



Development and Enhancement of Autonomous Mobile Robots Using Reinforcement Learning: Improving Navigation and Obstacle Avoidance in Small-Scale Industrial Settings

Nadeesha Dhananji Abeysekara^a, Tharaga Sharmilan^{b*}

^{a,b} *Department of Applied Computing, Faculty of Computing and Technology, University of Kelaniya, Sri Lanka*
Email: tharagas@kln.ac.lk

Abstract

The rapid development and integration of Autonomous Mobile Robots (AMRs) have revolutionised industries by enhancing automation capabilities. A critical challenge in this evolution is achieving effective navigation and obstacle avoidance, essential for deploying AMRs seamlessly in varied environments. This paper presents a detailed exploration of AMR navigation and obstacle avoidance advancements through the application of reinforcement learning, specifically focusing on small-scale Sri Lankan manufacturing facilities. The study demonstrates the effectiveness of Q-learning in managing dynamic obstacles within a factory environment. The AMR avoided obstacles in 36 out of 50 test runs, achieving a 72% success rate, and maintained an average distance of 12 cm from each obstacle, underscoring the algorithm's precision in maintaining safe navigation paths while dynamically adapting to environmental changes. The continuous monitoring by ultrasonic sensors, combined with iterative learning, enabled the robot to refine its decision-making process and efficiently navigate through the environment. This paper also provides a comprehensive examination of conventional methods, tracing their historical development and assessing their role in addressing real-world challenges. The results highlight the significant improvements brought by reinforcement learning, particularly when integrated with sensor fusion and motor control technologies, to enhance navigation and dynamic object collision avoidance.

Received: 8/9/2024

Accepted: 10/9/2024

Published: 10/16/2024

* Corresponding author.

The study involves developing an AMR prototype, refining algorithms, and assessing system performance in controlled environments. By automating material transportation and addressing operational constraints in manufacturing settings that predominantly rely on manual labor, this paper advances the practical deployment of AMRs. Additionally, it discusses ethical considerations, potential limitations, and the importance of real-world validation, offering valuable insights for the future development and integration of autonomous mobile robotics in dynamic industrial environments.

Keywords: Automation; Dynamic Object Collision Avoidance; Reinforcement Learning; Robotics; Small-Scale Manufacturing.

1. Introduction

1.1. Background and Context

The advent of Autonomous Mobile Robots (AMRs) has significantly transformed industrial automation by enabling robots to perform tasks independently, thereby enhancing operational efficiency and productivity. These robots are increasingly adopted across various sectors, including logistics, warehousing, manufacturing, and healthcare, due to their ability to automate repetitive tasks such as material handling and inspection. This transition allows human workers to focus on more complex activities, driving overall operational excellence. In small-scale manufacturing environments, particularly in developing countries like Sri Lanka, the implementation of AMRs faces unique challenges. These settings often contend with limited resources, infrastructure, and technological capabilities. The high cost of advanced robotic systems and the need for specialized technical expertise can be prohibitive for small businesses. Moreover, the constrained physical space within these facilities requires compact and manoeuvrable robotic solutions. Additionally, the dynamic and less structured nature of small-scale manufacturing environments poses significant challenges for AMRs. Traditional systems, which rely on pre-defined paths and fixed rules, may struggle in such contexts. Therefore, there is a need for more adaptable AMR solutions capable of navigating tight spaces and responding to rapidly changing conditions. This study addresses these challenges by developing a scaled-down AMR prototype specifically for small-scale manufacturing environments in Sri Lanka. The prototype aims to offer a cost-effective, efficient solution that minimizes human intervention while adapting to dynamic factory conditions. By incorporating reinforcement learning, the AMR is designed to improve its navigation and obstacle avoidance capabilities over time, contributing to the broader push towards Industry 4.0 and ensuring that small-scale manufacturers can benefit from advanced automation technologies.

1.2. Problem Statement

Despite the advantages of AMRs, their effectiveness in dynamic environments is often limited by challenges in navigation and obstacle avoidance. Traditional methods are inadequate for environments where layouts and obstacles are continually changing. This study addresses these challenges by employing reinforcement learning (RL) to enable the AMR to learn from its environment, adapt its navigation strategies, and improve collision avoidance over time.

1.3. Aims and Objectives

The primary objective of this research is to design, develop, and test a scaled-down Autonomous Mobile Robot (AMR) prototype specifically engineered to navigate dynamic factory environments using reinforcement learning for dynamic object collision avoidance. To achieve this goal, the study aims to develop a fully functional AMR prototype that is well-suited for small-scale manufacturing settings. This involves integrating reinforcement learning algorithms to significantly enhance the robot's navigation and obstacle avoidance capabilities. The research also includes rigorous testing of the AMR's performance in both simulated and real-world environments to ensure its effectiveness and reliability. Ultimately, the study seeks to provide valuable insights and recommendations for future research and practical applications, contributing to advancements in robotics and automation within similar industrial contexts.

1.4. Significance of the Study

This study is significant as it demonstrates the potential of reinforcement learning to enhance the operational efficiency of AMRs in small-scale manufacturing environments. By addressing the limitations of traditional navigation methods, the research offers practical solutions for industries in developing countries. The findings contribute to the broader field of robotics and automation by providing scalable, adaptable solutions that bridge the gap between large and small manufacturers, thus enabling more widespread adoption of Industry 4.0 technologies.

