Ocean Engineering 284 (2023) 115208

Contents lists available at ScienceDirect

Ocean Engineering

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

Deep reinforcement learning with dynamic window approach based collision avoidance path planning for maritime autonomous surface ships

Chuanbo Wu [a,c], Wangneng Yu [a,b,c,*], Guangze Li [a,c], Weiqiang Liao [a,b,c]

[a] School of Marine Engineering, Jimei University, Xiamen 361021, PR China
[b] Fujian Provincial Key Laboratory of Naval Architecture and Ocean Engineering, Xiamen 361021, PR China
[c] Fujian Engineering and Research Center of Offshore Small Green Intelligent Ship System, Xiamen 361021, PR China

ARTICLE INFO

Keywords:
Ship collision avoidance
Dynamic window approach
Deep reinforcement learning
Maritime autonomous surface ships

ABSTRACT

Automatic obstacle avoidance technology is one of the key technologies for ship intelligence. The purpose of this paper is to investigate the obstacle avoidance problem of maritime autonomous surface ships(MASS) in a complex offshore environment, and an obstacle avoidance strategy based on deep reinforcement learning and a dynamic window algorithm was proposed. To solve the collision avoidance problems that may occur during intelligent ship navigation, the action space of the proximal policy optimization (PPO) algorithm is defined according to the description of ship motion by linear and angular velocity in the dynamic window approach (DWA). The maximum detection distance of the MASS is utilized to construct the ship safety domain, which determines the state space containing the information of this ship and the nearest obstacle. To solve the problem of sparse reward, the reward function of the PPO is improved by combining the evaluation functions for distance, velocity and heading in the DWA. To verify the effectiveness of the algorithm, simulation experiments are performed in various situations. It is also shown that the improved algorithm can make the optimal collision avoidance decision from the complex environment and can effectively realize autonomous collision avoidance path planning for the MASS.

1. Introduction

With the rapid development of artificial intelligence technology and information technology, as well as the intelligent application of big data, the development of ship intelligence has improved (Aslam et al., 2020). Due to the advantages of high intelligence and autonomy, the MASS have a wide range of application needs in both military and civilian fields (Zhang et al., 2021a). Among them, electrically driven MASS applicable offshore can provide comprehensive high-end technical equipment for the tourism industry and marine resource development (Yu et al., 2017, 2018). Autonomous obstacle avoidance technology is one of the key technologies to achieve ship intelligence (Munim et al., 2020). Research on collision avoidance path planning algorithms can effectively solve the problem of how to safely and rapidly avoid various obstacles in the complex environment of the MASS.

Since offshore areas are complex and variable, there are not only static obstacles such as shorelines, islands, reefs, wrecks and beacons but also dynamic unknown obstacles and other ships, which obviously increases the difficulty of obstacle avoidance. Therefore, the investigation of the MASS obstacle avoidance needs to comprehensively consider the influence of dynamic obstacles and static obstacles on ship navigation. Many collision avoidance path planning algorithms have been proposed in the field of unmanned surface vessels (USVs) (Wenming et al., 2022; Zhong et al., 2022; Xu et al., 2022). The development of collision avoidance algorithms has evolved from traditional path planning algorithms such as the A* algorithm (He et al., 2022), artificial potential field (APF) algorithm (Sang et al., 2021) and rapid exploration random tree algorithm (RRT) (Xiong et al., 2020) to intelligent optimization algorithms (Xia et al., 2020, 2021).

The common traditional path generation algorithms include A*, APF, RRT, etc. These algorithms have been used in the field of ship collision avoidance with good results. Liang et al. (2021) proposed an improved A* algorithm considering the minimum course alteration algorithm (MCA), which can achieve safe navigation under multivessel encounter situations by optimizing the planned route under the International Regulations for Preventing Collisions at Sea (COLREGS) constraint by retaining important waypoints. Ju et al. (2020) effectively solved this problem by improving the A* algorithm. Chen et al. (2021) proposed an improved APF algorithm that can realize global optimal path planning by combining the improved ant colony optimization

* Corresponding author.
E-mail address: wnyu2007@jmu.edu.cn (W. Yu).

https://doi.org/10.1016/j.oceaneng.2023.115208
Received 15 March 2023; Received in revised form 31 May 2023; Accepted 18 June 2023
Available online 1 July 2023

(ACO) algorithm. Zhu et al. (2022) added the influence of the distance of the close point to the approaching point (DCPA) and the time to the close point of the approaching point (TCPA) on the ship collision hazard to the repulsive model of the APF algorithm, which can realize ship automatic collision avoidance effectively with the minimum parameter setting. Zhang et al. (2019) proposed an adaptive hybrid dynamic step and target attraction-RRT algorithm, which has good improvement in narrow water passage ability and open water navigation speed. 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.

With the development of AI, swarm intelligence algorithms and deep reinforcement learning algorithms are gaining increasing attention in the field of ship collision avoidance. Among them, ACO, the particle swarm optimization algorithm (PSO), and the deep reinforcement learning algorithm (DRL) are commonly used. Guo et al. (2020a) proposed a chaotic shared learning particle swarm optimization algorithm for solving the diffusion traveling salesman problem (TSP) and nonlinear multiobjective problems to obtain the global path. Wang et al. (2021) proposed an improved bacterial foraging optimization algorithm (BFOA), which solves the invalid search and repeated search problems of the traditional BFOA by the optimal swim search method. Zhang et al. (2021b) improved ACO from the perspective of trajectory smoothing and improving multiobjective solutions, which can plan optimal paths considering weather conditions. These methods yield favorable results for global static path planning problems, which involve planning based on a priori environmental data. However, practical application scenarios that involve only local information are distinct. Numerous uncertainties and unpredictable factors, such as weather conditions, sea state, the presence of other ships, sensor errors, and more, can influence ship perceptibility and motion state. Moreover, global path planning has limitations in practical applications, making it challenging to handle moving obstacles in dynamic environments and avoid collisions with other ships.

Model-free reinforcement learning methods can learn the optimal strategy by interacting with the environment and thus adapt to complex systems well. Hsu and Gau (2022) propose a reinforcement learning approach for collision avoidance and investigate optimal trajectory planning for unmanned aerial vehicle (UAV) communication networks. Chen et al. (2019) proposed a path planning and manipulating approach based on Q-learning, which can drive a cargo ship by itself without requiring any input from human experiences. The reinforcement learning algorithm is used for path optimization to generate a feasible path that can be tracked by the vehicle (Yoo and Kim, 2016).

DRL algorithms combined with neural networks can effectively solve the end-to-end decision problem. Li et al. (2021) used the APF to improve the action space and reward function of the deep Q-learning network (DQN), and the use of the improved DRL algorithm can effectively achieve autonomous collision avoidance path planning for USVs. Guo et al. (2020b) proposed a DRL-based autonomous path planning model that uses the deep deterministic policy gradient (DDPG) algorithm to learn to train the optimal algorithm to learn to train the best action strategy by using ship data provided by an automatic identification system (AIS). A composite learning method is proposed based on an asynchronous advantage actor–critic (A3C) algorithm, a long short-term memory neural network (LSTM) and Q-learning (Xie et al., 2020). A DRL-based path generation algorithm is proposed to determine the avoidance time and generate avoidance paths for the most dangerous ships in terms of collision risk in compliance with COLREGs (Chun et al., 2021). A DRL-based collision avoidance decision algorithm is proposed by combining a visual grid map to generate environmental information (Woo and Kim, 2020). Automatic obstacle avoidance algorithms based on deep reinforcement learning often encounter the issue of reward sparsity. This problem becomes more pronounced in complex sea environments where the reward function may not be well-designed to guide agent in avoiding obstacles effectively. Consequently, the performance of automatic obstacle avoidance algorithms can be affected. Additionally, previous studies on automatic obstacle avoidance have primarily focused on achieving the goal of safe avoidance without conducting comprehensive comparative analyses of obstacle avoidance paths. Therefore, further research is necessary to delve into the design of the reward function and conduct in-depth analyses of obstacle avoidance paths to enhance and optimize automatic obstacle avoidance algorithms.

We propose a dynamic obstacle avoidance algorithm based on proximal policy optimization (PPO) and the dynamic window approach (DWA). PPO is a DRL based on the policy gradient, which solves the problem of difficulty in determining the step size in the policy gradient algorithm. DWA can be applied to obstacle avoidance of the MASS in complex environments by establishing a dynamic window of linear and angular velocities and combining the evaluation function containing distance, velocity and heading to select the optimal velocity combination. To obtain the improved DRL algorithm, the action state space and reward function of the PPO algorithm are improved by using the idea of a dynamic window. Finally, simulation experiments are used to verify the effectiveness of the improved algorithm in various situations. The main contributions of this study are summarized as follows:

(1) A new DRL method is designed that can combine sensor information to address the obstacle avoidance problem of the MASS.

(2) The idea of DWA is utilized to improve the PPO algorithm action state space and reward function to solve the PPO sparse reward conundrum.

(3) The proposed improved algorithm solves the problem of obstacle avoidance path planning for the MASS in various dynamic and complex environments.

The rest of the paper is organized as follows. The problem of obstacle avoidance of intelligent ships in complex environments and briefly describes the ship kinematic model was described in Section 2. The improvement of DWA and PPO algorithms is mainly carried out in Section 3. Section 4 presents the design of simulation experiments and the analysis of experimental results. Section 5 presents the conclusion and future work. The rest of the paper is organized as follows. The problem of obstacle avoidance of the MASS in complex environments and a brief description of the ship kinematic model are described in Section 2. The improvement of the DWA and PPO algorithms is mainly carried out in Section 3. Section 4 presents the design of the simulation experiments and the analysis of the experimental results. Section 5 presents the conclusion and future work.

2. Problem description and ship motion model

2.1. Problem description

The path planning problem is divided into global path planning and local path planning. In the dynamic environment, the path planning and collision avoidance of the MASS can be described as a local path planning problem, the purpose of which is to determine the optimal conditions in the current state. The applicable scenario of the MASS in this paper is the offshore sea. Due to the complex and changing environment of ship navigation, there are not only static obstructions such as shorelines, islands, reefs, wrecks and navigation markers but also dynamic unknown obstructions and other ships, which bring danger to the safe navigation of the MASS. The DRL method is a combination of deep learning (DL) and reinforcement learning (RL) with strong perception and decision-making ability, which can provide a solution to the local optimal path planning problem of the MASS.

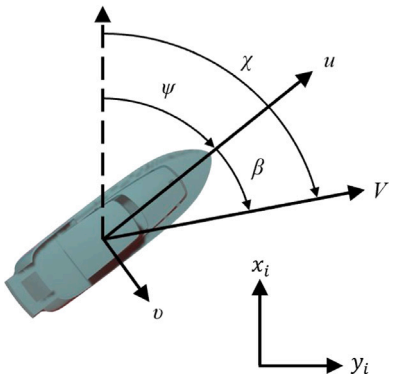


Fig. 1. The coordinate system.

2.2. Ship motion model

The conventional mechanistic modeling method uses the momentum principle and momentum moment principle of Newtonian rigid body mechanics, which correspond to translation and rotation in 3 directions. Therefore, the ship has 6 degrees of freedom. In practical applications, the motion of the ship does not change rapidly, so only three degrees of freedom are considered (surge, sway, and yaw motion). Therefore, the notation proposed by Fossen was adopted (Fossen et al., 2003). The kinematic model of the MASS can be described by Eq. (1), where ν and η are defined as the velocity vector and position vector, respectively. $R(\eta)$ is the rotation matrix from the body-fixed frame to the inertial frame.

$$\dot{\eta} = R(\eta)\nu \tag{1}$$

$$\nu = (u, v, \omega)^T \tag{2}$$

$$\eta = (x_i, y_i, \psi)^T \tag{3}$$

$$R(\eta) = \begin{bmatrix} cos(\eta) & -sin(\eta) & 0 \\ sin(\eta) & cos(\eta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

The coordinate system used in this paper is shown in Fig. 1, where x_i and y_i are the northward and eastward positions of the MASS in the inertial frame, respectively. u, v and ω are the surge, sway, and yaw motions of the vehicle in a body-fixed frame, respectively. u is the motion velocity along the x-axis direction; v is the motion velocity along the y-axis direction; and ω is the rotational angular velocity around the z-axis direction. The heading angle of the ship is defined as ψ, while the course angle and the side slip angle are denoted as χ and β, respectively. According to the definition, the side slip angle β can be calculated by the following equation $\beta = asin(\frac{v}{V})$, where V represents the total speed of the ship. The course angle can be described by the sum of η and β, as the expression $\chi = \eta + \beta$.

