Individual Lab Report 3

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 28, 2025

Carnegie Mellon University
Robotics Institute

1 Individual Progress

Since the last progress review, I worked on finalizing and testing the external infrastructure for the project. After finishing all the external dependencies, I collaborated with Bhaswanth on writing code for the localization stack of the rover. The software stack is finished, what is left is to test and tune the parameters.

1.1 External Infrastructure Testing

The external infrastructure for the project consists of the total station, TX2 chip next to the Moon Yard, and our LAN network router (see Figure 1). Each individual component was set up and communication was established since the last progress review. However, some parameters were not tuned and the TF transforms tree was not set up. The initial origin of the global map frame was at the Leica Mini 360 Prism that was hanging on the wall. I finalized the offset of the total station pose readings so that the origin is at the corner of the Moon Yard closest to the entrance. This is a much more intuitive coordinate frame for navigation and localization. The offsets used are discussed in the Reference Frame Documentation subsection.

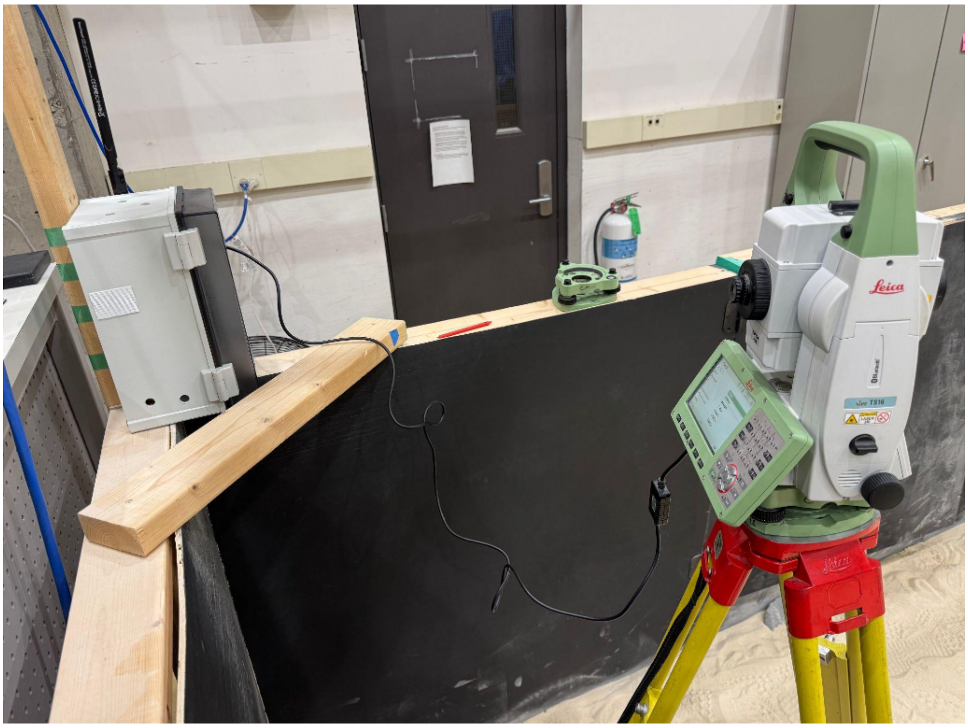

Figure 1: Testing the Total Station and TX2 Chip

1.2 Reference Frame Documentation

Before starting the localization stack, the reference frames that exist for our system needs to be defined first. Currently, there are a total of five reference frames. The frames are:

- **map**: This is the global frame with origin at the corner of the Moon Yard closest to the entrance.
- **odom**: This is the local frame origin that offers a continuous homogeneous transform to the `base_link` frame.

- **base_link**: This is the frame of the rover at its Center of Mass.
- **total_station_prism**: This is the frame of the Leica Prism sitting on top of the rover.
- **imu_link**: This is the frame of the IMU.

