Ocean Engineering 292 (2024) 116527

Contents lists available at ScienceDirect

Ocean Engineering

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

Deep reinforcement learning based collision avoidance system for autonomous ships

Yong Wang [b], Haixiang Xu [a,b], Hui Feng [a,b,*], Jianhua He [c], Haojie Yang [b], Fen Li [d], Zhen Yang [e]

[a] Key Laboratory of High Performance Ship Technology (Wuhan University of Technology), Ministry of Education, Wuhan, 430063, China
[b] School of Naval Architecture, Ocean and Energy Power Engineering, Wuhan University of Technology, Wuhan, 430063, China
[c] School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, United Kingdom
[d] Personnel Department of Wuhan University of Technology, Wuhan, 430070, China
[e] Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Zhuhai, 519000, China

ARTICLE INFO

Keywords:
Ship collision avoidance
Deep reinforcement learning
Parameter sharing
Navigation safety
Collision map

ABSTRACT

Autonomous ships is a key to avoid accidents caused by human errors and improve maritime safety. However, unlike the autonomous vehicles counterpart, collision avoidance for autonomous ships faces many challenges due to the hash driving environments, difficult ship control and large stopping distance. In this paper, we investigate a collision avoidance system for autonomous ships under complex encounter scenarios, such as busy ports. In the system various sensors are used to detect objects and perceive the maritime environments. To help the autonomous ships handle the complex and dynamic scenarios that may be encountered, a collision map used to describe the ships encounter scenarios is generated and utilized as the input of a deep reinforcement learning (DRL) model. The DRL model is applied to make collision avoidance and safe driving decisions. New reward functions are proposed to train the DRL model to generate safe ship maneuver actions to reduce collisions and ensure compliance with the Convention on the International Regulations for Preventing Collisions at Sea (COLREGs). Furthermore, a self-adaptive parameters sharing approach is designed for fast convergence and collision avoidance performance of the DRL model, where the parameters of the fully connected layers are shared and the correlation layers are self adapted for the DRL critic and actor networks. Simulation results show that the proposed system has high DRL convergence speed and excellent collision avoidance.

1. Introduction

Maritime navigation safety is a critical concern for the global shipping industry. As human errors is the leading cause of maritime collision accidents, autonomous ship is a key to avoid accidents and improve maritime safety.

With the development of maritime shipping, which undertakes more than 90% of international trade (Joung et al., 2020), marine traffic has become more and more complicated and the number of maritime accidents has been increasing. Based on the accident statistics compiled by the European Maritime Safety Agency for European Union flag ships between 2014 and 2022, it was found that navigational accidents constituted 43% of all incidents, with collision accidents accounting for 13% of these occurrences (EMSA, 2021). To reduce the ship accidents, International Maritime Organization (IMO) issued the interim guidelines for trial navigation of autonomous ships on the surface and the Maritime Autonomous Surface Ships (MASSs) is a pivotal role of it (Aiello et al., 2020; Yan et al., 2020). Collision avoidance is considered to be a highly complex and crucial autonomous operation for MASSs, which should consider the risk, efficiency of navigation, and characteristics of ship motion (Wang et al., 2020). Therefore, an effective autonomous collision avoidance algorithm, which possess a comprehensive comprehension of the maneuverability characteristics, traffic scenarios, navigation rules, cognitive ability and expert knowledge, is imminent for MASSs.

Research on ship autonomous collision avoidance has been a popular and challenging subject within the field of ship navigation safety. The traditional collision avoidance algorithm, such as artificial potential field (APF) (Lee et al., 2019), velocity obstacle (Shaobo et al., 2020), A* algorithm (Richards et al., 2019) and genetic algorithm (Wang et al., 2021) have been studied and developed for decades. These algorithms can work well in the static environment and simple driving environment. However, it is difficult for the traditional collision avoidance algorithms to handle complex dynamic environments, especially complex encounter scenarios for ships such as busy ports.

The deep reinforcement learning (DRL) has been applied to a wide range of applications and demonstrated great performance on dealing

* Corresponding author.
E-mail address: feng@whut.edu.cn (H. Feng).

https://doi.org/10.1016/j.oceaneng.2023.116527
Received 24 August 2023; Received in revised form 17 November 2023; Accepted 4 December 2023
Available online 12 December 2023

with complex and dynamic environments, and its application to ship autonomous driving and collision avoidance has also been studied. Du et al. (2021b) used an improved DRL method to make path planning for ships. Xu et al. (2022a) and Meyer et al. (2020) dealt with dynamic collision avoidance problem via DRL. However, there are some challenges on the convergence and safety of DRL in complex ships encounter scenarios where the number of ships is uncertain. More specifically, in this paper we will focus on three critical problems of applying DRL for ship collision avoidance. The first problem is how to improve the convergence speed of DRL algorithm. This is important as it serves as the basis for model application. The second problem is how to represent the complex ships encounter scenarios for DRL model. The third problem is to ensure the navigation safety and the DRL model based sailing will comply with the COLREGs, which is important for the models to obey the driving rules and be applicable to real world driving environments.

In view of the research gap and challenges on ship collision avoidance, this paper develops an effective DRL based collision avoidance system with three key mechanisms to solve the above research problems. The first one is parameter sharing in which the critic network and actor network for DRL model. Parameter sharing can reduce the number of model parameters and consider the correlation between actor and critic. However, balancing correlation and interference between actor and critic is important for this mechanism. In this paper, an improved parameter sharing mechanism, named self-adaptive parameter sharing, is applied to DRL model (self-adaptive parameters-shared DRL, SPS-DRL). The second one is the way to represent the complex ships encounter scenario. In this system, object detection module will provide accurate perception of the ships encounter scenario and the perceived information will be modeled into a map of objects with potential collisions. The potential collision map will be fed to the DRL model for safe driving decision making and collision avoidance. The third is on devising novel reward functions. Reward functions is critical for DRL models. With proper reward functions, the DRL model is expected to solve the problems of collision avoidance and compliance with the COLREGs. Simulations are run to evaluate the proposed solution, which demonstrate it feasibility and effectiveness for autonomous ship collision avoidance with high convergence speed and better applicability to ship collision avoidance.

The main contributions of this study are summarized as follows.

- A new sharing mechanism with hybrid structure for policy network and value network is proposed, which can improve the convergence speed and performance of DRL model.
- The collision map with three layers representing different features of ship navigation is designed, and the collision map can have a more precise description of ships encounter scenarios for DRL model.
- Several proper reward functions are developed to make DRL model obtain a safe and complaint with COLREGs operations for the ship collision avoidance.
- A collision avoidance method based on SPS-DRL is employed, and its effectiveness is evaluated to provide a way forward for MASSs.

The remainder of this paper is organized as follows. Section 2 reviews relevant literatures. In Section 3, the framework of collision avoidance system is introduced. Section 4 presents the design of collision map, the self-adaptive sharing mechanism, and the design of reward functions. Simulation results and discussion are presented in Section 5. Finally, the conclusions of this study and future works are discussed in section Section 6.

2. Literature review

2.1. Traditional ship collision avoidance methods

Ship collision avoidance algorithms have been studied and developed for decades. These algorithms can be roughly divided into artificial potential field (APF) method, geometric method, heuristic method, and artificial intelligence method (Zhang et al., 2021). The APF is widely used in ship collision avoidance for its simple principle and easy implementation. Naeem et al. (2016) introduced an enhanced APF that takes into account the COLREGs. Zhu et al. (2021) proposed a collision avoidance method that considers the COLREGs and the characteristic of ship motion by modifying the repulsive potential field in APF. APF follows a simple and efficient execution process, which attraction potential field guilds ship to the destination and repulsive potential field prevent ship from approaching obstacles. However, there is a special case when two potential fields are in opposite directions and the intensity of field is equal, then APF falls into the trap area of local minimum and cannot get the collision avoidance path. Geometric method includes visibility graph algorithm (Wu et al., 2020), Voronoi diagram algorithm (Niu et al., 2020), velocity obstacle algorithm (Shaobo et al., 2020; Yuan et al., 2020; Chen et al., 2018), fuzzy logic (Brcko et al., 2021; Fışkın et al., 2021) and so on. These algorithms implement ship collision avoidance based on the spatial geometry relationship between ships and obstacles. Therefore, such methods have little optimization space in the process of generating collision avoidance paths. Heuristic method is derived from optimization algorithms, such as the A* algorithm, genetic algorithm, particle swarm optimization algorithm, and so on. Richards et al. (2019) proposed a collision avoidance algorithm combining Voronoi algorithm and A* algorithm to find a minimum-risk path. Wang et al. (2021) conducted research on cooperative avoidance of multiple USVs (unmanned surface vehicles) using genetic algorithm. Xue (2022) used an improved particle swarm optimization to realize the ship grounding avoidance. For the static scenes, the heuristic methods have excellent global optimization capabilities. Nevertheless, the efficiency of heuristic algorithms will be significantly reduced in dynamic scenarios. Hence, the traditional methods, including APF, geometric methods, heuristic methods, have difficulties in dealing with complex encounter conditions with dynamic obstacles.

2.2. Deep reinforcement learning

For the study of DRL algorithms in ship collision avoidance, most research works focus on model-free DRL due to its easy implementation. model-free DRL models usually can be divided into value-function-based DRL and policy-gradient-based DRL (Kiran et al., 2021). The value-based DRL is applicable to discrete action spaces. And the typical value-based DRL algorithm is DQN (deep Q-learning). Chen et al. (2019) used the Q-learning method to complete the path planning and control of USV. Li et al. (2021) proposed an improved DQN algorithm for ship collision avoidance that employs the APF method to optimize the action space and rewards of DQN. The policy-gradient-based DRL methods, including the deep deterministic policy gradient (DDPG) method, the proximal policy optimization (PPO) method, and so on, are good at dealing with continuous action space. And Zhou et al. (2022) proposed an improved DDPG method to solve the problem of sparse reward, designing the memory pool modification and success pool modification to smooth the training process. Chun et al. (2021) proposed a method to quantitatively assess the collision risk, and combined the assessment result of collision risk with the PPO algorithm to generate a collision-free path for USV.

For the application of DRL to ship collision avoidance, there is a critical problem due to the fixed dimensions of input in DRL model, which means the own ship can only handle the situations with a fixed number of obstacles. The first solution to this problem is to find the most dangerous ship by assessing collision risk, then avoid collision with the ships one by one (Xu et al., 2022b; Du et al., 2021b; Xu et al., 2022b). The second approach is to design multidimensional matrices with different forms to describe the surrounding obstacles, such as azimuth matrix (Zhao and Roh, 2019), distance matrix (Meyer et al., 2020), and so on. However, these methods have some drawbacks. For example, the first one converts the multi-ships encounter conditions

to single-ship encounters, which means the interactions between ships are ignored. Meanwhile, both the first one and the second one use feature vectors to represent the states of obstacles and the own ship, and the correlation between features is less considered. In addition, as describing the state of ship, the regional features such as ship domain should also be provided. Woo and Kim (2020) used the grid map to describe the state of own ship, and defined three layers of the map, including path layer, dynamic obstacle layer, and static obstacle layer. Taking the grid map as the input of DRL model can fix the problem with a changing number of obstacles. However, this method of layer division ignored the relationship between layers and the features of obstacles, such as velocity, which cannot be described quantitatively. Meanwhile, the regional feature, such as ships domain that is significant for ship navigation safety, has also been less considered.

In addition to the applicability of DRL in ship collision avoidance, accelerating the convergence speed of DRL is a key problem that has not been solved effectively. One of the methods is use of parameter sharing mechanism. Parameter sharing comes from multi-task supervised learning (MTSL). In MTSL, different tasks have the same input data, and the parameter sharing mechanism is used to consider the correlation among multiple tasks and get the common information in model parameters (Dong et al., 2022). The most commonly used parameter sharing mechanism is hard parameters sharing, which means all the hidden layers of leaning network are the same and the output layer is different for different tasks. Sun et al. (2020) proposed a sparse sharing architecture for multiple tasks based on hard sharing mechanism. Meanwhile, to improve the performance of sharing network, some adaptive sharing modes had been proposed (Wang et al., 2022).

For the application of parameter sharing mechanism in DRL, D'Eramo et al. (2020) showed the effectiveness of sharing mechanism in multi-task reinforcement learning by providing theoretical guarantees. Schiewer and Wiskott (2021) applied the shared policy networks to model-based reinforcement learning. Considering the interference for gradient update in DRL model from different tasks, Teh et al. (2017) proposed the distilled policy to capture common features across tasks and each task policies were trained to stay close to distilled policy. For the DRL algorithms based on actor–critic framework, sharing mechanism can also be used in policy network and value network. Cobbe et al. (2021) analyzed the advantages and disadvantages of parameters shared PPO algorithm and proposed a reinforcement learning framework named phasic policy gradient, which split the optimization of DRL into two phases including advancing training phase and features distilling phase. Therefore, in the design process of sharing mechanism for DRL model, it is crucial to comprehensively consider the correlation and interference among multiple tasks.

It is observed that the existing research on DRL based ship collision avoidance mainly focused on finding the most dangerous ship and used feature vectors as input of DRL models, which is difficult to describe the complex encounter scenarios and represent the spatial relationship between ships. A few studies develop grid map to represent the relationship between encountered ships. However, regional feature was not considered in the development of grip map. Additionally, the parameter-sharing mechanism in DRL needs to effectively manage the correlation and interference among different task networks, which can accelerate the convergence and improve performance of DRL model. It is a crucial concern but rarely studied in the literature.

3. Framework of proposed collision avoidance method

In the collision avoidance operation utilizing DRL, the process consists of four sequential steps: environment information collection, environment modeling, determination of ship steering angle through DRL, and execution of the chosen action. The diagram for the framework of the collision avoidance decision-making system based on DRL is shown in Fig. 1.

As shown in Fig. 1, the environment perception is foundation of ship collision avoidance system, which can obtain the information of obstacles and own ship. However, in this paper, there are no actual sensors, and the information of obstacles and own ship is assumed to be accurately obtained all the time. After getting the information, the environment modeling operation should be executed. In this paper, environment modeling is used to represent the ship encounter scenario in detail based on the information of obstacles and own ship. The result of environment modeling operation would be the input of the DRL-based collision avoidance decision-making model. Then, the rudder angle command will be obtained by the decision-making model and executed by ship simulation model. Meanwhile, the environment will feedback a reward to the decision-making model. And then, the environment is updated and the above process is repeated.

