Individual Lab Report - 5

Lunar ROADSTER

Team I

Author: **Ankit Aggarwal**
Andrew ID: ankitagg
E-mail:
ankitagg@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: **Boxiang (William) Fu**
ID: boxiangf
E-mail: *boxiangf@andrew.cmu.edu*

Supervisor: **Dr. William "Red" Whittaker**
Department: Field Robotics Center
E-mail: *red@cmu.edu*

April 10, 2025

Carnegie Mellon University
Robotics Institute

1 Individual Progress

1.1 PDB Manufacturing

The Lunar ROADSTER Power Distribution Board has been manufactured. I was able to source all components and the team helped solder the boards. We made a spare board as well, to ensure our hardware risk mitigation plan remains on track. Figure 1 shows the fully soldered PCBs.

Figure 1: Manufactured PDBs

During the manufacturing of the E-box, we integrated the PCB. The PCB is functional and working on the rover. It has also helped us eliminate previous issues of high in-rush current and unstable voltage supply. The indicator LEDs also help us monitor supply to each actuator.

1.2 Wheel Testing

All 4 wheels have been printed, along with a spare. We tested the wheels in the Moon-Yard and they have improved mobility significantly. We have observed much better traction in each wheel, improving the drawbar pull and allowing us to manipulate sand more effectively. As the wheels are rigid and the grousers sink into the sand evenly, we speculate that the effective weight on each wheel has also been equalized. This has allowed us to steer more efficiently. Fig 2 shows the rover in the MoonYard with the 4 wheels mounted.

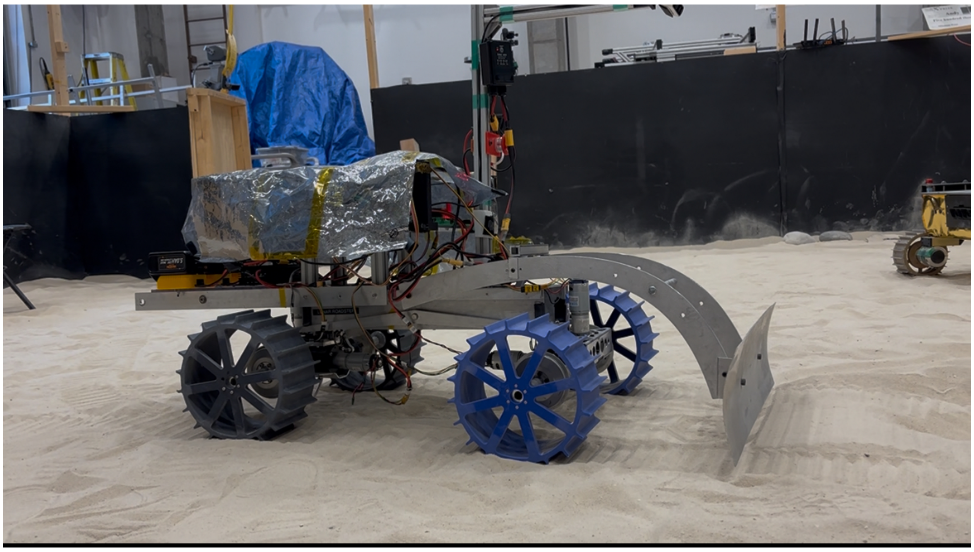

Figure 2: 4 printed wheels mounted on the rover

1.3 Tool Planner

1.3.1 Implementation

Based on the problem formulation detailed in the last ILR, I created the TransportPlanner class by using CraterGrader's code as a base. The pipeline is as follows:

1. **PointCloud to FilteredMap:** Collaborating with William, we wrote a script to convert the PointCloud generated by the FARO Laser Scanner into a grid map. This map is a 2D Matrix with each cell having x, y and depth data. This is saved as a CSV file.

2. **Initializing source and sink nodes:** The TransportPlanner traverses through the grid map and assigns source and sink nodes based of depth thresholds (currently, source is +2cm and sink -1cm).

3. **Setting up the optimization problem:** Using the ORTools library, we set up the optimization problem to minimize distance and volume of sand to be manipulated. The problem is solved using the Google Linear Optimization Package (GLOP).

