The USV Path Planning of Dueling DQN Algorithm Based on Tree Sampling Mechanism

Zhijian Huang*
Lab of Intelligent Control and Computation
Shanghai Maritime University
Shanghai 201306, China
zjhuang@shmtu.edu.cn

Sen Liu
Lab of Intelligent Control and Computation
Shanghai Maritime University
Shanghai 201306, China
202030110138@stu.shmtu.edu.cn

Guichen Zhang*
Lab of Intelligent Control and Computation
Shanghai Maritime University
Shanghai 201306, China
gczhang@shmtu.edu.cn

2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC) | 978-1-6654-0902-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/IPEC54454.2022.9777508

***Abstract*—The path planning and obstacle avoidance of USV (unmanned surface vessel) has become a research hotspot in recent years. Among them, the DQN algorithm has achieved good results in the obstacle avoidance and path planning problems of unmanned surface vessel. However, the algorithm suffers from the problems that the sampling method does not make full use of the stored information and the randomness of action selection during the training process is too large and the convergence is too slow. In this paper, we propose a Dueling DQN algorithm to optimize obstacle avoidance and path planning, which based on tree sampling mechanism. The Dueling DQN algorithm will decomposes the value function Q into a state-value function (V) and a dominance function (A). Meanwhile, the absolute value of TD-error is directly used as a priority indicator for priority sampling in the sampling process. Subsequently, the network model is built and experiments are conducted on each of the four maps. As a result, the convergence steps and loss values of the proposed algorithm on the four paths are better than those of the DQN algorithm. It shows that the dueling DQN algorithm can effectively use the stored information for optimal path planning.**

Keywords—USV (unmanned surface vessel), Dueling DQN (deep Q network), reinforcement learning, path planning

I. INTRODUCTION

At present, obstacle avoidance and path planning, as an important function to ensure the safety and efficient completion of operation tasks of surface unmanned vehicle, has become one of the important research topics of surface unmanned vehicle. Path planning is the process of finding paths between two points in a region while avoiding collisions with objects that exist in the region. An important feature of path planning for USV (unmanned surface vessel) is the uncertainty of operating environment. In complex environment, traditional path planning algorithm is difficult to carry out efficient path planning. Therefore, this paper proposes a path planning algorithm based on reinforcement learning theory.

The path planning algorithm of USV can be divided into global and local path planning. The idea of the global path planning algorithm is to plan the optimal obstacle avoidance path based on the distribution of obstacles in the known environment [1,2]. A* algorithm and Dijkstra algorithm are classic algorithms for solving static path planning optimization problems [3,4]. However, in the complex and changeable water environment, the above algorithms can hardly obtain the optimal performance of the planned path. Intelligent bionic algorithms are used to deal with these highly nonlinear path planning problems, which can effectively reduce path redundancy and loss, and obtain the global optimal path, but the real time ability is still insufficient [5,6].

The development of reinforcement learning provides an effective method for the interaction between an intelligent body and its environment, and its core idea comes from a dynamic programming theory proposed by Bellman, which solves the Markov decision optimal control problem in discrete form, thus laying the solution method and theoretical foundation of reinforcement learning [7-9]. Scholars have used deep neural networks to fit correlation functions in reinforcement learning to process the raw input data from sensors and have developed an end-to-end learning method known as deep reinforcement learning. For example, deep reinforcement learning methods for dual-Q learning reduce the problem of overestimation of execution value function approximation by decoupling execution selection from execution evaluation [10]. The deep reinforcement learning methods for Dueling network architecture explicitly separate the estimation of state values from execution dominance to obtain more accurate policy evaluation [11].

Deep reinforcement learning is very effective in navigation path planning and obstacle avoidance. In 2016, Tai et al. trained an agent based on deep Q network to avoid collision of obstacles in indoor mobile environment [12]. In 2017, they also extended this work to a range-sensor-based unmanned system that achieves continuous control using an asynchronous depth determination strategy gradient approach [13]. In 2018, Kenzo et al. extracted information from color images and derived motion commands to achieve a maples visual navigation system [14]. In 2019, Zhou et al. also proposed a path planning algorithm based on deep reinforcement learning, which provides a new solution for unmanned ships by integrating high-level intelligence [15].

