Sadaan Tahir - 25100224

EE 3002 L1 (Junior Design Studio - Robotics)

Spring 2025 - LAB 1

Prelab Tasks:

- **VMware Pro** setup: https://drive.google.com/drive/folders/1ZaVRQ1HTYU0Y5LrISL0aKQPgPhnVhaMk?usp=sharing
- The link for the zipped folder with **ROS Noetic** + **Gazebo v11** + **Ubuntu 20.04** already installed and ready to use can be found here: https://drive.google.com/file/d/1YNI00Z8jHZOTfESOSRe6wpu79dYyU4aL/view?usp=sharing

Task 1: ROS 1.1 Beginner Tutorials [20 MARKS]

To familiarize yourself with the basic principles of ROS, we'll begin by completing the **ROS 1.1 Beginner's Tutorial** as outlined on this ROS Wiki: http://wiki.ros.org/ROS/Tutorials.

These tutorials will guide you through the basic functionality of ROS: setting up your workspace, building and running your first ROS package, and exploring fundamental concepts like nodes, topics, messages, and services. By effectively navigating through this tutorial, you will build a solid foundation for tackling more advanced tasks in the labs to come.

You need to complete at least the first **13 tutorials** to get **full marks**. [20 MARKS]

Task 2: Custom Message Publisher and Subscriber in Python [30 MARKS]

Now, let us put our newfound ROS knowledge into practice by developing a **custom message publisher and subscriber** using Python. Before starting this task, it is advisable to install **Visual Studio Code** and the **C++** and **Python** extensions.

2.1 Setting Up:

Begin by setting up a new **catkin workspace**, naming it after yourself, such as **your_name_ws** (replace your_name with your actual name, for example like this: **sadaan_ws**). Inside this workspace, create a **ROS package** called **lab1** by using the catkin_create_pkg command.
[1 MARK]

2.2 Custom Message:

Create a **new message type** called **ComplexNumbe**r in the **msg folder** of your **lab1 package**. This message should include two fields: **real** (of type float32) and **imaginary** (also of type float32). After defining the message, update the **CMakeLists.txt** file to ensure the new message is recognized, and then **build** your package using the catkin_make command. [5 MARKS]

2.3 Publisher:

Write a Python script that repeatedly publishes instances of your ComplexNumber message at regular intervals. Use the **rospy** library to implement the publisher, and **publish** the messages to a **topic** named **/complex_numbers** or a more descriptive topic name that reflects the purpose of the data. [12 MARKS]

2.4 Subscriber:

Create a second Python script that **subscribes** to the **/complex_numbers** topic and listens for ComplexNumber messages. In this script, **compute** the **magnitude** and **phase** of each received complex number using the **message fields**. Then, display the calculated values in the console for easy observation. [12 MARKS]

Submission Requirements:

1. Include a **zip file** of the ROS package created in Task 2. Ensure the zip file contains the complete folder structure from the catkin workspace.
2. Submit a **LAB report** in PDF format (please don't submit word files). This report should provide any explanations for the tasks performed along with any screenshots. For this lab alone, explanations are the **only thing** required.