



BEng, BSc, MEng, MMath and MSc Degrees 2024–25

Open Assessment

Department Computer Science

Module Autonomous Robotic Systems Engineering (AURO)

Title Individual coursework

Issued: Monday 04/11/2024

Submission due: 12 noon, Monday 13/01/2025

Feedback and Marks due: Monday 17/02/2025

Word limit 2000 words for the report

Allocation of Marks 40% allocated to implementation and 60% to the report

Instructions

All students should submit their answers through the appropriate VLE submission point in the Assessment area of the VLE site by 12:00 noon on Monday, 13/01/2025. An assessment submitted after this deadline will be marked initially as if it had been handed in on time, but the Board of Examiners will normally apply a lateness penalty.

Your attention is drawn to the section about Academic Misconduct in your [Departmental Handbook](#).

Any queries you may have on this assessment should be posted on the Discussion Board on the Virtual Learning Environment (VLE) page for Autonomous Robotic Systems Engineering (AURO) in the appropriate discussion area. **No questions will be answered after 23/12/2024.**

Note on Academic Integrity

This is an open assessment, and you are permitted to refer to written and online materials to aid you in your answers. However, you must ensure that the work you submit is entirely your own, and for the whole time the assessment is live, you must not:

- communicate with other students on the topic of this assessment.
- seek advice or contribution from any other third party, including proofreaders, friends, or family members.

We expect, and trust, that all our students will seek to maintain the integrity of the assessment, and of their award, by ensuring that these instructions are strictly followed. Where evidence of academic misconduct is evident, this will be addressed in line with the [Academic Misconduct Policy](#) and, if proven, be penalised in line with the appropriate penalty table. Given the nature of these assessments, any collusion identified will normally be treated as cheating/breach of assessment regulations and penalised using the appropriate penalty table (see AM3.3 of the Academic Misconduct Policy).

Please note: The publication of your assessment submission online or elsewhere, or the sharing of these documents with others, is **NOT PERMITTED** until all students have completed the module assessment and reassessment (this will include any students with extensions to their reassessment date). We will notify you through the VLE at such time.

1. Task definition

This assessment focuses on the retrieval of items by autonomous mobile robot(s). You will be provided with a simulated world containing items of different types distributed throughout the environment, which the robot(s) must collect and deposit in the appropriate zone(s). This simulated world will also contain obstacles that the robot(s) need to avoid.

A robot can collect an item by driving into it and using a functionality to collect it, and will 'hold' the item until it uses another functionality to offload the item. Once an item has been collected and successfully returned to a zone, a new one will automatically spawn to replace it.

You must design and implement a solution to this task, using mobile robot(s) that autonomously collect the items and return them to the appropriate zone(s) in an efficient manner.

2. Deliverables

This assessment is worth 100% of the module mark, and is made up of 2 parts with the following weightings:

1. **Implementation:** 40%
2. **Report:** 60%

Guidance on what you are required to do for each part of the assessment is provided in the following sub-sections.

2.1 Implementation (40%)

You should engineer an autonomous robotic system that solves the assessment task efficiently, while obeying the following constraints:

- You must implement your solution in simulation, using the Robot Operating System (ROS). Specifically, you must use ROS 2 Humble Hawksbill, Gazebo Classic 11, and the Python client library `r1cpy`.
- You must use autonomously controlled (i.e. not teleoperated) TurtleBot3 Waffle Pi robot(s). Your solution may use up to 3 robot(s).
- You must include a README file that describes how to run your code, and details any packages or environment variables that your implementation depends on. It should also detail what scenarios your implementation should be exercised on, including values used to configure nodes and launch files as appropriate.

There are many different ways of approaching the assessment task, so there is scope for a variety of solutions. Your implementation will be assessed against the following criteria:

How effectively the task is achieved [25 marks]: You will be assessed on the intelligence of your autonomous robotic system, and how effectively it addresses different aspects of the task.

Multi-robot systems may be more sophisticated and perform the task more efficiently, but only if they are implemented appropriately. Marks will be awarded per the following criteria.

- Appropriateness of the obstacle avoidance solution [5 marks]
- Effectiveness of the search strategy [5 marks]. At most [3 marks] for a single robot solution.
- Effectiveness of the collection strategy [5 marks]. At most [3 marks] for a single robot solution.
- Efficiency of the task and how correctly it achieves the goals [10 marks]. At most [6 marks] for a single robot solution.

Use of concepts and comments [10 marks]: You will be assessed on the modularity of your solution and on the breadth of the ROS features that your implementation uses, and how well you have demonstrated your understanding of them (through correct usage). Your implementation should be easy to understand. You will be assessed on the structure of your code, the use of sensible naming conventions and comments, and how comprehensive your README file is. Marks will be awarded per the following split.

- Architecture [5 marks]
- Modularity [5 marks]

Provision of specific scenarios for running your solution [5 marks]: The instructions for running your solution should specify the simulation scenarios you specifically considered when designing your solution, including the parameters of launch files and or nodes. You will be assessed on how well the provided scenarios exercise your solution. Marks will be awarded per the following criteria.

