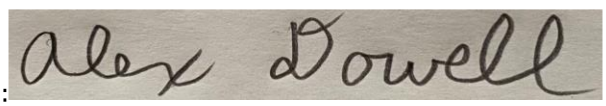
ME 459/5559 Exam # 2
Spring 2022
DUE Thursday, May 5th, 2022 at 11:00 PM

Rules:

1. You **MUST** work individually.
2. Any suspected cheating will be immediately reported to the associate dean.
3. The exam is a take-home exam.
4. Additional written and/or oral justification for any answer may be requested by the instructor.

I have read, understood, and agree to abide by the exam rules.

Printed Name: Alexander Dowell

Signature: 

1. **Conceptual Questions**

   (a) Explain in what scenarios would RRT be the best choice (of the RRT, A*, Dijkstra's $\boxed{5\%}$ options).

RRT does not find an optimal path but generally has considerably less time to compute especially when the C-space is particularly large. With that said, if the C-space is heavily constrained such that the possible paths are limited, RRT will likely be slower than A* and maybe slower Dijkstra. Therefore, a good scenario for RRT is where the C-space is large that is not heavily constrained, and a non-optimal path is acceptable.

(b) If you have a single starting point and 20 stops on your delivery run (traveling delivery $\boxed{5\%}$ man). How many possible paths are there that visit each of the stops? Make sure to show your work.

$$Possible\ routes = (number\ of\ stops)! = 20! = 2432902008176640000$$

(c) Describe how the genetic algorithm operates, and how it is applicable to the traveling $\boxed{4\%}$ salesman problem.

The genetic algorithm(GA) is a search-based optimization method that is used to find close to optimal solutions to difficult problems, specifically NP-hard problems. In GAs, there is a pool or a population of possible solutions to a problem. The solutions are then combined and mutated, to produce new set of solutions, and then process is repeated. Each solution has a fitness value, depending on what the GA is optimizing for, associated to it and the "fitter" solutions are given a higher chance to reproduce to create even more "fit" solutions. Since TSP problems become NP hard problems quickly, GA algorithms are good candidates for providing close to optimal solutions in a reasonable timeframe.

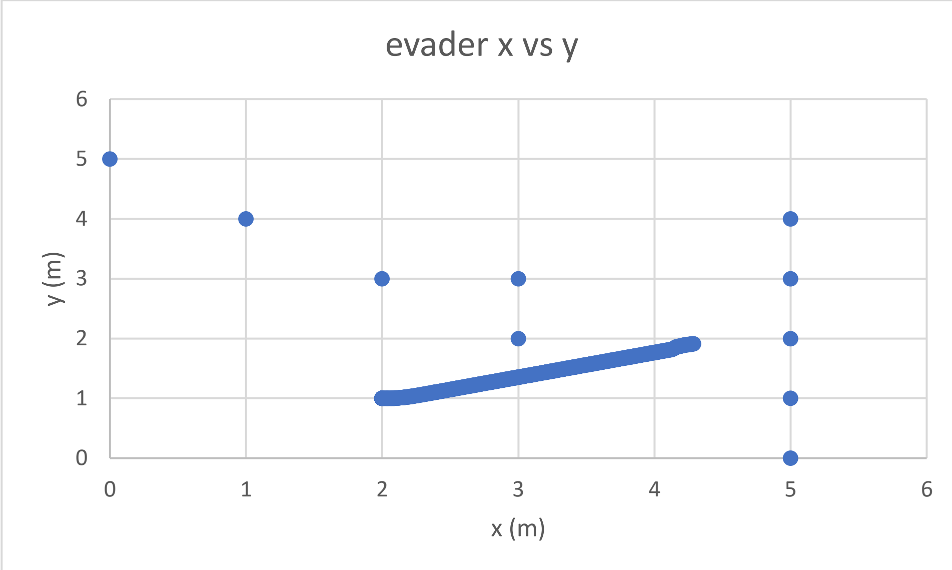
(d) What is the purpose of a Monte Carlo simulation? Additionally, give an example of a Monte Carlo simulation application (that is not the same as homework or lecture). 6%

