Hindawi
Journal of Robotics
Volume 2024, Article ID 9819037, 1 page
https://doi.org/10.1155/2024/9819037

Retraction

Retracted: Reinforcement Learning-Based Continuous Action Space Path Planning Method for Mobile Robots

Journal of Robotics

Received 23 January 2024; Accepted 23 January 2024; Published 24 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] W. Zhang and G. Wang, "Reinforcement Learning-Based Continuous Action Space Path Planning Method for Mobile Robots," *Journal of Robotics*, vol. 2022, Article ID 9069283, 9 pages, 2022.

Hindawi
Journal of Robotics
Volume 2022, Article ID 9069283, 9 pages
https://doi.org/10.1155/2022/9069283

Research Article

Reinforcement Learning-Based Continuous Action Space Path Planning Method for Mobile Robots

Weimin Zhang[1] **and Guoyong Wang**[2]

[1]*School of Electrical Engineering and Automation, Luoyang Institute of Science and Technology, Luoyang 471023, Henan, China*
[2]*School of Computer and Information Engineering, Luoyang Institute of Science and Technology, Luoyang 471023, Henan, China*

Correspondence should be addressed to Weimin Zhang; zwm@lit.edu.cn and Guoyong Wang; wgy@lit.edu.cn

Received 21 September 2022; Revised 30 September 2022; Accepted 5 October 2022; Published 15 October 2022

Academic Editor: Shahid Hussain

A reinforcement learning-based continuous action space path planning method for mobile robots is proposed in this article. First, the kinematic model of the mobile robot is analyzed, and on this basis, the optimal state space is constructed according to the minimum depth of the field value in the depth image to characterize the distance between the robot and the obstacle. Then, by setting the reward function of the mobile robot based on the artificial potential field method, the information of the robot's distance from obstacles is continuous, and a new reinforcement learning training process is proposed. Finally, by introducing a DDPG algorithm, the path planning of a mobile robot in an unknown environment is described as a Markov decision process, and the optimal planning of the mobile robot's continuous action space path is realized with a high success rate. The results show that compared with other three comparison methods, the final success rates of the proposed method are the highest, which are 97.2%, 99.1%, 98.4%, and 98.6%, respectively.

1. Introduction

The mobile robot can sense the environment information and its own state information through the sensor, so as to realize the autonomous movement in the obstacle environment and complete some operations [1–3]. A mobile robot plays an important role in various fields of life and can replace people to complete some special work [4, 5]. However, with the expansion of mobile robot application scenarios, its working environment is becoming quite complex. In order to achieve better adaptability to the environment, the autonomous learning ability of robots needs to be improved [6–8].

In order to successfully complete various tasks, mobile robots must avoid colliding with obstacles in the environment and complete the navigation from one point to another. The goal of mobile robot path planning is to find a collision-free optimal path from the starting position to the target position in the environment. Path planning can be divided into global path planning and local path planning according to the amount of environmental prior information. Global path planning means that environment information is all known relative to the mobile robot. The mobile robot models the environment and then finds an optimal path. The obstacle information in global path planning is known and fixed, so it belongs to static path planning and is also called offline planning. Local path planning requires mobile robots to obtain information from the environment constantly through sensors and real-time path planning. Therefore, local path planning belongs to dynamic planning and is also called online planning [9–11]. At present, research on navigation strategies using deep reinforcement learning has attracted considerable attention. Reinforcement learning technology can learn appropriate strategies from the state of the environment. In the process of interaction between the agent and the external environment, the agent obtains the surrounding environmental information through repeated trial-and-error learning, so as to continuously optimize the agent's strategy [12, 13]. Applying reinforcement learning to mobile robot path planning does not need to build maps. Through deep reinforcement learning networks, mature navigation strategies are trained so that mobile robots can navigate in unknown environments [14].

The traditional spatial path planning method of mobile robots is discontinuous and has a low success rate. A continuous action spatial path planning method for mobile robots based on reinforcement learning is proposed. Compared with the traditional service robot route recommendation method, the innovations of the proposed method are as follows:

(1) The artificial potential field is introduced to make the target point and obstacle produce attraction and repulsion force to the mobile robot, respectively, and the control performance of the robot is improved under the superposition of the two forces.

(2) By introducing the DDPG algorithm, the ability of the algorithm to deal with higher dimensional observation space and the stability of the algorithm are improved.

The remainder of this paper is arranged as follows: The second section is related work, introducing several representative research results. The third section introduces the path-planning algorithm based on DDPG. In section 4, experiments are designed to verify the performance of the proposed model. The fifth section is the conclusion.

