A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field

Lingyu Li[a,1], Defeng Wu[a,b,*,2], Youqiang Huang[a,3] and Zhi-Ming Yuan[c,4]

[a]School of Marine Engineering, Jimei University, Xiamen 361021, PRC
[b]Fujian Provincial Key Laboratory of Naval Architecture and Ocean Engineering, Xiamen 361021, PRC
[c]Department of Naval Architecture, Ocean & Marine Engineering, University of Strathclyde, Glasgow, G4 0LZ, United Kingdom

ARTICLE INFO

Keywords:
Deep reinforcement learning
Path planning
Artificial potential field
COLREGS collision avoidance

ABSTRACT

Improving the autopilot capability of ships is particularly important to ensure the safety of maritime navigation.The unmanned surface vessel (USV) with autopilot capability is a development trend of the ship of the future. The objective of this paper is to investigate the path planning problem of USVs in uncertain environments, and a path planning strategy unified with a collision avoidance function based on deep reinforcement learning (DRL) is proposed. A Deep Q-learning network (DQN) is used to continuously interact with the visually simulated environment to obtain experience data, so that the agent learns the best action strategies in the visual simulated environment. To solve the collision avoidance problems that may occur during USV navigation, the location of the obstacle ship is divided into four collision avoidance zones according to the International Regulations for Preventing Collisions at Sea (COLREGS). To obtain an improved DRL algorithm, the artificial potential field (APF) algorithm is utilized to improve the action space and reward function of the DQN algorithm. A simulation experiments is utilized to test the effects of our method in various situations. It is also shown that the enhanced DRL can effectively realize autonomous collision avoidance path planning.

1. Introduction

With the acceleration of globalization, maritime traffic has become increasingly important. The International Maritime Organization (IMO) report indicates that more than 80 percent of maritime accidents are attributed to human decision failures caused by people's misunderstanding of the situation and failure to comply with the International Regulations for Preventing Collisions at Sea (COLREGS). Therefore, enhancing the autopilot capabilities of ships has become an urgent problem to be solved [32, 34]. Furthermore, the marine environment is complex and variable. In some cases, manned ships are not suitable for performing tasks, while USVs are more suitable for coping with variable marine environments [42]. To effectively complete the task, the unmanned ship needs to have highly autonomous path planning and collision avoidance capabilities.

In the practical application of USVs, the conditions of maritime navigation in some areas, such as the coast of Fujian Province and the Taiwan Strait, are complicated. The Taiwan Strait is a maritime passage between Northeast Asian countries, Southeast Asia and countries along the Indian Ocean. The maritime map of the Taiwan Strait is displayed in Fig.1. According to statistics from the Fujian Maritime Safety Administration, there have been 119 collisions in the Taiwan Strait in the past five years. Therefore, the path planning method for USVs requires a comprehensive consideration of static obstacles and dynamic obstacles [10]. Many path planning methods with elimination collision functions have been proposed in the field of USVs, but most of them can only be realized under the conditions of known environmental information, which is contrary to practical applications.

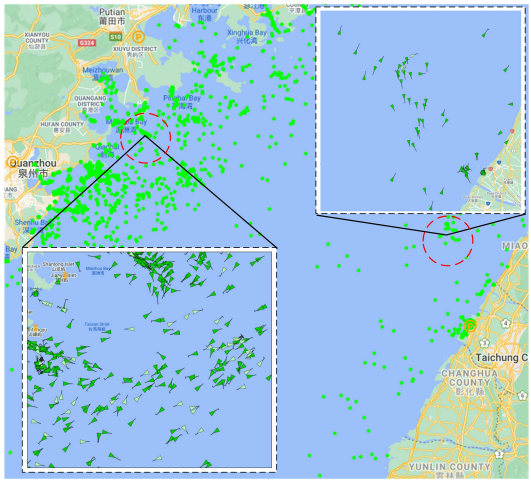

Figure 1: Maritime map of the Taiwan Strait.

There have been many research results on the path planning of ships. An improved A* algorithm-based path planning method was proposed in [2], which implemented the path planning of USVs in a complex environment. Song et al. proposed a USV smooth A* path planning method [21], that used a path smoother to smooth the A* result to provide a more continuous moving path. A path planning method that utilized the V-graph and grid map based on the A* algorithm was proposed in [41]. The method realized path

*Corresponding author:
arcwdf@gmail.com (Defeng Wu)
ORCID(s):

planning by establishing a grid map and designing an optimal path search strategy. Zhang et al. proposed an improved rapid-exploration random tree (RRT) [38], which used adaptive hybrid dynamic step size and adds attractiveness to the target to achieve USV path planning. Particle swarm optimization (PSO) is a commonly used and effective parameter optimize method [33], which can also used in path optimization problem. A PSO-based path planning method was proposed in [6], which established a mathematical model of USVs and marine environments and uses PSO to optimize path generation. An ant colony optimization (ACO) and clustering based algorithm was proposed to settle path planning of the USV in [16]. This method used an improved ACO algorithm to adaptively select a suitable search range, and used a smoothing mechanism to adjust the path to realize global path planning. Wang et al. proposed a membrane evolution artificial potential field method [27], which combined membrane calculation with a genetic algorithm and APF to handle the path planning problem. These methods belong to global static path planning methods based on a prior environmental data. However, global path planning has limitations in practical applications that cannot deal with moving obstacles in a dynamic environment, so it is difficult to avoid collisions with other ships.

The collision avoidance problem in a dynamic environment can be effectively addressed with local dynamic path planning based on local sensor data. In actual navigation, USVs primarily rely on perceptual sensors to obtain surrounding dynamic environment information, such as light detection and ranging (LiDAR), radar and vision sensors. Lyu et al. proposed a method to achieve deterministic path planning for USVs in the dynamic environment [17], that utilized the modified APF to settle the collision avoidance path planning problem. Beser et al. proposed a USV elimination collision path planning method based on Fast-Marching Square algorithm [1], which considered visual guidance aided navigation in the case of partial sensor failure. A velocity obstacle (VO) and improved PSO based method was proposed to achieve local path planning in [35], which converted the local path planning in the continuous space into the multi objective optimization problem under multiple constraints. These methods handle the local collision avoidance path planning problem by establishing the collision avoidance model of ships.

The collision avoidance model of ships has been proposed by many research groups. Goodwin et al. established the ship domain model of open water areas through observation and statistical analysis of open water areas [8]. The ship domain is a virtual safety zone composed of several sectors, and collision avoidance actions are executed when obstacles enter the virtual safety zone. To measure the peril of collision avoidance, Wang et al. proposed a time to close point of approaching (TCPA) and distance of closest point of approaching (DCPA) based risk assessment system to measure the risk of ship collision [28]. Kuwata et al. proposed a COLREGS and VO based method to eliminate collisions [29]. Zhen et al. proposed an elimination collision method using DSCBN cluster analysis of AIS ship data [40], which constructed the collision peril index of ships based on DCPA and TCPA. Xie et al. proposed an improved Beetle Antenna Search (BAS)-based method [36], which referred to the Model Predictive Control (MPC) idea to establish a real-time collision avoidance prediction optimization strategy with COLREGS as a constraint. The model-based method of ship collision avoidance path planning has good effects on the problems of known models. However, with the complexity of modern maritime systems, it is difficult to establish the complete collision avoidance model for many problems, and most model-based algorithms have difficulty predicting the uncertainty in practical applications.

Model-free reinforcement learning methods can adapt to complex systems well by learning the best strategy through interaction with the environment [22]. Q-learning is a classic value-based and model-free RL algorithm [30]. A method based on Q-learning and a neural network planner was proposed in [7], which was used to solve collision avoidance path planning. Chen et al. proposed a Q-learning based path planning method for unmanned ships [3], that obtained the best action strategy by learning the action-state model. However, reinforcement learning (RL) algorithms have insufficient perception of the environment, and it is difficult to explore all action-state information.

Deep reinforcement learning (DRL) combines the perception capability of deep learning (DL) and the decision-making capability of RL [13] and has been diffusely utilized in robot control and decision-making. The deep Q-learning network (DQN) was proposed in [19], which bridged the divide between high-dimensional sensory input and actions and has human-level decision-making capabilities. Subsequently, many researchers have studied the application of deep reinforcement learning in path planning. Woo et al. proposed a DQN-based elimination collision method [31], that included a semi-Markov decision model and neural network architecture specifically designed for USV elimination collision problems. Experimental results have proven that this method can effectively deal with the multi ship collision avoidance problem. However, visual image information was utilized as the input of the DQN, which will result in slower calculation speed. Guo et al. proposed an elimination collision path planning method based on DRL [9], that used automatic ship identification system (AIS) data to train the DRL model. However, many ships, such as small fishing boats and warships, will not install and use AIS. Zhao et al. proposed a proximal policy optimization (PPO)-based elimination collision method of USVs [39], that combined the ship motion mathematical model and COLREGS to achieve autonomous collision avoidance for USVs in a multi-ship environment. However, this method relies on the ship motion mathematical model and PPO is an on-strategy method, which is difficult to migrate to practical applications of ships.

The objective of this paper is to settle the elimination collision path planning conundrum of the USV with known local dynamic environmental information, and a path planning method with COLREGS collision avoidance function

based on DRL and APF is proposed. This method combines the characteristics of DQN and APF, and the APF algorithm idea is used to improve DQN. The DQN action-state model under the constraint of COLREGS is established, and the DQN action space and return function are improved by APF to realize USV path planning and collision avoidance in a dynamic environment. The main contributions of this study are summarized as follows:

(1). A DRL method is designed to handle COLREGS collision avoidance path planning, which can ensure that each action of the USV is the optimal solution in the current state.

(2). Simulated real-time sensor information is chosen as the input data of the DQN, which is used to simulate the practical navigation of the USVs.

