A novel path following approach for autonomous ships based on fast marching method and deep reinforcement learning

Shuwu Wang[a,b,c,d], Xinping Yan[a,b,c], Feng Ma[b,c,*], Peng Wu[d] and Yuanchang Liu[d]

[a] School of Energy and Power Engineering, Wuhan University of Technology, Wuhan, People's Republic of China
[b] Intelligent Transportation Systems Research Center, Wuhan University of Technology, Wuhan, People's Republic of China
[c] National Engineering Research Center for Water Transportation Safety, Wuhan University of Technology, Wuhan, People's Republic of China
[d] Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK

ARTICLE INFO

Keywords:
Path following
Autonomous ships
Fast marching method
Deep reinforcement learning
Navigation brain system.

ABSTRACT

Path following is one of the indispensable tools for autonomous ships, which ensures that autonomous ships are sufficiently capable of navigating in specified collision-free waters. This study proposes a novel path following approach for autonomous ships based on the fast marching (FM) method and deep reinforcement learning (DRL). The proposed approach is capable of controlling a ship to follow different paths and ensuring that the path tracking errors are always within a set range. With the help of the FM method, a grid-based path deviation map is specially produced to indicate the minimum distance between grid points and the path. Besides, a path deviation perceptron is specifically designed to simulate a range sensor for sensing the set path deviation boundaries based on the path deviation map. Afterwards, an agent is trained to control a ship following a circular path based on the DRL. Particularly, the approach is validated and evaluated through simulations. The obtained results show that the proposed method is always capable of maintaining high overall efficiency with the same strategy to follow different paths. Moreover, the ability of this approach exhibits a significant contribution to the development of autonomous ships.

1. Introduction

The maritime transport industry plays a pivotal role in the development of the world economy. However, ship accidents occur from time to time, bringing serious risks to society and the environment [1]. Improving the safety of maritime traffic and improving the efficiency of shipping have always been important research topics in academia [2][3]. At the same time, autonomous ships are considered to be a promising future for the development of maritime safety [4][5][6]. To address this problem, Yan et al. [7][8] put forward the Navigation Brain System (NBS), which is an intelligent system composed of three subsystems a perception module, a cognition module and a decision and manipulation module. As an integral part of this system, path following has attracted widespread attention from academia and industry [9].

In general, the path following of an autonomous ship consists of guidance and control modules [10][11]. The guidance module acts as a decision-maker and aims to generate behavior instructions according to the perception information. The line-of-sight (LOS) guidance law is a typical guidance algorithm, which was first introduced into the ship motion control by Fossen [12]. Afterwards, numerous studies were conducted to develop the LOS guidance law. The proportional [13], integral [14] and adaptive [15] LOS guidance laws were proposed to address different path following requirements. The traditional LOS [12] converts the path following task into the heading tracking task, and can meet requirements of speed control at the same time. The proportional LOS (PLOS) [13] is provided to follow the path of a moving target at a rate proportional to the rotation rate of the LOS in the same direction. And the integral LOS (ILOS) [14] is specially put forward for working in the presence of actuator gain uncertainty and unknown environment disturbances. The adaptive LOS (ALOS) [15] is usually designed based on the maneuverability of the ship.

Moreover, the control module makes actual commands in accordance with the behavior instructions provided by the guidance system. Various algorithms have been developed as control modules to implement path following. These include the proportional derivative (PD) and proportional integral derivative (PID) control [16], sliding mode control (SMC) [17], backstepping control [18] and model predictive control (MPC) [19], etc. The SMC [17] is a discontinuous nonlinear control method. Its feedback control law is not a continuous function of time and the structure of the control law varies according to the change of the state track position. The backstepping [18] introduces a known Lyapunov function into the control module and gradually corrects the deviation between the set trajectory and the actual trajectory to realize the global adjustment or tracking of the system. The MPC [19] solves a finite-time open-loop optimization problem online based on the current measurement information and applies the first element of the obtained control sequence to the controlled object.

However, most approaches have limitations of dependency on prior knowledge of dynamic modeling and uncertainty modeling. In addition, it is usually difficult to tune the parameters of the path-following algorithm with two layers

[*] Corresponding author
wangshuwu@whut.edu.cn (S. Wang); xpyan@whut.edu.cn (X. Yan); feng.ma.whut@gmail.com (F. Ma); peng.wu.14@ucl.ac.uk (P. Wu); yuanchang.liu@ucl.ac.uk (Y. Liu)
ORCID(s):

(guidance and control).

