Enhancing Navigation Efficiency in Robotics with PRM-DDPG

Abbas Nadhim Kadhim
Department of Electronics and Communication Engineering, Al-Nahrain University, Iraq
abbasnadhom74@gmail.com (corresponding author)

Muhammed Sabri Salim
Department of Electronics and Communication Engineering, Al-Nahrain University, Iraq
musabril1967@yahoo.com

Received: 29 March 2025 | Revised: 18 April 2025 | Accepted: 24 April 2025

 DOI: https://doi.org/10.48084/etasr.11213

ABSTRACT

This study describes a new way to plan the paths of mobile robots. It combines the large-scale, global planning power of Probabilistic Roadmaps (PRM) with the local, flexible decision-making power of Deep Reinforcement Learning (DRL). While PRM focuses on waypoint delineation, Deep Deterministic Policy Gradient (DDPG) focuses on real-time obstacle avoidance. By integrating these two approaches, the proposed PRM-DDPG algorithm significantly enhances the robot's navigation capabilities, allowing it to effectively handle both structured and complex environments. In the performed simulations, PRM-DDPG outperforms sampling-based methods, such as PRM and RRT in terms of path length, time efficiency, and obstacle avoidance, especially in difficult environments. In addition, the PRM-DDPG algorithm produced the shortest path of 27.0182 m with only six corners, while methods, such as ID3QN and Genetic Algorithm (GA), produced longer paths with more corners. Fewer corners indicate a smoother and more direct path. The results show that using both PRM and DDPG together produces paths that are faster and smoother than those produced by classical or pure machine learning methods alone. The proposed PRM-DDPG algorithm will advance mobile robotics by enabling smarter, more flexible, and more effective self-navigation systems for real-world applications.

Keywords-path planning; mobile robots; PRM; DDPG; DRL, PRM-DDPG

I. INTRODUCTION

Path planning for the mobile robot is an important and very crucial research area that focuses on the development of advanced algorithms, specifically designed to allow robots to traverse efficiently and autonomously through diverse environments. Planning a path seems to be a remedy for keeping several elements in view, including time, energy consumption, and the general goal [1]. Figure 1 shows the path planning flowchart of a mobile robot, where the robot processes sensor data, plans a path, moves, and checks if the goal is reached. Conventional path planning methods sometimes use existing heuristics and pre-existing maps. This can make robotic systems less able to adapt and operate in complex, uncertain, or completely new environments [2, 3]. In recent years, Reinforcement Learning (RL) has rapidly emerged as a powerful and new method to effectively address the challenges of mobile robot path planning. This creative approach allows robots to develop optimal navigation techniques by constantly engaging with their environment and using feedback in the form of incentives or penalties depending on their decisions, behaviors, and paths [4]. The careful use of RL ideas can greatly improve the ability of robots to make decisions and adapt to different dynamic environments. Navigation becomes better, more flexible, and more consistent in many different situations. Several RL methods, including Q-learning and DRL, have been successfully applied to various path planning challenges revealing their effectiveness and flexibility [5]. These advanced technologies enable robots to maneuver around their environment with more intelligence and understanding. Through continuous trial and error, they can choose the best paths, learn from mistakes, and gradually improve their performance. The development of RL for mobile robot path planning has the potential to transform robotic operations in challenging real-world environments and lead to a new era of robotic navigation, providing unparalleled autonomy and adaptability. However, one of the RL drawbacks is the computational and time-consuming nature of iteratively analyzing the environment to generate optimal policies. Especially in complicated or dynamic environments, where the agent must adapt to changing environments, this issue often leads to less-than-optimal learning [6]. To address the limitations of conventional path planning, which often lacks adaptability in complex environments, and the computational intensity and potential suboptimality of RL, which can hinder real-time decision making in complex or rapidly changing scenarios, the study proposes:

- a hybrid approach to mobile robot path planning by exploiting the advantages of global planning and local planning to improve the path planning of the robot
- to extend the PRM path planning approach with DDPG to a PRM-DDPG planner, which outperforms the sampling-based algorithms
- to evaluate and analyze the performance of PRM-DDPG on various application scenes compared to PRM and RRT.

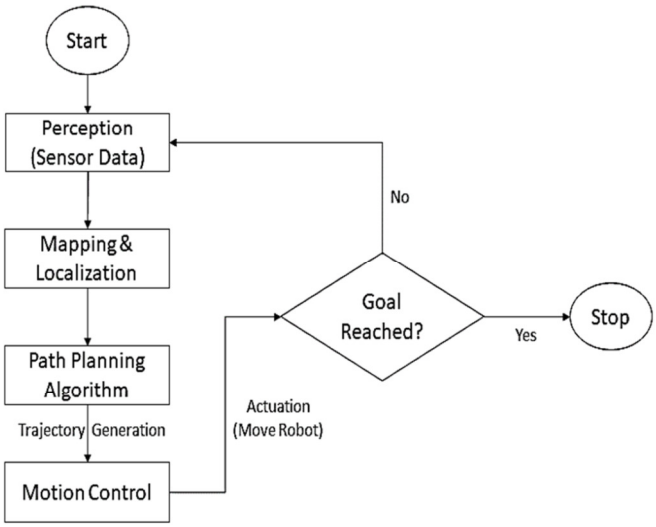


Fig. 1. Flowchart of mobile robot path planning.

