# Individual Lab Report - 03

# 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 28, 2025

Carnegie Mellon University
Robotics Institute

# Contents

# 1 Individual Progress

Since the previous progress review, I have spent most of my time working on the localization stack of the rover. I will be talking about this in detail in the Localization as well as the Motion Control and Teleoperation subsections. A bit of my work was also involved in generating the map and the hardware of our rover.

## 1.1 Localization

In the previous progress review and ILR02, I spoke about setting up the TS16 Leica-Geosystems Robotic Total Station in the Moon Yard to obtain precise position of the rover. With the total station fully operational, along with the VectorNav IMU (which Ankit set up previously) and wheel encoders, I implemented the localization stack in collaboration with William.

The localization stack heavily uses the `robot_localization` ROS2 package. The nodes are connected as shown in Figure 1. We use two instances of the `robot_localization` package, as follows:

1. Global Localization:
   - Leica TS16: Provides precise x, y, z coordinates using the total station
   - Used for absolute positioning in a known reference frame
2. Local Localization:
   - VectorNav IMU: Measures acceleration and angular velocity for state estimation
   - Wheel Encoders: Tracks wheel rotations for relative movement estimation
   - Helps in odometry-based localization, ensuring continuous tracking of the rover's path

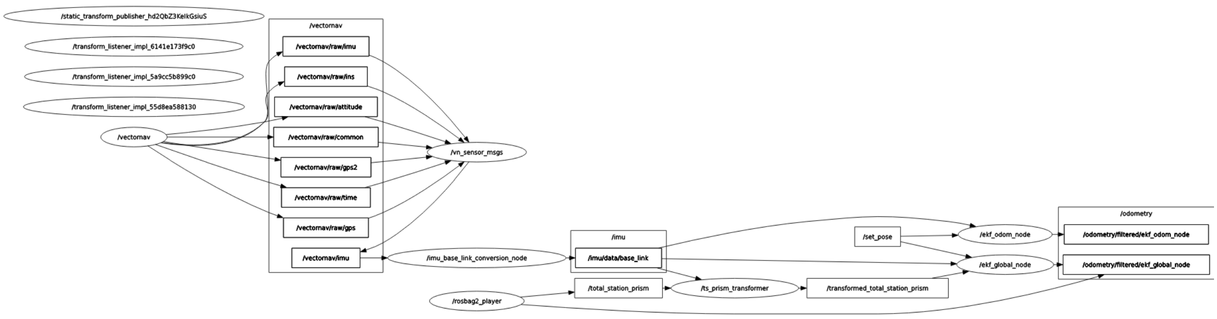


**Figure 1:** RQT Graph of Localization Stack

## Total Station (on TX2)

**Bringup**: `ros2 launch total_station total_station_launch.py`

**Usage**: Initiates the node that publishes total station data on the LAN network

**Nodes**: `/total_station_node`

**Publishes To**: `/total_station_prism`

**Subscribes From**: N/A. Reads total station data from the Leica using serial port in GeoCOM format.

## Vectornav

**Bringup**: `ros2 launch vectornav vectornav.launch.py`

**Usage**: Initiates the Vectornav IMU node that publishes data from the IMU. ROS readable formats are the ones labeled `\vectornav` without the `\raw` post-fix.

**Nodes**: `/vectornav` is the only node that are relevant to us

**Publishes To**: `/vectornav/imu` is the only publisher that are relevant to us. It publishes the IMU data in `sensor_msgs::msg::Imu` format.

**Subscribes From**: N/A

## IMU

**Bringup**: `ros2 launch imu imu_launch.py`

**Usage**: Initiates the node that converts the IMU readings in the IMU's frame and transforms it to the `base_link` frame.

