Individual Coursework

I. Overview

This project used a state-based control architecture to develop an autonomous robotic system [12]. To simplify complex behaviours the robot's behaviour was separated into operational phases including searching, navigating, collecting and depositing items. This separation created modularity, allowing the robot to focus on specific tasks without interference, improving performance and enabling future developments to the solution.

Transitions were determined by real-time sensor data, LiDAR scans provided obstacle detection enabling real-time navigation adjustments ensuring safe travel [6]. Camera inputs dynamically identified items and zones based on their properties, and odometry data ensured accurate positioning and orientation [2]. By separating data processing to the relevant states, this optimised decision-making, reduced computational requirements and improved efficiency [7].

This solution improved performance through state-specific checks and logic. When searching rotational scans were used to detect objects, and during navigation dynamic path adjustments were made allowing for efficient navigation while avoiding collisions [3]. This allowed the robot to respond to challenges, like lost targets and other robots, without failure.

This approach prioritised modularity, adaptability and scalability to provide effective task completion. This solution supports multi-robot system, balancing real-time collision avoidance with efficient search and navigation methods [11]. Ultimately providing an effective and robust solution for autonomous item collection and deposition.

II. Architecture

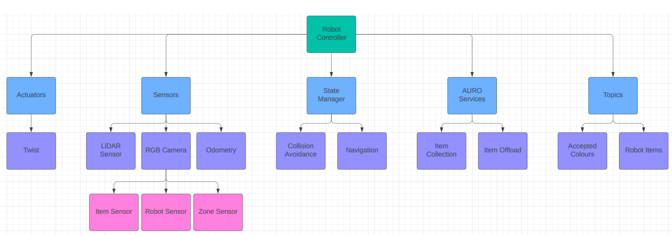


Fig. 1. Autonomous Robot System Architecture Diagram

The robot controller combines different components to provide successful completion of this project, through interactions between the robot and its environment.

The actuator uses the Twist class to directly control the robots movement based on the collected data and the robots current state, allowing for accurate navigation in its environment to complete objectives. The provided services are utilised allowing the robot to interact with it's environment, collecting and offloading items as required.

A finite state machine is used to control and implement the robots actions. Each state is designed to perform a specific task and upon completion transition into another state, that is based upon data from sensors and it's current state. All states have fallback methods to previous states allowing for continuous operation in cases of failure such as lost items or zones.

The sensors gather environmental data crucial for safe navigation and efficient completion of objectives. The LiDAR sensor detects the robots proximity to obstacles allowing for dynamic adjusts to the robots movement, avoiding collisions and maintaining accurate navigation towards the targeted object. The RGB camera gathers data about all items, zones, and robots, this provides essential data used to navigate through the environment towards the targeted object effectively as possible. The odometry sensor provides accurate localisation and position tracking for the robot to enable navigation.

To facilitate multi-robot collaboration, multiple topics are utilised to share information, this prevents any ineffective decision-making about item and zone selection resulting in a more efficient system. The 'accepted colours' topic stores which coloured item each zone accepts, preventing a robot from attempting to offload an item in the incorrect zone. The robot item topic ensures that robots are not attempting to collect the same items as one another reducing the likelihood of collisions between robots and improving overall efficiency.

III. Control

The RoboChart state machine describes the robots control system for transitioning between different states [10]. These transitions are triggered through object detection via the robots sensors and item interactions, ensuring a continuous flow between states with fallbacks to prevent the robot from becoming stuck in a single state. Each state performs actions to reach the states success criteria and complete the overall task [5] [8].

The initial *searching-without-item* state ensures the robot is in a safe position to rotate, then it rotates until an item is detected. Upon successful detection of an item, the robot transitions into *navigating-to-item* where it uses the cameras perception of the item to alter its movement to successfully navigate to the item. Whilst the robot is navigating, it performs collision avoidance methods to avoid other robots and obstacles. If the robot loses track of the item the state is reverted back to *searching-without-item* until an item is detected. Upon reaching the item the robot transitions into the *collecting-item* state where the robot attempts to collect the item, if successful it transitions into the *searching-with-item* state, or if unsuccessful it transitions back into *navigating-to-item* and reattempts the collection.