Based on this, the research content of this paper consists of the following parts: Firstly, an environmental modeling approach is developed by constructing a collision map using acquired environmental information. Secondly, the performance of DRL method is enhanced through the implementation of a self-adaptive parameters-shared mechanism. Then the reward functions and action are redesigned to better suit the specific requirements of ship collision avoidance. Lastly, a collision avoidance decision-making system is established by integrating the aforementioned components into the SPS-DRL framework.

4. Ship collision avoidance based on SPS-DRL

In this section, the collision map is introduced first, which serves as the input of DRL model. Subsequently, the self-adaptive parameters-shared mechanism is proposed to enhance the convergence speed of DRL. Lastly the design principles for the DRL model for ship collision avoidance is presented, including the reward function and action.

4.1. Construction of collision map

Collision map is designed to represent the information of encounter scenario. The map consists of two conceptual layers, namely spatial layer and velocity layer. The subsequent part provides a detailed explanation of the definition and characteristics of each layer.

(1) Map size

Before the introduction to conceptual layer, the size of collision map is discussed first. Due to the large inertia for ship motion, the collision avoidance maneuvers of ships need to be made in advance. As a result, the early warning range for collision avoidance should be large, typically around 5 to 6 nautical miles. However, directly using this warning range to construct the grid map would require a large grid size, which may compromise the efficiency of the collision avoidance algorithm and result in the loss of some details. To solve this problem, this paper proposed a method inspired by the pooling operations in convolutional networks, which involves building a scaling grid map.

Fig. 2 depicts the procedure for constructing a scaling map. Initially, the original map is created using the warning range (20,000 × 20,000 m sized), with the own ship positioned at the map's center. The number of grids in original map is 1000 × 1000. Then, the scaling map is generated by applying a mean pool operation with 10 × 10 filters and stride 10. The size of scaling map is 100 × 100 with large size grids (200 × 200 m sized). And the scaling map will be the input to ship collision avoidance model.

(2) Spatial layer

The spatial layer illustrates the spatial relationship between own ship and obstacles, including static and dynamic obstacles. Fig. 3 expresses the design principle of this layer. For simplicity of representation of the obstacle in grid map, the static obstacles are assumed to possess a circular shape. Additionally, a safety margin is incorporated around these static obstacles to ensure safe navigation. As for dynamic obstacles, ship domain is represented on grid map. Ship domain

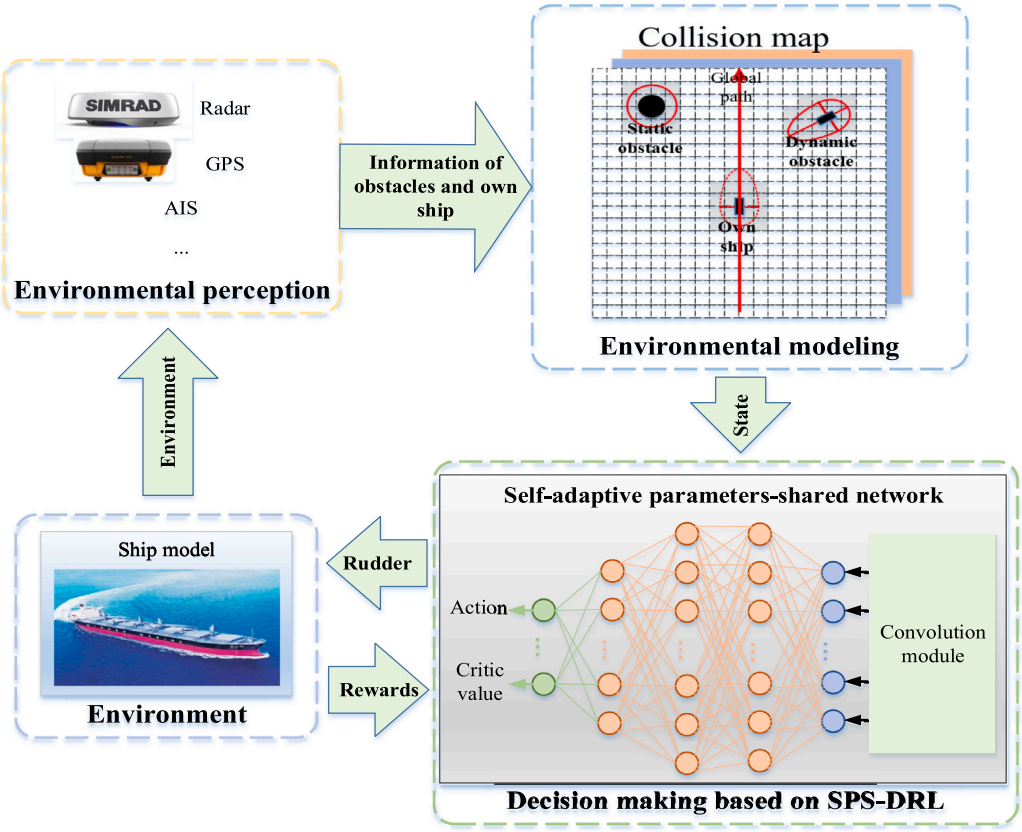


Fig. 1. The framework of collision avoidance decision-making system.

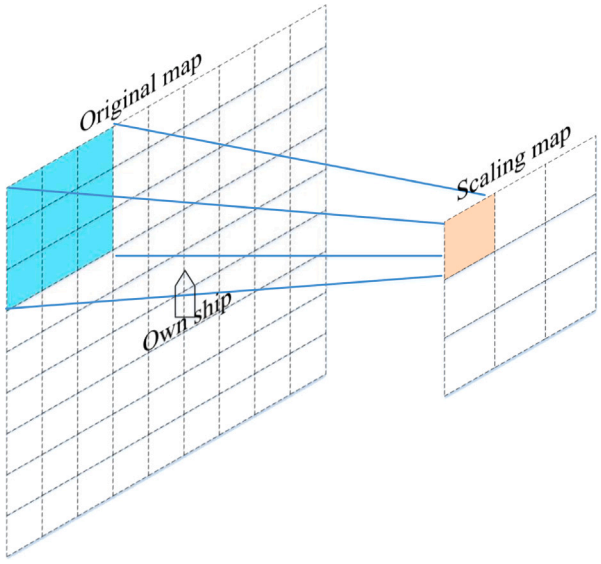


Fig. 2. Scaling map construction method.

means an area which other vessels would avoid entering, and there are many ship domains with different shapes. For example, Coldwell's domain shaped the area with ellipses of different sizes (Coldwell, 1983), Goodwin used threes disparate segments with different but constant radius to construct the ship domain (Goodwin, 1975). Fiskin et al. (2020) distinguished the approaches of determining ship domain into four groups: Empirical (statistical), analytical, knowledge-based and probabilistic. Meanwhile, they created an asymmetrical polygonal ship domain based on fuzzy rules, which had a comprehensive consideration of various navigational factors. However, in this paper, in order to simplify the layer construction process, the ship domain is depicted as a combination of a half-elliptical shape at the fore section and a half-circle shape at the aft section, as shown in Fig. 3. The dimensions of the major and minor axes of the elliptical shape are determined by the size and speed of the ship, with a specific calculation method proposed by Tam and Bucknall (2010). Besides, the global path of own ship is also presented on grid map.

To standardize the input of DRL model, all the grid map layers use an intensity value between 0 and 255. In spatial layer, the value of grid occupied by static obstacles, dynamic obstacles and own ship is equal to 125, which can ensure the superposition value is not out of range. The basic value of path grid is 125, and it is going to decay linearly. The decay coefficient is calculated by following equation.

$$\alpha = 0.5 \times \left(1 + \frac{dis_{os} - dis_i}{R_{os}}\right) \tag{1}$$

Here, α is the decay coefficient. dis_{os} and dis_i are the distance from own ship to target point and distance from the ith path grid to target point, respectively. R_{os} is the radius of early warning, which is equal to 10,000 m. the changes of decay coefficient can suggest the expected path direction. The closer the path grid is to the target point, the larger the grid value is.

(3) Velocity layer

the spatial layer is stationary, and it lacks dynamic information, such as speed. For dynamic obstacles, the velocity of dynamic obstacle must be considered in the avoidance decision process. To express this, a velocity layer is introduced in this research, as depicted in Fig. 4. In the velocity layer, each grids have a velocity attribute value. As the collision map is synchronized with the own ship's movement, the velocity attribute value represents the relative velocity with respect to own ship. In Fig. 4, the velocity attribute values of own-ship grids are zero. And the values of empty grids, static-obstacle grids, and path grids

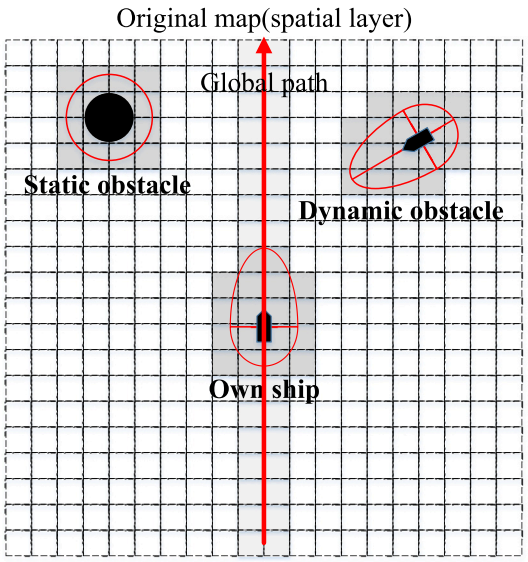


Fig. 3. Spatial layer.

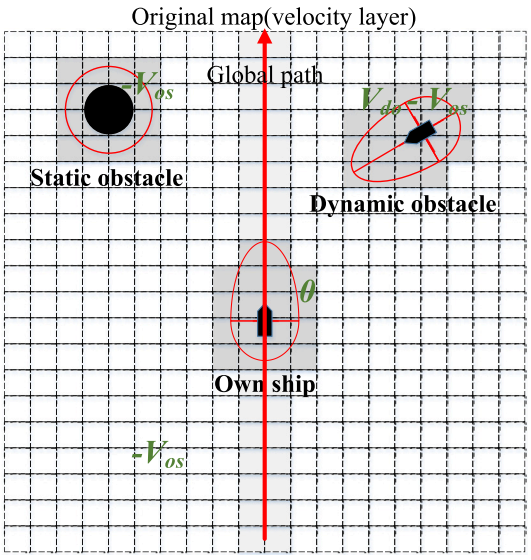


Fig. 4. Velocity layer.

are equal to $-V_{os}$, where V_{os} represents the speed of own ship. For the dynamic-obstacle grids, the values are obtained by the relative velocity between dynamic obstacle and own ship ($V_{do} - V_{os}$, V_{do} is the speed of dynamic obstacle).

To unify the value range of the input to the DRL model, the velocity attribute value of grid in velocity layer will map between 0 and 255, the specific mapping method shown as follows.

$$V_{attr} = 125 + \frac{V_{relate}}{V_{\max}} \times 125 \tag{2}$$

In this equation, V_{attr} is the velocity attribute value of grid. V_{relate} represents the relative velocity of grid. V_{max} is an artificial hyperparameter, which means the possible maximum of velocity.

4.2. Design of SPS-DRL

4.2.1. Self-adaptive parameters-shared mechanism

In DRL task, sharing parameters has a clear advantage that features trained by each objective can be used to better optimize the other, which can improve training samples' utilization rate and accelerate the convergence of model (Cobbe et al., 2021).

The common parameter sharing mechanisms in DRL model are divided into two categories: (a) hard sharing approaches, which have the same input layer and hidden layer, and work out the actor and critic value specifically on the output layer; (b) half-hard sharing approaches, which allow the critic and actor have separate network and share the input layer and partially hidden layer.

Since the sharing network is employed in DRL model, a disadvantage in the training process is appeared. There is an interference while optimizing the sharing parameters by critic loss and actor loss. It is hard to balance the weight between the critic loss and actor loss. The primary cause for the interference is the lack of independence between critic network and actor network. But complete independence network for critic and actor will increase the number of parameters greatly and reduce the convergence rate of model. In this paper, a self-adaptive sharing network is proposed and applied to the DRL model. The self-sharing network is shown in Fig. 5. As shown in Fig. 5, the basic network structure is the same for critic network and actor network, which means the same weight coefficient $\omega_{i,j,k}$ and bias coefficient bi are used in both network model. To keep the independence of critic and actor network, an independent self-adaptive layer is put into the hidden layer of critic and actor networks. The principle of self-adaptive layer is described by the following equations:

$$\hat{y}_i = y_i \cdot f\left(\beta_{i,j}\right) \tag{3}$$

$$f\left(\beta_{i,j}\right) = k \cdot \beta_{i,j} \tag{4}$$

where $\beta_{i,j}$ is the adaptive parameters, y_i represents the output value of traditional hidden layer, and $\hat{y}_i$ means the output value of improved hidden layer. The idea of self-adaptive layer is similar to the dropout method. For each neuron in the hidden layer, the relational degree with different task is different. Thus, $f\left(\beta_{i,j}\right)$ is the relational degree function for specific task. In this study, a linear relational degree function is adopted, with the linear coefficient k being set to 1. The larger the absolute value of this function is, the greater the neuron's relational degree of this task is. Positive value of relational degree means positive relational degree. Conversely, as the absolute value is close to 0, it represents that the neuron has nothing to do with this task. In Fig. 5, $f_C\left(\beta_{i,j}\right)$ and $f_A\left(\beta_{i,j}\right)$ are the relational degree function for critic and actor respectively.

In addition, in the process of model training, a cross training method is proposed in this study, which means that sharing parameters and adaptive parameters are not trained at the same time. When the sharing parameters are trained, the adaptive parameters are fixed. On the contrary, once the adaptive parameters are trained, the sharing parameters are fixed, and the training for sharing parameters and adaptive parameters is conducted alternately. The meaning of cross training method is that the sharing network is responsible for extracting features, and based on it, the adaptive layer is used to rebuild the network for specific task. Therefore, there is a sequence between sharing network and adaptive layer.

4.2.2. Network structure of SPS-DRL

As described in Section 4.1, a collision map with spatial layer and velocity layer is used as the input of DRL model. For practical consideration, the velocity layer is divided into east speed layer and north speed layer. Thus, the size of collision map is 100 × 100 × 3. As depicted in Fig. 6, the structure of SPS-DRL in this work for ship collision avoidance consists of two modules: the convolution module and the fully connected module. The convolution module is responsible for extracting features from the collision map. On the other hand, the fully connected module utilizes the extracted features to make decisions. The self-adaptive sharing network, introduced in Section 4.2.1, is employed in the fully connected module. The specific parameters pertaining to the structure of DRL network are illustrated in Fig. 6.

