Moonyard at Planetary Robotics Lab (PRL). I ran the entire teleop stack and made the rover make its "first steps" in the yard for our project. Simson and I made key observations on the traction and the motion of the rover, which helped us understand what hardware and mechanical updates would be required for the robot.

- **Dozer prototype tests**: I'm currently assigned to develop the entire dozer blade assembly for the robot. After multiple meetings and insights from our sponsor and from other team members, I put together the first prototype of the dozer blade using plywood. Simson and I made adjustments to the rover to incorporate this prototype and mount it temporarily in the front. We carried out the grading and traversability tests of the rover with the dozer in front, at PRL. We made key observations on how well or how badly the rover is pushing sand in different conditions, how the height and shape of the blade can make a difference, and how it is affecting the slippage and sinkage of the wheels, along with the pitching motion of the rover. Similar tests with a metal blade were also performed in January with the entire team. (See Figure 2 and Figure 3)

Figure 2: Dozer blade prototype

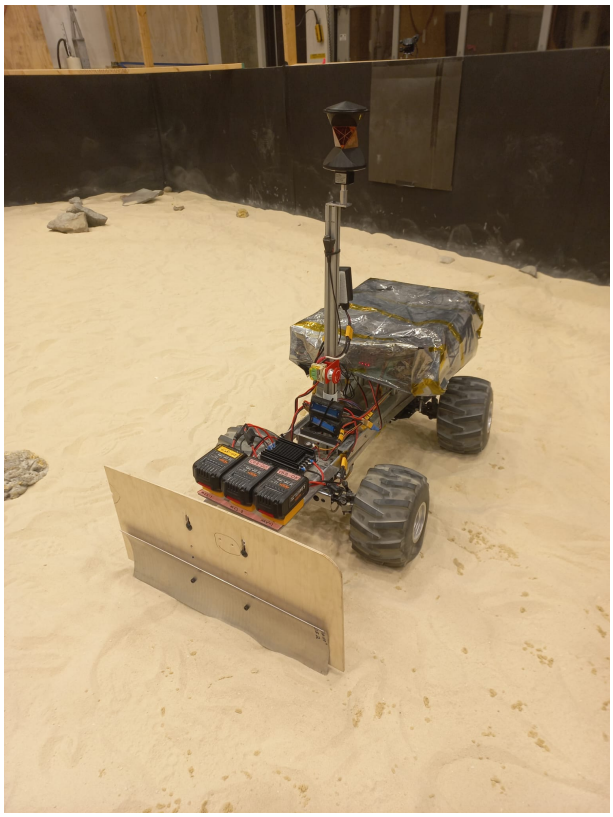


Figure 3: Dozer blade prototype 2

- **Dozer Blade and Mechanism Design**: Following the tests and initial discussions with the team and our sponsor, since the winter break, I have been working on developing and finalizing the design for our blade and the mechanism to mount it to the chassis and achieve 1DOF of motion. Multiple blade and mounting iterations were discussed with everyone, which I designed using SolidWorks (see Figure 4). I have been collaborating with Ankit and Simson to refine the design and develop a robust mechanism that can dynamically lift the dozer blade or bring it back down by 1-1.5 inches. This also involves selecting the right actuator for this task and making other modifications in the chassis and placement of other components on the rover.

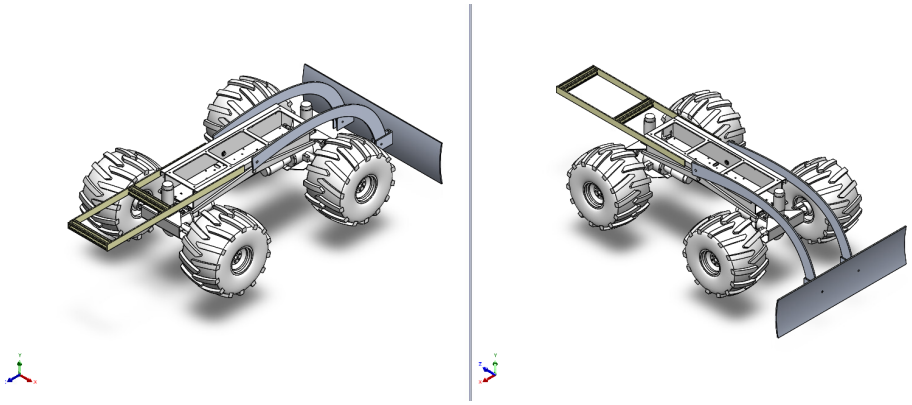


Figure 4: Dozer Design Iteration

- **FARO Scanner Setup and Moonyard Scanning**: I worked with the team in setting up the FARO Laser Scanner. We took initial scans of the Moonyard for fetching point cloud data and converting it into robot-readable format.

