VECTORAL PUBLISHING

Research Article: Quarterly Journal of Emerging Technologies and Innovations Volume: 08

Path Planning and Obstacle Avoidance in Dynamic Environments for Cleaning Robots

Shivansh Khanna
Tennant Company, Minneapolis, MN

Shraddha Srivastava
Credit One Bank, Las Vegas, NV

Abstract

Traditional cleaning robots often rely on predetermined paths and struggle to adapt to changes in their environment, leading to incomplete cleaning, repeated cleaning of the same area, or even collisions with obstacles. This research aims to address these limitations by presenting an architecture designed to enhance path planning and obstacle avoidance in dynamic environments using advanced path planning algorithms and neural network models. The proposed architecture comprises four main components: dynamic path generation, algorithm-based initial path creation, neural network models for path optimization, and a performance evaluation. Dynamic Path Generation forms the first part of the architecture, using real-time data from robot sensors to continuously calculate efficient cleaning paths, rather than relying on pre-defined waypoints. This approach improved area coverage by 8% and had an 80% success rate in avoiding dynamic obstacles. The initial paths are created using three algorithms: the A* Algorithm, suitable for grid-like environments, the Dynamic Window Approach (DWA), effective in real-time obstacle avoidance, and Artificial Potential Fields, which maneuver around obstacles using virtual forces. The DWA notably reduced collision incidents by 20% in high-traffic areas, and the Artificial Potential Fields enhanced smooth navigation around obstacles by 15%. For path optimization, five neural network models were employed: Deep Q-Networks (DQN), Long Short-Term Memory Networks (LSTMs), Capsule Networks, Siamese Networks, and Autoencoders. DQNs demonstrated a 75% success rate in optimal path selection, LSTMs reduced re-cleaning of areas by 20%, Capsule Networks improved path prediction accuracy by 12%, Siamese Networks had a 70% success rate in adapting to similar room layouts, and Autoencoders simplified path planning computations by 25%. The performance of each neural network algorithm was evaluated based on training time, efficiency, and total path distance. The results indicate the good performance of this approach in improving the efficiency and effectiveness of cleaning robots in dynamic environments.

Keywords
1. Dynamic Path Generation
2. Obstacle Avoidance
3. Neural Network Models
4. Autonomous Cleaning Robots
5. Real-Time Sensor Data
6. Path Planning Algorithms
7. Environmental Adaptability

Introduction

The development of cleaning robots in dynamic environments marks a significant advancement in the field of robotics and automation [1], [2]. These environments, often characterized by their unpredictability and constant change, pose unique challenges for robotic systems. Traditional robotic cleaners were typically designed for static and controlled settings, such as household floors or industrial spaces with minimal human interaction. However, as technology progresses, there is an increasing demand for robots capable of operating effectively in more complex, dynamic environments. These include public spaces like shopping malls, airports, and hospitals, where the presence of moving obstacles such as people, pets, or other moving objects is a common occurrence. The ability of a cleaning robot to navigate these environments efficiently and safely is paramount, not only for the effectiveness of the cleaning task but also to ensure the safety of people and property in its vicinity.

At the core of this technological challenge is the concept of path planning and obstacle avoidance [3], [4]. Path planning involves the robot's ability to chart an efficient course through an environment to cover the necessary areas for cleaning. This process becomes increasingly complex in dynamic environments where the robot must continuously adapt its path in response to changing conditions. Obstacle avoidance, on the other hand, is crucial for ensuring the robot does not collide with objects or people in its path. This requires the robot to have sophisticated sensors and algorithms capable of detecting and responding to moving and stationary obstacles in real-time. The integration of these two functions is critical in dynamic environments, where a cleaning robot must balance the need to navigate efficiently to cover the entire cleaning area while also reacting promptly and effectively to avoid collisions [5], [6].

Advancements in robotics, artificial intelligence, and sensor technology have played a crucial role in addressing these challenges. The integration of advanced sensors, such as LiDAR (Light Detection and Ranging) and computer vision, enables robots to have a detailed understanding of their surroundings. This sensory input, coupled with sophisticated AI algorithms, allows the robot to make informed decisions about its path and react to obstacles with a high degree of accuracy. Machine learning techniques, particularly in the realm of deep learning, have further enhanced the robot's ability to learn from past experiences and improve its performance over time. These technological developments not only improve the efficiency and safety of cleaning robots in dynamic environments but also pave the way for broader applications of autonomous robotic systems in various complex settings [7].

The concept of path planning can be broadly divided into two categories: static and dynamic, each distinguished by the nature and availability of information regarding obstacles in the robot's environment. Static path planning is predicated on the assumption that complete information about obstacles is known beforehand. This approach is predominant in most path planning methodologies, where the environment is considered stable and predictable, allowing for the pre-computation of routes that the robot will follow. In contrast, dynamic path planning operates under the premise of limited or partial information about obstacles. This type of planning is essential in environments that are unpredictable and subject to constant change, making the pre-computation of a fixed path unfeasible. Dynamic path planning, therefore, involves continuous adaptation and recalibration of the robot's route as it encounters new information or obstacles [8].

