ME 459/5559 – Robotics and Unmanned Systems
HW #4: DUE September 26th, 2022

LATE HOMEWORK WILL BE DEDUCTED 10% PER DAY AFTER THE DUE DATE

Problem 1:

Complete the following ROS tutorials at https://docs.ros.org/en/foxy/Tutorials.html

- Understanding ROS Nodes
- Understanding ROS Topics

Save a screenshot of the Turtlebot simulator and show/print the screenshot.

Problem 2:

Complete the following ROS tutorials at http://wiki.ros.org/ROS/Tutorials

- 12. Writing a Simple Publisher and Subscriber (Python)
- 13. Examining the Simple Publisher and Subscriber

Save a screenshot of the running code in tutorials 13, and print these screenshots to turn in.

Problem 3:

For this problem you will be using ROS2 and Gazebo to simulate the Turtlebot3 Burger platform. Help in loading the simulation can be found at http://emanual.robotis.com/docs/en/platform/turtlebot3/simulation/. Make sure to go to the Gazebo part of the E-manual. When setting the model type, replace "${TB3_MODEL}" with "burger."

Create a ROS2 node that makes the Turtlebot3 travel forward at 1.5 m/s (remember this speed is much faster than it can do in real life) for 5 seconds, then turns to the right at 0.15 rad/s for 2 seconds, then continues forward at 1.5 m/s for another 5 seconds before stopping. Log the position data, and velocity & angular velocity commands.

Create a subplot of the x & y position data, velocity & angular velocity commands versus time.

Submit your Python code.

Problem 4:

Create a ROS2 node/script that uses a feedback controller to control the heading of the Turtlebot3. Once you have adequately tuned the controller, collect the data (by writing to a log file) from a 90 degree step input (use a forward speed of 0.15 m/s).

Create a plot of the desired and actual heading versus time. What is the rise time, settling time, and percent overshoot of your controller?

Submit your Python code.

Problem 5:
Create a ROS2 node that follows the given set of prescribed waypoints: [0,0], [0,1], [2,2], [3, -3]. Start your robot at [0,0]. Create a plot of the X, Y coordinates to show how well your robot follows the desired path. Use a maximum translational speed of 0.15 m/s.

Submit your Python code.

Graduate Students Only:

Problem 6:
Redo Problem 5 with speeds ranging from 0.15 – 1.5 m/s (use at least 4 speeds) and provide the results regarding the ability of the turtlebot to follow the desired path/waypoints. A single plot showing the deviation from the desired path for each speed is the minimum acceptable information.

Discuss the results, what is the max speed we should use for the simulation environment of the turtlebot (if we want to go as fast as possible)?