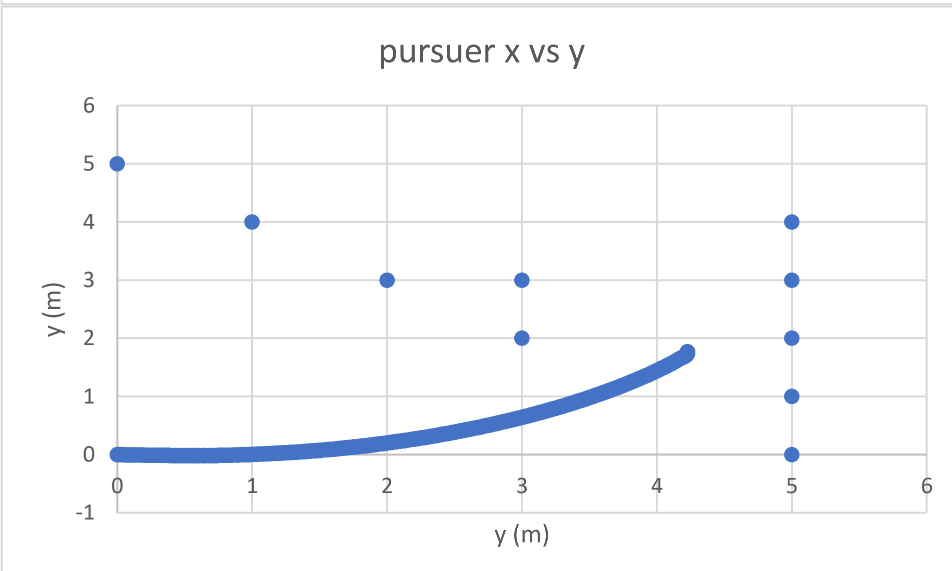
The purpose of a Monte Carlo simulation is to predict the likelihood of different outcomes when uncertainty is present by varying different parameters. Monte Carlo simulations are often used for financial planning. By simulating the market and varying what stocks to invest in and how much is invested financial planners can provide risk assessment predictions to clients.

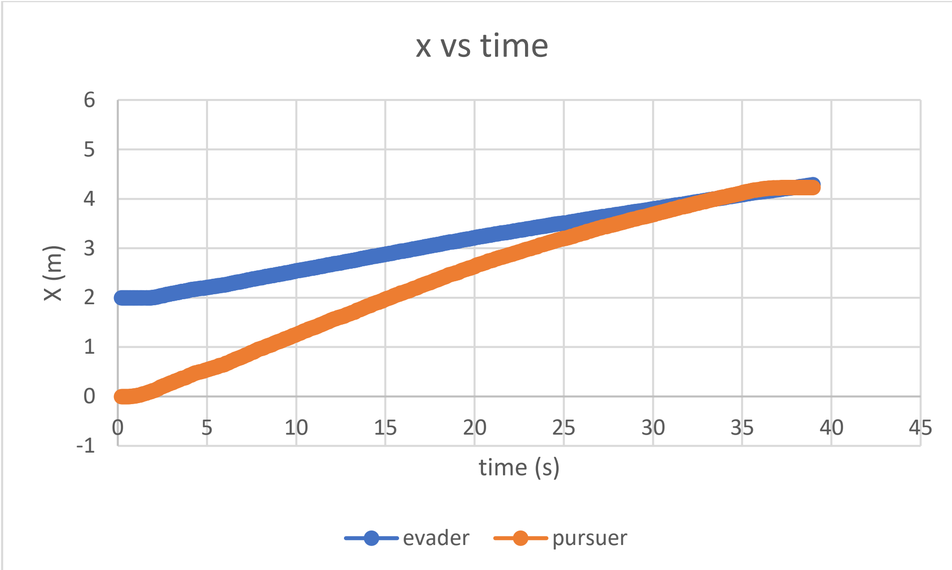
2. **Trajectory Generating and Following Evader** 30%