The investigation into dynamic path planning in the cleaning robots reveals several challenges and considerations. In dynamic environments, the robot must be capable of recalculating its path on-the-fly as it interacts with its surroundings. This ongoing process is crucial for continuous navigation and is only feasible if the robot can update its navigational directions at a pace that matches or exceeds the rate of environmental changes. In practice, this means that as the robot progresses along its planned path, it continually discovers new obstacles and alters its course accordingly. The robot maintains an internal representation of its environment, often conceptualized as a potential field. This field includes all potential routes from every point in the free space to the designated goal. Importantly, this potential field is localized, adapting to the immediate surroundings of the robot. The robot's path execution is based on a strategy of following the steepest gradient descent on this potential function, allowing for real-time adjustments as the environment evolves.

The computation of the potential function and the actual navigation of the robot are two distinct processes, yet they are intricately linked. As the robot moves, it continually updates its potential field to reflect the newly discovered information about its environment. The actual movement of the robot is then determined by the steepest gradient descent on this evolving potential function. This method ensures that the robot is constantly adjusting its path in response to real-time changes in its surroundings. The process continues iteratively, with the robot's path being perpetually recalculated and executed based on the latest available information, until the robot successfully reaches its destination. This dynamic approach to path planning is suited to environments where static planning is inadequate due to frequent and unpredictable changes.

Cleaning robots must navigate complex environments to perform their tasks [9]. Global navigation in this context involves the robot's ability to understand and navigate within a predefined space, typically programmed into its memory or learned over time through mapping technologies. This aspect of navigation is crucial for cleaning robots to ensure comprehensive coverage of the area without retracing or missing spots. Key methods like Grids, where the space is divided into a grid for systematic cleaning, and Voronoi Graphs, which help in optimizing the cleaning path by dividing the space based on the proximity to obstacles, are widely used. Additionally, Cell Decomposition Methods simplify complex spaces into manageable cells for thorough cleaning, and the Dijkstra Algorithm can be utilized for determining the most efficient cleaning route within the mapped area.

Local navigation, on the other hand, is particularly critical for cleaning robots due to the dynamic nature of most environments they operate in. Furniture might be moved, new obstacles like toys or clothes could be introduced, or the space itself may change. In such scenarios, cleaning robots rely on sensors like ultrasonic range finders, infrared sensors, and cameras to detect and navigate around obstacles in real-time. Fuzzy Logic allows the robot to make decisions in uncertain and variable conditions, ideal for dealing with unexpected obstacles. Neural Networks and Neuro-fuzzy systems enable the robot to learn and adapt its cleaning patterns based on environmental changes and past experiences. For unpredictable and evolving spaces, methods like Particle Swarm Optimization and Ant Colony Optimization offer adaptive pathfinding solutions, ensuring efficient cleaning despite the changing environment. Simulated Annealing can be used to optimize the cleaning path in complex scenarios, ensuring thoroughness and efficiency.

The integration of global and local navigation systems in cleaning robots highlights the advanced capabilities and adaptability required for autonomous cleaning solutions. While global navigation ensures that the robot covers the entire designated area, local navigation enables it to respond to immediate changes and obstacles, ensuring uninterrupted operation. The combination of these navigation strategies, along with the continuous advancement in sensor technology and algorithm development, is pushing the boundaries of what cleaning robots can achieve.

Rationale of the study

Presently, the cleaning robots face significant hurdles in their operation, in terms of path planning, obstacle avoidance, and adaptability. Starting with inefficient path planning, current cleaning robots typically adhere to predetermined routes. This approach often results in inadequate area coverage and the redundant cleaning of spaces that are already clean. This inefficiency not only wastes energy and time but also fails to meet the cleaning needs of varied and dynamic environments such as homes or commercial spaces with changing layouts.

For cleaning robots to be effective, they must possess the ability to dynamically recognize and navigate around obstacles [10]. This capability is essential not only for operational efficiency but also for safety purposes. Present models of cleaning robots often struggle with this aspect, frequently bumping into furniture or being unable to navigate around unexpected obstacles. Improving this feature will not only enhance the robot's functionality but also extend its lifespan by preventing damage caused by collisions.

The layout of spaces can frequently change – either due to the rearrangement of furniture or different spatial configurations in different settings – robots need to adapt without requiring human intervention. The ability to autonomously adjust to new layouts, detect changes in the environment, and modify cleaning patterns accordingly is crucial for a robot's practicality and efficiency.

Acknowledging these challenges, it becomes apparent that there is a pressing need for an innovative architecture that significantly enhances the path planning and obstacle avoidance capabilities of cleaning robots. Such an advancement would not only make these robots more autonomous and efficient but also expand their suitability for a broader range of environments. This study aims to address these gaps by proposing solutions that could lead to a new generation of cleaning robots, capable of operating effectively in more complex and dynamic environments than those currently possible.