In recent years, machine learning, especially reinforcement learning (RL), has been successful in complex control problems, such as the inverted pendulum [20], which provides new insight into the path following control of autonomous ships. In [21], an actor-critic method was applied to find a policy for course tracking of a ship, and the LOS guidance law was employed as a guidance module. Woo et al. [22] proposed a deep RL (DRL) based controller for path following of an unmanned surface vehicle with the help of the vector field guidance method, and the performance of different stages of the trained policy showed great self-learning ability of the proposed method. The applied RL approach makes it easy to get a good performance controller. However, to match the controller, it still requires significant efforts to tune the parameters of guidance modules. To address this challenge, Martinsen et al. [23] developed the DRL to solve the straight path following problem directly for underactuated marine vessels without using the two layers (guidance and control) structure, which avoids complex parameter tuning problems. Nonetheless, the adaptability for different path following tasks is not developed in the studies above mentioned. In addition, few studies are well discussed the path deviation boundary problem of path following.

By summarising current studies about path following of ships, research gaps remain. First, it typically requires considerable effort to adapt the guidance and control modules to each other. Moreover, most of the studies only focus on the robustness of path-following methods without well developing the maximum path deviation limitation. Finally, the reuse of the RL-based policy in different path following tasks needs to be further studied.

To address the above problems, an approach with the characteristic of parameters self-tuning, maximum path deviation limitation configurable and learned strategy reusable needs to be proposed. With the intention of achieving the above goals, a novel path following controller was designed based on the DRL. Meanwhile, a path deviation map and a path deviation perceptron were specially designed for the controller. The main contributions of the proposed method to path following of autonomous ships are as follows: 1) a parameters self-tuning path following controller based on the DRL is employed to guide the ship to follow the path directly instead of converting the path following problem into a course tracking problem; 2) through a novel design of the path deviation perceptron, great flexibility can be achieved by configurable maximum path deviation limitation before training. 3) a new path following Markov decision process (MDP) has been proposed to ensure the reusing of the learned control strategy.

The rest of this paper is organized as follows. In Section 2, the ship motion model and methodology used in this paper are introduced. Section 3 details the proposed path following approach. The proposed fast marching (FM) and DRL based path following method is verified and compared with traditional methods by simulations in Section 4. Section 5 concludes this paper and discusses future research.

2. Ship motion model and methodology

In this section, a manoeuvring mathematical model group (MMG) [24][25] based ship motion model, the ALOS guidance law, the FM method, and the DRL are described in detail.

2.1. Ship motion model

Usually, ship motion models are used to validate proposed control methods in simulations. In this way, the accuracy, precision and robustness of new methods can be tested at a low cost. Ship motion models are divided into two categories by different ways of modeling, which are hydrodynamic models [26][27] and responsive models [28]. The Abkowitz hydrodynamic model [29] and the MMG [24] are two hydrodynamic models widely used. The MMG-based ship motion model was adopted for validating in this study since it is more simple but accurate enough.

At the very beginning, a set of coordinate systems should be defined. Fig. 1 shows the two coordinate systems used in this study. $o_0-x_0y_0$ is the space-fixed coordinate system and $o-xy$ is the moving ship-fixed coordinate system. Plane x_0-y_0 coincides with the still water surface. In the coordinate system $o-xy$, o is taken at midship. x and y axes point towards the bow and starboard of the ship, respectively.

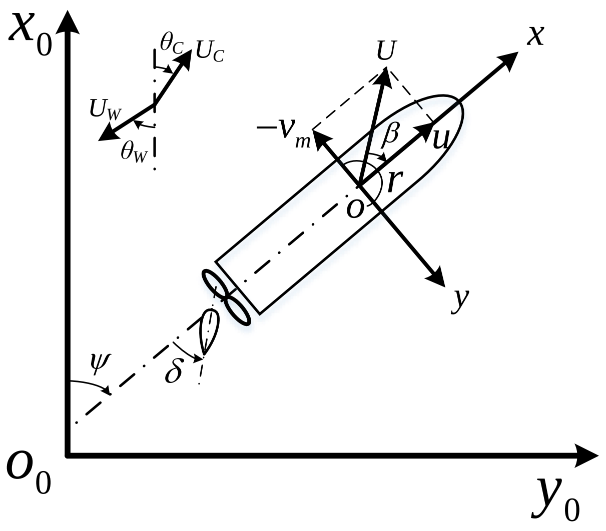

Figure 1: Coordinate Systems.

