Autonomous Driving Turtlebot

*Prerequisite to this tutorial: make a map -* [*http://learn.turtlebot.com/2015/02/01/11/*](http://learn.turtlebot.com/2015/02/01/11/)

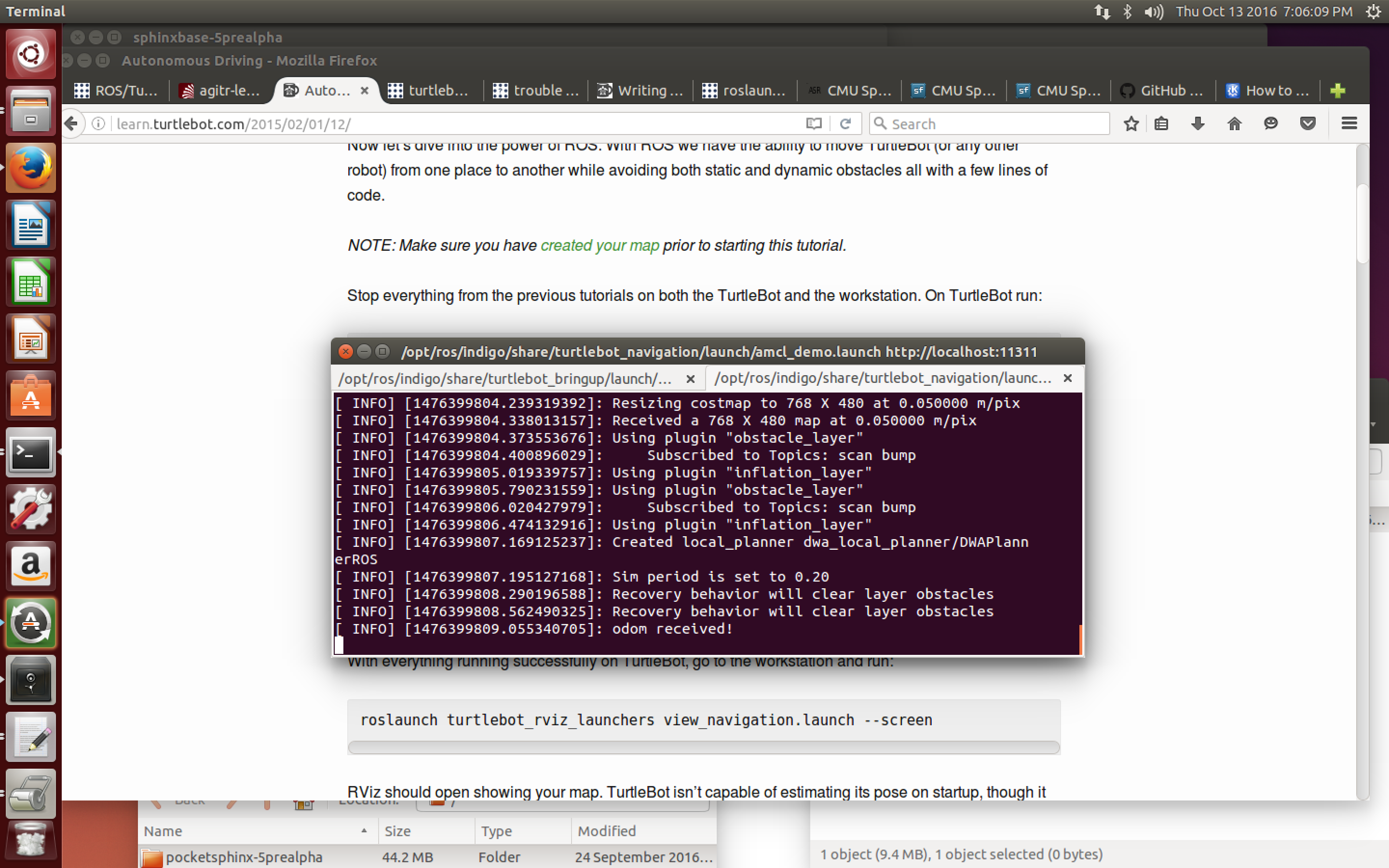
*Make sure ROS\_MASTER\_URI=http://IP\_ADDRESS\_OF\_TURTLEBOT and ROS\_HOSTNAME=IP\_ADDRESS\_OF\_WORK\_STATION*