**Nodes**: `/imu_base_link_conversion_node`

**Publishes To**: `/imu/data/base_link` is the IMU readings in the `base_link` coordinate frame.

**Subscribes From**: `/vectornav/imu`

## EKF

**Bringup**: `ros2 launch localization ekf_localization.launch.py`

**Usage**: Initiates the nodes that does global and local localization

**Node 1**: `/ekf_odom_node`

**Publishes To**: `odometry/filtered/ekf_odom_node` that gives the local localization of the rover. (See this link for explanation: https://discourse.ros.org/t/map-base-odom-as-alternative-for-rep-105-recommended-frame-order/25095)

**Subscribes From**: `/imu/data/base_link` and `encoder/odom`. There is also an action server `/set_pose` which could be used to change the pose of the localization manually.

**Node 2**: `/ekf_global_node`

**Publishes To**: `odometry/filtered/ekf_global_node` that gives the global localization of the rover.

**Subscribes From**: `/transformed_total_station_prism`, `/imu/data/base_link`, `encoder/odom`. There is also an action server `/set_pose` which could be used to change the pose of the localization manually.

**Node 3**: `/ts_prism_transformer`

**Publishes To**: `/transformed_total_station_prism` which transforms the total station coordinates to the coordinates at `base_link`

**Subscribes From**: `/total_station_prism` and `/imu/data/base_link`

**Node 4**: `/static_transform_publisher`

**Publishes To**: `/tf_static` with `parent_frame_id = base_link` and `child_frame_id = total_station_prism` which is the rigid transform between the two frames.

**Subscribes From**: N/A

## 1.2 Motion Control and Teleoperation

An important part of the localization stack is getting the correct odometry data to be used by the `robot_localization` package. To obtain this, I worked on the motion control stack, and its architecture is shown in Figure 2.

