Mobile Robotics Assignment #2 Name: ________________________
University of Ottawa Dr. W. Gueaieb

Total (60 points)

Instructions

- Individual work only. ABSOLUTELY no collaboration of any kind with anybody concerning any part of the assignment is allowed during the assignment period. Questions must be addressed to the professor during office hours (not by email).
- Source codes will neither be marked nor debugged. They will be executed to ensure proper functionality with different input/parameters. Only the program results are graded. It is the student's responsibility to make sure that their programs run as described.
- Source codes must be very well commented so that they are easy for the marker to understand when needed.
- You have to write your own programs. Copying a source code from the Internet or any other source is not acceptable and will be considered as plagiarism.
- All students must be aware of the rules and regulations posted on the course website, especially those on assignment submission and plagiarism.

Deliverables

- Submit your work online through the course website in exactly 4 files as follows:
  1. The source code of each node in a separate file (with a proper file name and extension).
  2. A "00Readme.txt" text file explaining how to create the package and run any of the nodes.
- Do NOT archive the files in a zip file.
- Do not forget to attach all the files BEFORE you click the "Submit" button.

Problems

(60$^{\text{pts}}$) **1.** In this problem, you will create a ROS package to drive the turtle robot (from the `turtlesim` package) from its current position to a destination point.

As per ROS' philosophy, the task is divided over a number of nodes (3 nodes in this case), as described below. Write a Python program for each node.

(a) (15 pts) Node `target_publisher` prompts the user for the (x, y) coordinates of a destination position. The node must keep prompting the user for new coordinates until $0 \leq x, y \leq 11$. Then, it publishes the target coordinates over topic `/turtle1/destination` at a rate of 10 Hz. The topic is of the same type as that of topic `/turtle1/pose`, which is used by the `turtlesim` package to report the robot's pose.

(b) (15 pts) Node `pose_reporter` reads the robot's pose from topic `/turtle1/pose` and reports the following signals, using the `loginfo` function, at a rate of 2 Hz. This can be done by publishing them to a new topic of your choice, for example.

- The robot's current positions (in m).
- The distance between the current and target positions (in m).
- The robot's current orientation (in degrees).
- The orientation error (in degrees) defined as the robot's current orientation minus the target orientation.

(c) (30 pts) Node `robot_driver` drives the robot to its destination by publishing its linear and angular velocity on the `/turtle1/cmd_vel` topic with a frequency of 20 Hz. The destination is considered to be reached if the robot gets within 10 cm from it.