Therefore, combining with the reinforcement learning related algorithms in recent years, this paper proposes an improved reinforcement learning method of deep Q-network algorithm, Dueling DQN algorithm based on tree sampling, and proves the superiority of this algorithm in USV path planning and obstacle avoidance through simulation experiments.

II. DUELING DQN ALGORITHM

A. Dueling DQN Network Model

In this paper, the priority sampling mechanism is

introduced into the deep reinforcement learning algorithm Dueling DQN network. A Dueling DQN model with prioritized sampling is developed.

Dueling DQN uses a memory playback mechanism. In each step, the agent stores the transformation tuple $e_t = (s_t, a_t, r_t, s_{t+1})$ in memory buffer $D = \{e_1, e_2, \ldots, e_t\}$, and then trains by uniformly sampling these samples. The other is to use two Q networks, namely $Q(s,a;\theta)$ and $Q(s,a;\theta^-)$, where θ represents the current parameter and θ^- represents the historical parameter. Each update iteration updates the current parameter to minimize the mean square Behrman error associated with the historical parameter. This process is expressed by minimizing:

$$L_i(\theta_i) = E\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)\right)^2\right] \quad (1)$$

Differentiating the loss function yields the following gradient:

$$\nabla_{\theta_i} L_i(\theta_i) = \left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)\right) \nabla_{\theta_i} Q(s, a; \theta) \quad (2)$$

In this way, the parameters of the neural network can be updated along the direction of decreasing gradient of the loss function:

$$\theta_{i+1} = \theta_i + \alpha \nabla_{\theta_i} L_i(\theta_i) \quad (3)$$

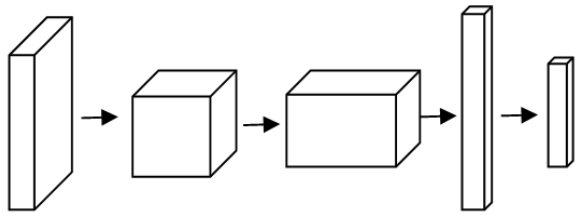

Fig. 1. Classical DQN algorithm Q network architecture

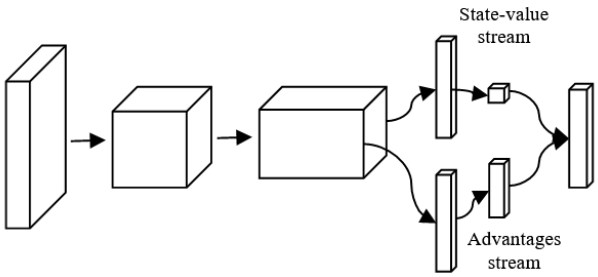

Fig. 2. Dueling DQN networks have two streams to evaluate the state values and the advantages of each action separately

Figure 1 illustrates the DQN network structure. Unlike the traditional DQN structure, the Dueling network structure consists of two streams: one for the state value estimation and the other for the state-independent motion dominance function. The underlying feature learning module is shown in Figure 2. The advantage of this approach using decomposition is that it generalizes the learning between actions without changing the reinforcement learning algorithm. The Dueling network is a single Q-network with two streams. The network can perceive which states are valuable or not without knowing the impact of each action in each state. This is especially important for behavior selection without influencing the environment for each state.

Decompose the value function Q into the state value function (V) and the dominance function (A), as shown in equation (4):

$$Q(s, a|\theta) = V(s, a) + A(s, a|\theta) \quad (4)$$

Among them, the $V(s,a)$ said state-value stream, is the long-term judgment of the current state. the $A(s, a|\theta)$ said advantage flow, is the current measurement state of good or bad behavior.

