RBE 500 — Foundations of Robotics

Instructor: Siavash Farzan

Spring 2022

ROS Assignment 0: Setting Up Robot Operating System (ROS)

Note: This assignment must be completed before the start of ROS Assignment 1 as you will not be able to begin work on Assignment 1 if your Linux Machine is not properly set up.*

0.1 Objectives

By the end of this assignment, you should have Robot Operating System (ROS) Noetic Ninjemys installed on a Linux Ubuntu 20.04 machine. Ubuntu could be installed as a virtual machine if needed.

0.2 General Information

Here are some general ideas that can help you get prepared for this assignment.

When working with robots, we need skills in Linux, ROS, Python, Git/GitHub, and Virtual Machine (VM).

- The best way to learn Linux is to spend time playing with it. It's just like the first time you had your Windows/Mac computer. Additionally, you may follow some tutorials online and try those commonly used commands in terminals. For example, it is recommended to go over chapter 1-3 of this tutorial: `http://swcarpentry.github.io/shell-novice/`
- https://www.ros.org/ is a Linux software library specifically designed for programming on robots. It has some features of OS, but is not actually an independent OS. With ROS, people do not need to worry about low-level communications and keep reinventing the wheel. On top of ROS, developers all over the world can work on their own software packages, and contribute to ROS community. These packages are similar to those libraries that we "import" in Python.
- A good way to learn ROS is to learn from ROS wiki (`http://wiki.ros.org/ROS/Tutorials`), which provides official tutorials. In addition, there is a reference book *A Gentle Introduction to ROS* by Jason M. O'Kane (free online: `https://www.cse.sc.edu/~jokane/agitr/`), which introduces useful design ideas behind ROS. Note that examples in this book and some tutorials on ROS wiki are written in C++. Please focus on high-level design ideas and rospy library only (not roscpp), since we recommend to use Python (instead of C++) in this class.

*Parts of this assignment are adapted from the course materials by Professor K. Karydis at UC Riverside.

- For Python, basically you need to have a rough idea about data structures, operators, flow control, etc. There are also many good tutorials online. For example, the tutorial on `W3Schools`. Going through the first 20 sections (until Python Functions) would be sufficient for this class.
- Git is a version control tool and GitHub is a website (or company) that offers Git-based version control service. It's good to learn Git in the sense that you can better manage your code. With Git, you can see all your change history, and have backups of each version. Many ROS packages that we are going to use in this course are hosted on GitHub. However, it's not strictly required in this class. Going through chapter 1-5 of this tutorial might be helpful: `http://swcarpentry.github.io/git-novice/`
- For Virtual Machine, there are mainly two kinds of software available online. One is VMware and the other is VirtualBox. The former has better utilization of GPU and hence supports better graphics, but it is not free of charge. The good news is that in recent years VMware has released a free "Player" version for individual users, which we will discuss later. On the other hand, VirtualBox is totally open source (free) for all platforms (Windows, Mac, Linux). However, it does not perform well in heavy simulation tasks in Gazebo. (Gazebo is a simulator that we are going to use throughout the course, together with ROS.)

Please familiarize yourself with the above concepts/tools, if they are new to you.

0.3 Overview

For all ROS Assignments in this course we will be using a Linux Ubuntu machine because Robot Operating System (ROS) only works on Linux machines. If you currently have a computer with the Ubuntu 20.04 operating system, or you can dual boot with Ubuntu 20.04 operating system, that's great! This is the best way to work on robots. In that case, you can skip the next two sections of this assignment and continue from Section 0.6.

However, if you have only a Windows or macOS operating system, you have two options:

- Install a virtual machine on your Windows/macOS. This may be a little slow while performing heavy simulation tasks in Gazebo, but will still do the job. (Gazebo is a robot simulator that we are going to use throughout the course, together with ROS.)
- Dual installation (e.g. installing Windows/macOS and Linux alongside each other). There are many resources on the internet on how to do this. (Please be careful about dual boot, since you have to take potential risks.)

In the following, we will go through some basic steps to get our development environment ready.

0.4 Download the Virtual Machine

Virtual machines (VMs) allow users to run an operating system that behaves like a completely separate computer in an app window on a desktop. One or more virtual "guest" machines run on a physical "host" machine. Each virtual machine runs its own operating system and functions separately from the other VMs, even when they are all running on the same hosts[†]. In this course,

[†] `https://www.vmware.com/topics/glossary/content/virtual-machine`

you can use a VM to run an instance of Ubuntu 20.04 in order to use ROS even if you don't have a Linux computer.

When you create a virtual machine, you can shutdown and start the same VM with the same files still stored on it; however, if you delete a VM, you also delete any files you have stored on it.

If you have a Windows laptop, you can download and install VMware Workstation Player, available at:
`https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html`.
Note that VMware Workstation 16 Player is available for free to students for non-commercial use.

If you have a Mac laptop, you can go to VMware Fusion webpage: register under "Use for Free with a Personal Use License" tab and download VMware Fusion Player using a free personal license:
`https://www.vmware.com/products/fusion.html`
For macOS you can use VMware Fusion 12.1.2.

