Progress Review 10

Individual lab report – 09 | March 16, 2016

TEAM DAEDALUS

Submitted By:

Mohak Bhardwaj

Team Members:
Dorothy Kirlew
Richa Varma
Shivam Gautam

Pranav Maheshwari

1. Individual Progress

The following responsibilities were taken up by me:

- Developing planning environment for simulation.
- Developing a preliminary version of local planner.
- Reinforced Parking Lot.
- Mapping and system testing in new parking lot.

PLANNING ENVIRONMENT

In order for the multiple planners in the simulation to work together seamlessly, it is necessary to have a common planning environment. This environment specifies the different parameters related to motion planning for the simulated vehicles. I worked together with Pranav and Shivam to come up with a map of the simulated parking lot and a corresponding environment file. The following are the different parameters of the planning environment:

1. **Robot Dimensions: 3m x 1.5m**
2. **Spot Dimensions: 3.5m x 2.5m**
3. **Number Of Spots: 104**
4. **Lane width: 5m**
5. **Environment Dimensions: 48m x 42.5m**
6. **Cell Resolution: 0.025m x 0.025m**
7. **Start Location(x,y): 2.5m, 2m**
8. **Exit Location(meters,x,y) 40m, 40m**

In addition to the above parameters, an additional file describes the coordinates of the parking spots and the orientation of the robot in that parking spot. A map image of the parking lot created from the above specifications was then parsed to get the pixel values of each cell (0 being free and 255 being an obstacle) which were stored in a text file. This file will act as a cost map or the planning environment. The costmap will be inflated near the obstacles according to the robot dimensions to ensure that the robot does not skim too close to the obstacles and collide eventually. An important parameter in the above environment specification is the cell resolution. The cell resolution has to be chosen very carefully to work with the motion model.

PRELIMINARY LOCAL PLANNER

The local planner is responsible for finding an obstacle free path to a parking spot and also return the path-cost of the most optimal path. As outlined in the simulation architecture, the global planner passes a list of spots ranked on the basis of different heuristics to the local planner. The local planner then evaluates each spot as a query and ranks the spots again on the basis of heuristics that take into account the actual cost of travelling to the spots. The most optimal spot is then calculated and the path to it is passed onto the visualizer as a list of nodes.

I am implementing an A* lattice based planner for planning in continuous state space of the robot i.e x, y and heading. The planner uses a lattice implementation consisting of rasterized poses to prevent the number of states in the open list from blowing up. This is a very critical step in the planner since the planning is being done in a continuous configuration space and not a discretized environment. A bicycle motion model is being used for the cars with six different motion primitives consisting of steering and distance control inputs. Using these primitives continuous car-like motions can be generated.

As a preliminary implementation, I used an open-source simulator for implementing motion planning algorithms using Python. In this simulator, obstacles can be added to the environment in different locations and paths can be visualized based on the algorithm implemented. I implemented the planner and bicycle motion model using just the distance to goal heuristic (in addition to cost of path from the start) and the results of different obstacle configurations can be seen in **Fig 1.** The red cells characterize the obstacles and the arrows signify the start and end pose of the robot. The green colored cells signify the raster cells that have been visited while planning a path from the start to end pose.

The planner performs sufficiently well but in order to obtain more real-time performance with suitable refresh rates, I am implementing the planner in C++. Currently, the data structures for nodes, poses and environment along with routines for environment loading, helper functions for finding euclidean distances, raster formation etc. have been successfully implemented and tested.

