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

**Maysam H.Qasim [*1],Dr. Salah Al-Darraji [2]**

**Abstract**

Dynamic path planning involves finding the most efficient path between the beginning and the destination in an unfamiliar and constantly changing environment while avoiding fixed and moving impediments. Using advanced sensors, mobile robots may traverse their environment without human intervention, ensuring safety and autonomy.

It is necessary to utilize more efficient algorithms to resolve the problem of inadequate robot performance in such environments and achieve intelligent path planning considering factors such as time, energy, and distance. Recently, reinforcement learning and deep neural networks techniques had been used recently to address these problems. By using a trial-and-error methodology to communicate with its surroundings, an artificial intelligence agent uses reinforcement learning to acquire an ideal behavioural approach predicated on reward signals from previous transactions. The reinforcement learning agent's learning process resembles the method used by humans and animals to learn. The fact that reinforcement learning may be used to different scientific and engineering domains is one of its most advantageous features. Reinforcement learning has shown to be an effective approach in recent years for managing difficult sequential decisions. It presents a fantastic chance to explore new technological horizons in areas where system models are non-existent or too complex, costly, or time-consuming to develop. This review article examines path planning strategies utilizing neural networks, such as deep reinforcement learning, the fundamental concepts of it as well as the components of a system that uses it. Including policy gradient, model-free learning, model- based learning, and actor-critic techniques.

**Keywords:** Path planning, Deep reinforcement learning approaches, Actor-critic, Mobile autonomous robots.

## I. Introduction

Autonomous mobile robots have become more and more necessary in recent years. These robots are used in many aspects of our everyday lives, such as cleaning, self-driving cars, military operations, and rescue missions. In the majority of these applications, the robot must maneuver through challenging and unfamiliar terrain without running into any impediments. To prevent accidents with both stationary and moving obstacles, these robots must devise a comprehensive path based on the existing environmental data. Subsequently, they must design a specific path to reach the checkpoints along the previously determined global Path, Utilizing data using sensors like LiDAR, RGBD, or RGB cameras. Analytical methods have traditionally been used to tackle the issue of path planning. However, these methods need enhancement for complex situations or emergencies, as they require precise placement in the environment and a detailed map for path planning [1, 2].

* pgs.maysam.qasm@uobasrah.edu.iq

Path planning is crucial for robots' ability to navigate on their own. Robotic path planning challenge involves setting the most effective path from the current location. Direct the robot to the designated target location in its working environment based on one or more optimization objectives, given that the site of the robot is already determined [3-5].

Reinforcement learning could be better suited for complex tasks, and deep learning needs to make –decision. Consequently, many researchers contemplated utilizing deep learning's aptitude for extracting information in combination with reinforcement learning's ability in decision-making for robot path planning.

In artificial intelligence, known as "deep reinforcement learning," intelligent systems are built, trained by interacting with their environments and evaluated in real-time. Deep reinforcement learning (DRL) approaches are frequently used in a variety of fields, such as robotics, machine translation, control systems, text generation, target identification, video prediction optimization, autonomous driving, text-based games, and more [6].

Artificial intelligence is a section of machine learning that replicates the functions of the brain in humans using artificial neural networks. Such techniques allow computers to explore potential outcomes that exceed human capacities [7]. This procedure relies heavily on intricate mathematical formulas. Furthermore, numerous formulas may be required for machine learning, which performs optimally with extensive datasets. If we had perfect foresight into the consequences of all our choices, deep reinforcement learning would be unnecessary [8]. We could develop an algorithm to determine the optimal decision for reaching a particular outcome. We require technologies such as deep reinforcement machine learning to assist us in addressing issues involving various variables in our intricate reality, as correct predictions are challenging.

The DRL showed an excellent capacity for learning and adaptability. The DRL technique serves as crucial for robotic path-planning. Figure (1) provides an overview of traditional Strategies for path-planning [9].