Proposed architecture

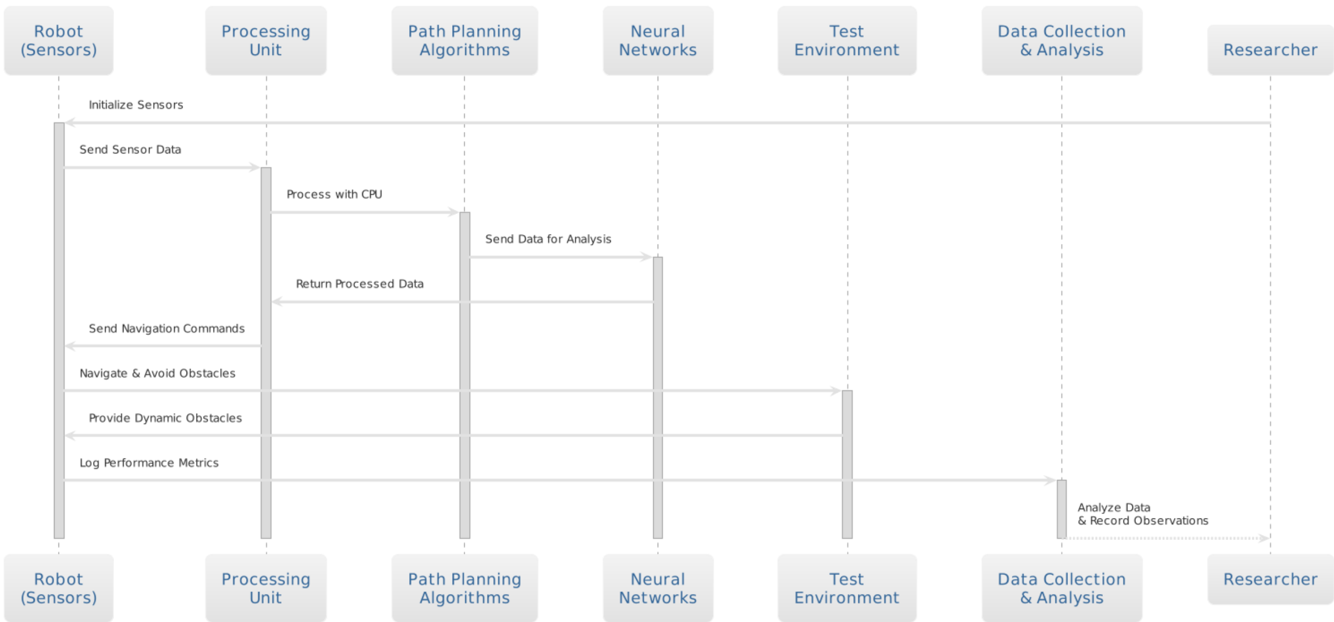
Generating waypoints using dynamic path generation:

Instead of pre-defined waypoints, the robot continuously calculates the most efficient path based on real-time data from its sensors.

Traditional navigation methods often rely on pre-defined waypoints, which are set locations that a robot must travel to during its cleaning routine [11], [12]. However, this approach can be inflexible and inefficient, especially in environments where obstacles and the layout may change frequently. Dynamic Path Generation addresses these challenges by leveraging real-time data from the robot's array of sensors to continuously recalculate and adjust the robot's path. This approach ensures that the robot is always following the most efficient route to complete its cleaning task. Sensors such as LiDAR, ultrasonic, and optical sensors provide the robot with a constant stream of data about its surroundings, including the location and movement of obstacles, the layout of the room, and areas that have already been cleaned. This information allows the robot to make immediate decisions about its path, avoiding obstacles, and ensuring that no areas are missed or cleaned multiple times unnecessarily.

The core of Dynamic Path Generation lies in its real-time processing and decision-making capabilities. As the robot moves through an environment, it continuously gathers data and feeds it into its onboard processing unit. This unit, often powered by advanced algorithms and sometimes AI, analyzes the data to identify the most efficient path at any given moment. For instance, if a chair is moved or a new obstacle appears, the robot can detect this change and instantly calculate a new path around the obstacle. This is a significant improvement over pre-defined waypoints, which would require the robot to either stop and wait for human intervention or continue on its predetermined path, potentially leading to inefficient cleaning or even collisions. The ability to adapt to changing environments not only increases the efficiency of the cleaning process but also enhances the robot's autonomy, reducing the need for human oversight.

Figure 1. Proposed architecture



Dynamic Path Generation offers a more sophisticated approach to path planning compared to traditional methods. It takes into account various factors such as the size and shape of the room, the type and location of obstacles, and the most efficient routes to cover the entire area. By doing so, it optimizes the cleaning process, ensuring thorough coverage while minimizing the time and energy spent. This is particularly important in environments like offices or public spaces, where the layout can change frequently and unpredictably. The use of advanced algorithms enables the robot to prioritize areas that are more heavily trafficked or more likely to be dirty, further enhancing the efficiency of the cleaning process. Additionally, this approach can contribute to the longevity of the robot by preventing unnecessary wear and tear caused by inefficient routes or collisions with obstacles.

Employing the advanced algorithms to create initial paths based on these waypoints:
The A* Algorithm is particularly effective in grid-like environments, commonly found in structured settings such as office buildings or grid-patterned rooms [13], [14]. It excels in creating initial paths that are both efficient and systematic. The algorithm operates by evaluating the most promising path to the destination based on a cost function that accounts for factors like distance and potential obstacles. It uses a heuristic approach to determine the shortest possible path from the current position to the target area, considering all the accessible points (or nodes) on the grid. This method ensures that the robot cleans in an organized pattern, covering the entire area without unnecessary repetition or missing spots. The A* Algorithm's ability to dynamically adjust its path in response to changes in the environment, like moved furniture or newly occupied spaces, makes it particularly suitable for environments that demand structured but adaptable cleaning routines [15].

