Research on the Local Path Planning for Mobile Robots Based on PRO-Dueling Deep Q-Network (DQN) Algorithm

Yaoyu Zhang, Caihong Li[*], Guosheng Zhang, Ruihong Zhou, and Zhenying Liang
School of Computer Science and Technology, Shandong University of Technology, Zibo 255049, China

***Abstract*—This paper proposes a Pro-Dueling DQN algorithm to solve the problems of slow convergence speed and waste of effective experience of the traditional DQN (Deep Q-Network) algorithm for the local path planning of mobile robot. The new algorithm introduces a priority experience playback mechanism based on SumTree to avoid forgetting the learning effective experiences as the number of samples in the experience pool increases. A more detailed reward and punishment function is designed for the new algorithm to reduce the blindness of extracting experience in the early stages of algorithm training. The feasibility of the algorithm is verified by comparative verification on ROS simulation platform and real scene, respectively. The results show that the designed Pro-Dueling DQN algorithm converges faster and the length of planned path is shorter than that of the original DQN algorithm.**

***Keywords*—Deep Q-Network (DQN) algorithm; local path planning; mobile robot; Pro-Dueling DQN algorithm; SumTree**

I. Introduction

It is crucial for robots to avoid obstacles and plan effective paths in the research of mobile robot navigation. There are many effective path planning methods for obstacle avoidance at present. The traditional methods mainly include A* algorithm [1], Dijkstra algorithm [2], fuzzy control algorithm [3], genetic algorithm [4], artificial potential field method [5] and neural network [6]. Reinforcement learning algorithm [7] has received widespread attention because it can solve the shortcomings of traditional algorithms such as strong dependence on environment in robot path planning. It does not require any prior knowledge, and optimizes the strategy by interacting with the environment and accumulating rewards. The combination of deep learning [8] and reinforcement learning has extended traditional reinforcement learning to multidimensional state space and action space in recent years. Deep reinforcement learning [9] combines the ability of deep learning algorithm to understand perception problems and the ability to fit the learning results of reinforcement learning algorithm [10]. It has been widely used in robot path planning research.

Q-learning algorithm is one of the reinforcement learning algorithms proposed by Watikins, which is independent of environmental prior model in the path planning problem of mobile robot[11]. However, the strategy of storing the state-action value function by a Q-value table will cause the disaster of dimension as the environment states become more and more complex. Mnih et al. proposed Deep Q-Network (DQN), which combined CNN (Convolutional Neural Networks) with Q-learning algorithm to solve the dimension disaster of Q-learning method, and pushed the research of deep reinforcement learning to a new level [12]. Z. Wang et al. innovated the network structure on the basis of DQN and divided the network into two parts: value function and advantage function to reduce the excessive dependence of states on the environment [13]. J.F. Zheng et al. proposed an improved DQN algorithm based on depth image information. PTZ (Pan/Tilt/Zoom) was used to obtain depth image information of obstacles, which improved the convergence speed of the network, but the stability and computational speed of the algorithm could not be guaranteed [14]. Xiaofei Yang et al. proposed a global path planning algorithm based on DDQN, which integrated an action mask method to deal with the invalid actions generated by the amphibious unmanned vehicle, but the algorithm training speed is not ideal[15]. Meng Guan et al. proposed a DQN path planning method combining heuristic reward and adaptive exploration strategy, designed a heuristic reward function based on artificial potential field method, and self-adaptively adjusted the balance between exploration and utilization in the algorithm, which accelerated the learning efficiency of the algorithm. However, the algorithm verification only stayed in the simulation stage, and the efficiency of the algorithm has not been verified in the real scene [16]. The new method improves the efficiency of the algorithm's exploration, but the length of exploration in the direction of exploration increases, resulting in excessive spatial dimensions.

This paper presents a Pro-Dueling DQN algorithm to solve the problems of poor convergence and waste of effective experience in the local path planning of mobile robot using DQN method. The research modifies the DQN neural network structure to combine the state value and action value to obtain a more accurate Q-value. The priority experience playback strategy [17] based on SumTree is adopted to give priority to the samples in the experience pool, and designs a reward and punishment function to solve the convergence difficulty problem caused by sparse rewards in unknown environments. This improves the utilization rate of effective experience in the algorithm, avoids the problem of local optimal solution, and accelerates the convergence speed of the algorithm. Comparing the convergence speed and planned path length of the two algorithms in simulation and real environments, experimental results show that the Pro-Dueling DQN algorithm performs better in various scenarios.