II. RELATED WORK

In the field of mobile robot path planning RL has become a popular method due to its ability to facilitate novel strategies for robots to navigate complex environments. Researchers have investigated various RL techniques to enhance robots' mobility and judgment capabilities. Authors in [7] applied the Dynamic Window Approach (DWA) with enhanced evaluation functions and Q-learning to optimize parameters, thereby enhancing mobile robot navigation in unfamiliar environments. Integrating Q-learning facilitates the dynamic adjustment of the weights of multiple evaluation functions. In the absence of a comprehensive global map, the robot is capable of dynamically reacting to barriers and maximizing its path towards a target. The efficacy of the algorithm in navigating complex scenarios that change over time is demonstrated by its ability to circumvent obstacles and reach destinations with greater efficiency than conventional stationary path planning approaches. This conclusion is evidenced by both simulations and real-world trials. Authors in [8] examined the consistency of paths traversed by an Autonomous Vehicle (AV) by employing a DDPG approach. This configuration ensures that the AV is capable of operating within a limited range of environments. According to the aforementioned method, an agent is capable of acquiring the ability to propose navigation pathways. This capability is enabled through the usage of the Lyapunov function, which is derived from the DDPG training reward structure. The primary objective of this function is to ascertain stability, with the findings indicating that the agent employs consistent routes, enabling the AV to achieve its objectives within a designated timeframe. This suggests that the method of stability analysis is a viable tool for assessing the agent's navigation performance in varied environments. Authors in [9] proposed a novel incremental training method, termed PRM+TD3, which integrates DRL with the PRM. This integration enables alternative path planning for mobile robots. The solution is initiated with training in a lightweight 2D environment to optimize network parameters, thereby addressing convergence issues in difficult scenarios. Subsequently, the instructed model should be transferred to a three-dimensional environment to facilitate enhanced training, a method that significantly enhances growth and improves the model's flexibility. The efficacy of the methodology under consideration exceeded that of certain alternative approaches. Authors in [10] proposed a novel integration of DDPG hierarchical RL and neural networks with the aim of enhancing the path planning capabilities of mobile robots. The integration of the algorithm into a two-wheel differential mobile robot resulted in enhanced path smoothness and convergence speed, thus optimizing navigation. The findings indicated that DDPG outperformed conventional methods by showing superior performance across diverse environments and by achieving a 91% reduction in convergence time compared to Q-learning. After a thorough evaluation of the available evidence, it can be concluded that the proposed method significantly enhances the mobility and adaptability of mobile robots in complex environments. This finding underscores the practical relevance of the method and lends credibility to its application in real-world scenarios. Authors in [11] developed an autonomous path planning model for unmanned ships with DRL, improving navigation safety and efficiency. A C++ electronic chart platform was designed to facilitate the development of a DDPG algorithm. This innovation empowered dynamic tracking through environmental interaction. The model was assessed across a range of scenarios to ascertain its adherence to navigation regulations. The advanced DRL demonstrated a reduction in navigation errors, along with enhanced convergence speed and accuracy, in comparison to previous methodologies.

Authors in [12] enhanced path planning algorithms by designing a model that replicates human learning, enabling efficient navigation of varied environments. The agent acquires the ability to discern the most efficacious course of action, contingent upon its circumstances and any impediments, through the usage of an RL method that has been implemented. This objective is accomplished through the application of a structured reward system that acknowledges the avoidance of collisions and the successful completion of objectives. The findings indicate substantial enhancements in convergence speed, planning success rate, and path accuracy when contrasted with conventional Deep Q Network (DQN) methodologies. The proposed PN-DQN model exhibits superior performance in dynamic adjustment to changes, achieving more precise and frequent estimation in complex environments. Authors in [13] examined whether the constraints of conventional DQN connected to reward modeling and experience replay can be solved, thereby enhancing robot path planning in complicated situations. The

robot's initial approach is redefined based on Rapidly-Exploring Random Tree (RRT), and a customized reward function for free locations motivated by the A algorithm is designed. The experimental results indicate that the enhanced DDQN facilitates the robot's effective location of optimal courses and successful navigation of obstacles, achieving a successful path planning rate of approximately 80% under testing conditions. This outcome substantiates the efficacy of the proposed method. The ID3QN algorithm is a sophisticated system that has been developed to address a range of challenges, including the avoidance of collisions, the optimization of path length, and the minimization of radiation exposure. Authors in [14] proposed an enhancement to mobile robot path planning in radioactive situations. The architecture of the neural network was optimized, action selection was improved, and a prioritized experience replay system was employed. The simulation results demonstrated that the ID3QN algorithm significantly outperformed conventional algorithms, such as A, GA and Ant Colony Optimization (ACO), with a 15.6% path length, 23.5% accumulated radiation dose, and faster convergence by approximately 2,300 episodes. In order to address the limitations identified in previous research, a hybrid approach is proposed in the current work that integrates the strengths of PRM for global planning and DRL for local planning. The proposed approach will facilitate the acquisition of optimal navigation strategies by robots through continuous interaction and feedback from their surroundings.

III. METHODOLOGY

The problem of path planning is described using the framework of a Markov Decision Process (MDP). This problem is expressed as a set of states S, actions A, transition probabilities $P(s'|s,a)$, rewards $R(s,a)$, and a discount factor $\gamma \in [0,1]$ [15]. The state space S, represents the observations that the agent receives from the environment, while the action space A consists of all possible actions that the agent can take. The actor network π (S, θ) defines the policy by mapping states to actions that maximize long-term rewards, whereas the critic network Q (S, A, φ) estimates the expected cumulative reward for a given state-action pair. It can be observed that both the actor and critic networks possess corresponding target networks π_t (S, θ_t) and Q_t (S, A, θ_t)) that undergo periodic updates to ensure the stability of the training process. The reward function R (s, a) is implicitly modeled by the critic network, which evaluates the quality of actions taken by the actor. The agent's interaction with the environment is characterized by the execution of an action A_t at a state S_t, which results in a transition to a new state $S_{(t+1)}$ based on the transition probability $P(s'|s,a)$. This transition is followed by the collection of a reward R (S_t, A_t). The discount factor γ, ensures that the agent achieves a balance between long-term benefits and immediate gratification. This iterative process refines the actor and critic networks to optimize the policy and value functions, thereby empowering the agent to learn ideal conduct in a continuous action environment.

A. The Mobile Robot Testing Environments

Two environments were constructed to test the mobile robot. The First Environment (FE) is characterized by its abundance of opportunities for mobility and discovery. In comparison, the Second Environment (SE) is characterized by a complex, small-scale scene with a high density of stationary objects. The robot must navigate and interact with numerous stationary objects, thus necessitating the incorporation of these hurdles, which introduce a unique set of challenges. While the small-scale environment necessitates meticulous consideration of the numerous challenges that influence navigation dynamics, the large-scale environment offers a greater array of opportunities for movement. As portrayed in Figure 2, the mobile robot was evaluated in two distinct settings.

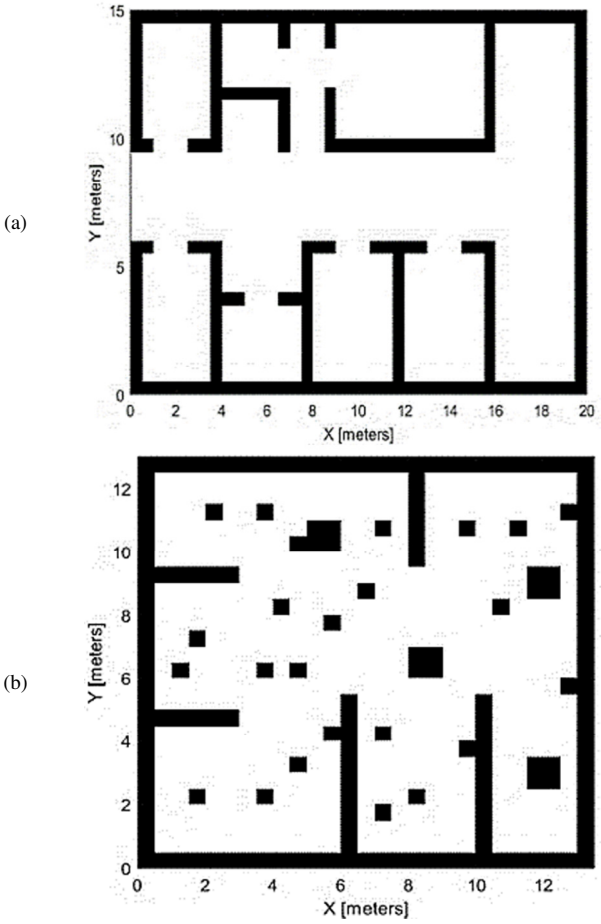


Fig. 2. The test mobile robot environments: (a) FS and (b) SE.

