Iraqi Journal of Intelligent Computing and Informatics (IJICI)
Vol. 2,1, December 2023, pp. 1~21
ISSN: 2791-2868, DOI: 10.52940/ijici.v2i1.61

Traversing Dynamic Environments: Advanced Deep Reinforcement Learning for Mobile Robots Path Planning - A Comprehensive Review

Maysam Hammed Qasim [1], Salah Al-Darraji [2]
[1,2] Department of Computer Science, Basra University, Basra, Iraq

Article Info

Article history:

Received August15, 2024
Revised September 13, 2024
Accepted September 15, 2024

Keywords:

Dynamic Path Planning
Deep Reinforcement Learning
Actor-Critic Technique
Autonomous Mobile Robots
Avoid Obstacles

ABSTRACT

Enabling mobile robots to navigate unpredictable and ever-changing environments while avoiding static and moving obstacles is a critical challenge for dynamic path planning. Advanced sensors have simplified the robot's work by enabling it to navigate autonomously without human intervention. Optimal path planning in dynamic environments requires sophisticated algorithms considering essential factors such as time, energy, and distance. These problems can be solved using deep neural networks (DNNs) and reinforcement learning (RL). An artificial intelligence (AI) agent learns from reward signals using trial and error to identify humans' optimal behavioral strategies. This review paper explores how deep reinforcement learning (DRL) techniques can be combined with other path-planning techniques to enhance the efficiency of these methods and solutions to address the problem of efficient navigation in unfamiliar environments with obstacles, with a focus on processes such as policy gradient, model-free and model-based learning, and the actor-critic approach. We comprehensively examine the key concepts, challenges, and recent developments in DRL, focusing on its application to revolutionize robotic navigation in complex scenarios.

Corresponding Author:

Maysam Hameed Qasim
Department of Computer Science, Basra University, Basra, Iraq
Email: pgs.maysam.qasm@uobasrah.edu.iq

1. INTRODUCTION

Autonomous mobile robots have become increasingly necessary in recent years. These robots are used in many aspects of our daily lives, such as cleaning, self-driving cars, military operations, and rescue missions. In most applications, the robot must move across difficult and unfamiliar terrain without colliding with obstacles. These robots must devise a global path based on available environmental data to avoid stationary and moving obstacles. Then, a specific path is created to reach the target points along a pre-defined global path, relying on sensors such as LiDAR, RGBD or RGB cameras. Path planning algorithms are classified into traditional and heuristic algorithms [1], as shown in Figure 1. Conventional methods were used to address the path-planning problem. Some of these algorithms are unsuitable for complex and unfamiliar environments because they require complete knowledge of the environment and a detailed map for path planning [2], [3].

Path planning is crucial for robots' ability to navigate independently and without human intervention. Robot path planning challenges involve determining the most efficient path from the starting point to the target point while avoiding collision [4], [5].

RL is the last machine learning (ML) type more suitable for complex tasks, while deep learning (DL) can extract information. As a result, many researchers have considered leveraging DL's ability to extract information and RL's ability to make decisions to plan the robot's path.

In DRL, intelligent systems are built, trained by interacting with their environments and evaluated in real time. DRL approaches are frequently used in various fields, such as robotics, machine translation, control systems, text generation, target identification, autonomous driving, text-based games, and more [6].

DRL techniques have demonstrated an excellent ability to learn and adapt. They have proven their efficiency in planning the robot's path by interacting with its environment, acquiring knowledge, and learning

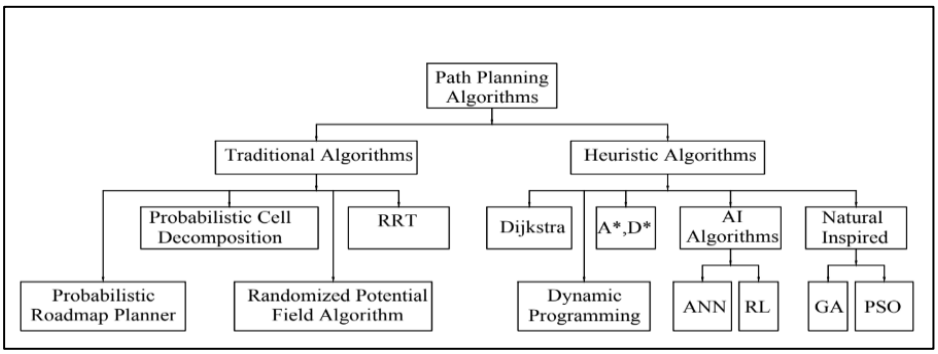

Figure 1. Shows path planning techniques

Finally, the rest of this work is organized as follows: Section 2 discusses AI-based path-planning methods and how they have been applied to solve the path planning problem. Section 3's primary goal is to summarize the concepts of DRL. Section 4 presents the most essential DRL path planning and obstacle avoidance strategies. Section 5 contains the paper's conclusions.

2. PATH PLANNING TECHNIQUES

This section discusses the most critical path-planning methods based on AI techniques and how they have been employed to solve path planning problems in complex and unfamiliar environments.

2.1. Path planning based on DL

ANN, support vector machines, and decision trees are examples of ML approaches that employ different techniques to create a predictive model using data. The models aim to predict and collect data [9].

Previously, when the processing speed of computers was limited, neural networks (NNs), which had several layers of interconnected neurons, were more effective in solving complex issues [10]. Now, DNNs use a wide range of connections and many layers of neurons. DL networks have greatly enhanced the accuracy of some basic ML tasks, allowing it to work on complex, high-dimensional issues such as distinguishing between dogs and cats in high-resolution (megapixel) images. DL allows rapid solution of complex problems involving many variables. Furthermore, it has extended ML to everyday tasks, such as speech and facial recognition on mobile devices [11].

NNs are employed in DL in ML to model and tackle complex problems. They consist of interconnected nodes organized into layers, which receive input and undergo processing and transformation. These networks are made to simulate the structure and functioning of the human brain. DL's versatility allows it to be employed in several ways to tackle the path planning issue.

Convolutional neural networks(CNNs) are utilized in algorithm processing pictures as input. NNs are employed to mimic education and deal with the Q-value problem in RL when dealing with a complicated state of action space [12].

In [13] the Deep SORT human tracking technique was utilized to monitor individuals' movements. The SSD Mobile net object recognition method was trained to expose common stains, litter on the ground, and footprints in places with substantial human presence. The dataset contains 1200 pictures for each of the four classifications: Stain, Foot Stain, Trash, and Human.

In [14] the authors presented a new and innovative method for multiple-path planning in real-time. This method combines the conventional graph-based search with semantic segmentation. A fully convolutional neural network (FCN) was initially developed to examine the ideal trajectory area produced by an A* path planning algorithm in several real-life and simulated settings. Incorporating auditory information into the localization data significantly improves the neural network's generalization capacity, even in incorrect localization findings. Subsequently, the FCN infers several possible path locations, which are subsequently employed as constraints for the subsequent A*-based path planning.

