Using ROS for ROB521 Labs

Welcome to ROB521 labs Winter 2024! The labs in this course are completed on the TurtleBot 3 Waffle Pi robot, with communication handled using the Robot Operating System (ROS). There are two major components for each lab, simulation and experiments with the simulated and physical TurtleBot 3. For both simulataion and experiments, we use Robot Operating System (ROS) for communication. Simulation is handled by Gazebo, which comes bundled with ROS installations. You are expected to finish implementation and tests in simulation before your lab session.

This document will provide preliminary information for how you can set up ROS with simulation for your group. Note that we will only be using **ROS1**, not the newer (and less tested) ROS2. For the remainder of this document, any time we refer to ROS, assume that we mean ROS1 unless we state otherwise. For this course, we recommend, and will only officially support, using **ROS Noetic Ninjamys**.

ROS1 only natively runs in Ubuntu, and although installing it usually isn't too difficult, it can be a bit tricky if you're not comfortable with linux or have never used ROS before. Whether or not you're running Ubuntu, there are many options for installing and using ROS, so we're going to go through some of them here.

1. The Easiest Option -- Use Our Campus Computers

Difficulty: ★

Software required

- A browser
- VSCode (optional but highly recommended)

Who should use?

Anyone who wants to get setup with minimal installation, or tried some of the other options and found that they didn't work for their particular setup. Also, this is the **only setup option** that we will fully support if there are technical issues.

Limitations

- Only one member from each group can be running a simulation at a time.
- If two groups are using a single machine at the same time, it may run a little slow.
- Few options for customizing the ROS install or the box that it's installed in.

How does it work?

Our lab computers already have Ubuntu 20.04 and Ros Noetic installed on them. We will provide each group with an IP address and a password, and you will be able to use your browser to log into these computers using VNC. You will be able to edit your code either on your machine using VSCode set up for remote editing (recommended), or directly in the VNC session with code editors available on the remote machines (not recommended).

2. Install Ubuntu

Difficulty: ★★★★★

Software required

- An install drive with Ubuntu on it
- ROS (See note on versions below)
- The Turtlebot3 simulation packages -- specifically turtlebot3, turtlebot3_simulations, and turtlebot3_msgs
- Code Editor

Who should use?

Anyone who really wants to get their hands dirty, and expects to be working with robots and/or ROS more in the future. We can basically guarantee that this will take the most upfront investment from you if you choose to go this route, but will also allow you the most options for customization and running things on your own. Plus, basically all software development in robotics (and machine leanring, if that interests you) is done in Linux, so you'll be doing yourself a favour in terms of future preparation.

Limitations

- Large initial setup effort investment.
- Minimal technical support from us.

How does it work?

Since ROS natively runs on Ubuntu, every other option in this guide is basically a watered down version of doing this anyways. You don't even need to throw away your existing operating system -- you can dual boot Ubuntu with both Windows and macOS. There are a zillion tutorials out there on how to do this, but beware -- this is often a bit of a bumpy process.

Note on ROS versions: ROS versions are tied to Ubuntu versions . ROS Noetic is primariy targeted at Ubuntu 20.04. For our course, we're going to assume that you have access to ROS Noetic (which is tied to Ubuntu 20.04).

If you do choose to install Ubuntu yourself, and you don't install 16.04, we also recommend that you use the provided Docker image (or another similar image) with Ubuntu 16.04 + ROS Kinetic. Although you *should* be able to run all of the labs from this course in ROS Noetic, this is untested, and we won't have the bandwidth to support this.

Resources

- Windows dual boot tutorial
- Mac dual boot tuturial

3. Virtualization with VMWare

Difficulty: ★★

Software required

- (Windows, Linux) VMWare Player -- free for non-commerical use
- (macOS) VMWare Fusion -- free for non-commercial use, requires sign up for an account
- 4GB of free space for the initial zip file download, and an additional 10GB of free space for the virtual OS once it's unzipped

Who should use?

Anyone who wants to run ROS on their own machine (as opposed to on ours), but wants the less painful installation option.

Limitations

- Large amount of free space required.
- Lower performance than Docker Virtualization.
- Minimal technical support from us.

How does it work?

VMWare Player and VMWare Fusion are both versions virtualization software that allows you run a different operating system (OS) inside of your main (or host) OS.

Resources

There are many tutorials online. As an example,

- How to install ubuntu 20.04 on windows using vmware workstation player
- How to install ubuntu 20.04 on VMware Fusion for Mac

4. Virtualization with Docker

Difficulty: ★★★

Software required

- Docker
- (If you have an Nvidia GPU) Nvidia-docker
- (Windows) Windows Sub-system for Linux (WSL)
- 5-10GB of free space (depending on image used)

Who Should Use?

If you want to run ROS on your own system, and want something that will probably perform better than the VMWare Virtualization option, *and* you want to make sure you're using the recommended ROS version for the course.

Docker is quickly becoming the virtualization tool of choice for most professional applications, so learning how to use it isn't a bad idea. That being said, we don't have the bandwidth to support docker-based installations beyond basic issues, so YMMV.

Limitations

- Large amount of free space required.
- More difficult to set up than VMWare Virtualization.
- Minimal technical support from us.

How does it work?

Similar to the VMWare option, Docker creates a "virtual" version of Ubuntu running inside of whatever your host OS is. Unlike VMWare, when you run a Docker container, you don't automatically see a full desktop, which typically ensures that you only use the computer resources that you absolutely need for whatever your application is, meaning you tend to get better performance in Docker than in VMWare. Docker also gives you far more options for running smaller applications. Docker is a pretty powerful and versatile piece of software, and the details of all of the things that you can do with it is beyond the scope of this document, but you should definitely check it out.

Resources

- Offical ROS repo on the docker site
- Official ROS tutorial on using docker

> Written with StackEdit.