Research article

INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
November-December 2020: 1–15

DOI: 10.1177/1729881420969081
journals.sagepub.com/home/arx
SAGE

Research on autonomous collision avoidance of merchant ship based on inverse reinforcement learning

Mao Zheng[1], Shuo Xie[2], Xiumin Chu[1], Tianquan Zhu[1,3] and Guohao Tian[1,3]

Abstract

To learn the optimal collision avoidance policy of merchant ships controlled by human experts, a finite-state Markov decision process model for ship collision avoidance is proposed based on the analysis of collision avoidance mechanism, and an inverse reinforcement learning (IRL) method based on cross entropy and projection is proposed to obtain the optimal policy from expert's demonstrations. Collision avoidance simulations in different ship encounters are conducted and the results show that the policy obtained by the proposed IRL has a good inversion effect on two kinds of human experts, which indicate that the proposed method can effectively learn the policy of human experts for ship collision avoidance.

Keywords

Inverse reinforcement learning, collision avoidance, cross entropy, projection, merchant ship

Date received: 14 June 2019; accepted: 5 October 2020

Topic Area: AI in Robotics; Human Robot/Machining Interaction
Topic Editor: Andrey Savkin
Associate Editor: Andrey Savkin

Introduction

In robotics, kinds of collision avoidance techniques have been widely tested in fields, such as smart cars, military robots, entertainment, and service robots, in different environments. These collision avoidance methods are quite specific to individual scenarios. Various collision avoidance methods could be broadly classified into two categories, that is, classical and reactive methods.[1] Early scholars mainly focused on classical methods, such as artificial potential field (APF),[2] cell decomposition,[3] roadmap planner, A* algorithm,[4,5] and so on. The major shortcoming of these classical methods is high computational costs and failure to respond to the uncertainty present in the environment, leading to changing control instructions. In recent years, reactive methods have been accepted as the most popular tool for unmanned vehicle collision avoidance, including Q-learning,[6] artificial neural network, genetic algorithm (GA),[7] particle swarm optimization (PSO), ant colony,[8] and some other evolutionary optimization algorithms,[9] even model predictive control (MPC).[10] Especially for moving obstacles and multiple vehicles, MPC and sliding-mode control could achieve better robustness to disturbance. As reactive methods could deal with

[1] National Engineering Research Center for Water Transportation Safety, Wuhan University of Technology, Wuhan, Hubei Province, People's Republic of China
[2] China Classification Society, Beijing, People's Republic of China
[3] School of Energy and Power Engineering, Wuhan University of Technology, Wuhan, Hubei Province, People's Republic of China

Corresponding author:
Shuo Xie, China Classification Society, Beijing 100007, People's Republic of China.
Email: xieshuo@whut.edu.cn

uncertainty present in dynamic environment much better than classical methods, most of the existing approaches for ship collision avoidance belong to reactive methods.[11]

With the rapid development of intelligent ships, the collision avoidance at sea becomes more and more prominent. Scholars have carried out a lot of research on collision avoidance[12,13] of unmanned surface vehicles (USVs) in recent years, achieving good collision avoidance performance in relatively simple static environments. However, as the kinematics of merchant ships are so different from USVs of which the sizes are small, the collision avoidance law for merchant ships is much more complex obviously.

To gain a proper collision avoidance action for a merchant ship in a specified encounter environment, the obvious solution is to establish the state-action mapping relations. For a single ship in dynamic environments, the state action corresponding to Q-value tables for simple discrete state-action decision problems, such as optimal policy search and path planning, have been put forward. Based on the ship kinematics and Q-learning, Yoo and Kim[14] conducted an automatic ship autopilot control program from start points to end points, among static obstacles, taking the currents into consideration. Chen et al.[15] treated the discretized ship rudder angle as a Q-learning action, corresponding to the ship's position states with grid map, and verified the effectiveness of the Q-learning for collision avoidance path planning. Zheng et al.[16] established a Markov decision process (MDP) discrete state strategy optimization method based on multiweight apprentice learning, achieving the scheduling policies, which perform close to experts' experience. Heuristic optimization-based algorithms include GA,[17] and PSO,[18] which have clear and simple structures, being widely used in collision avoidance for intelligent unmanned vehicles. These algorithms usually search the collision-free paths according to the gradient descent direction of a set objective function. In addition, some hybrid methods have also been tested. Shen et al.[19] combined deep Q-learning and A* algorithm to propose an intelligent collision avoidance method for unmanned vessels, considering the ship's characteristics and bumper areas. Human experience was introduced into A* grid map to improve the search efficiency, which obtains good collision avoidance performance in a complex environment.

Nowadays, machine learning and artificial intelligence tend to be important tools to solve real-time decision-making problems. With the development of deep reinforcement learning recently, scholars have also applied these methods to the controlling of unmanned ships. Based on deep Q-learning networks (DQN), Cheng and Zhang[20] proposed four kinds of objective functions, consisting the reward function and testing the collision avoidance algorithm of vessels. Abbeel[21] put forward the walking control policies for a quadruped robot using inverse reinforcement learning (IRL). With the application of deep learning in the deterministic policy gradient method, the decision-making actions of reinforcement learning can also be approximated as continuous actions using some functions. Continuous action reinforcement learning methods, such as Deep Deterministic Policy Gradient (DDPG) and Asynchronous Advantage Actor-Critic (A3C), have been tested in control and decision-making problems. Xu et al.[22] used the DDPG method to learn collision avoidance behavior in the continuous state and action space, and obtained an effective collision avoidance strategy. Kim et al.[23] also applied the DDPG algorithm to carry out ship collision avoidance policies, using the relative motion parameters between the own ship and the target ships (other ships in the area except the own ship), and the distance between the own ship and the target track. The state-space simplifies the complexity of learning tasks. In the literature,[24] a constrained DQN is proposed to reduce the complexity of the action space by adding constraints based on some collision avoidance rules on the sea, which improves the learning rate of DQN. Generally, the machine learning methods not only have the advantages of strong learning ability but also have the disadvantages of large requirements of training samples. How to obtain and exploit the training samples with high efficiency and accuracy is the key issue in the application of machine learning in ship collision avoidance.

The International Rules for Collision Avoidance at Sea (COLREGs) is the basic rule for ship collision avoidance handling on the sea. To make decisions in different ship encounters for maritime safety, Li et al.[25] constructed a dynamic personifying intelligent decision-making structure for vessel collision avoidance system, considering rules and human experience. Liu et al.[26] established the shortest path model to realize collision avoidance through path planning based on COLREGs. However, the practical collision avoidance of merchant ships has the following characteristics:

(1) Large size, large hysteresis and inertia

Since merchant ship has the characteristics of large size, small redundancy space, and large hysteresis and inertias, it is difficult to generate proper collision avoidance decision using conventional algorithms.

(2) Complexity and uncertainty of ship collision avoidance scenarios

The COLREGs does not specify all encounter scenarios and may even result in close-quarter situations in several encounter scenarios.

The above characteristics result in the specialty and complexity of collision avoidance of merchant ships. Therefore, the navigation experiences of human experts are still of great significance for learning a collision avoidance policy. To make use of human experts' experience, the most challenging work is to gain the reward functions of machine learning algorithm. Abbeel and Ng[27] proposed an appealing framework for apprenticeship learning. The reward function, while unknown to the apprentice, is

assumed to be a linear combination of a set of state features, which can be observed directly. Although it may be difficult to directly and correctly define the reward function, it is usually much easier to specify the state features on which the reward function depends. With this setting in mind, Abbeel and Ng put forward IRL algorithm to generate a policy that performs at least as well as a human expert with respect to the unknown reward function. Essentially, IRL algorithm is an efficient method for mimicking the expert's behavior, which was widely tested in many kinds of robots.

To improve the safety and rationality of the ship collision avoidance at sea, an IRL algorithm is proposed in this study to learn an approximate reward function and a collision avoidance policy that approaches the expert's demonstration operations. Two kinds of expert demonstration operations (safety and efficiency) are learned by the proposed IRL method in simulation tests, and the results indicate that the proposed IRL method can obtain a good collision avoidance policy, which has the similar performance with human experts.

Ship collision avoidance modeling

Collision avoidance of large merchant ships in open waters follows the principle of "using rudder instead of car," that is, only relying on steering to realize collision avoidance.[28] Therefore, the service speed of the ship is adopted in the entire collision avoidance process in this article and the rudder angle is the action in collision avoidance.

To ensure the accuracy of the collision avoidance model, the following four assumptions are made as follows: (1) The speed of the ship is stable and constant, and the maneuverability of the ship is also stable; (2) it is considered that the collision avoidance process can be simplified to three steering actions before the clearance, that is, two rudder commands to change the course and one rudder command for resailing; (3) ship motions in three degrees of freedom are considered in the collision avoidance process, that is, the sway, surge and yaw; and (4) the ship is located in still water without considering the impact of large wind and waves.