First, run the following 2 terminal commands on the Turtlebot

roslaunch turtlebot\_bringup minimal.launch

roslaunch turtlebot\_navigation amcl\_demo.launch map\_file:=/tmp/my\_map.yaml

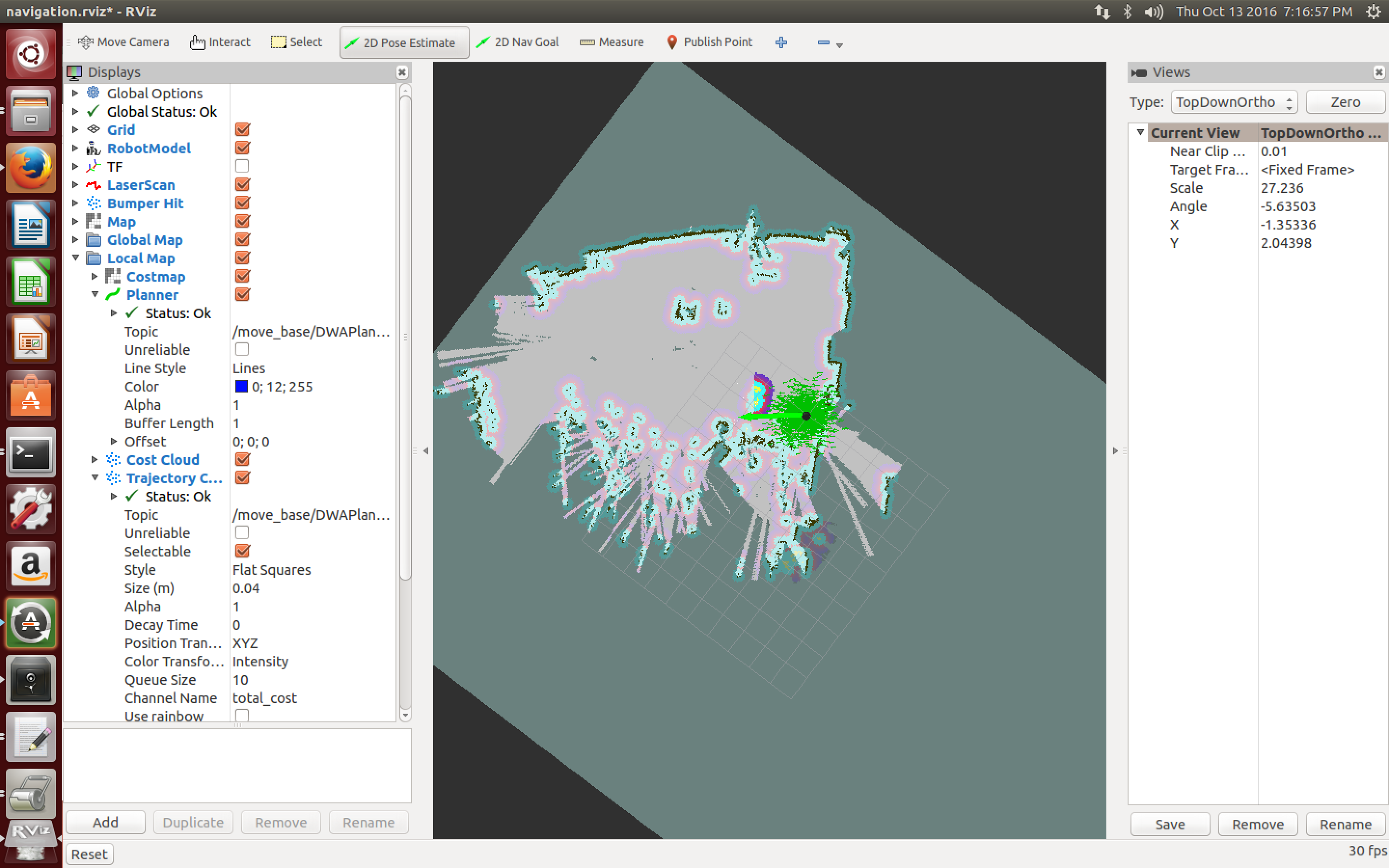
‘odom received’ means everything worked properly.