II. DQN Algorithm

DQN algorithm combines Q-learning algorithm with deep learning, uses network structure in deep leaning to predict Q-value, and generates Q-table dynamically. It not only avoids the disaster of dimensionality in complex space, but also solves the instability problem of approximate representation of value functions for nonlinear functions to a certain extent [18]. Fig. 1 shows the process of DQN algorithm.

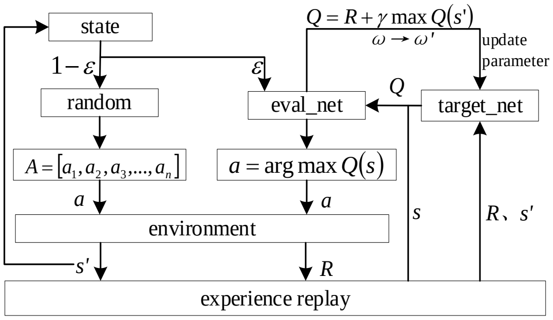


Fig. 1. DQN algorithm process diagram.

The algorithm defines two relatively independent networks with the same structure, namely eval_net and target_net. The agent interacts with the environment to achieve learning of the training network. All the parameters in the training network are assigned to the target network after the training of a fixed number of steps. The algorithm sets an experience replay unit to reduce the correlation of training samples and improve the instability of the action value function of neural network approximation reinforcement learning [19]. A batch of samples are evenly selected from the experience library and mixed together with the training samples to break the correlation between adjacent training samples and improve the utilization rate of the samples during each training. Where, the LOSS function is:

$$LOSS = \frac{1}{2}\left(Q_{\omega}(s) - \left(R + \gamma \max Q_{\omega'}(s')\right)\right)^2 \quad (1)$$

In Eq. (1), ω is a parameter in the eval_net, ω' is a parameter in the target_net. The parameters of target_net are synchronized with the training network every N steps to make the updated target more stable, that is, $\omega' \leftarrow \omega$.

III. Pro-Dueling DQN Algorithm

This research proposes a Pro-Dueling DQN algorithm to improve the training speed and convergence of DQN algorithm. Two different branches are introduced at the back end of DQN neural network to predict the value of state and action, then the results of these two branches are combined to output the Q-value to reduce the dependence of action on state. In addition, the priority experience playback based on SumTree replaces the uniform sampling playback mechanism of DQN algorithm to increase the sampling rate of important samples. The Pro-Dueling DQN algorithm includes the design of network structure of state space and action space, reward and punishment function and priority experience playback.

A. The Design of State Space and Action Space

The state space is the feedback of the environment information of the mobile robot. The input of the network is the state vector. The robot selects the subsequent action based on the state information, obtains the corresponding reward or punishment, and optimizes the strategy by accumulating the reward value.

The laser radar installed on the robot detects the surrounding environmental information. The detection range of radar sensor is 180°, and a group of data is returned every 15° with a total of 12 groups. Fig. 2 shows the laser radar information.

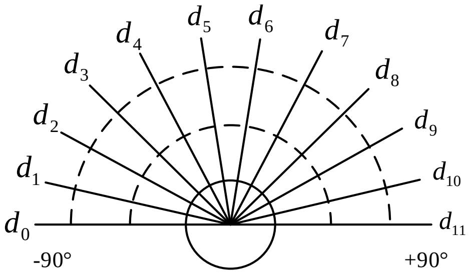


Fig. 2. Laser radar information.

The position information of the robot consists of the obstacle distance information returned by the radar in 12 directions d_n (n=0~11), the distance between the robot and the target point D_{g}, and the angle between the robot and the target point θ_{g}. The state space of the robot is defined as:

$$S = \left(d_n, D_g, \theta_g\right) \quad (2)$$

The robot's action space includes action information in five directions, defined as:

$$A = \left\{a_t, t = 0 \sim 4\right\} \quad (3)$$

The linear speed of the robot is constant, set as 0.15m/s, and the angular velocity is determined by its action. The relationship between the robot angular velocity (Angle_v) corresponding to the five action values of the robot is shown in Table I.