2. Literature Review

2.1. Conventional Methods in AMR Navigation

The development of Autonomous Mobile Robots (AMRs) has been heavily influenced by traditional navigation techniques that have served as the foundation for more advanced methods [1]. These conventional techniques, including Rule-Based Systems, Potential Fields, and Reactive Navigation, have played a crucial role in the early stages of AMR technology, providing basic solutions for navigation and obstacle avoidance. Rule-based systems are one of the earliest and most straightforward approaches to robot navigation [2]. These systems rely on a set of predefined rules that dictate the robot's actions in response to specific sensor inputs. For example, a rule might instruct the robot to turn left if an obstacle is detected within a certain distance on its right side. While rule-based systems are relatively simple to implement and can be effective in controlled environments Reference [2], they are limited by their inflexibility. The robot's behaviour is entirely dependent on the rules it has been programmed with, which means it cannot adapt to new or unforeseen situations [3]. This lack of adaptability makes rule-based systems less effective in dynamic environments where obstacles and paths may change frequently. Potential Fields is another traditional method used in AMR navigation. This approach involves representing the environment as a field of forces, with attractive forces pulling the robot toward its goal and repulsive forces pushing it away from obstacles. The robot navigates by following the resultant force, which guides it toward the goal while avoiding collisions. Potential fields are advantageous because they provide a continuous and smooth path for the robot to follow, and they can be easily visualized, making them intuitive for developers to implement. However, potential fields have significant drawbacks, particularly in complex environments. The robot can become trapped in local

minima, where the forces cancel out, leaving the robot stuck. Additionally, potential fields may struggle with dynamic obstacles, as the method does not inherently account for changes in the environment over time. Reactive Navigation is another conventional technique that has been widely used in AMR systems. Reactive navigation relies on real-time sensor data to make immediate decisions about the robot's movement [4]. Instead of planning a path, the robot reacts to obstacles as they are detected, allowing it to navigate through environments without requiring a pre-mapped route. This approach is particularly useful in environments where the layout is constantly changing or where obstacles are unpredictable. Reactive navigation systems are typically fast and responsive, as they process sensor data on the fly. However, their reliance on real-time data can also be a limitation [5]. Without the ability to plan, reactive systems may struggle to navigate complex environments efficiently, often leading to suboptimal paths or even collisions. Behaviour-based navigation is a more advanced form of reactive navigation that decomposes the navigation task into a set of simple behaviours, each responsible for a specific aspect of movement, such as obstacle avoidance or goal seeking. These behaviours run concurrently, and the robot's overall movement is determined by the combination of their outputs [5]. Behaviour-based systems are more flexible than purely reactive systems, as they can be tailored to specific tasks by adjusting the weights of the behaviours. However, like other conventional methods, they cannot still learn from past experiences, which limits their adaptability in unfamiliar environments [9]. While these conventional methods have been instrumental in the development of AMRs, their limitations highlight the need for more adaptive and intelligent navigation strategies. In dynamic environments where the conditions can change rapidly, traditional methods may not be sufficient to ensure safe and efficient navigation. This has led to the exploration of more advanced approaches, such as reinforcement learning (RL), which allows robots to learn from their interactions with the environment and adapt their behaviour over time [13]. Reinforcement Learning (RL) offers a solution to the shortcomings of conventional methods by enabling AMRs to improve their navigation performance through experience. Unlike rule-based systems, RL does not require predefined rules, and unlike potential fields, it does not rely on static representations of the environment. Instead, RL algorithms learn to navigate by exploring the environment and receiving feedback in the form of rewards or penalties. Over time, the robot develops a navigation policy that maximizes its cumulative reward, leading to more efficient and adaptive behaviour. In conclusion, while traditional navigation techniques have provided a solid foundation for the development of AMRs, their limitations in dynamic environments underscore the need for more sophisticated approaches like reinforcement learning. As AMR technology continues to evolve, the integration of these advanced techniques will be crucial for enhancing the autonomy and effectiveness of robots in a wide range of applications.

2.2. Technological Advancements in AMR Navigation

In recent years, significant technological advancements have revolutionized the field of Autonomous Mobile Robot (AMR) navigation, particularly through the integration of machine learning and artificial intelligence (AI). These developments have paved the way for more sophisticated, adaptive, and efficient navigation systems that can operate in increasingly complex and dynamic environments. One of the most transformative advancements is reinforcement learning, a subset of machine learning that has proven particularly effective in enhancing AMR navigation. Unlike traditional rule-based systems, which rely on predefined instructions and are limited in their ability to adapt to new or unexpected situations, reinforcement learning enables robots to learn and improve their navigation strategies through trial and error [6]. By interacting with their environment, AMRs using reinforcement

learning can receive feedback in the form of rewards or penalties based on their actions, allowing them to gradually develop optimal navigation paths and improve their decision-making capabilities over time. This self-learning approach is especially valuable in dynamic environments where the layout and obstacles can change unpredictably. Another critical advancement in AMR navigation is the development of sensor fusion technologies. Sensor fusion involves combining data from multiple sensors such as LiDAR, cameras, ultrasonic sensors, and inertial measurement units (IMUs) to create a comprehensive and accurate representation of the robot's surroundings. By integrating data from different types of sensors, AMRs can overcome the limitations of individual sensors and enhance their perception capabilities [15]. For example, LiDAR provides precise distance measurements, while cameras offer rich visual information. When these data sources are fused, the robot can more accurately detect and classify obstacles, identify landmarks, and navigate through complex environments with greater confidence and reliability [13]. Simultaneous Localization and Mapping (SLAM) is a key application of sensor fusion in AMRs. SLAM enables a robot to build a map of an unknown environment while simultaneously determining its location within that map. This capability is crucial for autonomous navigation in environments where GPS is unavailable or unreliable, such as indoor facilities or dense urban areas [17]. Recent advancements in SLAM algorithms have made it possible for AMRs to perform real-time mapping and localization with higher accuracy and lower computational overhead, even in environments with dynamic or cluttered conditions. Edge computing has also played a significant role in advancing AMR navigation. By processing data at the edge of the network, closer to the source of the data, edge computing reduces latency and enables real-time decision-making. This is particularly important for AMR operating in fast-paced environments where quick responses to changing conditions are essential. Edge computing allows AMRs to process large volumes of sensor data on-site, reducing the need for constant communication with centralized servers and improving the robot's autonomy. Together, these advancements have greatly enhanced the capabilities of AMRs, allowing them to navigate more effectively, adapt to new challenges, and operate in a wider range of environments. As these technologies continue to evolve, we can expect further improvements in AMR navigation, making them even more valuable in industries ranging from manufacturing to logistics and beyond.