In [15] a novel graph convolutional network model, TAM-GCN, was developed to address a significant limitation of the current graph convolutional network: its inability to effectively represent the dynamic interaction among various nodes in autonomous driving. TAM-GCN addresses this problem by incorporating a trainable adjacency matrix. An approach for surpassing a deep neural network uses the TAM

GCN to build a correlation between observed data and intended actions. The network is trained and optimized using the imitation learning technique.

In [16] this work utilizes motion profiles (MP) and compact road profiles (RP) to recognize dynamic objects and path areas effectively. These profiles greatly enhance recognition by reducing video data to a smaller dimension and increasing the sensing average. To ensure the avoidance of collisions at short distances and to assist in the navigation of vehicles at medium and long distances, many reference points and measurement points are consistently scanned at different depths to aid in planning vehicle paths. The authors utilized a deep network to train and execute semantic segmentation of R.P. in the spatial-temporal domain. In addition, the authors proposed an inference model called temporal shifting memory (TSM) for online testing. This model is designed to avert data overlap in sequent semantic segmentation, an essential process for edge device applications.

In [17] a persistent challenge in autonomous driving is the accurate categorization of LiDAR data in an outside setting, known as semantic segmentation. The authors presented a pioneering approach called hybrid CNN-LSTM for semantic segmentation of LiDAR point clouds. The system has a unique neural network architecture and an effective method for handling point cloud characteristics. Building upon Polar Net's approach of representing point clouds as vectors with uniform magnitude, the 3D point clouds were transformed into pseudo-images. The scientists developed an innovative neural network structure that combines the features of several channels produced by convolutional NNs with extended short-term memory networks to improve the representation of small object qualities. The procedure entailed feeding the pseudo image into an LSTM network that relied on the spatial filling curve. Experiments performed on the Semantic KITTI dataset demonstrate that the approach outperforms current cutting-edge techniques in terms of accuracy for semantic segmentation. Provide a theoretical study explaining how a network with sparse point cloud features may effectively distinguish small details.

2.2. Path planning based on RL

ML includes three basic models that specify how observations are represented: supervised, unsupervised, and RL [18]. Supervised learning is the primary approach in ML. Supervised learning involves a learning algorithm that provides data in the form of example pairs (x, y), which are used to train the function f(x). Here, y represents the observed output value that needs to be learned for a given input value x. The phrase supervised learning derives from the concept that y-values supervise and guide the learning process on the correct responses to each input value. The use of alternative learning methods becomes necessary when information is unlabeled. Unsupervised learning is synonymous with feature learning. Unsupervised learning uses an inherent metric, such as distance, to evaluate the properties of data items. Unsupervised learning often involves identifying patterns in the data, such as clusters or subgroups [19].

RL is the latest style in ML and is distinguished from previous models by three main factors: the ability to learn through interaction and be used to solve sequential choice problems. RL acquires knowledge through iterative interaction, unlike supervised and unsupervised learning techniques, which know more holistically [20].

RL aims to determine policy and the best action in any environmental situation. The agent acquires information through interactions with surroundings and collects data by selecting actions based on the rewards they receive in their surroundings [21], as shown in Figure 2. Agents can select specific activities to obtain information; RL is a distinct type of active learning. Agents are like children who develop a particular ability through play and exploration. The level of subject autonomy is a critical aspect that attracts researchers [22]. A RL agent develops a set of actions to be performed in different environmental scenarios based on past experiences. This is done by selecting the procedure or hypothesis to be tested and refining its understanding of effective strategies. RL only requires an environment that produces feedback signals of the agent's activities, while supervised learning relies on pre-existing datasets with labeled instances to approximate a function [23].

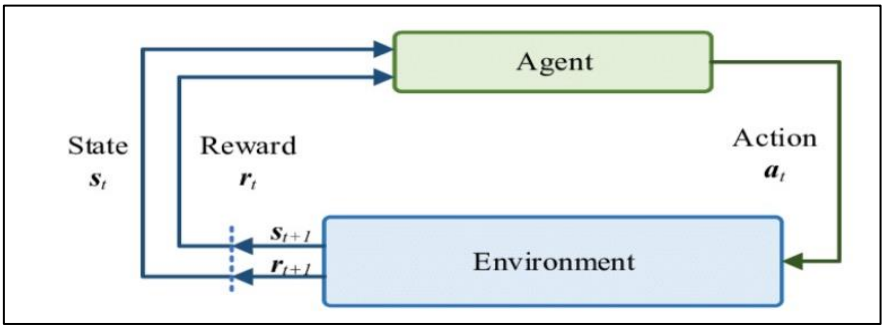

Figure 2. Shows the reinforcement learning architecture

RL can be used in a wider variety of situations than supervised learning due to its lower level of complexity. The basic ideas in RL are known as Markov decision process (MDP):

1. Agent: the learner and decision-maker [24].
2. Environment: the environment encompasses all entities with which the agent or agents interact.
3. State: An agent's epistemic state is data about its immediate surroundings at a specific moment. This data may consist of the agent's present location, the following objects, the space between the robot and its intended location point, and any past actions executed by the robot [25].
4. Action: if the agent is in a specific state, it selects an action based on its current behavioral rules (policy). The actions show discreteness in particular situations and continuity in others. Possible actions in a discrete action space contain movements like left, right, up, down, and more. The mobile robot can move from zero to 360 degrees in a continuous action space.
5. Policy: when the agent is in a state, it chooses an action to perform, guided by its existing behavior rules (policy). Policy dictates the behavior of the learning agent at a specific moment. A policy is a function that links perceived environmental situations to corresponding actions to be executed in those states [26]. It aligns with what would be referred to in psychology as a gathering of stimulus-response rules or relationships. At times, the policy could be a primary function or lookup table, while in other instances, it may require complex processing like a search technique. The policy is the crucial ingredient of a RL agent as it is solely responsible for defining behavior. Policies can be probabilistic [27].
6. Reward: a numerical rating that reflects the algorithm's efficacy concerning its environment. A reward signal establishes the objective in a RL scenario. In each step, the surrounding provides the RL agent with a singular numerical value known as a reward [28]. The reward signal determines which events are favorable or unfavorable for the agent. The reward given to the agent is contingent upon the activity taken by the agent and the present state of the agent's environment. The agent can only impact the reward signal by doing activities that directly affect the reward or indirectly by altering the condition of the environment [29].

The state-action-reward-state-action (SARSA) and Q-learning are popular and simple methods in RL [30]. SARSA is an on-policy temporal difference (TD) approach for policy control. SARSA evaluates the Q-value functions using TD. Updates to get the appropriate policy. Q-learning is a model-free approach, indicating that it doesn't rely on a model of the environment to guide the RL process. The agent acquires knowledge through practical encounters and formulates its prognostications on the environment. Q-learning is an off-policy technique that sets the optimal action based on the current state. Watkins proposed the Q-Learning method as a suitable approach for handling the trouble of path planning of mobile robots[31].