Ship maneuverability model

In this study, the most widely used KVLCC2 ship model[29] is selected as the object, and the ship maneuvering motion model is established to verify the training effect of machine learning on expert demonstration operation. Considering the accuracy and computational complexity, we establish the following nonlinear Nomoto model for KVLCC2

$$T\dot{r} + r + \alpha r^3 = K\delta \tag{1}$$

where K and T are the maneuverability indicators of KVLCC2, α is the nonlinear coefficient, δ is the rudder angle, r and $\dot{r}$ are the heading rate and accelerate, respectively. The Nomoto model represents the relationship between the ship heading and rudder, which is widely used in ship control. Assuming that $\boldsymbol{\eta} = [x \quad y \quad \psi]^{\mathrm{T}}$ and $\boldsymbol{v} = [u \quad v \quad r]^{\mathrm{T}}$ are the position and velocity vectors of the ship, then, the kinematic model of the ship is

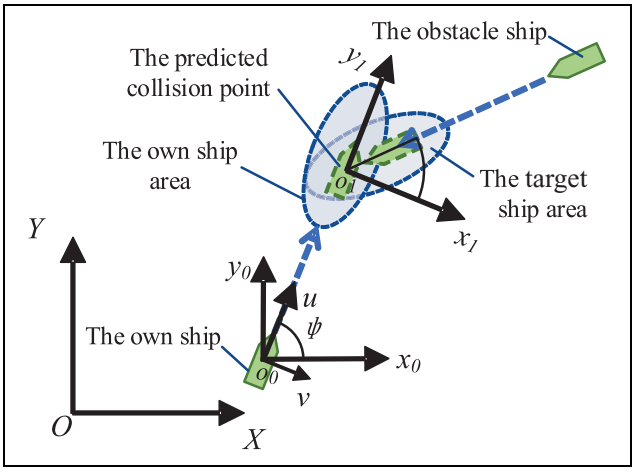

Figure 1. The coordinate system of ship collision avoidance.

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\psi)\boldsymbol{v}$$
$$\boldsymbol{R}(\psi) = \begin{bmatrix} \sin(\psi) & \cos(\psi) & 0 \\ -\cos(\psi) & \sin(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where ψ is the heading angle. When the ship is sailing at the service speed, the surge speed $u \approx U$ and the sway speed $v \approx 0$, where U is the service speed. Then, the final ship maneuverability model can be obtained based on equations (1) and (2).

Modeling of ship collision avoidance process

As shown in Figure 1, the geodetic coordinate system is defined as $X - O - Y$, and the body-fixed coordinate system of the ship is defined as $x_0 - o_0 - y_0$. The heading angle ψ can be defined by the angle between the surge u and the positive x-axis. The own ship's fixed coordinate system $x_1 - o_1 - y_1$ is established at the predicted collision point, and the relative position angle θ is defined by the angle between the course direction of the approaching ship and the positive x_1 axis.

Focusing on the mathematical expression of state-action space, the ship encounter state, collision avoidance policy, and rudder actions should be expressed in high-dimensional space for collision avoidance. The state-action dimension needs to be reduced as much as possible to avoid the dimensional disaster problem in machine learning, and the ship collision avoidance process should be simplified to reduce the difficulty of learning.

Generally, the collision encounters are detected by the perception system of a ship and the responsibility of collision avoidance is determined based on COLREGs. If the

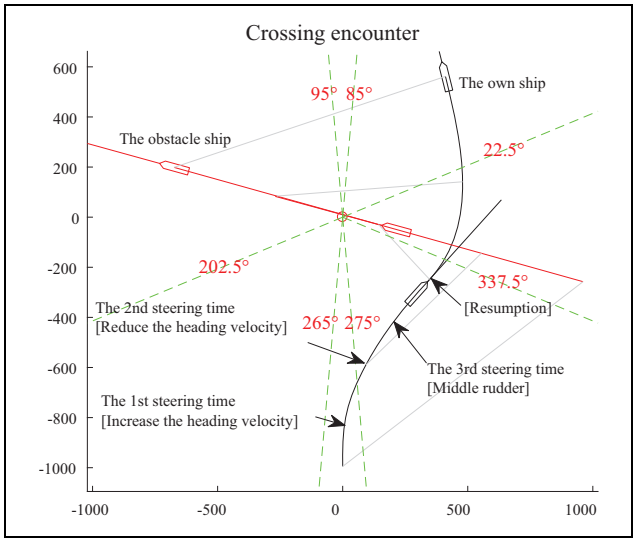

Figure 2. The collision avoidance process.

own ship is the given-way ship and has the responsibility of collision avoidance, the decision result, that is, the rudder command will be applied at the point of the last-minute action to avoid collision and stabilize the course. The detailed process of a typical collision avoidance is shown in Figure 2.

Firstly, the first steering time is defined by the moment that the own ship needs to steer to increase the heading velocity and change the course for avoidance; secondly, the second steering time is defined by the moment that another rudder angle is applied in the opposite direction to rapidly reduce the heading velocity if there is no collision risk; when the heading velocity of the ship is reduced to a certain extent, a middle rudder is adopted to keep the course, that is, the third steering time; finally, the own ship returns back to the original path by trajectory tracking when the two ships have passed the closest positions, which have the distance of closest point of approach (DCPA).

Remark: DCPA is the minimum distance between the closet points of the own ship and the approaching ship in two-ship encounters, which is an important indicator of collision risk.

In summary, the actions to be decided include the first steering time, the first rudder angle, the second steering time, and the second rudder angle. Besides, the third steering time is automatically decided when the heading velocity is reduced to a certain setting threshold.

Ship collision avoidance based on inverse reinforcement learning

Markov decision process of ship collision avoidance

A multistage MDP is shown in Figure 3 and can be described by tuples $\{S, A, P, R\}$, where S is the state, A is the action, P is the state-transition probability, and R is the

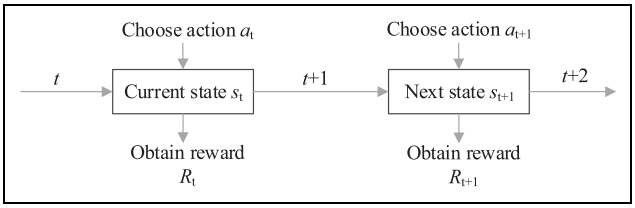

Figure 3. Markov decision process.

reward for the state action. The collision avoidance process can be described by a typical MDP.

Generally, the positions, speeds, and courses of the own ship and other ships can be used for the definition of the state S. In aspect of the positions and courses, it is considered that the relative position and course of each target ship is limited. The circumference of the own ship is divided into seven sectors by 22.5°, 85°, 95°, 202.5°, 265°, 275°, and 337.5° referring to the dividing method of collision avoidance responsibility in COLREGs, as shown in Figure 4(a). At the decision-making moment, the relative position of the target ship is located in one of the seven sectors based on the relative position angle θ and coded as state s_1. In aspect of the speed state, the speed ratios of own ship and target ships are regarded as another state s_2, which is shown in Figure 4(b). Then, the state S in MDP consists of s_1 and s_2.

$A = [a_1, a_2, ..., a_n]$ is the action space of the collision avoidance, each action a_j represents a possible action option for the current state S, which includes the rudder angle and moment. To reduce the complexity of calculation, the rudder angle value is discretized in this study. After taking an action A, the ship gets a reward $R : S \rightarrow \mathbb{R}$, where R is a mapping function from state S to a real number in $\mathbb{R}$. Assuming that $\prod$ denotes a set of rules for any possible selection of the action based on the state, and a policy $\pi \in \prod$ denotes a sequence of rules from the state to the action. Then, the goal of solving the MDP problem is to select a policy to maximize the value function, that is, the discounted sum rewards under this policy π at the decision-making moment

$$V^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(s_t) | s_0 = s, a_t \sim \pi(s_t), s_{t+1} \sim P(s_t, a_t)\right] \tag{3}$$

where $V^{\pi}(s)$ is the state value function under the policy π, which represents the discounted sum of R, γ is the discount factor to reduce the impact of future state on the current state, $E[\cdot]$ represents the expectation, and $P(s_t, a_t)$ represent the state-transition equations obtained by the established ship maneuverability model. Therefore, the optimization problem in MDP is

$$\pi^* = \arg\max_{\pi \in \Pi} V^{\pi}(s) \tag{4}$$

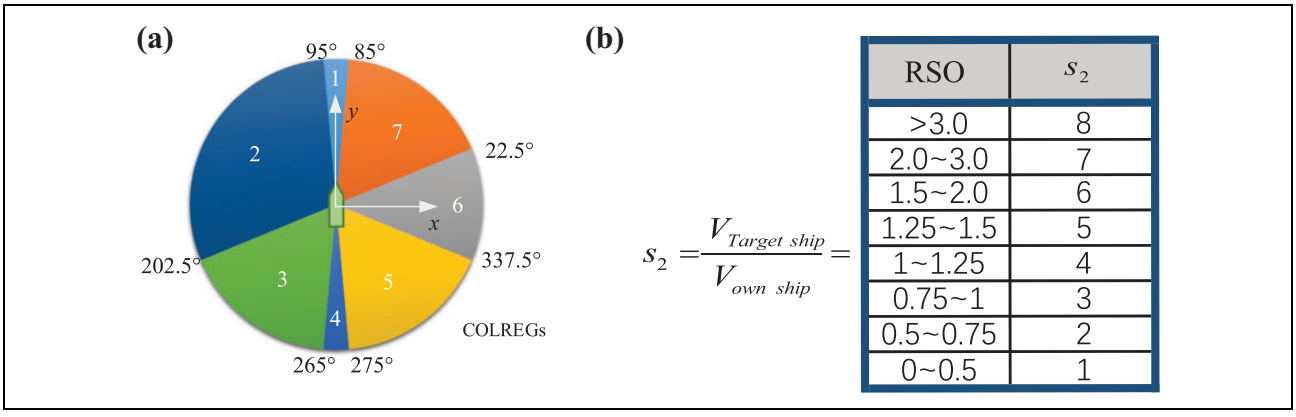

Figure 4. State definition of MDP: (a) state definition of relative position and (b) state definition of RSO. MDP: Markov decision process; RSO: relative speed ratio.

Table 1. State features of collision avoidance.

