CS4278/5478 *Intelligent Robots: Algorithms & Systems*, 2022-23 Semester 2 David Hsu

Lab 1pre

Robot Operating System (ROS) Basics

Instructions

- Recommended completion date: **30 January**.
- There is no required submission for this pre-lab.
- This is a preparatory step for Lab 1, which depends on ROS. If you try to learn ROS while doing Lab 1, there will not be enough time and you will be under a lot of stress in the middle of the semester. So start early **now**.

The Robot Operating System (ROS) is the most widely used free software platform for programming robot systems. It provides a well-designed architecture, many features, and a strong user community. It is a complex programming system. This exercise helps you to get started. After completing the exercise, you should have ROS running on your own computer.

ROS Installation

Officially, ROS only supports Ubuntu Linux. The ROS and Ubuntu versions are coupled. For example, ROS Kinetic requires Ubuntu 16.04 to work properly. Various packages require a specific version of Ubuntu. We have prepared a Docker with ROS pre-installed so that you can skip the tedious work of ROS installation. For beginners, we recommend that you follow the instructions below and use our Dockerized installation to simplify your work. If you are experienced with Linux, installing ROS directly on a linux system takes some time, but is generally straightforward and provides better system performance.

Install ROS Docker

1. Install Docker. Docker works like a virtual machine. It packages and runs an application in a loosely isolated environment called a *container*. A container can be created from *an image*, which contains everything needed to run that application. A shared docker image can run on different platforms, such as Linux, MacOS, and Windows. Install Docker by following this link. After installation, check that the following command runs successfully:

```
$ docker run hello-world
```

2. Download the ROS Docker image. We have created an image of Ubuntu 16.04 with ROS Kinetic pre-installed. Download this image, and run it on your computer.

Bash:

```
$ cd ~
$ mkdir lab1pre_catkin_ws
$ docker run --init --name lab1pre -p 6080:80 -v
"$(pwd)"/lab1pre_catkin_ws:/home/ir/catkin_ws adacomplab/ros_kinetic:latest
```

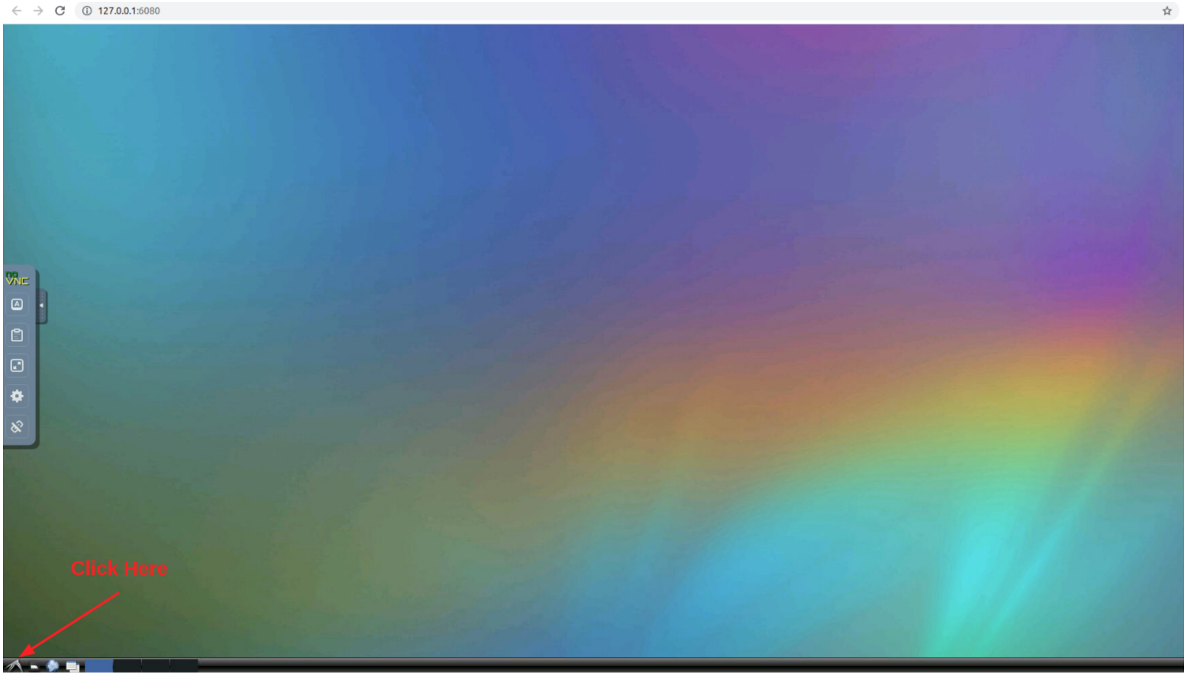
Windows Powershell:

```
$ cd ~
$ mkdir lab1pre_catkin_ws
$ docker run --init --name lab1pre -p 6080:80 -v
${pwd}\lab1pre_catkin_ws:/home/ir/catkin_ws adacomplab/ros_kinetic:latest
```