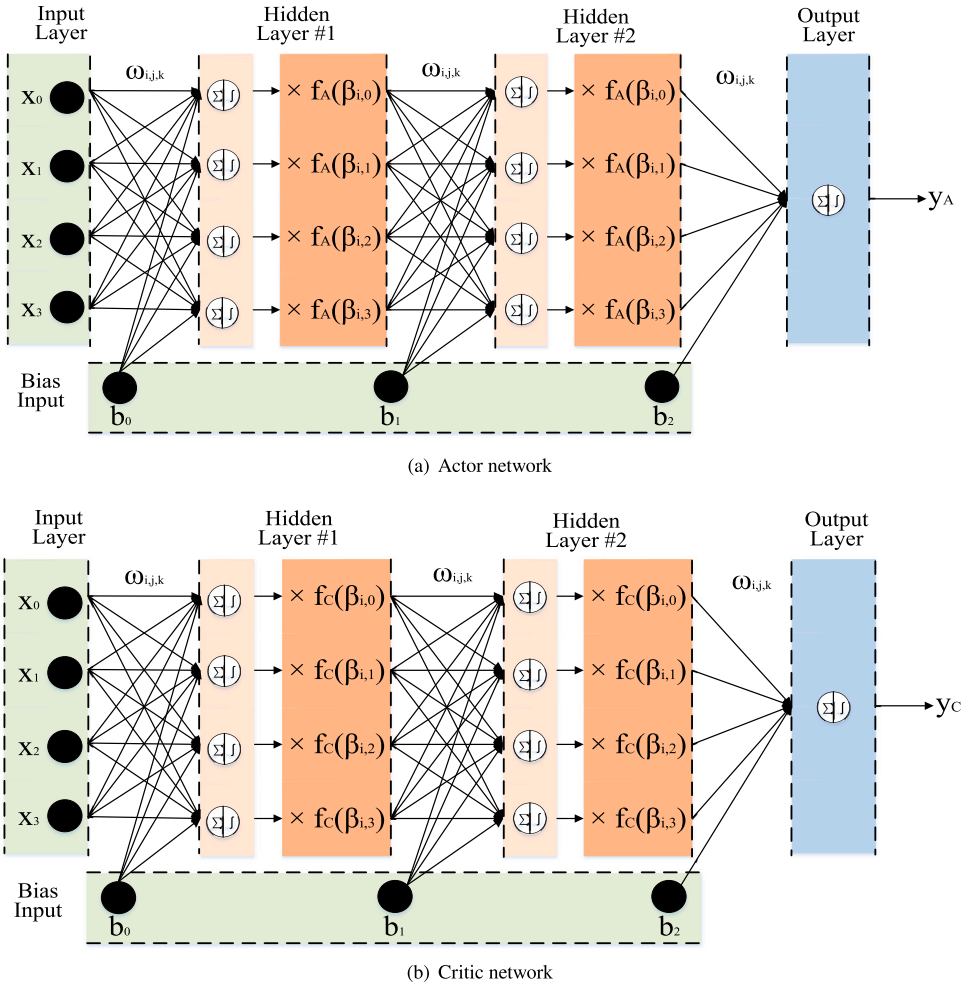


Fig. 5. Self-adaptive sharing network.

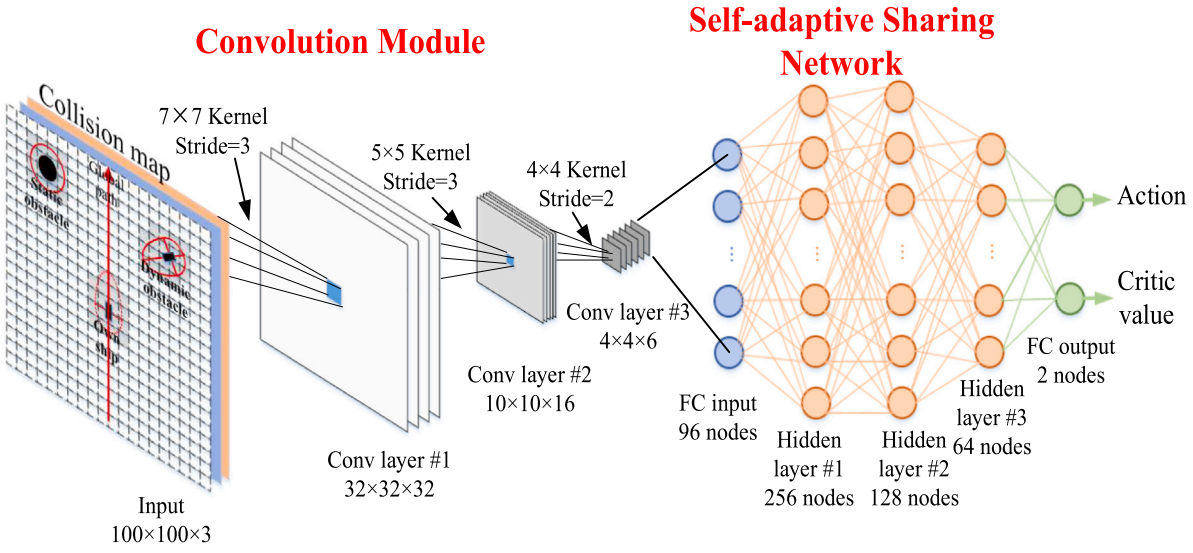


Fig. 6. Structure of the SPS-DRL.

4.3. Design of rewards and action for SPS-DRL

4.3.1. Action space

To prevent collisions, the ship will take actions such as adjusting its direction and changing its speed. Typically, in order to save energy and simplify the maneuvering operations, in the open sea, adjusting the direction of ship is the mainstream operation to avoid collision. Consequently, in this study, the rudder angle is defined as the action selected by the policy of the agent. the value of rudder angle is limited to a range of −35° to 35°. Negative values mean turning rudder to

left and positive values mean turning to the right. To reduce the noise of the output and avoid incorrect decision, a low-pass filtering process proposed by Rongcai et al. (2023) was used and expressed using Eq. (5).

$$A_n = a \cdot a_n + (1 - a) \cdot A_{n-1} \tag{5}$$

where a is a filter coefficient with a value 0.6, a_n is the current output of DRL model, A_{n-1} is the filter output value at previous moment.

4.3.2. Formulation of rewards

In the decision-making process using DRL model, the performed action should be evaluated using the reward function and the obtained rewards are used to train the critic and actor network. However, the design of reward function is related to the specific problems. To address the ship collision avoidance problem, the reward functions are defined as follows. In this paper, there exists six types of rewards established to avoid ship collision, including goal reward, collision reward, collision risk reward, rule following reward, error reward of track deviation, and time reward. These rewards are denoted as r_{goal}, $r_{collision}$, r_{risk}, r_{rule}, r_{error}, and r_{time} respectively. The weighted mean of these rewards is regarded as the final output of reward function, and the specific calculation formula is as follows.

$$R_{mean} = W \times \begin{bmatrix} r_{goal} & r_{collision} & r_{risk} & r_{rule} & r_{error} & r_{time} \end{bmatrix}^T \tag{6}$$

Here, R_{mean} is the weighted mean of rewards. W is the weight matrix, set to be [0.25, 0.1, 0.1, 0.3, 0.15, 0.1]. The specific calculation formulas of six type of rewards are introduced as follows.

(1) Goal reward

The goal reward is used to guide the own ship to the target point. In this paper, calculation method of goal reward proposed by Chun et al. (2021) is adopted. Fig. 7 illustrates the calculation principle of goal reward. As it depicts, P_t is the current position of own ship, and P_{t-1} is the position of own ship at the last moment. The concentric circle around P_{t-1} is the available position own ship reaching at current moment. Among these available positions, $P_{t_{max}}$ is the closest point to the goal. as shown in Fig. 7, if own ship moves to $P_{t_{max}}$ at present moment, it will get maximum goal reward. Eq. (7) presents the formula for calculating goal reward.

$$r_{goal} = \frac{L_{t-1} - L_t}{V_{OS}} \tag{7}$$

Where, L_t and L_{t-1} are the distance between current position and position at last moment (P_t and P_{t-1}) of own ship and the goal, respectively. V_{os} is the speed of own ship. According to the Eq. (7), the closer the own ship is to the goal, the higher the reward own ship gets. And the maximum and minimum of goal reward are 1 and −1, respectively.

(2) Collision reward

Collision reward is the penalty term for ships navigation. There are two navigation stages divided by the distance between own ship and dynamic obstacle in the calculating process of collision reward. The two navigation stages are shown in Fig. 8. As depicted in Fig. 8, there are two areas around the dynamic obstacle, which the red line is the boundary of the ship domain and the black dotted line is the swelling boundary of ship domain. When own ship enters the dotted line area, navigation state one starts, and a warning will be pop-up and the own ship will keep getting small penalties. Once own ship gets into the ship domain of dynamic obstacle, collision states happens, and own ship will get a large penalty value. In this work, it is assumed that the collision happens when one ship reaches the boundary of ship's domain. However, in Fig. 8, the specific calculation methods of d_{aft} and d_{fore} are described in previous study (Tam and Bucknall, 2010), and D_{aft} and D_{fore} can be calculated from Eq. (8) and Eq. (9).

$$D_{aft} = \eta \cdot d_{aft} \tag{8}$$

$$D_{fore} = \eta \cdot d_{fore} \tag{9}$$

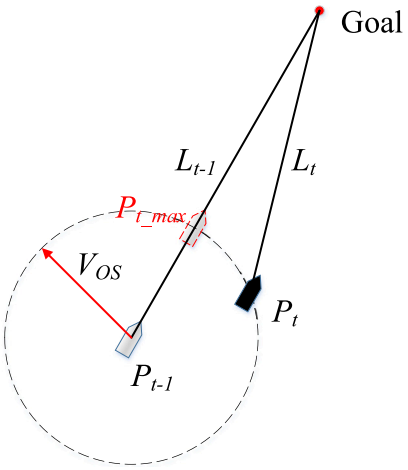


Fig. 7. Definition of goal reward.

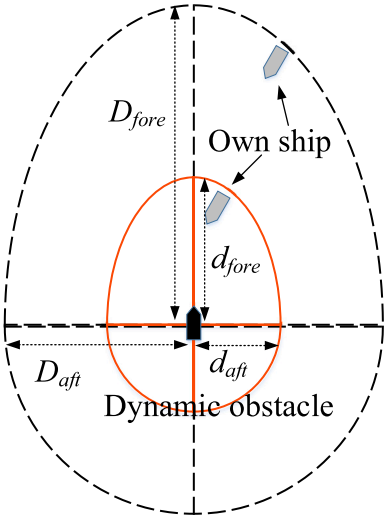


Fig. 8. Definition of navigation stages.

Here, D_{aft} and D_{fore} indicate the radius and the major semi-axis for the aft and fore section of the swelling ship domain. η is the swelling coefficient, which is set to be 2.5 in this paper. The specific value of penalty for own ship in different state are as follows.

$$r_{collision} = \begin{cases} -1 & \text{in state one} \\ -1000 & \text{collision happens} \end{cases} \tag{10}$$

(3) Collision risk reward

During the navigation of ships, collision risk (CR) is a very important factor for masters to assess the encounter situation and determine the time to avoid collision. Studies on CR have always forced on the ship domain method and CPA (closest point of approach) method. The ship domain method aims to create a distinct boundary around ships, which prevents other ships from getting into this domain and guarantees the navigation safety. Ship domain is an important factor in the process of navigational risk analysis. Rawson and Brito (2020) analyzed the navigational risk in waterway with special scenarios by ship domain. But, the ship domain cannot predict the CR in the future. For the CPA method, there are two important elements, including the distance of closest point of approach (DCPA) and the time to the closest point of approach (TCPA). DCPA and TCPA can effectively aid ship masters in evaluating potential risks in ship encounter scenarios.

Meanwhile, DCPA and TCPA can also assess quantitative CR (Chun et al., 2021). Otherwise, ship maneuver is another factor for the CR, Du et al. (2021a) took the first evasive maneuver as a basis to construct the ship domain, which is correlated with the risk perception. Therefore, in this paper, TCPA, DCPA, some parameters of ship domain and evasive maneuver are used to calculate the collision risk reward, and the specific calculation formula is shown as follows.

$$r_{risk} = 0.5 \times \tanh\left[\ln\left(\frac{DCPA}{d_{sd}}\right)\right] + 0.5 \times \tanh\left[\ln\left(\frac{abs\,(TCPA)}{t_{norm}}\right)\right] \quad (11)$$

In this equation, $tanh\,()$ is the hyperbolic tangent function, whose value range is -1 to 1. $ln\,()$ is the natural logarithm function. $abs\,()$ is the function to get absolute value of number. d_{sd} is equal to the average value of d_{aft} and d_{fore}, which are the parameters of ship domain shown in Fig. 8. t_{norm} represents a standard value related to the ship maneuverability, and it is equal to the time when the heading of ship changes 90 degrees in the turning test, setting to be 120 s d_{sd} and t_{norm} are the reference value for distance dimension and time dimension, respectively. When DCPA is smaller than d_{sd}, it means the collision will probably happen in the future. For the TCPA, while it is less than t_{norm}, the ship may not have enough time to change heading to avoid collision. In the process of risk calculation, when TCPA is less than zero, which means there is no risk of collision, r_{risk} is equal to 1. Otherwise, r_{risk} is calculated by Eq. (11).

