Indian Journal of Geo Marine Sciences
Vol. 50 (11), November 2021, pp. 978-987

COLREGs-compliant dynamic collision avoidance algorithm based on deep deterministic policy gradient

X L Xu[a], X L Zhou[b], P Cai[c] & Z Z Chu*[,a]

[a]School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai – 200 093, China

[b]Mechanical Engineering College, Beihua University, Jilin – 132 021, China

[c]Department of EE Data Architecture, Human Horizons Technology Co., Ltd., Shanghai – 200 082, China

*[E-mail: chu_zhenzhong@163.com]

Received 31 August 2021; revised 30 November 2021

In order to reduce collision avoidance accidents and improve the safety of ship navigation, a dynamic collision avoidance algorithm based on deep reinforcement learning is proposed in this paper. In order to avoid the fuzziness and uncertainty in the encounter process, the degree of risk is formulated to quantify the collision risk. International regulations for preventing collisions at sea (COLREGs) are quantified reasonably. Considering the factors of collision, position, speed, course and compliance with the COLREGs, the reward function of the algorithm is designed to ensure that the collision avoidance decision is safe and effective and meet the requirements of the COLREGs. Based on DDPG algorithm, the sample data processing mechanism is improved, the utilization rate of experience is improved, and the problems of long learning time and unstable training are solved. The navigation and collision avoidance for multiple ships are simulated respectively. The results show that this method can effectively avoid obstacle ships under the requirements of COLREGs, and it has good real-time performance and safety.

[Keywords: Collision avoidance, COLREGs, DDPG, Deep reinforcement learning, Unmanned surface vehicle]

Introduction

Various collision accidents during navigation are the core issues that researchers should pay attention to[1]. 89 – 96 % of collision accidents for ships are caused by human factors[2]. Therefore, the research on intelligent automatic collision avoidance method is of great significance to reduce human errors in accidents. The actual navigation environment is changing rapidly, which requires the Unmanned Surface Vehicle (USV) to have the ability to avoid sudden obstacles.

Abdallah *et al.*[3] used nonlinear optimization method to solve the collision avoidance problem of two ships. Ni *et al.*[4] carried out research on auxiliary decision-making. Cheng[5] introduced the fuzzy logic method into the control system of USV. Xu *et al.*[6] proposed a dynamic collision avoidance algorithm via layered artificial potential field with collision cone. Shen *et al.*[7] proposed an intelligent collision avoidance method for USVs based on deep Q-learning and A* algorithm. Further, Cheng & Zhang[8] proposed a concise obstacle avoidance algorithm with the deep Q-networks architecture. All the above studies are summarized in the Table 1.

There are still many challenges to be solved in the research of ship autonomous collision avoidance. Such as the wind, waves and current in the marine environment change greatly with time[9,10], many complex navigation scenes are difficult to be designed[8]. The collision avoidance action in unknown environment also needs to comply with the actual COLREGs constraints; analytical methods are difficult to solve this problem.

At present, with the rapid development of artificial intelligence technology, it has the characteristics of simple model, strong robustness and self-learning to adapt to the environment. Deep reinforcement learning technology[11] is widely used in the research of intelligent collision avoidance and path planning. This paper proposes an intelligent dynamic collision avoidance algorithm for multiple encounter scenarios for USVs. This algorithm is used to train agents, and the effectiveness of this method is verified by simulation of multiple ships and multiple encounter scenes.

Collision avoidance model

Ship motion is the complex six degree of freedom motion, which not only moves along the body-fitted

Table 1 — Performance summary of different collision avoidance methods

Current collision avoidance algorithms	Applicable characteristics	Aspects that need to be improved
Nonlinear optimization method	It takes rules as constraints and uses model predictive control to solve the optimization problem	The model is complex and it is difficult to establish an accurate and reliable model
Auxiliary decision-making method	It can avoid multiple dangerous target ships	It requires artificial expert experience, and it is difficult to establish a better decision for the complex multi ship encounter situation
Fuzzy logic method	It can avoid obstacles accurately and quickly in complex environment	It is only applicable to avoid static obstacles, and dynamic obstacles need to be considered
Artificial potential method	The algorithm is simple and efficient	The problem of local minima needs to be avoided
Deep Q-learning algorithm	It has strong learning ability and can solve complex problems.	It needs a large amount of calculation, and can only deal with the limited state and action space, but it can not deal with the infinite space effectively.
A* algorithm	It is simple, effective and accurate	The space requirement is too large and the optimal search path cannot be guaranteed when there are multiple minimum values
Velocity obstacle method	It can avoid static and dynamic obstacles at the same time	It is difficult to integrate the COLREGs

coordinate axis, but also rotates around the body-fitted coordinate axis. For different research objectives, ship motion can usually be reasonably simplified[12,13]. For most ships, the motion of the surge, sway and yaw is mainly concerned in the process of collision avoidance. The influence of heave, pitch and roll on collision avoidance is very small, that is, collision avoidance studies the motion of ships in the horizontal plane. Therefore, the plane motion of ship is considered in this paper. It is assumed that the USV can be regarded as a rigid body and the geodetic coordinate system is a typical inertial coordinate system.

Collision risk model

The dynamic model and relevant parameters of the ship used in this simulation are shown in literature[14]. In order to simplify the expression, the owner USV is abbreviated as OU and the target obstacle ship is abbreviated as TS. The coordinates of OU is (x_O, y_O), heading angle is φ_O, the speed is φ_O. The coordinates of TS is (x_T, y_T) and the heading angle is φ_T, the speed is v_T. The intersection angle of heading is C_T; where, $C_T = \varphi_T - \varphi_O$. If $C_T < 0$, then $C_T = C_T + 360°$. The relative azimuth angle of OU is θ_O, the relative azimuth angle of TS is θ_T. v_R is the relative velocity. The angle between the connecting line of the positions for two ships and the direction of v_R is α. It is shown in Figure 1. DCPA is the Distance of Closest Point of Approach (CPA) between OU and TS, $DCPA = R_T * sin\alpha$. TCPA is the Time to Closest Point of Approach (CPA), $TCPA = R_T * cos\alpha / v_R$[15]. When $TCPA < 0$, TS has passed the CPA for the two ships, it is no longer a threat to OU. When $TCPA > 0$, TS has not passed the CPA for the two ships, so the collision risk remains.

