RobotLab 1: Teleop

ENGN 4627/6627 Robotics 2018 S2

July 26, 2018

1 Purpose

This lab is meant to get you started with the TurtleBots that you will be using for the rest of the semester.

In this lab, you will first learn how to bring up a turtlebot, test its DEMO programs. You will also learn how to compile from the source code of the Demo program.

After this, you will be asked to modify the source code, to see how the DEMO program can be changed to have different functions or behaviours.

2 Preparation

This lab assumes that you're comfortable with ROS and with writing Python code.

If you're not already comfortable with ROS or Python, you should probably look over the relevant tutorials again (Tutorial 1 and Lab 0), to make sure you understand the core concepts.

3 Task 1: Basic teleop DEMO and compilation

1. Log in to your TurtleBot.
2. Work your way through some of the TurtleBot tutorials . Specifically, do the following tutorials:
   - TurtleBot Bringup found here `http://wiki.ros.org/turtlebot_bringup/Tutorials/indigo/TurtleBot`
   - TurtleBot Teleoperation found here `http://wiki.ros.org/turtlebot_teleop/Tutorials/indigo/Keyboard`

3. Launch Teleop and get familiar with its behaviours.
4. Locate *turtlebot_teleop* package and the source code for teleop python. Create a new package using source codes and launch file from *turtlebot_teleop*, change the package name and type & name of the node, and recompile. Repeat the above test.

Hint: use rospack find and rospack libs-only-L

```
# Create a workspace:
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws/
$ catkin_make
$ source devel/setup.bash

# or having this automatically run every time a new terminal is
    opened
$ echo source ~/catkin_ws/devel/setup.bash >> ~/.bashrc

# Create a package:
$ cd ~/catkin_ws/src
$ catkin_create_pkg <package_name> [depend1] [depend2] [depend3]

# Modify the package and compile:
$ cd ~/catkin_ws
$ catkin_make

# Run:
$ roslaunch ...
# or
$ rosrun ...
# Note: rosrun requires the rosserver (via roscore or roslaunch)
    to be active first
```

4 Task 2: Tweaked Teleop program*

(* This project idea is courtesy of Konolige, @ Stanford University, for course of CS225B Robotics.)

The above teleop program works well, but it has the annoying issue of having to keep pressing keys to move the robot.

Your task is to design a ROS keyboard teleop node that uses the IJKLM keys in the following way:

i I,M: increase / decrease forward velocity

ii J,L: nudge the robot's angle to the left / right

iii 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 via ROS from your laptop, do not control the turtlebot using a ssh link.

NOTE: Your robot programs need to constantly check the readings of your turtlebot's **bumper sensor** and **cliff sensors**, and should play an **alarm** sound as a warning signal and **stop** the turtlebot, should any of the sensors be triggered.

5 What to Hand In (due in Week 3)

Pack your source code and your Lab report (3 page A4) in a single Zip file. Name your zip file in the following format:
" ⟨**your_last_name**⟩_⟨**your_uni_id**⟩_**Lab**_⟨ **X** ⟩**.zip** ".

Upload your single zip file to Wattle, at the corresponding Lab upload link. Since this lab is done by your group, the source code you hand in will be very similar if not identical.

However, your Lab Report must be your own **individual work**, which should not be similar to your group mates'. Your overall Lab marks will be based on the performance of your DEMO, the clarity of your source code, and the quality of your Lab Report.

The Lab Report will be based around answering some technical questions. These reports do not have to be formal, but are important to test your understanding of the work done in this lab.

The report will be marked based on your answer, which should demonstrate your understanding of the subject asked. There will be 4-6 questions, and each question contributes 0.5-2 marks, the demonstration 4 marks, while the clarity of source code contributes to the remaining 1 marks.

In summary: your demonstration + the clarity of your source code (5 marks), and correctness and quality of your 3 page lab report (5 marks). In total, these make up 10 marks as the full mark for each lab.

DUE DATE:

Lab1 lab demonstration will be due in the last 1.5 hours of Week 3's Lab sessions.

The lab report (see next page) will due on Sunday evening of the same week.

6 Turtlebot Lab 1 Report

In your Lab 1 Report, please answer the following 5 questions.

(Note: Lab Report must be your individual work.)

Question 1: Please briefly describe the steps that you have used to create and compile a new teleop Python package (please list and comment on each step) (1 mark)

Question 2: What is the purpose of a ROS callback function for a topic subscriber, and when is it called? (0.5 mark)

Question 3: When you code a program for your turtlebot, you are asked to constantly check the bumper and cliff sensor readings. Why so? (1 mark)

Question 4: Wheels and Legs are the two most popular choice of robot locomotion mechanisms. Please list three Pros and three Cons for each of these two types of locomotion, and give your brief explanations. (1.5 mark)

Question 5: What are the four basic types of wheels? Which one (or ones) can generate omni-directional motion? (1 mark)