ISSN: 2688-8645

USV Obstacle Avoidance Path Planning Based on Reinforcement Learning

Tengbin Zhu[1]

[1]Shanghai Maritime University, Shanghai 201306, China.

Abstract

In order to realize the adaptive navigation of USV in specific environments, an automatic obstacle avoidance path planning model based on improved Q-Learning algorithm is established. Models were built for the unmanned maneuvering characteristics, intensive learning, and environmental space. In the reinforcement learning algorithm, the Q-value updated method of Q-Learning are improved, the configuration of the award-winning path in the USV avoidance path planning process is optimized, and the algorithm learning efficiency is improved. The action selection is determined according to the action characteristics of the USV; Optimize the path based on the USV cycle. Finally, the simulation environment is established by using MATLAB GUI platform. The simulation results show that the USV can use this intensive learning method to plan a better obstacle avoidance route that accords with the ship's motion characteristics and successfully avoid multiple obstacles.

Keywords

USV; obstacle avoidance; cycle motion characteristics; Reinforcement learning.

1. Introduction

In the 21st century, the ocean has gradually become a battlefield for all countries' comprehensive strength, and the research of unmanned boats has become a hot spot in the development of international science and technology. Western countries have begun research on surface unmanned boats earlier. The United States, Israel and other countries have made many research results in this field. The unmanned boats developed not only apply to the military, but also to civilian development. In recent years, China has carried out some research in the field of unmanned boats and achieved some results. However, in terms of key technology research and development, there is still a large gap with developed western countries. Automatic obstacle avoidance, as the core problem of unmanned boat research, represents the level of intelligence of unmanned boat to a certain extent.

Many scholars (Wu Bo, 2014, YAN Ru-jian, et al.,2010. CACCIA et al.,2008, Gao X., et al.,2011, Lee and Kim, 2016, Pehlivanoglu, 2012.) in the world have conducted in-depth research on the problem of automatic obstacle avoidance paths. For example, Chen Chao et al. (2013) proposed a heuristic search based on viewable A * algorithm to overcome the poor flexibility of traditional viewable methods and improve planning efficiency. Zhuang Jiayuan et al. (2011) proposed the Dijkstra algorithm for distance optimization based on electronic charts to improve the planning accuracy and reduce the memory occupation and operation time. The genetic algorithm proposed by J. Holland (1973) encodes chromosomes for individuals. It is controlled through several processes of selection, crossover, and random mutation. Setting fitness functions is more helpful to find the global optimal solution. Fan Yunsheng et al. (2017) proposed an improved genetic algorithm to generate an initial population for path search through random fast search to improve the convergence efficiency and speed of path planning. Tan Baocheng et al. (2012) used the Euclidean distance between two points as an evaluation function in the A *

ISSN: 2688-8645

algorithm. The forward search and backward search were performed alternately to reduce the path planning time. Chen Zhuo et al. (2019) proposed an unmanned boat path planning algorithm based on an evolutionary potential field model. A potential field path evaluation equation and a differential evolution algorithm were introduced into the potential field model, and a smoothing algorithm was used to optimize the potential field path twice. Zhang Chuang (2016) proposed a two-dimensional particle swarm trajectory planning algorithm based on fuzzy logic, which combined SVM models to simulate different environmental models.

This paper uses the Q-learning reinforcement learning method to solve the problem of obstacle avoidance path planning. This algorithm is an adaptive learning algorithm based on prior knowledge. It has been widely studied and applied in industrial development and expanded to the field of artificial intelligence, which can effectively Improved the intelligent decision-making level of industrial machinery and equipment.

2. Unmanned Swing Model

2.1. Unmanned Boat Plane Coordinate System

For the study of obstacle avoidance path of unmanned boat, only the motion model in the plane coordinate system of three degrees of freedom, that is, longitudinal, lateral and swaying, is needed.