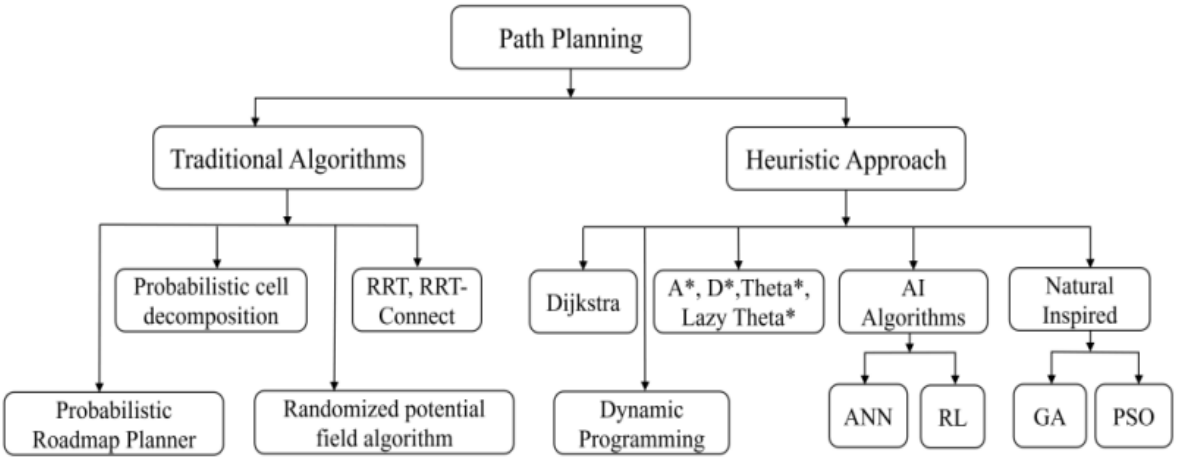

Figure 1.Overview of path planning techniques.

Ultimately, the residue of this work is organized as follows: A review of path planning approaches is discussed in Section II. Summarizing deep reinforcement learning concepts is the main objective of Section III. Deep Reinforcement learning strategies are covered in Section IV. Section V. contains the paper's conclusion.

## II. Path Planning Techniques

### A. Deep Learning (DL)

Artificial neural networks, support vector machines, and decision trees are examples of machine learning approaches employ different approaches to create a predictive model using data. The models aim to predict and collect data. Machine learning aims to estimate a function based on data mathematically [10].

Previously, when computers had limited processing speed, neural networks, which had several layers of interconnected neurons, could have been more effective in solving complex issues. Advanced deep-learning techniques and developing increasingly powerful computers have led to significant progress[11] .

Now, deep neural networks employ a wide variety of connections and several layers of neurons. The Deep learning and deep networks have significantly enhanced the precision of certain essential machine learning tasks, allowing for machine learning in intricate, high-dimensional issues like distinguishing between dogs and cats in high-resolution (mega-pixel) photographs. Deep learning allows for the rapid solution of intricate problems involving many variables. Furthermore, it has expanded machine learning to commonplace tasks, such as speech and facial recognition on mobile devices[12] .

The Neural networks are employed in deep learning in the area of machine learning to model and tackle complex problems. Neural networks 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. Deep learning's versatility allows it to be employed in several ways to tackle the path planning issue.

CNNs are utilized in algorithms designed to process pictures as input. Neural networks are employed in mimicking education and to deal with the Q-value problem in reinforcement learning when dealing with a complicated state of actions space [13].

In [14], the Deep SORT human tracking technique was utilized to monitor the movement of individuals. 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. There are 1200 pictures for each of the four classifications in the dataset: Stain, Foot Stain, Trash, and Human.

In [15], 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 capacity to generalize, even in the presence of 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 [16], a novel graph convolutional network model, TAM-GCN, was developed to address a significant limitation of the current graph convolutional network, which is 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 is devised by utilizing 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 [17], this work utilizes motion profiles (M.P.) and compact road profiles (R.P.) 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 [18], 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 consists of a unique neural network architecture combined with 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 neural networks 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.