(3). The APF algorithm is utilized to improve the action space and reward function of the DQN to solve the sparse reward conundrum.

The rest of the paper is organized as follows. Path planning and COLREGS collision avoidance problems are described in Section 2. Section 3 introduces the collision avoidance path planning method based on DQN and APF. The simulation experimental design and experimental result analysis are presented in Section 4. Conclusions and future work are presented in Section 5.

2. Collision avoidance path planning problem

2.1. Problem definition

Path planning problems are separated into global path planning and local path planning. The path planning and collision avoidance of USVs in the dynamic environment can be described as local path planning problems, which aim to determine the optimal condition in the present state.

To determine the optimal condition in the present state of the USV, a method that is good at solving the local optimal decision-making problem is needed. The DRL method is a combination of DL and RL, which has strong perception and decision-making capabilities, and can provide solutions for the local optimal path planning problem of the USV.

The local optimal path planning problem based on DRL is composed of a reward function and boundary conditions. The state space is represented by the set X, and x is the concrete state of the state set X. $X_{obs} \subset X$ is the set of obstacle structure states that represent forbidden districts of the USV. $X_{path} \subset X$is the set of states which represent feasible districts of the USV. The initial and goal states are respectively represented by $X_{start}$and X_{goal}.The problem definition of local optimal path planning can be delimited as follows:

Definition 1. *The path planning is executed to discover an available path φ, which can be written as: $\varphi : [0, n] \rightarrow \{\varphi(0) = X_{start}, \varphi(n) = X_{goal}\} \in X_{path}, n \in \mathbb{R}$ and $n \geq 1$. Let x_t be the state of the USV at time t, and let x_{t+1} be the next state. The local optimal path planning problem is performed to find the optimal path $\varphi(t) \in \varphi : [0, n]$, which is the collision-free path.*

Definition 2. *For a given local optimal path planning problem (X_{path}, x_t, x_{t+1}). Let $r(t)$ be the reward to the x_{t+1} along the path $\varphi(t)$. The total reward function $r(\varphi)$can hence be formally determined as follows:*

$$r(\varphi) = \int_0^n \varphi(t), \{\varphi(t) \in [0, n], \varphi : [0, n] \rightarrow X_{path}\} \quad (1)$$

2.2. COLREGS

Before applying DRL to solve path planning and obstacle avoidance problems, maritime collision avoidance rules should be considered. COLREGS is a mandatory maritime traffic regulation formulated by IMO to ensure ship safety and reduce ship collisions. The various situations of maritime traffic and the corresponding avoidance directions are stipulated by COLREGS. Therefore, USVs should also formulate actions based on COLREGS to guarantee the security of maritime navigation [26]. According to the COLREGS rule, the relative position of the two ships is divided into four obstacle avoidance strategy areas as displayed in Fig.2.

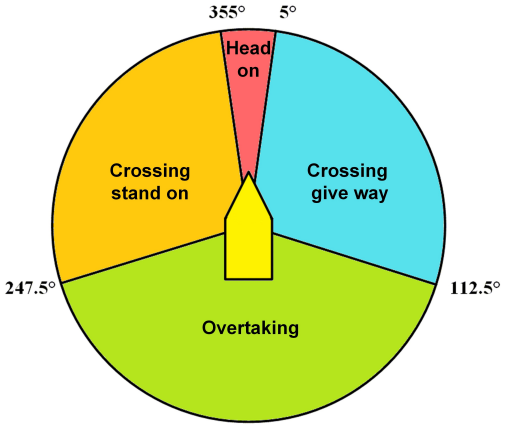

Figure 2: Four collision avoidance action zones divided according to relative position.

The four collision avoidance rules involved in COLREGS Chapter 2 Regulation 13 to 17 are as follows. The corresponding collision avoidance actions are displayed in Fig.3.

(a) Head-on

When the relative azimuth of the target-ship (TS) and own-ship (OS) is in $[355°, 360°]$ or $[0°, 5°]$ and there is a risk of collision, both parties should turn to the right, to pass by the port side of the other ship. The head-on situation is displayed in Fig.3 (a).

(b) Crossing give-way

When two ships meet and there is a risk of collision, the relative position of the target ship and the own ship is in $[5°, 112.5°]$. In this case, the target-ship is a stand-on ship, and the own-ship should give way to the target-ship. According to COLREGS, the own-ship must turn right to avoid collision. The crossing give-way situation is displayed in Fig.3 (b).

(c) Overtaking

When the own-ship chases the target ship in a certain direction 22.5 degrees behind the target-ship, the target-ship is a stand-on ship and the own-ship should give way to the

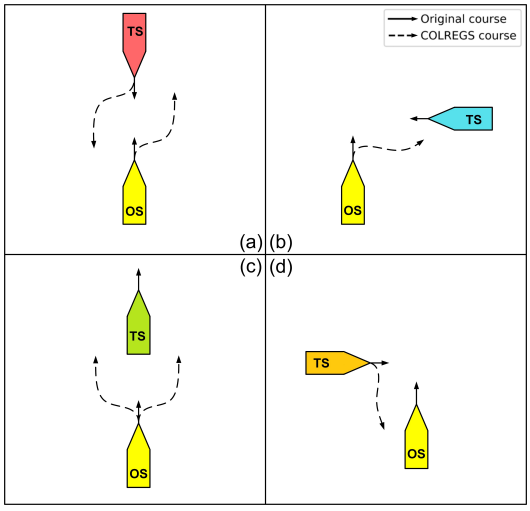

Figure 3: Collision avoidance strategies corresponding to the four situations specified by COLREGS.

target-ship. The overtaking situation is displayed in Fig.3 (c).

(d) Crossing stand-on

When two ships meet and there is a risk of collision, if the relative position of the target ship and the own ship is in [247.5°, 355°]. In this case, the ship is a stand-on ship, and the target-ship should give way to the own-ship. If the target-ship does not take collision avoidance actions, the own-ship should take appropriate collision avoidance actions to prevent collision accidents. The crossing stand-on situation is displayed in Fig.3 (d).

2.3. COLREGS-based multiship collision avoidance

The COLREGS can be extended to scenarios where own-ship encounters multiple target-ships. The multi-ship collision avoidance scenario under the COLREGS can be summarized as displayed in Fig.4.

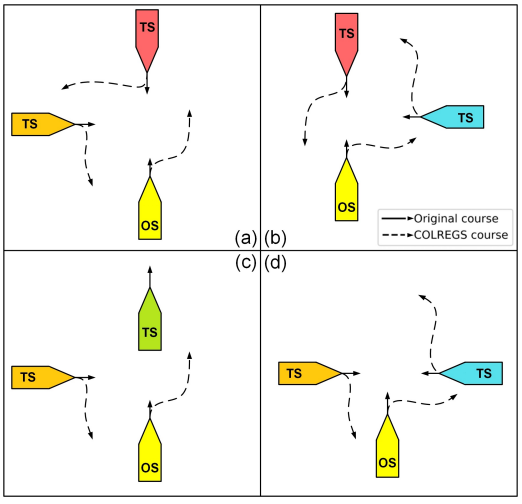

Figure 4: COLREGS based multiship collision avoidance strategies.

In Fig.4(a), Fig.4(b) and Fig.4(d), the own ship encountered two different types of target ships. In these scenarios, they should comply with the COLREGS and move to the right to avoid collision. In the single-ship collision avoidance scenario, the overtaking action described in the COLREGS does not stipulate that the ship must move left or right. However, in the multiship collision avoidance scenario, the overtaking action of the own ship under the COLREGS must move to the right as displayed in Fig.4 (c), otherwise it will collide with the target ship.

3. Method for real-time path planning and collision avoidance based on DQN and APF

3.1. DQN

Reinforcement learning is generally described using the Markov decision process (MDP) [20]. The RL agent takes action a in a certain state s and interacts with the environment to change its state to $s'$and obtain the reward r. The probability of transition from state s to state s' is called state transition probability $P^a_{s\to s'}$. The agent continuously interacts with the circumstance to study, and finally learns the optimal policy to attain the goal. The RL process is displayed in Fig.5. In realistic RL tasks, it is difficult to know the reward function and state transition probability of the environment. If the RL algorithm does not rely on environmental modeling, it is called "model-free learning", which is more challenging than model learning [14].

Q-learning is a model-free RL algorithm, whose main advantage is the utilization of temporal-difference algorithm [24] to achieve off-strategy learning. The Bellman equation is used to solve the optimal strategy of the Q-learning MDP. The final strategy of Q-learning is obtained through the state-action value function $Q(s,a)$, where s' and a' represent the next state and action, and γ represents the attenuation of future reward r'.

$$Q(s,a) = Q(s,a) + \alpha[r' + \gamma \underset{a}{max} Q(s',a') - Q(s,a)] \quad (2)$$

The deep Q-learning network has evolved from Q-learning, which associates Q-learning with DL to learn control strategies directly from high-dimensional data. During the update process, DQN first evaluates the strategy and then improves the strategy. It calculates the rewards of all actions in the next state and selects the action with the largest reward value. The DQN algorithm consists of two deep neural networks which are used to determine the Q-value, and an experience e replay memory $M_t = \{e_1, e_2, \cdots, e_t\}$. The structure of the DQN algorithm is displayed in Fig.6.