The *searching-with-item* state is similar to the initial state and performs the same actions however, this state is looking for a zone to offload the item. Having located a zone the robot transitions into the *navigating-to-zone* state which similar to the *navigating-to-item* state navigates towards the zone utilising the cameras perception and avoiding an obstacles or robots as necessary, until the robot detects that the zone is close enough to offload the item and transitions into the *depositing-item* state to offload the item. If unsuccessful the robot returns to the *navigating-to-zone* state and reattempts the offload, if successful the robot transitions into the *searching-without-item* state and the process restarts.

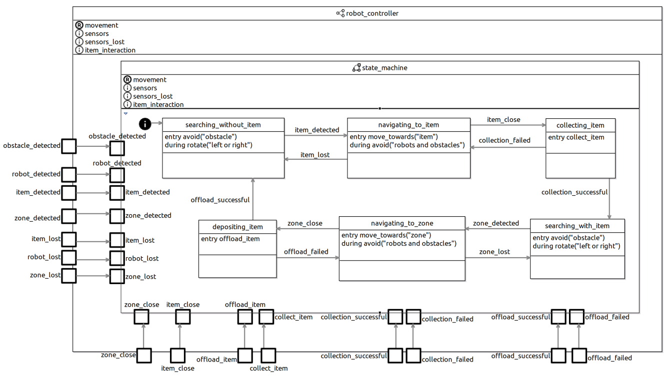


Fig. 2. RoboChart Control Diagram

IV. Evaluation

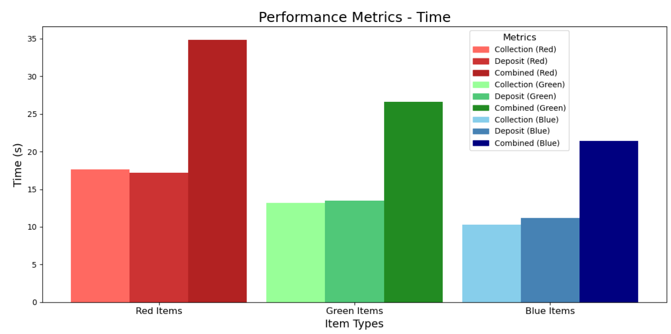
To evaluate how this system performed, simulations were used to provide visual data such as collision avoidance, navigational planning and item and zone selection. Alongside this, a complexity analysis was performed to evaluate the efficiency of the algorithm. Quantitative metrics were collected to further analyse performance gaining numerical insights into specific factors. This collection of data provides a wide range of information allowing a thorough evaluation of this systems performance across multiple areas.

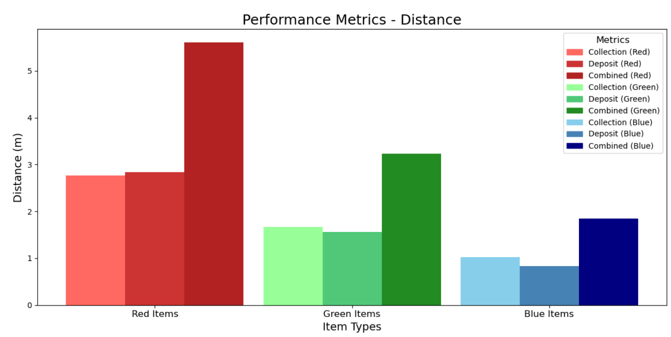
During simulations, the system consistently avoided any collisions whilst effectively achieving its objectives, demonstrating robust collision avoidance. This was achieved through LiDAR obstacle detection and dynamic navigational adjustments, real-time data was used to alter the robots velocity whilst continually navigating towards the target. This provided effective navigation without disruptions leading to efficient task completion.