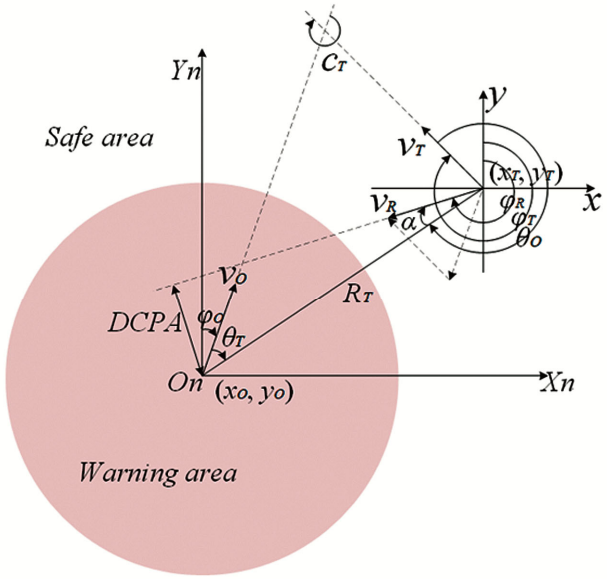

Fig. 1 — Description of motion variables for ships

In this paper, the working space of the USV is divided into safety area and warning area, which changes from time to time. In the safe area, the only task of the USV is to sail to the target, and the corresponding state is safe sailing state. The calculation formula given in Zhao[16] is adopted for the safety area:

$$R_{safe} = v_R\left(T_n + \frac{\Theta \times N \times SDA}{v_R}\right) \quad \ldots (1)$$

where, v_R is the relative velocity, T_n is the time required for our USV to turn 90° at full rudder, Θ is the coefficient, N is the visibility coefficient. SDA is the safe distance of approach of two ships[16].

$$SDA = (L_o + L_T) + 2 \times P + (L_o \times \pi/135 + L_T \times \pi/45) \ldots (2)$$

Among them, L_o L_T is the length of OU and TS, P is the variance of kalman filter.

When the obstacle ship is within the warning area, the corresponding state of USV is collision avoidance state. The USV needs to comply with the COLREGs and take corresponding collision avoidance actions. Then, in order to prevent it from entering the collision avoidance state again, the keeping state is set, that is, its heading angle is constant in this state until the USV successfully avoids the obstacle ship.

In order to quantify the degree of collision risk, it is defined as follows:

$$\Omega = min\left(R_{Ti}/R_{safe}\right) \quad \ldots (3)$$

R_{Ti} is the distance between OU and target ship T_i (i = 1,2,…,n), which is applicable to collision avoidance for multi ships, the collision risk degree is determined by calculating the Ω value of each TS. When Ω > 1, the USV is in the safe state and its task is to drive to the target. When Ω ≤ 1, the USV is in collision avoidance state, it shall give priority to the ship with the smallest R_{Ti}/R_{safe}.

COLREGs model

This algorithm only studies the case that OU is give-way ship. Φ is the parameter of COLREGs[17]. When the two ships do not constitute an encounter situation, $\Phi = 0$, other cases are defined as follows:

(1) Head-on ($\Phi = 1$): Both ships are directly in front of each other at an angle of ±5°. The relative azimuth of OU θ_O satisfies the condition $\theta_O \leq 5^\circ$ or $\theta_O \geq 355°$. The relative azimuth $\theta_T \leq 5^\circ$ or $\theta_T \geq 355°$. In Figure 2, OU's velocity is pointing due north, and TS's position falls in the yellow area, OU should turn right to avoid TS.

(2) Starboard crossing: i) Starboard crossing-small angle ($\Phi = 2$): The condition $\theta_T \leq 45°, 185° \leq C_T < 210°$ is satisfied. The position of TS falls in the pink sector area, and its velocity direction falls in the gray sector area. In this case, OU should turn right to avoid TS. ii) Starboard crossing-large angle ($\Phi = 3$): The condition $\theta_T \leq 112.5^\circ, 210° \leq C_T \leq 360°$ is satisfied, OU should turn left to avoid TS.

(3) Overtaking: When the encounter situation is overtaking, the condition $112.5^\circ \leq \theta_T \leq 247°$ is satisfied, and the velocity component of OU in TS's direction is larger than that of TS, that is $v_O > v_T * cosC_T$. i) Overtaking 1($\Phi = 4$): When α < 90°, DCPA>0, OU should turn right. ii) Overtaking 2($\Phi = 5$): When $270° < \alpha \leq 360°$, DCPA≤0, OU should turn left.

Algorithm design

Design of state space

The state space of this algorithm contains the linear velocity v_O, angular velocity ω_O, heading angle Φ. The distance between TS_i and OU R_T, azimuth angle of TS_i relative to OU θ_T, azimuth angle of OU relative to TS_i θ_0. Encounter situation of two ships is Φ. Distance between OU and target is R_{G_t}. That is $S_t = (v_O, \omega_O, \psi, R_T, \theta_T, \theta_o, \sigma_i, R_{G_t})$. All states in the algorithm are normalized, and the normalized range is [-1,1].

Selection of action set

USVs should have the ability to respond quickly to complex environments; USVs must realize three basic skills: acceleration, deceleration and turning. In order to ensure that the collision avoidance decision output by the algorithm has good operability, the thrust (acceleration and deceleration) and rudder angle (left and right turns) are taken as the action space of the agent, namely $a = [\tau_u, \delta]$.

Design of reward function

The reward function[18-20] in this paper includes position reward function, heading angle reward function and speed reward function.

