Ocean Engineering 287 (2023) 115649

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

A novel intelligent collision avoidance algorithm based on deep reinforcement learning approach for USV

Yunsheng Fan *, Zhe Sun, Guofeng Wang

College of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China
Key Laboratory of Technology and System for Intelligent Ships of Liaoning Province, Dalian 116026, China

ARTICLE INFO

Keywords:
Unmanned surface vehicles
Deep reinforcement learning
Autonomous collision avoidance
Artificial intelligence

ABSTRACT

Enhancing the efficiency of unmanned surface vehicles (USVs) collision avoidance can yield a significant impact, as it can result in safer navigation and lower energy consumption. This paper introduces a robust approach employing deep reinforcement learning theory to facilitate informed collision avoidance decisions within intricate maritime environments. The restrictions on USV maneuverability and international regulations for preventing collisions at sea are studied and quantified, particularly focusing on the shape and size changes of the ship's domain caused by USV speed. Based on the deep Q network, an improved methodology is designed, incorporating a noisy network, prioritized experience replay, dueling neural network architecture, and double Q learning, resulting in a highly efficient sampling, exploration, and learning process. To curtail computational expenses associated with USVs, a novel dynamic area restriction technique is proposed. Furthermore, an innovative USV state clipping method is introduced to mitigate training complexities. By utilizing the Unity platform, a virtual environment characterized by complexity and stochasticity is constructed for training and testing the collision avoidance of USVs This novel approach surpasses the performance of the pre-improvement algorithm across multiple collision avoidance effectiveness indicators and performance metrics.

1. Introduction

Due to its autonomous characteristics, research on unmanned surface vehicles (USVs) has become a crucial component across various fields, including ocean exploration, environmental monitoring, and rescue operations (Negenborn et al., 2023). Ensuring safe and intelligent navigation and conducting maritime missions have always necessitated effective collision avoidance for USVs, making it a key issue. Consequently, the enhancement of the control effect of USVs is critical. Hence, the exploration of intelligent collision avoidance algorithms tailored for USVs bears substantial practical significance

Numerous classical algorithms are available for achieving effective collision avoidance in USVs. These algorithms can be categorized into two groups: dynamic collision avoidance and static path planning, each tailored to different research objectives. In static path planning, the Dijkstra algorithm, which employs breadth-first search, can chart the shortest path between two points (Xu et al., 2007). Another notably efficient algorithm is A*, which employs an evaluation function to strategize path planning (Liu et al., 2019). Liang et al. (2021) optimized the A* algorithm by refining ship waypoint settings and eliminating unnecessary waypoints, resulting in improved ship path planning. Shi et al. (2019) proposed an enhanced A* algorithm that incorporates motion primitive constraints to enable autonomous planning, optimization, and autonomous navigation of USVs. Dynamic collision avoidance centers on the real-time decision-making process to handle changing marine conditions. The Artificial Potential Field (APF) method considers the motion of objects in the surrounding environment as a result of gravity from the goal and repulsion from obstacles. However, this approach is susceptible to getting trapped in local optima (Rasekhipour et al., 2016). Xu et al. (2020b) proposed a layered artificial potential field algorithm that optimizes collision avoidance behavior and assesses collision hazards. Zhang et al. (2022) addressed the planning deficiencies of the APF algorithm in environments with multiple static obstacles and achieved improved ship navigation by modifying the potential field function. The Velocity Obstacle (VO) algorithm, based on the velocity vector, constructs a restricted access area for each USV to facilitate collision avoidance decisions (Kim and Oh, 2016). This algorithm has been widely employed in USV collision avoidance. Wang et al. Wenming et al. (2022) proposed a proactive velocity obstacle algorithm that predicts potential courses, leading to better compliance with the COLREGs for collision avoidance. Additionally, the Dynamic Window (DW) algorithm and Fast Marching Method (FMM) are two

* Corresponding author at: College of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China
E-mail address: yunsheng@dlmu.edu.cn (Y. Fan).

https://doi.org/10.1016/j.oceaneng.2023.115649
Received 11 March 2023; Received in revised form 10 August 2023; Accepted 17 August 2023
Available online 11 September 2023

classical collision avoidance algorithms that consider kinematics, including acceleration and speed constraints (Sun et al., 2021; Liu et al., 2017).

Many recent successes in USV collision avoidance research were kick-started by artificial intelligence theory. Its way of imitating navigator behavior enabled USVs to learn, from navigation information, how to make real-time collision avoidance decisions. Deep reinforcement learning (DRL) theory combines deep learning (DL) and reinforcement learning (RL), providing an efficient approach to decision-making problems. DRL is particularly well-suited for meeting the latest requirements of autonomous collision avoidance in USVs, and it offers advantages when dealing with manned ships. These requirements are challenging for other algorithms to achieve. In particular, the deep reinforcement learning approach offers high real-time performance and scalability, making it convenient to deploy and implement. DRL theory has achieved remarkable success in many fields, including electronic games (Vinyals et al., 2015), navigation (Kiran et al., 2021), operation acceleration (Fawzi et al., 2022), and the game of Go (Silver et al., 2017). Therefore, deep reinforcement learning approaches have been adopted in USV collision avoidance research to achieve proficient autonomous collision avoidance. Compared to traditional USV collision avoidance algorithms, the DRL approach offers an advantage in handling the controlled object and environment, which difficult to model accurately (Shaobo et al., 2020). In the context of USV collision avoidance problem, obtaining learning samples or prior knowledge based on the imitation of navigator behavior is challenging, especially for USVs where collision avoidance sample information is difficult to obtain or difficult to obtain in large quantities. However, the reinforcement learning algorithm, which can learn without human guidance on desired behavior, offers significant advantages.

While the reinforcement learning approach came into being earlier, its widespread application was constrained by computational costs and expressive capability until the birth of AlphaGo and the development of deep learning approach. This has sparked widespread interest in deep reinforcement learning, and it is now applied in many fields (Zhang et al., 2021). However, the application of deep reinforcement learning approach to USV collision avoidance problems was relatively late, and a little exploratory research in this area only emerged around 2017. In recent years, investigations have primarily focused on the following aspects, which are also burning issues in the application of deep reinforcement learning to USV collision avoidance:

(1) Select an algorithm suitable for the USV collision avoidance problem. Most researchers choose the Deep Deterministic Policy Gradient (DDPG) algorithm (Lillicrap et al., 2015) or the Deep Q-Network (DQN) algorithm (Mnih et al., 2015) for designing collision avoidance algorithms for USVs. The former is suitable for continuous actions, while the latter can only handle discrete actions. Many improved algorithms for USV collision avoidance have been developed based on these approaches (Woo and Woo, 2020; Guo et al., 2020; Xu et al., 2020a).
(2) Design a suitable reward signal for training USV collision avoidance. As a key aspect of training, the reward signal based on the characteristics of USV collision avoidance is a more appropriate choice. This can include factors like compliance with maneuverability and international regulations for preventing collisions at sea (COLREGs) (Cheng and Zhang, 2018; Chun et al., 2022; Xu et al., 2022).
(3) An excellent and comprehensive training environment is essential for training agents of the USV's collision avoidance (Zhang et al., 2019; Shen et al., 2019).

Based on the research in these three aspects, there are still some shortcomings in the current deep reinforcement learning methods for USV collision avoidance. Firstly, many studies utilize a fixed environment for training, which leads to a lack of practical relevance. Furthermore, the use of multiple random number seeds for training can effectively validate the algorithm's universality, stability, and superiority, yet this aspect is often overlooked in most of the articles. Moreover, the considerations of maneuverability and COLREGs are crucial. Taking the aforementioned issues into account, a novel collision avoidance algorithm for USVs that combines an enhanced deep reinforcement learning approach is proposed. This algorithm does not require prior knowledge yet achieves effective collision avoidance. The main contributions of this paper are as follows:

(1) The consideration of maneuverability and adherence to COLREGs is factored into USV collision avoidance. The impact of changes in USV speed on the size and shape of the ship domain is incorporated into the training for USV collision avoidance. Building on these considerations, suitable state, action, and reward signals are devised for USV training.
(2) To mitigate estimation bias and enhance learning efficiency, the approach of utilizing the double learning and dueling architecture is adopted. Given the distinct attributes of a multitude of diverse and time-sensitive USV training samples, the prioritized experience replay framework is applied to modify the sampling approach, thereby enhancing the training value of crucial samples. In response to the significant complexity and stochastic nature of the simulation environment in this study, a noisy network is deliberately designed to promote exploratory behavior and amplify exploration efficiency.
(3) To boost the collision avoidance performance of USVs, an algorithmic training optimization method is introduced in this paper. This method takes into account the distinct characteristics of USVs and introduces state clipping and dynamic area constraints. Through reshaping the information used for neural network learning, it establishes a more effective training environment.

This paper is structured as follows: Section 2 introduces the COLREGs, ship domain, dynamic area, and motion model of USVs. Section 3 presents fundamental concepts in deep reinforcement learning and the algorithm improvements proposed in this paper. Section 4 covers the training process for USV collision avoidance. Section 5 discusses the simulation and comparison of the NPD3QNU algorithm within the environments designed in this study. The final section provides a summary and outlines prospects for future research.

2. Construction of USVs collision avoidance model

2.1. Collision avoidance characteristics of USV

Researching efficient algorithms for USV collision avoidance can have a significant impact, as it can improve its safety and intelligence. USV collision avoidance is a meaningful task that necessitates consideration of the complex and real-time changes in the marine environment, along with the execution of appropriate actions. Therefore, designing a collision avoidance algorithm that balances between safety and efficiency presents challenging. Intelligent USVs must continuously perceive real-time environmental information and transmit it to the control system, yielding improved collision avoidance results and enabling safer and more intelligent navigation at sea. When encountering other obstacles and aiming to achieve autonomous collision avoidance, several principles and challenges require consideration:

(1) When sensing obstacles such as other USVs or reefs and assessing the risk of collision, it is crucial to adhere to the principles of seamanship. The objective is to promptly choose the appropriate and skillful maneuver, ensuring the execution of a safe, reasonable, and reliable collision avoidance behavior. This ensures that each USV can encounter and navigate past obstacles safely.

(2) When encountering obstacles and needing to avoid them, it is advisable to alter the navigation course instead of relying on frequent rudder changes. Altering the course offers a more stable and controlled maneuver. Additionally, it is generally recommended to avoid altering the speed of the USV unless an urgent need arises.
(3) Due to the intricate marine environment, an ideal collision avoidance algorithm should showcase high real-time performance, outstanding generalization ability, robust noise immunity, and long-term stability. Moreover, the algorithm should take into account the influence of various environmental factors, COLREGs, and maneuverability principles. This comprehensive approach ensures that the autonomous collision avoidance algorithm maintains a higher level of practical applicability.
(4) After successfully avoiding a collision, USVs should promptly resume their previous navigation status. They can either return to their original route or continue sailing directly towards their destination.
(5) When multiple USVs encounter each other, it is crucial to avoid situations where one USV's avoidance maneuver leads to an urgent collision avoidance scenario with other obstacles. Therefore, the designed collision avoidance algorithm should possess decision-making capabilities to effectively handle encounters involving multiple USVs.

2.2. USV motion model

As shown in Fig. 1, the schematic diagram illustrates the three degree of freedom (3DOF) motion parameters among the own USV, obstacle USV, and terminal. The own (USV_U) is brown. The obstacle USV (USV_{O_i}) is blue. The pink circle represents the terminal for USV navigation. (x_U, y_U) is the position of the own USV. (x_T, y_T) is the position of the terminal. (x_{S_i}, y_{S_i}) is the position of the static obstacle. (x_{O_i}, y_{O_i}) is the position of the obstacle USV. φ_U is the course of the own USV. φ_{O_i} is the course of the obstacle USV. ϑ_T is the absolute azimuth of the terminal and the own USV. ϑ_{O_i} is the absolute azimuth between the obstacle USV and the own USV. ϑ_{S_i} is the absolute azimuth between the static obstacle and the own USV. ϕ_{TU} is the relative azimuth between the terminal and the own USV. V_U is the speed of the own USV. V_{O_i} is the speed of the obstacle USV. d_{UO_i} is the distance between the obstacle USV and the own USV. d_{US_i} is the distance between the static obstacle and the terminal. d_{UT} is the distance between the static obstacle and the own USV. ϕ_{O_iU} is the relative azimuth between the obstacle USV and the own USV. ϕ_{TU} is a relative azimuth between the terminal and the own USV. ϕ_{S_iU} is the relative azimuth of the static obstacle and the own USV. δ is the rudder angle of the own USV. θ_{UO_i} is the deviation in course between the two USVs.

