COLREGs-Compliant Multi-Ship Collision Avoidance Based on Multi-Agent Reinforcement Learning Technique

Guan Wei and Wang Kuo *

Abstract: The congestion of waterways can easily lead to traffic hazards. Moreover, according to the data, the majority of sea collisions are caused by human error and the failure to comply with the Convention on the International Regulation for the preventing Collision at Sea (COLREGs). To avoid this situation, ship automatic collision avoidance has become one of the most important research issues in the field of marine engineering. In this study, an efficient method is proposed to solve multi-ship collision avoidance problems based on the multi-agent reinforcement learning (MARL) algorithm. Firstly, the COLREGs and ship maneuverability are considered for achieving multi-ship collision avoidance. Subsequently, the Optimal Reciprocal Collision Avoidance (ORCA) algorithm is utilized to detect and reduce the risk of collision. Ships can operate at the safe velocity computed by the ORCA algorithm to avoid collisions. Finally, the Nomoto three-degrees-of-freedom (3-DOF) model is used to simulate the maneuvers of ships. According to the above information and algorithms, this study designs and improves the state space, action space and reward function. For validating the effectiveness of the method, this study designs various simulation scenarios with thorough performance evaluations. The simulation results indicate that the proposed method is flexible and scalable in solving multi-ship collision avoidance, complying with COLREGs in various scenarios.

Keywords: multi-ship collision avoidance; COLREGs; multi-agent reinforcement learning; ORCA; ship maneuverability

1. Introduction

The continuous increase in the number of maritime transportation vessels results in the waterways becoming more congested. Obviously, this situation will cause serious traffic hazards. When there are many ships around, it is easy to make wrong decisions by only relying on people to control the ship. According to data, it is found that about 89–96% of sea collisions are caused by human error [1]. To avoid this situation, ship automatic avoidance has become one of the most important research issues in the field of marine engineering. However, due to the complex motion model of ships and the low control accuracy, most of the algorithms cannot meet the requirements. Artificial intelligence technology (AI) is currently the most applicable method to solve this problem [2]. Deep reinforcement learning (DRL) is a new research hotspot in the field of artificial intelligence, and has made great progress in both theory and applications, in particular, the I-go agent named ‘Alpha-Go’, which was created by the ‘Google Deep Mind’ team and which beat the top I-go players [3]. It has also made substantial breakthroughs in decision-making control [4]. Deep reinforcement learning (DRL) consists of two parts: deep learning and reinforcement learning. Among them, deep learning has a strong perceptual ability, and is widely used in image analysis [5], speech recognition [6] and other fields. Reinforcement learning is known for its decision-making ability, first proposed by Sutton in 1984 [7]. It uses a reward and punishment system, gains experience from the environment, then adjusts strategies through multiple training to adapt to the environment and ultimately achieves the desired results. The task of autonomous ship avoidance involves interactions with the environment.
and objects, and many scenarios involve interactions between multiple ships; and this problem becomes more complex during this common interaction.

This paper proposes a COLREGs-compliant multi-ship collision avoidance method based on multi-agent reinforcement learning CA-QMIX. A 3-DOF ship model is utilized, and an Optimal Reciprocal Collision Avoidance algorithm is used to detect the risk of collision and provide a safe velocity. The simulation studies validate the effectiveness of the algorithms in the multi-ship collision avoidance problems.

2. Literature Review and Motivation

The automatic collision avoidance technology of ships is key in guaranteeing the safety of navigation. Recently, the related theories and technologies have been gradually improved. Miele et al. [8] proposed a method based on the multi-subarc sequential gradient-restoration algorithm to solve two cases of the collision avoidance problem: two ships moving along the same rectilinear course and orthogonal courses. Phanthong et al. [9] described path replanning techniques and proposed an algorithm based on the A star algorithm to avoid against stationary and dynamic obstacles with the optimal trajectory. Cheng et al. [10] proposed an optimization method based on a genetic algorithm which was applied to avoid collision and to seek the trajectory. However, these methods do not comply with the COLREGs, which should not be ignored for ocean-going ships.

Methods complying with COLREGs collision avoidance have been proposed for multi-ships in the open sea. Wilson et al. [11] proposed a new navigation method called a line-of-sight counteraction navigation algorithm (LOSCAN). The algorithm aided maneuver decision-making for a two-ship collision avoidance complying with COLREGs. However, this method is not capable of dealing with multi-ship collision avoidance. Liang et al. [12] proposed the minimum course alteration algorithm (MCA) to avoid moving ships, or obstacles constrained by COLREGs. The results of simulation presented that the algorithm was credible in collision avoidance. Chen et al. [13] designed an intelligent collision avoidance control system, which integrated the collision avoidance navigation and the nonlinear optimal control methods. To avoid collision, two fuzzy indicators including collision risk and collision avoidance acting timing were developed. Johansen et al. [14] describes a concept for a ship collision avoidance system, which is based on model predictive control. COLREGs and collision hazards associated with each of the alternative control behaviors are evaluated on a finite prediction horizon. Hu L et al. [15] designed a multi-objective optimization algorithm which cooperated a hierarchical sorting rule to prioritize the objective of course or speed change preference over other objectives such as path length and path smoothness. All of these methods can complete the two-ship collision avoidance task, and comply with COLREGs. However, when the ship encounters more complex scenarios, such as four ships colliding at the same time, these methods cannot achieve collision avoidance navigation.

