HOCHSCHULE
RHEIN-WAAL
Rhine-Waal University
of Applied Sciences

LABORATORY

Autonomous Mobile Robotics

SESSION # 3

More on Simulation

SESSION OVERVIEW

Today's session will cover running simple SLAM for Turtlebot 2 via Gazebo and Rviz.

Objective:

1. To be able to map virtual environments using the turtlebot packages
2. To enable the Turtlebot to autonomously navigate a mapped environment

Happy learning!

1. MAPPING

To map a world in Gazebo, we first launch the `turtlebot_world` launch file as in the last session. For example, we launch the WillowGarage world as follows (also refer Figure 1):

```
$ roslaunch turtlebot_gazebo turtlebot_world.launch
world_file:=worlds/willowgarage.world
```

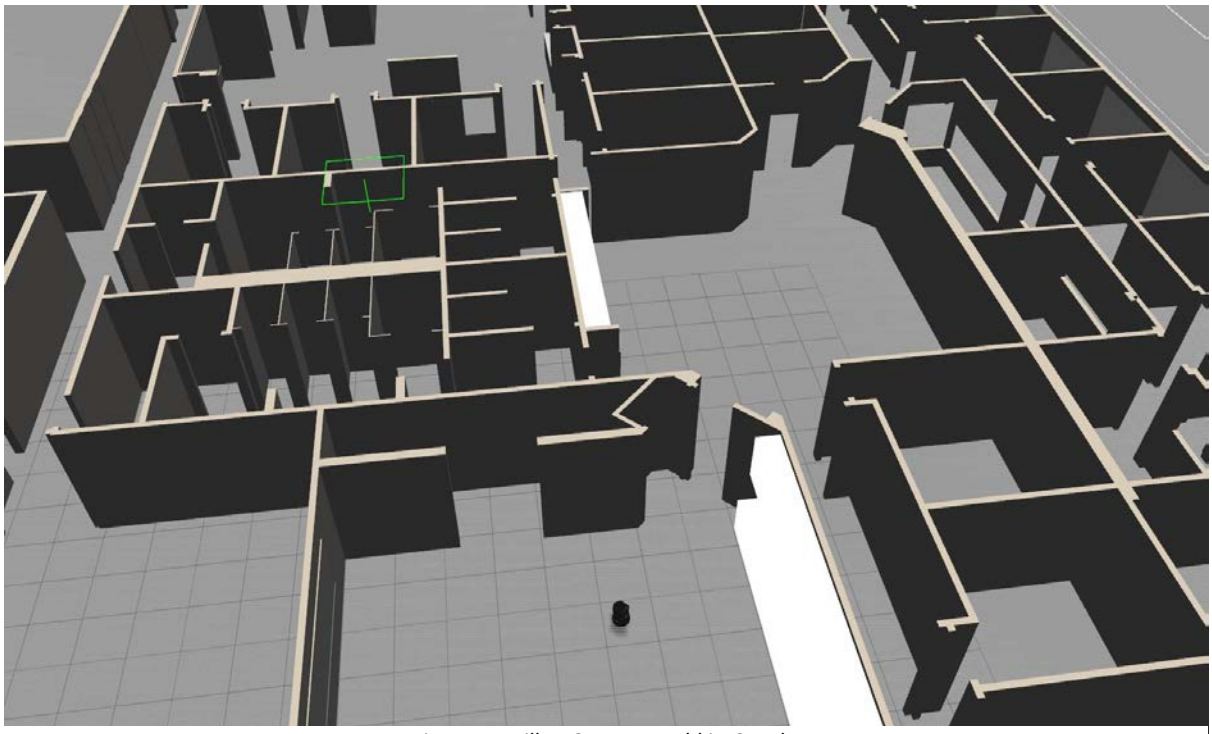

Figure 1: WillowGarage world in Gazebo

The turtlebot has a simple mapping mode/package which can be used in Gazebo as well, by typing the following command in a new terminal:

```
$ roslaunch turtlebot_gazebo gmapping_demo.launch
```