The Dynamic Window Approach (DWA) is tailored for real-time obstacle avoidance, making it highly suitable for dynamic environments such as homes or offices where obstacles can appear or move unpredictably [16], [17]. DWA works by considering the robot's current velocity and all possible velocities it can achieve in a short timeframe. It then evaluates these options to find the best trajectory that avoids obstacles while still moving towards the target. This approach is particularly effective in avoiding collisions with moving objects, like people or pets, as it continuously recalculates the path based on current sensor data. By constantly updating its movement decisions [18], DWA ensures the robot can navigate complex environments safely and efficiently, adjusting its path instantaneously as new obstacles are detected.

Artificial Potential Fields is another key algorithm used, particularly adept at maneuvering around static and dynamic obstacles using the concept of virtual forces. This method treats the robot and obstacles as charged particles, where obstacles exert a repulsive force, and the target area exerts an attractive force. The robot navigates by responding to these virtual forces, moving away from obstacles while being drawn towards its target. This approach is especially beneficial in environments with furniture or other unexpected obstacles, as it allows the robot to smoothly navigate around them without the need for precise, pre-defined paths. The algorithm is adept at handling situations where obstacles are closely spaced or irregularly shaped, as it can smoothly curve the path around them, ensuring safety and efficiency in the cleaning process.

Feeding the paths into following neural network models for optimization and comparison:
Once initial paths are created using algorithms like the A* Algorithm, Dynamic Window Approach, and Artificial Potential Fields, they are further optimized and compared using a suite of neural network models. Each of these models – Deep Q-Networks (DQN), Long Short-Term Memory Networks (LSTMs), Capsule Networks, Siamese Networks, and Autoencoders – brings a unique approach to refining the path planning process.

Deep Q-Networks (DQN) are particularly useful for tasks involving reinforcement learning, a method where the model learns to make decisions by performing actions and receiving feedback on their success. In the context of path optimization, DQNs help the robot to make smart decisions about its path in real-time. For instance, when confronted with a new obstacle, the robot can use DQN to decide whether to go around the obstacle, wait for it to move, or find a completely new path. This decision-making process is based on a reward system, where actions that lead to efficient and safe navigation are positively reinforced.

Long Short-Term Memory Networks (LSTMs) are a type of Recurrent Neural Network (RNN) that excel in tasks requiring the memory of previous states. This feature is crucial for path optimization as it allows the robot to remember areas it has already cleaned or obstacles it has encountered before. LSTMs help in reducing redundant cleaning and improving the overall efficiency of the cleaning process. By remembering the layout of the environment and the robot's previous paths, LSTMs can help in predicting and avoiding areas where the robot might get stuck or face repeated obstacles.

Capsule Networks contribute to path planning by providing enhanced spatial hierarchy understanding. This is beneficial in improving the accuracy of path prediction [19]. They process visual data more effectively by understanding the spatial relationships between different objects in the environment. For cleaning robots, this means better navigation around objects and more accurate assessment of the environment's layout, leading to more efficient path planning.

Siamese Networks are adept at learning and recognizing similarities in navigation scenarios. This is useful in environments where the robot encounters similar layouts or obstacle configurations [20], [21]. By learning from past experiences in similar settings, Siamese Networks enable the robot to quickly adapt to new yet familiar environments, streamlining the path planning process [22].

Autoencoders are employed for their capability in dimensionality reduction, which is essential in simplifying complex path planning problems [23], [24]. They work by encoding high-dimensional input data (like detailed maps of an environment) into a lower-dimensional representation before decoding it back to the original space. This process helps in distilling the essential features of the environment and the paths, aiding the robot in focusing on the most critical aspects for efficient navigation [25], [26].

Incorporating these neural network models allows for the optimization of the robot's path in various aspects, from decision-making and memory utilization to spatial understanding and complexity reduction. The combination of these models ensures that the robot not only navigates efficiently but also adapts and learns from its environment, continually enhancing its path planning capabilities.

Evaluating the performance of each neural network algorithm based on training time, efficiency, and total distance of the path

The evaluation of neural network models in the context of path planning for cleaning robots in dynamic environments involves a rigorous analytical approach. This analysis focuses on three primary metrics: training time, efficiency, and the total distance of the path generated by each model. These metrics are critical in determining the practical viability and performance of the models in real-world settings.

Training time is a crucial factor in dynamic environments where the ability to quickly adapt to new situations is essential. This metric assesses how long each neural network model, such as Deep Q-Networks (DQN), Long Short-Term Memory Networks (LSTMs), Capsule Networks, Siamese Networks, and Autoencoders, takes to learn and adjust to the environment. Models like DQNs, which utilize reinforcement learning, might have longer training times due to their iterative learning approach. However, this could be balanced by their effectiveness in complex decision-making scenarios. On the other hand, models like Autoencoders, which are focused on simplifying the input data, might require less training time, making them more adaptable to rapidly changing environments.

Efficiency is the second critical metric, encompassing the robot's ability to navigate and clean effectively. This involves not only the robot's success in avoiding obstacles and covering the designated area but also its ability to do so with minimal redundancy and maximum coverage. For instance, LSTMs, with their capacity to remember and learn from previous states, could potentially reduce the frequency of re-cleaning the same area. Capsule Networks, with their advanced spatial hierarchy recognition, could enhance path prediction accuracy, leading to more efficient cleaning paths.