2 Challenges

2.1 Sensors and Motor Control Lab

While interfacing with the IMU and the Servo Motor, I only faced an issue with the operation of the servo motor. The HiTec Servo provided was jittering constantly when it reached the specified angle. I tried to debug it using different power supplies, and making sure all wires were connected properly, but continued to face the problem. Unable to determine the exact culprit of this (I suspect it was some interference from other components or common-ground issues), I decided to use a different servo motor. Fortunately, the new one worked well and I was able to mitigate this problem.

2.2 MRSD Project

One major bottleneck I was facing in the project was researching and developing a dozer blade. Dozer blades are a fairly new avenue for me, as a mechanical engineer, to explore, so reading literature, going through numerous resources, and understanding the reasons behind the shape and size of the blade used up more time than allotted. To our benefit, our sponsor has commendable experience in developing such systems and parts, and through regular meetings, I'm able to proceed with the design and development of the blade and the mechanism.
Another challenge I faced was setting up the VectorNav IMU. Initially, I was assigned the task of doing so, but the firmware on the IMU was outdated, and there was no simple way to update it. Moreover, the ROS packages available were also not compatible with the old firmware. In order to stick to the timeline as much as possible, the team decided to off-load the task and Ankit took up this task. Luckily, he got in touch with customer support and was able to successfully set up the IMU.

3 Teamwork

3.1 Sensors and Motor Control Lab

Apart from my contribution to this task, the team divided other tasks as follows:

- **Ankit**: He interfaced the potentiometer and the stepper motor with the controller. He also made the stepper motor controllable using the potentiometer.
- **William**: He developed the GUI for this lab task, and was responsible for the overall integration of all the components and the final demonstration during the lab.
- **Bhaswanth**: He interfaced the ultrasonic sensor, wrote the transfer function for the same, applied a filter to smoothen out the sensor readings, and implemented debouncing for the push-button.
- **Simson**: He interfaced the DC Motor (with encoder) and the Temperature sensor.

3.2 MRSD Project

Throughout the Winter break and in the first month of the semester, the team divided the tasks as follows:

- **Ankit**: He worked on the wheel design for the rover and 3D printing of the prototype of the same. He collaborated with Simson to devise a circuit diagram of the rover. He worked on getting the firmware update for the VectorNav IMU. He is also the Project Management for this semester.
- **William**: He developed a crater distribution scaled for the MoonYard. He collaborated with Bhaswanth to set up the teleoperation of the rover and for the ZED Camera setup. He has also set up a LAN for our project.
- **Bhaswanth**: He was responsible for setting up Jetson and the operations terminal. He set up the encoder drivers and worked with William to set up teleoperation and the ZED Camera. He also set up the Docker and GitHub for our team.
- **Simson**: He has worked with me in preliminary testing and dozer design and prototyping. He is working on creating a robot-readable map using the scans of the Moonyard from the FARO Laser Scanner, by processing the point cloud data to generate an occupancy grid map.

4 Plans

4.1 Sensors and Motor Control Lab

I will not be using the servo motor in the Capstone project. Instead, I will use a linear actuator for the lifting mechanism of the dozer arms and blade.

4.2 MRSD Project

We plan to do the following by the next Progress Review:

- I will complete the dozer blade and mechanism design and start working on fabricating and carrying out tests on the mechanism.
- The team will also optimize the cables on the rover by efficient routing of the wires and placement of the electronic components.
- We also aim to test the 3D-printed prototype of the wheel.
- We will be scanning the Moonyard using the ZED Camera to aid in the development of the map using the point cloud data it provides.
- The team will also begin work on the localization stack.

5 Sensors and Motor Control Quiz

5.1 Question 1: Reading a Datasheet

1. What is the sensor's range?
   Ans: The measurement range along each axis is ±3g.

2. What is the sensor's dynamic range?
   Ans: The maximum dynamic range is 7.2g and the minimum dynamic range is 6g.

3. What is the purpose of the capacitor CDC on the LHS of the functional block diagram on p.1? How does it achieve this?
   Ans: The capacitor CDC functions as a decoupling capacitor. It removes high-frequency interference from the power supply while keeping the sensor's voltage steady. This capacitor accomplishes its task by intercepting any quick changes in voltage, whether up or down, allowing the sensor to work reliably.