B. Training Procedure

The training method for an agent designed to navigate a 20 m × 15 m environment involves the avoidance of obstacles. The agent has undergone extensive training over 10,000 epochs to refine its navigation capabilities and strengthen its proficiency in recognizing and avoiding potential obstacles in the allocated area, achieving a total of 12,434,464 steps. The training session has a duration of 4 h, 52 min, and 27 sec. Comprehensive training enables agents to adapt to diverse difficulties and enhance their decision-making skills in real-time situations. As presented in Figure 3, the episode reward progression for an RL agent trained on a task exhibits significant fluctuations, with occasional spikes reaching high

values, while the average reward displays a smoother trend of performance over time. It is important to note that the variability in rewards is attributable to exploration noise, stochastic environments, and sensitivity to conditions. However, stability is achieved through the use of experience replay and noise decay. Sensory noise and hardware constraints, including inaccuracies or actuator limits, have the potential to adversely affect performance. These issues can be mitigated by implementing filtering mechanisms, robust training methodologies, and hardware-aware strategies.

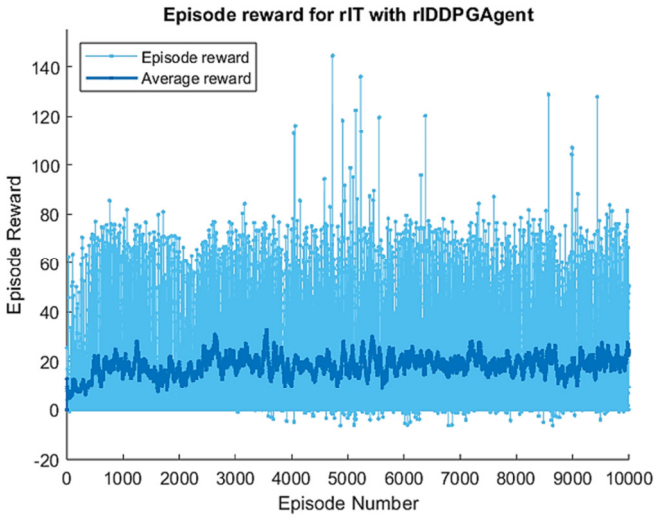


Fig. 3. The reward achieved by the agent.

C. *Probabilistic Roadmaps Algorithm*

The convergence of robotics and computational geometry results in the development of the PRM algorithm, which represents a significant advancement in path planning methodologies. The PRM algorithm, developed in the 1990s, was specifically designed to address the complexities inherent in motion planning within high-dimensional spaces [16]. This approach entails the formulation of a probabilistic framework, a strategy that facilitates navigation through challenging environments, a task that conventional deterministic methods occasionally encounter difficulties with. The PRM approach is divided into two primary phases: the query phase and the learning phase. The method of training involves the usage of random point sampling within the environment's free configuration space. These sites are then connected depending on the presence of roads that are free of obstacles. This process produces a road map that captures the connectivity of the designated area. The algorithm uses a pre-built road map to search the existing network of links in the query phase, with the objective of identifying a workable route. This process is initiated at specific points, delineated by the initial and final coordinates [17]. Figure 4 illustrates the PRM working procedure, while Figure 5 provides a PRM algorithm demonstration. This approach enables effective navigation of challenging environmental layouts giving the ability to manage spaces with many dimensions and modify them to match various contexts by varying sample density and roadmap connection [18]. The system's efficacy is further demonstrated by its ability to function effectively in scenarios where the configuration space becomes too complex for conventional techniques. This capability is attributed to its random search of potential paths [19]. However, it is important to note that the PRM algorithm is not without its limitations. The efficacy of the generated roadmap is significantly influenced by the sampling strategy employed. Suboptimal sampling may result in disconnected paths or less efficient routes [16]. Furthermore, the process of generating a roadmap for the first time can necessitate a substantial amount of computing power, particularly in scenarios where obstacles undergo rapid changes or where real-time processing is required [20].

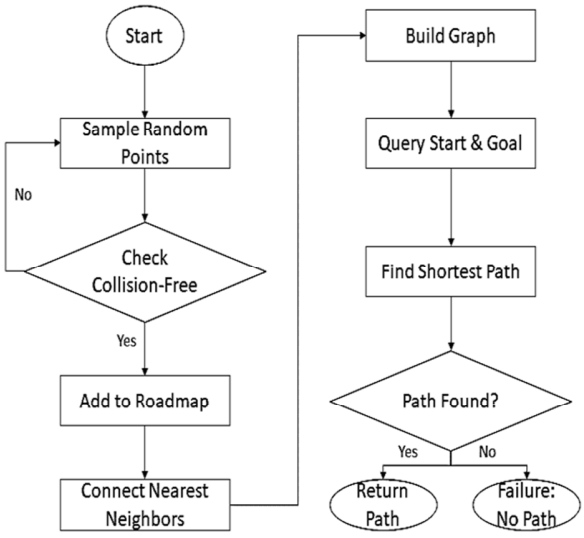


Fig. 4. Flowchart of the PRM algorithm.

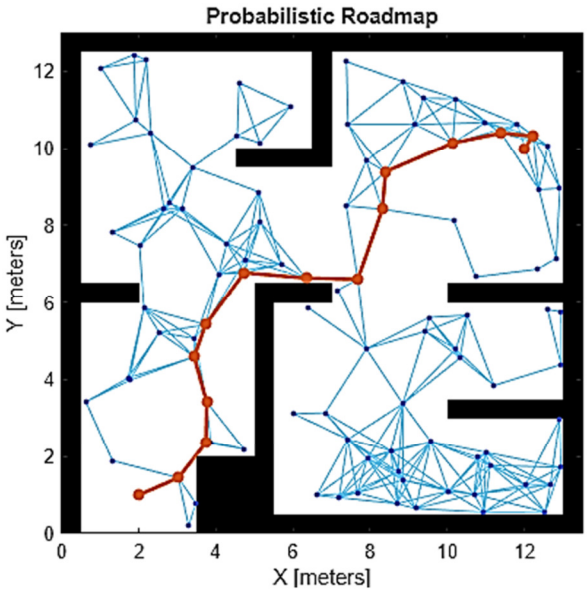


Fig. 5. Demonstration of the PRM algorithm.