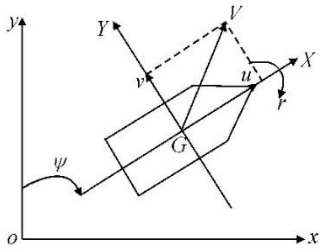

Figure 1. Three-degree-of-freedom plane coordinate system

The coordinate system of the ship's plane motion is shown in Figure 1. Among them, xoy represents the inertial coordinate system, o is the origin of the coordinates. The specified axis x points to the north direction and the axis y points to the east direction. The coordinate system XGY is the appendage coordinate system. The absolute motion of the ship and its surrounding fluid and the interaction force between the fluid and the hull are studied. The axis X points to the bow direction and the axis Y points to the starboard side of the ship. The center of gravity of the ship is represented by G, the origin of the attached coordinate system. The moving speed of a ship V is that the component on the Y axis is decomposed in the attached coordinate system, v is the ship's traverse speed, and the speed of axial direction X is the ship's forward speed u. Indicates the angular velocity r and the heading angle ψ.

2.2. Analysis of Cycle Motion Characteristics

The force of the ship when studying the obstacle avoidance path of the unmanned boat is analyzed by using the MMG model. Unlike the motion trajectory of a car, the direction of movement of the ship cannot be turned directly at 90 degrees, so it will produce a loop, that is, the arc trajectory of the ship under the action of the car and the rudder. The MMG model (Lu Mengmeng et al, 2016) decomposes the force of an unmanned boat into one-way forces acting on the hull, propeller, and rudder. This model is more versatile in a clear physical sense. According to the nature of force generation, the fluid forces and moments on the unmanned boat can be divided into inertial and viscous fluid forces and moments.

ISSN: 2688-8645

According to the simulation of the MMG model, it can be obtained that the trajectory of the fixed-speed rotation of the ship at different rudder angles in still water is approximately circular, as shown in Figure 2.

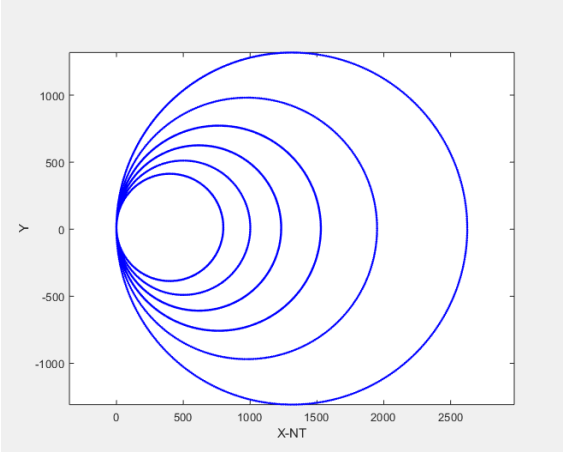

Figure 2. The trajectory of the MMG model cycle

3. Reinforcement Learning Principles and Models

3.1. Markov Decision

Most reinforcement learning algorithms can be performed within the framework of a Markov decision process. Markov's (Yan Zheping et al., 2018 and Kober, 2013) state transition process is a tuple $< S, P, A, R, \gamma >$, which is the state set S, the state transition probability matrix P, the action set A, the return function R and the discount factor γ, $0 \le \gamma \le 1$. The Markov decision process is determined based on the current state of the next state, $P[S_{t+1} \mid S_t] = P[S_t, A_t]$. For a specific state s, specific action a, and its next state s', $P_{ss'}$ is used to represent the state transition probability $s \to s$':

$$P_{ss'} = P[S_{t+1} = s' \mid S_t = s] = P[S_t = s, A_t = a]$$

Assume that there are n states. Then the state transition probability set can be defined by a matrix as

$$P = \begin{bmatrix} P_{11} \cdots P_{1n} \\ \vdots \quad\quad \vdots \\ P_{n1} \cdots P_{nn} \end{bmatrix}$$

The number i row in the matrix indicates the number i state, then the probability of its next state $1,2,3\ldots,n$ is respectively $P_{i1}, P_{i2}, P_{i3}, \cdots P_{in}$. Obviously the sum of all probabilities in this line is 1.

