Individual Lab Report 2

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*

February 14, 2025

Carnegie Mellon University
Robotics Institute

1 Individual Progress

Since the last progress review, I worked on the external infrastructure for the project. This includes setting up the total station, the NVIDIA TX2 relay, and the software to publish the pose updates provided by the total station via our LAN network. My contributions are discussed in turn:

1.1 Total Station Setup

The total station is used to provide accurate measurements of the rover's location in the Moon Yard. It is one out of three sensors that will be used for the localization stack of the rover, with the other two being an IMU and wheel odometry. Using documentation given to us by Crater Grader and help from my teammate Bhaswanth, we were able to set up the total station and calibrate its frame of reference so it stays fixed relative to the Moon Yard. We are then able to switch to measuring mode and track a moving Leica Prism relative to the calibrated fixed reference frame. This enables us to find the position of the prism inside the Moon Yard. This allows us to localize the rover to a high degree of accuracy. Finally, the total station is connected to our laptop (or the TX2 relay) via serial port to process the arguments and publish them to a ROS topic. Figure 1 shows the total station placed at the setup location in the Moon Yard, with the cable connecting to our development laptop.

Figure 1: Total Station Setup

In summary (of the setup procedure provided by Crater Grader), the calibration of the total station are as follows:

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.

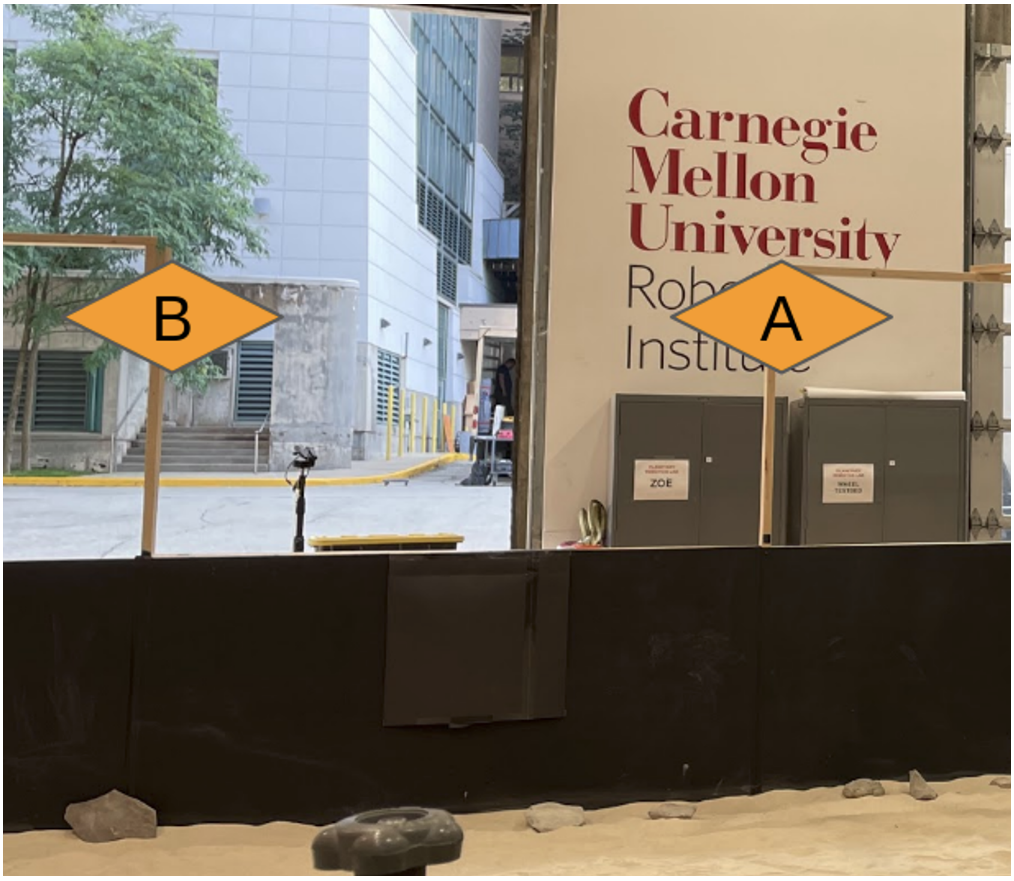

Figure 2: Total Station Calibration Locations

The measuring and tracking of the rover (more specifically the Leica Prism 360 mounted on top of the rover) are as follows:

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 (see Figure 3).