Throughout simulations the robot gathered sensor data to select items and zones. All items in view were analysed and scored based on various factors including value, diameter, and distance for items, and size and distance for zones. This analysis happened throughout navigation allowing for new data to dynamically change the targeted object. The simulations demonstrated that the robot altered its path ensuring the highest scoring object was always selected. A problem with this system was that the robot could not target an object outside its field of view, resulting in a potential to navigate past items. A potential solution would be using multiple cameras to provide a 360° view of the environment which would not only improve item and zone selection but also improve safety as these cameras could be combined with LiDAR sensors to accurately map the environment.

The complexity analysis of this system determined that operations including navigation, collision avoidance, and objects searches had a time complexity of O(n). Searching for items and zones involved iterating through all detected objects to determine the best one for navigation, and navigation itself involved scoring each object resulting in an overall time complexity of O(n). The space complexity of this system is dependant on the stored object lists, scaling with the number of objects detected, and resulting in a space complexity of O(n). While the constant sized sensors such as LiDAR have a complexity of O(1), the worst case scenario relating to items and zones result in an overall space complexity of O(n).

The quantitative data collected during multiple simulations lasting approximately 10 minutes provided various metrics used to evaluate this system. The average time taken and distance travelled to collect and offload any item was 27.6 seconds and 3.3 meters with averages for collection being 13.1 seconds and 1.7 meters and for offloading being 13.5 seconds and 1.6 meters. However these values differed between the different coloured items. This difference is due to the increased distanced between the item spawn location and its corresponding zone.





The average time taken by all robots to travel one meter was 8.36 seconds, however robot1 had an average of 6.22 seconds, robot2 an average of 8.25 seconds and robot3 an average of 8.45 seconds. This suggests that robot1 encountered fewer obstacles or robots during simulations resulting in easier navigation with less potential collisions. The simulations themselves corroborate this hypothesis as robot1 operated in larger more open spaces than the other robots.

Once a robot collected an item, it then exclusively focused on items of that colour. This strategy distributed the robots across the environment in an attempt to reduce potential collisions between robots and the complex avoidance methods required to avoid each other. During simulations, robots rarely had to navigate around each other, significantly improving their overall navigational efficiency.

V. Safety and Ethics

The ethical and safety implications of a robotic system which interacts with an environment are significant and must be considered [1]. In terms of safety, ensuring the robot operates and interacts with the environment without creating additional risks to other robots and humans in real-world scenarios. The system utilises various sensors and algorithms to avoid collisions at all costs. However, in real-world scenarios additional systems would have to account for unexpected sensor failures or environmental changes. The robot must have emergency protocols to manage these unexpected changes such as emergency stop functionality or emergency homing. In real-world scenarios, the impact of a failure is much greater and could result in devastating impacts, so robust testing is required to ensure reliability to prevent failures [9].

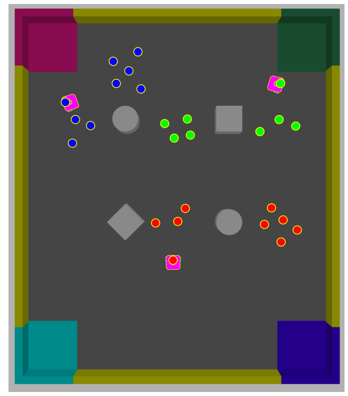
The ethical considerations for this system must consider the data privacy implications of using a camera and sensors. These devices could accidentally gather personal information which must be safeguarded in a real-world scenario, specifically if identifiable data is collected about specific locations. The robotic system must also operate without bias to ensure fair decisions are made, specifically in environments when involving sensitive or critical decisions.

This system utilises robust safety protocols to ensure safety measures are successful through the integration of multiple sensors and emergency features. The system avoids collecting personal data by only processing data relevant to the completion of the task, this also ensures the system is free from bias by reducing the effect of external factors on the data. The system would have to be tested in a variety of environments with a variety of data to ensure successful obstacle detection against different backgrounds and colours. Due the the dynamic nature of the real-world, sensor redundancy would be required alongside robust error handling algorithms to account for dynamic obstacles such as humans or animals [4].