B. Reinforcement Learning (RL)

Machine learning encompasses three fundamental paradigms that delineate how observations might be represented: supervised, unsupervised, and reinforcement learning[19]. Supervised learning is the primary and foundational approach in machine learning. Supervised learning involves the learning algorithm providing data in the form of (x, y) example pairs, which are used to train the function f(x). Here, y represents the observed output value that must be learned for a given value x input. The phrase supervised learning is derived from a concept that the y values serve as a means of overseeing the learning process and instructing it on the correct responses for each input value, utilizing alternative learning approaches becomes imperative when the information is devoid of labels. Unsupervised learning is synonymous with distinctive learning. Unsupervised learning utilizes an inherent metric, such as distance, to assess the characteristics of the data items. The task of unsupervised learning often involves the identification of data patterns; such as clusters or subgroups[20].

Reinforcement learning is the final pattern in the field of machine learning. Reinforcement learning distinguishes itself from previous models by three key factors: the ability to learn through interaction, the incorporation of incentives, and its use in solving sequential choice troubles. Reinforcement learning acquires knowledge by iterative interaction, unlike supervised and unsupervised learning techniques, which learn more holistically[21].

The dataset is generated in real-time. Reinforcement learning aims to identify the policy, a function that specifies the better action to do in any given environmental state. In reinforcement learning, an agent gains information through interactions with its surroundings. Reinforcement learning agents collect data by selecting actions according to the rewards they receive in their surroundings. Agents have the ability to select particular activities to gain information; reinforcement learning is a distinct kind of active learning. Our agents are like children who develop certain ability via play and exploration. The subject's level of independence is a key aspect that draws researchers[22].

The RL agent develops a set of actions to be executed in various environmental scenarios based on past experiences. It does this by choosing which action or hypothesis to test and refining its grasp of effective strategies. Reinforcement learning only necessitates an environment that produces feedback signals for the agent's activities, whereas supervised learning relies on pre-existing datasets with labeled cases to approximate a function[23].

Reinforcement learning could be used in a broader variety of situations compared to supervised learning due to its lower level of complexity. The figure (2) shows the architecture of reinforcement learning [24].

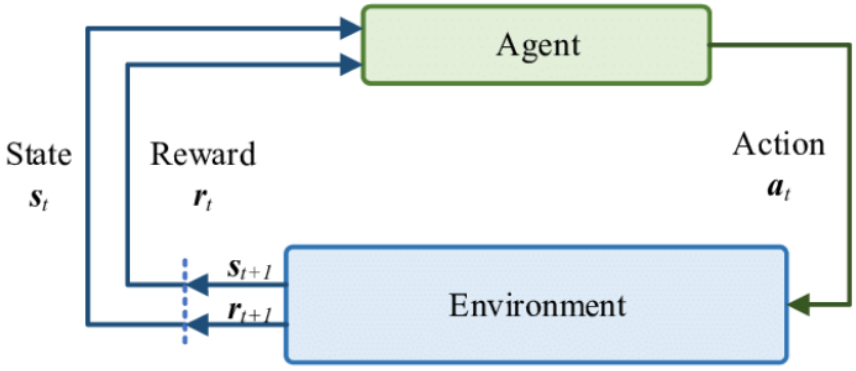


Figure 2 Reinforcement learning architecture

Basic ideas in Reinforcement Learning are known as Markov decision process (MDP):

- Agent: the learner and decision-maker [25].
- Environment: The environment encompasses all entities with which the agent or agents interact. Its purpose is to create the illusion of being an authentic case in the actual world for the agent. It is essential to showcase an agent's efficacy and ability to work effectively in a real-world application.