TABLE I. Corresponding Relationship between Robot Action and Rotation Angle

Action	Angle_v（rad/s）
0	-1.5
1	-0.75
2	0
3	0.75
4	1.5

B. Network Structure

The neural network used by Pro-Dueling DQN algorithm contains 14 inputs of the state space and 5 outputs of the action space, as shown in Fig. 3.

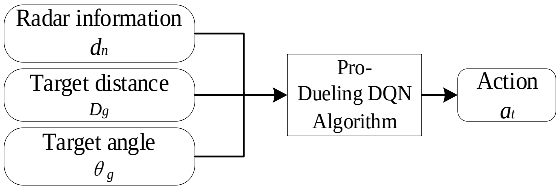


Fig. 3. Input and output of Pro-Dueling DQN network.

In the Pro-Dueling DQN algorithm, the output of the network includes a value function and an advantage function. The formula is as follows:

$$Q(s,a;\gamma,\alpha,\beta)= \\ V(s;\gamma,\beta)+A(s,a;\gamma,\alpha) \tag{4}$$

In Eq. (4), $V(s;\gamma,\beta)$ is a value function, $A(s,a;\gamma,\alpha)$ is the advantage function of taking different actions in this state, indicating the difference of taking different actions. γ is a network structure,α is the parameter of value function,β is the parameter of advantage function. It can be seen from the formula that $V(s;\gamma,\beta)$ function is only related to the state, and $A(s,a;\gamma,\alpha)$ depends on both state and action. The neural network outputs the value function and dominance function, respectively, and sums them to obtain the Q value. The robot only pays attention to the value of the state in some cases, and does not care about the difference caused by different actions by modeling $V(s;\gamma,\beta)$ function and $A(s,a;\gamma,\alpha)$ function. The approach works better with states that are less associated with an action.

Fig. 4 shows the network structure of the Pro-Dueling DQN algorithm. L_1 and L_2 are fully connected layers, which contains 128 and 64 hidden neuron nodes, respectively. In the network, input eigenvalues are used to obtain eigenvectors using a convolutional network. When outputting, two fully connected layers are used to correspond to the state value and advantage value, respectively. Finally, the state value and advantage value are added to obtain the action value of each action.

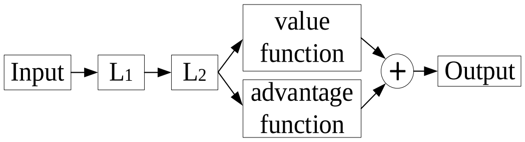


Fig. 4. Pro-Dueling DQN network structure.

C. The Design of Reward and Punishment Function

In the process of reinforcement learning, rewards and punishments obtained by mobile robot in its interactions with the environment are the key to completing tasks. In this research, the reward and punishment function is further refined, which including two parts: R_θ , the reward and punishment function of the angle between the robot and the target point, and *Rt*, the reward and punishment function of the distance between the robot and the obstacle. The final reward and punishment value R is obtained as follows by adding the two parts:

$$\begin{aligned} &R_\theta=\begin{cases} C,-\dfrac{1}{2}\pi<\theta<\dfrac{1}{2}\pi \\ -C, else \end{cases} \\ &R_t=\begin{cases} r_{goal}, d_c<c_d \\ r_{collide}, \min{}_x<c_o \end{cases} \\ &R=R_\theta+R_t \end{aligned} \quad . \tag{5}$$

In Eq. (5), C is a positive integer, θ is the angle of the robot to the target point, r_{goal} is a positive integer which representing a positive reward of the robot reaching the target point, d_c is the actual distance from the robot to the target, c_d is the threshold of reaching the target point. It means that the robot has reached the target point, when d_c is less than c_d. $r_{collide}$ is a negative integer, representing the penalty for the robot to encounter with obstacles, min_x is the radar minimum,c_o is the safe distance. It is determined that the robot collides with obstacle as the radar value is less than the safe distance.

D. Priority Experience Playback Based on SumTree

The research adopts the priority experience replay strategy to improve this situation. The most valuable experiences are extracted first as training. *TD-error* determines the priority of the sample. The target function is weighted according to the *TD-error* of the sample. The greater the deviation, the larger the sample weight, and the higher the priority p is.

