

Individual Lab Report 8

Progress Review 9

Shivam Gautam
sgautam@andrew.cmu.edu

February 24th, 2016

1 Individual Progress

I undertook the following tasks for this progress review-

1. Debugging Oculus Prime Hardware
2. Debugging Oculus Prime Software
3. Conducting mapping using GMapping and Hector- Slam
4. Conducting waypoint Navigation on Oculus Prime
5. Defined Scope and Architecture of the Simulation Environment

1.1 Debugging Oculus Prime Hardware

The Oculus Prime hardware we received from Xaxxon did not work as expected. The platform did not charge while docking. While we could charge the battery externally using a Li-Po charger, this was not a long term solution as the Oculus takes the dock as its origin during mapping. It was essential for the platform to be 'docked' before the navigation or mapping node could be launched.

I worked on identifying the root cause of the problem and started inspecting the connections and the contacts. I started off by checking all the connections for continuity. I then went on to check for the voltages at the docking station and the platform contacts. The dock showed a voltage of 14 Volts whereas the contacts were at 10 Volts. After trying with another dock, the platform persisted which could mean that the contacts on the platform were dropping voltage. After replacing the contacts with the contacts on the previous platform, the voltage drop remained. The problem was not solved by sanding the contacts.

After observing the voltage on the contacts using an oscilloscope, I found that the power brick dropped power intermittently to give an average voltage of 10V. After replacing the power supply on the dock, the problem was solved.

1.2 Debugging Oculus Prime Software

The Oculus Prime software did not run out of the box again. The scripts that it was supposed to launch- Oculus prime server, GMapping and navigation were not launched on startup. Initially, the platform would not even connect via its own WiFi.

The GUI provided by Xaxxon also did not launch these specified scripts. After looking into the software repository for the Oculus, we launched the scripts manually- first by actually accessing the OS by an external monitor and then later by SSH.

To solve the problem of the WiFi, I set up a router to which the Oculus Prime computer could automatically connect with a static IP. The network created by the router allowed any of our laptops to connect to it and in turn be connected to the Oculus Prime.

1.3 Conducting mapping using GMapping and Hector-Slam

Before the GMapping package on the Oculus prime was being used because of the issues with the platform, I worked on conducting mapping without using odometry data. The package that was most suitable for this was the Hector Slam package which did not rely on the odometry data.

The pipeline involved was as follows-

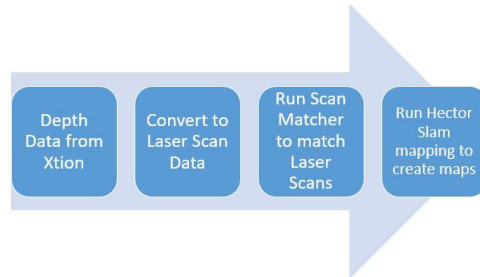


Figure 1: Pipeline for mapping without odometry

Mapping using the GMapping package on the Oculus Prime was finished before I could finish mapping using hector slam. Nevertheless, I plan to use this to generate better maps using the Hokuyo sensor in the future.

The map created by GMapping is a 2D projection of the 3D world and consists of a PGM file (the actual 2D map) and a YAML file (the metadata for the map). The map created can later be edited in a bitmap editor. Maps were created by teleoperating the platform inside a mock-parking lot. A few of the maps created were as follows-

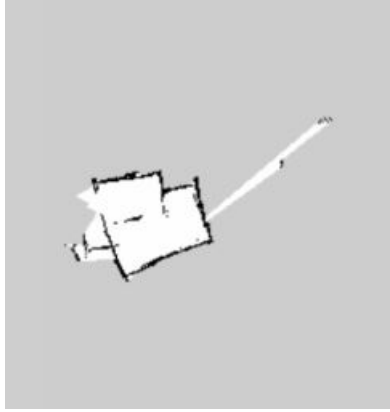


Figure 2: Unedited Map of mock parking lot



Figure 3: Edited Map of mock parking lot

1.4 Conducting waypoint Navigation on Oculus Prime

The created map was used to conduct waypoint navigation as well. The platform was given a go-to waypoint and then later a sequence of waypoints. The platform was able to follow both the trajectories after facing some initial issues with the localisation. Due to the size of our mock parking lot, the AMCL localisation module did not have great accuracy. The times it did localise itself, it was able to follow the waypoints satisfactorily. This is depicted in figure 4.

1.5 Defined Scope and Architecture of the Simulation Environment

We decided to make a simulation environment that would show how our own planner for the environment would handle a large volume of cars. The planner has three different aspects to it-

