University of Westminster
Department of Computer Science & Engineering

6ELEN018W – Applied Robotics Coursework (Semester 1)

Module leader	Dr D. Dracopoulos
Unit	Coursework
Weighting:	50%
Qualifying mark	30%
Description	
Learning Outcomes Covered in this Assignment:	- Design, implement and evaluate velocity controllers and trajectory controllers in robotics applications; - Evaluate the behaviour of complex robotic systems
Handed Out:	December 2025
Due Date	5/1/2026 13:00
Expected deliverables	1. Python code in the form of Jupyter notebook files for modelling and control (*.ipynb files) and all necessary input files to run the code 2. Report in PDF format
Method of Submission:	Online via Blackboard
Type of Feedback and Due Date:	Individual feedback via Blackboard within 3 weeks of submission **All marks will remain provisional until formally agreed by an Assessment Board.**

Assessment regulations

Refer to section 4 of the "How you study" guide for undergraduate students for a clarification of how you are assessed, penalties and late submissions, what constitutes plagiarism etc.

Penalty for Late Submission

If you submit your coursework late but within 24 hours or one working day of the specified deadline, 10 marks will be deducted from the final mark, as a penalty for late submission, except for work which obtains a mark in the range 40 – 49%, in which case the mark will be capped at the pass mark (40%). If you submit your coursework more than 24 hours or more than one working day after the specified deadline you will be given a mark of zero for the work in question unless a claim of Mitigating Circumstances has been submitted and accepted as valid.

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website :**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

It is recognised that on occasion, illness or a personal crisis can mean that you fail to submit a piece of work on time. In such cases you must inform the Campus Office in writing on a mitigating circumstances form, giving the reason for your late or non-submission. You must provide relevant documentary evidence with the form. This information will be reported to the relevant Assessment Board that will decide whether the mark of zero shall stand. For more detailed information regarding University Assessment Regulations, please refer to the following website:**http://www.westminster.ac.uk/study/current-students/resources/academic-regulations**

6ELEN018W Applied Robotics - Assignment
Deadline 5/1/2026, 13:00

Dr Dimitris C. Dracopoulos
Email: `d.dracopoulos@westminster.ac.uk`

Your task for this coursework is to control a mobile robot in an unknown environment. The robot objective is to navigate the unknown environment in an optimum (or near optimum) way while interacting with it, receiving rewards and penalties. The robot uses intelligent control algorithms to control its actions and find its optimum path.

You will be assessed for 4 parts (the details of which are described in the following sections of this specification):

1. Implement a reinforcement learning control approach which the robot uses to determine optimum actions, based on rewards and penalties received by the environment. The Q-learning algorithm will be used.
2. Implement a neural network approach which controls the robot and determines its next action. Python and the `sklearn` library will be used.
3. Implement the reinforcement learning control approach for different mazes which are defined and read in a file.
4. Write a short report (no more than 5 pages) discussing your results with the inclusion of appropriate diagrams.

The Problem

It is highly desirable to have robots which are able to adapt and operate in unknown or changing environments and circumstances which they have not encountered before.

Here, we consider a mobile robot which navigates an unknown environment, trying to find the optimum path to a goal location while avoiding collisions with the obstacles. The environment is a maze shown in Figure 1. A black square represents a location where there is an obstacle and the robot cannot move to such a cell.

At every time step t, the available actions for the robot are: move north (`North` action), move east (`East` action), move south (`South` action) or move west (`West` action).

The starting and the target (goal) positions of the robot could be any square in the maze which is not occupied by an obstacle. The robot's task is based on the current position and a pre-specified end position (the goal): to take a optimum action at time step t so that the time required (number of steps) to reach the destination (target) is minimised.

The number of combinations of all possible inputs, i.e. (starting position, target goal position) is $(28 \times 27 = 756)$.

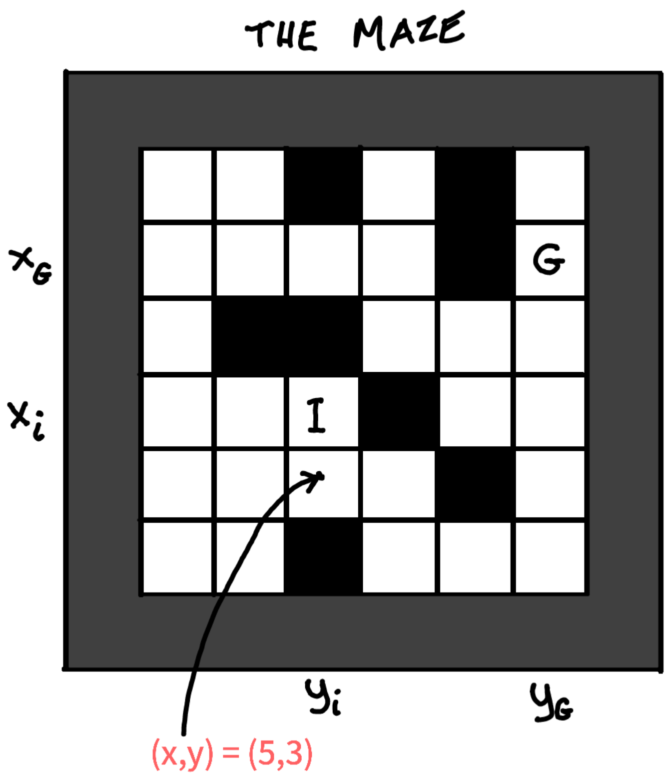

Figure 1: The maze environment for the robot for tasks 1 and 2. I is the initial position of the robot with coordinates (x_i, y_i). G is the target (goal) position of the robot with coordinates (x_G, y_G).