2. Related Research Studies

At present, some scholars have conducted in-depth research on the path planning of mobile robots and achieved some results. The authors in [15] constructed a global planner for finding the shortest safe path by combining the evolutionary algorithm, mutated cuckoo optimization algorithm, and genetic algorithm. On this basis, a mobile robot navigation system for path optimization is constructed based on a sensor source, map format, and basic controller. However, when the number of nodes is too large, the algorithm will consume a lot of time and memory, resulting in low efficiency. In [16], aiming at the mutual restriction between the execution speed and path quality of mobile robots, a new path planning strategy was proposed by ignoring all static obstacles outside the robot and the destination, focusing on the processing of key areas around obstacles and target points to improve the execution speed, and finding a linear shortcut between any two points in the path to improve the path quality. However, this method cannot achieve good results in a dynamic environment. The authors in [17] proposed a centralized decoupling algorithm for solving the multirobot path-planning problem in grid graphs, which can be set automatically as needed. On the one hand, a group of robots can be moved from their initial positions to the target positions; on the other hand, they can adapt to the target configuration adjustment through continuous replanning. However, this method cannot effectively avoid the local optimization in path planning. The authors in [18] proposed a trajectory-planning algorithm for parallel parking mobile robots based on polynomial parameterization and genetic algorithm optimization by defining a new law of motion to avoid obstacles on the road without interruption and guide the vehicle from the initial posture close to the parking space to the final posture in the parking space in a stable way. However, this method will still produce a large amount of calculation when the map accuracy is large.

In [19], the authors designed an adaptive firefly algorithm for mobile robot path planning by optimizing the adaptive parameters of the firefly algorithm to solve the problem that the traditional firefly algorithm is easy to fall into the local optimal solution. However, this method does not fundamentally improve the success rate and path optimization performance of the path optimization algorithm. The authors in [20] aimed at the problem that the traditional graph neural network depends on the message aggregation mechanism and is not conducive to the priority processing of important information. Based on a mechanism similar to a key query, the relative importance of features in messages received from various adjacent robots is determined and the path optimization of autonomous mobile robots is realized. However, this method will incur a large time cost in the process of judging the relative importance of messages, making the convergence speed of the algorithm slow. The authors in [21] constructed global path planning and local path planning strategies based on the hybrid artificial fish swarm algorithm. By developing the scoring function, the time of local path planning is shortened. On this basis, an obstacle avoidance and real-time navigation algorithm for the multi-robot cooperative path is proposed. However, this method will produce a lot of computation and cannot be used for real-time path planning.

3. Path Planning Based on DDPG Deep Reinforcement Learning

3.1. Kinematic Model of Mobile Robots. In the process of studying the kinematic model of mobile robots, the Pioneer3 mobile robot is taken as a research object, and its model is shown in Figure 1.

As shown in Figure 1, the Pioneer3 mobile robot is composed of the rear driving wheel and the front steering wheel, in which the rear wheel is the driving wheel, the speed is v, and the front wheel is the steering wheel. The position of the mobile robot is represented by a three-dimensional state vector $P(x, y, \alpha)$. (x, y) represents the midpoint of the robot's rear axle. The midpoint of the rear axis is used as a reference point to represent the position coordinates of the robot. α represents the included angle between the robot-fixed coordinate system and the spatial-fixed coordinate system, that is, the direction angle of the robot. β represents the steering angle of the robot, which is the angle of the steering wheel. The wheelbase of the driving wheel and steering wheel is Z_1.

The kinematic model of the Pioneer3 mobile robot is shown in the following formula:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 \\ \sin\alpha & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \dfrac{v\tan\beta}{Z_1} \end{bmatrix}. \tag{1}$$

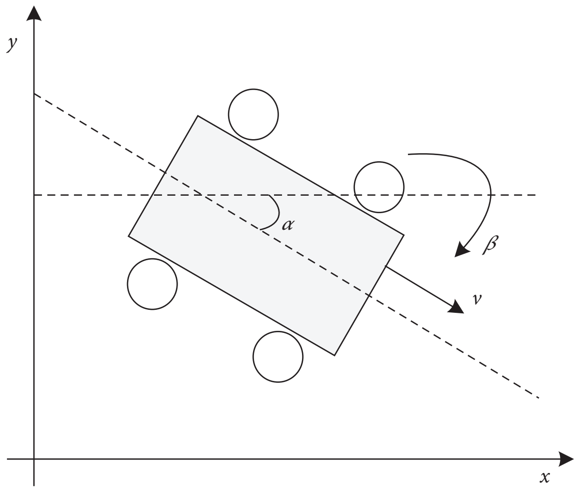

Figure 1: Basic structure of the Pioneer3 mobile robot.

3.2. Construction of the Optimized State Space. Traditionally, reinforcement learning is generally used in mobile robot path planning. Although unsupervised autonomous learning can be realized, there is also an inevitable problem; that is, it is highly dependent on training set information, and the learning process is very long. In addition, the solution to the problem is often not achieved overnight, but through the transition sequence of multiple state actions to reach the target state to obtain the final delayed return. In this way, the robot must repeatedly access these state-action transition sequences in order to find the optimal state-action transition sequence [22]. Therefore, reinforcement learning can converge to the optimal solution only when it runs to a certain extent. In the field of large and complex real problems, especially in the field of multirobots, the learning speed of reinforcement learning is very slow and learning efficiency is very low [23]. There will be problems such as "dimension disaster" and "state space combination explosion."

