CSC 415/515 Robotics
Project 2

Due: Beginning of the Final Exam

1 Overview

You are to set up ROS to teleoperate a robot and track its position in world coordinates.

1.1 Teleoperation

The ROS keyboard teleop program works well (see this wiki page: `http://www.ros.org/wiki/turtlebot_teleop/Tutorials/Teleoperation`), but it has the annoying issue of having to keep pressing keys to move the robot. Design a ROS keyboard teleop node that uses the IJKLM keys in the following way:

1. I,M: increase / decrease forward velocity
2. J,L: bump the robot's angle to the left / right
3. K, space: stop the robot

Your node must continuously broadcast movement commands. You can use the teleop keyboard node in the turtlebot package as a template, if you wish. Note that you should teleop from your laptop, not by ssh'ing into the turtlebot.

1.2 Transform for World Coordinates

Write a node that outputs a TF frame with the following properties:

1. Stays fixed in the odom frame
   Let's call this frame `my_frame`. You want to publish a transform between odom and `my_frame`. Normally this transform will be static, that is, will not change over time. You need to publish the transform continuously, though, to keep it current.
2. Whenever the "t" key is pressed, the world coordinate system moves to the current pose of the robot
   When the key is pressed, `my_frame` "jumps" to a new position in the odom frame, by publishing the new position given by the current position of the robot in the odom frame.
3. Display a line from the frame to the robot's base frame in RVIZ

It helps to make the fixed frame of rviz the odom frame, since you can see how the robot and `my_frame` are registered to odom.

2 Project Requirements

Your project will be graded based on the source code that you turn in (50%), and the quality of the paper that you submit (50%).

1. Source code turned in for review
   (a) Source code should clearly show the team name and the participants.
   (b) Include instructions on how to run your code on the robot and simulator. Launch files are the best way to go. Your instructor will be happier if it is easy to run.
   (c) Source code should be readable, with comments liberally sprinkled to help guide the reader. We will be reading the code, so the easier you make it on me, the easier it is to give a good grade.
2. A well formatted paper (2-6 pages) is required. The paper should detail the algorithm used, any special techniques used, and how it is implemented.