- State: Epistemic state is the whole data that an agent has about its immediate surroundings at a specific moment. The data may consist of the agent's present location, next objects, the space between the robot and its intended location point, and any past actions executed by the robot [26].
- 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 specific situations and continuous in others. Possible actions in a discrete action space contain movements like left, right, up, down, and more. The mobile robot in a continuous action space can move from zero to 360 degrees.
- Policy: When the agent is in a state, it chooses an action to perform, guide by its existing behavior rules (policy). Policy dictates the behavior of the learning agent at a specific moment. Essentially, a policy is a function that links perceived environmental situations to corresponding actions to be executed in those states [27]. 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 basic function or lookup table, while in other instances; it may require complex processing like a search technique. The policy is the key ingredient of a reinforcement learning agent as it is solely responsible for defining behavior. Policies can be probabilistic in nature [28].
- Reward: A numerical rating that reflects the algorithm's efficacy concerning its environment. A reward signal establishes the objective in a reinforcement learning scenario. Every each step, the surrounding provides the reinforcement learning agent with a singular numerical value known as a reward[29]. The reward signal determines which events are considered favorable or unfavorable for the agent. In a biological system, rewards might be likened to the sensations of pleasure or pain. The features are the primary and distinctive characteristics of the problem encountered by 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 is unable to modify the procedure. 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 [30].

The State-action-reward-state-action (SARSA) and Q-learning are popular and simple methods in reinforcement learning [31]

SARSA is an on-policy temporal difference approach used for policy control. SARSA evaluates the Q-value functions using T.D. 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 reinforcement learning process. The agent acquires knowledge through practical encounters and formulates its prognostications on the environment. Q-learning is an off-policy technique that set 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 [32].

In [33], the IQL was explicitly built to enhance the obstacle avoidance performance of Q.L. 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 Q.L., DWA, and I.Z. in 14 different navigation scenarios with different layouts and types of dynamic obstacles.

In [34], 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 [35], 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 can adjust well to changing environmental conditions. The efficacy of DQDWA was validated by empirical testing and thorough simulations.

In [36], 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 [37], 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.

C. Deep Reinforcement Learning (DRL)

Recently, reinforcement learning and deep learning disciplines have converged, resulting in the development of novel procedure that can effectively solve complex troubles by iteratively adjusting their behaviors based on feedback. By employing an iterative process of

experimentation and learning from several engagements with the challenge. Deep reinforcement learning has introduced novel methodologies and achievements through model-based, policy-based, transfer, hierarchical reinforcement, and multi-agent learning progress [38].

Deep Reinforcement Learning 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 a range of applications such as gaming, Atari, and robotics [39].

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 Deep Reinforcement Learning (DRL) enables a mobile robot to actively engage with its surroundings and independently acquire knowledge to choose the optimal course [40]. Mobile robots with DRL techniques have demonstrated remarkable abilities in accurately accomplishing tasks, maneuvering complex environments, and evading obstacles. Notable DRL techniques, like Deep Q-learning Network (DQN), Double-DQN, actor-critic (A2C, A3C), Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Soft Actor-Critic (SAC), among others. The strategy utilizes an action-reward framework to mimic human learning behavior. The system incentivizes the agent to engage in positive acts and imposes penalties for negative ones [41].

We will examine key concepts in DRL, including model-free and model-based learning, off-policy and on-policy approaches, policy gradient theory, and actor-critic techniques. Next, we will analyze recent research that employed prominent deep reinforcement learning techniques to address path design and dynamic avoiding obstacles.

## III. Concepts in Deep Reinforcement Learning

### A. Model-free learning vs model-based learning

Reinforcement learning is classifiable as model-based learning or model-free learning. Model-free learning is a core technique for reinforcement learning where agents ( Robot) evaluate actions and acquire knowledge of their consequences using techniques based

on experience[42]. These algorithms repeatedly perform actions and adjust their policy (the strategy guiding their actions) to maximize rewards based on the observed outcomes.

Model-free reinforcement learning may be further categorized into techniques based on value, policy, and actor-critic. The value-based DRL techniques utilizes temporal difference (T.D.) learning and deep neural network to estimate the function's value [6, 37]. 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 reinforcement learning (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 the utilize of model-free approaches in order to create the environment model [43].

Model-free reinforcement learning (RL) approaches are beneficial for intricate issues in which constructing a sufficiently precise environment model is difficult. Model-based learning depends on the development of 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. In order 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 [44].