A separate distinction between the map and odom frame is made due to the way ROS handles continuous and discontinuous homogeneous transforms. In simplified terms, the transform between odom and base_link is continuous over time, whereas the transform between map and base_link may be discontinuous due to discrete updates from the total station. Official documentation for this can be found under REF 105. The current TF transform tree is depicted in Figure 2.

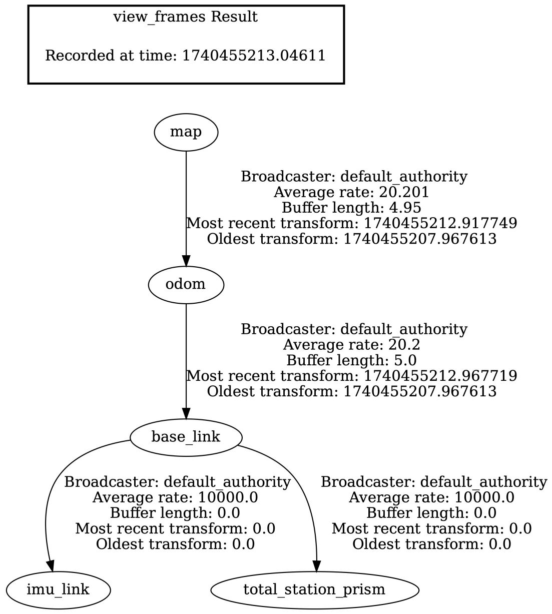

Figure 2: TF Tranform Tree for Lunar ROADSTER

In the above homogeneous transforms, the child frames of base_link are all static transforms. The transforms from map to odom and odom to base_link are non-static and vary over time. The static transforms do not change over time and are set as shown in Figure 3. Additionally, the wheel base and the total station offset parameters were also calculated. The wheel base is used to convert encoder ticks to velocity (for wheel odometry localization) and the total station offset is used to set the map frame to a intuitive coordinate frame. Their values are shown in Figure 4.

1.3 Localization Stack

After finalizing the frame transforms for the project, I collaborated with Bhaswanth on implementing the localization stack of the rover. This was done using code left by Crater Grader and the robot_localization ROS package. There are three input streams that feed into the localization stack. The first data stream is from the Vectornav IMU. Ankit set up the drivers and SDK, while I transformed the data to the base_link frame. The second data stream is from the wheel encoders. The raw encoder ticks is transformed to linear and angular velocity before feeding it to robot_localization. Finally, the third data stream is from the total station. The data is again transformed to the base_link frame before feeding it to robot_localization.

Static Parameters and Where to Find Them

Total Station Beacon

Location: `localization/config/total_station_beacon_static_tf_params.yaml`. Confusingly, it appears also in `ts_prism_transformer.cpp` and is not linked to the YAML file.

Bringup: EKF launch file

Parent Frame: `base_link`

Child Frame: `total_station_prism`

Transformation: `{X, Y, Z, R, P, Y} = {-0.20033, -0.020133, -0.75628 0 , 0, 0}`

IMU

Location: `imu/config/imu_static_tf_params.yaml`

Bringup: IMU launch file

Parent Frame: `base_link`

Child Frame: `imu_link`

Transformation: `{X, Y, Z, R, P, Y} = {0.071406, 0, -0.0033096, 3.14159265359 , 0, 0}`. The IMU frame defaults the z-axis to point downwards, whereas the base frame has z-axis pointing upwards.

