# ME 640: Autonomous Mobile Robotics

# Final Exam

# April 9th, 9:00 AM – April 10th, 9:00 AM, 2018

# Prof. Steven Waslander

Exam length: 24 hours. There are three (3) questions in total. **Submit your completed work on UW-LEARN as a single PDF solution file with an accompanying single code zip archive containing three folders labeled Q1, Q2, Q3.**  Handwritten solutions are acceptable, but please include scans or photos in the submitted pdf file that are legible. All figures must be in the pdf and be sure to generate figures regardless of whether your solution was fully successful or not.

You are allowed to use any resources available to you, including course notes, course code, textbooks, the internet etc. However, **you are not allowed to communicate with any other human being about the exam or its questions at any point in the period of time you are taking the exam**. Please also refrain from talking to anyone about this exam who has not yet completed it, or passing on questions or solutions to anyone until the exam is over. There is one exception to this rule, you may email me ([stevenw@uwaterloo.ca](mailto:stevenw@uwaterloo.ca)) or phone in questions (cell: 647 760-6702) at any time, and I will try to respond as soon as possible. Keep in mind I may not be available at all hours, so be sure to read through and/or start all the questions before it gets too late for me to respond.

The exam is marked out of 100. Marks are indicated as [x] at the start of each question. Show all your work, and be sure to at least outline answers to all questions.

Good luck!

1. **EKF SLAM in a Changing Environment [35]:** Many uses for mobile robots have been identified in the retail sales environment, including cleaning, greeting, information provision and restocking of shelves. Some of the challenges of working in such spaces is that they are both cluttered environments and cost sensitive applications, so SLAM solutions with low cost sensors are a natural fit. In this problem, we will improve our default EKF SLAM algorithm to rely on 2D bearing only visual measurements and a single range beacon from a charging station.

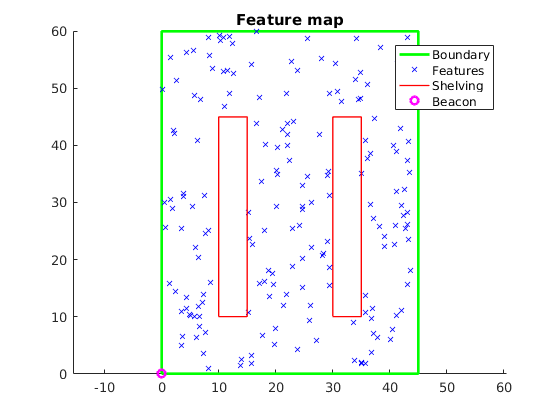
The robot in this problem will be a simple two-wheel robot, and will start at the bottom-left corner of the map. The robot will need to track a desired path through the environment, relying on its vision and range measurements for state estimation and mapping.

Figure 1: Store map for Problem #1

* 1. Define a motion model for a two-wheeled robot, and construct a controller to track a desired path through the environment, as depicted in Figure 1. The robot has a 1 m diameter round footprint, and should stop and change directions when it comes into contact with an object in the environment. Describe how you will update the robot motion with velocity and rotation rate commands, how you will check for collisions with the environment, and how you will track the path from one line segment to the next. (u1 = v, u2 = w)
  2. Define measurement models for the visual features and range measurement. Describe your initialization method for each new feature observed in the scene. Assume the bearing only measurements to features are uniquely identifiable, and for convenience, an index to each feature has been provided in the store\_map.m problem definition. (ASSUME beacon goes through shelving)
  3. Implement the EKF SLAM algorithm on the map provided, with no information about the true feature locations. Define the coordinate frame of your map with (0,0) at the bottom left, the location of the range beacon and charging station, and assume an initial robot position and heading of (1,1, π/4). Navigate the robot through the environment by following a reference path of your own design at 2 m/s, but try to cover the whole environment without traveling through the shelving unit obstacles. Use a 5 Hz update rate for the range beacon and image features, with bearing only measurements for features in a 120 degree field of view out to 15 m, with additive Gaussian noise of 0.001 rad^2 for bearing measurements, 0.01 m^2 for range measurements, and an additive Gaussian disturbance model for the two wheeled robot with covariance,