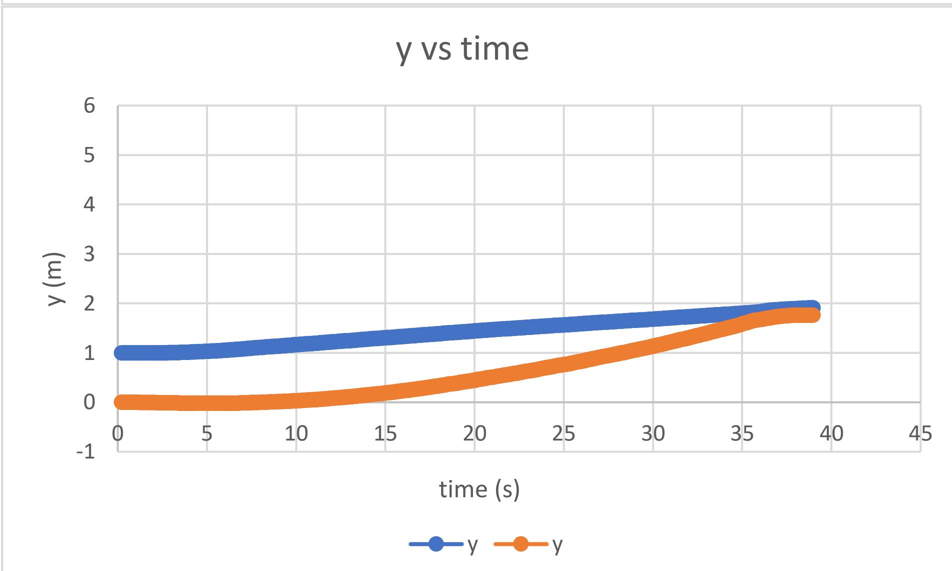
Your objective is to incorporate your Proportional Navigation pursuer turtlebot with an evader that generates and follows an optimal path from a start to the goal. **You must provide a plot of each bots' path along with the desired trajectory generated for the evader.** You are not required to include any obstacle avoidance into your pursuer path planning, but may find some small modifications are required to successfully chase (and catch) the evader prior to the evader reaching the goal location as well as make sure your evader does not collide with any obstacles. **In addition to the plot(s), provide the code used for the problem, and provide any comments/discussions necessary to interpret your results and describe any modifications you had to make in order to be successful.**

Launch File: *two turtlebot3 _test.launch*
World File: *umkc _test.world*
Obstacle List: (5,0),(5,1),(5,2),(5,3),(5,4),(0,5),(1,4),(2,3),(3,2),(3,3)
Evader Parameters: Name - *turtlebot1*, Start - (2.0,1.0), Goal - (7.0,2.0), Max Speed - 0.15m/s
Pursuer Parameters: Name - *turtlebot2*, Start - (0.0,0.0), Maximum Speed - 0.2m/s
Algorithms: Pursuer - *Proportional Navigation*, Evader - *Dijkstra's Algorithm*


evader x vs y
6
5
4
3
2
1
0
y (m)
0
1
2
3
4
5
6
x (m)


pursuer x vs y
6
5
4
3
2
1
0
-1
y (m)
0
1
2
3
4
5
6
y (m)





At approximately location 4,2 at approximately 35 seconds pursuer bot catches the evader bot and forces it off course and into an obstacle. The only modifications I made to my pursuer was the maximum speed. The only modifications I made to my evader was the maximum speed and pathfinding ability.

3. **Traveling Salesman Problem with A*** 25%

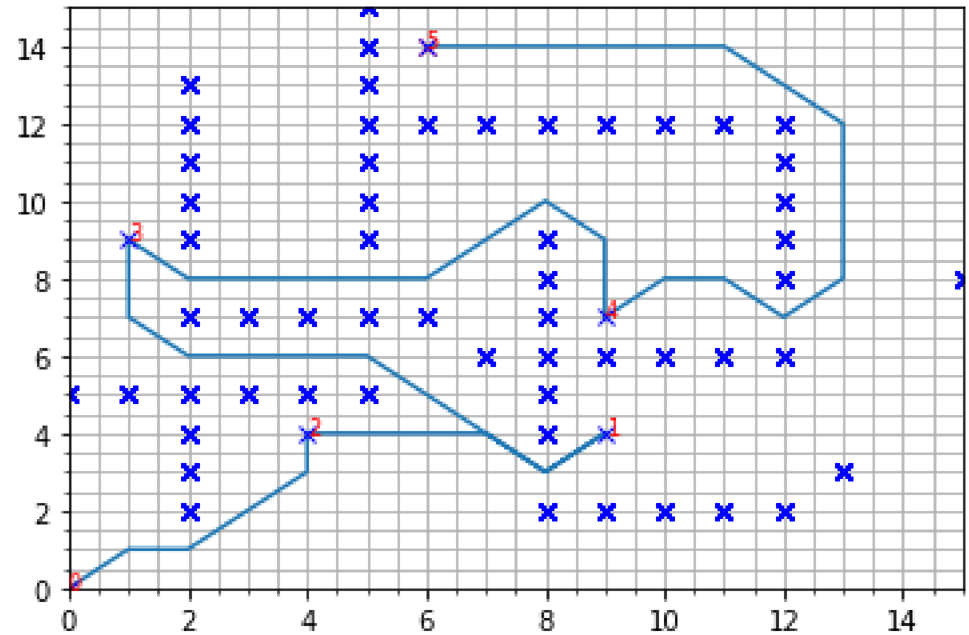
Your objective is to use the *A* Algorithm* to navigate through the modified maze (obstacle list posted below) to deliver five packages. You must start at (0,0) and visit each of the delivery points. Provide a plot showing the obstacles, A* paths, and delivery locations. Provide the total travel distance of your solution. How does this compare with the other possible paths (how good is the solution, are there other paths that have much higher travel costs)? Turn in a copy of your working code.