- Global Planner This will be responsible to rank and assign costs to parking

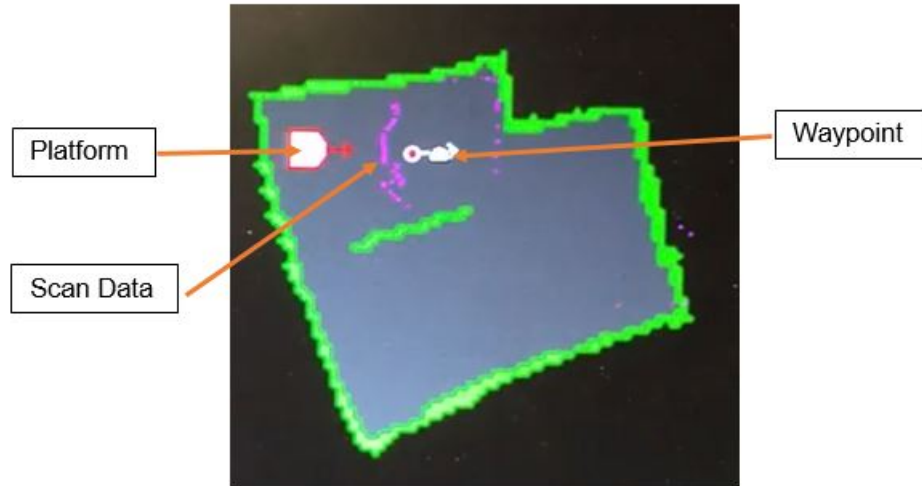


Figure 4: Waypoint Navigation

spots based on multiple heuristics. The decision to assign a spot to the car would be made by this unit. This information is then relayed to the simulation engine to be processed and rendered in the environment. This planner would take into account the information from the path planner.

- Path Planner The path planner takes in inputs from the Simulation Engine and the Global Planner to plan the most optimal path from any given point A to B, taking into account different heuristics, such as A* path cost, motion segments, type of motion and current state of the lot.
- Simulation Engine It accepts the data from other nodes to render an environment showing the state of all the vehicles in the parking lot. It will also update the other systems with the real-time state of the lot so that planning takes the dynamic environment into account. This is depicted in figure 5

2 Challenges

One of the biggest challenges I faced, as documented before, was working with the platform. This involved debugging the hardware as well as the software for the platform.

Another challenge arose when I tried to map a makeshift parking lot. The parking lot was experimental and we had to continuously fiddle with the structure and materials to create useful maps.

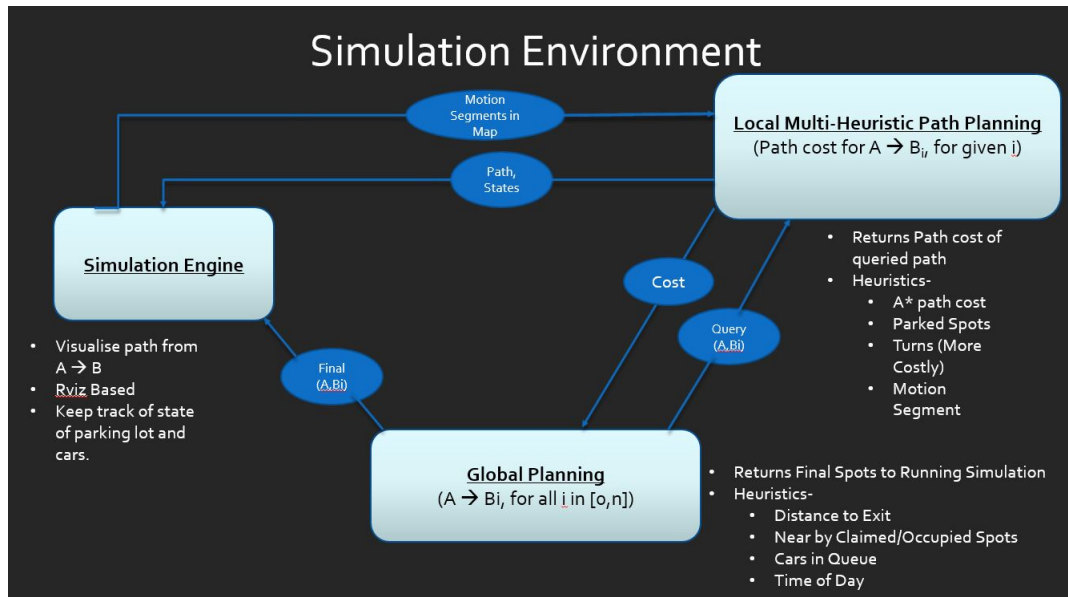


Figure 5: Simulation Engine

Redefining the scope of the project and coming up with a plan to achieve it was a challenge in itself and involved me working closely with Pranav and Mohak to come up with an agreed-upon scope.

3 Teamwork

As per the tasks for this week, I worked with Pranav and Mohak to create the map of the environment and test the locomotion of the platform. I worked with Dorothy to give her the exact requirements of the mock parking lot that she was building. Mohak, Pranav and I came up with the entire architecture and planning heuristics for the simulation environment. Dorothy, Richa and Mohak worked on integrating the planner with the communication system. Later Pranav too integrated the visualization tool with the communication system. Richa also started working on installing the OS on the new Intel SBC.

4 Plans

The team plans to integrate the communication system, planning system and the visualisation tool to the actual platform and start navigation in the mock parking lot based on that. Pranav would be working on writing a master script that integrates all the subsystems. Pranav, Mohak and I would be working on the simulation environment. I would be working on the global planner while

Mohak would be working on the path planner. Pranav would be working on the simulation engine. Dorothy and Richa would be working on getting the other platform ready from the software and the hardware side.