Hochschule Rhein-Waal
Marie-Curie Strasse 1
D-47533 Kleve

We use Rviz to visualize the mapping process, by running the following command in another terminal:

```
$ roslaunch turtlebot_rviz_launchers
view_navigation.launch
```

This should open up an Rviz window with a top (sky) view of the turtlebot that updates the topography dynamically, as the robot traverses the world. In order to control the turtlebot, we'll use the gamepads via the `turtlebot_telelop` package by running:

```
$ roslaunch turtlebot_teleop logitech.launch
```

Once you've generated an accurate map in Rviz, save the map by running the following the command in a new terminal:

```
$ rosrun map_server map_saver –f <full path with map
name>
```

This creates two files: one is a greyscale image file (.pgm) that contains the 2D plan of the world as generated by the robot, the other is a configuration file (.yaml) that stores origin, resolution and reference to the .pgm file among other data.

To save the map, we first create another directory. One room of the WillowGarage world was recorded and saved as follows:

```
$ mkdir ~/AMR_WS19/hello_turtle/maps

$ rosrun map_server map_saver –f
~/AMR_WS19/hello_turtle/maps/willowmap
```

Shown below in Figure 2 is the `willowmap.pgm` file. The black lines represent obstacles detected

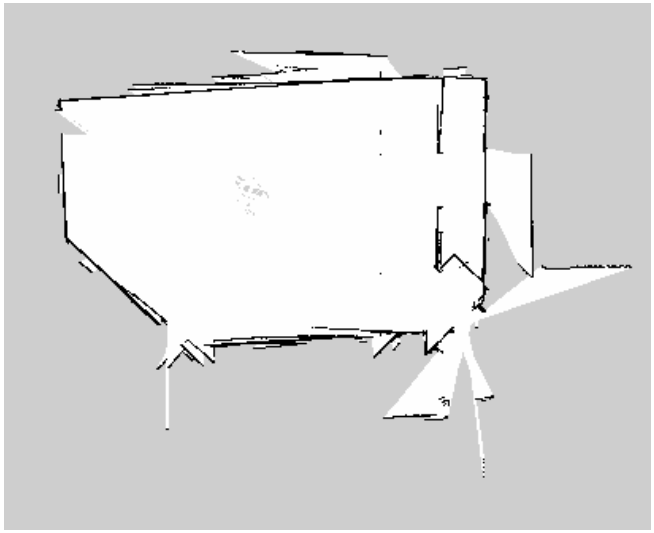

Figure 2: willowmap.pgm

by the Kinect sensors and the grey area corresponds to the uncharted areas.

Hochschule Rhein-Waal
Marie-Curie Strasse 1
D-47533 Kleve

2. AUTONOMOUS NAVIGATION

Once mapping has been done, close all the applications (terminals) and relaunch the turtlebot in the corresponding world. Next open up the virtual turtlebot navigation package in a new terminal window using the `map_file` argument:

```
$ roslaunch turtlebot_gazebo amcl_demo.launch
map_file:=<full path to yaml file>
```

This enables the collision detection and autonomous navigation of the robot in the defined map. Open the navigation view of Rviz to see the robot sensor data.

For the WillowGarage map, running the below command opens up the following window (Figure 3):

```
$ roslaunch turtlebot_gazebo amcl_demo.launch map_file:=
~/AMR_WS19/hello_turtle/maps/willowmap.yaml
```

Sometimes, using the ~ sign may not work, in that case, you have to type the full path address which