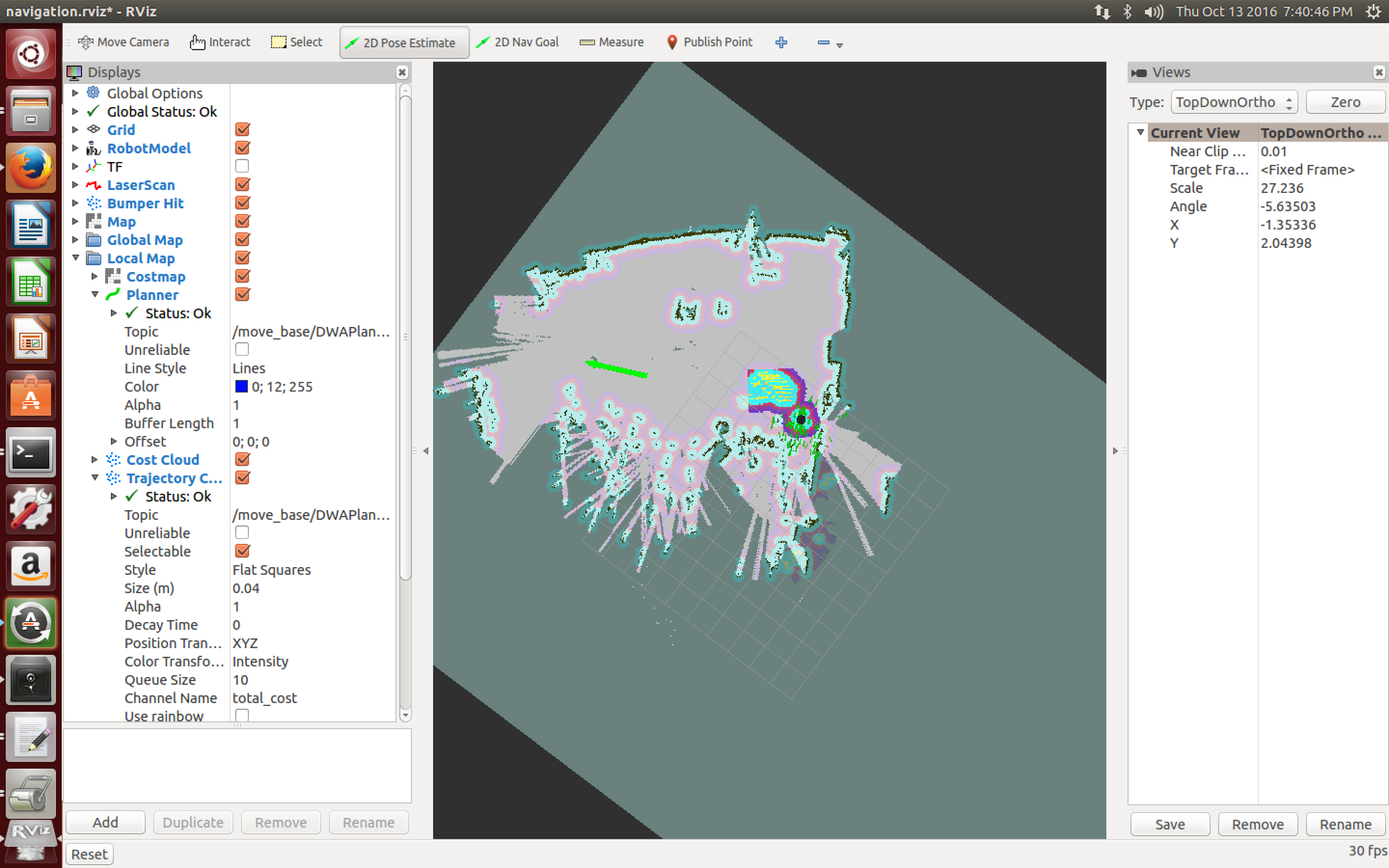
Next, on your workspace run:

roslaunch turtlebot\_rviz\_launchers view\_navigation.launch --screen

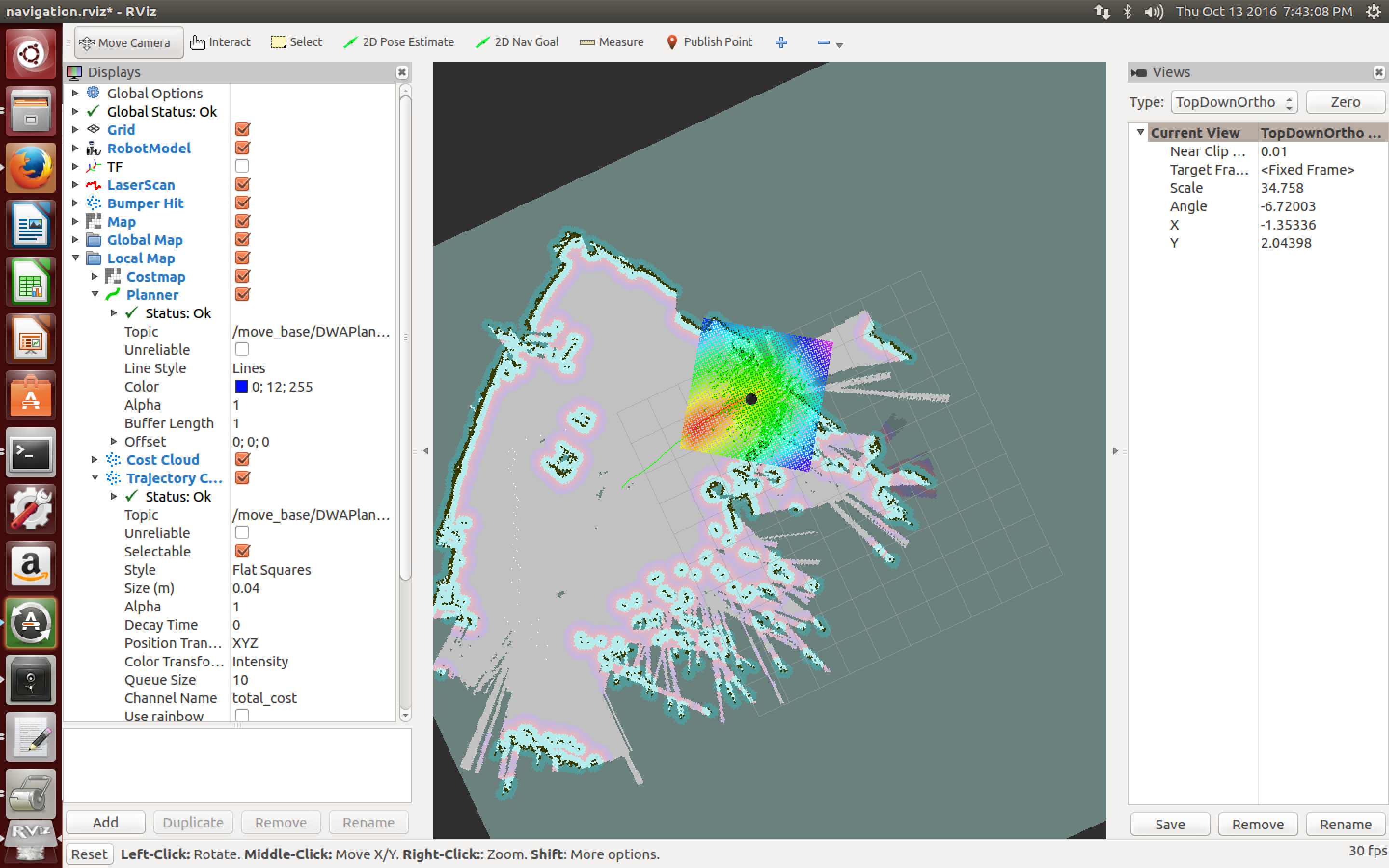
This will open and will display the map you created in RViz. Because Turtlebot relies on relative coordinates on the map, you will need to manually tell Turtlebot where he is located on the map and which direction he is facing. To do this, select “2D Pose Estimate” in the toolbar at the top of the RViz screen, and click where Turtlebot should be initialized and then drag the green arrow in the direction Turtlebot is facing.



Now, that Turtlebot knows his location on the map, click on “2D Nav Goal” and select the point on the map where Turtlebot should travel.



A line will appear mapping the projected path Turtlebot will take to get to the destination on the map. It will avoid any obstacles or people walking in his way.



Autonomous Docking

*Important: Make sure the docking station is up against a wall or secured to the ground, if not Turtlebot may push it around and prevent it from docking properly.*

Turtlebot is able to dock autonomously as long as it is within 3 meters line of sight of a docking station.

First, run the following terminal command on the Turtlebot

roslaunch turtlebot\_bringup minimal.launch

Then, run the following 2 terminal commands on the workstation

roslaunch kobuki\_auto\_docking minimal.launch --screen

roslaunch kobuki\_auto\_docking activate.launch --screen

No map is needed to autonomously dock Turtlebot. Turtlebot uses three IR receivers to do this.

Autonomous Driving with Code

First clone the following github repository: <https://github.com/markwsilliman/turtlebot/>

Go forward until told to stop – No obstacle avoidance

Run the following terminal command on the Turtlebot

roslaunch turtlebot\_bringup minimal.launch

Run the following terminal command on the workstation

python ~/helloworld/turtlebot/goforward.py

This code tells Turtlebot to go forward until CTRL + C is pressed to make Turtlebot stop moving. Because no map is loaded, and Turtlebot is not aware of his surroundings, he will not avoid obstacles.