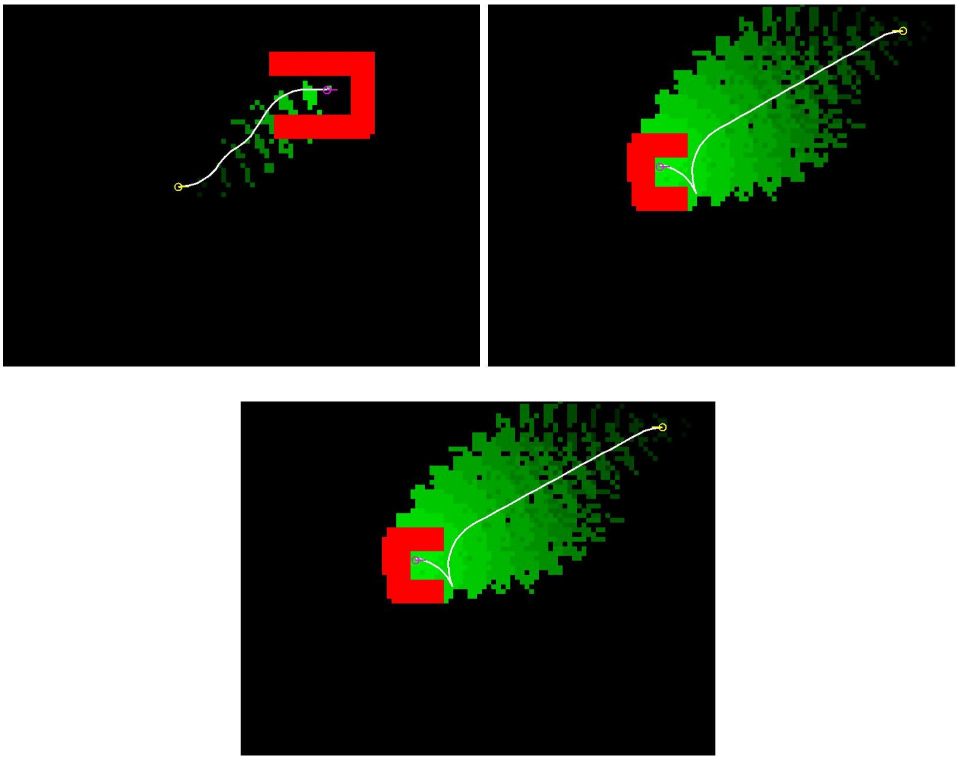

Fig 1: Preliminary results of A* lattice planner with forward and backward motion primitives

PARKING LOT

The current parking lot was insufficient for our purposes because of the flimsiness of the cardboard used to construct it. Due to this it was difficult to rearrange the parking lot in the same configuration every time which caused problems with localization and the parking lot had to mapped every time. Pranav and I worked on reinforcing the existing parking lot by wooden frames to provide support at the corners. Figure 2 shows a CAD model of the support frames.

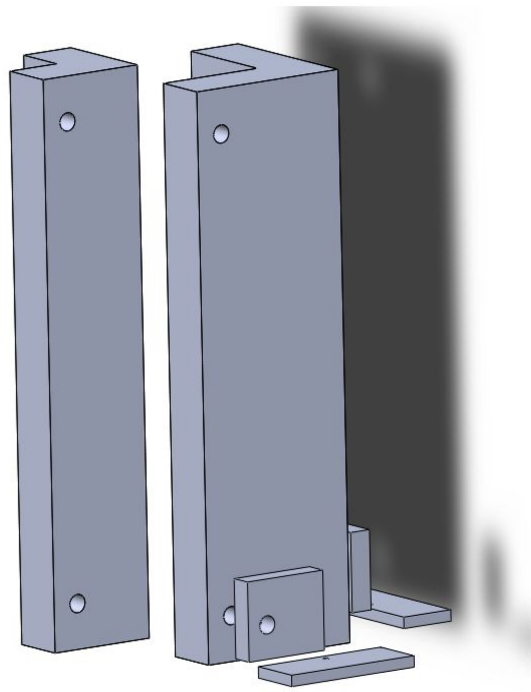

Fig 2: CAD Model of support frames for reinforcing current parking lot

The L-sections clamp the cardboard between them and can be tightened using screws. A butt-joint was created between the wood cuts using adhesive and additional pieces were joined to the bottom for extra support and stability. However, the wooden clamps proved to be insufficient and the structure was still not fully stable.

To mitigate this problem, we decided to keep the cardboard stretched by providing tension at the corners. We fixed all of the corners of the parking lot using a pole and four heavy boxes to keep the cardboard taut. These act as anchors and the positions have been marked so that the same configuration can be recreated every time. Figure 3 shows the current parking lot.

Figure 3: Current status of parking lot

(Collaborators: Pranav and Mohak)

MAPPING AND SYSTEM TESTING IN NEW PARKING LOT

Once the reinforced parking lot was created and the pipeline from the app to navigation was integrated Pranav, Shivam and I worked on mapping the new parking lot and testing the entire system. We specified different waypoints and tested the repeatability of the waypoint navigation through a park command sent through the app. In addition, the obstacle avoidance was tested by placing different obstacles in the path of the robot. The robot successfully planned path across different obstacles.

2. Challenges

1. Reinforcing the current parking lot was time consuming and required multiple iterations to get right. Lack of a permanent place for the parking lot is still a big risk moving forward.
2. Deciding on a common environment that satisfies the requirements for the different planners and visualization required deliberation and multiple iterations.
3. Deciding the most optimal data structures for the planners was difficult for me as I do not have extensive prior experience with data structures.

3. Teamwork

The other members of my team were working on the following subsystems:

1. Shivam: He worked with Pranav on mapping and navigation aspects of the platform and with both me and Pranav on the simulation. He has developed partial functionality for the global planner using SBPL and tested different planning algorithms like ARA* and MHA*.
2. Pranav: He worked with Shivam and me on the devloping the environment for the simulator and on the visualization aspect. He also worked with me on reinforcement of the parking lot and with Shivam on mapping and navigation. Later, the worked with Richa on integration of the visualization tool with communications. He also worked on the integration of the pipeline from the app to navigation and later with me on the mapping and testing in the new parking lot.
3. Dorothy: She worked with Richa on improving and testing the communication code and hardware for the old platform. She also worked on integration of planner into UI
4. Richa: She worked with Dorothy on improving and testing the communication code, with Pranav on integrating the visualization tool with communication and ordering parts for the integration of the old platform.

4. Plans

As per our critical path we will be working on completing the first version of our simulator by the end of this week. Pranav, Shivam and I will be working on completing our individual subsystems and starting integration with a basic implementation ready. Once our basic integration within a ROS framework is done and different services have been implemented, we will iteratively refine our planners and work toward a final finished simulator. Dorothy will be working on constructing a final parking lot and along with Richa on completing the integration of the old platform followed by testing of the system for robustness.