A novel ship collision avoidance awareness approach for cooperating ships using multi-agent deep reinforcement learning

Chen Chen[1], Feng Ma[2*], Xiaobin Xu[3], Jin Wang[4]

School of Computer Science & Engineering, Wuhan Institute of Technology, Wuhan, China[1]

Intelligent Transportation System Center, Wuhan University of Technology, Wuhan, China[2]

School of Automation, Hangzhou Dianzi University, Hangzhou, China[3]

School of Engineering, Liverpool John Moores University, UK[4]

Abstract： Ships are special machineries with large inertias and relatively weak driving forces. To simulate the manual operations of manipulating ships with Artificial intelligence (AI) is quite a difficult job, in which how to avoid collisions in crowded waters may be the most challenging task. This research proposes a cooperative collision avoidance approach for multiple ships using a multi-agent deep reinforcement learning (MADRL) algorithm. Each ship is modelled as an individual agent, controlled by a Deep Q-Network (DQN) method and described by a dedicated ship motion model. Each agent observes the state of itself and other ships as well as the surrounding environment. Then, agents analyse the navigation situation and make motion decisions respectively. In particular, specific reward function schemas are designed to simulate the degree of cooperation among agents. According to the International Regulations for Preventing Collisions at Sea (COLREGs), three typical scenarios of simulation are established to validate the proposed approach, which are head-on, overtaking and crossing. After sufficient training, the ship agents were capable of avoiding collisions under their cooperation in narrow crowded waters.

Keywords: Multi-agent Deep Reinforcement Learning (MADRL); Deep Q-Network (DQN); Maritime Autonomous Surface Ships (MASS); Multi-ship Cooperative Collision Avoidance; Reward Function

Highlights:

[1] Novel approach for multiple ships collision avoidance using a MADRL algorithm.

[2] Novel method to model different cooperative relationships among multiple ships.

1 Introduction

In 2018, the 99th session of the Maritime Safety Committee (MSC) of the International Maritime Organization (IMO) defined the objectives, concept, degrees of autonomy, methodology

and work plan of maritime autonomous surface ships (MASS) (Fan et al., 2020). MASS can offer a perfect solution to the dilemma of modern shipping industry, while safety is still the primary concern. Intelligent collision avoidance is a key ingredient for MASS, involving hazard identification, collision avoidance and manoeuvring decision-making. However, in formal systems research, ship collision avoidance methods are usually applicable on the condition that only the "own ship" is intelligent. This means that only the own ship makes decisions, and other ships are regarded as obstacles that always keep their motion status. Nevertheless, achieving collision avoidance is actually the result of cooperative behaviours by multiple ships. Therefore, it is necessary to simulate the actions of multi-ship cooperative collision avoidance.

In this research, the Multi-agent Deep Reinforcement Learning (MADRL) is used to address the problem of intelligent collision avoidance and cooperation modelling. In general, reinforcement learning (RL) can be considered as a method of mapping from environment to appropriate behaviours. An agent seeks a promising action by maximizing the corresponding value function, which is similar to the profit and loss consideration or balance of manual works. On this basis, cooperative collision avoidances among multiple ships can be modelled as the profit and loss allocation of decision-making among multiple RL agents. Moreover, navigation conventions and personalities of ship operators can be described as different reward functions in terms of collisions, cooperation and competition. After sufficient training, the artificial consciousness of ship collision avoidance is capable of making safe decisions and control, even if there is no cooperation between ship agents at all.

In order to achieve this goal, a novel multi-ship collision avoidance approach based on MADRL is proposed which takes the ship manoeuvrability into consideration in this research. The paper is organized as follows. Relevant references are briefly reviewed in Section 2. A novel MADRL-based approach is put forward in Section 3. Through a simulation case study, the approach is validated in Section 4. Section 5 concludes this study and provides directions for future research.

2 Literature review

2.1 Ship collision avoidance methods

In general, artificial ship collision avoidance mainly depends on ship position and motion relationship to determine the collision avoidance opportunity and make collision avoidance decisions using methods such as a ship domain-based approach (Szlapczynski and Szlapczynska, 2016), time

to the closest point of approach (TCPA) and distance at closest point of approach (DCPA) (Denker et al., 2016). Autonomous navigation and collision avoidance of an Unmanned Surface Vessel (USV) depends on automatic sensor fusion methods (Blaich et al., 2015; Chen et al., 2013; Eriksen et al., 2018; van der Sande and Nijmeijer, 2017), which are capable of discovering static and dynamic obstacles.

Autonomous collision avoidance decision-making of USV draws on the methods of robot collision avoidance. A* and B-spline (Wang et al., 2017), APF (Lazarowska, 2018), an ant colony optimization method (Song, 2014) are suggested for obstacle detection and avoidance. The evidential reasoning theory was used to evaluate collision risks (Zhao et al., 2016) to make collision avoidance decisions. The anti-collision system of USV was built on a neural-evolutionary fuzzy algorithm (Szymak and Praczyk, 2012) and an evolutionary neural network (Praczyk, 2015).