4. Write an equation for the sensor's transfer function.
   Ans: $V_{out} = 1.5V + 0.3V/ga$

5. What is the largest expected nonlinearity error in g?
   The largest expected nonlinearity error in g is ±0.0216 g.

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

7. How much noise do you expect in the X- and Y-axis sensor signals when your measurement bandwidth is 25 Hz?
   Ans:
   $$rmsNoise = NoiseDensity\sqrt{BW1.6}$$
   $$rmsNoise = 150\sqrt{1.625} = 948.683\mu g$$

8. 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.
   Ans: To determine the RMS noise experimentally, I would place the ADXL335 on a stable, vibration-free surface and power it with a stable voltage. Then, I would sample data from the X, Y, and Z axes at a high rate ($\geq$1 kHz) and collect a sufficiently high amount of data to ensure statistical analysis.
   The main assumption in the method is that the surface is stable and vibration-free and that the location (and the exact gravity at that location) is known.

5.2 Signal Conditioning

5.2.1 Filtering

1. Name at least two problems you might have in using a moving average filter.
   Ans: Outlier Sensitivity: Moving average filters can be affected by outliers, as extreme values in the data may disproportionately impact the smoothed result.
   Lag: These filters introduce a phase delay, causing a lag in the real-time response.

2. Name at least two problems you might have in using a median filter.
   Ans: Loss of detail: Median filters can smooth out sharp features or edges in the data, reducing clarity.
   Computation: Median filters require the sorting of data. Running algorithms to do so is computationally more complex than other simpler statistical methods.

5.2.2 Op-Amps

- 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).
  Ans: V_1 - Reference Voltage, V_2- Input Voltage. We have,

  $$V_{out} = V_{in}(1 + \frac{R_f}{R_i}) - V_{ref}\frac{R_f}{R_i}$$

  When we solve this equation, we get

  $$\frac{R_f}{R_i} = 1, V_{ref} = -3V$$

- V_1 - Reference Voltage, V_2- Input Voltage.

  Solving the same equation results in undefined values for V_{ref} and $\frac{R_f}{R_i}$, making the system infeasible and leaving no valid solutions.

  V_1 represents the reference voltage, and V_2 denotes the input voltage. The system follows the equation:

  $$V_{out} = V_{in}\left(1 + \frac{R_f}{R_i}\right) - V_{ref}\frac{R_f}{R_i} \tag{1}$$

  Solving this equation results in undefined values for V_{ref} and $\frac{R_f}{R_i}$, making the system infeasible and leaving no valid solutions.

  An analysis of the system equations shows no solution exists, regardless of whether V_1 is the reference and V_2 is the input or vice versa. Since both input and output have a $5V$ range, an offset seems sufficient, but this would require zero gain, negating the reference voltage and making calibration impossible.

  Consequently, the circuit does not yield a feasible solution.

5.3 Control

1. To implement PID control digitally, the Proportional term takes the difference between the desired and actual position as input. The Integral term accumulates this error over time to address any persistent discrepancies. The Derivative term calculates the rate of change of this error by comparing the difference between the current and previous errors, helping to predict and counteract rapid changes.

2. I would increase the Proportional Gain (Kp) if the system is sluggish, as a higher P gain will make the response larger. It will lead to a quicker response.

3. I would increase the Integral Gain (Ki) if the system has some steady-state error. This is because it will accumulate errors and amplify the control effort until the error is minimized.

4. If the system still has some overshoot, I will increase the Derivative Gain (Kd). This is the damping term and it will slow down the response and prevent large overshoots.

6 Appendix

Code for IMU and Servo Motor interfaced with GUI:

```
    void servoMotorController(int control) {
  /*
  INPUT: Integer in min/max range of 0 to 180 corresponding to desired angle
  OUTPUT: Void
  */

  SERIAL_PORT.print("Servo motor controller received command: ");
  SERIAL_PORT.print(control);
  SERIAL_PORT.print(";");

  // TODO: IMPLEMENT FUNCTION BELOW
  servoAngle = control;
  servo.write(servoAngle);
}

int servoMotorStateCallback() {
  /*
  INPUT: Void
  OUTPUT: Integer in min/max range of 0 to 180
  corresponding to servo motor angle
  */

  // TODO: IMPLEMENT FUNCTION BELOW
  return servoAngle;
}

double imuSensorCallback() {
  /*
  INPUT: Void
  OUTPUT: Double corresponding to sensed IMU pitch reading
  */

  // TODO: IMPLEMENT FUNCTION BELOW
  sensors_event_t a, g, temp;
  mpu.getEvent(&a, &g, &temp);

  // Pitch using accel data
  double pitchAccel = atan2(a.acceleration.x, a.acceleration.z) * 180 / PI;

  return pitchAccel;
}
```