1 Reinforcement Learning Control of the Robot

Implement using Python the Q-learning control algorithm we have seen in the lectures and tutorials. You should implement this from scratch, based on the code given to you in the tutorials and extending it.

You are NOT ALLOWED to use any library which implements the algorithm.

The robot learns the optimum Q values for determining the optimum action at every time step t while it interacts with the environment.

You should define the rewards and penalties for each state transition in such a way that the robot is able to find the optimum path (using the minimum number of steps) to the target goal location.

There are different optimal Q values for each target location of the robot and you should call your Q-algorithm implementation (e.g. a Python function) multiple times to evaluate these optimal Q values. You should save these to an appropriate Python data structure.

The robot uses these learned Q values to determine the next optimal action at the current time t. This can be done by just looking ahead at the action that maximises Q at the current location.

In your report, you should discuss how well the Q-learning algorithm of the robot works using appropriate methodologies such as metrics and/or diagrams.

2 Neural Network Control of the Robot

Implement using Python and the `sklearn` library a multilayer perceptron using the backpropagation algorithm to determine the next control action of the robot.

The robot uses the implemented neural network to decide (predict) which action to take next.

For training the neural network you should use optimum actions learned from the application of the Q-learning algorithm that you developed in the first part of this coursework. You should create a file with appropriate data (756 rows) corresponding to the form

```
(current_position_x, current_position_y, goal_x, goal_y) -> optimum_action
```

and then split the data into training and testing datasets. This file should be created by running your Q-learning algorithm for all the possible 756 configurations corresponding to all possible combinations of robot locations and destinations.

You should split the data into appropriate sizes of training and testing sets so that the neural network of the robot is able to generalise best on the testing set.

In your report, you should discuss how well the neural network of the robot performs for the training and the unseen data. This should be done using appropriate methodologies such as metrics and diagrams.

3 Reinforcement Learning Control of the Robot for Different Mazes

For this task, you should implement the Q-learning control algorithm so that the robot can control itself and find its optimum way for a maze of any size with any obstacle configuration (i.e. the obstacles can be located in different squares for different maze environments).

Your program should be able to read from a file the configuration of the maze (size and location of the obstacles) as well as the starting position and goal position of the robot. The format of the file MUST follow the format below:

```
6 5  # size of the maze, i.e. 6 rows and 5 rows
1 2  # starting position of the robot is row 1, column 2.
3 5  # target (goal position for the robot) is row 3, column 5
0 0 0 0 0
0 1 0 1 0
0 0 1 0 0
0 1 0 0 0
0 1 0 1 0
0 1 0 0 1
```

In this example, the last 6 rows describe the maze and a 0 corresponds to an empty location and a 1 corresponds to a location where there is an obstacle.

The top left square of the maze corresponds to row 1, column 1.

In your report you should demonstrate how well your implementation works, in order to find the optimum path of the robot for different starting and goal positions and for at least 2 different mazes.

4 Technical Report with Analysis of your Solution

You are expected to write a short report **in PDF format (no more than 5 pages)** which you will discuss all of your implementations for the 2 tasks.

You should discuss how you chose the split of training and testing data for the neural network as well as how you defined the rewards for the reinforcement learning approach.

Discuss in detail any of the choices made for the 2 different controllers and how they affect the performance of the control of the robot. You should include metrics, such as RMSE (root mean square error) and others, as well as any diagrams you think will demonstrate how well the controller performs.

Marking Scheme:

1. **Reinforcement Learning Controller**
   - Python implementation of the Q-learning controller: *20 marks* (correctness: 15 marks, efficiency: 3 marks, readability: 2 marks — The 15 marks for correctness will be awarded as:

- 15 marks for a totally correct implementation
  - 11–14 marks for a very good implementation with minor omissions/mistakes
  - 6–10 marks for an implementation with major mistakes/omissions
  - 3–5 marks for a poor implementation
  - 0–2 marks for a sketchy or not very relevant implementation.

)

- Correct choice of the next optimal action based on calculated Q values: *9 marks*
  - 9 marks for optimal choice
  - 6–8 marks for an appropriate but not optimal choice
  - 3–7 marks for a poor action choice
  - 0–2 marks for an incorrect choice.

