College of Engineering & Physical Sciences
Assignment Brief
[DRAFT – PENDING OF MODERATION]

CS4790 Robotics and Autonomous Systems	**Project**
Dr Martin Rudorfer - Room MB214G - Email: m.rudorfer@aston.ac.uk Dr Luis Manso - Room MB214D - Email: l.manso@aston.ac.uk	Preferred contact method: 1. **In-class interaction** 2. Office hours 3. Email

Assignment Brief:

This document describes the "*Project*" Individual Assignment component of the CS4790 *"Robotics and Autonomous Systems"* module. The objective of this practical assessment is to allow you to demonstrate your skills in, and understanding of, the design, implementation and composition of algorithms and methods for autonomous robot control. This piece of assessment is worth **75%** of the module.
It covers the Learning Outcomes 1-3 from the module specification:

1. Apply perception algorithms to extract information from sensors.
2. Apply classical machine learning algorithms to processed sensor data in order to recognise objects in the environment and for navigation.
3. Design a control method for a robotic system in a given context by selecting and composing sensing, classical machine learning and decision-making algorithms.

The project involves programming a robot simulated using Webots R2023b in a series of three missions covering different units of the module:

- Perception and behaviour coordination for a mobile robot
- Navigation with reactive obstacle avoidance
- Control of an autonomous manipulator for a pick-and-place task

You will need to submit **each mission project** and demonstrate it in a live demo in which you will be asked to explain how your robot works – see below for further details of grading and the submission requirements.

The three missions consider different types of robots in different contexts and will help you get a breadth of experience. The world, scenario, and robots will be given in each mission, and it is your task to design and implement the controller to achieve the desired behaviours. For each mission, you will be given a template project which includes the world and the API for the robot. **You are not allowed to change any part of the world file or the robot API.** Other than that, you can freely add new Python modules that your controller uses and also make use of 3rd party libraries (if referenced appropriately and used within terms & conditions, see details below).

Any **documentation should be included as code comments**. We will not consider any other documents.

Each mission gives 25 marks; the three missions add up to 75 marks in total.

If you find anything in the brief that is not clear, please let the module tutors know, so that it can be clarified.

Additional tips:

- Make sure you create regular backups. Using a version control system such as Git is highly recommended.
- Include in your submission anything you may have implemented even if it doesn't work.
- Make sure you overwrite the world file (not the code) with the one in the template before submitting (you cannot do the demo with a modified world file).

Descriptive details of Assignment:

- **Mission 1: Perception and Behaviour Coordination (25 marks)**
  - Scenario: You are developing a controller for an autonomous robot in a simulated environment populated with pins. The robot is equipped with an RGB camera, a LiDAR, infrared sensors, a GPS and a compass.
  - Desired behaviour: The robot should knock down all pins as fast as possible without assuming any location of the pins (use the robot's sensors instead).
  - In your code comments, you should provide a rationale for how you implement all behaviours as well as how you coordinate them.

- **Mission 2: Navigation with reactive obstacle avoidance (25 marks)**
  - Scenario: In a warehouse, there are different rooms to store goods. A mobile robot is used to transport items between them. In each room, there is a pick-up/drop-off location which the robot should navigate to (i.e. the goals are those locations, not the rooms where they are). The tasks are given by a sub-system that is simulated using the keyboard and include the source and target location (from A to B). New tasks may come up at any point in time and should be queued.
  - Desired behaviour: The robot should **purposefully** navigate to the target room as well as to the target position within the room. The robot should avoid any static or moving obstacles. Upon fulfilling its tasks, it should be ready to take further instructions.
  - In your code comments, you should provide a rationale for how you solve

global planning, local planning, and obstacle avoidance. You should also provide a rationale for your behaviour coordination strategy.

- **Mission 3: Pick-and-place with an autonomous manipulator (25 marks)**
  - Scenario: An autonomous manipulator is equipped with a camera at its end-effector in addition to the sensors in mission 1. The existing controller already provides behaviours to move the robot in joint and/or task space and to open/close the gripper. There are several objects as well as a tray on a table.
  - Desired behaviour: The robot should be able to detect the objects on the table using its camera (the tray will always be in the same position and does not need to be detected). The robot should pick the objects one by one and drop them into the tray.
  - In your code comments, you should provide a rationale for how you solve object detection and approaching the objects with the gripper.

For each mission, there are detailed instructions in the given template. You are only allowed to modify the controller of your robot, and not any other files.

Note that accomplishing the desired behaviours is only part of the mark, both your rationale and the documentation within the code count as well. See marking rubric at the end of this document.

Recommended reading and other online sources:

Please use existing Python packages. Whenever you directly use or copy code from third-party resources, you **MUST EXPLICITLY STATE** in the documented code what pieces of code have been copied and where the code has been copied from. If you fail to do so, you will not get marks for writing such code and you will most likely be **reported for academic malpractice**.

Please make sure that you are aware of the Regulations on Student Discipline:

https://www2.aston.ac.uk/clipp/documents/Quality/Regulations/2021-22/used/final-au-rsc-20-3967-a-regulations-on-student-discipline-202122.pdf

Key Dates:

The following are the most relevant key dates and times for the coursework:

03/06/2025 by 15:00	Mission 1
24/06/2025 by 15:00	Mission 2
15/07/2025 by 15:00	Mission 3

Late submissions will be treated under the standard rules for the School, with an **absolute** deadline of one week after which submissions will not be marked. The lateness penalty will be 10% of the awarded marks for each working day.

Submission Details:

Submissions will be *on-line* through Blackboard as a single ZIP file for each mission. The name of the files must be as follows:

- M1: <student_id>_M1.zip
- M2: <student_id>_M2.zip
- M3: <student_id>_M3.zip

For each mission, there will be a live demo scheduled with the module tutors in which you demonstrate how your robot works and explain your design decisions. The demos are **mandatory** and can take place via MS Teams or in person.

You need to demo the exact same version that you submitted.

Marking Rubrics:

For each of the missions you will receive up to 25 marks. The demo is mandatory, no marks will be given for the mission without a demo. You will get marks for the functionality that you **can** explain.
The missions are marked independently and the whole coursework gives in total up to 75 marks. Each mission will be marked based on your rationale, the performance of the robot, and the documentation of your code, as follows:

Rationale	**9**
There is little or no evidence of principled design decisions.	0-2
There is some evidence of principled design decisions, but they fall below the expected threshold or the student fails to explain the general design principles of their solution.	3-4
Adequate, principled design decisions have been made based on the content of the module and possibly further research.	5-6
The rationale is informed by the content of the module as well as further research to make the best possible design decisions.	7-9
Performance	**8**
The code does not run, the robot does not move, or there are other obvious flaws such as assuming the position of the obstacles or objects to interact with (this does not include walls or tables).	0-1
The code works, but there are major flaws which prevent the robot from executing the desired behaviour. There are significant gaps in the explanations of the algorithms developed or their implementation.	2-3
The implementation is generally sound, and the robot executes the desired behaviour with only minor errors or inconsistencies (if any).	4-7
The robot behaviour is excellent and flawlessly masters a variety of edge cases (e.g., including obstacles). The student can explain how their implementation works in detail.	8-9
Documentation	**8**
There may be some comments in the code, but they fall short of the expected documentation.	0-2
Most functions and classes have been documented with their purpose, inputs, and outputs, and additional comments throughout help	3-5

understanding the code. Relevant design decisions can be inferred from the comments.	
All functions are documented with their purpose, inputs, and outputs. Additional comments throughout help understanding the code. The rationale and design decisions are clearly stated.	6-7