Progress Review 10

Individual lab report – 09 || March 16, 2016

Team Daedalus

Pranav Maheshwari

Team Members:

Mohak Bhardwaj

Dorothy Kirlew

Richa Varma

Shivam Gautam

1. Individual Progress

- ❖ Navigation in mock parking lot
- ❖ Building the Simulation Environment
- ❖ Integration of Subsystems
- ❖ Reinforcement of parking lot

Navigation Subsystem

- **Mapping**

For mapping, my task was to use the mock parking lot and finalize some specifications for our final parking lot such that it supports our perception subsystem and provides the platform an ideal environment to navigate in. The test setup can be seen in Figure 1.

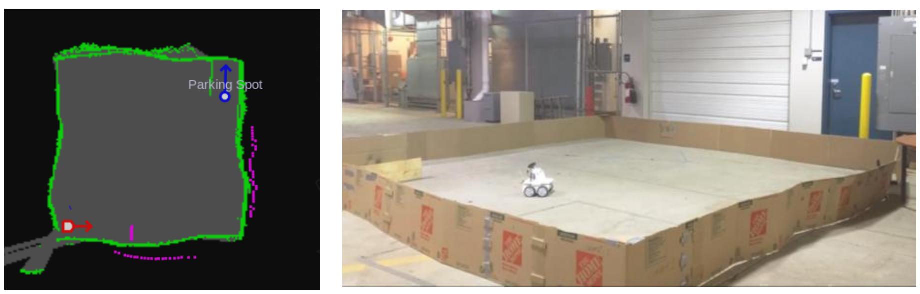

Figure 1: The map of the mock parking lot and the physical setup itself.

Shivam and I used the mock parking lot made by Dorothy to test the platform. Through our initial tests we could ascertain that the height of the walls were adequate and the area itself was appropriately sized for the platform to localize itself and traverse in. We also tested the obstacle avoidance capabilities of our platform and, as seen in Figure 2, it was quite comfortably able to plan its way around the obstacle.

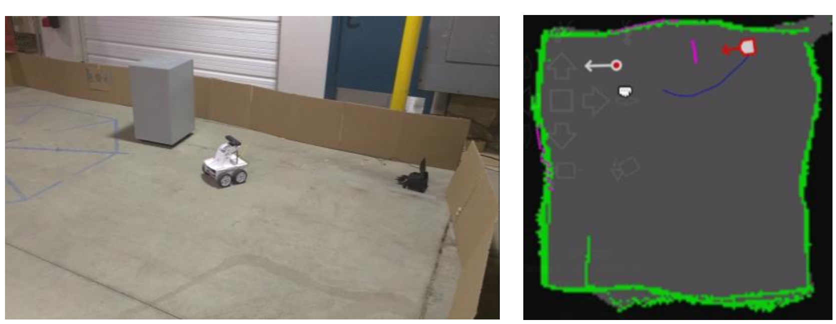

Figure 2: Blue path shows the trajectory calculated to avoid the obstacle

- **Waypoint Navigation**

  After running some tests and creating a temporary setup for the parking lot, I tested my navigation script, which is responsible for accepting waypoints from the multi-agent planner and transmitting them to the ROS Navigation Stack. This script accepts data over a ROS topic and then uses the TelNet API to transmit a waypoint to the platform. Waypoints, which in our case will be parking spots, can be added either through the web interface or the TelNet API itself. Once the testing of this script was complete, I was in a position to further integrate it with the rest of our architecture, more about which has been covered in the Integration section of this report.

Simulation Environment

I've been working on a ROS node which acts as the rendering engine, which publishes data to RViz, and a backend script which creates random data for the simulation to run, such as the arrival time and return time of a vehicle.