A random sampling method combined greedy sampling and uniformly distributed table sampling is used to solve the overfitting problem caused by greedy priority in the process of function approximation. This method ensures that the probability of sampling from the storage container is monotonous, and the lowest priority sample has a non-zero probability of being drawn. The sampling probability is as follows:

$$P(i)=\sum\nolimits_k^{P_i^\alpha} P_k^\alpha \ . \tag{6}$$

In Eq. (6), P_i is the priority of the ith sample, P_k is the priority of any sample, α is used to adjust the degree of priority. It is reduced to uniform sampling, when $\alpha=0$. k is the number of batches of samples.

IV. Analysis and Verification of Pro-Dueling DQN Algorithm

In this research, the environments of discrete obstacles, U-shaped obstacle and mixed obstacles are set for training to verify the feasibility of the designed Pro-Dueling DQN algorithm. The algorithm is compared in different environments with the traditional DQN algorithm. The environments are built on Gazebo of ROS platform. Their informations are projected into Rviz. The path planned by the mobile robot from the starting point to the target point is displayed on Rviz. In Rviz, the initial position of the robot is the starting point, the gray box represents the target point, the gray cylinder shows the obstacle, and the blue solid line expresses the trajectory of the robot. A comparison graph of the average return value of each round of the two algorithms is drawn to observe the convergence of the algorithm more clearly and intuitively. The horizontal axis represents the number of training, and the vertical axis expresses the average reward of each training. At the same time, the path lengths of the robot by the two algorithms for path planning are recorded. Table II illustrates the parameter settings in the experiment.

TABLE II. EXPERIMENTAL PARAMETER SETTING

Parameter	Initialization value
learning rate	0.0001
attenuation factor	0.999
experience pool capacity	10000
number of learning experiences per round	128
maximum number of steps per round	400

A. Simulation Verification in Discrete Obstacles Environment

Fig. 5 shows the 7m×7m discrete obstacles environment set in Gazebo. The initial position of the robot is (-2.5, -2.5), and the target point coordinate is (2.2).

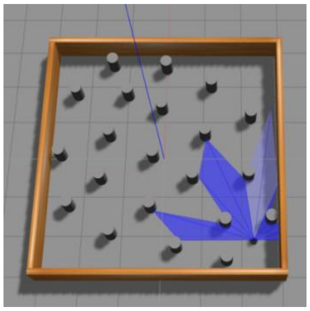


Fig. 5. The discrete obstacles simulation environment.

In the discrete obstacles environment, the two algorithms conduct 200 rounds of training, and record the average reward of each round. Fig. 6 shows the recording results. It can be seen from the figure that the average reward of the Pro-Dueling DQN algorithm gradually increases after 25 rounds, indicating that the success rate of the robot in the process of finding the target point is getting higher and higher, and the algorithm tends to converge after the 100th round of training. The convergence speed of it is faster than the traditional DQN algorithm, and the average reward of the convergence algorithm has less fluctuation and is relatively stable.

Fig. 7 shows the paths planned by the model after convergence of the two algorithms. The path planned by the Pro-Dueling DQN algorithm from the starting point to the target point is smoother and has fewer path steps than the traditional one, as can be seen in the figure. The number of steps taken by the designed strategy is 235, while the number is 283 by the traditional one.

B. Simulation Verification in U-shaped Obstacle Environment

Fig. 8 shows a 5m×5m U-shaped obstacle environment set up in Gazebo. The starting point coordinate of the robot is (-2.0, 0.0) and the target point coordinate is (1.5, 1.5).

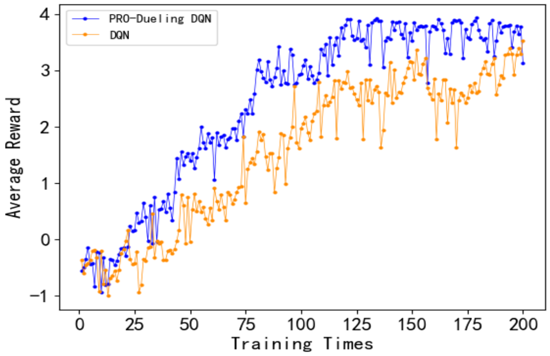


Fig. 6. The comparison of average reward per round in discrete obstacles environment.