D. Reinforcement Learning

RL is a machine learning principle in which an agent learns to optimize decisions by taking actions within an environment to maximize cumulative rewards over time. Inspired by the principles of behavioral psychology, it underscores the significance of reinforcement in the context of learning, using the following components: the agent, the environment, actions, rewards, and states [21]. The agent interacts with the environment, acting according to its current state and receiving rewards as feedback. As presented in Figure 6, this interaction loop constitutes a fundamental component of the RL process, showing the manner in which an agent engages with an environment through the execution of actions, subsequently receiving feedback in the form of rewards. This cycle continues as the agent learns to maximize cumulative rewards, thereby improving its decision-making over time. The objective of this initiative is to devise a comprehensive policy, or more precisely, a strategy for action in diverse states, with the overarching aim of optimizing long-term benefits [22]. Common RL algorithms, such as Q-learning and policy gradients, estimate policies and value functions in disparate ways. The application of RL in robotics, gaming, and autonomous vehicles highlights its efficacy in handling complex tasks, particularly in settings with delayed incentives [21, 23]. Its adaptive qualities help agents to increase their performance via experience, hence guiding good decisions.

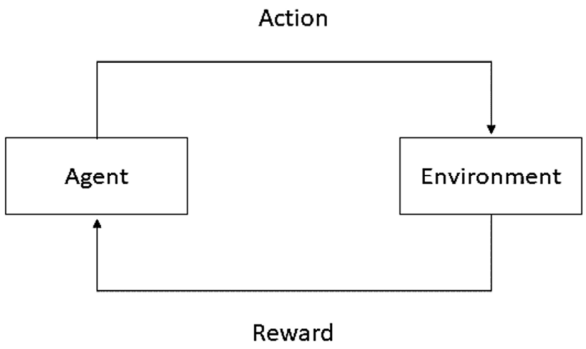


Fig. 6. RL block diagram.

By enabling robots to develop optimal navigation methods through interaction with their environment, RL emerges as a potent instrument for mobile robot path planning. In the context of mobile robots, this relates to the navigation of obstacles and the optimization of travel time by moving from a starting point to a target [24]. The usage of RL in the field of mobile robot path planning typically entails the construction of a state space that encompasses the robot's surroundings, in conjunction with the actions it is capable of executing to ensure mobility. The RL algorithm is designed to incentivize positive behaviors, such as the completion of objectives or the successful navigation of challenges, while discouraging undesired actions. The algorithm's ability to adapt and scale to different levels of complexity is a significant strength, making it a promising approach for a wide range of applications. In dynamic environments where conditions are subject to change and robots are capable of modifying their approach based on newly acquired knowledge, RL can prove beneficial [25]. Furthermore, it has the capacity to manage intricate, high-dimensional state spaces that traditional methodologies are unable to effectively address. Nevertheless, this approach is not without its drawbacks. Real-time applications may encounter challenges when training an RL model due to its tendency to consume substantial processing resources and data volumes. Additionally, it is important to regulate the trade-off between exploration and exploitation, as an excess of exploration may result in suboptimal performance [26].

1) Deep Deterministic Policy Gradient

The advanced RL method, DDPG has been developed for the purpose of addressing problems within continuous action space. It efficiently learns policies that maximize expected rewards by combining the advantages of policy gradient methods with value-based approaches [15]. Deep neural networks in DDPG function as actors, determining the optimum course of action in a given situation. Figure 7 presents the flow chart of the DDPG agent, using an experience replay technique, wherein transitions are stored in a memory buffer, and subsequently sampled randomly to facilitate the learning process of the agent. This approach leads to more consistent learning and helps to differentiate consecutive samples' associations using target networks, which are gently updated replicas of the actor and critic networks, thus enhancing the stability of the training process. The algorithm is a valuable instrument in a variety of disciplines, including robotics, gaming, and artificial intelligence, given the dynamic nature of these environments. DDPG facilitates the efficient acquisition of proficiency in challenging control tasks by capitalizing on the strengths of both deep learning methods and RL techniques [8, 27].

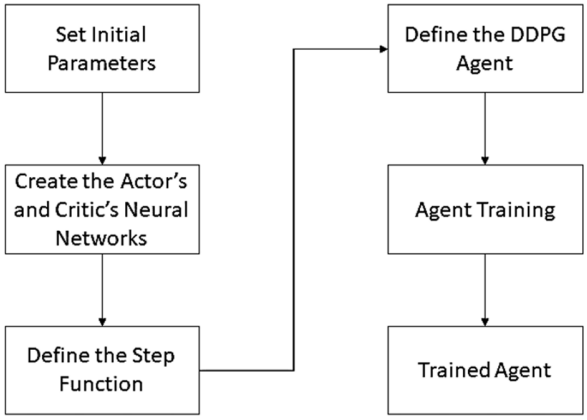


Fig. 7. Diagram of the DDPG agent's workflow.

E. PRM-DDPG Path Planning

A robotic navigation system integrates the PRM and DDPG, merging global and local planning methodologies. PRM specifies waypoints, while DDPG emphasizes real-time obstacle evasion through the usage of DRL algorithms. This collaborative approach enhances navigational efficiency and operational safety in complex environments by leveraging a combination of global and local planning methodologies.

IV. DEEP REINFORCEMENT LEARNING

A. Observation States

The objective of the DRL model is to optimize the cumulative reward R during the interactions between the agent and its environment. At any specified time, step t, the state S_t is constituted by the lidar data O_m, the geometrical relations—namely, the angle and distance O_g between the agent and the target location—as well as the linear velocity $v_{(t-1)}$ and angular velocity $\omega_{(t-1)}$ from the preceding time step. In the context of the DRL model's application as a planner f, the observed states S_t are associated with the linear velocity v_t and angular velocity ω_t, which are defined in two dimensions and represented as the action set A, and generated as outputs of the deep neural network, for the mobile robot at the current time step, as:

$$v_t, \omega_t = f(S_t) = f(O_m, O_g, v_{t-1}, \omega_{t-1}) \quad (1)$$

B. Reward Function

The proper design of a reward function is of critical importance in the context of RL. In order to address the challenge posed by the sparse rewards for the navigation task in a low-dimensional mobile robot state space, the usage of constant reward shaping has been proposed. A dense reward function is advantageous in environments that resemble a maze, as it can circumvent local minima and sporadic referees. This phenomenon facilitates the convergence process, resulting in enhanced efficiency and expedited outcomes [28]. The design of the reward function is intended to motivate the agent to maintain a considerable distance from the closest obstacle, hence reducing the probability of experiencing a negative outcome in various scenarios. Avoiding danger areas is an effective strategy that enables the agent to navigate more successfully and ensure safety. Furthermore, the agent's motivation is enhanced by a meticulously designed positive reward system that facilitates the attainment of higher linear velocities, thus suggesting an improvement in performance in straight-line movement. Conversely, a negative reward is initiated when the agent exhibits high angular velocities, which may be indicative of undesirable motions. The efficacy of the reward system in deterring the agent from engaging in circular movements is evident. This behavior is effectively guided, ensuring that the agent's actions are directed towards achieving optimal performance and stability in its objectives while navigating the environment. The reward function is:

$$R = \begin{cases} 0.015 & \text{Avoid nearest obstacle} \\ 2 & \text{Straight line motions} \\ -0.3 & \text{Going in circles} \end{cases} \quad (2)$$