Heading angle ψ is defined as the angle between x_0 and x axes. δ is the rudder angle and r is the yaw rate. u and v_m indicate the ship velocity components in x and y axis directions, respectively. U is the total velocity and β is the draft angle of the ship. Gravity center G of the ship is located at $(x_G, 0)$ in $o-xy$ system. Then, the lateral velocity component at the center of gravity v is expressed as $v = v_m + x_G r$.

The MMG based ship motion model of a ship can be defined as follows:

$$\left.\begin{array}{l}\left(m+m_{x}\right)\dot{u}-\left(m+m_{y}\right)v_{m}r-x_{G}mr^{2}=X\\ \left(m+m_{y}\right)\dot{v}_{m}+\left(m+m_{x}\right)ur+x_{G}m\dot{r}=Y\\ \left(I_{zG}+x_{G}^{2}m+J_{z}\right)\dot{r}+x_{G}m\left(\dot{v}_{m}+ur\right)=N\end{array}\right\},\quad(1)$$

where m is the ship mass. m_x and m_y are the added mass in x and y axis directions, respectively. x_G is the longitudinal coordinate of the ship gravity center. I_{zG} is the moment of ship inertia around the gravity center and J_z is the added moment of inertia. $\dot{u}$, $\dot{v}_m$ and $\dot{r}$ are the change rate of u, v_m and r, respectively. X, Y and N represent the longitudinal force, the transverse force and the transverse moment, respectively, and are expressed as follows:

$$\left.\begin{array}{l}X=X_{H}+X_{R}+X_{P}+X_{A}\\ Y=Y_{H}+Y_{R}+Y_{A}\\ N=N_{H}+N_{R}+N_{A}\end{array}\right\},\quad(2)$$

where subscripts H, R and P denote the hull, the rudder and the propeller of the ship, respectively. The subscript A denotes the wind forces.

The expressions X_H, Y_H and N_H related to the hull are defined as follows:

$$\left.\begin{array}{l}X_{H}=(1/2)\rho LdU^{2}X_{H}'\left(v_{m}',r'\right)\\ Y_{H}=(1/2)\rho LdU^{2}Y_{H}'\left(v_{m}',r'\right)\\ N_{H}=(1/2)\rho L^{2}dU^{2}N_{H}'\left(v_{m}',r'\right)\end{array}\right\},\quad(3)$$

where ρ is the density of water. L is the ship length between perpendiculars and d is the ship draft. X_H' and Y_H' are the longitudinal and lateral force coefficients of the ship. N_H' is the force moment coefficient.

The expressions of effective rudder forces and moments acting on the rudder are defined as:

$$\left.\begin{array}{l}X_{R}=-\left(1-t_{R}\right)F_{N}\sin\delta\\ Y_{R}=-\left(1+a_{H}\right)F_{N}\cos\delta\\ N_{R}=-\left(x_{R}+a_{H}\cdot x_{H}\right)F_{N}\cos\delta\end{array}\right\},\quad(4)$$

where t_R is the steering resistance deduction factor. F_N is the rudder normal force. a_H is the rudder force multiplier. x_R and x_H are the longitudinal positions of the rudder and the acting point of the additional lateral force, respectively.

The hydrodynamic force acting on the propeller is defined as:

$$X_{P}=\left(1-t_{P}\right)T,\quad(5)$$

where T is the propeller thrust and t_P is the propeller thrust deduction factor.

The surge force, lateral force, and yaw moment due to steady and constant wind are expressed as follows:

$$\left.\begin{array}{l}X_{A}=(1/2)\rho_{a}A_{X}V_{A}^{2}C_{XA}\left(\theta_{A}\right)\\ Y_{A}=(1/2)\rho_{a}A_{Y}V_{A}^{2}C_{YA}\left(\theta_{A}\right)\\ N_{A}=(1/2)\rho_{a}A_{Y}LV_{A}^{2}C_{NA}\left(\theta_{A}\right)\end{array}\right\},\quad(6)$$

where ρ_a denotes the air density. V_A denotes the relative wind speed and θ_A denotes the relative wind direction. A_X and A_Y are the front and the profile wind pressure area, respectively. C_{XA}, C_{YA} and C_{NA} denote the aerodynamic force coefficients expressed as a function of the relative wind direction θ_A.