VI. Simulation Scenario

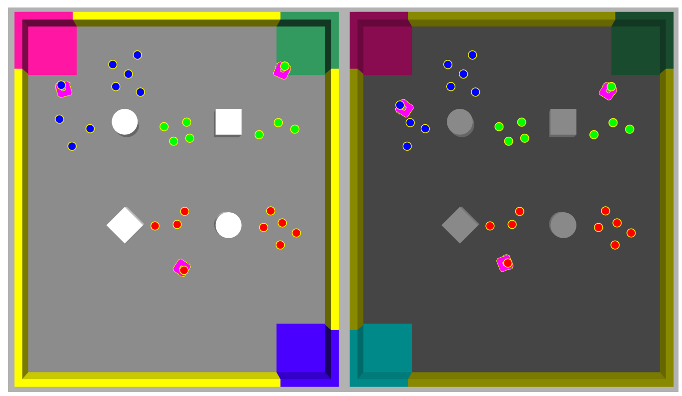
A. Scenario 1 - Standard Operating Conditions

This scenario evaluated all system aspects including collision avoidance, item and zone selection, navigational planning and multi-robot coordination. A overall focus on the system allowed a complete evaluation, aiming to achieve efficient item collection and offload. Creating a benchmark for comparisons between scenarios by ensuring all components functioned successfully in ideal conditions.



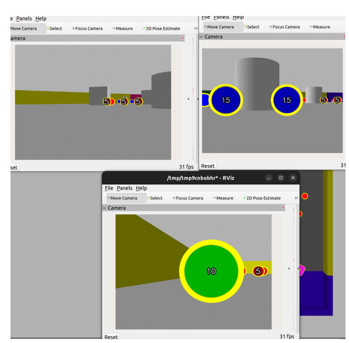
B. Scenario 2 - Reduced Zone Operation

This scenario focused on the zone selection processes impact on performance. Robots only detected active zones, preventing inefficiencies from incorrectly offloading items. Location a zone where items had previously been offloaded,preventing an item from being offloaded in multiple zones, stopping other items being offloaded at all. This evaluates the systems adaptability, robustness of decision-making algorithms, and multi-robot coordination.



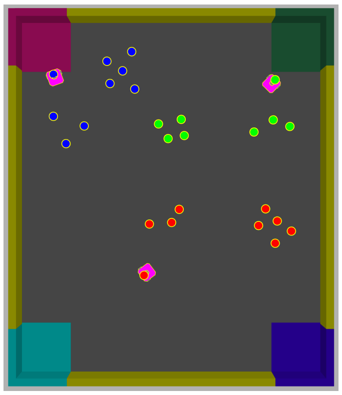
C. Scenario 3 - Sensor Noise Simulation

This scenario tests the robustness and reliability of the robot despite sensor noise. The robot uses constant sensor updates to track object locations, enabling navigation despite inaccuracies. Recovery methods allowed robots to find new objects and retry failed collections and offloads until completion. This reflects the real-world and the robots ability to function in less than ideal environments, testing error handling and failures due to poor sensor data.



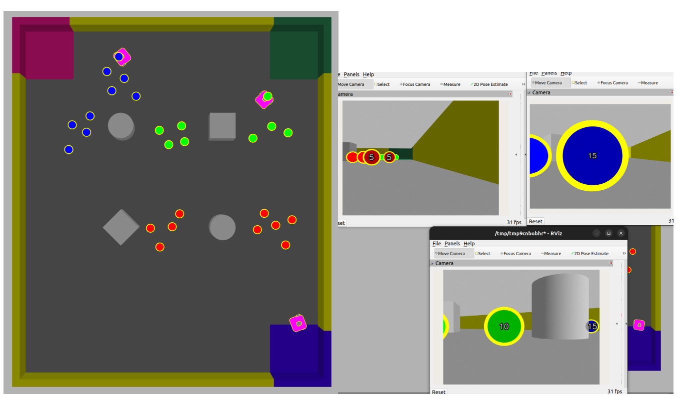
D. Scenario 4 - Obstacle Free Environment