In [32]the IQL was explicitly built to enhance the obstacle avoidance performance of QL in dynamic scenarios by including the concept of distortion and an optimization mode. An analysis was conducted to compare the computational time, collision rate, traveled distance, and success rate of IQL with QL and DWA in 14 navigation scenarios with various layouts and dynamic obstacles.

In [33] the QAPF learning method, which integrates Q-learning with the artificial potential field, is proposed as a resolution for mobile robot path planning challenges. The QAPF learning algorithm consists of three operations: exploration, exploitation, and APF weighting. These are employed to overcome the limitations of the conventional Q-learning approach for path planning in both familiar and unfamiliar contexts.

In [34] the research introduced dynamic weighting coefficients based on Q-learning for DWA (DQDWA) using a Q-table that includes robot statuses, ambient circumstances, and weight coefficient actions. DQDWA may utilize the Q-table to dynamically choose the best pathways and weight coefficients that adjust well to changing environmental conditions. The efficacy of DQDWA was validated by empirical testing and thorough simulations.

In [35] the authors employed the accomplishment motivation model to modify the Q- Learning algorithm to generate different path variations. The Motivated Q-Learning (MQL) method was implemented in an environment consisting of three scenarios: one with no obstacles, one with uniformly distributed obstacles, and one with randomly placed obstacles.

In [36] the improved Q-learning for the mobile robot approach utilizes the following strategies to boost performance: The final path is more efficient and seamless due to the implementation of 8 optical self-adaptive action spaces, path extensions, and dynamic exploration factors.

2.3. Path planning based on DRL

RL and DL disciplines have recently converged, resulting in experimentation and learning from several engagements with the challenge. DRL has introduced novel methodologies and achievements through model-based, policy-based, transfer, hierarchical reinforcement, and multi-agent learning progress [37].

DRL intends to acquire the most advantageous behaviors that provide the highest rewards across various environmental conditions. This is achieved through engaging with intricate, multi-dimensional environments, conducting experiments with diverse actions, and assimilating knowledge from received feedback. One of the primary factors driving interest in this form of learning is its compatibility with contemporary computer systems, allowing for its effective implementation across various applications such as gaming, Atari, and robotics [38].

DRL offers solutions for trajectory planning in uncertain circumstances owing to technique developments. Unlike traditional trajectory planning methods that need significant effort to address complicated, high-dimensional problems, the recently proposed DRL enables a mobile robot to actively engage with its surroundings and independently acquire knowledge to choose the optimal course [39]. Mobile robots using DL techniques have demonstrated remarkable abilities to accurately complete tasks, maneuver complex environments, and avoid obstacles. Among the prominent DL techniques are deep learning network (DQN), double DQN, actor-critic (A2C, A3C), deep deterministic policy gradient (DDPG), double delay DDPG (TD3), soft actor-critic (SAC), and others. The strategies use a reward framework to mimic human learning behavior, and the system motivates the agent to engage in positive actions and imposes punishments for negative actions [40].

We will discuss key DRL concepts and comparisons, including model-free and model-based learning, off-policy and on-policy approaches, policy gradient theory, and active critic techniques. Next, we will analyze recent research that has used DRL techniques and how to combine them with other methods to solve path planning and dynamic obstacle avoidance problems.

3. BASIC CONCEPTS IN DRL

In this section, it is explained the basic concepts of DRL and the techniques based on these concepts.

3.1. Model-free learning vs model-based learning

RL is classifiable as model-based learning or model-free learning. Model-free learning is a core technique for RL where agents (Robots) evaluate actions and acquire knowledge of their consequences using techniques based on experience [41]. These algorithms repeatedly perform actions and adjust their policy (the strategy guiding their actions) to maximize rewards based on the observed outcomes. Model-free RL may be further categorized into techniques based on value, policy, and actor-critic. Value-based DRL techniques utilize TD learning and DNNs to estimate the function's value[36]. The environment model comprises the likelihood of state transitions and the expected reward. However, in actual scenarios, they may not be accessible for all potential states. Model-free RL techniques utilize the agent's experience to directly learn the most optimum value functions or policies without relying on a comprehensive model of the environment. This is achieved by approximating the ideal policy through a trial-and-error procedure. The quantity of agent samples of data regarding environment interaction needed for training model-based algorithms is lower than that required for model-free techniques. However, model-based algorithms still require model-free approaches to create the environment model [42].

Model-free RL approaches are beneficial for intricate issues that make constructing a sufficiently precise environment model difficult. Model-based learning depends on developing internal representations of the environment to optimize reward. Preferences are prioritized above action outcomes; the agent with a greedy approach will consistently attempt to do actions that provide the highest possible reward, regardless of potential consequences. For a model-based system to learn all of the transition probabilities, it must utilize dynamic programming methodologies to determine the chance of an agent changing states [43].

The system's model-based component uses a cross-entropy optimizer to change the model. This change aims to decrease the collision probability in the following step. It accomplishes this by forecasting the future condition based on the current condition and the activity performed. Each method, whether model-based or model-free, has its advantages and limitations. Model-free methods may exhibit reduced efficacy and require a larger dataset to attain satisfactory performance, although they are frequently easier to execute and facilitate expedited experiential learning. Model-based strategies exhibit reduced sensitivity to environmental changes and enhanced efficacy with less data but pose more application challenges [44]. AlphaZero method is a model-based approach, and Q-learning is model-free.

3.2. On-policy vs. off-policy

The process by which the behavior policy acquires knowledge is essential to developing techniques for RL. It focuses on creating a policy by analyzing actions and rewards. It chooses an activity to perform. On-policy learning involves updating the value of a desired action by consistently utilizing the original behavior of the function's policy that was used to select the action [45]. Off-policy refers to the situation in which learning occurs by storing the values of an action other than the one chosen by the behavior policy [46]. The SAC technique is a policy, and the A3C method is a policy.

3.3. Policy gradient theory

The value function is optimized using policy gradient (PG) over a parameterized family of policies. This Technique offers a minimum of two advantages. Initially, actions are selected from a well-defined parametric distribution [47]. Secondly, having less knowledge about the parameters of the parametric family, which has to be learned, arises from approximation policies. This leads to more efficient learning if one has prior information or intuition about the potential optimal policies, such as Gaussian distributions [48]. DDPG is based on PG theory.

3.4. Temporal-difference learning

TD learning combines dynamic programming principles (DP) with Monte Carlo. Like Monte Carlo techniques, TD procedures do not necessitate a model of the environment's dynamics to acquire knowledge from direct experience. Like DP, TD techniques iteratively refine their estimates by incorporating previously learned estimates without waiting for an outcome [49]. The relationship between TD, DP, and Monte Carlo approaches is a recurring subject in the context of RL. TD employs two distinct policy control techniques: SARSA, which is an on-policy method, and Q-learning, which is an off-policy method [50].

