Individual Lab Report

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

ILR06

January 28, 2016

Contents

List of Figures

1 Individual Progress

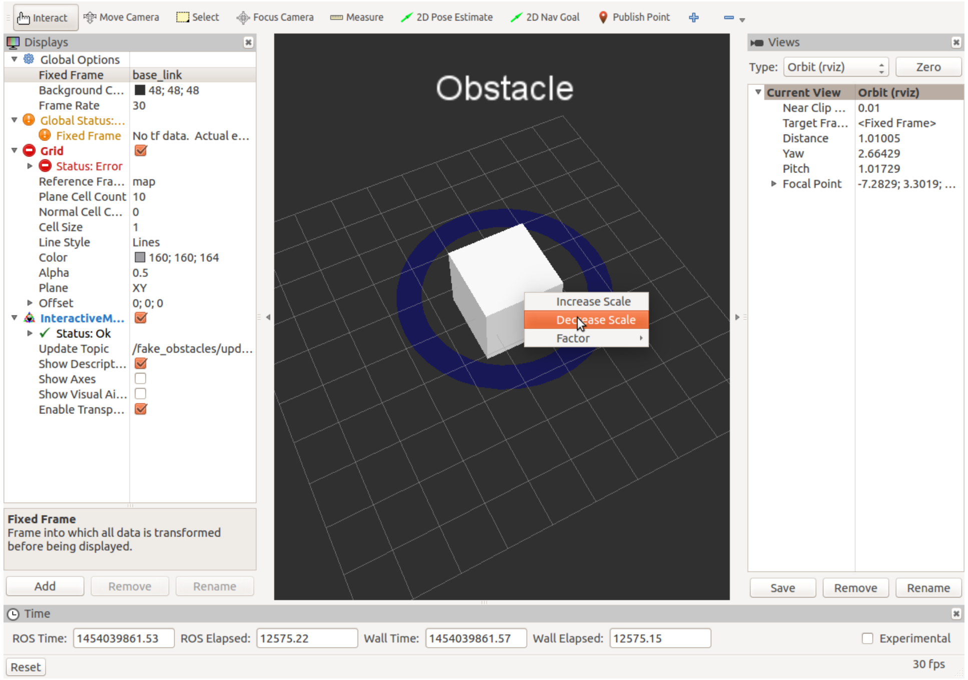
During the past couple of weeks, I worked upon writing a ROS package `int_markers` to interactively add obstacles to the environment. The motivation behind writing this package was to add fake obstacles in the path of the boat to test the path planner using the simulator and tune the motion primitives of the boat. Another reason for writing the package was to test the performance of the real boat in a cluttered environment during field testing. Due to the cold weather, the traffic on the rivers is pretty sparse these days. Also, due to safety considerations with other boats around, we were recommended by NREC engineers to use fake obstacles first. The source code of the package is available at `https://github.com/auto-pirates/recboat-ros/tree/master/src/int_markers`. William integrated this package into the system.

The package `int_markers` performs the following tasks –

- Add any number of obstacles in the map using RViz.
- Move the obstacles independently in the $x - y$ co-ordinates of the map by clicking on the markers attached with each obstacle.
- Increase or decrease the size of each obstacle independently using the context menu by clicking on the obstacles.

Figure 1 shows a single obstacle on a grid with a blue circle around it and a context menu. The context menu opens only when the obstacle is pressed while the markers are always visible. The context menu has four options namely –

- **Increase Scale** – To increase the size of obstacle
- **Decrease Scale** – To decrease the size of obstacle
- **Factor** – The factor by which the size is increased or decreased
  - **x**10 – Increase the size by a factor of 10.
  - **x**20 – Increase the size by a factor of 20.



Global Options	
Fixed Frame	base_link
Background C...	■ 48; 48; 48
Frame Rate	30
Global Status:...	
Fixed Frame	No tf data. Actual e...
Grid	☑
Status: Error	
Reference Fra...	map
Plane Cell Count	10
Normal Cell C...	0
Cell Size	1
Line Style	Lines
Color	■ 160; 160; 164
Alpha	0.5
Plane	XY
Offset	0; 0; 0
InteractiveM...	☑
✓ Status: Ok	
Update Topic	/fake_obstacles/upd...
Show Descript...	☑
Show Axes	□
Show Visual Ai...	□
Enable Transp...	☑

Fixed Frame
Frame into which all data is transformed before being displayed.

Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	1.01005
Yaw	2.66429
Pitch	1.01729
Focal Point	-7.2829; 3.3019; ...

ROS Time: 1454039861.53 ROS Elapsed: 12575.22 Wall Time: 1454039861.57 Wall Elapsed: 12575.15 □ Experimental

30 fps

Figure 1: Single Marker on a Grid

2 Implementation

I would first discuss the structure and architecture of the interactive markers in ROS. Interactive markers are very similar to the basic markers used in ROS with the additional capability to translate and rotate in the $3D$ environment interactively. As shown in Figure 2, a number of different controls can be attached with each obstacle. The cube, the markers and the context menu shown in Figure 1 are all examples of the various controls available for use with interactive markers.

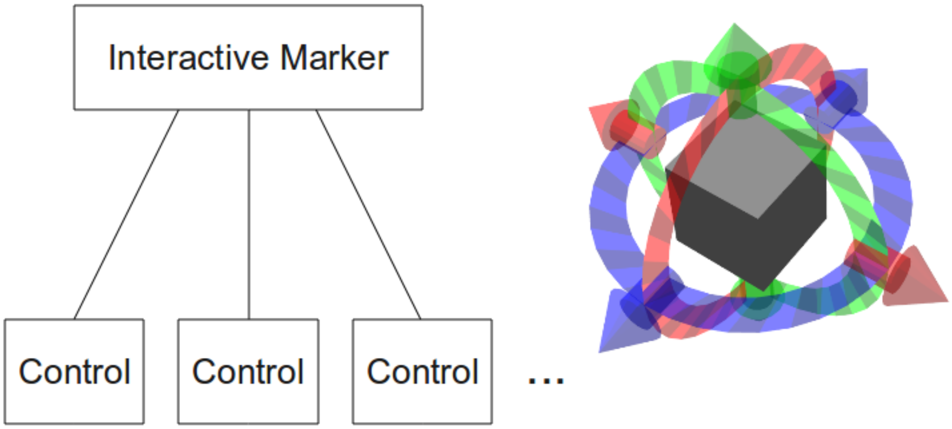


Figure 2: Interactive Markers Structure

In order to send and receive data from the RViz to the node that creates them, we need to create a server object. Figure 3 shows this interactions taking place.