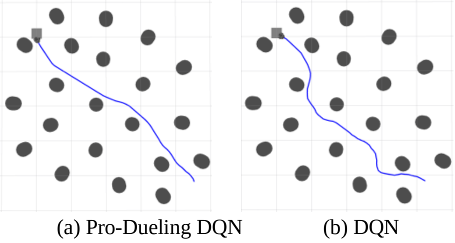

(a) Pro-Dueling DQN (b) DQN

Fig. 7. Path planning in discrete obstacles environment.

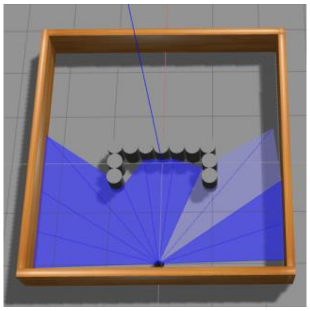


Fig. 8. U-shaped obstacle simulation environment.

Fig. 9 shows the record of the average return value of each round after 400 rounds of training by the Pro-Dueling DQN algorithm and the traditional DQN algorithm. Data displayed in the figure show that the average reward of the Pro-Dueling DQN algorithm after the 50th round is significantly higher than that of the traditional DQN algorithm, which indicates that the robot by the Pro-Dueling DQN algorithm can reach the target point more times. The Pro-Dueling DQN algorithm tends to converge after 150 rounds. The convergence speed of the Pro-Dueling DQN algorithm is faster, and the average reward of the convergence algorithm fluctuates less, indicating that the designed algorithm is more stable.

Fig. 10 shows the paths planned by the model after convergence of the two algorithms. The path planned by the Pro-Dueling DQN algorithm from the starting point to the target point is shorter than that by the traditional DQN algorithm. The number of steps is 268, while the number of the traditional one is 298.

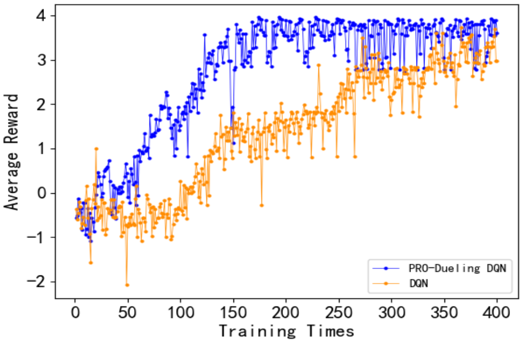
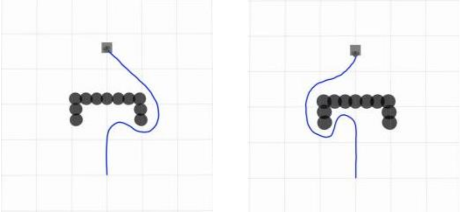


Fig. 9. Comparison of average reward per round in U-shaped obstacle environment.



(a) Pro-Dueling DQN (b) DQN

Fig. 10. U-shaped obstacle path planning.

C. *Simulation Verification in Mixed Obstacles Environment*

Fig. 11 and Fig. 12 show two 6m×6m mixed obstacles environments set up in Gazebo, which include discrete obstacles, 1-shaped obstacles and U-shaped obstacles. In Fig. 11, the starting point coordinate of the robot is (-2.0, 2.0) and the target point coordinate is (2.1,-2.0). In Fig. 12, the starting point coordinate of the robot is (-2.0,-2.0) and the target point coordinate is (2.5, 2.0).

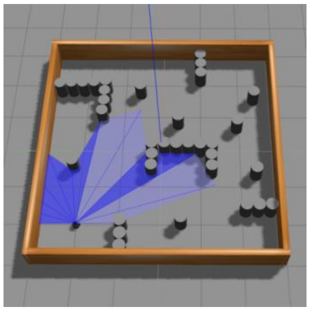


Fig. 11. Mixed obstacles environment (1).

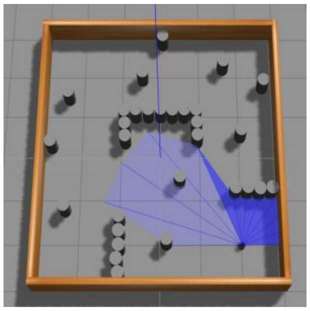


Fig. 12. Mixed obstacles environment (2).