C. The Agent

A DDPG agent was developed using the R2024b framework of MATLAB. This agent integrates two critical components: the actor network, which determines the appropriate actions to execute based on the prevailing environmental conditions, and the critic network, responsible for assessing the agent's actions. The critic network comprises two distinct components: a state part and an action part. The state input received from the state component is transmitted through a fully linked layer comprising 50 and 25 nodes, respectively. Upon receiving action input, the action component traverses a fully connected layer comprising 25 nodes. It has been demonstrated that every layer exhibiting complete connectivity exhibits Rectified Linear Unit (ReLU) activation behavior. ReLU is a conventional choice for the activation function in every fully connected layer. It is a straightforward operation, does not saturate in the positive domain, and facilitates the faster training of deep networks. The state part and action portion are coupled together through a ReLU layer and a full connection layer, creating the Critic network. Concurrently, the actor network receives state input and outputs action signals within a designated range. It has three completely linked layers of 50, 50, and 2 nodes, respectively, as shown in Figure 8. ReLU and tanh are, respectively, the activation functions of the fully connected layers. The learning rate for the actor is set to 1e-4, while the critic's learning rate is configured to 1e-3. The lower learning rate for the actor ensures consistent updates, as the policy directly influences actions; the higher rate for the critic facilitates more rapid Q-value function learning. Ensuring training stability is achieved by employing L2 regularization of 1e-4 to avoid overfitting, and a gradient threshold of 1 to restrict significant gradient changes. While larger batches necessitate more memory, smaller batches can result in noisier updates. Therefore, the mini batch size of 128 offers a balanced approach between computational efficiency and the capacity to generalize well. A storage capacity of 1e-6 for the experience buffer provides ample space for the accumulation of past experiences, reducing the probability of overfitting and update correlation. Exploration necessitates the usage of noise options. When set at 0.1, variance provides early training with sufficient randomness to facilitate efficient exploration. As training progresses and the agent's learned strategy is refined, the variance decay rate of 1e-5 progressively diminishes noise, facilitating the transition from exploration to exploitation. The selection of these numbers is intended to establish a balance between continuous action, stability, effective exploration, and convergence. Following the determination of the network framework, the approximation methods for training the deep neural network are presented from the following equations. The target value y_i integrates the immediate reward and the discounted future reward estimated by the target networks, ensuring that the agent learns long-term rewards. The critic is updated by minimizing a mean squared error loss L, which measures the difference between the predicted Q-value and the target value. The actor is updated using the policy gradient, with the critic directing the actor toward actions that enhance the Q-value. In order to stabilize the training process, the target networks are updated in a gradual manner using a smoothing factor, denoted by τ, which serves to blend the current and target parameters. Collectively, these updates empower the agent to enhance its policy and value function while preserving stability during the training process [15]:

$$y_i = R_i + \gamma Q_t(S_i'; \pi_t(S_i'; \theta_t); \varphi_t) \quad (3)$$

$$L = \frac{1}{2M}\sum_{i=1}^{M}(y_i - Q(S_i, A_i; \varphi))^2 \quad (4)$$

$$\nabla_\theta J \approx \frac{1}{M}\sum_{i=1}^{M} G_{ai} G_{\pi i} \quad (5)$$

$$G_{ai} = \nabla_A Q(S_i, A; \varphi) \quad \text{where } A = \pi(S_i; \theta) \quad (6)$$

$$G_{\pi i} = \nabla_{\theta}\pi(S_i;\theta) \quad (7)$$

$$\varphi_t = \tau\varphi + (1-\tau)\varphi_t \quad (8)$$

$$\theta_t = \tau\theta + (1-\tau)\theta_t \quad (9)$$

where R_i is the immediate reward received after taking action and $\gamma \in [0,1]$ is the discount factor, which determines how much the agent values future rewards compared to immediate rewards. M is the batch size (number of experiences sampled for training), G_{ai} and $G_{\pi i}$ are the gradient of the critic and actor output, respectively, and the φ and θ are the critic and actor parameters, respectively.

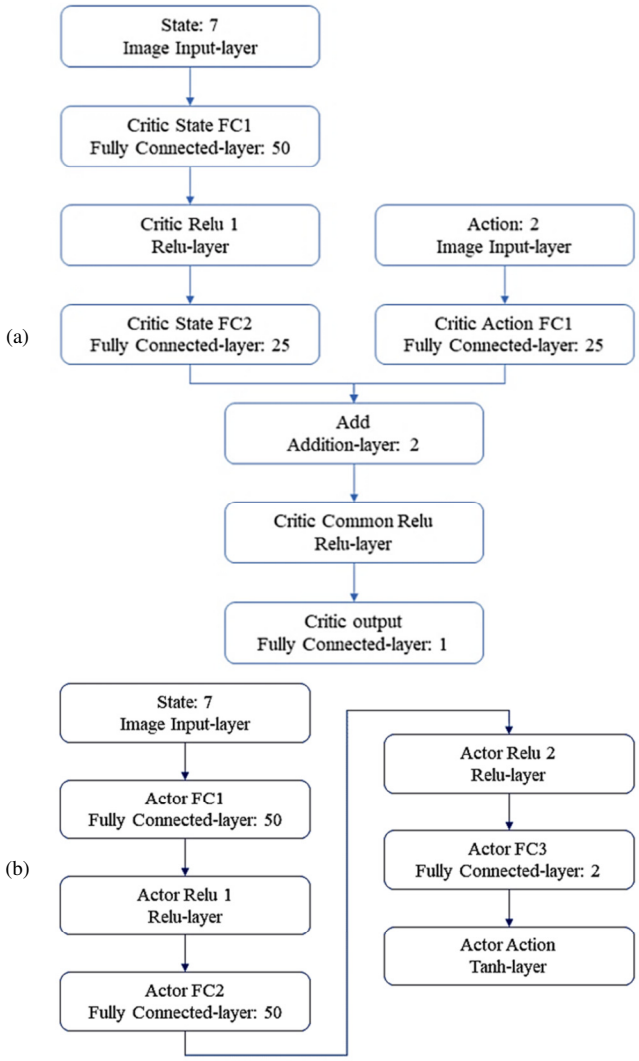


Fig. 8. Deep neural network of DDPG: (a) critic network structure, (b) actor network structure.