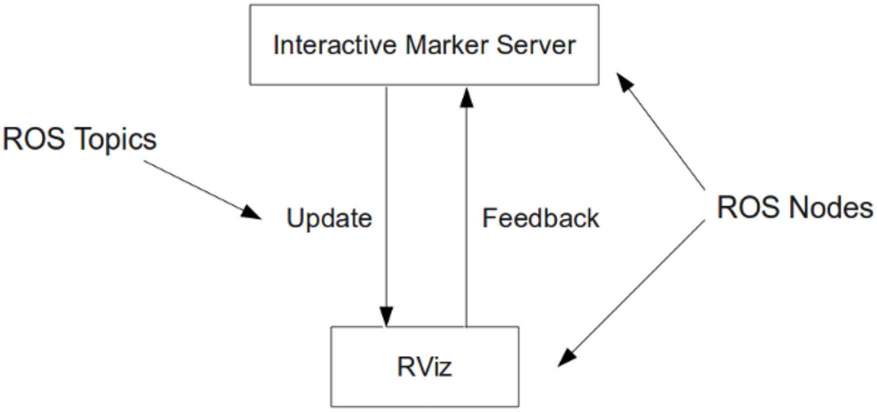
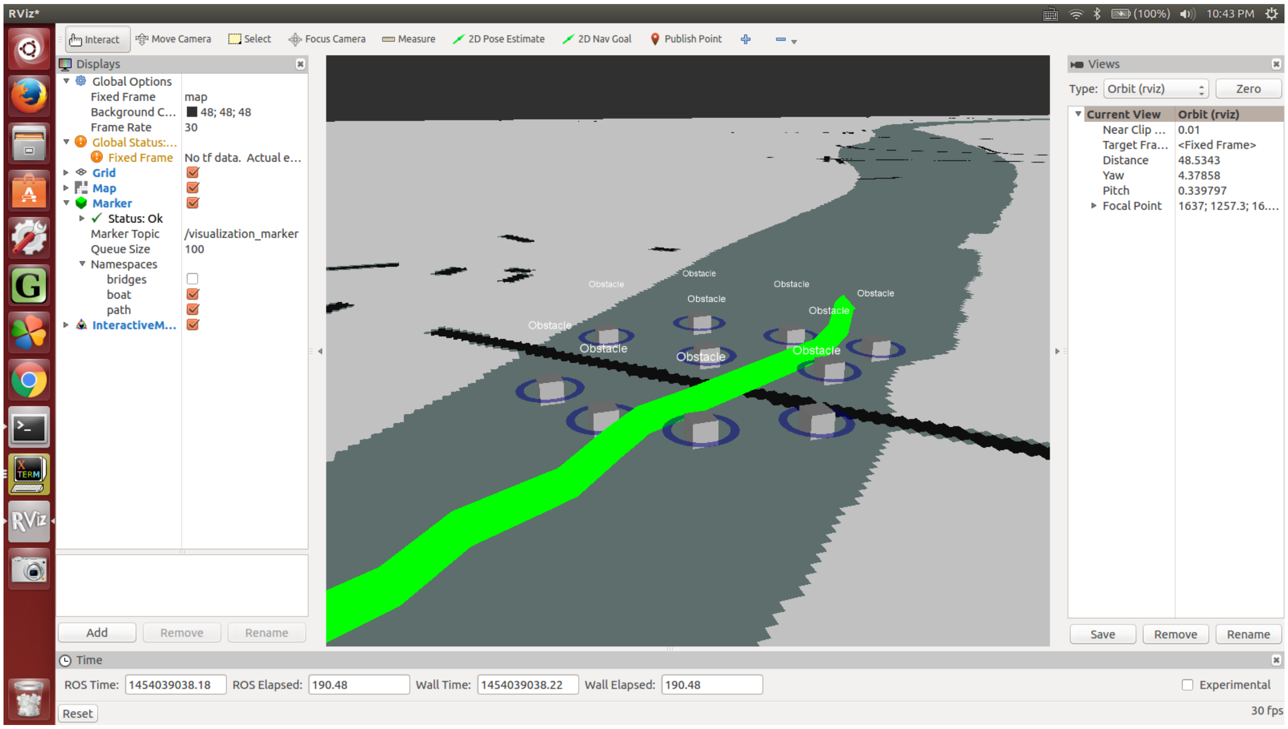


Figure 3: Interactive Marker Architecture

Figure 4 and 5 shows the path generated by the path planning algorithm for 2 different positions of the obstacles. There are a total of 10 obstacles. The green line shows the path generated by the path planner.



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	
bridges	□
boat	☑
path	☑
InteractiveM…	☑