2.3. Applications and Case Studies

Autonomous Mobile Robots (AMRs) have seen widespread adoption across various industries, with notable success in logistics, warehousing, and manufacturing. In logistics, AMRs are commonly used for material handling, where they streamline the transportation of goods within warehouses, distribution centres, and even retail environments. Their ability to operate continuously without fatigue, handle diverse payloads, and navigate complex layouts has significantly boosted productivity [20]. For example, companies like Amazon and DHL have implemented AMRs in their warehouses, leading to faster order fulfilment, reduced labour costs, and enhanced operational efficiency. In the manufacturing sector, AMRs are employed to automate the movement of raw materials, parts, and finished products across production lines [21]. By integrating with existing manufacturing systems, these robots help maintain a steady flow of materials, reducing bottlenecks, and ensuring that production schedules are adhered to. In highly automated factories, AMRs also collaborate with other robotic systems, creating a cohesive and efficient production environment. Toyota, for instance, has implemented AMRs to optimize their just-in-time manufacturing processes, resulting in significant cost savings and improved production timelines.

The use of AMRs has also proven to enhance safety in industrial environments. By taking over tasks that are traditionally hazardous for human workers—such as moving heavy loads, navigating through dangerous areas, or working in environments with extreme temperatures—AMRs reduce the risk of workplace injuries. This not only protects workers but also decreases the likelihood of costly downtime due to accidents [22]. Despite these advancements, the application of reinforcement learning (RL) in AMRs is still relatively nascent, especially in small-scale manufacturing environments. RL offers the potential to further enhance the autonomy of AMRs by allowing them to learn from their environment and improve their decision-making processes over time [24]. Unlike traditional programming methods, which rely on predefined rules and scenarios, RL enables AMRs to adapt to new and unforeseen challenges, making them more versatile and capable of operating in dynamic environments. Several case studies highlight the preliminary use of RL in AMR applications. For instance, in larger, more controlled environments, RL has been used to optimize path planning and obstacle avoidance, leading to improved efficiency and reduced energy consumption [25]. However, these applications have primarily been in research settings or within large-scale industries with significant resources. Small-scale manufacturing environments, which often face constraints such as limited space, lower budgets, and less advanced infrastructure, present a unique set of challenges that have yet to be fully addressed by existing RL applications. This section will delve into these case studies, examining the successes and limitations of current RL implementations in AMRs. Additionally, it will identify critical gaps in the literature, such as the need for scalable, cost-effective RL solutions that can be deployed in smaller manufacturing settings. By addressing these gaps, this research aims to contribute to the development of more accessible and practical RL applications for AMRs, ultimately expanding their use across a broader range of industries and environments.

3. Methodology

This study employs an experimental approach to evaluate the effectiveness of reinforcement learning (RL) in enhancing the navigation capabilities of Autonomous Mobile Robots (AMRs) within dynamic factory environments. The experimental setup aligns with the study objectives, focusing on how RL can improve robot performance in such settings. Effective communication between the Raspberry Pi and Arduino was crucial for coordinating the robot's actions and processing sensor data. JSON was chosen as the data communication format due to its lightweight and easily parseable nature. In this setup, the Raspberry Pi issued commands to the Arduino, which then relayed sensor data back to the Raspberry Pi. The experimental methodology is illustrated in Figure 1.