Go forward 3 meters and avoid obstacles

Now, the following code, will tell Turtlebot to go forward 3 meters.

Run the following 2 terminal commands on the Turtlebot

roslaunch turtlebot\_bringup minimal.launch

roslaunch turtlebot\_navigation amcl\_demo.launch map\_file:=/tmp/my\_map.yaml

Run the following terminal command on the workstation

roslaunch turtlebot\_rviz\_launchers view\_navigation.launch --screen

In RViz, initialize Turtlebots position and direction using 2D Pose Estimate, as described above.

Run the following terminal command on the workstation in a new terminal window

python ~/helloworld/turtlebot/goforward\_and\_avoid\_obstacle.py

Turtlebot will avoid obstacles if anything is placed in front of the 3-meter path.

Specific location on the map

Now, lets try telling Turtlebot to go to a specific location on the map.

Run the following 2 terminal commands on the Turtlebot

roslaunch turtlebot\_bringup minimal.launch

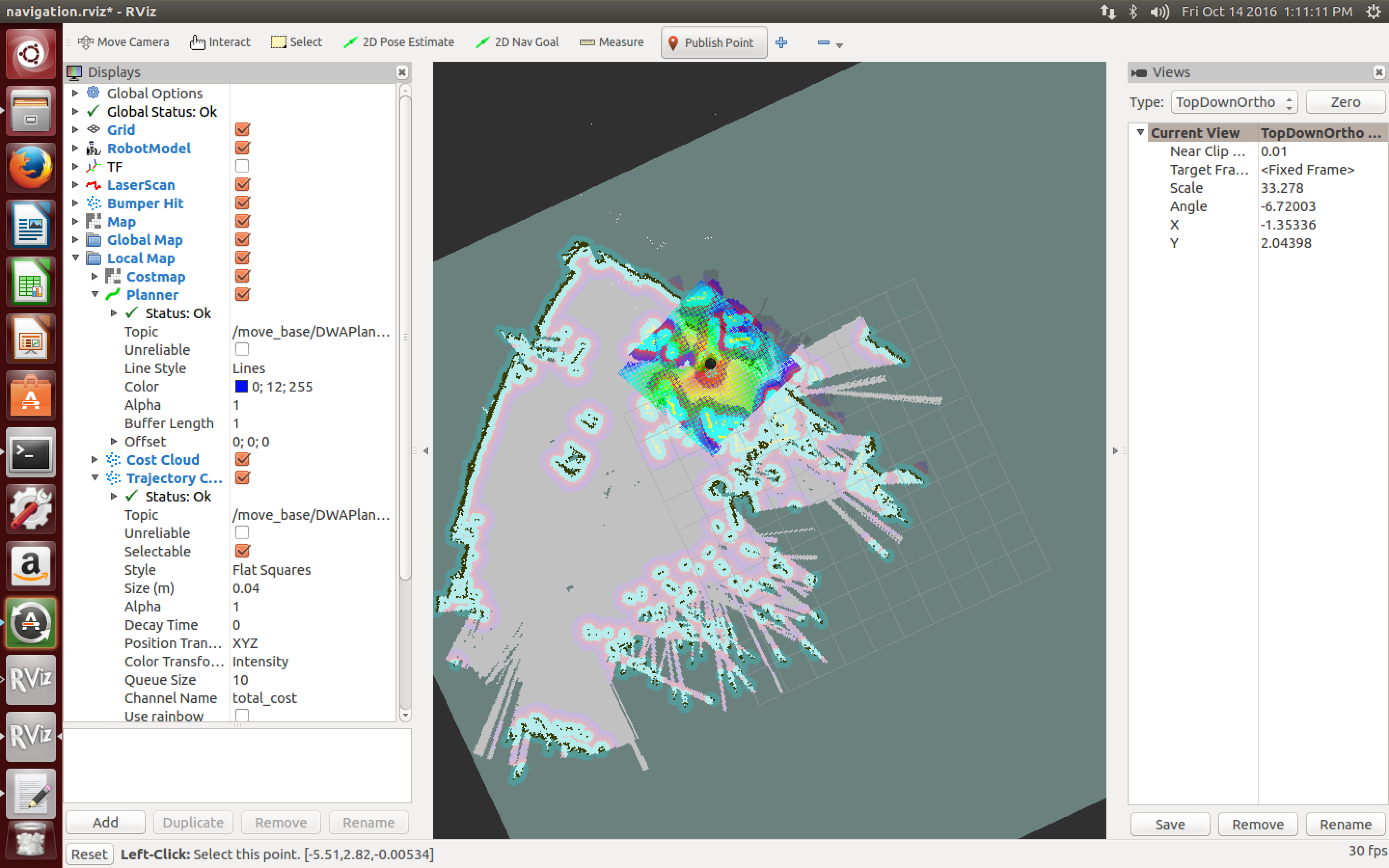
roslaunch turtlebot\_navigation amcl\_demo.launch map\_file:=/tmp/my\_map.yaml