The Norrbin mathematical model for simulating ship navigation is shown as follows,

$$\begin{cases} T\dot{\eta} + \eta + \alpha\eta^3 = K\delta \\ \eta = \dot{\varphi} \end{cases} \tag{1}$$

where, T, α, and K represent the following and gyration performance of USV. These parameters are determined through identification. δ is the actual rudder angle. φ is the USV course. η is the change rate of the USV course. These variables describe the changes in the direction of the USV during navigation (Sun et al., 2020). This paper focuses on the "LanXin" USV, which is equipped with a thrust vector control engine. The equation for the steering gear characteristics can be expressed as follows,

$$\ddot{\delta} + 2\zeta\omega_n\dot{\delta} + {\omega_n}^2\delta = k{\omega_n}^2\delta_r \tag{2}$$

where, ω_n, ζ and k are intrinsic frequency, damping ratio and proportionality coefficient. δ_r is the target rudder angle (Sun et al., 2018). The range of the rudder angle for the "LanXin" USV is $[-35°, +35°]$.

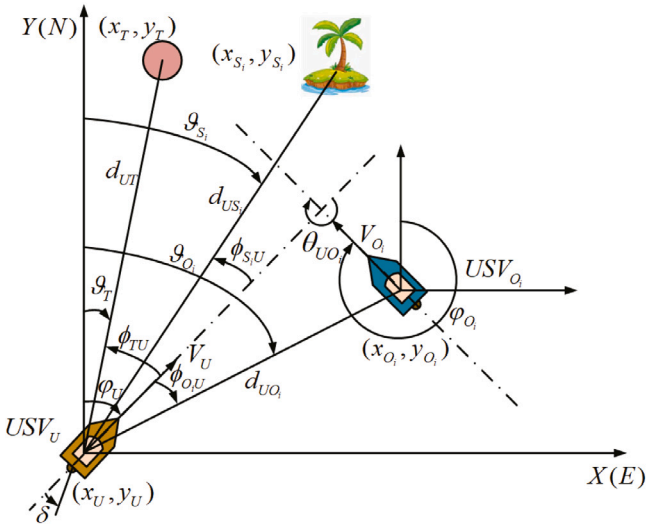


Fig. 1. Schematic diagram of USV motion.

2.3. Ship domain

For safety reasons, each USV is assigned an inviolable ship domain (SD). The concept of a ship domain, defined as "a two-dimensional area surrounding a ship that other ships must avoid", was first introduced by Fujii (Fujii and Tanaka, 1971). The size and shape of the ship domain are influenced by factors such as the size and speed of the USV. Basing the ship domain model proposed by Tam (Tam and Bucknall, 2010), this paper considers the characteristics of the "Lan Xin" USV and devises a ship domain model that adjusts in size and shape with speed. As shown in Fig. 2(a), the ship domain consists of two parts: a semi-ellipse in front of the USV and a semi-circle behind it. r_a is the semi-major axis. r_c is the semi-minor axis. The semi-circle is characterized by the radius r_b. The formulas for determining these parameters are as follows,

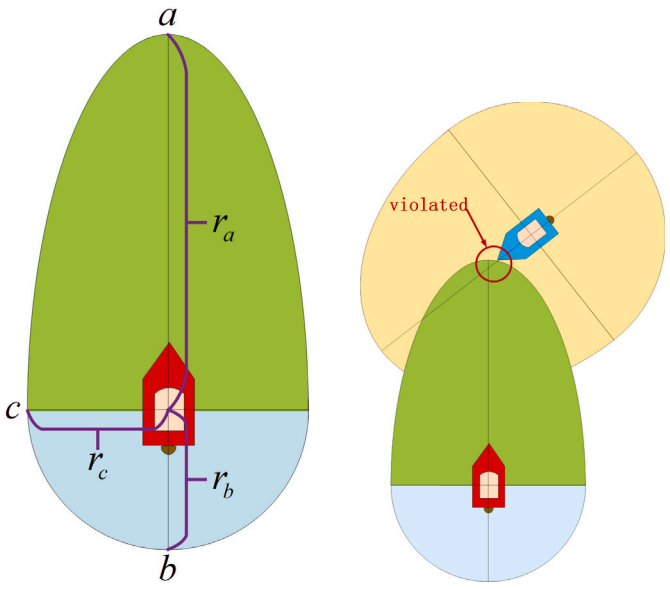
$$r_a = V_{USV}t_1 + r_{\min} \tag{3}$$

$$r_b = \begin{cases} V_{USV}t_1 + r_{\min} & V_{USV} < V_{cr1} \\ V_{USV}t_2 - V_{cr2}t_2 + r_{\min} & V_{cr1} \le V_{USV} < V_{cr2} \\ r_{\min} & V_{cr2} \le V_{USV} \end{cases} \tag{4}$$

$$r_c = r_b \tag{5}$$

where, V_{USV} is the speed of USV. t_1 and t_2 are rate of change. $r_{\min}$ is the minimum ship domain radius. V_{cr1} and V_{cr2} are two critical velocity values. Fig. 3 shows the trend of changes in the length of the major and minor axes of the ellipse that determine the size and shape of the ship domain. The major axis, r_a, always shows an upward trend and increases continuously with increasing USV speed until reaching the maximum navigational speed that "LanXin" can reach. The minor axis, r_b, increases until it reaches a critical point and then gradually decreases until it reaches the minimum value, $r_{\min}$. Additionally, Fig. 2(b) shows the definition of violation concerning this USV ship domain Szlapczynski and Szlapczynska (2020). Considering the impact of USV speed on the size and shape of the ship domain can make the training results of the algorithm more closely aligned with the actual conditions of the "LanXin" USV. This can improve the practicality of the algorithm's training outcomes.

The collision risk index (CRI) is a crucial factor in evaluating encounter situations between USVs. A fundamental concept in calculating the CRI is the dynamic area (DA), represented as R. This dynamic area is visualized as a large circular region encompassing the own USV, and holds a pivotal role in CRI assessment.



(a) Shape and size. (b) Definition of violation.

Fig. 2. Ship domain.

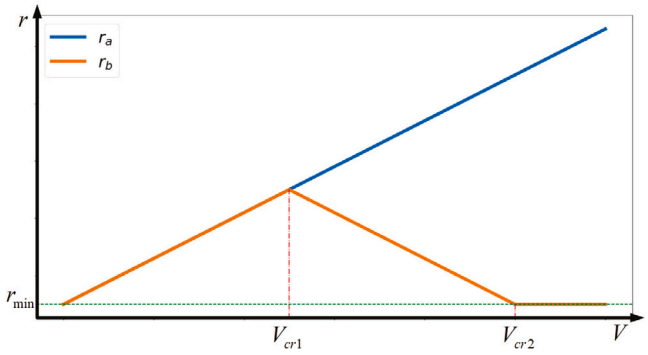


Fig. 3. Changes of USV ship domain.

The CRI is a crucial parameter in USV collision avoidance, as it defines the current collision risk level. As shown in Fig. 4, V_{O_iU} represents the relative speed of the obstacle USV with respect to the own USV. ϖ is the relative speed direction of the obstacle USV with respect to the own USV. Two important parameters, namely the distance at the closest point of approach (DCPA) and the time to the closest point of approach (TCPA), play a key role in calculating the CRI. The DCPA and TCPA can be determined as follows,

$$\begin{cases} C_{DCPA} = d_{UO_i} \sin(\varpi) \\ C_{TCPA} = d_{UO_i} \cos(\varpi)/V_{O_iU} \end{cases} \tag{6}$$

Therefore, the membership function u_{DCPA} and u_{TCPA} can be calculated directly based on C_{DCPA} and C_{TCPA} directly (Fan et al., 2022) as follows,

$$u_{DCPA} = \begin{cases} 1, & |C_{DCPA}| \le r \\ 0.5 - 0.5\sin[\frac{\pi}{R-r} \times \frac{C_{DCPA}(R+r)}{2}], & r < |C_{DCPA}| \le R \\ 0, & |C_{DCPA}| > R \end{cases} \tag{7}$$

If $C_{TCPA} > 0$,

$$u_{TCPA} = \begin{cases} 1, & C_{TCPA} \le T_1 \\ [\frac{T_R - C_{TCPA}}{T_R - T_r}]^2, & T_r < C_{TCPA} \le T_R \\ 0, & C_{TCPA} > T_R \end{cases} \tag{8}$$

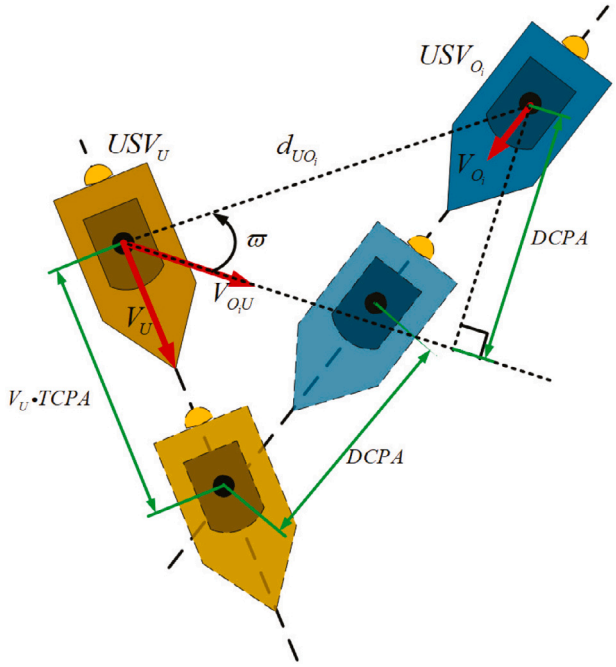


Fig. 4. DCPA and TCPA.

If $C_{TCPA} \le 0$,

$$u_{TCPA} = \begin{cases} 1, & |C_{TCPA}| \le T_r \\ [\frac{T_R + C_{TCPA}}{T_R - T_r}]^2, & T_r < |C_{TCPA}| \le T_R \\ 0, & |C_{TCPA}| > T_R \end{cases} \tag{9}$$

T_r and T_R are expressed as follows,

$$T_r = \begin{cases} \frac{\sqrt{r^2 + C_{DCPA}^2}}{V_{OU}}, & r \ge |C_{DCPA}| \\ 0, & r < |C_{DCPA}| \end{cases} \tag{10}$$

$$T_R = \begin{cases} \frac{\sqrt{R^2 - C_{DCPA}^2}}{V_{OU}}, & R \ge |C_{DCPA}| \\ 0, & R < |C_{DCPA}| \end{cases} \tag{11}$$

where, r is the size of the ship domain. The CRI can be expressed as,

$$u_{CRI} = \begin{cases} 0 & u_{DCPA} = 0 \\ 0 & u_{DCPA} \ne 0, u_{TCPA} = 0 \\ \max(u_{DCPA}, u_{TCPA}) & u_{DCPA} \ne 0, u_{TCPA} \ne 0 \end{cases} \tag{12}$$

2.4. COLREGs

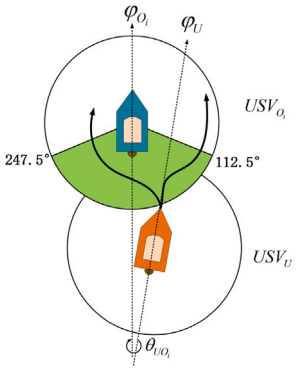
The COLREGs play a paramount role in regulating the collision avoidance behavior of USVs. They provide specific guidelines on the appropriate avoidance behavior that each USV should adopt in various encounter situations, thus ensuring the safety of USV collision avoidance.

Considering the constraints imposed by the COLREGs and leveraging the excellent maneuverability characteristics of the "LanXin" USV, this paper formulates a set of comprehensive principles to address different encounter situations denoted as E:

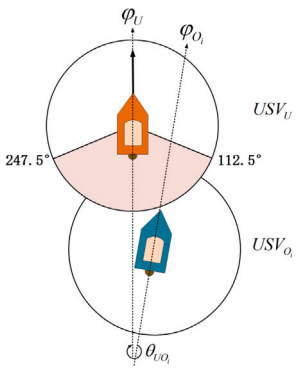
(1) If:

(a) USV_U is at the relative azimuth of the USV_{O_i} in $[\varphi_{O_i} + 112.5°, \varphi_{O_i} + 247.5°]$.

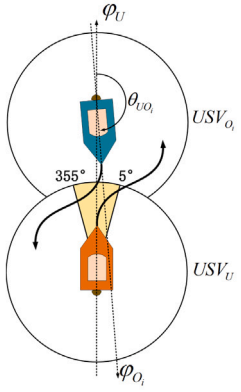
(b) $V_U > V_{O_i} \cos(\theta_{UO_i})$.



