ECE5242 **Project 3: SLAM**

Code due date: [redacted] on CMSX, <NetID>_project3.zip
Report due date: [redacted] on CMSX, <NetID>_project3.pdf

In this project, you will implement the structure of mapping and localization in an indoor environment using information from an IMU and range sensors. You will integrate the IMU orientation and odometry information from a mobile robot with a 2D laser range scanner (LIDAR) in order to build a 2D occupancy grid map of the walls and obstacles in the environment. Training sets of odometry, inertial, and range measurements from a mobile robot will be provided for this project.

Training Data Download: Now available at https://upenn.box.com/v/ECE5242Proj3-train
Test Data Release: [redacted] https://upenn.box.com/v/ECE5242Proj3-test

:These data sets contain timestamped sensor values, corresponding to the raw sensor readings. Download these files and be sure you can load and interpret the file formats. You can find the 'docs/platform_config.pdf' file and utility functions (load_data.py and p3_util.py) that explains about the robot and data.

Upload: on Canvas

(1) Code [redacted], <NetID>_project3.zip)

: Do include the resulting map images but do not include data when you submit.

(2) Write-up ([redacted], < NetID >_project3.pdf)

: Write a project report including the following sections: Introduction, Problem Formulation, Technical Approach, Results and Discussion.

: Make sure your result includes SLAM image of training and test data set (If you have video files, please include them!)

1. SLAM: First, you should run your robot using pure wheel odometry measurements and yaw gyro readings. Make a 2D map using this data before correcting using range readings. You should then be able to provide a visualization of the motion of the robot within a 2D map. Up to this point, you will have done the first phase of this project. Next, you will need to simultaneously localize the robot pose, and construct the surrounding 2D map using a pose filter and occupancy grid algorithm. The cython files will be provided, along with python binding instructions, to help you experiment with 2D LIDAR scan matching.

2. Help functions for python are provided. Please see files in the 'docs' directory for data description and additional implementation details.

3. TESTING. You should then make sure that your program can take input sensor readings from unknown environment. We will release the test data at [redacted] in order to give some time to run your program with test data. You should be able to show your results with proper visualization from an unknown test dataset.