A collision avoidance decision-makingalgorithm for Tokyo Bay ferry based on deep renforcement learning

Master's Thesis

A COLLISION AVOIDANCE DECISION-MAKING ALGORITHM FOR TOKYO BAY FERRY BASED ON DEEP REINFORCEMENT LEARNING

September 2022

Graduate School of Marine Science and Technology

Tokyo University of Marine Science and Technology

Master's Course of Maritime Technology and Logistics

WU CHAOYANG

Master's Thesis

A COLLISION AVOIDANCE DECISION-MAKING ALGORITHM FOR TOKYO BAY FERRY BASED ON DEEP REINFORCEMENT LEARNING

September 2022

Graduate School of Marine Science and Technology

Tokyo University of Marine Science and Technology

Master's Course of Maritime Technology and Logistics

WU CHAOYANG

Abstract

Ship collision avoidance is the core issue of ship navigation safety. Deep reinforcement learning algorithm has a broad application prospect in the field of ship collision avoidance, which improves the flexibility and adaptability of ships in the navigation environment. Based on DDQN (Double Deep Q Network) algorithm in deep reinforcement learning, this thesis proposes a ship collision avoidance decision algorithm suitable for Tokyo Bay dangerous sea area, which effectively improves the navigation safety of the sea area. The main research contents of this thesis are as follows:

Firstly, according to the existing research results, deficiencies are put forward from three aspects: the application value, timeliness and process nature. The basic principle of deep reinforcement learning and its advantages in ship collision avoidance decision-making are also introduced in detail.

Secondly, a highly real ship simulation model is constructed. To improve the application value of the algorithm, the model combines the actual navigation data provided by Automatic Identification System (AIS), embeds the Tokyo Bay ferry Bumper model and considers the Japanese Maritime Traffic Safety Law.

Then, the framework of ship collision avoidance decision algorithm is established from the four aspects of state space, action space, reward function and network structure. After establishing the ship collision avoidance simulation model using python language, the AIS data of ships in different dangerous situations are used as the input of the model. With continuous training of DDQN neural network, the automatic collision avoidance strategies of ships in various collision avoidance situations are obtained and stored in the database. For the training results, this thesis shows the ship navigation path under different reward values with the increase of training times in the situation of two ships and multiple ships meeting, which reflects the process nature of the algorithm.

Finally, the ship collision avoidance simulation model is applied to actual navigation. The database can provide the corresponding navigation strategy for the pilot according to the real-time AIS information, so that the ferry can avoid collision and reach the target point safely and effectively, ensuring the timeliness of the algorithm.

Key Words: Ship collision avoidance; Deep Reinforcement Learning; DDQN algorithm

Content

Abstract 1
1 Introduction 1
1.1 Background 1
1.2 Related Research 2
1.3 Research Content and Technology Route 6
2 Deep Reinforcement Learning 8
2.1 Reinforcement Learning 8
2.1.1 Markov Decision Process 8
2.1.2 Value-based reinforcement learning 9
2.2 Deep Reinforcement Learning 10
2.2.1 DQN Algorithm 10
2.2.2 DDQN Algorithm 12
2.3 Summary 13
3 Ship Simulation Model 15
3.1 AIS 15
3.1.1 The Component of AIS 15
3.1.2 Influence of AIS on ship collision avoidance 15
3.2 Maritime Traffic Safety Law 16
3.2.1 Ship encounter situations 16
3.2.2 Regulations 17
3.3 Ship domain model based on AIS and regulations 17
3.3.1 Ship domain model 17
3.3.2 Ferry's ship domain model 18
3.4 Summary 22
4 Algorithm Design and Simulation Training 24
4.1 Algorithm Design 24
4.1.1 State Space Design 24
4.1.2 Action Space Design 24
4.1.3 Reward Function Design 25
4.1.4 Network Structure Design 26

4.1.5 Model Process 28
4.2 Simulation Platform 29
4.3 Simulation Training Experiment 30
4.4 Experiment Result Analysis 32
4.4.1 2-ship Case 32
4.4.2 3-ship Case 33
4.5 Summary 35
5 Application Verification Experiment 36
5.1 Application Background 36
5.2 Verification for Tokyo-Wan Ferry Routes 37
5.3 Summary 39
6 Conclusion and Prospect 40
6.1 Conclusion 40
6.2 Prospect 41
Reference 42
Acknowledgement 46

1 Introduction

1.1 Background

In recent years, with the development of the economy, trade between countries is still very frequent, although it is greatly affected by epidemic situation and other factors. Among them, shipping takes up about 80% of the global trade in goods, which makes the maritime transportation activities still frequent. A large number of ships enter and leave the port, leading to frequent maritime traffic accidents. The collision between ships not only threatens the life safety of crews and passengers, but also cause huge losses to goods and pollutes the marine environment. According to relevant data[1], about 80% of maritime traffic accidents are caused by human errors of pilots. In order to ensure the safety of ships, crews, passengers and cargo, and prevent the marine environment from being polluted, it is extremely important to study the automation and intelligence technology of ships.

The ship collision avoidance is a key to ensure navigation safety. With the increasing degree of automation of ships and the continuous development of navigation technology, ships can complete automatic collision avoidance operations through navigation aids such as AIS, Electronic Chart Display and Information System (ECDIS) and collision avoidance algorithm. At present, the intelligent decision-making of ship collision avoidance based on AIS, the ship collision avoidance based on evolutionary genetic algorithm of intelligent algorithm and the ship collision avoidance algorithm based on Bayesian network all have certain capabilities to solve the problem of ship collision avoidance, but they also have their limitations. They cannot learn and improve the collision avoidance strategy by themselves.

Deep Reinforcement Learning[2] (DRL) is a rapidly developing artificial intelligence technology in recent years, which plays an important role in various fields. At present, deep reinforcement learning has been widely used in the fields of robot control[3], automatic driving[4-6], financial forecasting[7], traffic control[8] and so on. Artificial intelligence technology is gradually infiltrating into various fields, and the concept of unmanned driving is getting closer and closer. In the aspect of ship collision avoidance, the deep reinforcement learning algorithm can process a large amount of ship navigation data through deep learning[9], and then generate collision avoidance strategies through reinforcement learning[10] to deal with all kinds of encounter situations.

Tokyo Bay is an important port in Japan, and Uraga Traffic Route is the most important waterway in Tokyo Bay. At the south entrance of Uraga Traffic Route, there is a cross-cutting ferry route: Kurihama-Kanaya route. There are two scheduled ferry flights on this route-Tokyo-Wan Ferry, which will cross with ships entering and leaving the Uraga traffic route. If there is a slight negligence in human operation, there will be a certain risk of collision. Therefore, this thesis proposes a decision-making algorithm of ship collision avoidance based on deep reinforcement learning to avoid the collision between ferries and other ships.

In view of the above background, this thesis based on the DDQN (Double Deep Q Network) algorithm[11] of deep reinforcement learning, fully considering the Maritime Traffic Safety Law[12] of Tokyo Bay, provides a real-time collision avoidance strategy for ferry pilots during the voyage, which avoids the safety problems caused by pilots' inexperience or poor vision, reduces the collision risk of the ship in various encounter, and has certain economic value and broad application prospects.

1.2 Related Research

In 1950s and 1960s, scholars at home and abroad began to study the ship collision avoidance, and put forward related concepts such as ship domain, distance of closest point of approach (DCPA), time of closest point of approach (TCPA)[13]. With the development of science and technology, many navigational aids have also been developed. First, the Vessel Traffic Service (VTS), which collects target signals through shore-based radar, was developed. In the 1960 s, the Automatic Radar Plotting Aid (ARPA) was developed for ship collision avoidance. ARPA often forms a system with radar, and calculates the navigation information such as ship's heading and speed, so as to predict whether there is a risk of collision with the surrounding ship or obstacles. In 1980s, the expert system was applied to the field of ship collision avoidance. The famous collision avoidance expert system proposed by Liverpool University and Tokyo Merchant Marine University has solved the collision avoidance problem of ship in common encounter situations through a large amount of prior knowledge to choose routes. In the 1990s, Zhaolin Wu and Zhongyi Zheng[14-16] proposed a decision-making method of collision avoidance based on fuzzy theory and expert knowledge. By combining fuzzy mathematics theory and practical investigation method, a model of the best turning range and action time was constructed, which laid the foundation for the study of ship collision avoidance.

On the basis of predecessors, many classic algorithms of ship collision avoidance have emerged. Petres et al.[17] used artificial potential field method to make ships construct virtual repulsive force to avoid obstacles under the influence of wind speed changes. The artificial potential field method has the advantages of high computational efficiency and simple algorithm, but if the parameters of the force field are set unreasonably, there is the disadvantage of falling into the local minimum value. In order to avoid this problem, Song et al. studied the collision avoidance of ships by the Fast March method[18, 19], and proposed a collision avoidance method[20-22] that considers the characteristics of the ship’s motion by adding the bow guidance area, and completed the real ship test. Considering the current interference, a multi-layer fast step collision avoidance navigation algorithm is also proposed. Its advantage is that it can plan the path according to different optimization criteria, and the simulation results verify the effectiveness of the algorithm. Sun et al.[23] also proposed an automatic collision avoidance method which can change the speed and heading of the ship at the same time by transforming the static guide area of the ship head into a dynamic domain based on the fast step method. Taking the local waters area of a port as a simulated navigation area, the collision avoidance

experiments under three kinds of encounter situations are carried out, and the results show that this method has good obstacle avoidance ability. Zhang et al.[24-27] used the near-field diagram method and fuzzy theory to study obstacle avoidance in complex environment, and realized the accurate and rapid collision avoidance operation of ships in complex environment, so as to reach the target safely. Phanthong et al.[28] proposed a real-time path replanning collision avoidance algorithm combining heuristic A * algorithm and forward-looking multi-wave velocity sonar, which can simultaneously avoid the surrounding ships and obstacles. Based on the velocity obstacle method, Tian et al.[29] proposed a multi-ship automatic collision avoidance algorithm that can avoid ships and obstacles simultaneously through simulation on electronic chart.

But the above methods do not consider the Convention on the International Regulation for Preventing Collisions at Sea[30] (COLREGs). According to statistics[31], 56% of ship collision accidents are caused by violation of collision avoidance rules. Therefore, it is very important to consider the requirements of navigation collision avoidance rules when studying ship automatic collision avoidance algorithms[32].

At present, under the premise of considering the requirements of collision avoidance rules, there are many research results on ship collision avoidance[33-35]. Shtay et al.[36] realized the automatic collision avoidance of two ships by fuzzy logic method. Based on the improved artificial potential field method, Lee et al.[37] reasonably designed the fuzzy expert rule set to avoid other ships according to the rules. Based on fuzzy logic method, Perera et al.[32, 38] established a set of complete fuzzy logic rules to follow the collision avoidance rules by introducing the experienced actions of the pilots, and made collision avoidance experiment with ship models and simulated ships, which achieved good results. Because the method based on fuzzy logic needs to artificially set up expert experience, it is difficult to establish appropriate fuzzy rules in complex multi-ship encounter situations. By combining reactive collision avoidance with LOS (Line of Sight) path tracking method, Moe et al.[39] uses set-based theory to realize path tracking and collision avoidance mode switching, which helps the two ships navigate safely according to collision avoidance rules. Abdelaal et al.[40] regarded the problem of ship collision avoidance as a nonlinear optimization problem and took collision avoidance rules as constraints. The model predictive control method was used to solve this problem. Finally, the effectiveness of the proposed collision avoidance algorithm was verified by simulation examples of ships in different encounter situations. Benjamin et al.[41] regarded the problem of ship collision avoidance as a multi-objective optimization problem, and used the way of controlling behaviour and the method of interval planning to realize the constraint of navigation rules. Under the premise of complying with the collision avoidance rules, the algorithm verification was completed by simulation and real ship experiments.

With the rapid development of artificial intelligence theory and algorithms, many scholars apply intelligent algorithms to the field of ship collision avoidance. At the beginning of this century, Harris[42] proposed a method to study ship collision avoidance using neural networks. Smierzchalski[43] used the genetic algorithm to plan the path of ship collision avoidance to ensure the safety of ship navigation. Yang[44] realized a ship automatic collision avoidance agent

platform based on the technology and theory of multi-agent system. Zhuo et al.[35] used fuzzy neural network to realize collision avoidance decision by assisting pilots. Li Na Li, Guoquan Chen and so on[45-48] combined machine learning, expert system, fuzzy mathematics and mathematical analysis, resumed the ship humanoid intelligent collision avoidance decision model and optimized the ship collision avoidance decision scheme. Deep reinforcement learning has become a hot topic with the breakthrough in the computer field. In the field of ship collision avoidance, Zhao et al.[49] constructed an adaptive heuristic evaluation algorithm for ship collision avoidance based on the real-time acquisition of state information between ships. Through the reward signal, the optimal steering decision of ship from encounter formation to passing clearance dynamic process is obtained by reinforcement learning training, which can meet the collision avoidance rules and complete the safe collision avoidance. Compared with the existing distributed collision avoidance decision method, it has better real-time and economy. Haiqing Shen et al.[50] proposed a new multi-ship automatic collision avoidance method based on deep reinforcement learning, especially in restricted waters. A ship collision avoidance training method and algorithm combining ship manoeuvrability, human experience and navigation rules are introduced in detail. Through systematic numerical verification and experimental verification, it is shown that the developed ship automatic collision avoidance method based on deep reinforcement learning has great feasibility in a highly complex navigation environment. Wang et al.[51] established a path planning model of unmanned ship based on Q-Learning to realize the adaptive navigation of unmanned ship in unknown environment. Using the Q-learning algorithm based on Markov process and python and pygame platform, the simulation environment is established to effectively plan the better path in the unknown environment and avoid multiple obstacles. Zhang et al.[52] studied the adaptive navigation problem of autonomous surface ship under uncertain environment. In order to realize the intelligent obstacle avoidance of port MASSs, an autonomous navigation decision model based on hierarchical deep reinforcement learning is proposed. The model is mainly composed of two layers: scene partition layer and autonomous navigation decision-making layer. The scene division layer quantizes the sub-scenes according to the COLREGs. The navigation situations of ships are divided into entities and attributes. A deep Q-learning algorithm is designed at the decision-making level. By using the environmental model, ship motion space, reward function and search strategy, the environmental state is learned in the quantized sub-scene, and the navigation strategy is trained. The improved deep reinforcement learning algorithm effectively improves the navigation safety and collision avoidance performance of ships. Zhou et al.[53] proposed a ship intelligent collision avoidance algorithm based on deep Q network (DQN) reinforcement learning method. The simulation experiments were carried out for the encounter situation of two ships and multiple ships, which provided a reference for ships to effectively avoid incoming ships under the requirements of COLREGs. Guo et al.[54] proposed an autonomous path planning model based on the combination of the Deep Deterministic Policy Gradient (DDPG) algorithm and the artificial potential field method, which transformed the navigation rules and ship encounter into the restricted navigation area to ensure the validity and accuracy of the model. The established comparative experiments show

that the improved model can realize the autonomous path planning and realize the intelligent path planning of ships in the unknown environment. Joohyun Woo et al.[55] proposed a ship collision avoidance method based on deep reinforcement learning. This method is applicable to the decision-making stage of collision avoidance. If it is necessary to avoid collision, the direction of collision avoidance operation should be determined. In addition, by using the visual recognition ability of deep neural network, the complex fuzzy situations encountered by ships are analyzed, and a grid graph representation method for ship encounters is proposed. Wu et al.[56] proposed an Autonomous Navigation and Obstacle Avoidance (ANOA) method based on deep reinforcement learning to solve the obstacle avoidance problem of ship autonomous navigation. The ANOA was combined with the actual control model of ship surging, swaying and yawing to obtain higher success rate in dynamic environment.