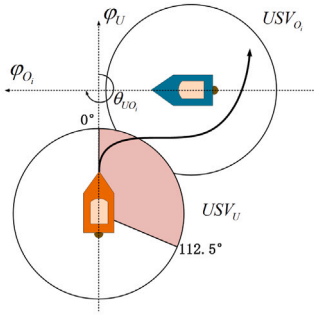
(a) Overtaking Give Way.



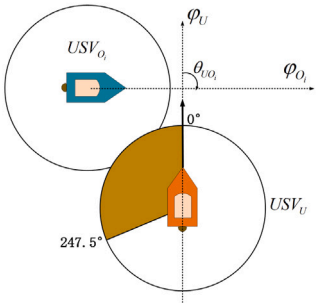
(b) Overtaking Stand On.



(c) Head On.



(d) Crossing Give Way.



(e) Crossing Stand On.

Fig. 5. Encounter situations.

(c) $d_{UO_i} \leq R$, and $u_{CRI} > 0$.

It is the overtaking-give-way encounter situation. As shown in Fig. 5(a), USV_U should either turn to the starboard or port side, while the obstacle USV should maintain its course. It represents as $E = E_1$.

(2) If:

(a) USV_{O_i} is at the relative azimuth of the USV_U in $[\varphi_{O_i} + 112.5°, \varphi_{O_i} + 247.5°]$.

(b) $V_U < V_{O_i} \cos(\theta_{UO_i})$.

(c) $d_{UO_i} \leq R$, and $u_{CRI} > 0$.

It is the overtaking-stand-on encounter situation. As shown in Fig. 5(b), USV_U should stand on, while USV_{O_i} should either turn to the starboard or port side. It represents as $E = E_2$.

(3) If:

(a) USV_{O_i} is at the relative azimuth of the USV_U in $[\varphi_U + 345°, \varphi_U + 360°) \cup [\varphi_U, \varphi_U + 15°]$.

(b) $165° < \theta_{UO_i} < 195°$.

(c) $d \leq R$, and $u_{CRI} > 0$.

It is the head-on encounter situation. As shown in Fig. 5(c), both USV should turn to the starboard side. It represents as $E = E_3$.

(4) If:

(a) USV_{O_i} is at the relative azimuth of the USV_U in $[\varphi_U, \varphi_U + 112.5°)$.

(b) $180° < \theta_{UO_i} < 360°$.

(c) $d \leq R$, and $u_{CRI} > 0$.

(d) Two USVs do not in the encounter situation of (1), (2) and (3).

It is the crossing-give-way encounter situation. As shown in Fig. 5(d), USV_U should turn to the starboard side, while USV_{O_i} should stand on. It represents as $E = E_4$.

(5) If:

(a) USV_{O_i} is at the relative azimuth of the USV_U in $[\varphi_U + 274.5°, \varphi_U + 360°)$.

(b) $0° < \theta_{UO_i} < 180°$.

(c) $d \leq R$, and $u_{CRI} > 0$.

(d) Two USVs do not in the encounter situation of (1), (2) and (3).

It is the crossing-stand-on USV encounter situation. As shown in Fig. 5(e), USV_U should stand on, while USV_{O_i} should turn to the starboard side. It represents as $E = E_5$.

(6) If the USV_{O_i} violates the principle or loses maneuverability, USV_U should be give way. It represents as $E = E_6$.

3. Intelligent collision avoidance algorithm

The interactive architecture of the deep reinforcement learning approach proves highly effective in addressing intricate control challenges. With its distinctive framework, this approach attains human-level performance without prior knowledge of the learning process. Significantly, the training method of this approach closely aligns with the decision-making process involved in USV collision avoidance. By crafting an appropriate training environment and devising corresponding reward signals, the training process can be executed with efficiently, resulting in safe obstacle avoidance behavior.

3.1. Deep reinforcement learning

Adopting the deep reinforcement learning approach in USV collision avoidance systems has a unique characteristic that the agent can continuously enhances its performance by evolving from a state of initial unfamiliarity with effective collision avoidance behavior to adeptly managing diverse scenarios. Deep reinforcement learning can be categorized into model-based and model-free approaches; this paper specifically focuses on the model-free theory. Model-free deep reinforcement learning further divided into two methods: value function-based and policy gradient-based. The former suits discrete action problems, while the latter for continuous action problems. The fundamental framework for model-free deep reinforcement learning is the Markov decision process (MDP), wherein the agent makes decisions, takes actions within the environment, observes state transitions, receives reward signals, and refines its behavior in an iterative loop. The deep reinforcement learning approach comprises four key elements: the policy π, which describes the mapping between states and actions; the reward signal G_t, which evaluates the agent's behavior; the value function V, which estimates the quality of the agent's behavior; and whether the algorithm is trained with or without a model. During the training process, the

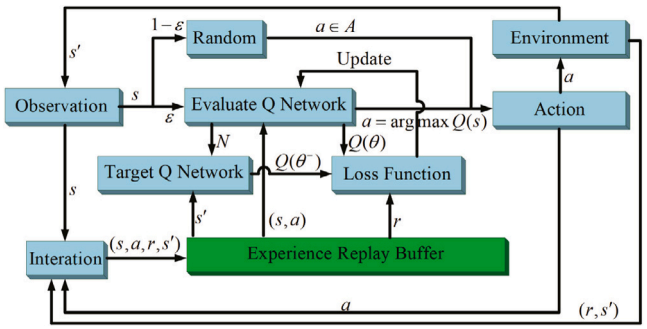


Fig. 6. Framework of DQN Algorithm.

agent employs the current state S_t and the neural network framework θ to make action decisions A_t within the maritime environment. The changes in both the USVs and the environment lead to the formation of a new state S_{t+1}, and a reward signal R_{t+1} is obtained. Through continuous learning, the USV agent progressively deepens its comprehension of collision avoidance predicaments and becomes adept at addressing diverse collision avoidance scenarios. This iterative learning process enhances its decision-making capabilities, leading to refined collision avoidance behavior.

In this paper, the DQN algorithm, which is rooted in the temporal difference (TD) algorithm, is chosen. It synergizes the advantages of the Monte Carlo (MC) algorithm (Bojesen, 2018) and the dynamic programming (DP) algorithm (Wang et al., 2017) to achieve efficient learning outcomes (Sutton and Barto, 2018). The DQN algorithm, founded on Q-learning, utilizes a neural network for approximating Q-values. This capability empowers the algorithm to adeptly address control challenges with higher dimensions and augmented complexity. It is loss function is as follows,

$$L(\theta) = \mathbb{E}[(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta))^2] \tag{13}$$

The framework of the DQN algorithm is depicted in Fig. 6. During the exploration phase in the environment, the agent selects an action based on the greedy value ε and the current state s. The action a is chosen either with a probability of ε based on the neural network or randomly with a probability of $1-\varepsilon$ from the action space A. The agent then receives a new state s' and a reward signal r, and this interaction (s, a, r, s') is stored in the experience replay buffer.

In the training phase, the network parameters are updated at each step based on the extensive collected experiences. During each exploration step, a certain number of interactions are sampled. The states and actions are fed into the evaluation Q network to generate $Q(\theta)$, while the next states s' are inputted into the target Q network to generate $Q(\theta^-)$. These values are then used in Eq. (13) to calculate the loss function of the target network. To ensure more stable training, the target network is not updated at every step; instead, the network parameters θ from the evaluated network are copied to the target network every N steps, resulting in $\theta^- = \theta$.

3.2. Improvement of USv collision avoidance algorithm

Since the DQN algorithm employs a bootstrap method for updates, the estimation of future action value may not be accurate, leading to overestimation issues that are challenging to identify without reference standards (Van et al., 2016). This inconsistency in overestimation can have negative impacts.

To address this problem, a double-learning algorithm is proposed. In this algorithm, when the target network outputs the Q value, the evaluated network is used to select the corresponding action, effectively mitigating the overestimation issue. The Q value output by the newly evaluated network is shown as follows,

$$Y_t^{Double} = r + \gamma Q(s', \arg\max_a Q(s', a; \theta); \theta^-) \tag{14}$$

The dueling network architecture (Wang et al., 2016) is another effective improvement method. By decoupling the action value function into a state value function and an advantage function, the value assessment of specific states and their corresponding actions can be made more accurate. The action value function can be expressed as follows,

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + A(s, a; \theta, \alpha) \tag{15}$$

To enhance the effectiveness of each step in USV collision avoidance training, it is not optimal to uniformly sample from the experience replay buffer when dealing with a large amount of interaction information (Schaul et al., 2015). Coincidentally, the TD-error χ_i can directly reflect the value of each interaction at the current time for training. Therefore, by considering the priority p_i, interactions with higher importance can be sampled more frequently, leading to improved training outcomes. The sampling probability for each sample is calculated as follows,

$$P(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{16}$$

where, $P(i)$ represents the sampling probability. k is the total number of samples. α is the parameter that controls the priority effect. Since uniform random sampling is no longer used, importance sampling weights (ISW) need to be incorporated to correct the bias of policy estimation. It can be calculated as follows,

$$\omega_i = \left(\frac{1}{N}\frac{1}{P(i)}\right)^\beta \tag{17}$$

where, β is used to adjust effect of non-uniform sampling.

The training environment in this paper is designed to be rich, with the aim of enhancing the collision avoidance agent's training and significantly developing its exploration ability, leading to better training results. In traditional methods, the ε-greedy policy is used to encourage exploration, but it is considered to be blind. Therefore, the addition of purposeful noise can greatly enhance the exploration capability of the USV agent (Fortunato et al., 2017). The noise is added as follows,

$$y \doteq (g^w + h^w \odot \varepsilon^w)x + g^b + h^b \odot \varepsilon^b \tag{18}$$

where, the weight is divided into two parts: g^w without noise, and $h^w \odot \varepsilon^w$ with noise. The symbol $\odot$ denotes element-wise multiplication. Similarly, the bias is divided into g^b and $h^b \odot \varepsilon^b$. The noise ε is structured as follows,

$$\begin{cases} \varepsilon_{i,j}^w = n(\varepsilon_i)n(\varepsilon_j) \\ \varepsilon_j^b = n(\varepsilon_j) \end{cases} \tag{19}$$

where, $n(x) = \mathrm{sgn}(x)\sqrt{|x|}$. ε_i and ε_j follows a normal distribution with $\varepsilon \sim N(0, k)$.

To tackle the issue of significant variations in the magnitude of scalar values within the USV state vector input to the neural network, state clipping technique is employed to enhance network training. By normalizing the input, the problem of large gradient descent during training caused by large inputs can be mitigated, resulting in more effective and robust neural network training. In the training environment in this paper, all distance values are divided by the side length of the environment, denoted as R_E. Similarly, azimuth angles and their rate of change are divided by 2π and $c_{\dot{\varphi}}$, respectively. Rudder angles and their rate of change are divided by $\frac{35\pi}{180}$ and $c_{\dot{\delta}}$, respectively. This normalization process ensures that each component of the state vector contributes uniformly to the training process.

The distance for learning in this paper is determined by a novel method called dynamic area restriction. This method takes into consideration the fact that when obstacles are situated too far from the own

USV, changes in their distance have negligible impact on the navigation of the own USV. Consequently, allocating computational resources to learn such conditions would be wasteful. As a result, the distance for learning in this paper is determined as follows,

$$d_{DA} = \begin{cases} d, & d < R \\ R, & d \geq R \end{cases} \tag{20}$$

So, the new loss function for the improved algorithm can be obtained as follows,

$$L(\zeta)_{NPD3QNU} = \mathbb{E}_{\epsilon,\varepsilon^-}[\mathbb{E}_{(s,a,r,s')\sim D} [r + \gamma \max_{a'} Q(s', a', \varepsilon^-; \zeta^-) - Q(s, a, \varepsilon; \zeta)]^2] \tag{21}$$

3.3. Design of training elements

Applying deep reinforcement learning theory to address the issue of USV collision avoidance necessitates suitable state, action, and reward signals for the learning process.

When considering the state space, it becomes crucial to possess a concise, comprehensive, and non-redundant representation of the current collision avoidance encounter situation at sea during USV training. In this paper, the following state space is designed,

$$S = \{\varphi_U, \dot{\varphi}_U, \delta_U, \dot{\delta}_U, V_U, \vartheta_T, d_T, d_{UO_1}, \vartheta_{O_1}, \varphi_{O_1}, \ldots, d_{UO_m}, \vartheta_{O_m}, \varphi_{O_m}, d_{US_1}, \vartheta_{S_1}, \ldots, d_{US_n}, \vartheta_{S_n}\} \tag{22}$$

In state space, the own USV information includes φ_U, $\dot{\varphi}_U$, δ_U, $\dot{\delta}_U$, and V_U. The terminal information includes ϑ_T, and d_T. The obstacle USV information in the state space consists of m groups of ϑ_{O_m}, d_{UO_m}, and φ_{O_m} for m obstacle USVs. The static obstacles information in the state space includes n groups of ϑ_{S_n} and d_{US_n} for n static obstacles.