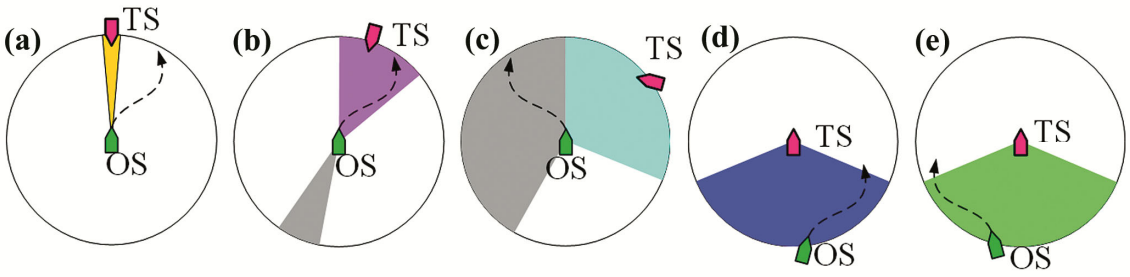

Fig. 2 — Encounter situation when OU is give-way ship: (a) Head-on, (b) Starboard crossing-small angle, (c) Starboard crossing-large angle, (d) Overtaking 1, and (e) Overtaking 2

Position reward function

In the unknown environment of an obstacle ship, the unmanned ship should not only avoid the obstacle ship, but also reach the target quickly. Therefore, the position reward function designed in this paper consists of two parts: collision avoidance item and target guidance item.

(1) Collision avoidance item: this paper establishes a collision avoidance item model based on Gaussian distribution. The relative distance between the USV and obstacle ship is R_{Ti}. The collision avoidance term is described by formula (4)

$$f_{collision} = \begin{cases} -\frac{1}{\sqrt{2\pi}} e^{-(1-\frac{R_{Ti}}{R_{safe}})^2} & , R_{Ti} \le R_{safe} \\ 0 & , R_{Ti} > R_{safe} \end{cases} \quad \ldots (4)$$

(2) Target guidance item: in order to enable the unmanned ship to avoid obstacles and quickly approach the target point, the target guidance item is shown in formula (5)

$$f_{target} = -\left(\frac{R_{T_t}}{R_{T_0}}\right)^2 \quad \ldots (5)$$

R_{T_t} represents the distance between the USV and the target at time t, and R_{T_0} is the initial distance between the USV and the target.

(3) Arrival item: When the distance between the USV and the target is less than R_{arrive}, it is considered that the USV reaches the target point. Therefore, arrive reward is defined as follows:

$$f_{arrive} = \begin{cases} 10, R_{G_t} \le R_{arrive} \\ 0, R_{G_t} > R_{arrive} \end{cases} \quad \ldots (6)$$

Heading angle reward function

In order to shorten the navigation time, it should avoid being in a state of deviation from the course, that is, ensure that it drives towards the destination. Therefore, the design of the heading reward function is

$$f_{heading} = \begin{cases} exp\left[-\frac{m}{(\psi_e)^2+(\dot{\psi}_e)^2}\right], if\ \psi_e > \psi_k \\ 0, otherwise \end{cases} \quad \ldots (7)$$

where, $\psi_e = \psi - \psi_d$, which is the deviation of the heading angle, and ψ_k is the threshold of deviation, m defines the parameters of the exponential function that relate to the convergence speed.

Speed reward function

(1) Destination item: In order to prevent the USV from moving too fast near the target point, the destination reward function is defined as:

$$f_{speed} = \begin{cases} \frac{-|v_o|}{v_{omax} R_{OG}} & , R_{G_t} \le \mathrm{R}_v \\ 0 & , R_{G_t} > \mathrm{R}_v \end{cases} \quad \ldots (8)$$

where, $\boldsymbol{v_o}$ is the speed of USV, $\boldsymbol{v_{omax}}$ is the maximum speed of USV, and R_{Tt} is the distance between USV and the target point. We define that when the distance between the USV and the destination is R_v, the USV begins to decelerate.

(2) Sway item: In order to make the USV sail to the destination at the heading angle towards the target, the speed of sway should not be too large. u and v are the speeds of surge and sway respectively. Therefore, the sway function is

$$f_{sway} = \begin{cases} -\varsigma, if\ |u|<|v| \\ 0, otherwise \end{cases} \quad \ldots (9)$$

where, ς is a positive.

COLREGs function

In addition to ensuring the safety and effectiveness of collision avoidance, the reward function for whether the collision avoidance decision meets the COLREGs is designed.

$$f_{COLREGs} = \begin{cases} 0, Compliance\ with\ COLREGs \\ -\varrho, Against\ COLREGs \end{cases} \quad \ldots (10)$$

where, ϱ is positive, which is the punishment for violating COLREGs.

Combined with the weight $\lambda_{collision}$, λ_{target}, λ_{arrive}, $\lambda_{heading}$, λ_{speed}, λ_{sway}, $\lambda_{COLREGs}$, the comprehensive expression of the reward function is as follows:

$$R = \begin{bmatrix} \lambda_{collision} \\ \lambda_{target} \\ \lambda_{arrive} \\ \lambda_{heading} \\ \lambda_{speed} \\ \lambda_{sway} \\ \lambda_{COLREGs} \end{bmatrix}^T \begin{bmatrix} f_{collision} \\ f_{target} \\ f_{arrive} \\ f_{heading} \\ f_{speed} \\ f_{sway} \\ f_{COLREGs} \end{bmatrix} \quad \ldots (11)$$

Improved sample data processing mechanism

Deep Deterministic Policy Gradient (DDPG) algorithm[21-23] is an algorithm based on Actor-Critic framework. Because the experience of agent is not equally important for the learning of network model, the experience with high immediate return[24] and TD-error[25] are more important than other experiences, and these experiences should be used more efficiently. Based on DDPG, this paper improves the sample data processing mechanism, which can make the experience take into account the immediate return mechanism and TD-error priority mechanism as much as possible.