According to the research content of this paper, the ship motion is simplified. Since it is imprecise to assume the ship's trajectory as a straight line, the arc-shaped motion trajectory of the ship at the current velocity combination (v_i, ω_i) is taken as the predicted trajectory, as shown in Fig. 1, whereas the arc radius of the ship's trajectory is calculated as shown in Eq. (5).

$$M_R^i = \frac{v_i}{\omega_i} \tag{5}$$

where M_R^i predicts the circular radius of the ship's trajectory when $\omega_i = 0$ for linear motion and (v_i, ω_i) is a set of desired velocity pairs(see Fig. 2). When the angular velocity of the ship is not equal to zero, the

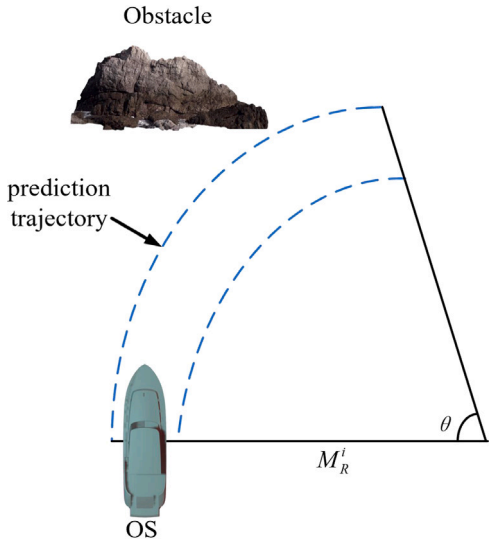


Fig. 2. Circular prediction trajectory.

coordinates of the trajectory sailing in Δt time are calculated as shown in Eq. (6) below.

$$\begin{cases} x_{t+1} = x_t + \frac{v}{\omega}\sin(\theta_t + \omega\Delta t) - \frac{v}{\omega}\sin\theta_t \\ y_{t+1} = y_t - \frac{v}{\omega}\cos\theta_t - \frac{v}{\omega}\cos(\theta_t + \omega\Delta t) \\ \psi_{t+1} = \psi_t + \omega\Delta t \end{cases} \tag{6}$$

where (x_t, y_t) and (x_{t+1}, y_{t+1}) are the current position of the ship and the position after Δt, respectively.

3. Collision avoidance method based on DWA and PPO

3.1. Dynamic window algorithm

The dynamic window method is essentially a reactive collision avoidance algorithm that samples multiple velocity combinations (u, ω) from the range of linear velocity $[u_{min}, u_{max}]$ and angular velocity $[\omega_{min}, \omega_{max}]$ allowed by the predicted Δt time and to simulate and calculate the trajectory of each velocity combination based on the ship's motion model and environmental information (Peng et al., 2020). Then, the evaluation value of each trajectory is estimated based on the evaluation function, and the optimal speed combination $(u_{best}, \omega_{best})$ is selected to change the ship's heading and speed. The above steps are cycled continuously to reach the target point while avoiding the obstacle.

$$V_s = \{(u, \omega) \mid u \in [u_{min}, u_{max}], \omega \in [\omega_{min}, \omega_{max}]\} \tag{7}$$

Fig. 3 shows a schematic diagram of the search space for the dynamic window method. In Eq. (7), V_s denotes the velocity space under the velocity and angular velocity constraints, u is the linear velocity and ω is the angular velocity.

$$V_a = \{(u, \omega) \,||u| \le \sqrt{2 \cdot dist(u, \omega) \cdot u_{min}}, \; |\omega| \le \sqrt{2 \cdot dist(u, \omega) \cdot \omega_{min}}\} \tag{8}$$

In Eq. (8), V_a denotes the set of ambient allowable velocities and $dist(u, \omega)$ is the distance from the obstacle.

$$V_d = \{(u, \omega) \,|u \in [u_o - \dot{u}\Delta t, u_o + \dot{u}\Delta t] \cap \omega \in [\omega_o - \dot{\omega}\Delta t, \omega_o + \dot{\omega}\Delta t]\} \tag{9}$$

In Eq. (9), V_d denotes the current dynamic window, which indicates the velocity space that can be reached after Δt from the current velocity (u_o, ω_o).

$$V_r = V_s \cap V_a \cap V_d \tag{10}$$

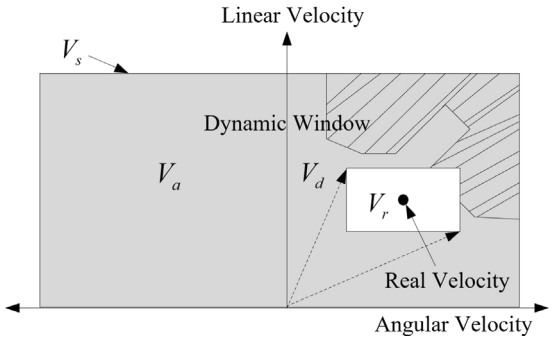


Fig. 3. The search space of the dynamic window algorithm.

In Eq. (10), V_r denotes the set of desirable velocities at the next moment.

The predicted trajectory of the vessel in the prediction time and the formed velocity space are evaluated and selected to choose the optimal velocity set (u_{best}, ω_{best}). In terms of evaluation rules, the higher the speed needed, the farther the distance relative to the obstacle is, and the smaller the difference with the target heading is, the better, so the objective function is proposed for selection.

$$G(u,\omega) = \alpha \cdot dist(u,\omega) + \beta \cdot vel(u,\omega) + \gamma \cdot head(u,\omega) \tag{11}$$

Eq. (11) is the objective function, which aims to select the optimal velocity pair (u_{best}, ω_{best}) in this velocity space after determining V_r, where $dist(u,\omega)$ is the distance to the obstacle, $vel(u,\omega)$ is the linear velocity magnitude, and $head(u,\omega)$ is the deviation between the heading angle and the azimuth of the target.

3.2. PPO algorithm

The PPO algorithm is a deep reinforcement learning algorithm based on the AC framework (Schulman et al., 1999), which obtains the optimal policy based on the policy gradient (PG) (Schulman et al., 2017). The PPO algorithm consists of a critic network (CN) and an actor network (AN). The CN learns the relationship between the environment and the reward and obtains the current action dominance function, and the AN continuously adjusts the parameters of the policy according to the action dominance function The AN continuously adjusts the parameters of the strategy to increase the probability of obtaining high rewards.

PPO is a policy gradient algorithm, that interacts with the environment by using the policy $\pi_\theta(a \mid s)$, calculates the expected reward $J(\theta)$ of its policy with the obtained data, calculates the gradient to update the policy parameter $\theta \rightarrow \theta'$, and finally repeats the above process with the updated policy $\pi'_\theta(a \mid s)$. The expected reward $J(\theta)$ of the strategy and the gradient $\nabla J(\theta)$ of the strategy are calculated as shown in Eq. (12) and (13).

$$J(\theta) = E_{\tau\sim\pi_\theta(\tau)}[\sum_{t=1}^{T} R(s_t,a_t)] = E_{\tau\sim\pi_\theta(\tau)}[R(\tau)] \tag{12}$$

$$\nabla J(\theta) = E_{\tau\sim\pi_\theta(\tau)}[(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t \mid s_t))R(\tau)] \tag{13}$$

The large variance of the gradient estimation in the strategy gradient approach can prevent the strategy from moving in a better direction. Therefore, the first problem that needs to be solved is to reduce the variance of the gradient estimation. The PPO algorithm is based on the actor-critic (AC) framework, and formulating the reward based on the dominance function can be a good solution to this problem. The dominance function can be used to evaluate the advantage of an action over other actions in the current state so that good actions have positive reward values and bad actions have negative reward values. The formula for the advantage function $A^\pi(s,a)$ is shown in Eq. (14).

$$\begin{cases} Q^\pi(s,a) = \sum E_{\pi\theta}[R(s_t,a_t) \mid s,a] \\ V^\pi(s) = \sum E_{\pi\theta}[R(s_t,a_t) \mid s] \\ A^\pi(s,a) = Q^\pi(s,a) - V^\pi(s) \end{cases} \tag{14}$$

where $Q^\pi(s,a)$ is the value of the evaluated action and represents the expectation of the sum of the rewards of the intelligence in this state all the way to the final state after choosing this action. $V^\pi(s)$ is the value of the evaluated state and represents the expectation of the sum of the rewards of the intelligence in this state all the way to the final state.

The principle of the PPO algorithm is to improve the utilization of samples by obtaining samples from the old strategy $\pi_{\theta old}(a \mid s)$ through a significant sampling method. Meanwhile, the parameters of the new strategy $\pi_\theta(a \mid s)$ are periodically updated into the old strategy $\pi_{\theta old}(a \mid s)$. As the training proceeds, the strategy entropy between the two strategies increases. For this reason, the update step needs to be determined, and when the update step is not appropriate, the updated parameters corresponding to the strategy payoff function will be reduced, and the more it is learned, the worse it gets, leading to the final failure of the algorithm to converge. This problem can be solved by restricting the change range $r_t(\theta)$ of the action output probability of the old and new strategy networks to a certain region through Eq. (15), which is the core of the PPO algorithm. The framework of the PPO algorithm is shown in Fig. 4.

$$r_t(\theta) = \frac{\pi_\theta(a_t,s_t)}{\pi_{\theta old}(a_t,s_t)} \tag{15}$$

The expression of the objective function of the PPO algorithm is

$$L^{CLIP}(\theta) = \hat{E}_t[min(r_t(\theta), clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t)] \tag{16}$$

where ε is the hyperparameter, usually set to 0.2, and $\hat{A}$ is the dominance function. When $\hat{A} > 0$, it means that this action is better than the average action, so the probability of selecting this action increases; when $\hat{A} < 0$, it means that this action is worse than the average action, so the probability of selecting this action decreases, but the probability distribution of the actions obtained by the network cannot be too far apart, so it is truncated at $1+\varepsilon$ and $1-\varepsilon$, respectively, to limit the magnitude of the policy update. The objective function restriction range is shown in Fig. 5.

3.3. Proposed DWA-PPO algorithm for collision avoidance

For the task of obstacle avoidance path planning for the MASS sailing in offshore waters, the PPO algorithm and the DWA algorithm can plan safe and less energy-intensive routes in complex environments. The advantage of the PPO algorithm is that it performs well for continuous control problems. The data collected by the deep reinforcement learning algorithm are temporal in nature, and the environmental states are correlated. The neural network can solve the continuous state action space problem through its powerful characterization capability, but it also suffers from slow convergence, sparse rewards, and the tendency to fall into deadlock regions. The DWA algorithm has the advantages of low computational complexity, high efficiency, and good real-time obstacle avoidance effect, but it also suffers from insufficient foresight and poor performance for poor obstacle avoidance or dynamic obstacles and choosing the next best path instead of the global optimal path every time. Therefore, the PPO algorithm is improved by the DWA algorithm so that the improved algorithm absorbs the advantages of DWA and improves the defects of PPO. The structure of the DWA-PPO fusion improvement algorithm is shown in Fig. 6.