Designing a variety of different rudder angle change actions in the action space is feasible, and these actions can be designed as follows:

$$\begin{cases} A = \{\Delta\delta_1, \Delta\delta_2, \ldots, \Delta\delta_k\} \\ \delta_r \leftarrow \delta_r + \Delta\delta_k \end{cases} \tag{23}$$

where, $\Delta\delta_k$ represents the change in the rudder angle command. δ_r is the current rudder angle. This paper defines a total of 11 actions within the action space. These actions encompass a diverse spectrum of rudder angle adjustments, ranging from small to large changes, as well as no change at all. The action space is represented as $A = \{c| -5° \leq c \leq 5°, c \in Z\}$. This comprehensive set of actions enables the agent to navigate a wide array of possible rudder angle scenarios.

The reward signal serves as the foundation for implementing the constrained collision avoidance behavior. In this paper, the reward signal designed for collision avoidance training is split into the following two parts,

(1) Goal reward

(a) Terminal reward: Each training episode concludes at a terminal, which represents the end of USV navigation. The design of the terminal reward encourages desirable behavior and has an impact on the entire training environment through bootstrap. When $\sqrt{(x_U - x_T)^2 + (y_U - y_T)^2} < r_{\min} + r_T$, it is considered that the USV has reached the terminal, and USV agent receives the associated reward as shown follows,

$$R_T = k_T \frac{R_E}{r_T} \tag{24}$$

where, $r_{\min}$ is the minimum radius of the USV's own ship domain. r_T is the radius of the terminal. R_E is the size of the simulation environment.

(b) Collision reward: Avoiding obstacles is another important objective in training. Penalizing collisions can train the USV to maintain a safe distance from obstacles. When $\sqrt{(x_U - x_O)^2 + (y_U - y_O)^2} < r$ or $\sqrt{(x_U - x_O)^2 + (y_U - y_O)^2} < r_O$, it is considered a collision with the obstacle USV. The collision reward is designed as follows,

$$R_O = k_O \frac{R_E}{r_O V_O} \tag{25}$$

If $\sqrt{(x_U - x_O)^2 + (y_U - y_O)^2} < r + r_S$, it is considered a collision with a static obstacle. The collision reward is designed as follows,

$$R_S = k_S \frac{R_E}{r_S} \tag{26}$$

(c) COLREGs reward: COLREGs provide constraints on USV behaviors. Integrating COLREGs into the training process through rewards is an effective approach to instilling the trained USV agent with regulated avoidance behavior. Based on the requirements of COLREGs suitable for USV discussed in Section 2 of this paper, when $E \in \{E_3, E_4\}$ and $a \notin \{c|0° \leq c \leq 5°, c \in Z\}$,

$$R_C = k_C u_{CRI} \tag{27}$$

In this manner, if the USV violates COLREGs in high collision risk areas, it receives a more severe penalty. When $E \in \{E_3, E_4\}$ and $a \in \{c|0° \leq c \leq 5°, c \in Z\}$, or $E \in \{E_1, E_2, E_5, E_6\}$, the k_C is 0. The absence of a straight-line reward design is intentional, as it will be addressed in the subsequent section (d) to avoid redundancy.

(d) Seamanship reward: In the absence of obstacles or no obligation to give way, the own USV is expected to maintain a straight course as much as possible. Consequently, the following seamanship reward is designed to regulate the navigation behavior of the USV: If $a \notin \{0°\}$ and $u_{CRI} = 0$,

$$R_\delta = k_\delta \tag{28}$$

(2) Guidance reward.

The use of guiding rewards enriches the rewards received by the agent in the environment, thereby effectively facilitating training and addressing the issue of sparse rewards.

(a) Course reward: Choosing the behavior that brings the own USV closer to the terminal is deemed advantageous, and the course reward is formulated as follows,

$$R_\phi = \mathrm{k}_\varphi(\varphi_\mathrm{k} - \phi) \tag{29}$$

where, φ_k is a critical value that is determined based on the specific characteristics of the USV. $\phi = |\varphi_U - \vartheta_T|$.

(b) Better course reward: Actions that result in a smaller value of ϕ are considered favorable behaviors, and the enhanced course reward is formulated as follows,

$$R_{\Delta\phi} = \begin{cases} r_s & , \phi_t < \phi_{t'} \\ r_b & , \phi_t > \phi_{t'} \\ r_{e1} & , \phi_t = \phi_{t'}, \phi_t = 0 \\ r_{e2} & , \phi_t = \phi_{t'}, \phi_t \neq 0 \end{cases} \tag{30}$$

Therefore, the reward signal for USV collision avoidance training can be expressed as,

$$R = R_T + R_O + R_S + R_C + R_\delta + R_\varphi + R_{\Delta\phi} \tag{31}$$

Through the design of an appropriate state space, action space, and reward signal, the architecture of the algorithm update can be constructed effectively. Fig. 7 illustrates the complete training process.

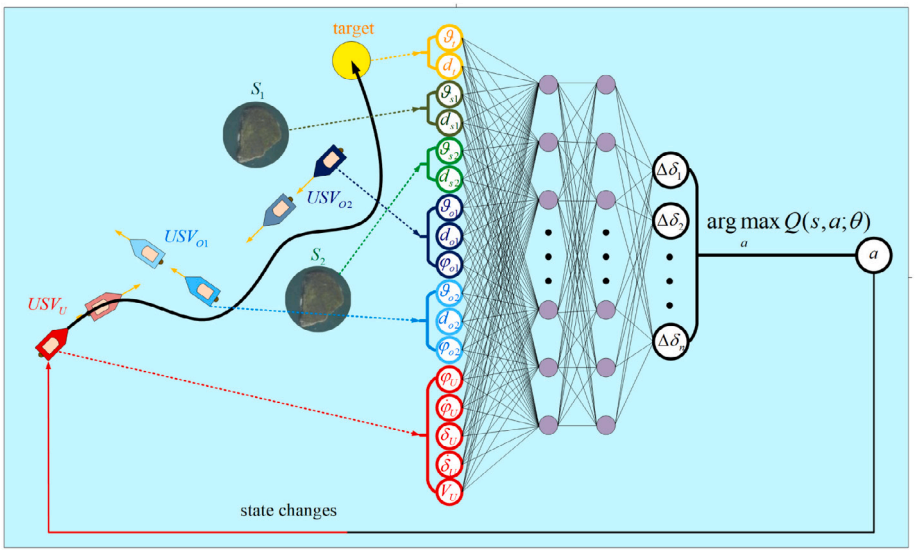


Fig. 7. Training Process.

The current USV state is fed into the neural network, and the action is determined by selecting the one with the maximum value. Subsequently, the environment is updated in accordance with this chosen action.

4. Experiments

4.1. Establishment of experimental environment

Fig. 8 shows the training environment utilized in this study. The upper-left section represents the sea area near Dalian, where specific parameters for the "LanXin" USV are determined. Additionally, the lower-left part of the figure presents a virtual environment, within the agent training environment enclosed by a red square of side length $R_E = 1000$m. The right part of the figure provides a detailed design diagram. Due to the fact that many studies have designed collision avoidance training environments that are static, this is not conducive to algorithm training. Therefore, in order to train an USV collision avoidance agent capable of dealing with more complex and uncertain environments, this study designed a stochastic obstacle generation method. Within this environment, the own USV maintains a fixed initial position at $(100, 100)$ in each episode. The yellow circle represents the terminal, which also has a fixed initial position at $(950, 950)$ with a radius of $r_t = 30$m. The blue USV symbolizes the obstacle USV, with 360 randomly assigned initial positions. These obstacle USVs are distributed around a circumference of a circle centered at $(500, 500)$, with an angular interval of $\mu_{EO} = 1°$ and a distance from the circle's center given by $d_{EO_i} = \frac{400\sqrt{2}V_{O_i}}{V_U}$. In this figure, for each initial position of the obstacle USV, there are eight corresponding positions that generate static obstacles. These positions are symmetric with the obstacle USV's route as the reference line. Additionally, at intervals of $\mu_{ES} = 60°$, there exists a potential obstacle position at a distance of d_{ES} from the center of the circle. Two more obstacle positions are located at a distance of $0.5d_{ES}$ from the center of the circle, perpendicular to the course of the obstacle USV. During each episode, two static obstacles with a radius of $r_T = 30$m are randomly selected from the eight static obstacle positions. The speed of the obstacle USV remains fixed at 3 m/s, while the own USV has 15 different speeds represented by $V_U = \{4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 8.5, 9, 9.5, 10, 10.5, 11\}$. Consequently, there are a total of $15 \times 360 \times 7 \times 8 = 302400$ distinct training environments.

Table 1
Parameters for "LanXin" USV.

Parameter	Value
Length between perpendiculars	7.04 m
Breadth	2.60 m
Speed	$\lesssim$ 35kn
Draft (full load)	0.32 m
Block coefficient	0.6976
Displacement (full load)	2.73 m^3
Rudder area	0.2091 m^2

For this experiment, the algorithm framework is implemented using TensorFlow 1.15 within Python 3.6. The training and testing environments are built using Unity. The GPU setup consists two RTX 2080Ti cards, which are utilized for accelerated computations throughout the training process.

4.2. Design of training process

The selected parameters of the "LanXin" USV are shown in Table 1. The corresponding USV model parameters are $T = 0.332$, $\alpha = 0.001$, $K = 0.707$, $\omega_n = 0.958$, $\zeta = 0.811$ and $k = 0.923$. The virtual USV in the Unity simulation environment uses exactly these model parameters.

Based on the state, action, and reward signal design in Section 3, $k_t = 30$, $k_O = -189$, $k_S = -175$, $k_C = -36$, $k_\delta = -41$, $k_\varphi = 170$, $\varphi_k = 2$, $r_s = 5$, $r_b = -11$, $r_{e1} = 19$, $r_{e2} = -18$. The hyperparameters for training are shown in Table 2. A learning rate of 0.0001 is selected to ensure training stability. A higher discount factor of 0.99 is employed, allowing the agent to consider future rewards from longer time steps, which corresponds to approximately 229 steps based on $\log_\gamma 0.1$. This choice results in challenging yet robust training. The experience replay buffer possesses a capacity of one million, without discarding any interaction samples. To facilitate exploration, Gaussian noise with a mean of 0 and a variance of 0.1 is added. The greedy value is set to 1. The batch size is 600. The parameter for adjusting sample priority is set to 0.4, and the parameter for importance sampling started at 0.5 and increased by 1.25×10^{-6} with each step until it reached 1. To avoid division by zero, a small parameter of 0.01 was introduced.

The neural network architecture, illustrated in Fig. 9, comprises multiple layers, each containing 900 neurons, including their respective noisy layers. The input layer for the state consists of 13 neurons,

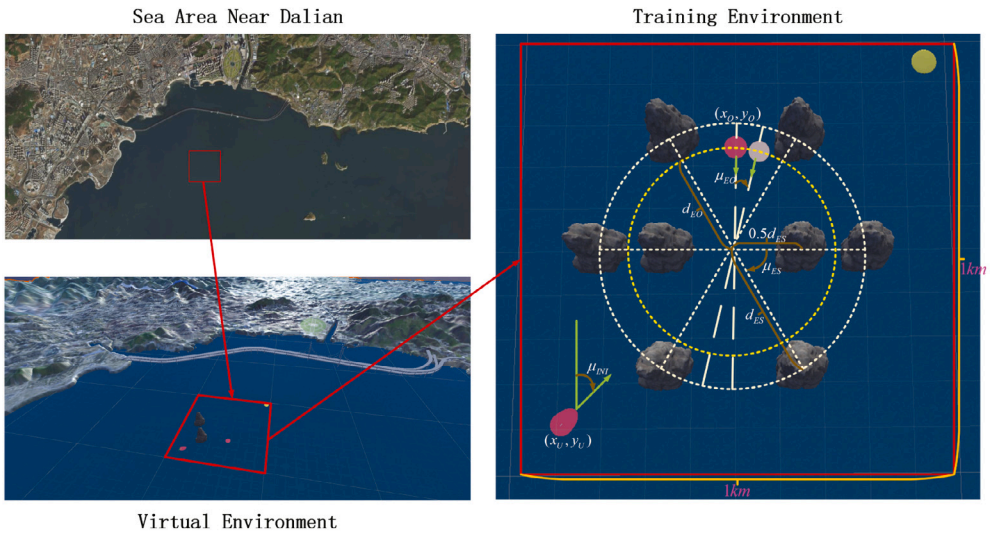


Fig. 8. Training environment.