Note: This problem should only require small modifications to your TSP code from Homework #7 to include A* as a function to calculate the distance rather than using the distance formula for each combination of waypoints.

Grid Info: Xmin = 0, Xmax = 15, Ymin = 0, Ymax = 15, Grid Size = 0.5, Robot Radius = 0.5

Delivery Points: (9,4), (4,4), (1,9), (9,7), (6,14)
Obstacle List: (2,2), (2,3), (2,4), (2,5), (0,5), (1,5), (2,5), (3,5), (4,5), (5,5), (8,2), (9,2), (10,2), (11,2), (12,2), (13,3), (8,4), (8,5), (8,6), (8,7), (8,8), (8,9), (8,7), (2,7), (3,7), (4,7), (5,7), (6,7), (7,6), (9,6), (10,6), (11,6), (12,6), (15,8), (2,9), (2,10, (2,11), (2,12), (2,13), (5,9), (5,10), (5,11), (5,12), (5,13), (5,14), (5,15), (6,12,0.5), (7,12), (8,12), (9,12), (10,12), (11,12), (12,8), (12,9), (12,10), (12,11), (12,12)
Note: See obstacle.py on Canvas for obstacle list typed in already. This should save you from any copy-paste errors.

An example of the required plot is shown below, although this is just an example and is not representative of this exact problem.



The shortest path is = (2, 1, 3, 4, 5)
The cost is = 52.87005768508881

The path is good but I think it could be slightly better, specifically between city 4 and 5. There are many other paths that would be substantially worse.

4. **Monte Carlo Simulation** 25%

Your objective is to modify the cannon ball simulation scripts posted on Canvas in order to quantify the effects of different wind data on the landing location of the cannon ball. The file wind data.zip contains 14 wind files that will be used in your Monte Carlo simulation. Using the simulation setup parameters below, show the output from the Monte Carlo simulation (landing location is the important output). An example is shown below (this plot has different initial conditions and is just an example).

The goal is to reach the farthest range possible given the simulation scenarios. Based upon your results, what $\theta_{initial}$ should be used? What is the expected range of the best $\theta_{initial}$? Make sure to show how you calculated this value (use 95% confidence). Your answer should be based upon the information gathered from all 14 wind files (total of 14,000 simulations). Does your answer make sense? **Explain!**

Simulation Parameters:

$V_{muzzle} = 500m/s$ - Initial velocity

$z_{initial} = 1{,}250m$ - Initial altitude

$z_{final} = 1{,}200m$ - Landing altitude

$\mu_{Cd} = 0.05$ & $\sigma_{Cd} = 0.005$ - Drag coefficient and Cd st. dev.