Based on the above analysis of the research on ship collision avoidance at home and abroad, it can be seen that ship collision avoidance is gradually developing towards intelligence and automation. At present, the application of deep reinforcement learning in ship collision avoidance has overcome the problems of dimension disaster and limited state space of reinforcement learning, and improves the ship's self-learning and adaptive ability in navigation environment. However, it still faces the following problems:

(1) Application value. Most of the current studies only verify the effectiveness of the algorithm through simulation experiments, without considering the actual navigation environment and real ship data and other information, so the algorithm cannot be applied to actual maritime navigation.

(2) Timeliness. The long training cycle of the algorithms and the high cost in the real training environment cannot provide a real-time collision avoidance strategy in time.

(3) Process Nature. Most of the simulation experiments of algorithms only show the final results of the training, but do not show the process of reinforcement learning self-learning and optimization, which lacks certain persuasion.

In view of the existing problems, this thesis makes corresponding improvements. In terms of application value, this thesis takes Tokyo-Wan Ferry Route as an example, combined with AIS information and ship data, constructs a ship collision avoidance decision algorithm model that can effectively reduce the collision risk of the route. In terms of timeliness, the database stores the training of a large number of ships in the past few years, in order to obtain the optimal collision avoidance strategy under the corresponding situation, and gives the corresponding strategy according to the real-time navigation environment during ferry sailing, saving a lot of time and the training process. In terms of process nature, the simulation experiment in this thesis not only gives the final training results, but also shows the process of exploring the maximum reward value in the training process. By analyzing the value of the reward function, the optimization idea of the algorithm is explained.

1.3 Research Content and Technology Route

The main research content of this thesis is the application of deep reinforcement learning algorithm for ship collision avoidance in high collision risk waters. In this thesis, a ferry collision avoidance decision-making algorithm based on DDQN deep reinforcement learning algorithm is proposed. Taking the Tokyo-Wan ferry route as the scene, a highly simulated navigation environment is established. The ferry automatic collision avoidance model is obtained by training, which provides a safe and effective navigation guarantee for ferry. The main contents of this thesis are as follows:

Chapter 1: Introduction. This chapter mainly introduces the research background and significance of this thesis, and analyzes the research results of scholars at home and abroad in the field of ship collision avoidance and deep reinforcement learning.

Chapter 2: Deep reinforcement learning algorithm. This chapter introduces the relevant theoretical knowledge of deep reinforcement learning, from the basic framework of reinforcement learning, Markov decision process, reinforcement learning algorithm based on value function to the principles and advantages of DQN algorithm and DDQN algorithm in deep reinforcement learning.

Chapter 3: Ship simulation model based on AIS and rules. Firstly, this chapter introduces AIS, an important source data in this thesis, and analyzes the influnce of AIS on ship collision avoidance. Secondly, it considers the maritime traffic safety law of Tokyo Bay, and divides and simplifies the encounter situations according to the laws and regulations. Finally, the knowledge of the ship domain is introduced, AIS and maritime traffic safety law are integrated into the ship domain model, and the ship domain model of ferry is constructed in the simulation environment.

Chapter 4: Ship collision avoidance decision-making algorithm design and simulation experiments. Firstly, the state space, action space, reward function and network structure of the ship collision avoidance decision-making algorithm are designed. Then, the python is used to build the simulation platform, and the AIS data is input into the model for training. The results of different encounter situations are randomly selected and analyzed, and compared with the manual navigation route.

Chapter 5: Application verification of ship collision avoidance decision-making algorithm. Firstly, this chapter introduces the background of the south entrance of the Tokyo Bay Uraga Traffic Route, and then applies the trained ferry automatic collision avoidance strategy model to real-time data for verification, which provides a new strategy for the operation of the Tokyo-Wan ferry route.

Chapter 6: Conclusion and prospect. This chapter summarizes the work of the full thesis, and prospects the future research work in view of the shortcomings of this study.

The technology route of this thesis is shown in Fig. 1-1:

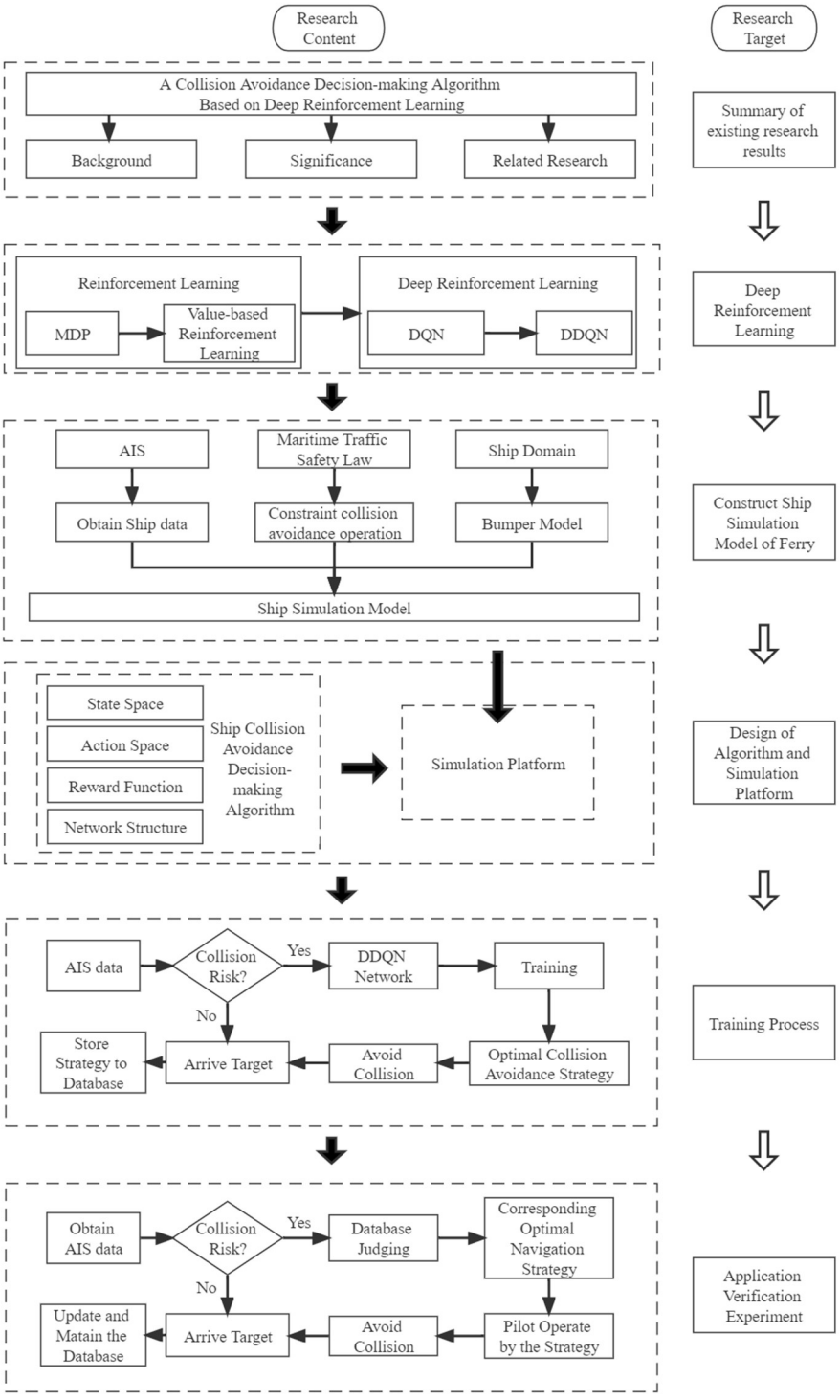

Fig.1-1 Technology route

2 Deep Reinforcement Learning

2.1 Reinforcement Learning

Reinforcement Learning (RL) discusses how an agent can get its maximum reward in a complex and uncertain environment. In the process of reinforcement learning, the agent interacts with the environment. The agent gets a state in the environment and use it to output an action and a decision. Then this decision will be put into the environment, and the environment will output the reward for the next state and the current decision according to the decision taken by the agent. The purpose of the agent is to get as much reward as possible from the environment.

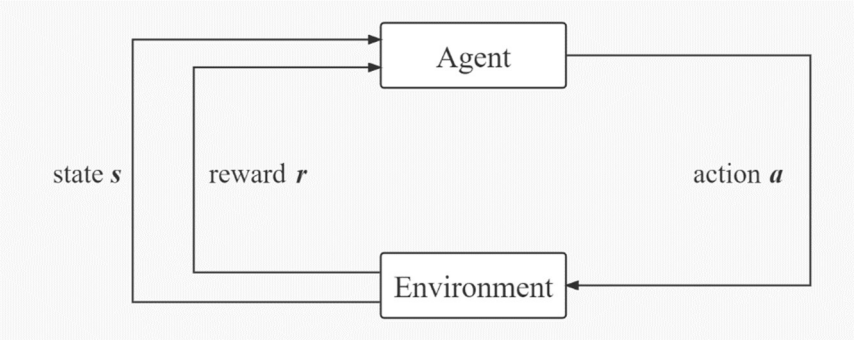

Fig. 2-1 The concept of Reinforcement Learning

2.1.1 Markov Decision Process

Most reinforcement learning problems can be described by Markov Decision Process (MDP) model. Markov decision-making process is described by tuples (S, A, P, r, γ) , where:

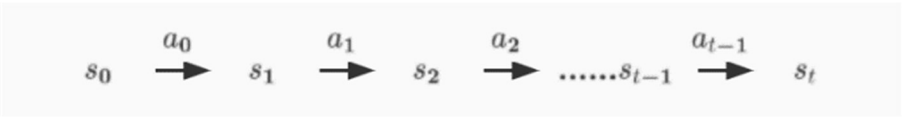

Fig.2-2 The transition of Markov states

(1) S represents the state space. The state in the Markov decision process satisfies the Markov property, that is, the next state of the system is only related to the current state and independent of the historical state. It can be seen from Fig. 2-2 that state s can represent the information of historical changes, so historical changes need not be stored.

(2) A represents action space. The ultimate goal of reinforcement learning is to choose the action that maximizes the cumulative reward value of the whole decision-making process.

(3) P is the state transition probability, that is, under the premise of the current state s

and action a, the probability of the next state is s'.

$$P(s,s') = P_r(s_{t+1} = s' | s_t = s,\ a_t = a) \tag{2-1}$$

(4) $R(s,s',a)$ represents the reward function. The reward is feedback from the environment to the agent, and the feedback reward is only related to the current state and action, such as the score of the game.

(5) $\gamma \in (0,1)$ represents the discount factor when calculating the cumulative reward. The earlier the action, the smaller the impact on the current situation. Therefore, it is often necessary to add a discount factor to calculate the cumulative reward, so that the current reward R decays to $\gamma^{k-1}R$ after k time steps.

In a certain state s_t, the agent chooses action a according to the strategy. The probability of action distribution in this state is expressed by π, which is called strategy. Usually, nonlinear function is used to approximate:

$$\pi(a \mid s) = P(a_t = a \mid s_t = s) \tag{2-2}$$

Actually, reinforcement learning algorithm is the process of finding the strategy that can obtain the maximum reward in a given MDP.

2.1.2 Value-based reinforcement learning

The agent of reinforcement learning algorithm can not perceive the future value through feedback reward, and can only know the current action. Therefore, value-based reinforcement learning algorithm is introduced. Reinforcement learning algorithms are roughly divided into two categories: value-based method and policy-based method. In the value-based method, the value function enables the agent to evaluate actions and future values, that is, the ability to select actions with the max future value rather than those with the max current value.

Value function includes state value function and state-action value function, which are used to evaluate the value of a certain state and the value of taking a certain action in a certain state.

The state value function $V^{\pi}(s)$ represents the interaction with the environment under the policy π. When the state s appears, the expected value of the next cumulative reward can be obtained, as shown in Equation 2-3:

$$V^{\pi}(s) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s] \tag{2-3}$$

Accordingly, the state-action value function $Q^{\pi}(s,a)$ represents that the state s takes action a, and then the strategy π is used to interact with the environment, and the cumulative reward expectation can be obtained, as shown in Equation 2–4:

$$Q^{\pi}(s,a) = E_{\pi}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) | s_0 = s, a_0 = a] \tag{2-4}$$

The value function corresponds to the evaluated strategy one by one. The value-based reinforcement learning algorithm finds the corresponding optimal strategy π^* by calculating the optimal value function. The Bellman optimal equations of the optimal value state function $V^*(s)$ and the optimal state-action value function $Q^*(s,a)$are shown in Equations 2–5 and 2–6 :

$$V^*(s) = \max_a R_s^a + \gamma \sum_{s' \in S} P_{ss}^a\ V^*(s') \tag{2-5}$$

$$Q^*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \max_{a'} Q^*(s',a') \tag{2-6}$$

According to the optimal state-action value function $Q^*(s,a)$, the optimal strategy π^* can be obtained by the greedy method :

$$\pi^*(a|s) = \begin{cases} 1, if\ a = \underset{a}{\operatorname{argmax}}\, Q^*(s,a) \\ 0, otherwise \end{cases} \tag{2-7}$$

In addition, in the process of reinforcement learning, "Exploration and Exploitation" are two core issues. Exploration refers to obtaining the best strategy by trying different actions. Exploitation refers to taking actions that are known to get the maximum reward. In fact, exploration and utilization are contradictory. To maximize the cumulative reward value, a good compromise between exploration and exploitation must be reached. This thesis adopts *ε-greedy* strategy to achieve:

$$\pi(a|s) = \begin{cases} 1 - \varepsilon + \frac{\varepsilon}{|A(s)|}, if\ a = \underset{a}{\operatorname{argmax}}\, Q(s,a) \\ \frac{\varepsilon}{|A(s)|}, otherwise \end{cases} \tag{2-8}$$

In the equation, $|A(s)|$ represents the number of optional actions in the action set.

The strategy requires that in each action selection, the non-optimal action is randomly selected with probability ε, and ε decreases with the increase of the number of learning iterations.

2.2 Deep Reinforcement Learning

In recent years, the deep reinforcement learning combined with deep learning and reinforcement learning has solved the problem of dimension disaster of traditional reinforcement learning algorithm in the face of high-dimensional state space and action space. Deep neural networks, which are widely used in deep learning, have powerful characterization and perception capabilities, and can deal with high-dimensional raw data such as images and audio. DRL uses deep neural network to process state data and fit strategy function, which combines the advantages of deep learning and reinforcement learning, and provides a new solution for complex decision-making problems.

The ship collision avoidance decision-making algorithm in this thesis is based on the DDQN deep reinforcement learning algorithm, so the DDQN algorithm and its predecessor DQN algorithm are briefly introduced.

2.2.1 DQN Algorithm

DQN algorithm[57] is one of the classic deep reinforcement learning algorithms, proposed by DeepMind team in 2013, is a model-free, value-based and end-to-end deep reinforcement learning algorithm. The algorithm framework of DQN comes from Q-learning algorithm. On the basis of Q-learning algorithm, deep neural network and special training mechanism are introduced. The reinforcement learning problem is skillfully transformed into a problem that can be trained by supervised learning methods, which opens a new era of deep reinforcement learning.

