**F1TENTH Autonomous Racing** **(Due Date:)**

# Lab 1: Introduction to ROS

*Instructor:* Dr. Rahul Mangharam *Name:* Tuan Nguyen, *StudentID:* 20979605

**Course Policy:** Read all the instructions below carefully before you start working on the assignment, and before you make a submission. All sources of material must be cited. The University Academic Code of Conduct will be strictly enforced.

## 1 Workspaces and Packages

### 1.1 Written Questions

1. (**Python & C++**) CMakeList is a file that specifies how to build a package and where to install it. The CMakeList file for a package is also related to the CMakeList file for C++ objects in which it specifies how to build the C++ code and where to install it.
2. (**Python & C++**) CMakeList is needed for Python nodes to make sure that python scripts get installed properly and use the right python interpreter. Python nodes do have executable objects.
3. (**Python & C++**) home/catkin_ws
4. (**Python & C++**) Sourcing the setup.bash files gives us access to ros commands and ros packages. The setup.bash in /noetic directory gives access to ros commands while the setup.bash in /devel directory gives access to ros packages.

## 2 Publishers and Subscribers

### 2.1 Written Questions

1. (**C++**) ROS NodeHandle is a C++ object that represents a node, and a single node can have multiple nodehandles.
2. (**Python**) There is no nodehandle object in python. The role of rospy.init_node() is to initialize a node
3. (**C++**) ros::spin() and ros::spinOnce() are subscriber's callback handles. While ros::spin() would execute the callback function whenever a new message is received, ros::spinOnce() would only execute once.

4. (**C++**) ros::rate() controls the frequency at which a program loop is running
5. (**Python**) The callbacks for subscribers are controlled using the function rospy.Subcriber(). rospy.spin() is need in python because it keeps a node from exiting until the node has been shutdown

## 3 Implementing Custom Messages

### 3.1 Written Questions

1. (**C++**) The header file of the message file is needed so that the custom message can be imported in any C++ node.
2. (**Python & C++**) *Header header* is a special data type in ROS which contains a timestamp and coordinate frame information.*Header header* can be included in custom messages and it provides additional fields which are *seq*, *stamp*, *nsecs* and *frame_id*

## 4 Recording and Publishing Bag Files

### 4.1 Written Questions

1. (**Python & C++**) Bag files can be saved in any directory. This is done by going to the directory in which I want to save the published messages and then running the command *rosbag record* in the terminal.
2. (**Python & C++**) The bag files will be saved in the package directory in which the corresponding *rosbag record* commands are located. In this case, the directory of the saved bag files cannot be changed unless I manually move the files.