During the training process, the current net generates empirical data by interacting with the environment, the target net learns optimization strategies from empirical data. DQN uses the current net to explore and to provide diverse data to continuously optimize the target net. This structure could solve the problem of exploitation and exploration in reinforcement learning [37]. Exploration emphasizes the discovery of more information from known and unknown envi-

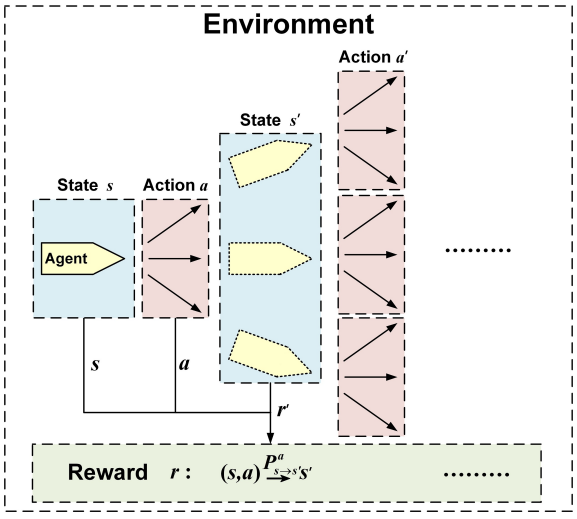

Figure 5: Reinforcement learning process.

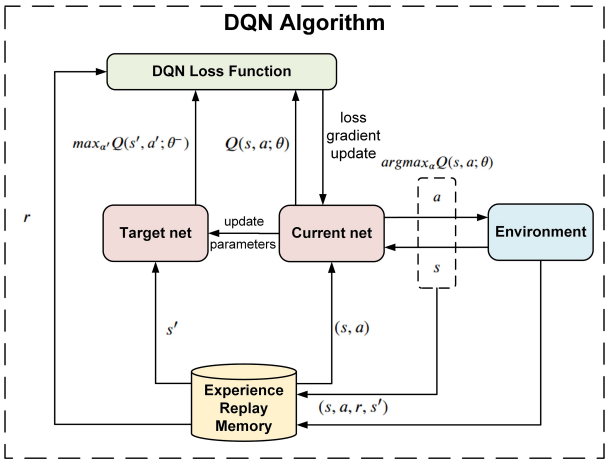

Figure 6: Deep Q-learning network algorithm structure.

ronments, and exploitation emphasizes the maximization of reward from information.

The ϵ-greedy algorithm [25] takes both exploration and exploitation into consideration for selection actions. It randomly selects an action from the action space N_{action} with probability $P = \epsilon$, and selects the action a corresponding to the maximum state-action value function $Q(s, a)$ with probability $P = 1-\epsilon$. The selection action process can be written as follows:

$$a = \begin{cases} \text{random} N_{action}, & P = \epsilon \\ argmaxQ, & P = 1-\epsilon \end{cases} \tag{3}$$

The deep neural network approximates the optimal state-action value function $Q(s, a)$ of the DQN to $Q(s, a; \theta_t)$, where θ_t is the neural network parameter. The update process of θ_t can be written as follows:

$$\theta_{t+1} = \theta_t + \alpha[r + \gamma \underset{a'}{max} Q(s', a', \theta^-) - Q(s, a; \theta)]\nabla Q \tag{4}$$

The loss function of DQN is the residual discrepancy between the true value and the predicted value, which is utilized to iteratively update the action-state value function. The Bellman equation of the loss function $L(\theta)$ can be written as follows:

$$L(\theta) = \mathbb{E}_M[(r + \gamma \underset{a'}{max} Q(s', a', \theta^-) - Q(s, a; \theta))^2] \tag{5}$$

Another feature of DQN is the experience replay memory pool. It stores the experience $e_t = (s_t, a_t, r_t, s_{t+1})$ obtained by the agent interacting with the environment at each step t in the experience pool $M_t = \{e_1, e_2, \cdots, e_t\}$. In the training process, a batch of the experience is randomly selected to train the neural network can reduce the data correlation and enhance the stability of the network. The Bellman equation for the iterative update of the action-state value function can be written as follows:

$$Q_{t+1}(s, a) = \mathbb{E}_{s'}[r + \gamma \underset{a'}{max} Q_t(s', a')] \tag{6}$$

The combination of DQN and APF to achieve collision avoidance path planning will be described in Section 3.3.

3.2. APF

APF is a local path planning method [12], and its basic algorithm idea is to add virtual attraction and repulsion potential fields to the environment. There is a virtual repulsion potential field around the obstacle preventing the agent from moving to the obstacle. The virtual attractive potential field around the target point attracts the agent to approach it. Finally, the combined force of attraction and repulsion is calculated to guide the agent to move.

We use p_a nd p_g to represent the coordinates of the agent and goal. The attractive potential field function $U_{att}(p_a)$ and repulsion field function $U_{rep}(p_a)$ can be written as follows:

$$U_{att}(p_a) = \frac{1}{2}\xi\rho(p_a, p_g)^2 \tag{7}$$

$$U_{rep}(p_a) = \begin{cases} \frac{1}{2}\eta(\frac{1}{D} - \frac{1}{D_{rep}})^2, & D \le D_{rep} \\ 0, & D > D_{rep} \end{cases} \tag{8}$$

Where ξ and η represent the coefficient of attraction and repulsion, and $\rho(p_a, p_g)$ represent the Euclidean distance between the agent and the goal. The Euclidean distance between the agent and the obstacle is represented by D, and D_{rep} represents the distance threshold of the repulsion potential field.

The total potential field $U(p_a)$ and the resultant force F can be written as follows:

$$U(p_a) = U_{rep}(p_a) + U_{att}(p_a) \tag{9}$$

$$F = -\nabla U(p_a) \tag{10}$$

3.3. Proposed APF-DQN for collision avoidance path planning

3.3.1. Modified algorithm structure

In practical applications, the USV cannot obtain complete prior information of the environment, it can only obtain environmental information within certain range centered on itself through various sensors. The maximum detection range of sensors is set to D_{max}. We assume that the environmental information in the range can be completely detected by the sensors. The ship domain is the effective area around a ship which a navigator would like to keep free with respect to other ships and stationary obstacles [4]. There are three definitions of safety criterion in ship domain: own domain not violated, target domain not violated and domains not overlapping. The most commonly used safety criterion is that the own ship domain has not been violated [23]. Therefore, the ship domain is based-on the safety criterion of own ship domain not violated and simplified into circular domain. It is mainly used to evaluate the risk of collision, calculate the reward of the action of the DRL agent and as an event trigger mechanism to mandatory collision avoidance actions. The design of the circular domain is displayed in Fig.7.

In order to evaluate the risk of collision and calculate the reward of the action of the DRL, the circular domain is divided into safe zone, collision avoidance zone and mandatory collision avoidance zone. The minimum safe encounter distance for ship is 2 nautical miles. The advance distance is the minimum distance for the ship to avoid collision, which is generally set to 6 8 times the length of the ship L. If the ship length is 100 meters, the collision avoidance distance threshold should be: $8L + 2NM = 800m + 3704m \approx 45L$. Therefore, the threshold of USV circular domain is set to 45 times the USV length and USV is calculated as a mass point. The thresholds of the collision avoidance zone and the mandatory collision avoidance zone are represented by D_{cz} and D_{mz} respectively. When the obstacle is in safe zone, there is no risk of collision between own ship and the obstacle. When the obstacle is in collision avoidance zone $D_{mz} \leq D < D_{cz}$, own ship needs to avoid collision to ensure navigation safety. When the obstacle is in mandatory collision avoidance zone $D < D_{mz}$, it will trigger the mandatory collision avoidance action of own ship to ensure navigation safety.

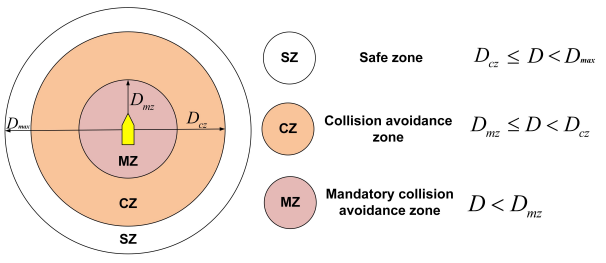

Figure 7: Circular domain around the agent.

After obtaining prior information of the local environment, local collision avoidance path planning can be realized. The conundrum of collision avoidance path planning can be solved by the DQN algorithm and APF algorithm. The advantage of the DQN algorithm is that it can realize collision avoidance path planning without the aid of environmental prior information, but it also has the problem of sparse reward [5]. APF considers the advantages of simplicity, effectiveness and excellent real-time capability, but only using the APF algorithm will have the problem of local minimum and unreachable location. Therefore, we use APF to improve the DQN algorithm, so that the improved algorithm adopt the advantages of APF and improves the defects of DQN. The improved algorithm structure of APF-DQN is displayed in Fig.8.

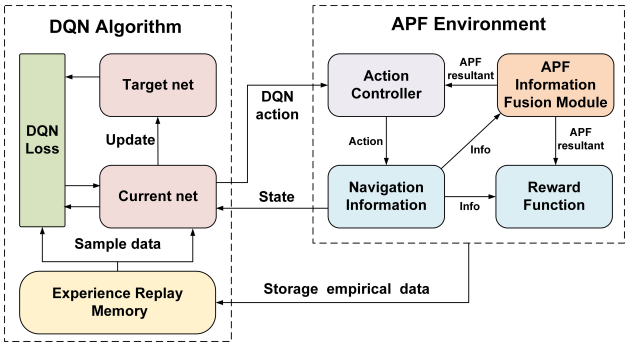

Figure 8: Improved algorithm structure combining DQN and APF.

