Individual Lab Report

Bikramjot Singh Hanzra
Team B – Auto Pirates
Teammates – Tushar Chugh, Shiyu Dong, Tae-Hyung Kim, William Seto
ILR07

February 11, 2016

Contents

List of Figures

1 Individual Progress

During the past couple of weeks, I worked upon extending the functionality of the ROS package `int_markers` about which I discussed in detail in the last ILR. Just to give the some context, until the last Progress Review this package had functionality to add static obstacles to the environment. So as proposed in the work overview of the last ILR, I worked on adding dynamic obstacles to the environment.

2 Implementation

The implementation section is divided into 2 subsections –

- Simulating motion of obstacles
- Collision Detection

2.1 Simulating motion of obstacles

The motion of the dynamics obstacles are simulated in real time using Newtonian equation of motion for a body in a plane. An obstacle is instantiated by specifying an initial position (x_x, x_y) in the environment. The initial acceleration is sampled from an uniform distribution and the object starts to accelerate.

The updates in position (x_x, x_y), velocity $(\dot{x}_x, \dot{x}_y)$ and acceleration $(\ddot{x}_x, \ddot{x}_y)$ are given by the equations 1, 2, 3 and 4.

$$\dot{x}_x := \dot{x}_x * t + \frac{\ddot{x}_x t^2}{2} \tag{1}$$

$$\dot{x}_y := \dot{x}_y * t + \frac{\ddot{x}_y t^2}{2} \tag{2}$$

$$x_x := \dot{x}_x t \tag{3}$$

$$x_y := \dot{x}_y t \tag{4}$$

During the motion of the obstacles, it is constantly accelerated and deaccelerated based on samples from an uniform distribution. The maximum acceleration that an obstacle can reach is controlled using a threshold.

Figure 1 shows the log messages generated that show the position (x_x, x_y), velocity $(\dot{x}_x, \dot{x}_y)$ and acceleration $(\ddot{x}_x, \ddot{x}_y)$ of the obstacle. Note that in the figure, the position and velocity are changing at each time step.

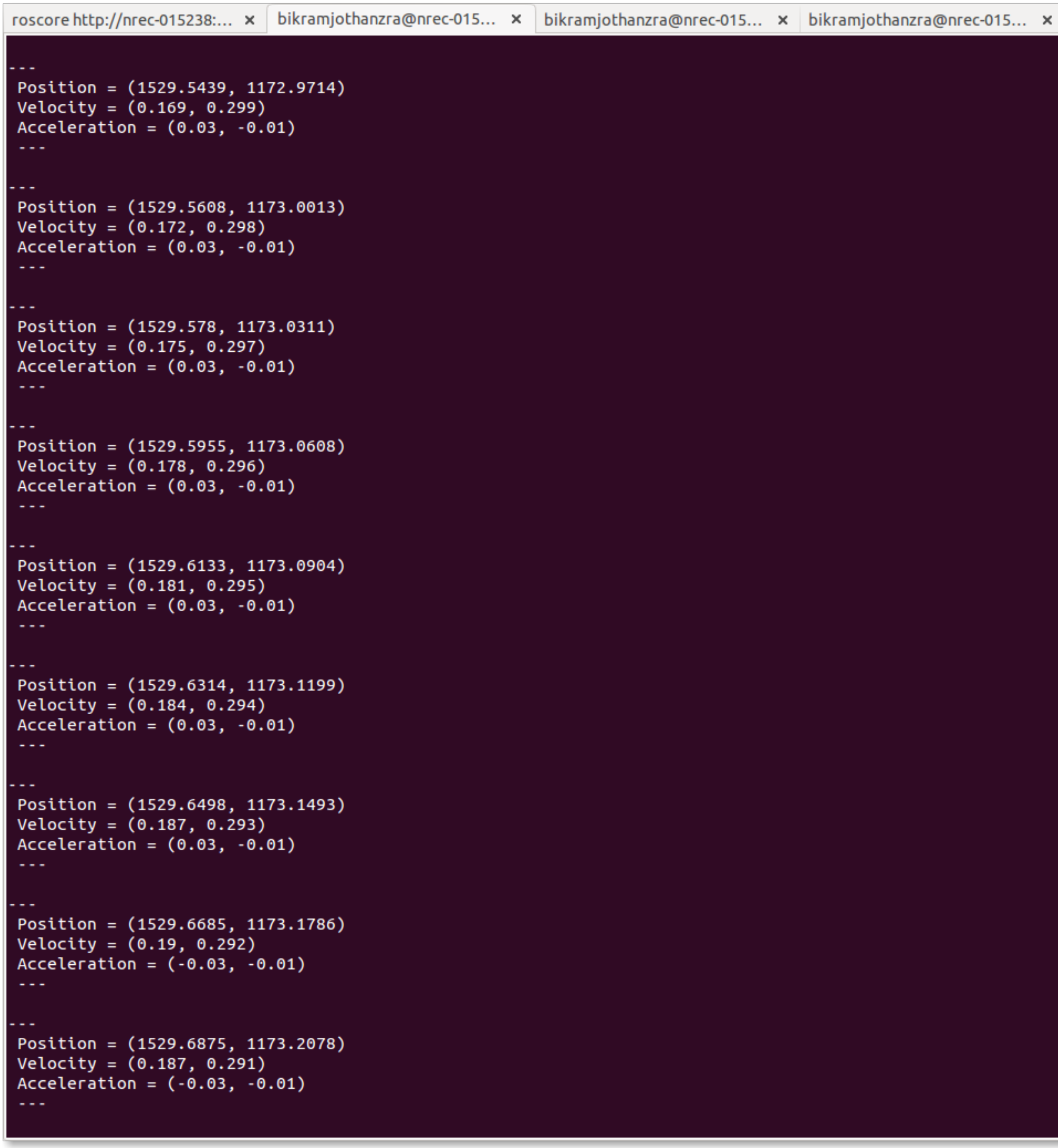
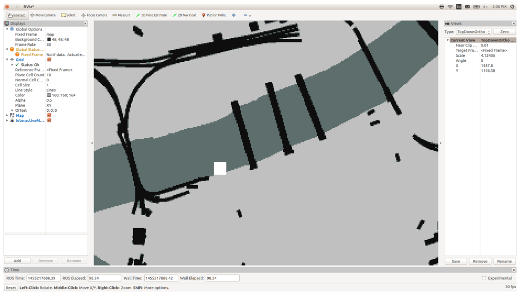
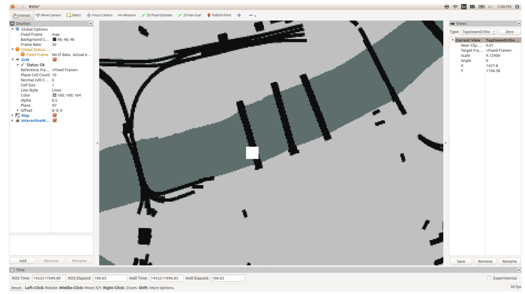
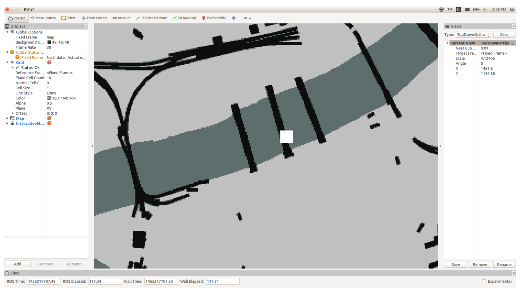
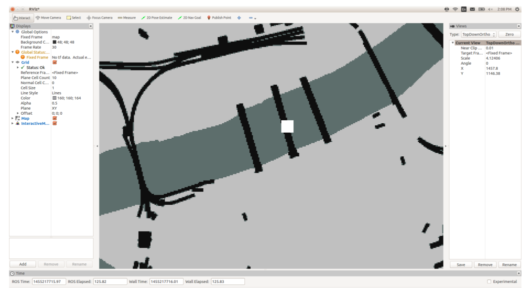


