Progress Review 7

Individual lab report – 06 || January 27, 2016

Team Daedalus

Pranav Maheshwari

Team Members:

Mohak Bhardwaj

Dorothy Kirlew

Richa Varma

Shivam Gautam

1. Individual Progress

- Literature survey for implementing waypoint navigation on the platform
- Integration and testing of locomotion with emergency stop
- Development of Visualization Tool

Navigation Subsystem

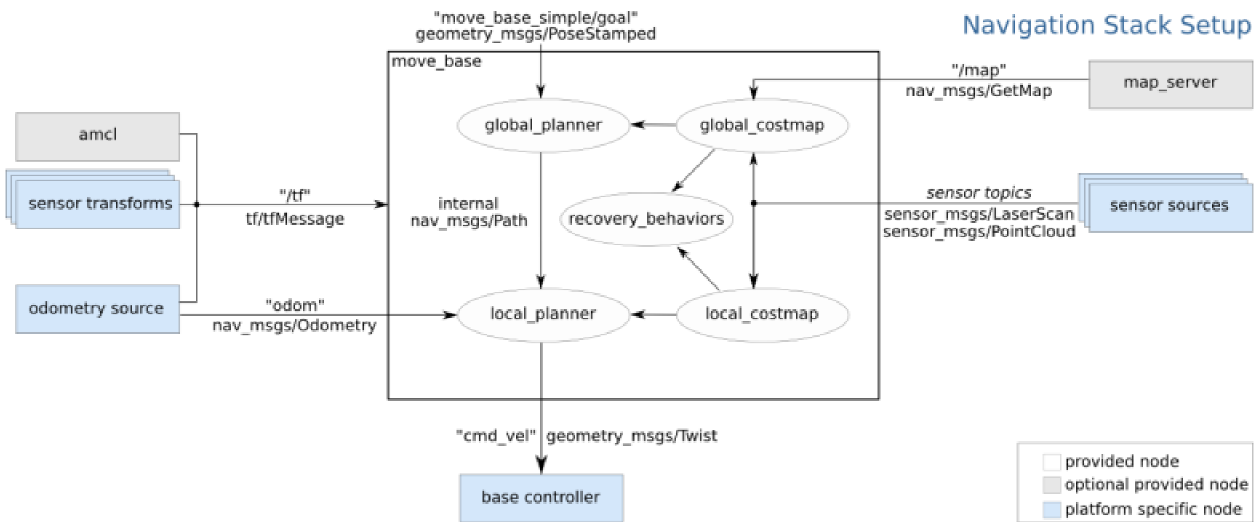

Figure 1: Architecture of ROS navigation stack.
Reference: http://wiki.ros.org/navigation/Tutorials/RobotSetup

As can be seen in Figure 1, the ROS Navigation stack is capable of accepting sensor data, a pre-generated map, odometry data, etc. and process it all via its global and local planner to generate velocity commands which are then used to drive the actuators. Adaptive Monte Carlo Localization (AMCL) is used to perform localization of the robot against a known map.

For path planning there are two cost maps which are used. One is the global cost map used to generate long-term plans over the entire map and other is local cost map used for short term plans and obstacle avoidance.

The twist messages published by the navigation stack can't be directly executed by our platform since it is capable of only holonomic turns. To take care of this, a local ROS node running on the platform processes these commands and breaks them down in to segmented arc moves (by performing turn-in-place and linear moves sequentially).

To integrate the ROS Navigation Stack with the rest of our system, I'll work on a script that can accept goal positions and then track the platform till it reaches there. It will also publish a message back to the mobile app once the platform reaches its destination.

Emergency stop implementation

To help with testing and debugging of the obstacle detection/ emergency stop node, I created a python script which makes the platform stop and resume motion depending upon the present state of the system. These emergency messages are being published based on the data being received from the range sensors. These messages are received by the master node which then further gives "stop" or "resume" commands to the locomotion node. The flow of logic can be seen in Figure 2. This is an important feature as it ensures that our platform won't collide with objects and stop at a safe distance.

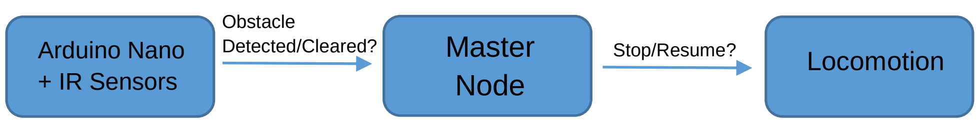

Figure 2: Flow diagram for Emergency Stop

Visualization Tool

The visualization tool created for Fall Validation Experiment needs some further work so that it is capable enough to show the active status of the parking lot. This will help us during integration and testing as we'll be able to actively track the status of the system.

