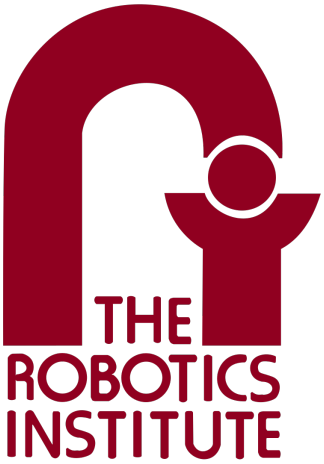
Individual Lab Report - 04



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*

March 21, 2025

Carnegie Mellon University
Robotics Institute

Contents

1 Individual Progress

Since the previous progress review, I have spent my time working mostly on the localization and navigation software stacks. A bit of my work was also involved in debugging the encoder telemetry issue.

1.1 Encoder Telemetry

During our Moon Yard tests, we noticed that no odometry data was being read from the drive motors, but odometry could be read from the steer motors. Initially, we suspected a code error. Our teleoperation code was built upon Crater Grader's code, and we originally intended to use a third Roboclaw motor controller to operate the dozer blade, just like the previous team did. However, we switched to a Linear Actuator Controller for the dozer's actuator instead, but forgot to update the code to remove the Roboclaw-specific section. This oversight could have caused bit shifting in the data due to the missing Roboclaw input. I added a simple fix in the code for this. However, after testing again, we saw that the issue still persisted.

To troubleshoot further, we began unit testing by checking all hardware connections. We discovered that the drive motors were connected in reverse, and one of the Roboclaw encoder connections was loose. Fixing them finally fixed the issue, and data could properly be read from the `/encoder_telemetry` topic.

1.2 Localization

I collaborated with William in completing the software implementation of the localization stack and we tested it in the Moon Yard. While testing, we observed that when we set the global fixed frame to `map`, the rover flew off unexpectedly. At first, we thought the problem was with the odometry because we weren't receiving drive motor encoder data from the `/encoder_telemetry` topic. After fixing that issue as I explained above, we tested again and visualized on Rviz. We observed that the issue still persisted with the `map` frame (Figure 1), but when we switched to the `odom` frame, everything worked fine (Figure 2). This shows that the local localization works correctly and the issue is with the global localization. We think the issue has something to due with transforming the `total_station_prism` to the `map` frame instead of `base_link` frame. We will make the necessary code changes to incorporate this and test again.

1.3 Navigation

I worked with Simson on the initial navigation stack setup. We built the ROS Humble `navigation2` packages on our Jetson Xavier board. We faced some issues with the `gazebbo_ros_pkgs` due to incompatibility with ARM processors, but we resolved this issue by removing its related files as dependencies from the `CMakeLists.txt` and `package.xml` files, as they were not required. We also configured the launch files to integrate our robot's URDF, and we successfully loaded the costmap that Simson generated into RViz as well.