Demonstrate the SLAM results by plotting the true and estimated vehicle path through the environment, the feature estimates and error ellipses for the observed features after 1, 10, 30 and 60 seconds. (T >=60s)

* 1. Add 10% correspondence errors to the image feature measurements and correct for them using an outlier rejection method (RANSAC?). Select 10% of the features observed in each measurement, and associate the measurement with one of the other features within a 15X15m window in front of the robot (=/-7.5m to the left and right and 15m forward). How do you detect whether a feature measurement can be used for SLAM? What performance degradation results in terms of robot pose and average feature position for the run, if any?

1. **Bicycle State Estimation with Multi-path Errors [35]:**  A recent Indiegogo campaign has promised the world Speedforce, a smart bicycle computer, with an embedded GPS, wheel odometer and steering angle potentiometer inside the stem of the bicycle steering column. However, they soon realized the odometer could not be trusted at low speeds, and the placement of the GPS antenna in the stem led to low quality position and velocity estimates. Your job is to design a 2D pose estimation system (*x*, *y*, *θ*) for the Speedforce that is robust to the common modes of failure of these sensors in urban cycling.



Figure 2: The SpeedForce Smart Bike Stem for Problem #2.

* 1. Using the normal bicycle model from the course notes, implement a 2 Hz simulation of the bicycle, with length of 1 m between wheel centers, moving through an open environment at a nominal speed of 8 m/s and a steering angle trajectory of  for 100 seconds. Present plots of the resulting trajectory in the x,y plane, with heading and steering angle indicated once every 10 timesteps.
  2. The bicycle will be ridden by an unpredictable human, so it is not possible to know the inputs to the bicycle in advance, and therefore a modified model will be required for construction of a state estimator. Define a bicycle motion model that includes position (*x*,*y*), heading (*θ*), velocity (v) and steering angle (*δ*) as the unknown system states. Both the velocity and steering angle are not controlled by the computer, and so can no longer be thought of as inputs. Instead, model the system without inputs, and note that changes in the velocity and steering angle will still occur through additive Gaussian disturbances. The disturbance model covariance will be

.

The bicycle has minimum and maximum steering angles of +/- 25∘.

* 1. In typical operation, measurements will come from the GPS receiver and the steering angle potentiometer in the stem. The GPS receiver will provide position and velocity measurements in x and y at 2 Hz, with additive Gaussian noise of covariance of (0.5)2 m2 in *x* and *y*, and (1.0)2 m2/s2 in *vx* and *vy*. The steering angle potentiometer returns a steering angle measurement at 2 Hz converted into radians with additive Gaussian noise covariance of (2)2 deg2. Define measurement models for each of the sensors.

* 1. A wheel odometry unit can also be integrated into the system, with a pickup on the front fork and a magnet that generates a pulse in the pickup each time it passes. Therefore, the wheel odometry measurement comes to the computer in the form of a digital pulse once per revolution of the wheel, and the microcontroller provides a pulse count at 1 Hz, which is the number of pulses received within the last 1000 milliseconds. On average, 1 out of every 100 pulses is missed, and a missed pulse event occurs uniformly. The wheel circumference is fixed to 2100 mm. Define a measurement model for the wheel odometry measurement. What is the minimum speed above which the error due to collecting only a single pulse per revolution drops below 5% of the true speed?
  2. Define a particle filter to estimate the full state of the bicycle using all three measurement sources. Implement the simulated measurements and your particle filter. Use a forward speed greater than the minimum speed identified in part d. Present plots of 1) the simulated measurements and true states, 2) the results of the estimator, including the particle sets with heading indicators at timesteps 10, 30, 50, 70, 90 and 100, as well as the true states throughout the trajectory, 3) the particle mean states and the true states with respect to time. What is an appropriate number of particles, and what strategy will you use to avoid particle deprivation while maintaining computational performance?