Figure 3: Static Transforms for Lunar ROADSTER

Wheel Base

Location: `motion_control/config/odom_node_params.yaml` (`motion_control/src/odom_node.cpp` also has default parameters if loading the YAML file fails)

Bringup: Motion Control launch file

Parent Frame: N/A

Child Frame: N/A

Parameters:

- "half_wheel_base_m", 0.2775
- "qp_steer_to_radian", 0.000099223014335
- "qp_drive_to_pos_m", 0.0008298755187
- "qpps_drive_to_speed_ms", 0.0008298755187
- "qp_tool_to_fs", -0.00000666933

Total Station

Location: `JetsonTX2/src/total_station/config/total_station.yaml`

Bringup: `ros2 launch total_station total_station_launch.py` (on TX2)

Parent Frame: N/A

Child Frame: N/A

Parameters:

- ts_prism_x_offset: 2.25
- ts_prism_y_offset: 0.00
- ts_prism_z_offset: 2.2
- ts_prism_z_rotation: 0.0

Figure 4: Static Parameters for Lunar ROADSTER

To conform with REF 105 convention, a local localization stack and a global localization stack were implemented. The local localization stack uses only the first two data streams that are contained on the physical rover. The global localization stack uses all three data streams, including the total station. The node and topic dependencies are depicted in the RQT Graph in Figure 5. Note that when the graph was taken, the wheel encoder node was not publishing and the total station data was publishing from a recorded ROS bag.

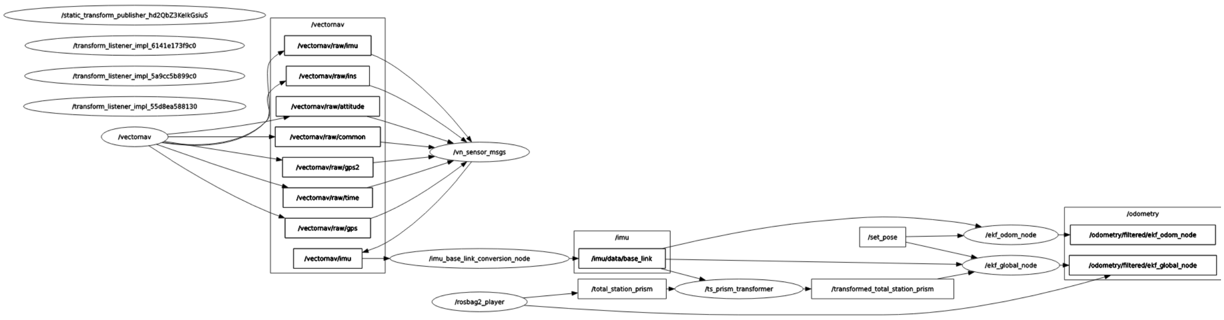

Figure 5: RQT Graph of Localization Stack

2 Challenges

One significant challenge faced was integrating the published total station data with our localization stack. The total station state publisher was written by Crater Grader and modified by LunarX. For some reason, the previous team decided to connect the `total_station_prism` frame with the `map` frame. This was not correct according to REF 105 convention and resulted in an incorrect TF tree shown in Figure 6.

