ME 134 - Lab #2 Robot Navigation

Released: Monday 04/30/2018 (80 points) Due: Wednesday 05/09/2018

In this lab we will be using the navigation stack in ROS to integrate localization, mapping, and path planning to navigate our Turtlebot. We will be using the move_base ROS package named integrate all our data and do path planning, the gmapping package to do the mapping, odometry to do the localization. The goal of this lab will be to specify a goal point for the robot, and to have it autonomously navigate to this goal region.

1 Setting up the Turtlebot and the Navigation Stack

Start up all the turtlebot hardware, sensors, mapping, and localization.

1. Bring up the Turtlebot. As you saw in the mobile_base.launch.xml file, this has been modified to launch the move_base node.

   $ roslaunch turtlebot_bringup minimal.launch

2. Run the laser scanner. In ROS Kinetic, the command is slightly different as shown below. Be sure to set permissions for the laser scanner as you did in the previous lab.

   $ sudo chmod a+rw /dev/ttyACM0

   $ rosrun hokuyo_node hokuyo_node

3. Setup the transforms for the laser frame using tf_setup.py. You can find the python script here.

4. Run the amcl node in order to localize the turtlebot within the generated map. The amcl node uses odometry and laser scans to localize the turtlebot.

   $ rosrun amcl amcl _use_map_topic:=true _base_frame_id:=base_footprint

5. Run the slam_gmapping node in order to build the map of the environment.

   $ rosrun gmapping slam_gmapping _xmin:=-10 _xmax:=10 _ymin:=-10 _ymax:= 10 _map_update_interval:=5

6. Check your rqt_graph and transform tree at this point. Make sure everything is connected as it should be. If not, redefine topics being published/subscribed to, or redefine your transforms. Once they are fixed, save the rqt_graph and transform tree for your lab report.

2 Set Up the OptiTrack System

1. Follow the instructions on the OptiTrack instruction sheet to launch the vrpn client and get OptiTrack communicating with your ROS system.

2. Confirm that the OptiTrack system is properly communicating by checking the rqt_graph and transform tree.

3 Send a Message to the Navigation Stack

Now that the navigation stack is set up, it is awaiting goal commands from the user (you). Once it receives the goal, it should (in theory) run A* to plan a path to that goal point and execute the resulting plan.

1. Create a folder for your group, and navigate to that folder. Start recording a bag file from that folder.

   $ rosbag record -a

2. Run the node you wrote for the pre-lab for sending a goal message to the navigation stack. If you were unable to complete that portion of the prelab, we will have a node at the lab for you to run. Initially avoid any obstacles in your first attempt.

   $ python *your_code.py*

3. Change the commanded coordinates to move around any obstacles in the arena, and build up a map of the arena.

4. Send a command that sends the turtlebot through one of the obstacles.

5. Stop recording for your bag file, and check that the expected contents are there.

   $ rosbag info *your_bag_file*

4 Lab Writeup

If everything worked as expected during the lab, the writeup will be very short. If issues arose, you will be expected to comment on those issues and explain why they arose and how you would remedy those issues. The lab writeup should answer the following five questions:

1. Print out the rqt_graph that resulted. Are there any issues with how the nodes/topics are connected? If so, comment on what those issues are and how you would change them (be specific).

2. Print out the transform tree that resulted. Are there any issues with transform tree? If so, comment on what those issues are and how you would change them (be specific).

3. After sending your message to the goal, did the Turtlebot execute the expected plan? If not, print out both the transform tree and rqt_graph to show that the messages were being properly sent/received.

4. If you encountered any other issues during the lab, comment on what those issues were and why they arose. The goal here is to convince us that you understand the underlying processes enough to debug the issue. If you did not encounter any issues in the lab, you may skip this problem and get full credit.

5. If your turtlebot experience any collisions with obstacles during the lab, give the reasons why (be specific).

6. What is the default topic that the move_base node publishes to in order to drive the Turtlebot? What topic did it actually publish to in order to drive the Turtlebot in the lab (hint: look into the updated move_base.launch.xml file).