With the development of RL, Chen et al. (Chen et al., 2019) proposed an approach of operating a vessel based on Q-learning for smart ships without any input from human experiences. Zhao and Roh (Zhao and Roh, 2019) put up with an obstacle avoidance model based on deep reinforcement learning (DRL). Chen et al. (Chen et al., 2020) made use of Deep Q-Network (DQN) to control a cargo ship directly, while avoiding collisions, keeping its position in the middle of the route as much as possible.

Traditional multi-agent collaboration problems are generally addressed by distributed constraint optimization (DCOP) (Leite et al., 2014). DCOP refers to a distributed constrained optimization problem that decision variables and mathematical constraints are distributed in different individuals. Li et al. (Li et al., 2019) applied this method to multi-ship collision avoidance, predicting ship trajectories based on ship dynamics, giving different candidate rudder angles, evaluating the collision risk by each rudder angle, and then using optimization strategies to find the most effective collision avoidance plan for ships. However, in this research all ships are controlled by a system decision module, and each agent has no independent decision-making intelligence.

Collective motions are widespread in nature, such as the concerted movements of fish, ants, birds, *etc*. A number of relevant studies applied swarm control to multi-robot, unmanned vehicle formation control, crowd evacuation, *etc*. There are many models about collective motions, while a leader-follower model is one of the most widely applied. This method adopts a centralized control structure, while one agent is the leader and the other agents are followers. The leader-follower method

is widely applied to design formation control for USVs (Zhou et al., 2015; Sun et al., 2018). The individual intelligence in swarm dynamics is simple. Zhou et al. (Zhou et al., 2019) made use of the DRL for USV formation path planning. However, this kind of formation control is often very different from the real multi-ship collision avoidance, since any single agent in this research does not realize independent decision-making.

2.2 Multi-agent deep reinforcement learning (MADRL)

With the success of DRL, it has been applied to multi-agent systems, and MADRL has been developed. MADRL is a stochastic game based Markov decision-making process (Foerster et al., 2016), which can be described as a tuple $(n, S, A_1, \dots, A_n, T, \gamma, R_1, \dots, R_n)$, where n is the number of agents, S is a finite set of environment states, $A = A_1 \times \dots \times A_n$ is the collection of action sets, $A_1, \dots, A_n$, one for each agent in the environment. T is the state transition probability function, controlled by the current state S and one action from each agent: $T: S \times A_1 \times A_2 \dots \times A_n \to S'[0,1]$. R is the return function, R_i is the reward of agent i in state S after taking joint action in state S'.

In the multi-agent case, the state transitions are the result of the joint action of all the agents. The policies $M_i: S \times A \to M$, form the joint policy M together. Accordingly, the reward for each agent is:

$$R_i^M = E[R_{t+1} | S_t = s, A_t, i = a, M] \tag{1}$$

The Bellman equation is

$$v_i^M(s) = E_i^M[R_{t+1} + \gamma V_i^M(S_{t+1}) | S_t = s] \tag{2}$$

$$Q_i^H(s, a) = E_i^M[R_{t+1} + \gamma Q_i^M(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \tag{3}$$

According to different rewarding schemes, different games can be created, such as a fully cooperative one, a fully competitive one and transition between cooperation and competition, which is also called mixed games.

Collaborative agents performed better than an independent agent through experiments (Tan, 1993). Tampuu et al. (Tampuu et al., 2017) extended the DQN algorithm to multi-agent environments in the Pong videogame, with the two agents controlled by independent DQN. By manipulating reward rules, they demonstrated how competitive and collaborative behaviours emerge. MADRL has reached the level of professional players in the first person multi player game and cooperated with other real players (Jaderberg et al., 2019).

As discussed above, this research uses the MADRL to realize the cooperative collision avoidance awareness of multiple ships. Each ship is regarded as an agent which observes the state of itself and the others as well as the surrounding environment, judges the navigation situation and makes decisions respectively in the multi-ship encounters. In addition, different agent reward function schemas are designed to simulate the states of cooperation mode, such as a fully competitive one, a fully cooperative one, and transition between cooperation and competition. Finally, repeated training is carried out in different encounter scenarios to realize the cooperative collision avoidance awareness among multiple ships.