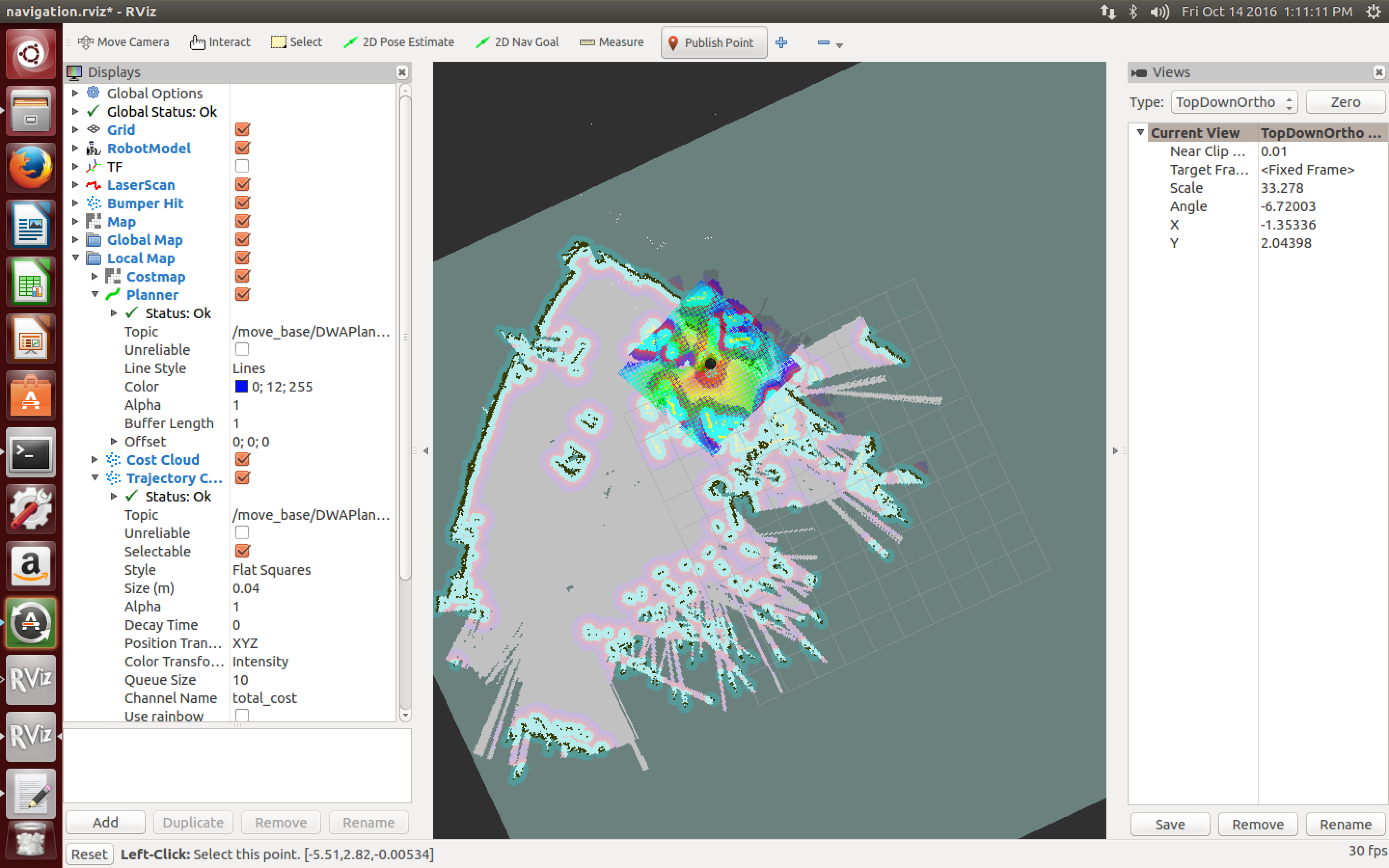
Run the following terminal command on the workstation

roslaunch turtlebot\_rviz\_launchers view\_navigation.launch --screen

As described above, in RViz, initialize Turtlebots position and direction using 2D Pose Estimate.

