

UNIVERSITY OF GIRONA

PROBABILISTIC ROBOTICS

LAB 0

---

# Introduction to ROS

---

*Author:*  
Di MENG

*Supervisor:*  
Dr. Ridao RODRIGUEZ  
PEDRO

February 23, 2017



# 1 Introduction

The goal of this lab is to grasp a brief knowledge about ROS(Robot Operating System) and know how to use ROS for some robot applications. Ros is a set of software libraries and tools which have powerful functionality and can be accessed with python and c++.

There are mainly three parts about this lab, which are installing Ubuntu and ROS, learning ROS from tutorials and programming some simple codes and testing in simulator.

## 2 Installing Ubuntu and Ros

The Ubuntu and Ros were installed successfully without any problems following the website:<http://wiki.ros.org/kinetic/Installation/Ubuntu>.

## 3 ROS learning

From the tutorial, I learned how to create the workspace and package and have a brief knowledge of the node, topic and message. Basically, nodes use ROS to communicate with other nodes and publish or subscribe topic using messages. Topic saves many nodes that published and also can be subscribed. The user can subscribe some information from robot or simulator to decide and perform some actions and publish some performance to control the robot.

## 4 Testing and Simulating

### 4.1 Run the ROS simulator

Using command line "roscore "to launch the ros system. Using command line "roslaunch turtlesim turtlesim \_node "to run the simulator. Then we can see the virtual turtle in the middle of the simulator.

### 4.2 Run the package

Before running the package, the catkin\_make has to be run first. In order to let the bash know the location of package, "source ~/catkin\_ws/devel/setup.bash

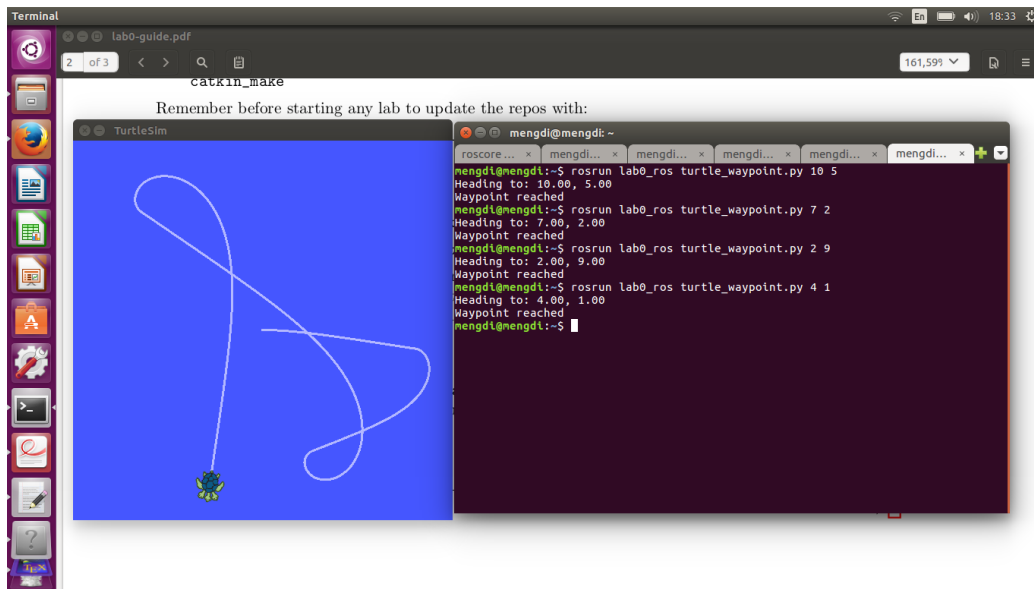


Figure 1: Running waypoint package

"has to be run every time when the user wants to run the package. But there is a simpler way which is to put this command in the `.bashrc` file, so that users don't need to run it every time.

Then the turtle would perform the specified action in the simulator. This lab is to control the turtle to go to an exact position no matter where it is.

## 5 Problems faced and the solution

Problems:

1. The robot keeps moving and can never reach the goal position.
2. Default parameter setting goes wrong
3. Unknown.

Solutions:

1. The reason why robot can never reach the goal position is that the goal position is an exact point but an area. Because the robot is moving with a constant velocity, it can never reach a point. This problem was

fixed by using a tolerance. The robot is considered to reach the goal when it gets into this area. In addition, The iterate function will run every 0.3 seconds and if the velocity times time interval which is the step distance is larger than the tolerance, it would never reach as well. The velocity was set as a linear function of distance between current position and the goal position so that the robot would lower the speed when it is close to final point.

2. The code given by Lab0 guide is the command line which can be used in terminal but not in the package programming, this was kind of confusing. After checking the correct way to set parameter in server, problem was solved.
3. For unknown problems, the most probably situation is the syntax error, such as losing a backlash or unexpected capital letter. Python is a simple language but still should be paid attention on the syntax.