TurtleBot 2 Navigation Exercises

DISCLAIMER: This exercise in Gazebo is a low quality example of the autonomous navigation of a real TurtleBot. Better TurtleBot simulation controllers are needed for Gazebo.

Driving without steering TurtleBot 2

Reference: ROS Robotics By Example, Chapter 4 –Navigating the World with TurtleBot explains TurtleBot navigation (mapping a room and autonomous navigation) for a real TurtleBot 2. These exercises outline the information and commands for autonomous navigation using the TurtleBot Simulator. Pages 175 – 193 in this book will provide a description of the commands and additional information about TurtleBot's autonomous navigation.

Copy the files simulation_map.pgm and simulation_map.yaml from the Github website: https://github.com/FairchildC/ROS-Seminar-files

Take screenshots of the following steps to record your results in a MS Word document.

(Pages 175) To start Gazebo and the TurtleBot 2 simulator, first open a terminal window and type the command:
$ roslaunch turtlebot_gazebo turtlebot_world.launch

If this command gives you an error, add the following line to your .bashrc file:
export TURTLEBOT_3D_SENSOR=kinect

Use the **pwd** command in the directory where the simulation_map files are stored to find the path to that working directory. Open the simulation_map.yaml file in the editor (gedit) and change the first line of the file **to contain the path to your simulation_map.pgm file**. Then save and close the file.

In a second terminal window, start the amcl operation: (REMOVE <CR>)
$ roslaunch turtlebot_gazebo amcl_demo.launch
map_file:=/<pwd_path_to_files>/simulation_map.yaml

Look for the following text on your window:
odom received!

Open another terminal window and view navigation on rviz:
$ roslaunch turtlebot_rviz_launchers view_navigation.launch
The rviz window should appear.

Orient the rviz map to align with the Gazebo World!

TurtleBot should be told its current location on the map as it corresponds to the Gazebo environment. Click the **2DPose Estimate** button on the top toolbar of rviz and then click on the location on the map where TurtleBot should be located. Orient the **large green arrow** by dragging the cursor in the direction TurtleBot is facing.

To get TurtleBot to navigate autonomously across the map, click on the **2D Nav Goal** button on the top toolbar. Next, click on the map location where you want TurtleBot to go and drag the green arrow in the direction TurtleBot should be facing when he reaches that location.

(In Gazebo, TurtleBot 2 acts like a "drunken sailor" as it tries to reach its goal – so be patient. It is normal for it to spin as it progresses down the path.)

Navigating to a designated location

Kill all processes and close all windows to restart Gazebo and TurtleBot 2 Simulator.

(page 183) In the first terminal window, type the command:
$ roslaunch turtlebot_gazebo turtlebot_world.launch

Open another terminal window, check the initial pose of TurtleBot:
$ rostopic echo /odom/pose/pose -n1
See that TurtleBot is essentially at (0,0) and facing in the x direction of Gazebo.

In a third terminal window, launch gmapping_demo:
$ roslaunch turtlebot_navigation gmapping_demo.launch

Open another terminal window and move TurtleBot ahead about 1 meter by typing the command: (Tab complete will help here!)
$ rostopic pub /move_base_simple/goal geometry_msgs/PoseStamped
"header:
seq: 0
stamp:
secs: 0
nsecs: 0
frame_id: 'map'
pose:
position:
x: 1.0
y: 0.0
z: 0.0
orientation:
x: 0.0
y: 0.0

z: 0.0
w: 1.0"

(Sometimes TurtleBot does not move or is slow to move. Be patient. Clicking on the Gazebo window or TurtleBot will refresh the screen.)

In the second window, check TurtleBot's final pose with the command:
$ rostopic echo /odom/pose/pose -n1

See that TurtleBot moved approximately 1 meter.

Navigating to waypoints with a Python script using a map

Kill all processes and close all windows to restart Gazebo and TurtleBot 2 Simulator.

Copy the file MoveTBtoGoalPoints2.py from the Github website:
https://github.com/FairchildC/ROS-Seminar-files

(page 185) In the first terminal window, type the command:
$ roslaunch turtlebot_gazebo turtlebot_world.launch

Open another terminal window, launch **amcl** operation: (REMOVE <CR>)
$ roslaunch turtlebot_gazebo amcl_demo.launch
map_file:=/<pwd_path_to_files>/simulation_map.yaml

Next, open a third window and launch rviz and display the map:
$ roslaunch turtlebot_rviz_launchers view_navigation.launch

Open a fourth window to display TurtleBot's initial pose on the map. Type the command:
$ rostopic echo /initialpose

When you use the **2D Pose Estimate** button in rviz to indentify TurtleBot's initial position and orientation, information on TurtleBot's initial pose will appear in the **echo /initialpose** terminal window.

Find other points on the map by using the **Publish Point** button on the rviz toolbar, type the following command:
$ rostopic echo /clicked_point

Then use the **Publish Point** button and click on a point on the map.

The goal locations of (0, -2..0) and (-2.0, 2.0) have been selected on the map in Gazebo and used in the Python script MoveTBtoGoalPoints2.py. Other locations can be selected as you wish. Execute this script with the command:
$ python MoveTBtoGoalPoints2.py

C heck the robot's final position on the map by typing the command:
$ rostopic echo /amcl_pose

To view only the position and orientation, use the command:
$ rostopic echo /odom/pose/pose -n1

Show a comparison of the final position that amcl displays with the final position odom displays.

Tested 11/1/2018