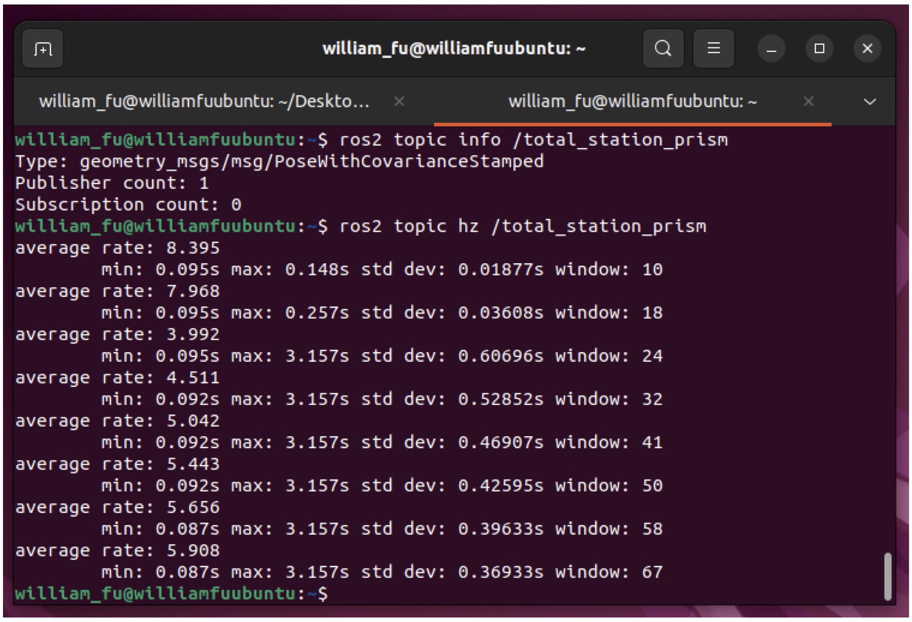

Figure 3: Prism Location being Published on ROS Topic

1.2 NVIDIA TX2 Setup

The TX2 chip is used to connect to the total station via serial port and convert the tracked Leica Prism 360 location to a ROS message format. The ROS message is then published to the `/total_station_prism` topic on our LAN network, which will be subscribed to by the Jetson Xavier on the rover to use in its localization stack.

For the serial port connection between the total station and the TX2, the data is transmitted in GeoCOM format. The ROS node on the TX2 chip reads in this GeoCOM formatted data and converts it into a standard ROS readable `PoseWithCovarianceStamped` format. A ROS topic called `/total_station_prism` is then initialized that publishes the formatted data. I did not write the code for the TX2 as the chip is directly inherited from Crater Grader. The majority of my time spent was on understanding Crater Grader's method for publishing pose information and reproducing the results. I then taught what I learned to my team members so that we do not have to reinvent the wheel and instead focus our efforts on other important tasks.

1.3 Communication Setup

I already discussed setting up our LAN network in ILR1 and will not go over it here. However, at the time of ILR1, I was not able to establish communication between the TX2 and the Jetson Xavier on our rover. It turned out that the domain ID for the two devices were set differently and the firewall on the Xavier was blocking the communication requests. I was able to establish a common domain ID using `export ROS_DOMAIN_ID={ID}` and disabled the firewall on the Xavier. This established the connection between the TX2 and the Xavier. In fact, Figure 3 shows the terminal on my laptop receiving messages on the `/total_station_prism` topic, which was actually being published by the TX2 and not connected physically to my laptop. Figure 4 shows the hardware for the communication setup (TX2 chip and LAN router).