(4) Rule reward

The role of rule reward is to guide the own ship to comply with the COLREGs, which defines the give-way vessel and the stand-on vessel in encountered situation. the give-way ship should be in duty bound to take action to avoid collision. the stand-on ship should keep the course and speed, and pay close attention to the encountered situation. Four encountered situation are specified in COLREGs, as shown in Fig. 9. Meanwhile, the desired collision avoidance actions for each encountered situations in COLERGs are also shown in Fig. 9. In this work, the own ship will get two types of rule rewards, namely reward for following rules and reward for the violation of rules. The values for these two types of rewards are 1 and -1, respectively. For example, in the "Head-on" and "Crossing (give way)" situations, own ship should pass through the port side of other ship. The classification method of ship encounter scenarios in Thyri et al. (2020) is adopted. When other ship gets into the early warning range, the rule reward obtained by own ship is 1 if own ship is on the port side of other ship and other ship is also on the port side of own ship, as shown in Fig. 10. In Fig. 10, the area around ship is consistence with the warning range introduced in Section 4.1 and the side length is 10.8 nautical miles. Otherwise, it is -1, and as they are past and clear, the value of rule reward is fixed at 1. In addition, during the encounter, all obstacle ships keep a constant course and speed. Eq. (12) presents the formula for calculating the rule reward.

$$r_{rule} = \begin{cases} 1 & \text{following COLREGs or past and clear} \\ -1 & \text{otherwise} \end{cases} \quad (12)$$

(5) Error reward

The error reward is defined as a reward given according to the track deviation of own ship, and the definition of track deviation is described

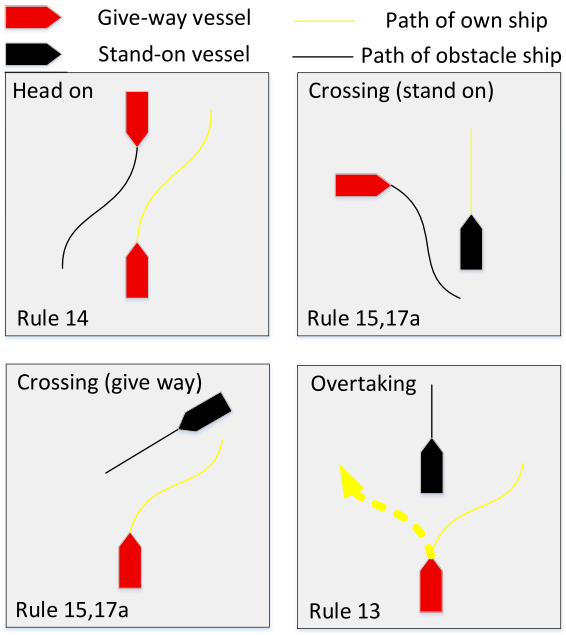


Fig. 9. Desired collision avoidance action for each encountered situation in COLERGs.

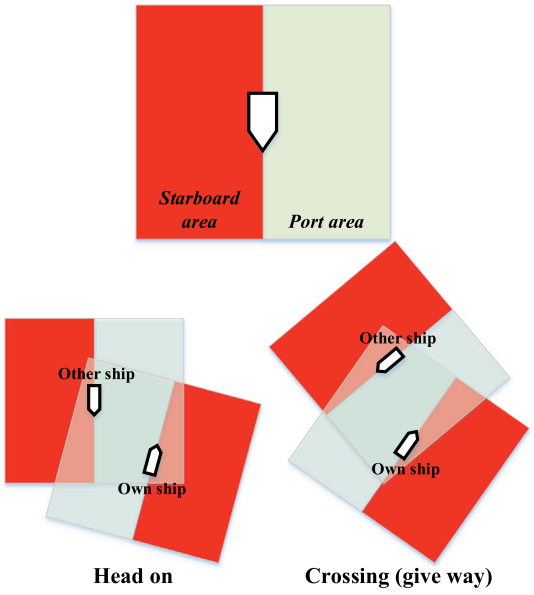


Fig. 10. The situation with following rules reward.

in Fig. 11. As depicted in Fig. 11, the track deviation is the distance between the position of own ship and the reference path, and there is a safe area around the reference path. When own ship is in the safe area (P_0 in Fig. 11), the own ship will obtain the maximum error reward. Once it sails out of the safe area (P_1 in Fig. 11), the error reward will gradually decrease. The specific calculation formula is presented as follows.

$$r_{error} = \begin{cases} 1 & y_e < e_{safe} \\ \tanh\left(-\ln\left(\frac{y_e}{e_{norm}}\right)\right) & y_e > e_{safe} \end{cases} \quad (13)$$

Here, y_e is the track deviation of own ship. e_{safe} represents the safe threshold of track deviation, which means it is acceptable when y_e is less than e_{safe}. e_{norm} denotes the standard value of track deviation,

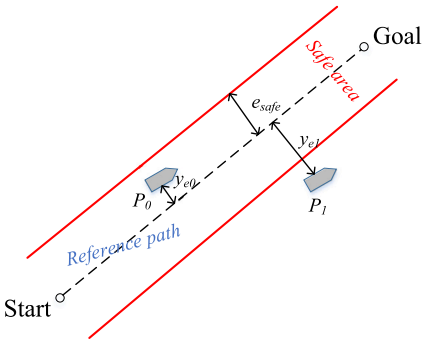


Fig. 11. Definition of track deviation.

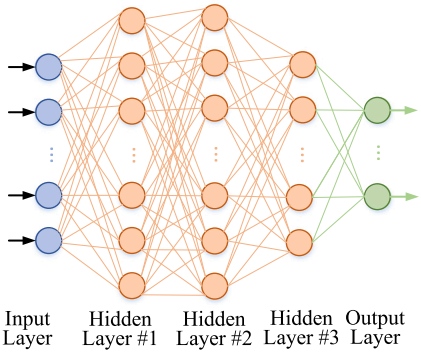


Fig. 12. Basic network structure.

equal to $e_{safe} \cdot e_{norm}$ (is the natural number). In this paper, e_{safe} is set to be 3.5 times width of ship.

(6) Time reward

The value of time reward is always equal to −1, representing the cost of time consumption in navigation. The function of time reward is to guide own ship to find the shortest path to the destination. In other word, the longer the path to destination is, the greater the cumulative time consumption is, and the total reward will become small.

In this study, R_{mean} is the normal navigation reward, and in the training process of DRL model, some special scenes, such as having a collision, running to the maximum steps, sailing out of the designated area, circling around for a long time, and reaching the goal, will terminate the training episode. For these scenes, some rewards, called done reward in this paper, should be given. The done reward should be large enough to punish or encourage the agent. Among these scenes, reaching the goal is the only positive scene, and the done reward of it is equal to 1000. All other scenes are negative, and done rewards are set to be −1000. Besides the special scenes, the done reward is equal to zero. Total reward is the sum of normal navigation reward and done reward. Eq. (14) represents the calculation formula.

$$R_{total} = R_{mean} + R_{done} \tag{14}$$

Where R_{total} is the total reward of every steps in episode. R_{done} is the done reward.

5. Simulation verification and discussion

In this section, numerical simulations are carried out to verify the proposed system SPS-DRL. First, to compare the effectiveness of different sharing mechanism in DRL, some simulations based on Gym environment are conducted. In addition, the availability of SPS-DRL in ship collision avoidance has been confirmed by handling various encounter scenarios.

5.1. Simulation comparison for different parameters-shared mechanisms

Four Gym environments are used to test the DRL algorithm with corresponding network: (a) Bipedal Walker-v3; (b) Half Cheetah-v2; (c) Hopper-v2; (d) Walker2d-v2. Meanwhile, in order to improve the algorithm performance, some settings are applied to the DRL algorithm, shown as follows:

- Normal distribution has been used to conduct state normalization, which is beneficial to the training of network.
- The negative effects caused by overlarge rewards or tiny rewards have been considered by rewards scaling method (Engstrom et al., 2020).
- Learning rate decay has been applied as the guarantee of stationarity of network training in the later stage. And in this paper, the learning rate proceeds in a linear decay.

Table 1
Comparison of the total number of parameters in different environments and models.

Parameters \ Algorithm / Gym environment	HPS-PPO	HHPS-PPO	SPS-PPO
Bipedal Walker-v3	28 297	28 134	28 937
Half Cheetah-v2	27 533	27 309	28 173
Hopper-v2	26 567	26 439	27 207
Walker2d-v2	27 533	27 309	28 173

Table 2
Hyperparameters for the training of the PPO algorithm.

Parameters	Values
Discounted rate	0.99
Advantage ratio	0.95
Max evaluation times	1465
Bath size	2048
Time step	3×10^6
Clipping hyperparameter	0.2
Learning rate of actor network	3×10^{-4}
Learning rate of critic network	3×10^{-4}

- Expect for learning rate decay, gradient clip is also used to guarantee the stationarity of network training, which can avoid the gradient explosion.

In the simulation, a DRL algorithm called PPO and proposed by Schulman (Schulman et al., 2017) is adopted, and the basic network structure is shown in Fig. 12, there are three hidden layer: #1 and #2 have 128 neurons; #3 has 64 neurons. The PPO algorithm with hard parameters-shared network (HPS-PPO) means actor and critic use the basic network in common. PPO algorithm with half-hard parameters-shared (HHPS-PPO) network means hidden layer #1 and hidden layer #2 are shared and the remaining hidden layers are independent for actor and critic. Meanwhile, to ensure that the total number of parameters is consistent in different model, the hidden layer #3, independent for actor and critic in HHPS-PPO, has 32 neurons. PPO algorithm with self-adaptive parameters-shared network (SPS-PPO) means the actor and critic share the basic network and have three independent self-adaptive layer. Table 1 shows the total number of parameters in different environments and models. As shown in Table 1, in the same gym environment, the number of parameters of different models is approximately on the same order of magnitude, and SPS-PPO has slightly more parameters than others due to the existence of self-adaptive layer.

In this simulation process, the hyperparameters of PPO algorithms are shown in Table 2. The model is evaluated after each iteration, and the evaluation rewards are averaged after each 20 times. For the different gym environments and algorithms, the results of rewards in each evaluations are shown in Figs. 13–16. In Figs. 13–16, (a), (b)

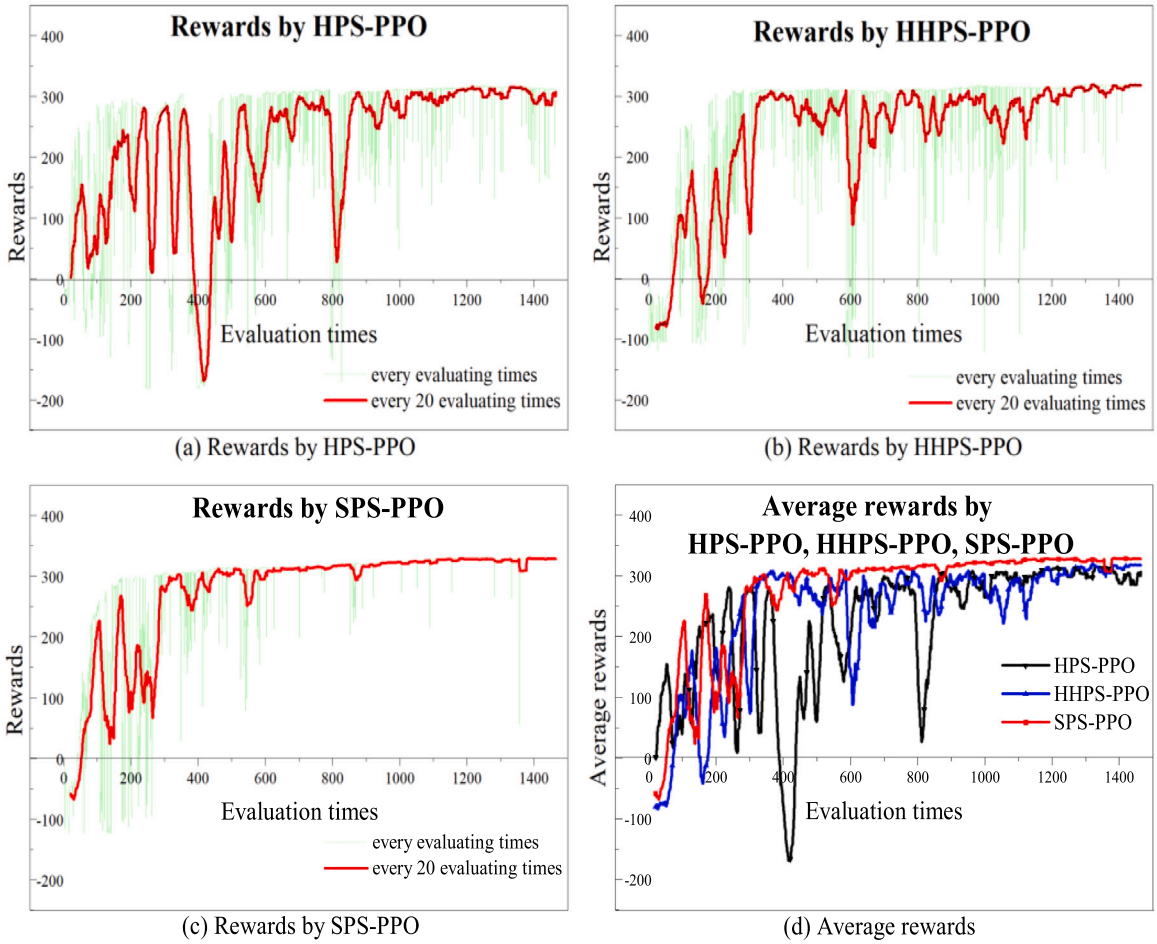


Fig. 13. Rewards obtained in Bipedal Walker-v3.

and (c) describe the variation of rewards for different algorithms in specified gym environment. In these figures, the green line represents the accumulated rewards for each evaluation times, and the red line denotes the averaged rewards for each 20 evaluation times. (d) depicts the comparison of averaged rewards of different algorithms.

According to Figs. 13–16, compared to the HPS-PPO and HHPS-PPO, the SPS-PPO can effectively improve the speed of convergence and increase the accumulated rewards. In Bipedal Walker-v3 environment, the maximum number of rewards is around 300. SPS-PPO is convergent at about 600 evaluation times, and HHPS-PPO and HPS-PPO are convergent at about 1200 and 1000 evaluation times, respectively. Also, the rewards by SPS-PPO is slightly more than HHPS-PPO and HPS-PPO. Moreover, after model convergence, SPS-PPO is more stable than others. In Half Cheetah-v2 environment, although, there are some oscillations in SPS-PPO, the obtained rewards are far more than others. In common, The SPS-PPO performs better than HHPS-PPO and HPS-PPO in the rest environment.

5.2. Simulation for ship collision avoidance system based on SPS-DRL

PPO algorithm is also employed and the structure of SPS-DRL model is shown in Fig. 6. And a fast container ship with single propeller was selected as the example ship. Also, to simplify the encounter scene, all ships in simulation is assumed to have the same parameters. The major parameters of example ship in numerical simulations are presented in Table 3.