B. Tree Optimization Sampling Mechanism

The uniform sampling method does not fully utilize the stored information, from which Dueling network can learn more valuable information, and is limited by the memory capacity, and the new memory will overwrite the newly received information. In this paper, the concept of tree-like proportional priority sampling is introduced to improve the convergence speed, and uniform sampling is replaced by proportional preferential sampling in Dueling networks.

The mechanism uses the absolute magnitude of the TD-error as a priority indicator to measure which experiences contribute more to the learning process. TD-error is defined as two value changes. If the TD-error is larger, it means that the experience value used to update has more information and therefore has a higher priority. For a new experience, the network cannot sense the TD-error size, so the design algorithm gives the highest priority to the new experience, ensuring that all experiences are replayed at least once. To implement priority sampling, again, define the sampling probability as:

$$j \sim P(j) = p_j^\alpha / \sum_i p_i^\alpha \quad (5)$$

where $P_j > 0$ is the priority of transfer j. The absolute value of TD-error is used as a direct priority indicator. The index α determines how much priority is used, $\alpha = 0$ corresponds to uniform sampling, and $\alpha = 1$ corresponds to the case of pure greedy sampling.

Importance sampling (IS) is also used to adjust the updated model by reducing the weights of common samples. Define the importance sampling weights w_j as :

$$w_j = (N \cdot P(j))^{-\beta} / \max_i w_i \quad (6)$$

where β compensates for the non-uniform probability $P(j)$. When $\beta = 1$, the importance sampling (IS) weights fully compensate for $P(j)$.

The sampling process uses the "sum tree" data structure (shown in Figure 3), which is a branching structure. Each branch at the bottom of the structure stores the priority of one

memory, the sum of two branches is the priority of the previous branch, and the top branch is the sum of the priorities of all memories in the memory pool. The leaf node is the experience pool information containing the priority metrics, from which the training data are sampled. This data structure simplifies the computation of priority sums and reduces the time complexity of updating and sampling.

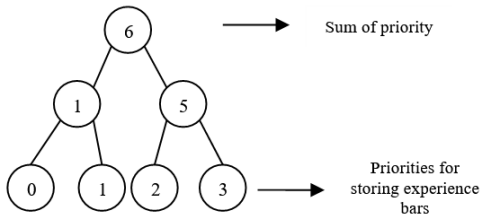

Fig. 3. Structure of the sum tree

C. Algorithm Modeling

In this study, the priority sampling mechanism is introduced into the deep reinforcement learning algorithm Dueling DQN network to establish the Dueling DQN model with priority sampling, and the complete algorithm model is as follows:

1) Initialize the experience pool and environment.

Observe state S_0 and choose action $A_0 \sim \pi(S_0)$ according to the greedy strategy.

USV explores each step to get the experience tuple $(S_{t-1}, A_{t-1}, R_t, \gamma_t, S_t)$.

Calculate TD-error.

$$\delta_j = R_j + \gamma_j Q_{\text{target?}}\left(S_j, \arg\max_a Q(S_j, a)\right) - Q(S_{j-1}, A_{j-1})$$

Calculate the memory storage tuple priority $p_j \leftarrow |\delta_j|$, get the sampling probability $j \sim P(j) = p_j^\alpha / \sum_i p_i^\theta$, calculate the importance weight $w_j = (N \cdot P(j))^{-\beta} / \max_i w_i$, and store the experience tuple to the experience pool with probability P_t.

Sampling from the experience pool according to the sampling probability.

Calculate the label .

$$\gamma_t = \begin{cases} R_t & (\textit{termination status}) \\ R_t + \gamma \max_{A_t} Q(S_t, A_t; \theta, \alpha, \beta) & (\textit{else}) \end{cases}$$

Minimize the cost function based on gradient descent theory.

Replace the Dueling network parameters to the target network as per T step.

The network architecture uses Dueling network, which uses two streams state value and action advantage are calculated separately. Therefore, its space complexity increases, assuming that the action dimension is M. The storage consumption is increased $O(1)+O(M)$, and the total storage consumption is $O(M)$. The sampling and updating time complexity is N in a storage memory pool with capacity $O(\log N)$. The algorithm network structure is shown in Figure 4.