3 A proposed approach

3.1 Mathematical modelling of ship motions

Ship manoeuvring motions are used to forecast the state changing of a ship when it takes specific action, making the training environment consistent with the real world. In this area, ship manoeuvring motions are generally presented with a standard three degree-of-freedom MMG model (Chen et al., 2020) that considers surge, sway, and yaw for simplification. Fig. 1 illustrates the static earth-fixed $o_0 - x_0y_0z_0$ and the dynamic body-fixed $o - xyz$ coordinate systems. The origin of $o - xyz$ locates at the middle of the ship O. x-, y - and z- axes are positive to the bow of a ship, the starboard of the ship, and downwards of the water surface xy respectively. Assuming that the ship presented in Fig. 1 is maneuvering at surge speed u and sway speed v, the ship speed is $V = \sqrt{u^2 + v^2}$. The heading angle is ψ. The ship is turning with a rudder angle δ at yaw rate $r=\dot{\psi}$.

The MMG model used in this research describes the hydrodynamic force and the moment in three aspects: hull, propeller and rudder. The motion equations are expressed as follows:

$$\begin{aligned}(m + m_x)\dot{u} - (m + m_y)vr - x_G m r^2 &= X_H + X_P + X_R \\ (m + m_y)\dot{v} + (m + m_x)ur + x_G m\dot{r} &= Y_H + Y_R \\ (I_Z + x_G^2 m + J_Z)\dot{r} + x_G m(\dot{v} + ur) &= N_H + N_R\end{aligned} \tag{4}$$

where subscripts H, P, and R denote hull, propeller, and rudder, respectively, with force (X and Y) and moment (N). m is the ship mass, m_x and m_y are added mass due to motions in surge and sway directions. $\dot{u}$, $\dot{v}$ and $\dot{r}$ are surge, sway and yaw acceleration, and I_z, J_z are the moments of inertia, where $I_Z \approx (0.25L_{pp})^2$ m. If not particularly specified, the parameters, such as velocity (u, v, r, and V), acceleration ($\dot{u}$, $\dot{v}$, and $\dot{r}$), force (X and Y), and moment (N) are defined on or around midships.

According to the MMG model, the trajectory and status of a ship can be predicted under different initial conditions (positions, speeds, rudder angles and different angular velocities).

3.2 MDP of multi-ship cooperative collision avoidance

For the multi-ship cooperative collision avoidance, each ship is an agent capable of observing environment, collecting data and autonomous learning. Its state space is formulated on the current rudder angle, position, speed and heading of each ship, which can be represented by

$$S = [angle_1, x_1, y_1, v_1, \psi_1, angle_2, x_2, y_2, v_2, \psi_2, \cdots, angle_n, x_n, y_n, v_n, \psi_n] \quad (5)$$

where n is the number of the agents, $\boldsymbol{x}$ is the X-coordinate, $\boldsymbol{y}$ is the Y-coordinate and ψ is the heading of the ship. For simplification of the model and computing, the speed of the simulated ship is set to be constant v.

This research defines the action space as [-5, 0, 5], meaning that the rudder angle turns 5° to the left, remains unchanged, or 5° to right respectively. Considering the steering angle of a ship is generally between ±35°, the rudder angle after taking an action must also be within this range.

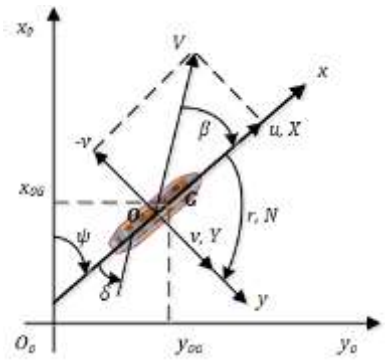


Fig.1. Applied earth-fixed and body-fixed coordinate systems

For the multi-ship system, it is necessary to define the reward value of a single agent first. In fact, each factor that affects the choices of a helmsman should be described as appropriate rewards or punishments. Since many factors have a direct or indirect influence on the decisions of the helmsman, it might take enormous effort to list up these factors perfectly in a reward function, which should be a huge engineering problem. Hence, this research only selects five typical factors from different perspectives, aiming to demonstrate the applicability of the proposed approach.

(1) Approaching a destination. Generally speaking, each ship should reach its destination. If the ship cannot approach the destination, the navigation is considered as failure. This reward is set as

$$r_{destination}=\begin{cases}\lambda_{destination}, & \text{approaching the destination}\\ -\lambda_{destination}, & \text{else}\end{cases}$$

(6)

where $\lambda_{destination}$ is a constant greater than 0. When the ship agent is approaching its destination, the reward is set to $\lambda_{destination}$. When the ship cannot approach its destination, the reward is set to $-\lambda_{destination}$. This policy will encourage the ship not to deviate from the navigation destination.

(2) Lane deviation. Lane deviation is an abnormal behaviour, which is easy to lead to accidents. Therefore, lane deviation is not encouraged while navigating. Hence, this reward is denoted as,

$$r_{lane}=\begin{cases}\lambda_{lanein}, & \text{in lane}\\ -\lambda_{laneout}, & \text{lane deviation}\end{cases} \quad (7)$$

where λ_{lanein} denotes the reward value when the ship is sailing in the route, and $-\lambda_{laneout}$ denotes the punishment when the ship is out of the route.