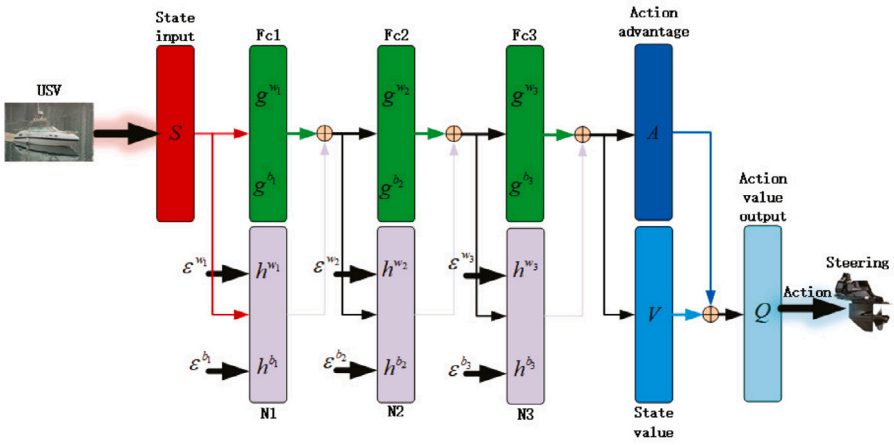


Fig. 9. Neural network architecture.

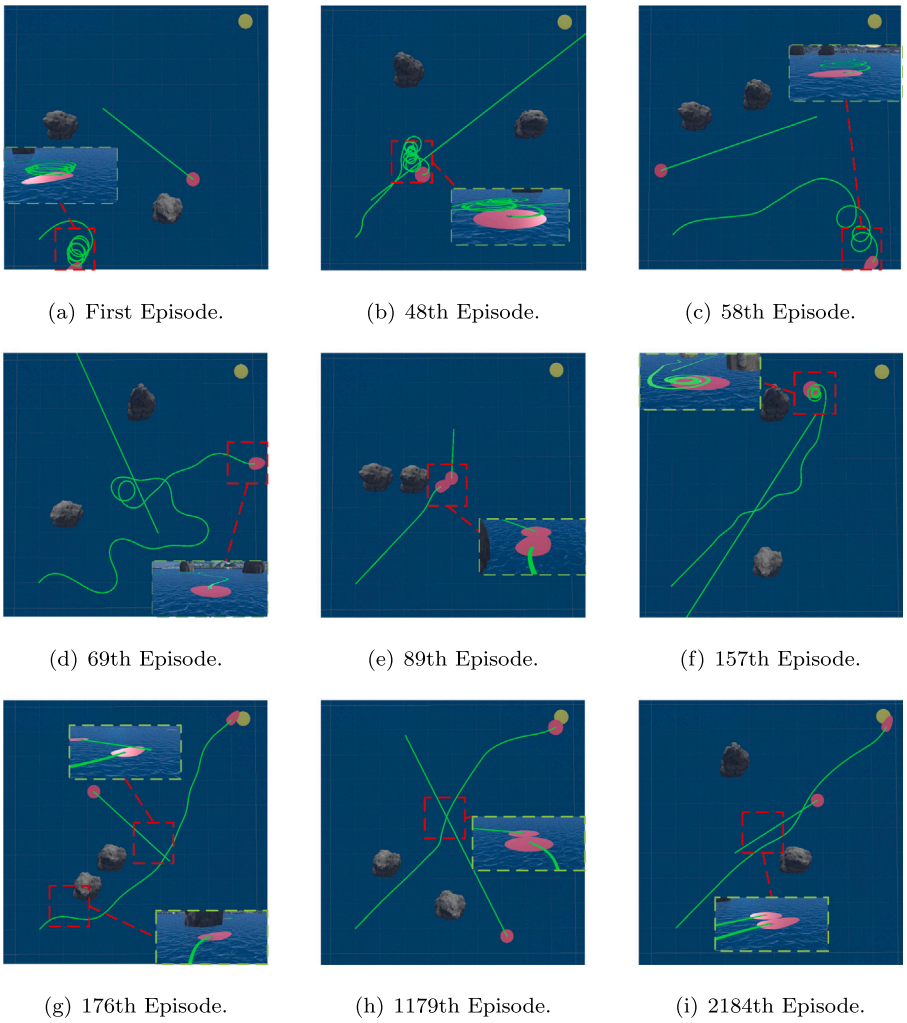
Table 2
Hyper parameters for training.

Hyperparameter	value
Learning rate (α)	1×10^{-4}
Discount factor (γ)	0.99
Target network update frequency	3000
Replay memory size	1×10^{6}
Batch size	600
Noise variance	0.1
Noise mean	0
Greedy value	1
Importance sampling	0.5
Linearly anneal of importance sampling	1.25×10^{-6}
Priority experience replay	0.4
Replay start size	2000

while the output layer for the action is composed of 11 neurons. The activation function is Leaky_ReLU, and the optimizer used is Adam.

4.3. Training

The training progress of the USV agent is illustrated in multiple figures. In Fig. 10(a), which represents the first episode, the USV displays unfamiliarity with the environment, resulting in a circling behavior attributed to random initial parameters of the neural networks. The changes in rudder angle and course are depicted in Fig. 11(a). The rudder angle closes 35 degrees, indicating the rotational behavior of the USV. However, such interactions are not conducive to later stages of training. Fortunately, the priority experience replay method effectively addresses this issue by selectively utilizing valuable experiences. Fig. 10(b) and Fig. 11(b) show the training progresses to the 48th episode, the USV realizes that circling near the starting point is not a desirable behavior by learning the guidance reward signal. As a result, it begins exploring areas further away from the initial position. This exploration behavior is evident in the training effects depicted in Fig. 10(c) (58th episode) and Fig. 10(d) (69th episode), where the USV continuously explores the unknown environment, accumulating interaction experiences, and their change in rudder angle and course are depicted in Fig. 11(c) and Fig. 11(d). In Fig. 10(e) (89th episode), a collision event prompts the USV to learn the importance of obstacle avoidance. Changes in rudder angle and course are depicted in Fig. 11(e), and it can be observed that in this training episode, the initial straight-line navigation successfully demonstrates the effectiveness of the guiding reward. However, during the collision avoidance maneuver, learning is still insufficient to respond effectively to the situation. Continuing the training process, Fig. 10(f) (157th episode) and Fig. 11(f)demonstrate the USV's improved understanding of obstacle avoidance behavior. The training effect of the 176th episode, depicted



(a) First Episode. (b) 48th Episode. (c) 58th Episode.

(d) 69th Episode. (e) 89th Episode. (f) 157th Episode.

(g) 176th Episode. (h) 1179th Episode. (i) 2184th Episode.

Fig. 10. The training effect of different training stages.

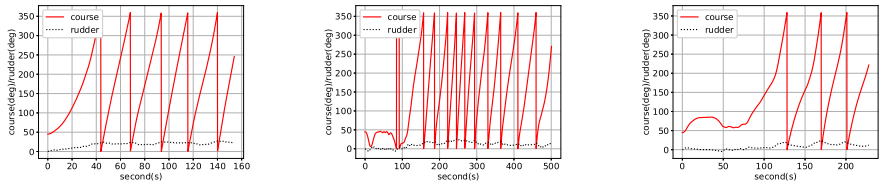
in Fig. 10(g) and Fig. 11(g), illustrates the USV's first successful reach of the terminal. This achievement highlights the success of designing the guiding reward once again. Finally, Fig. 10(h) and Fig. 10(i) represent the training effects of the 1179th and 2184th episodes, respectively, showcasing the continuous optimization of the agent's policy over an extended training period. Their change in rudder angle and course are depicted in Fig. 11(h) and Fig. 11(i). While the basic behaviors of collision avoidance have been successfully learned, there remain some issues such as unnecessary steering or COLREGs violations. These aspects will be progressively refined through extended learning periods.

Based on the comprehensive training process described above, model-free deep reinforcement learning algorithms do not require accurate modeling of the environment and USV. Through trial and error, it can better capture the intrinsic dynamics of the environment enabling more precise control of USV collision avoidance, which is difficult for other algorithms to achieve. Furthermore, acquiring real-world collision avoidance data for USV, compared to large ship, is challenging. Therefore, this paper adopts deep reinforcement learning algorithms, which generate behavioral experiences through action policies to perform rotations, exploration, and collision experiences. This approach facilitates a complete learning process through the target policy.

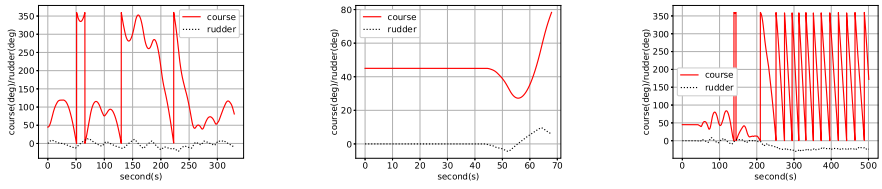
The average reward during the USV collision avoidance training is shown in Fig. 12. It is evident that, as training progresses and leveraging the appropriate reward signal designed in this study, the average reward curve quickly starts to rise and gradually improves, stabilizing at a relatively high level. In the later part of this curve, there are occasional drops in the average reward curve, occasional drops are noticeable, resulting from collisions or the inability to reach the destination; however, such instances are exceedingly rare.

To mitigate the impact of pseudorandom numbers, each algorithm is trained ten times with ten distinct random number seeds, and the outcomes are averaged. The average reward for each algorithm is then computed on 5000 episodes using a sliding average with a window size of 101. The comparison of average rewards for each algorithm is presented in Fig. 13. The algorithm that incorporating all proposed improvements exhibits the fastest initial improvement rate and achieves the highest average reward in the later stages. Especially, as shown in curves NPD3QNU and Noisy DQN in this figure, the inclusion of a noisy network significantly enhances the exploration capability of the USV, leading to a rapid increase in average reward. The dueling network architecture and priority experience replay contribute to enhancing algorithm performance, although the impact of each individual improvement alone is not very significant. Besides, the double DQN does not show significant improvement. In order to verify the effectiveness of double DQN, Fig. 14 depicts the results of ablation experiments for each improvement in this study. It is evident that the NPD3QNU algorithm outperforms the best training results. Removing any of these improvements would result in less effective training, particularly in cases involving the use of a noisy network, which exhibits a notable training effect.

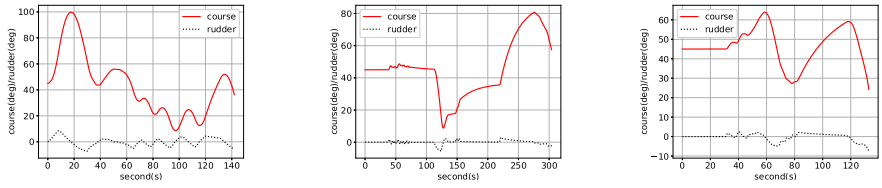
Since the utilization of noise networks in USV collision avoidance training is relatively uncommon, there have been no prior experiments



(a) First Episode. (b) 48th Episode. (c) 58th Episode.



(d) 69th Episode. (e) 89th Episode. (f) 157th Episode.



(g) 176th Episode. (h) 1179th Episode. (i) 2184th Episode.

Fig. 11. The course and rudder of different training stages.

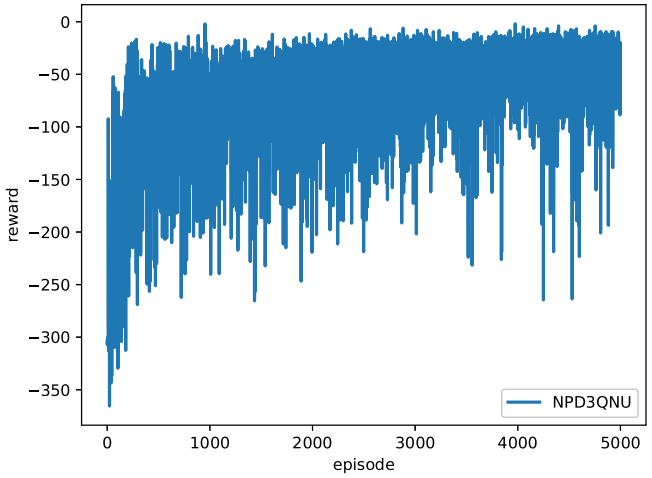


Fig. 12. Average reward of training.