4. **Transport assignments:** The solver outputs optimal transport assignments, defined as a Struct containing source node location, sink node location and volume of sand to be transported.

5. **Creating waypoints:** Using the transport assignments, the TransportPlanner creates waypoints. It currently outputs 2 goal poses per assignment, corresponding to the locations of the source and sink nodes. The yaw at each pose is calculated using the inverse tangent of Δx and Δy. We may add an additional pose for the rover to safely exit the manipulation area without affecting the manipulated area.

6. **Output to navigation stack:** The created goal poses are stacked in an array sequentially and sent to the navigation stack using a shared pointer.

Figure 3 shows the current outputs of the transport planner. It needs to be tuned and we will also be testing it with various different maps.

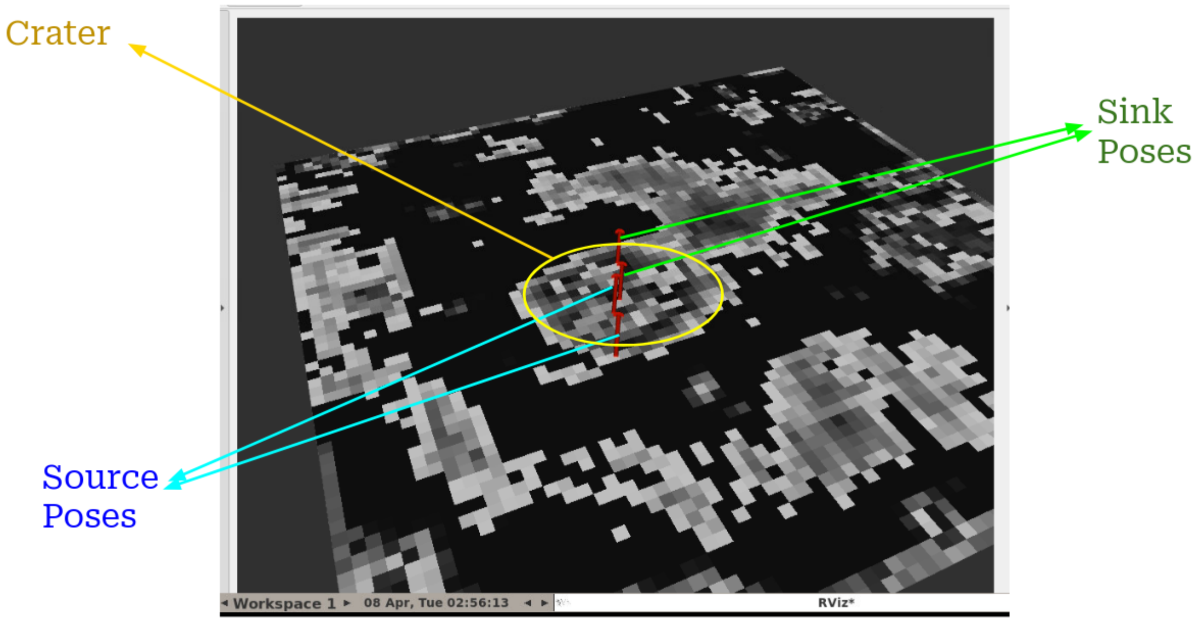

Figure 3: TransportPlanner output poses

1.3.2 Integration

I also worked on integrating the TransportPlanner with the BehaviourExecutive Node (Finite State Machine Planner). Fig 4 shows the debug statements of the BehaviourExecutive node.

```
[behavior_executive_node-1] ~~~~~~~ Machine iteration
[behavior_executive_node-1]     Pre-Signal: YES
[behavior_executive_node-1]       State L0: PLAN_TRANSPORT
[behavior_executive_node-1]       State L1: TRANSPORT
[behavior_executive_node-1] PLAN_TRANSPORT
[behavior_executive_node-1] Number of source nodes: 436
[behavior_executive_node-1] Number of sink nodes: 480
[behavior_executive_node-1] Source volume: 13.252
[behavior_executive_node-1] Sink volume: 7.73717
[behavior_executive_node-1] Number of transport assignments: 2
[behavior_executive_node-1]     obj value: 13.0855
[behavior_executive_node-1] ~~~~~~~ Machine iteration
[behavior_executive_node-1]     Pre-Signal: TRANSPORT_PLANNED
[behavior_executive_node-1]       State L0: GET_TRANSPORT_GOALS
[behavior_executive_node-1]       State L1: TRANSPORT
[behavior_executive_node-1] GET_TRANSPORT_GOALS
```

