Individual Lab Report 7

Lunar ROADSTER

Team I

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

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

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

September 25, 2025

Carnegie Mellon University
Robotics Institute

1 Individual Progress

Since the last progress review, I worked on two main tasks. The first task is debugging the software connections on the new Orin compute with the Arduino and joystick. My second task is on researching and starting to implement the validation unit. This will be used in our workflow to determine whether an adequate job is performed on each crater. Inadequate craters will be re-graded so that the crater's gradient is below our specified requirement of 5 degrees.

1.1 Debugging Software Connections

1.1.1 Joystick

The new NVIDIA Orin compute unit uses JetPack 6.0 SDKs for its development environment. However, it inadvertently broke many Logitech drivers (see here). We are currently using a Logitech F710 joystick to remotely teleop our rover (see Figure 1). The broken drivers meant that the Orin was unable to communicate with the joystick. Therefore, it renders us unable to teleop the rover. It took me some time to figure out what drivers were missing and how to fix this issue. I eventually found this YouTube video which goes over how to resolve this problem and followed it to resolve the joystick connection issue. The joystick is now able to control the rover in the same way as the old NVIDIA Xavier was.

Figure 1: F710 joystick teleoperating our rover

1.1.2 Arduino

A similar problem arose with the Arduino to Orin connection. The new SDKs and upgraded dependencies meant that we were not able to establish connection between the Orin and the Arduino. We originally thought that the Arduino board was the issue (due to 4 months of not using it) and that the board might be fried. We spent considerable time unit testing Arduino code/hardware and physical connections without any luck. After exhausting the failure points on the Arduino, we switched to debugging on the Orin. We tried to downgrade the firmware for the serial port gateway and changing

the baud rate – all to no avail. Finally, Bhaswanth uninstalled and reinstalled Micro-ROS and the connection was suddenly established. We tried several times afterwards to make sure the connection is maintained before starting other nodes for testing.

As soon as the teleop node was turned on and a steering input from the joystick was given, the wheels jolted to one direction and severely damaged the steering pinions. Bhaswanth and I then performed unit and wire testing to see what could be the reason for the sudden jolt from the steer input (see Figure 2). After spending some time, we found two problems. The first problem is that the power connections for the steer RoboClaw and drive RoboClaw were cross-wired. This meant that power signals that were meant for drive were being sent to steer, causing the sudden jolts. A second problem is that one wire that was soldered to the front steer motor came loose. This meant it was unable to transmit information back to the RoboClaw controllers and caused the PID to become open-loop. Fixing both problems resolved the steering issue and teleop works as intended.

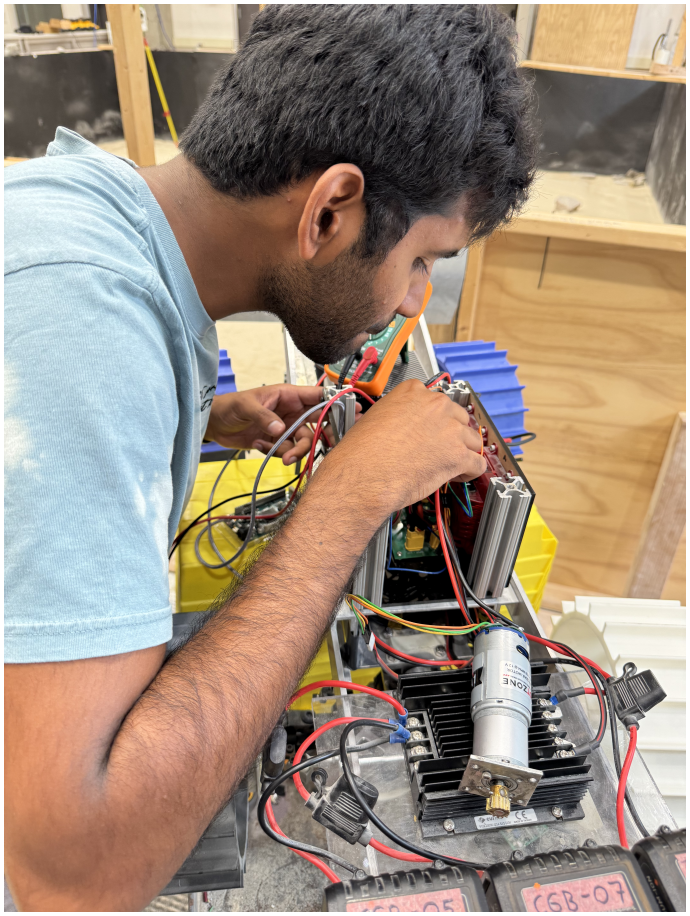

Figure 2: Debugging wiring of the rover

1.2 Validation Unit

After the debugging tasks, I started researching and began implementing the skeleton code of the validation unit. This unit is used to validate the smoothness of a graded crater. If the gradient is less than 5 degrees, then the crater is considered graded. If it does not meet this benchmark, then the crater would be re-graded. I settled on an original methodology of using the point cloud from the ZED camera and elevation mapping to determine gradients. Elevation mapping is a package I wrote for my SLAM course project at CMU and can be found here. It gives the height information of discrete grid

cells of the Moon Yard (see Figure 3). The entire validation stack would preliminarily have the following steps:

1. Obtain point cloud from ZED camera FOV
2. Use RANSAC to fit a plane
3. Discretize map into grids and bin points in each cell
4. Compute an elevation map relative to the RANSAC plane
5. Compute gradients using finite differences
6. Remove excessive gradients (walls)
7. Compute maximum gradient

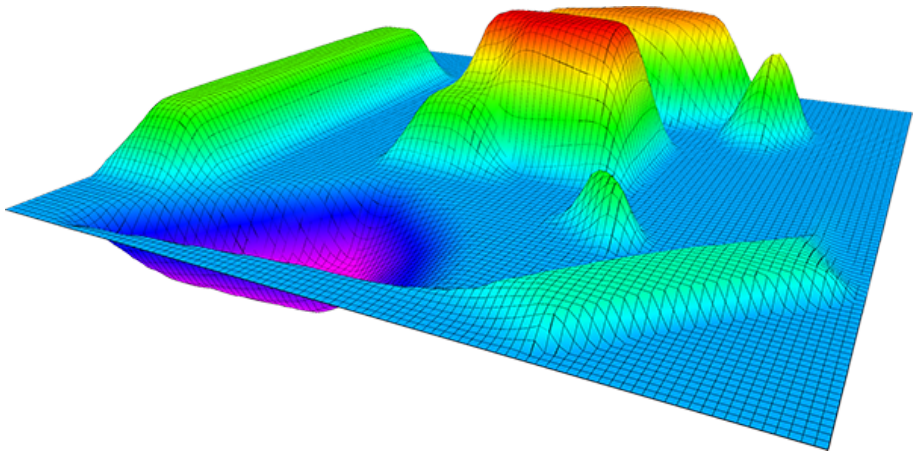

Figure 3: Elevation map of the environment

2 Challenges

A significant challenge faced is debugging the various issues facing the rover. Issues such as missing driver dependencies and gateway connection issues are small problems by itself, but finding solutions for them can take considerable time and effort. The time spent debugging meant that I was not able to spend as much time on implementing new elements. This meant that I would have to make up the lost productivity later in the semester.

Another challenge faced was that during testing for the steer input, the sudden jolts significantly wore-down the steering pinions. The original pinions with the exact number of tooth were discontinued and is no longer replaceable. The team has decided to switch to a brand new pinion with a differing tooth count so that replacement parts are no longer a problem for us. However, this would require recalibrating some of the steering commands (such as QPPS, RPM values, etc).

3 Teamwork

A breakdown of the contributions of each team member are tabulated below:

- **Ankit Aggarwal**: Ankit's work involved finalizing the methodology for the perception stack. He worked with Deepam and finalized a Deep Learning method to detect craters and extract geometry information. Ankit collected a preliminary dataset for the YOLO Model and a video to run inferences. He also worked with Deepam to refine the dozer performance by debugging the jittery motion. Additionally, Ankit manufactured mounts for the ZED and Orin.
- **Deepam Ameria**: Deepam's work involved finalizing the methodology for the perception stack of our system. He worked with Ankit and decided to go ahead with a Deep Learning based method to detect craters and extract geometry information. For this task, Deepam worked with Ankit to carry out initial research and also collect preliminary data (images of craters) for training a YOLO model and run inference on a test video. He also worked with Ankit on refining the Dozer and debugged the problem of jittery motion of the actuator at intermediate positions. Deepam also worked with Bhaswanth and Ankit for solving the Arduino Reset issue.
- **Bhaswanth Ayapilla**: Bhaswanth's initial work involved working on the resection method for the total station. He worked with the rest of the team in performing hardware maintenance of the rover, and closely collaborated with William in performing unit testing to fix the connection issues. Bhaswanth also worked with Simson in finalizing the global controller methodology. Additionally, he is working with Ankit and Deepam to debug the Arduino reset issue and implement the software fix.
- **Simson D'Souza**: Simson's work focused on map generation and refining the crater detection logic. He explored different plane-fitting techniques and simpler approaches for identifying gradable craters. However, the original method proved to be more reliable and accurate, so he reverted to it. In addition, Simson collaborated with Bhaswanth on the navigation planner and contributed to formulating the global navigation controller methodology, which he will soon begin implementing.

4 Plans

From now until Progress Review 9, I will be mainly working on implementing the validation stack and integrating it into our software stack. I will also need to tune the RANSAC parameter values and rigid transforms from the ZED camera frame to the `base_link` transform so that the elevation map output is relative to the rover's frame. If time permits, I plan to also start on researching on how to localize the rover using the SkyCam camera.