(3) Ship domain. Ship domain is a concept invented by traditional marine technologies (Szlapczynski and Szlapczynska, 2016). In practice, collision avoidance is difficult for a cargo ship due to its large tonnage, huge inertia, and relatively weak driving forces. Therefore, an imaginary region, namely a ship domain, should be defined in advance which is generally 7 times longer than the ship's length and 3 times wider than its width. When an obstacle has entered this area, caution warnings will be triggered, which is a tense situation for all the crews. An experienced helmsman should try to avoid this situation. This reward can be denoted as,

$$r_{danger}=\begin{cases}-\lambda_{danger}, & \text{in ship domain}\\ 0, & \text{else}\end{cases}$$

(8)

where $-\lambda_{danger}$ denotes the punishment when some other object enters the ship's domain.

(4) Collision. To avoid collision is the first priority for ships. When colliding with some objects, such as other ships, rocks or a coastline, the ship should be punished. This reward is denoted as,

$$r_{collision}=\begin{cases}-\lambda_{collision}, & \text{if collision}\\ 0, & \text{else}\end{cases}$$

(9)

where $-\lambda_{collision}$ denotes the punishment value. Moreover, if the target collides with something, the present episode of training can be considered as failed and the training process will restart. In particular, $\lambda_{collision}$ should be assigned with a relatively large value, since avoiding collisions should always be the priority.

(5) Avoidance rules. The ship collision avoidance rules are very complex. This research selects one typical rule for modelling. The ships tend to avoid the coming ship from its right side and to sail through the stern of the other ship. The avoidance of violating this process can be regarded as unreasonable. Other rules or conventions can also be modelled by this method.

$$r_{regulation}=\begin{cases}-\lambda_{regulation}, & \text{breaking the rule}\\ 0, & \text{else}\end{cases} \tag{10}$$

where $-\lambda_{regulation}$ denotes the punishment when the ship breaks the rule.

Based on these five factors above, the agent reward can be defined as,

$$r=r_{destination}+r_{lane}+r_{danger}+r_{collision}+r_{regulation} \tag{11}$$

As elaborated previously, the factors that affect the ship are more than these five discussed in this section. The reason for choosing these five lies in that they are coming from different perspectives. More factors based on another perspective can be modelled similarly.

3.3 Different cooperative relationships between ship agents

Compared with a single agent, each agent is affected not only by the environment, but also by other agents in a multi-agent system. Therefore, each agent in a multi-agent system must observe the state and behaviour of other agents, and the state transition and reward value of each agent are affected by the joint action of all agents.

Similarly, each ship agent must observe the state and action of other agents, and their own state and behaviour will also affect other agents for the multi-ship system. This research assumes that there are two ship agents in the system. When these two ships encounter, the two agents will be in different cooperative relationships, making different decisions if their cooperation goals are different.

(1) Fully cooperative

Each agent not only considers its own navigational safety, but also avoids putting the other one in danger, when ships encounter. Such two ship agents are fully cooperative. To achieve this goal, both agents are penalized whenever one agent is in danger. In other words, the goal of the two ship agents is to maximize the sum of their cumulative returns.

(2) Fully competitive

On the contrary, agents only focus on their own safety, even if their decisions will put the other in danger. The goal of the two ships is to maximize each one's own cumulative returns, regardless of the reward value and safety of the other. Such two ship agents are fully competitive.

(3) A mixed game

When two ships encounter, they form the relationship of the transition between cooperation and competition if they are neither fully cooperative nor fully competitive.

Suppose the two ship agents' rewards are r_1 and r_2 which can be calculated by Equation (11) after performing a certain action.

The return function of Ship 1 can be defined as,

$$R_1=r_1+\rho_2 r_2 \tag{12}$$

Accordingly, the return function of Ship 2 can be defined as,

$$R_2=\rho_1 r_1+r_2 \tag{13}$$

Then the return function of the system is the sum of the reward functions of the two agents,

$$R=R_1+R_2 \tag{14}$$

As shown in Table 1, when ρ_1 and ρ_2 are both equal to 1, the return function can be maximized only when both ships obtain positive returns, and the two agents are fully cooperative. While ρ_1 and ρ_2 are both equal to 0, each agent only considers to maximize its own reward and the two agents are fully competitive. While ρ_1 and ρ_2 are from 0 to 1, the two agents are in a mixed stochastic game.

Table 1 Cooperative relationships between multi-ships

ρ_1	$\boldsymbol{\rho_2}$	Cooperative relationships
1	1	Fully cooperative
0	0	Fully competitive
[0,1]	[0,1]	Mixed game

It is appropriate to simulate and learn the decision-making of crew members with different personalities in multi-ship encounters in this way. As a result, agents can select optimal actions in different modes.