R means that a return value from every steps of a path starting from the state s, and finally reaching the end point after a series of state transitions. According to the calculation formula of cumulative return $G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$, use the strategy π to define the distribution of the selected

ISSN: 2688-8645

action a in a given state s, and define the state-return function of the state s and at the action a as the expected value:

$$q_\pi(s,a) = E_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a]$$

The goal of reinforcement learning is to give a Markov decision process to find the optimal strategy π* so that it can maximizes the cumulative return expectation under this decision, that is:

$$\pi^* = \arg\max q_\pi(a,s)$$

3.2. Improved Q-learning Algorithm

The Q-learning algorithm is a method for solving reinforcement learning control problems using temporal differences. The obstacle avoidance of unmanned boats is considered as a control problem for reinforcement learning. Given six elements of reinforcement learning: state set S, action set A, instant reward R, attenuation factor γ, exploration rate ò, learning rate α, calculating the optimal action value function Q^* and Optimal strategy π^*.

The Q-learning algorithm (Li Guangchuang et al., 2019) is applied to reinforcement learning problem solving without the need for an environmental state transformation model. It is not a model-based reinforcement learning problem solving method. For its control problem solving, it combines dynamic programming and Monte Carlo method for value iteration, that is, updating the strategy by updating the value function, generating new states and instant rewards through the strategy, and then updating the value function. Continuing until the value function and strategy converge.

According to the Q-Learning algorithm, a Q-Table is established to save all states and all actions that will be taken, initialized as $\vec{0}$. In the current state S, ϵ-greedy method is used to select a new action A, and the state is updated after the action is performed as S'; for the value function of the state S', the greedy method is used to take the action a as A' that maximizes Q. Expressed mathematically as

$$Q(S,A) = Q(S,A) + \alpha(R + \gamma maxQ(S',a) - Q(S,A))$$

At this time, the selected action will not be executed, but will only participate in the update of the value function. In the whole algorithm, the values in the Q-table are constantly updated, and continuously updated until the end point is reached. Then based on the updated value to determine what action to take in a certain state is best. When solving the obstacle avoidance path of the unmanned boat, every Q-table update is a process of finding the optimal action, so as to obtain the control of the unmanned boat.

It can be known from the update formula of Q-table that the update of the value Q is based on the judgment error back for learning passed between the real value and the predicted value. In order to improve the update efficiency, an update method $Sarsa(lambda)$ is introduced. Refer to Chen Shenglei et al., 2008, the updated formula for the improved Q value is

ISSN: 2688-8645

$$\delta = R + \gamma * Q(S', A') - Q(S, A)$$
$$E(S, A) = E(S, A) + 1$$
$$Q(s, a) = Q(s, a) + \alpha * \delta * E(s, a)$$
$$E(s, a) = r * lambda * E(s, a)$$

When $lambda$ =0, the method is the Sarsa strategy algorithm. The strength of all step updates remains the same as the largest. E means a matrix similar to Q-table's structure, initialize E to 0 matrix. When agent takes action A and transfers from state S to S' then E(S,A) change from 0 to 1. When $lambda \in (0,1)$, it means the update strength of all steps are different, the larger lambda is the larger update intensity is. Lambda can be understood as the decay value coefficient of the step, that is, the action closer to the end point is more important. When it is larger, the algorithm converges faster, but the optimization capability is insufficient. When it is smaller, it improves the obstacle avoidance path optimization capability but the convergence speed decreases.