With the development of artificial intelligence, a number of collision avoidance methods based on deep reinforcement learning (DRL) have been developed. Shen et al. [16] presented a training method based on DRL for ship collision avoidance, which incorporated the ship maneuverability, human experience and COLREGs. Through the experimental validation of three self-propelled ships, it demonstrated that the method based on DRL had great potential to realize automatic collision avoidance. Sawada et al. [17] proposed a multi-ship automatic collision avoidance method based on DRL in continuous action space, and the obstacle zone by target was used to compute risk of collision. The trained agent has passed a large number of simulation scenarios. Li et al. [18] utilized the artificial potential field (APF) algorithm to improve the action space and reward function of DRL. To avoid collision, the method trained agents complying with COLREGs. The results of simulation showed that the improved DRL could realize automatic collision avoidance. Zhao et al. [19] proposed a method which used the Deep Neural Network (DNN) to map the states of encountered ships to a ship’s own steering commands in terms of rudder angle. The policy gradient-based DRL algorithm was used to train the DNN for collision avoidance.
complying with COLREGs. The simulation results indicated that the multi-ship model was able to avoid collision. Xu et al. [20] formulated the collision avoidance strategy and designed the state, action, reward function and network structure to improve the DDPG algorithm. The results showed that the method can give reasonable collision avoidance actions and realize effective collision avoidance. The advantages and disadvantages of the literature is shown in Table 1.

Table 1. The Simple Summary of the Literature Review.

| Type                                    | Reference | Technique                                | Advantages                                                                 | Disadvantages                                                                 |
|-----------------------------------------|-----------|------------------------------------------|----------------------------------------------------------------------------|------------------------------------------------------------------------------|
| Ship collision avoidance                | [8]       | multi-subarc sequential gradient-restoration algorithm | Solving two cases of the collision avoidance problem                       | Ship’s actions do not conform to COLREGs                                       |
|                                         | [9]       | A star algorithm                         | Avoiding against the stationary and dynamic obstacles with the optimal trajectory |                                                                             |
|                                         | [10]      | Genetic algorithm                        | Avoiding collision and seek the trajectory                                |                                                                              |
|                                         | [11]      | A line of sight counteraction navigation algorithm | Two-ship collision avoidance complied with COLREGs                      |                                                                              |
| COLREGs-compliant ship collision avoidance| [12]      | Minimum course alteration algorithm      | Avoid moving ships or obstacles constrained by COLREGs                    |                                                                              |
|                                         | [13]      | Nonlinear optimal control method         | Collision risk and collision avoidance acting timing are developed        | When the ship encounters more complex scenarios, such as four ships will collide at the same time, these methods cannot achieve collision avoidance navigation |
|                                         | [14]      | Model predictive control                 | COLREGs and collision hazards associated with each of the alternative control behaviors are evaluated on a finite prediction horizon |                                                                              |
|                                         | [15]      | Multi-objective optimization algorithm   | Cooperating a hierarchical sorting rule                                  |                                                                              |
| COLREGs-compliant multi-ship collision avoidance based on DRL | [16]      | DRL incorporated the ship maneuverability, human experience and COLREGs | The experimental validation of three self-propelled ships                 | Discrete action element                                                        |
|                                         | [17]      | DRL in continuous action space           | The risk of collision is reduced                                          | Poor convergence                                                              |
|                                         | [18]      | Utilizing the artificial potential field (APF) algorithm to improve DRL | Improving the action space and reward function of DRL                     | -                                                                            |
|                                         | [19]      | The policy-gradient based DRL algorithm | Multi-ship collision avoidance                                           | Heading angle retention                                                       |
|                                         | [20]      | DDPG                                     | The method can give reasonable collision avoidance actions and realize effective collision avoidance | -                                                                            |

In this study, a novel intelligent method based on multiple agent reinforcement learning, named the CA-QMIX algorithm is proposed. The COLREGs and ship maneuverability are considered for achieving multi-ship automatic collision avoidance. The Optimal Reciprocal Collision Avoidance (ORCA) algorithm is used to detect and reduce the risk of
collision. The safe velocity computed by the ORCA is adopted to avoid collision. This study also utilizes the three-degrees-of-freedom (3-DOF) Nomoto ship motion mathematical model to simulate the maneuvers of a ship. Finally, the state space, action space and reward functions are designed for improving the convergence rate of training. The simulation results indicated that the proposed method has excellent flexibility and scalability for solving multi-ship collision avoidance complying with COLREGs in various scenarios.

The organization of this paper is stated as follows: In Section 3, the method for detecting the risk of collision is given first. Then, the ship motion model and COLREGs, which are considered as the basis of ship collision avoidance, are illustrated. Section 4 describes the principles and applications of the multi-agent reinforcement algorithm. Section 5 is the simulation result of the multi-ship collision avoidance. Section 6 is a summary of this paper.

3. Ship Collision Avoidance Problem

3.1. Problem Definition

The solution to the multi-ship collision avoidance problem can be roughly divided into the following two categories:

1. Single-agent collision avoidance: The own-ship (OS) is considered as an agent, the target-ship (TS) is seen as a dynamic obstacle;
2. Multi-agent collision avoidance: Each ship is an agent, and there are partnerships between them.

In this study, we aim at successfully completing multi-ship collision avoidance and reach the target point. The good or bad behavior of collision avoidance is not only related to own-ship (OS) actions, but also related to the target-ship (TS) actions; under the trend that the communication network (5G or 6G network) will gradually cover the world in the future, the interaction between ships will be more convenient, and the advantages of the multi-agent method will become more obvious. Thence this paper defines multi-ship collision avoidance as a multi-agent problem, and uses the multi-agent reinforcement learning algorithm to solve the problem.

3.2. The Ship Motion Model and Collision Detection

Establishing a suitable ship motion mathematical model is necessary before using the algorithm. This paper uses the Nomoto three-degrees-of-freedom (3-DOF) model [21] and the principal dimensions of the ship form [22]. The coordinate systems are shown in Figure 1, and the principal dimensions of the ship are given in Table 2.