The above command first creates the `lab1pre_catkin_ws` folder on the host machine, which can be found at `~/lab1pre_catkin_ws`. Then, the docker command pulls the docker image `adacomplab/ros_kinetic:latest` if it is not on your machine. Finally, the docker command starts a container `lab1pre`:

- It has the host folder `~/lab1pre_catkin_ws` mounted at `~/catkin_ws`. Think of the mount as a shared folder between the virtual machine and the host.
- It serves the container's graphical user interface at `http://127.0.0.1:6080/` on the host.

You can treat the container `lab1pre` as a Ubuntu virtual machine that has run the steps 1.1-1.6 in ROS Kinetic Installation Guide.



3. There are two ways to get a command-line interface of the container. The first uses the web GUI. Go to `http://127.0.0.1:6080/`, you should see the above image. Click the bottom-left button shown in the image above and choose `System Tools/LXTrminal`. The second way uses the `docker exec` command. In a terminal on your host, run:

```
$ docker exec -it lab1pre bash
```

4. Create a ROS catkin workspace using the command-line interface

```
$ source /opt/ros/kinetic/setup.bash
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/
$ catkin_make
```

It is **important** to keep in mind that when the docker container is removed/deleted, only the changes in the mounted folder `catkin_ws` are saved. To save other changes, use `docker commit`. See examples here. You are free to `start` or `stop` the container (analogous to powering up/down a virtual machine) from the Docker Desktop GUI or by the commands

```
$ docker stop lab1pre
$ docker start lab1pre
```

Docker does not provide native support GUI. If you are not using the terminal from the web GUI, you need to specify the `DISPLAY` environment variable (typically `:1.0`). With the web GUI open, try the following command on your host's terminal (`docker exec -it lab1pre bash`):

```
$ DISPLAY=:1.0 firefox
```

You should see the firefox browser opening up in the web GUI.

Install Ubuntu and ROS

If you are familiar with Ubuntu and ROS, an alternative is to install them directly on your computer. While the direct installation takes a little more effort to set up initially, it is easier to use and provides improved system performance.

1. Install Ubuntu 16.04.
2. Install ROS Kinetic.

ROS Basics

ROS provides the services normally expected from an operating system, including hardware abstraction, low-level device control, implementation of common functions, interprocess messaging, and package management. It also provides tools and libraries for acquiring, building, writing, and running code on multiple computers. For an overview, see the tutorial slides and the video recording on LumiNUS.

Follow the official ROS tutorials (Sections 1.1.1–1.1.12) and try out the examples on your own system:

- Understand concepts such as ROS master, topics, nodes, messages (Sections 1.1.5–1.1.6),
- Create your own *catkin* workspaces and packages (Sections 1.1.2–1.1.4),
- Write publishers and subscribers (Sections 1.1.11–1.1.13).
- Write services and clients (Sections 1.1.14–1.1.16).

Follow the video tutorials in turtlesim - ROS Wiki (optional).