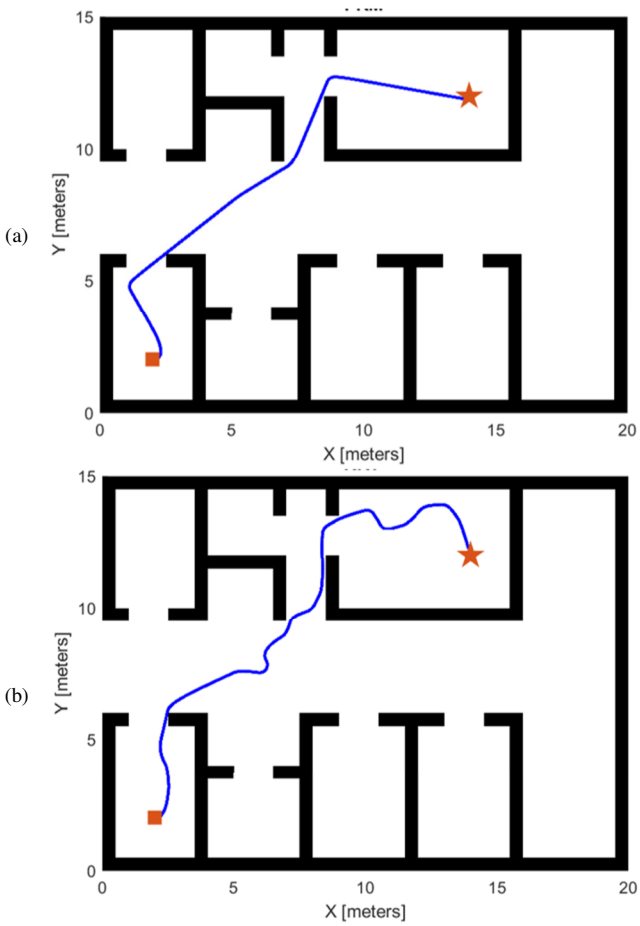
V. RESULTS AND DISCUSSION

The simulation study used MATLAB version R2024b, an AMD Ryzen 7 5800H CPU, Radeon Graphics, and 16 GB RAM in two environments, comparing robot parameters, such as linear velocity, angular velocity, 2D lidar scanning distance, and scanning angle, while the parameters of the robot are:

- Linear velocity (0m/s - 0.3 m/s)
- Angular velocity (-1 rad/s – 1 rad/s)
- 2D lidar scanning distance (0.2 m-3 m)
- Lidar scanning angle (-1.178-1.178)

A. Comparison of the Proposed PRM-DDPG Approach with the Sampling-based Method

The findings indicate that PRM-DDPG exhibits superior performance in comparison to both PRM and RRT in a 20 m × 15 m static area, as evidenced by metrics, such as path length, time efficiency, and path smoothness. Moreover, when evaluated in comparison to alternative algorithms, PRM-DDPG demonstrates superior performance in terms of obstacle avoidance. In order to calculate the path length, the start point was set at (2, 2) and the goal point was set at (14, 12). The results show that PRM-DDPG achieved a path length of 19.70 m, whereas PRM's path length was 22.45 m, and RRT was measured at 23.12 m. Figure 9 illustrates path planning using three distinct algorithms in the FE.



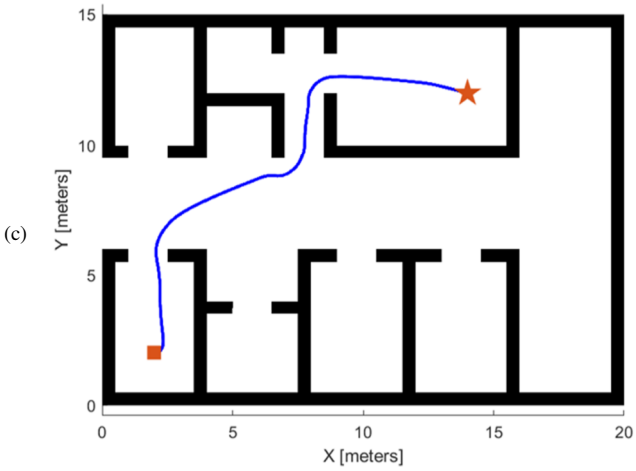


Fig. 9. Path planning of: (a) PRM, (b) RRT, and (c) PRM-DDPG in the FE.

The path generated by the PRM Probabilistic Roadmap Method appears smooth, but this depends on the quality of the sampled nodes. While RRT generates a greater number of curved paths than PRM, it remains a viable method for navigating the surroundings. The usage of PRM-DDPG appears to ensure an optimal trajectory, potentially through the application of learned policies to enhance navigation. From initiation to the achievement of the objective, all three strategies effectively provide a road forward. Although RRT may comprise a greater number of waypoints, PRM and PRM-DDPG are more likely to generate pathways that are more seamless. Consequently, by effectively addressing environmental constraints, RL has the potential to enhance the performance of the PRM-DDPG. The SE is characterized by a greater level of complexity in comparison to the first, marked by an augmented presence of static obstacles. The starting point is designated as (1, 12), while the target point is located at (9, 1). Furthermore, the implementation of the PRM-DDPG approach exhibits superior performance in complex environments. A comparison of PRM-DDPG with the PRM and RRT methods reveals that PRM-DDPG exhibits superior performance in addressing challenges posed by additional obstacles. This enhances the efficacy of the pathfinding process, leading to enhanced efficiency. However, the augmented number of obstacles results in the DDPG agent requiring more time to navigate around them efficiently, causing a slight delay in progress. As depicted in Figure 10, the three methods are represented in the SE, while Table I outlines the results for the three algorithms in the two environments. Path length: PRM-DDPG identifies the shortest path in both contexts, suggesting that the learning-based strategy (DDPG) may efficiently maximize paths. From the perspective of arrival time, in the FE: it is evident that PRM-DDPG is the most efficient solution in terms of both time and computational resources, with a level of performance that nearly exceeds that of PRM and significantly surpasses that of RRT, and in the SE: although PRM-DDPG maintains its status as the shortest path, RRT exhibits a notably higher level of processing speed.

