Last modification: October 22, 2020

RBE3002: Unified Robotics IV
D-Term 2020/2021
Lab 1: Introduction to ROS

Introduction

The laboratory activities across the course are aimed at teaching you how to implement algorithms for mobile robot path planning, localization and navigation. The main objective of Lab 1 is to familiarize you with the Robot Operating System (ROS) and the TurtleBot. This lab will serve as a first building block for the upcoming assignments and the final project. This lab is to be completed individually, and it is expected that you demonstrate a working knowledge of ROS to be used in the course. Starting from Lab 3, you will work in groups.

Objectives

Upon successful completion of this lab, you will be able to:

- Demonstrate knowledge of ROS nodes, messages, topics.
- Demonstrate knowledge of ROS tools.
- Demonstrate knowledge of using the TurtleBot in the Gazebo simulator.

Before You Start

Learning Linux

Ubuntu is the standard operating system used with ROS, and the one we will use in this course. Aside from learning to use ROS, learning to use Ubuntu is a valuable skill for all RBE majors due to its wide adoption in industry and academia. We will be using Ubuntu 18.04 and ROS Melodic. If you would like to install it on your own laptop to dual-boot with your current OS, you can find easy-to-follow instructions on the Ubuntu website. If you are new to Linux, there are many online tutorials to help you get started. Some useful references:

- Settling into Unix: `http://matt.might.net/articles/settling-into-unix/`
- A survival guide for Unix newbies: `http://matt.might.net/articles/basic-unix/`
- Tips, tricks, and tools Linux/Unix: `http://matt.might.net/articles/best-tools-for-using-and-learning-linux-and-unix/`

Other Linux/Unix tutorial for beginners:

- Unix: http://www.ee.surrey.ac.uk/Teaching/Unix/
- Ubuntu 18.04 help page: http://releases.ubuntu.com/bionic/
- Using the Terminal: https://help.ubuntu.com/community/UsingTheTerminal

Lab Work

Go to the ROS tutorials page: http://www.ros.org/wiki/ROS/Tutorials. Note that there are different tutorials for different versions of ROS (Hydro, Indigo, Kinetic, etc). Make sure you follow the ones for ROS Melodic and catkin. There are also tutorials for both C++ and Python; in that case, do the Python tutorials.

1. Complete the Beginner Level ROS tutorials 1–6, 12–13
2. Complete the Intermediate Level ROS tutorials 3–5.
3. Complete the TF tutorial at http://wiki.ros.org/tf/Tutorials.
4. Go to the Turtlebot3 website: http://emanual.robotis.com/docs/en/platform/turtlebot3/overview/ and complete the following Turtlebot3 tutorials. Read through the Lab 1 questions document before completing the Turtlebot3 tutorials. Note that you will need to put the model:=burger tag at the end of each command line, for example:

   ```
   roslaunch turtlebot3_fake turtlebot3_fake.launch model:=burger
   ```

   (a) Tutorial 6: Setup: http://emanual.robotis.com/docs/en/platform/turtlebot3/setup/#setup Follow only Sections 6.1.2 and 6.1.3, which are required for the simulated Turtlebot. The subsequent section about networking is for the real robot, which we won't use. **Make sure to replace "kinetic" with "melodic" in all the commands. Also, we're using Ubuntu 18.04, not 16.04, so ignore the comments that say that 16.04 is recommended.**

   (b) Tutorial 11: Simulation http://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/#simulation

   (c) Tutorial 8.2: Teleoperation (keyboard), make sure you can teleoperate the robot. http://emanual.robotis.com/docs/en/platform/turtlebot3/teleoperation/#teleoperation

   We encourage you to explore and see what else you can find, there are several packages that come with the Turtlebot that will serve as useful examples for future labs.

5. One of the nice aspects about ROS is that it makes it easy to switch between simulation and the robot. A lot of your debugging can happen in simulation using Gazebo.

   (a) First, try running the simulator by following the instructions in Tutorial 11

   (b) Now in a new terminal start the teleop node you used

   ```
   roslaunch turtlebot_teleop keyboard_teleop.launch
   ```

   (c) Make sure you can control the robot, since you will use this extensively in the next lab!

Deliverables

1. All sign offs (online/in-person)

Signoff	Description	Points
Bring up the robot in Gazebo	Launch the robot in Gazebo and show it in *rviz*	25
Control the robot	Use the keyboard to drive around in the robot in simulation	25
Move the robot with a message	Use the command line to send a message to move the robot	50

2. Completed Lab 1 Questions document

Grading Rubric

[100 points] All pre-lab and in-lab procedures are completed and demonstrated prior to the deadline. All deliverables and sign-off sheet are submitted on time. The code is well commented and structured. Upload your completed code to a gist and share the link. You will need to submit 4 images of data:

1. Robot in the original position in Gazebo.
2. Robot in *rviz*.
3. Robot in the another position in Gazebo.
4. ROS message sent to the robot.

You will lose 20 pt for every picture of a screen (i.e. you use your phone to take a picture of your screen). You will also lose 5 pt for every screenshot you submit. There are proper ways to get image data out of *rviz* and *Gazebo*: show us you know how to use them.
Point Breakdown:

- 50% → questions
- 50% → signoffs

Some Tips to Make Your Life Easier

Setting Your Environment Variables Automatically

When you open a new terminal, Bash creates a new *environment*. An environment is a collection of variables, such as `PATH`, that help Bash and other commands to do their job. ROS makes heavy use of environment variables, and, if these variables are not set correctly, it won't work.

To check if the variables are set correctly, run this command:

```
$ set | grep ROS
```

If the output of this command is empty, then the variables have not been set and ROS won't work.

To set up your environment, every time you open a new terminal, you should issue the command:

```
$ source ~/catkin_ws/devel/setup.bash
```

but this gets tedious quickly and it's easy to forget. A better approach is to have Bash run that command automatically upon opening a new terminal.

To tell Bash to execute that command automatically, use a text editor and create/open the file `~/.bashrc`, for instance with the following command:

```
$ gedit ~/.bashrc
```

Make sure that the last line of this file is (assuming you followed the tutorials and used `~/catkin_ws` as your Catkin workspace):

```
$ source ~/catkin_ws/devel/setup.bash
```

Save and close the file. Next, open the file `~/.profile` and make sure it contains somewhere the following lines:

```
if [ -f ~/.bashrc ]; then
  . ~/.bashrc
fi
```

If the file does not exist or these lines are missing, create the file and add those lines.

Now open a new terminal. The output of `set | grep ROS` should be correct.