Q-learning method is a typical off-policy learning strategy, which separates the target strategy and the behavior strategy. The temporal difference method[58] (TD) is used to estimate the Q value, and the Q value of the current state is estimated by the Q value of the subsequent state. The Q value is updated with the learning rate α, as shown in Equation 2–9.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)) \tag{2-9}$$

$r_t + \gamma \max_a Q(s_{t+1}, a)$ is the TD target, $r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$ is TD deviation.

The traditional Q-learning algorithm expresses and stores the Q value of each state in Q-table, which is not suitable for high dimensional state space. Therefore, DQN has made three optimizations.

(1) Value Function Approximation

The value function approximation method is helpful to solve the dimension disaster problem caused by the huge state and continuous action of Q-learning in the actual task. Combined with the deep neural network, the value function is expressed as $Q(s, a; \theta)$, where θ is the weight of deep neural network.

(2) Experience Replay

Only using deep neural network approximation of state-action function, the result often does not converge. The DQN algorithm introduces the Experience Replay, and eliminates the correlation between samples by random sampling, which improves the performance of the algorithm. The experience replay constructs an Experience Replay Buffer D to save many sample data. Each group of data is saved as a transfer sample in (s, a, r, s') format, where s is the current state, a is a certain action taken under state s, r is the reward value obtained, and s' is the corresponding next state. The agent interacts with the environment many times in strategy π, and the collected data are stored in the Experience Replay Buffer D. The transfer samples in the Replay Buffer can be reused and distributed independently. When the Replay Buffer is filled, the data that first enters the Replay Buffer are automatically deleted. During training, the Mini-batch samples are randomly selected and the random gradient descent method is used to update the network weight to ensure the performance of the model.

(3) TD-based Target Network

Relying on the above two optimization techniques, DQN is not stable because the update of an action value function depends on other action value functions, and setting a target network based on TD helps to solve this problem. The target network is the same as the initial parameters of the current network, but only the current network is updated during training. The parameters of the target network are fixed and converted into regression problems. After a fixed number of times, the parameters of the current network and the target network are updated synchronously once. The process that the output value of the model is close to the target value is the process of minimizing the mean square error.

The loss function of current Q value and target Q value is:

$$L(\theta) = E[(\left(r + \gamma \max_{a'} Q(s', a'; \theta^-)\right) - Q(s, a; \theta))^2] \quad (2\text{-}10)$$

Where $(r + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_t; \theta^-))$ represents the target Q value, $Q(s, a; \theta)$ is the current Q value,

Update parameters by derivation of network weight parameters:

$$\nabla L = E[(\left(r + \gamma \max_{a'} Q(s', a'; \theta^-)\right) - Q(s, a; \theta))\nabla Q(s, a; \theta)] \quad (2\text{-}11)$$

After using these three important techniques, the *ε-greedy* strategy is also needed to avoid overfitting. A suitable ε value helps to explore better reward values and prevent falling into local optimum.

The complete framework of DQN algorithm is shown in Fig. 2–3.

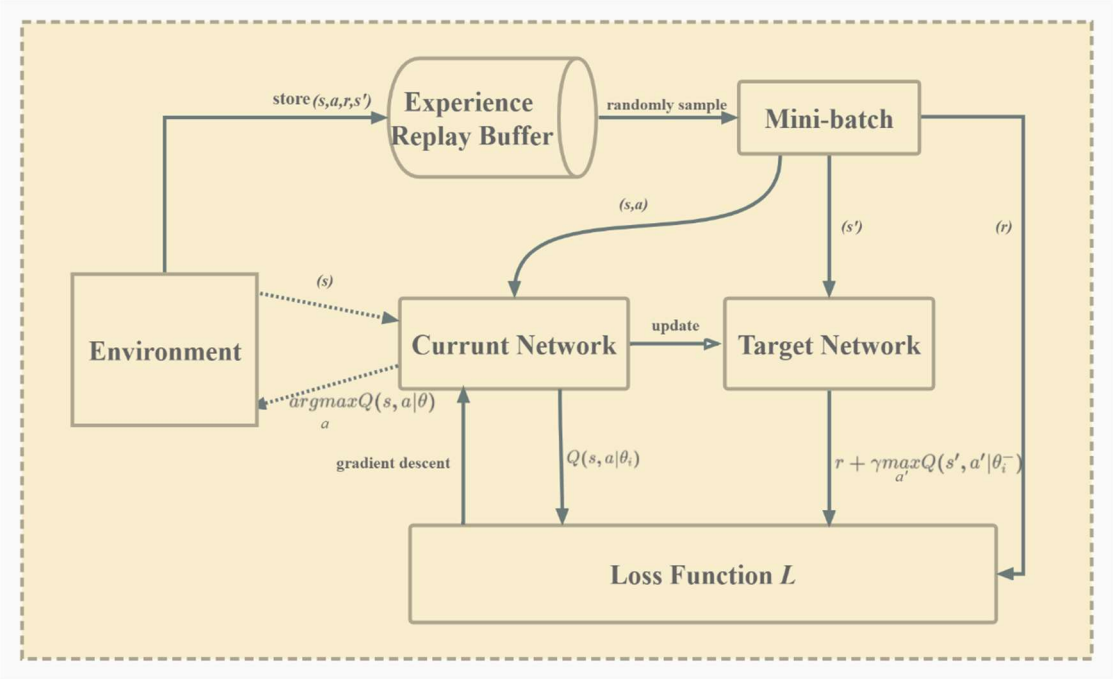

Fig.2-3 The structure of DQN algorithm

DQN algorithm successfully combines neural network with reinforcement learning by combining value function approximation method, establishing Experience Replay, setting up additional target network and reasonable "Exploration and Exploitation" strategy, which not only ensures the rationality of neural network output action Q value, but also improves the stability and versatility of the algorithm.

2.2.2 DDQN Algorithm

Although DQN algorithm has a great improvement in performance compared with Q-learning algorithm, its algorithm framework is still the same as Q-learning algorithm, so it has

the same overestimation problem as Q-learning algorithm. DDQN algorithm uses two value networks, one network is used to perform action selection, and then updates the current network with the action value of another value function. The action selection and value estimation are separated, which greatly reduces the overestimation problem and shortens the training time.

When setting the neural network with approximate median function in DDQN algorithm. $Q(s,a;\theta)$ represents the value function of the current network output, $Q(s,a;\theta^-)$ represents the target value function of the output of the target network, r represents the cumulative reward value of the migration state, and γ is the attenuation factor. The optimization formula of the target value network is shown in Formula 2–12.

$$r+\gamma Q\left(s',\underset{a}{\operatorname{argmax}}\,Q(s',a';\theta);\theta^-\right) \tag{2-12}$$

First, the action that enables the next state to obtain the maximum reward value is selected by the main network with the weight of θ, and then passed to the target network with the weight of θ^- for evaluation. The weight of the target value network is updated by the Experience Replay. The loss function of the current Q value and the target Q value is:

$$L(\theta)=E[(\left(r+\gamma Q\left(s',\underset{a}{\operatorname{argmax}}\,Q(s',a';\theta);\theta^-\right)\right)-Q(s,a;\theta))^2] \tag{2-13}$$

Compared with DQN algorithm, the training process of DDQN algorithm has no significant change, and almost no increase in the amount of computation. It also uses the two networks that DQN originally has. The only difference is that in the DQN algorithm, the action strategy of the current network is evaluated by the target network, while the DDQN algorithm uses the updated current network to evaluate the action, which is helpful to improve the stability of the algorithm.

DDQN algorithm as a value-based deep reinforcement learning algorithm, its action space is still discrete, in intelligent control[59], chess game[60], autonomous navigation[61] and other discrete strategy scenarios have a good application. In this thesis, when studying the automatic collision avoidance of ships, the training data is obtained through the AIS, the discrete action space is established, the simulation environment is built by python, and the reasonable reward function is formulated. The decision framework of ship collision avoidance based on DDQN algorithm is constructed and applied to the ship collision avoidance problem of Tokyo-Wan Ferry Route.

2.3 Summary

This chapter mainly introduces some basic theoretical knowledge related to deep reinforcement learning. Firstly, the Markov decision process is introduced, and the reinforcement learning method based on value function used in this thesis is expounded. Finally, the basic concept of deep reinforcement learning is briefly summarized, and the DQN algorithm is introduced in theory, and the DDQN algorithm is extended to the ship decision-making

problem.

3 Ship Simulation Model

In order to build a ship collision avoidance simulation environment in Tokyo Bay, it is necessary to integrate the knowledge of AIS, maritime traffic safety law and ship domain. Among them, AIS collects and provides ship navigation data, maritime traffic safety law regulates and restricts the navigation behavior between ferries and other ships in the encounter situations, and the ship domain judges and indicates the safe distance between ships.

3.1 AIS

Automatic Identification System is a new navigation aid system which can realize the automatic exchange, detection and recognition of ship information and navigation state between ships and shores. It consists of VHF (Very High Frequency), GPS locator and communication controller connected with ship monitors and sensors, which can be exchanged automatically. The AIS on the ship receives the information of other ships covered by VHF while sending important information such as ship position, speed, course, ship name, call sign, realizing automatic response. AIS as an open data transmission system, can connect with radar, electronic chart display and information system (ECDIS), ARPA, VTS and other terminal equipment and the Internet to form a maritime traffic management network. Even without radar detection, it can obtain traffic information and effectively reduce ship collision accidents.

3.1.1 The Component of AIS

AIS is composed of shore-side facilities and ship-borne equipment. The shore-side facilities of AIS established in the Tokyo Bay include maritime traffic center, maritime security department and AIS land-base station. The maritime traffic center provides navigation information and navigation control information for ships passing through waters and traffic route; AIS land-base station is responsible for receiving information from ships and sending necessary navigation information provided by maritime traffic center; the maritime security department is responsible for supporting maritime emergencies. In addition to receiving the navigation information sent by the shore-side facilities, the ship-borne equipment also exchanges the navigation data with the surrounding ships, including the name, call sign, ship speed and ship course.

3.1.2 Influence of AIS on ship collision avoidance

With the development of science and technology, AIS equipment is more and more advanced, the function is more and more complete. One of the biggest uses of AIS is to realize automatic identification by autonomous and continuous work in the surrounding area, and exchange the navigation information of the ship with the surrounding ship at all times, which solves the problem of obtaining ship information in ship collision avoidance.

However, AIS is only a ship navigation aid system, which cannot rely on itself to achieve

automatic collision avoidance of ships. In actual navigation, the pilot needs to rely on their own experience to judge the collision risk and avoid collision after getting the information of the surrounding ship. In order to realize that the ship can automatically provide a collision avoidance strategy for the pilot after receiving AIS data, it is also necessary to combine AIS with other algorithms. The DDQN algorithm studied in this thesis is just an algorithm to find the optimal strategy from the environment.

3.2 Maritime Traffic Safety Law

Usually, ships sailing at sea follow the provisions of the COLREGs, and the ferry route studied in this thesis is within the waters of the Tokyo Bay. In 1972, Japan Coast Guard has enacted the Maritime Traffic Safety Law in response to increasing congestion in the Tokyo Bay and the increase in the transport of dangerous goods to ensure maritime traffic safety. Therefore, the ship simulation model and automatic collision avoidance rules in this thesis will give priority to comply with the provisions of Maritime Traffic Safety Law.

3.2.1 Ship encounter situations

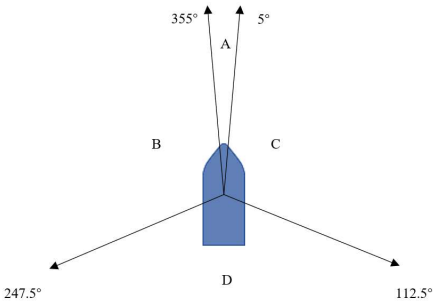

Fig.3-1 Regions of the ship encounter situation

Fig.3-1 is the division of ship encounter situations. Ship encounter situations are divided into head-on situation, crossing situation and overtaking situation. As shown in Fig. 3-1, t head-on situation is the range of A region (355°~5°), B and C regions (247.5°~355°, 5°~112.5°) are both crossing situations, D region (112.5°~247.5°) is called overtaking situation. In case of the head-on situation, do not distinguish the stand-on ship and the give-way ship, the two ships should take the right turn to avoid collision. In the crossing situation, if the target ship is on the starboard side of the own ship, the own ship needs to execute avoidance operation; if the target ship is located on the port side of the ship, the target ship needs to turn to avoid, and the own ship just needs to maintain speed and direction. If the target ship is located at a position bigger than 22° behind the ship beam and the speed is faster than the speed of the own ship, the overtaking situation is formed. The own ship should sail at a constant speed and direction, and the overtaking ship should steer and avoid. This thesis only considers the encounter situation in the case of good sea visibility.

3.2.2 Regulations

This thesis mainly studies the dangerous encounter situations of the ferry on the Tokyo-Wan Ferry Route and the ships entering and leaving the south entrance of the Uraga Traffic Route. The Uraga Traffic Route is a route to implement the traffic separation scheme. The ships entering and leaving the traffic route entrance do not have enough space and time to avoid the collision. For the provisions of Maritime Traffic Safety Law, this thesis focuses on two regulations:

(1) When the ferry in the Tokyo Bay crosses the route, it always takes the ferry as give-way ship, giving priority to ensuring the sailing of ships entering and leaving the Uraga Traffic Route.

(2) When crossing the route, the ferry sails as close to the right angle as possible.

Therefore, the ship collision avoidance problem studied in this thesis can be simplified as the case of only analyzing the range B or C of other ships when the ferry is used as give-way ship, that is, only considering the crossing situation. Ships entering and leaving the Uraga Traffic Route need not consider ship collision avoidance. Ferries need to continuously cross two opposite ship traffic flow, so it is easy to encounter multiple ships with collision risk.

3.3 Ship domain model based on AIS and regulations

3.3.1 Ship domain model

Ship domain refers to the area around itself which other ships and objects try to avoid entering. It was proposed by the Japanese scholar Fujii Yahei et al.[13] in the 1960s, and then a large number of scholars such as Goodwin[62], Coldwell[63] carried out further research and constructed many ship domain models with different sizes and shapes. Now it is widely used in collision avoidance decision-making and risk assessment.

This thesis selects the Bumper model[64] applicable to the Tokyo Bay. The classic Bumper model is shown in Fig. 3-2:

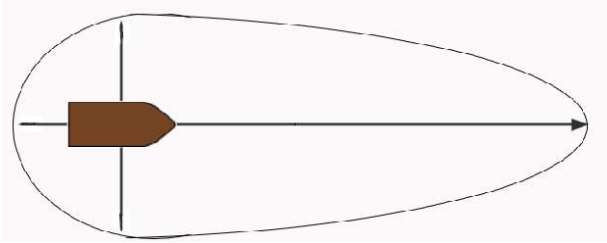

Fig.3-2 Typical Bumper model

Because the simulation experiment is carried out in the grid navigation environment, we also improve the Bumper model to rectangular Bumper model, as shown in Fig. 3-3:

Fig.3-3 Rectangular Bumper model

The ships in this thesis use the rectangular Bumper model when sailing. Once the Bumper regions of the two ships overlap, they are judged to be ships with collision risk.

3.3.2 Ferry's ship domain model

Information of two ferries serving on the Tokyo-Wan Ferry Route is shown in Tab.3-1.