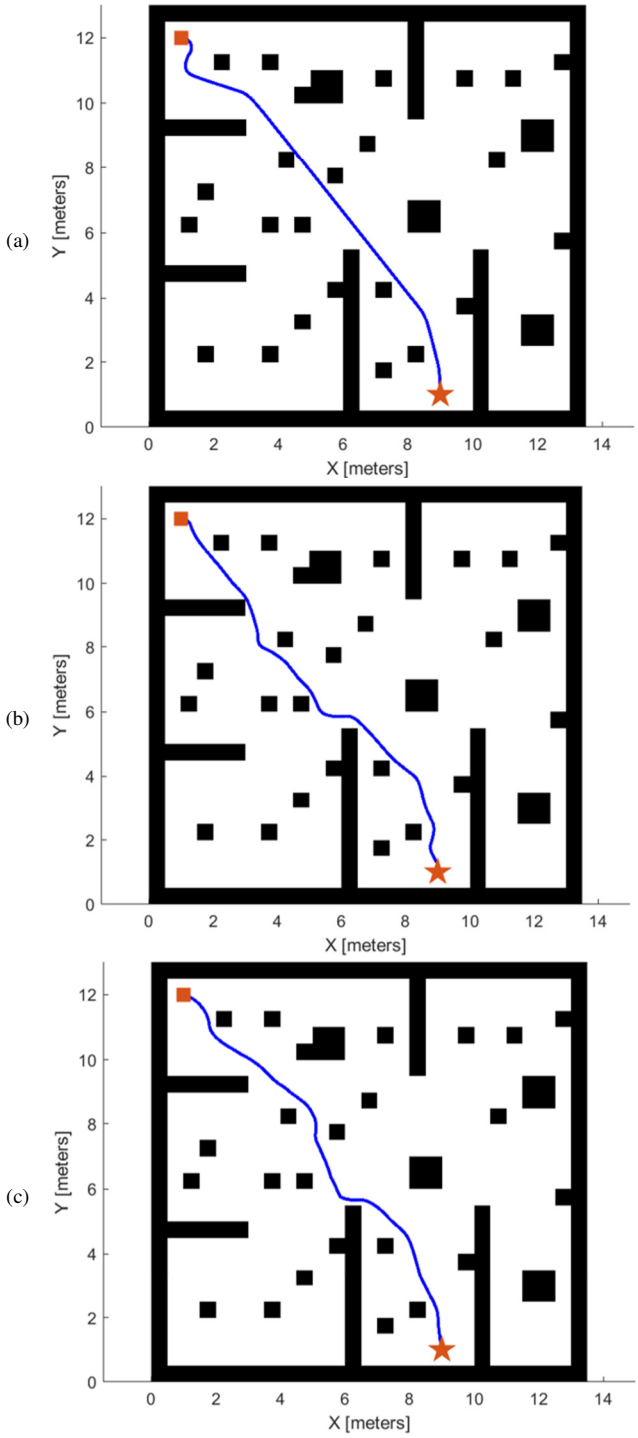


Fig. 10. Path planning of: (a) PRM, (b) RRT, and (c) PRM-DDPG in the SE.

B. Comparison with Previous Works

Authors in [13] mention that the study enhances robot path planning using an improved Double DQN algorithm. This algorithm integrates strategies from A and RRT to efficiently navigate complex environments and optimize obstacle avoidance. Table II, presents a synopsis of PRM-DDPG in

comparison to alternative methodologies. Authors in [14] introduce the improved dueling deep double Q network (ID3QN) algorithm, which is designed to optimize path planning for mobile robots in radioactive environments, enhancing safety and efficiency during operations.

In Table III, a synopsis of PRM-DDPG is provided in comparison to the other techniques mentioned in [14] across two environments. The results for each method, including the path length (m) and the number of corners (turns) are presented. It was observed that the shortest path produced by PRM-DDPG at 27.0182 had six corners, and the fewest corners were generated by PRM-DDPG and DQN, indicating a smoother or more direct path. It must be noted that alternative methods (ID3QN, GA) yield pathways of greater length (≥ 29.7990 m) and higher number of corners. The findings indicate that, when used in conjunction with DDPG, PRM has the capacity to generate pathways that are both more concise and more seamless in comparison to those produced by classical or entirely RL-based methodologies when employed in isolation.

TABLE I. THE RESULTS OF THE THREE ALGORITHMS IN THE TWO ENVIRONMENTS

	Algorithm	Path length (m)	Time (sec)
FE	PRM	22.45	66.7
	RRT	23.12	74.2
	PRM-DDPG	19.70	66.3
SE	PRM	14.44	48.5
	RRT	14.61	47.8
	PRM-DDPG	14.26	50.3

TABLE II. COMPARISON WITH OTHER TECHNIQUES.

Algorithm	Path length (m)	Number of corners
Improved DDQN [13]	22.97	9
A	25.31	9
Proposed algorithm DDPG-PRM	22.21	2

TABLE III. COMPARISON WITH OTHER STUDIES

Algorithm		ID3QN [14]	D3QN	DQN	A	GA	ACO	Proposed PRM-DDGP
FE	Path length (m)	29.7990	30.3848	35.3137	30.9706	29.7990	31.5563	27.0182
	Number of corners	10	9	6	11	9	8	6
SE	Path length (m)	29.7990	31.5563	32.1412	30.9706	31.5563	31.5563	28.0354
	Number of corners	13	11	12	13	11	9	8

VI. CONCLUSIONS

This work presents an innovative methodology for mobile robot path planning, integrating the advantages of Probabilistic Road Mapping (PRM) for extensive planning and Deep Reinforcement Learning (DRL) for detailed planning. The integration of these methodologies has been shown to enhance the robot's navigational proficiency, enabling it to navigate both organized and intricate situations with adeptness. These findings underscore the significance of using sophisticated learning algorithms in robotic navigation, facilitating the development of more autonomous and flexible robotic systems for practical applications. The proposed PRM-DDPG algorithm is evaluated against alternative approaches across two distinct settings [14]. The evaluation process yields a shortest path measuring 27.0182 m and comprising only six corners. In contrast, alternative approaches, such as ID3QN and Genetic Algorithm (GA), yield pathways of greater length (≥ 29.7990 m) that are distinguished by an augmented number of corners. The findings suggest that the integration of PRM with DDPG yields pathways that are both more concise and efficient in comparison to those generated by conventional or machine learning methodologies. This work contributes to the advancement of mobile robotics by facilitating the development of autonomous navigation systems that are more intelligent and efficient.

REFERENCES

[1] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, Oct. 2023, Art. no. 120254, https://doi.org/10.1016/j.eswa.2023.120254.

[2] A. Hoang, S. T. Nguyen, T. V. Pham, T. M. Pham, L. V. Trieu, and T. T. Cao, "A Bayesian Neural Network-based Obstacle Avoidance Algorithm for an Educational Autonomous Mobile Robot Platform," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12183–12189, Dec. 2023, https://doi.org/10.48084/etasr.6304.

[3] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang, and Z. Cao, "Review of Autonomous Path Planning Algorithms for Mobile Robots," *Drones*, vol. 7, no. 3, Mar. 2023, Art. no. 211, https://doi.org/10.3390/drones7030211.

[4] K. Zhu and T. Zhang, "Deep reinforcement learning based mobile robot navigation: A review," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 674–691, Oct. 2021, https://doi.org/10.26599/TST.2021.9010012.