![Figure 1. The ship motion mathematical model.](image-url)
Table 2. Principal dimensions of the ship.

| Parameters           | Value |
|----------------------|-------|
| Length (m)           | 52.5  |
| Beam (m)             | 8.6   |
| Draft (m)            | 2.29  |
| Rudder area (m²)     | 1.5   |
| Max rudder angle (deg)| 40    |
| Max rudder angle rate (deg/s) | 20 |
| Nominal speed (kt)   | 15    |
| K index              | −0.085|
| T index              | 4.2   |

Where the ship’s velocity set includes surge velocities $u_v$, sway velocities $v_v$ and yaw rate $r_v$. The $\psi$ denotes heading angle, and $\psi_d$ stands for desired heading angle. Hence the error of heading angle is $\psi_e = \psi - \psi_d$. The rudder characteristics are expressed as:

$$\begin{bmatrix} \psi \\ r \\ \delta \end{bmatrix} = \begin{bmatrix} r \\ (K\delta - r)/T \\ (\delta_E - \delta)/T_E \end{bmatrix}$$

(1)

where $\delta$ and $\delta_E$ are the real rudder angle and the command rudder angle. $T_E$ is the time constant of the steering gear.

To reduce collision risk, this paper performs collision detection using the ORCA (Optimal Reciprocal Collision Avoidance) [23] method. The schematic diagram of ORCA is illustrated in Figure 2. Similarly, the ship domain concept is suggested to calculate the collision risk area and is used to define a safe area. To increase security, the area of the ship domain is further expanded into a circle (taking $d_5$ as diameter [24]).

$$d_5 = L \cdot v^{1.26} + 30v + u$$

(2)

Figure 2. Establish the safe area by designing the ship domain. (a) The design of the ship safe domain; (b) The information of multi-ship motion.

As is shown in Figure 2a, a hexagon-shaped collision danger risk area is created, then taking the smallest circumscribed circle of the hexagon, the final collision danger risk area is formed. The multi-ship collision avoidance problem can be simplified into the collision avoidance behavior of circular areas with a radius $R_{OS}$ and $R_{TS}$. As Figure 2 shows, $P_{OS}$...
and \( P_{TS} \) stand for the location of OS and TS, respectively. However, the movement of the areas are still constrained by the ship model.

In the velocity coordinate system of Figure 3a, assuming that TS is stationary, if OS does not collide with TS during movement, then the velocity of OS cannot be selected from the velocity obstacle \( V_{OS|TS}^{ts} \) (Figure 3a gray part). The definition of the velocity obstacle implies that if \( V_{OS} - V_{TS} \in V_{OS|TS}^{ts} \), or equivalently if \( V_{TS} - V_{OS} \in V_{TS|OS}^{ts} \), then OS and TS will collide at some moment before time \( t_s \) (one time step).

\[
V_{OS|TS}^{ts} = \{ v \exists t \in [0, t_s] : tv \in D(P_{OS} - P_{TS}, R_{TS} + R_{OS}) \} \tag{3}
\]

where \( D \) is a circle with center \( P_{TS} - P_{OS} \) and radius \( R_{TS} + R_{OS} \). The \( V_{OS|TS}^{ts} \) is geometrically a truncated cone with its apex at the origin and its two sides tangent to the circle (center \( P_{TS} - P_{OS} \) and radius \( R_{TS} + R_{OS} \)). The cone is truncated by arc (center \( (P_{TS} - P_{OS})/t_s \) and radius \( (R_{TS} + R_{OS})/t_s \)). Generally speaking, OS cannot sail for \( t_s \) time with current velocity, otherwise OS will collide with the TS.

![Figure 3. Calculate the optimal velocity by ORCA: (a) When TS is stationary, the collision velocity is calculated; (b) When TS is moving, the collision velocity is calculated; (c) The geometric meaning of the vector \( u \) and \( n \) are expressed; (d) The geometric meaning of the optimal velocity is expressed.](image)

When \( V_{TS} \) is considered, the set of velocities which make OS collide with TS are \( V_{OS|TS}^{ts} \oplus V_{TS} \) (Figure 3b gray part). Finally, the complementary set of \( V_{OS|TS}^{ts} \oplus V_{TS} \) is the safe velocity recorded as \( V_{safe} \). If the ship trails with \( V \in V_{safe} \), the collision will be avoided; but it is a difficult problem to select the optimal velocity from the set \( V_{safe} \).

For solving the above problem, a method named Optimal Reciprocal Collision Avoidance (ORCA) is presented. Firstly, a vector \( u \) illustrates minimal change to make OS’s velocity \( V_{OS} \notin V_{OS|TS}^{ts} \).

\[
u = \left( \text{argmin}_{V_{OS}} \partial V_{OS|TS}^{ts} \| V_{OS} - V_{OS}^{*} \| \right) - V_{OS} \tag{4}
\]

where \( \partial V_{OS|TS}^{ts} \) is the limit point of the \( V_{OS|TS}^{ts} \). The vector \( n \) is a normal vector of the \( V_{OS|TS}^{ts} \) boundary. \( n \) points to the inside of the collision area and the starting point is at the intersection of the vector \( u \) and the boundary. Combining these variables, \( ORCV_{OS|TS}^{ts} \) is defined as the optimal reciprocal collision avoidance velocity of OS.

\[
ORCV_{OS|TS}^{ts} = \left\{ V_{OS}^{*} \left| \left( V_{OS}^{*} - (V_{OS} + \frac{1}{2}u) \right) \cdot n \leq 0 \right. \right\} \tag{5}
\]

where \( ORCV_{OS|TS}^{ts} \) is the optimal reciprocal collision avoidance velocity set for OS avoiding collision with TS in time \( t_s \). When multiple ships avoid collision, each ship computes the optimal velocity set by ORCA, and the intersection of these velocity set form a polyhedron.
For achieving multi-ship collision avoidance, we defined a velocity set $\text{ORCV}_{OS}^{t_s}$, in which OS adopting $V \in \text{ORCV}_{OS}^{t_s}$ can avoid colliding with all TSs.

$$\text{ORCV}_{OS}^{t_s} = D(0, V_{OS}^*) \cap \left( \bigcap_{OS \neq TS} \text{ORCV}_{OS|TS}^{t_s} \right)$$

(6)

3.3. COLREGs

Before using the method to solve the multi-ship collision avoidance problem, COLREGs needs to be considered. The OS must react to avoid the TSs while complying with COLREGs, and subsequently, return to its predefined path once safety is confirmed. As illustrated in Figure 4, a diagram centered on the OS is divided into four parts:

- **Head-on**: When two vessels (OS and TS) are meeting on opposite or approximately opposite routes within an azimuth angle of $(0^\circ, 5^\circ)$ or $(355^\circ, 360^\circ)$, this situation should be judged as a head-on situation. The two vessels (OS and TS) should alter their course to starboard, so that each vessel should pass on the port side of the other to avoid collision;
- **Port crossing**: When a vessel (OS) is crossing on its port side within an azimuth angle of $247.5^\circ - 355^\circ$, this situation should be judged as a port crossing situation. The vessels (OS) are not the give-way vessels so they shall keep their original course and speed;
- **Starboard crossing**: When a vessel (OS) is crossing on its starboard side within an azimuth angle of $5^\circ - 112.5^\circ$, this situation should be judged as the starboard crossing situation. The vessels (OS) shall alter their course to the starboard side to avoid collision;
- **Overtaking**: When a vessel (OS) is chasing another vessel (target-ship, TS) within an azimuth angle of $112.5^\circ - 247.5^\circ$ directly behind it, this situation should be judged as overtaking. The vessels (OS) shall alter their course to starboard or port side to avoid collision.

![Figure 4. The categorization of TSs based on COLREGs.](image)

The collision avoidance behaviors conforming to COLREGs are shown in Figure 5.
3.4. COLREGs-Based Multi-Ship Collision Avoidance

The COLREGs can be extended to scenarios where OS encounters multiple TSs. The multi-ship collision avoidance under the COLREGs can be summarized as Figure 6.

In Figure 6, the OS encounters two TSs in different directions, they should all comply with COLREGs and alter course to starboard to avoid collision. In the same way, when three ships encounter a similar situation, they should all alter course to starboard. In summary, when a multi-ship (≥3) encounter is occurring, each ship should follow the COLREGs and alter course to starboard for avoiding collision.

The process of collision avoidance combined with COLREGs is shown in Figure 7. Firstly, the instantaneous ship domain should be calculated and expanded to a safe round area. Then, OS detects whether TSs enter into the OS’s safe area, and judges the encounter scenario by TSs’ instantaneous positions. To avoid collision, OS must select $V \in ORCV_{OS}^k$, which is calculated by ORCA. After one time step, OS checks whether TSs leave the OS’s safe area. If there are TSs still in the safe area, the above steps are repeated.
4. Algorithm Background

4.1. Algorithm Model and CTDE

4.1.1. CTDE

Multi-agent reinforcement learning is a recent research topic. Centralized Training Decentralized Execution (CTDE) [25] is arguably the simplest method to train and execute. However, multi-agent reinforcement learning has two difficulties:

- Observational limitations: When the agent interacts with the environment, the agent cannot obtain the global state $s$ of the environment, and can only see the local observation information within its own observation range $o$;
- Instability: When multiple agents learn together, the changing strategies and the mutative actions caused mean that the value function of agent $i$ cannot stably update.

Therefore, to solve the above problems, this study proposes using the Centralized Training Decentralized Execution (CTDE) framework to relax the limitation conditions and allow agents to access the global information during training.

4.1.2. DEC-POMDP Model

QMIX algorithm takes the DEC-POMDP model [26] as the standard for the cooperative multi-agent tasks model. All variables are grouped into a tuple $G = \{S, U, P, r, Z, O, N, \gamma\}$, where $s \in S$ denotes the true state of the environment. Each agent $i = \{1 \ldots N\}$ chooses an
action $u_i \in U$ at each time step, generating a joint action vector $u := [u_i]^{N}_{i=1} \in U^N$. Function $P(s'|s, u) : S \times U \times S \rightarrow [0, 1]$ decides all of the state transition dynamics. Every agent uses the same joint reward function $r(s, u) : S \times U \rightarrow \mathbb{R}$, and $\gamma \in [0, 1)$ is the discount factor. Each agent has its own observation $z \in Z$, according to the observation function $O(s, i) : S \times N \rightarrow Z$. Each agent also has an action observation history $\tau_i \in \Gamma := (Z \times U)^*$, on which it conditions its stochastic policy $\pi_i = (u_i|\tau) : \Gamma \times U \rightarrow [0, 1]$.

4.2. IQL and VDN

The IQL (Independent Q-Learning) [27] algorithm treats the rest of the agents directly as part of the environment. That is, each agent is solving a single-agent task. The value function of the agent $i$ is $Q_i(\tau_i, u_i)$. Only relying on $Q_i(\tau_i, u_i)$ for decision-making is unstable. Obviously, due to the existence of agents in the environment, the environment is a non-stationary state, so convergence cannot be guaranteed, and the agent can easily get caught up in endless exploration.

It is necessary to use $Q_{\text{total}}(\tau, u)$ to learn in global sight. Sunehag [28] proposed the VDN algorithm which used $Q_i(\tau_i, u_i)$ to finish the factorization of $Q_{\text{total}}(\tau, u)$, the formula is as follows:

$$Q_{\text{total}} = \sum_{i=1}^{N} Q_i(\tau_i, u_i)$$

The VDN just accumulates the local action value functions of each agent to get the joint action value function, so that it satisfies the conditions as the same additivity of $Q_{\text{total}}(\tau, u)$ and $Q_i(\tau_i, u_i)$. But it does not integrate the single agent local value function during learning.

4.3. QMIX Algorithm

4.3.1. IGM Condition and Constraint

In order to follow the advantages of VDN, centralized learning is used to obtain distributed strategies; QMIX [29] first defines the conditions called IGM (Individual-Global-Max):

$$\argmax_{u} = \left( \begin{array}{c}
\argmax_{u_1} Q_1(\tau_1, u_1) \\
\vdots \\
\argmax_{u_N} Q_N(\tau_N, u_N)
\end{array} \right)$$

Generally speaking, if Equation (8) is satisfied (doing the argmax of $Q_{\text{total}}(\tau, u)$ and $Q_i(\tau_i, u_i)$ are equivalent), then gaining optimal actions by local $Q_i(\tau_i, u_i)$ is trivially tractable.