1. **Aerial Drone Racing – Planning and control under Acceleration Constraints [30]:**  Drone racing is currently an exciting way to spend your hard earned money on the weekends, thanks to frequent crashes with obstacles in the environment. To change that, we want to develop a high-speed motion planner that understands the acceleration limits of the vehicle and plans maximum velocity, feasible paths through cluttered environments.

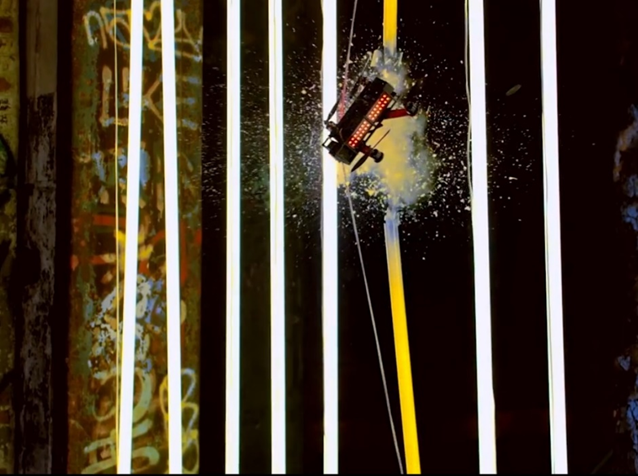


Figure 3: Aerial Drone smashing through fluorescent light

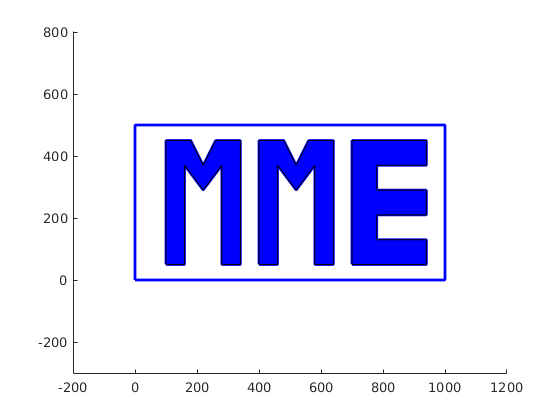
* 1. A small rotorcraft UAV (drone) is limited in its forward velocity by the need to stay in the air and the maximum thrust available to counteract drag forces. As a result, a vehicle has a maximum bank angle and top speed based on its thrust to weight ratio. Define a simple 2D kinematic motion model for the drone with (x, y, vx, vy) as states, and (ax, ay) as inputs. If a vehicle weighs 1kg, has a maximum thrust of 2mg, what is the maximum bank angle and lateral acceleration it can apply at full thrust in level flight? What are the constraints on ax and ay?
  2. While incorporating the constraints on acceleration inputs defined in part a, define a motion planning algorithm to find a feasible trajectory from any start point to any goal location in the environment. What collision checking strategy will you use? Describe your specific approach for identifying paths through the narrow passages in the map. (implying bridge or gaussian sampling)
  3. For the map presented in Figure 4 and provided with the exam, demonstrate your motion planner on a single set of start and end points. Show a plot of the planning process, and a separate plot of the final trajectory selected around the course. Further, present a plot that shows the acceleration commands used and how they satisfy the flight constraints of the quadrotor. 

Figure 4: Racing Map for Problem #3

1. Modify your planning method to seek an optimal solution, instead of merely a feasible solution. Define an optimality metric, or a path cost function, and evaluate possible trajectories using the cost function. Is your method guaranteed to find an optimal path if it runs long enough?
2. Implement the optimal planner for traveling between any two points in the environment. Present an initial feasible and a refined path and plot the evolution of your cost function for improvements to the path.