Figure 4: Jetson TX2 Chip and LAN Router

2 Challenges

The first challenge was gaining SSH access to the TX2. The TX2 relay was originally set up by Crater Grader. We obtained their documentation along with their login details. However, we did not that that the LunarX team used the TX2 relay after Crater Grader. They changed the login details so that we were not able to login using Crater Grader's details. We spend a considerable amount of time trying different login combinations before a team member from Crater Grader remembered that LunarX used the equipment. We were finally able to ask LunarX and obtained the login details.

A second challenge faced was that at first the TX2 chip was not able to read the data sent by the total station. It turns out that the data format was originally configured incorrectly. It was sending in the Leica Image File (LIF) instead of the required GeoCOM format. The documentation did not specify this detail, and we spent a considerable amount of time trying to debug this issue. We only found out the issue by going through the source code of the ROS2 node and figuring out the required input data format.

3 Teamwork

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

- **Ankit Aggarwal**: Ankit worked on the rover drive system and maintenance of the steering mechanism. This was in collaboration with Deepam on the hardware side of the rover. His work will be integrated with my work as the motor encoders used in the steering mechanism will be integrated in the localization stack that I am working on. Ankit also worked on setting up the IMU, which relates to my work as I will be using the output from the IMU for localization. Finally, Ankit is the project manager for the team, and tracks the overall progress of our project.
- **Deepam Ameria**: Deepam worked on the finalizing the design of the dozer assembly (dozer blade, dozer arms, mounting brackets, lifting mechanism, linear

actuator). He also collaborated with Ankit to discuss iterations and ideas on the lifting mechanism, and with Simson to manufacture the dozer blade. Currently, his work does not have much to do with my work on localization. However, my next task after completing localization will be on the dozer tool planning. His dozer hardware design will be integrated then as the tool planning software stack will be used to control the hardware.

- **Bhaswanth Ayapilla**: Bhaswanth worked with me in setting up the total station and working on the localization stack. This is directly related to my work as we were working together on setting up the external infrastructure of the project. Afterwards, Bhaswanth worked on trying to visualize the location of the rover on RViz. This is related to my work as I will be using the visualization to see the localization in the world frame and tune the error parameters of the EKF. Finally, Bhaswanth tried to get the already existing limit switches working. This relates to Ankit's work on fixing the steering mechanism. His work also relates to Simson in interfacing the ZED camera with ROS, which will be used in navigation.
- **Simson D'Souza**: Simson worked on processing point cloud data from the FARO scanner, setting up the ZED 2i and Intel RealSense D435i cameras, and tested mapping using the RealSense to generate a 2D costmap. He also designed the electrical circuitry diagram and collaborated with Deepam on fabricating the dozer blade. This is related to my work as the costmap map in addition to the localization of the rover will feed into the navigation stack and plan a path for the rover to take.

4 Plans

From now until ILR3, I plan to work on the localization stack of the rover. This effort would be in collaboration with my teammate Bhaswanth. The preliminary plan for the localization stack is to split into a global and local localization stack. The local localization stack uses internal sensors such as wheel odometry and the IMU and effectively uses "dead reckoning" to estimate its pose. The global localization uses internal sensors plus the external total station to localize in the global Moon Yard frame. The reasoning for the split is to ensure the local localization is continuous (the global localization may encounter discontinuities) so as to not cause problems with other planning algorithms we are planning to implement. Another reason is that if we were to suddenly lose connection to the total station, the rover would still be able to reasonably localize while waiting to reestablish communications.

Additionally, I also plan to set up the TF frame tree for the rover. Currently, we plan on having two fixed transforms. This would be from the `ZED camera` frame to the rover base frame (`base_link`), and the `IMU` frame to the `base_link` frame. We will also have two variable transforms. This will be from the `base_link` frame to the `odom` frame (for local localization) and from the `base_link` frame to the `map` frame (for global localization). The definitions of each frame is specified in REP-105 and can be found here: `http://www.ros.org/reps/rep-0105.html`.