3.4 The network structure of a multi-ship cooperative system

As shown in Fig.2, the multi-agent network is modelled by a multi-layer perceptron. The input of the system is its state space, represented by $[angle_1, x_1, y_1, v_1, \psi_1, angle_2, x_2, y_2, v_2, \psi_2]$. There are 128 nodes in its first layer, which is a fully connected. The second layer is also a fully connected layer, with 64 nodes. The output layer consists of three nodes, corresponding to the three actions of action space.

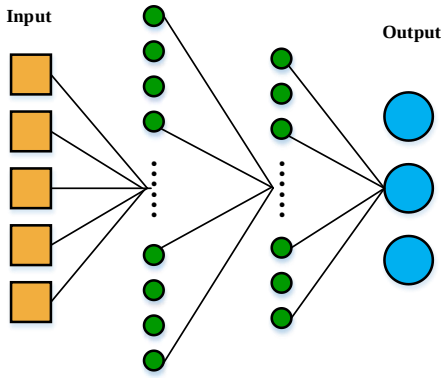


Fig.2. The network model of the multi-ship system

To ensure the stability of convergence, the DQN algorithm is adopted, which stores the current state, action, return and next state in a replay buffer, sampling through the greedy policy. The goal of the system is to make the difference between the target Q network and the Q network as little as possible. More importantly, each ship agent is controlled by the DQN algorithm with the same structure and parameters. The training parameters of its network model are shown in Table 2.

Table 2 Cooperative Relationships between multi-ship

Parameter	Value
Learning Rate	0.0002
Discount Rate	0.99
Minibatch Size	128
Replay Memory Size	20000
Target Network Update Frequency	1000
Initial exploration	1

4 A Case Study and Validation

4.1 Experimental platform

To verify the effectiveness of the proposed approach, the PyCharm was used to establish a simulation environment. As discussed previously, this research only used two agents to reduce calculation and to speed up the convergence. Moreover, the two ship agents chose a KVLCC2 tanker as the motion model, which is the standard object of modelling in navigation studies (Liu et al., 2016). Simulations are performed with the model-scale ship parameters as presented in Table 3.

Table 3 Basic parameters of the KVLCC2 within the MMG model

Attributes	Value
Length (m)	7

Attributes	Value
Width (m)	1.17
Draught (m)	0.46
Block coefficient (-)	0.81
Propeller revolution per second (1/s)	10.4
Range of rudder angles (deg)	-35~35

A scenario editor is designed and developed based on Pygame and Tinker. In this scenario editor, it is possible to set the scenario size, the ship size, the departure, the destination, the ship speed, *etc*. Moreover, the reward function can be set for each ship agent based on the description in Section 3 in this scenario editor.

According to the International Regulations for Preventing Collisions at Sea (COLREGs), this scenario editor modelled three scenarios, head-on, overtaking, and crossing (Zhao and Roh, 2019).

4.2 Training in different scenarios

The training was carried out separately with three different scenarios. As discussed previously, cooperative and competitive agents emerged by adjusting the cooperation coefficient of the two ship agents. The video of the trained ship sailing cooperatively in different scenarios can be found online (https://www.youtube.com/watch?v=h7ssNImWECg&list=PLia6EPeX0ULyw6FRlo0MZyYGiC9rlze-C).