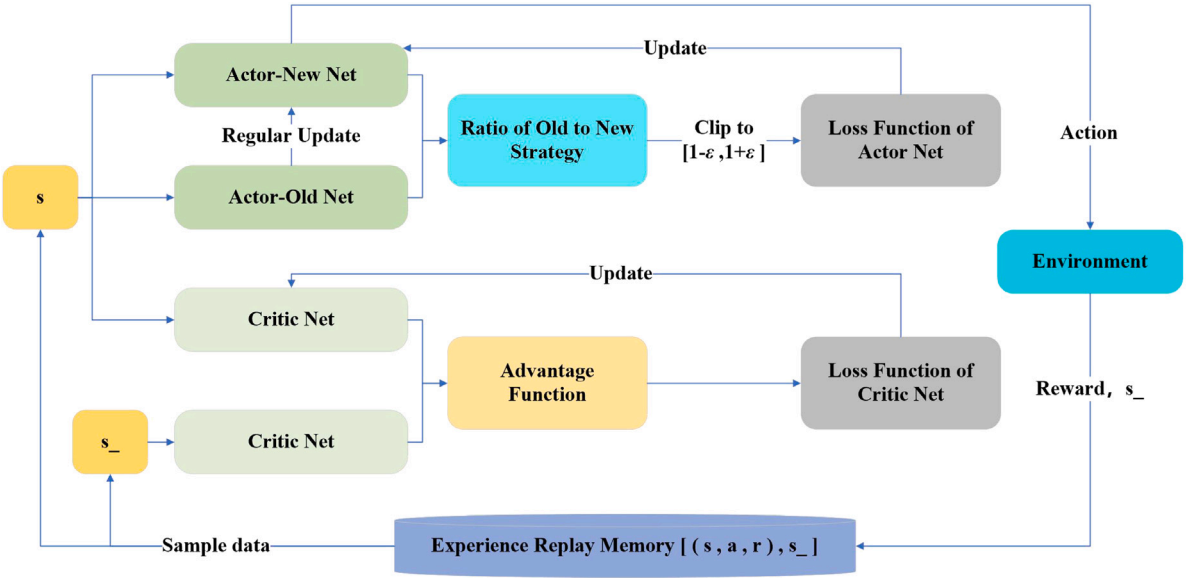


Fig. 4. PPO algorithm structure.

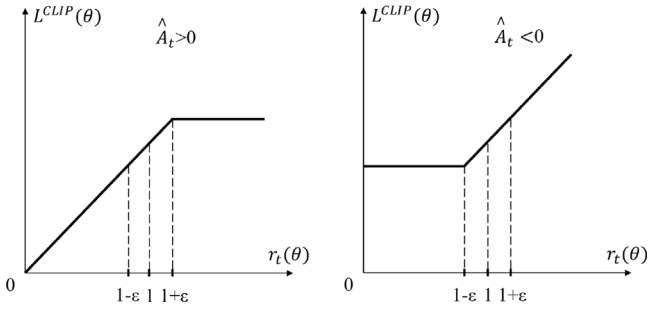


Fig. 5. Restricted range of the objective function.

3.3.1. Improved action space and state space

In this paper, we design a reasonable state, action space and reward function according to the actual operation state of the ship, use the sensors of the MASS to detect and collect the environment information, obtain the state information of the MASS at the current moment through the positioning and sensing system, input this data information as the state feature vector into the algorithm network, select the optimal action at the current moment according to the fusion algorithm, obtain the reward, and optimize the strategy by accumulating the reward.

Based on the sampling method of velocity by dynamic window in the DWA algorithm, the action space of the fusion algorithm is designed, including the angular velocity ω_r and linear velocity v_r describing the direction and velocity of ship motion. The range of angular velocity ω_r and linear velocity v_r is normalized considering the ship kinematics and practical requirements where angular velocity $\omega_r \in [-1, 1]$ and linear velocity $v_r \in [0, 1]$. The action space a is defined as:

$$a = (\omega_r, v_r) \tag{17}$$

The state space represents the environmental information that the agent can perceive and is the basis for the agent to make decisions and evaluate its long-term benefits. A reasonable design of the state space can ensure the convergence of the DRL algorithm and improve the performance. In this study the state space is mainly composed of two parts: the environmental information S_{env} and the motion state S_{os} of this ship.

Environmental information S_{env} mainly includes information such as obstacles and the status of incoming ships, among which the distance from each ship to surrounding ships and each obstacle is the most intuitive and important indicator reflecting the current environmental information. In practice, the environmental information during ship navigation is obtained by radar and sensor detection, so the ship safety domain model is designed as shown in Fig. 7. The figure presents a model diagram depicting the ship safety domain, represented by a circular shape centered at the ship's center point and with a radius of d_{sensor}. This model primarily illustrates the maximum detection range achievable by the ship. The rays emanating from the ship symbolize the detection signals transmitted by the ship's sensor system, covering a complete 360° range around the ship. It is important to highlight that the figure displays only a subset of the schematic rays and not the complete set. By constructing the ship safety field, the position information of multiple obstacles and incoming ships in the field of this ship can be accurately obtained.

Since the algorithm needs to determine the nodes of the input and output layers of the neural network before starting the training, the number of nodes in the input layer cannot be changed during the training process. However, the number of obstacles and incoming ships within each domain of ship safety is dynamically changing. Since there are many obstacles in the offshore complex environment, if the number of nodes in the initial input layer is increased to reflect the information of all obstacles and incoming ships, the whole network will become too large and complex, which obviously increases the time required for training the algorithm and reduces the convergence effect. For this reason, in this study, only the nearest and most dangerous obstacles or incoming ships are considered (Wu and Yu, 2022). Therefore, the environmental information S_{env} is defined as shown in Eq. (18).

$$S_{env} = [x_T, y_T, v_T, \theta_T, d_T, \Delta\theta_T] \tag{18}$$

Where x_T, y_T, v_T and θ_T are the position, speed and direction of movement of the nearest obstacle in the two-dimensional plane, d_T is the distance from the ship to the obstacle, and $\Delta\theta_T$ is the angle difference between the direction of movement of the obstacle and the bow direction of the ship.

The ship's motion information S_{os} mainly includes the ship's position, speed, bow direction and the distance and orientation between the ship and the target point. The definition of the ship's motion state information S_{os} is shown in Eq. (19).

$$S_{os} = [x_o, y_o, v_o, \omega_o, \theta_o, d_{goal}, \theta_{goal}] \tag{19}$$

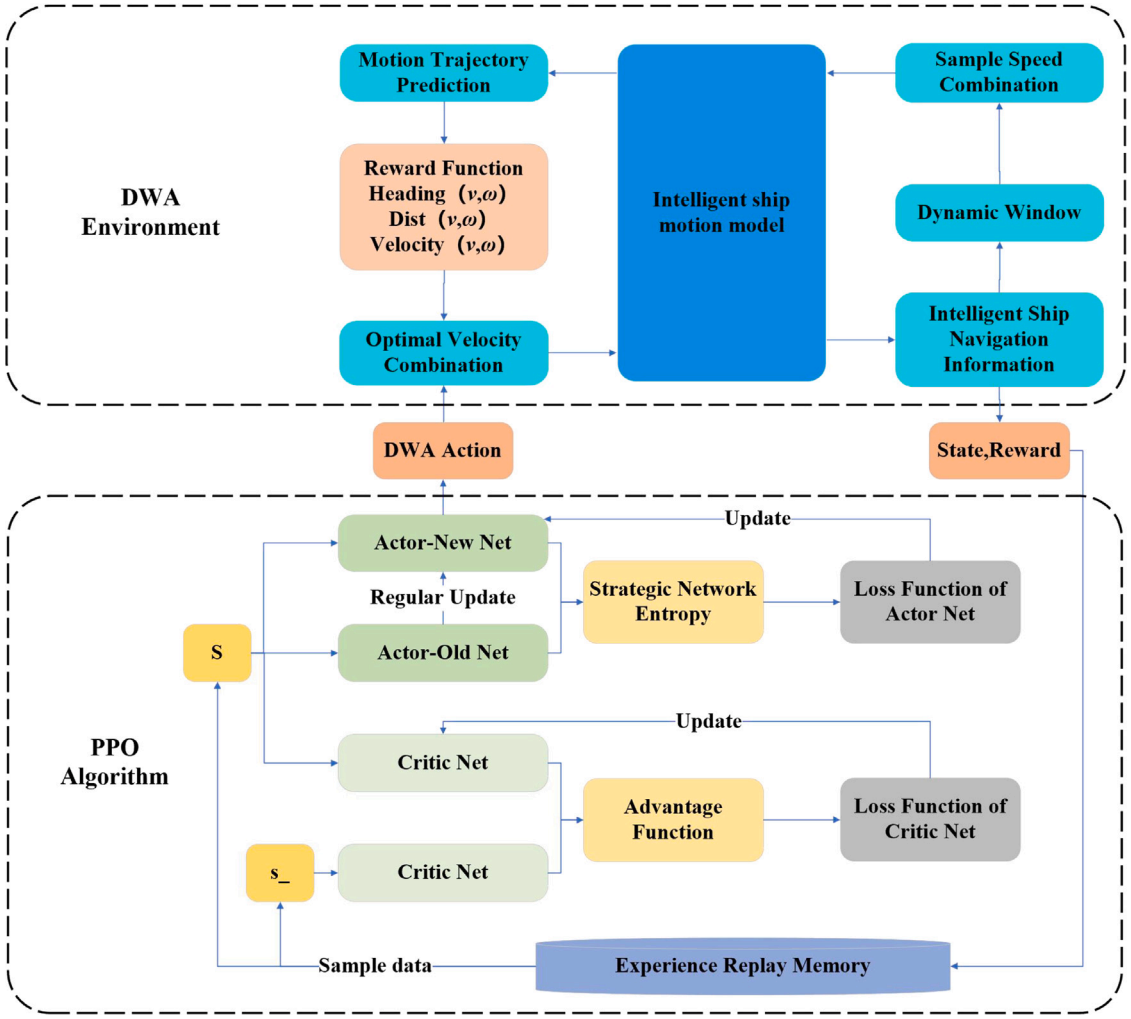


Fig. 6. Improved algorithm structure combining PPO and DWA.

Table 1
Hyperparameters of algorithm training.

Hyperparameters	Symbol	Value
Reward Discount Rate	γ	0.9
Actor Network Learning Rate	lr_{Actor}	3×10^{-4}
Critic Network Learning Rate	lr_{Critic}	3×10^{-4}
GAE parameter	λ	0.95
Strategic Network Entropy	ϵ	5
Target Actor Network Update Frequency	K	10
Maximum number of training sessions	M	1500
Experience replay storage pool size	b	2048
Batch size for experience replay learning	b_{min}	64

where x_o, y_o, v_o, w_o and θ_o denote the position, linear velocity, angular velocity, and direction of motion of the ship in the two-dimensional plane, respectively. d_{goal} is the distance between the ship and the focus, and θ_{goal} is the orientation of the end point to the ship.

3.3.2. Design of reward function

Since the traditional PPO algorithm is used for obstacle avoidance path planning, the reward function only contains a positive reward for reaching the end point, a negative reward for hitting an obstacle, and a negative reward for taking a step, and the intelligent agent does not receive any positive feedback for other actions before reaching the end point. In complex environments, the agent may fail to learn because of the sparse reward function. Therefore, in this paper, we focus on the control objective of dynamic obstacle avoidance of intelligent ships, combine the evaluation function in the DWA algorithm, design a suitable reward function to effectively evaluate the current state of the intelligent ship, guide the intelligent body to make the correct collision avoidance decision, ensure the convergence of the algorithm and improve its performance.