- At most 1 mark per valid scenario. Diversity of scenarios will be taken into account.

You will be provided with code that defines the assessment task, which should be downloaded from the VLE. This code sets up the simulated world, and runs a ROS node that spawns items and keeps track of when they are collected and returned to the appropriate zone(s) by the robot(s). You will also be provided with a ROS node that processes images from a robot's camera, and publishes information about any items that are detected. You **must** use this code **without modification**, as it defines the assessment parameters of the task.

You are allowed to use any of the example code provided as part of the teaching delivery for the AURO module, including solutions to practical exercises. The use of 3rd party packages is also permitted, **as long as** you cite the original source and include instructions on how to build them.

Note that a solution that does not build correctly may be awarded 0 marks.

2.2 Report (60%)

You must write a report that details the design and implementation of your solution. You should describe the design of your solution, analyse the performance of your solution, and present the results in your report. Finally, your report should evaluate the strengths and weaknesses of your solution, and reflect on related safety implications and ethical considerations. The report **must** be structured as follows:

1. **Overview [5 marks]:** This section should describe at a high-level what methodology you used in your solution. You should explain what influenced your approach to solve the given problem [200 words]. Marks will be awarded for a concise description.
2. **Architecture [13 marks]:** This section should have a diagram illustrating the high-level architecture and a description of the architecture using the diagram. You should identify the nodes responsible for implementing the control strategy adopted [300 words]. Marks will be awarded per the following criteria.
 - Appropriateness of the diagram [3 marks]
 - Depth, breadth and clarity of the discussion of the chosen diagram [5 marks]
 - Correct identification and description of nodes [5 marks]
3. **Control [12 marks]:** In this section, you should provide RoboChart state machines for the control nodes identified in Section 2 on Architecture. You should explain the state machines used [300 words]. Marks will be awarded per the following criteria.
 - Appropriateness of the control architecture used for the task [3 marks]
 - Depth, breadth and clarity of the description of the machines used [3 marks]
 - Demonstrated understanding of the design and its behaviour [3 marks]
4. **Evaluation [10 marks]:** In this section, you should provide an evaluation of your solution through one or more methodologies, for example, simulation, complexity analysis, data analysis, descriptive, experimental [600 words].
 - At most 3 marks will be awarded per chosen evaluation method [9 marks]
 - Justification of the chosen evaluation methods [1 mark]
5. **Safety and ethics [10 marks]:** This section should include a discussion of the ethical and safety implications of the problem and your solution in particular. You should also reflect on how these topics would relate to your solution's approach, if it were implemented in a real-world scenario [300 words].
 - Identification of ethical and safety implications [3 marks]
 - Depth, breadth and clarity of the discussion on the implications [3 marks]
 - Demonstrated understanding of the implications for real-world scenarios [4 marks]
6. **Simulation scenario [5 marks]:** This section should include screenshot(s) of the simulation and explain and reflect on the overall strategy you have implemented. You should justify the choice of simulation scenarios [300 words].

The remaining **[5 marks]** of the report mark will be based on its presentation (structure, figures, adherence to the structure, and use of referencing).

Your report should be formatted using **A4** paper size, with a minimum font size of 12pt and minimum margins of at least **2cm**, and any references should follow the IEEE referencing style¹. The report must not exceed the word limit of 2000 words (excluding references). If your report exceeds this

¹ <https://subjectguides.york.ac.uk/referencing-style-guides/ieee>

word limit, the marker will stop reading when they reach the limit, and base the mark on what they have read so far. There is no limit on the number of images.

3. Submission

The submission of your implementation and report for this assessment should abide by the following rules.

3.1 Anonymity

You should identify yourself only by your examination number. Your name, username, email address, or any other identifying information must not be present anywhere in your submission.

This includes all ROS package metadata, code comments, environment variables, Linux terminal prompts, PDF metadata, etc. You must not include any images that could deanonymize your submission.

3.2 Electronic submission

You should submit your deliverables via the submission point on the VLE as **two separate files**: your report should be submitted as a PDF file, and your implementation should be submitted as a single ZIP file. The contents of the ZIP file should be structured as shown in Figure 1.

```
submission.zip
├── README.txt
├── rosgraph.png
└── src
    └── Your ROS packages
```

Figure 1: Required structure of electronic submission ZIP file.

The source code of your implementation should be included in a directory called `src` containing your ROS packages. You **should not** include your entire ROS workspace – i.e. you should exclude the `build`, `install`, and `log` directories created by `colcon`.

Your README file should be in plain text format. You should also include a PNG file exported from `rqt_graph` in a file named `rosgraph.png` that shows the ROS graph of your solution.

Your report must be a single PDF file.

It is your responsibility to ensure that your implementation runs under the ROS 2 Linux environment provided to you as part of AURO before submission.

END OF PAPER