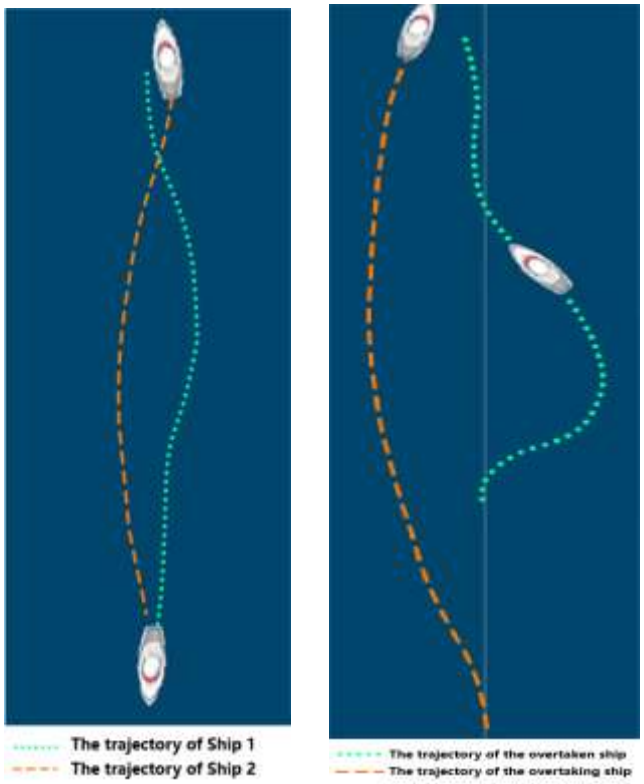
4.2.1 Head-on

This scenario size was set to 240 pixels × 560 pixels, where the top-left corner was taken as the origin (0, 0). The initial position of Ship 1 was (120, 30), and its destination was (120, 560). While the initial position of Ship 2 was (120, 560), and its destination was (120, 0). The speed of the two ship agents was 1.0 pixels per second with initial heading angle set to 0. It was found that the two agents were capable of avoiding collision only in the fully cooperative scheme after training. Due to the narrow waterway, two ship agents in the fully competition and mixed games could not spare enough space for each other. Hence, it was difficult to avoid collision and impossible to sail safely. Based on Fig. 3(a), it can be inferred that both ship agents turned to the left in the head-on encounter. When one ship agent left the domain of the other, they both turned starboard and returned to the middle of the waterway.

4.2.2 Overtaking

The overtaking encounter scenario size was also set to 240 pixels × 560 pixels, where the top-left corner was taken as the origin (0, 0). The initial position of the two ship agents was (120, 480), and their destination was (120, 0). The speed of the ship agent overtaking was 1.5 pixels per second, while the one of the ship being overtaken was 0.4 pixels per second.

Similarly, it was found that the two agents were capable of avoiding collision only in the fully cooperative scheme after sufficient training. Based on Fig. 3(b), it can be inferred that the ship being overtaken turned starboard while the overtaking ship turned to left in the overtaking situation. When the overtaking process was over, the overtaken ship turned left and returned to the middle of the waterway.



(a) Head-on (b) Overtaking

Fig.3. The trajectories of ship agents in the collision avoidance process

4.2.3 Crossing

The size of crossing encounter scenario was set to 480 pixels × 480 pixels, where the top-left corner was taken as the origin (0, 0). The initial position of Ship 1 was (240, 0), and its destination was (240, 480) while the initial position of Ship 2 was (0, 240), and its destination was (480, 240). The speed of the two ship agents was 1.0 pixels per second with initial heading angle set to 0. In this

scenario, the multi-agent system had acquired the cooperative collision avoidance intelligence through training in three cooperative schemes.

(1) Fully cooperative

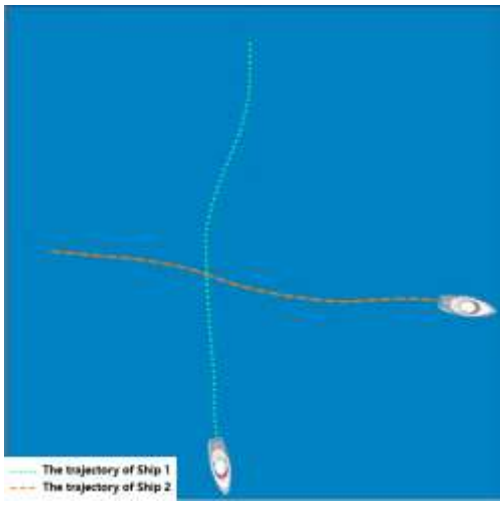
As discussed above, when both ρ_1 and ρ_2 were set to 1, the two agents were fully cooperative, and the goal was to achieve optimal the return value of the two agents in all. From Fig. 4(a), it can be seen that both ships turned starboard and passed through the port side of each other. Furthermore, Ship 1 passed through the stern of Ship 2. It can be concluded that the collision avoidance of the two ships followed "right hand collision avoidance", which met the requirement of the COLREGs.

(2) Fully competitive

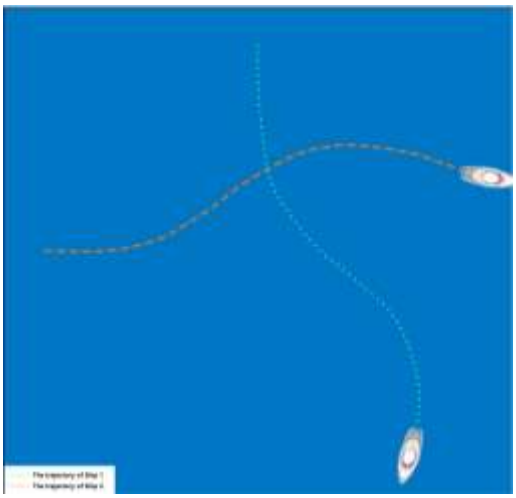
Both ρ_1 and ρ_2 were set to 0, the two agents only took their own safety and efficiencies into consideration. From the experimental results, both agents turned left and passed through the starboard side of the other one, and Ship 1 passed through the bow of Ship 2, as shown in Fig. 4(b). Although the collision avoidance was successful, it did not conform with the navigation rules, which was still very dangerous in practice.