3.5. Actor-critic methods

utilize TD techniques to separate the policy from the value function through a unique memory structure [51]. The policy framework is commonly known as the actor because it dictates to the actor to be taken. The estimated value function is called the critic, as it simultaneously assesses the decisions the actor created.

Learning is fundamentally linked to policy: the critic must gain expertise and evaluate the policies the actor implements. The critique is presented as a type of TD error. According to Figure 3, this scalar signal is the critic's only output and propels all learning in the actor and critic [52].

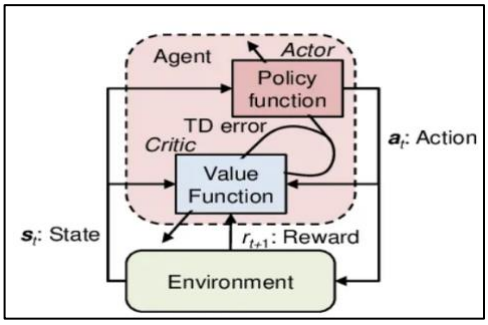

Figure 3. Shows the structure of actor-critic technique

The notion of reinforcement comparison approaches is naturally expanded to TD. Learning and RL by utilizing actor-critic methods. The critic often functions as a state-value function. After each decision, the critic evaluates the current condition to see if the outcome exceeded or fell short of expectations [53].

Actor-critic techniques optimize the policy and value functions using the benefits of both actors only (policy-function) and critic-only (value-function) techniques. In actor-critic approaches, the policy makes decisions depending on the present situation, while the critic analyses the actor's performance to approximate the function's value.

Subsequently, the parameterized policy is modified to enhance performance by including the value function and employing gradient ascent [54].

4. DRL TECHNIQUES

In this section, it is discussed the latest DRL techniques, highlights their challenges, and discusses ways to integrate them with other algorithms to solve path-planning problems in dynamic and static environments.

4.1. Deep Q-Learning (DQN)

DQN is an approach that mixes DNNs with Q-learning to determine the best action to take in a particular situation. The objective is to enable agents to learn about the optimal course of action in complex and multi-dimensional environments. It can handle environments with large state spaces by employing a neural network to approximate the Q function. The Q-function estimates the expected total reward for each possible action in a specific state. The network is updated repeatedly by mixing exploitation and exploration strategies throughout the episodes [55].

The DQN approach is extensively employed in path-planning applications due to its self-learning capability and adaptability to complex situations.

In [56] the Improved Dueling Deep Double Q Network algorithm (ID3QN) addresses the issues of overestimation and insufficient sample use in the classic DQN technique. It achieves this by utilizing an asymmetric neural network structure, optimizing the neural network structure, employing a double network to estimate action values, enhancing the action selection mechanism, implementing a priority experience replay mechanism, and redesigning the reward function.

In [57] the authors utilize the DQN and Artificial Potential Field (APF) algorithms to forecast the optimal Path for a mobility robot. The DQN is constructed and trained to achieve this aim. Subsequently, the APF shortest path method is incorporated into the DQN algorithm.

In [58] the AG-DQN method is designed to solve the Pathfinding problem of an AGV in an RMFS. It offers a quicker training procedure and reduces decision-making time compared to the A* technique. The AG-DQN technique utilizes a trained neural network that solely relies on the layout data of the current system to guide the AGV in completing a set of tasks assigned at random.

In [59] the agents are implemented by combining the deep Q networks approach, namely the D3QN and rainbow algorithms. These algorithms are used for obstacle avoidance and goal-oriented navigation tasks. The Rainbow DQN, because of its enhanced updates and improved estimates, achieved more goals and experienced fewer collisions during training than the D3QN agents.

In [60] the authors enhanced the DQN approach for Path planning for autonomous mobile robots. The reward function is enhanced by incorporating heading angle and distance errors. Additionally, a DHD (distance-heading angle-direction) reward function is devised by integrating the movement direction. This modification aims to enhance the algorithm's execution and prevent it from getting stuck in local optima. A weight-sampling learning approach is developed to grow the usage rate of training samples and accelerate the convergence speed of the algorithm.

4.2. Deep deterministic policy gradient (DDPG)

This methodology is a model-free off-policy approach developed explicitly to acquire knowledge about continuous activities. It integrates principles from Deterministic PG and DQN [61]. The system incorporates Experience Replay and slow-learning target networks from DQN. It is based on DPG and can operate in continuous action spaces [62].

In [63] Robotics involves the crucial challenge of maneuvering robots over expansive settings while evading moving impediments. A refined DDPG path planning approach incorporates sequential linear path planning (SLP) to address this issue. The authors aim to progress the reliability and effectiveness of standard DDPG approaches by including SLP to achieve a better balance between reliability and immediate performance. The system utilizes the Simultaneous Localization and Mapping (SLAM) algorithm to create a sequence of smaller objectives determined by a rapid computation of the robot's intended trajectory. Subsequently, The DDPG technique is utilized to provide these intermediate objectives for path planning while guaranteeing the avoidance of obstacles.

In [64] the authors employed a DRL-based technique known as Structure of Reconfigurable of DDPG (RS-DDPG) for robots. This method incorporates an event-triggered reconfigurable actor-critic network framework for motion policy, which dynamically adjusts its structure to mitigate the issue of the value of action overestimation. Subsequently, the temporal convergence of the policy motion may be improved by utilizing the action value that exhibits minimal divergence in valuation. A dynamic incentive system is developed for Flexible networks to address the absence of sample data.

In [65] the authors employed the DDPG technique for path planning mobile robots. A deep neural network structure may be constructed to improve the capabilities of robots' decision-making by using the DL Tensor Flow. Employs multi-sensing data collection by integrating image and LIDAR information to improve perceptive abilities. A meticulously crafted network model, a lightweight multimodal data-fusion network, has been established, which includes the idea of modalities separating learning. By integrating sensory data, robots enhance their understanding of their environment and improve their ability to make accurate decisions. Utilizing the artificial potential field technique for generating the reward function can lead to quicker convergence of the neural network and higher success rates in guiding mobile robots.

In [66] the authors employed the DDPG technique to accomplish the task of Path planning in a challenging continuous environment. Create a stochastic obstacle model for mobile sensors to replicate the complexity of target tracking situations and reduce mistakes by adjusting the parameters of the target network. Enhance the reward function to expedite the movement of the mobile sensor toward the goal location.

In [67] the DDPG technique is used with an LSTM network-based encoder to understand an indeterminate number of obstacles. Based on the LSTM network, the encoder utilizes the most recent environment data, which includes the prominent obstacles. It applies the secure processing guideline to produce a state vector with a defined length.

4.3. Twin delayed DDPG (TD3)

DDPG exhibits some instances of achieving exceptional performance, but it frequently demonstrates instability about hyperparameters and other tweaking forms. An example of a common failure situation in DDPG is when the learned Q-function overestimates Q-values excessively. This results in policy violation since it exploits the faults of the Q-function. The TD3 approach incorporates three crucial strategies to address this trouble: clipped double Q-learning, delayed policy updates, and target policy smoothing. TD3 is an effective method for DRL navigation [68].