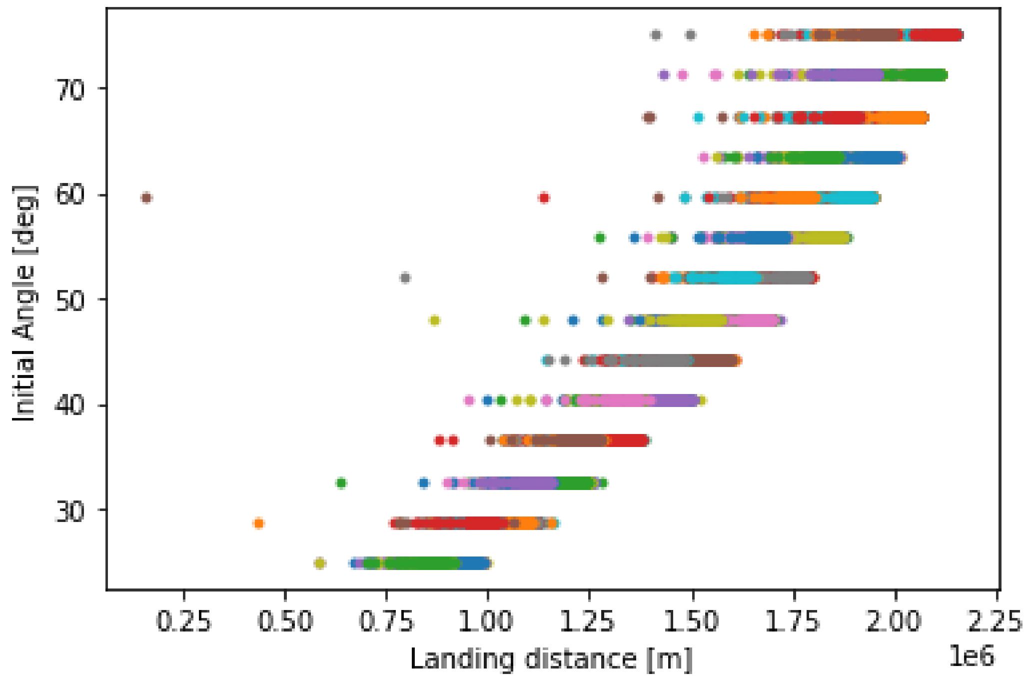
$R = 0.1m$ & $\sigma_S = 0.002m^2$ - Radius and area st. dev.

$\mu_M = 13kg$ & $\sigma_M = 0.5kg$ - Mass and mass st. dev.

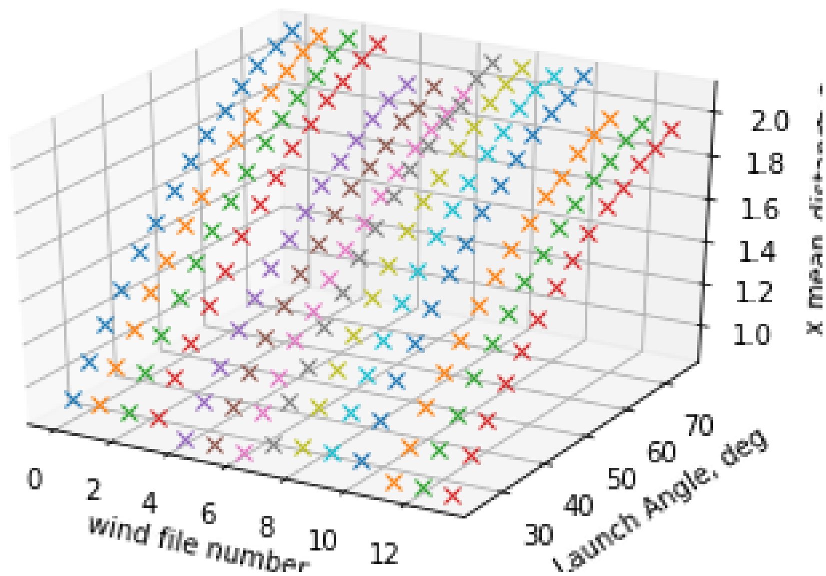
$\theta_{initial} = [25,37.5,50,62.5,75]$ - Initial angles to search

$\Delta_t = 1.0s$ - Time step

$N = 1000$ - Number of MC simulations for each configuration



Average landing distance at an initial Angle of 75 degree is 1.93 Km.



Based off my plots the highest launch angle of 75 degrees will launch the cannon ball the furthest. I didn't run all the simulations, but I knew that this result is incorrect. I only ran 200 simulations. I suspect 50 degree would be the best.

Question:	1	2	3	4	Total
Percent:	20	30	25	25	100
Score:					