The rewards obtained by the agent at each time step are divided into three parts: normal action reward R_o, end-point reward R_{goal}, and collision reward R_{col}.

$$R = \begin{cases} R_o, & normal \quad action \\ R_{goal}, & reach \quad end-point \\ R_{col}, & collision \end{cases} \tag{20}$$

The normal action reward R_o is defined as the reward obtained by moving one time step when the intelligent ship does not collide and does not reach the end point. To solve the problem of sparse rewards in the improved algorithm, the reward function is improved according to the evaluation function in the DWA algorithm. The improved normal action reward function is defined as shown in Eq. (21).

$$R_o = \begin{cases} R_{head}, & Heading(v,\omega) \\ R_{dis}, & Distance(v,\omega) \\ R_{vel}, & Velocity(v,\omega) \end{cases} \tag{21}$$

R_{head} corresponds to the angular component $Heading(v,\omega)$ in the evaluation function of DWA, and this reward value is inversely proportional to the angular difference between the motion direction and the

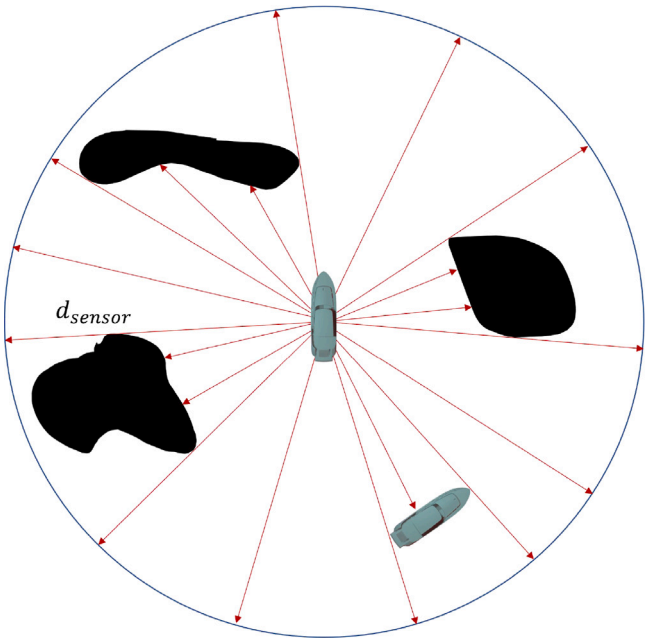


Fig. 7. Ship Safety Domain.

end direction of the intelligent ships, the larger the angular difference is, the lower the reward value. R_{head} is defined as shown in Eq. (22).

$$R_{head} = -\frac{\|\theta_o - \theta_{goal}\|}{\pi} \tag{22}$$

R_{dis} corresponds to the distance component $Distance(v, \omega)$ in the evaluation function of DWA, and this reward value is multiplied proportionally to the distance between the intelligent ship and the nearest obstacle, the further the distance from the obstacle, the higher the reward value. R_{dis} is defined as shown in Eq. (23).

$$R_{dis} = -\frac{d_{sensor} - d_{obs}}{d_{sensor}} \tag{23}$$

R_{vel} corresponds to $Velocity(v, \omega)$, the velocity component of the evaluation function of DWA, and this reward value is proportional to the speed of the intelligent ship, the faster the speed is, the higher the reward value. R_{vel} is defined as shown in Eq. (24).

$$R_{vel} = -\frac{v_{max} - v}{v_{max}} \tag{24}$$

The endpoint reward R_{goal} is defined as the reward given based on the distance between the current position of the intelligent ship and the endpoint. When the distance between the intelligent ship and the end point is less than $d_{sensor}/2$, the end point reward value is 100; otherwise the end point reward value is 0. R_{goal} is defined as shown in Eq. (25).

$$R_{goal} = \begin{cases} 100, d \le \frac{d_{sensor}}{2} \\ 0, d > \frac{d_{sensor}}{2} \end{cases} \tag{25}$$

The collision reward R_{col} is defined as the reward given according to the distance between the current position of the intelligent ship and the nearest obstacle or the oncoming ship. When the distance between the intelligent ship and the nearest obstacle or the oncoming ship is less than $d_{sensor}/2$, the end-point reward value is −100; otherwise the end-point reward value is 0. R_{col} is defined as shown in Eq. (26).

$$R_{col} = \begin{cases} -100, d \le \frac{d_{sensor}}{2} \\ 0, d > \frac{d_{sensor}}{2} \end{cases} \tag{26}$$

4. Simulation experiments

To verify the feasibility and practicality of the deep reinforcement learning and DWA-based obstacle avoidance path planning algorithm proposed in this paper, a simulation experimental environment is built using Python, and training experiments are conducted in both simple and complex environments. This chapter describes the construction of the DRL training environment and the analysis of the experimental results. The rest of this chapter is organized as follows. Section 4.1 describes the construction of the DRL training environment. Section 4.2 presents the experimental results and analysis.

4.1. Simulation environment and DRL model design

The algorithm model is constructed using the PyTorch framework and Python, and a virtual simulation environment that displays the training process in real time is created using Pygame. The visualization interface is a 1000 × 1000 pixel 2D map, and the ship's motion range is set to the size of the 2D map. If the ship crosses the boundary of the 2D map, the ship is considered to be in collision.

The state space and action space were introduced in the previous Section 3.3.1. After several experiments, the model of the DWA-DRL algorithm was designed to consist of three neural networks, the current actor network, the target actor network, and the critic network, all with three hidden layers. The hyperparameters of algorithm training are set as shown in Table 1. Among them, the reward decay rate γ is 0.9, and its higher value represents the higher attention of the intelligence to the future behavior; the neural structures of the current actor network and the target actor network are the same. To obtain a good learning effect and avoid the situation in which the intelligences fall into the local optimum, the network learning rate lr_{Actor} is set to 3×10^{-4}; the maximum size of the experience replay storage pool is set to 2048, and the batch size of experience replay learning is set to 64; the number of neurons in the hidden layer is 64, 32 and 10, respectively, and the input information of the input layer mainly includes the environmental information and the information of the motion state of this ship. Its data type is a two-dimensional array of shape (13, 1), which is input to the neural network after flattening the data into one-dimensional data; the output of the neural network is the linear and angular velocity of the MASS, and the tanh activation function is selected according to the range of action, and its output range is [−1,1], and then it is linearly transformed to the range of normal ship speed and rotational speed. The update mode of the actor network is soft update, that is, the current structure of the current actor network and the target actor network is shown in Fig. 8. The number of hidden layer neurons of the critic network is 64, 32 and 10, and the learning rate lr_{Critic} is set to 3×10^{-4}. The structure of the critic network is shown in Fig. 9. Both the actor network and the critic network use the Adam network optimizer. Based on the above training conditions and parameters, this paper conducts training experiments on the collision avoidance path planning model for the MASS.

The model continuously interacts with the environment to learn action strategies, and the learning effect is measured by the cumulative reward value of each training set. Fig. 10 depicts the total rewards obtained by the improved algorithm and the original PPO algorithm in each training set. Upon observing the image, it can be noted that the improved algorithm achieves a convergence value of approximately 80 within around 100 rounds and steadily approaches the peak value of around 90 during the subsequent training process. On the other hand, the PPO algorithm exhibits fluctuations between 100 rounds and 300 rounds, remaining in a semi-convergent state with reward values fluctuating considerably around 50. It eventually converges to approximately 80 by round 300, albeit with some remaining fluctuations. To summarize, the improved algorithm surpasses the PPO algorithm in terms of both convergence speed and stability.

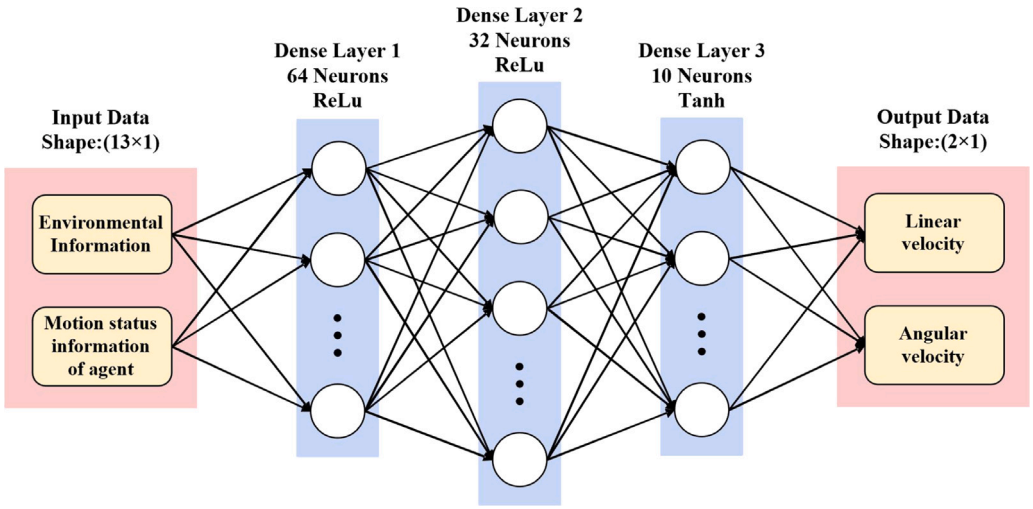


Fig. 8. Structure of the current Actor network and the target Actor network.

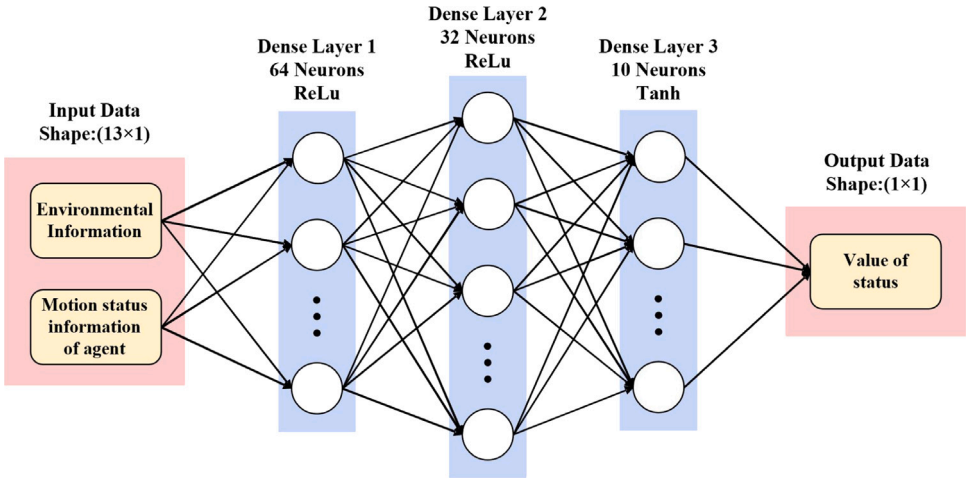


Fig. 9. Structure of the critic network.

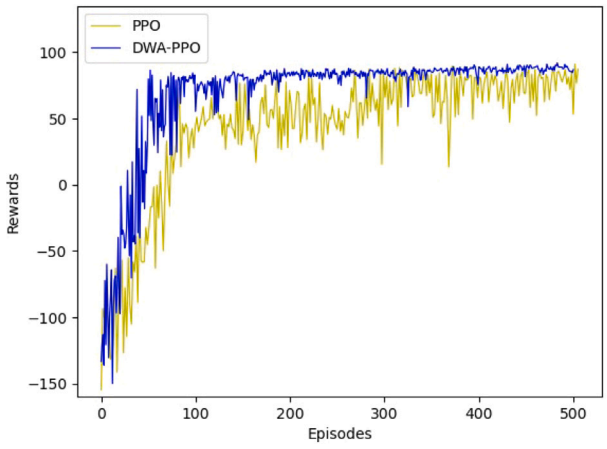


Fig. 10. Total reward for each episode.

