CS/ME/ECE/AE/BME 7785

Lab 0

Before you get started...

When released, the robots we work with in this course (Turtlebot 3), originally used Ubuntu 16.04 and ROS Kinetic Kame but, since those are are reaching their EOL, we will be switching to Ubuntu 20.04 and ROS Noetic Ninjemys. Be careful when following Turtlebot3 guides blindly as they were originally written for ROS Kinetic Kame.

Ubuntu is the standard operating system used with ROS, and the one we advise you use in this course. We will be using **Ubuntu 20.04**. If you don't have this Ubuntu version installed already, then you have several options:

- dual-boot your laptop to run both Ubuntu and your current OS **(best option)**
- set up Ubuntu on a virtual machine. There will be significant challenges to making this work well. For MacOS some have reported good results using Parallels, but again, use this approach with care, our two GUI tools (rviz and Gazebo) may not work well.
- use one of the lab desktops we provide in Klaus 1209. This is the least flexible solution since it relies on a shared machine.

Note 1: while ROS has some limited functionality on Windows, it won't work well enough to complete this course, so this is not a valid option.

Note 2: As an alternative to Ubuntu 20.04 / ROS Noetic, you could try an older combination: Ubuntu 18.04 / ROS Melodic Morenia. We will have a few robots available set up with 18.04 to support those who need it.

Note 3: We are using ROS 1 for this course which does not support anything newer than Ubuntu 20.04.

Lab

1. If you are new to ROS, read the ROS introduction page: http://wiki.ros.org/ROS/Introduction.
2. Install ROS using the instructions at http://wiki.ros.org/noetic/Installation/Ubuntu. During step 1.4, this course recommends you perform a Desktop-Full Install. Make sure you are using **ROS Noetic** and not another version!

3. Complete the beginner ROS tutorials (1-12) found at http://wiki.ros.org/ROS/Tutorials. Make sure you are using the **catkin** build system. There are tutorials in both C++ and Python. Feel free to use either language, however note that ROS Python code is typically easier to understand and debug (for your instructors as well as in general). C++ is typically performs faster and has some advantages (like Nodelets), but we will not be doing anything requiring that kind of optimization.

4. Complete intermediate tutorials number 2-5 http://wiki.ros.org/ROS/Tutorials.

5. Next install the Turtlebot3 Packages. There are directions here https://emanual.robotis.com/docs/en/platform/turtlebot3/quick-start/. **Note this page defaults to ROS Kinetic, make sure you click the button at the top for Noetic and follow the correct Noetic directions.** You'll know you're on the right page if your instructions include commands like,

   ```
   sudo apt-get install ros-noetic-joy ros-noetic-teleop-twist-joy
   ```

   The next two steps involve simulation. If your laptop cannot handle the simulation environment or there are other errors let us know.

6. Now that you're starting to grasp at ROS, it is useful to be able to test scripts you write in a simulation environment before deploying them on the hardware in the lab. The people working on the Turtlebot3 have released a great simulation package for Gazebo (http://gazebosim.org/). Follow the tutorials in Step 6 of the quick start guide found at https://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/ make sure to install and test the Gazebo simulator (step 6.1) and *fake_node* (step 6.4). **Note again these pages defaults to ROS Kinetic, make sure you click the button at the top for Noetic and follow the correct Noetic directions.**

7. Make sure you can run the *turtlebot3_teleop* example unsing the Turtlebot3 fake node and Gazebo simulator. The simulated robot should move around based on your keyboard command!

Submission

There is nothing to submit for this lab, but please post any issues or questions you run into during this process in Slack. While this lab does not count towards your grade, it is the foundation for the rest of the course. If you are having problems, please avoid waiting until Lab 1 to address them.