Individual Lab Report

Shiyu Dong
Team B – Auto Pirates
Teammates – Tushar Chugh, Bikramjot Hanzra, Tae-Hyung Kim, William Seto
ILR08

February 25, 2016

1 Individual Progress

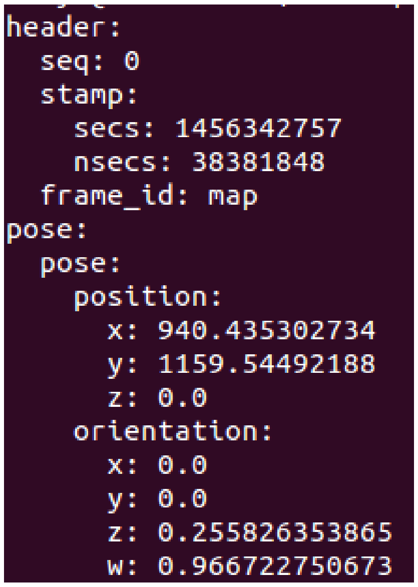
Because of the weather condition and time required to fix the boat engine, we were not able to go on a field test for this progress review. As a result, I mainly worked on improving the visualization and the simulator to test the path planner.

When we tested path planner before, we were playing the rosbag file in the same time and use the recorded GPS location as the input for the path planner. This is not good enough, because although we can test the replanning as boat moves, the boat is not moving according to the planned path. To improve the simulator to test path planner, we should get rid of the rosbag file and enable the boat to follow the planned path.

1.1 2D Pose Estimate in RViz

To achieve the goal mentioned before, firstly I need to specify the start point of the boat. In RViz, I can subscribe to the */initialpos* topic to get the position and orientation of the point we specified as the start point.

Figure 1 shows an example of the initialpos topic. We can get *x,y,z* as the position and *x,y,z,w* as the orientation (expressed in Quaternion).

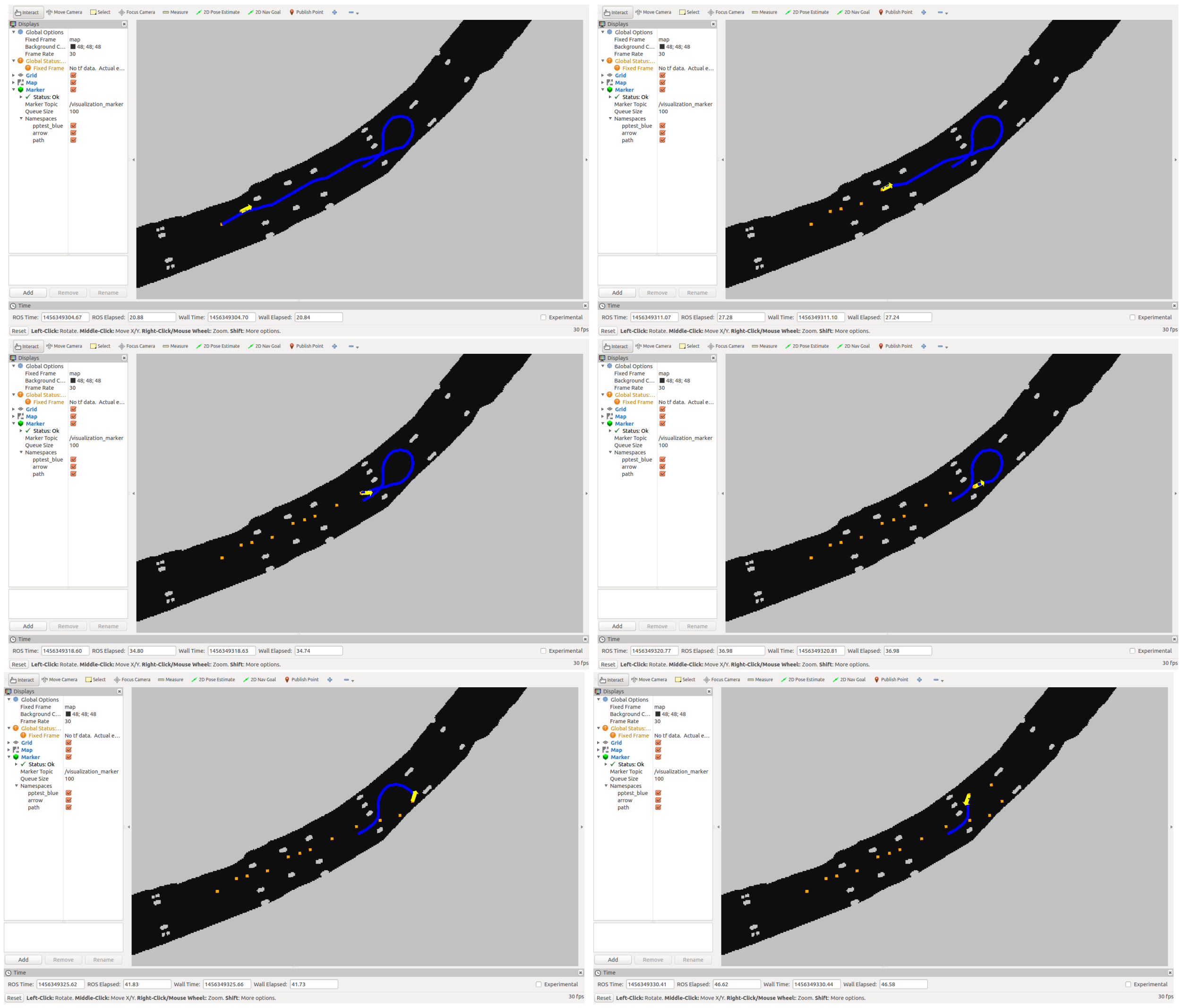


```
header:
  seq: 0
  stamp:
    secs: 1456342757
    nsecs: 38381848
  frame_id: map
pose:
  pose:
    position:
      x: 940.435302734
      y: 1159.54492188
      z: 0.0
    orientation:
      x: 0.0
      y: 0.0
      z: 0.255826353865
      w: 0.966722750673
```