4.2. Experimental results

The experimental design approach in this paper follows a two-step process. First, the proposed algorithm is trained to enable the MASS to select a safe path from the starting point to the end point. Subsequently, a series of environments is constructed, comprising both simple and complex scenarios. The simple environments include three typical encounter postures: head-on, overtaking, and crossing. On the other hand, the complex environments consist of multiple dynamic and static obstacles. The agent is trained in these environments to efficiently reach the end point while ensuring safe avoidance of dynamic obstacles.

To compare the advantages of the improved algorithm against other algorithms, this paper conducts a comparative analysis from two key perspectives: the cumulative change in the ship's heading and the closest distance between the ship and the target ship or obstacle during the collision avoidance process. These metrics serve as quantitative measures to assess the performance and effectiveness of different algorithms in obstacle avoidance scenarios. In the process of automatic collision avoidance of the MASS, the accumulated change in ship heading, as a data indicator to measure the simplicity and complexity of the collision avoidance process, can clearly reflect the complexity and practicality of the obstacle avoidance path calculated by the obstacle avoidance algorithm. Therefore, the less the accumulated change in the ship's heading is, the simpler the automatic obstacle avoidance process of the MASS is, and the higher the practicality of the corresponding algorithm is. Another important index to measure the safety performance in the collision avoidance process is the distance between this ship and the target ship. Although the further the distance from the target ship is, the safer it is, the corresponding path length increases with the increase of

Table 2
Initial information table of the three encounter situations.

Encounter situations	Ship information	Initial position	Initial orientation	Velocity
Head-on	Own ship	(200,200)	45°	5 m/s
	Target ship	(800,800)	225°	2 m/s
Crossing	Own ship	(200,200)	45°	5 m/s
	Target ship	(750,250)	135°	4 m/s
Overtaking	Own ship	(200,200)	45°	5 m/s
	Target ship	(400,400)	45°	2 m/s

that distance. Considering the energy cost issue, this paper introduces the minimum collision avoidance distance d_{min} and the safe collision avoidance distance d_{safe} allowed by the MASS as the reference of this index. When the distance between two ships is less than d_{min}, the safety of the collision avoidance process is poor; when the distance between two ships is greater than d_{min} and less than d_{safe}, the safety of the collision avoidance process is proportional to the distance between two ships; and when the distance between two ships is greater than d_{safe}, the collision avoidance process is very safe. According to the calculation method of the minimum full rudder avoidance distance in the literature (Hua, 2019), combined with the MASS length generally in 5~10 m, the minimum full rudder avoidance distance of the MASS can be calculated as approximately 80 ~ 100 m. Since the ship loading state, draft depth ratio, wind and waves influence ship rotation etc. are not considered, so in order to be more suitable for practical application, a sufficiently large margin needs to be left, and this paper sets the MASS. The minimum collision avoidance distance allowed in this paper is d_{min} =80 m, and the safe collision avoidance distance is d_{safe} = 110 m.

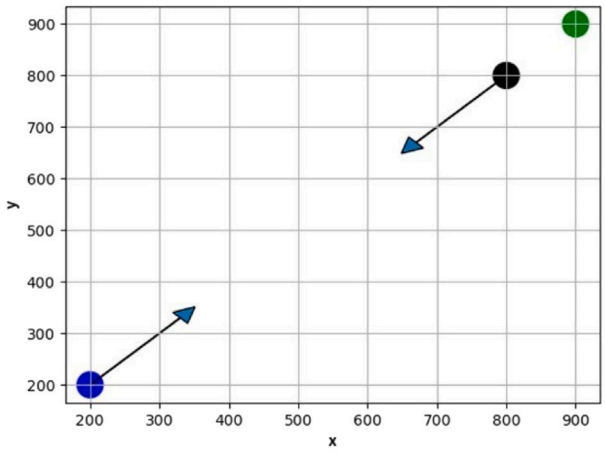
4.2.1. Basic meeting situation experiment

To test the collision avoidance effect of the DWA-PPO algorithm proposed in this paper under the three basic encounter situations of head-on, overtaking and crossing, the simulation environments of the three situations are constructed, and the relevant parameters are shown in Table 2.

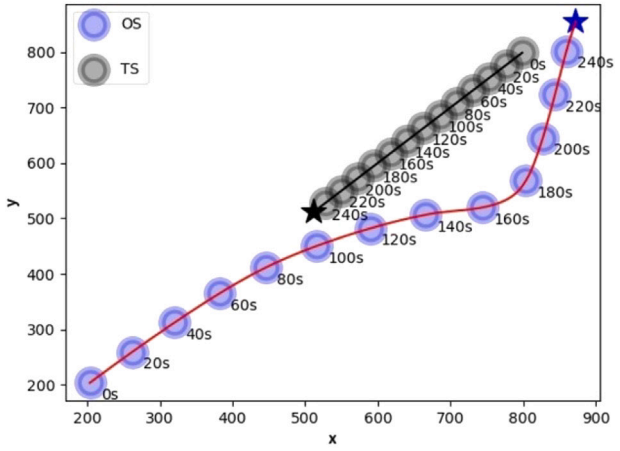
1.Head-on

In the head-on collision avoidance simulation experiment, the obstacle ship is set to come from the front of the ship. The initial position of the ship is (200,200), the initial motion direction is 45°, and the speed is 5 m/s; the initial position of the obstacle ship is (800,800), the initial motion direction is 225°, and the speed is 2 m/s. The scene setting is shown in Fig. 11(a). The effect of collision avoidance of the proposed algorithm is shown in Fig. 11(b), and it can be seen that the obstacle ship can be effectively avoided.

The comparative analysis of the results of the improved algorithm and other algorithms is presented in Figs. 12(a)–12(c). Fig. 12(a) illustrates the trajectory comparison of various algorithms, showing that each algorithm successfully avoids obstacles and reaches the end point. However, there are differences in the effectiveness of obstacle avoidance, which requires further analysis. Fig. 12(b) compares the cumulative change in heading for each algorithm. It is evident that the improved algorithm outperforms the other algorithms in terms of the amount of cumulative heading change. Notably, compared to the APF algorithm, the improved algorithm reduces the cumulative heading change by 3.75 rad. Similarly, compared to the DDPG algorithm, which already has good results, the improved algorithm reduces the cumulative heading change by 0.47 rad. The significant reduction in cumulative heading change achieved by the improved algorithm implies that fewer rudder operations are required during collision avoidance, making it more practical in real-world applications. In Fig. 12(c), the comparison of the distance between the ship and the obstacle ship in the avoidance trajectory of each algorithm is shown. The improved algorithm performs significantly better in terms of the closest distance between the ship and the target ship. The closest distance achieved by the improved algorithm is 133.56 m, which exceeds the minimum



(a) Head-on situation



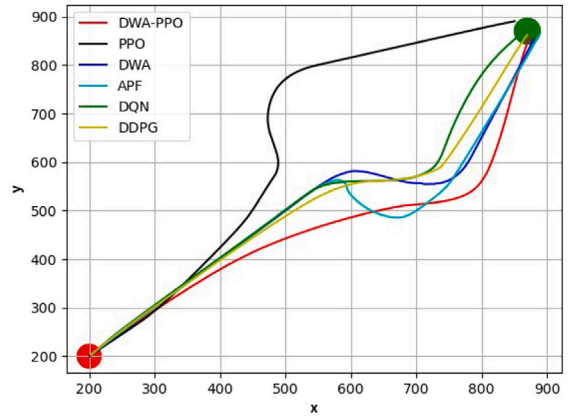
(b) Collision avoidance process

Fig. 11. Experimental and analytical diagrams of collision avoidance for the head-on situation.

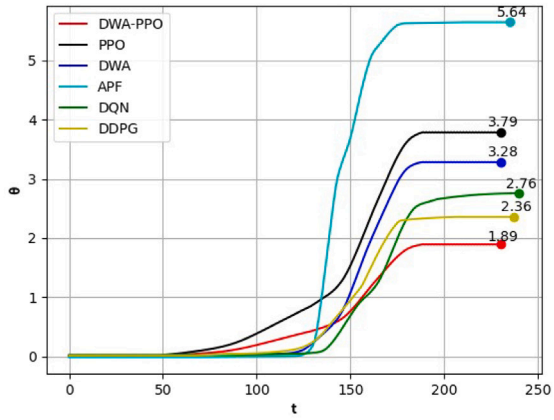
safe collision avoidance distance required by the MASS. Hence, the collision avoidance process planned by the improved algorithm is considered very safe. On the other hand, the other algorithms, including DWA, APF, DQN, DDPG, and PPO, exhibit much smaller distances between the ship and the obstacle ship in their collision avoidance trajectories, indicating less safety in the collision avoidance process. It is worth noting that the PPO algorithm achieves a nearest distance of 102.83 m, which is larger than the minimum collision avoidance distance allowed by the MASS but smaller than the safe collision avoidance distance. Thus, there is still a certain level of collision risk in the collision avoidance process of the PPO algorithm. Based on the above comparisons, the improved algorithm demonstrates superior obstacle avoidance ability and safety compared to other algorithms (see Table 3).

2.Overtaking

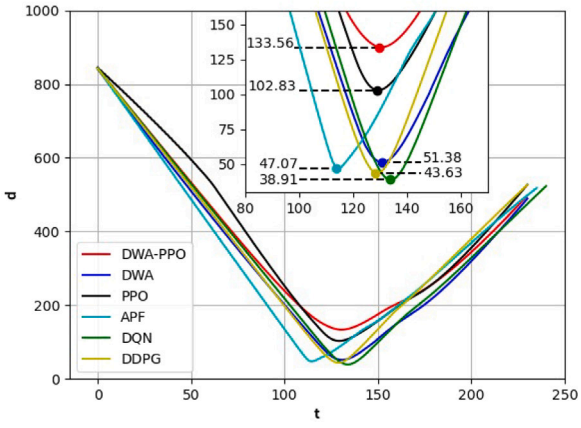
In the overtaking collision avoidance simulation experiment, the obstacle ship of this ship is set to come from directly behind and overtake the obstacle ship. The initial position of the ship is (200,200), the initial motion direction is 45°, and the speed is 5 m/s; the initial position of the obstacle ship is (400,400), the initial motion direction is 45°, and the speed is 2 m/s. The scene setting is shown in Fig. 13(a). The effect of collision avoidance of the proposed algorithm is shown in



(a) Simulation experimental results of the three methods



(b) The cumulative change of course



(c) The change in distance between the OS and TS

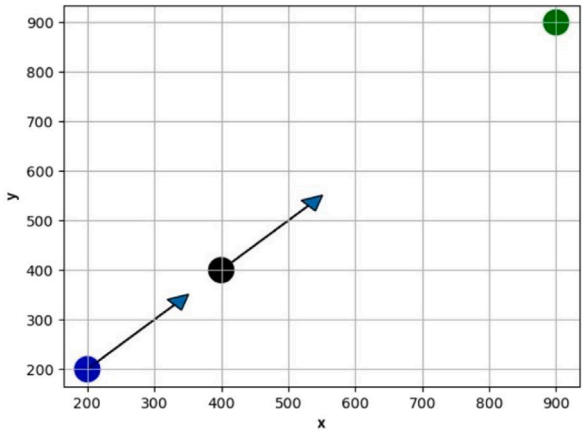
Fig. 12. Experimental and analytical diagrams of collision avoidance for the head-on situation.

Fig. 13(b), and it can be seen that the obstacle ship can be effectively avoided.

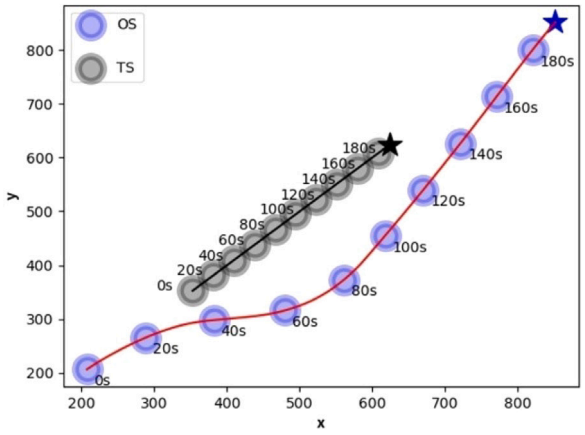
The comparative analysis of the results of the improved algorithm proposed in this paper and other algorithms is shown in Figs. 14(a)–14(c). Fig. 14(a) shows the comparison graph of trajectories of various algorithms, and it can be seen that each algorithm can successfully

Table 3
Analysis of the results of collision avoidance for the head-on situation.