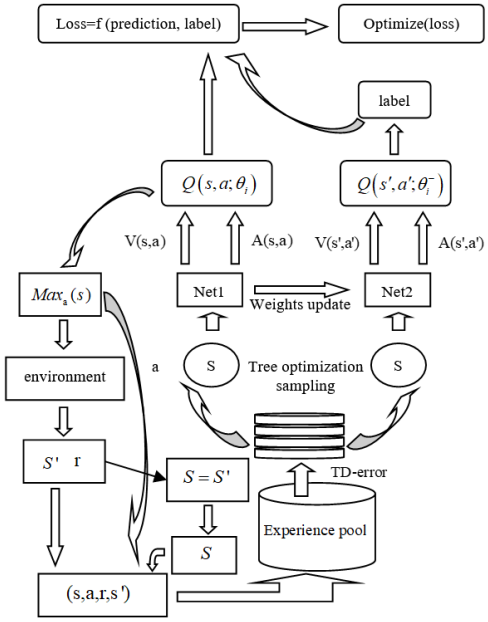

Fig. 4 Network structure of Dueling DQN based on tree sampling mechanism

III. Experimental Design

A. Experimental Process

In this paper, a new path planning algorithm for unmanned surface vessel is proposed by combining optimal sampling method with Dueling DQN. After the neural network input the location information of the agent and processed it, the action with the largest Q value is selected from the action space to execute, to realize the end-to-end autonomous obstacle avoidance control and path planning of the agent. The experimental flow chart is shown in Figure 5.

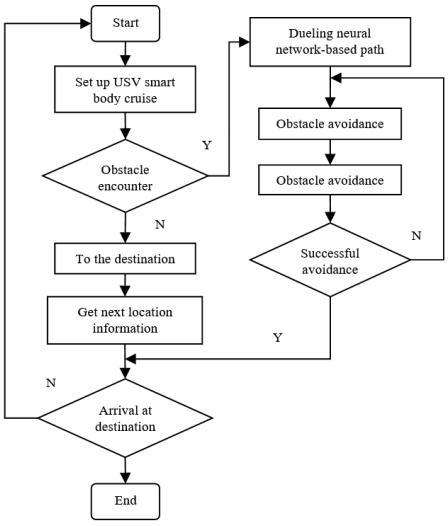

Fig. 5. Path planning flow chart

B. Reward Function

In this experiment, since the action space is finite and discrete, the excitation function can be generalized. When the USV reaches the target point, the reward value is 10; when the USV collides with the black block, it means that the USV does not perform effective obstacle avoidance, and the reward value is set to -10 as a penalty. The reward value for not reaching the target point and not touching the black obstacle is set to 0. The expression of the incentive function is:

$$R = \begin{cases} 10 & (\textit{arrive at the destination}) \\ 0 & (\textit{No collision, no arrival}) \\ -10 & (\textit{have a collision}) \end{cases}$$

C. Environment Construction

In order to verify the effectiveness of the proposed algorithm in USV path planning. In the simulation experiment, maps of four different target points are set for the proposed algorithm, as shown in Figure 6. Using the control variable method, all the hyperparameters are the same as DQN. The network learning rate is 0.0001. In order to keep the weight of the neural network within a reasonable range, the discount coefficient is set as 0.9, and the update rate of the target network is set as 0.01. Action choice uses greedy strategy $\varepsilon-$. The initial value of ε is set to 0.5. The importance of ε sampling starts at 0.4 and increases to 1. The increment of each sample is set to 0.001. The hyperparameter C is set to 0.6. The batch draw is set to 64 and the experience pool size is 3000.