In the process of simulation, own ship (OS) has the standard speed, which is equal to 15.5 knots (about 7.8 m/s). And the speed of dynamic obstacle is changing in different scenes and fixed in one scene. For the static obstacles, the radius is between 100 m and 500 m. However, in this paper, the perceptual accuracy of sensors is not considered. Once the obstacles enter the warning range, they will be detected and the information of obstacles, such as position, heading, speed, and so on, will always be available.

Table 3
Parameters of ship in numerical simulations.

Parameters	Values
Length (m)	175.00
Breadth (m)	25.40
Draught (m)	8.50
Tonnage (t)	21 222
Block coefficient	0.559
Rudder area (m^2)	33.038
Aspect ratio of rudder	1.822
Diameter of propeller (m)	6.533
Velocity (kn)	15.50

To fully describe the actual motion of ships in simulation, the mathematical model of six-degree-of-freedom motion is required, and the six degrees of freedom motions include surging, swaying, heaving, rolling, pitching and yawing. Usually, when ships are sailing on the horizontal plane, it is customary assuming that the ship is a rigid body. Then, the pitching, heaving and rolling degrees of motion are neglected. Therefore, the mathematical model of ship motion is simplified to three degrees of freedom, which only include surging, swaying, and yawing. To make it easier to represent the three degrees of freedom motion of ships, the global and inertial coordinate systems are established (see Fig. 17). In the Fig. 17, $O_g - X_g Y_g$ is the global coordinate systems,

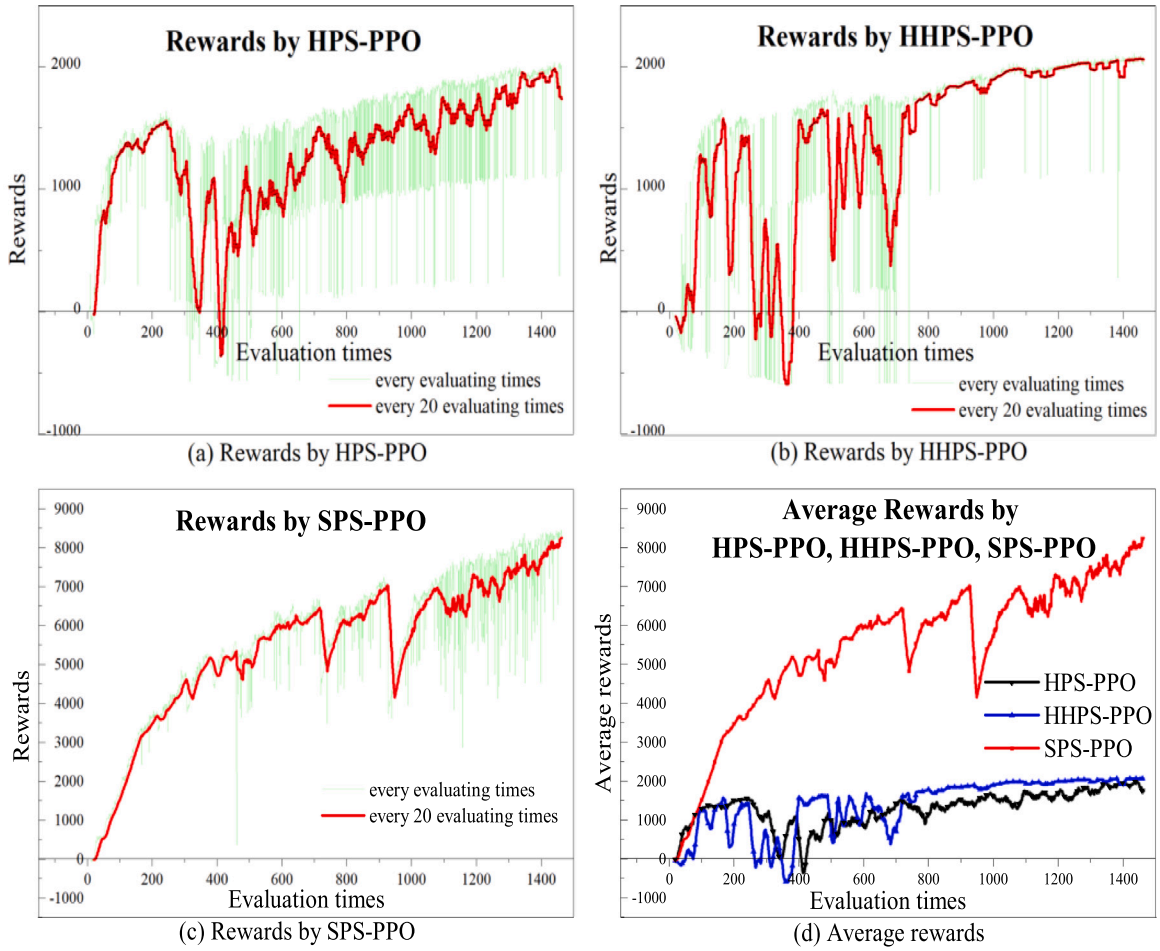


Fig. 14. Rewards obtained in Half Cheetah-v2.

which is fixed, and it is also called north-east coordinate system. $o_i{-}x_i y_i$ is the inertial coordinate system and it is attached to the ship. ψ represents the heading angle of ship.

In the inertial coordinate system, the motion of ship is expressed in followed equation (Jia and Yang, 1999).

$$\begin{aligned} m(\dot{u} - vr) &= X_I + X_H + X_P + X_R \\ m(\dot{v} + ur) &= Y_I + Y_H + Y_P + Y_R \\ I_{zz}\dot{r} &= N_I + N_H + N_P + N_R - Y_H \cdot x_C \end{aligned} \tag{15}$$

In this equation, the origin of inertial coordinate system is located on the center of gravity. Here, m is the mass of ship. x_C represents the coordinate value of ship's center in x-direction. I_{zz} is the inertia moment in z-direction pointing to ship's base line. u, v, r denote the velocity of ship in the x_i, y_i, and yaw directions, respectively. $\dot{u}, \dot{v}, \dot{r}$ are the accelerations in corresponding directions. X, Y, N represent the external force in x-direction and y-direction and yawing moment, respectively. The subscripts I, H, P, R denote the different type of external force and yawing moment, which represent fluid inertia force of ship hull, fluid viscous force of ship hull, hydrodynamic force of propeller, and hydrodynamic force of rudder. The specific calculation methods of external forces and yawing moment can be found in Jia and Yang (1999) and Zhang and Zhang (2020). The transformation method of ship's speed between inertial coordinate system and global coordinate system is expressed as Eq. (16)

$$\begin{aligned} \dot{x}_g &= \cos(\psi) \cdot u - \sin(\psi) \cdot v \\ \dot{y}_g &= \sin(\psi) \cdot u + \cos(\psi) \cdot v \\ r_g &= r \end{aligned} \tag{16}$$

Here, $\dot{x}_g, \dot{y}_g$ represent the global speed of ship in north direction and east direction, respectively. r_g is the yaw speed of ship in global coordinate system.

In COLREGs, obstacle encounter scenarios include head on situation, crossing situation, overtaking situation. In addition, in this paper, single-static obstacle encounter scenarios and multi-obstacles scenario are also considered. Fig. 18 describes the rewards obtained by SPS-DRL model during the training process with respect to the collision avoidance scenarios. In this simulation process, the hyperparameters of PPO algorithm is shown in Table 2. But, the size of experience pool is modified to 5120. Then, the max evaluation time is about 585. As depicted in Fig. 18, the reward is less than 0 at the beginning and gradually increases, which means the collision avoidance performance of own ship improves during the learning process. Finally, the reward converged near 2400 without any significant change.

5.2.1. Scenario 1: head on

In the head-on scenario, we defined two head-on encounter situations, including starboard-to-starboard and port-to-port. In the encounter situation, only one obstacle ship (OBS) was approaching with a ship speed of 15 m/s. Fig. 19 shows the simulation results for collision avoidance and the relative distance between OS and OBS during the head-on encounter scenario. The red dot line represents the simulation results of OS and the blue dot dash line represents the trajectory of OS with standing on. The yellow line represents the reference path for OS. In Section 4.1, the radius of warning range was about 10,000 m. Therefore, considering the uncertainties, the relative distance between OS and the initial position of OBS detected by OS was set randomly between 9000 and 10,000. Meanwhile, when OBS was out of warning

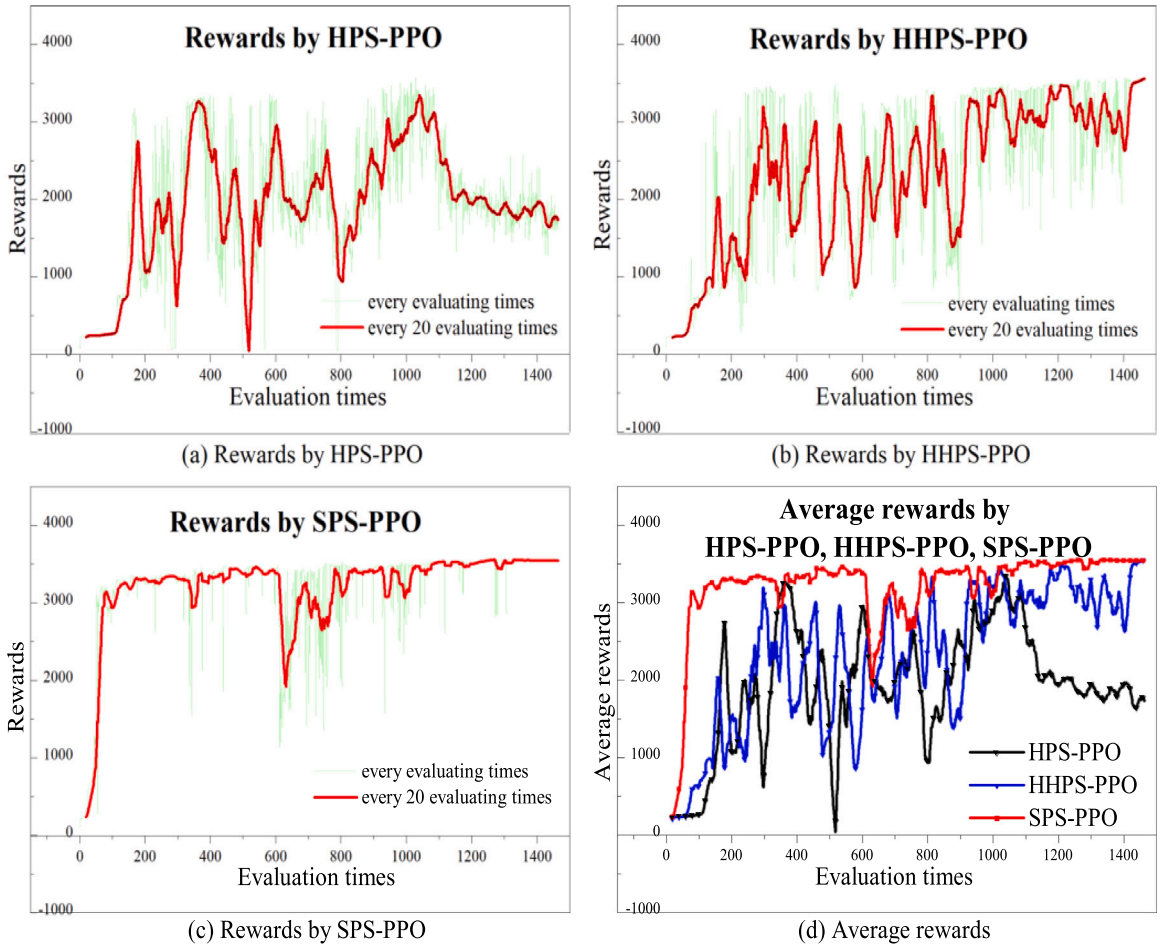


Fig. 15. Rewards obtained in Hopper-v2.

range, the information of it would not be recorded. In Fig. 19, the initial position of OBSs located at (323, 9255) in starboard-to-starboard situation and (−325, 9332) in port-to-port situation.

In the head-on condition, as OBS enters the warning range of OS, the kinematic information of OS and OBS are used to evaluate the collision risk. Once there is a risk of collision, a starboard side avoidance maneuver is taken by OS. The heading of OS had changed to about 30 degrees starboard at approximately 170 s in Fig. 19 (a) and (b). When the collision risk is reduced or the OBS has been past and clear, the OS should return to original course or reference path. In Fig. 19 (a) and 19 (c), at approximately 375 s, the collision risk is reduced, OS began to turn back to the original course. After the relative distance had gradually increased, OS started returning to the reference path, and had backed to the reference path at approximately 1400 s. In Fig. 19 (b) and 19 (d), at approximately 330 s, the collision risk had been reduced, but the relative distance was still decreasing. Therefore, to balance the efficiency of navigation and risk, OS turned the heading to original course and sailed parallel to the reference path. After a period of time, OS gradually got close to the reference path, and had backed to the reference path at approximately 1500 s. As shown in Fig. 19, the minimum distances between OS and OBS are 689 m and 1129 m in starboard-to-starboard situation and port-to-port situation, respectively, and they are both larger than ship domain and safe enough for the navigation of ships.

5.2.2. Scenario 2: crossing

In the crossing scenario, we also defined two crossing encounter situations, including crossing with small angle and crossing with large angle. Fig. 20 shows the simulation results for collision avoidance and the relative distance between OS and OBS during the crossing encounter scenario. The initial positions of OBS located at (2363, 9375) in small angle crossing situation and (4812,8660) in large angle crossing situation, and the intersection angle between initial course of OS and OBS was 33 degrees in small angle crossing situation and 60 degrees in large angle crossing situation. In addition, to ensure the existence of collision risk, the speeds of OBS were different in different encounter situation. Here, the speeds of OBS were 6 m/s and 7.6 m/s in small angle crossing situation and large angle crossing situation, respectively.

In crossing encounter condition, a starboard side avoidance maneuver should also be taken by OS when there is a risk of collision. As shown in Fig. 20, The collision avoidance process is about from 240 s to 610 s and from 520 s to 769 s in crossing situation with small angle and crossing situation with large angle, respectively. In crossing situation with small angle, there are two starboard side avoidance maneuver. The first avoidance maneuver is about 10 degrees. After that, the collision risk is still exist in the future. Then, the further starboard side maneuver happened. At approximately 610s, collision risk reduced and OS started to turn to original course. OS had backed to the reference path at approximately 1480 s. In crossing situation with large angle, to guarantee the navigation efficiency, OS followed the reference path from 0 s to 520 s. With the collision risk increasing, the avoidance maneuver happened at approximately 520 s. At approximately 769 s, OS was past and clear and started to back to reference path. Finally, OS had finished the collision avoidance maneuver at approximately 1589 s. In Fig. 20(c) and (d), it can be seen that the minimum relative distances between OS and OBS are 1061 m and 1311 m during entire crossing encounter scenario, and they are also both safe enough for ship navigation.