For achieving this effect, the QMIX algorithm sets a sufficient condition:

$$\frac{\partial Q_{\text{total}}}{\partial Q_i} \geq 0, \forall i \in N$$

If $Q_{\text{total}}(\tau, u)$ and $Q_i(\tau_i, u_i)$ satisfy monotonicity, then equation 4 holds. For the purpose of achieving the constraints, QMIX uses an architecture to implement.

4.3.2. Overall Framework

The overall framework of the QMIX algorithm is shown in Figure 8. The network structure mainly consists of three parts:

- Agent network (Figure 8a): It is represented by the DRQN network. In the partially observable setting, agents using RNN can use all their action-observation history information to get the current state. Its input at each step is the current individual observation $o_i^t$ of the agent and the action $u_{i-1}$ at each time step;
- Hypernetwork: Hypernetwork [30] is used to calculate network weights and biases in the mixing network. Its inputs are global state inputs. The outputs are the weights and the bias, where the weights need to be greater than 0 ($W \geq 0$), so the activation
function is the absolute activation function. Biases use the common Relu activation function, because it does not have a requirement for the value range;

- Mixing network (Figure 8c): Its weights and biases are generated by the Hypernetwork, and its role is to mix the $Q_i(\tau, u_i)$ of each agent into a monotonic $Q_{total}(\tau, u)$ of the whole system, and to also make the training more stable by increasing the system information.

$Q_{total}(\tau, u)$

Figure 8. (a) The overall QMIX architecture; (b) Agent network structure; (c) Mixing network structure.

QMIX is trained to minimize the following loss:

$$L(\theta) = \sum_{i=1}^{b} \left[ (Q_{total}(\tau, u) - y_{total}(r, \tau'; \theta))^2 \right]$$  \hspace{1cm} (10)

where $b$ is the batch size of transitions sampled from the replay buffer, and the role of $y_{total}$ [31] is to update networks. $\theta^ -$ are the parameters of a target network as in DQN.

$$y_{total} = r + \gamma \max_{u'} Q_{total}(\tau', u'; s; \theta^-)$$  \hspace{1cm} (11)

4.3.3. Algorithm Implementation

During the multi-ship collision avoidance, each ship is an agent to participate in the training; the multi-agent reinforcement learning algorithm named QMIX is employed to solve the problem. The iterative updating process of the algorithm is shown in Figure 9. The parameters of the training are shown in Table 3.
Algorithm 1 Algorithm: QMIX with multi-ship collision avoidance

1: Initialize reply memory $\zeta$;
2: Initialize $Q_t$ and $Q_{total}$;
3: Initialize target parameter $\theta^- = \theta$;
4: for episode = 1 to $M$ do
5: Observe initial state $s$ and observation $o$ for each ship;
6: for $t = 1$ to $T$ do
7: set $u^t_i = \arg \max_{u^t_i} Q_t(\tau^t_i, u^t_i)$ for each ship;
8: Take action $u^t$ and retrieve next observation and reward $(o^t, r^t)$;
9: Store transition $(\tau^t, u^t, r^t, \tau^{t+1})$ in $\zeta$;
10: Sample a transition $(\tau, u, r, r')$;
11: Update $\theta$ by minimizing loss:
   \[ L(\theta) = \sum_{i=1}^{b}[(Q_{total}(\tau, u) - y_{total}(r, \tau'; \theta))^2] \]
12: Update target network parameters $\theta^- = \theta$ with period $l$;
13: end for
14: end for

Figure 9. QMIX algorithm for the multi-ship collision avoidance.

Table 3. The parameters of the training algorithm.

| Parameter                | Value  |
|-------------------------|--------|
| Discounted rate $\gamma$ | 0.99   |
| Lambda $\lambda$        | 0.95   |
| Time steps $T_{max}$     | 10,000 |
| The epoch of target network $E$ | 100    |
| The learning rate for RMSprop $r_{RMS}$ | $5 \times 10^{-4}$ |
| Clipping hyperparameter $\epsilon$ | 1      |

4.4. CA-QMIX Algorithm

4.4.1. Action Space

In the process of ship collision avoidance, the crew changes heading and speed to ensure navigational safety. Likewise, during automatic collision avoidance, the performance of turning is considered for designing the action $u$, where $u \in [-\psi, \psi]$ and $\psi$ is the change in course angle. The command of the rudder angle is obtained by the ship motion mathematical model. At each time step, each ship chooses an action $u^t_i$, giving rise to a joint action vector $[u^t_i]_{i=1}^{N}$, where $N$ is the number of ships.

4.4.2. State Space

The state space is defined as the set of information about the environment that the ship receives at a given time step $t_s$. The observed state includes each ship location $P_{location}$, the location of goal $P_{goal}$, heading angle $\psi$, desired heading $\psi_d$, velocity $V$ and the ship length $L$.

4.4.3. Reward Function

Reward function is an evaluation of ship movements, and calculated as the sum of the remuneration accumulated from each ship. This process is expressed in:

$$ R_{total} = \sum_{i=1}^{N} R_i $$ (12)
The reward of each ship $R_i$ is the sum of the rewards accumulated in each episode. The objective of the study is to avoid OS and TSs collision, and to maneuver OS complying with COLREGs. Consequently, the reward function can be defined to reward the agent for reaching the destination and for avoiding the collision by complying with COLREGs.

$$R_i = R_{\text{goal}} + R_{\text{collision}} + R_{\text{COLREGs}}$$ (13)

The goal reward function $R_{\text{goal}}$ is to guide the ship to reach the destination. It is expressed as a formula by:

$$R_{\text{goal}} = \begin{cases} 0, & \text{if } \|P_t - P_{\text{goal}}\| \leq \frac{d_5}{1} - \lambda_{\text{goal}}(\|P_t - P_{\text{goal}}\| - \|P_{t-1} - P_{\text{goal}}\|), \\ \text{if otherwise} \end{cases}$$ (14)

where $P_t$ is the ship current location at $t$ episode, and $\lambda_{\text{goal}}$ is a hyperparameter. As the distance between the ship and the destination gets shorter, the agent obtains more substantial reward value. The reward value reaches maximum, when the distance becomes less than $\frac{d_5}{1}$.