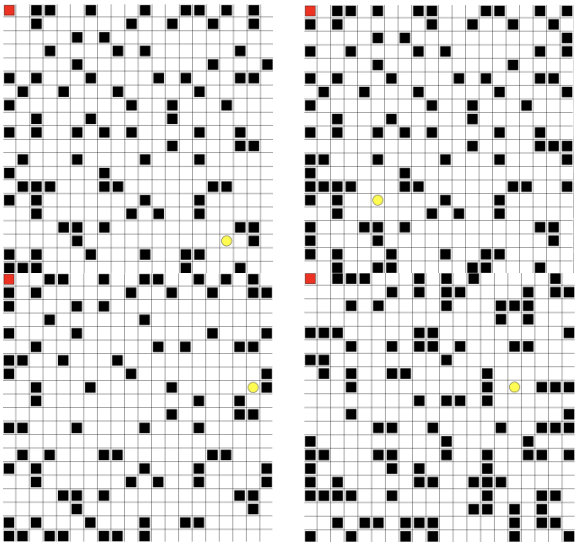

Fig. 6. Simulation environment

Using the current USV location information as input, a four-layer fully connected neural network is built, in which the neurons in the first and second layers of the hidden layer are 50 and 20, respectively. The RMSprop algorithm is used to train the network, and a 1*4 matrix is output, corresponding to the four discrete actions of the USV, respectively, and the action selection is performed according to a greedy strategy.

IV. Experimental Results and Analysis

Experimental configuration: Intel Core i9 (CPU frequency 2.3Ghz, RAM 16GB) and two graphics cards NVIDIA GTX1080 Ti (3584cores, 11GB memories), and Ubuntu 16.04 operating system.

During training, the USV interacts with the environment, and at the same time, the environment gives punishment or reward, thus effectively guiding the USV to train the optimal path. After 10,000 training sessions, the algorithm began to converge and the USV training path began to shorten. After 13,000 times of training, the algorithm gradually converges, the USV gradually stabilizes in the shortest path, and the time consumption is greatly shortened. After 25,000 training sessions, the algorithm basically converges and USV trains an optimal path. Table I shows the average value of data generated by Dueling DQN algorithm in different training time under the above four different paths. The average number of steps required to reach the target under the four different paths were 111.3, 102.4, 170.4 and 116.1, respectively, and the average time consumption was 13.3s, 15.2s, 25.5s and 18.3s, respectively. Table II shows the relevant data of DQN algorithm. As can be seen from the table, the average number of steps in the training process was 175.3, 162.8, 196.2 and 178.1 respectively. The average time to reach the target was 29.1s, 27.2s, 32.3s, 29.8s.

TABLE I. Dueling DQN Algorithm Related Data

DOUBLE DQN	**ROUTE**			
	1	**2**	**3**	**4**
Step	111.3	102.4	170.4	116.1
Time/s	13.3	15.2	25.5	18.3

TABLE II. DQN Algorithm Related Data

DQN	**ROUTE**			
	1	**2**	**3**	**4**
Step	175.3	162.8	196.2	178.1
Time/s	29.1	27.2	32.3	29.8

It can be seen from the data that the Dueling DQN algorithm combined with optimized sampling converges more quickly than the DQN algorithm, thus effectively avoiding obstacles and planning an effective path, and the path diagram is shown in Figure 7.

For the four different path planning methods mentioned above, the Dueling DQN algorithm is compared with the DQN algorithm, and the results are analyzed using the loss value and the cumulative reward value as the evaluation index, and the loss value is shown in Figure 8. It can be seen from the figure that the DQN algorithm is not stable in the late stage, and when the Dueling DQN algorithm converges, the DQN algorithm still has large fluctuations and cannot achieve the effective path planning goal.

From Figure 9, it can be seen that when using the Dueling DQN network for USV path planning, the number of steps required for the algorithm to converge on each of the four paths is about 12,000, 13,000, 14,000, and 15,000. The DQN algorithm requires 13,000 iterations, 14,000 iterations, 17,000 iterations, and 27,000 iterations on each of

the four paths iterations. It can be seen that the proposed algorithm has higher efficiency in the surface unmanned boat obstacle avoidance experiments.