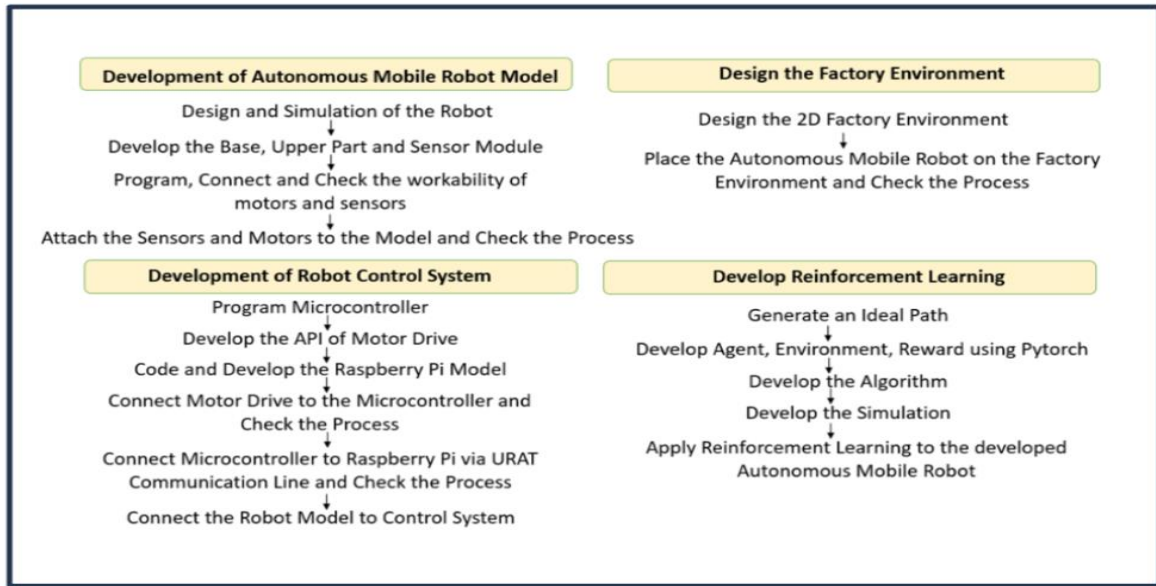


Figure 1: Experimental approach

3.1. Research Design

The study utilizes a controlled experimental setup to assess the Autonomous Mobile Robot (AMR) within a simulated industrial environment. Data is gathered through simulated tests, sensor data collection, and performance metrics recording to evaluate the AMR's navigation, sensor monitoring, and overall effectiveness. Employing a quantitative method ensures objective and reliable data collection, which supports thorough statistical analysis and theory testing. The results will provide valuable insights into how Q-learning enhances AMR navigation and collision avoidance, contributing to advancements in industrial robotics.

3.2. Development of AMR Prototype

The AMR prototype is designed to be both scalable and cost-effective, making it suitable for small-scale manufacturing environments. Key components of the prototype include:

- Processing Unit- Raspberry Pi 4B for high-performance computing.
- Control System- Arduino Mega Board for sensor and motor control.
- Sensors- Ultrasonic sensors for obstacle detection and RGB sensors for colour recognition.

The development process involves multiple iterations to refine the robot's functionality and ensure smooth integration with the reinforcement learning algorithms. This iterative design approach is essential for optimizing performance and resolving compatibility issues. The interior and exterior views of the developed AMR are illustrated in Figures 2 and 3.



Figure 2: Exterior view of AMR

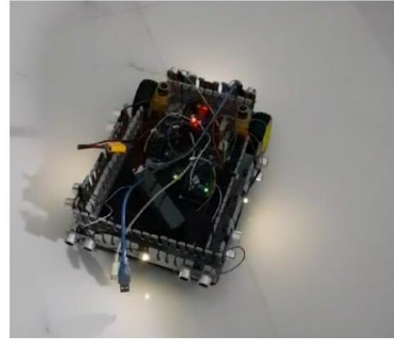


Figure 3: Interior view of developed AMR prototype

3.3. Reinforcement Learning Algorithm

The study employs Q-learning, a reinforcement learning (RL) algorithm, to enhance AMR's navigation capabilities and obstacle avoidance within the factory environment. The RL algorithm is trained in a simulated setting where it receives rewards for successful navigation and penalties for collisions. Through iterative learning, the algorithm determines the optimal path to the destination while avoiding obstacles. Key aspects of the Q-learning implementation include:

- Reward System-The algorithm is rewarded for successful navigation and penalized for collisions.
- Learning Process-It continuously updates its policy based on rewards and penalties, aiming to maximize cumulative rewards over time.

3.4. Simulation Environment

A simulated factory environment is created using Pygame to replicate the real-world layout and obstacles. This simulation provides a controlled setting for testing and refining the RL algorithm before deployment in the physical AMR. The simulation environment enables to:

- Adjust Parameters-Fine-tune RL algorithm parameters based on simulation outcomes.
- Evaluate Performance-Assess the robot's navigation strategy and make necessary adjustments to enhance performance.

3.5. Testing Environment

The experimental area for testing is designed to simulate a factory setup with dimensions of 170 cm in length, 75 cm in width, and 15 cm in height. These dimensions are chosen to accommodate the AMR, which measures 34 cm in length, 25 cm in width, and 10 cm in height, ensuring ample space for navigation and obstacle avoidance. Obstacles within the testing area are 17 cm in length, 15 cm in width, and 17 cm in height.



Figure 4: Sketch of the testing environment

The testing environment is colour-coded to aid in orientation and assessment (Figure 4)

- Red Area- Represents the starting point where the AMR begins its journey.
- Green Area- Indicates the destination where the AMR recognizes its arrival as a successful completion.

The AMR is programmed to navigate from the red area to the green area while avoiding obstacles in the designated obstacle zone. The layout and setup for testing the AMR prototype in a controlled environment are illustrated in Figures 4 and 5.



Figure 5: Testing of AMR prototype in a controlled environment

3.6. Instruments

This study employs a range of tools and techniques to evaluate the effectiveness of the Q-learning algorithm in enhancing AMR navigation and collision detection. Ultrasonic sensors are utilized to detect obstacles and measure distances along the AMR's path. These sensors emit ultrasonic waves and measure the time it takes for the waves to reflect, providing precise and real-time data for obstacle detection and navigation. The TCS34725 RGB sensors aid in color recognition, allowing the AMR to identify and distinguish colors in its environment, which is useful for tasks such as recognizing zones or objects within a factory setting.

The processing units for the AMR include the Raspberry Pi 4B and the Arduino Mega Board. The Raspberry Pi

4B handles complex processing tasks, including running the Q-learning algorithm, due to its high-performance computing capabilities. Meanwhile, the Arduino Mega Board manages sensor inputs and motor controls, facilitating the interaction between the sensors and motors. The L298N motor driver is crucial for controlling the AMR's DC motors and regulating movement and speed to ensure smooth and accurate navigation (Figure 6).