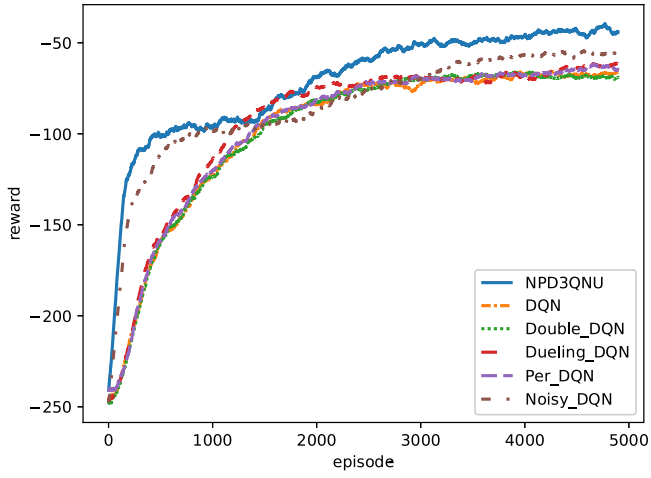


Fig. 13. Average reward under 10 seeds.

determining the most suitable approach for incorporating noise. In order to determine the optimal number of noise layers for collision avoidance training, this paper explores all eight possible configurations: all layers with noise, no noise for all layers, only one layer with noise, and only one layer without noise. Furthermore, the noise parameters are resampled at each step of the training. As shown in Fig. 15, it is evident that the average reward increases as more noise layers are added. Based on the collision avoidance experiment conducted in this study, it is determined that three noise layers provide the most effective results.

In Fig. 16, it is observed that incorporating collision avoidance characteristics notably improves the training effect. The dynamic area constraints method proposed in this paper effectively enhances the training outcome. While another method, state clipping, tailored specifically for USV collision avoidance, significantly enhances the training effectiveness.

4.4. Test

In this section, several representative encounter situations are selected to evaluate the training effectiveness of collision avoidance agents.

Fig. 17 illustrates an encounter situation involving the own USV and two static obstacles, representing encounter situation No. 4. One

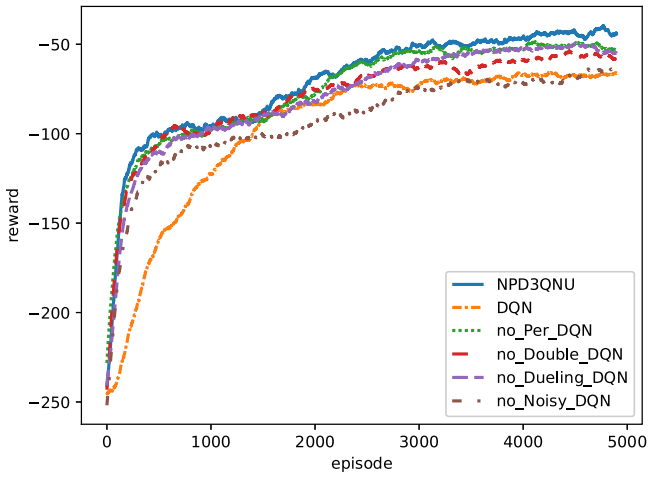


Fig. 14. Ablation study under 10 seeds.

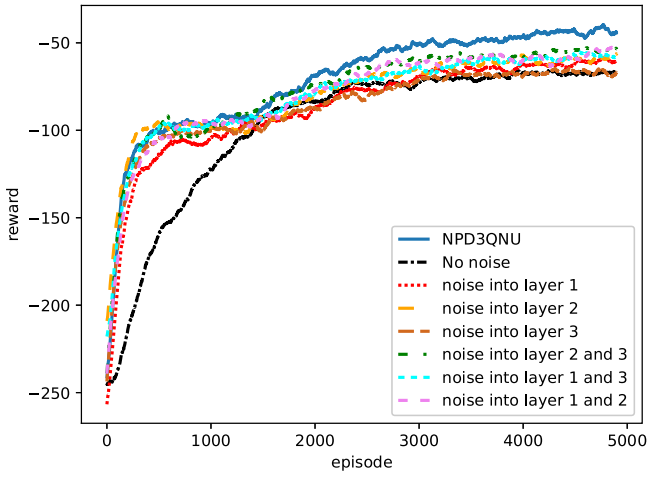


Fig. 15. Average reward of different noise addition ways under 10 seeds.

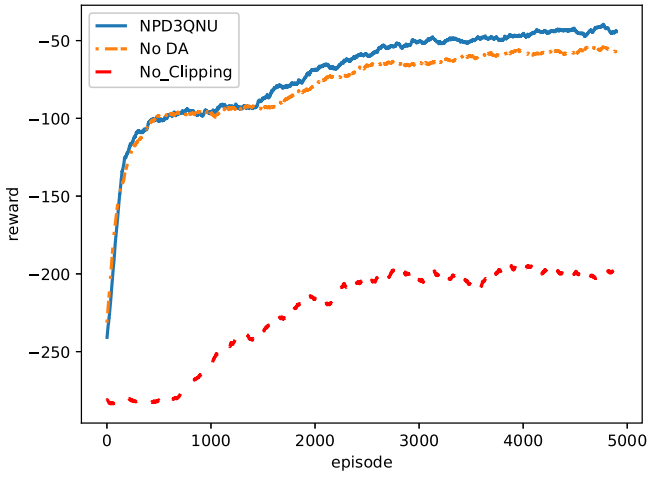


Fig. 16. Effect of dynamic area restriction and USV state clipping under 10 seeds.

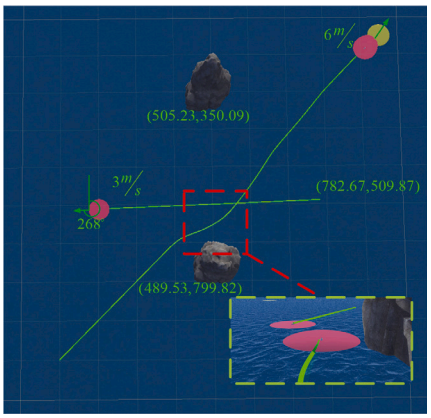
obstacle is in close proximity to the own USV, while the other is further away. The obstacle USV is located at coordinates $(782.67, 509.87)$ with a course of 268°. The static obstacles are positioned at $(505.23, 350.09)$ and $(489.53, 799.82)$. The own USV's speed is 6 m/s. For the NPD3QNU algorithm, Fig. 17(a) showcases the path taken during the test. Fig. 17(b) depicts the changes in the rudder angle δ_U and heading angle φ_U. Fig. 17(c) shows the variations in the distance between the own USV and the obstacles. Test 1 in Table 3 provides details of total course changes, decision steps, and rudder times for the NPD3QNU algorithm. Similarly, for the DQN algorithm, the test results are shown in Fig. 17(d), Fig. 17(e), Fig. 17(f), and test 2 in Table 3. The NPD3QNU reduces 77.04% in total course deviation, 91.20% in total course changes, 2.01% in decision steps, and 72.03% in rudder times compared to the pre-improvement algorithm. The improved algorithm exhibits fewer rudder and heading changes, resulting in better compliance with COLREGs.

Fig. 18 shows the No. 1 encounter situation. The obstacle USV is positioned at coordinates $(315.56, 460.80)$ with a course of 78°. There are also static obstacles located at $(562.37, 206.56)$ and $(722.94, 700.74)$. The own USV's speed in this scenario is 9 m/s. Fig. 18(a), Fig. 18(b), Fig. 18(c) and test 3 in Table 3 shows the test results for the NPD3QNU algorithm, and Fig. 18(d), Fig. 18(e), Fig. 18(f) and test 4 in Table 3 for DQN algorithm. The NPD3QNU reduces 22.51%, 2.95%, 0.01%, and 62.00% in total course deviation, total course changes, decision steps, and rudder times, compared to the pre-improvement algorithm. The improved algorithm performs exceptionally well in avoiding both static and dynamic obstacles, exhibiting no unprovoked steering behavior.

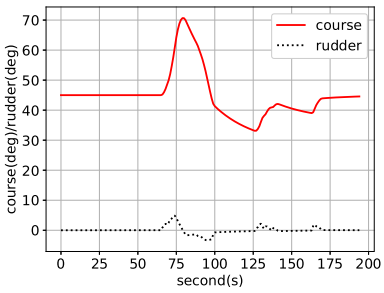
Fig. 19 shows an encounter situation where two static obstacles are far away and is in the No. 6 encounter situation. The obstacle USV is positioned at coordinates $(357.33, 607.51)$ with a course of 127°. There are also static obstacles located at $(202.24, 535.56)$ and $(617.22, 223.85)$. The own USV's speed in this scenario is 9.5 m/s. Fig. 19(a), Fig. 19(b), Fig. 19(c) and test 5 in Table 3 shows the test results for NPD3QNU algorithm, and Fig. 19(d), Fig. 19(e), Fig. 19(f) and test 6 in Table 3 for DQN algorithm. The NPD3QNU reduces 9.60%, 33.50%, −0.01%, and 69.70% in total course deviation, total course changes, decision steps, and rudder times, compared to the pre-improvement algorithm. The pre-improvement algorithm resulted in unexplained steering behavior at the beginning, which negatively impacted its collision avoidance performance. However, the improved algorithm, NPD3QNU, demonstrated exceptional performance in this environment. It effectively avoided collisions and maintained a smooth trajectory throughout the encounter, resulting in a significant improvement compared to the pre-improvement algorithm.

Fig. 20 shows an encounter situation where two static obstacles are not far away, and the obstacle USV is in the No. 3 encounter situation. The obstacle USV is positioned at coordinates $(684.62, 684.62)$ with a course of 225°. There are also static obstacles located at $(393.93, 606.07)$ and $(577.65, 789.78)$. The own USV's speed in this scenario is 6.5 m/s. Fig. 20(a), Fig. 20(b), Fig. 20(c) and test 7 in Table 3 shows the test results for NPD3QNU algorithm, and Fig. 20(d), Fig. 20(e), Fig. 20(f) and test 8 in Table 3 for DQN algorithm. The NPD3QNU reduces 42.40%, 61.10%, 1.64%, and 91.10% in total course deviation, total course changes, decision steps, and rudder times, compared to the pre-improvement algorithm. The improved algorithm for collision avoidance demonstrates better performance without unnecessary rudder actions.

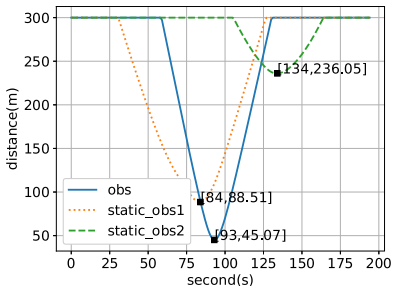
In addition, this paper also considers the collision avoidance ability in multi-USV encounter situations. The training and testing environment for multi-USV encounters is depicted in Fig. 21. The obstacle USVs are positioned at $(329, 171)$, $(670, 230)$ and $(750, 620)$ with courses of 315°, 320° and 225°. The static obstacles in this environment are located at $(250, 400)$ and $(500, 580)$. The obstacle USV's speed is 3 m/s, while the own USV's speed is 5.66 m/s. As shown in Fig. 21(a), the own USV encounters the first obstacle USV in the No. 4 encounter situation and follows the starboard maneuver as per the COLREGs. After successfully avoiding the obstacle, the own USV resumes its original navigation state, as shown in Fig. 21(b). Subsequently, as depicted in Fig. 21(c), the own USV encounters the second and third obstacle USVs in the No. 4 and No. 3 encounter situations, respectively. The own USV effectively



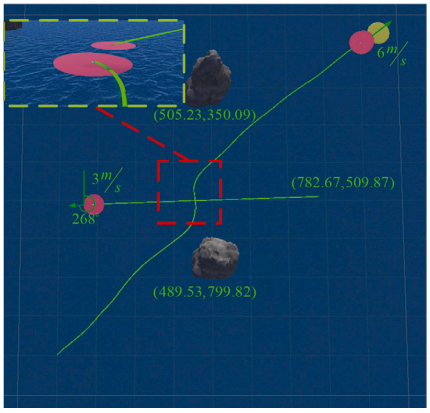
(a) Path planning by NPD3QNU.



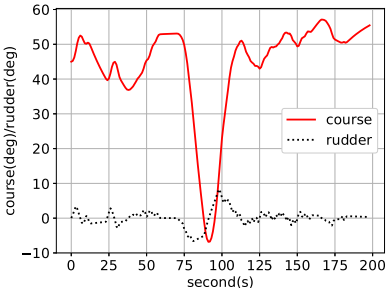
(b) Course and Rudder for NPD3QNU.