- **Environment Structure:**

  Based on the parameters decided by Shivam, Mohak and I, an environment file had to be created which could be accessed by everyone for their planner and rendering related tasks. I used OpenCV to create an image of a parking lot, as seen in Figure 3, which can be accessed as a 2D array by others. Each pixel in this image corresponds to a 2.5cm x 2.5cm patch in the environment. The parking spots have a dimension of 3.5m X 2.5m and the lane width is 5m.

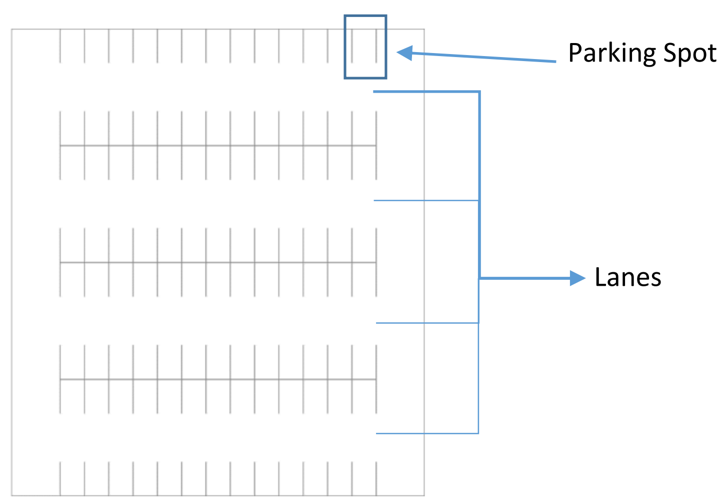

Figure 3: Map of the parking lot

- **Rendering Engine:**

  This publishes various markers to RViz which denote the position, path and destination of a vehicle. It also keeps a track of the active state of the parking lot and relays this information to the planners through a ROS service.

  The environment in RViz, as seen in Figure 4, will have a map of the parking lot, on which all the other markers will be rendered. The marker, acting as a vehicle, will have a continuous motion from its destination to the origin. The path that the vehicle is following will also be drawn for clarity regarding the state of the parking lot.

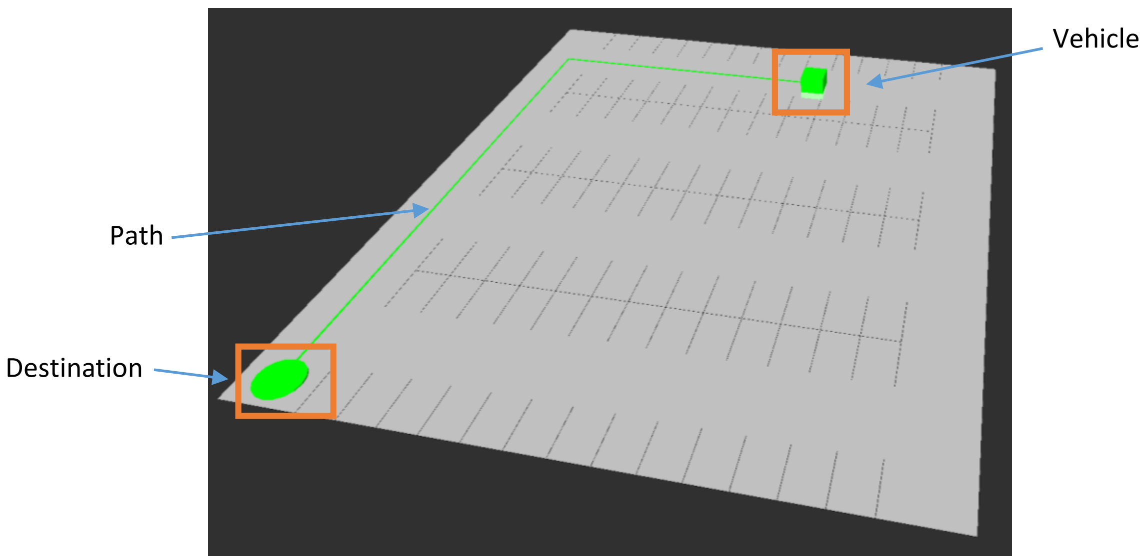

Figure 4: A vehicle travelling towards its destination along a path

  There are two services running on this node which populate all the data that the planners, local and global, need to calculate the cost of various paths and parking spots. This data includes the current position of the vehicles in the parking lot, free spots, and vehicle density across the parking lot, etc.