3.3. Action Selection Strategy

After setting the initial and end points of the drone, you need to determine the set of action behaviors. Ship motion is a continuous behavior process, so discrete generalization is required. In the simulation process, the search direction of the unmanned boat view is divided into four discrete actions of up, down, left, and right, and the search behavior of the diagonal direction is also added. Taking the mass point of the unmanned boat as the center, the movement space direction model is determined to be clockwise from north to north $\{0, 45°, 90°, 135°, 180°, 225°, 270°, 315°\}$, and the movement step of the up, down, left, and right directions is set to 1, and the movement step of the diagonal direction is set to $\sqrt{2}$. The action model matrix is

$$A = [0\ 1, 1\ 1, 1\ 0, 1\ -1, 0\ -1, -1\ -1, -1\ 0, -1\ 1]$$

In the reinforcement learning system, on the one hand, the unmanned boat needs virtual trial and error to find the optimal search strategy, that is, exploration; on the other hand, it must consider the entire path planning. The unmanned boat has the greatest expectation of receiving rewards, that is, utilization. The ϵ-greedy strategy (Wang Chengbo et al., 2018) is used to preferentially select the search action that maximizes the action value function, while taking into account the possibility of other actions. The probability of choosing this action is

$$\pi(a \mid s) = \begin{cases} 1 - \grave{o} + \dfrac{\grave{o}}{|A(s)|} & \text{, a=argmaxQ(s,a)} \\ \dfrac{\grave{o}}{|A(s)|} & \text{,其它} \end{cases}$$

In the simulation experiment, the next action is selected based on the value $\grave{o}$. A random number is generated first, and an action is selected randomly if the number is smaller than $\grave{o}$; else the larger one is selected to make the largest one, and the value of the search rate is gradually reduced as the exploration progresses $\grave{o} = \grave{o}$/maxtier, maxtier indicating the maximum number of iterations.

After multiple rounds of iterative the Q-table updates are completed, obstacle avoidance path is planned based on the values in the table Q. When there is more than one $Max(Q(S, a))$, the

ISSN: 2688-8645

action is randomly selected in action A (1,3,5,7) since the step length of the action to move up, down, left and right is smaller than the step size of the diagonal movement, It takes more cost to change the direction of the ship in the process of imitating. If the previous action still has the maximum value in the new state, the original action is preferentially maintained.

4. Environment Model Simulation

In this paper, deep reinforcement learning technology is applied to obstacle avoidance path planning for unmanned boats. First, the unmanned boat space environment must be modeled. Based on the MATLAB GUI platform, a two-dimensional simulation environment is constructed, and the environment model is designed as a 40 × 40 two-dimensional map. In the two-dimensional coordinate system, the unmanned boat is regarded as a particle, and each pair of integer coordinate points corresponds to a state of the unmanned boat. Each state can be mapped by an element in the environmental state set S. In the simulation environment model, each state coordinate has a mark, which is 1 or 0, respectively. 0 represents the navigable area, which is displayed as a white area in the environment model; 1 represents the obstacle area, and is displayed as a black area in the environment model.

In the experimental simulation of the specific ship type and navigation environment, the working space was modeled by the cell decomposition method (Su Jintao, 2015), and a large enough sea area including the starting point and the end point of the unmanned boat was divided into rectangular units of equal area. In the working space of the boat, the full rudder rotation diameter at the rated speed is selected as the side length of the unit rectangle divided by the two-dimensional map according to the unmanned boat's maneuvering motion characteristics. Obstacles are inflated. The size of the inflated area can be determined according to the safety area of the ship and the division of rectangular units, so that the unmanned boat can meet real-time requirements and safety requirements during operation. This article simulates the starting ship, the end point, and various irregular obstacles that may be encountered during the voyage in the environmental model. The starting point and the end point are particle-sized, and the obstacles are treated as inflated. The location information of these obstacles is unknown. See Figure 3 for details of the MATLAB GUI simulation environment model.