Firstly, TD-error needs to be calculated, and its formula is as follows:

$$\delta_t = r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) - Q^{\pi}(s_t, a_t) \quad \ldots (12)$$

where, $r_{t+1} + \gamma Q^{\pi}(s_{t+1}, a_{t+1})$ is the pre-estimated state action value function, $Q^{\pi}(s_t, a_t)$ is the state action value function at the current time, indicates that the agent follows the policy π, starts in the current state s_t and takes action a_t.

Secondly, the priority needs to be determined, and its formula is as follows:

$$Y_i = r_t + \varepsilon Y_j = |\delta_t| + \varepsilon \quad \ldots (13)$$

where, Y_i and Y_j are the priority based on immediate return priority mechanism and the priority based on TD-error priority mechanism respectively; r_t is the immediate return of experience; ε is a positive constant to ensure that each transfer message has a non-zero priority.

Finally, according to order of the priority, Y_i and Y_j are arranged from large to small, and $rank(i)$ and $rank(j)$ are obtained, and also get the empirical composite average ranking.

$$u(k) = \frac{rank(i)+rank(j)}{2} \quad \ldots (14)$$

The priority of the composite is calculated:

$$Y_k = [1/u(k)]^{\beta} \quad \ldots (15)$$

where, the parameter β represents the degree of priority used by the algorithm, and its range value is [0,1]. When, $\beta = 0$, it represents uniform sampling.

The probability of sampling is

$$P_k = \frac{Y_k}{\Sigma_n Y_n} \quad \ldots (16)$$

Where, n is the number of experience.

Design of network

The network structure has two hidden layers. The nodes of each hidden layer are 400 and 300, respectively, and the output action matrix. The state matrix is input into the critic network, which has 400 nodes in the second layer and 300 nodes in the third layer. The action matrix is also input to the critic network. There are 300 neuron nodes in the second layer. The neurons in the third layer of the network input by the state space matrix and the neurons in the second layer of the network input by the action matrix are combined for linear transformation and input to the neuron nodes in the fourth layer. There are 300 neuron nodes in this layer, and finally the value of the action is output. The connection mode between all neuron nodes of the network is full connection mode, and the network structure diagram is shown in Figure 3. Based on the above definition, the pseudo code is shown in Table 2.

Simulation

Real ship test has the disadvantages of high risk, long debugging cycle and high cost, and it takes a long time to build the algorithm test platform. In order to reduce the waste of time, make researchers focus on the algorithm, and test the effectiveness of the algorithm conveniently and quickly. In this paper, the mathematical model of ship motion with three degrees of freedom is used to design an algorithm simulation system which can provide an algorithm similar to the actual situation at sea. This is helpful for researchers to better test the proposed collision avoidance control algorithm and apply it to engineering practice.

In this part, the effectiveness of the algorithm is verified by simulation. The computer configuration is

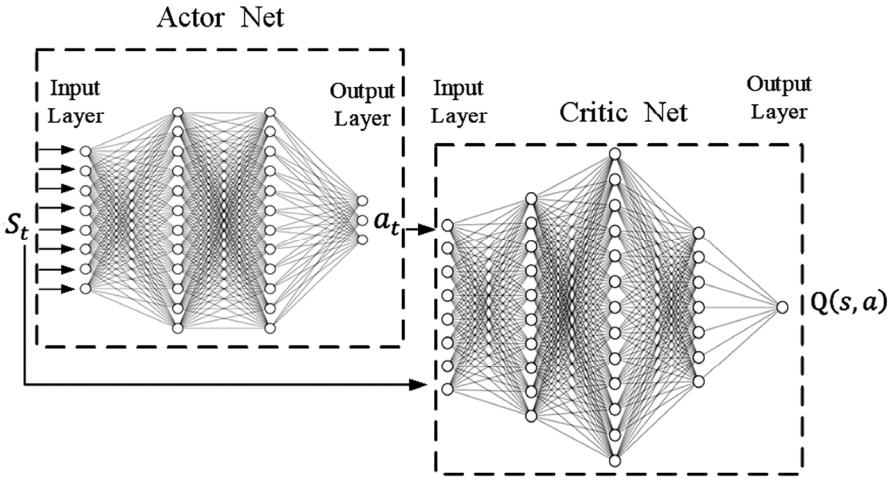

Fig. 3 — Structure of network

Table 2 — The pseudo code of the algorithm

Initialize experience buffer, initialize the parameters of Online Policy Net $\mu(s;\theta^{\mu})$ and Online Q Net $Q(s,a;\theta^{Q})$, assign the parameter to Target Policy Net and Target Q Net， that is $\theta^{\mu'} \leftarrow \theta^{\mu}$, $\theta^{Q'} \leftarrow \theta^{Q}$.
Initialize a random normal distribution N with var^2 variance, which is used to interfere with actions to explore the environment.
For episode=1:M
Get the initial state s_1
For step=1:T
According to the existing strategies and explored interference, input s_t and output a_t, $a_t \sim N[\mu(s_t/\theta^{\mu}), var]$
The agent makes action a_t, obtains return r_t and subsequent state s_{t+1}, and calculates TD-error
Sort the experience according to priority $Y_i = r_t + \varepsilon$ from large to small to get $rank(i)$
Sort the experience according to priority $Y_j = |\delta_t| + \varepsilon$ from large to small to get $rank(j)$
Make a compound average ranking of experience, get $u(k) = \frac{rank(i)+rank(j)}{2}$ and calculate the priority $Y_k = [1/u(k)]^{\beta}$ of experience
Empirical sampling probability $P_k = \frac{Y_k}{\sum_n Y_n}$, where n is the number of experiences. m experiences are sampled with probability P_k and stored in experience buffer 1
Sample experience from experience buffer 1 for network learning
Calculate the gradient obtained by time difference of experience, update the network parameters
Calculate the policy gradient ∇ and update the parameters of the actor network $\theta^{\mu} = \theta^{\mu} + \alpha\nabla$, where α is the learning rate
Every J episodes, the network parameters of the actor network are assigned to the actor target network $\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$.
Every K episodes, the network parameters of the critic network are assigned to the critic target network $\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$.
End for
End for

