Homework 1

ECE 172A
Introduction to Intelligent Systems

January 19, 2017

Make sure you follow these instructions carefully during submission. Not doing so may result in a penalty.

- Homework 1 is due by 11:59 PM, January 30, 2017.
- All problems are to be solved using MATLAB unless mentioned otherwise.
- You should avoid using loops in your MATLAB code unless you are explicitly permitted to do so.
- Submit your homework electronically by following the two steps listed below -

1. Upload a pdf file with your write-up on Gradescope. This should include your program outputs along with your answer to each question. Include your MATLAB code in an Appendix at the end of your write-up. Make sure the report mentions your full name and PID. Finally, carefully read and include the following sentences at the top of your report:

   Academic Integrity Policy: Integrity of scholarship is essential for an academic community. The University expects that both faculty and students will honor this principle and in so doing protect the validity of University intellectual work. For students, this means that all academic work will be done by the individual to whom it is assigned, without unauthorized aid of any kind.

   By including this in my report, I agree to abide by the Academic Integrity Policy mentioned above.

2. Send an email to arangesh@ucsd.edu with the subject line ECE 172A HW1. The email should have one file attached. Name this file: **ECE_172A_hw1_lastname_studentid.zip**. This file should contain all of your MATLAB scripts and functions (as .m files only!) in a folder called **code** (Note that you have to include your code in the write-up in addition to this). This should include all files necessary to run your code out of the box.

Problem 1. MATLAB basics (5 points)

(i) Input $A = \begin{bmatrix} 2 & 59 & 2 & 5 \\ 41 & 11 & 0 & 4 \\ 18 & 2 & 3 & 9 \\ 6 & 23 & 27 & 10 \\ 5 & 8 & 5 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$ in MATLAB.

(ii) Point-wise multiply A with B and set it to C.

(iii) Calculate the inner product of the 2nd row and 5th row of C.

(iv) Find the minimum and maximum values and their corresponding row and column indices in matrix C.

(v) Subtract the first row of C from all its rows (including the first row). Set the resulting matrix to D.

(vi) Repeat step (iv) for matrix D.

Functions you may want to use: *min(), max(), find(), bsxfun()*.
In your report, include:

- The result for each sub-question above.
- Your code in the Appendix.

Problem 2. Robot traversal (15 points)

Your boss wants a robot to traverse a room without colliding with any objects. To see if this task is feasible, your boss asks you to simulate such a robot.

In this problem, refer to the directions as North, East, South, and West. The goal of the simulation is to have the robot (white pixel) move from the North side of the room to the South side while avoiding the objects (dark gray pixels). The simulated robot moves by adding a vector **[v, h]** to its current position, where **v** and **h** represent the vertical and horizontal directions respectively. The North-west pixel of the room has coordinates **[1, 1]**, and the South-east pixel has coordinates **[vSize, hSize]**, where **vSize** and **hSize** are provided in the code.

(i) The framework of the simulation is provided in *nav.m*. Without changing the code, run *nav.m* by pressing **F5**, and answer the following questions:

- What does **loc** keep track of?
- In what direction should the robot move based on line 14 (adding **[1, 0]**)?
- Add a new object at **[4, 7]**. Explain what happens and why.
- If the robot were implemented based on the current state of the simulation, briefly explain why this robot would not be considered an intelligent system.

(ii) Uncomment the **if** statement and explain how this improved robot handles the object you placed at **[4, 7]**. Assuming that the function *detectObject()* can be practically implemented using sensors, briefly explain how this improved robot is a more intelligent system than the robot in part(i). Save the path of the simulated robot as an image once it reaches the goal and include it with your report.

(iii) Add another object at **[5, 5]**. Modify the code to solve this new case. The robot can only move one unit at a time (i.e. one move per iteration of the while loop), and it cannot move diagonally. Do not introduce any new variables or functions, and do not hard-code a solution. Only modify the logic in *nav.m* between the START and STOP comments. Save the path of the simulated robot as an image once it reaches the goal and include it in your report.

(iv) Finally, add another object at **[5, 7]**. Modify the code once again to solve this new case. The same rules apply - The robot can only move one unit at a time (i.e. one move per iteration of the while loop), and it cannot move diagonally. Do not introduce any new variables or functions, and do not hard-code a solution. Only modify the logic in *nav.m* between the START and STOP comments. Save the path of the simulated robot as an image once it reaches the goal and include it in your report.