	Cumulative change of course	Nearest distance between OS and TS
APF algorithm	5.64 rad	47.07 m
DWA algorithm	3.28 rad	51.38 m
DQN algorithm	2.76 rad	38.91 m
DDPG algorithm	2.36 rad	43.63 m
PPO algorithm	3.79 rad	102.83 m
DWA-PPO algorithm	1.89 rad	133.56 m



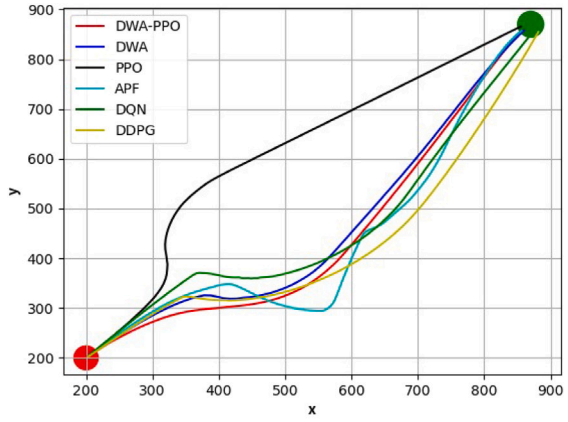
(a) Overtaking situation



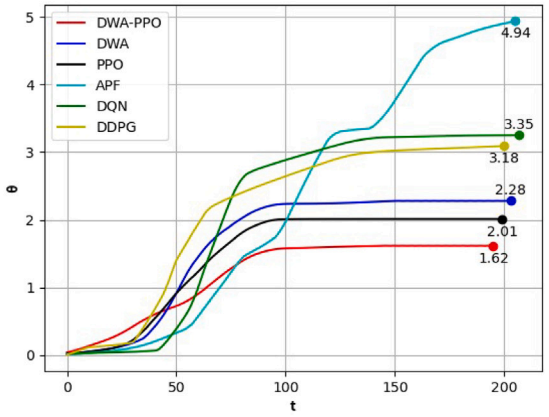
(b) Collision avoidance process

Fig. 13. Experimental and analytical diagrams of collision avoidance for the overtaking situation.

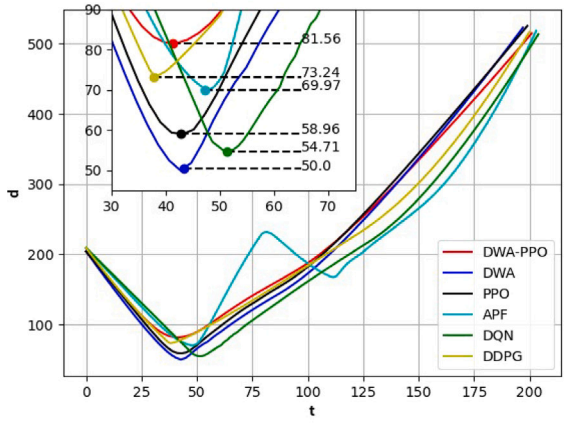
avoid obstacles and reach the end point, but they differ in terms of obstacle avoidance effect, which needs to be analyzed specifically. Fig. 14(b) shows the comparison of the cumulative change in heading for each algorithm, and it can be seen that the improved algorithm performs significantly better in terms of the amount of cumulative heading change. The comparison with the APF algorithm is particularly significant, as the improved algorithm reduces the cumulative heading change by 3.32 rad; compared with the PPO algorithm, which has better results, the improved algorithm reduces the cumulative heading change by 0.39 rad. The improved algorithm significantly reduces the cumulative change in heading during collision avoidance, so the corresponding rudder turning operation can be reduced, which is more beneficial to the application of this algorithm in practice. Fig. 14(c)



(a) Simulation experimental results of the three methods



(b) The cumulative change of course



(c) The change in distance between the OS and TS

Fig. 14. Experimental and analytical diagrams of collision avoidance for the overtaking situation.

shows the comparison of the distance between this ship and the obstacle ship in the avoidance trajectory of each algorithm. It can be seen that the improved algorithm performs significantly better in terms of the closest distance between this ship and the target ship. The nearest distance between this ship and the target ship in the collision avoidance trajectory of the improved algorithm is 81.56 m, which is slightly larger than the minimum collision avoidance distance allowed by the smart ship, so the collision avoidance process planned by the improved algorithm is relatively safe. In contrast, the nearest distance between this ship and the obstacle ship in the collision avoidance trajectory of other algorithms is much smaller than the minimum collision avoidance distance allowed by the MASS, and the collision avoidance process is less safe. After the above comparison, the improved algorithm has better collision avoidance ability and safety (see Table 4).

Table 4
Analysis of the results of collision avoidance for the overtaking situation.

	Cumulative change of course	Nearest distance between OS and TS
APF algorithm	4.94 rad	69.97 m
DWA algorithm	2.28 rad	50.00 m
DQN algorithm	3.35 rad	54.71 m
DDPG algorithm	3.18 rad	73.24 m
PPO algorithm	2.01 rad	58.96 m
DWA-PPO algorithm	1.62 rad	81.56 m

Table 5
Analysis of the results of collision avoidance for the crossing situation.

	Cumulative change of course	Nearest distance between OS and TS
APF algorithm	5.32 rad	99.12 m
DWA algorithm	4.08 rad	85.36 m
DQN algorithm	3.81 rad	108.27 m
DDPG algorithm	3.96 rad	112.15 m
PPO algorithm	3.38 rad	104.39 m
DWA-PPO algorithm	3.24 rad	118.18 m

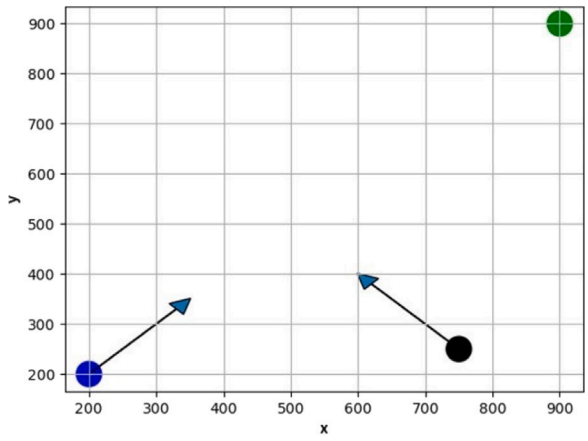
Table 6
Initial information on the environment of multi-ship encounter situation.

	Initial position	Initial orientation	Velocity
Own ship	(50, 50)	45°	5 m/s
Target ship1	(80, 290)	315°	1.2 m/s
Target ship2	(450, 190)	135°	1.5 m/s
Target ship3	(400, 400)	45°	1.5 m/s
Target ship4	(830, 930)	202°	1.5 m/s
Target ship5	(1050, 1050)	234°	1.5 m/s
Target ship6	(400, 650)	135°	1.5 m/s
Target ship7	(750, 300)	315°	1.5 m/s

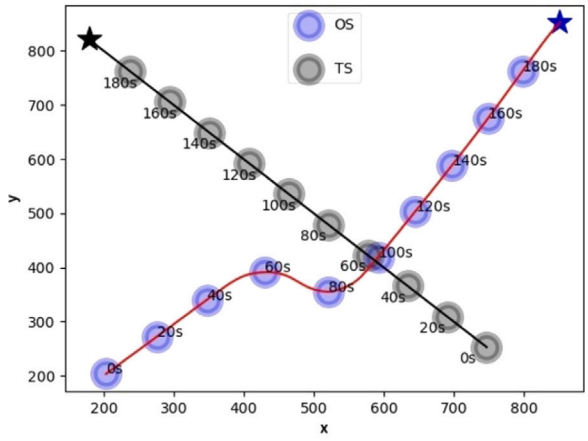
3.Crossing

In the crossing collision avoidance simulation experiment, the obstacle ship of this ship is set to come from directly behind and overtake the obstacle ship. The initial position of the ship is (200,200), the initial motion direction is 45°, and the speed is 5 m/s; the initial position of the obstacle ship is (750,250), the initial motion direction is 135°, and the speed is 4 m/s. The scene setting is shown in Fig. 15(a). The effect of collision avoidance of the proposed algorithm is shown in Fig. 15(b), and it can be seen that the obstacle ship can be effectively avoided.

The comparative analysis of the results of the improved algorithm proposed in this paper and other algorithms is shown in Figs. 16(a)–16(c). Fig. 16(a) displays the comparison graph of trajectories for various algorithms, demonstrating that each algorithm can successfully avoid obstacles and reach the end point. However, they differ in terms of obstacle avoidance effectiveness, which requires further analysis. Fig. 16(b) compares the cumulative change in heading for each algorithm, revealing that the improved algorithm performs significantly better in terms of the amount of cumulative heading change. Notably, compared with the APF algorithm, the improved algorithm reduces the cumulative heading change by 2.08 rad, and compared with the PPO algorithm, it reduces the cumulative heading change by 0.14 rad. This reduction in cumulative heading change indicates that the improved algorithm minimizes the need for rudder turning operations, enhancing its practical application. Fig. 16(c) presents the comparison of the distance between this ship and the obstacle ship in the avoidance



(a) Crossing situation



(b) Collision avoidance process

Fig. 15. Experimental and analytical diagrams of collision avoidance for the crossing situation.

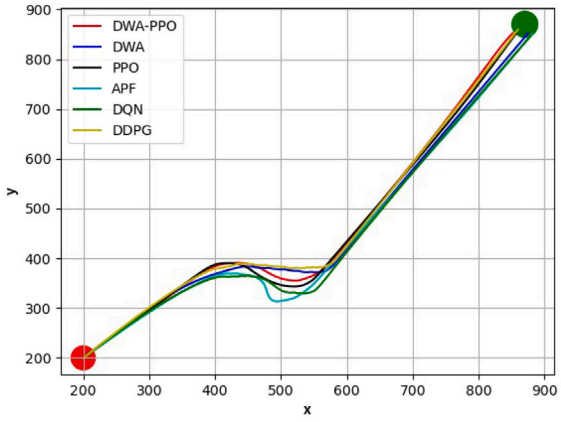
trajectory of each algorithm. The improved algorithm demonstrates significantly better performance in terms of the closest distance between this ship and the target ship. The nearest distance between the improved algorithm and the obstacle ship is 118.18 m, surpassing the safe collision avoidance distance of the smart ship, ensuring a very safe collision avoidance process. Compared to the DDPG algorithm, the improved algorithm improves the nearest distance to the obstacle ship by 6.03 m. In contrast, the other algorithms (APF, DWA, DQN, and PPO) exhibit a nearest distance smaller than the safe collision avoidance distance, indicating potential collision risks during their avoidance processes. Based on the above comparisons, the improved algorithm demonstrates superior obstacle avoidance ability and safety (see Table 5).

4.2.2. Multi-obstacle complex situation experiment

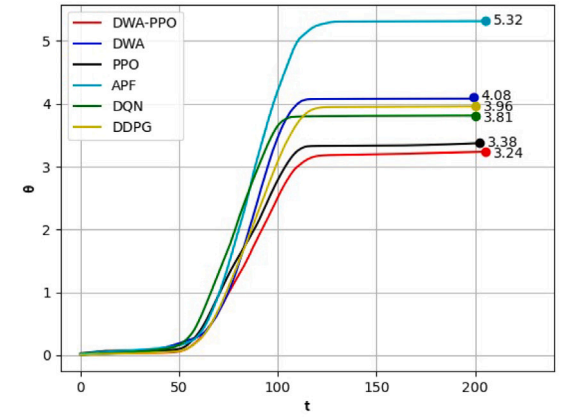
1.Multi-ship encounter situation

In order to test the collision avoidance effect of the DWA-PPO algorithm proposed in this paper in a complex dynamic environment, a complex simulation environment containing multiple dynamic obstacles is constructed, and the relevant parameters are shown in Table 6.