On the premise of ensuring the training results, how to greatly accelerate the efficiency of learning and the speed of reinforcement learning and training and expand reinforcement learning to larger and more complex applications has always been a hot spot in the field of reinforcement learning. Especially, in today's atmosphere, where most studies emphasize the performance of online, real-time adaptability to quickly adapt to the changes in the environment, the improvement of learning efficiency of reinforcement learning has increasingly become the research focus of researchers [24].

At the beginning of learning, the robot is very strange to the unknown environment and does not establish a stable state space. It must make actions randomly to obtain the reward value of different actions in each state, so as to slowly establish the understanding of the environment. For the motion decision-making of the robot, at the beginning of training, the robot collects the real-time depth image information in the environment as a real-time state, randomly gives the line speed and angular speed to move, calculates the reward value of this movement, collects the depth image information after the movement, and repeats this process until enough states and the probability distribution of the reward values of the corresponding actions are collected, Finally, training is completed when the reward values converges to a certain value.

In the process of repeated trial-and-error learning, the state when the robot moves to a place far away from the obstacle can be called the state not required for obstacle avoidance. Due to the random action selection, it is often in the collection state without obstacles for a long time so that the constructed state space contains a large number of states not required for obstacle avoidance, as shown in Figure 2.

What we hope is that the robot can move a large number of random near the obstacle and collect the current state to construct the state space, so as to obtain the obstacle avoidance ability when encountering the obstacle. Therefore, in the process of training, it is hoped that the robot can establish a state space for obstacle avoidance motion. In order to improve the efficiency of subsequent training, in the construction process of the initial state space, the distance between the robot and the obstacle is characterized according to the minimum depth of the field value in the depth image. Combined with robot kinematics, the motion of the robot is constrained and its training process is guided. A new reinforcement learning training process for path planning is proposed.

3.3. Algorithm Design

3.3.1. DDPG Algorithm Design. The path planning of the mobile robot in an unknown environment can be described as a Markov decision process. The process of the mobile robot interacting with the environment in offline time can be described as a decision sequence shown in the following formula:

$$\{S_0, A_0, R_1, S_1, A_1, R_2, \ldots, S_{T-1}, A_{T-1}, R_t, S_t, A_t\}. \tag{2}$$

In formula (2), S_t represents the state of the robot at the moment t, A_t represents the action of the robot at the moment t, and R_t represents the reward value obtained by the robot at the moment $t-1$. The mobile robot maximizes the total reward by finding an optimal action strategy. The total reward is defined as follows:

$$G_t = R_{t+1} + \gamma R_{t+2} + \cdots = \sum_{\tau=0}^{+\infty} \gamma^{\tau} R_{t+\tau+1}. \tag{3}$$

In formula (3), λ represents the discount factor, which is used to calculate the cumulative reward.

The strategy π represents the mapping from the state to action probability distribution, which is generally the probability distribution of the state. The output of the deterministic policy gradient algorithm is a specific action. The definition of the deterministic strategy π is shown in formula:

$$a = \pi(s, \theta). \tag{4}$$

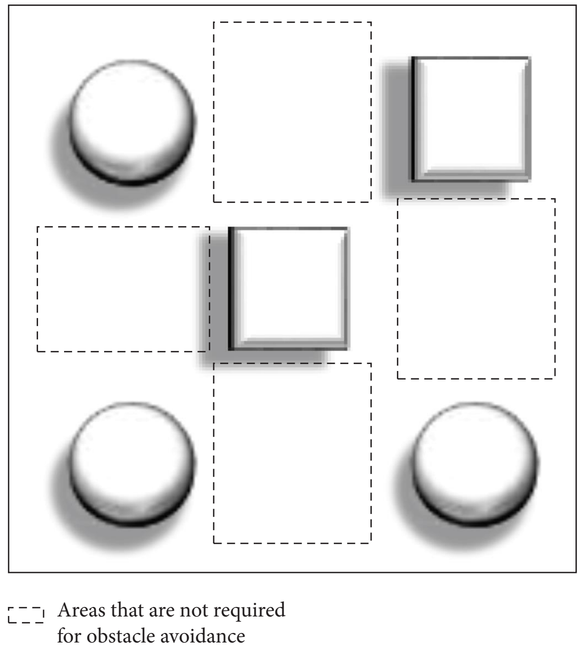

Figure 2: Collection status of areas not required for obstacle avoidance.

In formula (4), θ represents the parameters of the deterministic strategy, and the purpose of its update is to find an appropriate parameter θ to optimize the strategy π.