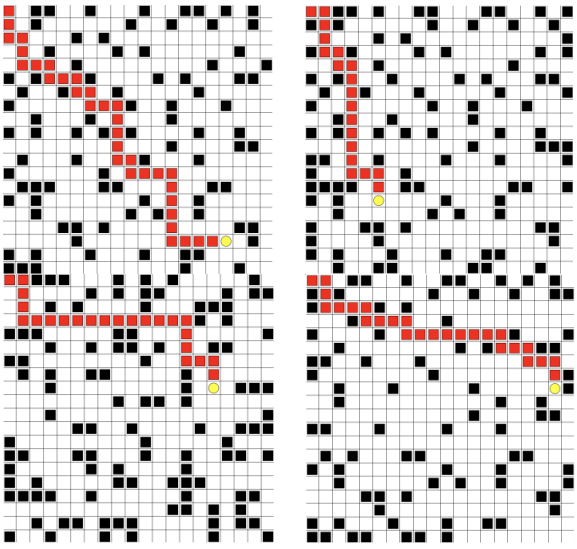

Fig. 7. Four types of path planning based on Dueling DQN network

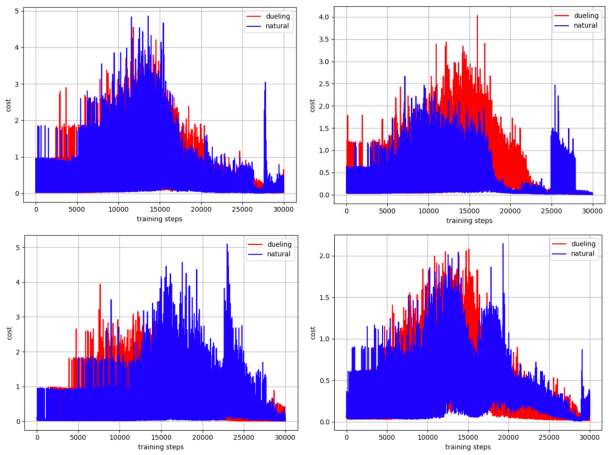

Fig. 8. Comparison of the loss values of the two algorithms

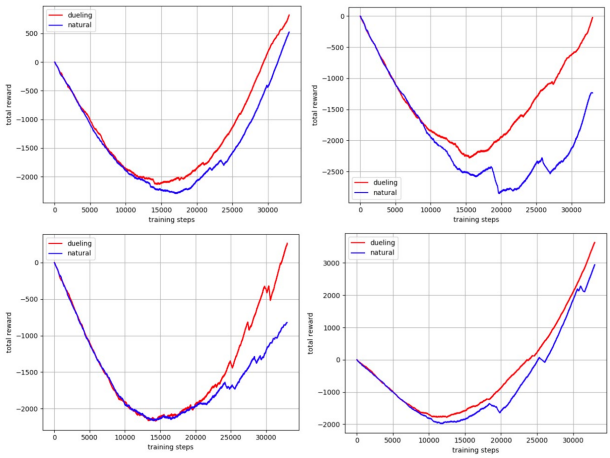

Fig. 9. Comparison of the cumulative reward values of the two algorithms

The above simulation results show that the tree-first sampling-based Dueling DQN network can achieve global path planning for USV and is more efficient than the DQN algorithm, which can effectively avoid obstacles in the environment and converge faster.

V. Conclusion

The tree-based sampling mechanism Dueling DQN proposed in this paper is a model-free end-to-end intelligent body path planning algorithm. The algorithm replaces the Q-value matrix with the output of the neural network and uses a preferential empirical replay mechanism, which solves the problem that previous algorithms Q-learning algorithms have huge Q-tables that cannot be learned as well as improves the learning efficiency of the intelligences. The algorithm obtains data by interacting with the environment without environmental information and thus trains the network until convergence. Compared with the DQN path planning algorithm, the path planning algorithm proposed in this paper is more general comparative tests show that the Dueling DQN algorithm based on the tree sampling mechanism is more efficient and reasonable than the DQN algorithm.