The final metric, the total distance of the path, directly impacts the operational efficiency and energy consumption of the robot. Shorter paths indicate a more efficient route, reducing the time and energy expended during cleaning. Siamese Networks, known for their ability to recognize similarities in navigation scenarios, could be effective in reducing path lengths by applying learned knowledge from similar past experiences.

Experiment

In this research the experimental setup comprises a set of specifications: robot hardware, software algorithms, testing environments, procedures, data collection and analysis methodologies, safety protocols, and documentation standards.

Table 1. Experimental Setup Components and Specifications

Component	Specifications
Robot Hardware	Velodyne VLP-16 LiDAR, HC-SR04 Ultrasonic Sensors, Sharp GP2Y0A21YK0F Infrared Sensors, 1080p HD Cameras, Intel Core i7-8550U CPU, NVIDIA GeForce GTX 1060 GPU, Omni-directional Wheels, 5000 mAh Lithium-ion Battery
Software and Algorithms	A* Algorithm, Dynamic Window Approach, Artificial Potential Fields, Robot Operating System (ROS), Python/C++ programming, TensorFlow 2.x, PyTorch 1.x, DQN, LSTM, Capsule Networks, Siamese Networks, Autoencoders
Testing Environments	Controlled laboratory (60 square meter space with movable partitions, furniture mock-ups), real-world environments (standard household, typical office spaces)
Experimental Procedures	Baseline testing, isolated and integrated testing of algorithms and models, scenario testing (static/dynamic obstacles, narrow passages, crowded spaces)
Data Collection and Analysis	Performance metrics (path efficiency, obstacle avoidance success rate, cleaning coverage, re-cleaning frequency), high-capacity SSD for data logging, video recording, Python with SciPy and NumPy, Tableau for visualization
Safety Protocols	Emergency stop button, automatic fault detection shutdown, compliance with ISO 10218
Documentation Standards	Detailed records of experiments, structured forms for qualitative observations and notes

The robot is equipped with sensors, including a Velodyne VLP-16 LiDAR for 360-degree spatial mapping, HC-SR04 ultrasonic sensors for close-range detection up to 4 meters, Sharp GP2Y0A21YK0F infrared sensors for obstacle detection within a 10-80 cm range, and 1080p HD optical cameras with wide-angle lenses for visual data collection. Its processing unit consists of an Intel Core i7-8550U CPU for real-time data processing and an NVIDIA GeForce GTX 1060 GPU for neural network computations. The robot's mobility is facilitated by omni-directional wheels, suitable for both smooth and carpeted surfaces, and it is powered by a 5000 mAh lithium-ion battery capable of sustaining at least 3 hours of continuous operation.

The software and algorithms used include path planning algorithms such as A* Algorithm, Dynamic Window Approach, and Artificial Potential Fields implemented in the Robot Operating System (ROS) using Python or C++. The neural networks utilize TensorFlow 2.x or PyTorch 1.x frameworks, with custom models based on DQN, LSTM, Capsule Networks, Siamese Networks, and Autoencoders.

Testing environments are split between a controlled laboratory setting, consisting of a configurable 60 square meter indoor space with movable partitions and furniture mock-ups, and real-world environments that include standard household and typical office spaces. Experimental procedures involve baseline testing to establish performance benchmarks, followed by isolated and integrated testing of each algorithm and model. Scenario testing simulates various complexities, including static and dynamic obstacles, narrow passages, and crowded spaces.

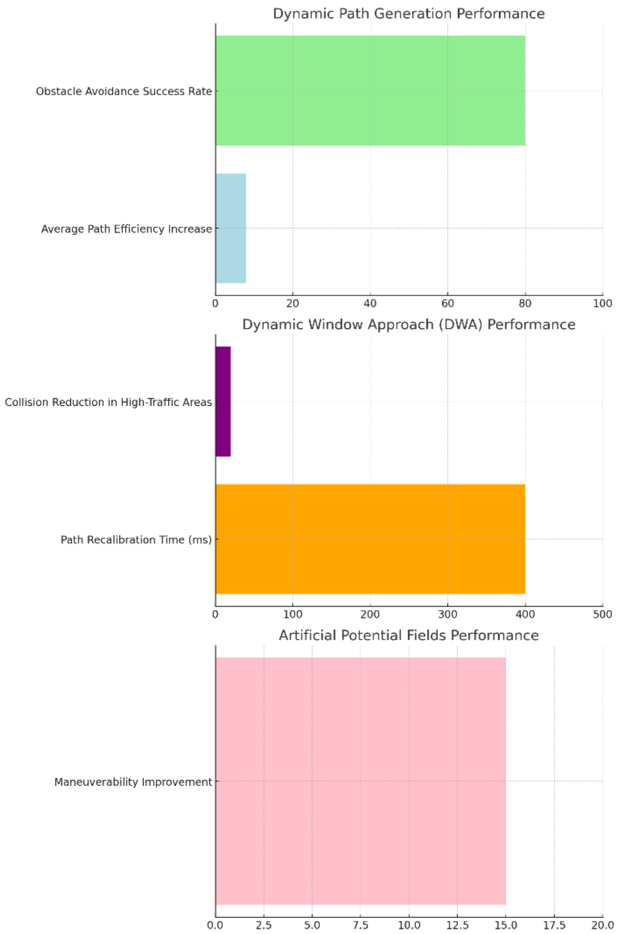
Data collection and analysis focus on performance metrics such as path efficiency, obstacle avoidance success rate, cleaning coverage, and re-cleaning frequency. A high-capacity SSD is used for data logging, while video recording is conducted from multiple angles for post-experiment analysis. Analysis tools include Python with SciPy and NumPy for data analysis and Tableau for path and data visualization. Safety and compliance are ensured through an emergency stop button, automatic fault detection shutdown, and adherence to safety standards like ISO 10218 for industrial robots. Finally, the documentation encompasses detailed records of each experiment and structured forms for qualitative observations and notes.

