Advanced Control Laboratory (085705)
ROS #2: Building a Robot Controller

This lab focuses on building a Robot motion controller that will communicate via ROS.
The team with the best code, would run it on the actual Pioneer Robot.
be sure to save the code into a folder labled ROS2.

1. You are required to write a PD based controller for the Pioneer Robot with the following header
   function [robot_path] = MyController (Goal, Odometry_topic, Velocity_topic)
   Your controller will be given a destination in the form of a 3D point, "Goal".
   These are the objectives of your controller:

   (a) get the robot as close to the destination as possible using only Odometry readings.

   (b) store the estimated position of your robot during the mission "robot_path".

   Here are some guidelines to help you in your mission:

   - All guidelines from ROS#2 pre-lab also apply here, be sure to go over it and make sure you are not missing anything.
   - Use the flow diagram you created in the pre-lab to help you with writing the code.
   - Remember to break down the path to the destination into segments.
   - Remember your robot should first rotate (using a rotation controller) and only than move forward (using translation controller).
   - Use the PD_Block.m function, as a PD controller for your implementation for both rotation and translation.
   - The PD_const for the translation controller is $\begin{bmatrix} 0.6 \\ 0.05 \\ 100 \end{bmatrix}$.
   - The PD_const for the rotation controller is $\begin{bmatrix} 0.01 \\ 0.001 \\ 200 \end{bmatrix}$.
   - Remember you need to communicate with the robot !!
     Use what you've learned during ROS#1, send velocity commands to the robot and receive odometry messages from it. The ROS topics "Odometry_topic" & "Velocity_topic" are given to the controller, you do not need to create them inside the controller.

2. Build a main function that uses your MyController function in order to test your controller in Gazebo

   (a) you should create the required Publishers and Subscribers in the main function and transfer them to your controller, "Odometry_topic" & "Velocity_topic".

   (b) your main function should plot offline, the path the robot traveled using "robot_path".