2. **Neural Network Controller**

- Python implementation: *10 marks*
  - 10 marks for a totally correct implementation
  - 7–9 marks for a very good implementation with minor omissions/mistakes
  - 4–6 marks for an implementation with major mistakes/omissions.
  - 0–3 marks for a poor implementation or not using the required libraries.
- Appropriate split of data sets: *3 marks*
- Correct construction of training/testing data based on the Q-learning implementation: *7 marks*
  - Correct construction of data: 7 marks
  - 5–6 marks for minor mistakes/omissions
  - 3–4 marks for major mistakes/omissions
  - 0–2 marks for poor or totally incorrect construction of the data.
- Calculation of appropriate metrics: *5 marks*
  - Correct calculation of a number of different metrics: 5 marks
  - Correct calculation for inadequate choice of metrics: 3–4 marks.
  - Poor or incorrect calculation of metrics: 0–2 marks.
- Appropriate inputs and outputs and values of the parameters for the training, as given by the specification: *3 marks*

3. **Reinforcement Learning Controller for Different mazes**

- Reading the maze from an input file and setting up a data structure describing the maze: *3 marks*
- Python implementation: *12 marks*
  - 12 marks for a correct implementation
  - 8–11 marks for a very good implementation with minor omissions/mistakes
  - 4–7 marks for an implementation with major mistakes/omissions.
  - 0–3 marks for a poor implementation or not using the required libraries.

4. **Analysis Technical Report**

- Justification of choices made : *10 marks*

- Excellent coverage of the justification: 9–10 marks.
  - Good but with minor omissions/mistakes justification 6–8 marks.
  - Major omissions/mistakes or insufficient coverage of justification: 3–5 marks.
  - Incorrect or no relevant justification: 0–2 marks.
- Metrics and diagrams explaining the results obtained: *10 marks*
  - Excellent choice of metrics and diagrams with appropriate explanation: 9–10 marks.
  - Limited choice of metrics/diagrams or without sufficient coverage of the explanation: 6–8 marks.
  - Incorrect explanation or poor choice of metrics and diagrams or diagrams not clear enough: 3–5 marks.
  - Irrelevant choice of metrics/diagrams or almost not existing discussion of them.
- Clarity of the technical report and completeness: *5 marks*
  - Clarity: 2.5 marks
  - Completeness: 2.5 marks
- Citations and references: *3 marks*
  - Proper citations: 3 marks.
  - Sufficient and relevant references: 2 marks.
- Penalty for exceeding 5 pages: *-20 marks*

Submission of assignments using a different method other than Blackboard will not be accepted and zero (0) marks will be awarded in such cases.

Deadline: Monday 5th of January 2026, 13:00.

Submission Instructions

Files to submit: All the Python Jupyter notebooks of your code, any required input files to run the code and the report in PDF format. **The notebook files should run on Jupyter lab and notebooks which run only in other environments will receive no marks**. All your files should be submitted in a single zip file.

You should submit via BlackBoard's Assignment functionality (do NOT use email, as email submissions will be ignored.), all the files described above. A single zip file with the name `wNNNNNNNN` (where `wNNNNNNN` is your university ID login name) containing all the above files should be submitted.

Note that Blackboard allows you to make a submission multiple times. Make sure before submitting (i.e. before pressing the `Submit` button), that all the files you want to submit are contained there (or in the zip file you submit).

In the case of more than one submissions, only your last submission before the deadline given to you will be marked, so make sure that all the files are included in the last submission attempt and the last attempt is before the coursework deadline.

Request to mark submissions which are earlier than the last submission before the given deadline will be ignored as it is your responsibility to make sure everything is included in your last submission.

The following describes how to submit your work via BlackBoard:

1. Access `https://learning.westminster.ac.uk` and login using your username and password (if either of those is not known to you, contact the Service Desk, tel: +44 (0) 207 915 5488 or log a call via `https://servicedesk.westminster.ac.uk`.).

2. Click on the module's name, `MODULE: 6ELEN018W.2025 APPLIED ROBOTICS` found under `My Modules & Courses`.

3. Click on the `Assessment->Submit Coursework->Coursework`.

4. Click on `View Assignment`.

5. Attach your zip file containing all of the required files, by using the `Browse` button.

6. Create a Word or PDF file with the following information:

   - *Comments*: Type your full name and your registration number, followed by:
     "I confirm that I understand what plagiarism is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged."

7. Attach the file with the statement above.

8. Check that you have attached both the zip and the statement file.

9. Click the `Submit` button.

If Blackboard is unavailable before the deadline you must email the Registry at `studentcentre@westminster.ac.uk` with **cc:** to myself and your personal tutor before the deadline with a copy of the assignment, following the naming, title and comments conventions as given above and stating the time that you tried to access Blackboard. You are still expected to submit your assignment via Blackboard. Please keep checking Blackboard's availability at regular intervals up to and after the deadline for submission. You must submit your coursework through Blackboard as soon as you can after Blackboard becomes available again even if you have also emailed the coursework to the above recipients.