Lab 1a
Introduction to ROS

OVERVIEW
In this lab, you will learn about the Willow Garage ROS software, calibrate your robot, and create a tool for characterizing odometry performance.

TASKS
Computer setup. This semester, your group will be assigned one of seven TurtleBots to work with. The robots are all named after fictional turtle characters. Each robot has a laptop associated with it, with the robot name labeled at the top of the keyboard.

From time to time, it will be useful to have another computer to work with while you are completing the labs, in order to run software remotely on the robots. Your personal laptops will suffice, but you can also use another robot laptop if it is not currently in use.

Therefore, each laptop will have two user accounts set up. The “turtlebot” account is for running the onboard robot software, and the “remotebot” account is for running software on a robot remotely and performing other miscellaneous tasks. Your group should choose a password for the turtlebot account on your robot’s laptop, and share the password with the class wizard, and with the professor.

ROS Tutorials. Load the ROS tutorial page at http://www.ros.org/wiki/ROS/Tutorials. Along with your group, and taking time to make sure that everyone is following what is going on, complete at least the following tutorials:

- 2. Navigating the ROS Filesystem
- 3. Creating a ROS Package
- 5. Understanding ROS Nodes
- 6. Understanding ROS Topics
- 7. Understanding ROS Services and Parameters
- 11. Writing a Simple Publisher and Subscriber (Python)
- 13. Examining the Simple Publisher and Subscriber

For tutorial 3, create your package in the ~/ros directory, which should already exist and be on the ROS_PACKAGE_PATH.

It’s probably a good idea to take turns reading over each tutorial, performing the steps, and explaining the material to your group members. There will be an in-class quiz on the material in the tutorials, so make sure that everyone understands what’s going on!
Modify the simple publisher and subscriber. Make a new program, reverser, that subscribes to messages on the chatter topic, and publishes the same message, backwards, on a topic named rchatter. For instance, if the talker program publishes the message

```
hello world 123
```

the reverser program should publish the message

```
321 dlrow olleh
```

You should be able to verify that the reverser program is working by running “rostopic echo /rchatter”.


Verify the calibration. Write a program to listen to the joystick node (see the joystick teleop tutorial: [http://www.ros.org/wiki/joy/Tutorials/WritingTeleopNode](http://www.ros.org/wiki/joy/Tutorials/WritingTeleopNode)). Each time a particular button is pressed on the joystick, the robot should drive forward exactly 1 meter (or as close as you can get), and subsequently turn 90 degrees to the right (or as close as you can get). Starter code will be provided.

Show the professor your program working with your robot in an uncalibrated and a calibrated state.

WHAT TO TURN IN

You should schedule a demonstration with the professor to show:

- completion of the ROS tutorials, and knowledge of their contents
- your finished reverser node
- your drive_square program working both with and without calibrated odometry

You should also use the git software to submit the code you have written for this lab. Instructions on how to submit will be provided.