ME 401/5501 – Robotics and Unmanned Systems
HW #6: DUE October 24th, 2022

For this assignment 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."

Use the multi_spawns_turtles_launch.py script to simulate the pursuer and the evader. NOTE: the evader's name space is "turtle" and the pursuer is none.

Make an evader script that controls the "turtle" (evader) to go from [2,2] to [9,9]. In addition add a condition to allow evader to move to random waypoints.
HINT: If you want to do this quick, just use your old code to send waypoints to your turtlebot and put it in your evader script. Instead of sending pathfinding waypoints, you can either hardcode the straight-line waypoints, and for the random waypoints use Python's random command to create random waypoints.

You must use the LIDAR (not the evader's true position) for the pursuit

Proportional Navigation - Pursuit

Problem 1:

With the evader moving from [2,2] to [9,9], create a pursuit script/node that chases after the evading turtblebot. You must use proportional navigation. You will need to tune your PN constant to provide suitable response. You must collide with the evading turtlebot within a simulation time of 30 seconds.

Provide evidence to show your pursuit vehicle working (x vs y plot of both evader and pursuer for example).

Identify your PN constant.

Problem 2:

With the evader moving from [2,2] to [9,9], test out your working pursuit script. Although random between tests, perform 3 tests with PN constant = 10% of problem 1 value, 100%, and 1000%. How does the performance change in in terms of capturing the random evader? Provide plots/figures to support what you see.

Problem 3:

With the evader moving randomly ,test out your pursuit script. Comment on how it performs (run this simulation a few times to compare). Provide plots/figures to support what you see.

Problem 4:

Using an evader with a maximum speed of 0.1 m/s (and your own pursuer of 0.2 m/s) and a starting location of (2,2) and a goal location of (9,9), create an evader that can out-maneuver your pursuer (pursuer starts at (0,0).

Problem 5:

Work with a partner to test your evader and pursuer (take turns) identical to Problem 4. You can accomplish this through sharing scripts or by connecting two machines in ROS2. See the link below for a quick tutorial on how to do this (quite easy).

https://roboticsbackend.com/ros2-multiple-machines-including-raspberry-pi/

Show plots of your pursuer and evader.

Problem 6:

Work with a partner to test your evader/pursuer scripts on the **real** turtlebots in the lab. Show your results.