In [69] to get the ultimate Q-value, the author enhanced the precision of the Q-value estimate and enhanced the capacity to learn; the authors propose a revised version of the TD3 method incorporating the dueling critic network architecture. This design separates and recombines the state value and action trait functions. Additionally, the authors include the dueling network architecture into the critic network to enhance the precision of the Q-value estimation. The findings indicate that the suggested model surpasses the old model because of its ability to design paths.

In [70] to address the low success rate and slow learning speed of the TD3 approach in the planning of mobile robot paths, researchers are examining an enhanced TD3 algorithm. To mitigate the effects of inaccuracies in value estimation, the Technique of prioritized experience replay is implemented, along with the development of dynamic delay updating algorithms. These methods reduce training time while enhancing the benefits and increasing the success rate. Currently, simulated trials are being employed to validate the algorithm's effectiveness for planning mobile robot paths.

In [71] the path planning method of mobile robots utilizes the Prioritized Experience Replay (PER) technique and Long Short Term Memory (LSTM) neural network. This approach effectively addresses problems related to slow convergence and incorrect perception of dynamic obstacles by employing the TD3 technique. This unique approach has been designated as PL-TD3. The authors use the Policy Evaluation with Repeated Updates (PER) approach to enhance the method’s convergence rate. Subsequently, the LSTM neural network was utilized to improve the dynamic obstacle detection technique. Based on the testing results, PL-TD3 outperforms TD3 in terms of both execution time and execution path length across all situations.

In [72] the authors suggested a method for designing lifting paths by employing DRL for hybrid action spaces. The network architecture was devised using the TD3 technique. To tackle the issue of limited rewards in long-distance path planning, a proposed solution involves creating a unique reward function and implementing hindsight experience replay. Real-time path planning is feasible in unfamiliar surroundings due to the ability to create an easy-to-follow path.

In [73] the authors proposed that the Advanced TD3 model can devise drone trajectories energy-efficiently at the edge level. The TD3 is the most sophisticated approach in PG RL, now considered state-of-the-art in this field. The TD3 model incorporates the drone's continuous action space while employing the frame stacking method. The authors expanded the range of observation for agents to achieve both fast and stable convergence. They also modified the TD3 model using Offline RL to decrease the training overhead for the RL model.

4.4. Asynchronous advantage actor-critic (A3C)

In 2016, DeepMind introduced A3Cs. PG and DQN became outdated due to their simplicity, resilience, efficiency, and capacity to provide superior outcomes in typical RL assignments. A3C consists of several autonomous agents, often networks, each possessing a distinct weight. These agents interact simultaneously with independent replicas of the environment. Consequently, they can allocate significantly less time to explore a more extensive range of state-action possibilities. A3C is an on-policy method, so utilizing an experience replay buffer is unnecessary. It exhibits greater resilience to hyperparameter adjustment than DDPG [74].

In [75] the authors suggested a three-step technique, detailed in the following order: A path planner that uses footprints to calculate cover and metrics for the path length for different Smorphi shapes. Second, the optimization of PPO and A3C methods. This creates energy-efficient and optimal configurations for Smorphi robots by maximizing rewards. Third, a Markov decision process (MDP) to represent and analyze the Smorphi

design space enables sequential decision-making. The proposed approach employs a validated technique using two separate environment maps. It subsequently evaluates the results by comparing them to the Pareto front solutions obtained by NSGA-II and the suboptimal random shapes.

In [76] the authors presented a technique for training neural controllers for differential-drive mobile robots operating in a congested environment to reach a given destination safely. The researchers devised a training pipeline that allows for the expansion of the process to many compute nodes. The authors showcased the ability to train and evaluate neural controllers efficiently on an actual robot in a dynamic setting by employing the asynchronous training methodology in A3C.

In [77] the authors suggested using a mean-A3C (M-A3C) method to find the robot's final motion in continuous state and action spaces without needing a reference gait. The authors utilized the M-A3C algorithm in a physical simulation environment to train several virtual robots independently and simultaneously with the help of various sub-agents. The trained model was used to regulate the robot's walking to decrease the need for frequent training sessions on the physical robot, accelerate the training process, and guarantee the proper implementation of the desired walking pattern. Ultimately, a bipedal robot is created to confirm the practicality of the suggested approach. Multiple studies indicate the proposed technique may reliably offer the biped robot uniform and seamless gait planning.

In [78] the Dec-POMDP model-based IL-A3C algorithm is designed to conquer the constraints of conventional centralized path planning techniques. Afterward, the IL-A3C performance evaluation is carried out by measuring metrics such as the mean path planning length, mean path planning time, mean likelihood of a collision, and mean planning success rate across several dimensions. The simulation outcome demonstrates that ILA3C has excellent performance in environments characterized by a sparse distribution of barriers, and it can be easily expanded to accommodate a team consisting of 128 robots. Comparatively, the centralized algorithms A3C and CBS are contrasted with IL-A3C, revealing that IL-A3C exhibits superior stability, scalability, and success rate compared to A3C and CBS. Growing IL-A3C into a large-scale robot team is a straightforward task.

In [79] to accelerate the learning process, the authors have suggested implementing a sophisticated double-layered multi-agent system that utilizes a two-dimensional grid to represent a state space. This system provides a hierarchical representation of a two-dimensional grid space and leverages actions based on the A3C technique. Both the top and lower levels included the state space. The top layer promptly evaluates the learning outcomes obtained from the bottom layer's use of A3C, leading to a decrease in the overall duration of learning. The efficacy of this approach was confirmed by experimentation with a virtual simulator for autonomous surface vehicles, and the time needed to attain a 90% success rate in meeting the aim decreased by 7.1% compared to the standard double-layered A3C approach. Through almost 20,000 learning sessions, the suggested approach surpassed the conventional double-layered A3C by obtaining a target achievement of 18.86% higher.

4.5. Soft actor-critic (SAC)

Using stochastic policy, the SAC methodology integrates DL techniques and merges the maximum entropy concept into an actor-critic network. The SAC technique excels in DRL techniques because of its exceptional exploration abilities and quick reaction to complex situations [80]. The SAC method stands out from other algorithms due to its superior sampling efficiency and robustness in dealing with slow convergence. The method learns from off-policy, which is the underlying cause. The primary characteristic of the change of the goal function in the context of SAC is that the objective is to optimize rewards and policy entropy. High entropy in policy facilitates exploration, mitigating the vulnerability to convergence. Consequently, this technique has demonstrated its effectiveness in path planning.

In [81] the authors employed a multi-agent actor-critic approach called SAC with Heuristic-Based Attention (SACHA). This method incorporates heuristic-based attention mechanisms for actors and critics, promoting agent collaboration. SACHA trains a neural network for each agent to focus on the shortened Path heuristic that guides several agents within its vicinity. SACHA enhances the current multi-agent actor-critic paradigm by incorporating a dedicated critic for all agents to estimate Q-values.