For collision avoidance and fulfilling COLREGs, this paper designs the reward functions $R_{\text{collision}}$ and $R_{\text{COLREGs}}$.

$$R_{\text{collision}} = \begin{cases} 0, & \text{if } V \in \text{ORCV}_{\text{OS}}^t, \\ -r_{\text{collision}}, & \text{if otherwise} \end{cases}$$ (15)

$$R_{\text{COLREGs}} = \begin{cases} r_{\text{COLREGs}}, & \text{if turn right} \\ -r_{\text{COLREGs}}, & \text{if otherwise} \end{cases}$$ (16)

However, sometimes the goal reward function is contradictory to the collision avoidance reward. Thereby, the whole process is divided into two stages including normal sailing and collision avoidance, as shown in Figure 10.

![Figure 10. QMIX algorithm for the multi-ship collision avoidance.](image-url)
5. Method for Path Planning and Collision Avoidance Based on CA-QMIX

According to the previous section, the design of the ship collision avoidance system has been presented with an explanation of important parts in detail. In this section, we trained the agent to avoid collision using the CA-QMIX algorithm. The proposed CA-QMIX algorithm has been evaluated with simulation tests for diverse environments. The setting of the ship collision avoidance simulation scenarios included two-ship encounter situations and a multi-ship encounter situation. The collision avoidance scenarios of two ships were used to evaluate whether the algorithm conforms to COLREGs. The scalability of the algorithm was then verified by multi-ship encountering scenarios.

5.1. Two Ships Collision Avoidance in Four Scenarios

To guarantee the performance of the algorithm, each ship making the decision must comply with COLREGs, and reach a destination after successfully avoiding collision. In the training phase, the state input of each ship consists of its state, as observed by itself. The output of the algorithm is the rudder angle. At each training iteration, each ship selects an optimal action based on state and observation to generate trajectories. But an episode will end if ships collide with others or ships all reach their destinations.

The average reward is computed as the sum of rewards accumulated from each ship, and the reward functions follow the rules designed in Section 3. When the average reward value tends to be stable as shown in Figure 11, the training process is completed, and the optimal agent can be obtained. All ships can automatically avoid collision, and strictly follow the COLREGs.

![Figure 11. Average reward for multi-ship collision avoidance.](image)

For inspecting the trained agent, the case (a), case (b), case (c) and case (d) were set to conduct the simulation. The origin and destination parameters of the simulation were shown in Table 4, and the simulation scenarios were setup as shown in Figure 12.
After detecting the collision risk, each ship adopts a different rudder angle and velocity to solve the danger. The velocity is limited to [0, 7.5 (m/s)]. After finishing collision avoidance, they continued to their destinations. In this process, the ship movement not only complied with the COLREGs, but also completed collision-free navigation.

In case (a), ship I and ship II encountered head-on situations (reciprocal courses in the range of 10°). In detecting collision risk, the ships quickly altered course to starboard. After finishing collision avoidance, they continued to their destinations. In this process, the ship movement not only complied with the COLREGs, but also completed collision-free navigation.

In case (b), a port crossing situation occurred if the ship IV was appearing within an azimuth angle (247.5°, 355°) of ship III. When two ships were in a dangerous ship domain, they altered course to starboard and finished the collision avoidance task by selecting a safe speed.

In case (c), ship V detected that ship VI was coming from the starboard side of ship V. Two ships were encountering a starboard crossing situation. Ship V altered course to starboard to avoid collision.

In case (d), ship VII was overtaking ship VIII in the range 135°. For avoiding collision, ship VII altered course to starboard and completed overtaking.

From Figure 12, the ships motion trajectories and collision avoidance behaviors can be observed. The results show that each ship followed the COLREGs and reached its destination, indicating that the trained agent can complete the collision avoid task.

According to Figure 13, the detailed information of the ship’s navigation can be obtained. Since the rudder angle and the speed of ship I and ship II are the same, only the information of ship I is expressed. The rudder angle of each ship is limited to [-30°, 30°]. And the velocity is limited to [0, 7.5 (m/s)]. After detecting the collision risk, each ship adopts a different rudder angle and velocity to solve the danger.
Three ships

| Scenario | Ship Number | Origin (m) | Destination (m) |
|----------|-------------|------------|-----------------|
| Three ships | Ship I | (0, 900) | (0, -900) |
| | Ship II | (779.4, 450) | (-779.4, -450) |
| | Ship III | (-779.4, 450) | (779.4, -450) |
| Four ships | Ship I | (0, -900) | (0, 900) |
| | Ship II | (-900, 0) | (900, 0) |
| | Ship III | (0, 900) | (0, -900) |
| | Ship IV | (900, 0) | (-900, 0) |

5.2.1. Three Ships Collision Avoidance Scenarios

According to Figure 6, the three ships encountering scenario was created. We need to evaluate the performance of the algorithm from two aspects. One is the assessment of collision avoidance and whether the ship can avoid collision and comply with COLREGs. The other is whether the ship can reach the destination. In Figure 14, three ships were at risk of collision in the central area. To avoid collision, three ships altered course to starboard while complying with COLREGs. The trajectories of the ships were safe and smooth. Figure 15 illustrates the change in their speed and rudder angle.

Figure 13. Rudder angle and velocity of ship I, III, V and VII.

In conclusion, the CA-QMIX algorithm can ensure compliance with the COLREGs under the premise of successful collision avoidance. In addition, the proposed method demonstrated its excellent collision avoidance performance, flexibility of application scenarios and scalability potential.

5.2. Simulation for Multi-Ship Collision Avoidance

In order to verify the scalability of the algorithm, scenarios of three and four ships encountering are set. The origin and destination of each ship are shown in Table 5.