This scenario maximised the navigational efficiency without the need for obstacle avoidance. Allowing robots to travel in direct paths maximising velocity and minimising deviation resulting in maximum task efficiency. The average total number of items collected increased from 66 to 72 demonstrating the time consumption to avoid collisions. This data evaluated the maximum performance of the core processes, providing information about how environmental conditions affect the systems performance.



E. Scenario 5 - Reduced Zone Operation and Sensor Noise Simulation

The strategy focuses on the robots performance in a non ideal environment with inactive zones and sensor noise. The recovery methods and constant sensor updates provides reliability in case of failures ensuring that tasks are still successful despite the challenges. This demonstrates the systems effectiveness in conditions similar to the real-world.



References

[1] http://www.facebook.com/ElectronicsWeekly, *Ecs 2019 – kista, sweden*, Electronics Weekly, Jul. 2019. [Online]. Available: https://www.electronicsweekly.com/news/research-news/seven-ethics-robotics-autonomous-systems-2019-07/.

[2] O. Media, *Robotics vision processing: Object detection and tracking*, Embedded Computing Design, Aug. 2019. [Online]. Available: https://embeddedcomputing.com/application/industrial/robotics-vision-processing-object-detection-and-tracking.

[3] S. Moe, K. Y. Pettersen, and J. T. Gravdahl, "Set-based collision avoidance applications to robotic systems," *Mechatronics*, vol. 69, p. 102 399, Aug. 2020. DOI: 10.1016/j.mechatronics.2020.102399. (visited on 01/10/2021).

[4] E. Papadimitriou, H. Farah, G. van de Kaa, F. Santoni de Sio, M. Hagenzieker, and P. van Gelder, "Towards common ethical and safe 'behaviour' standards for automated vehicles," *Accident Analysis Prevention*, vol. 174, p. 106 724, Sep. 2022. DOI: 10.1016/j.aap.2022.106724.

[5] Z. Ahmad, *State machines vs behavior trees: Designing a decision-making architecture for robotics*, www.polymathrobotics.com, Jan. 2023. [Online]. Available: https://www.polymathrobotics.com/blog/state-machines-vs-behavior-trees.

[6] A. Szczepaniak, *Leo rover blog - how is lidar used in robotic navigation? pros and cons*, www.leorover.tech, Jan. 2023. [Online]. Available: https://www.leorover.tech/post/how-is-lidar-used-in-robotic-navigation-pros-and-cons.

[7] C. Yang, *Vision-guided robotics - how cameras are transforming robotics*, TechNexion, Aug. 2023. [Online]. Available: https://www.technexion.com/resources/vision-guided-robotics-how-cameras-are-transforming-robotics/.

[8] *Finite state machine behavior tree in robotics*, Mingukwon.com, 2024. [Online]. Available: https://www.mingukwon.com/posts/finite-state-machine-&-behaviour-tree (visited on 01/01/2025).

[9] A. Singh, *Azorobotics*, AZoRobotics, Sep. 2024. [Online]. Available: https://www.azorobotics.com/Article.aspx?ArticleID=709.

[10] *State machines in robotics — restackio*, Restack.io, 2025. [Online]. Available: https://www.restack.io/p/state-machines-knowledge-state-machines-robotics-cat-ai (visited on 01/01/2025).

[11] *Collision avoidance system - an overview — sciencedirect topics*, www.sciencedirect.com. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/collision-avoidance-system.

[12] Y. Name, *Robotis e-manual*, ROBOTIS e-Manual. [Online]. Available: https://emanual.robotis.com/docs/en/platform/turtlebot3/features/.