In RViz, select “Publish Point” in the toolbar above. Hover over the location on the map that Turtlebot should travel to. Do not click on that location, instead in the bottom left corner of RVIS, there will be 3 coordinates. The first is the x coordinate, the second is the y coordinate, and the third is altitude. Record the x and y coordinate of the destination location, and although altitude is not exactly zero, 0 should always be used anyways.

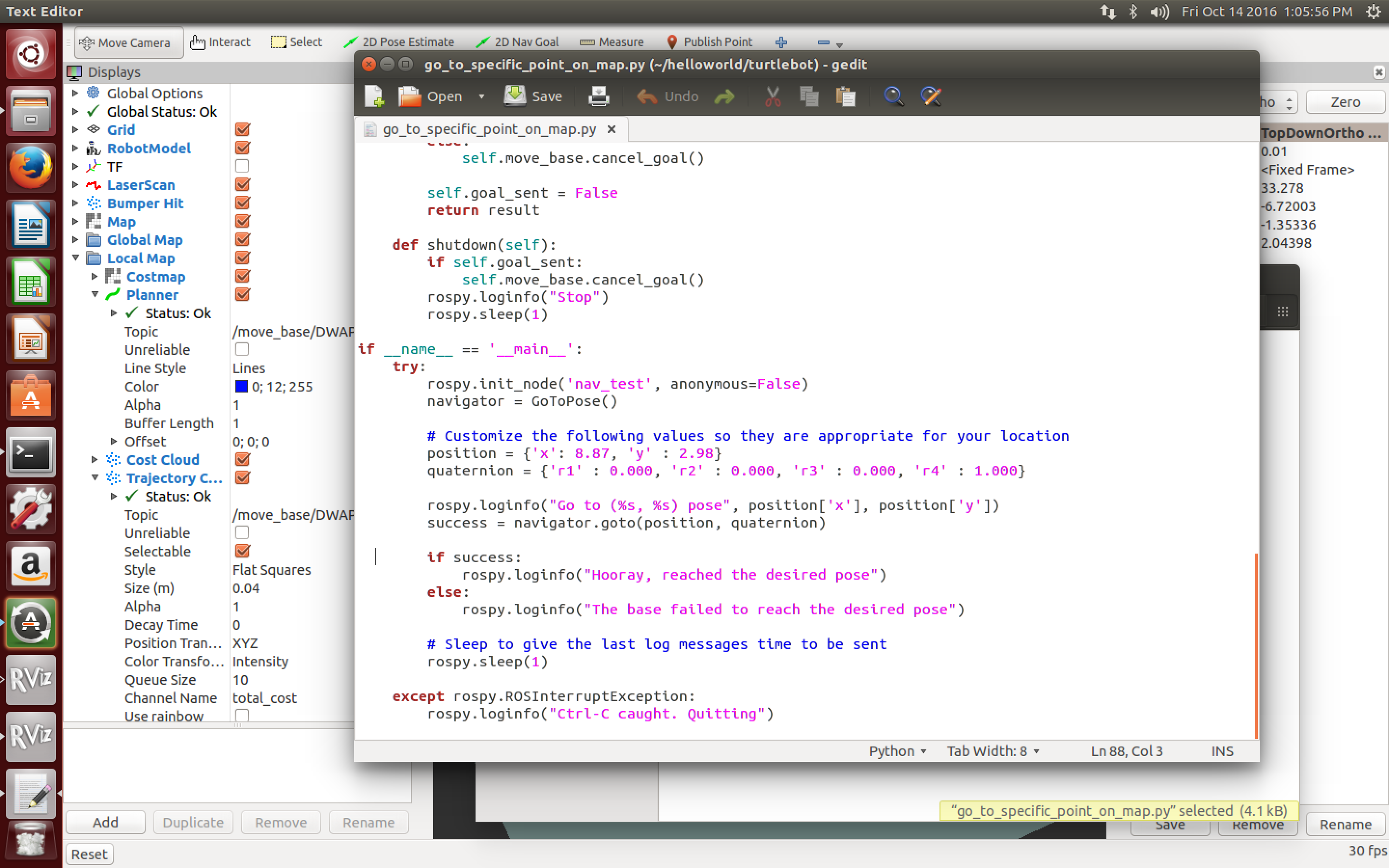


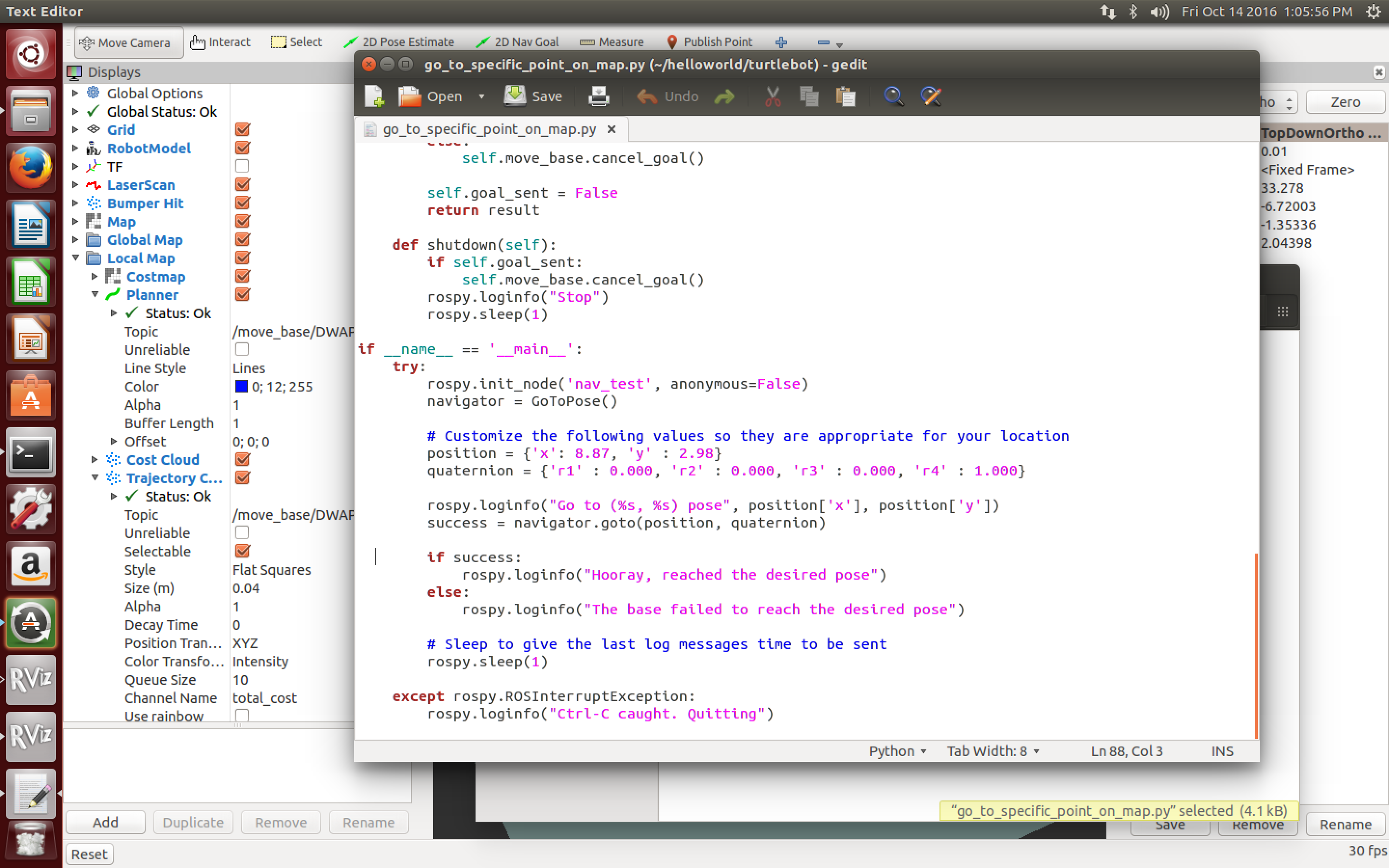


Run the following terminal command on the workstation

gedit ~/helloworld/turtlebot/go\_to\_specific\_point\_on\_map.py

You will need to change line 83. Here, input the destination coordinates that were written down from RViz. Save that file.





Finally, run the following terminal command on the workstation

python ~/helloworld/turtlebot/go\_to\_specific\_point\_on\_map.py

Turtlebot will now go to the specified point while avoiding any obstacles or people that come into the way. RViz will show the projected travel path Turtlebot is taking and will show what obstacles it sees around him.