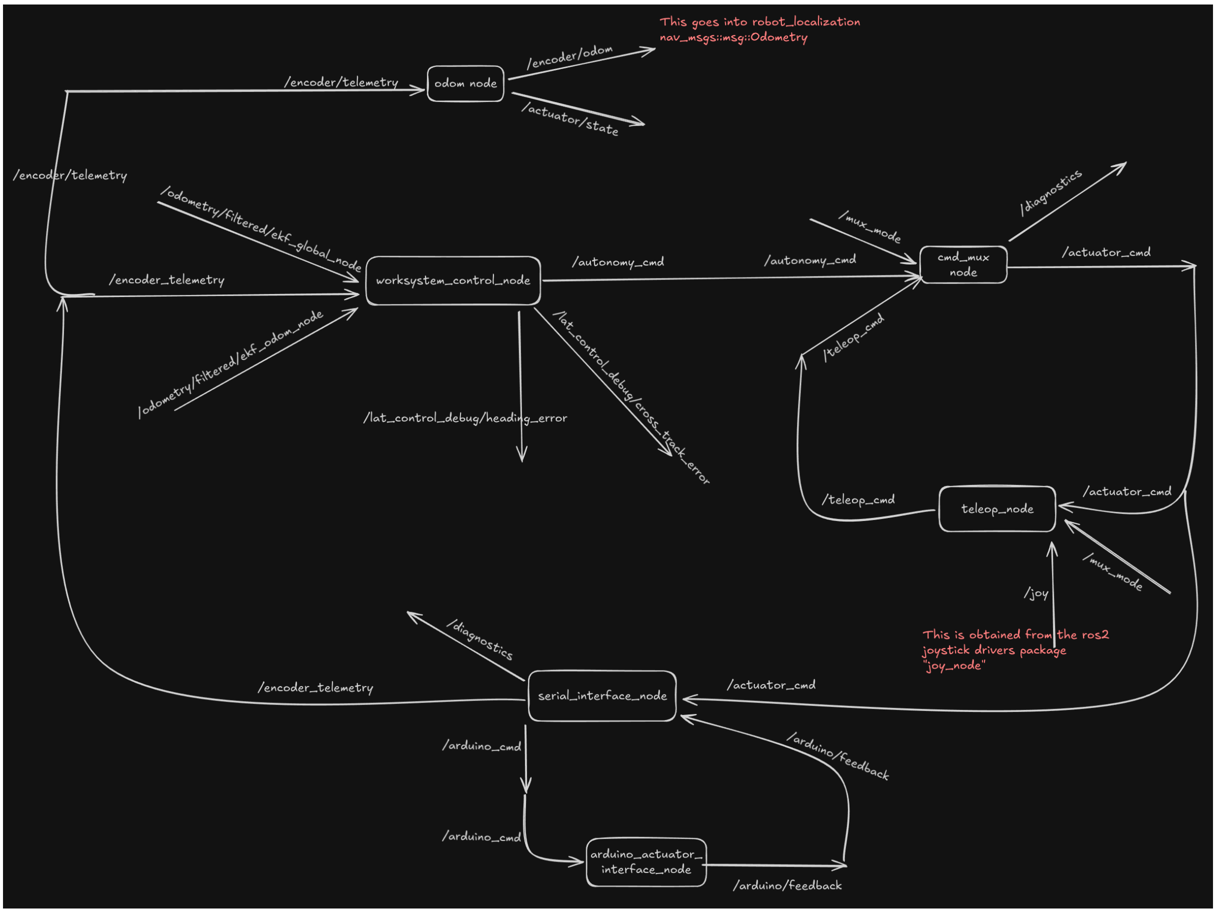


**Figure 2:** Motion Control Software Architecture

As a part of this, I also implemented the teleoperation code for the rover. Until now, we have been using Crater Grader's software to run the rover, but now we can operate it using our own code.

## 1.3 Mapping and Hardware

I collaborated with Simson to map the Moon Yard. We used the FARO laser scanner, which is a high-precision 3D scanning device that captures detailed point cloud data by emitting laser beams and measuring their reflections to create an accurate 3D model of the Moon Yard. Unlike our initial test, where we captured only a single scan, this time we conducted three scans from different locations within the Moon Yard. To evaluate how effective our scans are, we dug several craters before scanning (see Figure 3). Each scan took approximately 11 minutes, with color enabled and set to high quality and resolution. Simson merged the three scans into a single, high quality map. This map will be used to identify craters for the rover to grade, and will be used for navigation as well.

I collaborated with Deepam to actuate the dozer assembly using a linear actuator. We used the L16-P Miniature Linear Actuator with Feedback (50mm 63:1 12 volts) along with the Linear Actuator Control (LAC) Board. The LAC allows us to use the L16 -P actuator as a linear servo. The board reads the position signal from the L16's potentiometer,