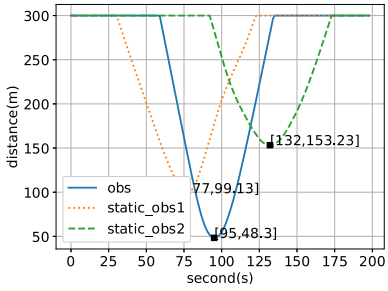
(c) Distance for NPD3QNU.



(d) Path planning by DQN.



(e) Course and Rudder for DQN.



(f) Distance for DQN.

Fig. 17. Test environment 1.

Table 3
Collision avoidance test result.

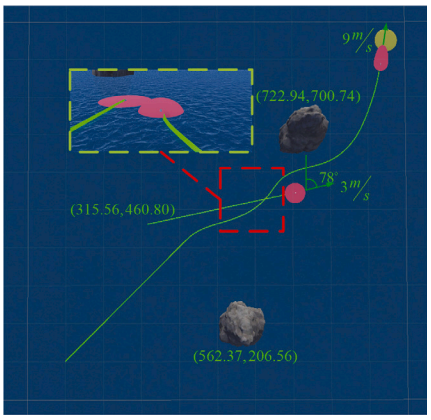
Test	COLREGs	Course deviation	Course changes	Decision steps	Rudder times
1	Comply	835.53	80.76	195	33
2	Against	3717.64	917.06	199	118
3	Comply	1601.97	176.36	136	43
4	Comply	2067.42	181.72	137	113
5	Comply	907.15	77.70	125	20
6	Comply	1003.44	118.57	124	66
7	Comply	971.96	81.17	180	10
8	Comply	1686.95	208.56	183	112

avoids the obstacles while complying with the COLREGs. Finally, in Fig. 21(d), the own USV navigates swiftly to the terminal.

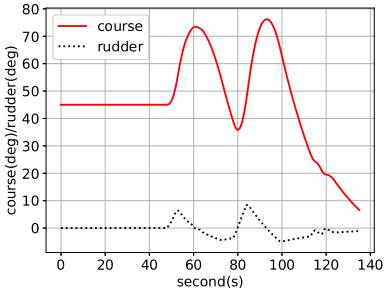
In summary, the improved algorithm demonstrates significant learning capabilities in both single and multi-obstacle USV encounters throughout this paper, despite the limited number of learning episodes. During training, it achieves higher average reward values and faster learning progress. During testing, it exhibits superior collision avoidance performance across various encounter situations. Overall, the algorithm showcases remarkable learning effects and achieves favorable collision avoidance outcomes in different situations.

5. Conclusion

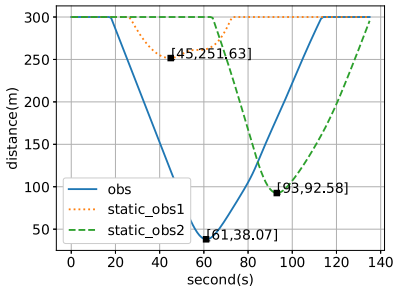
A comprehensive training and testing system for USV collision avoidance has been developed, incorporating the NPD3QNU algorithm



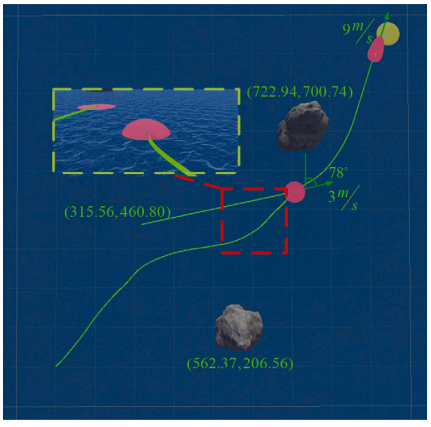
(a) Path planning by NPD3QNU.



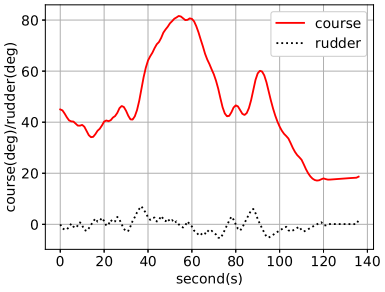
(b) Course and Rudder for NPD3QNU.



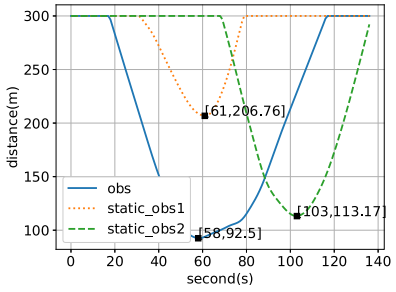
(c) Distance for NPD3QNU.



(d) Path planning by DQN.



(e) Course and Rudder for DQN.



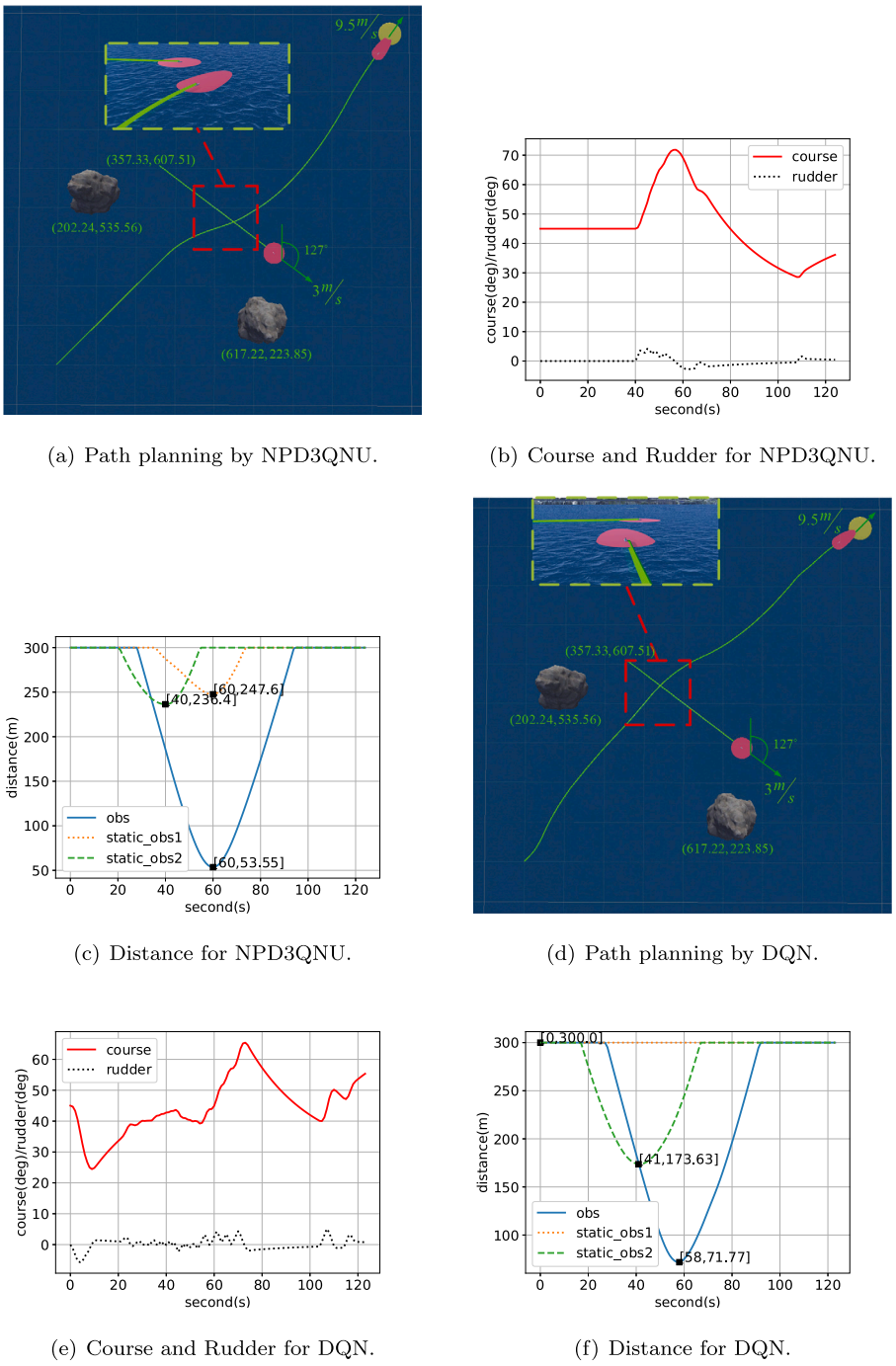
(f) Distance for DQN.

Fig. 18. Test environment 2.

specifically proposed for USVs. The agent demonstrates remarkable collision avoidance capabilities even with minimal prior knowledge. The system takes into account the maneuverability of the USV and the guidelines of COLREGs, considering the variations in shape and size of ship domain due to different USV speeds. To enhance exploration and accelerate early-stage learning, a noisy network is introduced. The system also incorporates priority experience replay, double learning, and dueling architecture to optimize sampling, training bias, and network design. A novel approach to reward signal, action, and state representation for USV training is designed, taking into account the unique characteristics of USVs. Additionally, novel methods such as state clipping and dynamic area restriction are proposed to improve learning efficiency and reduce computational complexity. Through extensive testing in various single or multi-obstacle USV encounter situations, the improved algorithm demonstrates superiority in terms of total course deviation, total course changes, rudder times, and compliance with COLREGs.

While this work has made some progress in USV collision avoidance, there are several issues that could be addressed in future research:

1. To enhance the collision avoidance agent's generalization ability, future studies will focus on enriching the training environment without making a distinction between scenarios with multiple obstacles and those with a single obstacle.



(a) Path planning by NPD3QNU.

(b) Course and Rudder for NPD3QNU.

(c) Distance for NPD3QNU.

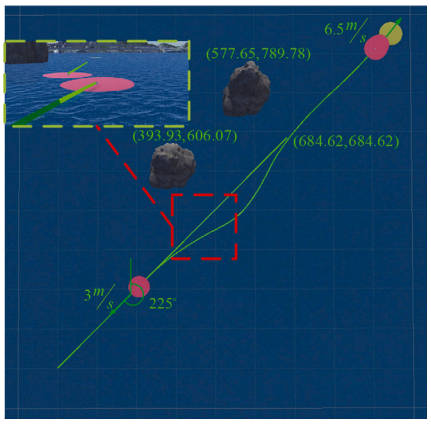
(d) Path planning by DQN.

(e) Course and Rudder for DQN.

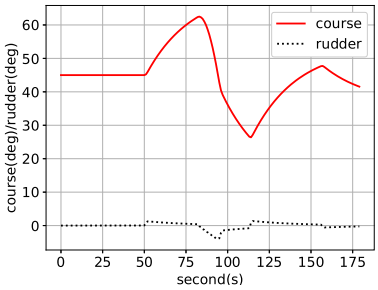
(f) Distance for DQN.

Fig. 19. Test environment 3.

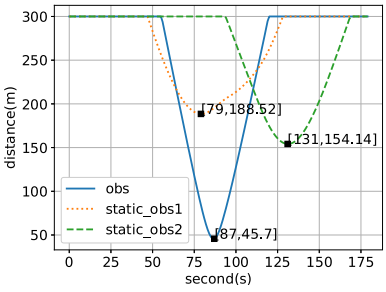
2. Similar to the No. 1 encounter situation, the action value function exhibits two peaks, corresponding to the options of turning to port or starboard. To address this issue, future research will explore the use of a distributional method that can provide a more refined differentiation of action values.
3. As shown in Fig. 22, the heights of the bar chart in the figure represent the average time of 10 random seed experiments, while the black line represents the magnitude of their variances. The improved algorithm, due to the utilization of the SumTree structure, exhibits increased time consumption compared to the original algorithms. To address this issue, future research will focus on optimizing the code efficiency to reduce computational overhead.
4. In the future, research will be conducted on how to effectively conduct real-USV experiments with the algorithm applied to "LanXin" USV. This includes exploring how to integrate the algorithm platform on USV and ensuring the safety of collision avoidance experiments with real obstacle USV. Addressing these crucial issues will enhance the practical significance of algorithm research.



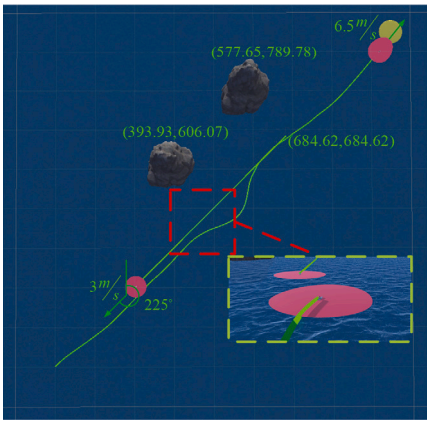
(a) Path planning by NPD3QNU.