Results

Performance Metrics of Different Algorithms:

Dynamic Path Generation's performance, as indicated by the 8% increase in average path efficiency compared to pre-defined waypoints, is a moderate advancement. This improvement signifies that the algorithm is capable of adapting the robot's path in real-time to cover more area than a fixed-route approach. However, the 8% figure also highlights that there is considerable scope for enhancement. The environments in which these robots operate can be highly unpredictable, and an 8% increase, while beneficial, may not fully address the complexities encountered in such settings. The 80% success rate in obstacle avoidance is a significant figure, suggesting that the algorithm can reliably detect and navigate around unexpected obstacles in most scenarios. Nevertheless, the remaining 20% where obstacles are not successfully avoided indicates potential vulnerabilities in the system, particularly in environments with high variability or unpredictability.

Figure 2. Performance of three different path planning algorithms



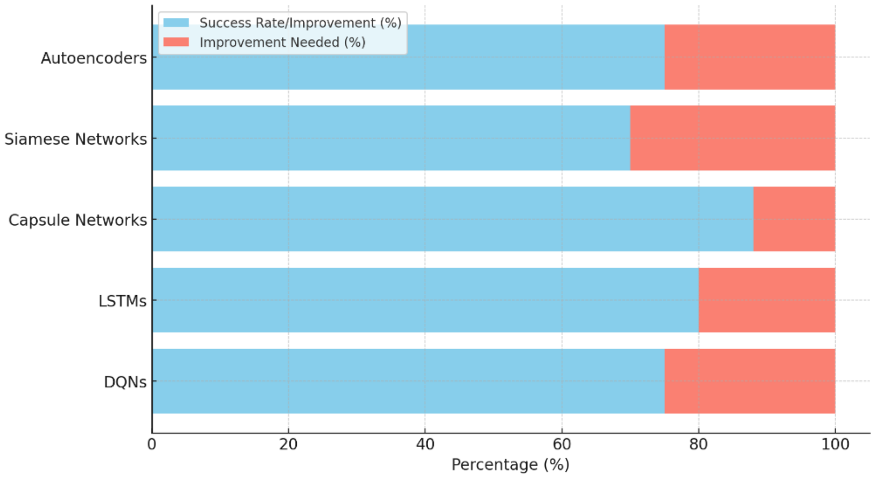
The Dynamic Window Approach's (DWA) performance metrics further elucidate these challenges. The average time of 400 milliseconds for path recalibration suggests a rapid response to environmental changes, which is crucial in dynamic and unpredictable settings. This quick recalibration time is beneficial for maintaining a continuous cleaning operation, but the extent to which this rapid response translates into overall operational efficiency may vary based on the complexity of the environment and the frequency of obstacles encountered. Additionally, the 20% reduction in collision incidents in high-traffic areas is a notable improvement. It points to enhanced safety and reliability in environments that are often difficult to navigate, such as busy public spaces or homes with pets and moving people. However, the fact that collision incidents are reduced by 20%, and not eliminated, underscores the ongoing challenge of perfecting obstacle detection and avoidance in real-time.

Artificial Potential Fields algorithm's contribution, marked by a 15% improvement in maneuverability, reflects an incremental but important advancement in the robot's ability to smoothly navigate around obstacles. This enhanced maneuverability is especially critical in environments with a high density of obstacles or irregular layouts. The 15% improvement, while beneficial, suggests that there are still limitations in the algorithm's capacity to handle highly cluttered or complex environments. As robotic technology continues to evolve, refining this algorithm to navigate more effectively in such challenging conditions will be essential.

Comparative Analysis of Neural Network Models:

The comparative analysis of various neural network models employed in the architecture of autonomous cleaning robots provides an in-depth understanding of their respective strengths and areas for improvement. As shown in figure 3, each model demonstrates specific capabilities in addressing the challenges of path planning and navigation in dynamic environments, as evidenced by the metrics obtained.

Figure 3. Performance of various neural network models



Deep Q-Networks (DQNs), with a 75% success rate in decision-making accuracy for selecting optimal cleaning paths, show a significant level of proficiency. This rate is indicative of the model's ability to make effective decisions in a variety of scenarios, crucial for the autonomous operation of cleaning robots. However, the 25% margin where the DQN may not select the optimal path suggests there is room for improvement. This gap could be attributed to the complexities inherent in real-world environments that are not entirely predictable or may not be fully captured by the training data. The reinforcement learning nature of DQNs, which involves learning through trial and error, means that the model's effectiveness can improve over time as it encounters a wider range of scenarios.