3.3.2. Improved collision avoidance action

The DQN algorithm has a good effect on the problem of discrete action space, and it is troublesome to handle the defect of continuous action space. To solve the collision avoidance problem of USV path planning, the action space of DQN was modified to a unit vector that only represents the direction of movement. The APF result is mapped to the action step coefficient τ, and the final action step is obtained by multiplying the basic action step v_{base} and the action step coefficient τ.The modified action space allows the continuous action of the USV to be calculated as discrete values in DQN. In addition, the APF repulsion potential field function has been improved according to the collision avoidance area design. The repulsion potential field function can be written as follows:

$$U_{rep}(p_a) = \begin{cases} 0, & D_{cz} \leq D < D_{max} \\ \frac{1}{2}\eta_{cz}(\frac{1}{D-D_{mz}} - \frac{1}{D_{cz}})^2\rho^2, & D_{mz} \leq D < D_{cz} \\ \frac{1}{2}\eta_{mz}(\frac{1}{D} - \frac{1}{D_{mz}})^2\rho^2, & D < D_{mz} \end{cases} \tag{11}$$

where η_{cz} and η_{mz} represent the repulsion coefficient of the collision avoidance zone and the mandatory collision avoidance zone of the repulsion potential function.

When the obstacle is in the safe zone or no obstacle is detected, the agent is only affected by the attractive potential field of the target point. In this case, the moving direction

of the agent is obtained by adding the DQN output action direction $\overrightarrow{A_q}$ and gravity direction $\overrightarrow{F_a}$, and the action of the agent can be written as follows:

$$A = (\overrightarrow{A_q} + \overrightarrow{F_a})v_{base} \tag{12}$$

When the obstacle is in the collision avoidance zone, the agent is affected by the repulsion potential field of the obstacle and the attractive potential field of the target point. At this time, the magnitude and direction of the resultant force obtained by the agent are divided into F and $\overrightarrow{F}$. The sigmoid activation function is used to map the resultant force to the action step coefficient. It is a neural network activation function that can map any real number to (0, 1). The sigmoid activation function can be written as follows:

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{13}$$

The value of action step coefficient τ after being mapped by the sigmoid function is in the interval (0.5, 1), and the calculation process of the action step coefficient can be written as follows:

$$\tau = sigmoid(F) \tag{14}$$

The final action step is 1 to 1.5 times the basic action step. The action calculation process can be written as follows:

$$A = (\tau + \frac{1}{2})(\overrightarrow{A_q} + \overrightarrow{F_a})v_{base} \tag{15}$$

When the obstacle is in the mandatory collision avoidance zone, the agent has a high risk of collision with the obstacle. To prevent collisions, the agent is forced to execute collision avoidance direction $\overrightarrow{A_c}$ that conforms to the COLREGS so that the agent could gradually approach the direction that conforms to COLREGS. The action of the agent can be written as follows:

$$A = (\frac{1}{2}\overrightarrow{A_q} + \overrightarrow{A_C})v_{base} \tag{16}$$

3.3.3. Improved reward function

The reward function is utilized to evaluate the actions of the agent. However, when traditional DQN is used for path planning and collision avoidance, the agent can only obtain positive and negative sparse reward functions by reaching the target point and colliding with obstacles. Therefore, the other actions will not receive any positive or negative feedback, and most of the data cannot reflect its own quality. The model will not receive any feedback until it receives the first reward, so it may stop learning and fail to improve.

To settle the sparse reward conundrum of the DQN, the reward function R has been improved. The reward function is divided into the normal action reward function R_a, collision avoidance action reward function R_{ca}, and end reward R_{end}.The modified reward function can be written as follows:

$$R = \begin{cases} R_a, & D \geq D_{cz} \\ R_{ca}, & D_{mz} \leq D < D_{cz} \\ R_{ca}, & 0 < D < D_{mz} \\ R_{end}, & \text{Other} \end{cases} \tag{17}$$

The normal action reward function is calculated based on the distance between the point of the agent and target and the APF attractive potential field. Taking the inertial coordinates as the benchmark, the distance between the initial point of the agent and target is represented by d_{max}, the distance between the current position of the agent and target is represented by d_{goal}, and the direction angle of the attraction of APF to the agent and the current direction angle of the agent are respectively represented by ϕ_{att} and ϕ_a.The reward function value of normal action gradually increases as the distance between the point of the agent and target decreases. The normal action reward function can be written as follows:

$$R_a = sigmoid(|\frac{\phi_a - \phi_{att}}{\phi_{att}}|)\frac{d_{max} - d_{goal}}{d_{max}} \tag{18}$$

The collision avoidance reward function is affected by APF results and COLREGS actions. When the obstacle is in the collision avoidance zone or the mandatory collision avoidance zone, the agent is simultaneously affected by the repulsion of the obstacle and the attraction of the target point. The direction angle of the resultant force received by the agent is ϕ_F. At this time, the reward function value decreases as the distance between the agent and obstacle increases. The collision avoidance reward function can be written as follows:

$$R_{ca} = \begin{cases} R_a, & \text{OS stand-on} \\ R_{give-way}, & \text{OS give-way} \end{cases} \tag{19}$$

$$R_{give-way} = \begin{cases} \frac{1}{2}(\lambda_1 + \lambda_2)\frac{D}{D_{cz}}, & D_{mz} \leq D_{cz} \\ \frac{1}{2}\lambda_2\frac{D}{D_{mz}}, & 0 \leq D < D_{mz} \end{cases} \tag{20}$$

$$\lambda_1 = sigmoid(|\frac{\phi_a - \phi_{att}}{\phi_{att}}|) \tag{21}$$

$$\lambda_2 = sigmoid(|\frac{\phi_a - \phi_F}{\phi_F}|) \tag{22}$$

The modified reward function converts the reward value of each action-state into a continuous value between (0, 1), which solves the sparse reward problem. The calculation process of the reward function is displayed in Fig.9.

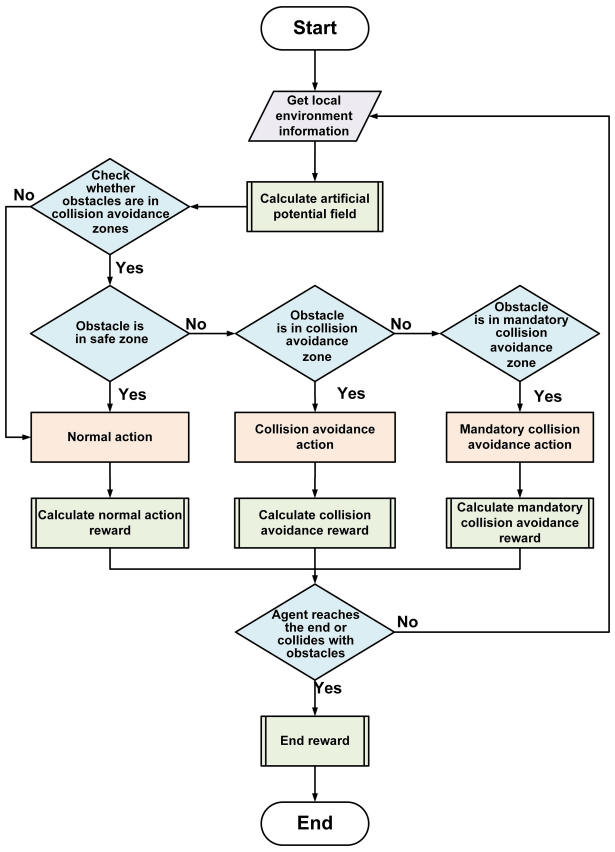

Figure 9: Flow chart of reward function.

Remark 1: The three methods of DQN, Deep Deterministic Policy Gradient (DDPG) [15] and Asynchronous Advantage Actor-Critic (A3C) [18] have their own advantages and disadvantages in solving problems, the deficiencies of RL are improved from different points of view. DQN is effective in solving discrete action space, but it performs poorly in continuous action space. DDPG has a high algorithmic similarity with DQN, it can be regarded as the expansion of DQN in the continuous action space. It has a good effect on solving continuous space problems, but the deterministic policy will lead to insufficient exploration of the action-state space. A3C algorithm is an asynchronous and concurrent RL method, which has good effect when training in multi-threaded GPU clusters. Navigation strategy can be expressed in discrete or continuous action space, but in most situations, the navigation strategy of ship can be expressed in a limited action space. Taking these factors into consideration, DQN algorithm is chosen to improve from another point of view.

4. Simulation experimental results

To verify the utility and feasibility of the collision avoidance path planning method based on DQN and APF, the TensorFlow framework and Python are used to build the algorithm model, and Python GUI is used to create a visual simulation environment that can observe the training and verification process in real time.The visual simulation environment and algorithm model, as well as the analysis of experimental results are introduced in this chapter. The rest of this chapter is organized as follows. The visual simulation environment and model design are introduced in Section 4.1.The experimental results are presented in Section 4.2. Application of the algorithm model in the environment of multiple moving obstacles will be introduced in Section 4.3.

4.1. Simulation environment and DRL model design

4.1.1. Visual training and verification simulation environment

The visual simulation environment is designed as a top view map of 500*500 meters, and the motion of the agent is also calculated in meters. The action scope is limited to the map. If the agent moves beyond the boundary of the map, it will be considered a collision, and the environment will be initialized. Static and dynamic obstacles are utilized to simulate islands and other ships encountered during the navigation of the USV.

The black polygon in the simulation environment is used to simulate the island. If the agent moves to the black polygon, it means a collision occurs and the environment will be initialized. The goal is represented by a green circle in the simulation environment. When the agent reaches the goal, the environment will be initialized. In the simulation environment the agent and the obstacle ship are represented by blue and red boat-shaped patterns, and the collision avoidance zone and mandatory collision avoidance zone of the agent are represented by blue dashed lines.

4.1.2. DRL model design and training

In the action space design, we assume that the USV heading angle is 0°, and the action space is designed to be nine different action directions as:

$$N_{action} = \{-60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°\}$$

In practical applications, the USV can only collect real-time environmental information through sensors. To simulate the navigation of the USV, we assume that the USV can perceive 360° environmental information within a certain distance, which is utilized as the input of the APF-DQN.