Tab. 3-1 Tokyo-Wan Ferry Information

Ship Name	Weight(t)	Speed(kt)	Length(m)	Starting Point
Shirahama Maru	3351	13	79.1	(35.221°N,139.715°E)
Kanaya Maru	3580	13	79.0	(35.170°N,139.817°E)

Because Shirahama Maru and Kanaya Maru are the ferries sailing opposite at the same time, and the navigation paths and the ship specifications are similar to each other, this thesis selects Shirahama Maru as the object of simulation training and research. In this thesis, the daily AIS data of the Uraga Traffic Route in 2014 are collected, and about 5100 different encounter situations of Shirahama Maru in the Tokyo Bay Ferry are selected. Then the trajectory of each situation is simulated, and the dangerous encounter situations with collision risk are selected. By inputting ship data in each case into the model for training, a new collision avoidance trajectory can be obtained to verify the strategy of the algorithm. In addition, all data will be normalized before model training to make model training faster and more accurate.

Combined with the ship information of the ferry, in order to establish the ferry ship domain model suitable for the Tokyo Bay sea area and improve the decision-making accuracy of the ferry collision avoidance algorithm, the following issues need to be considered:

(1) Prediction of the surrounding ships' location information

When receiving AIS data to calculate the location and other information of the ferry and other surrounding ships, since the AIS data are discrete in time, the time points of AIS data broadcast by the ferry and other surrounding ships may be different.

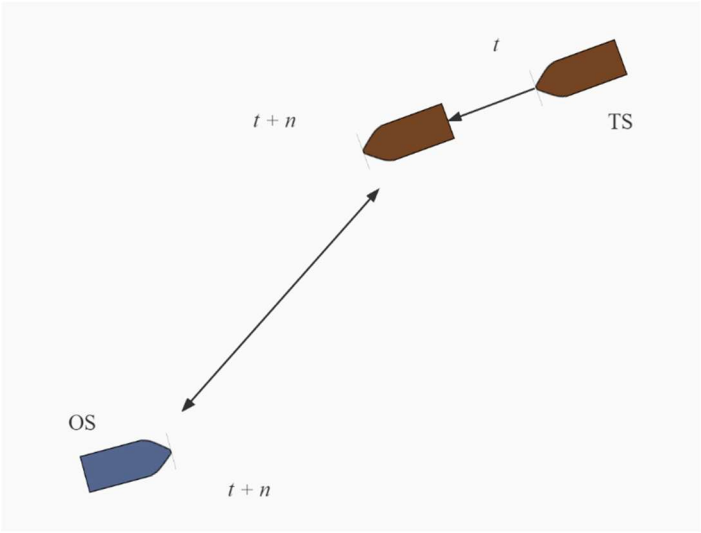

Fig.3-4 Error map of ship position

As shown in Fig. 3-4, the target ship broadcasts ship dynamic information at t time and the own ship broadcasts ship dynamic information at $t + n$ time. There is a time difference between the two ships for AIS data dissemination. In order to reduce the error of ship position information caused by broadcast time difference, it is necessary to estimate the ship position at $t + n$ by the dynamic information of other ships at t time.

Due to the high update rate of ship dynamic information, and the small change of ship speed and heading in a short time, the path of ships after dynamic information updating can be regarded as a uniform linear motion. In Fig. 3-4, the ship position information of other ships at time $t + n$ can be estimated by the uniform linear equation.

(2) Overlay of ship data around ferry

Due to the number of ships around the ferry of every navigation is limited, in order to reflect the truly location distribution of the ships surrounding the ferry, it is necessary to overlay the location distribution information of the surrounding ships of different flights at different times. Considering that the Tokyo Bay Ferry is a ferry of two fixed flights, combined with the ferry information, the flights schedule and the sea environment in Tokyo Bay, it is set that only the distribution data of the surrounding ships during each flight of the ferry are overlayed. Fig. 3-5 is the overlay graph of ship distribution data around the ferry. The black point in the graph is the ship distribution information obtained by AIS, and the red line is the transverse and longitudinal tangent of the ferry.

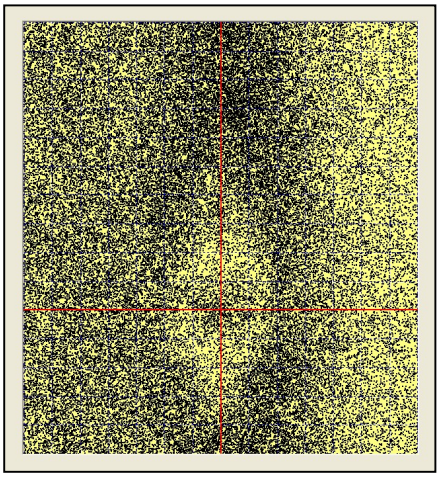

Fig.3-5 Overlay of ship distribution data around the ferry

(3) Determination of boundaries in the field of ships

Among the many methods to determine the boundary of the ship domain, the most used method is to determine the boundary through the change of the density of the surrounding ships. The density distributions of ships with different lengths are shown in Fig. 3-6.

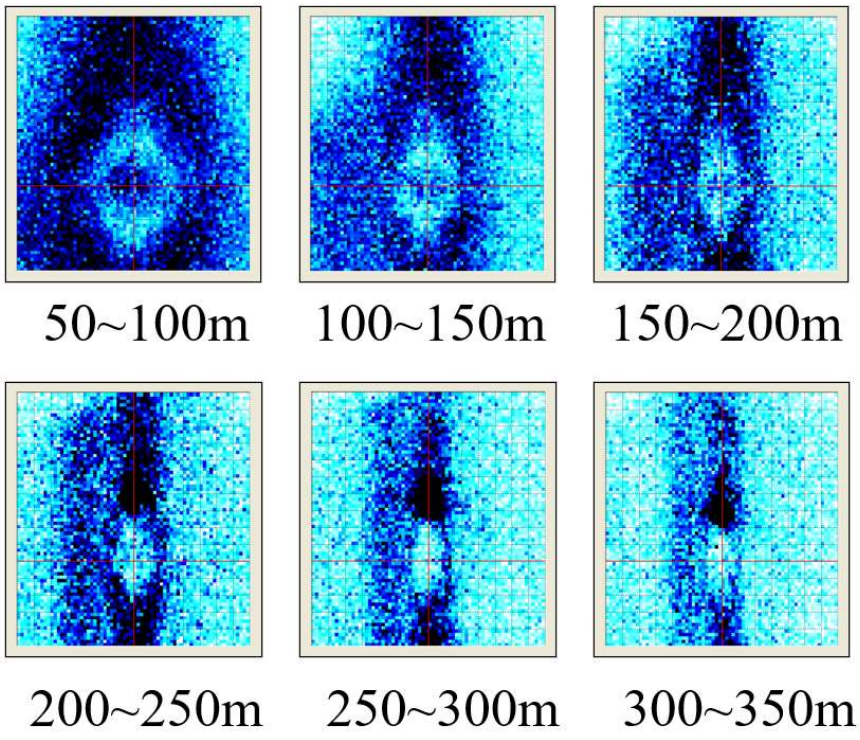

Fig.3-6 Distribution map of ship density for different lengths

The lengths of the Tokyo Bay ferries are about 80 meters. Fig. 3-6 shows that the high-

density areas of the port and starboard of the ships between 50~100 meters are about three times the ship length, and the high-density areas of the bow and stern of the ships are also about three times the captain.

By fitting the ship density distribution data in Fig. 3-6, the boundary relationship curves of ship bow, stern and lateral can be obtained, as shown in Figs. 3-7, 3-8, 3-9.

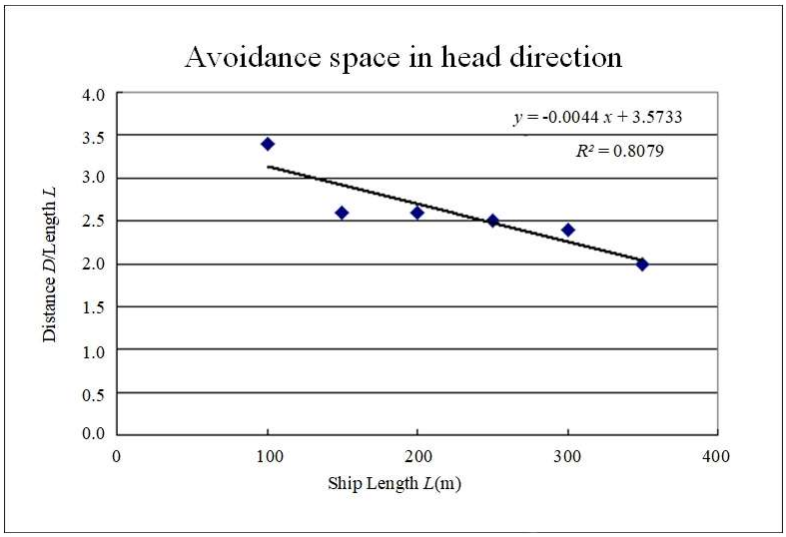

Fig.3-7 Fitting curve of ship head

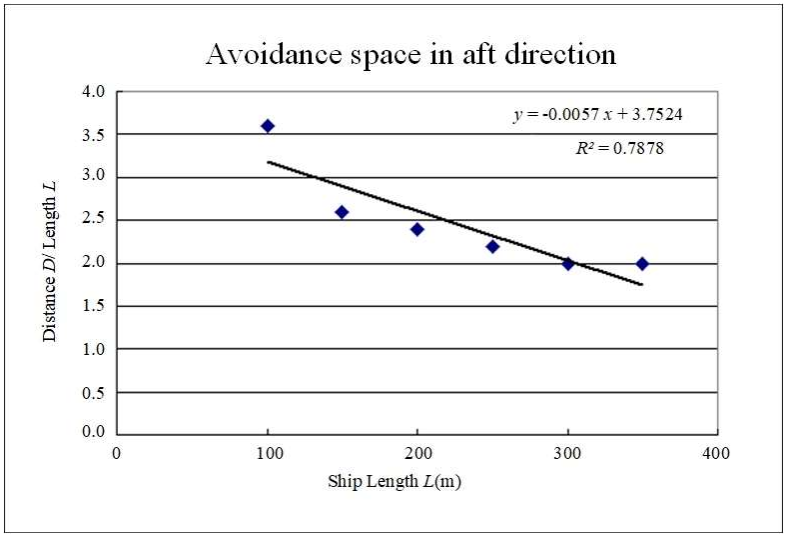

Fig.3-8 Fitting curve of ship aft

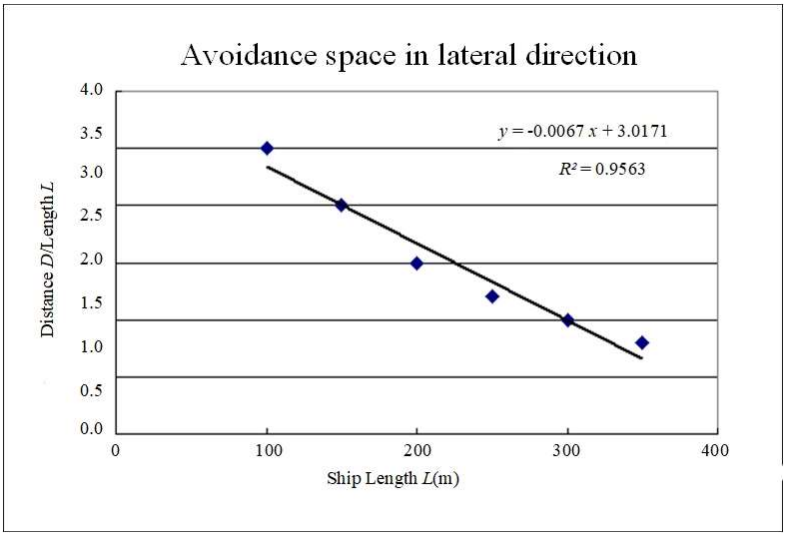

Fig.3-9 Fitting curve of ship lateral

Substituting the length L of the ferry into the formula in Figs. 3-7, 3-8, 3-9, it can be seen that the distance of ship head is about 3.2 L, the distance of ship aft is about 3.3 L, and the distances of the port and starboard are about 2.5 L.

So the rectangular Bumper model of the Tokyo Bay ferry is shown in Fig. 3-10:

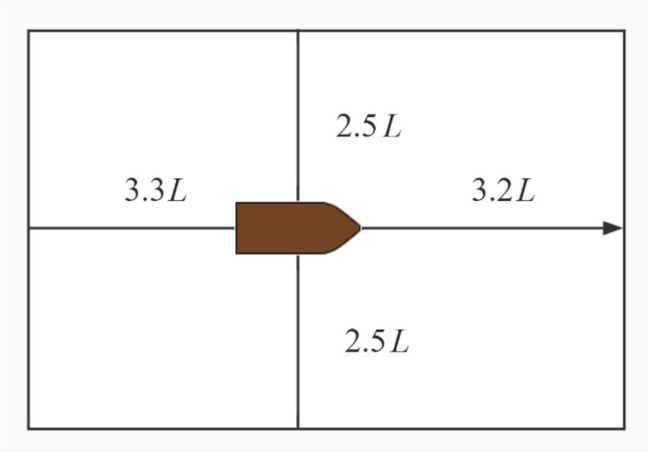

Fig.3-10 Rectangular Bumper model of the ferry

3.4 Summary

Firstly, this chapter introduces AIS as the source of ship information data, and then analyzes the encounter situations and collision avoidance requirements of Tokyo Bay ferry under the constraints of regulations combined with the Maritime Traffic Safety Law applicable to Tokyo Bay, which is helpful for simulation experiments. Finally, through the ship density distribution graph of Tokyo Bay, combined with the relevant knowledge of the ship domain, the ferry ship domain model is constructed, which provides more accurate model information for

the simulation and research of the ship collision avoidance decision-making algorithm.

4 Algorithm Design and Simulation Training

4.1 Algorithm Design

The design of ship collision avoidance decision-making algorithm in this thesis mainly includes the design of state space, action space, reward function and network structure based on ferry model, and the model process of the whole algorithm.

4.1.1 State Space Design

The state in reinforcement learning is a complete description of the environment. The simulation environment in this thesis is completely observable, which is in line with the Markov decision process model. Therefore, in order to realize the automatic collision avoidance path decision of ships, the DDQN algorithm needs to consider the observation of the surrounding environment state first. In Section 3.3.3, based on AIS system, this thesis uses python to construct a two-dimensional grid map of the environment around the Tokyo-Wan Ferry Route. In this map, the static and dynamic information of the ferry and other ships is observable.

This thesis divides the state of the surrounding environment of the ferry into two parts:

(1) When there are no other ships or obstacles around, that is, no collision risk, the target of the ferry is to reach the target point. At this time, the DDQN state space should take into account the current position, velocity and heading of the ferry.

(2) When there are ships or obstacles with collision risk around, the algorithm model needs to let the ferry to execute collision avoidance operation to prevent collision, and then enter the state (1). At this time, the DDQN state space also needs to consider the distance, angle, DCPA, TCPA and other state information between the ferry and the target ship or obstacle.

4.1.2 Action Space Design

The Tokyo-Wan Ferry Route, as a public transport route, has served on both sides of Kurihama and Kanaya since 1957, connecting Yokosuka, Kanagawa and Futtsu, Chiba, greatly saving traffic time.

So ferries are always sailing on planned routes and schedules. If the pilot needs to avoid collision with other ships or obstacles during navigation, corresponding measures to prevent the occurrence of collision should be taken. Because the frequent change of speed is harmful to the ship's main engine, and it takes a lot of time, in order to safety and good maneuverability, the pilot usually takes the method of changing the course to complete the collision avoidance operation.