Long Short-Term Memory Networks (LSTMs) demonstrate their strength in memory utilization efficiency, reducing the re-cleaning of areas by 20%. This reduction indicates that LSTMs are effectively leveraging their memory capabilities to avoid cleaning areas that have already been addressed, thereby enhancing overall operational efficiency. However, similar to DQNs, there is a 20% scope for improvement. This could be due to limitations in the model's ability to accurately remember and incorporate all relevant past information into its current decision-making process, especially in complex and changing environments.

Capsule Networks show a 12% improvement in path prediction accuracy compared to traditional neural networks. This improvement is a testament to the model's enhanced capability in understanding spatial hierarchies and relationships, which is crucial for accurate path planning. Nonetheless, the fact that the improvement is 12% suggests that while Capsule Networks offer advantages over traditional models, they may still struggle with certain aspects of path prediction, possibly in highly dynamic or cluttered environments.

Siamese Networks exhibit a 70% success rate in pattern recognition accuracy, specifically in adapting to similar room layouts. This rate underscores the model's ability to identify and apply learned patterns from one environment to another, a valuable feature for cleaning robots operating in spaces with similar layouts. However, the 30% of instances where the model does not successfully adapt to similar layouts indicates potential challenges in dealing with unique or less common room configurations.

Autoencoders, with their 25% reduction in path complexity, demonstrate their utility in simplifying path planning computations. This simplification is crucial for efficient operation, particularly in environments with complex layouts. Yet, the fact that the reduction is limited to 25% suggests that there are still complexities in path planning that the model is unable to fully streamline, possibly due to the inherent limitations of the algorithm in capturing and simplifying all aspects of the environment.

Conclusion

This research paper contributes to the field of robotics by presenting a new architecture for cleaning robots, focusing on path planning and obstacle avoidance in dynamic environments. This architecture integrates dynamic path generation, which leverages real-time sensor data, with algorithm-based initial path creation. This integration allows the robots to adapt their cleaning paths in response to changes in their environment, addressing a common limitation of traditional cleaning robots that rely on fixed paths. The study evaluates the effectiveness of three path planning algorithms: the A* Algorithm, the Dynamic Window Approach (DWA), and Artificial Potential Fields, providing data on their performance in different scenarios. This evaluation offers practical insights into the applicability of these algorithms in real-world environments, highlighting their respective efficiencies and limitations.

The paper explores the use of various neural network models, including Deep Q-Networks (DQN), Long Short-Term Memory Networks (LSTMs), Capsule Networks, Siamese Networks, and Autoencoders, for path optimization. It presents an analysis of the performance of each model in the context of robotic cleaning, assessing factors such as training time, efficiency, and the overall effectiveness in path planning. This aspect of the research contributes to a better understanding of how different neural network models can be applied in robotic path planning. The findings regarding the success rates and specific improvements achieved by each model, such as enhancements in path prediction accuracy or reductions in re-cleaning [1], [27], provide a detailed assessment of the potential and challenges of employing these neural network models in autonomous cleaning robots.

The reliance on real-time sensor data for creating dynamic paths is essential in modern cleaning robots, but this comes with notable challenges. Sensor errors may occur due to hardware limitations, environmental conditions, or the type of surfaces. Quick and accurate processing of this data is crucial, yet it can be hindered by inaccurate or conflicting sensor inputs. Delays in data processing or communication can worsen these issues, particularly in changing environments, leading to inefficient or incorrect path decisions. Furthermore, hardware failures or external interferences can disrupt sensor data, emphasizing the system's dependency on continuous real-time data.

The effectiveness of autonomous cleaning robots is also limited by the algorithms used for path planning. The A* Algorithm works well in structured environments but is less effective in complex or irregular spaces due to its reliance on a grid for path calculation. The Dynamic Window Approach reduces collisions but may struggle in unpredictable environments, as it predicts obstacle movements within a timeframe. Artificial Potential Fields, which navigate around obstacles using virtual forces, can be problematic in cluttered spaces, leading to complex and inefficient path planning.

The use of neural network models in advanced cleaning robots has its own constraints. Models like Deep Q-Networks and Siamese Networks, despite their success, can fail in certain situations. These failures stem from limitations in training data, model complexity, and the unpredictability of real-world scenarios. These models require extensive training and computational resources, which can be impractical or costly, particularly for large-scale or commercial use. Their ability to adapt to environments different from their training data is also questionable. For instance, Siamese Networks might not perform well in significantly different environments. The hardware and cost requirements for these advanced algorithms and models further limit their widespread adoption, posing barriers, especially for smaller or consumer-level applications.

References

[1] A. Pandey, A. Kaushik, A. K. Jha, and G. Kapse, "A technological survey on autonomous home cleaning robots," *International Journal of Scientific*, 2014.

[2] C. Luo and S. X. Yang, "A real-time cooperative sweeping strategy for multiple cleaning robots," in *Proceedings of the IEEE Internatinal Symposium on Intelligent Control*, 2002, pp. 660–665.