Fig. 13 records the average reward of 500 rounds of training in the environments conducted by the Pro-Dueling DQN algorithm and the traditional DQN algorithm. The data in the figure show that the average reward value of the Pro-Dueling DQN algorithm after 100th round is significantly higher than that of the traditional one, indicating that the robot reach the target point more times by the Pro-Dueling DQN algorithm. The Pro-Dueling DQN algorithm tends to converge after 380 rounds. The convergence speed of it is faster than the traditional one.

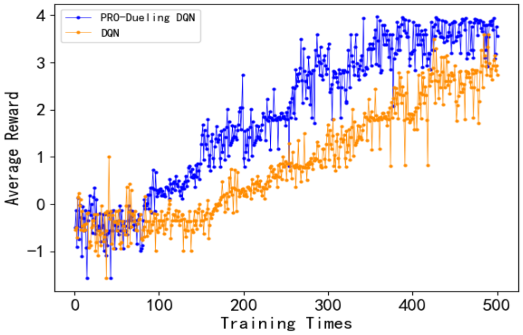
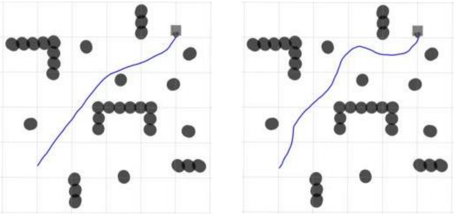


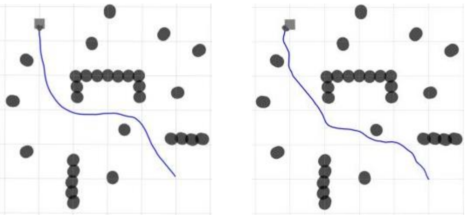
Fig. 13. Comparison of average reward value per round in mixed obstacles environment.

Fig. 14 and Fig. 15 show the path planned by the model after convergence of the two algorithms. The paths planned by the Pro-Dueling DQN algorithm from the starting point to the target point are shorter than the paths planned by the traditional DQN algorithm in both figures. In Fig. 14, the mixed obstacles environment (1), the number of steps taken by the Pro-Dueling DQN algorithm is 258, while the number by the traditional DQN algorithm is 311. The data in Fig. 15, the other mixed obstacles environment, are 302 and 337, respectively.



(a) Pro-Dueling DQN algorithm (b) DQN algorithm

Fig. 14. Path planning in mixed obstacles environment (1).



(a) Pro-Dueling DQN (b) DQN.

Fig. 15. Path planning in mixed obstacles environment (2).

The feasibility and effectiveness of the designed Pro-Dueling DQN algorithm in robot path planning are verified through simulation training in different obstacle environments, and are compared with the traditional DQN algorithm. The simulations show that the convergence speed of Pro-Dueling DQN algorithm is faster and more stable than DQN algorithm. The path planned by the Pro-Dueling DQN algorithm is shorter and smoother in the same training times and operating environment.

Fig. 16 shows the comparison of the planned steps of the two algorithms from the starting point to the target point in the above three simulation environments. The data in the figure show that the Pro-Dueling DQN algorithm uses fewer steps than the traditional DQN algorithm in each environment.

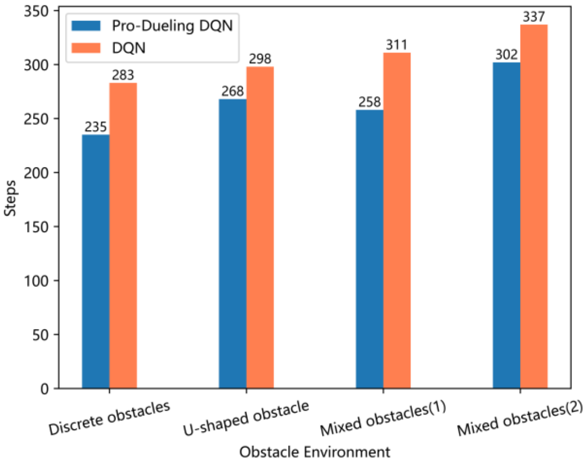
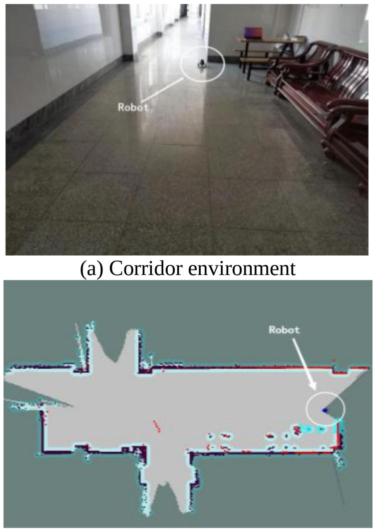