Table 1
List of parameters used for aerodynamic force coefficients C_{XA}, C_{YA} and C_{NA} in this study.

	C_{XA}	C_{YA}	C_{NA}
n	3	3	2
a_0	0.159	0	0
a_1	-1.459	-0.004	-0.002
b_1	0.139	-0.756	0.001
a_2	0.063	0	0.133
b_2	-0.012	0.002	-0.042
a_3	0.262	0.002	-
b_3	-0.077	0.118	-
w	0.970	0.998	1.006

The head wind of the ship is defined as $\theta_A = 0°$, the starboard wind as $\theta_A = 90°$, and the following wind as $\theta_A = 180°$. θ_A and V_A are defined as follows:

$$\left.\begin{array}{l}\theta_{A}=\arctan 2\left(v_{A},u_{A}\right)\\ V_{A}=\sqrt{u_{A}^{2}+v_{A}^{2}}\\ u_{A}=u+U_{W}\cos\left(\theta_{W}-\psi\right)\\ v_{A}=v_{m}+U_{W}\sin\left(\theta_{W}-\psi\right)\end{array}\right\},\quad(7)$$

where u_A and v_A are the relative wind velocity components in x and y axis directions, respectively. U_W denotes the absolute wind speed and θ_W denotes the absolute wind direction.

For the full-scale (320 m) KVLCC2 ship, the front wind pressure area A_X corresponds to 1161 m^2, and the profile wind pressure area A_Y corresponds to 4258 m^2 [30]. Therefore, wind pressure areas A_X and A_Y for a 7 meters KVLCC2 ship are equal to 0.56 m^2 and 2.04 m^2, respectively. In addition, the aerodynamic force coefficients C_{XA}, C_{YA} and C_{NA} predicted by Yasukawa [30] are shown with red solid lines in Fig. 2. The following function was used to fit the coefficients in this research.

$$f\left(\theta_{A}\right)=a_{0}+\sum_{k=1}^{n}\left(a_{k}\cos\left(kw\theta_{A}\right)+b_{k}\sin\left(kw\theta_{A}\right)\right),\quad(8)$$

Parameters of equation (8) for coefficients C_{XA}, C_{YA} and C_{NA} are listed in Table 1. And the blue dashed lines shown in Fig. 2 indicate the coefficients used in this research.

The effect of sea current [31] was also considered in this research. The horizontal current components in surge and sway are defined as:

$$\left.\begin{array}{l}u_{C}=U_{C}\cos\left(\theta_{C}-\psi\right)\\ v_{C}=U_{C}\sin\left(\theta_{C}-\psi\right)\end{array}\right\},\quad(9)$$

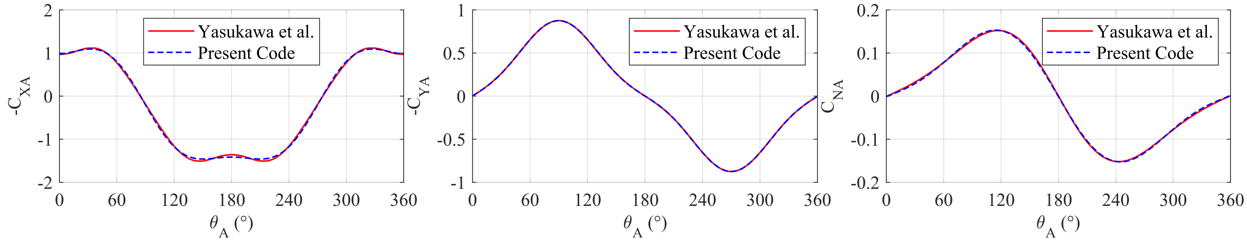

Figure 2: Aerodynamic force coefficients.

where U_C and θ_C are the current velocity and direction, respectively.

When calculating the trajectory of the ship in the space-fixed coordinate system $o_0 - x_0 y_0$, the relative velocity components u and v should be replaced with the absolute velocity components u_r and v_r, which are defined as follows:

$$\left.\begin{array}{l} u_r = u - u_C \\ v_r = v - v_C \end{array}\right\}, \tag{10}$$

Other parameters not developed in detail refer to the work of Yasukawa [24]. And the 7 meters KVLCC2 ship motion model was used in this study.

