

The Chinese University of Hong Kong
Department of Electronic Engineering
2023-24 Term 2

ELEG4701 Intelligent Interactive Robot Practice
Laboratory 3: Introduction to ROS

Objective

In this lab session, you will learn:

1. Install ROS on Ubuntu
2. Run rosnod and control turtle with keyboard
3. Let the turtle run a circle. Finish this task in the terminal
4. Create a new package and add new dependence in the package.xml file
5. Let the turtle run a circle
6. Let the turtle stop at a desired position

Name: _____

SID: _____

Date: _____

1 Part 1: Install ROS on Ubuntu

- Please follow the guidelines in the slides.
- After you finish the installation, **run the turtle and control it with the keyboard.**
- `$ rosrun turtlesim turtle_teleop_key`
- After you finish this task, please show it to the TA.
- Slide p11–p16, or visit <https://wiki.ros.org/noetic/Installation/Ubuntu>

Checked by TA: _____

Time: _____

2 Part 2: Run a Circle using Commands in the Terminal

Please follow these steps:

1. Run ROS core
2. Run turtle sim
3. Pub a topic msg
4. Modify the commands in the command line and let the turtle run a circle

You can use any parameters as if it runs a circle:

```
rostopic pub -r 10 /turtle1/ cmd_vel geometry_msgs /Twist
x:5.0
y:0.0
z:0.0
angular:
x:0.0
y:0.0
z:10.0
```

Note: Input `rostopic pub -r 10 /turtle1/ cmd_vel` then double click 'Tab' button, and the rest part will automatically appear. If you input the command below by hand, it will not work. (Q5 is the same)

5. Call a service request, generate a new turtle with the name `turtle_last 3 digits of you SID`, such as `turtle_135`, using the absolute position command, let the new turtle draw a square with length 5.

```
rosservice call /spawn 3.0 3.0 0.0 'turtle2' (x: 3.0 / y: 3.0 / theta: 0.0, name:'turtle2')
```

This step produces a new turtle with the coordinates (3,3) named turtle2.

Note: Input `rosservice call /spawn` then double click 'Tab' button, and the rest part will automatically appear.

```
rosservice call /turtle2/teleport_absolute 8.0 3.0 0.0
```

This step moves the turtle to (8,3)

```
rosservice call /turtle2/teleport_absolute 8.0 8.0 0.0
```

This step moves the turtle to (8,8) and so on to complete the **square** with side length 5 - you will see something like:

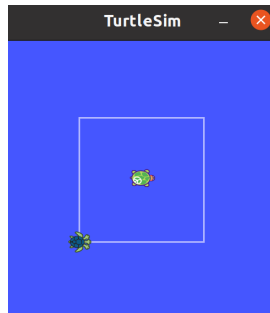


Figure 1: Every turtle will be different

6. Read `rosvbag` command line info on <http://wiki.ros.org/rosvbag/Commandline>, use the keyboard to control the turtle to move any track, and record the track with the name “cmd_record_yourSID” and let the turtle repeat the track automatically – and show it to TA for inspection.

Close every terminal window

Then follow slide page 16, run a `turtlesim`

Open a new terminal (we can mark it as terminal 1 in our mind): `rosvbag record -a -O cmd_record_sid`

Use the keyboard to move the turtle arbitrarily.

In terminal 1, press Ctrl+C at the same time; this means “force stop”

Then input: `rosvbag play cmd_record_sid`

You will see the turtle will move automatically, and its track is the one you created with the keyboard just now.

Show your result of #5 (square turtle) and #6 to the TA to get the score

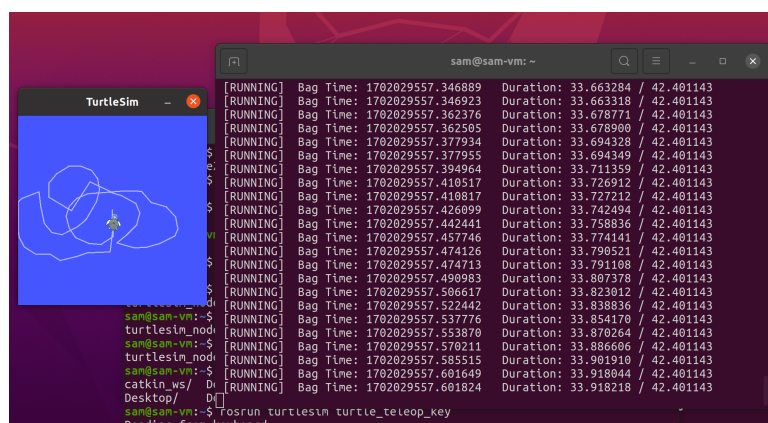


Figure 2: Every path will be different

Checked by TA: _____

Time: _____

3 Part 3: Create a new package, run a circle

Please follow these steps:

- (a) Please follow the guidelines in the slides to create a new ROS workspace with the name `catkin_ws`;
- (b) Create a new package named `eleg_t03_topic`, using commands below:

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg eleg_t03topic_yoursid std_msgs roscpp rospy geometry_msgs turtlesim
$ cd ~/catkin_ws
$ catkin_make
$ source ~/catkin_ws/devel/setup.bash
```

- (c) Edit the source code “`your_node.py`,” rename it as `velocity_publisher.py`, and finish the to-do tasks in it;

`.py` file needs to be right-clicked to change the properties to an executable file

mistake1: `/usr/bin/env: 'python\r': No such file or directory`. This is because the code in that file contains the code: `#!/usr/bin/env python`(First line of the file), In ubuntu, this will be changed to `#!/usr/bin/env python\r`, And `\r` will be taken as an argument by the shell.

Processing method: `sudo apt-get install dos2unix`

mistake2: unindent mistake, Python indentation error, code top frame, then press four spaces

Todo0: Check package.xml and add rospy dependencies
`<build_depend>rospy</build_depend>`
`<build_export_depend>rospy</build_export_depend>`
`<exec_depend>rospy</exec_depend>`

In the Home folder, press Ctrl+H, edit the `.bashrc` file and add

`source ~/catkin_ws/devel/setup.bash`

at the end of the file.

TODO 1 - TODO 4: Refer to `velocity_publisher.py`

- (d) Copy the source code to create a new file with the name `velocity_rectangle`, and modify the code thus the turtle can run a rectangle. If you can not finish this task, Please analyze why you were unable to complete the task and write your answer below.
- (e) Ans: topic `/turtle1/cmd_vel`, can only control linear and angular velocity, but the running time of each command is difficult to set precisely in ROS. So based on Q3's source, it is difficult to run a rectangle.
If you are interested, you can try position control.
After you finish this, please show it to the TA.

Checked by TA: _____

Time: _____

4 Part 4: Stop the turtle at a desired position

Please follow these steps:

1. Copy the `move_turtle.py` to `src` folder;
2. `catkin_make` in workspace;

```
cd ~/catkin_ws
catkin_make
```

3. Edit the source code and finish the tasks list in it. Refer to `move_turtle.py`
4. Input the command: `roslaunch eleg_t03_topic_XXX move_turtle.py 3.14 1 8`
5. After you finish this, please show it to the TA.

Checked by TA: _____

Time: _____

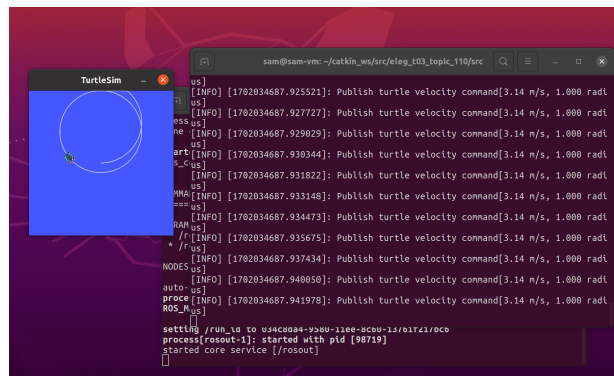


Figure 3: Result for Part 3

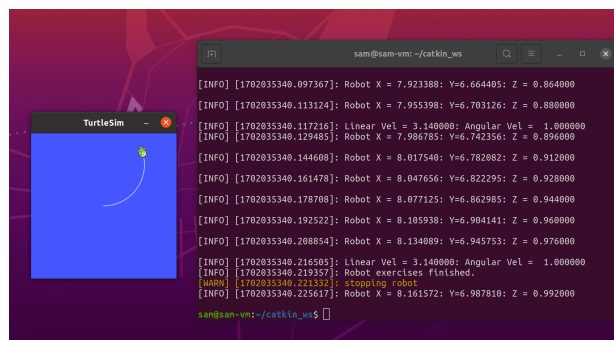


Figure 4: Result for Part 4