In the DDPG algorithm, the parameter θ is updated by the following formula:

$$\theta\leftarrow\theta+\beta\gamma^{t}\nabla_{\theta}\pi(S_t,\theta)[\nabla_{a}q(S_t,a,\eta)]_{a=\pi(S_t,\theta)}. \tag{5}$$

In formula (5), η represents the parameters of the value function. η can be updated by the following formula:

$$\eta\leftarrow\eta+\alpha[y-q_{\pi}(S_t,A_t;\eta)]\nabla_{\eta}q_{\pi}(S_t,A_t;\eta). \tag{6}$$

Compared with the DPG algorithm, the DDPG algorithm uses the experience replay and target network technology, which can deal with higher dimensional observation space and improve the stability of the algorithm. The flow of the algorithm is as follows:

(1) We initialize the policy network $\pi(S,\theta)$, value network $q(s,a)$, and parameters θ and η. Then, we initialize the target policy network $\pi'(s)$, target value network $q\prime(s,a)$, and parameters $\theta'\leftarrow\theta$ and $\eta'\leftarrow\eta$. We also initialize the target network learning rate σ. The batch size of each learning sample is N, and the experience pool size is R.

(2) We select the action $a_t=\pi(s_t)$ in the state S_t.

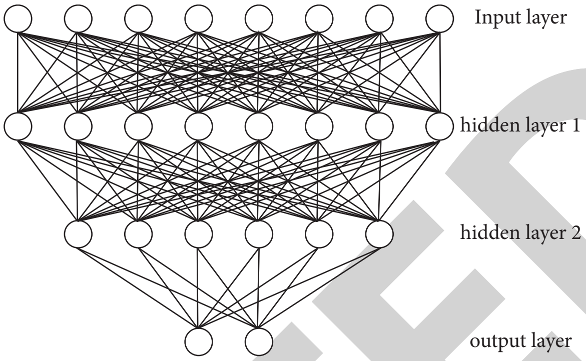

Figure 3: The basic structure of the path-planning strategy neural network.

(3) We perform actions to get rewards r_{t+1} and S_{t+1} from the environment.

(4) We save $\{S_T,a_t,r_{t+1},S_{t+1}\}$ to the experience pool.

(5) When the samples in the experience pool meet the conditions for starting training, P samples are randomly selected from the experience pool, $\{(S_t,a_t,r_{t+1},S_{t+1})\}_{i=1}^{N}$.

(6) We update the policy network and value network:

$$y_i=r_{i+1}+\gamma q(s_{i+1},\pi(s_{i+1},\theta'),\eta'), \tag{7}$$

$$\eta\leftarrow\eta+\alpha\frac{1}{N}\sum_{i}[y_i-q(s_i,a_i,\eta)]\nabla_{\eta}q(s_i,a_i,\eta), \tag{8}$$

$$\theta\leftarrow\theta+\beta\frac{1}{N}\sum_{i}\nabla_{\theta}\pi(h_i,\theta)[\nabla_{a}q(h_i,a,\eta)]_{a=\pi(h_i,\theta)}. \tag{9}$$

(7) We update the target network at regular intervals:

$$\eta'\leftarrow\varepsilon\eta+(1-\varepsilon)\eta', \tag{10}$$

$$\theta'\leftarrow\varepsilon\theta_c+(1-\varepsilon)\theta'. \tag{11}$$

3.3.2. Design of the Reward and Punishment Function and State Space. In order to simplify the model of path planning, it is assumed that the robot moves at a fixed speed; that is, the robot has a fixed moving distance in each time step. Therefore, the steering angle β of the robot is taken as an action space, and the dimension is 1. In the training of deep reinforcement learning, the purpose of the robot is to avoid obstacles and move to the target point at the same time. The state space of the robot is defined as follows:

$$Z = \begin{cases} \dfrac{(x, y)}{\rho}, \\ \dfrac{\delta}{2\pi}, \\ \dfrac{D_{\min,o}}{\rho}, \\ \dfrac{\left[\left(x - x_{\min,o}\right), \left(y - y_{\min,o}\right)\right]}{\rho}, \\ \dfrac{D_{\min,g}}{\rho}, \\ \dfrac{\left[\left(x - x_{\min,g}\right), \left(y - y_{\min,g}\right)\right]}{\rho}. \end{cases} \tag{12}$$

In formula (12), (x, y) represent the position of the robot in the current map, δ represents the orientation of the robot in the current map, ρ represents the standardization coefficient, $D_{\min,o}$ represents the distance between the robot and the nearest obstacle, $D_{\min,g}$ represents the distance between the robot and the nearest target point, $(x - x_{\min,o})$, $(y - y_{\min,o})$ represent the distance information between the robot and the nearest obstacle, and $(x - x_{\min,g})$, $(y - y_{\min,g})$ represent the distance information between the robot and the nearest target. In actual motion, the distance between the robot and obstacles in the environment can be obtained by using sensors.

In the process of reinforcement learning, the quality of the reward function affects the effect of reinforcement learning. According to the basic framework of reinforcement learning, the agent evaluates the action through the feedback of the environment and selects the action that can obtain the maximum reward after learning. Therefore, the reasonable design of the reward function plays a vital role in reinforcement learning. Usually, the distance information between the robot and the target point is processed as the reward value; that is, the closer the robot is to the end point, the greater the reward it will get in each step of movement. However, this method of setting the reward function does not consider the changes between the robot and obstacle. A negative reward value is given only when the robot hits the obstacle, and the information of the robot's distance from the obstacle is not continuous. In view of the problems of setting the reward function in the abovementioned way, an artificial potential field method is proposed to set the reward function.