Fig. 16. Comparison of path steps between Pro-Dueling DQN algorithm and DQN algorithm.

D. Verification Experiments in Real Scenarios

The trained algorithm model is loaded into the robot, and the paths planned by the Pro-Dueling DQN algorithm and the DQN algorithm are tested and compared in the real environment. The ROS integrated SLAM (Simultaneous Localization and Mapping) function package is used to build the corridor environment model of teaching building. Fig. 17 shows the corridor environment and map model.



(a) Corridor environment

(b) Map model

Fig. 17. Corridor environment.

The path planning experiments are carried out in the map model built by the robot. Fig. 18 shows the addition of temporary obstacles in the corridor environment during the experiment to verify the real-time obstacle avoidance performance of the local path planning algorithm. The robot uses Radar to scan obstacle information, and the pink parts in the map model represent the unknown obstacles detected in real time. The target point is selected in the map, and the robot moves towards the target point from the starting point. The obstacle information is detected and fed back in real time by Radar, so that the robot plans a collision-free path from the starting point to the target point. Rviz is used to display and record the path planned by the two algorithms. Fig. 19 shows the planned paths. In the environment, the path length planned by the Pro-Dueling DQN algorithm is 7.835 meters, and that by the DQN algorithm is 8.563 meters. Both paths planned by the two algorithms can avoid temporary obstacles to reach the target point, while the paths planned by the Pro-Dueling DQN algorithm are shorter than those planned by the DQN algorithm, and the obstacle avoidance paths are smoother when encountering obstacles.

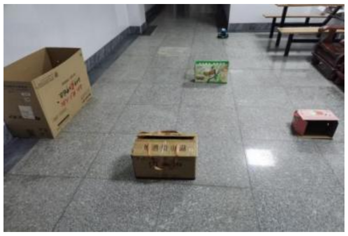
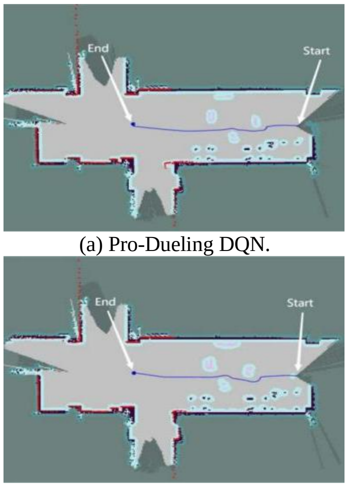


Fig. 18. Obstacle environment.



(a) Pro-Dueling DQN.

(b) DQN

Fig. 19. Results of path planning in corridor environment.

V. Conclusion

This paper proposes a Pro-Dueling DQN algorithm based on DQN algorithm and SumTree algorithm to solve the local path planning problem of mobile robot in unknown environment. The effectiveness of the proposed algorithm is verified by comparison experiments on ROS simulation platform and real environment. The experimental results show that the trained Pro Dueling DQN algorithm model can perform better in robot local path planning tasks, obtain smoother paths compared to the original algorithm, and complete tasks more efficiently and quickly. At the same time, the model has certain adaptability and can plan feasible paths in unknown environments in real-time based on sensor information.

However, the robot designed in this research has fewer actions, resulting in a large swing range in the training process. The planned paths in the complex and dense obstacles environments are not smooth, and the trained paths are not the optimal shortest ones. So future work will design more detailed action spaces, increase training time, and enable robot to plan better paths.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Shandong Province, China (No. ZR2021MF072).

REFERENCES

[1] T.T. Sang, J.C. Xiao, J.F. Xiong, H.Y. Xia,, and Z.Z. Wang, Path planning method of unmanned surface vehicles formation based on improved A* algorithm, *Journal of Marine Science and Engineering*, 2023, vol. 11, no. 1, pp. 176-176.