as follows: Intel Core i7-9750H six core processor, NVIDIA GTX1660Ti 6 GB graphics card, 16 GB DDR3L memory. The optimization of neural network parameters uses Adam optimizer, learning rate of actor network is $\alpha^{\mu} = 1 \times 10^{-4}$, learning rate of critic network is $\alpha^{Q} = 1 \times 10^{-3}$. The discount factor is $\gamma = 0.99$, the soft replacement coefficient $\tau = 0.001$. The parameters are updated every 10 episodes, the hidden layer of the neural network uses modified nonlinear elements, the final output layer of the actor network is tanh layer. In Gaussian distribution $\sigma = 0.2$, the maximum number of time steps in each episode is 500 (that is, when the time step reaches 500, the episode ends). The size of the experience pool is 300000 samples, and the size after sampling is 10000. The simulation environment is a rectangular area of 600×500 m. L_o, L_T is all 1.8 m, which is the length of our USV and the target ship, respectively. v_{Omax}, a_{max}, ω_{max}, α_{max} are the maximum linear velocity, linear acceleration, angular velocity and angular acceleration of our USV, which is 3.5 m/s, 0.4 m/s^2, 0.2 rad/s, 0.05 rad/s^2, respectively. v_{Tmax} is the maximum velocity of the target ship, which is 3.5 m/s.

The task of path planning

It is trained with 1000 episodes, the maximum number of steps in each episode is 600, and the network is saved every 10 episodes. It can be seen in Figure 4 that at the beginning of the training, the USV does not know how to navigate, and it turns around near the starting point. After training 50 episodes, the initial sailing direction of the USV is wrong, but then sails towards the target, but does not reach the target accurately in the end. After training 200 episodes, the USV can drive to the target, but its track fluctuates greatly. After training 1000 episodes, the USV have learned to sail towards the target, compared with the previous track, it is the smoothest and shortest.

In order to compare the learning efficiency and oscillating amplitudes of the improved DDPG algorithm (improved sample data processing mechanism) and the original DDPG algorithm more conveniently, the two algorithms are both trained for 20 times. The results of the first 1000 episodes are extracted, and the average cumulative reward of the two methods is calculated. The results are shown in Figure 5. One can clearly see from the figure that initially the USV is in the exploratory learning phase and does not map the relationship between state and action very well, resulting in a relatively low average reward for the USV. Figure 5 shows the improved algorithm before 80th episode has almost the same oscillating amplitude extent as the DDPG algorithm. However, with increased training, the USV can successfully map the relationship between state and optimal action and the average reward obtained gradually increases. In the later stages of training, the USV effectively uses the learned knowledge to reach the goal position safely, resulting in an overall positive return. The learning efficiency of the improved algorithm is higher, because the average score of the

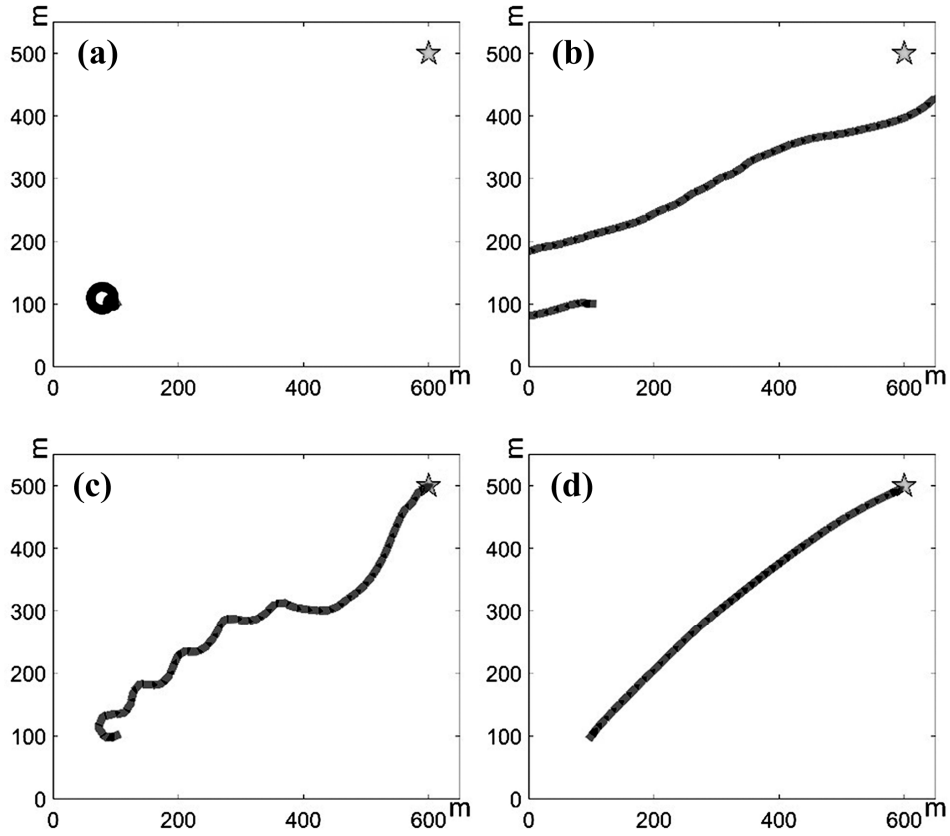

Fig. 4 — Training effect of path planning task: (a) Test after 1 training episode, (b) Test after 50 training episodes, (c) Test after 200 training episodes, and (d) Test after 1000 training episodes

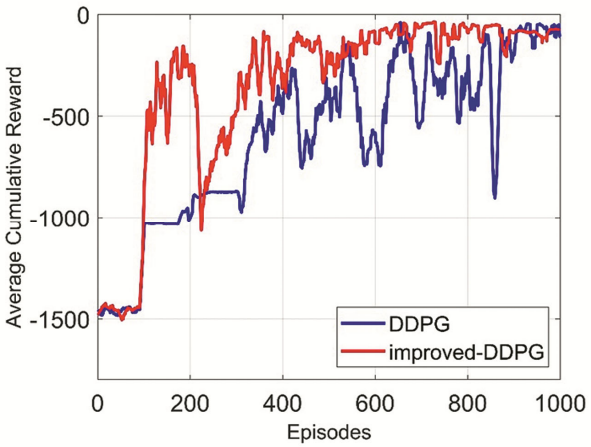

Fig. 5 — The comparison part of average cumulative reward

improved algorithm is better than the DDPG algorithm most of the time. After about 220th episodes, the reward of the DDPG algorithm is still volatile, while the improved algorithm is still in the process of continuous optimization. Accordingly, the improved algorithm overcomes problems associated with oscillating amplitudes and low learning efficiency.

