

ME 134 HW 1 - MOTION PLANNING

Released: Monday 04/23/2018

Due: Monday 04/30/2018 - 6pm

In this homework, you will learn how to use the map constructed by gmapping to plan a path that avoids obstacles. You will learn how to use the built-in global planner on the Turtlebot to autonomously navigate to designated target points. This will build off your results from Lab 1, and require you to modify the built-in global planner on the Turtlebot's navigation stack, and send designated goals to the Turtlebot.

1 Building a Map - Occupancy Grid (50 Points)

1. Run the bag file from the previous lab, and build a map (occupancy grid) using the ROS gmapping node. If you are having problems processing your group's data, you may use [this](#) bag file. Print out the resulting map using the map_server node. Also print out a graph of the connecting nodes/topics (you may use rqt_graph).
2. In 2-3 sentences, describe the data format of the map output by the gmapping node. How is it represented, and what do the numbers mean?
3. What is the resolution of the map (i.e. how many meters per cell)? In either your bag file or the provided one, what is the origin of the map, and what would be the real-world position of the center of cell [3,2] (relative to the origin). Show how you got your results.

2 Modifying the Navigation Stack (50 Points)

The navigation stack plans and controls the motion of your robot in ROS. In this section, you will begin examining how the navigation stack works, and how you can interface with it using the move_base node. Read about the move_base node [here](#), and the navigation stack [here](#).

1. What topics does the move_base node publish to and subscribe to? In a sentence, what is the purpose of each of these topics?
2. Where does the previously constructed map factor into the move_base node?
3. What is the default global planner used by move_base? Do the following in order to change the default global planner:

```
$ roscd turtlebot_bringup
```

```
% cd launch/includes
```

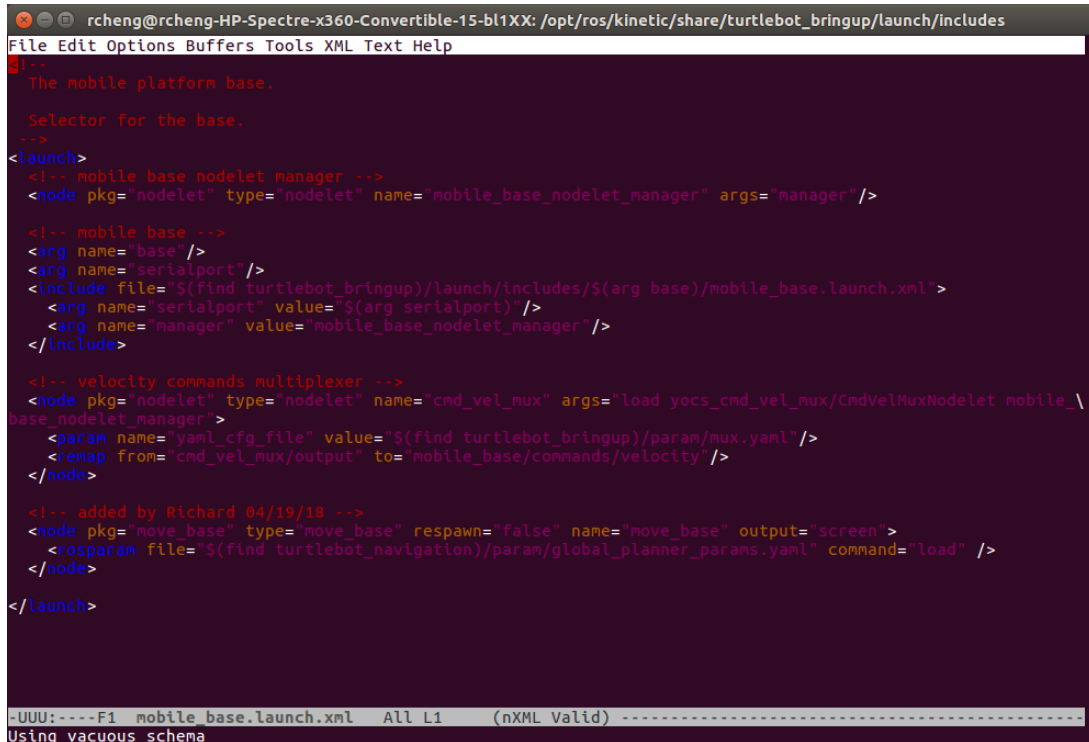
Replace the mobile_base.launch.xml file with the file [here](#).

```
$ roscd turtlebot_bringup
```

```
% cd param
```

Add [this](#) file in the param folder.

Briefly, what is the purpose of the last 3 lines of the launch file shown below. What is the resulting global planner being used? (hint: Look at the file you just uploaded to the param folder)



```
rcheng@rcheng-HP-Spectre-x360-Convertible-15-bl1XX: /opt/ros/kinetic/share/turtlebot_bringup/launch/includes
File Edit Options Buffers Tools XML Text Help
<!--
The mobile platform base.

Selector for the base.
-->
<launch>
  <!-- mobile base nodelet manager -->
  <node pkg='nodelet' type='nodelet' name='mobile_base_nodelet_manager' args='manager' />

  <!-- mobile base -->
  <arg name='base' />
  <arg name='serialport' />
  <include file='$(find turtlebot_bringup)/launch/includes/$(arg base)/mobile_base.launch.xml'>
    <arg name='serialport' value='$(arg serialport)' />
    <arg name='manager' value='mobile_base_nodelet_manager' />
  </include>

  <!-- velocity commands multiplexer -->
  <node pkg='nodelet' type='nodelet' name='cmd_vel_mux' args='load yocs_cmd_vel_mux/CmdVelMuxNodelet mobile_
base_nodelet_manager'>
    <param name='yaml_cfg_file' value='$(find turtlebot_bringup)/param/mux.yaml' />
    <remap from='cmd_vel_mux/output' to='mobile_base/commands/velocity' />
  </node>

  <!-- added by Richard 04/19/18 -->
  <node pkg='move_base' type='move_base' respawn='false' name='move_base' output='screen'>
    <rosparam file='$(find turtlebot_navigation)/param/global_planner_params.yaml' command='load' />
  </node>
</launch>

UUU:----F1 mobile_base.launch.xml All L1 (nXML Valid) -----
Using vacuous schema
```

3 Sending Messages to Designate Target Positions (40 Points)

Write a properly formatted message to send to the move_base node that will give it the target position. Use your answer to Problem 2(1) to determine where to publish your message to and what type of message you should publish.

You may use the template [here](#), though you are also free to write your own script. Verify that your messages are being received by the move_base node.