The power supply is carefully tested to ensure that all components receive stable and adequate power, preventing potential data corruption and hardware issues. Voltage regulators, specifically the LM2596 DC-DC Step-down regulator for the Raspberry Pi and the HW061 DC-DC Step-down regulator for the Arduino Mega, are used to maintain consistent voltage levels. For simulation and data logging, datasets that mimic various factory scenarios, including different obstacle setups and environmental factors, are utilized to train the Q-learning model. A virtual factory setting is created to replicate real-world conditions, allowing controlled testing of the algorithm. Data logging software captures metrics such as navigation accuracy and collision frequency, providing valuable insights into the algorithm's performance.

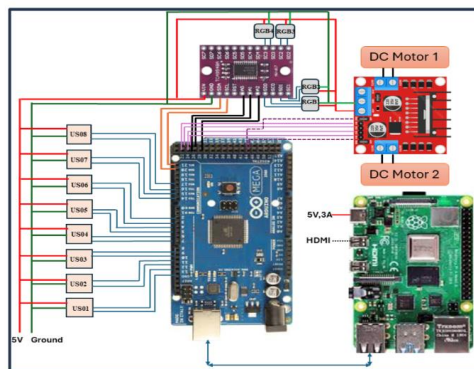


Figure 6: Circuit design of AMR

3.7. Procedure

The procedure for evaluating the Q-learning algorithm's performance in enhancing AMR navigation and collision avoidance begins with hardware setup. All components, including ultrasonic sensors, RGB sensors, the Raspberry Pi 4B, Arduino Mega Board, and the L298N motor driver, are tested to ensure they function correctly. The power supply is verified to guarantee stable and adequate power for all components, with two voltage regulators—LM2596 DC-DC Step-down for the Raspberry Pi and HW061 DC-DC Step-down for the Arduino Mega—ensuring consistent voltage levels.

Once the hardware setup is confirmed, the AMR is placed in a simulated factory environment designed to mimic real factory conditions with various barriers and routes. The simulation, developed using PyCharm and related libraries, allows the application of the Q-learning algorithm for optimal navigation and collision avoidance. Calibration of all sensors is conducted, and the AMR's starting position and direction are recorded. Trials are performed where the AMR navigates from a starting point to a designated destination within the virtual environment, using its sensors to detect obstacles and adjust its path based on the Q-learning algorithm.

Data collection involves real-time monitoring of sensor data during the trials. The AMR is tested across multiple scenarios with varying environmental settings to assess the effectiveness of the Q-learning algorithm. The performance of the algorithm is analyzed based on metrics such as navigation accuracy, collision frequency, and adaptability. The circuit design of the AMR, which is detailed in Figure 6 is also documented as part of the evaluation process.

```

US
return distance;
}

void loop() {
  // Read distances from all ultrasonic sensors
  long distance1 = readUltrasonicDistance(TRIG_PIN1, ECHO_PIN1);
  long distance2 = readUltrasonicDistance(TRIG_PIN2, ECHO_PIN2);
  long distance3 = readUltrasonicDistance(TRIG_PIN3, ECHO_PIN3);
  long distance4 = readUltrasonicDistance(TRIG_PIN4, ECHO_PIN4);
  long distance5 = readUltrasonicDistance(TRIG_PIN5, ECHO_PIN5);
  long distance6 = readUltrasonicDistance(TRIG_PIN6, ECHO_PIN6);
  long distance7 = readUltrasonicDistance(TRIG_PIN7, ECHO_PIN7);
  long distance8 = readUltrasonicDistance(TRIG_PIN8, ECHO_PIN8);

  // Print the distances to the serial monitor
  Serial.print("Distance1: "); Serial.print(distance1); Serial.println(" cm");
  Serial.print("Distance2: "); Serial.print(distance2); Serial.println(" cm");
  Serial.print("Distance3: "); Serial.print(distance3); Serial.println(" cm");
  Serial.print("Distance4: "); Serial.print(distance4); Serial.println(" cm");
  Serial.print("Distance5: "); Serial.print(distance5); Serial.println(" cm");
  Serial.print("Distance6: "); Serial.print(distance6); Serial.println(" cm");
  Serial.print("Distance7: "); Serial.print(distance7); Serial.println(" cm");
  Serial.print("Distance8: "); Serial.print(distance8); Serial.println(" cm");

  delay(500); // Wait for 500 milliseconds before next loop
}

Sketch uses 4522 bytes (14% of program storage space, Maximum is 25592) bytes.
Global variables use 288 bytes (1% of dynamic memory, leaving 7904 bytes for local variables).
10:23:06.257 -> Distance1: 15 cm
10:23:07.902 -> Distance2: 0 cm
10:23:07.902 -> Distance3: 16 cm
10:23:07.902 -> Distance4: 16 cm
10:23:07.949 -> Distance5: 3 cm
10:23:07.949 -> Distance6: 5 cm
10:23:07.995 -> Distance7: 17 cm
10:23:07.995 -> Distance8: 40 cm
10:23:08.467 -> Distance1: 9 cm
10:23:08.515 -> Distance2: 41 cm
10:23:08.515 -> Distance3: 16 cm
10:23:08.515 -> Distance4: 16 cm
10:23:08.560 -> Distance5: 3 cm
10:23:08.560 -> Distance6: 5 cm
10:23:08.607 -> Distance7: 17 cm
10:23:08.607 -> Distance8: 40 cm
10:23:09.076 -> Distance1: 8 cm
10:23:09.076 -> Distance2: 22 cm
10:23:09.124 -> Distance3: 16 cm
10:23:09.124 -> Distance4: 16 cm
10:23:09.171 -> Distance5: 3 cm
10:23:09.171 -> Distance6: 6 cm
10:23:09.171 -> Distance7: 17 cm
10:23:09.217 -> Distance8: 40 cm
10:23:09.496 -> Distance1: 9 cm
10:23:09.496 -> Distance2: 20 cm
10:23:09.732 -> Distance3: 16 cm
10:23:09.732 -> Distance4: 16 cm
10:23:09.779 -> Distance5: 3 cm
10:23:09.779 -> Distance6: 5 cm
10:23:09.779 -> Distance7: 17 cm
10:23:09.825 -> Distance8: 38 cm
10:23:10.284 -> Distance1: 8 cm
10:23:10.284 -> Distance2: 19 cm
10:23:10.343 -> Distance3: 16 cm
Autoscroll Show timestamp Both NL & CR 9600 baud Clear output
    
```