In [82] the authors developed a novel method called SAC-M, which combines the adaptive SAC with automated entropy techniques. These approaches enable the computerized adjustment of temperature settings, allowing the entropy to fluctuate between various states to regulate the extent of exploration.

In [83] to provide real-time optimum feedback management in the navigation task, we utilize a unique mixed auxiliary reward structure and sum-tree prioritized experience replay (SAC-SP). This approach treats the navigation job as a Markov Decision Process, encompassing static and movable obstacles. To enhance the efficiency of robust learning for AGVs, propose a unique approach incorporating mixed auxiliary incentives. Next, the AGVs can be effectively utilized by implementing the SAC-SP technique for time navigation using a mix of effective auxiliary reward structures. The proficient policy network can generate

real-time optimum feedback actions based on the placements of obstacles, the objective, and the states of the AGV.

In [84] the authors proposed a SAC Residual-like (R-SAC) method for agricultural settings, aiming to provide security for the avoidance of obstacles and Path-planning intelligence for robots. To address the time-consuming issue in the exploration phase of RL, the authors propose an offline expert experience pre-training Technique. This technique increases the effectiveness of training in RL. Additionally, the method enhances the reward system by including multi-step TD-error, effectively resolving training-related issues.

5. CONCLUSION

Mobile robots face significant challenges in achieving autonomous navigation, especially in uncertain environments. To scan its surroundings, determine its location, and plot a course toward a goal, the position of the intended destination is crucial in a navigation system because it is an input to the path-planning technique. The robot often requires multiple sensors. However, DRL methods solve navigation challenges without a pre-defined map by identifying the most efficient course of action. This article explores several methodologies to address the challenge of path planning in mobile robots by taking advantage of DNNs and RL. This group can provide a reliable answer. This review provides a comprehensive analysis of several methods and their specific applications. Although DL methods have exceptional capabilities, they also present distinct challenges. It has an enhanced ability to detect and understand subtle differences in data, which requires a large amount of computer processing and data. However, ongoing research has identified several strategies that may mitigate these challenges. Domain randomization techniques improve the quality of training data, while intrinsic incentives and reward shaping lead to higher reward concentration and overall performance.

LSTM-based RNNs have been used to study the time-dependent features of navigational data, increasing the effectiveness of DRL approaches. Due to their advantages, it is critical to carefully evaluate the use of these tactics when implementing DRL techniques in path-planning activities. With advances in DRL-based route planning, navigation efficiency through unfamiliar locations has been greatly improved. In navigation, DRL is essential for creating autonomous mobile robots that are intelligent and adaptable in real-world scenarios as we advance into the Fourth Industrial Revolution, which began with AI and robotics.

REFERENCES

[1] N. T. Lam, I. Howard, and L. Cui, "A literature review on path planning of polyhedrons with rolling contact," in *2019 4th International Conference on Control, Robotics and Cybernetics (CRC)*, 2019, pp. 145-151.

[2] S. Feng, B. Sebastian, and P. Ben-Tzvi, "A collision avoidance method based on deep reinforcement learning," *Robotics*, vol. 10, p. 73, 2021.

[3] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 14128-14147, 2022.

[4] M. Silva, "Introduction to the special issue on mobile service robotics and associated technologies," *Journal of Artificial Intelligence and Technology*, vol. 1, p. 145, 2021.

[5] H. Hakim, Z. Alhakeem, and S. Al-Darraji, "Goal location prediction based on deep learning using RGB-D camera," *Bulletin of Electrical Engineering and Informatics*, vol. 10, pp. 2811-2820, 2021.

[6] Y. Zhao, Y. Zhang, and S. Wang, "A review of mobile robot path planning based on deep reinforcement learning algorithm," in *Journal of Physics: Conference Series*, 2021, p. 012011.

[7] N. Sharma, R. Sharma, and N. Jindal, "Machine learning and deep learning applications vision," *Global Transitions Proceedings*, vol. 2, pp. 24-28, 2021.

[8] Y. Lin, J. McPhee, and N. L. Azad, "Comparison of deep reinforcement learning and model predictive control for adaptive cruise control," *IEEE Transactions on Intelligent Vehicles*, vol. 6, pp. 221-231, 2020.

[9] J. Parmar, S. Chouhan, V. Raychoudhury, and S. Rathore, "Open-world machine learning: applications, challenges, and opportunities," *ACM Computing Surveys*, vol. 55, pp. 1-37, 2023.

[10] K. Sharifani and M. Amini, "Machine learning and deep learning: A review of methods and applications," *World Information Technology and Engineering Journal*, vol. 10, pp. 3897-3904, 2023.

[11] M. M. Moein, A. Saradar, K. Rahmati, S. H. G. Mousavinejad, J. Bristow, V. Aramali, *et al.*, "Predictive models for concrete properties using machine learning and deep learning approaches: A review," *Journal of Building Engineering*, vol. 63, p. 105444, 2023.

[12] D. W. Jorgenson, M. L. Weitzman, Y. X. ZXhang, Y. M. Haxo, and Y. X. Mat, "Can neural networks predict stock market?(LON: MSMN Stock Forecast)," *AC Investment Research Journal*, vol. 220, 2023.

[13] B. Ramalingam, A. V. Le, Z. Lin, Z. Weng, R. E. Mohan, and S. Pookkuttath, "Optimal selective floor cleaning using deep learning algorithms and reconfigurable robot hTetro," *Scientific Reports*, vol. 12, p. 15938, 2022.

[14] H. Zhou, X. Yang, E. Zhang, J. Zhao, C. Ye, and Y. Wu, "Real-time Multiple Path Prediction and Planning for Autonomous Driving aided by FCN," in *2022 6th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, 2022, pp. 1-6.

[15] X. Hu, Y. Liu, B. Tang, J. Yan, and L. Chen, "Learning dynamic graph for overtaking strategy in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 11921-11933, 2023.

[16] G. Cheng and J. Y. Zheng, "Sequential semantic segmentation of road profiles for path and speed planning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 23869-23882, 2022.

[17] S. Wen, T. Wang, and S. Tao, "Hybrid CNN-LSTM architecture for LiDAR point clouds semantic segmentation," *IEEE Robotics and Automation Letters*, vol. 7, pp. 5811-5818, 2022.