Table 5. Multi-ship scenarios parameters setup.

Figure 14. Three ships encountering scenario. (Unit: 10 m).
When they sailed to the positions of the first figure, they detected the collision risk. For encounter situation. Initially, four ships sailed to their destinations along straight lines. Avoiding collision, the command that changed their course to turn right was issued. After avoiding collision, the command that changed their course to turn right was issued. After avoiding collision, the command that changed their course to turn right was issued. When they sailed to the positions of the first figure, they detected the collision risk. For encounter situation. Initially, four ships sailed to their destinations along straight lines. Avoiding collision, the command that changed their course to turn right was issued. After avoiding collision, the command that changed their course to turn right was issued. After avoiding collision, the command that changed their course to turn right was issued.

Figure 15. The rudder angle and velocity of three ships.

5.2.2. Four Ships Collision Avoidance Scenarios

To prove the scalability of the algorithm, we set up this complex simulation scenario, as shown in Figure 16, and the origin and destination coordinates of each ship are shown in Table 5. Where four ships navigate to a center point, if they do not adopt the appropriate collision avoidance behaviors, they will collide each other in central area.

Figure 16. Four ships encountering scenario. (Unit: 10 m).

Figure 17 illustrates the simulation process of collision avoidance with a four-ship encounter situation. Initially, four ships sailed to their destinations along straight lines. When they sailed to the positions of the first figure, they detected the collision risk. For avoiding collision, the command that changed their course to turn right was issued. After arriving in the positions of the second figure, the four ships move circularly to pass the hazardous area. Next, the four ships arrived at their positions in the third figure—the risk of collision had been solved. They changed course to more quickly arrive at the destination course. The rudder angle and velocity of the four ships in this process are shown in Figure 17.

Figure 17. The rudder angle and velocity for four ships.
5.3. Complementary Simulation for Multi-Ship Collision Avoidance

In this section, further multi-ship encounter scenarios were set to increase the validity of the proposed CA-QMIX method. For confirming that the collision avoidance of two ships complies with COLREGs in all scenarios, the further simulations are shown in Figure 18. Based on the previous simulations, Figure 18 illustrates the collision avoidance process under the various conditions.

Figure 18. Additional two-ships encountering scenarios. (The simulation of the more two-ship encountering scenarios).

Figure 19 illustrates the setup of the multi-ship simulation. According to this section, we confirmed the successful performance of the CA-QMIX algorithm in various simulation scenarios and through thorough performance evaluations. Therefore, it can be concluded that the proposed CA-QMIX algorithm could enable ships to avoid collision and get to their target destinations in different scenarios. It also indicates that this study provides a more versatile decision-making model for intelligent ship behavior.
6. Conclusions

In this study, an intelligent ship behavior decision-making method is proposed for multi-ship collision avoidance based on the multi-agent reinforcement learning algorithm, which could ensure the safety of a ship’s voyage in different multi-ship encounter scenarios.

To reduce collision risk, this paper performs collision detection using the ORCA (Optimal Reciprocal Collision Avoidance) method. The proposed algorithm adopts the CTDE framework and DEC-POMDP model to train the agents to avoid collision. Then, the multi-ship model was trained on the rich encountering situations based on the CA-QMIX algorithm. Furthermore, multi-ship collision avoidance also needed to comply with COLREGs. Hence, we designed a novel reward function for solving the collision problem, and as a result, changing course complied with COLREGs. To improve the efficiency of training, the procedure of the reward functions was defined. Subsequently, multiple ships were trained on different scenarios defined by COLREGs.

In the simulations, the proposed algorithm was validated in various simulated scenarios, and its performance was evaluated by the navigational trajectories, the rudder angle and speed. The results of simulation on various scenarios indicated that the proposed algorithm could implement multi-ship collision avoidance while complying with COLREGs and incorporating rudder characteristics. This algorithm demonstrated its flexibility and scalability. It was therefore able to be applied to a wide range of tasks.

For future work, we will focus on improving the ship motion mathematical model. The model-based multiple-agents reinforcement learning can achieve a good sample efficiency and a stable performance. For use in the real world, a more accurate model will be integrated to enhance the maneuverability of a ship. In addition, the collision-free path should be optimized to improve navigational efficiency. Finally, a hardware simulation...
will be implemented for verifying the feasibility of multi-ship collision avoidance, and the
simulation results will be compared with other relevant methods.

**Author Contributions:** Conceptualization, G.W. and W.K.; methodology, W.K.; software, W.K.;
validation, G.W.; formal analysis, G.W. and W.K.; investigation, G.W. and W.K.; writing—original
draft preparation, W.K.; writing—review and editing, G.W. and W.K.; supervision, G.W.; project
administration, G.W.; funding acquisition, G.W. All authors have read and agreed to the published
version of the manuscript.

**Funding:** The paper is partially supported by National Natural Science Foundation of China
(NO. 51409033, 52171342), and the Fundamental Research Funds for the Central Universities
(NO. 3132019343). The authors would like to thank the anonymous reviews for their valuable comments.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available from the
corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

