PROBABILISTIC ROBOTICS

Lab report 1: Introduction to ROS

Author: **RAABID HUSSAIN**

Objective:-

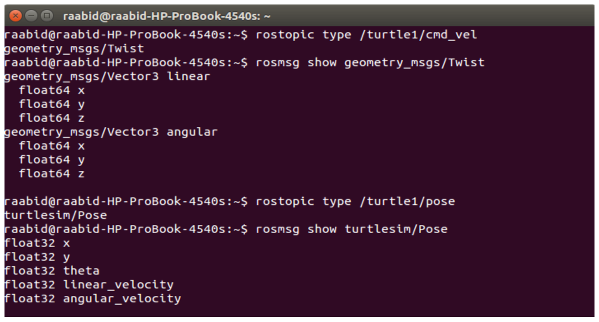
The objective of the lab-work was to get familiarized with the ROS software which is a software framework that has gained popularity in last few years and is emerging as the leading software platform used in robotics. The lab was divided into two parts:-

- The first task consisted of following a set of tutorials available online at wiki.ros.org. These tutorials consisted basic commands and controls in ROS. Some of these tutorials involved understanding the “turtlesim” framework of ROS which was later used in the following task.
- The second task consisted of moving the turtle of turtlesim framework from its current location to the desired position using an efficient trajectory.

Methodology:-

The task was rather a simple task and we were also provided with a skeleton of the python code as a starting point for this task. Thus, lack of familiarization with python language and linux systems proved to be the major obstacles in completing this task. Moreover, most tutorials on the internet being in c++ did not alleviate the task either.

The main features of the ROS framework that were utilized in this task were the nodes and topics. Nodes represent the processes that control various tasks. The nodes communicate with each other through messages whose contents are a topic name and the data structure: it is communicating. Topics are used to publish (send) and subscribe (receive) these messages. A set of request and reply is known as a service. It was necessary to first identify these communication parameters. The main parameters of the turtle that were used were pose and cmd_vel. Pose (structure used = Pose) was used to find the current position of the turtle whereas as cmd_vel (structure used = Twist) was used to send velocity parameters to the turtle in order to move it. ROS commands were run in the “Terminal” for this task, the results of which are shown in Figure 1.



```
raabid@raabid-HP-ProBook-4540s: ~
raabid@raabid-HP-ProBook-4540s:~$ rostopic type /turtle1/cmd_vel
geometry_msgs/Twist
raabid@raabid-HP-ProBook-4540s:~$ rosmsg show geometry_msgs/Twist
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z

raabid@raabid-HP-ProBook-4540s:~$ rostopic type /turtle1/pose
turtlesim/Pose
raabid@raabid-HP-ProBook-4540s:~$ rosmsg show turtlesim/Pose
float32 x
float32 y
float32 theta
float32 linear_velocity
float32 angular_velocity
```

Figure 1: Description of the messages being transmitted

The next thing was to start accommodating these parameters into the skeleton. The main problem faced here was the change in syntax. As all the tutorials in the first task contained commands to be run using the terminal, it was a little hard to convert those commands into the syntax required by the skeleton file.

After defining the publisher and subscriber and writing code to read the turtle's current position and sending the velocity for the turtle to move, the code was tested. However, it was found that the turtle was stationary. rqt_graph was used to identify the nodes and topics relationship and figure out the problem which stated that the nodes had not developed a communication channel with each other.

After establishing the appropriate relationships among the nodes, the code for moving the turtle to the desired location was written. There were many approaches to this task:-

- First move in the x-direction to the desired coordinate and then in the y-direction.
- Adjust the linear and angular velocities of the turtle according to the distance to the target and thus follow a circular trajectory.
- First change the angle of the turtle to face the target location and then simply move forward.

Since the third option gave the optimum path, it was chosen by inserting an extra while loop in the skeleton for the angle and then using the while loop provided in the skeleton to simply move forward. The speed of the turtle was kept proportional to the difference in both the angle and the forward movement. Figure 2 represents the trajectory followed by the turtle.

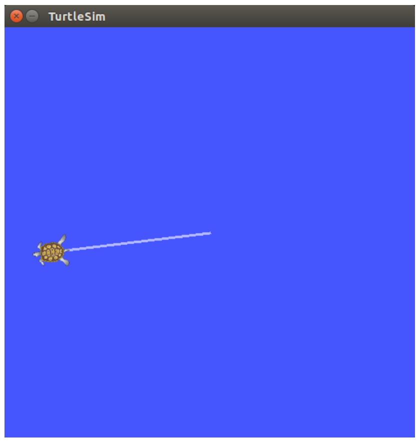


Figure 2: Trajectory followed by the turtle

Conclusion:-

ROS is a powerful tool gaining popularity in the field of robotics. It provides a platform for simulation and practical application to various form of robotics. In this lab-work, the basic knowledge of ROS was gained and a small program successfully implemented that used the turtle simulator in ROS to move the turtle to the desired location.