Computer Assisted Surgery and Medical Robotics

Lab0 - Introduction to ROS

Submitted By

Fakrul Islam Tushar

Md. Kamrul Hasan

Submitted To

Prof. Xevi Cufí, PhD

xevi.cufi@gmail.com

Prof. Pere Ridao, PhD

pere.ridao@udg.edu

January 20, 2019

Contents of the report **Page No**

1. Introduction

Controlling or plan to movement of the robots is an essential part of the medical robotics. In this lab we were get introduced to the Robot Operating System (ROS). Robot Operating System (ROS), which is a collection of software packages to aid researchers and developers using robotic systems. ROS is sometimes called a meta operating system because it performs many functions of an operating system, but it requires a computer's operating system such as Linux [1]. One of its main purposes is to provide communication between the user, the computer's operating system, and equipment external to the computer. This equipment can include sensors, cameras, as well as robots [1]. ROS command can be issued in Python or C++ written to cause the robot to respond as commanded.

The objective of the lab was following:

I. To know about ROS
II. To perform an introduction to ROS packages, nodes, and topics
III. To use ROS commands with the turtlesim simulator

2. Experimental Setup

To perform the tasks of this ROS was required of an operating system with Linux. In lab following the instruction we were able to move the turtlesim. Afterwards, Ubunto operating system on the Jave virtual machine from Oracle was installed at home PC following the instructions from the website of the ROS [2]. Then we followed the guideline links which were provided in the Lab 0 to be able to write a Node in Python to make the robot moves to the desired position which defined.

Figure: Turtlesim screen

3. Experimental problems faced

I. As we are not use to with the ubuntu operating systems we find it difficult to operate the basic instruction.

II. Different problems appeared while setting up the ROS and turtlesim simulator on home PC.

III. Using the same code from laboratory the movement of the turtlesim is not the same in-home PC.
Possible explanation could be having some missing packages which did not enable me to observe the same results as the ones while running the code in the laboratory.

4. Solution

I. The turtle state can be obtained by subscribing the'/turtle1/pose' topic. Therefore, we specify the type of topic received to be Pose. On receiving a new message from this topic, a callback function of the name **self.callback** shall be automatically called. Since the aim is to maneuver the turtle to the desired position, we set up a publisher'/turtle1/cmd vel' with message type Twist, allowing us to send linear x, linear y, linear z, angular x, angular y, and angular z values.

II. The input (desired x and y positions) can either be given as an argument when running the node ,or obtained from the param server.

III. the turtle was checked if it has reached desired position (within a tolerance level). If not, we give it a linear x value proportional to the distance to the desired position, and angular z value proportional to the deviation of the turtle's angle to the desired one. The proportional constants are set through experimentation.

5. Code

Code was added with the submission.

Reference

[1] 2018: Computer Assisted Surgery and Medical Robotics **Lab0 - Introduction to ROS** [lab manual]. Girona (Spain): University of Girona.

[2] http://wiki.ros.org/kinetic/Installation/Ubuntu (Accessed November 30, 2018)