The APF-DQN consists of two neural networks, the current net and target net with three hidden layers. The structure of the deep neural network is displayed in Fig.10. The input data consists of three parts: the 360∘ environmental information scanned by the sensor, the distance and angle between the USV and the destination and the location of USV. The environmental information includes the distance and angle data of the surrounding objects of the USV, which type is a two-dimensional array of shape $(2, 360)$. The location of USV and destination data type are two-dimensional array of shape $(2, 1)$. The data will be flattened into one dimension before being input into the neural network, and the result of the action will be output after being calculated by the fully connected layer.

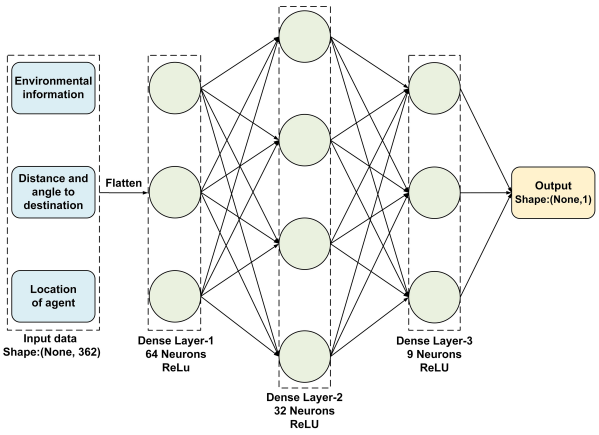

Figure 10: Neural network structure of the current net and the target net.

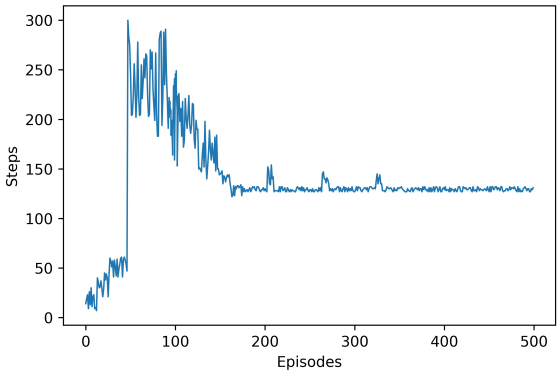

Figure 11: Number of steps in each training episode.

The neural network hyperparameters of APF-DQN are designed through experiments. The hyperparameters of the neural network are displayed in Table 1.The higher the reward decay rate γ, the greater the agent pays attention to future actions.The time for the objective function to converge to the minimum and whether it can converge to the minimum are both determined by the learning rate lr. To maintain a better learning effect, the learning rate is set to 0.01 at the beginning of training. Then, the Adam algorithm is used as an optimizer of the learning rate [11] to achieve adaptive update of the learning rate. To maintain a certain exploratory behaviour of the agent, the parameter ϵ of the ϵ-greedy algorithm is set to 0.1. The maximum size of the experience replay memory pool is set to 5000, and the batch size of experience replay learning is set to 64. The APF-DQN neural network has a delayed parameter update mechanism. When the parameters of the current net are updated c times, the target net replicates the parameters of the current net and updates it once. The target net update interval is set to 100.

Table 1
Hyper parameters of the DQN training algorithm.

Hyper parameter	Symbol	Value
Reward decay rate	γ	0.9
Learning rate	lr	0.01
ϵ-greedy	ϵ	0.1
Experience replay memory	M_{max}	5000
Replay batch size	b	64
Target net update interval	c	100

The maximum number of action of the agent in a single training session is 300, and the reward and penalty values for reaching the goal and collision are set to 200 and -200. The model learns the action strategy by continuously interacting with the environment, and the learning effect is represented by the cumulative reward value of each training episode.

The number of steps in each training episode is displayed in Fig.11. The curve of the first 60 training episodes shows that the number of steps in a single training of the model is rather small. It can be inferred that the agent touched an environmental obstacle near the starting point or moved beyond the map boundary to trigger the training termination condition. At approximately about the 60th training, the action steps of the agent reached the maximum, which triggered the training termination condition. Therefore, it can be inferred that the agent has learned how to avoid collisions with static obstacles. The increased number of training steps means that the agent has learned more action strategies. After approximately the 65th training episode, the number of steps in each episode of training began to decrease. It can be inferred that the agent is constantly trying to explore the environment randomly at this time. After approximately 160 training episodes, the single training steps of the model remain at 130 steps, which suggests that the agent has found a path to the goal. After 200 training episodes, three obvious fluctuations appeared in the moving step curve, which may be caused by the exploratory strategy of the ϵ-greedy algorithm or the change of the collision avoidance path.

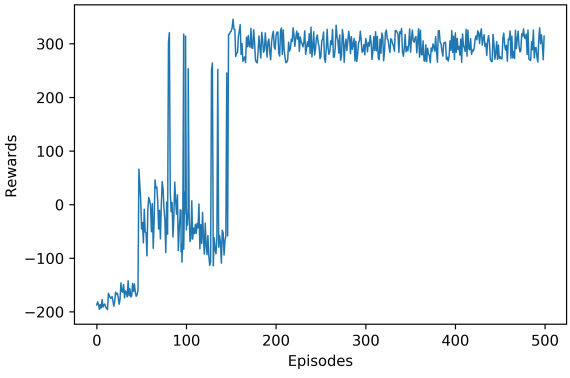

Figure 12: Total reward for each episode.

The total reward for each training episode is displayed in Fig.12.Combining the results of Fig.11 and Fig.12 can verify the above inference of the training process. The curve of the first 60 training episodes shows that the total reward

value approaches the penalty value, indicating that the agent triggered the training failure mechanism. Between 60 and 80 training sessions, the total reward value fluctuates to approximately 0. The experimental results show that the model is trying random exploratory actions and triggers the training termination condition. Between 60 and 80 training sessions, according to the total reward and the number of action steps, the model is trying a different strategy. The agent reaches the goal through some action strategies, and the training failure constraint is triggered by the remaining action strategies. After approximately 160 training episodes, the reward value curve fluctuates to approximately 300, and the result shows that the agent reaches the goal and triggers the reward mechanism. According to the total reward curve, it can be seen that the fluctuation of the action step has no obvious effect on the reward. The result shows that the agent has also reached the goal, which verifies the above hypothetical analysis of the fluctuation of the action step.

4.2. Experimental results

To verify the effect of APF-DQN on path planning and collision avoidance problems. The APF-DQN, DQN, DDPG, RRT, A* and APF methods are used for path planning comparison experiments. The experimental results will be displayed in the visual simulation environment.

4.2.1. Path planning experiment

Two groups of starting and ending points were utilized to test the ability of the six methods to handle path planning problems under the same conditions to verify the capability of the trained algorithm model. According to the characteristics of algorithms, they can be divided into two categories: DRL-based methods (DQN, APF-DQN and DDPG) and classic methods (APF, RRT and A*).

The start and goal point coordinates of the first set of experiments are set at both ends of the simulation environment, and the goal point is placed beside the environmental obstacle. The purpose of the experimental design is to simulate the route planning mission of the USV sailing to the shore. The experimental results of the first group are displayed in Fig.13, which includes the path planning trajectory, the distance to the destination and the displacement distance.

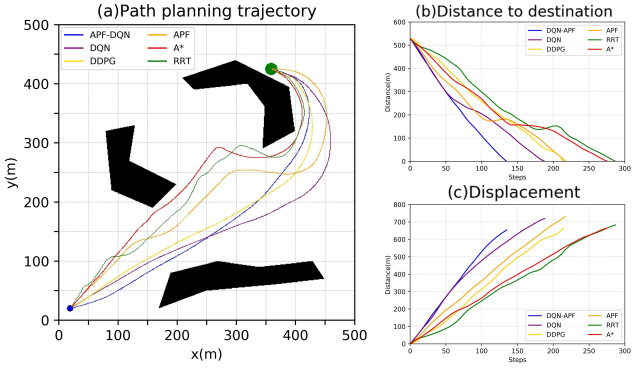

Figure 13: The first group of six path planning methods simulation experiment results.

Experimental results show that APF-DQN method uses the fewest moving steps and shortest distance to reach the destination in the first group, and the DRL-based method performs better than the classic method. It can be seen from Fig.13 (a) that the three classic methods perform better than the DRL-based methods in the front part of the trajectory. However, the performance of classic algorithms is poor in the latter part of the trajectory, which is caused by their algorithm characteristics. RRT and A* are respectively a sampling based method and a search based method, their sampling or searching node behavior is hard to generate smooth and optimal trajectories. The APF method is affected by the attraction of the destination and the repulsion of obstacles, so that the agent can only move in the direction where the attractive force is greater than the repulsive force. In three DRL-based methods, it can be seen from Fig.13 (a) and Fig.13 (c) that DDPG and APF-DQN achieved similar results on trajectories and distances, and both are better than DQN method. Compared with the DQN method, the improved action space and reward function of APF-DQN can effectively guide the path planning trajectory of the agent.

The start and goal points of the second set of experiments are set beside the same environmental obstacle. The experimental results of the second group are shown in Fig.14, which includes the path planning trajectory, the distance to the destination and the displacement distance. The experimental results show that although the trajectory of the APF-DQN method is different from other trajectories, the trajectory displacement is similar to RRT, A*, DDPG and DQN, and the comprehensive result is the best. The reason for the poor results of the APF method is that the obstacle surround the destination, and the repulsive force field of the obstacle directly acts on the agent, making it unable to approach the obstacle. The agent can only move in the direction where the attraction field is greater than the repulsion field.

