16-681 MRSD Project Course 1

Individual Lab Report - 02

Name: Dhruv Tyagi

Team C: Lunar Autonomous Regolith Excavator [LunAR-X]

Teammates: Vibhakar Mohta, Anishram Senathi, Hariharan Ravichandran, Vivek Chervi

Feb 16th 2023

Contents

1. Individual Progress 2
   1.1 Sensor Placement 2
   1.2 Joystick setup 2
   1.3 High-level software & network architecture 2
2. Challenges 3
3. Teamwork 4
4. Plans 4

1. Individual Progress

1.1 Sensor Placement

In continuation of the simulation work from the previous report, I tried out multiple different placement configurations in that simulation setup. This was to figure out the best configuration for the 2 Realsense cameras which will maximize the view of the environment while also maintaining adequate view of the berm that the robot will build.
The final configuration chosen has been visualized in Fig 1.

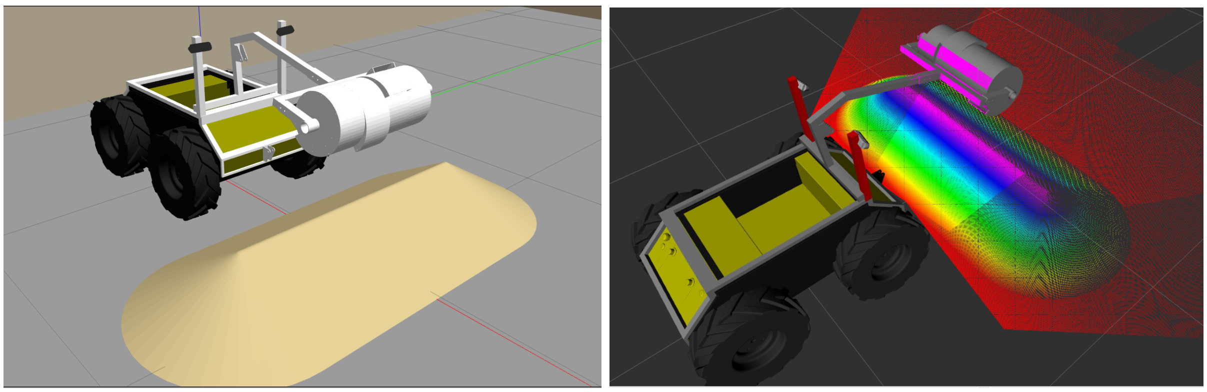

Fig 1. Sensor placement configuration

This configuration has the 2 cameras mounted on 2 different masts, facing 8 degrees outwards and 37 degrees downwards. This meets our requirements and also ensures higher accuracy in the dump region due to an overlap of the 2 camera-views.

1.2 Joystick setup

Another task was to set up the joystick for tele-operation. We have decided to keep a control station outside the Moonyard/workspace for monitoring and controlling the robot whenever required. The tele-operation commands will be sent through this control station. To enable this, I set up a docker image which will interface with the joystick and publish commands over a wireless network to the robot. I tested if all the joystick buttons and sticks were functional and were publishing the right commands.

1.3 High-level software & network architecture

Since we have multiple hardware platforms, we need an efficient and clear architecture for the network. This will define the responsibility and dependency of each of the platforms. A high-level architecture is visualized in Fig 2. Following is a short description of each of the platforms:

- Nvidia Xavier AGX:
    - This is present on the robot itself.
    - Responsible for all the on-board decision-making on the robot, it is the core brain of the robot and the platform which will have all of the autonomy software.
- Nvidia TX2:
    - This is present on the Total station.
    - The total station is the equipment for the external localization of the robot. This TX2 platform will take in the data from the total station and publish it over the network for the robot's localization.
- Control Station:
    - This can be any personal computer outside of the workspace
    - Responsible for monitoring and controlling the system whenever required.

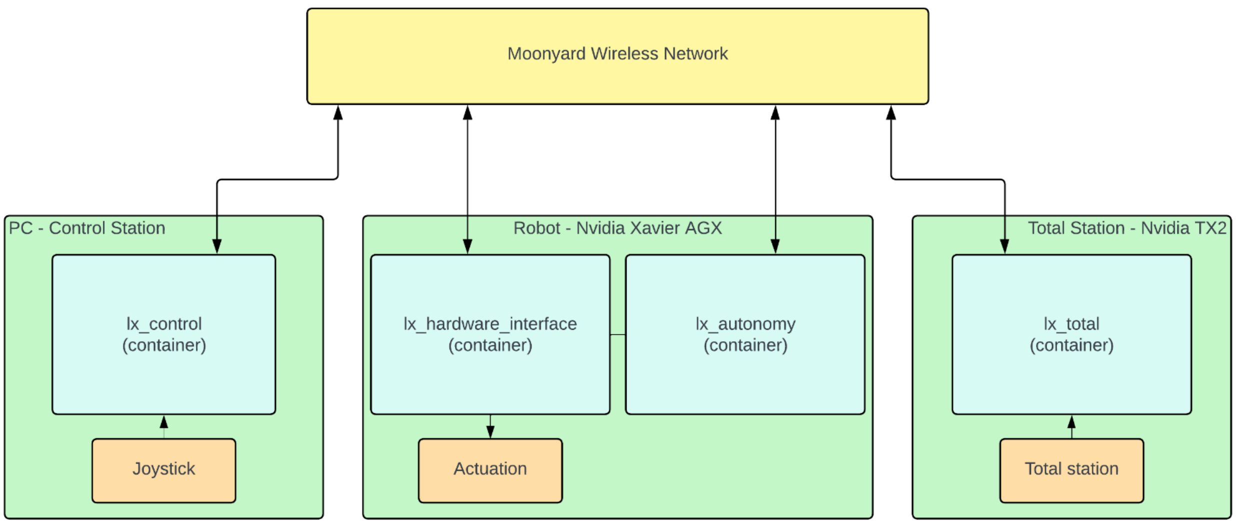

Fig 2. High-level software & network architecture

2. Challenges

A major challenge I faced was while trying to configure the docker image to mount dynamically plugged in USB devices (the joystick in our case). To interface a joystick input with a docker container, we need to specify the device ID at the start of the container itself. But what if the joystick was disconnected and has to be reconnected? This would require us to restart the entire docker container as well, which is not ideal.
The closest solution I found was to bind-mount the "/dev" folder of the host machine to the container. Although this is still not ideal for a personal computer (where security is key) and can only be implemented on the devices we will connect to the Xavier

platform. For the tele-operation joystick, we will have to continue to make sure that it is already plugged into the computer before the container is started.

3. Teamwork

- Vibhakar Mohta: Vibhakar worked on setting up and getting the sensors functional. He also participated actively in brainstorming about electronics design.
- Hariharan Ravichandran: Hariharan led the drum assembly manufacturing and the design for the lifting mechanism.
- Anish Senathi: Anish was heavily involved in the drum assembly manufacturing and also discussed a few ideas for the power circuit.
- Vivek Chervi: Vivek led the electronics design and made the preliminary architecture also taking into consideration everyone's inputs.

4. Plans

The primary task now is to flesh out the core autonomy architecture and start programming the utilities which would be required by the rest of the software. This will be done along with setting up the docker image for the core autonomy repository, which will run on the Xavier platform. This needs special attention to detail as we plan to integrate CUDA compatibility in the future for processing the data from the cameras and the base docker configurations do not directly support CUDA capabilities. The last task includes testing if the Xavier platform receives the joystick commands wirelessly over the multi-container network described in section 1.