(3) Mixed game

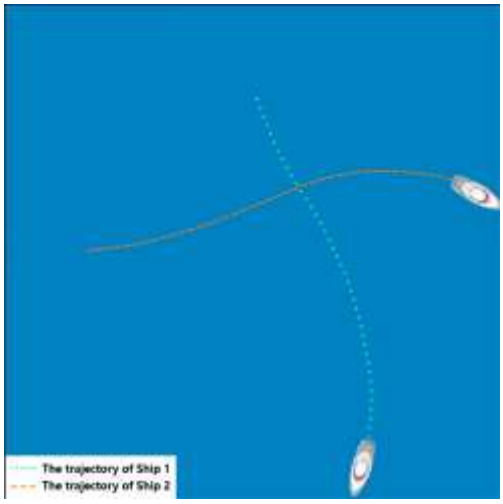
In this experiment, both ρ_1 and ρ_2 were set to 0.5. As a result, the two agents played a mixed game. From the experimental results, both agents turned left and passed through the starboard side of the other one, and Ship 1 passed through the bow of Ship 2, as shown in Fig. 4(c). The collision avoidance process of the two ships also went against the collision convention. However, the "dangerous situation" had not appeared since the two agents took early actions.



(a) Fully cooperative



(b) Fully competitive



(c) A mixed game

Fig.4. The trajectories of ship agents in collision avoidance process of crossing

5 Conclusion

In order to simulate the cooperative collision avoidance awareness between multi-ships, this research analysed the cooperation mechanism between agents using MADRL and established several cooperative schemas by determining the coefficient in reward functions. According to the rules of ship collision avoidance, this study modelled different scenarios and verified the proposed method. Different from the traditional nonlinear optimization-based method, each MADRL agent had an independent operation consciousness and was capable of making relatively reasonable decisions even without the cooperation of the other agent, which is highly similar to the human consciousness. Overall, it provided new solutions for bionic modelling of ship operations, which is of important theoretical and practical significance.

However, it was found that the incensement of agents led to an exponential growth of action space, which made the training time-consuming in a more complex avoidance experiment. Therefore, it is necessary to develop new methods to reduce the amount of calculation. On the other hand, it might be a wise way to imbed human knowledge into the MADRL-based model to speed out the convergence in finding the optimal route.

Acknowledgments

This research is supported by Zhejiang Province Key R&D projects (2021C03015), NSFC-Zhejiang Joint Fund for the Integration of Industrialization and Informatization (U1709215), Zhejiang outstanding youth fund (R21F030005).

References

Blaich, M., Kohler, S., Schuster, M., Schuchhardt, T., Reuter, J., Tietz, T., 2015. Mission integrated collision avoidance for USVs using laser range finder. MTS/IEEE OCEANS 2015 - Genova: Discovering Sustainable Ocean Energy for a New World 0–5. https://doi.org/10.1109/OCEANS-Genova.2015.7271415

Chen, C., Chen, X.-Q., Ma, F., Zeng, X.-J., Wang, J., 2019. A knowledge-free path planning approach for smart ships based on reinforcement learning. Ocean Engineering 189, 106299. https://doi.org/10.1016/j.oceaneng.2019.106299

Chen, C., Ma, F., Liu, J., Negenborn, R.R., Liu, Y., Yan, X., 2020. Controlling a cargo ship without human experience using deep Q-network. Journal of Intelligent and Fuzzy Systems 39, 7363–7379. https://doi.org/10.3233/JIFS-200754

Chen, J., Pan, W., Guo, Y., Huang, C., Wu, H., 2013. An obstacle avoidance algorithm designed for USV based on single beam sonar and fuzzy control. 2013 IEEE International Conference on Robotics and Biomimetics, ROBIO 2013 2446–2451. https://doi.org/10.1109/ROBIO.2013.6739838

Denker, C., Baldauf, M., Fischer, S., Hahn, A., Ziebold, R., Gehrmann, E., Semann, M., 2016. E-Navigation based cooperative collision avoidance at sea: The MTCAS approach. 2016 European Navigation Conference, ENC 2016. https://doi.org/10.1109/EURONAV.2016.7530566

Eriksen, B.O.H., Wilthil, E.F., Flåten, A.L., Brekke, E.F., Breivik, M., 2018. Radar-based maritime collision avoidance using dynamic window. IEEE Aerospace Conference Proceedings 2018-March, 1–9. https://doi.org/10.1109/AERO.2018.8396666

Fan, C., Wróbel, K., Montewka, J., Gil, M., Wan, C., Zhang, D., 2020. A framework to identify factors influencing navigational risk for Maritime Autonomous Surface Ships. Ocean Engineering 202. https://doi.org/10.1016/j.oceaneng.2020.107188

Foerster, J.N., Assael, Y.M., De Freitas, N., Whiteson, S., 2016. Learning to communicate with deep multi-agent reinforcement learning, in: Advances in Neural Information Processing Systems.