[2] B.Y. He, Application of Dijkstra algorithm in finding the shortest path, *Journal of Physics: Conference Series*, 2022, vol. 2181, no. 1.

[3] H.B. Gao, S.Y. Lu, and T. Wang, Motion path planning of 6-DOF industrial robot based on fuzzy control algorithm, *Journal of Intelligent & Fuzzy Systems*, 2020, vol. 38, no. 4, pp. 3773-3782.

[4] K. Hao, J.L. Zhao, Z.S. Li, Y.L. Liu, and L. Zhao, Dynamic path planning of a three-dimensional underwater AUV based on an adaptive genetic algorithm, *Ocean Engineering*, 2022, vol. 263.

[5] G.Q. Zhang, J. Han, J.Q. Li, and X.K. Zhang, APF-based intelligent navigation approach for USV in presence of mixed potential directions: Guidance and control design, *Ocean Engineering*, 2022, vol. 260.

[6] W.Z. Du, Q.M. Zhang, Z.X. He, and X. Wang, Real Time Neural Network Path Planning Algorithm for Robot, *International Journal of Frontiers in Engineering Technology*, 2021, vol. 3, no. 5.

[7] A. Khan, F. Jiang, S. Liu, and O. Ibrahim, Playing a FPS doom video game with deep visual reinforcement learning, *Automatic Control and Computer Sciences*, 2019, vol. 53, no. 3, pp. 214-222.

[8] B. Tamir, J. William, and Y. Kazuya, Deep learned path planning via randomized Reward-Linked-Goals and potential space applications, *CoRR*, 2019, vol. abs/1909.06034.

[9] Z. Li, S.H. Yuan, X.F. Yin, X.Y. Li, and S.X. Tang, Research into autonomous vehicles following and obstacle avoidance based on deep reinforcement learning method under map constraints, *Sensors*, 2023, vol. 23, no. 2, pp. 844-844.

[10] Z. Yu, J. Bi, and H.T. Yuan, A path planning method for complex naval battlefields based on improved DQN algorithm, *Journal of Intelligent Science and Technology*, 2022, vol. 4, no. 3, pp. 418-425.

[11] C. Watkins, and P. Dayan, Q-learning, *Machine Learning*, 1992, vol. 8, no. 3-4, pp. 279-292.

[12] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Grabes, M. Riedmiller, A.K. Fidjeland, and G. Ostrovski, Human-level control through deep reinforcement learning, *Nature*, 2015, vol. 518, no. 7540, pp. 529-533.

[13] Z. Wang, N.D. Freitas, and M. Lanctot, Dueling network architectures for deep reinforcement learning, *CoRR*, 2015, vol. abs/1511.06581.

[14] J.F. Zheng, S.R. Mao, Z.Y. Wu, P.C. Kong, and H. Qiang, Improved path planning for indoor patrol robot based on deep reinforcement learning, *Symmetry*, 2022, vol. 14, no. 1, pp. 132-132.

[15] X.F. Yang, Y.L. Shi, W. Liu, H. Ye, W.B. Zhong, and Z.R. Xiang, Global path planning algorithm based on double DQN for multi-tasks amphibious unmanned surface vehicle, *Ocean Engineering*, 2022, vol. 266, no. P1.

[16] M. Guan, F.X. Yang, J.C. Jiao, and X.P. Chen, Research on path planning of mobile robot based on improved Deep Q Network, *Journal of Physics: Conference Series*, 2021, vol. 1820, no. 1, pp. 012024-.

[17] Y.Y. Zhang, X.P. Rao, C.Y. Liu, X.B. Zhang, and Y. Zhou, A cooperative EV charging scheduling strategy based on double deep Q-network and Prioritized experience replay, *Engineering Applications of Artificial Intelligence*, 2023, vol. 118.

[18] E. Erkan, and M.A. Arserim, Mobile robot application with hierarchical start position DQN, *Computational Intelligence and Neuroscience*, 2022, vol. 2022.

[19] Y.B. Chen, D.C. Li, H.G. Zhong, O.W. Zhu, and Z.Q. Zhao, The determination of reward function in AGV motion control based on DQN, *Journal of Physics: Conference Series*, 2022, vol. 2320, no. 1.