The artificial potential field method is a virtual force method. In the artificial potential field, the target end point will attract the mobile robot, while the obstacle will repel the robot. The superposition of these two forces is the control force of the mobile robot's motion process. Under the action of the control force, the mobile robot plans a path to the end point. Among them, the attractive force of the robot near the end point will become larger. If the robot approaches an obstacle, the repulsion force will become greater. In the classical artificial potential field, the attractive field function is shown in the following formula:

$$L_1(x) = \frac{1}{2}\kappa_1 d\left(x, x_g\right). \tag{13}$$

In formula (8), $\kappa_1 > 0$ represents the coefficient constant of the attractive field and $d(x, x_g)$ represents the distance of the robot from the target point. At the target point, the attractive potential energy L_1 is the smallest. The attractive function can be obtained by calculating the negative derivative of formula (8), as shown in the following formula:

$$F_1(x) = -\nabla L_2(x) = -\kappa_2 d\left(x, x_g\right). \tag{14}$$

The formula of the repulsion field function is shown in the following formula:

$$L_2(x) = \begin{cases} \dfrac{1}{2}\kappa_2\left(\dfrac{1}{d\left(x, x_{g0}\right)} - \dfrac{1}{d_{\max}}\right), & d\left(x, x_{g0}\right) \leq d_{\max}, \\ 0, \quad d\left(x, x_{g0}\right) > d_{\max}. \end{cases} \tag{15}$$

In formula (15), $\kappa_2 > 0$ represents the coefficient constant of the repulsion field, $d(x, x_{g0})$ represents the straight-line distance of the robot from the target point, and $d_{\max}$ represents the maximum influence range of the obstacle. When $d(x, x_{g0}) > d_{\max}$, the robot is not affected by obstacles. The repulsive potential energy at obstacles has the maximum value. The repulsion function can be obtained by calculating the negative derivative of formula (10), as shown in the following formula:

$$F_2(x) = \begin{cases} \left(\dfrac{1}{d\left(x, x_{g0}\right)} - \dfrac{1}{d_{\max}}\right)\dfrac{\kappa_2 \partial d\left(x, x_{g0}\right)}{d^2\left(x, x_{g0}\right)\partial x}, & d\left(x, x_{g0}\right) \leq d_{\max}, \\ 0, \quad d\left(x, x_{g0}\right) > d_{\max}. \end{cases} \tag{16}$$

The resultant potential field $L(x)$ and resultant force $F(x)$ received by the mobile robot during movement are shown in the following formula:

$$\begin{cases} L(x) = L_1(x) + L_2(x), \\ F(x) = F_1(x) + F_2(x). \end{cases} \tag{17}$$

When the robot has not reached the target or touched an obstacle, according to the idea of the artificial potential field, the reward value includes two parts: (1) the negative reward value of the distance information between the robot and the nearest obstacle and (2) the positive reward value of the distance information between the robot and the target point.

Table 1: Neural network-related parameter settings.

Parameters	Value	Meaning
Network learning rate	0.001	Network learning speed
Reward discount rate	0.92	Future rewards at the current value
Soft update parameters	0.01	Update parameters of the strategy network and target network
Steps per round	250	Maximum number of exploration steps per round
Total number of rounds	20000	Maximum number of rounds
Experience pool capacity	60000	Experience storage limit
Batch size	32	Update network training batch size

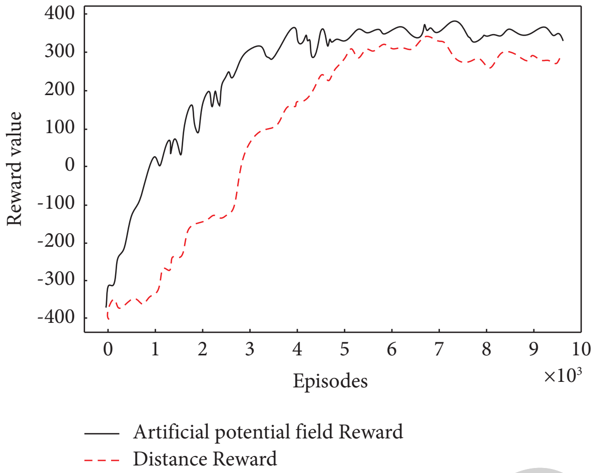

Figure 4: Total reward value per episode.