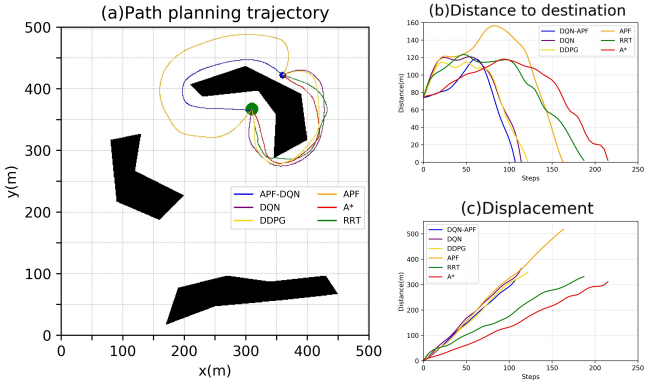

Figure 14: The second group of six path planning methods simulation experiment results.

Two sets of experimental results show that the APF-DQN method can effectively solve the path planning problem. The experimental results from Fig.13 and Fig.14 proved that the DRL-based method performs better on the global trajectory, and the continuous action space DRL method is better than the discrete action space DRL method for path planning prob-

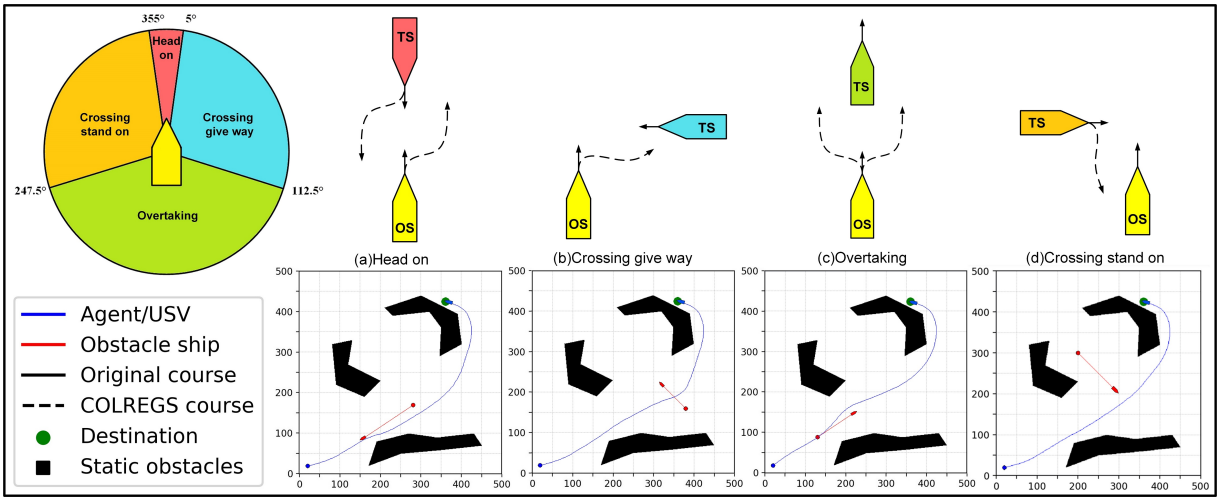

Figure 15: APF-DQN COLREGS collision avoidance experimental results.

lems.

4.2.2. Collision avoidance experiments

The collision avoidance experiments were designed to verify whether APF-DQN can achieve COLREGS-compliant collision avoidance while implementing path planning. The four COLREGS situations and experimental results are illustrated in Fig.15.

In the head-on collision avoidance simulation experiment, the course of the obstacle ship is set to face the USV. The head-on collision avoidance path planning trajectory is demonstrated in Fig.16 (a). From the navigation trajectory results, it can be seen that the USV turned right to avoid the obstacle after detecting it. The head-on collision avoidance trajectory snapshot is displayed in Fig.16(b). It can be seen that when the obstacle ship is detected by the USV, their relative position is in line with the head-on situation. The USV is a give-way ship and the obstacle ship is a stand-on ship; hence, the USV moves to the right to make way for the obstacle ship. The collision avoidance operation of the USV complies with COLREGS Chapter 2 Regulation 14.

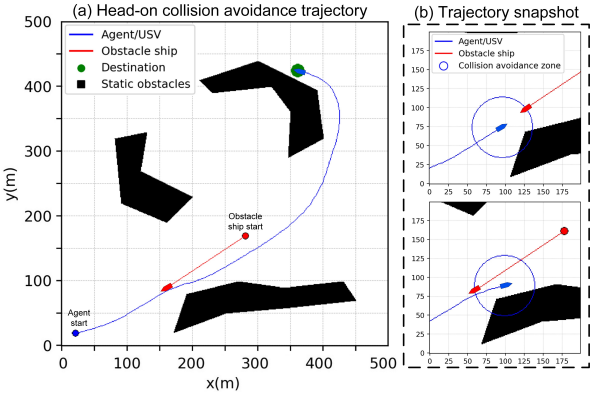

Figure 16: Head-on situation experiment result.

In the crossing give-way collision avoidance simulation experiment, the course of the obstacle ship is set to the right side of the USV. The crossing give-way collision avoidance path planning trajectory is demonstrated in Fig.17 (a). From the navigation trajectory results, it can be seen that the USV turned right to avoid the obstacle after detecting it. The crossing give-way collision avoidance trajectory snapshot is displayed in Fig.17 (b). It can be seen that when the obstacle ship is detected by the USV, their relative position is in line with the crossing give-way situation. The USV is a give-way ship, and the obstacle ship is a stand-on ship; hence, the USV moves to the right to make way for the obstacle ship. The collision avoidance operation of the USV complies with COLREGS Chapter 2 Regulation 15 and 17.

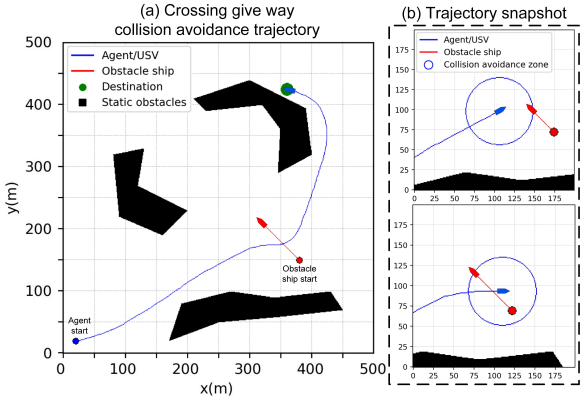

Figure 17: Crossing give way situation experimental result.

In the overtaking collision avoidance simulation experiment, the course of the obstacle ship is set to travel in the same direction as the USV. The overtaking collision avoidance path planning trajectory is shown in Fig.18 (a). From the navigation trajectory results, it can be seen that the USV turned left to avoid the obstacle after detecting it. The overtaking collision avoidance trajectory snapshot is illustrated in Fig.18 (b). It can be seen that when the obstacle ship is

detected by the USV, their relative position is in line with the overtaking situation. The USV is a give-way ship and the obstacle ship is a stand-on ship; hence, the USV moves to the left to make way for the obstacle ship. The collision avoidance operation of the USV complies with COLREGS Chapter 2 Regulation 13.

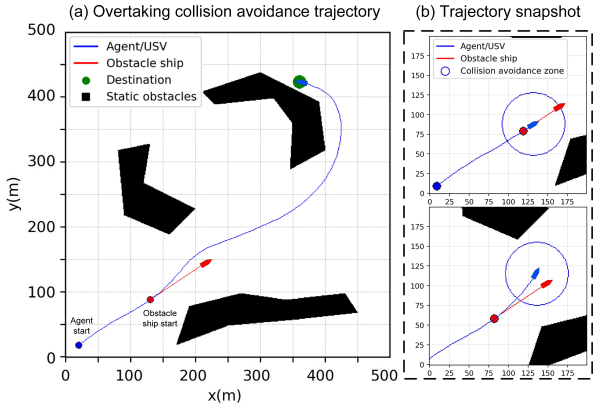

Figure 18: Overtaking situation experiment result.

In the crossing stand-on collision avoidance simulation experiment, the course of the obstacle ship is set to the right side of the USV. The crossing stand-on collision avoidance path planning trajectory is illustrated in Fig.19 (a). From the navigation trajectory results, it can be seen that the USV kept its course and continue sailing after detecting it. The crossing stand-on collision avoidance trajectory snapshot is displayed in Fig.19 (b). It can be seen that when the obstacle ship is detected by the USV, their relative position is in line with the crossing stand-on situation. The USV is a stand-on ship and the obstacle ship is a give-way ship, hence, the USV keeps its course. The collision avoidance operation of the USV complies with COLREGS Chapter 2 Regulation 17.

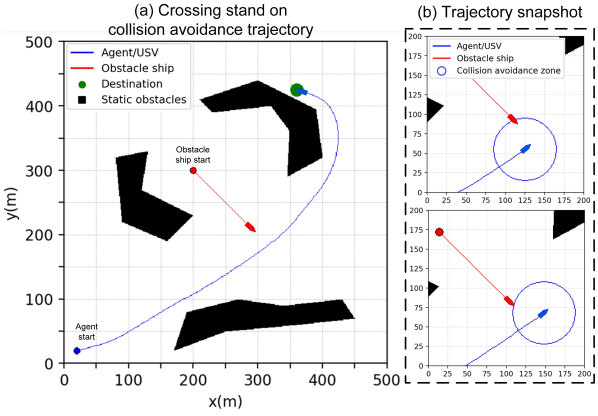

Figure 19: Crossing stand on situation experiment result.