[5] Y. Zhao, Y. Zhang, and S. Wang, "A Review of Mobile Robot Path Planning Based on Deep Reinforcement Learning Algorithm," *Journal of Physics: Conference Series*, vol. 2138, no. 1, Dec. 2021, Art. no. 012011, https://doi.org/10.1088/1742-6596/2138/1/012011.

[6] H. Sun, W. Zhang, R. Yu, and Y. Zhang, "Motion Planning for Mobile Robots—Focusing on Deep Reinforcement Learning: A Systematic Review," *IEEE Access*, vol. 9, pp. 69061–69081, 2021, https://doi.org/10.1109/ACCESS.2021.3076530.

[7] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment," *Autonomous Robots*, vol. 45, no. 1, pp. 51–76, Jan. 2021, https://doi.org/10.1007/s10514-020-09947-4.

[8] M. Cabezas-Olivenza, E. Zulueta, A. Sanchez-Chica, U. Fernandez-Gamiz, and A. Teso-Fz-Betoño, "Stability Analysis for Autonomous Vehicle Navigation Trained over Deep Deterministic Policy Gradient," *Mathematics*, vol. 11, no. 1, Jan. 2023, Art. no. 132, https://doi.org/10.3390/math11010132.

[9] J. Gao, W. Ye, J. Guo, and Z. Li, "Deep Reinforcement Learning for Indoor Mobile Robot Path Planning," *Sensors*, vol. 20, no. 19, Jan. 2020, Art. no. 5493, https://doi.org/10.3390/s20195493.

[10] J. Yu, Y. Su, and Y. Liao, "The Path Planning of Mobile Robot by Neural Networks and Hierarchical Reinforcement Learning," *Frontiers in Neurorobotics*, vol. 14, Oct. 2020, https://doi.org/10.3389/fnbot.2020.00063.

[11] S. Guo, X. Zhang, Y. Zheng, and Y. Du, "An Autonomous Path Planning Model for Unmanned Ships Based on Deep Reinforcement Learning," *Sensors*, vol. 20, no. 2, Jan. 2020, Art. no. 426, https://doi.org/10.3390/s20020426.

[12] L. Lv, S. Zhang, D. Ding, and Y. Wang, "Path Planning via an Improved DQN-Based Learning Policy," *IEEE Access*, vol. 7, pp. 67319–67330, 2019, https://doi.org/10.1109/ACCESS.2019.2918703.

[13] F. Zhang, C. Gu, and F. Yang, "An Improved Algorithm of Robot Path Planning in Complex Environment Based on Double DQN," in *Advances in Guidance, Navigation and Control*, 2022, pp. 303–313, https://doi.org/10.1007/978-981-15-8155-7_25.

[14] Z. Wu, Y. Yin, J. Liu, D. Zhang, J. Chen, and W. Jiang, "A Novel Path Planning Approach for Mobile Robot in Radioactive Environment Based on Improved Deep Q Network Algorithm," *Symmetry*, vol. 15, no. 11, Nov. 2023, Art. no. 2048, https://doi.org/10.3390/sym15112048.

[15] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning." arXiv, Jul. 05, 2019, https://doi.org/10.48550/arXiv.1509.02971.

[16] J. Jermyn, "A Comparison of the Effectiveness of the RRT, PRM, and Novel Hybrid RRT-PRM Path Planners," *International Journal for Research in Applied Science and Engineering Technology*, vol. 9, no. 12, pp. 600–611, Dec. 2021, https://doi.org/10.22214/ijraset.2021.39297.

[17] X. Ma, R. Gong, Y. Tan, H. Mei, and C. Li, "Path Planning of Mobile Robot Based on Improved PRM Based on Cubic Spline," *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, 2022, Art. no. 1632698, https://doi.org/10.1155/2022/1632698.

[18] Q. Li, Y. Xu, S. Bu, and J. Yang, "Smart Vehicle Path Planning Based on Modified PRM Algorithm," *Sensors*, vol. 22, no. 17, Jan. 2022, Art. no. 6581, https://doi.org/10.3390/s22176581.

[19] A. M. Saeed and K. S. Rijab, "Enhancing Performance of Path Planning PRM Algorithm for Automated Boat Using PID Controller," *Journal of Global Scientific Research*, vol. 9, no. 11, pp. 3678–3689, Nov. 2024, https://doi.org/10.5281/zenodo.14039138.

[20] L. Qiao, X. Luo, and Q. Luo, "An Optimized Probabilistic Roadmap Algorithm for Path Planning of Mobile Robots in Complex Environments with Narrow Channels," *Sensors*, vol. 22, no. 22, Jan. 2022, Art. no. 8983, https://doi.org/10.3390/s22228983.

[21] Z. Ding, Y. Huang, H. Yuan, and H. Dong, "Introduction to Reinforcement Learning," in *Deep Reinforcement Learning: Fundamentals, Research and Applications*, H. Dong, Z. Ding, and S. Zhang, Eds. Singapore: Springer, 2020, pp. 47–123.

[22] H. Surmann, C. Jestel, R. Marchel, F. Musberg, H. Elhadj, and M. Ardani, "Deep Reinforcement learning for real autonomous mobile robot navigation in indoor environments." arXiv, May 28, 2020, https://doi.org/10.48550/arXiv.2005.13857.

[23] S. S. Mousavi, M. Schukat, and E. Howley, "Deep Reinforcement Learning: An Overview," in *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, Cham, Switzerland, 2018, pp. 426–440, https://doi.org/10.1007/978-3-319-56991-8_32.

[24] J. Xin, H. Zhao, D. Liu, and M. Li, "Application of deep reinforcement learning in mobile robot path planning," in *2017 Chinese Automation Congress (CAC)*, Jinan, China, Oct. 2017, pp. 7112–7116, https://doi.org/10.1109/CAC.2017.8244061.

[25] C. Zhu, "Intelligent Robot Path Planning and Navigation based on Reinforcement Learning and Adaptive Control," *Journal of Logistics, Informatics and Service Science*, vol. 10, no. 3, pp. 235–248, 2023, https://doi.org/10.33168/JLISS.2023.0318.

[26] Q. Yao *et al.*, "Path Planning Method With Improved Artificial Potential Field—A Reinforcement Learning Perspective," *IEEE Access*, vol. 8, pp. 135513–135523, 2020, https://doi.org/10.1109/ACCESS.2020.3011211.

[27] W. Hu, Y. Yang, and Z. Liu, "Deep Deterministic Policy Gradient (DDPG) Agent-Based Sliding Mode Control for Quadrotor Attitudes," *Drones*, vol. 8, no. 3, Mar. 2024, Art. no. 95, https://doi.org/10.3390/drones8030095.

[28] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, Oct. 2020, https://doi.org/10.1109/LRA.2020.3026638.