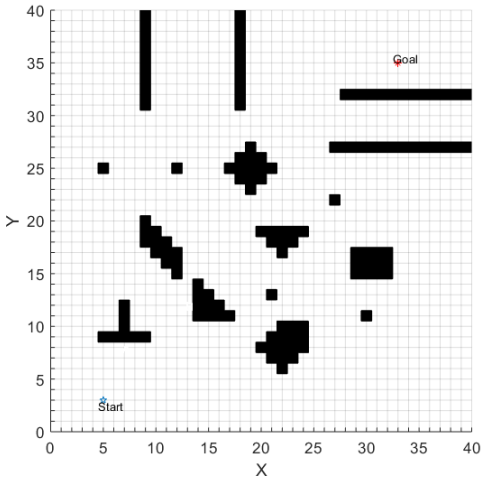

Figure 3. Environmental space simulation

ISSN: 2688-8645

5. Trajectory Optimization of Obstacle Avoidance Path

The movement process of the unmanned boat is a continuous state change process, and the obstacle avoidance path planning algorithm of reinforcement learning discretizes the continuous process in steps. The discrete point trajectory map is calculated and the discrete points are connected. Polyline trajectory diagram, such as the line segment OBDE shown in Figure 4. In the above, the action selection direction is selected at 45-degree intervals, so the corners of the obtained path trajectory are 90 degrees or 145 degrees. According to the maneuvering characteristics of the ship, the unmanned boat's motion trajectory should be a smooth curve, and the obstacle avoidance path trajectory should be optimized according to the ship's cycle motion trajectory map based on the MMG model in Chapter 2.2.

The optimized route is shown as arc OACE in Figure 4(). When the unmanned boat is sailing in the OBC section, a certain rudder angle (δ) is output at point A, which is the length of the half of the rectangular unit in front of point B. Under the action of the rudder, the unmanned boat determines the radius of the half-rectangular unit Speed rotation. It can be seen from the simulation diagram that the unmanned boat turns the trajectory arc AC under the control of the rudder angle δ at point A, and quickly returns to the rudder at the position of point C. When the unmanned boat is sailing in the CDE section, it also starts to make feasible turning trajectory that meets the limited parameters at the long position C of the half of the rectangular unit before reaching point D. The second steering at the anti-rudder angle makes the unmanned boat sail along the plan The line arc CE reaches the rudder at point E. At this time, the radius of the turning circle is (1/tan22.5∘) side lengths of rectangular units.

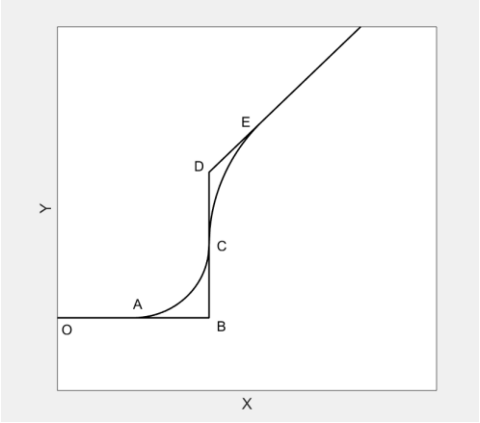

Figure 4. Schematic diagram of route trajectory optimization

6. Experimental Simulation Results and Analysis

Some model parameter settings in the experiment: learning rate alpha = 0.3; discount factor gamma = 0.95; lambda = 0.5; exploration rate epsilon = 0.5; set the maximum number of trials trials = 5000; maximum iterations per attempt (maxiter) = 2000; convergence target (convgoal) The standard deviation of the number of steps to reach the end point is taken as 0.25; the step deviation of the convergence process is calculated according to the number of iterations (avgtrials) is 10; the initial position of the unmanned boat (6, 4) and the end point (33, 35) are set. Once the unmanned boat collides with an obstacle during the experimental iteration process, it will return to the previous step and re-select the action according to the action selection strategy.