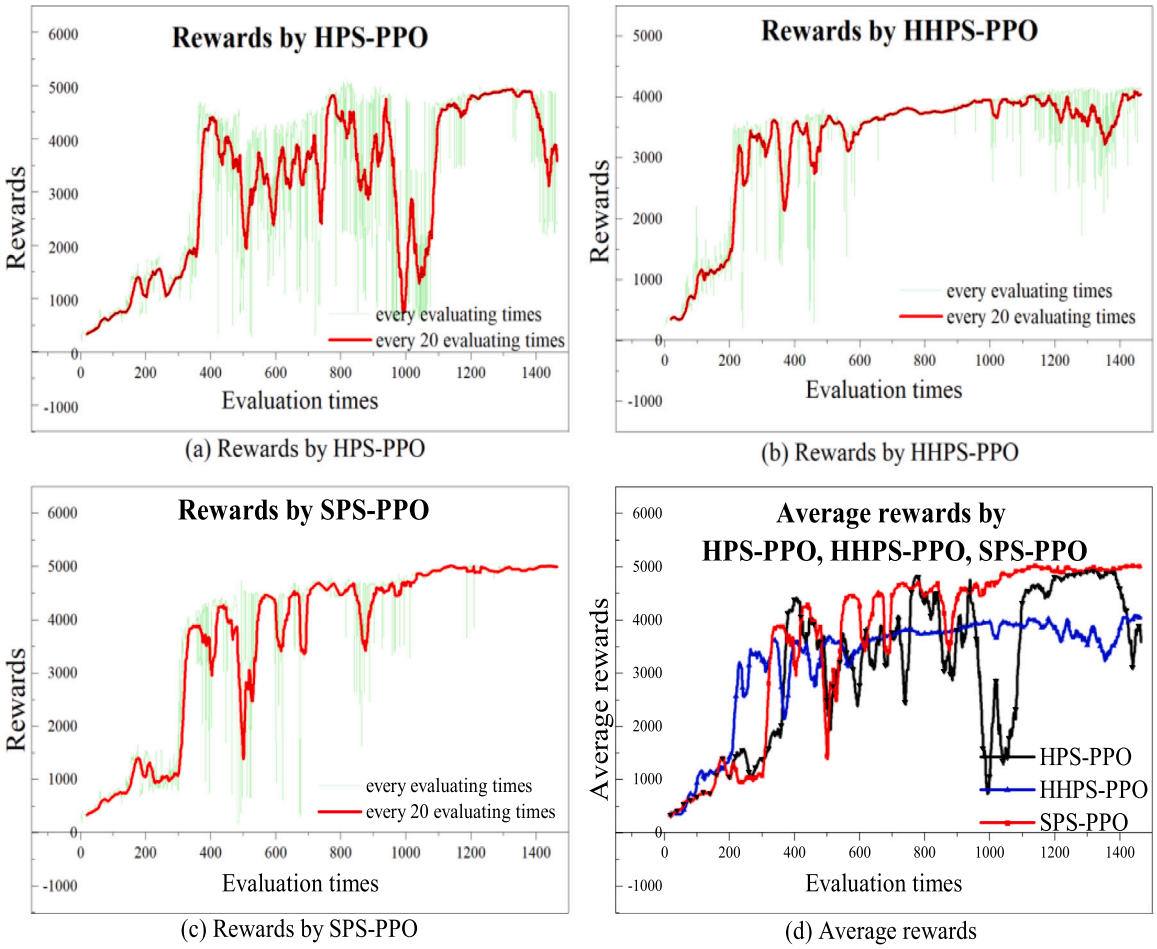


Fig. 16. Rewards obtained in Walker2d-v2.

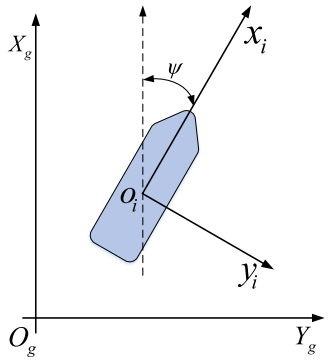


Fig. 17. Global and inertial coordinate systems.

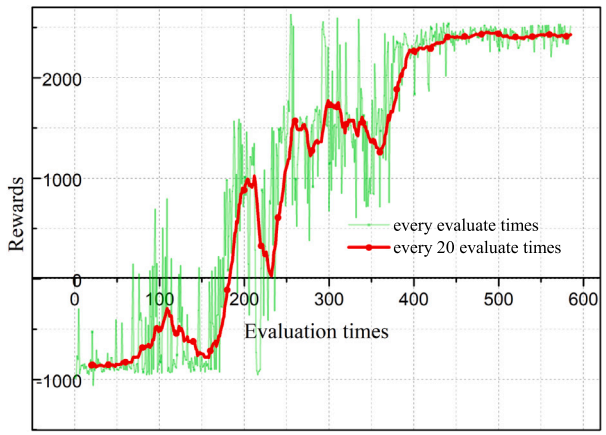
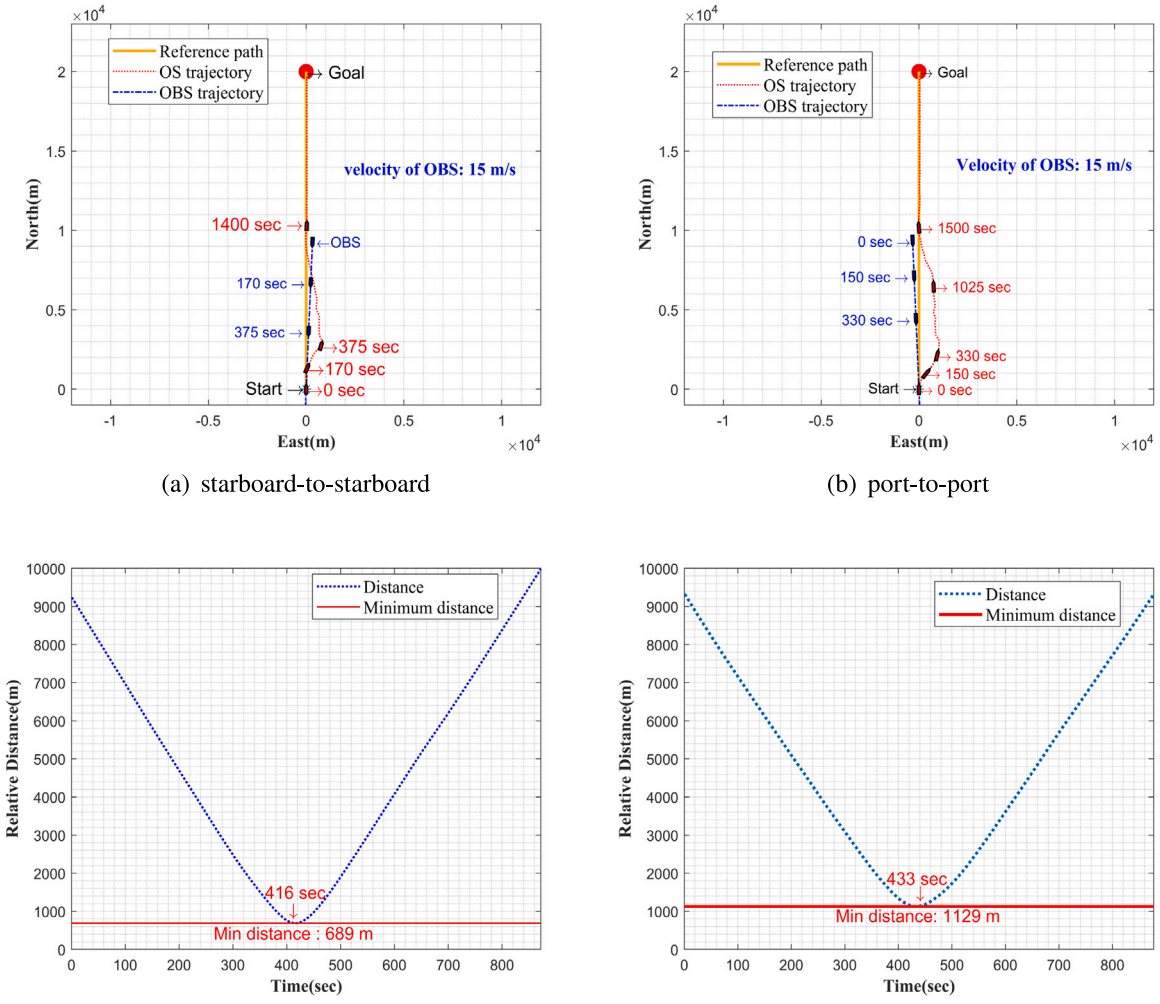


Fig. 18. Rewards according to the evaluation.

5.2.3. Scenario 3: overtaking and single static obstacle

In this section, we defined the overtaking scenario and single static obstacle scenario. Fig. 21 shows the simulation results for collision avoidance during the overtaking and static obstacle encounter scenario. Fig. 21(c) shows the relative distance between OS and OBS during the overtaking encounter scenario and Fig. 21(d) shows the relative distance between OS and static obstacle. In overtaking situation, the initial position of OBS located at (58, 9648) and the heading was the same as OS. To make sure that OS could overtake the OBS quickly, the velocity of OBS was set to 1.5 m/s. In the static obstacle scenario, obstacle had been simplified to a circle with the center located at (−36, 9536) and the radius of obstacle is 572 m.

In overtaking encounter scenario, there are also two stages of collision avoidance maneuver. In the first stage, OS took a starboard side avoidance maneuver to balance the efficiency of navigation and collision risk and declared the intention of collision avoidance. After that, OS began to maintain the navigation with initial course. As the distance between the two ships approaches, the risk of collision gradually increased. At approximately 1150 s, a second starboard side avoidance maneuver was taken to keep a safe distance between OS and OBS. As gradually approaching the target point, OS began to sail



(a) starboard-to-starboard

(b) port-to-port

(c) Relative distance in starboard-to-starboard situation

(d) Relative distance in port-to-port situation

Fig. 19. Collision avoidance for head-on encounter scenario.

towards the goal at approximately 2266 s. For the static obstacle, the collision avoidance of OS was similar with the overtaking scenario except that OS would turn to the reference path immediately once the obstacle was past and clear in static obstacle scenario. However, in overtaking encounter situation, OS would keep parallel to the reference path after OBS was past and clear because the potential risk of collision still existed. As shown in Fig. 21 (c) and (d), the minimum relative distances are also safe enough for ship navigation.

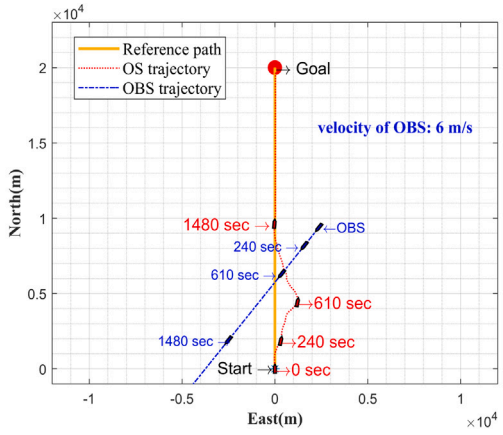
5.2.4. Scenario 4: multiple obstacles

In this part, we defined four multiple obstacles scenarios, including two multi-static-obstacles scenarios and two multi-dynamic-obstacles scenarios. In multi-static-obstacles scenario, we defined five static obstacles, and the information of static obstacles was shown in Table 4. Fig. 22 shows the simulation results for collision avoidance during the multi-static-obstacles encounter scenario. There are four static obstacles around reference path and one static obstacle far from it but still in the warning range of OS. Comparing static scenario 1 and static scenario 2, there are differences in the position and radius of obstacles.

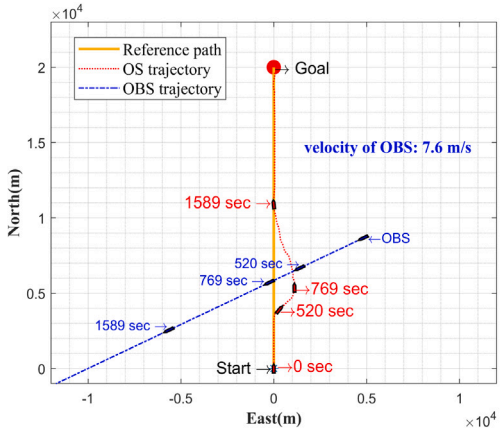
As shown in Fig. 22, in the multi-static-obstacles encounter scenario, OS avoided the static obstacles successfully and arrived target point. At the initial moment, static obstacle 1 and 5 were observed by OS, then OS took a starboard side maneuver with small angle. As the risk of collision decreased, OS gradually turned to the original course. At approximately 911 s, OS had a further starboard side maneuver to avoid static obstacle 2 and 3. After avoiding static obstacle 3, OS would back to the reference path, but the static obstacle 4 made OS keep a straight line. Comparing the static scenario 1, obstacles in static scenario 2 moved westward. Therefore, the avoidance angle of OS in static scenario 2 was slightly small than that in static scenario 1 and the time for OS to maintain original course in static scenario 2 was greater than that in static scenario 1. Fig. 22 (c) and (d) shows the relative distance between OS and obstacles and Table 5 provides the minimum relative distance between OS and static obstacles during the collision avoidance and the time for OS to arrive it. In Fig. 22 (c) and (d), it can be seen that the min-distances between OS and the periphery of static obstacles are 590.18 m in static scenario 1 and 643.30 m in static scenario 2, which are safe enough to navigate.

In multi-dynamic-obstacles scenario, we defined three dynamic obstacles (OBS 1, OBS 2, OBS 3), and the initial information of dynamic obstacles was shown in Table 6. Fig. 23 shows the simulation results for collision avoidance during the multi-dynamic-obstacles encounter scenario. In Fig. 23, OS forms an overtaking situation with OBS 1 and a crossing situation with OBS 2 and OBS 3. The trajectory of OBS 1 was right relative to reference path in dynamic scenario 1 and it was left relative to reference path in dynamic scenario 2. The headings of OBS 2 and OBS 3 were different in dynamic scenario 1 and dynamic scenario 2, and the specific values of heading were shown in Table 6.