Currently, as can been seen in Figure 3, the visualization tool uses a template based on the mock parking lot and then overlays images on different layers to show the origin, goal and state of the platform. I worked on creating visual elements which will be used for the interface and also created a routine which will be followed to update the status of the parking lot.

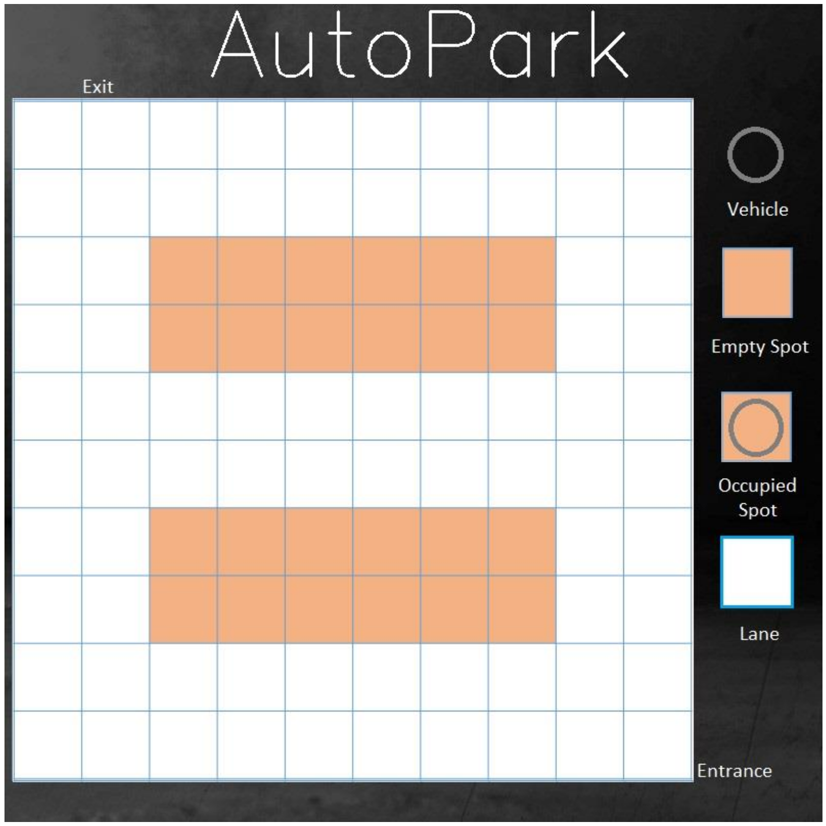

Figure 3: Basic layout of the visualization tool

2. Challenges

The main challenge in the literature survey was gathering information from different sources and integrating it all. Since we need to tweak the ROS Navigation Stack as per our application, it is important to have a good low level understanding of it.

There are a lot of interdependencies in everyone's work now. This makes it important to have a clear understanding of all the different data structures you're dealing with and the overall routine of the software architecture. As the Technical Leader of the project, I plan on streamlining this so that we face fewer issues during integration. The challenge with this lies in the fact that I'll have to be involved with every work package to a certain extent.

Also, based on our Fall Semester experience, we have allocated a lot of time for integration, thereby setting deadlines which would require a lot of work to be done very soon. This creates a lot of challenges in setting up schedules and delivering work on time.

3. Teamwork

As per the task distribution, I helped Dorothy with testing of her subsystem to ensure that it operated robustly. Shivam worked on redesigning the PCB to be used for our obstacle detection subsystem. Richa and Dorothy worked on the communication subsystem to ensure that we have three working XBees performing reliable bi-directional communication. Mohak has started the work of creating a map, which is a primary requirement of the ROS Navigation Stack to work effectively. He has also worked on creating the multi-agent planner which processes the current state of the parking lot to rank the empty spots.

4. Plans

As per the schedule, the next task I need to work on is implementing a waypoint navigation system on the platform, which accepts origin and destination targets as input and plans a trajectory accordingly. This work depends on the mapping and multi agent planning work being done my Mohak and Shivam. I'll also be working on the visualization tool used to show the active state of the parking lot. This work depends on receiving updated occupancy maps from the communication subsystem. Richa and Dorothy are working on the testing and integration of the communication subsystem, which will now use multiple XBees. Dorothy will also work on fabrication and assembly of the mock parking lot used for testing of the final integrated subsystem. We plan to wrap up our individual subsystems in the first half of February so that we have adequate time for integration and testing of the complete subsystem.