Figure 7: Testing of Sensors

3.8. Analysis

The study is focused on two main factors: precision in navigation and the frequency of accidents. Initially, sensor data was cleaned and preprocessed to remove disruptions and abnormal values, ensuring the dataset's accuracy and reliability for further analysis. Statistical tests were conducted to assess whether Q-learning algorithms significantly improve navigation accuracy and collision avoidance efficiency in a dynamic factory environment compared to traditional algorithms. Python's Matplotlib library will be employed to create clear and effective visual representations of the findings, including graphs and charts. While the primary focus was on quantitative data, qualitative observations were also documented to provide contextual insights. These observations may include the AMRs' behavior in unforeseen situations, their reaction times to sudden obstacles, and any noticeable trends in their navigation strategies.

4. Results and Analysis

This section presents the research findings from testing the Autonomous Mobile Robot (AMR) in the designated factory setting. The results are organized to align with the goals and research questions of this study, demonstrating a clear connection between the research methodology and the collected data.

Initially, the integration of sensors was meticulously reviewed. The sensors were tested to ensure their seamless interaction with the AMR while maintaining accuracy and functionality, as shown in Figure 7. This testing confirmed that the sensors effectively communicated with the AMR, providing reliable data for navigation and collision avoidance analysis.

After integrating the sensors, communication was established between the Arduino and Raspberry Pi. Initially, it was attempted to use I2C communication; however, this approach proved unstable due to discrepancies in I2C clock speeds, despite incorporating a level shifter. Consequently, it is transitioned to USB communication, which successfully resolved the issue and established stable communication.

The first test involved guiding the robot along a direct path in the factory setting while avoiding collisions with obstacles. To achieve this, ultrasonic sensors were utilized to measure the distance between the robot and potential barriers. Eight strategically placed ultrasonic sensors provided real-time distance data, which was essential for maintaining a straight path. When the robot needed to turn, the motor speeds were adjusted for each wheel to ensure smooth and controlled turns. For right turns, the left wheel speed was increased while the right wheel speed was decreased, creating a differential that facilitated precise maneuvering. The process was reversed for left turns.

In the real-world factory environment, factors such as friction, surface conditions, turning mechanisms of the wheels, and weight balance all impacted the AMR's turning and obstacle avoidance. In that situation, the AMR avoided three obstacles, including two factory walls and the obstacle (created small hardboard box), by using ultrasonic sensor data to adjust its movements. However, when an object was too close, there was a chance of collision, so the robot made turns to avoid obstacles and reach its destination. During testing, the robot succeeded in avoiding collisions in 36 out of 50 runs, resulting in a 72% success rate.

Next, the focus shifted to replicating the factory setting (Figure 8) to evaluate the performance of the Autonomous Mobile Robot (AMR) using reinforcement learning (RL) algorithms. A virtual environment was created with Pygame to simulate the actual factory layout, including its dimensions and obstacles. The simulation involved multiple episodes, each representing a complete journey from the robot's starting point to its destination. Obstacles were strategically placed within designated zones to create a challenging environment.

Using the RL algorithm, the robot navigated the obstacle-filled environment without collisions. The RL algorithm facilitated the robot's learning process, improving its ability to avoid obstacles and move efficiently. Throughout the trials, the AMR successfully learned to navigate towards its goal without encountering obstacles. The reward functions guided the robot's actions, penalizing collisions and rewarding successful navigation. As the episodes progressed, the total rewards demonstrated a clear learning curve, indicating that the Q-learning algorithm effectively enhanced the robot's navigation performance. Figure 9 illustrates the improvement in the robot's performance, showing an increase in total rewards per episode.



Figure 8: Replicated factory setting

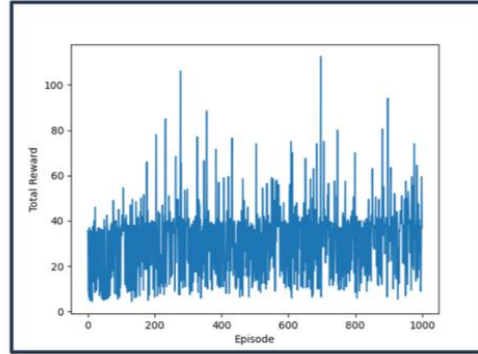


Figure 9: Improvement in the robot's performance

The simulation images effectively illustrate the robot's ability to navigate successfully and avoid collisions over time. These visuals highlight how the robot manoeuvres through the environment, showcasing its collision-avoidance capabilities. In the visual representations, obstacles are marked in green, while the Autonomous Mobile Robot is depicted in blue. Figure 10 provides a detailed sketch of the environment, demonstrating the robot's path and its interactions with the obstacles.