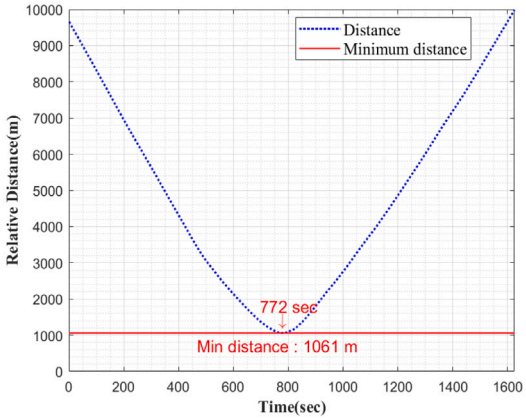
As shown in Fig. 23, in the multi-dynamic-obstacles encounter scenario, OS could also avoid the dynamic obstacles successfully and



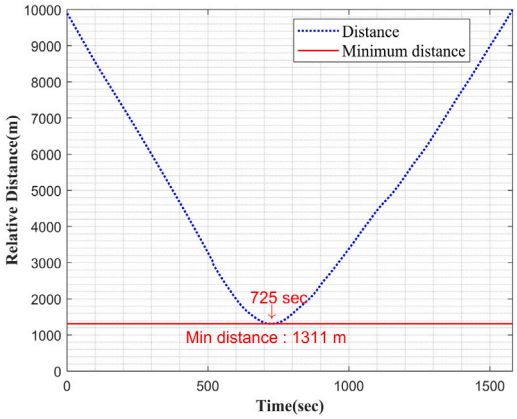
(a) Crossing with small angle



(b) Crossing with large angle



(c) Relative distance in Crossing situation with small angle



(d) Relative distance in crossing situation with large angle

Fig. 20. Collision avoidance for crossing encounter scenario.

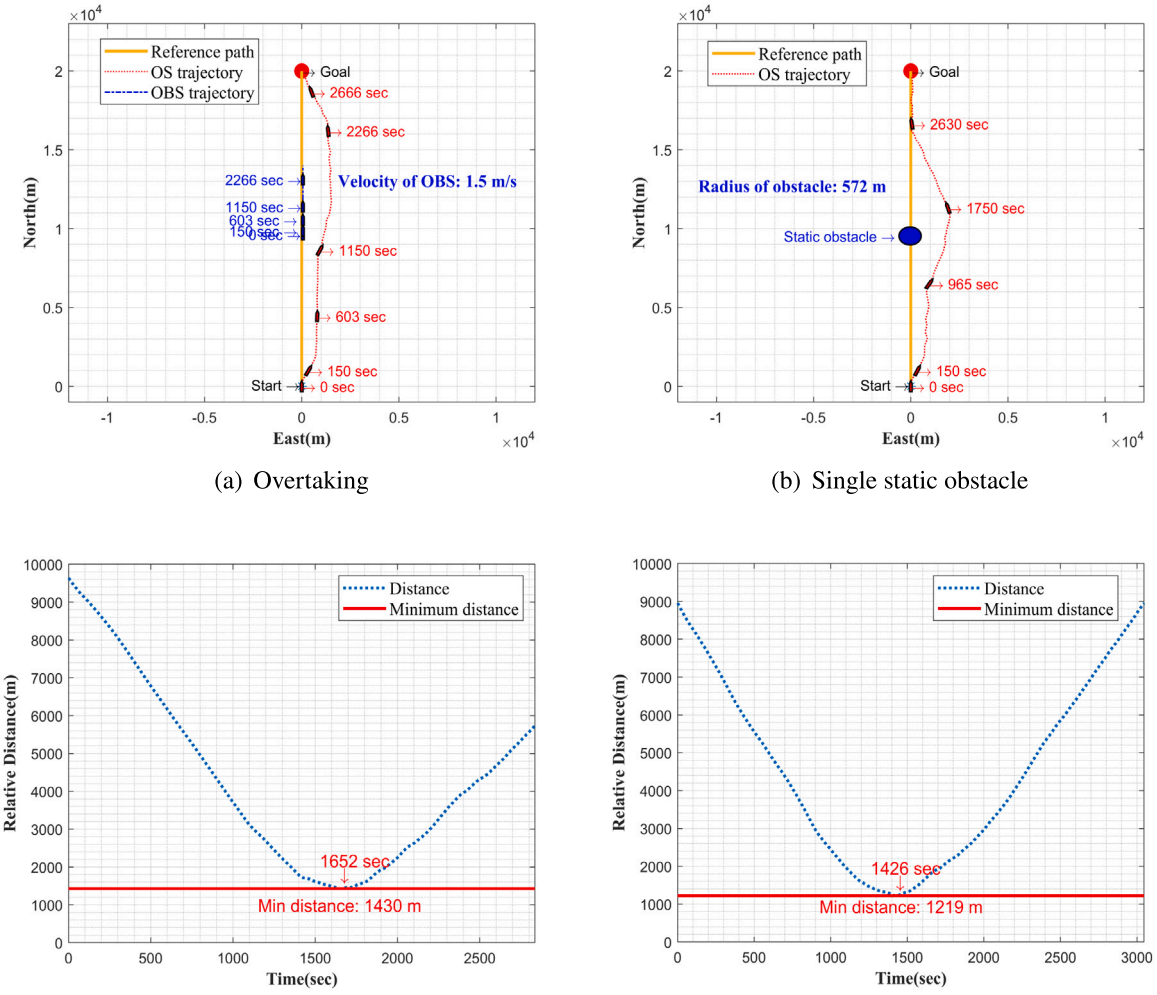
Table 4
Comparison of the total number of parameters in different environments and models.

Scenario	Obstacles	Position (east, north)	Radius (m)
Static scenario 1	Static obstacle 1	(−956, 9848)	464
	Static obstacle 2	(43, 10848)	464
	Static obstacle 3	(1043, 11848)	464
	Static obstacle 4	(43, 14848)	464
	Static obstacle 5	(−2956, 4848)	464
Static scenario 2	Static obstacle 1	(−1276, 9411)	594
	Static obstacle 2	(−276, 10411)	594
	Static obstacle 3	(723, 11411)	594
	Static obstacle 4	(−276, 14411)	594
	Static obstacle 5	(−3276, 4411)	594

Table 5
Comparison of the total number of parameters in different environments and models.

Scenario	Obstacles	Min-distance (m)	Time (s)
Static scenario 1	Static obstacle 1	1964.23	1122
	Static obstacle 2	1350.45	1410
	Static obstacle 3	**590.18**	**1541**
	Static obstacle 4	1480.32	1986
	Static obstacle 5	3086.29	523
Static scenario 2	Static obstacle 1	2022.49	1103
	Static obstacle 2	1269.28	1317
	Static obstacle 3	**643.30**	**1399**
	Static obstacle 4	1695.36	1947
	Static obstacle 5	3354.10	565

arrive target point. At the initial moment, OS should take a starboard side maneuver to avoid OBS 2 and stand on relative to OBS 3 in crossing situation and take maneuver to avoid OBS 1 in overtaking situation. At approximately 221 s in dynamic scenario 1 and 330 s in dynamic scenario 2, OS turned to the right. After a period of time, the collision risk gradually decreased and OS would turn to the original course, and the times this process occurred in dynamic scenario 1 and dynamic scenario 2 were from approximately 575s to 795 s and from 571 s to 850 s, respectively. Meanwhile, in this process, OS formed an encounter situation with OBS 3. But, there was no risk of collision as OS maintained the original course. After OBS 2 and OBS 3 sailed past and clear, OS begun to avoid OBS 1. In dynamic scenario 1, the position of OBS 1 was on the right of reference path, and OS taken a further starboard side maneuver to keep safe distance from OBS 1 at approximately 1150 s. As the goal approached, at approximately 2120 s, OS changed the course and sailed towards the goal. In dynamic scenario 2, the position of OBS 1 was on the left of reference path, and there was enough distance between OS and OBS 1. Therefore, from 1190 s to 1721 s, OS almost sailed with original course. Form 1721 s to 2040 s, OS was approaching the reference path. However, because of the potential risk between OS and OBS 1, OS would keep a certain lateral distance with OBS 1. Fig. 23 (c) and (d) shows the relative distance between OS and OBSs and Table 7 provides the minimum relative distance between OS and OBSs during the entire collision avoidance process and the time for OS to arrive it. As shown in Fig. 23 (c) and (d), in dynamic scenario 1 and dynamic scenario 2, the minimum relative distances were equal to 996.03 m and 859.14 m respectively, which both happened between OS and OBS 2, and OS arrived it at 601 s and 859 s.



(a) Overtaking (b) Single static obstacle

(c) Relative distance in overtaking situation (d) Relative distance in static obstacle situation

Fig. 21. Collision avoidance for overtaking scenario and static obstacle.

Table 6
Comparison of the total number of parameters in different environments and models.

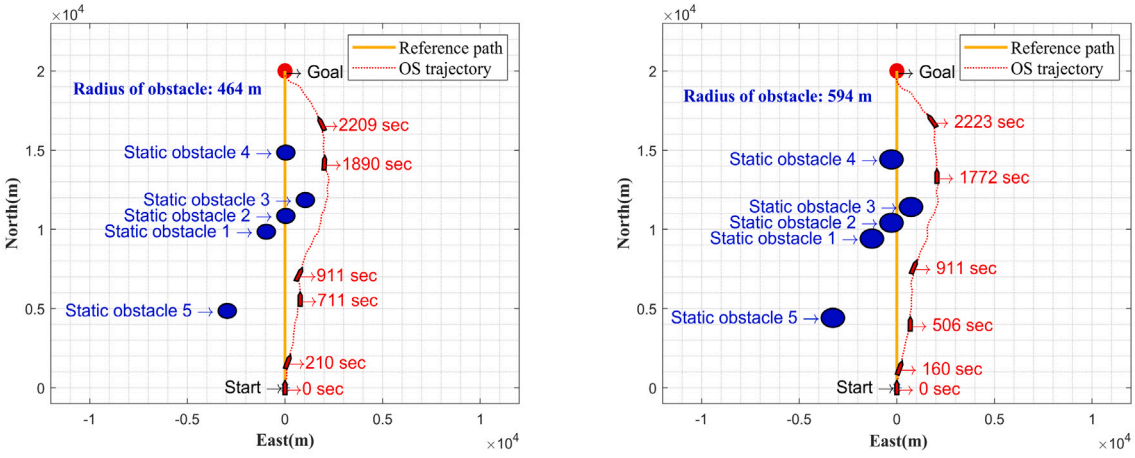
Scenario	Obstacles	Initial position (east, north)	Velocity (m/s)	Heading (°)
Static scenario 1	OBS 1	(276, 9304)	1.7	0
	OBS 2	(5534, 7920)	10.0	240
	OBS 3	(−5534, 10920)	10.0	118
Static scenario 2	OBS 1	(−431, 9708)	1.8	0
	OBS 2	(6382, 6683)	11.0	253
	OBS 3	(−6382, 9683)	11.0	106

Table 7
Comparison of the total number of parameters in different environments and models.

Scenario	Obstacles	Min-distance (m)	Time (s)
Dynamic scenario 1	OBS 1	1493.12	1674
	OBS 2	**996.03**	**601**
	OBS 3	1255.23	869
Static scenario 2	OBS 1	1197.43	1103
	OBS 2	**859.14**	**569**
	OBS 3	1748.46	809

5.3. Discussion

In this part, we discuss on the simulation results and the performance of the proposed methods. Firstly, the self-adaptive parameters sharing mechanism is verified in four Gym environment. It can be seen from Figs. 13 to 16 that the performance of the DRL algorithm with self-adaptive parameters sharing mechanism is better than other sharing mechanism. In section Section 5.1, the SPS-PPO algorithms not only obtain higher total rewards, but also has faster convergence speed in four Gym environment. Secondly, the ship collision avoidance system based on SPS-DRL and collision map is applied in single-obstacle encounter scenarios and multi-obstacles encounter scenarios Section 5.2. In the single obstacle simulation, four encounter situations are tested, including head on situation, crossing situation, overtaking situation and single-static-obstacle situation. In the four situations, ship can avoid the obstacle successfully and reach the destination. Otherwise, the minimum distance between OS and obstacle is 1129 m during the collision avoidance in the four situations, and it is safe enough for ship navigation. For the dynamic situation, especially in head on situation and crossing situation, the COLREGs should be obeyed. As it can be seen from Figs. 19 and 20, the ship follows the rules and avoids collision



(a) Static scenario 1

(b) Static scenario 2

(c) Relative distance in Static scenario 1

(d) Relative distance in Static scenario 2

Fig. 22. Collision avoidance for multiple static obstacles.