The pilot changes the course by controlling the rudder. Since the Maritime Traffic Safety Law stipulates that the ship crosses the route entrance at an angle of nearly 90°, this thesis controls the change of the course per unit time of the ferry between -5° and 5°. The action space of DDQN is a discrete action space, and the pilot's turning the rudder to change the course is a

continuous control action. Therefore, this thesis divides the action of changing the course during the ferry navigation into five discrete actions: 5°, 2°, 0°, -2°, -5°. As shown in Formula 4-1:

$$A = [5, 2, 0, -2, -5] \tag{4-1}$$

4.1.3 Reward Function Design

The reward function is the enhanced signal R of environmental feedback after the ferry makes the action selection strategy interacting with the environment. It is used to evaluate action strategies. If the reward value is helpful to achieve the goal, it will be rewarded with a positive value, or punished with a negative value.

The ferry needs to use the learned action to obtain more reward from the environment, in order to fully explore the environment space to obtain the optimal action strategy. The reward function is divided into two parts to adapt the dynamic environment. One part of the reward function r_o is based on the distance between the ferry and the other ships. The other part of the reward function r_g based on the distance to the goal point. If the ship become closer to the goal point, the reward will be stronger, or it will be reduced. The two parts of the reward function are shown in Equations 4-2 and 4-3.

$$r_o = \begin{cases} 1, \ d_{ol} \le d_s < d_o \\ -1, \ d_o \le d_s < d_{ol} \\ -2, d_o and \ d_{ol} \le d_s \\ -10, d_o \le 0 \\ 0, \ else \end{cases} \tag{4-2}$$

$$r_g = \begin{cases} 5, \ d_g < 5 \\ 0.5, \ d_g < d_{gl} \\ -1, \ d_g > d_{gl} \\ \frac{10}{d_g}, else \end{cases} \tag{4-3}$$

In the equations, in r_o part, d_o is the distance of the ferry from the other ship, d_{ol} is the distance of the ferry from the other ship at the last time, d_s is the safe distance between ships. In r_g part, d_g is the distance of the ferry from the goal position, d_{gl} is the distance of the ferry from the goal position at the last time. The virtual simulation environment is a two-dimension map, 1000 meters is represented by 1 pixel. All the distances have already transferred into a two-dimension data.

During the collision avoidance process, when d_s is longer than d_{ol} but shorter than d_o, the reward r_o of the ferry will be added by 1. When d_o is less than d_s and d_{ol} is more than d_s, the reward r_o will be set to -1. When both d_o and d_{ol} are shorter than d_s, it puts ferry in a dangerous situation and the reward r_o will be punished to reduce by -2. If the ferry still keeps the negative decision making, the collision will occur and the reward r_o will impose a penalty of -10. During the process of approaching the goal point, the reward function r_g is used. When d_g is less than 5, it means the ferry has already reach the goal, the r_g will be added by 5. When d_g is less than d_{gl}, the ferry is closer to the goal point than the last time, the r_g will be set to 0.5. On the contrary, the reward r_g is reduced by -1. In other else cases,

the reward r_o will be calculated according to d_g.

Above all, the reward of the ship is designed as:

$$R = r_o + r_g \tag{4-4}$$

In addition, the condition of the R is based on the ferry sailing in the general area, if the ferry is out of boundary, the R will be punished a reward of -100.

4.1.4 Network Structure Design

The DDQN algorithm, like the DQN algorithm, has two networks, namely the current network and the target network, and the two networks have the same structure. Because the state input of the ship collision avoidance simulation environment constructed in this thesis is low-dimensional data, it is not complicated and good to handle, so the convolutional layer is not used in the designed network model, but the full connection layer is used.

The input of the network model is the state s observed in the surrounding water environment, and the output is the next action a of the ferry. Before inputting these state variables into the full connection layer, these different dimensions should be normalized.

Fig. 4-1 shows the deep neural network model built in this thesis:

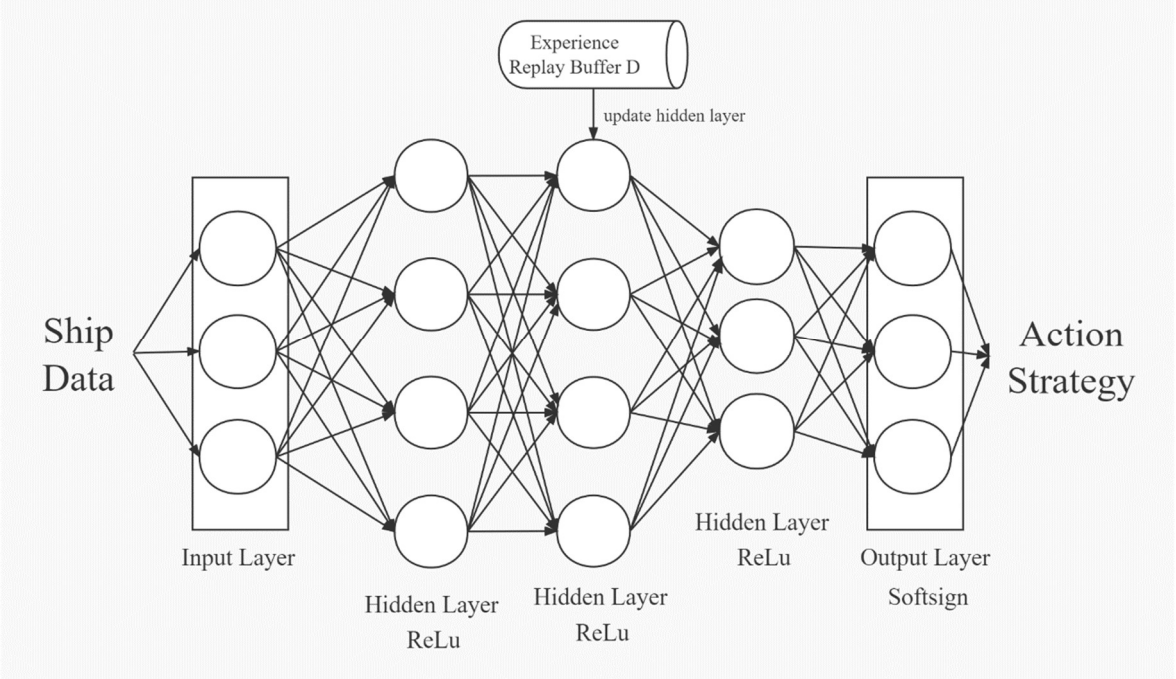

Fig.4-1 The structure of Deep Reinforcement Learning

As shown in Fig. 4-1, the constructed neural network contains an input layer, three hidden layers and an output layer. Through a lot of attempts and adjustments, the network structure and various hyperparameters are adapted to the algorithm in this thesis. Three fully connected hidden layers are set up in the neural network, and the number of neurons is 200,200 and 150, respectively. The activation function is ReLU function, and the activation function of the output

layer is Softsign. In the hyperparameters, the discount γ is set to 0.99, the minimum ε-*greedy* is set to 0.01, and the learning rate is set to 0.0002. The size of the Experience Replay Buffer is 10000, and the Mini-batch is 32. In terms of updating the weights of the neural network, the number of samples stored in the Experience Replay Buffer is checked. If the number of samples is greater than the batch of gradient descent, the random gradient descent method is used to update the weights of the current network. When the ferry takes an action, the network is updated once, but the target network is not updated with the current network. By calculating the total step length of the ferry, the target network is updated every 1000 steps, and the Adam optimizer is used. After completing these settings, train ferry collision avoidance strategies in each situation.

The training process in each situation is shown in Fig. 4-2:

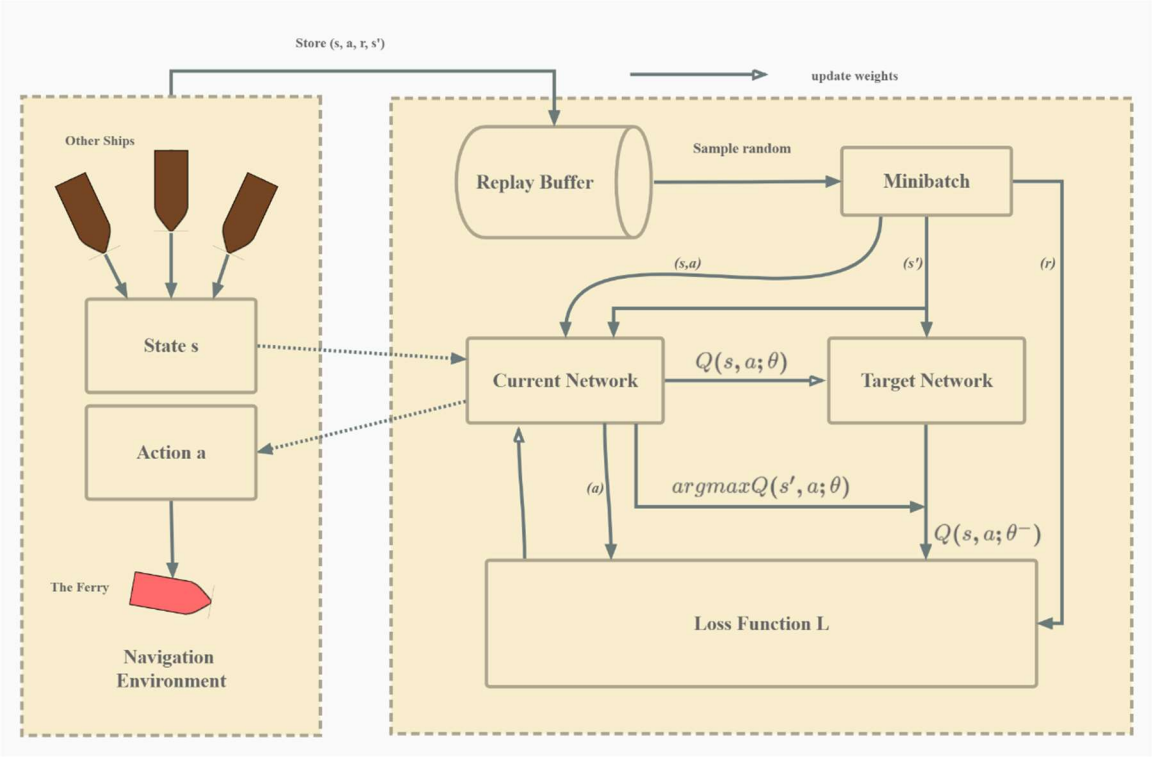

Fig.4-2 The structure of DDQN algorithm with navigation environment

The DDQN algorithm obtains the ship state from the navigation environment as input, puts the state into the current network and the target network, and trains and calculates through the loss function L. The loss function aims to evaluate action a, so that the current value network has the maximum Q value under the target value network state s', as shown below.

$$L(\theta) = \left(\left(r + \gamma Q\left(s', \underset{a}{\operatorname{argmax}}\, Q(s', a'; \theta); \theta^-\right)\right) - Q(s, a; \theta)\right)^2 \tag{4-5}$$

If the action makes the agent achieve the goal better, it will get a positive reward, otherwise,

it is a negative reward. Besides, the initial state s, the action a, the reward of this state r and the state s', all are stored in the Experience Replay Buffer. The two networks can extract data from the replay buffer randomly to train experience and adjust action strategy continuously. The algorithm will train over and over until it can get the best strategy. Now the estimated value of the discounted function is evaluated using a different policy, which solves the overestimation issue.

The process of deep reinforcement learning for ship collision avoidance is the process of neural network convergence in the DDQN algorithm. In this process, randomly extract ships transitions from Experience Replay Buffer as training samples and update network parameters over and over with the loss function. After training iteratively, the convergent optimal collision avoidance strategy is obtained.

4.1.5 Model Process

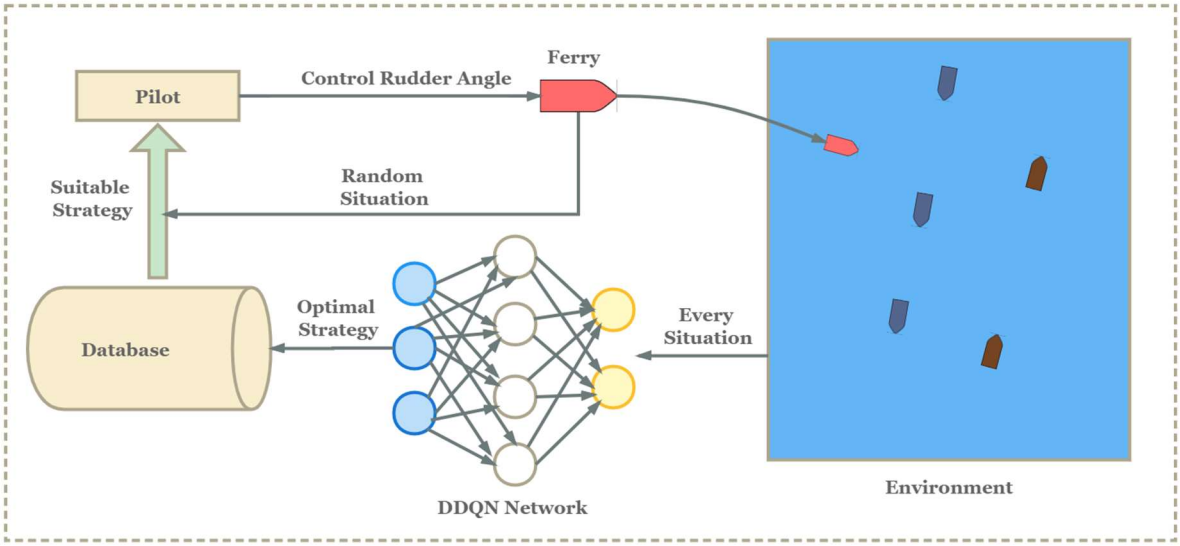

Fig.4-3 The model process of ship collision avoidance decision-making algorithm

Fig. 4-3 is the model process of the ship collision avoidance decision-making algorithm in this thesis. The advantage of deep reinforcement learning is to calculate and process massive data, use DDQN neural network to train each collision avoidance situation, and then store the optimal collision avoidance strategy in the database. When the ferry encounters any random situation, the pilot can obtain the most suitable collision avoidance strategy for the current situation, efficiently and safely through the dangerous area at the south entrance of the Uraga Traffic Route.

The pseudo code of the DDQN algorithm is detailed as follows:

Algorithm 1 The improved DDQN Algorithm

Input: Initial navigation environment states s

Output: Action strategy by weights parameter θ for DDQN

Random initialize current network Q with weight parameter θ.
Initialize target network Q' with weight parameter $\theta' \leftarrow \theta$.
Initialize experience replay buffer D
for episode = 1, M **do**
  Get the current initial state s;
  for t = 1, T **do**
    Get action a based on current strategy $\pi(s_t)$.
    Execute action a_t to get reward r_t and next state s_{t+1}
    Save transition (s_t, a_t, r_t, s_{t+1}) into replay buffer$D$
    Sample random minibatch of transitions (s_i, a_i, r_i, s_{i+1}) from D.
    Set

$$y_i = \begin{cases} r_i, & if \quad s_{i+1} \quad is \quad terminal \\ r_i + \gamma Q'(s_i, argmax_a Q(s_{i+1}, a_i, \theta), \theta'), & otherwise \end{cases}$$

    Do a gradient descent step with loss $\frac{1}{N}\sum_{i=1}^{N}(y_i - Q(s_i, a_i; \theta))^2$
    Update target parameters θ
    Copy weights into target network $\theta' \leftarrow \theta$ every 1000 steps
  end for
  Choose action $a_t \sim \pi_\theta(s_t)$
end for

4.2 Simulation Platform

The DDQN algorithm proposed in this thesis uses python to program the model and simulation platform. In order to train the model, a deep reinforcement learning network is established based on Tensorflow, and numpy, matplotlib, time, scipy and math libraries are used. Action space, state space, navigation angle, reward function and other parameters are also built by python.