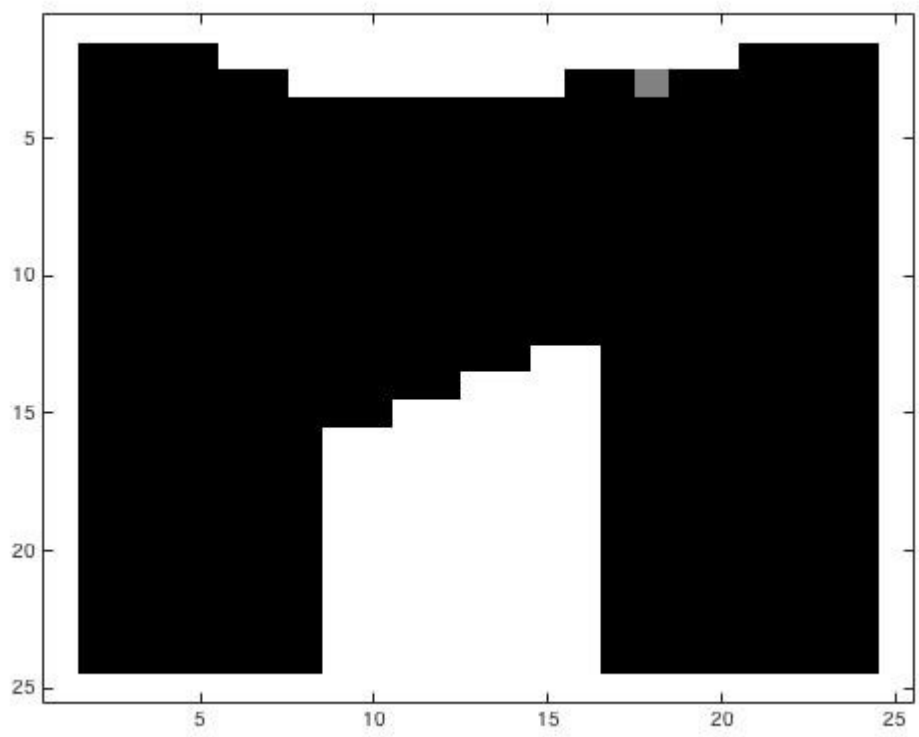
Hint: You may use the function *haveIBeenHereBefore()* that we have provided.

In your report, include:

- Answers to all questions above in no more than 2 sentences each.
- Images showing the path of the simulated robot once it reaches the goal for parts (ii), (iii) and (iv).
- Include the code block that implements your robots' logic (the part between START and STOP) for parts (ii), (iii) and (iv) in the Appendix.

Problem 3. Edge following (10 points)

You are given a floor plan for a house and are tasked with designing a robot that navigates around the inner edge of the house. Below is the floor plan and the bot location at 0% completion.



Run the main script (*runMe.m*) to see what path the bot is taking. Edit the function *get_new_pos.m* to get the job done. Read the comments inside the given files for clarification.

Your code should be able to run in both clockwise and anti-clockwise directions (controlled using **BOT_DIR** variable in *runMe.m*). The robot cannot move diagonally.

In your report, include:

- 3 snapshots of the animation window at 50% distance covered, 90% distance covered, and 50% distance covered for the second time, for both clockwise and anti-clockwise directions (total of 6 images).
- Make sure you include the contents of your *get_new_pos.m* code in the Appendix.

Problem 4. Literature Survey(10 points)

Prepare a one page report on a specific intelligent robotics related story that you find interesting. Consider the following points in your report:

1. Needs to be a relatively recent story (not older than 2010).
2. Identify the type of robot, types of sensors (both external and internal), types of planning and controllers that may be used, application domain and your assessment of the significance of such robotic systems.
3. Make sure that one of the sensory modality used is vision (some sort of camera).
4. What do you suppose are the challenges encountered in the development of such a vision subsystem?
5. Discuss any technical issues which need resolution in realizing widespread use of such robots.
6. Discuss any ethical issues which should be considered in making such robots.