Progress Review 2

Tae-Hyung Kim (tk1)

Team B / Auto Pirates

Teammates: Tushar Chugh, Shiyu Dong, William Seto,
Bikramjot Hanzra

ILR #03

October 30, 2015

1. Individual Progress

I've worked SBPL(Search Based Planning Library) simulation on the ROS this week. First, I tried to install ros-jade-sbpl and ros-jade-sbpl-lattice-planner package. However, I had a couple of errors while running sbpl-lattice-planner. After some trouble shooting, I realized that the ROS jade version, recent distribution, was not supporting packages stably. I decided to switch to ROS indigo version. It was good choices to proceed further steps because it has been supporting various packages. In the course of implementing sbpl simulation on stage package, I faced various problems. First, it did not have robot model in stage simulation. Second, it needed several other dependent packages like move-base, navigation, tf (transformation), and fake localization, etc. Although I couldn't get to the my goal, implanting sbpl simulation on the ROS, I learned several things, how to use rviz (ros visualizing tool), what Occupancy Grid Map is. Lastly, I reached to load Occupancy Grid Map on the rviz. (Figure 1)

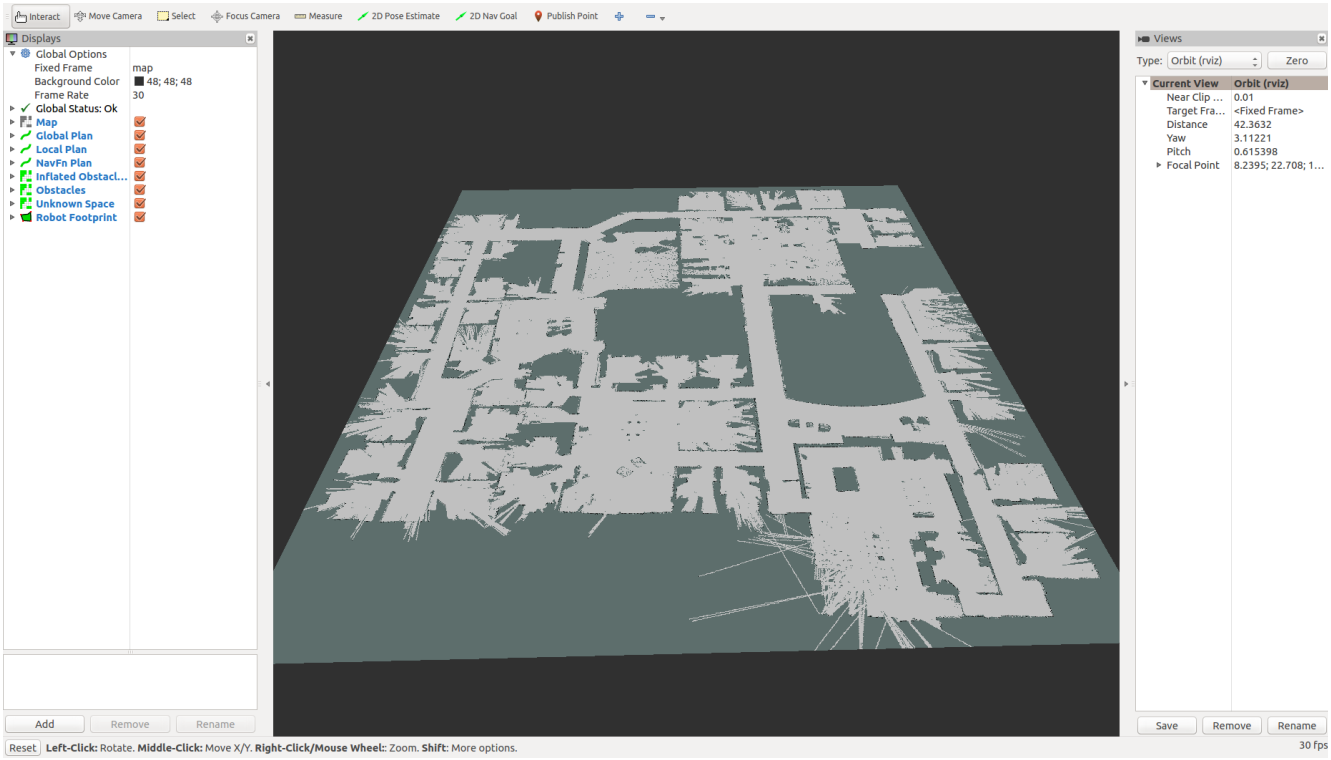

Figure 1. Occupancy Grid Map on the rviz

On Wednesday, I had meeting with Andrew Dornbush, the software researcher in SBPL lab. I explained what our autonomous boat project was and what our plan for path planning library was. As I was responsible for implanting path planning, I discussed how to apply SBPL(Search Based Planning Library) to our autonomous boat system and how to approach to analyze the SBPL source code. What I've heard from him was that we could use ARA* and AD* path planning algorithms. Specifically, ARA* is used in non-dynamic environment and simpler to analyze it. Otherwise, AD* is utilized in dynamic environment and more difficult to approach it. Next, the approaching ways to apply SBPL to our system are to modify the cost map and apply time tables. After some discussion, he let me know that what study resources were. Most of them were the tutorials of ROS packages related to navigation. Then, I asked him when he was available in a week.