The system's model-based component uses a cross-entropy optimizer to change the model. This change aims to decrease the probability of a collision 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 [45].

B. On policy vs. off policy

The process by which the behavior policy acquires knowledge is a basic aspect of the development of techniques for reinforcement learning. Reinforcement learning focuses on acquiring a policy by analyzing actions and rewards. It chooses an activity to perform. On-policy learning involves updating the value of a chosen action by consistently utilizing the original behavior of function's policy that was used to select the action [46]. Off-policy refers to the situation when learning occurs by storing the values of an action other than the one chosen by the behavior policy [47].

C. Policy Gradient Theory

The value function is optimized using policy gradient (P.G.) over a parameterized family of policies. This Technique offers a minimum of two advantages. Initially, actions are selected from a well-defined parametric distribution [48]. Secondly, having less knowledge about the parameters of the parametric family that 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 [49].

D. Temporal-Difference Learning

Temporal-difference (T.D.) learning combines dynamic programming principles (D.P.) with Monte Carlo. Like Monte Carlo techniques, T.D. procedures do not necessitate a model of the environment's dynamics to acquire knowledge from direct experience. Like dynamic programming (D.P.), temporal difference (T.D.) techniques iteratively refine their estimates by incorporating previously learned estimates without waiting for an outcome[50]. The relationship between Temporal Difference (T.D.), Dynamic Programming (D.P.), and Monte Carlo approaches is a recurring subject in the context of reinforcement learning. T.D. employs two distinct policy control techniques: State-action-reward-state-action (SARSA), which is an on-policy method, and Q-learning, which is an off-policy method [51].

E. Actor-critic methods

utilize temporal difference (T.D.) techniques to separate the policy from the value function through the use of a unique memory structure[52]. Policy framework is commonly known as the actor because it dictates to actor to be taken. The estimated value function is referred to as the critic as it assesses the decisions created by the actor simultaneously. Learning is fundamentally linked to policy: the critic must gain expertise and assess the policies being implemented by the actor. The critique is presented as a type of T.D. errors .According to Figure (3), this scalar signal is the critic's only output and propels all learning in the actor and critic [53, 54].

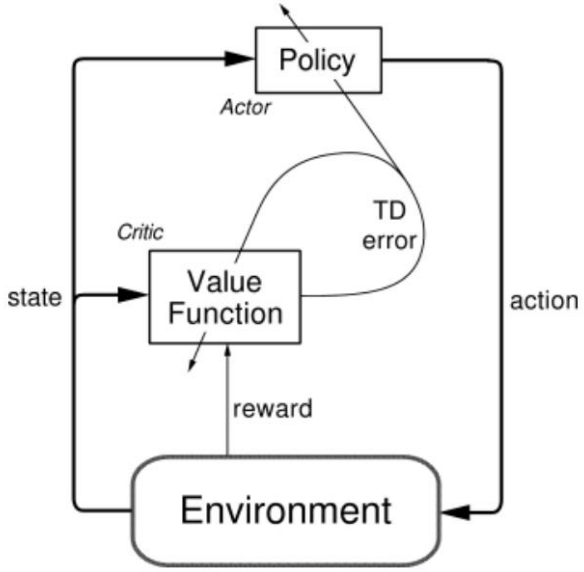


Figure 3.Structure of actor-critic

The notion of reinforcement comparison approaches is naturally expanded to T.D. learning and reinforcement learning 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 has exceeded or fallen short of expectations [55].

Actor-critic techniques optimize the policy and value functions using the benefits of both actors-only (policy-based) and critic-only (value-based) technique. In actor-critic approaches, the policy is responsible for making decisions depending on the present situation, while the critic analyses the actor's performance. An approximation of the function’s value.

Subsequently, the parameterized policy is modified to enhance performance by including the value function and employing gradient ascent [56].

## IV. Deep Reinforcement Learning Techniques

### A. Deep Q-Learning (DQN):

Deep Q-Learning is an approach that mixes deep neural networks with Q-Learning, a strategy to determine the best action to take in a certain situation[57]. The objective is to enable agents to learn about the optimal course of action in complex and multi-dimensional environments. Deep Q-learning 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 [58].