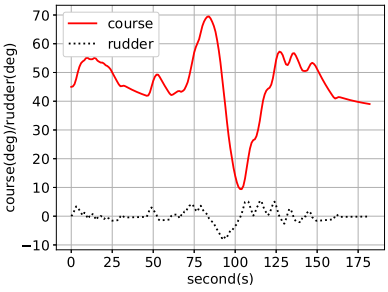
(b) Course and Rudder for NPD3QNU.



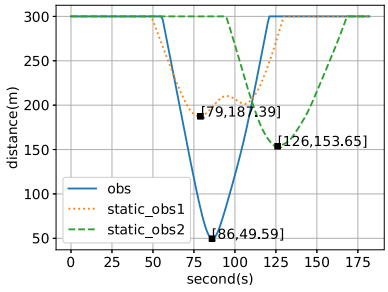
(c) Distance for NPD3QNU.



(d) Path planning by DQN.



(e) Course and Rudder for DQN.



(f) Distance for DQN.

Fig. 20. Test environment 4.

CRediT authorship contribution statement

Yunsheng Fan: Data curation, Conceptualization, Supervision. **Zhe Sun:** Methodology, Software, Writing – original draft. **Guofeng Wang:** Visualization, Review, Investigation.

Declaration of competing interest

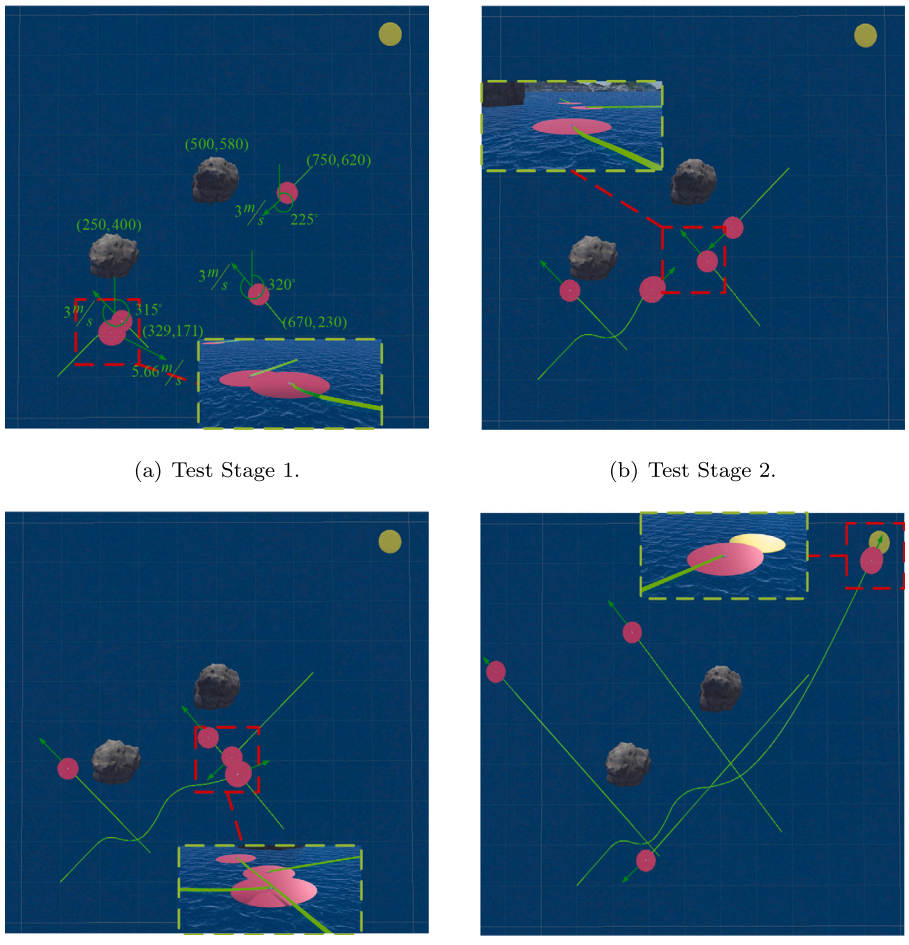
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This research was funded by “National Key Research and Development Program of China” (Grant number 2022YFB4301401)), “National Natural Science Foundation of China” (Grant number 61976033), “Pilot Base Construction and Pilot Verification Plan Program of Liaoning Province of China” (Grant number 2022JH24/10200029), “Key Development Guidance Program of Liaoning Province of China”(Grant number 2019JH8/10100100), “China Postdoctoral Science Foundation”(Grant number 2022M710569).



(a) Test Stage 1.

(b) Test Stage 2.

(c) Test Stage 3.

(d) Test Stage 4.

Fig. 21. Multi-USV encounter situation.

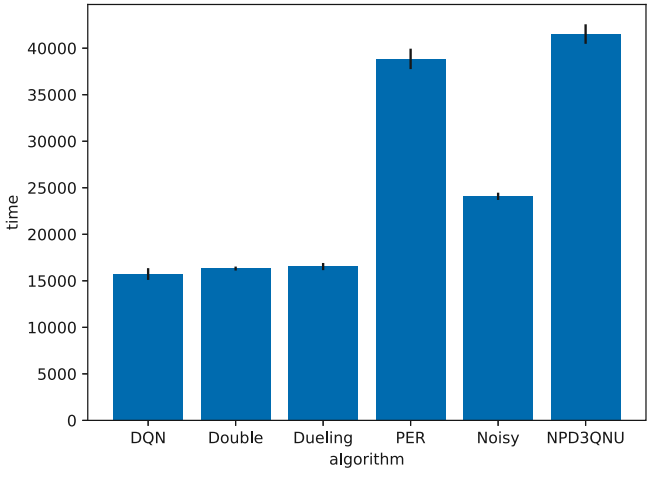


Fig. 22. Training time.

References

Bojesen, T.A., 2018. Policy-guided Monte Carlo: Reinforcement-learning Markov chain dynamics. Phys. Rev. E 98 (6), 063303.

Cheng, Y., Zhang, W.D., 2018. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. Neurocomputing 272, 63–73.

Chun, D., Roh, M.I., Lee, H.W., 2022. Deep reinforcement learning-based collision avoidance for an autonomous ship. Ocean Eng. 234, 109216.

Fan, Y.S., Sun, Z., Wang, G.F., 2022. A novel reinforcement learning collision avoidance algorithm for USVs based on maneuvering characteristics and COLREGs. Sensors 22 (6).

Fawzi, A., Balog, M., Huang, A., 2022. Discovering faster matrix multiplication algorithms with reinforcement learning. Nature 610 (7930), 47–53.

Fortunato, M., Azar, M.G., Piot, B., 2017. Noisy networks for exploration. p. 1706, ArXiv preprint arXiv:10295.

Fujii, Y., Tanaka, K., 1971. Traffic capacity. J. Navig. 24 (4), 543–552.

Guo, S., Zhang, X., Zhang, Y., 2020. An autonomous path planning model for unmanned ships based on deep reinforcement learning. Sensors 20 (2), 426.

Kim, M., Oh, J.H., 2016. Study on optimal velocity selection using velocity obstacle (OVVO) in dynamic and crowded environment. Auton. Robot. 40 (8), 1459–1470.

Kiran, B.R., Sobh, I., Talpaert, V., 2021. Deep reinforcement learning for autonomous driving: A survey. IEEE Trans. Intell. Transp. Syst 23 (6), 4909–4926.

Liang, C., Zhang, X., Watanabe, Y., 2021. Autonomous collision avoidance of unmanned surface vehicles based on improved a star and minimum course alteration algorithms. Appl. Ocean Res. 113, 102755.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., 2015. Continuous control with deep reinforcement learning. p. 1509, arXiv preprint arXiv:02971.

Liu, Y., Bucknall, R., Zhang, X., 2017. The fast marching method based intelligent navigation of an unmanned surface vehicle. Ocean Eng. 142, 363–376.

Liu, C., Mao, Q., Chu, X., 2019. An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning. Appl. Sci.-Basel 9 (6), 1057.

Mnih, V., Kavukcuoglu, K., Silver, D., 2015. Human-level control through deep reinforcement learning. Nature 518 (7540), 529–533.

Negenborn, R.R., Goerlandt, F., Johansen, T.A., 2023. Autonomous ships are on the horizon: Here's what we need to know. Nature 615 (7950), 30–33.

Rasekhipour, Y., Khajepour, A., Chen, S.K., 2016. A potential field-based model predictive path-planning controller for autonomous road vehicles. IEEE Trans. Intell. Transp. Syst 18 (5), 1255–1267.

Schaul, T., Quan, J., Antonoglou, I., 2015. Prioritized experience replay. p. 1511, ArXiv preprint arXiv:05952.
Shaobo, W., Yingjun, Z., Lianbo, L., 2020. A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method. Ocean Eng. 215, 107910.
Shen, H.Q., Hashimoto, H., Matsuda, A., 2019. Automatic collision avoidance of multiple ships based on deep Q-learning. Appl. Ocean Res. 86, 268–288.
Shi, B., Su, Y., Wang, C., 2019. Study on intelligent collision avoidance and recovery path planning system for the waterjet-propelled unmanned surface vehicle. Ocean Eng. 182, 489–498.
Silver, D., Schrittwieser, J., Simonyan, K., 2017. Mastering the game of go without human knowledge. Nature 550 (7676), 354–359.
Sun, X.J., Wang, G.F., Fan, Y.S., 2018. An automatic navigation system for unmanned surface vehicles in realistic sea environments. Appl. Sci. 8 (2), 193.
Sun, X.J., Wang, G.F., Fan, Y.S., 2020. Model identification and trajectory tracking control for vector propulsion unmanned surface vehicles. Electronics 9 (1), 22.
Sun, X.J., Wang, G.F., Fan, Y.S., 2021. A formation autonomous navigation system for unmanned surface vehicles with distributed control strategy. IEEE Trans. Intell. Transp. Syst. 22 (5), 2834–2845.
Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT press.
Szlapczynski, R., Szlapczynska, J., 2020. Review of ship safety domains: Models and applications. Ocean Eng. 145, 277–289.
Tam, C.K., Bucknall, R., 2010. Collision risk assessment for ships. J. Mar. Sci. TechnolNature 15 (3), 257–270.
Van, H.H., Guez, A., Silver, D., 2016. Deep reinforcement learning with double q-learning. In: Proc. AAAI Conf. Artif. Intell, Vol. 30, no. 1.
Vinyals, O., Babuschkin, I., Czarnecki, W.M., 2015. Grandmaster level in StarCraft II using multi-agent reinforcement learning. Nature 575 (7782), 350–354.
Wang, Z., Schaul, T., Hessel, M., 2016. Dueling network architectures for deep reinforcement learning. In: Int. Conf. Mach. Learn.. PLMR, pp. 1995–2003.
Wang, F.Y., Zhang, J., Wei, Q., 2017. PDP: Parallel dynamic programming. IEEE-CAA J. Automatica Sin 4 (1), 1–5.
Wenming, W., Jialu, D., Yihan, T., 2022. A dynamic collision avoidance solution scheme of unmanned surface vessels based on proactive velocity obstacle and set-based guidance. Ocean Eng. 248, 110794.
Woo, J., Woo, N., 2020. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. Ocean Eng. 199, 107001.
Xu, M.H., Liu, Y.Q., Huang, Q.L., 2007. An improved Dijkstra's shortest path algorithm for sparse network. Appl. Math. Comput. 185 (1), 247–254.
Xu, X., Lu, Y., Liu, X., 2020a. Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs. Ocean Eng. 217, 107704.
Xu, X., Lu, Y., Liu, G., 2022. COLREGs-abiding hybrid collision avoidance algorithm based on deep reinforcement learning for USVs. Ocean Eng. 247, 110749.
Xu, X., Pan, W., Huang, Y., 2020b. Dynamic collision avoidance algorithm for unmanned surface vehicles via layered artificial potential field with collision cone. J. Navig. 73 (6), 1306–1325.
Zhang, G., Han, J., Li, J., 2022. APF-based intelligent navigation approach for USV in presence of mixed potential directions: Guidance and control design. Ocean Eng. 260, 111972.
Zhang, X.Y., Wang, C.B., Liu, Y.C., 2019. Decision-making for the autonomous navigation of maritime autonomous surface ships based on scene division and deep reinforcement learning. Sensors 19 (18), 4055.
Zhang, J., Zhang, C., Chien, W.C., 2021. Overview of deep reinforcement learning improvements and applications. J. Internet Technol. 22 (2), 239–255.