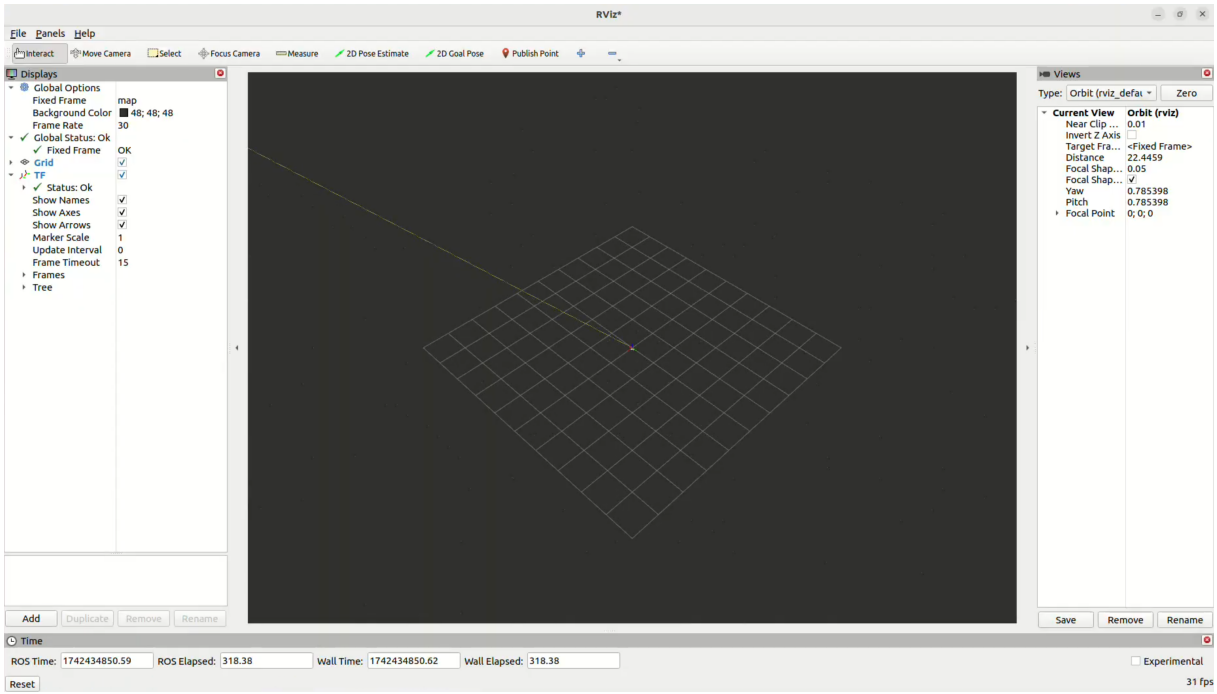


Figure 1: Global Localization Flying Off

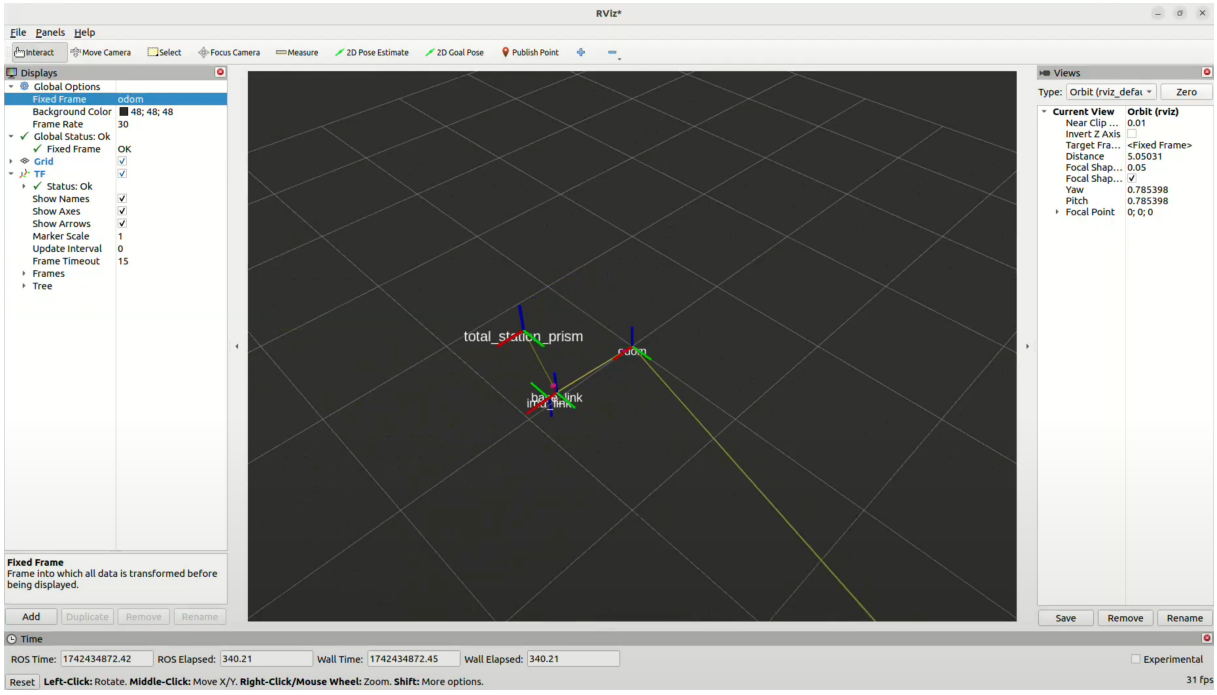


Figure 2: Local Localization Working Well

2 Challenges

The first challenge that the team faced was debugging the odometry data from the drive motors. The team spent a considerable amount of time on this and it was also a small blocked for the localization tests. This issue made us realize the risks associated with poor wiring and the team will work on replacing all the jumpers and connections correctly.

The biggest challenge for us is the localization stack, and it is critical since both the navigation and tool planner stacks depend on it. Localization issues are currently our main blocker, but we are treating this as a top priority and believe we are close to resolving it. We will be testing again very soon.

3 Team Work

- **Bhaswanth Ayapilla:** My work with William involved testing the localization stack in the Moon Yard. During testing, we realised that the issue is now with the global localization and we are debugging it together. I worked with Deepam in helping him implement dozer teleoperation. I also worked with Simson on the initial navigation stack setup on our Jetson board, and we will be collaborating together more on completing the navigation stack.

- **Ankit Aggarwal:** Ankit's work mainly focused on the tool planner methodology. He took inputs from the team for insights on the best way to set up the planner to minimize integration issues. He worked with Simson and Deepam for to set up a manufacturing plan for the E-Box. He also worked on debugging wheel odometry with me and William. Additionally, he worked with Deepam to mitigate the issue of rover breakdown due to a worn-out rear drive axle.