  In the coming week, we'll be setting up the ".srv" files needed for these services to run which will define the data being sent and received during these service requests.

- **Backend Script:**

  This script creates a random distribution of vehicles and assigns a time of arrival and departure to each vehicle. These times are calculated by using modified Rayleigh and Chi-squared distribution for arrival and return respectively. The resulting distribution can be seen in Figure 5. As the time progresses, this script issues service requests to the planners to get an optimal path and destination, which it then further relays to the rendering engine.

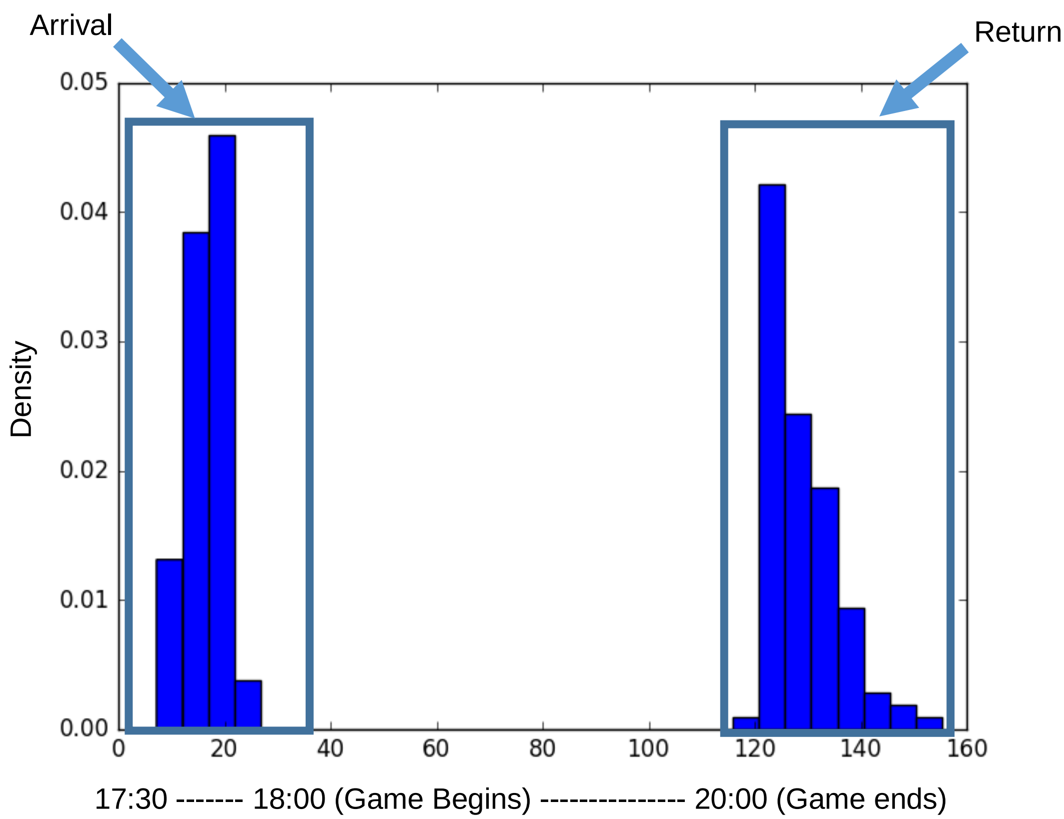

Figure 5: Distribution of Arrival and Return times of Vehicles

This distribution is adequate for our Use Case, that of a Pirates' baseball game. If the game is from 18:00 to 20:00, we expect people to show up in advance and leave after the game ends with different types of people taking varying amounts of time. For example a group of youngsters will take much shorter time than a family with kids to pack up and leave once the game is over.