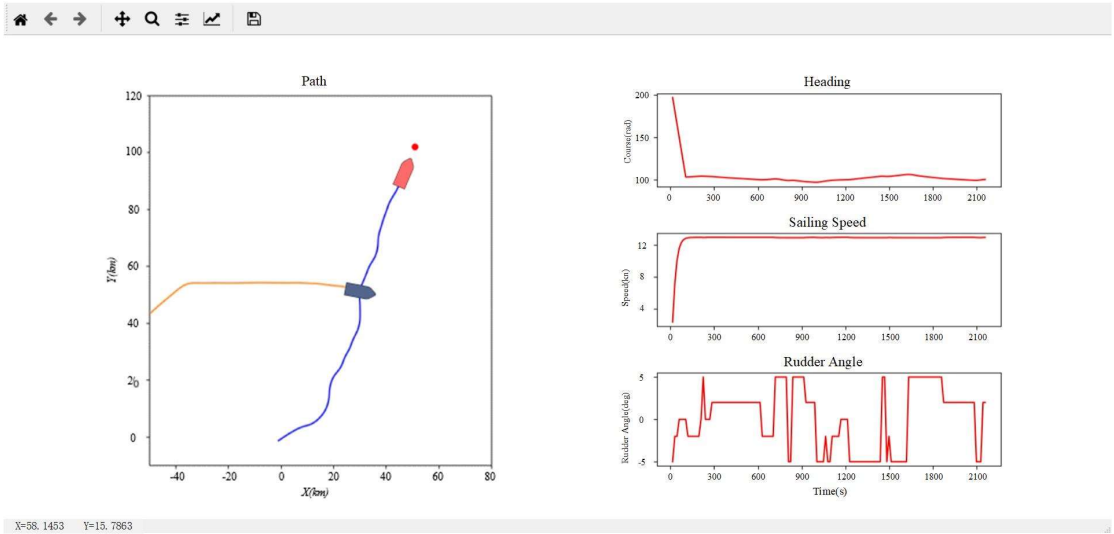

Fig.4-4 Simulation Platform Visualization Window

Fig. 4-4 show the visualization window constructed in this thesis. The left side is the simulation map to show the path, and the right side is the ship direction, ship speed and rudder angle from top to bottom. The simulation map is a rasterized two-dimensional map, and each pixel represents a certain distance. The longitude and latitude are transformed into the corresponding two-dimensional plane coordinates by calculation.

In the simulation map, one pixel represents 1 km in the actual environment, and the parameters of the ship and the environment are all converted into pixel units. The left side of Fig. 4-4 shows the training scene with Kurihama as the starting point and Kanaya as the target point. For the convenience of calculation, Shirahama Maru starts from the (35°13'22'N, 139°43'15'E), and the two-dimensional point is set to (0, 0). With Kanaya (35°10'12'N, 139°49'2'E) as the target point, the point is set to (50.95, 102.1) in the two-dimensional coordinate, so the positive direction of the horizontal axis on the map is south. By modifying the corresponding coordinates and origin, the ferry model from Kanaya to Kurihama can be trained. In the training process, the ferry will sail from the starting point to the target point, and avoid other ships through the feedback of the reward function.

Because the length of the unit pixel of the simulation map is 1km, while the Tokyo Bay Ferry is only about 80 meters long, and its ship domain is only about 500m × 400m rectangular, less than a quarter of the unit pixel size. Therefore, in the program code part, the collision risk is still judged by the ship domain, but in the map display, the larger ship figures are used instead to facilitate observation.

4.3 Simulation Training Experiment

The training data source of ship automatic collision avoidance algorithm in this thesis is based on the daily ship AIS data of Tokyo Bay in 2014. For example, Fig. 4-5 is the processed AIS data screenshot of a certain day in 2014, and combined with the different encounter situations of about 5100 navigations in Tokyo Bay ferry throughout the year, the trajectory of each case is analyzed and simulated, and 4277 dangerous encounter situations with collision risk are screened out. Fig. 4-6 is all the navigation paths of Shirahama Maru in the day of March 16, 2014, and the map direction is adjusted to be consistent with the direction of the simulation map in this thesis.

	A	B	C	D	E	F	G	H	I	J	K
79724	2014/3/16 22:09	4	4.6	35.19369	139.7795	8.7	SAGAMIMA	431002093	52	40	9
79725	2014/3/16 22:09	217	342.3	35.22059	139.7186	0	NO8 TOA M	431101065	52	38	9
79726	2014/3/16 22:09	16	13.1	35.17042	139.7693	12.3	TOKUEI MA	431200143	80	75	10
79727	2014/3/16 22:09	353	346.9	35.22206	139.7141	0	DOSHIRO M	431009622	52	62	9
79728	2014/3/16 22:09	0	147	35.2226	139.7138	0	ARIMA MAF	431006958	52	40	9
79729	2014/3/16 22:09	0	1.2	35.23971	139.7817	11.2	PLOVER PA(	563037200	80	183	32
79730	2014/3/16 22:09	312	290.6	35.27901	139.6789	0	TAISEIMARU	431100849	80	66	11
79731	2014/3/16 22:09	10	5.1	35.18533	139.7779	8.7	CAPE GREEI	351950000	70	300	50
79732	2014/3/16 22:09	223	334.4	35.22124	139.7193	0.1	NO2 TOA M	431101119	52	40	9
79733	2014/3/16 22:09	220	355.4	35.22155	139.7197	0.1	NO6 TOA M	431101161	52	40	8
79734	2014/3/16 22:09	230	192.8	35.2842	139.6754	0	ARASAKIMA	431000742	52	38	9
79735	2014/3/16 22:09	236	247	35.28407	139.6756	0	SHINANO M	431000795	52	38	9
79736	2014/3/16 22:09	234	165.5	35.2846	139.6764	0	SURUGA MA	431004067	52	38	9
79737	2014/3/16 22:09	236	193.2	35.28415	139.6755	0	KANTO MAF	431000841	52	40	9
79738	2014/3/16 22:09	17	288.6	35.28267	139.6784	0	NADESHIKO	431000393	80	69	12
79739	2014/3/16 22:09	237	226.4	35.28453	139.6764	0	NAGATOMA	431005054	52	38	9
79740	2014/3/16 22:09	47	55.6	35.02057	139.6244	10.9	KYOKUHO M	431301714	80	63	10
79741	2014/3/16 22:09	39	316.7	35.27393	139.685	0	YOYU MARU	431003205	80	65	10
79742	2014/3/16 22:09	19	204.8	35.28166	139.6781	0	POSEIDON-	431214000	90	78	20
79743	2014/3/16 22:09	17	84.5	35.28363	139.6787	0.1	TUNA PRIN(	352241000	70	120	16
79744	2014/3/16 22:09	40	313.3	35.22242	139.7177	0	SAKISHIMA	431200133	70	82	15
79745	2014/3/16 22:09	44	166.1	35.22202	139.717	0	SHIN CHOU	432743000	99	71	16
79746	2014/3/16 22:09	233	167.1	35.28435	139.6759	0	SHONAN M	431000124	52	38	9
79747	2014/3/16 22:09	234	210.5	35.28428	139.676	0	AOBAMARU	431003385	52	38	9
79748	2014/3/16 22:09	217	342.3	35.22059	139.7186	0	NO8 TOA M	431101065	52	38	9
79749	2014/3/16 22:09	354	182.1	35.22206	139.7139	0	URAGA MAI	431010513	52	38	9
79750	2014/3/16 22:09	219	305.9	35.22093	139.7189	0	NO7TOA M/	431100961	52	38	9
79751	2014/3/16 22:09	0	1.5	35.24129	139.7818	11.3	PLOVER PA(	563037200	80	183	32
79752	2014/3/16 22:09	6	4.3	35.1949	139.7796	8.7	SAGAMIMA	431002093	52	40	9
79753	2014/3/16 22:09	352	358.4	35.2524	139.7822	12.6	KATSUEI MA	431300795	70	86	15
79754	2014/3/16 22:09	353	346.9	35.22206	139.7141	0	DOSHIRO M	431009622	52	62	9
79755	2014/3/16 22:09	16	11.9	35.1721	139.7697	12.4	TOKUEI MAI	431200143	80	75	10
79756	2014/3/16 22:09	0	147	35.2226	139.7138	0	ARIMA MAF	431006958	52	40	9
79757	2014/3/16 22:09	312	290.6	35.27901	139.6789	0	TAISEIMARU	431100849	80	66	11
79758	2014/3/16 22:09	10	4.4	35.18653	139.778	8.7	CAPE GREEI	351950000	70	300	50
79759	2014/3/16 22:09	234	165.5	35.2846	139.6764	0	SURUGA MA	431004067	52	38	9
79760	2014/3/16 22:09	236	247	35.28407	139.6755	0	SHINANO M	431000795	52	38	9
79761	2014/3/16 22:09	221	355.4	35.22155	139.7197	0.1	NO6 TOA M	431101161	52	40	8
79762	2014/3/16 22:09	223	334.4	35.22124	139.7193	0.1	NO2 TOA M	431101119	52	40	9

Fig.4-5 Processed AIS data

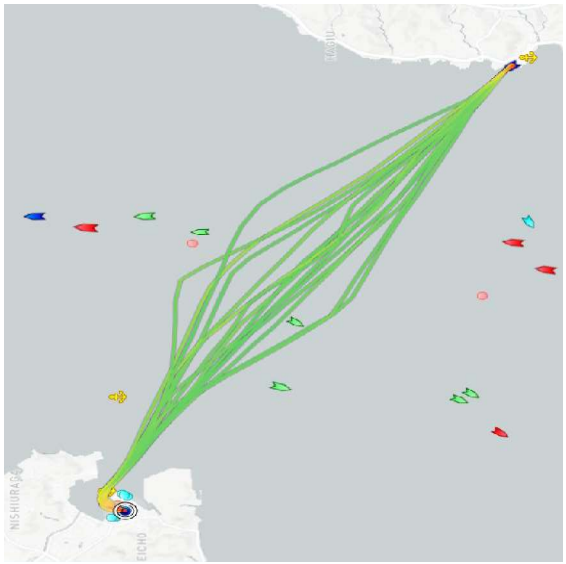

Fig.4-6 All-day sailing track of Shirahama Maru

The ferries will encounter some passing ships in each voyage, but the Tokyo Bay have requirements for the distance between ships entering and leaving the Uraga Traffic Route. Through the analysis of all the dangerous encounter situations, it is found that the ferry will only have collision risk between one to two passing ships in a single voyage. So the 4277

dangerous encounter situations only include 2-ship encounter situations and 3-ship encounter situations, of which 1582 are 2-ship encounter situations, accounting for about 37 % of the total. Therefore, the ship collision avoidance decision-making algorithm in this thesis can be simplified to the problem of automatic collision avoidance when only considering the situation of 2-ship crossing situation and 3-ship crossing situation.

The AIS data of ferry and ship with collision risk in each dangerous situation are used as training data to input the ship collision avoidance decision-making algorithm for training. After the training begins, the current episode will end due to ship collision, ferry arrival at the target point or training time overtime. After continuous training, in order to obtain higher reward value, the ferry navigation strategy will continue to optimize until the maximum reward value can avoid the ship and reach the target point. Finally, the optimal ship navigation strategy trained by the algorithm is stored in the database.

4.4 Experiment Result Analysis

Analyzing the simulation results to verify the effectiveness, accuracy, and stability of the algorithm. In the simulation platform, the pink boat represents the ferry, and the blue line represents the trajectory of the ferry. The brown and blue boat represents the ship sailing northwards or southwards of the Uraga Traffic Route. Their trajectories are indicated by green line and yellow line. The target point Kanaya is indicated by red dot. The influence of the ships lengths and widths is ignored. The heading angle is the initial angle of ship, and the ship speed is a constant in the whole process. This simulation platform does not take the wind velocity and direction and ocean currents and dynamics into consideration.

For each dangerous situation, simulation training is carried out, and the corresponding navigation strategy is obtained. All the dangerous encounter situations are divided into 2-ship case and 3-ship case. For these two cases, this thesis chooses a random example to analyze. In fact, all the 4277 situations have their own optimal route strategy, but it is difficult to show all of them due to paper space limit.

4.4.1 2-ship Case

In the case of two ships, the ferry only needs to avoid a ship in one direction. This thesis randomly selects a two-ship encounter situation from 07: 20 to 08: 10 on July 3, 2014 to analyze the optimization of navigation strategy by the algorithm. In this situation, the stand-on ship is INGENUITY, a cargo ship from the Republic of Marshall Islands, with a length of 172 meters. During the encounter, the ship speed is kept at 10.8 nautical miles. The give-way ship is Shirahama Maru.

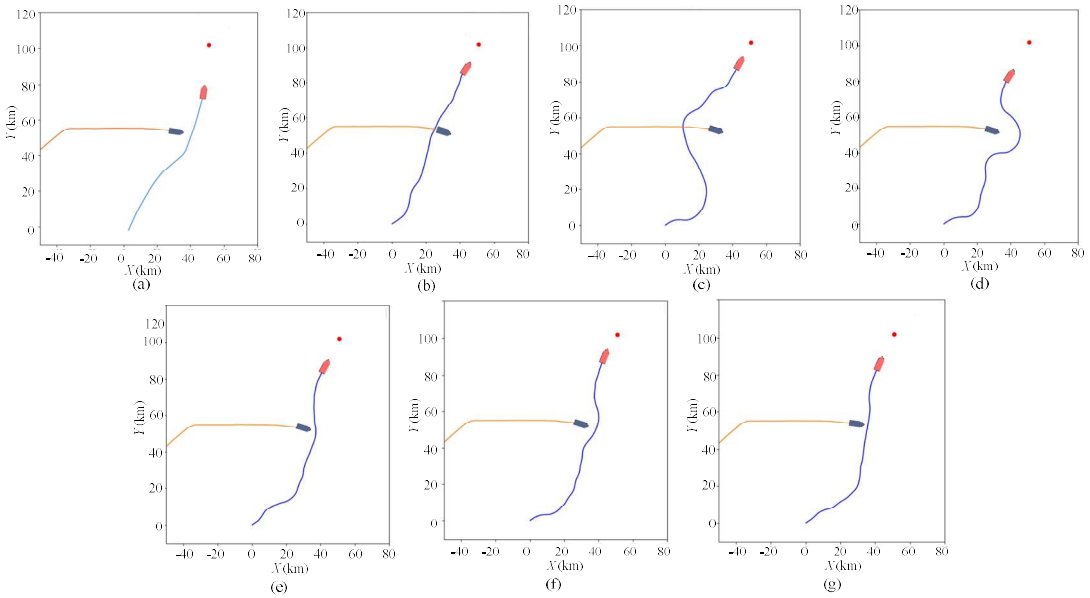

Fig 4-7 The training process of two-ship encounter situation

Fig. 4-7(a) is the original route operated by the ferry pilot, and the reward value obtained after the simulation training is 60.5. Fig. 4-7(b) ~ Fig. 4-7(g) are six representative episodes selected from 2732 episodes in the optimization process for training this situation. Fig. 4–7(b) shows that the ferry directly reaches the target point at the initial stage of training, but the risk of collision under this navigation strategy is very high. It basically does not avoid collision with other ships, and the reward value is lower than that of the manual operation, only 49.5. Figs. 4-7(c) and 4-7(d) show whether the ferry chooses to turn left or right for collision avoidance according to the reward value. Figs. 4-7(e) and 4-7(f) optimize the path without collision risk after selecting the right-turn collision avoidance, reduce the magnitude of the steering to save the navigation time, and finally obtain the optimal strategy, as shown in Fig. 4-7(g), with a reward value of 64.5. Compared with Figs. 4-7(a) and 4-7(g), it can be found that the optimal strategy obtained by the algorithm is similar to the route of the pilot's manual operation, but Fig. 4-7(g) has advanced steering to complete collision avoidance, which has higher reward value and higher efficiency.