State features	Description
φ_1–φ_{15}	The sample proportions of DCPA range from 0–100 m, 100–200 m,···, 1400–1500 m, respectively.
φ_{16}–φ_{21}	The sample proportions of heading change of the own ship range from 0–10°, 10–20°,···, 50–60°, respectively.
φ_{22}–φ_{27}	The sample proportions of heading change of the target ship range from 0–10°, 10–20°,···, 50–60°, respectively.

where π^* is the optimal policy, which satisfies $V^{\pi^*}(s) \geq V^{\pi}(s) | \pi \in \Pi, s \in S$. Bellman[30] has proved the existence of the maximum value function $V^{\pi^*}(s)$, and it does not change with time in a certain environment.

Construction of the state features

The state features are the indicators of MDP process to construct the final reward, which are very important for reinforcement learning. In the collision avoidance of large ships at sea, the most significant indicators to characterize the process of collision avoidance include two major aspects: (1) DCPA, which is the distance to closest points of approach between two ships and (2) the maximum heading changes of two ships. The former indicator represents the safety level of collision avoidance, while the latter indicator represents the efficiency level.

Similarly, to reduce the complexity of the machine learning, the finite-state features are set, as given in Table 1. Each state feature represents the proportion of collision avoidance samples in a certain interval to large numbers of stochastic collision avoidance samples, which were conducted by simulation programs, thus, all of the 27 state features of ship collision avoidance process are defined.

Stochastic policy optimization based on cross entropy

The reinforcement learning for MDP is an optimum policy searching process. The idea of introducing noise cross-entropy (CE) algorithm[31] is to randomize a deterministic optimization problem and solve it using rare event simulation and optimization techniques. The main steps of CE are as follows: (1) Generating random data samples and (2) generating new samples with a certain distribution and optimizing the sample distribution.

Without losing generality, the reward function R of reinforcement learning can be represented by a linear combination function

$$R(\boldsymbol{s}) = R_W(\boldsymbol{s}) = \sum_{k=1}^{n} \omega_k \phi_k(\boldsymbol{s}) = \boldsymbol{W}^T \cdot \boldsymbol{\phi} \tag{5}$$

where $\boldsymbol{W} = (\omega_1, \omega_2, ..., \omega_n)$ is the weight matrix for the state $\boldsymbol{s}$ and $\boldsymbol{\phi}$ is the state feature.

For a random weight matrix $\boldsymbol{W} = (\omega_1, \omega_2, ..., \omega_n)$, a set of random policies $\prod = [\pi_1, \pi_2, \ldots, \pi_n]$ is generated by CE algorithm. Then, the state features $\boldsymbol{\phi}$ are obtained by executing the action a_i mapped by each policy π_i under the current state s_i. After that, the immediate reward can be calculated by equation (5), and the value function can be updated by equation (3).

In each iteration of CE, the policy $\prod_t$ is obtained using the Gauss distribution in high-dimensional space, and the mean and variance of $\prod_t$ are as follows

$$\begin{aligned} \mu_{t+1} &= \frac{1}{\beta} \sum_{i=1}^{\beta} \pi_i^{(t)} \\ \sigma_{t+1}^2 &= \frac{1}{\beta} \sum_{i=1}^{\beta} (\pi_i^{(t)} - \mu_{t+1})^T (\pi_i^{(t)} - \mu_{t+1}) \end{aligned} \tag{6}$$

where β is the sample selection ratio of the policy, that is, only β policies with the largest value function are taken in

Algorithm 1. The noise CE algorithm

Input: The mean and variance of initial policy distribution, μ and σ; the number of policies generated in each iteration, n; the selection ratio, β; the noise in each iteration Z_t.
Output: The optimal policy π^*.
1: **for** $t = 1$ to n **do**
2: Use the distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ to generate N policies $\pi^{(1)}, \pi^{(2)}, \cdots, \pi^{(N)}$;
3: Calculate the value function of each policy based on Eq. (3).
4: Select $\beta \cdot N$ policies with the largest value function, and calculate the new mean and variance of the selected policies, $\tilde{\mu}_{t+1}$ and $\tilde{\sigma}_{t+1}$.
5: $\mu_{t+1} \leftarrow \tilde{\mu}_{t+1}$
6: $\sigma_{t+1} \leftarrow \tilde{\sigma}_{t+1}$
7: $t \leftarrow t + 1$
8: **end for**
9: **return** the policy with the largest value function π^*

each iteration and the sample mean and variance of the β policies are calculated as the mean and variance of the random policies for the next iteration. The CE algorithm converges fast, but it is easy to fall into suboptimal solution. To deal with this problem, Szita and Lörincz[32] introduce Z noise component in the variance and achieve better global optimization results

$$\sigma_{t+1}^2 = \frac{\sum_{i=1}^{\beta} (\pi_i^{(t)} - \mu_{t+1})^T (\pi_i^{(t)} - \mu_{t+1})}{\beta} + Z_{t+1} \tag{7}$$

where $Z_{t+1} = C \cdot (t+1) + d$ and C, d are the constants. The calculation steps of the noise CE algorithm can be denoted in Algorithm 1.

Expert policy approximation based on projection method

As a search process of reinforcement learning, the noise CE method needs to search the optimal policy on the premise of defining the weight matrix of the reward function. The projection method[27,33] is used in this study to obtain the approximated reward of expert policy using the weight matrix $\boldsymbol{W}$ as the medium.

Firstly, the state feature expectations of expert demonstration samples are calculated

$$\mu_{\mathrm{E}} \approx \hat{\mu} = \frac{1}{k} \sum_{i=1}^{k} \sum_{l=0}^{\infty} \gamma^l f(s_l) \tag{8}$$

where k is the expert demonstration sample size and $f(s_l)$ is the state feature of samples s at time l. Then, the weight vector $\boldsymbol{W}^{(0)}$ is initialized randomly, and an initial strategy $\mu^{(0)}$ is generated randomly. Based on $\boldsymbol{W}^{(1)} = \mu_E - \mu^{(0)} \bar{\mu}^{(0)} = \mu^{(0)}$, the first generation weight vector $\boldsymbol{W}^{(1)}$ and the state feature mean $\bar{\mu}^{(0)}$ are obtained. After that, the weight matrix W can be updated by the following equation

$$\boldsymbol{W}^{(i)} = \mu_E - \bar{\mu}^{(i-2)} - \frac{\left(\mu^{(i-1)} - \bar{\mu}^{(i-2)}\right)^T \left(\mu_E - \bar{\mu}^{(i-2)}\right)}{\left(\mu^{(i-1)} - \bar{\mu}^{(i-2)}\right)^T \left(\mu^{(i-1)} - \bar{\mu}^{(i-2)}\right)} \left(\mu^{(i-1)} - \bar{\mu}^{(i-2)}\right) \tag{9}$$

Moreover, the flowchart of projection algorithm is shown in Figure 5.

The state feature expectation corresponding to the policy π^* can become close to the state feature expectation of the expert demonstration sample based on the projection method.

In summary, the reward function is obtained by projection-based IRL, and the CE-based RL method is used to search the optimal policy. The reward function is updated by the difference between the expectations of state features of the current policy and the expert demonstration until the convergence condition is satisfied. The final flowchart of the policy search is shown in Figure 6.

Simulation experiments

Acquisition of the expert demonstration samples

To conduct the IRL simulation experiments, large amounts of expert demonstration samples are acquired. A simulation software for ship collision avoidance operation based on the established ship maneuverability model is developed, as shown in Figure 7.

The software generates different encounter scenarios and judges the responsibility of collision avoidance according to the COLREGs. If the own ship has the responsibility of avoidance, the experts need to drag the horizontal slider to control the rudder angle of own ship to change the course. The software will automatically add the simulation results into training samples.

Validation of the proposed projection-based inverse reinforcement learning method

To reduce the randomness, fixed encounter scenarios are adopted. In range of 0–360°, as shown in Figure 4(a), the interval of the relative position angle of the target ship is 2.88°. The other ship's speeds are set as 4, 6, 8, 12, 14, 18, 25, and 30 knots, respectively. The own ship's speed is set as 10 knots, that is, the service speed. With this kind of method, 1000 encounter scenarios are designed and used in the simulation software to obtain demonstrations by experts, and these typical encounter scenarios were called base scenarios. In fact, on the one hand, the movements of ships are so complex that it is impossible to establish all encounter scenarios. In our research, testing set scenarios was classified according to the encounter situation judgment methods in COLREGs. On the other hand, as the directions and speeds of target ships are the major concerns of captains, the states of MDP in our research only consist