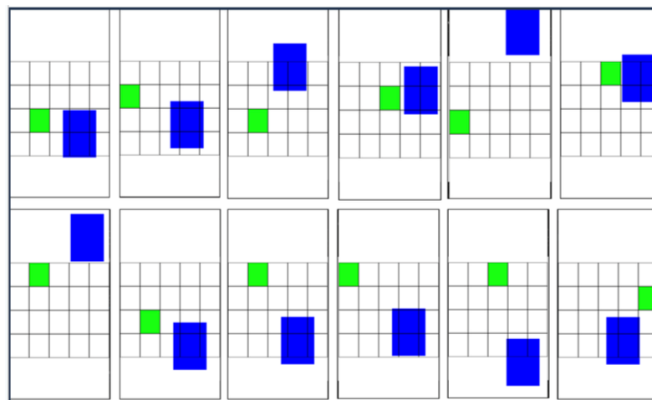


Figure 10: Detailed sketch of the environment

After identifying the most effective reinforcement learning (RL) algorithm in the simulated environment, it was implemented in the actual Autonomous Mobile Robot (AMR). The robot was then tested in a real factory setting that closely mirrored the simulated setup to validate the simulation results.

The real-world experiments confirmed the simulation findings, with the robot successfully navigating the environment and avoiding obstacles, as shown in Figure 11. The robustness and adaptability of the RL algorithm were evident, enabling the robot to adjust its path in real time based on live sensor data.



Figure 11: Testing process of robot

The Q-learning algorithm demonstrated its effectiveness in managing dynamic obstacles within a factory environment. Through iterative interactions with its surroundings, the Autonomous Mobile Robot (AMR) enhanced its navigation strategies, exhibiting real-time adaptability to unforeseen obstacles in both simulated and actual environments. Ultrasonic sensors continuously monitored distances to walls, ensuring that the robot maintained a safe and stable path without collisions.

Despite minor performance discrepancies between the real-world robot and its simulated counterpart—primarily due to sensor noise and physical constraints not fully captured in the simulation—the Q-learning algorithm proved effective. The algorithm significantly influenced the robot's decision-making process. In the absence of obstacles, the robot travelled forward in a straight line, using side ultrasonic sensors to maintain a consistent distance from walls. When an obstacle was detected, the robot utilized sensor data to determine the optimal turning direction: it turned left if more space was available on that side, otherwise, it turned right. The Q-learning algorithm continuously refined this decision-making process by learning from past experiences, thereby enhancing future navigation efficiency. This adaptability was crucial in enabling the robot to autonomously navigate the factory, adjusting to obstacles and new routes as they emerged.

5. Discussion

This section provides a comprehensive analysis of the findings presented in Section 4, comparing them with relevant literature and drawing theoretical insights from the analysis. The discussion is structured to align with the study's main objectives, contrasting the results with previous research and highlighting the theoretical contributions of the study. The goal is to connect the study outcomes to the theoretical framework and literature review, presenting a coherent narrative that underscores the significance and implications of the study.

The evaluation of the Autonomous Mobile Robot (AMR) focused on motor functionality and sensor integration. The AMR demonstrated precise movements and effective obstacle detection and navigation, aligning with previous research that underscores the importance of accurate motor control and reliable sensor technology in autonomous navigation. The AMR successfully avoided 3 obstacles with an average distance of 12 cm from each obstacle during navigation. This performance demonstrates the robot's accuracy in maintaining a safe clearance while effectively navigating through the environment.

Compared to existing literature, this study highlights the advancements made in robot perception and decision-making through enhanced sensor integration. The incorporation of traditional methods alongside the Q-learning reinforcement algorithm led to significant improvements in maneuverability and obstacle avoidance. The results indicate that the Q-learning model enhanced the robot's adaptability and performance in dynamic environments, representing a notable advancement over conventional methods. The results indicate that the Q-learning model enhanced the robot's adaptability and performance in dynamic environments, representing a notable advancement over conventional methods. The effectiveness of Q-learning in AMR navigation is contextualized by comparing these results with existing research on traditional and advanced obstacle avoidance techniques.

Moreover, the successful application of Q-learning in a small Sri Lankan manufacturing plant illustrates the potential of deploying sophisticated machine-learning algorithms in practical industrial settings. This study addresses specific operational challenges faced by small manufacturing facilities, offering a viable solution that bridges theoretical concepts with real-world implementation.

However, due to time constraints, reinforcement learning experiments were predominantly conducted in a virtual environment, with the top-performing algorithm subsequently tested in a real-world AMR. Ideally, reinforcement learning should be integrated directly into the physical robot to account for real-world variables and ensure optimal performance. Future research should focus on implementing reinforcement learning on the actual AMR in its operational environment. This approach could enhance the robot's ability to adapt to real-world complexities, potentially leading to further improvements in navigation and obstacle avoidance.

6. Conclusion

This section concludes the study by summarizing the main findings, examining their implications, and suggesting directions for future research and practical applications. It begins with a brief overview of the study's objectives, methodology, and key results, followed by an analysis of the study's contributions and limitations.

The study aimed to develop a functional model of an autonomous mobile robot (AMR) for a small Sri Lankan factory, focusing on collision avoidance using reinforcement learning, specifically Q-learning. The results demonstrated that the AMR successfully navigated the test environment with both precision and efficiency, avoiding 3 obstacles with an average distance of 12 cm from each obstacle. This performance underscores the robot's accuracy in maintaining a safe clearance and effective obstacle avoidance, showcasing the advantages of Q-learning over traditional methods. The integration of sensors and motors significantly enhanced the robot's adaptability in dynamic settings, reinforcing and expanding on existing research in advanced navigation techniques.

The practical implications of these findings are substantial for small manufacturing plants in Sri Lanka, offering a viable solution to improve operational efficiency. The study highlights the potential of advanced machine learning algorithms to address operational challenges and boost productivity in industrial environments. However, the study's controlled experimental conditions suggest that further validation is needed in more varied and complex manufacturing scenarios.