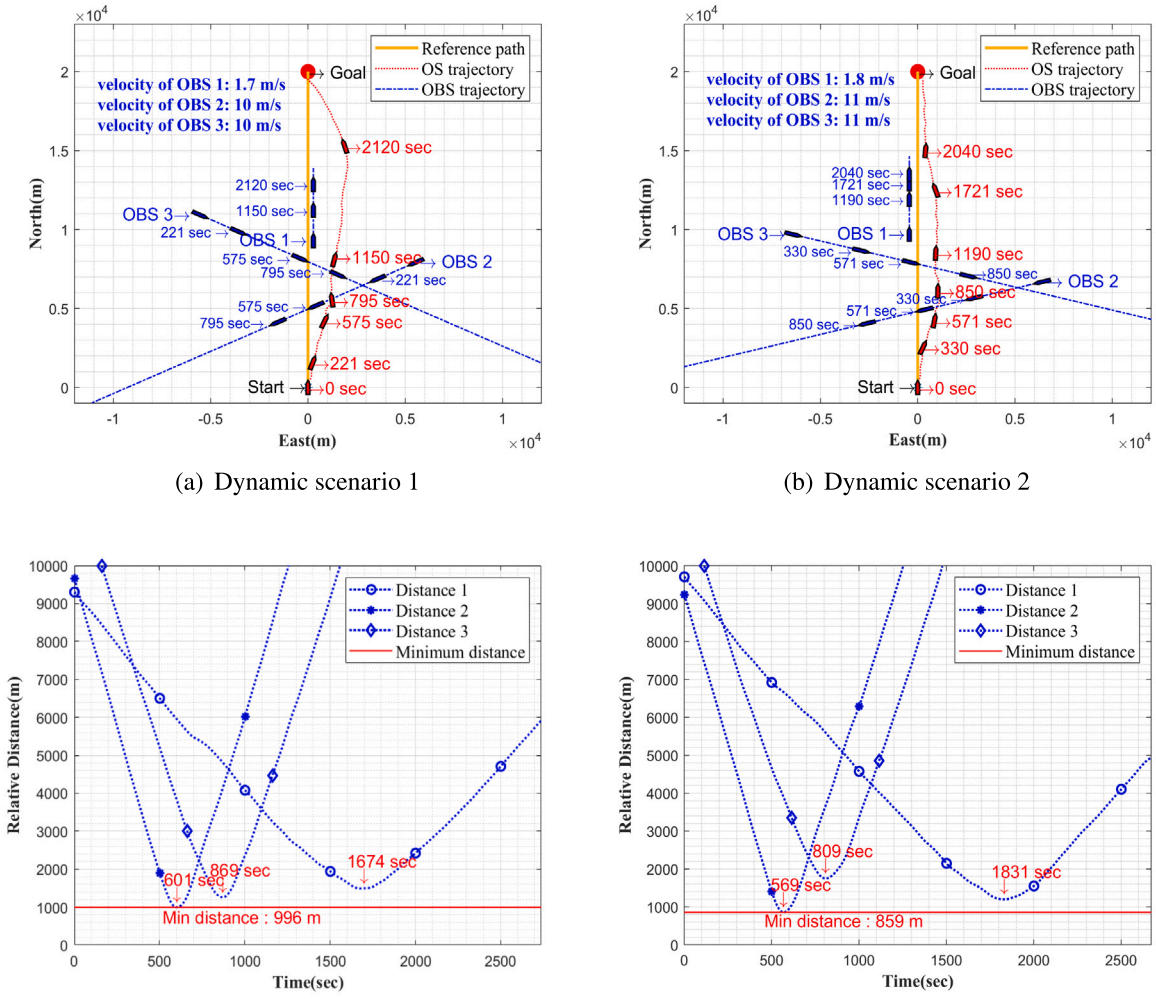
from starboard side. In addition, in the multi-obstacles simulation, the collision map can deal with the scenario where the number of obstacles is not fixed, and the ship can also avoid obstacles successfully and arrive at the target. And minimum distance between OS and obstacle (590 m) is also safe enough for ships. In conclusion, the proposed method can effectively cope with complex dynamic encounter scenarios for ship navigation.

6. Conclusion

In this paper, a parameter-sharing DRL based collision avoidance method was proposed. In the proposed method, a self-adaptive sharing network was designed to improve the convergence rate of model and applied in DRL model. The design of collision map was a crucial part of the proposed method, which was used to describe the nearby environment of OS and as the input of DRL model. The proposed collision map makes DRL model more suitable for the complex ship encounter scenarios. To guide the ship to navigate safely and comply with COLREGs in the process of collision avoidance, new reward functions are designed for the training of DRL network. After that, ship can deal with the complicated encounter scenarios by proposed method. As a result of the above design, this paper makes the following main conclusions .

- The proposed self-adaptive sharing network can accelerate the convergence and improve the performance. In four gym environments, the DRL algorithm with self-adaptive network receives higher rewards and needs less convergence time.
- The collision map can represent complex and dynamic situations with multiple obstacles. It can describe not only dynamic features of ships, but also regional features, such as ship domain.
- After guided by the designed reward functions, the proposed method can select suitable avoidance strategy, complying with COLREGs and ensuring the safety.
- According to the collision avoidance experiments under various scenarios, the effectiveness of the proposed method was verified.

It is noted that there are still some limitations of the proposed collision avoidance system. First, the collision map can represent the ship encountered situation well, but it makes DRL perform on a high dimensional state, which may increase the training time of DRL model. Second, the result of collision avoidance may be affected by the marine environment during the actual voyage. Therefore, it will be better to consider the marine environment should be considered in collision avoidance simulation system. Finally, the DRL model is trained and tested in the simulation environment in this paper, it can be improved with performance verification in the real environment. In the future research, the effects of marine environment, including wind, wave, and current will be taken into account in the collision avoidance decision-making. In addition, the performance of proposed DRL-based collision avoidance method will be further verified and validated by indoor model experiments and field tests.



(a) Dynamic scenario 1

(b) Dynamic scenario 2

(c) Relative distance in Dynamic scenario 1

(d) Relative distance in Dynamic scenario 2

Fig. 23. Collision avoidance for multiple Dynamic obstacles.

CRediT authorship contribution statement

Yong Wang: Conceptualization, Formal analysis, Methodology, Software, Writing – original draft, Writing – review & editing. **Haixiang Xu:** Conceptualization, Funding acquisition, Project administration, Supervision. **Hui Feng:** Conceptualization, Resources, Supervision, Writing – review & editing. **Jianhua He:** Formal analysis, Supervision, Validation, Writing – review & editing. **Haojie Yang:** Investigation, Visualization. **Fen Li:** Investigation, Visualization, Writing – review & editing. **Zhen Yang:** Conceptualization, Investigation, Software.

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

Data will be made available on request.

Acknowledgments

The authors appreciate the constructive suggestions from reviewers and the Associate Editor. This work is supported by the Key Research and Development Program of Guangdong Province, China (No. 2020B1111500002) and National Natural Science Foundation of China under Grant No. 51979210, 52371374. This work was partly funded by EPSRC with RC Grant reference EP/Y027787/1, UKRI under grant number EP/Y028317/1, the Horizon Europe MSCA programme under grant agreement No 101086228.

References

Aiello, G., Giallanza, A., Mascarella, G., 2020. Towards shipping 4.0 A preliminary gap analysis. Proc. Manuf. 42, 24–29.

Brcko, T., Androjna, A., Srše, J., Boć, R., 2021. Vessel multi-parametric collision avoidance decision model: Fuzzy approach(article). J. Mar. Sci. Eng. 1–24.

Chen, C., Chen, X., Ma, F., Zhao, X., Wang, J., 2019. A knowledge-free path planning approach for smart ships based on reinforcement learning. Ocean Eng. 189, 106299.

Chen, P., Huang, Y., Mou, J., van Gelder, P., 2018. Ship collision candidate detection method: A velocity obstacle approach. Ocean Eng. 186–198.

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

Cobbe, K.W., Hilton, J., Klimov, O., Schulman, J., 2021. Phasic policy gradient. In: Meila, M., Zhang, T. (Eds.), Proceedings of the 38th International Conference on Machine Learning. In: Proceedings of Machine Learning Research, Vol. 139, PMLR, pp. 2020–2027, URL https://proceedings.mlr.press/v139/cobbe21a.html.

Coldwell, T.G., 1983. Marine traffic behaviour in restricted waters. J. Navig. 36 (3), 430–444. http://dx.doi.org/10.1017/S0373463300039783.

D'Eramo, C., Tateo, D., Bonarini, A., Restelli, M., Peters, J., 2020. Sharing knowledge in multi-task deep reinforcement learning. In: International Conference on Learning Representations. URL https://openreview.net/forum?id=rkgpv2VFvr.

Dong, X.-M., Kong, X., Zhang, X., 2022. Multi-task learning based on stochastic configuration neural networks. Front. Bioeng. Biotechnol. 10, http://dx.doi.org/10.3389/fbioe.2022.890132, URL https://www.frontiersin.org/articles/10.3389/fbioe.2022.890132.

Du, L., Banda, O.A.V., Huang, Y., Goerlandt, F., Kujala, P., Zhang, W., 2021a. An empirical ship domain based on evasive maneuver and perceived collision risk. Reliab. Eng. Syst. Saf. 107752.

Du, Y., Zhang, X., Cao, Z., Wang, S., Liang, J., Zhang, F., Tang, J., 2021b. An optimized path planning method for coastal ships based on improved DDPG and DP. J. Adv. Transp. 2021, 7765130.

EMSA, 2021. Annual overview of marine casualties and incidents 2021.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., Madry, A., 2020. Implementation matters in deep policy gradients: A case study on ppo and trpo. arXiv preprint arXiv:2005.12729.

Fışkın, R., Atik, O., Kişi, H., Nasibov, E., Johansen, T.A., 2021. Fuzzy domain and meta-heuristic algorithm-based collision avoidance control for ships: experimental validation in virtual and real environment(article). Ocean Eng. 108502.

Fiskin, R., Nasiboglu, E., Yardimci, M.O., 2020. A knowledge-based framework for two-dimensional (2D) asymmetrical polygonal ship domain. Ocean Eng. 107187.

Goodwin, E.M., 1975. A statistical study of ship domains. J. Navig. 28 (3), 328–344. http://dx.doi.org/10.1017/S0373463300041230.

Jia, X., Yang, Y., 1999. Mathematical Model of Ship Motion - Mechanism Modeling and Identification Modeling. Dalian Maritime University Press.

Joung, T.-H., Kang, S.-G., Lee, J.-K., Ahn, J., 2020. The IMO initial strategy for reducing Greenhouse Gas (GHG) emissions, and its follow-up actions towards 2050. J. Int. Marit. Saf., Environ. Affairs Shipping 4 (1), 1–7.

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

Lee, M.-C., Nieh, C.-Y., Kuo, H.-C., Huang, J.-C., 2019. An automatic collision avoidance and route generating algorithm for ships based on field model. J. Mar. Sci. Technol. 27 (2), 101–113.

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.

Meyer, E., Heiberg, A., Rasheed, A., San, O., 2020. COLREG-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning. IEEE Access 8, 165344–165364. http://dx.doi.org/10.1109/ACCESS.2020.3022600.

Naeem, W., Henrique, S.C., Hu, L., 2016. A reactive COLREGs-compliant navigation strategy for autonomous maritime navigation. IFAC-PapersOnLine 49 (23), 207–213.

Niu, H., Ji, Z., Savvaris, A., Tsourdos, A., 2020. Energy efficient path planning for unmanned surface vehicle in spatially-temporally variant environment. Ocean Eng. 196, 106766.

Rawson, A., Brito, M.P., 2020. A critique of the use of domain analysis for spatial collision risk assessment. Ocean Eng. 108259.

Richards, C., Odom, C., Morton, D., Newman, A., 2019. Minimum-risk routing through a mapped minefield. Networks 73 (3), 358–376.

Rongcai, Z., Hongwei, X., Kexin, Y., 2023. Autonomous collision avoidance system in a multi-ship environment based on proximal policy optimization method. Ocean Eng. 113779.

Schiewer, R., Wiskott, L., 2021. Modular networks prevent catastrophic interference in model-based multi-task reinforcement learning. In: International Conference on Machine Learning, Optimization, and Data Science. Springer, pp. 299–313.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. pp. 1–12. http://dx.doi.org/10.48550/arXiv.1707.06347.

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.

Sun, T., Shao, Y., Li, X., Liu, P., Yan, H., Qiu, X., Huang, X., 2020. Learning sparse sharing architectures for multiple tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34, (05), pp. 8936–8943.

Tam, C., Bucknall, R., 2010. Collision risk assessment for ships. J. Mar. Sci. Technol. 15, 257–270.

Teh, Y., Bapst, V., Czarnecki, W.M., Quan, J., Kirkpatrick, J., Hadsell, R., Heess, N., Pascanu, R., 2017. Distral: Robust multitask reinforcement learning. Adv. Neural Inf. Process. Syst. 30.

Thyri, E.H., Basso, E.A., Breivik, M., Pettersen, K.Y., Skjetne, R., Lekkas, A.M., 2020. Reactive collision avoidance for ASVs based on control barrier functions. In: 2020 IEEE Conference on Control Technology and Applications (CCTA).

Wang, H., Fu, Z., Zhou, J., Fu, M., Ruan, L., 2021. Cooperative collision avoidance for unmanned surface vehicles based on improved genetic algorithm. Ocean Eng. 222, 108612.

Wang, Y., Yanushkevich, S., Hou, M., Plataniotis, K., Coates, M., Gavrilova, M., Hu, Y., Karray, F., Leung, H., Mohammadi, A., et al., 2020. A tripartite theory of trustworthiness for autonomous systems. In: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, pp. 3375–3380.

Wang, T., Zhuang, F., Sun, Y., Zhang, X., Lin, L., Xia, F., He, L., He, Q., 2022. Adaptively sharing multi-levels of distributed representations in multi-task learning. Inform. Sci. 591, 226–234.

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

Wu, Q., Wang, T., Diaconeasa, M.A., Mosleh, A., Wang, Y., 2020. A comparative assessment of collision risk of manned and unmanned vessels. J. Mar. Sci. Eng. 8 (11), 852.

Xu, X., Cai, P., Ahmed, Z., Yellapu, V.S., Zhang, W., 2022a. Path planning and dynamic collision avoidance algorithm under COLREGs via deep reinforcement learning. Neurocomputing 468, 181–197.

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

Xue, H., 2022. A quasi-reflection based SC-PSO for ship path planning with grounding avoidance. Ocean Eng. 247, 110772.

Yan, X., Liu, J., Fan, A., Ma, F., Chen, L., 2020. Overview of the development and trends of intelligent ship technology. Ocean Eng. 42 (3), 15–20.

Yuan, X., Zhang, D., Zhang, J., Zhang, M., Guedes Soares, C., 2020. A novel real-time collision risk awareness method based on velocity obstacle considering uncertainties in ship dynamics. Ocean Eng. 108436.

Zhang, X., Wang, C., Jiang, L., An, L., Yang, R., 2021. Collision-avoidance navigation systems for maritime autonomous surface ships: A state of the art survey. Ocean Eng. 235, 109380.

Zhang, G., Zhang, X., 2020. Ship Motion Mathematical Model and MATLAB Simulation. China University of Mining and Technology Press.

Zhao, L., Roh, M.-I., 2019. COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. Ocean Eng. 191, 106436.

Zhou, C., Wang, Y., Wang, L., He, H., 2022. Obstacle avoidance strategy for an autonomous surface vessel based on modified deep deterministic policy gradient. Ocean Eng. 243, 110166.

Zhu, Z., Lyu, H., Zhang, J., Yin, Y., 2021. An efficient ship automatic collision avoidance method based on modified artificial potential field. J. Mar. Sci. Eng. 10 (1), 3.