After PR2, I found useful simulation package, a turtlebot stage simulation package (ros-indigo-turtlebot-stage) to grasp navigation and path planning. I tested it navigating to the goal position on the rviz. When I set the goal position of turtlebot (Figure 2), the simulation showed the turtlebot navigating through the path to goal position (Figure 3). I assumed that this turtlebot simulation helped us to grasp how to operate the navigation.

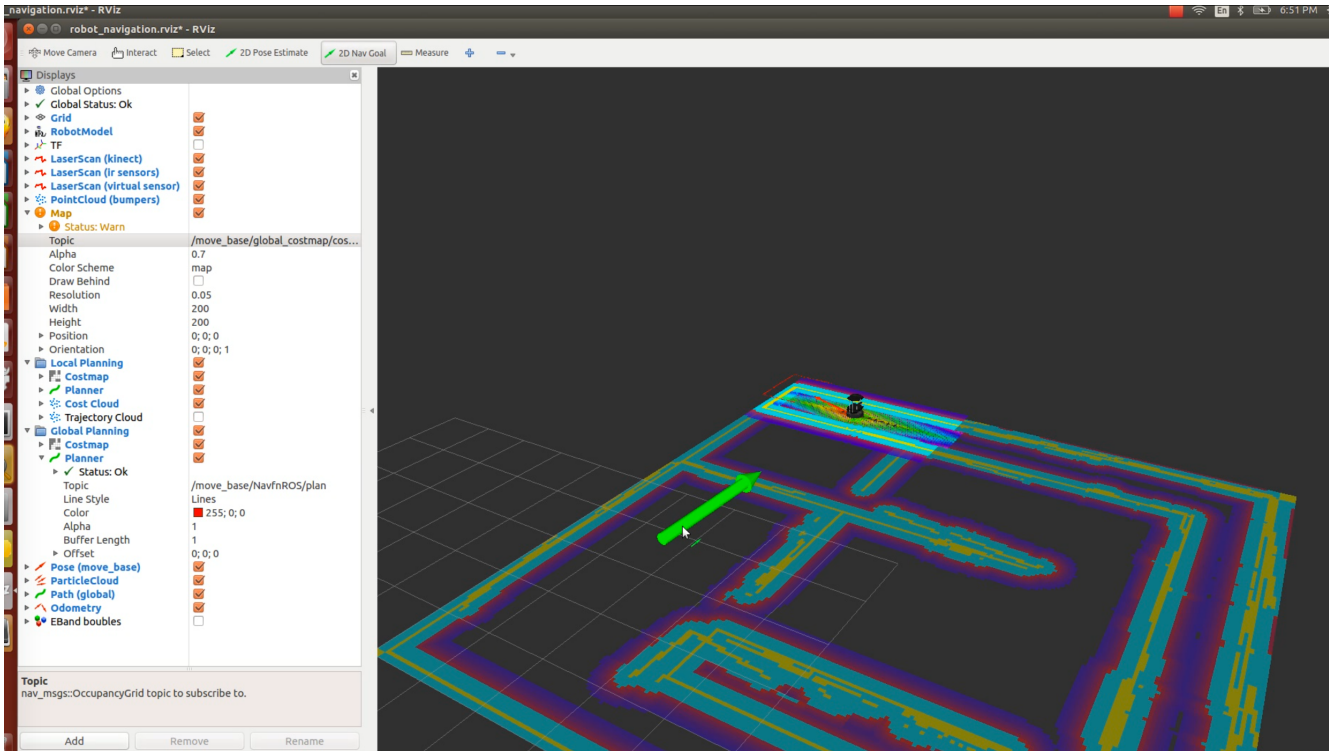

Figure 2. Setting the navigation goal to the turtlebot

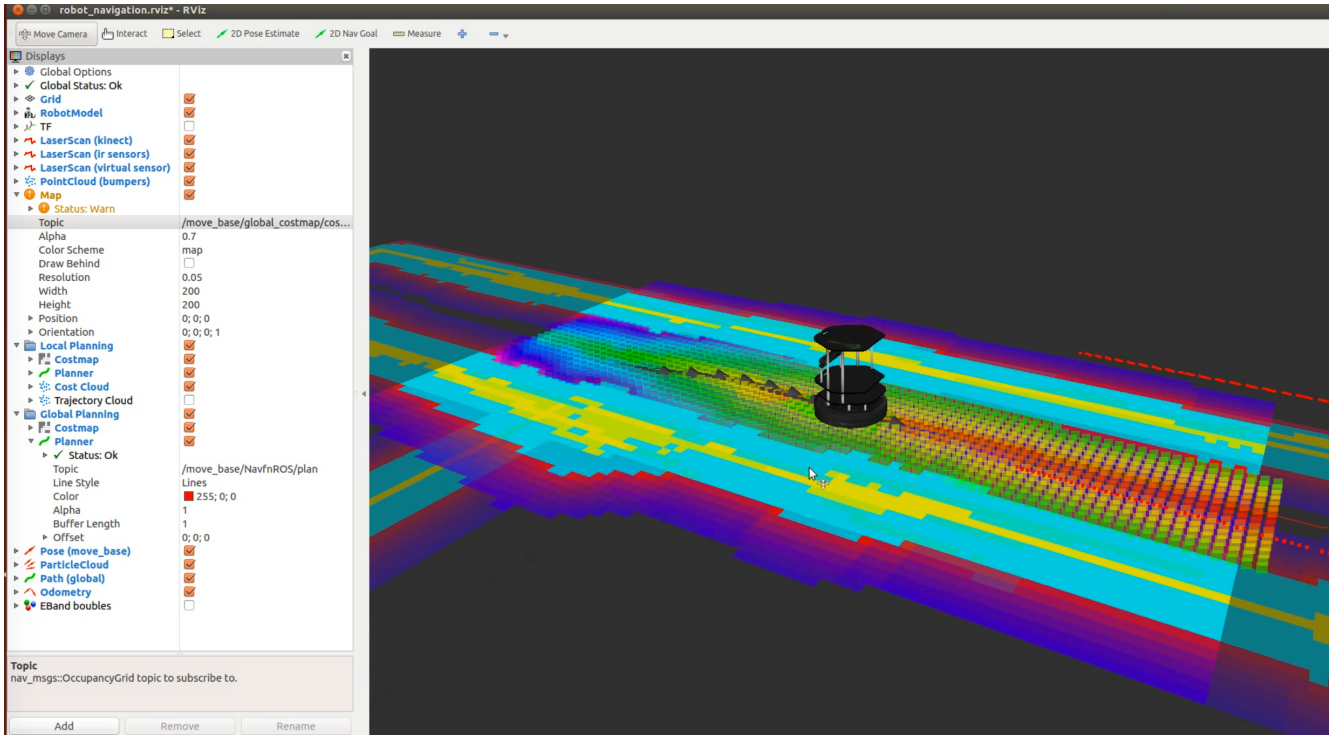

Figure 3. Navigation of turtlebot with path planning

2. Challenges

One of this week challenges was CAD assignment. I spent a lot of time about 8 hours to draw ten parts. I couldn't find menus that I implemented so I was following some youtube videos to notice the function of menus. And I got some helps from Shiyu to make drawing files.

Next, I was troubled to demonstrate simulation for PR2. The goal was not achieved on the PR2 presentation day. The reason why I couldn't succeed our goal was that I knew the part of the ROS packages. As path planning package is entangled with several other packages, move-base, navigation, tf, stage simulation, and localization package. But, I didn't install full of packages depended with SBPL library. The other reason was that there were not complete the world config files on stage simulation. As I didn't have the world config file specifying robot model, I had an error which ROS stage node failed to load robot model. After PR2, I found the turtlebot-stage package and I could see the simulation was operating. I realized that surveying top to bottom of ROS is more useful for us than bottom to top of that. And other thing that I realized that we should choose stable distribution version of ROS,

indigo. Because a lot of packages supported the indigo ROS. it allowed me to waste less time to implement simulation than the jade ROS. As well, it provided me study material, the world config files on the stage in the turtlebot simulation package.

3. Teamwork

1) Shiyu Dong: Shiyu researched on the data format of radar and preprocessing of data. He also revised the risk management and fall validation for perception sections of CoDR.

2) Bikram Hanzra: Bikram created the model of the boat in solidworks. He also worked with Tushar to explore functionalities of SBPL. In addition to it, he simulated world of dummy environment on gazebo.

3) William Seto: William worked on the integration of OpenCPN and ROS. Also, he worked with the team to revise CoDR (FVE, WBS and critical path)

4) Tushar Chugh: Tushar did hands-on graph algorithm, and built SBPL library. He wrote PCB conceptual design and revision of CODR.

4. Future Plans

First, I plan to write the technical documents about the map generation from open street map to Occupancy Grid map. I'll make the sample map in that way and test it on the stage simulation on the ROS.

Second, I'll make the robot model considered to boat's kinematic features, boat dimension and its velocity. Considering these factors, I can decide map size and its grid size.

Lastly, I plan to survey the way how each grid correspond its longitude and latitude from open street map and GPS.