Integration

The task of integration was carried out in incremental steps. Firstly, the entire pipeline, as seen in Figure 6, was integrated. This ensured that everything independent of the navigation of the platform is in working order.

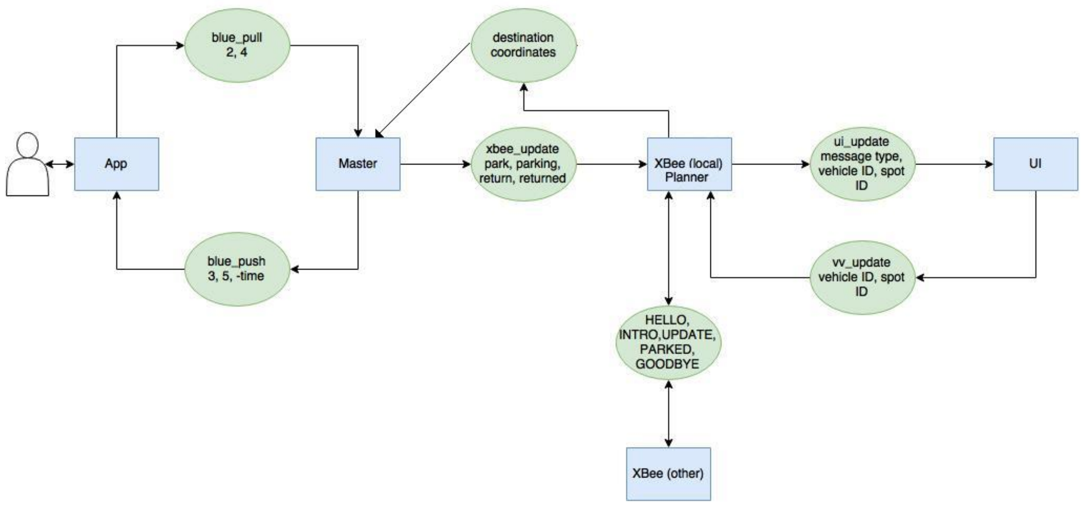

Figure 6: ROS Communication Architecture (Graphic created by Dorothy)

Richa and I worked on the integration of all these different subsystems. To ease the process of integration, we used GitHub as it allows us to make quick changes and effectively collaborate with each other. We collected all the necessary files and set them up in the ROS environment. I also created a launch file which accepts multiple arguments, thereby allowing us to selectively launch and test nodes from our subsystems. We tested the Bluetooth connection between the platform and the Android phone to ensure that the mobile app is compatible with our setup. To test the standalone functionality of XBee, we transmitted some serial data back and forth between two XBees which ensured that our platform had all the required packages for our Communication subsystem to work. Finally, after the environment was setup, we tested the integration of these subsystems which went quite well.

The final outcome of this was that we could start a routine through our mobile app and make platforms collaborate with each other. After ensuring the robust performance of this routine, I further integrated the navigation node with this setup, which completed the entire architecture of our system. This setup was then tested on the modified parking lot.

Parking Lot

After verifying that the current parking lot is of adequate dimensions in all respects, I decided to reinforce the walls as they were pretty flimsy in their present state. This flimsiness results in uncertain configuration every time the parking lot is propped up. To set some fixed corners and keep the walls upright, Mohak and I created wooden frames to provide support by reinforcing the corner joints. The exploded view of our CAD can be seen in Figure 7. We used scrap material and tools from the woodshop for fabrication.