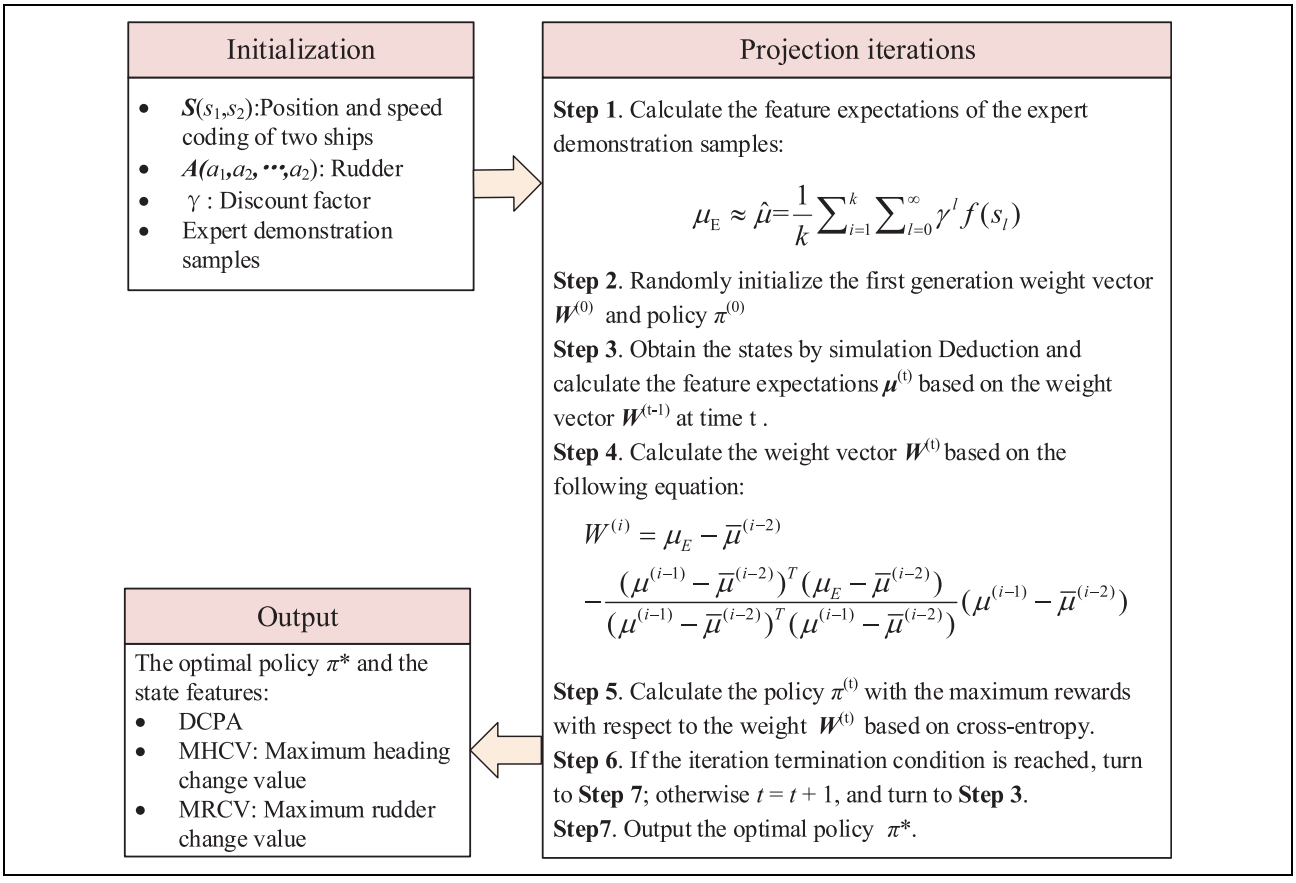

Figure 5. Flowchart of projection algorithm.

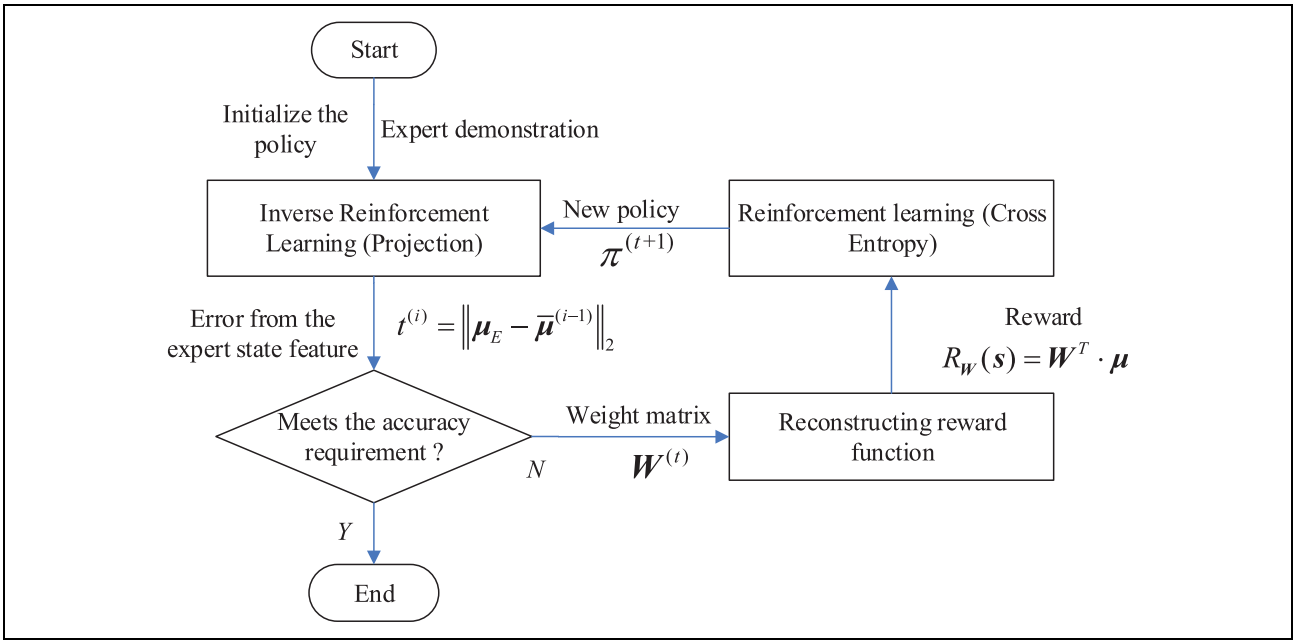

Figure 6. Flowchart of the policy searching.

of s_1, s_2, and the training set data were conducted by software, removing lots of random events. As a result, the state features could be determined by initial states and policies.

In addition, the experts are divided into two categories, that is, the safety experts and efficiency experts. The safety experts give priority to the safety in collision avoidance,

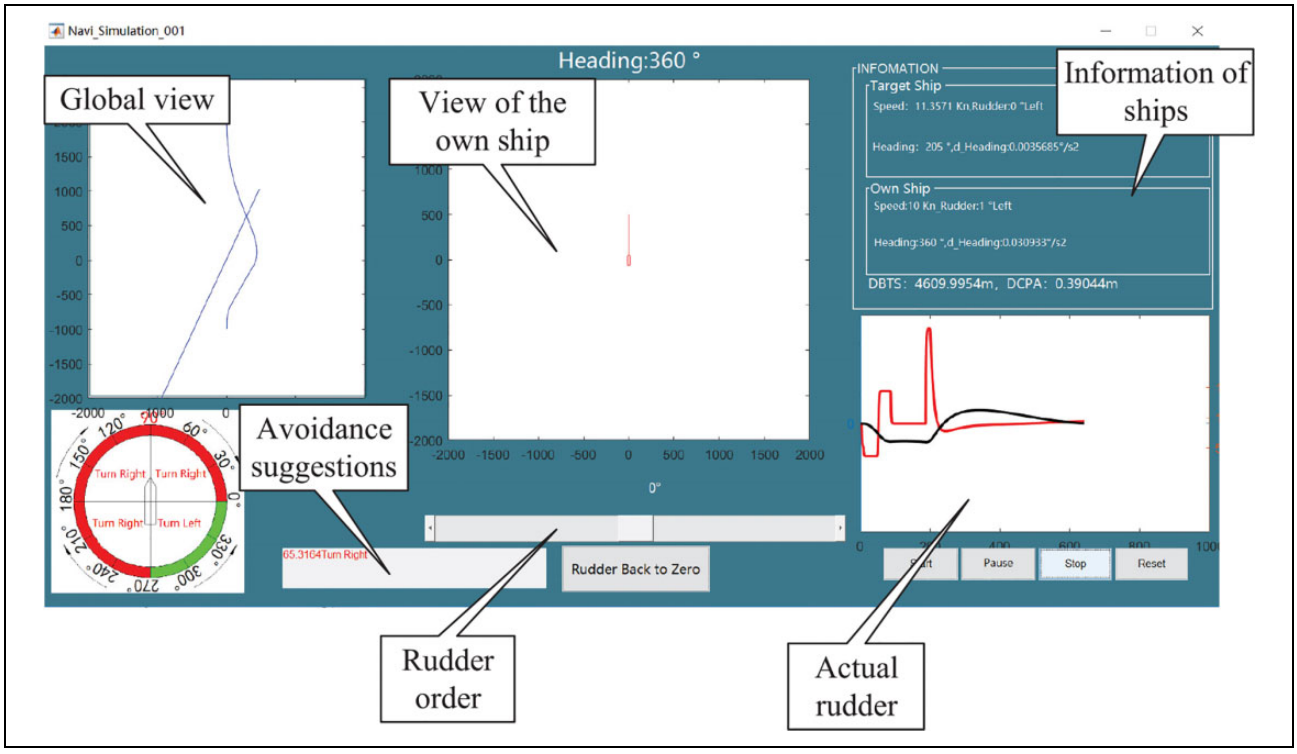

Figure 7. Interface of the simulation software for ship collision avoidance.