[18] Y. Casali, N. Y. Aydin, and T. Comes, "Machine learning for spatial analyses in urban areas: a scoping review," *Sustainable cities and society*, vol. 85, p. 104050, 2022.
[19] G. T. Nwaila, S. E. Zhang, J. E. Bourdeau, H. E. Frimmel, and Y. Ghorbani, "Spatial interpolation using machine learning: from patterns and regularities to block models," *Natural Resources Research*, vol. 33, pp. 129-161, 2024.
[20] M. Kim, Y. Ham, C. Koo, and T. W. Kim, "Simulating travel paths of construction site workers via deep reinforcement learning considering their spatial cognition and wayfinding behavior," *Automation in Construction*, vol. 147, p. 104715, 2023.
[21] R. Gu, Z. Yang, and Y. Ji, "Machine learning for intelligent optical networks: A comprehensive survey," *Journal of Network and Computer Applications*, vol. 157, p. 102576, 2020.
[22] P. Liu, H. Qi, J. Liu, L. Feng, D. Li, and J. Guo, "Automated clash resolution for reinforcement steel design in precast concrete wall panels via generative adversarial network and reinforcement learning," *Advanced Engineering Informatics*, vol. 58, p. 102131, 2023.
[23] Y. Niu, X. Yan, Y. Wang, and Y. Niu, "Three-dimensional collaborative path planning for multiple UCAVs based on improved artificial ecosystem optimizer and reinforcement learning," *Knowledge-Based Systems*, vol. 276, p. 110782, 2023.
[24] M. Natarajan and A. Kolobov, *Planning with Markov decision processes: An AI perspective*: Springer Nature, 2022.
[25] L. Bramblett, S. Gao, and N. Bezzo, "Epistemic prediction and planning with implicit coordination for multi-robot teams in communication restricted environments," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5744-5750.
[26] A. Yala, P. G. Mikhael, C. Lehman, G. Lin, F. Strand, Y.-L. Wan, *et al.*, "Optimizing risk-based breast cancer screening policies with reinforcement learning," *Nature medicine*, vol. 28, pp. 136-143, 2022.
[27] R. F. Prudencio, M. R. Maximo, and E. L. Colombini, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
[28] Y. Septon, T. Huber, E. André, and O. Amir, "Integrating policy summaries with reward decomposition for explaining reinforcement learning agents," in *International Conference on Practical Applications of Agents and Multi-Agent Systems*, 2023, pp. 320-332.
[29] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Information Fusion*, vol. 85, pp. 1-22, 2022.
[30] A. Plaat, *Deep reinforcement learning* vol. 10: Springer, 2022.
[31] X. Huang and G. Li, "An Improved Q-Learning Algorithm for Path Planning," in *2023 IEEE International Conference on Sensors, Electronics and Computer Engineering (ICSECE)*, 2023, pp. 277-281.
[32] E. S. Low, P. Ong, and C. Y. Low, "A modified Q-learning path planning approach using distortion concept and optimization in dynamic environment for autonomous mobile robot," *Computers & Industrial Engineering*, vol. 181, p. 109338, 2023.
[33] U. Orozco-Rosas, K. Picos, J. J. Pantrigo, A. S. Montemayor, and A. Cuesta-Infante, "Mobile robot path planning using a QAPF learning algorithm for known and unknown environments," *IEEE Access*, vol. 10, pp. 84648-84663, 2022.
[34] M. Kobayashi and N. Motoi, "Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles," *IEEE Access*, vol. 10, pp. 17018-17029, 2022.
[35] H. Hidayat, A. Buono, K. Priandana, and S. Wahjuni, "Modified Q-Learning Algorithm for Mobile Robot Path Planning Variation using Motivation Model," *Journal of Robotics and Control (JRC)*, vol. 4, pp. 696-707, 2023.
[36] F. Qian, K. Du, H. Wang, T. Chen, X. Meng, S. Wang, *et al.*, "Path Planning Algorithm of Mobile Robot Based on Improved Q-learning Algorithm," in *2023 IEEE 6th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2023, pp. 133-136.
[37] L. Li, D. Wu, Y. Huang, and Z.-M. Yuan, "A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field," *Applied Ocean Research*, vol. 113, p. 102759, 2021.
[38] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, "Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, pp. 73-84, 2020.
[39] L. Chen, Z. Jiang, L. Cheng, A. C. Knoll, and M. Zhou, "Deep reinforcement learning based trajectory planning under uncertain constraints," *Frontiers in Neurorobotics*, vol. 16, p. 883562, 2022.
[40] Ó. Pérez-Gil, R. Barea, E. López-Guillén, L. M. Bergasa, C. Gómez-Huélamo, R. Gutiérrez, *et al.*, "Deep reinforcement learning based control for Autonomous Vehicles in CARLA," *Multimedia Tools and Applications*, vol. 81, pp. 3553-3576, 2022.
[41] C. Diehl, T. Sievernich, M. Krüger, F. Hoffmann, and T. Bertram, "Umbrella: Uncertainty-aware model-based offline reinforcement learning leveraging planning," *arXiv preprint arXiv:2111.11097*, 2021.
[42] A. Morris and F. Cushman, "Model-free RL or action sequences?," *Frontiers in Psychology*, vol. 10, p. 2892, 2019.
[43] W. Xu, "Design, Development, and Control of an Assistive Robotic Exoskeleton Glove Using Reinforcement Learning-Based Force Planning for Autonomous Grasping," 2023.
[44] R. Hashemi, S. Ali, N. H. Mahmood, and M. Latva-Aho, "Deep reinforcement learning for practical phase-shift optimization in RIS-aided MISO URLLC systems," *IEEE Internet of Things Journal*, vol. 10, pp. 8931-8943, 2022.
[45] L. He, N. Aouf, and B. Song, "Explainable Deep Reinforcement Learning for UAV autonomous path planning," *Aerospace science and technology*, vol. 118, p. 107052, 2021.
[46] L. Federici, A. Zavoli, and G. De Matteis, "Deep Reinforcement Learning for Robust Spacecraft Guidance and Control," Ph. D. Dissertation, Sapienza University of Rome, 2022.
[47] X. Li, H. Liu, J. Li, and Y. Li, "Deep deterministic policy gradient algorithm for crowd-evacuation path planning," *Computers & Industrial Engineering*, vol. 161, p. 107621, 2021.
[48] L. Yang, J. Bi, and H. Yuan, "Dynamic path planning for mobile robots with deep reinforcement learning," *IFAC-PapersOnLine*, vol. 55, pp. 19-24, 2022.
[49] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," *arXiv preprint arXiv:2203.04955*, 2022.
[50] M. Salimibeni, A. Mohammadi, P. Malekzadeh, and K. N. Plataniotis, "Multi-Agent Reinforcement Learning via Adaptive Kalman Temporal Difference and Successor Representation," *Sensors*, vol. 22, p. 1393, 2022.
[51] S. Gattu, "Autonomous Navigation and Obstacle Avoidance using Self-Guided and Self-Regularized Actor-Critic," in *Proceedings of the 8th International Conference on Robotics and Artificial Intelligence*, 2022, pp. 52-58.
[52] A. J. M. Muzahid, M. A. Rahim, S. A. Murad, S. F. Kamarulzaman, and M. A. Rahman, "Optimal safety planning and driving decision-making for multiple autonomous vehicles: A learning based approach," in *2021 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, 2021, pp. 1-6.