However, there are still many shortcomings in this paper and need to be improved. First, in the simulation experiment stage, the simulation environment used is static, while in the actual situation the USV intelligences need to face a complex and changing ocean environment with more unknown factors. Based on this, in order to improve the practical application of the algorithm, it is necessary to build the real marine environment that can be simulated, and then to conduct the research of obstacle avoidance and path planning algorithms in this environment. Secondly, the proposed algorithm in this paper is currently only in the experimental validation stage and has not been combined with real USV intelligences; therefore, it is not yet possible to evaluate the relevant performance of the proposed algorithm in practical applications.

Acknowledgment

This work is supported by the NSFC of China under grant No.51779136, and NSFC of Shanghai under grant No. 21ZR1426600, No. 21DZ1201004.

References

[1] J. Hong, K. Park. A new mobile robot navigation using a turning point searching algorithm with the consideration of obstacle avoidance. Adv. Manuf. Technol, 2011, 52: 763-775.

[2] P. Yao, H. Wang, H. Ji. Gaussian mixture model and receding horizon control for multiple UAV search in complex environment. Nonlinear Dyn., 2017, 88(2): 903-919.

[3] R. Song, Y. Liu, R. Bucknall. Smoothed A* algorithm for practical unmanned surface vehicle path planning. Appl. Oceans Res, 2019, 83(920).

[4] Y. Singh, S. Sharma, R. Sutton, D. Hatton, A. Khan. Feasibility study of a constrained Dijkstra approach for optimal path planning of an unmanned surface vehicle in a dynamic maritime environment. 18th IEEE International Conference on Autonomous Robot Systems and Competitions, 2018: 117-122.

[5] B. Song, Z. Wang, L. Zou, L. Xu, F. E. Alsaadi. A new approach to smooth global path planning of mobile robots with kinematic constraints. Mach. Learn. Cybern., 2019, 10(1): 107-119.

[6] M. Duguleana, G. Mogan. Neural networks based reinforcement learning for mobile robots obstacle avoidance. Expert Syst. Appl, 2016, 62: 104-115.

[7] Bellman R. Terminal Control, Time Lags, And Dynamaic Programming. Proceedings of The National Academy of Science of The Unite States of Americal. 1957, 43(10): 927-930.

[8] CUI Junxiao, ZHU Mengting, WANG Haiyan et al. Value Iterative Algorithm based on Reinforcement Learning. Computer Knowledge and Technology.2014, 10(31): 7348-7350.

[9] Chen Xiaoqian, Liu Ruixiang. Uav Path Planning Method Based on

Least Square Strategy Iteration. Computer Engineering and Applications.2020, 56(01); 191-195.

[10] Huang H, Lin M, Yang, L T, Zhang Q C. Autonomous Power Management With Double-Q Reinforcement Learning Method. IEEE Transactions on Industrial Informatics, 2020, 16(3): 1938-1946.

[11] Zhang D J, Yu F R, Yang R Z. Blockchain-Based Distributed Software-Defined Vehicular Networks: A Dueling Deep Q-Learning Approach. IEEE Transactions on Cognitive Communications and Networking, 2019, 5(4): 1086-1100.

[12] Tai L, Liu M. Towards cognitive exploration through deep reinforcement learning for mobile robots. ArXiv:1610.01733, 2016.

[13] Tai L, Paolo G, Liu M. Virtual-to-real deep reinforcement learning: continuous control of mobile robots for mapless navigation. ArXiv:1703.00420, 2017.

[14] Kenzo L T, Francisco L, Javier R D S. Visual navigation for biped humanoid robots using deep reinforcement learning. IEEE Robotics and Automation Letters, 2018, 3(4): 3247-3254.

[15] Zhou XY, Wu P, Zhang H F, Guo WH, Liu YC. Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning. IEEE Access, 2019, 7(2019): 165262-165278.