[3] A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni, "Path Planning and Trajectory Planning Algorithms: A General Overview," in *Motion and Operation Planning of Robotic Systems: Background and Practical Approaches*, G. Carbone and F. Gomez-Bravo, Eds. Cham: Springer International Publishing, 2015, pp. 3–27.

[4] H.-Y. Zhang, W.-M. Lin, and A.-X. Chen, "Path Planning for the Mobile Robot: A Review," *Symmetry* , vol. 10, no. 10, p. 450, Oct. 2018.

[5] V. Kunchev, L. Jain, V. Ivancevic, and A. Finn, "Path Planning and Obstacle Avoidance for Autonomous Mobile Robots: A Review," in *Knowledge-Based Intelligent Information and Engineering Systems*, 2006, pp. 537–544.

[6] P. Švestka and M. H. Overmars, "Coordinated path planning for multiple robots," *Rob. Auton. Syst.*, vol. 23, no. 3, pp. 125–152, Apr. 1998.

[7] K. Karur, N. Sharma, C. Dharmatti, and J. E. Siegel, "A Survey of Path Planning Algorithms for Mobile Robots," *Vehicles*, vol. 3, no. 3, pp. 448–468, Aug. 2021.

[8] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, "Path Planning for Autonomous Mobile Robots: A Review," *Sensors* , vol. 21, no. 23, p. 7898, Nov. 2021.

[9] J. Kim *et al.*, "Control strategies for cleaning robots in domestic applications: A comprehensive review," *Int. J. Adv. Rob. Syst.*, vol. 16, no. 4, p. 1729881419857432, Jul. 2019.

[10] H. Shakhawat Hossen Prayash, M. R. Shaharear, M. F. Islam, S. Islam, N. Hossain, and S. Datta, "Designing and Optimization of An Autonomous Vacuum Floor Cleaning Robot," in *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, 2019, pp. 25–30.

[11] T. Bai, Z. Fan, M. Liu, S. Zhang, and R. Zheng, "Multiple Waypoints Path Planning for a Home Mobile Robot," in *2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, 2018, pp. 53–58.

[12] T. Davies and A. Jnifene, "Multiple Waypoint Path Planning for a Mobile Robot using Genetic Algorithms," in *2006 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, 2006, pp. 21–26.
[13] B. Fu *et al.*, "An improved A* algorithm for the industrial robot path planning with high success rate and short length," *Rob. Auton. Syst.*, vol. 106, pp. 26–37, Aug. 2018.
[14] S. M. Persson and I. Sharf, "Sampling-based A* algorithm for robot path-planning," *Int. J. Rob. Res.*, vol. 33, no. 13, pp. 1683–1708, Nov. 2014.
[15] A. K. Guruji, H. Agarwal, and D. K. Parsediya, "Time-efficient A* Algorithm for Robot Path Planning," *Procedia Technology*, vol. 23, pp. 144–149, Jan. 2016.
[16] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment," *Auton. Robots*, vol. 45, no. 1, pp. 51–76, Jan. 2021.
[17] M. Kobayashi and N. Motoi, "Local Path Planning Method Based on Virtual Manipulators and Dynamic Window Approach for a Wheeled Mobile Robot," in *2021 IEEE/SICE International Symposium on System Integration (SII)*, 2021, pp. 499–504.
[18] M. Dobrevski and D. Skočaj, "Adaptive Dynamic Window Approach for Local Navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 6930–6936.
[19] T. Molnar and E. Culurciello, "Capsule Network Performance with Autonomous Navigation," *arXiv [cs.LG]*, 08-Feb-2020.
[20] Y. Yeboah, C. Yanguang, W. Wu, and S. He, "Autonomous Indoor Robot Navigation via Siamese Deep Convolutional Neural Network," in *Proceedings of the 2018 International Conference on Artificial Intelligence and Pattern Recognition*, Beijing, China, 2018, pp. 113–119.
[21] M. Gao, L. Jin, Y. Jiang, and B. Guo, "Manifold Siamese Network: A Novel Visual Tracking ConvNet for Autonomous Vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1612–1623, Apr. 2020.
[22] V. Kok, M. Olusanya, and A. Ezugwu, "A Few-Shot Learning-Based Reward Estimation for Mapless Navigation of Mobile Robots Using a Siamese Convolutional Neural Network," *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 12, no. 11, p. 5323, May 2022.
[23] J. Sergeant, N. Sünderhauf, M. Milford, and B. Upcroft, "Multimodal deep autoencoders for control of a mobile robot," in *Proc. of Australasian Conf. for robotics and automation (ACRA)*, 2015.
[24] K. Hu and P. O'Connor, "Learning a Representation Map for Robot Navigation using Deep Variational Autoencoder," *arXiv [cs.RO]*, 05-Jul-2018.
[25] B. Ivanovic, K. Leung, E. Schmerling, and M. Pavone, "Multimodal Deep Generative Models for Trajectory Prediction: A Conditional Variational Autoencoder Approach," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 295–302, Apr. 2021.
[26] S. Odaibo, "Tutorial: Deriving the Standard Variational Autoencoder (VAE) Loss Function," *arXiv [cs.LG]*, 21-Jul-2019.
[27] P. Fiorini and E. Prassler, "Cleaning and Household Robots: A Technology Survey," *Auton. Robots*, vol. 9, no. 3, pp. 227–235, Dec. 2000.