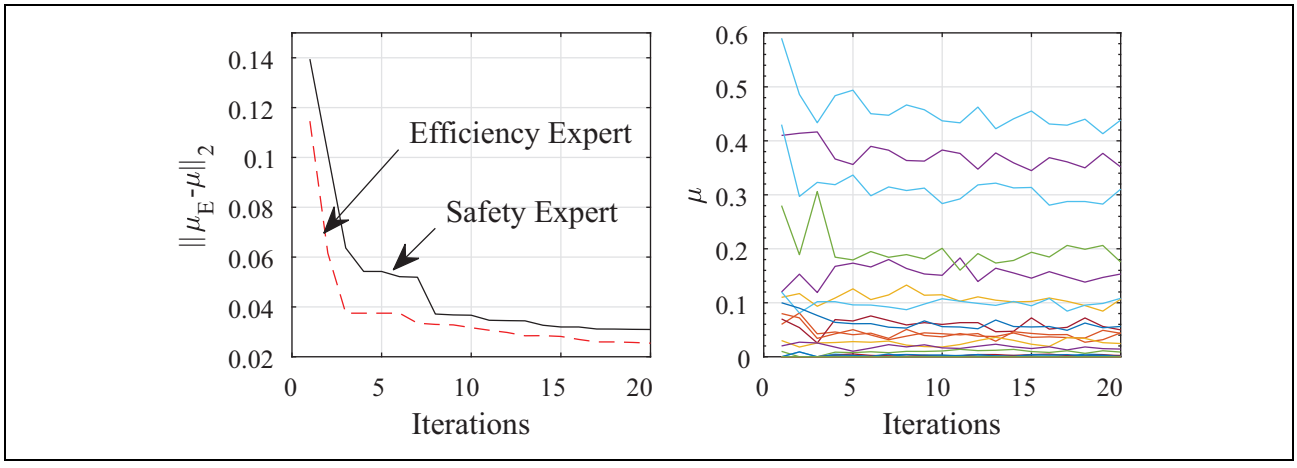

Figure 8. The feature deviation curve.

who usually use a larger rudder angle to achieve a larger DCPA and heading change, so as to keep the two ships as far away as possible to ensure the safety. While the efficiency experts give priority to the efficiency, who usually use a smaller rudder angle to shorten the ship's voyage under the premise of the safety between two ships, so as to improve the economy.

Demonstrations of these two kinds of experts are obtained by four sailors and experts using the simulation software. The discount factor is set as $\gamma = 0.99$. The sample selection ratio of the CE algorithm is set as β=10% and the noise factor is set as $Z_t = 0.2 \cdot t + 1$. The obtained 1000 samples of different encounter scenarios are used for learning of each policy π in the IRL method. The software runs on a computer with I7-6700 (four-core, eight-thread) CPU and the features in IRL method converge in about 6000 s. The feature deviations between the demonstrations and the learned policy are shown in Figure 8.

It can be seen from Figure 8 that the feature deviations can converge to a good level within about eight iterations. Moreover, the comparison of the state features between the convergent policy and the expert demonstrations is shown in Figures 9 and 10.

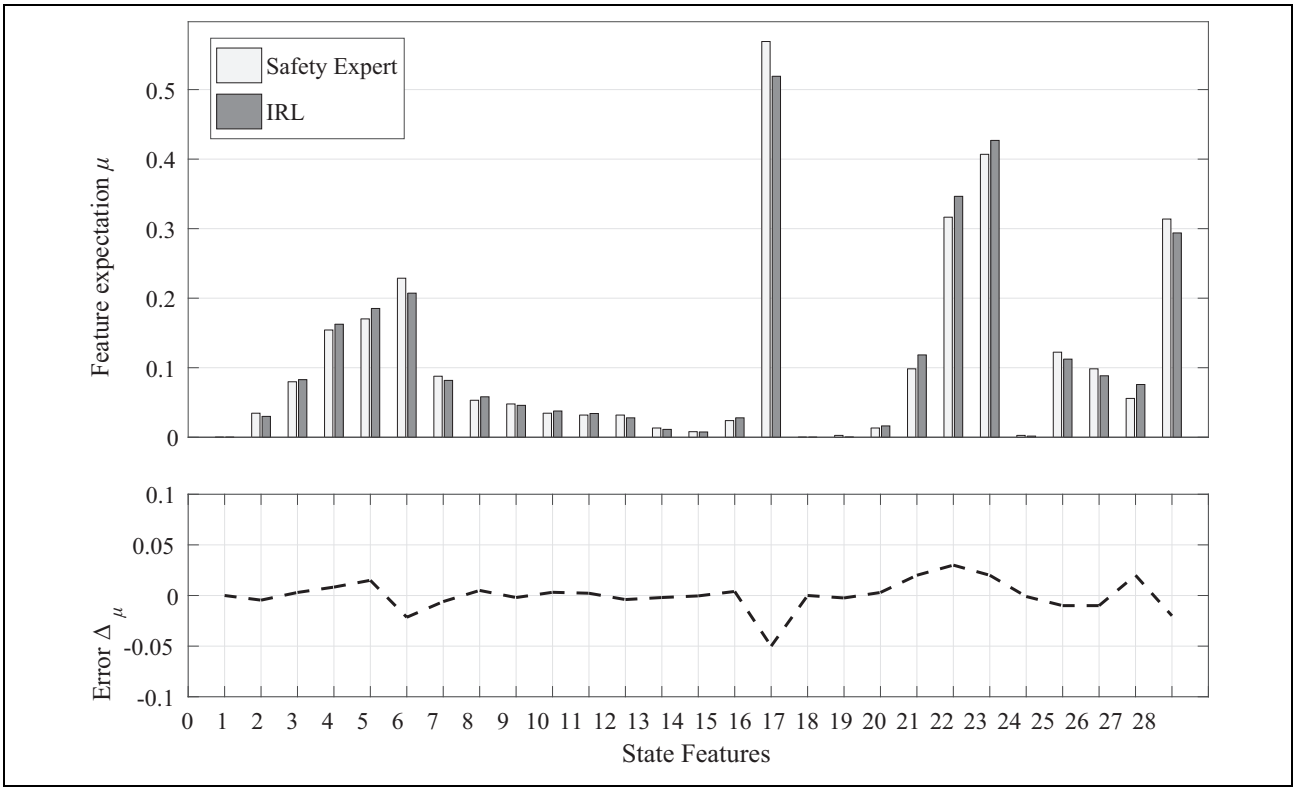

Figure 9. The state features of safety expert and IRL policy. IRL: inverse reinforcement learning.

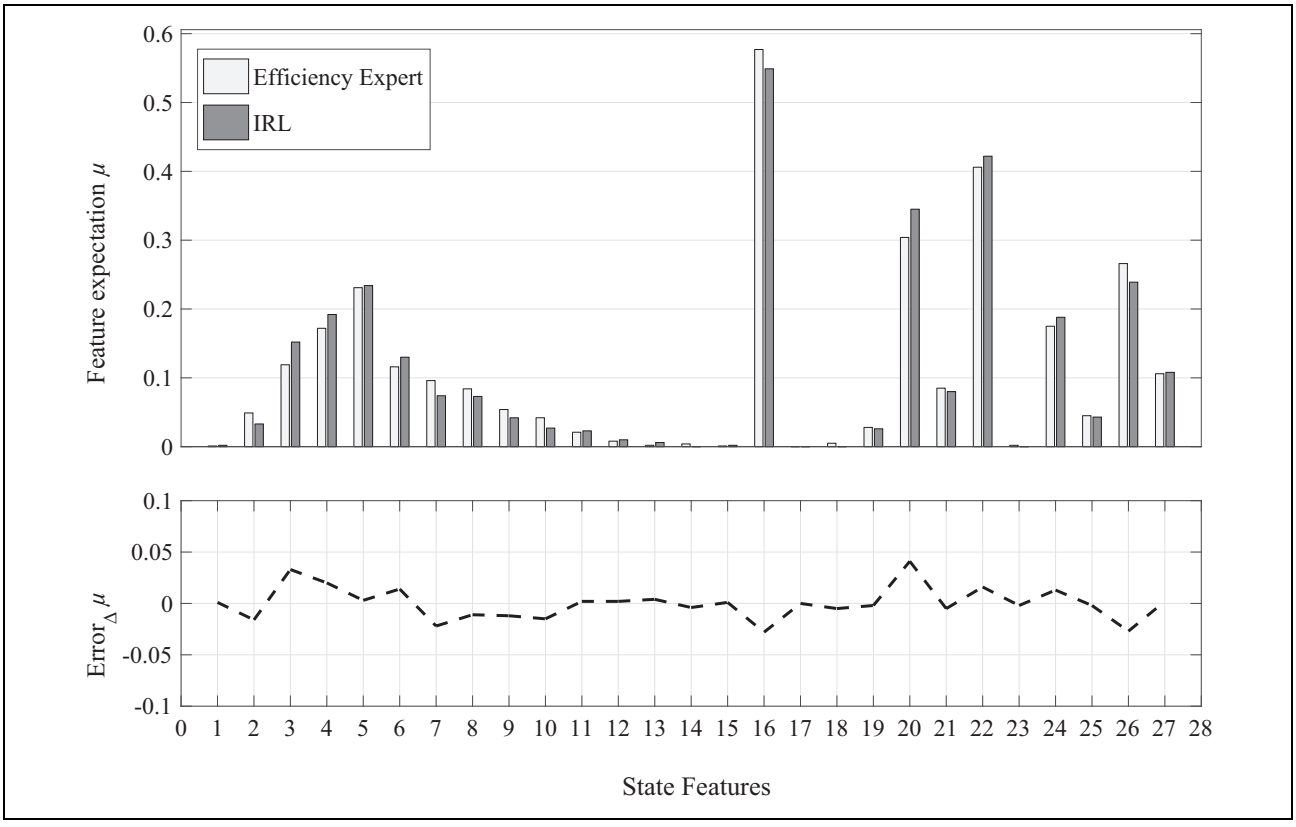

Figure 10. The state features of efficiency expert and IRL policy. IRL: inverse reinforcement learning.