Jaderberg, M., Czarnecki, W.M., Dunning, I., Marris, L., Lever, G., Castañeda, A.G., Beattie, C., Rabinowitz, N.C., Morcos, A.S., Ruderman, A., Sonnerat, N., Green, T., Deason, L., Leibo, J.Z., Silver, D., Hassabis, D., Kavukcuoglu, K., Graepel, T., 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. Science 364, 859–865. https://doi.org/10.1126/science.aau6249

Lazarowska, A., 2018. A New Potential Field Inspired Path Planning Algorithm for Ships, in: 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR). IEEE, pp. 166–170.

Leite, A.R., Enembreck, F., Barthès, J.P.A., 2014. Distributed Constraint Optimization Problems: Review and perspectives. Expert Systems with Applications. https://doi.org/10.1016/j.eswa.2014.02.039

Li, S., Liu, J., Negenborn, R.R., 2019. Distributed coordination for collision avoidance of multiple ships considering ship maneuverability. Ocean Engineering 181, 212–226. https://doi.org/10.1016/j.oceaneng.2019.03.054

Praczyk, T., 2015. Neural anti-collision system for Autonomous Surface Vehicle. Neurocomputing 149, 559–572. https://doi.org/10.1016/j.neucom.2014.08.018

Liu, J., Quadvlieg, F., Hekkenberg, R., 2016. Impacts of the rudder profile on manoeuvring performance of ships. Ocean Engineering 124, 226–240. http://dx. doi.org/10.1016/j.oceaneng.2016.07.064

Song, C.H., 2014. Global path planning method for USV system based on improved ant colony algorithm. Applied Mechanics and Materials 568–570, 785–788. https://doi.org/10.4028/www.scientific.net/AMM.568-570.785

Sun, Z., Zhang, G., Lu, Y., Zhang, W., 2018. Leader-follower formation control of underactuated surface vehicles based on sliding mode control and parameter estimation. ISA Transactions 72, 15–24. https://doi.org/10.1016/j.isatra.2017.11.008

Szlapczynski, R., Szlapczynska, J., 2016. An analysis of domain-based ship collision risk parameters. Ocean Engineering. https://doi.org/10.1016/j.oceaneng.2016.08.030

Szymak, P., Praczyk, T., 2012. Using neural-evolutionary-fuzzy algorithm for anti-collision system of unmanned surface vehicle. 2012 17th International Conference on Methods and Models in Automation and Robotics, MMAR 2012 286–290. https://doi.org/10.1109/MMAR.2012.6347873

Tampuu, A., Matiisen, T., Kodelja, D., Kuzovkin, I., Korjus, K., Aru, Juhan, Aru, Jaan, Vicente, R., 2017. Multiagent cooperation and competition with deep reinforcement learning. PLoS ONE 12, 1–12. https://doi.org/10.1371/journal.pone.0172395

Tan, M., 1993. Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents, in: Machine Learning Proceedings 1993. https://doi.org/10.1016/b978-1-55860-307-3.50049-6

van der Sande, T., Nijmeijer, H., 2017. A Target Tracking System for ASV Collision Avoidance Based on the PDAF, Lecture Notes in Control and Information Sciences. https://doi.org/10.1007/978-3-319-55372-6_20

Wang, L., Wu, Q., Liu, J., Li, S., Negenborn, R., 2019. State-of-the-Art Research on Motion Control of Maritime Autonomous Surface Ships. Journal of Marine Science and Engineering 7, 438. https://doi.org/10.3390/jmse7120438

Wang, Z., Xiang, X., Yang, J., Yang, S., 2017. Composite Astar and B-spline algorithm for path

planning of Autonomous Underwater Vehicle. 2017 IEEE 7th International Conference on Underwater System Technology: Theory and Applications.

Zhao, L., Roh, M. Il, 2019. COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. Ocean Engineering 191, 106436. https://doi.org/10.1016/j.oceaneng.2019.106436

Zhao, Y., Li, W., Shi, P., 2016. A real-time collision avoidance learning system for Unmanned Surface Vessels. Neurocomputing 182, 255–266. https://doi.org/10.1016/j.neucom.2015.12.028

Zhou, B., Liao, X., Huang, T., Chen, G., 2015. Leader-following exponential consensus of general linear multi-agent systems via event-triggered control with combinational measurements. Applied Mathematics Letters. https://doi.org/10.1016/j.aml.2014.09.009

Zhou, X., Wu, P., Zhang, H., Guo, W., Liu, Y., 2019. Learn to Navigate: Cooperative Path Planning for Unmanned Surface Vehicles Using Deep Reinforcement Learning. IEEE Access 7, 165262–165278. https://doi.org/10.1109/ACCESS.2019.2953326