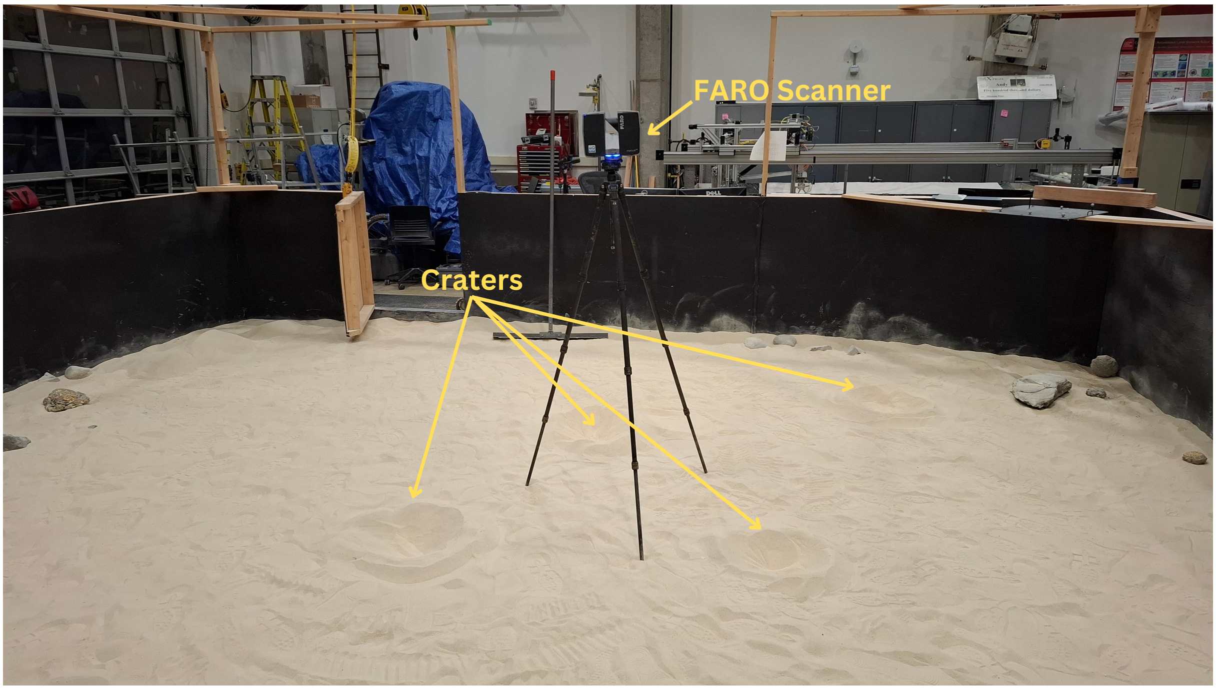


**Figure 3:** FARO Laser Scanner in the Moon Yard

compares it with the input control signal, and then commands the actuator to move via an onboard H-bridge circuit. We connected the board to a PWM pin on the Arduino Due and wrote a simple test script to verify the actuation. However, upon testing, we found that the current actuator struggles to handle the load, so we placed a new order for a more powerful linear actuator with a higher gear ratio of 150:1.

## 2 Challenges

One of the challenges we encountered during the initial testing of the localization stack was that the Jetson AGX Xavier on the rover couldn't receive the total station data published by the TX2. After some troubleshooting, we realized that when I initially built the docker image, I set the container's network mode to default as "bridge", instead of "host". It took us a while to figure this out and I rebuilt the docker container as "host" later. This delay prevented us from testing that day.

Another challenge was since Ankit and Deepam got their access to FRC only this week due to Tim's availability, the rover was disassembled to be fitted with the dozer arms. As a result, we were unable to test the localization stack. Since the manufacturing is now complete and the rover has been fitted with all the electronics again, we will test very soon.

## 3 Team Work

- **Bhaswanth Ayapilla:** My main work was in collaboration with William in implementing the localization stack for the rover. The code for the global localization and local localization is complete, and we will be testing it in the coming week. I also collaborated with Simson in mapping the Moon Yard using the FARO laser scanner, which we will use for navigation. I helped Deepam to operate the linear actuator for the dozer assembly. Additionally, I collaborated with Ankit in creating the PDB schematic.

- **Ankit Aggarwal:** Ankit worked with Deepam in manufacturing the dozer assembly for the rover. He collaborated with Simson to create occupancy maps from the mapped point cloud data of the Moon Yard. He collaborated with Deepam for the PDB conceptual design, and with me to create the schematic of the PDB. Additionally, to design the new electronics box, he used inputs from the entire team.

- **Deepam Ameria:** Deepam spearheaded the manufacturing of the dozer blade assembly. He collaborated with Ankit to manufacture individual components like the dozer arms, blade, mounts, etc. at the FRC workshop. He collaborated with Ankit and me for the actuator selection and implementation to make an active dozer assembly. He also worked on conceptualizing the PDB for our system by drafting out the power requirements of the system and benchmarking the components.

- **Simson D'Souza:** Simson worked with me in mapping the moon yard using the FARO laser scanner. He then processed the data to generate an occupancy grid map, which will be used for navigation. To obtain an accurate occupancy grid map, several parameters had to be tuned, and Ankit assisted him with it.

- **Boxiang (William) Fu:** William's work was in collaboration with mine in implementing the localization stack for the rover. This includes local localization (odom to base_link transform) using the IMU and wheel encoders, and global localization (map to base_link) using additional data from the total station.

# 4 Plans

My plan until ILR04 is to work on the navigation stack of the rover. As of now, we operate the rover only on teleoperation mode. Operating the rover in fully autonomy is one of our primary goals for this semester and this is a big task too. I will be collaborating with Simson for this. Our work will involve working on the simulations first, and then translating the simulation to real world. Additionally, I will be testing the localization stack along with William.