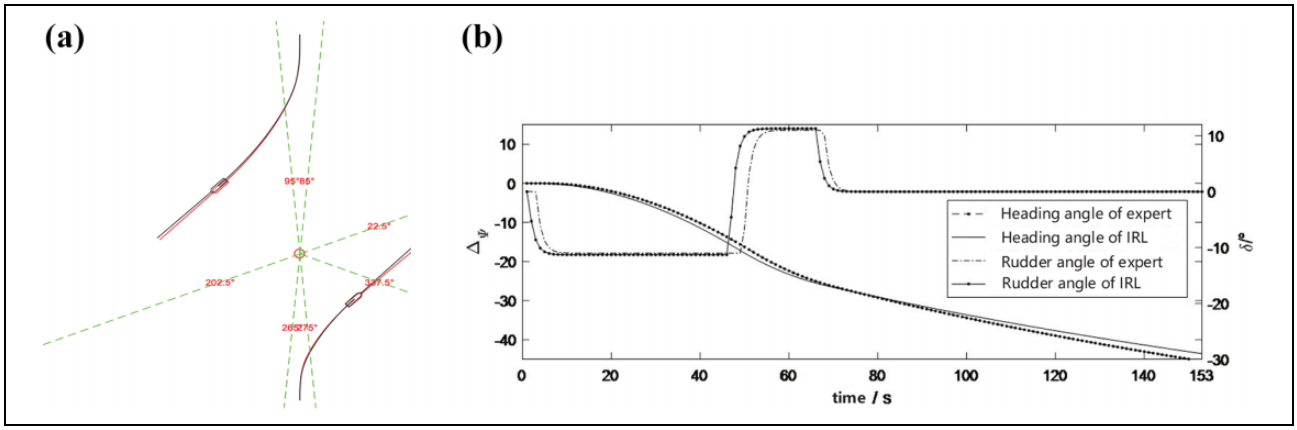

Figure 11. Simulation results in head-on encounter compared with safety experts. (a) Trajectories of two ships in head-on encounter compared with safety experts. (b) Headings and rudders of two ships in head-on encounter compared with safety experts.

In Figure 9, the maximum error between the state features of the safety expert demonstrations (the white bar) and those of the policy trained by IRL (the gray bar) is less than 5%, which indicates that the proposed IRL method can obtain a collision avoidance policy similar to the safety expert. In the aspect of the DCPA, the expectation of the sixth feature represents the sample proportion of DCPA between 500 m and 600 m, which is the largest expectation with respect to DCPA and shows that the collision avoidance samples given by the safety experts are mostly concentrated in this area. The sample distribution of DCPA between 600 m and 1200 m is more uniform than that between other ranges, which indicates that the searched policies achieve a larger distance between two ships during more collision avoidances, corresponding to larger DCPAs.

In the aspect of the heading change, the largest expectations are the 16th, 22nd, 21st, and 27th feature expectations. The 16th and 22nd feature expectations are the sample proportions of that the own ship and target ships keeping course, respectively, corresponding to the keeping course scenarios, in which the target ship has avoidance responsibility. The 21st and 27th feature expectations represent the sample proportions of that the maximum heading change of the own ship and target ships varies between 40° and 50°, respectively. It can be seen that the safety experts prefer to control the heading change between 40° and 50°.

In Figure 10, the maximum error between the state features of the efficiency expert demonstrations (the white bar) and those of the policy trained by IRL (the gray bar) is also less than 5%. In the aspect of the DCPA, the fifth feature expectation is the largest expectation with respect to DCPA, which indicates that the collision avoidance samples given by efficiency experts are more likely to be completed with a medium DCPA (about four to five times the length of the ship). In the aspect of the heading change, the largest expectations are the 20th and 26th feature expectations except for the 16th and 22nd features for keeping the course, which indicates that more samples have the heading change between 30° and 40°, indicating that efficiency experts prefer to choose less DCPAs and heading changes to achieve higher efficiency level.

Simulation verification of random collision avoidance

To show the decision-making performance of the proposed IRL method in different collision avoidance scenarios more intuitively, 1000 base scenarios, including head-on, crossing, and overtaking encounters, are selected as typical ship encounter scenarios. The learned policies through IRL are used to control ships during collision avoidance, and the results are compared with the demonstrations of safety experts and efficiency experts in the same encounter scene, as shown in Figures 11 to 16. In Figure 11(a), 12(a), 13(a), 14(a), 15(a), and 16(a), the red curves are the expert demonstrations and the black curves are the ship trajectories controlled by the learned polices. It can be seen that the simulation results of learned polices are very close to those of the safety experts and efficiency experts. From Figure 11(b), 12(b), 13(b), 14(b), 15(b), and 16(b), it can be seen that the rudder angle of the IRL policy is also similar to that of the expert demonstrations, although the steering time is slightly different.

In addition, the DCPA and maximum heading angle changes of expert demonstrations and learned policies are plotted as a box diagram, as shown in Figure 17. It can be seen that the DCPA values of the safety experts and corresponding safety IRL policy are about 480 m, which are larger than those of the efficiency experts and corresponding efficiency IRL policy (about 400 m), indicating that the safety experts and safety IRL policy achieve more safe avoidance results. On the contrary, the efficiency experts and efficiency IRL policy adopt smaller rudder change values to realize collision avoidance with smaller heading angle changes, which means that the own ship can return to the original route faster after collision avoidance is completed. Both own ship and target ship were controlled by the same policy in the simulation software. For example, the target ship in safety experts' demonstrations is

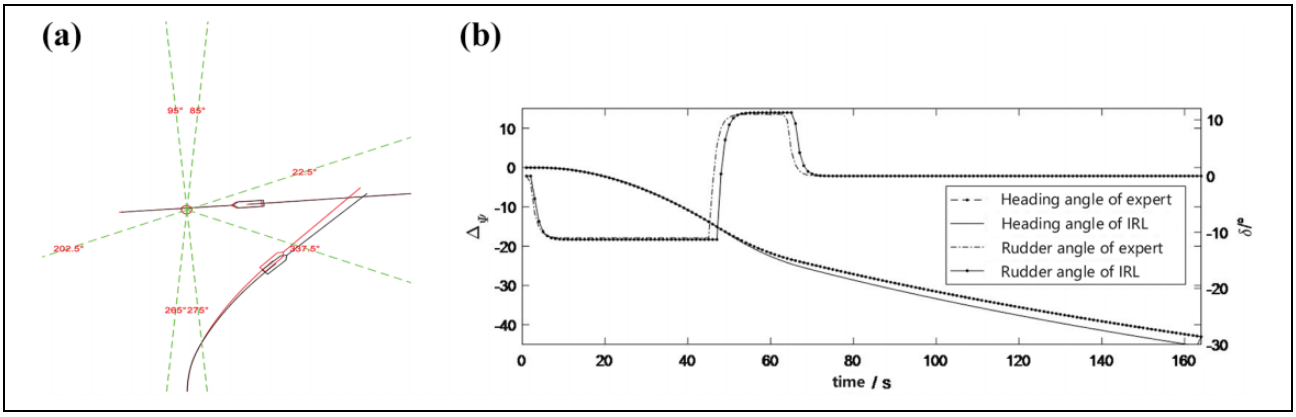

Figure 12. Simulation results in crossing encounter compared with safety experts. (a) Trajectories of two ships in crossing encounter compared with safety experts. (b) Headings and rudders of two ships in crossing encounter compared with safety experts.

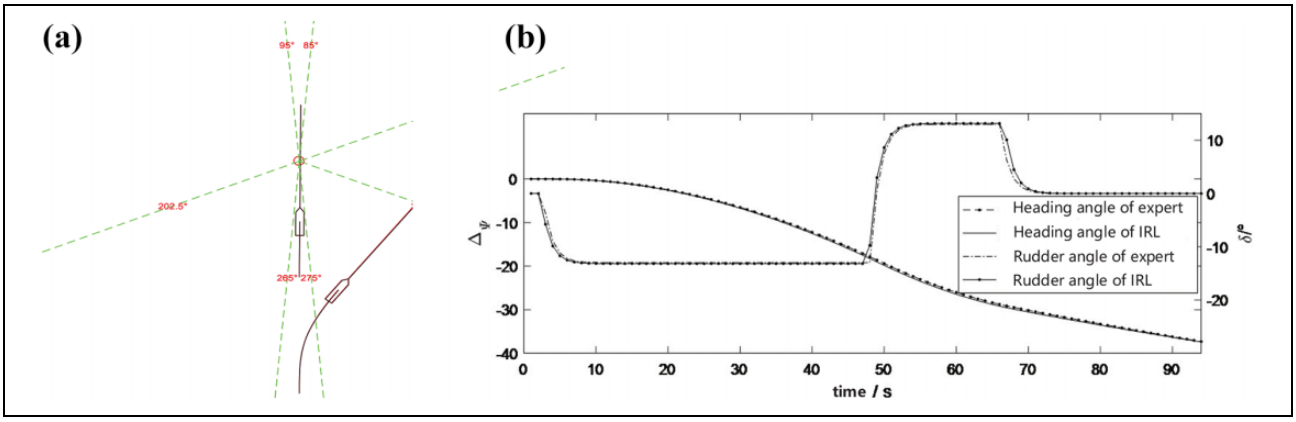

Figure 13. Simulation results in overtaking encounter compared with safety experts. (a) Trajectories of two ships in overtaking encounter compared with safety experts. (b) Headings and rudders of two ships in overtaking encounter compared with safety experts.