Dynamic collision avoidance of multiple USVs

Combined with the collision avoidance task, 6 obstacle ships with random initial velocity and position. When the obstacle ship reaches the boundary of the designated area, it will sail in the form of specular reflection.

Figure 6(a) represents the trajectory of all ships. TS 1, 2, 3, 4, 5 and 6 are generated at the same time. The USV detects TS 6, and the encounter situation is overtaking 1. As shown in Figure 6(b), in step 21th, the USV enters the collision avoidance state, and TS 6 is at the center of the sector. The USV takes a right turn to avoid and escapes the danger in step 24th. Then the USV detects TS 3, but it does not pose a threat to the USV, so it does not need to be considered. The USV detects TS 1 during sailing, and the encounter situation is overtaking 2. As shown in Figure 6(c), in step 71th, the USV enters the collision avoidance state, and the USV takes a left turn to avoid. In step 75th, it enters keeping state. In step 84th, the USV returned to safe sailing state and sails to the target. In the figure, the USV avoided two obstacle ships, its collision avoidance actions

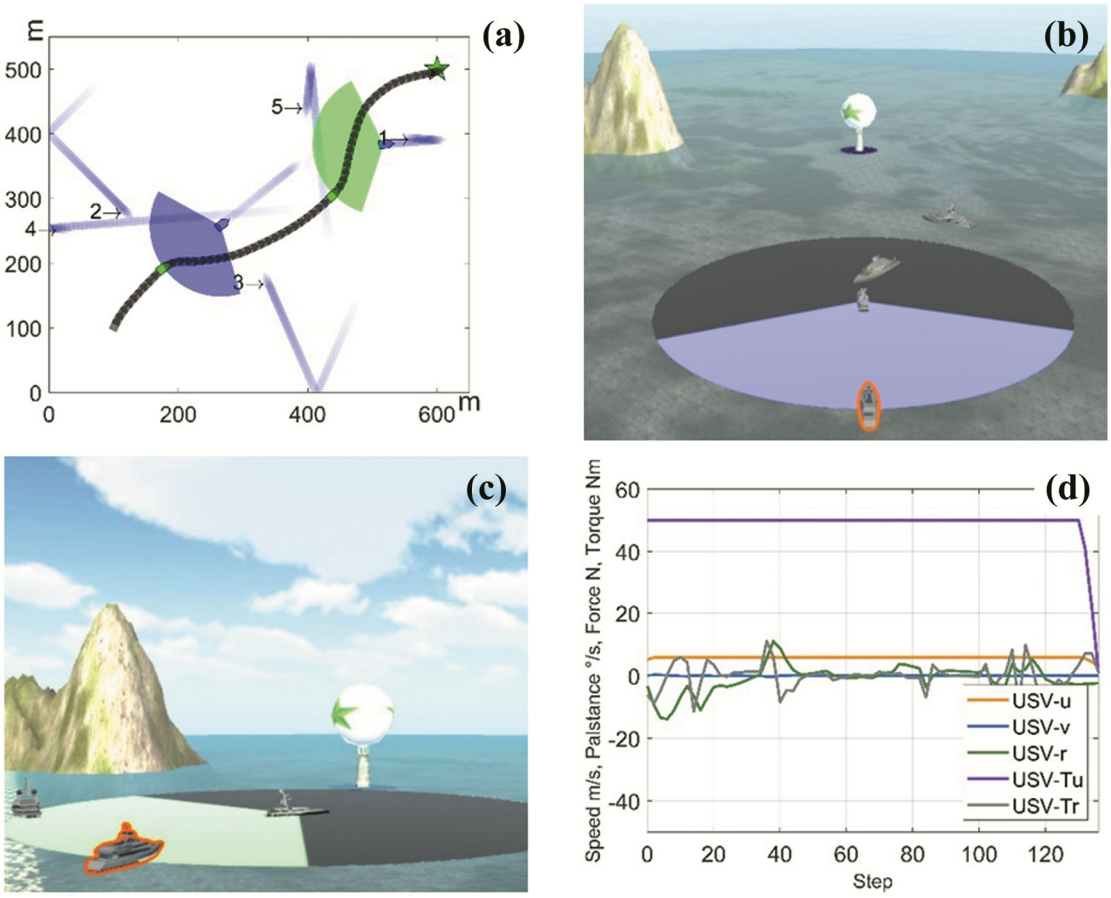

Fig. 6 — Encounter scenario 1: (a) Trajectory of the OU and TS, (b) The 3D view of avoiding for the first time, (c) The 3D view of avoiding for the second time, and (d) The curve of velocity and thrust

complied with the COLREGs, and finally reached the target safely, with a total of 118 steps. The velocity and thrust curves of USV during the whole collision avoidance process are shown in Figure 6(d).

Through the verification of the encounter situation, when the encounter situation of two ships is overtaking 1 and overtaking 2, the algorithm has the ability to avoid dynamic obstacle ships according to COLREGs.

In Figure 7, Figure 7(a) is the trajectory of all ships. In step 21th, OU enters the collision avoidance state and forms the starboard crossing-small angle situation with TS 6. In step 21, OU enters the keeping state, and in step 31th, it enters the safe sailing state. In step 71th, the USV detects TS 1, and the encounter situation is starboard crossing-large angle. As shown in Figure 7(c), OU took a left turn to avoid, successfully avoided the TS 1 and successfully reached the target, with a total of 119 steps in the whole process.