0.5 Install Linux on VMware

Once you have your VMware installed, let's create a new VM and install Ubuntu 20.04. First, download Ubuntu 20.04 disc image from official website (64-bit PC (AMD64) desktop image):
`http://releases.ubuntu.com/20.04/`
You should now have a `.iso` file.

For Windows:

- Open VMware, and select “Create a New Virtual Machine”
  - Choose the `.iso` disc image you downloaded as the “Installer disc image file”
  - Enter some information about this Linux, including Name, User name, and Passworrd
  - Enter a Virtual machine name, e.g. “Ubuntu 20.04”
  - Allocate at least 30GB (preferred 50GB or more) for the Maximum disk size
  - Store virtual disk as a “single” file
  - Under the “Customize Hardware” settings: use the slider to allocate at least 5GB of RAM (i.e. 5120 MB) for better performance
  - Select “Finish”

For macOS:

i) Open VMware Fusion, and select the “Install from disc or image” option.
   - Select “Continue”, and use the Ubuntu `.iso` file you downloaded above as the disc image.
   - Select “Continue”, and use the "easy install" set up with the default settings. Enter a password for your Ubuntu user account, and select “Continue” again.
   - Select “Finish” to launch the virtual machine.

   It will take a few minutes for the Ubuntu 20.04 to be installed on the virtual machine. Once installed, log in to your user account using the password you specified above.

ii) Now shut down the virtual machine (shut down, not pause. Please be sure you are aware of the difference).

iii) Next we want to allocate an appropriate amount of RAM to the VM. With the Ubuntu shut down, from the VMware menu, go to Virtual Machine → Settings...

   - Click on the “Processors & Memory” icon.
   - Under the “Memory” section, use the slider to allocate at least 5GB of RAM (i.e. 5120 MB). You should be able to allocate 5GB of RAM in 8GB systems. On a 16GB machine, you could easily allocate over 8GB (5GB should be sufficient for our course assignments). You will not be able to edit this if your VMware is not shut down.

iv) Finally, we want to allocate at least 30GB (preferred 50GB or more) of Hard Disk to Ubuntu. Under the Settings:

   - Click on the “Hard Disc” icon.
   - Use the slider to increase the Disk size to 30GB or more.

   Note that the disk size 30GB will not be allocated instantly, but will grow gradually as you add more stuff.
   Close the Settings window.

Great. Now you have a (virtual) Linux Ubuntu computer. Take your time and get familiar with it!

0.6 Install ROS

Once you are familiar with Linux, you can start installing Robot Operating System (ROS). In this class, we will be using ROS version Noetic. You will learn more about ROS in the next assignment.

First, in Ubuntu, open up a terminal window (right click anywhere → Open Terminal) and in it run

```
echo "export SVGA_VGPU10=0" >> ~/.profile
```

In general, to install ROS, we need to follow ROS installation tutorial:
`http://wiki.ros.org/noetic/Installation/Ubuntu`

The main steps are:

i) Set up your computer to accept software from packages.ros.org
(**note: the commands here are supposed to be one line, with no break in between. Be careful when copying and pasting into the terminal.**)

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc)
   main" > /etc/apt/sources.list.d/ros-latest.list'
```

ii) Set up your keys

```
sudo apt install curl
curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo
   apt-key add -
```

iii) Update package index

```
sudo apt update
```

iv) Install ROS desktop full

```
sudo apt install ros-noetic-desktop-full
```

v) Make sure your environment variables get correctly setup every time a new shell is launched by adding to your `.bashrc` file. Run the following commands

```
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

vi) Install more dependencies for building packages

```
sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-
   generator python3-wstool build-essential
```

vii) Initialize rosdep

```
sudo rosdep init
rosdep update
```

0.7 Common VMware Issues and Solutions

- If you get the error below every time you boot with VMware:
  `'VMware Cannot Connect the Virtual Device sata0:1'`
  Go to your VMware settings: in the Hardware tab choose CD/DVD, and in Device Status, uncheck the option ”Connect at power on”.

- (For Mac computers) if you get the error below when booting your virtual machine for the first time:
  `'Could not open /dev/vmmon: Broken pipe.'`
  On your computer (not in your VM) go to Systems Preferences → Security & Privacy → General. Under the section “Allow apps downloaded from”, press allow for VMware. Then restart VMware.

- If Gazebo is repeatedly crashing and shutting down, try shutting down the VM and with the VM still shut down, go to Virtual Machine → Settings → Display and uncheck “Accelerate 3D Graphics”. This may cause Gazebo to have more lag but it should also stop it from crashing.

- If you cannot connect to the internet, click the wifi icon at the top right and make sure that “Enable Networking” is checked. If it is already checked, try unchecking and rechecking it. When you are connected, the VM should recognize your connection as an ethernet connection and the wifi icon should be replaced with two arrows pointing in opposite directions (one up, one down). If it still will not connect, try restarting your VM.

0.8 Learn from ROS Tutorials

Once you have ROS Noetic installed, you can follow the tutorials on ROS Wiki and rospy documentation:

```
http://wiki.ros.org/ROS/Tutorials
http://wiki.ros.org/rospy_tutorials
```

Have fun!