In the mandatory collision avoidance simulation experiment, the course of the obstacle ship is set to the right side of the USV and its collision avoidance action does not follow the rules. The crossing stand-on collision avoidance path planning trajectory is demonstrated in Fig.20 (a). From the navigation trajectory results, it can be seen that the USV turned left to avoid the obstacle after detecting it. The mandatory collision avoidance trajectory snapshot is displayed in Fig.20 (b). It can be seen that when the obstacle ship is detected by the USV, their relative position is in line with the crossing stand-on situation. The USV is a stand-on ship and the obstacle ship is a give-way ship, but the obstacle ship did not abide by COLREGS to give way to the USV. In this situation, the mandatory collision avoidance action determined by the circle domain of own ship will be triggered, the USV travels to the left to avoid collision with the obstacle ship. According to COLREGS Chapter 2 Regulation 15, the obstacle ship should give way to the USV, and the USV should keep stand-on, but according to COLREGS, if the giving way ship fails to comply with the regulations, the standing ship can take mandatory actions to avoid collision.

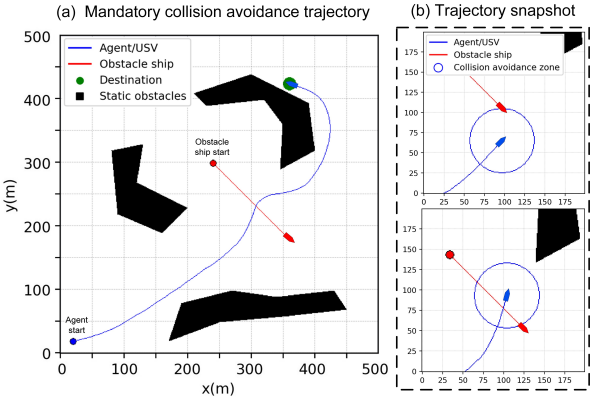

Figure 20: Mandatory collision avoidance experiment result.

The results of the five sets of simulation experiments prove that the proposed method can achieve COLREGS collision avoidance while completing path planning.

4.3. Multi-obstacle path planning and collision avoidance verification

In this section, three groups of multi-obstacle collision avoidance experiments are designed to test our method. In first group of experiments, four mobile obstacle ships are added to the environment, which represent four COLREGS collision avoidance situations. The multi-obstacle collision avoidance trajectory and trajectory snapshot are displayed in Fig.21.

Fig.21 (a) shows that the USV encounters four obstacle ships in sequence during the process of sailing to the goal point. When the USV detects that obstacle ship A enters the collision avoidance area from the right side, it moves to the right to avoid collision with the obstacle ship, and the crossing give-way trajectory snapshot is demonstrated in Fig.21 (b).Then the USV continues to sail when obstacle ship B is detected to enter the collision avoidance zone from the left, and the crossing stand-on trajectory snapshot is illustrated in Fig.21 (c).The USV detects that obstacle ship C is in the collision avoidance zone and moves in the same direction as the USV. To avoid collision, the USV moves to the left to

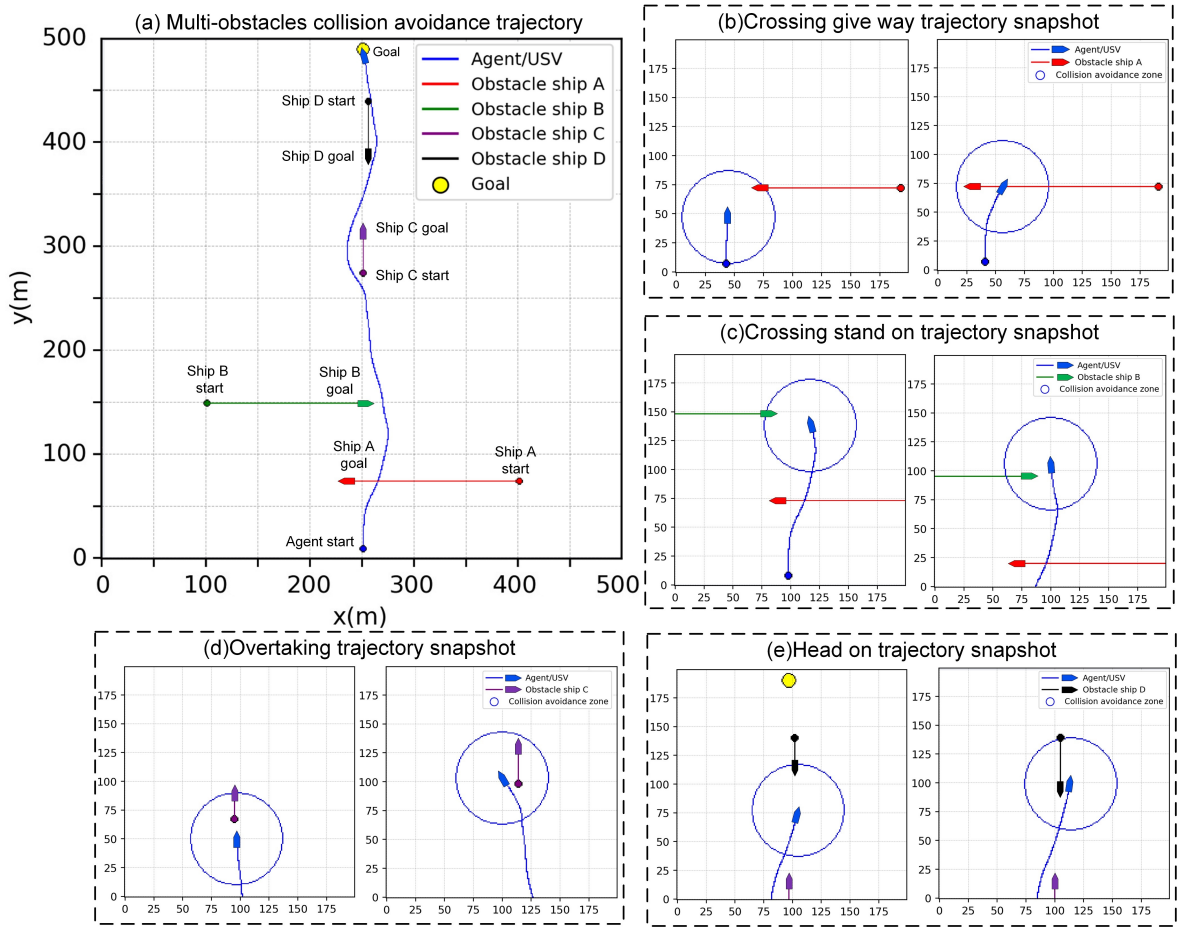

Figure 21: Multi-obstacle collision avoidance trajectory and trajectory snapshot.

give way to obstacle ship C, and the overtaking trajectory snapshot is shown in Fig.21 (d).In the last segment of the trajectory, the USV detects that obstacle ship D is located in the collision avoidance zone and is driving in the opposite direction to the USV. To avoid collision, the USV moves to the right to give way to obstacle ship D, and the head-on trajectory snapshot is displayed in Fig.21 (e).The experimental results show that APF-DQN can effectively deal with the collision avoidance problem in the process of path planning, and the collision avoidance action of USV complies with the COLREGS.

In the second group of experiments, the scenario when own ship encountered two types of target ships at the same time is displayed in Fig.22, which is designed to simulate the multiship encounter scenario of case Fig. 4 (c).

Fig.22 (a) shows the collision avoidance path planning trajectory when own ship encounters two types of target ships at the same time while sailing to the goal. It can be seen from Fig.22 (b), when USV detects that the obstacle ships A and B enter the collision avoidance zone at the same time, USV takes an action to the starboard side to avoid collision. The collision avoidance action of own ship complied with the COLREGS-based strategies shown in Fig.3 and the multiship collision avoidance strategies is illustrated in Fig.4 (c).

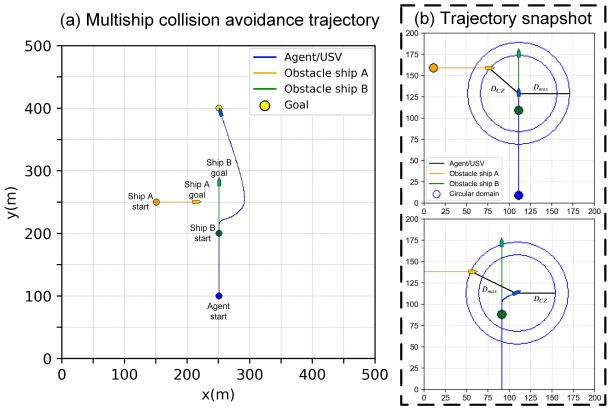

Figure 22: Multiship collision avoidance trajectory and trajectory snapshot of case Fig.4(c).

In the third group of experiments, the scenario when own ship encountered two types of target ships at the same time is displayed in Fig.23, which is designed to simulate the multiship encounter scenario of case Fig. 4(a).

The collision avoidance path planning trajectory of own ship is displayed in Fig. 23 (a). It can be seen from Fig. 23 (b), when the obstacle ships A and C enter the collision

avoidance zone of USV at the same time, USV sails to the starboard side to avoid collision. The collision avoidance action of own ship complied with the COLREGS-based strategies shown in Fig. 3 and the multiship collision avoidance strategies is illustrated in Fig. 4 (a).

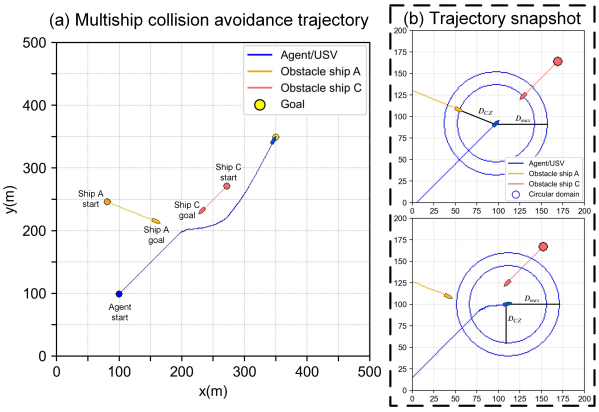

Figure 23: Multiship collision avoidance trajectory and trajectory snapshot of case Fig.4(a).

