Individual Lab Report - 02

Lunar ROADSTER

Team I

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

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

Teammate: **Boxiang (William) Fu**
ID: boxiangf
E-mail: *boxiangf@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*

February 14, 2025

Carnegie Mellon University
Robotics Institute

Contents

1 Individual Progress

Since the previous progress review, I have put in most of my time on the localization setup of the rover. This includes setting up the Leica TS16 robotic total station and working on the software. Additionally, I worked on the limit switches for the steering and the ZED camera as well. I will discuss my contributions in more detail below:

1.1 Localization

The TS16 Leica-Geosystems Robotic Total Station will be used to track a prism mounted on the rover and obtain its precise position within the Moon Yard. Back in December during our winter break, the entire team received training from Warren "Chuck" Whittaker on setting up the total station. Using this knowledge and the existing documentation from Crater Grader, me and my teammate William were able to set up the total station in the Moon Yard, fix a frame of reference by calibrating it using a mini-prism, and then continuously measuring the position of a larger prism with respect to the new fixed frame. By mounting this prism on the mast of the rover, we were able to track its position precisely and also publish them to a ROS topic.

Figure 1: TS16 Leica-Geosystems Robotic Total Station in the Moon Yard

The following are the steps to set up and calibrate the total station:

1. Prepare Leica mini-prism by attaching S-hook into the eyebolt.
2. Change the target settings to "Leica Mini 360".
3. Go to home -> setup, and choose the "Orientate to Line" method.
4. Place the mini-prism in location A (see Figure 2) via S-hook attachment.
5. Press "Measure" to measure the first location.

6. Move the mini-prism to location B (see Figure 2).

7. Press "Measure" to measure the second location.

8. Press "Set" to finish calibration.

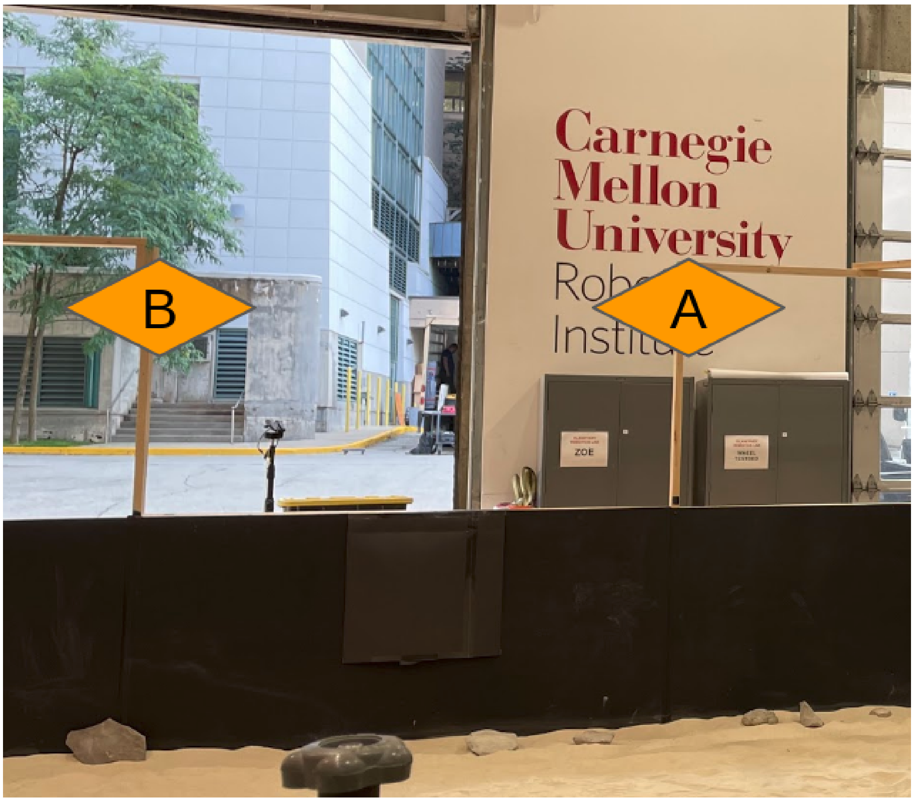

Figure 2: Total Station Calibration Locations in PRL

The following are the steps to measure and track the Leica Prism 360 mounted on the mast of the rover:

1. Plug in laptop or TX2 into the total station via serial connection.

2. Launch ROS node using `ros2 launch total_station total_station_launch.py`.

3. Find the Leica Prism 360 using the "Power Search Feature".

4. Turn on measure mode, using "Measure" button.

5. The prism location should be publishing on the `/total_station_prism` topic.

1.2 Limit Switches

The rover we use has a minor steering issue due to the ball-and-pinion mechanism used for steering. This mechanism is secured to the steering motor's shaft using a grub screw. However, when the wheels are turned beyond a certain limit, the grub screw tends to come loose, causing the entire steering system to fail. To prevent this, we considered using limit switches to restrict the motors from steering past their breaking point. The rover already has built-in limit switches, and I attempted to interface them with an Arduino, but I was unable to read any signals from them. Currently, the teleoperation code implements a software limit, but as we continue testing, we will gain a better understanding of whether an additional set of limit switches is necessary. I collaborated with Ankit on this task.

1.3 ZED Camera

We previously set up the ZED 2i depth camera to work with the Jetson AGX Xavier. Now, we need to integrate it with ROS2 for navigation and, later, for validation. I am currently working on this along with my teammate, Simson.

2 Challenges

One of the issues that I am currently facing with the localization stack is visualizing data on Rviz. I'm using a VectorNav IMU ROS2 wrapper to read IMU data, but no transformation data is being published to either the `/tf` or `/tf_static` topics. This makes it difficult to directly visualize the IMU's output in Rviz. I'm currently reviewing the code to resolve this issue.

Another challenge I'm facing is interfacing the ZED 2i depth camera with ROS2 on the Jetson. The Jetson runs Ubuntu 20.04, while the Docker container running ROS2 Humble is based on Ubuntu 22.04. Installing the ZED SDK inside the container has led to CUDA incompatibility and library version conflicts. For now, I have successfully installed the correct CUDA and SDK versions, but the mismatched library versions are preventing the ROS2 wrapper from building correctly. I'm currently working on fixing these dependencies.

3 Team Work

- **Bhaswanth Ayapilla:** My work was in collaboration with William's in setting up the total station and working on the localization stack. I have been working on visualizing the rover's pose on Rviz. I tried to get the already existing limit switches working, and this relates to Ankit's work on fixing the steering mechanism. My work also relates to Simson in interfacing the ZED camera with ROS, which will be used in navigation.

- **Ankit Aggarwal:** Ankit worked on fixing the rover drive system to allow the team to resume testing, which involved creating a secure assembly and collaborating with me to introduce limit switches for robustness. He collaborated with Deepam to discuss design iterations of the dozer blade. He interfaced the IMU with ROS2, which enabled me and William to take forth our work on localization. He also took ownership of the PCB assignment which facilitated the team to prioritize tasks on the project.

- **Deepam Ameria:** Deepam worked on finalizing the design of the dozer assembly, which includes the dozer blade, dozer arms, mounting brackets, lifting mechanism, and linear actuator. He also collaborated with Ankit to discuss iterations and ideas on the lifting mechanism, and with Simson to manufacture the dozer blade. The team also worked together on devising a test plan. His work does not relate to mine directly at this stage.

- **Simson D'Souza:** Simson worked on processing the point cloud data from the FARO scanner. His work relates to mine in setting up the ZED 2i depth camera, which will be used for navigation and validation. He also worked on setting up the Intel RealSense D435i camera and tested mapping using RealSense to generate a 2D costmap. Additionally, he designed the electrical circuitry diagram and collaborated with Deepam on fabricating the dozer blade.

- **Boxiang (William) Fu:** William's work was in collaboration with mine in setting up the external infrastructure for localization. This includes setting up the total station, the TX2 relay that's connected to the total station, and the LAN network for the total station data to be transmitted to the rover. Both of our work also relates to Simson's since the map he is producing is used in localization as the world frame.

4 Plans

Before ILR3, I will be working on the localization stack of the rover along with William. We have split the task into global localization (using the total station) and local localization (IMU + wheel encoders). We will work together collaboratively and fine tune the EKF as we conduct more tests. I will also complete the integration of the ZED 2i camera with ROS2 so that the navigation task can begin quickly too. Once this is done, I will be working on translating the teleoperation stack from Crater Grader to our software stack. This will serve as the foundation for the navigation stack and give us a clearer understanding of how to design the FSM planner and tool planner effectively.

The team plans on completing the entire dozer assembly and conducting tests to evaluate its functionality. Additionally, we will test the wheel design and make iterative improvements as needed. Our goal is to finalize the rover's hardware as soon as possible, so that everyone can shift focus to the computational tasks such as navigation and planning.