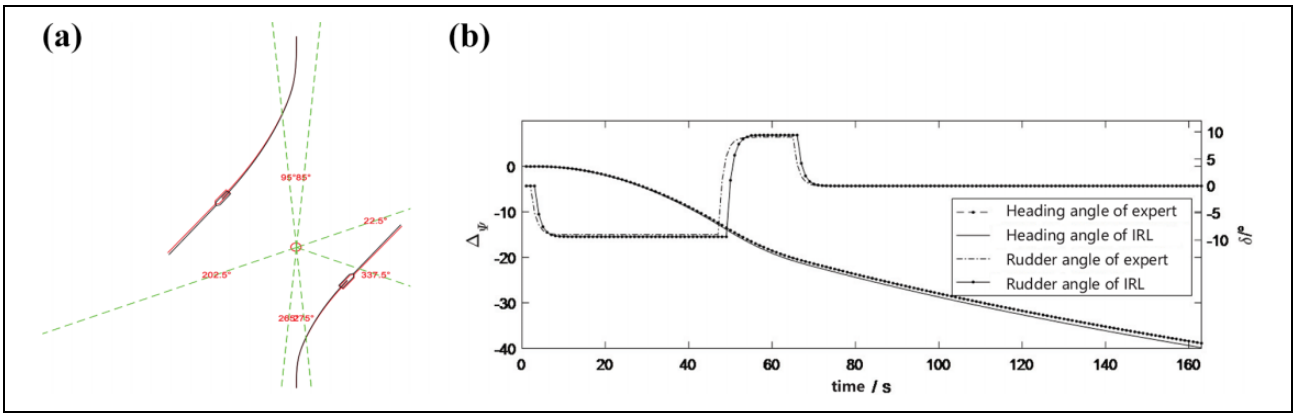

Figure 14. Simulation results in head-on encounter compared with efficiency experts. (a) Trajectories of two ships in head-on encounter compared with efficiency experts. (b) Headings and rudders of two ships in head-on encounter compared with efficiency experts.

controlled by safety IRL policy. For the same policy, the statistical average heading changes of other ships are less than the own ship since the speeds of other ships in most samples are faster than that of own ship and the experts tend to steer with smaller rudder angles in advance for high-speed ships.

Simulation comparisons of the proposed inverse reinforcement learning method and normal reinforcement learning method

To compare the performance of IRL and concise reinforcement learning applied in collision avoidance scenarios,

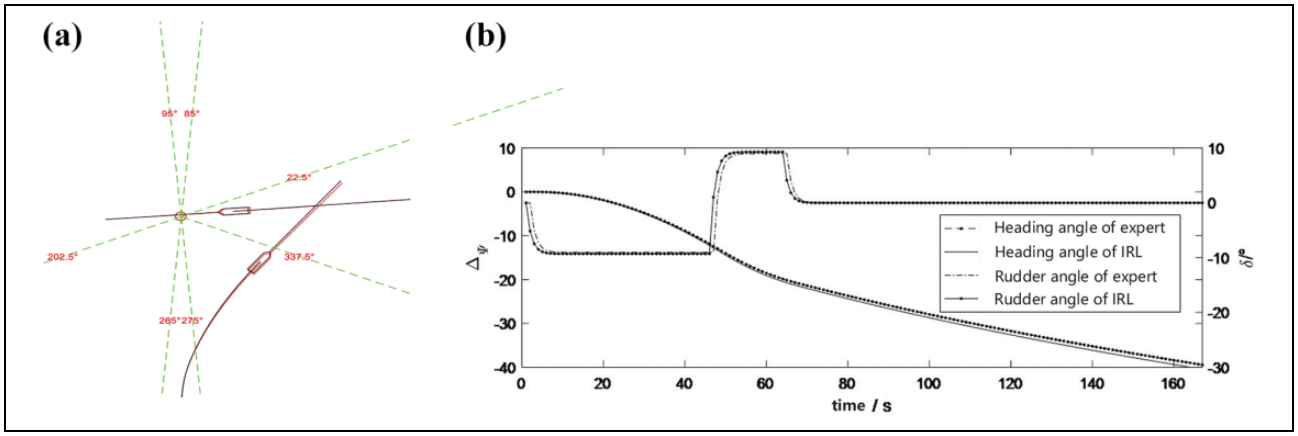

Figure 15. Simulation results in crossing encounter compared with efficiency experts. (a) Trajectories of two ships in crossing encounter compared with efficiency experts. (b) Headings and rudders of two ships in crossing encounter compared with efficiency experts.

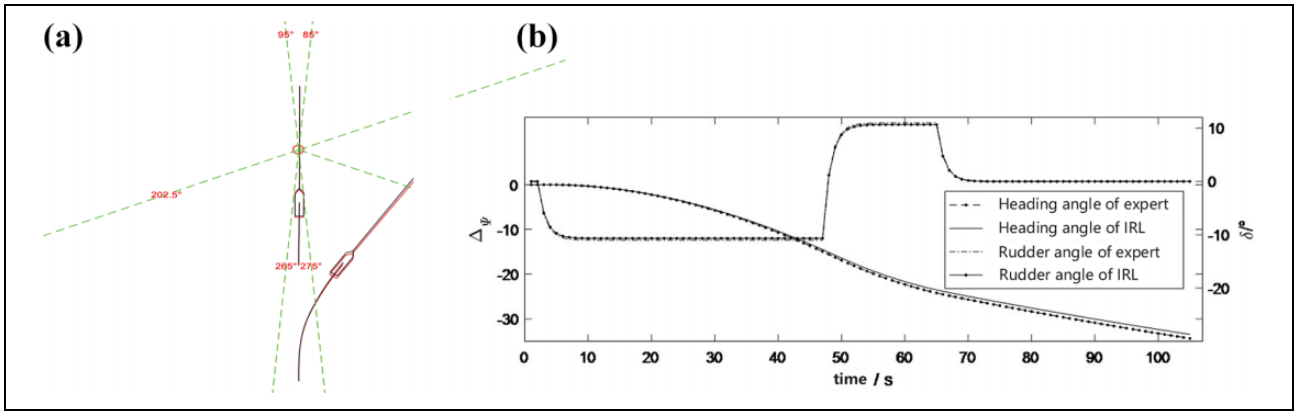

Figure 16. Simulation results in overtaking encounter compared with efficiency experts. (a) Trajectories of two ships in overtaking encounter compared with efficiency experts. (b) Headings and rudders of two ships in overtaking encounter compared with efficiency experts.

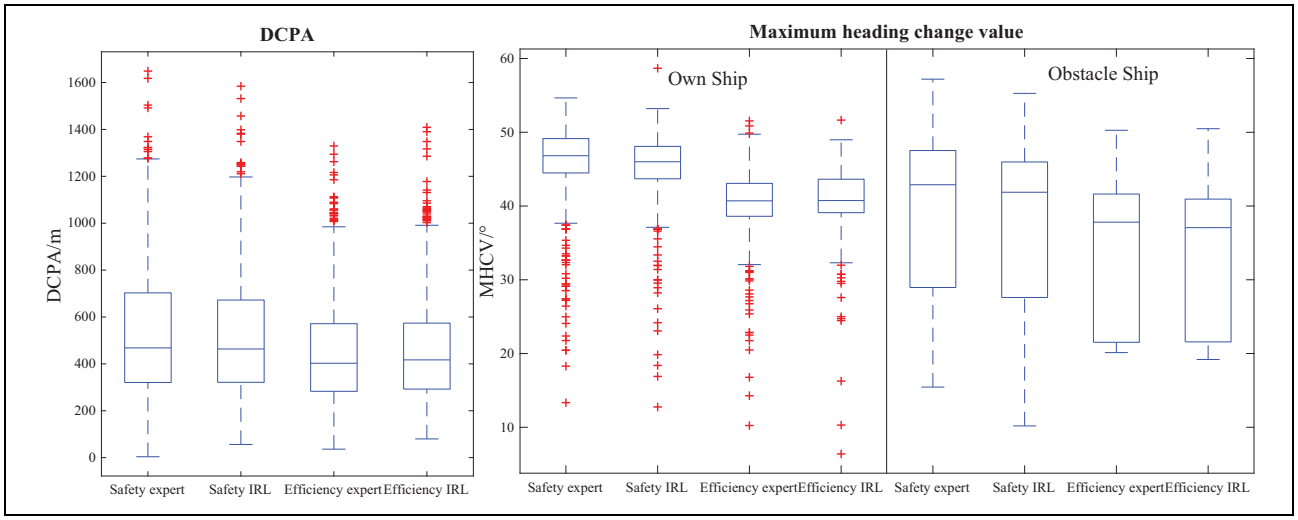

Figure 17. Statistics of experts' demonstrations and IRL policy simulations. IRL: inverse reinforcement learning.

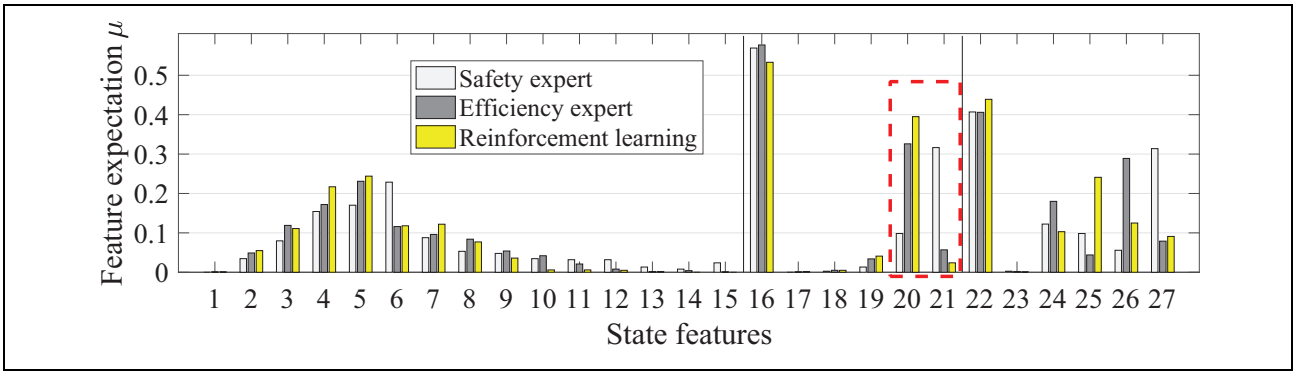

Figure 18. The state features of experts and reinforcement learning policy.

concise reinforcement learning was also tested. It is difficult to define the reward functions in collision avoidance. As the state features of MDP is a 27-dimension vector, a linear combination reward function based on state features could be defined as follows

$$R(s) = R[\boldsymbol{\mu}(s)] = \sum_{k=1}^{n} \omega_k \phi_k(s) = \boldsymbol{W}^T \cdot \boldsymbol{\mu} \tag{10}$$

where $\boldsymbol{W} = (\omega_1, \omega_2, \ldots, \omega_n)$ is the weight vector and $\boldsymbol{\phi}$ is the state feature of test samples conducted by reinforcement learning. As the weight $\boldsymbol{W}$ could influence the reward function value directly, it could be defined according to the safety requirements for navigation. The feature expectations of reinforcement learning in the specific collision avoidance scenarios are shown in Figure 18.