Through the verification of the encounter situation, when the encounter situation of the two ships are starboard crossing-small angle and starboard crossing-large angle, the algorithm has the ability to avoid dynamic obstacle ships according to COLREGs.

The current study defines that if the USV complies with the COLREGs and finally reaches the target, the voyage is regarded as a success, and otherwise it is a failure. We extracted the success rate of the first 5000 episodes of training, which includes five situations. The final success rate of each situation reaches 95 %. It can be seen that this algorithm can control USV to plan the path reasonably and comply with the COLREGs to avoid dynamic obstacle ships.

Above all, we can see that this algorithm can avoid multiple burst dynamic ships and complies with COLREGs, which has the ability of dynamic collision avoidance.

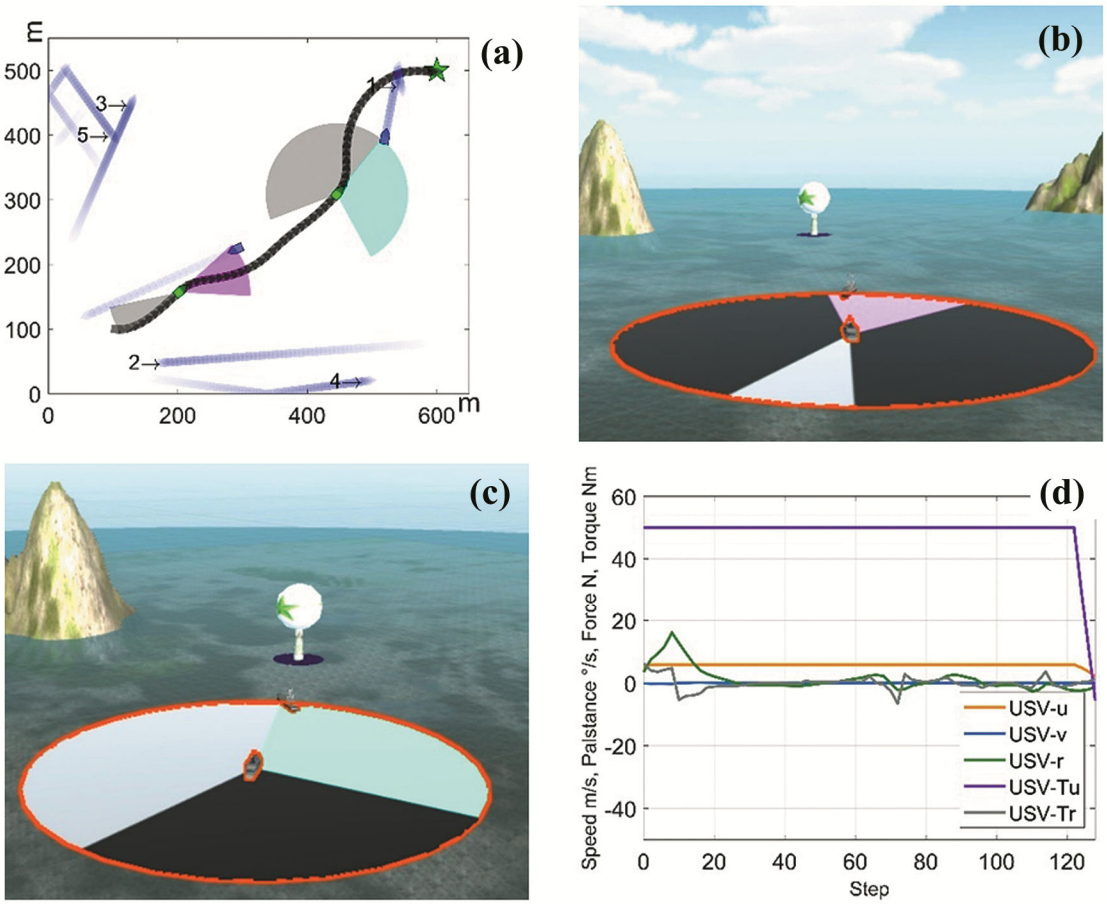

Fig. 7 — Encounter scenario 2: (a) Trajectory of the OU and TS, (b) The 3D view of avoiding for the first time, (c) The 3D view of avoiding for the second time, and (d) The curve of velocity and thrust

Conclusion

A dynamic collision avoidance algorithm under COLREGs constraints base on DDPG is proposed. By analyzing and quantifying the collision risk and collision avoidance time, the calculation method of risk degree is formulated, and COLREGs is divided in detail and quantified. According to the real-time navigation information obtained between ships, the state set and action set of avoidance process are designed to ensure the integrity and computability of navigation information in the input neural network. Combined with the requirements for safety and compliance with COLREGs in avoidance decision-making, reward function is design. Based on DDPG algorithm, the sample data processing mechanism is improved to improve the utilization of experience. The deep neural network is used to train the agent. After training 2000 episodes, the USV learned to sail towards the target, compared with the previous track, it is the smoothest and shortest. Then multi ship encounter scene is simulated, 6 obstacle ships with random initial velocity and position is set. The first 5000 episodes of training are extracted and the final success rate reaches 97 %. It is verified that the algorithm has the characteristics of real-time and security when avoiding multi burst ships.

This paper studies the collision avoidance algorithm in open sea area. For more complex or dangerous navigation sea area, USVs need to navigate according to special requirements. At present, there is little research on collision avoidance in this aspect, and there are some challenges. However, it is a problem that must be solved by USVs (especially unmanned transport ships), which is also the direction of our follow-up efforts.

Acknowledgements

This work is partly supported by the National Natural Science Foundation of China under Grant U2006228, 52171313. This work is also supported by

the financial support of University of Shanghai for Science and Technology (H-2021-304-044).

Conflict of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Author Contributions

XLX: Ideas, evolution of overarching research goals and aims, creation of models; PC: Designing computer programs, implementation of the computer code and supporting algorithms; XLZ: Presentation of the writing work; and ZZC: Application of statistical, other formal techniques to analyze or synthesize study data.