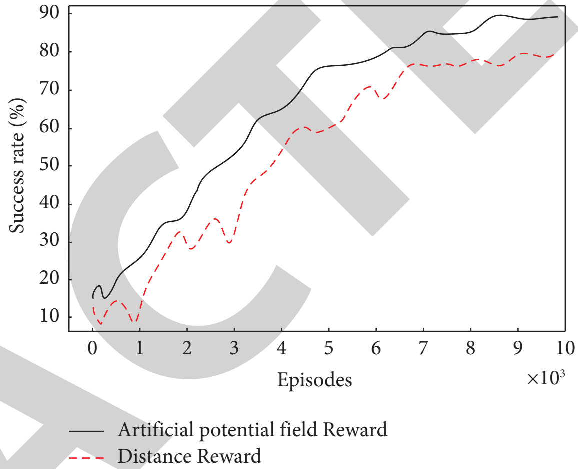

Figure 5: Change trend of the overall success rate.

The sum of the two reward values is taken as the final reward value obtained after the robot performs each action, as shown in the following formula:

$$C = C_1 + C_2 = \frac{1}{2^{d_{g/\kappa}}} - \frac{1}{2^{d_{o/\kappa}}}. \tag{18}$$

In formula (18), $d_{g/\kappa}$ represents the distance between the robot and the target point and $d_{o/\kappa}$ represents the distance between the robot and the obstacle.

The reward function of the robot action is obtained as shown in the following formula:

$$R = \begin{cases} 200, \\ \\ \dfrac{1}{2^{d_{g/\kappa}}} - \dfrac{1}{2^{d_{o/\kappa}}}, \\ \\ -200. \end{cases} \tag{19}$$

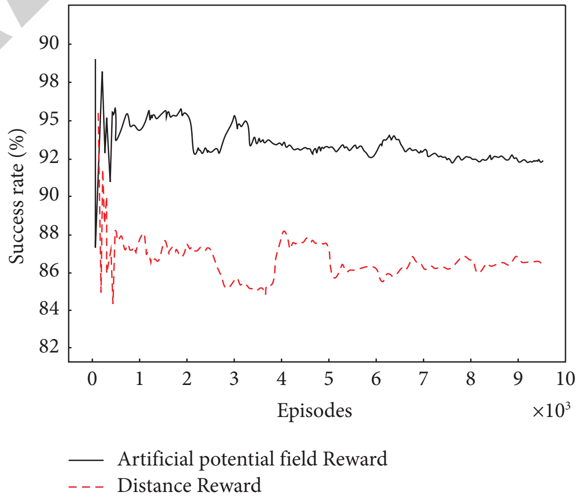

Figure 6: Change trend of the final success rate of training.

3.3.3. Neural Network Design. The designed path-planning strategy neural network is composed of four layers of the neural network, and its network structure is shown in Figure 3.

In Figure 3, the input layer is the current state of the mobile robot, the output layer is the linear velocity and angular velocity of the mobile robot, and there are two hidden layers in the middle. The input layer contains 12 neurons, and hidden layer 1 contains 250 neurons. The connection mode between the input layer and hidden layer 1 is full connection, and the ReLU nonlinear activation function is adopted. The input of hidden layer 2 contains 250 neurons. Hidden layer 1 and hidden layer 2 are fully connected, and the ReLU activation function is used. The output layer contains two neurons. Hidden layer 2 and the output layer are also fully connected, using the sigmoid activation function. The value network adopts a similar network structure. The main difference from the strategy network is that the input layer includes two parts: one is the current

Table 2: The total length of paths planned by different algorithms in different situations.

Method	Situation 1 (m)	Situation 2 (m)	Situation 3 (m)	Situation 4 (m)
Proposed method	2.8	4.6	5.9	8.3
Reference. [15]	3.7	5.5	7.2	9.5
Reference. [18]	3.2	4.9	6.5	8.8
Reference. [19]	3.5	5.1	6.6	9.2

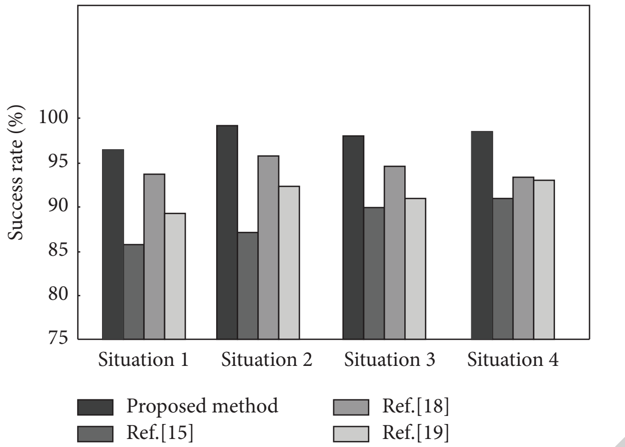

Figure 7: The success rate of path planning of different methods in different situations.

state of the robot and the other is the current action value of the robot. The output layer contains 1 neuron.