4.4.2 3-ship Case

In the case of three ships encounter situation, the ferry should avoid collision twice. However, in the situation selected in this thesis, the Tokyo Bay Ferry Route is located at the south entrance of the Uraga Traffic Route, and the Uraga Traffic Route implements the traffic separation scheme. The ships leaving the Tokyo Bay should sail on the west side of the route, and the ships entering the Tokyo Bay should sail on the east side of the route. Therefore, the ferry needs to avoid ships in a certain direction first, and then avoid ships in another direction,

without encountering two ships at the same time. This thesis randomly selected a 3-ship encounter situation from 09: 20 to 10: 00 on January 25th, 2014 as an example, the stand-on ships are DHT HAWK and SHOKO MARU, DHT HAWK is a Hong Kong tanker, the ship length is 333 meters, the ship speed is 12.7 nautical miles. SHOKO MARU is a Japanese tanker, with a length of 77 meters and a speed of 11.6 nautical miles during the encounter. The give-way ship is still Shirahama Maru.

Input the initial data of the ship for simulation training, the 3-ship encounter situation training time is longer, four representative episodes are selected from 4115 episodes, as shown in Figs. 4-8(b) ~ 4-8 (e). Their reward values are 128.5, 147, 159.5 and 171. It can be seen that in the initial stage of training, the ferry takes a small course to avoid collision, reaching the target point directly, being close to other ships and the reward value is very low. Fig.4-8(c) shows that the reward value increases as the distance between the ferry and the stand-on ship increases, but the sailing route becomes longer. Therefore, it is necessary to reduce the steering angle to obtain a larger reward value until it is optimized to Fig. 4-8(e) so that the collision avoidance strategy is both safe and efficient. Fig. 7(a) is the original route operated by the pilot and its reward is 167.5. Compared with Fig. 4-8(a) and 4-8(e), although the steering directions of the two strategies are opposite, the collision avoidance processes are both well completed. However, the reward value of the strategy obtained by the algorithm training is greater than that obtained by the manual operation route, which shows that the algorithm can help the pilot to obtain a better collision avoidance strategy.

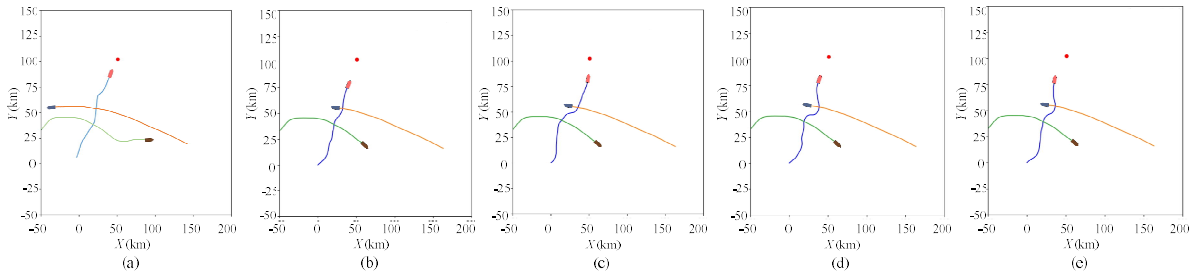

Fig 4-8 The training process of two-ship encounter situation

Figs. 4-7 and 4-8 are just examples of many simulation results. When training each 2-ship and 3-ship encounter situation, the algorithm can give the optimal collision avoidance strategy in each case, and continuously store the strategies in various situations into the database. Then in the actual navigation, by judging the information of other ships in the environment, the database can give the corresponding collision avoidance strategy to the pilot. According to this strategy, the ferry can pass through this dangerous sea area safely and efficiently.

In practical application, the database will store all collision avoidance strategies after training in AIS history records, and will be maintained and updated every certain period to cope with the complicated and changeable navigation environment.

4.5 Summary

Firstly, this chapter introduces the design and process of ship collision avoidance decision-making algorithm, and explains the state space, action space, reward function and network structure of the algorithm. Secondly, the simulation platform based on python and the deep reinforcement learning network based on Tensorflow are constructed. The AIS data of Tokyo Bay in 2014 are processed and input into the algorithm model for training, and the ship collision avoidance strategies in various situations are obtained. Finally, the encounter situations of two ships and three ships is randomly selected for analysis to verify the feasibility of the algorithm.

5 Application Verification Experiment

In Chapter 4, the ship collision avoidance decision-making algorithm model is trained using the AIS data of Tokyo Bay in 2014, and the corresponding navigation collision avoidance strategies in different situations are stored in the database. In practical applications, the database will store every encounter situation in every year to deal with complex navigation environment. This chapter verifies the effectiveness of the model trained in Chapter 4 through AIS data of ships from other dates in other years, and introduces the application background and value of this algorithm for Tokyo-Wan Ferry Route.

5.1 Application Background

The Uraga Traffic Route connects the Pacific Ocean and Tokyo Bay, and undertakes huge ship traffic. Due to the narrow route, the minimum width is only 6.5 km, and there are many ships, so the possibility of traffic accidents is high. The Tokyo Bay Ferry Route studied in this thesis is located at the entrance to the south of the Uraga Traffic Route, connecting the Kurihama Port in Yokosuga and the Kanaya Port in Futtsu, which is the only sea route that can cross the Tokyo Bay. As shown in Fig. 5-1, the 48-hour ship trajectory of Tokyo Bay from October 1st, 2014 to October 3rd, 2014 is shown. The blue trajectory in the black circle at the bottom of Fig. 5-1 is the Tokyo-Wan Ferry Route, greatly saves cross-strait traffic time. The author of this thesis has also taken the ferry on this route for field research. The rich experience of the pilot can skillfully avoid a large number of ship flows entering and leaving the Tokyo Bay, and also ensure the timely arrival of the target point.

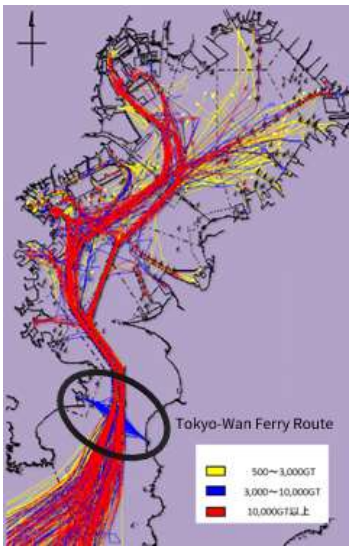

Fig 5-1 Vessel track in Tokyo Bay from 0:00 on October 1, 2014 to 0:00 on October 3, 2014

In this thesis, the ship automatic collision avoidance strategy model trained by the ship collision avoidance decision-making algorithm is applied to the Tokyo-Wan Ferry Route. By obtaining the AIS data of the surrounding ships, the similar situation is found in the database of the ship automatic collision avoidance strategy model, which provides reference information for the pilot's collision avoidance operation.

5.2 Verification for Tokyo-Wan Ferry Routes

In the process of ferry navigation, forecasting the trend of surrounding ships in advance is helpful to the planning of navigation strategy. The decision-making of ferry collision avoidance in this thesis can be divided into the following processes:

(1) Acquisition of the navigation information of the ferry

The first information needed for planning the collision avoidance strategy of the ferry is the navigation data of the ferry itself, which are provided by the AIS and the ferry cab.

(2) Judgment of collision risk vessels around ferry

The navigation data of ships in the surrounding 10 nautical miles are obtained by AIS. Combined with the speed, course and ship domain of each ship, the risk of collision with the ferry is determined.

(3) Generation of ferry automatic collision avoidance strategy

a. Ferry collision avoidance strategies without collision risk

When there is no collision risk between the ferry and the surrounding ships, the navigation strategy of the ferry is to apply the ship automatic collision avoidance strategy model to the target point.

b. Ferry collision avoidance strategy with collision risk

When one or more ships in the surrounding ships encounter with the ferry in a crossing situation, according to the navigation information of the stand-on ship, the future navigation trajectory is predicted. Then, the corresponding ferry collision avoidance strategy in similar cases is selected from the database of the ship automatic collision avoidance strategy model and fed back to the ferry pilot. The pilot can operate the course of the ferry or use the automatic driving system for automatic collision avoidance navigation through this strategy.

This thesis selects the AIS data on January 1st, 2018 to verify the proposed ship automatic collision avoidance strategy model. There were 14 flights of Shirahama Maru and Kanaya Maru on the Tokyo-Wan Ferry Route, and a total of 28 voyages. Among them, Shirahama Maru had 4 encounters with two ships, 8 encounters with three ships, and Kanaya Maru had 4 encounters with two ships, but there were 7 encounters with three ships. Since the two ferries sail from their respective ports at the same time, and almost pass through the ship traffic flow concentrated in the middle of the route at the same time, the number of times the two ships dangerous encounter situations is very close.

The AIS data of the 14 voyages of Shirahama Maru on January 1, 2018 are input into the ship collision avoidance decision-making algorithm in this thesis to observe whether the ferry can obtain the corresponding strategy to complete the voyage safely.

Since the simulation is whether the algorithm can timely feedback to the pilot the trained collision avoidance strategy in response to the actual navigation of the ferry, this thesis only takes the AIS data of the relevant ships in this period from the departure of the ferry to the start of maintaining constant speed as the input, and the subsequent navigation is completely completed according to the automatic collision avoidance strategy. Fig. 5-2 shows the simulation results of the verification experiment for each voyage, and the geographical environment is added to make the results more intuitive. Among them, Shirahama Maru is represented by a pink ship, and its trajectory is blue. the blue ship represents the ship entering the Uraga Traffic Route, its trajectory is orange. The ship leaving the Uraga Traffic Route is represented by a brown ship whose trajectory is green.

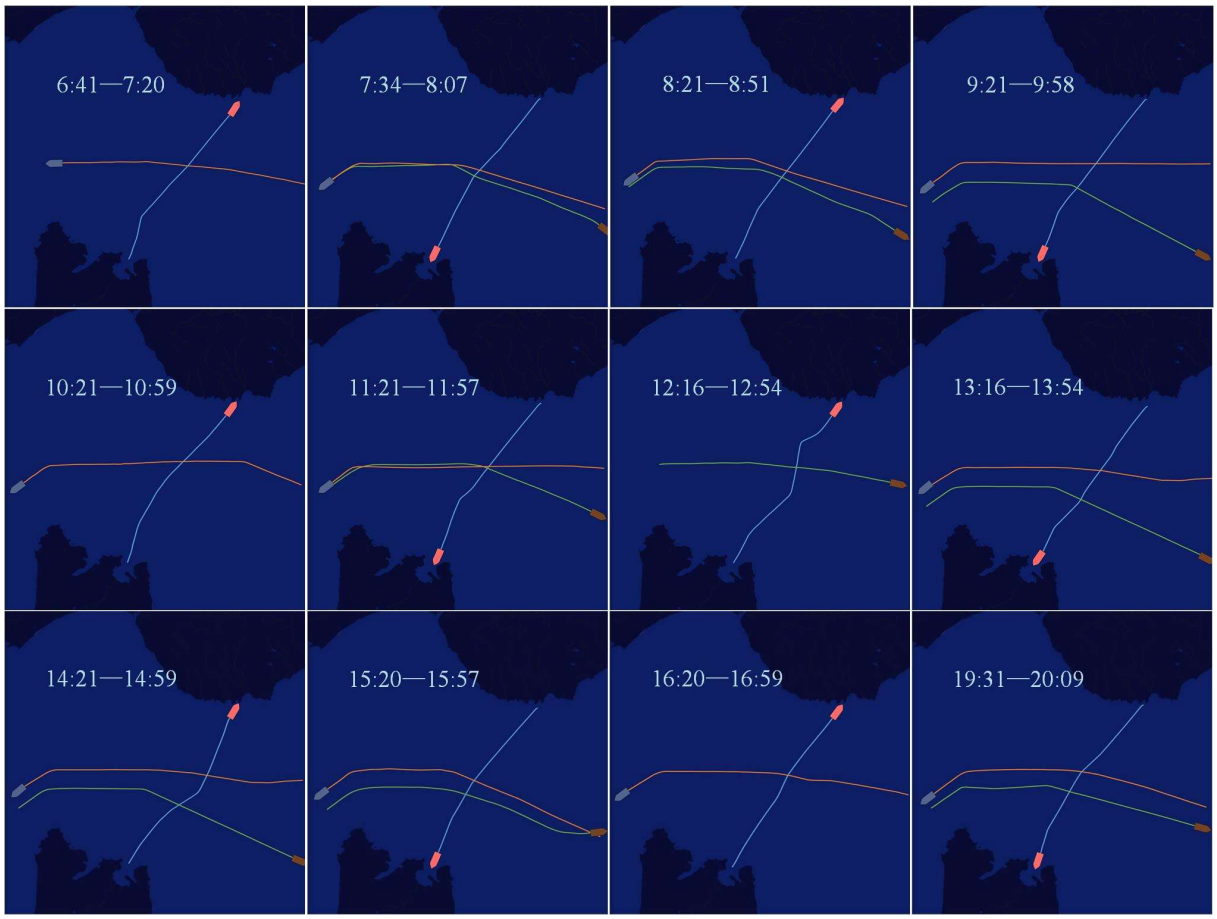

Fig-5-2 Simulation results of Shirahama Maru by the algorithm

The paths in the figure are the ship trajectories extracted from the first training in each encounter situation, which omit the safe paths of flights at 17: 00 and 18: 00. They all reach the target point safely and on time, and met the requirements of the Maritime Traffic Safety Law, which verifies the effectiveness of the algorithm.

Therefore, the ship collision avoidance decision-making algorithm based on DDQN algorithm can provide a reasonable ferry navigation strategy for the pilot according to the real-time ship data information obtained by AIS, so that the ferry can safely and effectively complete collision avoidance operation and smoothly reach the target point, avoid the safety problems caused by lack of experience or poor vision of the pilot, and provide a new assistant driving strategy for the operation of Tokyo-Wan Ferry Route.

5.3 Summary

Firstly, this chapter introduces the application value of automatic collision avoidance strategy model for Tokyo-Wan Ferry Route. Secondly, the AIS data of ships on January 1st, 2018 are collated and input into the ship automatic collision avoidance strategy model trained by the AIS data source in 2014, and the simulation results in a total of 12 situations are obtained, which verifies the effectiveness of the model and provides a new strategy for the operation of the Tokyo-Wan Ferry Line. It effectively improves the navigation safety of the waters and ensures the safety of life and property of ships and passengers.

6 Conclusion and Prospect

6.1 Conclusion

In recent years, with the breakthrough in the field of computer, deep reinforcement learning algorithm has developed rapidly and is widely used in various fields. In the field of ship collision avoidance, deep reinforcement learning has also made many research results. By learning and comparing various deep reinforcement learning algorithms, and using the knowledge of ship collision avoidance, this thesis proposes a decision-making algorithm for ship collision avoidance based on the DDQN algorithm in deep reinforcement learning and the ship domain model according to the actual situation of Tokyo-Wan Ferry Route. Through simulation training and verification experiments, the automatic collision avoidance strategy for ferry in various encounter situations is obtained, which effectively reduces the collision risk at the south entrance of the Uraga Traffic Route in Tokyo Bay.