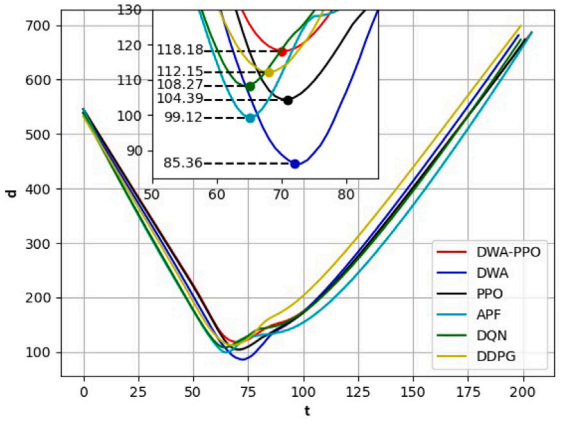
In the multiship encounter situations collision avoidance experiment, the scene is set as shown in Fig. 17(a). The initial position of this ship is (50,50), the target point position is (950,910), the initial



(a) Simulation experimental results of the three methods



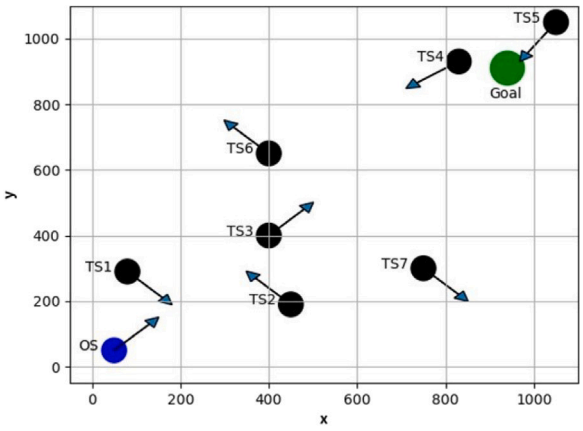
(b) The cumulative change of course



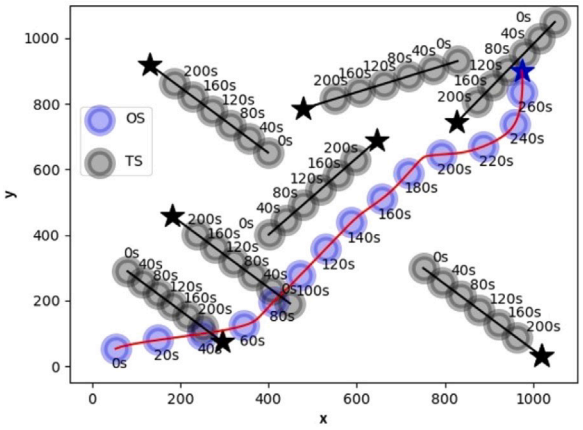
(c) The change in distance between the OS and TS

Fig. 16. Experimental and analytical diagrams of collision avoidance for the crossing situation.

motion direction is 45°, and the speed is 5 m/s; the motion parameters of all obstacle ships are shown in Table 6. The collision avoidance effect of the algorithm proposed in this paper is shown in Fig. 17(b), and it can be seen that it can effectively avoid the obstacle ship and reach the end point safely.

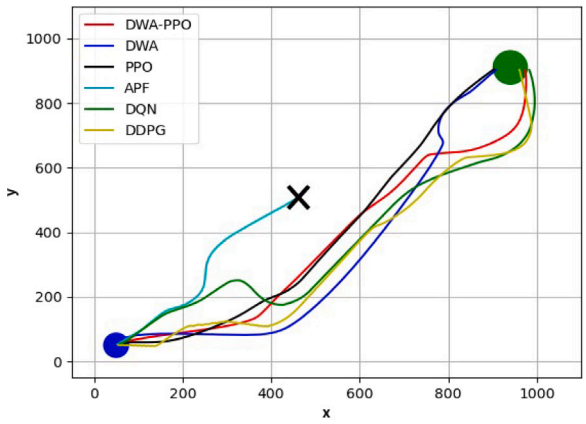


(a) Multiship situation

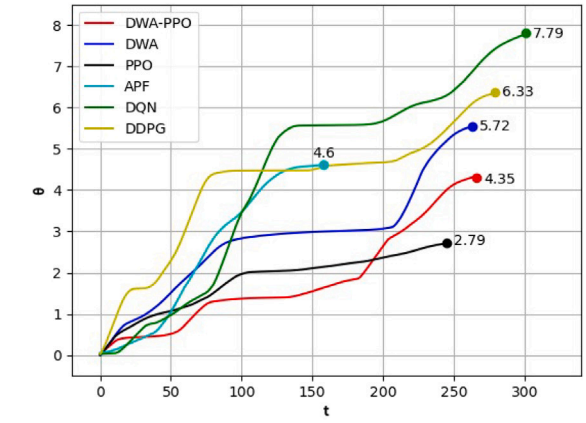


(b) Collision avoidance process

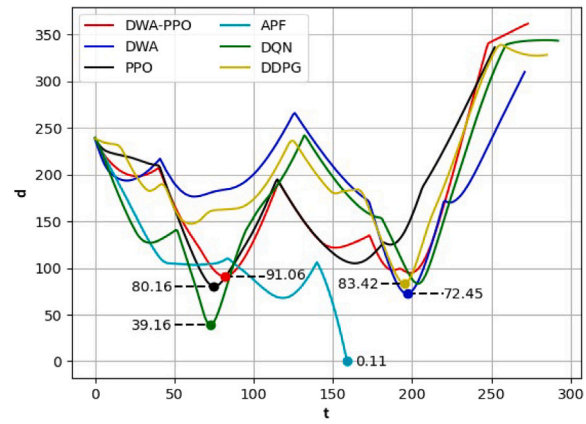
Fig. 17. Experimental and analytical diagrams of collision avoidance for the multiship situation.



(a) Simulation experimental results of the three methods



(b) The cumulative change of course



(c) The change in distance between the OS and TS

Fig. 18. Experimental and analytical diagrams of collision avoidance for the multiship situation.

The comparative analysis of the results of the improved algorithm proposed in this paper and other algorithms is shown in Figs. 18(a)–18(c). Fig. 18(a) displays the comparison graph of trajectories for various algorithms, indicating that all algorithms, except for the collision between APF and the target ship, can successfully avoid obstacles and reach the end point. However, they differ in terms of obstacle avoidance effectiveness, which requires further analysis. Fig. 18(b) compares the cumulative change in heading for each algorithm. It can be observed that the improved algorithm performs significantly better in terms of the cumulative change in heading when compared to the other algorithms, except for a slightly better performance of the improved algorithm compared to the PPO algorithm, where it reduces the cumulative change in heading by 1.33 rad. This reduction in cumulative change in heading indicates that the improved algorithm minimizes the need for significant changes in ship heading, leading to more efficient and controlled navigation. Fig. 18(c) presents the comparison of the distance between this ship and the obstacle ship in the obstacle avoidance trajectory of each algorithm. The nearest distance between this ship and the target ship in the collision avoidance trajectory of the improved algorithm is 91.06 m, which exceeds the minimum collision avoidance distance allowed by the smart ship, ensuring a relatively safe collision avoidance process. In contrast, the nearest distance between this ship and the obstacle ship in the collision avoidance trajectory of other algorithms is smaller than that of the improved algorithm. Therefore, the improved algorithm performs significantly better in terms of the closest distance between this ship and the target ship, indicating enhanced safety in the collision avoidance process (see Table 7).

2.Multi-ship with static obstacles situation

To test the collision avoidance effect of the improved algorithm proposed in this paper in the environment of multi-ship with static

Table 7
Analysis of the results of collision avoidance for the multiship encounter situation.

	Cumulative change of course	Nearest distance between OS and TS
APF algorithm	4.60 rad	0.11 m
DWA algorithm	5.72 rad	72.45 m
DQN algorithm	7.79 rad	39.16 m
DDPG algorithm	6.33 rad	83.42 m
PPO algorithm	3.02 rad	80.16 m
DWA-PPO algorithm	4.35 rad	91.06 m

Table 8
Initial information on the environment of multi-ship with static obstacles situation.

	Initial position	Initial orientation	Velocity
Own ship	(50, 50)	45°	5 m/s
Target ship1	(900, 300)	148°	3 m/s
Target ship2	(350, 800)	270°	3 m/s

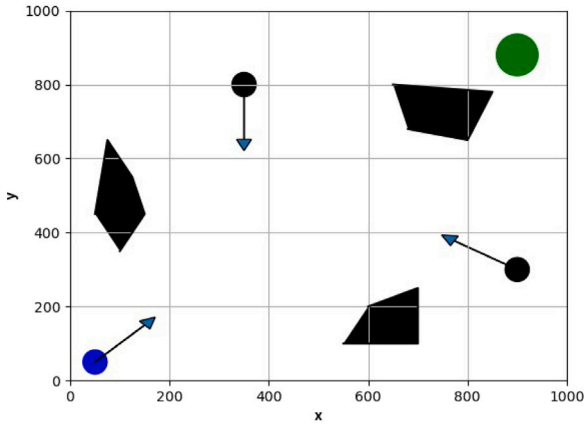
Table 9
Analysis of the results of collision avoidance for the multiship encounter situation.

	Cumulative change of course	Nearest distance between OS and TS
APF algorithm	6.16 rad	49.38 m
DWA algorithm	5.85 rad	71.28 m
DQN algorithm	7.84 rad	81.02 m
DDPG algorithm	5.52 rad	70.22 m
PPO algorithm	6.66 rad	60.84 m
DWA-PPO algorithm	3.87 rad	86.12 m

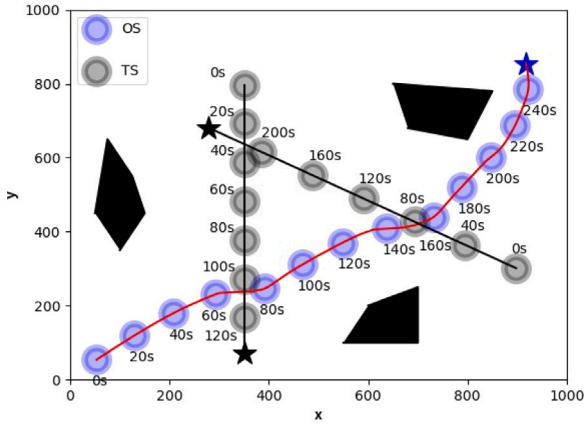
obstacles, a complex simulation environment containing multiple target ship with multiple static obstacles is constructed, and the relevant parameters are shown in Table 8.

In the multi-ship with static obstacles in the collision avoidance simulation experiment, the scene is set as shown in Fig. 19(a) The initial position of this ship is (50,50), the target point position is (900,900), the initial motion direction is 45°, the speed is 5 m/s; the motion parameters of all the obstacle ships are shown in Table 8. The collision avoidance effect of the algorithm proposed in this paper is shown in Fig. 19(b), and it can be seen that it can effectively avoid all target boats with static obstacles and reach the end point safely.