4. Experiment and Analysis

4.1. Experimental Environment. All the experiments in this chapter are implemented on the Windows7 professional 64 bit system and Python3.6. The simulation environment is designed by using pyglet, a multimedia framework under Python. A shape map with a pixel size of 1000 * 1000 is created. The pixel coordinates of four inner corners in the map are (300, 300), (700, 300), (300, 700), and (700, 700), respectively. The size of the robot in the environment is 100-pixel long and 50-pixel wide. The front end of the robot has five sensors to detect the distance, which can obtain the distance from the environmental boundary in all directions. The initial pose of the robot is $(500, 250, -\pi/2)$, and the speed is $v = 50$. The purpose of path planning is to enable the mobile robot to return to the initial position from the initial pose. The mobile robot cannot turn around in the environment under set speed conditions, so the mobile robot will complete the path planning around the map.

The pose of the robot can be solved by the following formula , and t is the sampling time.

$$\begin{cases} x_{P+1} = x_P + t \cdot v \cdot \ \cos \alpha_P, \\ \alpha_{P+1} = \alpha_P + t \cdot v \cdot \dfrac{\tan \beta_P}{Z_1}, \\ y_{P+1} = y_P + t \cdot v \cdot \ \sin \alpha_P. \end{cases} \tag{20}$$

After building the neural network and environment, we set the relevant parameters of the neural network according to Table 1.

4.2. Simulation Analysis. In this paper, deep reinforcement learning is trained based on the reward function built by the artificial potential field method and by using the distance information of the mobile robot from the target point. During the training process, the change trend of the total reward value, the change trend of the overall success rate, and the final success rate of training are shown in Figures 4–6, respectively.

It can be seen from Figures 4–6 that the reward function based on the artificial potential field method used in this paper converges more rapidly than the method that only takes the information of the distance from the target point of the mobile robot as the reward function. In addition, under the same training times, its success rate is also higher, and it has better goal orientation and obstacle avoidance ability.

The following is a comparative analysis of the proposed method and the mobile robot path-planning method proposed in [15], [18], and [19] under the same conditions for four different situations. The total length of mobile robot paths planned by different algorithms in different situations is shown in Table 2. The success rate of the proposed method in four different situations compared with the other three methods is shown in Figure 7.

It can be seen from Table 2 that in four different situations, compared with the other three comparison methods, the total lengths of the final planned path of the proposed method are both the shortest, which are 2.8 m, 4.6 m, 5.9 m, and 8.3 m, respectively. Compared with the other three methods, the maximum increased values are 0.9 m, 0.9 m, 1.3 m, and 1.2 m, respectively, and the minimum increased values are 0.4 m, 0.3 m, 0.6 m, and 0.5 m, respectively. This is because the method proposed in this paper uses the artificial potential field method to set the reward function, collects the information of the robot from the continuous obstacle, improves the performance of the path planning of the algorithm, and obtains the optimal path with a shorter path length.

As can be seen from Figure 7, in four different situations, compared with the other three comparison methods, the final success rates of the proposed method are the highest, which are 97.2%, 99.1%, 98.4%, and 98.6%, respectively. Compared with other comparison methods, it has been improved to a certain extent. The results show that the learning efficiency and training speed can be improved by combining the kinematic constraint motion of the robot and guiding its training process to establish the optimal state space for obstacle avoidance motion. The proposed algorithm can describe the path planning of the mobile robot in the unknown environment as a Markov decision-making process by introducing the DDPG algorithm. The stability of the algorithm and the success rate of path planning are greatly improved by using experience replay and target network technology.

5. Conclusion

Aiming at the problems of poor continuity and low success rate of the mobile robot spatial path-planning method, a continuous action spatial path-planning method of the mobile robot based on reinforcement learning is proposed. Through simulation experiments, the proposed method is compared with the other three methods. The basic ideas are as follows: ① Through the analysis of the kinematic model of the mobile robot, the optimal state space is constructed. ② The reward function of the mobile robot is set based on the artificial potential field method, and the information of the robot's distance from the obstacle is continuous. ③ By introducing the deep deterministic policy gradient (DDPG) algorithm, the success rate of mobile robot path planning is improved. By introducing the depth deterministic strategy, the gradient algorithm can use experience replay and target network technology to deal with higher dimensional observation space and improve the stability of the algorithm and the success rate of path planning.

This paper studies the path planning of mobile robots in static and dynamic environments only for one robot. Although there is research on the dynamic environment, the situation of multiple robots is different from the dynamic environment, so independent training and knowledge sharing need to be considered. In addition, there is a problem of cooperation among multiple robots, and there will be some new problems to be further solved.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Key Scientific Research Projects of Colleges and Universities in Henan Province (No.20A520029, No.21B520012).

References

[1] J. R. Bourne, E. R. Pardyjak, and K. K. Leang, "Coordinated bayesian-based bioinspired plume source term estimation and source seeking for mobile robots," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 967–986, 2019.

[2] C. Jiang, S. Sun, and J. Liu, *Global Path Planning of Mobile Robot Based on Improved JPS+ algorithm [C]*, Chinese Automation Congress (CAC), Shanghai, China, 2020.

[3] J. Guo, C. Li, and S. Guo, "A novel step optimal path planning algorithm for the spherical mobile robot based on fuzzy control," *IEEE Access*, vol. 8, no. 12, pp. 1394–1405, 2020.