Figure 1: rostopic echo /initialpos

1.2 Update the boat position

As we can also set the destination using **2D Pose Estimate** in RViz. We can now run the path planner to generate the path. To update the boat position, I was using the 5th way points from the planned path as the new boat position. To simulate the real condition better, I added some normal distribution noise to the boat position every time I replan the path.



Global Options
Fixed Frame: map
Background C…: 48; 48; 48
Frame Rate: 30
Global Status:…
Fixed Frame: No tf data. Actual e…
Grid
Map
Marker
Status: Ok
Marker Topic: /visualization_marker
Queue Size: 100
Namespaces
pptest_blue
arrow
path

ROS Time: 1456349304.67 ROS Elapsed: 20.88 Wall Time: 1456349304.70 Wall Elapsed: 20.84 Experimental
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options. 30 fps

ROS Time: 1456349311.07 ROS Elapsed: 27.28 Wall Time: 1456349311.10 Wall Elapsed: 27.24 Experimental
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options. 30 fps

ROS Time: 1456349318.60 ROS Elapsed: 34.80 Wall Time: 1456349318.63 Wall Elapsed: 34.74 Experimental
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options. 30 fps

ROS Time: 1456349320.77 ROS Elapsed: 36.98 Wall Time: 1456349320.81 Wall Elapsed: 36.98 Experimental
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options. 30 fps

ROS Time: 1456349325.62 ROS Elapsed: 41.83 Wall Time: 1456349325.66 Wall Elapsed: 41.73 Experimental
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options. 30 fps

ROS Time: 1456349330.41 ROS Elapsed: 46.62 Wall Time: 1456349330.44 Wall Elapsed: 46.58 Experimental
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel:: Zoom. Shift: More options. 30 fps

Figure 2: Updated Path

The improved simulator is shown as figure 2, as we can see the path is updated as the position change of the boat.

1.3 Visualization Arrow marker

As shown in figure 2, I also change the marker for the boat from point to an arrow, to better show the orientation of the boat.

For the arrow marker, we need to initialize *scale.x*, *scale.y*, *scale.z* as the length, width and height of the arrow, set *pose.position.x*, *pose.position.y*, *pose.position.z* as the position of the arrow, and set *pose.orientation.x*, *pose.orientation.y*, *pose.orientation.z*, *pose.orientation.w* as the orientation of the arrow (in Quaternion).

2 Challenges

The main challenge is the conversion between Euler angles (i.e, yaw, pitch, roll) and Quaternions (i.e, x, y, z, w), as SBPL uses Euler angles but Rviz uses Quaternions. It takes me some time to figure out the way to convert the units and Tushar helped me figure out by using the tf::Quaternion package.

2.1 Quaternion to Euler Angles

The following code converts Quaternion to Euler angles. We should initialize the Quaternion q, and then create a 3×3 matrix m as a function of q. Then we can use function getRPY to get the roll, pitch and yaw.

```
 tf::Quaternion q(msg.pose.pose.orientation.x,
msg.pose.pose.orientation.y,
msg.pose.pose.orientation.z,
msg.pose.pose.orientation.w);
 tf::Matrix3x3 m(q);
 double roll, pitch, yaw;
m.getRPY(roll, pitch, yaw);
```

2.2 Euler Angles to Quaternion

The following code converts Euler angles to Quaternion. We should initialize the 3×3 matrix m, and use function setRPY to set the roll, pitch and yaw of the boat. (In simulation we only want to see the yaw of the boat so we set roll and pitch to be zero.) The we can use function getRotation to get the Quaternion from m.

```
 tf::Matrix3x3 m;
m.setRPY(0.0, 0.0, yaw);
 tf::Quaternion q;
m.getRotation (q);
```

3 Teamwork

- Tae-Hyung worked on testing the GPS devices we bought. He used serial port to get GPS data and passed the data to ROS.
- Tushar worked with Tae-Hyung to using USB serial port to get GPS data and helped me in the conversion of Quaternion and Euler angles. He also updated the project website.
- William proposed and implemented a better radar filtering and mapping method – by using the OctoMap ROS package. It can remove the noise well and also increase the speed and efficiency.
- Bikram improved the simulator by adding teleoperated obstacles using keyboard events.

4 Future Plan

The OctoMap package we are currently using gives us a good filtering and mapping of radar data. We'll try to integrate it with path planning and test it in the field test.

The things we're going to test test in the field test:

- The replanning without deleting the environment.
- The integration of path planning and perception (using OctoMap package).
- Test the safety distance near shore and pylons and set a corresponding cost for it in the costmap.
- Test different motion primitives files to find out the one with best performance.

Since we have a lot of staff to test, we should spend more time to prepare for it. In the meantime, I'll work on the rules-of-the-road to find out what else we can do except for the middle line strategy.