

Turtlebot Tele-operation

Author: Sterling McLeod

June 20, 2016

Introduction

The assignment will train you to move the Turtlebot 2 platform via velocity commands. First, you will start processes on the Turtlebot by using a *launch* file. After the Turtlebot has been started, you will publish a specific msg, *geometry_msgs/Twist*, to drive the robot. Lastly, you will design a basic keyboard interface for a user to operate the Turtlebot through a keyboard. By doing all of this, you will become more familiar with publishing and subscribing messages and how to structure a ROS node. To complete this assignment, you should already be familiar with creating a ROS package, building a ROS node, and how to publish and subscribe to messages.

System Info

Operating System, ROS Version: Ubuntu 14.04, Indigo

Supplemental Material

Visit <http://wiki.ros.org/ROS/Tutorials> for a list of tutorials on ROS. For this assignment, the following tutorials will be most relevant:

1. Understanding ROS Nodes
2. Understanding ROS Topics
3. Creating a ROS msg and srv
4. Writing a Simple Publisher and Subscriber (C++), Writing a Simple Publisher and Subscriber (Python)
5. Using `rqt_console` and `roslaunch`

Tasks

Task 1. Run and observe the keyboard tele-operation provided by ROS..

1. Change permissions on the interface between the robot and computer: "sudo chmod 777 /dev/ttyUSB0"

2. Run the "minimal.launch" file in the package turtlebot_bringup
3. Drive the robot around with the keyboard interface.

Task 2. Create a ROS package implementing keyboard tele-operation.

1. Accept keyboard commands for the following:
 - (a) Move forward or backwards for X number of seconds at a fixed speed
 - i. Limit the time to 2 seconds
 - (b) Move forward or backwards X distance
 - i. Limit the distance to 0.33m
 - (c) Turn 90 degrees clockwise and counter-clockwise
 - (d) Turn X degrees
 - i. Limit to 90 degrees
 - (e) Change the linear and angular velocities of the robot
 - i. Limit the linear velocity to $0.33\frac{m}{s}$ and angular velocity to $0.52\frac{rad}{s}$.
 - (f) The keys used for tele-operation do not need to match the keys used in the keyboard_teleop.launch file.
 - (g) The robot does not need to stop smoothly.
2. Create a launch file to run all nodes necessary for your keyboard tele-operation node.
3. Run *rqt_graph* with your launch file running and save an image of it (there's an icon in the top-right of the window for this).
4. Create a README file explaining how to run your code.