[4] W. Chi, Z. Ding, J. Wang, G. Chen, and L. Sun, "A generalized voronoi diagram-based efficient heuristic path planning method for RRTs in mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 5, pp. 4926–4937, 2022.

[5] K. R. Jensen-Nau, T. Hermans, and K. K. Leang, "Near-optimal area-coverage path planning of energy-constrained aerial robots with application in autonomous environmental monitoring," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1453–1468, 2021.

[6] C. Tatino, N. Pappas, and D. Yuan, "Multi-robot association-path planning in millimeter-wave industrial scenarios," *IEEE Networking Letters*, vol. 2, no. 4, pp. 190–194, 2020.

[7] N. S. Monteiro, V. M. Gonçalves, and C. A. Maia, "Motion planning of mobile robots in indoor topological environments using partially observable Markov decision process," *IEEE Latin America Transactions*, vol. 19, no. 8, pp. 1315–1324, 2021.

[8] Q. Lu, D. Zhang, W. Ye, J. Fan, S. Liu, and C. Y. Su, "Targeting posture control with dynamic obstacle avoidance of constrained uncertain wheeled mobile robots including unknown skidding and slipping," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6650–6659, 2021.

[9] Z. Jian, S. Zhang, S. Chen, Z. Nan, and N. Zheng, "A global-local coupling two-stage path planning method for mobile robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5349–5356, 2021.

[10] F. Ugalde Pereira, P. Medeiros de Assis Brasil, M. A. De Souza Leite Cuadros, A. R. Cukla, P. Drews Junior, and D. F. Tello Gamarra, "Analysis of local trajectory planners for mobile robot with robot operating system," *IEEE Latin America Transactions*, vol. 20, no. 1, pp. 92–99, 2022.

[11] R. Zhou and L. Hui, "Path planning of mobile robot based on improved dynamic programming algorithm," *Computer Engineering and Application*, vol. 56, no. 21, pp. 20–24, 2020.

[12] X. Y. Zhong, J. Tian, H. S. Hu, and X. Peng, "Hybrid path planning based on safe a algorithm and adaptive window approach for mobile robot in large-scale dynamic environment," *Journal of Intelligent and Robotic Systems*, vol. 99, no. 1, pp. 65–77, 2020.

[13] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6932–6939, 2020.

[14] S. K. Angappamudaliar Palanisamy, D. Selvaraj, and S. B. K. Ramasamy, "Hybrid multi-objective optimization approach intended for mobile robot path planning model," *Journal of Intelligent and Fuzzy Systems*, vol. 42, no. 3, pp. 2681–2693, 2022.

[15] M. Alireza, D. Vincent, and W. Tony, "Experimental study of path planning problem using EMCOA for a holonomic mobile robot," *Journal of Systems Engineering and Electronics*, vol. 32, no. 6, pp. 1450–1462, 2021.

[16] R. Fareh, M. Baziyad, T. Rabie, and M. Bettayeb, "Enhancing path quality of real-time path planning algorithms for mobile robots: a sequential linear paths approach," *IEEE Access*, vol. 8, no. 3, pp. 167090–167104, 2020.

[17] S. D. Han and J. Yu, "DDM: fast near-optimal multi-robot path planning using diversified-path and optimal sub-

problem solution database heuristics," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1350–1357, 2020.

[18] R. Vieira, E. Argento, and T. Revoredo, "Trajectory planning for car-like robots through curve parametrization and genetic algorithm optimization with applications to autonomous parking," *IEEE Latin America Transactions*, vol. 20, no. 2, pp. 309–316, 2022.

[19] G. H. Xu, T. W. Zhang, and Q. Lai, "A new path planning method of mobile robot based on adaptive dynamic firefly algorithm," *Modern Physics Letters B*, vol. 34, no. 29, pp. 159–167, 2020.

[20] Q. Li, W. Lin, Z. Liu, and A. Prorok, "Message-aware graph attention networks for large-scale multi-robot path planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5533–5540, 2021.

[21] Y. Q. Huang, Z. K. Li, and Y. Jiang, "Cooperative path planning for multiple mobile robots via HAFSA and an expansion logic strategy," *Applied Sciences-Basel*, vol. 9, no. 4, pp. 32–40, 2019.

[22] H. Wang, C. Hao, and P. Zhang, "Path planning of mobile robots based on A~* algorithm and artificial potential field algorithm," *China Mechanical Engineering*, vol. 30, no. 20, pp. 2489–2496, 2019.

[23] D. S. Lyu, Z. W. Chen, Z. S. Cai, and S. Piao, "Robot path planning by leveraging the graph-encoded Floyd algorithm," *Future Generation Computer Systems*, vol. 122, no. 56, pp. 204–208, 2021.

[24] J. Lima, P. Costa, and P. Costa, "A* search algorithm optimization path planning in mobile robots scenarios," in *Proceedings of the [C]//International Conference on Numerical Analysis and Applied Mathematics (ICNAAM)*, AIP publishing, Rhodes, Greece, 2019.