This thesis mainly completed the following work:

(1) In this thesis, the AIS data of ships in 2014 are screened and sorted out, and it is concluded that the ferry will only encounter up to three ships during the navigation. At the same time, combined with the provisions of the Japanese Maritime Traffic Safety Law, the encounter situations between ferry and other ships are simplified as two-ship crossing situation and three-ship crossing situation, which greatly reduces the complexity of the model.

(2) Based on the ship information and ship domain knowledge, a rectangular Bumper model of the ferry is established by plotting the ship density distribution map of Tokyo Bay and fitting the ship avoidance curve.

(3) Reasonable state space, action space, reward function and network structure are designed for ship collision avoidance decision-making algorithm. Based on python, the simulation environment and visual simulation platform of ship collision avoidance are constructed. The DDQN-based ship collision avoidance decision-making algorithm is used to train 4277 encounter situations and store the optimal collision avoidance strategies in various situations.

(4) The AIS data of January 1st, 2018 is used to verify the effect of the ship automatic collision avoidance strategy model in real-time navigation, which effectively improves the navigation safety of the waters at the south entrance of the Puhe waterway in Tokyo Bay, and provides a new strategy for the operation of the Tokyo Bay ferry route.

It can be seen that this thesis is quite different from other ship collision avoidance research using deep reinforcement learning. Based on the geographical environment and AIS data of

Tokyo Bay, this thesis does not simply assume the collision avoidance in the process of tracing the starting point and the target point in the simulation platform, so it has good application value. In terms of the timeliness of the algorithm, the long training process of the deep reinforcement learning algorithm is advanced, and a large number of training results are stored in the database, so as to quickly judge and provide strategies in actual navigation, instead of real-time training under collision risk. Finally, the gradual analysis of the training process rather than the single presentation of the final training results reflects the process nature of deep reinforcement learning algorithm learning to the optimal strategy, which is more convincing.

6.2 Prospect

Although the proposed ship collision avoidance decision-making algorithm has realized the research on the automatic collision avoidance strategy of the Tokyo Bay Ferry Route, there are still many shortcomings. In future research, the following aspects need to be further studied:

(1) Consideration of the effects of wind, waves, etc.

Tokyo Bay is a sea area with large waves. This thesis ignores the influence of wind, wave and current on ship navigation and operation. Future research can increase the influence of wind speed, wind direction, wave and tidal current on the existing basis.

(2) Improving the universality of the algorithm

The ship collision avoidance decision-making algorithm proposed in this thesis takes Tokyo-Wan Ferry Route as the research object. Although it has certain practical application value, it is insufficient in other waters. In the following research, encounter situations such as head-on situation and overtaking situation should also be considered.

(3) Consideration of speed changes in collision avoidance

The collision avoidance action in this thesis is to change the course. Although the collision avoidance strategy in most cases is satisfied, some avoidance operation that need to change ship speed are not considered. In the future research, the speed variables should be added to the action space of the algorithm.

(4) Regular updating and maintaining of databases

In this thesis, the simulation training part only uses the data of 2014. In practical application, the database will store all navigation strategies in AIS historical data, and update and maintain them through new navigation data every day. With the increasing amount of data, more and more situations will be covered, so that all situations at the south entrance of Uraga Traffic Route can be finally coped with.

Reference

[1] PERERA L, CARVALHO J, GUEDES SOARES C. Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions [J]. Journal of marine science and technology, 2011, 16(1): 84-99.

[2] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. nature, 2015, 518(7540): 529-33.

[3] TAI L, PAOLO G, LIU M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation; proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), F, 2017 [C]. IEEE.

[4] ZHU M, WANG X, WANG Y. Human-like autonomous car-following model with deep reinforcement learning [J]. Transportation research part C: emerging technologies, 2018, 97: 348-68.

[5] AL-NIMA R R O, HAN T, CHEN T. Road tracking using deep reinforcement learning for self-driving car applications; proceedings of the International Conference on Computer Recognition Systems, F, 2019 [C]. Springer.

[6] YU L, SHAO X, WEI Y, et al. Intelligent land-vehicle model transfer trajectory planning method based on deep reinforcement learning [J]. Sensors, 2018, 18(9): 2905.

[7] DENG Y, BAO F, KONG Y, et al. Deep direct reinforcement learning for financial signal representation and trading [J]. IEEE transactions on neural networks and learning systems, 2016, 28(3): 653-64.

[8] ISELE D, RAHIMI R, COSGUN A, et al. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning; proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), F, 2018 [C]. IEEE.

[9] LECUN Y, BENGIO Y, HINTON G. Deep learning [J]. nature, 2015, 521(7553): 436-44.

[10] SUTTON R S, BARTO A G. Reinforcement learning: An introduction [M]. MIT press, 2018.

[11] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double q-learning; proceedings of the Proceedings of the AAAI conference on artificial intelligence, F, 2016 [C].

[12] JAPAN MINISTRY OF LAND I, TRANSPORT AND TOURISM. Maritime Traffic Safety Law [Z]. 1972

[13] FUJII Y, TANAKA K. Traffic capacity [J]. The Journal of navigation, 1971, 24(4): 543-52.

[14] ZHENG Z, WU Z. Fuzzy decision to avoid vessel collision [J]. Journal of Dalian Maritime University, 1996, 22(2): 5-8.

[15] WU Z, ZHENG Z. Model of ship's optimization opportunity for taking collision avoidance action in decision-naking for collision avoidance [J]. Journal of Dalian Maritime University, 2000, 26(4): 1-4.

[16] ZHENG Z, WU Z. Variable structure adaptive robust control algorithm applied to fin control for ship roll stabilization with uncertain system [J]. Journal of Dalian Maritime University, 2000, 26(4): 5-8.

[17] PêTRèS C, ROMERO-RAMIREZ M-A, PLUMET F. Reactive path planning for autonomous sailboat; proceedings of the 2011 15th International Conference on Advanced Robotics (ICAR), F, 2011 [C]. IEEE.
[18] SETHIAN J A. A fast marching level set method for monotonically advancing fronts [J]. Proceedings of the National Academy of Sciences, 1996, 93(4): 1591-5.
[19] GóMEZ J V, LUMBIER A, GARRIDO S, et al. Planning robot formations with fast marching square including uncertainty conditions [J]. Robotics and Autonomous Systems, 2013, 61(2): 137-52.
[20] LIU Y, SONG R, BUCKNALL R. A practical path planning and navigation algorithm for an unmanned surface vehicle using the fast marching algorithm; proceedings of the OCEANS 2015-Genova, F, 2015 [C]. IEEE.
[21] LIU Y, BUCKNALL R. The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method [J]. Applied Ocean Research, 2016, 59: 327-44.
[22] SONG R, LIU Y, BUCKNALL R. A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment [J]. Ocean Engineering, 2017, 129: 301-17.
[23] SUN X, WANG G, FAN Y, et al. An automatic navigation system for unmanned surface vehicles in realistic sea environments [J]. Applied Sciences, 2018, 8(2): 193.
[24] WANG M, ZHANG R. Research on fuzzy ND obstacle avoidance method of unmanned surface vessel [J]. Computer Engineering, 2012, 38(21): 164-7.
[25] TANG P, QIAO L, ZHANG R. Near-field reactive obstacle-avoidance for USV [J]. Journal of Huazhong University of Science and Technology(Natural Science Edition), 2011, 39(S2): 400-2.
[26] TANG P, ZHANG R, LIU D, et al. Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle [J]. Ocean engineering, 2015, 106: 128-40.
[27] ZHANG R, TANG P, SU Y, et al. An adaptive obstacle avoidance algorithm for unmanned surface vehicle in complicated marine environments [J]. IEEE/CAA Journal of Automatica Sinica, 2014, 1(4): 385-96.
[28] PHANTHONG T, MAKI T, URA T, et al. Application of A* algorithm for real-time path re-planning of an unmanned surface vehicle avoiding underwater obstacles [J]. Journal of Marine Science and Application, 2014, 13(1): 105-16.
[29] TIAN Y, HUANG L, XIONG Y, et al. On the velocity obstacle based automatic collision avoidance with multiple target ships at sea; proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS), F, 2015 [C]. IEEE.
[30] ORGANIZATION I M. Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs) [Z]. International Maritime Organization London, UK. 1972
[31] STATHEROS T, HOWELLS G, MAIER K M. Autonomous ship collision avoidance navigation concepts, technologies and techniques [J]. The journal of Navigation, 2008, 61(1): 129-42.
[32] PERERA L P, CARVALHO J P, SOARES C G. Autonomous guidance and navigation based on the COLREGs rules and regulations of collision avoidance; proceedings of the Proceedings of the international workshop advanced ship design for pollution prevention, F, 2009 [C].

[33] KANG Y, ZHU D, CHEN W. Review of research on collision avoidance path planning for ships [J]. Ship & Ocean Engineering, 2013, 42(5): 141-5.
[34] LIU D, WU Z, JIA C. A summary of recent researches on ship's intelligent collision avoidance decision-making and controlling system [J]. Journal of Dalian Maritime University, 2003, 29(3): 52-6.
[35] ZHUO Y, HEARN G E. A ship based intelligent anti-collision decision-making support system utilizing trial manoeuvres; proceedings of the 2008 Chinese Control and Decision Conference, F, 2008 [C]. IEEE.
[36] SHTAY A D, GHARIB W. An intelligent control system for ship collision avoidance [J]. King Saud University, Kingdom of Saudi Arabia, 2009.
[37] LEE S-M, KWON K-Y, JOONGSEON J. A fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines [J]. International Journal of Control, Automation, and Systems, 2004, 2(2): 171-81.
[38] PERERA L P, FERRARI V, SANTOS F P, et al. Experimental evaluations on ship autonomous navigation and collision avoidance by intelligent guidance [J]. IEEE Journal of Oceanic Engineering, 2014, 40(2): 374-87.
[39] MOE S, PETTERSEN K Y. Set-based Line-of-Sight (LOS) path following with collision avoidance for underactuated unmanned surface vessel; proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED), F, 2016 [C]. IEEE.
[40] ABDELAAL M, HAHN A. Nmpc-based trajectory tracking and collision avoidance of unmanned surface vessels with rule-based colregs confinement; proceedings of the 2016 IEEE Conference on Systems, Process and Control (ICSPC), F, 2016 [C]. IEEE.
[41] BENJAMIN M R, LEONARD J J, CURCIO J A, et al. A method for protocol‐based collision avoidance between autonomous marine surface craft [J]. Journal of Field Robotics, 2006, 23(5): 333-46.
[42] HARRIS C, HONG X, WILSON P. An intelligent guidance and control system for ship obstacle avoidance [J]. Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 1999, 213(4): 311-20.
[43] SMIERZCHALSKI R, MICHALEWICZ Z. Modeling of ship trajectory in collision situations by an evolutionary algorithm [J]. IEEE Transactions on Evolutionary Computation, 2000, 4(3): 227-41.
[44] YANG S, SHI Z, LI L, et al. Design and realization of vessel automatic collision avoidance simulation platform [J]. Navigation of China, 2009, (3): 50-4.
[45] LI L, YANG S, YIN Y. Study of simulation platform construction for automatic ship collision avoidance and its test method [J]. Navigation of China, 2006, (3): 47-50.
[46] LI L, WANG J, CHEN G. Integrated machine learning strategy of PIDVCA theory[J]. Information and Control, 2011, 40(3): 359-68.
[47] LI L, CHEN G, LI G, et al. Research on decision-making methods of ship humanoid intelligent collision avoidance [J]. Navigation, 2014, (2): 42-9.
[48] LI L, CHEN G, SHAO Z, et al. Construction of the PIDVCA system and its evaluation standard [J]. Journal of Dalian Maritime University, 2011, 37(4): 1-5.

[49] ZHAO Z, WANG J. Ship automatic anti-collision avoidance path simulation based on reinforcement learning in different encounter situations [J]. Science Technology and Engineering, 2018, 18(18): 218-23.
[50] SHEN H, HASHIMOTO H, MATSUDA A, et al. Automatic collision avoidance of multiple ships based on deep Q-learning [J]. Applied Ocean Research, 2019, 86: 268-88.
[51] WANG C, ZHANG X, ZHANG J, et al. Method for intelligent obstacle avoidance decision-making of unmanned vessel in unknown waters [J]. Chinese Journal of Ship Research, 2018, 13(6): 72-7.
[52] ZHANG X, WANG C, LIU Y, et al. Decision-making for the autonomous navigation of maritime autonomous surface ships based on scene division and deep reinforcement learning [J]. Sensors, 2019, 19(18): 4055.
[53] ZHOU S, YANG X, LIU K, et al. COLREGs-Compliant method for ship collision avoidance based on deep reinforcement learning [J]. Navigation of China, 2020.
[54] GUO S, ZHANG X, ZHENG Y, et al. An autonomous path planning model for unmanned ships based on deep reinforcement learning [J]. Sensors, 2020, 20(2): 426.
[55] WOO J, KIM N. Collision avoidance for an unmanned surface vehicle using deep reinforcement learning [J]. Ocean Engineering, 2020, 199: 107001.
[56] WU X, CHEN H, CHEN C, et al. The autonomous navigation and obstacle avoidance for USVs with ANOA deep reinforcement learning method [J]. Knowledge-Based Systems, 2020, 196: 105201.
[57] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning [J]. arXiv preprint arXiv:13125602, 2013.
[58] TESAURO G. Temporal difference learning and TD-Gammon [J]. Communications of the ACM, 1995, 38(3): 58-68.
[59] YAN J, HUANG Q, ZHOU X. Energy-saving optimization operation of central air-condition system based on Double-DQN algorithm [J]. Journal of South China University of Technology, 2019, 1.
[60] TANG Z, SHAO K, ZHAO D, et al. Recent progress of deep reinforcement learning: from AlphaGo to AlphaGo Zero [J]. Control Theory & Applications, 2017, 34(12): 1529-46.
[61] HAN S-H, CHOI H-J, BENZ P, et al. Sensor-based mobile robot navigation via deep reinforcement learning; proceedings of the 2018 IEEE International Conference on Big Data and Smart Computing (BigComp), F, 2018 [C]. IEEE.
[62] GOODWIN E M. A statistical study of ship domains [J]. The Journal of navigation, 1975, 28(3): 328-44.
[63] COLDWELL T. Marine traffic behaviour in restricted waters [J]. The Journal of Navigation, 1983, 36(3): 430-44.
[64] HARA K. PROPOSAL OF MANOEUVRING STANDARDS TO AVOID COLLISION IN CONGESTED SEA AREA [J]. 1991.

Acknowledgement

It’s a great experience to study in Tokyo University of Marine Science and Technology. I am really grateful to my supervisor Mr. Tamaru for giving me this opportunity. With his guidance, I have a new understanding of my research that enable deep reinforcement learning have application scenarios. And the data of training is all provided by Mr. Tamaru. Without his help, I won’t finish my thesis. In this year of living in Tokyo, I learned a lot of knowledge about ships and oceans at campus, and my free time also broadened my horizons, a very different society. If I have a chance in the future, I will come back to Tokyo again. In the meantime, I also make many good friends, such as Bai Wenbin, Shen Gangliang and Nemesio Menlendez. All of these make me a good memory. Thank you all very much.