ISSN: 2688-8645

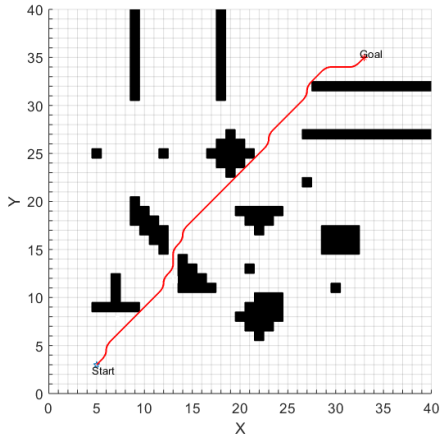

Figure 5. Unmanned obstacle avoidance path

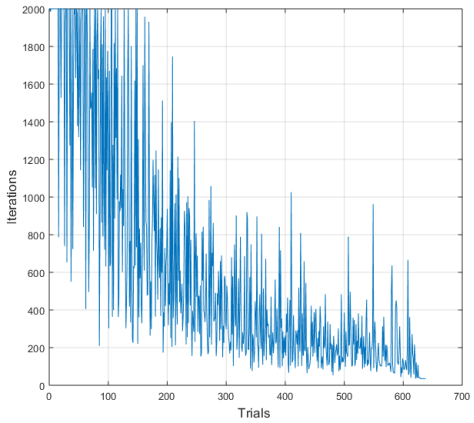

Figure 6. Number of trial and error search iterations

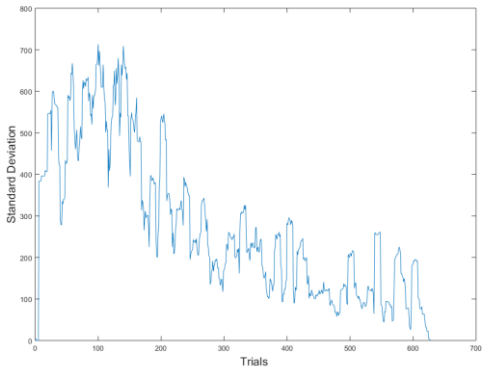

Figure 7. Convergence target change

The results of the obstacle avoidance path planning of the unmanned boat are shown in Figure 5. After learning the reinforcement learning model, the unmanned boat successfully avoided each obstacle and reached the end point. The number of steps of the unmanned boat was 35, which was close to Optimal.

In combination with Figure 6 and Figure 7, in the initial iteration, the unmanned boat can not judge the safe area and obstacle area in the simulation environment, so that it often gets into trouble. At the same time, the search path is random, and it is often unable to find within the specified iteration number After the effective path is obtained, the value of the objective function changes greatly. After 150 iterations, the system gradually plans an effective path. During the process, the collision obstacles still occur many times and the number of planned path steps fluctuates greatly. The overall value of the objective function shows a downward

ISSN: 2688-8645

trend; After 400 to 600 iterations, the collision phenomenon gradually eases and the number of planned path steps fluctuates. The objective function is still decreasing, but it has not reached the set requirements. Until 600 iterations, the probability of random search is gradually approaching the minimum. After the 638th iteration, the requirements of the objective function were reached, and the reinforcement learning system planned the obstacle avoidance path that finally reached the end point.

7. Conclusions and Prospects

The obstacle avoidance path planning model given by this article can comprehensively describe and solve the problem of obstacle avoidance path planning for specific environments. Simulation was performed on the MATLAB GUI platform. In the early stage of interaction with the environment, the unmanned boat knew too little about the state of the environment, and there were collisions and large steps in path planning. As the number of iterations increases, the unmanned boat system accumulates learning experience, gradually adapts to the environment, and finally successfully plans the path and reaches the end. However, to really apply reinforcement learning to unmanned boats, the reinforcement learning algorithm still needs a lot of improvements. Although the improved algorithm has accelerated the calculation efficiency, its convergence speed is still slow, the number of iterations is large, and it takes a lot of time. In the actual voyage, the ship's behavior has complex continuity, and this article is simply divided into 8 Action, this is also the future direction of the unmanned boat reinforcement learning algorithm needs detailed research to improve.

