ROS for Robot Arm Specialization - ROS Essentials Course

Assignment #2

Time estimated to complete: 7-10 days

Overview:

This assignment will let them understand the loop sense->think->act for any robot. It will combine a publisher and subscriber into the same node to control the robot motion

Write your First ROS Program to Control the Motion of a Robot

Description: In this assignment, you will apply the knowledge you learned to make a robot move and track its location. You will learn which topics are used to control robot motion and determine its position. Let's get started!

Assignment Instructions

This assignment will take you step by step to develop your first ROS program by applying the concept you have learned so far.

You can first watch the companion video before reading the following.

https://youtu.be/8jzqA8xZvrU

The objective is to develop a simple ROS program that:

- subscribes to the topic that gives information about the robot position
- publishes to the topic that makes the robot move.
- develop a method called move(distance) which makes the robot moves a certain distance then stops.
- develop a method called rotate(distance) which makes the robot rotate a certain angle then stops.

We will use the Turtlesim simulator to develop this program.

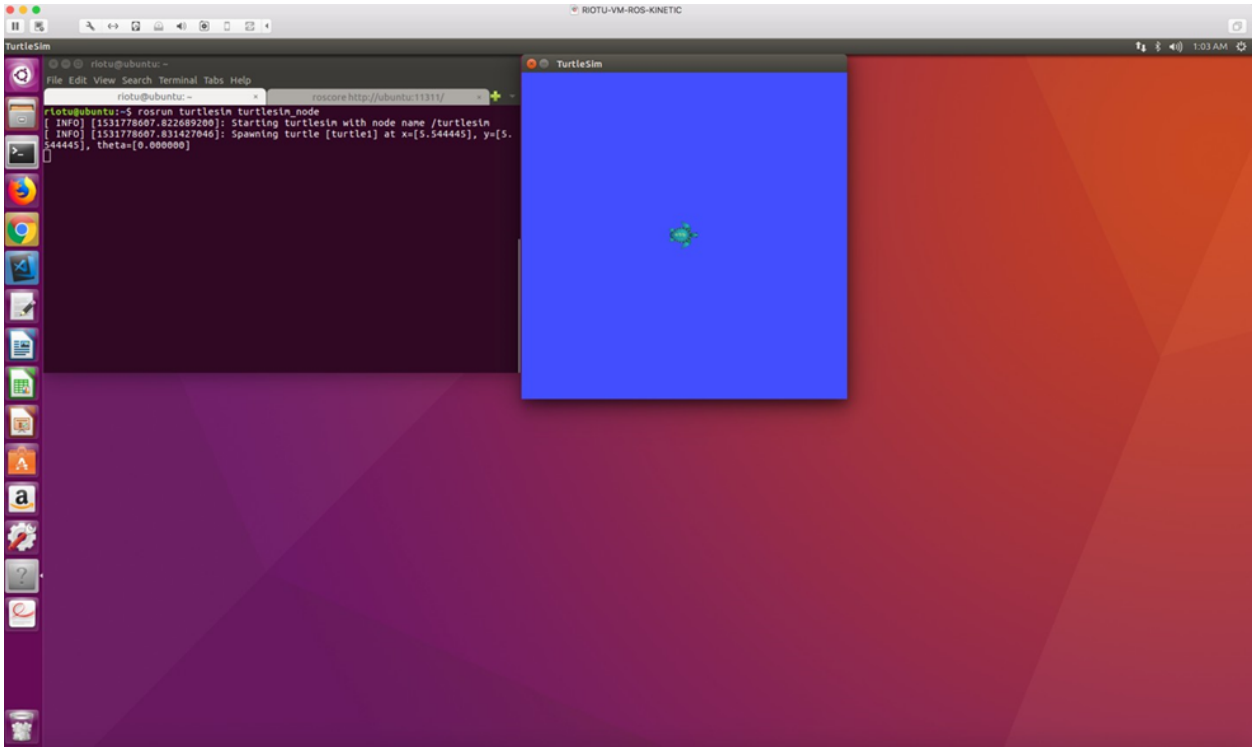
Start the Turtlesim Simulator

First, run the Turtlesim simulator using this command:

1. > roscore

2. > rosrun turtlesim turtlesim_node You

will have an interface similar to this one:



You will develop your first program to control this robot, STEP BY STEP following the instruction below.
We want to make this robot move and display its location. So let us get started.

Q1. Find the topic name of the pose (position and orientation) of turtlesim and its message type. Display the content of message of the pose.

Q2. Find the topic name of the velocity command of turtlesim and its message type. Display the content of message of the velocity command.
Remember that velocity command is the topic that makes the robot move.

Q3. Write a simple ROS node in a script file called turtlesim_pose.py, which subscribes to the topic of the pose, and then prints the position of the robot in the callback function.
Q4. Complete the previous code in turtlesim_pose.py to add a publisher to the velocity and make the robot move for a certain distance. Hint: you can use the rule
Distance = linear_speed * time
Q5. Complete the previous code in turtlesim_pose.py to add a publisher to the velocity and make the robot rotate in place for a certain angle. Hint: you can use the rule
Angle = angular_speed * time
Q6. Use your code above to make the robot move 1 meter and rotate 90 degrees.

What to Submit

- Submit a 3-5 minute video explaining your work and showing response to all questions above and a demonstration of the working example. Put the video on YouTube as unlisted and share the link.
- Submit the code move.py with your modification. Submit all through Phi Classroom.

Deadline: Thursday 26 May 2022.