Portfolio 1 Submission Instructions

Please follow the submission instructions carefully. A failure to do so will result in mark deductions. Make sure you attempt all questions in Part A, B and C.

1. All questions must be submitted via Blackboard's MATLAB grader. Multiple attempts are allowed and your best score will be counted. After submitting an answer to the problem, your grade should appear in My Grades on Blackboard.

2. Solutions to all questions must also be presented within a single pdf document and submitted via blackboard. This is in case we need to look back over your code.

3. The submitted pdf should include your student number in its same (eg. n#######Portfolio1.pdf)

4. Portfolio 1 is **due at 11:59pm Friday 20th March** and should be submitted through Blackboard. Late submissions will receive a mark of zero. If you make multiple submissions, the most recent on-time submission will be graded.

Part A - Functions (6 Marks)

As part of a robotics project, you have been tasked with writing some MATLAB functions to help model a robot's power and energy consumption. While the robot is turned on, the voltage and the current drawn is recorded at various points in time. This data is recorded in a matrix, where the first column represesnts time data (s), the second column represents voltage data (V) and the third column represents the current data (A).

Questions:

1. Create a function that accepts the data matrix as an input. The function should ouput the maximum electrical power that was drawn by the robot. Note that $P = vi$.

   (2 marks)

2. Create a function that accepts the data matrix as an input. The function should output a plot of the electrical power vs. time. In the same figure, plot the point where the maximum electrical power occurs with an 'o' marker.

   (2 marks)

3. Create a function that accepts the data matrix as an input. The function should output the total amount of electrical energy used by the robot. Energy can be described mathematically as $E = \int_{t_0}^{t_f} P \, dt$. Your function should approximate this value using the `trapz` function in MATLAB. You may assume that time is recorded in even intervals in the data matrix.

   (2 marks)

Useful Functions: `max`, `plot`, `trapz`.

Part B - Conditional Statements (6 Marks)

The next task assigned for the robotics project is to write some MATLAB functions that help with the robot's decision making.

Questions:

1. The robot is initially equipped with a sensor on its front. The sensor records the distance to the nearest obstacle measured in cm. The rules for how the robot moves are as follows:

   - The robot should move forward if the obstacle is at least 20 cm away.
   - Otherwise, the robot should stop.

   Create a function that accepts a scalar input representing the distance reading from the sensor (in cm). The function should output what decision the robot should make. It should output 'F' if it should move forward or 'S' if it should stop.

   (2 marks)

2. The robot has some extra sensors added in order to navigate around a 2D maze. It now has a sensor on its left, a sensor on its front, and a sensor on its right. Each sensor records the distance to the nearest obstacle in that direction (measured in cm). The rules for how the robot moves are as follows:

   - The robot is only allowed to move in a direction if the obstacle is at least 20 cm away.
   - If there are multiple allowable directions, the robot decides as follows:
     - The robot always moves forward if allowed.
     - The next preference is to turn left or right. If both directions are allowed, the robot should turn in the direction with the farthest obstacle. If the obstacles are equal distances away, it will choose right over left.
     - The robot should stop if it is not allowed to move in any of the directions.

   Create a function that accepts three scalar inputs: a left sensor reading, a forward sensor reading and a right sensor reading (all in cm). The function should output what decision the robot should make. It should output 'F' if it should move forward, 'L' if it should turn left, 'R' if it should turn right, or 'S' if it should stop.

   (4 marks)

Useful Control Structures: `if`, `elseif`, `else`.