- **Deepam Ameria:** Deepam's primary work was to try different actuators of varying gear ratios and finalize the best one for our use case. He worked with me on making the tool capable of teleoperation. He collaborated with Simson to develop an ideal terrain by flattening the MoonYard and creating craters of various shapes and sizes, in order to develop a global map using FARO Laser Scanner. He worked with Ankit to mitigate the issue of the rover breaking down due to a worm out rear axle. They scavenged the spares off a twin rover and successfully replaced it on our ROADSTER. He also used the E-Box design made by Ankit to laser-cut the walls of the E-Box at TechSpark.

- **Simson D'Souza:** Simson worked on refining the global costmap and tuned parameters to obtain an accurate ground plane. To achieve this, he collaborated with Deepam to flatten the Moon Yard and create craters of various diameters and depths, allowing for a more precise terrain model. He worked on the navigation stack setup, collaborating with me to configure and integrate it on the NVIDIA Jetson. Furthermore, in collaboration with Ankit, the required parts for E-box manufacturing were finalized.

- **Boxiang (William) Fu:** William's work was in collaboration with mine in debugging the localization stack for the rover. We finalized the local localization (odom to base_link transform) using the IMU and wheel encoders, but the global localization (map to base_link) needs to be debugged. He also worked on the sensor stack using the RealSense depth camera, which will then be converted to an elevation map. The localization and sensor stacks will relate to mine and Simson's work on navigation, and also to Ankit and Deepam's work on the tool planner.

4 Plans

My plan until ILR05 is to fix the localization issue and work on the navigation stack of the rover. I will be collaborating with William for the localization tests and with Simson for navigation. The next steps for navigation are:

- Spawn the robot at a defined position in RViz
- Integrate with the localization stack
- Configure the local costmap params to utilize point cloud data from the RealSense camera
- Fine-tune Nav2 parameters for optimal performance on our robot and ensure accurate robot localization and real-time updates in RViz