References

[1] Wu Bo, 2014. Autonomous collision avoidance algorithm based on maneuvering characteristics marine USV[D]. Wuhan: Wuhan University of Technology.

[2] Yan Ru-jian, Pang Shuo, Sun Han-bing, et al.,2010. Development and missions of unmanned surface vehicle [J]. Journal of Marine Science and Application, 9(4): 451–457.

[3] CACCIA M, BIBULI M, BONO R, et al.,2008. Basic navigation, guidance and control of an unmanned surface vehicle [J]. Autonomous Robots,25(4): 349–365.

[4] Gao X, Jia Q, Sun H, et al.,2011. Research on path planning for 7-DOF space manipulator to avoid obstacle based on A* algorithm[J]. Sensor Letters, 9(4): 1515-1519.

[5] Lee J, Kim D W, 2016. An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph[J]. Information Sciences, 332(3): 1-18.

[6] Pehlivanoglu Y V, 2012. A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV[J]. Aerospace Science and Technology, 16(1): 47-55.

[7] Chen Chao, Tang Jian, 2013. Path planning and design of surface unmanned boat based on visual method [J]. Shipbuilding of China, (1): 129-135.

[8] Zhuang Jiayuan, Wan Lei, Liao Yulei, et al., 2011. Research on global path planning of unmanned surface watercraft based on electronic charts [J]. Computer Science, 38 (9): 211-214.

[9] Holland J H, 1973. Erratum: Genetic Algorithms and the Optimal Allocation of Trials. [J]. Siam Journal on Computing, 2 (2): 88-105.

[10] Fan Yunsheng, Zhao Yongsheng, Shi Linlong, et al.,2017. Global path planning of unmanned surface craft based on electronic chart rasterization [J]. China Navigation, 40 (1): 47-52.

[11] Tan Baocheng, Wang Pei, 2012. Improvement and Implementation of A * Path Planning Algorithm [J]. Journal of Xi'an University of Technology, 32 (4): 325-329.

[12] Chen Zhuo, Mao Yunsheng, Song Lifei, et al.,2019. Path planning algorithm of unmanned boat based on evolutionary potential field model [J]. Journal of Wuhan University of Technology, 43 (1): 113-117.

ISSN: 2688-8645

[13] Zhang Chuang, 2016. Research on Track Planning and Sliding Mode Control of Under-Driven Ships Based on Integrated Navigation Data [D]. Dalian: Dalian Maritime University.

[14] Lu Mengmeng, Zhang Qiang, 2018. Automatic search mode of ship dynamic fan based on MMG model [J]. Journal of Shandong Jiaotong University, 26 (02): 83-88.

[15] Yan Zheping, Yang Zewen, Wang Lu, et al., 2018. Research status of Markov theory in unmanned systems [J]. China Ship Research, 13 (06): 9-18.

[16] Kober,Jens,J.Andrew Bagnell,et al., 2013. Reinforcement learning in robotics:A survey[C].The International Journal of Robotics Research,1238-1274.

[17] Li Guangchuang, Cheng Lianglun, 2019. Research on obstacle avoidance path planning of robotic arms based on deep reinforcement learning [J]. Software Engineering, 22 (3): 12-15.

[18] Sheng-Lei Chen, Yan-Mei Wei, 2008. Least-Squares SARSA (λ) Algorithms for Reinforcement Learning (C), Natural Computation. ICNC '08. Fourth International Conference on 2008.

[19] Wang Chengbo, Zhang Xinyu, Zou Zhiqiang, Wang Shaobo, 2018. Path planning of unmanned ships based on Q-Learning [J]. Ship and Ocean Engineering, 47 (5): 168-171.

[20] Su Jintao, 2015. Path planning of unmanned surface boat [J]. Command Control and Simulation, 37 (6): 36-40.