The results of the comparative analysis between the proposed improved algorithm in this paper and other algorithms are depicted in Figs. 20(a)–20(c). Fig. 20(a) illustrates the trajectory comparison of the algorithms, highlighting their successful obstacle avoidance and reaching the destination. However, there are variations in terms of obstacle avoidance effectiveness, necessitating a specific analysis. Fig. 20(b) presents the comparison of the cumulative change in heading for each algorithm, demonstrating that the improved algorithm outperforms the others significantly in terms of cumulative heading change. Notably, compared to the DQN algorithm, the improved algorithm reduces the cumulative heading change by 3.97 rad. Similarly, when compared to the DDPG algorithm, which exhibits better results, the improved algorithm reduces the cumulative heading change by 1.65 rad. By significantly reducing the cumulative change in heading during collision avoidance, the improved algorithm minimizes the need for ruddering operations, thereby enhancing the practical application of the algorithm. In Fig. 20(c), the comparison of the distance between this ship and the obstacle ship in the avoidance trajectory for each algorithm is shown. The improved algorithm demonstrates superior performance in terms of the closest distance between this ship and the target ship. The nearest distance achieved by the improved algorithm and the obstacle ship is 86.12 m, slightly exceeding the minimum collision avoidance distance required by the smart ship, thereby ensuring a relatively safe collision avoidance process. Comparatively, the improved algorithm



(a) Multiship situation



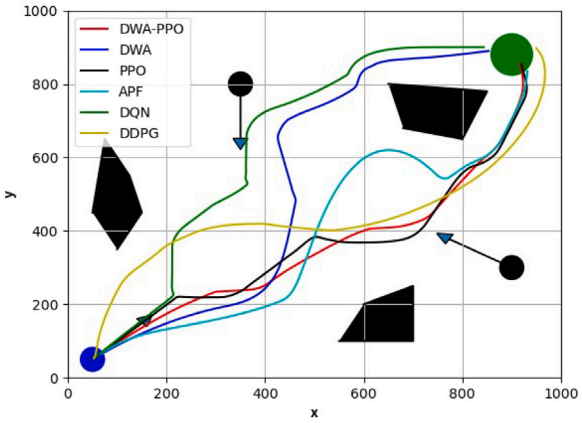
(b) Collision avoidance process

Fig. 19. Experimental and analytical diagrams of collision avoidance for the multiship situation.

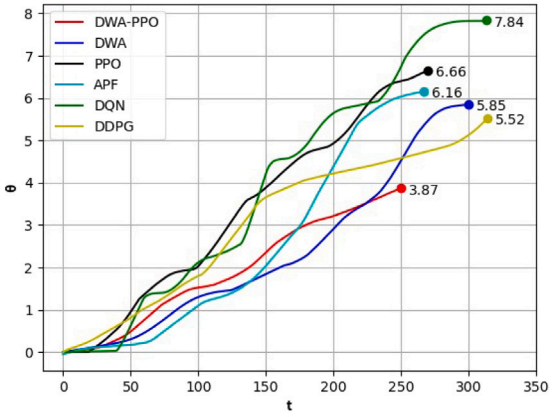
improves the nearest distance between the ship and the obstacle ship in the avoidance trajectory by 5.1 m compared to the DQN algorithm, which exhibits better results. Conversely, other algorithms have significantly smaller nearest distances, falling below the minimum avoidance distance stipulated by the MASS, resulting in poor safety during the avoidance process. Based on the aforementioned comparison, it is evident that the improved algorithm possesses superior obstacle avoidance ability and safety (see Table 9).

5. Conclusion and future work

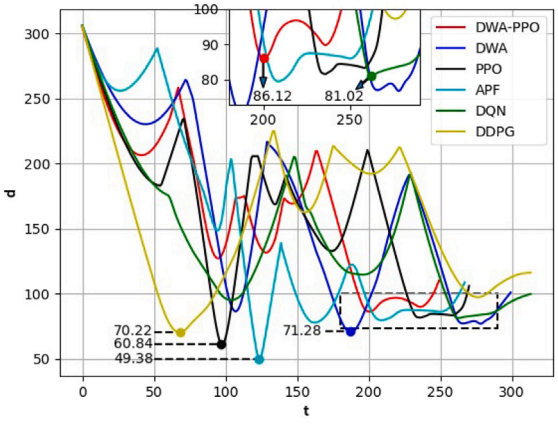
The purpose of this paper is to examine the obstacle avoidance path planning algorithm applicable to the MASS. An obstacle avoidance path planning algorithm based on deep reinforcement learning (PPO) and the dynamic window method is proposed to address the limitations of the traditional dynamic window method (DWA) algorithm in terms of trajectory foresight, heading change, safety, and random obstacle avoidance in complex environments. The reward sparsity problem is effectively solved, and the training effect of the algorithm is improved by enhancing the ship motion description in the DWA algorithm and modifying the evaluation function for the action space, state space, and reward function of the PPO algorithm. The effectiveness of the algorithm is then verified in three classical encounter situations and complex environments. The experimental results demonstrate that



(a) Simulation experimental results of the three methods



(b) The cumulative change of course



(c) The change in distance between the OS and TS

Fig. 20. Experimental and analytical diagrams of collision avoidance for the multiship situation.

optimal collision avoidance decisions can be made in complex environments using the improved algorithm, thereby enabling autonomous collision avoidance path planning for the MASS.

However, the proposed method also has some limitations. The DWA-PPO algorithm is suited for a model-free reinforcement learning (DRL) environment, and its model output represents only one action strategy, which can be influenced by the physical factors of the ship in practical applications. The DWA-PPO algorithm relies on deep neural networks trained in a simulated environment, which introduces uncertainties regarding the ship model and the real-world environment. Future work should consider the impact of uncertain environmental factors on collision avoidance decisions to enhance the reliability of the proposed method in practical applications. Additionally, the influence of sea visibility on sensor data accuracy is an important consideration. Thus, combining environmental effects and ship kinematic models to achieve precise motion control through deep reinforcement learning is part of the planned future research. Moreover, exploring the integration of model-based reinforcement learning algorithms with the proposed method aims to enhance applicability and stability.

CRediT authorship contribution statement

Chuanbo Wu: Conceptualization, Writing – original draft, Writing – review & editing, Visualization. **Wangneng Yu:** Writing – original draft, Writing – review & editing, Methodology, Supervision, Formal analysis, Project administration, Funding acquisition. **Guangze Li:** Writing – original draft, Writing – review & editing. **Weiqiang Liao:** Writing – review & editing, Project administration, Funding acquisition.

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

The data that has been used is confidential

Acknowledgments

The authors would like to express appreciation for the financial support provided by the National Natural Science Foundation of China (52171308), Key Project of Fujian Provincial Science and Technology Department, China (2021H0021) Natural Science Foundation of Fujian Province, China (2022J01333) and National Key Research and Development Program of Ministry of Science and Technology (2021YFB3901500).

References

Aslam, S., Michaelides, M.P., Herodotou, H., 2020. Internet of ships: A survey on architectures, emerging applications, and challenges. IEEE Internet Things J. 7, 9714–9727.

Chen, Y., Bai, G., Zhan, Y., Hu, X., Liu, J., 2021. Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning. IEEE Access 9, 40728–40742.

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 Eng. 189, 106299.

Chun, D.-H., Roh, M.-I., Lee, H.-W., Ha, J., Yu, D., 2021. Deep reinforcement learning-based collision avoidance for an autonomous ship. Ocean Eng. 234, 109216, 1.

Fossen, T.I., Breivik, M., Skjetne, R., 2003. Line-of-sight path following of underactuated marine craft. IFAC Proc. Vol. 36, 211–216.

Guo, X., Ji, M., Zhao, Z., Wen, D., Zhang, W., 2020a. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm. Ocean Eng. 216, 107693.

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

He, Z., Liu, C., Chu, X., Negenborn, R.R., Wu, Q., 2022. Dynamic anti-collision A-star algorithm for multi-ship encounter situations. Appl. Ocean Res. 118, 102995.

Hsu, Y.-H., Gau, R.-H., 2022. Reinforcement learning-based collision avoidance and optimal trajectory planning in UAV communication networks. IEEE Trans. Mob. Comput. 21, 306–320.

Hua, K., 2019. Dangerous situation and collision avoidance in ship navigation. Mar. Technol. 5, 80–85.

Ju, C., Luo, Q., Yan, X., 2020. Path planning using an improved A-star algorithm. In: 2020 11th International Conference on Prognostics and System Health Management. PHM-2020 Jinan, pp. 23–26.

Li, L., Wu, D., Huang, Y., Yuan, Z.-M., 2021. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. Appl. Ocean Res. 113, 102759.

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

Munim, Z.H., Dushenko, M., Jimenez, V.J., Shakil, M.H., Imset, M., 2020. Big data and artificial intelligence in the maritime industry: a bibliometric review and future research directions. Marit. Policy Manag. 47, 577–597.

Peng, Y., Huang, Z., Li, S., 2020. Research on automatic obstacle avoidance navigation of mobile robot based on dynamic window approach. Process Autom. Instrum. 41, 26–29+33.

Sang, H., You, Y., Sun, X., Zhou, Y., Liu, F., 2021. The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations. Ocean Eng. 223, 108709.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 1999. Actor-critic algorithms. Adv. Neural Inf. Process. Syst. 12.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

Wang, X., Kou, X., Huang, J., Tan, X., 2021. A collision avoidance method for intelligent ship based on the improved bacterial foraging optimization algorithm. J. Robot. 2021, 1–10.

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., Kim, N., 2020. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning. Ocean Eng. 199, 107001.

Wu, C., Yu, W., 2022. Research on dynamic ship safety domain model based on safety level. Shipbuild. China 63, 218–229.

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. Marit. Policy Manag. 47, 1–15.

Xia, G., Sun, X., Xia, X., 2021. Multiple task assignment and path planning of a multiple unmanned surface vehicles system based on improved self-organizing mapping and improved genetic algorithm. J. Mar. Sci. Eng. 9, 556.

Xie, S., Chu, X., Zheng, M., Liu, C., 2020. A composite learning method for multi-ship collision avoidance based on reinforcement learning and inverse control. Neurocomputing 411, 375–392.

Xiong, C., Zhou, H., Lu, D., Zeng, Z., Lian, L., Yu, C., 2020. Rapidly-exploring adaptive sampling tree*: A sample-based path-planning algorithm for unmanned marine vehicles information gathering in variable ocean environments. Sensors 20, 2515.

Xu, X., Lu, Y., Liu, G., Cai, P., Zhang, W., 2022. COLREGs-abiding hybrid collision avoidance algorithm based on deep reinforcement learning for USVs. Ocean Eng. 247, 110749.

Yoo, B., Kim, J., 2016. Path optimization for marine vehicles in ocean currents using reinforcement learning. J. Mar. Sci. Technol. 21, 334–343.

Yu, W., Liao, W., Yang, R., li, S., 2017. Development of multi-energy control system for marine micro-grid based on photovoltaic-diesel generator-battery. Shipbuild. China 58, 170–176.

Yu, W., Zhou, P., Wang, H., 2018. Evaluation on the energy efficiency and emissions reduction of a short-route hybrid sightseeing ship. Ocean Eng. 162, 34–42.

Zhang, X., Wang, C., Jiang, L., An, L., Yang, R., 2021a. Collision-avoidance navigation systems for Maritime Autonomous Surface Ships: A state of the art survey. Ocean Eng. 235, 109380.

Zhang, G., Wang, H., Zhao, W., Guan, Z., Li, P., 2021b. Application of improved multi-objective ant colony optimization algorithm in ship weather routing. J. Ocean Univ. China 20, 45–55.

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. J. Mar. Sci. Eng. 7, 132.

Zhong, W., Li, H., Meng, Y., Yang, X., Feng, Y., Ye, H., Liu, W., 2022. USV path following controller based on DDPG with composite state-space and dynamic reward function. Ocean Eng. 266, 112449.

Zhu, Z., Lyu, H., Zhang, J., Yin, Y., 2022. An efficient ship automatic collision avoidance method based on modified artificial potential field. J. Mar. Sci. Eng. 10.