5. Conclusion

In this paper, a path planning method with collision avoidance function based on DRL and APF was proposed to achieve collision avoidance path planning. Real-time collision avoidance in path planning is essential in guaranteeing navigation security in crowded seas.

The 360° environmental information detected by the sensors of the drone is utilized as the input of the DQN. The algorithm method of APF is used to improve the action space and the reward function of the Deep Q-learning network algorithm, and the improved method is named APF-DQN. To convert the discrete action space of APF-DQN into a partial continuous action space, the potential field is mapped to the action step. The reward function is designed based on the artificial potential field. There are different reward functions according to different situations, which can effectively reflect the experience of the agent in various situations.

To settle the collision avoidance problems that may occur during USV navigation, the location of the obstacle ship is divided into four collision avoidance zones according to the COLREGS. The ability of APF-DQN to handle collision avoidance path planning problems is verified in the visual simulation environment. The simulation experiment results show that the collision avoidance path planning problem can be settled by the APF-DQN, and the collision avoidance actions conform to COLREGS.

Although the method proposed in this paper has many advantages, it also has some limitations. The APF-DQN algorithm appertain to model-free DRL. The output of the algorithm model only represents an action strategy, which will be affected by the physical factors of the ship in practical applications. The deep neural network is trained in the simulated environment, so the trained APF-DQN will be affected by the ship model and environmental uncertainty in practical applications.

For future work, we will try to consider the role of uncertain environmental factors in collision avoidance decision-making to improve the reliability of the proposed method in practical applications. Moreover, the impact of visibility at sea on the accuracy of sensor data is also an issue that cannot be ignored. Therefore, considering the environmental impact and ship kinematics model to achieve precise motion control with deep reinforcement learning is our future work. We will also attempt to combine it with model-based DRL algorithm to achieve better applicability and stability.

Acknowledgements

The authors are grateful for the constructive comments raised by the reviewers which helped significantly improve the quality of the paper. The authors are also grateful for the helpful discussions and guidance provided by Captain Fusheng Li from Jimei University, China. The authors would like to express appreciation for the financial support provided by the National Natural Science Foundation of China (51809113, 51249006), Fujian Province Science and Technology Department (2019H0019, 2019H6 017), Fujian Education Department (JT180266), and Fujian Provincial Department of Ocean and Fisheries (FJ HJF-L-2020-6).

References

[1] Beser, F., Yildirim, T., 2018. Colregs based path planning and bearing only obstacle avoidance for autonomous unmanned surface vehicles. Procedia computer science, 131, 633–640.

[2] Campbell, S., Naeem, W., 2012. A rule-based heuristic method for colregs-compliant collision avoidance for an unmanned surface vehicle. IFAC proceedings volumes, 45, 386–391.

[3] Chen, C., Chen, X.Q., Ma, F., Zeng, X.J., Wang, J., 2019. A knowledge-free path planning approach for smart ships based on reinforcement learning. Ocean Engineering, 189, 106299.

[4] Coldwell, T., 1983. Marine traffic behaviour in restricted waters. The Journal of Navigation 36, 430–444.

[5] Dann, M., Zambetta, F., Thangarajah, J., 2019. Deriving subgoals autonomously to accelerate learning in sparse reward domains, in: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 881–889.

[6] Ding, F., Zhang, Z., Fu, M., Wang, Y., Wang, C., 2018. Energy-efficient path planning and control approach of usv based on particle swarm optimization, in: OCEANS 2018 MTS/IEEE Charleston, IEEE. pp. 1–6.

[7] Duguleana, M., Mogan, G., 2016. Neural networks based reinforcement learning for mobile robots obstacle avoidance. Expert Systems with Applications, 62, 104–115.

[8] Goodwin, E.M., 1975. A statistical study of ship domains. The Journal of navigation, 28, 328–344.

[9] Guo, S., Zhang, X., Zheng, Y., Du, Y., 2020. An autonomous path planning model for unmanned ships based on deep reinforcement learning. Sensors, 20, 426.

[10] Huang, Y., Wu, D., Yin, Z., Yuan, Z.M., 2021. Design of ude-based dynamic surface control for dynamic positioning of vessels with complex disturbances and input constraints. Ocean Engineering 220, 108487.

[11] Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: International Conference on Learning Representation, pp. 1–5.

[12] Lazarowska, A., 2019. Discrete artificial potential field approach to mobile robot path planning. IFAC-PapersOnLine, 52, 277–282.
[13] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature, 521, 436–444.
[14] Lee, S.W., Shimojo, S., O'Doherty, J.P., 2014. Neural computations underlying arbitration between model-based and model-free learning. Neuron, 81, 687–699.
[15] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. Continuous control with deep reinforcement learning, in: International Conference on Learning Representation.
[16] Liu, X., Li, Y., Zhang, J., Zheng, J., Yang, C., 2019. Self-adaptive dynamic obstacle avoidance and path planning for usv under complex maritime environment. IEEE Access, 7, 114945–114954.
[17] Lyu, H., Yin, Y., 2019. Colregs-constrained real-time path planning for autonomous ships using modified artificial potential fields. The Journal of Navigation, 72, 588–608.
[18] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning, in: International Conference on Machine Learning, PMLR. pp. 1928–1937.
[19] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature, 518, 529–533.
[20] Papadimitriou, C.H., Tsitsiklis, J.N., 1987. The complexity of markov decision processes. Mathematics of operations research, 12, 441–450.
[21] Song, R., Liu, Y., Bucknall, R., 2019. Smoothed a* algorithm for practical unmanned surface vehicle path planning. Applied Ocean Research, 83, 9–20.
[22] Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.
[23] Szlapczynski, R., Szlapczynska, J., 2017. Review of ship safety domains: Models and applications. Ocean Engineering 145, 277–289.
[24] Tesauro, G., 1992. Practical issues in temporal difference learning, in: Advances in neural information processing systems, pp. 259–266.
[25] Tokic, M., 2010. Adaptive ϵ-greedy exploration in reinforcement learning based on value differences, in: Annual Conference on Artificial Intelligence, Springer. pp. 203–210.
[26] Vagale, A., Bye, R., Oucheikh, R., Osen, O., Fossen, T., 2021. Path planning and collision avoidance for autonomous surface vehicles ii: a comparative study of algorithms. Journal of Marine Science and Technology doi:10.1007/s00773-020-00790-x.
[27] Wang, D., Wang, P., Zhang, X., Guo, X., Shu, Y., Tian, X., 2020. An obstacle avoidance strategy for the wave glider based on the improved artificial potential field and collision prediction model. Ocean Engineering, 206, 107356.
[28] Wang, X., Liu, Z., Cai, Y., 2017a. The ship maneuverability based collision avoidance dynamic support system in close-quarters situation. Ocean Engineering, 146, 486–497.
[29] Wang, X., Liu, Z., Cai, Y., 2017b. The ship maneuverability based collision avoidance dynamic support system in close-quarters situation. Ocean Engineering, 146, 486–497.
[30] Watkins, C.J., Dayan, P., 1992. Q-learning. Machine learning, 8, 279–292.
[31] Woo, J., Kim, N., 2020. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. Ocean Engineering, 199, 107001.
[32] Wu, D., Liao, Y., Hu, C., Yu, S., Tian, Q., 2020. An enhanced fuzzy control strategy for low-level thrusters in marine dynamic positioning systems based on chaotic random distribution harmony search. International Journal of Fuzzy Systems , 1–17.
[33] Wu, D., Ma, Z., Li, A., Zhu, Q., 2011. Identification for fractional order rational models based on particle swarm optimisation. International journal of computer applications in technology 41, 53–59.
[34] Wu, D., Ren, F., Qiao, L., Zhang, W., 2018. Active disturbance rejection controller design for dynamically positioned vessels based on adaptive hybrid biogeography-based optimization and differential evolution. ISA transactions 78, 56–65.
[35] Xia, G., Han, Z., Zhao, B., Wang, X., 2020. Local path planning for unmanned surface vehicle collision avoidance based on modified quantum particle swarm optimization. Complexity, 2020.
[36] Xie, S., Chu, X., Zheng, M., Liu, C., 2019. Ship predictive collision avoidance method based on an improved beetle antennae search algorithm. Ocean Engineering, 192, 106542.
[37] Yogeswaran, M., Ponnambalam, S., 2012. Reinforcement learning: exploration–exploitation dilemma in multi-agent foraging task. Opsearch, 49, 223–236.
[38] Zhang, Z., Wu, D., Gu, J., Li, F., 2019. A path-planning strategy for unmanned surface vehicles based on an adaptive hybrid dynamic stepsize and target attractive force-rrt algorithm. Journal of Marine Science and Engineering, 7, 132.
[39] Zhao, L., Roh, M.I., 2019. Colregs-compliant multiship collision avoidance based on deep reinforcement learning. Ocean Engineering, 191, 106436.
[40] Zhen, R., Riveiro, M., Jin, Y., 2017. A novel analytic framework of real-time multi-vessel collision risk assessment for maritime traffic surveillance. Ocean Engineering, 145, 492–501.
[41] Zhu, B., Li, C., Song, L., Song, Y., Li, Y., 2017. A* algorithm of global path planning based on the grid map and v-graph environmental model for the mobile robot, in: 2017 Chinese Automation Congress (CAC), IEEE. pp. 4973–4977.
[42] Zhu, G., Ma, Y., Hu, S., 2020. Single-parameter-learning-based finite-time tracking control of underactuated msvs under input saturation. Control Engineering Practice, 105, 104652.