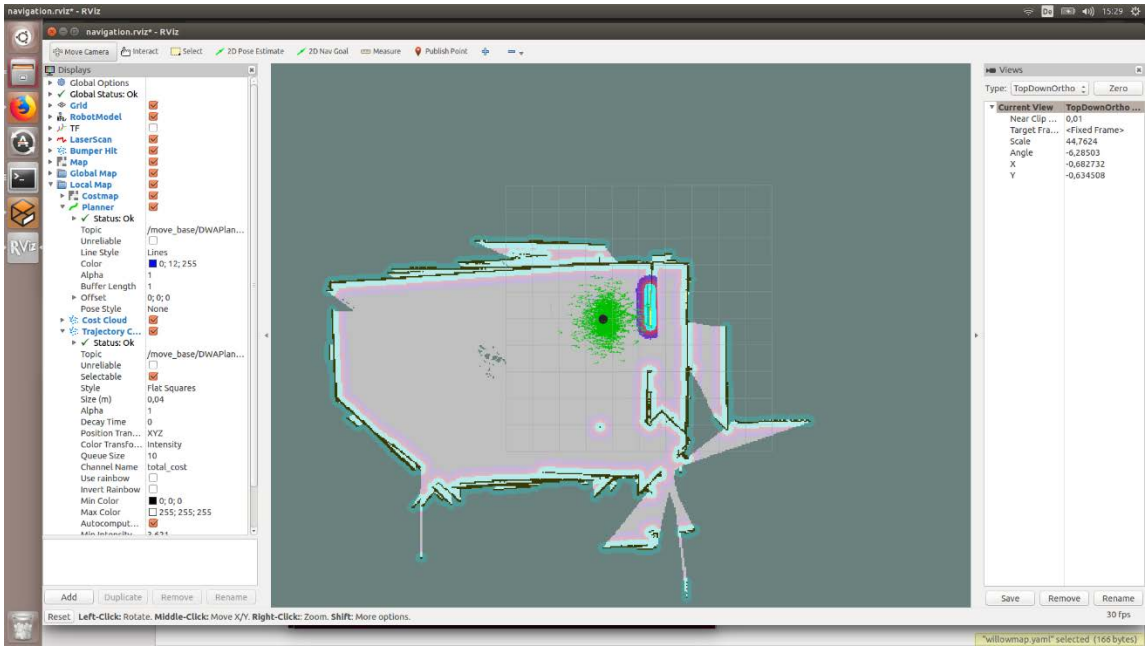

Figure 3: RViz in Navigation view

starts with `/home/....`

2.1. INITIAL POSE ESTIMATION

If the turtlebot is initialised at some other point in the map than when mapping was done, it is necessary to specify the initial pose else it can lead to odometry issues and could cause the package to crash. This can be done in 2 ways:

2.1.1. VIA RVIZ

Once Rviz is open, you can specify the new initial pose by clicking the 2D Pose Estimate button [Figure 4] and then clicking at some point in the map and dragging the mouse in the direction of the robots orientation. This is useful since it involved just picking a point on the map.

Hochschule Rhein-Waal
Marie-Curie Strasse 1
D-47533 Kleve

2.1.2. VIA COMMAND LINE

If you know the new initial pose coordinates, you can provide them as arguments while launching the `amcl_demo` package:

```
$ roslaunch turtlebot_gazebo amcl_demo.launch
map_file:=<map file location> intial_pose_x:=<floating
point number> intial_pose_y:=<floating point number>
intial_pose_a:=<floating point number>
```