[53] X. Zhan, X. Zhu, and H. Xu, "Model-based offline planning with trajectory pruning," *arXiv preprint arXiv:2105.07351*, 2021.
[54] Q. Zhang, W. Pan, and V. Reppa, "Model-reference reinforcement learning for collision-free tracking control of autonomous surface vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 8770-8781, 2021.
[55] J. Zhang, C. Zhang, and W.-C. Chien, "Overview of deep reinforcement learning improvements and applications," *Journal of Internet Technology*, vol. 22, pp. 239-255, 2021.
[56] 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, p. 2048, 2023.
[57] A. Sivaranjani and B. Vinod, "Artificial Potential Field Incorporated Deep-Q-Network Algorithm for Mobile Robot Path Prediction," *Intelligent Automation & Soft Computing*, vol. 35, 2023.
[58] L. Luo, N. Zhao, Y. Zhu, and Y. Sun, "A* guiding DQN algorithm for automated guided vehicle pathfinding problem of robotic mobile fulfillment systems," *Computers & Industrial Engineering*, vol. 178, p. 109112, 2023.
[59] M. Quinones-Ramirez, J. Rios-Martinez, and V. Uc-Cetina, "Robot path planning using deep reinforcement learning," *arXiv preprint arXiv:2302.09120*, 2023.
[60] X. Xu, Y. Cao, and X. Liu, "Improved DQN Algorithm for Path Planning of Autonomous Mobile Robots," 2023.
[61] H. Gong, P. Wang, C. Ni, and N. Cheng, "Efficient path planning for mobile robot based on deep deterministic policy gradient," *Sensors*, vol. 22, p. 3579, 2022.
[62] Z. Wang, Y. Wei, F. R. Yu, and Z. Han, "Utility optimization for resource allocation in multi-access edge network slicing: A twin-actor deep deterministic policy gradient approach," *IEEE Transactions on Wireless Communications*, vol. 21, pp. 5842-5856, 2022.
[63] Y. Chen and L. Liang, "SLP-improved DDPG path-planning algorithm for mobile robot in large-scale dynamic environment," *Sensors*, vol. 23, p. 3521, 2023.
[64] H. Sun, C. Zhang, C. Hu, and J. Zhang, "Event-triggered reconfigurable reinforcement learning motion-planning approach for mobile robot in unknown dynamic environments," *Engineering Applications of Artificial Intelligence*, vol. 123, p. 106197, 2023.
[65] J. Tan, "A method to plan the path of a robot utilizing deep reinforcement learning and multi-sensory information fusion," *Applied Artificial Intelligence*, vol. 37, p. 2224996, 2023.
[66] K. Zhang, Y. Hu, D. Huang, and Z. Yin, "Target Tracking and Path Planning of Mobile Sensor Based on Deep Reinforcement Learning," in *2023 IEEE 12th Data Driven Control and Learning Systems Conference (DDCLS)*, 2023, pp. 190-195.
[67] X. Gao, L. Yan, Z. Li, G. Wang, and I.-M. Chen, "Improved deep deterministic policy gradient for dynamic obstacle avoidance of mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, pp. 3675-3682, 2023.
[68] Y. Zhang, C. Zhang, R. Fan, S. Huang, Y. Yang, and Q. Xu, "Twin delayed deep deterministic policy gradient-based deep reinforcement learning for energy management of fuel cell vehicle integrating durability information of powertrain," *Energy Conversion and Management*, vol. 274, p. 116454, 2022.
[69] H. Jiang, K.-W. Wan, H. Wang, and X. Jiang, "A Dueling Twin Delayed DDPG Architecture for mobile robot navigation," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 193-197.
[70] P. Li, Y. Wang, and Z. Gao, "Path planning of mobile robot based on improved td3 algorithm," in *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2022, pp. 715-720.
[71] Y. Tan, Y. Lin, T. Liu, and H. Min, "PL-TD3: A Dynamic Path Planning Algorithm of Mobile Robot," in *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2022, pp. 3040-3045.
[72] Z. Yin, K. Wang, and X. Ma, "A Real-time Smooth Lifting Path Planning for Tower Crane Based on TD3 with Discrete-Continuous Hybrid Action Space," in *Proceedings of the 14th International Conference on Computer Modeling and Simulation*, 2022, pp. 88-93.
[73] D. Hong, S. Lee, Y. H. Cho, D. Baek, J. Kim, and N. Chang, "Energy-efficient online path planning of multiple drones using reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 9725-9740, 2021.
[74] R. Singh, J. Ren, and X. Lin, "A review of deep reinforcement learning algorithms for mobile robot path planning," *Vehicles*, vol. 5, pp. 1423-1451, 2023.
[75] M. Kalimuthu, A. A. Hayat, T. Pathmakumar, M. Rajesh Elara, and K. L. Wood, "A deep reinforcement learning approach to optimal morphologies generation in reconfigurable tiling robots," *Mathematics*, vol. 11, p. 3893, 2023.
[76] M. Caruso, E. Regolin, F. J. Camerota Verdù, S. A. Russo, L. Bortolussi, and S. Seriani, "Robot Navigation in Crowded Environments: A Reinforcement Learning Approach," *Machines*, vol. 11, p. 268, 2023.
[77] J. Leng, S. Fan, J. Tang, H. Mou, J. Xue, and Q. Li, "M-A3C: a mean-asynchronous advantage actor-critic reinforcement learning method for real-time gait planning of biped robot," *IEEE Access*, vol. 10, pp. 76523-76536, 2022.
[78] G. Shen, Y. Cheng, Z. Tang, T. Qiu, and J. Li, "Research on multi-robot path planning based on deep reinforcement learning," in *Second International Conference on Electronic Information Engineering and Computer Communication (EIECC 2022)*, 2023, pp. 141-150.
[79] D. Lee, J. Kim, K. Cho, and Y. Sung, "Advanced double layered multi-agent Systems based on A3C in real-time path planning," *Electronics*, vol. 10, p. 2762, 2021.
[80] T. Zhao, M. Wang, Q. Zhao, X. Zheng, and H. Gao, "A path-planning method based on improved soft actor-critic algorithm for mobile robots," *Biomimetics*, vol. 8, p. 481, 2023.
[81] Q. Lin and H. Ma, "SACHA: Soft actor-critic with heuristic-based attention for partially observable multi-agent path finding," *IEEE Robotics and Automation Letters*, 2023.
[82] Y. Chen, F. Ying, X. Li, and H. Liu, "Deep Reinforcement Learning in Maximum Entropy Framework with Automatic Adjustment of Mixed Temperature Parameters for Path Planning," in *2023 7th International Conference on Robotics, Control and Automation (ICRCA)*, 2023, pp. 78-82.
[83] H. Guo, Z. Ren, J. Lai, Z. Wu, and S. Xie, "Optimal navigation for AGVs: A soft actor–critic-based reinforcement learning approach with composite auxiliary rewards," *Engineering Applications of Artificial Intelligence*, vol. 124, p. 106613, 2023.
[84] J. Yang, J. Ni, Y. Li, J. Wen, and D. Chen, "The intelligent path planning system of agricultural robot via reinforcement learning," *Sensors*, vol. 22, p. 4316, 2022.