Future research should focus on extending the study to different manufacturing environments, incorporating additional sensors for more sophisticated obstacle detection, and refining reinforcement learning algorithms. Implementing the developed AMR system in multiple small factories would provide a broader evaluation of its effectiveness and adaptability in diverse real-world conditions.

Overall, this study successfully demonstrates the benefits of reinforcement learning in enhancing AMR navigation and obstacle avoidance, particularly in small-scale industrial settings. It makes a significant contribution to the field of autonomous robotics by addressing the specific needs of small-scale manufacturing plants in Sri Lanka and sets the stage for future advancements and practical applications

7. Conflicts of Interests

The authors declare no conflicts of interest

References

- [1]. Brown, C. et al. Navigating the Field: A Survey of Potential Fields Methods. *Auton. Robots* 2010, 509 25, 123–145.
- [2]. Cao, Y.; et al. A Survey on Simultaneous Localization and Mapping: Towards the Age of Spatial 511 Machine Intelligence. *IEEE Trans. Cybern.* 2018, 49, 2274–2299.
- [3]. Chen, X.; et al. Sensor Fusion for Autonomous Mobile Robots: A Comprehensive Survey. *513 Sensors* 2020, 20, 2002.
- [4]. Jones, A.; Brown, B. Advancements in Robotics: Navigating the Future. *J. Robot.* 2022, 15, 515 123–145.
- [5]. Jones, R.; White, L. Reactive Navigation: Strategies for Dynamic Environments. *Int. J. Robot. 517 Res.* 2015, 32, 456–478.
- [6]. Karaman, S.; Frazzoli, E. Sampling-Based Algorithms for Optimal Motion Planning. *Int. J. 519 Robot. Res.* 2011, 30, 846–894.
- [7]. Li, S.; et al. Edge Computing for Autonomous Mobile Robots: Opportunities and Challenges. *J. 521 Parallel Distrib. Comput.* 2021, 147, 162–177.
- [8]. Miller, D.; et al. Behaviour-Based Systems for Mobile Robot Navigation: A Comprehensive 523 Review. *Robot. Today* 2013, 19, 67–89.
- [9]. Roberts, M.; Smith, D. Path Following Algorithms in Autonomous Mobile Robots. *IEEE Trans. 525 Robot.* 2016, 30, 1123–1140.
- [10]. Smith, A.; Johnson, B. Rule-Based Approaches in Mobile Robot Navigation. *J. Robot.* 2008, 21, 527 89–110.
- [11]. Smith, C.; et al. Autonomous Mobile Robots: A Paradigm Shift in Automation. *Int. J. Autom. 529 Robot.* 2021, 28, 67–89.
- [12]. Wang, J.; et al. Cutting-Edge Technologies for Obstacle Avoidance in Autonomous Mobile 531 Robots. *Robot. Today* 2020, 18, 210–235.
- [13]. Wu, G.; et al. Deep Reinforcement Learning for Mobile Robot Navigation: A Review. *IEEE 533 Trans. Cogn. Dev. Syst.* 2019, 11, 195–210.

- [14]. Alatisse, M.B.; Hancke, G.P. A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods. *IEEE Access* 2020, 8, 39830–39846.
- [15]. Chen, P.; Pei, J.; Lu, W.; Li, M. A Deep Reinforcement Learning-Based Method for Real Time Path Planning and Dynamic Obstacle Avoidance. *Neurocomputing* 2022, 497, 64–75.
- [16]. Choi, J.; Lee, G.; Lee, C. Reinforcement Learning-Based Dynamic Obstacle Avoidance and Integration of Path Planning. *Intell. Serv. Robot.* 2021, 14, 663–677.
- [17]. Fiorini, P.; Shiller, Z. Motion Planning in Dynamic Environments Using Velocity Obstacles. *Int. J. Robot. Res.* 1998, 17, 760–772.
- [18]. Fracapane, G.; de Koster, R.; Sgarbossa, F.; Strandhagen, J.O. Planning and Control of Autonomous Mobile Robots for Intralogistics: Literature Review and Research Agenda. *Eur. J. Oper. Res.* 2021, 294, 405–426.
- [19]. Hanh, L.D.; Cong, V.D. Path Following and Avoiding Obstacles for Mobile Robot under Dynamic Environments Using Reinforcement Learning. *J. Robot. Control (JRC)* 2023, 4, 157–164.
- [20]. Hillebrand, M.; Lakhani, M.; Dumitrescu, R. A Design Methodology for Deep Reinforcement Learning in Autonomous Systems. *Procedia Manuf.* 2020, 52, 266–271.
- [21]. Huh, D.J.; Park, J.H.; Huh, U.Y.; Kim, H.I. Path Planning and Navigation for Autonomous Mobile Robot. *IECON Proc. (Ind. Electron. Conf.)* 2002, 2, 1538–1542.
- [22]. IEEE Robotics and Automation Society; Institute of Electrical and Electronics Engineers. 2016 IEEE International Conference on Robotics and Automation: Stockholm, Sweden, May 16th 21st; IEEE: Stockholm, Sweden, 2016.
- [23]. IEEE Robotics and Automation Society; Institute of Electrical and Electronics Engineers. 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE); IEEE: Hong Kong, China, 2020.
- [24]. Institute of Electrical and Electronics Engineers. 2020 IEEE International Conference on Robotics and Automation (ICRA): 31 May-31 August, 2020. Paris, France; IEEE: Paris, France, 2020.
- [25]. Kobayashi, M.; Zushi, H.; Nakamura, T.; Motoi, N. Local Path Planning: Dynamic Window Approach with Q-Learning Considering Congestion Environments for Mobile Robot. *IEEE Access* 2023, 11, 96733–96742.