Individual Lab Report

Shiyu Dong
Team B – Auto Pirates
Teammates – Tushar Chugh, Bikramjot Hanzra, Tae-Hyung Kim, William Seto
ILR06

January 28, 2016

1 Individual Progress

We mainly focused on the integration of our project for this progress review. We originally wrote all the functionality in separate nodes, hence it takes much time to run all the code one by one. For this progress review, I mainly wrote the launch file to start all the nodes in the same time and also take in all the input arguments and constant numbers to avoid confusion.

```
<launch>


  <node pkg="rosbag" type="play" name="rosbag" output="screen" args="$(find radar_stuff)
  /rosbag/3river.bag -l" launch-prefix="xterm -e"/>

  <node name="markerobs" pkg="int_markers" type="menu.py" args="5" launch-prefix="xterm -e"/>


  <rosparam subst_value="true">
    /map_ref_x: 577336.797385
    /map_ref_y: 4484193.218161
    /map_scale_x: 4.997183
    /map_scale_y: 4.997080
    /map_pix_height: 2455

    /map_bridges: "$(find map)/bigger_bin.bmp"
    /map_nobridges: "$(find map)/bigger_bin_nobdg.bmp"
  </rosparam>

  <node name="rviz" pkg="rviz" type="rviz" args="$(find launch)/launch/rvizconfig.rviz" launch-
  prefix="xterm -e"/>

  <node name="map_server" pkg="map_server" type="map_server" args="$(find map)/map.yaml" />

  <node name="radar_stuff" pkg="radar_stuff" type="radar_rviz.py" output="screen" launch-
  prefix="xterm -e"/>

  <node name="pptest" pkg="pptest" type="pptest" args="$(find map)/map1.cfg $(find map)
  /outputnobridges.mprim" output="screen" launch-prefix="xterm -e"/>


</launch>

```

Figure 1: Launch File

According to the launch file shown in figure 1, we can see that 6 ROS nodes are started when we run the launch file. They are:

- rosbag: play the rosbag file we recorded in field test.
- rviz: takes in the rviz config file, and run rviz for visualization.
- map_server: run the map server to show the occupancy grid map in rviz.
- radar_stuff.py: the node to filter radar data, segment bridges and detect obstacles.
- pptest: the node to take in the occupancy grid map and the motion primitive file, and do the path planning.

- markerobs: create fake obstacles using interactive markers to test the path planning functionality.

Also, we've kept all the constant value in the launch file. These parameters are mainly for the map we're using, including map size, corresponding UTM coordinate, and grid size of each pixel for the map. William and I have tried to figure out the pixel size when we exported the map from QGIS. Although we now still havn't figured out how to get the accurate pixel size, we have made the pixel size as close as to 5 meters/pixel.

2 Challenges

The launch file is common way for big ROS project to start all the nodes. There are some tricks I learned when trying to write the launch file.

The first thing is the way to put in an input argument. The simplest way to do this is to give the path for the file we want:

```
args = "<path_for_the_file>"
```

But the path for the input files may be different in other team member's computers. Therefore a better way to do this is put the file in one ROS package, and search for the package to find the corresponding file. For example, the map server takes the *map.yaml* file as an input, so we made a dummy package *map* to store all the map files. By using the following roslaunch command, ROS is able to find the file in the map package.

```
<node name = "map_server" pkg = "map_server" type = "map_server" args = "$(find map)
    /map.yaml" />
```

Another issue is that all the node will be started and shown in the same window at almost the same time when we run the launch file, so that it's hard to read the output on screen. I solved this problem by using the following command:

```
launch-prefix="xtern -e"
```

This will start the corresponding node start in the new window. And then by using the following command:

```
output = "screen"
```

We are able to print on the screen.

3 Teamwork

- Bikramjot Hanzra: Bikram wrote the code to add interactive markers as fake obstacles to test path planning. We are able to change the size, location and number of fake obstacles in this way. This will be a good way to simulate and test when we are not able to do the field test.
- Tushar Chugh: Tushar worked in integration of path planning with radar data. He contributed a lot to improve the path planner, mostly increase the speed of replanning and make the path planning more stable.

- William Seto: William also worked a lot in the integration. He increased the speed of the planner by changing it from ARA* to AD*. He found the information of bridge pylons in QGIS so that we can add bridge pylons to our map. He also worked with Bikram in adding the interactive markers as fake obstacles to test the path planner.
- Tae-Hyung Kim: Tae-Hyung worked on the robot localization package in ROS and tried to use the navigation stack in simulation.

4 Future Plan

Since in last semester I mainly worked on the perception part of our project, I am not quite familiar with the path planning part. As we now focus on the integration, it's important for each team member to know how each part in our project works.

I am now trying to learn the path planning knowledge, including using SBPL library, update the costmap and editing motion primitives. So that I can contribute more in integration and increase the speed of our code.

Also we are planning to go on field test next Wednesday. I'll work with other team members to plan for the field test. We will test the path planning performance near shores (as we added cost near shores) and near bridge pylons (as we added pylons in the occupancy grid map). We will also test the replanning when we change destination or add fake obstacles.