2.2. SENDING NAVIGATION GOALS

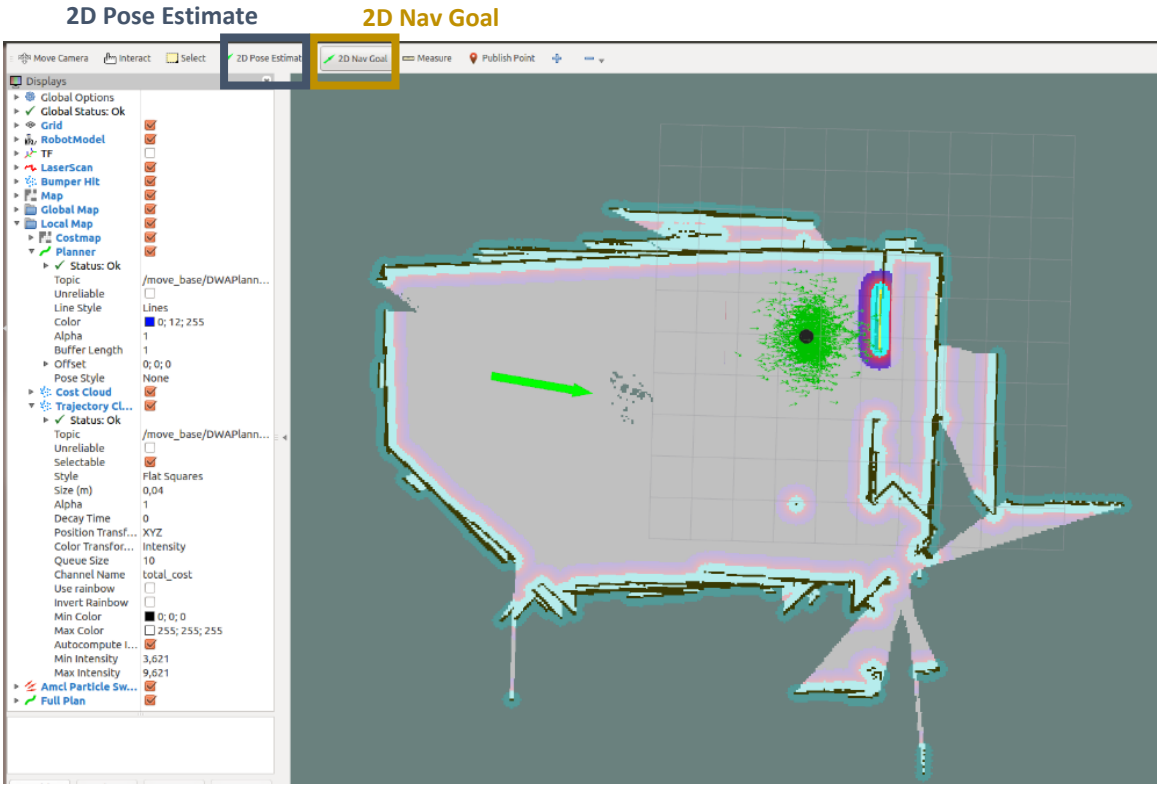

Figure 4: 2D NavGoal Command in RViz GUI

You can easily send the destination pose by clicking on 2DNavGoal button and then clicking on any point within the map and dragging the mouse to specify the robots final orientation (refer Figure 4).

***

For today's tasks, make sure you have already created your account on Gitlab.

Hochschule Rhein-Waal
Marie-Curie Strasse 1
D-47533 Kleve

TASKS

1. Download the `maze4.world` file from the Gitlab repository and place it in the `custom_gazebo_worlds` folder. In Rviz, add Image to the Displays tab and play around with the settings so that the turtlebot camera output is visible in Rviz itself. This way you could possibly avoid having 2 windows side by side while mapping.
   Map this world and save it as `maze` in the `maps` directory.

2. Next send a navigation goal that places the turtlebot right in front of the warehouse robot, but facing the other way.

3. Now insert a model into this world (something not too large, say, for example a human) somewhere in front of the turtlebot. Send a new navigation goal. Is the turtlebot able to respond to this new change in its map?
   Try introducing a larger models that obstructs it's path. How does the turtlebot respond?

4. We can use command line arguments to specify the initial pose of the turtlebot. But how or where do we get this initial pose from? [**HINT:** play around with `rostopic list`]

REFERENCES AND FURTHER READING

[1] (n.d.). Retrieved from Gazebo - Robot Simulator: http://gazebosim.org/

[2] *A "Getting Started" Guide for Developers Interested in Robotics*. (n.d.). Retrieved from Learn Turtlebot and ROS: https://learn.turtlebot.com/

[3] *rviz*. (n.d.). Retrieved from ROS.org: http://wiki.ros.org/rviz

Hochschule Rhein-Waal
Marie-Curie Strasse 1
D-47533 Kleve