The deep Q-network (DQN) approach is extensively employed in path-planning applications due to its self-learning capability and adaptability to complex situation.

In [58], the Improved Dueling Deep Double Q Network algorithm (ID3QN) addresses the issues of overestimation and insufficient sample use in the classic Deep Q Network (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 [59], 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 with the objective of obtaining the aim. Subsequently, the APF shortest path method is incorporated into the DQN algorithm.

In [60], 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 in order to guide the AGV in completing a set of tasks assigned at random.

In [61], the agents are implemented utilizing a combination of the deep Q networks approach, namely the D3QN and rainbow algorithms. These algorithms are utilized for both obstacle avoidance and goal-oriented navigation tasks. The Rainbow DQN, because of its

enhanced updates and improved estimates, was able to achieve a more significant number of goals and experience fewer collisions during training compared to the D3QN agents.

In [62], 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.

B. Deep Deterministic Policy Gradient (DDPG):

This methodology is a model-free off-policy approach developed explicitly for acquiring knowledge about continuous activities. It integrates principles from DPG (Deterministic Policy Gradient) and DQN (Deep Q-Network) [63]. The system incorporates Experience Replay and slow-learning target networks from DQN. It is based on DPG and can operate in continuous action spaces[64].

In [65], Robotics involves the crucial challenge of maneuvering robots over expansive settings while evading moving impediments. A refined deep deterministic policy gradient (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 Deep Deterministic Policy Gradient (DDPG) technique is utilized to provide these intermediate objectives for path planning while guaranteeing avoidance of obstacles.

In [66], the authors employed a DRL-based technique known as Structure of Reconfigurable of Deep Deterministic Policy Gradient (RS-DDPG) for robots. This method incorporates an event-triggered reconfigurable actor-critic network framework for motion policy, which dynamically adjusts it’s the structure of network to mitigate issue the value of action overestimation. Subsequently, the temporal convergence the policy motion may be improved by utilizing the actions value that exhibit minimal divergence in valuation. A dynamic incentive system is developed for Flexible networks to address the absence of sample data.

In [67], the author employs the DDPG technique for path planning mobile robot. A deep neural network structure may be constructed to improve the capabilities s of robots' decision-making by employing the Deep Learning 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 [68], 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 [69], 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.

C. Twin Delayed DDPG (TD3):

DDPG exhibits some instances of achieving exceptional performance, but it frequently demonstrates instability about hyper parameters 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. Twin Delayed DDPG (TD3) approach incorporates three crucial strategies to address this trouble [70].

- Clipped Double-Q Learning.
- "Delayed" Policy Updates.
- Target Policy Smoothing.

(TD3) is an effective method for DRL navigation. In [71] , To get the ultimate Q-value, enhance the precision of the Q-value estimate, and enhance 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 [72] , 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. In order 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 the training time while enhancing the benefits and increasing the success rate in training. Currently, simulated trials are being employed to validate the algorithm's effectiveness for planning mobile robot paths.

In [73], 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 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 algorithm'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 [74] , the authors suggested a method for designing lifting paths by employing deep reinforcement learning for hybrid action spaces. The network architecture was devised using the TD3 technique. In order to tackle the issue of limited rewards in long-distance path planning, a proposed solution involves the creation of a unique reward function and implementing hindsight experience replay. Real-time path planning is feasible in unfamiliar surroundings due to the ability to create a path that is easy to follow.

In [75] The authors proposed that the Advanced TD3 model can devise drone trajectories energy-efficiently at the edge level. The TD3 is the best sophisticated approach in policy gradient reinforcement learning, 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.

D. Asynchronous Advantage Actor-Critic (A3C)

In 2016, DeepMind introduced A3Cs. Policy gradients 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 [44]. A3C is an on-policy method, so utilizing an experience replay buffer is unnecessary. It exhibits greater resilience to hyper parameter adjustment than DDPG [76].