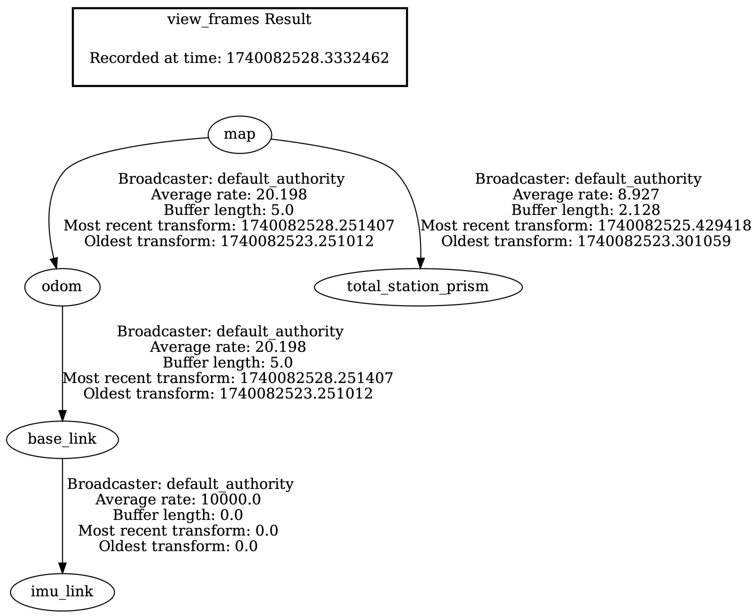

Figure 6: Incorrect TF Transform Tree

Another challenge that we faced was that while the Jetson was able to receive the topics published by the TX2 chip, once we enter the docker container housing our code, the topic connection was lost. The reason turned out to be that the container was incorrectly set up. The docker driver was set up as “bridge” when instead it should be set as “host” (see Figure 7). Changing this allowed the container to communicate with the host system.

Figure 7: Incorrect Jetson Docker Driver Setup

A final challenge was on the availability of the physical rover for testing. Since the dozer hardware needs to be fitted this week, the rover was taken apart and unavailable for testing. This meant that we were not able to fully test our localization stack as we are missing data from the wheel encoder topic. As a compromise, we tested and tuned the localization stack using only the total station Leica 360 Prism and the IMU. Once the rover is rebuilt, we will further add on data from the wheel encoder topic and test the full localization stack.

3 Teamwork

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

- **Ankit Aggarwal**: Ankit collaborated with Deepam to manufacture the dozer assembly and for the PDB conceptual design. He also collaborated with Simson to create occupancy maps from the mapped point cloud data of the Moon Yard. Finally, Ankit collaborated with Bhaswanth to create the schematic of the PDB. To design the new electronics box, Ankit used inputs from everyone on the team to ensure accessibility.

- **Deepam Ameria**: Deepam spearheaded the manufacturing of the Dozer Blade Assembly. Individual components like the dozer arms, blade, mounts, etc. were manufactured at the FRC workshop in collaboration with Ankit. Deepam also collaborated with Ankit and Bhaswanth for actuator selection and implementation to make an active dozer assembly. Finally, Deepam worked on conceptualizing the PDB for our system by drafting out the power requirements of the system and benchmarking the components.

- **Bhaswanth Ayapilla**: Bhaswanth's 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. Bhaswanth also collaborated with Simson in mapping the Moon Yard using the FARO laser scanner, which we will use for navigation. Additionally, Bhaswanth helped Deepam to operate the linear actuator for the dozer assembly. Finally, Bhaswanth collaborated with Ankit in creating the PDB schematic.
- **Simson D'Souza**: Simson worked on mapping the moon yard using a FARO laser scanner in collaboration with Bhaswanth. 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 that.

4 Plans

From now until ILR4, I plan to finish the testing and tuning of the localization stack. This includes both the global and local localization stack. Additionally, Bhaswanth added a wheel slip estimation node that uses the difference between the global and local localization to estimate how much the wheel is slipping. I plan on collaborating with him to also test this.

After finishing localization, my next task is to implement the Finite State Machine (FSM) node. This should be a relatively quick task as we would be adapting Crater Grader's FSM planner. Preliminarily, we plan on having four states plus an error state for the FSM planner. They are as follows:

1. **IDLE**: Sets wheel velocity to 0.
2. **AUTOGRADER**: Rover's wheel and steering is controlled by the joystick, but the tool is controlled autonomously.
3. **FULL_AUTONOMY**: Both the rover's motion and the tool is controlled autonomously.
4. **FULL_TELEOP**: Both the rover's motion and the tool is controlled by the joystick.
5. **ERROR_MODE**: Error mode when something unexpected occurs.

Additionally, I also plan on collaborating with Simson with the rover's mapping subsystem. Simson is currently in charge of generating a global map of the Moon Yard. I will be using the ZED camera to generate a local map from the camera's field-of-view and update the global map over time. This allows the map to be updated in places that the rover has already dozed a traversable path.

Finally, with the preliminary design review (PDR) due the week after Spring Break, the team plans on spending time preparing the PDR presentations and recording demonstrations for various subsystems.