Figure 4: BehaviourExecutive Node debug output

1. **Previous State - MAP_EXPLORED:** If the system receives a current and a desired map, this state transitions forward.
2. **State - PLAN_TRANSPORT:** In this state, the system sets up and solves the optimization problem using the grid map given by the previous state. When the transport assignments are finalizes, it transitions forward. All required debugging parameters are logged and printed on the terminal.
3. **State - GET_TRANSPORT_GOALS:** In this state, the system creates waypoints using the goal assignments and stacks them into a shared pointer. It transitions when all waypoints are added.

4. **Next State - GOALS_REMAINING:** Here the system checks for any remaining goal poses. Since GET_TRANSPORT has added new goals, the system transitions to the state for creating trajectories using the Navigation Planner.

1.4 Rover Hardware Maintenance

1. **Steering Motors:** The persistent rear steering issue has been solved. We glued the pinion onto the shaft and then drilled a radial through hole. A dowel pin has been added to the radial hole, securing the pinion in place. Further, the gearbox of the front steering motor got damaged and was replaced it.

2. **Electronics Debugging:** The team manufactured my E-Box design. However, there were issues relating to wire to pin mapping which needed to be identified and solved. Further, all jumper connections were secured in place using electrical tape.

2 Challenges

The only challenge with PDB manufacturing was re-learning how to solder well. The onboard connectors were challenging to solder as they had a lot of open space where a wire would usually be placed.

With the tool planner, tuning parameters still continues to be a challenge. As the environment is super sensitive to even footsteps, it generates 'false sink nodes' which causes haywire solutions. This would need a new and final map to tune parameters for, before SVD. The integration of the tool planner with the FSM was challenging as it is my first time working with a ROS2 acrchitecture of this scale. Figuring out all dependent packages and files took longer than expected.

3 Teamwork

My main work was implementing the tool planner and testing it using the FARO laser scan. I worked with William to integrate the Tool Planner into the FSM and he helped me visualize the planner outputs in RViz. I collaborated with Simson, Deepam and Bhaswanth to solder the PCBs. I also worked with them to debug and finalize the wiring connections in the new manufactured E-Box and general hardware debugging of the rover.

Bhaswanth Ayapilla: Bhaswanth's initial work was in collaboration with William in debugging the global localization stack and correcting the yaw of the rover. He worked with Simson on the navigation stack and integrating it with localization. He collaborated with Deepam, Simson and I in soldering the PCBs, assembling the E-box, testing it and troubleshooting issues, and performing quality assurance of the entire hardware setup.

Deepam Ameria: Deepam worked with Simson, Bhaswanth and I to solder the PCBs and also for manufacturing, assembling, testing and debugging the E-box and the other hardware of the rover. He also worked on understanding and charting out the software architecture, mainly to understand how the planning stack fits into the whole system. He also worked on visualising the planning outputs with William.

Simson D'Souza: Simson's primary responsibility was setting up the navigation stack and integrating it with the localization stack, which was done in collaboration with Bhaswanth.

Additionally, the moon yard was scanned using a FARO scanner with Bhaswanth's assistance. Alongside Bhaswanth, Deepam, and I, he contributed to soldering PCBs, assembling the E-box, wiring new connections based on the updated E-box design, and troubleshooting hardware issues on the rover.

Boxiang Fu: William's initial work was mainly in collaboration with Bhaswanth in debugging the global localization stack of the rover. He worked with Bhaswanth on integrating localization with the mapping stack to output a global elevation map. Next, he collaborated with the team to write the skeleton code for the FSM behavior tree by getting each subsystem's expected inputs and outputs.

4 Plans

The team plans to be ready for SVD and extensively test our system. My plans are:

1. Tune the Tool Planner to generate optimal goal poses
2. Implement code for back-blading (independent of the tool planner)
3. Perform testing and trial runs for SVD
4. Quality assurance for rover hardware