In [77], The authors suggested a three-step technique, detailed in the following order: A path planner that use footprints to calculate cover and metrics for the length of the path for different Smorphi shapes. (2) The optimization of (PPO) and (A3C) methods. This creates energy-efficient and optimal configurations for Smorphi robots by maximizing rewards. (3) Utilizing a Markov decision process (MDP) to represent and analyze the Smorphi design space, enabling sequential decision-making. The proposed approach employs a validated technique, utilizing 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 [78], 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 [79], is an ongoing process where a robot communicates with its surroundings. The authors suggested using a mean-asynchronous advantage actor-critic (M-A3C) reinforcement learning method to find the robot's final motion in continuous state and action spaces without the need for a reference gait. The authors utilized the M-A3C algorithm in a physical simulation environment to independently train several virtual robots simultaneously with the help of various sub-agents. The trained model was utilized to regulate the robot's walking in order 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 suggest that the proposed technique may reliably offer uniform and seamless gait planning for the biped robot.

In [65], 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 [80], 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 (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. 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.

E. Soft actor-critic (SAC):

The SAC methodology integrates deep learning techniques and merges the maximum entropy concept into an actor-critic network over the use of stochastic policy. The SAC technique excels in deep reinforcement learning techniques because of its exceptional exploration abilities and quick reaction to complex situations [81]. The soft Actor-Critic method stands out from other algorithms due to its superior sampling efficiency and robustness in dealing with slow convergence. The algorithm 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 both rewards and policy entropy. High entropy in policy facilitates exploration, mitigating the vulnerability to convergence. Consequently, this Technique demonstrated its effectiveness in path planning.

In [82], the authors employed a multi-agent actor-critic approach called Soft Actor-Critic (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[83], the authors developed a novel method called SAC-M, which is a combination of the adaptive soft actor-critic (ASAC) and soft actor-critic with automated entropy (SAC-A) algorithms. These approaches enable the automated adjustment of temperature settings, allowing the entropy to fluctuate between various states to regulate the extent of exploration.

In[81] , the authors improved the path planning algorithm using the soft actor-critic methodology. They achieved this by enhancing the reward function, allowing mobile robots to navigate obstacles and reach their destination point quickly. This algorithm also utilizes state dynamic normalization and priority replay buffer methods.

In [84], 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, effectively utilize the AGVs by implementing the SAC-SP technique to 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 [85], the authors proposed a Soft Actor-Critic Residual-like (R-SAC) method for agricultural settings, aiming to provide secure for avoidance obstacle and Path planning intelligent for robots. To address the time-consuming issue in the exploration phase of reinforcement learning, the authors propose an offline expert experience pre-training Technique. This technique increases the effectiveness of training in reinforcement learning. Additionally, the method enhances the reward system by including multi-step TD-error, effectively resolving training-related issues.

## V. Conclusions

Mobile robots have significant challenges in achieving autonomous navigation, especially in uncertain environments. In order to survey its surroundings, ascertain its position, and devise a path toward the target, the intended destination position is crucial in the navigation system as it serves as an input for the path planning technique. A robot often requires many sensors. However, deep reinforcement learning approaches solve the challenges of navigating without a predefined map by identifying the most effective course of action. This article explores several methodologies for addressing the path planning challenge in mobile robotics by utilizing deep neural networks. The collection of reinforcement learning and deep neural networks can provide a dependable answer. This review provides a comprehensive analysis of several approaches and their respective applications.

Although deep learning approaches possess exceptional capabilities, they also provide distinct challenges. They have an enhanced ability to detect and understand little differences in the data, which requires a significant amount of computer processing and a large number of data. However, ongoing research has identified many strategies that might ease these challenges. Domain randomization techniques improve the quality of the data's training, while intrinsic incentives and reward shaping lead to a higher concentration of rewards and overall performance.