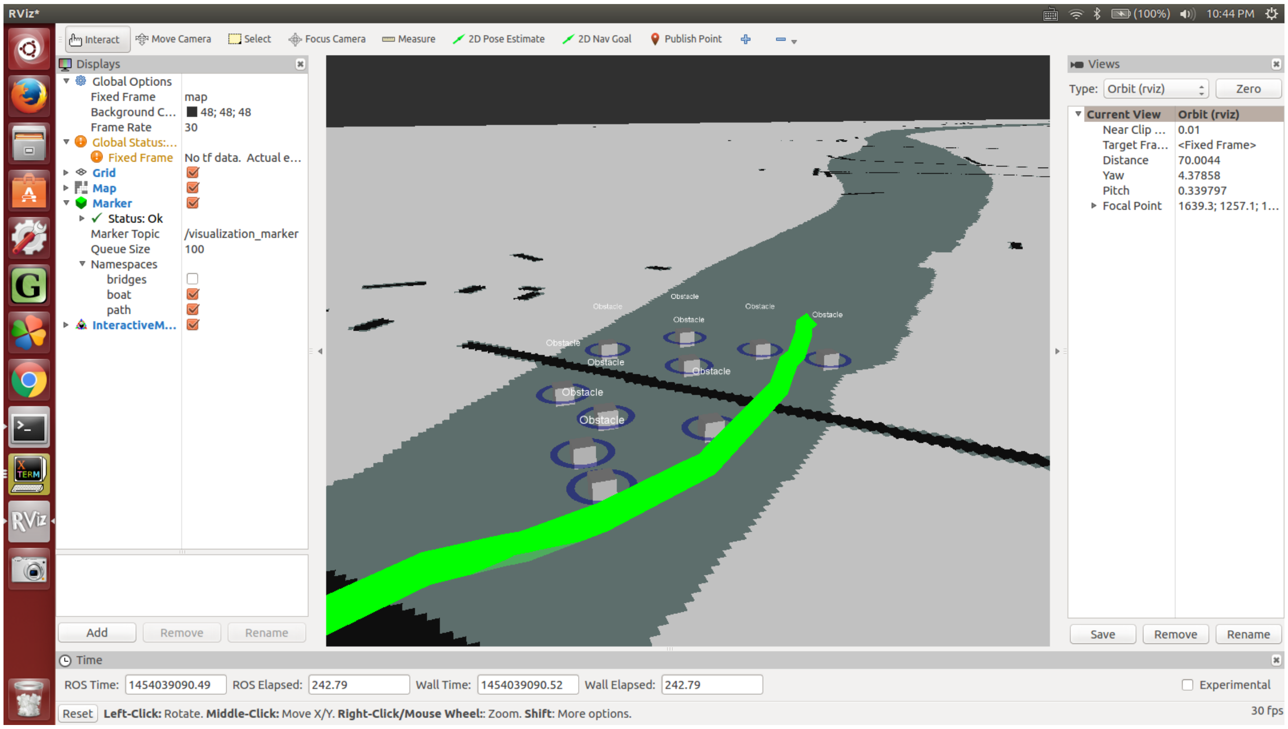
Type: Orbit (rviz) Zero

Current View	Orbit (rviz)
Near Clip …	0.01
Target Fra…	<Fixed Frame>
Distance	48.5343
Yaw	4.37858
Pitch	0.339797
Focal Point	1637; 1257.3; 16.…

ROS Time: 1454039038.18 ROS Elapsed: 190.48 Wall Time: 1454039038.22 Wall Elapsed: 190.48 □ Experimental

30 fps

Figure 4: Path Planned using the Interactive Markers with markers at random position 1



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	
bridges	☐
boat	☑
path	☑
InteractiveM...	☑

Type: Orbit (rviz)

Current View	Orbit (rviz)
Near Clip ...	0.01
Target Fra...	<Fixed Frame>
Distance	70.0044
Yaw	4.37858
Pitch	0.339797
Focal Point	1639.3; 1257.1; 1...

ROS Time: 1454039090.49 ROS Elapsed: 242.79 Wall Time: 1454039090.52 Wall Elapsed: 242.79 ☐ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

Figure 5: Path Planned using the Interactive Markers with markers at random position 2

3 Challenges

The major challenge I faced was that the documentation for the `interactive_markers` package on ROS website was pretty terse. Since this is the first time I have worked on using markers in RViz, I had to search a lot on forums to figure out the errors that I was getting. Writing the callback function was the most difficult part since the callback function does not take arguments and I had to make the variables global. This consumed a considerable amount of time.

We are having problems going for field testing because of the inclement weather conditions. This past Monday, we wanted to test our path planning algorithm on the boat but we were unable to do so because of the cold weather.

4 Teamwork

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

- **Tushar Chugh** – Tushar worked on integrating the radar data from the perception subsystem with the path planning subsystem. He also worked on adding continuous re-planning option to the path planner.
- **Shiyu Dong** – Shiyu worked on writing a launch file to start all the nodes at once. This helped us reduce the time to start the system.

- **Tae-Hyung Kim** – Tae-Hyung is working on integrating the GPS data with the ROS navigation stack.
- **William Seto** – William worked on integrating most of our code including the `int_markers` package. He also created new maps with bridge information and cleaned up the code on GitHub.

5 Work Overview for Coming Week

We need to add the functionality of simulating dynamic obstacles in the environment in the `int_markers` package. I will be working on this task.

The filtering code takes a lot of time to run which slows down the overall pipeline. We are considering the option to run this code on a GPU because most of the filtering can be in parallel. This should give us better performance. Tae-Hyung has a NVIDIA Jetson tk1 and we will perform initial testing on this. I plan to take the lead on this task. If the results are promising, then we might buy a GPU from our team budget.

References

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