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

Lab 1: Introduction to ROS

Instructor: INSTRUCTOR *Name:* STUDENT NAME, *StudentID:* ID

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. CMakeList is a meta build system that can be used to generate build files for a specific environment such as make files. Yes, it is related to make files. CMakeList is a higher level language that generates make files.
2. Yes, we use CMakeList for Python in ROS, but we do not create an executable for Python.
3. The workspace directory.
4. This adds several environment variables that ROS needs in order to work. The first one is needed for the ROS distribution and its base packages to be recognized, and the second one is needed for ROS to recognize our workspace and our customized packages.

2 Publishers and Subscribers

2.1 Written Questions

1. As described by the documentation, the ros::NodeHandle class serves two purposes. First, it provides RAII-style startup and shutdown of the internal node inside a roscpp program. Second, it provides an extra layer of namespace resolution that can make writing subcomponents easier. Yes, we can have more than one nodehandle objects in a node.
2. No, Python does not have a nodehandle object. "rospy.init_node" serves the purpose of ros::NodeHandle for starting the node and communicating with the ROS master.
3. ros::spin() and ros::spinOnce() are both ROS message callback handlers. The difference between them is that ros::spinOnce() will return after the call, so that later portions of the

program can be executed, while ros::spin() will not return after calling, that is, the program will not execute anything down this line.

4. ”A ros::Rate object allows you to specify a frequency that you would like to loop at. It will keep track of how long it has been since the last call to Rate::sleep(), and sleep for the correct amount of time”, according to the ROS documentation.

5. We use spin() in python to control callbacks for the subscribers.

3 Implementing Custom Messages

3.1 Written Questions

1. Because C++ does not recognize the message file itself. Catkin_make will generate a header file with message file that can be included by the program.

2. ”Header header” is a general mechanism for setting frame IDs for libraries like tf, according to the ROS documentation. We can include it in our message file. It is a standard metadata for higher-level stamped data types, generally used to communicate timestamped data in a particular coordinate frame.

4 Recording and Publishing Bag Files

4.1 Written Questions

1. The bag file is saved in the directory where ”rosbag record” is called. We can change where it is saved by changing the directory at which we run the ”rosbag” command, we can also use the ”-o” argument and specify where we want to save the bag file.

2. The bag file is saved in the directory where the command is called. We can change where it is saved by changing the directory at which we run the ”roslaunch” command, we can also use the ”-o” argument and specify where we want to save the bag file.