Figure 1: Log messages showing the Position, Velocity an Acceleration of the obstacle

Figure 2 shows the path taken by the dynamic obstacle at four time steps. First the obstacle moves across the river and then almost parallel to the bridge.

1. 2.

3. 4.

Figure 2: The figure shows (in clockwise direction) the path taken by the dynamic obstacle

2.2 Collision Detection

One important task while simulating the obstacles was that it should not go out of the river and it should not collide with the banks of the river. In order to implement this goal, I followed the ROS architecture to avoid collisions. I generated a binary grid map specifying where an obstacle is allowed to move. The value 1 was set for areas where the obstacles could go and 0 where it could not go. Whenever an obstacle is about to enter an area where it is not allowed to, it's direction is changed towards the unoccupied grids.

3 Challenges

Last time the major challenge, I faced was that the documentation for the `interactive_markers` [1] package on ROS website was pretty terse, but now I am feeling comfortable in using the package. So, most of the work was programming the dynamics of the obstacles.

One of the major challenge that I faced during debugging was that OpenCV and RViz have their coordinate frames defined differently. OpenCV just like MATLAB places the coordinate frame origin on the top-left of the image and RViz places the coordinate frame origin on the bottom-left of the image. I did not knew that RViz has such an architecture. I spent a lot of time debugging this bug.

The weather continues to hinder our field testing. We have not finalized the date for our next field test because the temperature is pretty low currently. This causes a lot of fog on the river and makes it tough for Mike (*Lead Test Engineer from NREC*) to drive the boat.

4 Teamwork

The work done by the rest of the team members is discuss below –

- **Tushar Chugh** – Tushar worked on improving the path planning subsystem. He worked on adding incremental updates to the path planner.
- **Shiyu Dong** – Shiyu worked on tuning the motion primitives of the boat. He also worked with William on tuning the hyperparameters of the binomial filter.
- **Tae-Hyung Kim** – Tae-Hyung started working on integrating the GPS (which he recently bought) with the path planner code he is working on. He also explored ways to follow the rules of the road.
- **William Seto** – William worked on manipulating the costmap to follow the rules of the road. He also worked in collaboration with Shiyu on exploring ways to filter the RADAR data.

5 Work Overview for Coming Week

In the next couple of week, I will be working on integrating the dynamic obstacles into the system. Currently, the motion of the dynamic obstacles is random. Real obstacles in a river will usually follow the rules of the road e.g. they will try to stay on the right side of the river.

I will be working on making the obstacles stay on the right side of the river where their aim will be to move from one end of the map to another. I will be formulating it as an optimization problem where there is penalty to go to the left side of the river.

References

[1] `interactive_markers` package documentation
`http://wiki.ros.org/interactive_markers`