References

1 Kim D, Hirayama M & Okimoto, Ship collision avoidance by distributed tabu search, *Trans Nav: Inter J Mar Navig Saf Sea Transp*, 9 (1) (2015) 23-29. *doi:* 10.12716/1001.09.01.03

2 Statheros T, Howells G & Maier K, Autonomous ship collision avoidance navigation concepts, *Technol Tech*, 61 (1) (2008) 129-142. *doi:* 10.1017/S037346330700447X

3 Abdelaal M & Hahn A, Nmpc-based trajectory tracking and collision avoidance of unmanned surface vessels with rule-based colregs confinement, In: *2016 IEEE Conference on Systems, Process and Control (ICSPC)*, 12 (2016) pp. 23-28. *doi:* 10.1109/SPC.2016.7920697

4 Ni K, Liu Z, CAI Y & Wang X, Ship collision avoidance decision aids based on genetic algorithm, *J Shanghai Maritime Univ*, 38 (1) (2017) 12-15.

5 Chen Y, Sun Z, Huang Y & Zhang W, Fuzzy categorical deep reinforcement learning of a defensive game for an unmanned surface vessel, *Int J Fuzzy Syst*, 21 (2) (2019) 592-606. *doi:* 10.1007/s40815-018-0586-0

6 Xu X, Pan W, Huang Y & Zhang W, Dynamic collision avoidance algorithm for USVs via layered APF with collision cone, *J Navig*, 73 (6) (2020) 1306-1325. *doi:* 10.1017/S0373463320000284

7 Shen H, Guo C, Li T & Yu Y, An intelligent collision avoidance and navigation approach of unmanned surface vessel considering navigation experience and rules, *J Harbin Eng Univ*, 39 (6) (2018) 998-1005

8 Chen Y & Zhang W D, Concise deep reinforcement learning collision avoidance for underactuated unmanned marine vessels, *Neurocomputing*, 272 (5) (2018) 63-73. *doi:* 10.1016/j.neucom.2017.06.066

9 Wang Z, Yang S, Xiang X, Vasilijević A, Mišković N, *et al.*, Cloud-based mission control of USV fleet: Architecture, implementation and experiments, *Control Eng Pract*, 106 (2021) p. 104657. *doi:* 10.1016/j.conengprac.2020.104657

10 Chu Z, Chen Y, Zhu D & Zhang M, Observer-based adaptive neural sliding mode trajectory tracking control for remotely operated vehicles with thruster constraints, *Trans Inst Meas Control*, 43 (13) (2021) 2960-2971. *doi:* 10.1177/01423312211004819

11 Chu Z, Sun B, Zhu D, Zhang M & Luo C, Motion control of unmanned underwater vehicles via deep imitation reinforcement learning algorithm, *IET Intell Transp Syst*, 14 (7) (2020) 764-774. *doi:* 10.1049/iet-its.2019.0273

12 Xiang G & Xiang X, 3D trajectory optimization of the slender body freely falling through water using Cuckoo Search Algorithm, *Ocean Eng*, *235* (2021) p. 109354. *doi:* 10.1016/j.oceaneng.2021.109354

13 Zhang J, Xiang X, Lapierre L, Zhang Q & Li W, Approach-angle-based three-dimensional indirect adaptive fuzzy path following of under-actuated AUV with input saturation, *Appl Ocean Res*, 107 (2021) p. 102486. *doi:* 10.1016/j.apor.2020.102486

14 Xu X, Lu Y, Liu X & Zhang W, Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs, *Ocean Eng*, 217 (2020) p. 107704. *doi:* 10.1016/j.oceaneng.2020.107704

15 Xiu B & Guang J, Research on the domain model of ship collision avoidance actio, *J Dalian Marit Univ: Nat Sci Edt*, 29 (1) (2020) 9-12.

16 Zhao W, Collision avoidance and maritime safety of ships, *Dalian Marit Univ*, 2016.

17 Imo, Convention on the International Regulations for Preventing Collisions at Sea (COLREGs), *London*, 1972.

18 Wierstra D, Legg S & Hassabis D, Human-level control through deep reinforcement learning, *Nature*, 518 (7540) (2018) 529-533. *doi:* 10.1038/nature14236

19 Qiao G, Leng S, Maharjan S, Zhang Y & Ansari N, Deep reinforcement learning for cooperative content caching in vehicular edge computing and networks, *IEEE Internet Things J*, 7 (1) (2019) 247-257. *doi:* 10.1109/JIOT.2019.2945640

20 Adam S, Busoniu L & Babuska R, Experience Replay for Real-Time Reinforcement Learning Control, *IEEE Trans Syst Man Cybern*, 42 (2) (2021) 201-212. *doi:* 10.1109/TSMCC.2011.2106494

21 Qiu C, Hu Y, Chen Y & Zeng B, Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications, *IEEE Internet Things J*, 6 (5) (2019) 8577-8588. *doi:* 10.1109/JIOT.2019.2921159

22 Hu J, Zhang H & Song L, Reinforcement learning for decentralized trajectory design in cellular UAV networks with sense-and-send protocol, *IEEE Internet Things J*, 6 (4) (2018) 6177-6189. *doi:* 10.1109/JIOT.2018.2876513

23 Qiu C, Hu Y, Chen Y & Zeng B, Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications, *IEEE Internet Things J*, 6 (5) (2019) 8577-8588. *doi:* 10.1109/JIOT.2019.2921159

24 Xu J, Hou Z, Wang W, Xu B, Zhang K, *et al*., Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks, *IEEE Trans Industr Inform*, 15 (3) (2018) 1658-1667. *doi:* 10.1109/TII.2018.2868859

25 Kim M, Han D, Park J & Kim J, Motion planning of robot manipulators for a smoother path using a twin delayed deep deterministic policy gradient with hindsight experience replay, *Appl Sci*, 10 (2) (2020) p. 575. *doi:* 10.3390/app10020575