LSTM-based RNN have been used to study the dependent on time features of navigational data, resulting in enhanced effectiveness for DRL approaches. It is crucial to thoroughly evaluate the use of these tactics when implementing DRL techniques in path-planning activities due to the advantages they offer. Thanks to developments in (DRL)-based planning of paths, the efficiency of navigating across unfamiliar locations has greatly improved. In navigation the Deep reinforcement learning is essential for creating smart and adaptable mobile autonomous robots in real-world scenarios as we advance in the fourth industrial revolution, which began with artificial intelligence and robotics.

**References: -**


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

[2] 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.

[3] 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.

[4] 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.

[5] A. Shareef and S. Al-Darraji, "Grasshopper optimization algorithm based path planning for autonomous mobile robot," *Bulletin of Electrical Engineering and Informatics*, vol. 11, pp. 3551-3561, 2022.

[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-a 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] 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.

[10] 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.

[11] 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.

[12] 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.

[13] D. W. Jorgenson, M. L. Weitzman, Y. X. ZXhang, Y. M. Haxo, and Y. X. Mat, "Can Neural Networks Predict Stock Market?," *AC Investment Research Journal*, vol. 220, 2023.

[14] 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.

[15] 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.

[16] 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*, 2023.

[17] 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.
[18] 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.
[19] 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.
[20] 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.
[21] 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.
[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] 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.
[25] M. Natarajan and A. Kolobov, *Planning with Markov decision processes: An AI perspective*: Springer Nature, 2022.
[26] L. Bramblett, S. Gao, and N. Bezzo, "Epistemic Prediction and Planning with Implicit Coordination for Multi-Robot Teams in Communication Restricted Environments," *arXiv preprint arXiv:2302.10393*, 2023.
[27] 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.
[28] 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.
[29] 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.
[30] P. Ladosz, L. Weng, M. Kim, and H. Oh, "Exploration in deep reinforcement learning: A survey," *Information Fusion*, vol. 85, pp. 1-22, 2022.
[31] A. Plaat, *Deep reinforcement learning* vol. 10: Springer, 2022.
[32] 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.
[33] 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.

[34] 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.
[35] M. Kobayashi, H. Zushii, T. Nakamura, and N. Motoi, "Local Path Planning: Dynamic Window Approach with Q-learning Considering Congestion Environments for Mobile Robot," *IEEE Access,* 2023.
[36] 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.
[37] 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.
[38] 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.
[39] 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.
[40] 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.
[41] Ó. Pérez-Gil, R. Barea, E. López-Guillén, L. M. Bergasa, C. Gomez-Huelamo, R. Gutiérrez, *et al.*, "Deep reinforcement learning based control for Autonomous Vehicles in CARLA," *Multimedia Tools and Applications,* vol. 81, pp. 3553-3576, 2022.
[42] 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.
[43] A. Morris and F. Cushman, "Model-free RL or action sequences?," *Frontiers in Psychology,* vol. 10, p. 2892, 2019.
[44] W. Xu, "Design, Development, and Control of an Assistive Robotic Exoskeleton Glove Using Reinforcement Learning-Based Force Planning for Autonomous Grasping," Virginia Tech, 2023.
[45] 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.
[46] 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.
[47] 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.
[48] 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.
[49] 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.
[50] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," *arXiv preprint arXiv:2203.04955,* 2022.

[51] 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.

[52] 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.

[53] 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.

[54] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*: MIT press, 2018.

[55] X. Zhan, X. Zhu, and H. Xu, "Model-based offline planning with trajectory pruning," *arXiv preprint arXiv:2105.07351*, 2021.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] M. Quinones-Ramirez, J. Rios-Martinez, and V. Uc-Cetina, "Robot path planning using deep reinforcement learning," *arXiv preprint arXiv:2302.09120*, 2023.

[62] X. Xu, Y. Cao, and X. Liu, "Improved DQN Algorithm for Path Planning of Autonomous Mobile Robots," 2023.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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.

[68] 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.
[69] 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*, 2023.
[70] 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.
[71] 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.
[72] 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.
[73] 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.
[74] 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.
[75] 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.
[76] 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.
[77] 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.
[78] 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.
[79] 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.
[80] 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.
[81] 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.
[82] 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.
[83] 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.

[84] 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.

[85] 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.