How to conduct the collision avoidance according to COLREGs is the major challenge in our research for reinforcement learning. For example, in some encounter scenarios, ships, on one hand, should obey COLREGs, turning to special direction instead of the other direction to avoid potential collision. On the other hand, the responsibility of collision avoidance is so complex that in different encounter scenarios, own ship need not avoid collisions. As a result, it is difficult for reinforcement learning to generate proper policies. Therefore, we develop an expert system, which could judge whether the avoidance action is right based on COLREGs. If the avoidance action does not obey COLREGs, the expert system could generate determine factor as the hard constraints. In Figure 18, the state features of reinforcement learning are similar with the state features of efficiency expert, showing that the weight vector of reward function prefers efficiency more than safety on the basis of satisfying with the safety requirements of COLREGs. Similarly, other optimization algorithms, such as A*, APF, and GA, also need hard constraints based on the expert system of COLREGs.

On the contrary, it is relatively easy for human to control a ship to avoid collision in most encounter scenarios, especially in some complex scenarios. Human experts' prior knowledge about collision avoidance is so valuable that it could improve the validity and practicability of algorithm. IRL is more suitable to collect collision avoidance policies.

In summary, the IRL algorithm proposed in this article can easily obtain the decision-making policies of human experts, so that the algorithm has a similar collision avoidance performance with human drivers.

Conclusions

An IRL method through CE-based policy optimization and projection-based policy approximation is proposed in this study to realize ship collision avoidance. The main works of this article are concluded as follows:

1. The ship maneuverability model is established, and the expert demonstration operation software is developed to obtain collision avoidance samples through the expert operation.
2. The distributions of DCPA and maximum heading angle change are taken as the state features, and the collision avoidance policy of expert demonstrations is obtained by the proposed IRL method. The learned policy has similar performance with the expert demonstrations, which indicates that the proposed IRL method is suitable for collision avoidance policy training of merchant ships.

However, the practicability of the IRL method also depends on the reasonableness of expert demonstrations. Therefore, it is necessary to follow the captain's driving habits of real merchant ships and collect real operation data extensively to expand the samples for IRL so as to find a reasonable trade-off between the safety and efficiency for autonomous collision avoidance. Subsequently, further research on the proposed IRL method will focus on data collection of real ship navigation data collection.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Key Research and Development Program of China [No. 2018YFB1600400], the Development of Ship Situation Intelligent Awareness System [MC-201920-X01], the High-tech Ship Project of Ministry of Industry and Information Technology [MC-201712-C07], the National Natural Science Foundation of China [No. 52001243], the Fundamental Research Funds for the Central Universities [WUT:203144003], and the Fund of National Engineering Research Center for Water Transport Safety [No.C2019002].

ORCID iDs

Mao Zheng https://orcid.org/0000-0002-5533-2024
Shuo Xie https://orcid.org/0000-0001-9281-5351

References

1. Patle BK, Ganesh BL, Pandey A, et al. A review: on path planning strategies for navigation of mobile robot. *Def Technol* 2019; 15(4): 582–606.
2. Xue YZ, Wei Y, and Qiao Y. The research on ship intelligence navigation in confined waters. *Adv Mater Res* 2012; 442: 398–401.
3. Seda M. Roadmap methods vs. cell decomposition in robot motion planning. In: *Proceeding of the 6th WSEAS international conference on Signal Processing, Robotics and Automation*, Corfu Island, Greece, 16-19 February 2007, pp. 127–132.
4. Ma Y, Gan L, Zheng Y, et al. Autonomous ship safe navigation using smoothing A* algorithm. *Open Cybern Syst J* 2014; 8: 72–76.
5. Phanthong T, Maki T, Ura T, et al. Application of A* algorithm for real-time path re-planning of an unmanned surface vehicle avoiding underwater obstacles. *J Mar Sci Appl* 2014; 13(1): 105–116.
6. Chen C, Chen X Q, Ma F, et al. A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Eng* 2019; 189: 106299.
7. Wang X, Shi Y, Ding D, et al. Double global optimum genetic algorithm-particle swarm optimization-based welding robot path planning. *Eng Optim* 2016; 48(2): 1–24.
8. Liu J, Yang J, Liu H, et al. An improved ant colony algorithm for robot path planning. *Soft Comput* 2017; 21(19): 106–113.
9. Savkin AV and Wang C. A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles. *Robotica* 2013; 31(6): 993–1001.
10. Sun X, Wang G, Fan Y, et al. Collision avoidance using finite control set model predictive control for unmanned surface vehicle. *Appl Sci* 2018; 8(6): 926–944.
11. Hoy M, Matveev AS, and Savkin AV. Algorithms for collision free navigation of mobile robots in complex cluttered environments: a survey. *Robotica* 2015; 33(3): 1–35.
12. Song L, Chen Z, Dong Z, et al. Collision avoidance planning for unmanned surface vehicle based on eccentric expansion. *Int J Adv Robot Syst* Epub ahead of print 10 July 2019. DOI: 10.1177/1729881419851945.
13. Wu P, Xie S, Liu H, et al. Autonomous obstacle avoidance of an unmanned surface vehicle based on cooperative manoeuvring. *Ind Robot* 2017; 44(1): 64–74.
14. Yoo B and Kim J. Path optimization for marine vehicles in ocean currents using reinforcement learning. *J Mar Sci Technol* 2016; 21(2): 334–343.
15. Chen C, Ma F, Liu J L, et al. A novel path planning approach for unmanned ships based on deep reinforcement learning. *World Sci Proc Ser Comput Eng Inform Sci* 2018; 1: 626–633.
16. Zheng M, Yang FQ, Dong ZP, et al. Carrier-borne aircrafts aviation operation automated scheduling using multiplicative weights apprenticeship learning. *Int J Adv Robot Syst* 2019; 16(1): 370–388.
17. Kim H, Kim S H, Jeon M, et al. A study on path optimization method of an unmanned surface vehicle under environmental loads using genetic algorithm. *Ocean Eng* 2017; 142: 616–624.
18. Chen YL, Cheng J, Lin C, et al. Classification-based learning by particle swarm optimization for wall-following robot navigation. *Neurocomputing* 2013; 113(8): 27–35.
19. Shen HQ, Guo C, Li T S, et al. An intelligent collision avoidance and navigation approach of unmanned surface vessel considering navigation experience and rules. *Harbin Gongcheng Daxue Xuebao* 2018; 39(6): 998–1005.
20. Cheng Y and Zhang W. Concise deep reinforcement learning obstacle avoidance for under-actuated unmanned marine vessels. *Neurocomputing* 2018; 272: 63–73.
21. Abbeel P. *Apprenticeship learning and reinforcement learning with application to robotic control*. Palo Alto: Stanford University, 2008.
22. Xu H, Wang N, Zhao H, et al. Deep reinforcement learning-based path planning of underactuated surface vessels. *Cyber-Phys Syst* 2019; 5(1): 1–17.
23. Kim DH, Lee SU, Nam JH, et al. Determination of ship collision avoidance path using deep deterministic policy gradient algorithm. *J Soc Nav Arch Korea* 2019; 56(1): 58–65.
24. Zhang R, Wang X, Liu K, et al. Ship collision avoidance using constrained deep reinforcement learning. In: *Proceedings of the IEEE 5th international conference on behavioral, economic, and socio-cultural computing*, Kaohsiung, Taiwan, 12–14 November 2018, pp. 115–120. Berlin, Germany: Springer.
25. Li LN, Wang JL, and Chen GQ. Integrated machine learning strategy of PIDVCA theory. *Inf Control* 2011; 40(3): 359–368.
26. Liu DD, Shi GY, Li WF, et al. Decision support of collision avoidance based on shortest avoidance distance and collision risk. *Shanghai Haishi Daxue Xuebao* 2018; 39(1): 13–18.
27. Abbeel P and Ng AY. Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of the 21st international conference on machine learning*, Banff, Canada, 04–08 July 2004, pp. 1–8. New York, US: ACM.

28. Zhao Z and Wang JX. Ship automatic anti-collision path simulations based on reinforcement learning in different encounter situations. *Sci Technol Eng* 2018; 18(18): 218–223.
29. Liu J, Quadvlieg F, and Hekkenberg R. Impacts of the rudder profile on manoeuvring performance of ships. *Ocean Eng* 2016, 124: 226–240.
30. Bellman R. Dynamic programming. *Science* 1966; 153(3731): 34–37.
31. De Boer PT, Kroese DP, Mannor S, et al. A tutorial on the cross-entropy method. *Ann Oper Res* 2005; 134(1): 19–67.
32. Szita I and Lörincz A. Learning tetris using the noisy cross-entropy method. *Neural Comput* 2006; 18(12): 2936–2941.
33. Wu L, Wen C, Zhou M, et al. A real-time monitoring method using random projection and k-nearest neighbor rule for batch process. *Int J Adv Robot Syst* 2017; 14(6): 1–6.