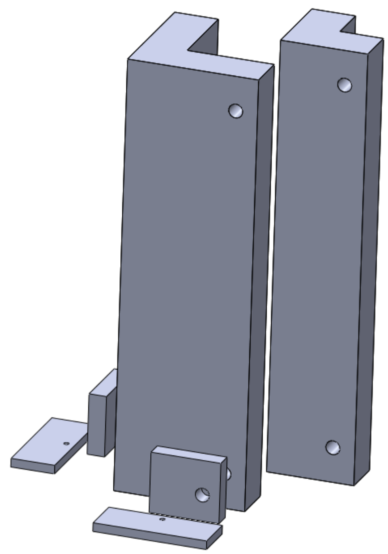

Figure 7: Exploded view of corner joints' support

This setup consists of two L sections sandwiching the walls between them and keeping them up-right. The L sections are further attached to a block of wood for further support and adding some weight. We used a butt joint and glue to create these L sections. Our initial testing with these wooden supports didn't yield expected results and we had to come up with a different support mechanism.

To simplify the problem, we decided to fix the corners of the parking lot and keep our long cardboard strip stretched. This tension keeps the cardboard in a straight line. To fix the corners, we took 3 heavy boxes filled with scrap material and a fixed pole as our anchor points. This setup had much better results and we believe is good enough for our platform to navigate and localize in. The modified state of the parking lot can be seen in Figure 8.

Figure 8: Current parking lot (Collaborators: Pranav and Mohak)

2. Challenges

The main challenge during this work was to figure out the proper specifications for our parking lot. Since making and modifying the mock parking lot takes considerable amount of effort, it is important that we conduct extensive testing before finalizing the specifications. This required a substantial amount of time. Finally, we decided that our mock parking lot, in its current configuration works pretty well and all that needs to be done is to make a more stable fixture with similar dimensions. This will now be fabricated using wood.

For the simulator, since we have three people actively working on a common environment, we had to go through multiple iterations before we could decide an environment structure which satisfied everyone's requirements. The architecture of the simulator is also something which required a lot of back and forth since there are multiple ways and paths to carry out internal communication. This required significant deliberation before we could lock down on the combination of ROS service/client and publisher/subscriber present in our simulator.

Apart from this, since we are nearing system integration now, there are a lot of inter-dependencies in everyone's work leading to multiple bottlenecks which reduces the team's efficiency. Also, the documentation we prepared earlier, to assist during integration, had errors which wasted some of our time during debugging. We aim and plan to have better communication and collaboration in the coming weeks to alleviate this issue.

I also faced some issues in starting the Navigation Stack on the platform which I believe was due to inadequate power supply. We've ordered new batteries to fix that.

3. Teamwork

I've been closely working and collaborating with Mohak and Shivam for all my Simulator related tasks. Together, we created the environment structure and the overall architecture of our Simulator. Shivam and I also worked together on the Mapping and Navigation aspects of the platform. We setup and mapped the mock parking lot to test the localization and navigation of the platform. Richa and I worked together on the integration of various subsystems. Richa also worked on fixing the old platform and placed orders to acquire all the missing components. Mohak and Shivam worked on the multi-heuristic planners for the Simulator. They've managed to achieve partial functionality in their subsystems and have prepared a skeleton structure of their code. Mohak and I also collaborated for the reinforcement of the parking lot. Together, we designed, fabricated and assembled all the parts needed to reinforce the parking lot. Dorothy and Richa merged and tested the planner and XBee nodes. Dorothy also created revised communication architecture, which shall be helpful for future reference.

4. Plans

For all our future tasks, we'll be following the critical schedule. We need to have a beta version of our Simulator ready by the end of this week. This would be a working version with some heuristics implemented on the planners and the vehicles being rendered in the environment. I shall be working on the integration and testing of the simulator. Once all the integration is done, we'll be spending rest of our time in testing and fixing along the way. We expect the simulator to go under multiple iterations before it's fully done.

Dorothy and Richa would be working on various aspects of our physical system. They'll fabricate a wooden parking lot for our final environment. The second platform, as and when it arrives, will also require some prepping before it is ready for testing. Once that is done, we'll be in a position to have an end to end run of our Demo 1 for the Spring Validation Experiments.