**References**

1. Guan, W.; Peng, H.; Zhang, X.; Sun, H. Ship Steering Adaptive CGS Control Based on EKF Identification Method. *J. Mar. Sci. Eng.* **2022**, *10*, 294. [CrossRef]
2. Statheros, T.; Howells, G.; Maier, K.M.D. Autonomous ship collision avoidance navigation concepts, technologies and techniques. *J. Navig.* **2008**, *61*, 129–142. [CrossRef]
3. Zhao, D.; Shao, K. Deep reinforcement learning overview: The development of computer go. *Control. Theory Appl.* **2016**, *6*, 17. [CrossRef]
4. Liu, Q.; Zhai, J. A brief overview of deep reinforcement learning. *Chin. J. Comput.* **2018**, *1*, 27. [CrossRef]
5. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vision* **2015**, *115*, 211–252. [CrossRef]
6. Graves, A.; Mohamed, A.R.; Hinton, G. Speech Recognition with Deep Recurrent Neural Networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; Volume 38, pp. 6645–6649. [CrossRef]
7. Sutton, R.S.; Barto, A.G. Reinforcement learning: An introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [CrossRef]
8. Miele, A.; Wang, T. Maximin approach to the ship collision avoidance problem via multiple-subarc sequential gradient-restoration algorithm. *J. Optim. Theory Appl.* **2005**, *124*, 29–53. [CrossRef]
9. Phanthong, T.; Maki, T.; Ura, T.; Sakamaki, T.; Aiyarak, P. Application, Application of A* algorithm for real-time path re-planning of an unmanned surface vehicle avoiding underwater obstacles. *J. Mar. Sci. Appl.* **2014**, *13*, 105–116. [CrossRef]
10. Cheng, X.; Liu, Z.; Zhang, X. In Trajectory Optimization for ship Collision Avoidance System Using Genetic Algorithm. In Proceedings of the OCEANS 2006-Asia Pacific, Singapore, 16–19 May 2006; pp. 1–5. [CrossRef]
11. Liang, C.; Zhang, X.; Watanabe, Y.; Deng, Y. Autonomous collision avoidance of unmanned surface vehicles based on improved a star and minimum course alteration algorithms. *Appl. Ocean. Res.* **2021**, *113*, 102755. [CrossRef]
12. Wilson, P.A.; Harris, C.J.; Hong, X. A line of sight counteraction navigation algorithm for ship encounter collision avoidance. *J. Navig.* **2003**, *56*, 111–121. [CrossRef]
13. Chen, Y.-Y.; Ellis-Tiew, M.-Z.; Chen, W.-C.; Wang, C.-Z. Fuzzy risk evaluation and collision avoidance control of unmanned surface vessels. *Appl. Sci.* **2021**, *11*, 6338. [CrossRef]
14. Johansen, T.A.; Perez, T.; Cristofaro, A. Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3407–3422. [CrossRef]
15. Hu, L.; Naeem, W.; Rajabally, E.; Watson, G.; Mills, T.; Bhuiyan, Z.; Raeburn, C.; Salter, I.; Pekcan, C. A multiobjective optimization approach for COLREGs-compliant path planning of autonomous surface vehicles verified on networked bridge simulators. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 1167–1179. [CrossRef]
16. Shen, H.; Hashimoto, H.; Matsuda, A.; Taniguchi, Y.; Terada, D.; Guo, C. Automatic collision avoidance of multiple ships based on deep Q-learning. *Appl. Ocean. Res.* **2019**, *86*, 268–288. [CrossRef]
17. Sawada, R.; Sato, K.; Majima, T. Automatic ship collision avoidance using deep reinforcement learning with LSTM in continuous action spaces. *J. Mar. Sci. Technol.* **2021**, *26*, 509–524. [CrossRef]
18. Li, L.; Wu, D.; Huang, Y.; Yuan, Z.-M. A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field. *Appl. Ocean. Res.* **2021**, *113*, 102759. [CrossRef]
19. Zhao, L.; Roh, M.-I. COLREGs-compliant multiship collision avoidance based on deep reinforcement learning. *Ocean. Eng.* **2019**, *191*, 106436. [CrossRef]
20. Xu, X.; Lu, Y.; Liu, X.; Zhang, W. Intelligent collision avoidance algorithms for USVs via deep reinforcement learning under COLREGs. Ocean. Eng. 2020, 217, 107704. [CrossRef]

21. Fossen, T.I. Guidance and Control of Ocean Vehicles; John Wiley and Sons: Chichester, UK, 1999; ISBN 0 471 94113 1.

22. Perez, T.; Ross, A.; Fossen, T. A 4-dof Simulink Model of a Coastal Patrol Vessel for Manoeuvring in Waves. In Proceedings of the 7th IFAC Conference on Manoeuvring and Control of Marine Craft, International Federation for Automatic Control, Lisbon, Portugal, 20–22 September 2006; pp. 1–6.

23. Alonso-Mora, J.; Breitenmoser, A.; Ruffi, M.; Beardsley, P.; Siegwart, R. Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots; Springer: Berlin/Heidelberg, Germany, 2013; pp. 203–216. [CrossRef]

24. ´Smierzchalski, R. Ships’ Domains as Collision Risk at Sea in the Evolutionary Method of Trajectory Planning. In Information Processing and Security Systems; Springer: Berlin/Heidelberg, Germany, 2005; pp. 411–422. [CrossRef]

25. Oliehoek, F.A.; Span, M.T.; Vlassis, N. Optimal and approximate q-value functions for decentralized POMDPs. J. Artif. Intell. Res. 2008, 32, 289–353. [CrossRef]

26. Oliehoek, F.A.; Amato, C. A Concise Introduction to Decentralized POMDPs; Springer: Berlin/Heidelberg, Germany, 2016.

27. Busoniu, L.; Babuška, R.; De Schutter, B. Multi-Agent Reinforcement Learning: An Overview. In Innovations in Multi-Agent Systems and Applications-1; Springer: Berlin/Heidelberg, Germany, 2008; Volume 38, pp. 156–172. [CrossRef]

28. Sunehag, P.; Lever, G.; Gruslys, A.; Czarnecki, W.M.; Zambaldi, V.; Jaderberg, M.; Lanctot, M.; Sonnerat, N.; Leibo, J.Z.; Tuyls, K.; et al. Value-decomposition networks for cooperative multi-agent learning. arXiv 2017, arXiv:1706.05296. [CrossRef]

29. Rashid, T.; Samvelyan, M.; Schroeder, C.; Farquhar, G.; Foerster, J.; Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 4295–4304.

30. Ha, D.; Dai, A.; Le, Q. Hypernetworks. arXiv 2016, arXiv:1609.09106.

31. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G. Human-level control through deep reinforcement learning. Nature 2015, 518, 529–533. [CrossRef] [PubMed]