2.2. Adaptive LOS (ALOS) guidance law

The LOS guidance law has the advantages of accurate target tracking and simple calculation. It was widely used in the fields of underwater and surface vehicle path following control. And it showed a good performance. The path for LOS guidance is usually formed by segmented lines or curves located between waypoints. The straight-line path was selected for the path following of ships in this research. During the path following, the first step is to find the LOS points. There are different ways of generating the LOS point [32][33]. The way named as LOS-2 described in [32] is adopted in this research. Furthermore, the radius of the LOS circle is defined as follows:

$$R_{\mathrm{LOS}} = \begin{cases} nL, |e| \leq nL \\ e + mL, |e| > nL \end{cases} \tag{11}$$

where R_{LOS} denotes the radius of the LOS circle. L is the ship length and e is the path deviation. The coefficients m and n in this study are set to 1 and 3, respectively.

When the ship starts to move away from the current target waypoint or enters the current acceptance circle, the target waypoint point of LOS guidance will change to the next waypoint [33]. Usually, the radius of the acceptance circle is set with a constant value in traditional LOS guidance law. However, different acceptance circles are required for the path following of ships. When the included angle between adjacent path lines is small, the ship needs to make a turn early, which means that a big acceptance circle is required. On the contrary, a small acceptance circle is required for high path tracking accuracy when the included angle is big. The ALOS [33] was specially designed for such requirements. The radius of the acceptance circle is defined as follows:

$$R(\Delta\theta) = \begin{cases} l\left(\frac{\pi}{\Delta\theta} - 1\right)^2 L + R_{\min}, \Delta\theta \geq \theta_0 \\ R_{\max}, 0 \leq \Delta\theta < \theta_0 \end{cases}, \tag{12}$$

$$\theta_0 = \frac{\pi}{\sqrt{\frac{R_{\max} - R_{\min}}{lL}} + 1}, \tag{13}$$

where $R(\Delta\theta)$ is the radius of the acceptance circle needed. $\Delta\theta$ is the included angle between adjacent path lines. L is ship length. $R_{\min}$ and $R_{\max}$ are the maximum and minimum acceptance circle radius, respectively. l is an undetermined coefficient, which needs to be determined according to the ship maneuverability. In this research, tuned coefficient l is set to 13.0. $R_{\min}$ and $R_{\max}$ are set to $0.5L$ and $9L$, respectively.

2.3. Fast marching method

The FM method [34] was designed to calculate the arrival times of a monotonic wave by solving the Eikonal equation. Usually, the FM method works on a grid map whose points are divided into three categories: 1) Far points, which denote the points that the wave has not arrived and whose arrival times are not computed yet; 2) Trial points, which represent the points that the wave will arrive soon, whose arrival times have been calculated already but may be updated in later computations; 3) Accepted points, which indicate the points that the wave has already passed and whose arrival times will not be changed.

Starting from the original Accepted point, arrival times of the points on the grid map are iteratively calculated. In the beginning, all points of the grid map are marked as Far points and infinity arrival times, and the start point is marked as Accepted point, whose arrival time is set to zero. During each iteration, points adjacent to the Accepted point are marked as Trial points, and arrival times of these points are calculated. After that, the Trial point with the shortest arrival time will be marked as the Accepted point, and a new iteration begins.

The detailed calculation process of the FM method refers to [35] and [36]. The total travel cost from the start point can be obtained after the iteration computation.

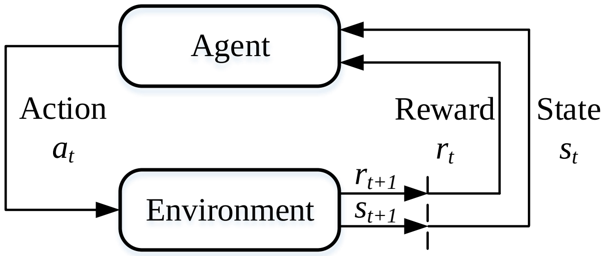

Figure 3: The MDP interaction framework of agent and environment.

2.4. Deep reinforcement learning

The DRL is described in this section. Firstly, the MDP normally used for environment modeling is introduced. And then policy gradient and deep deterministic actor-critic algorithms are discussed.

2.4.1. Markov decision process (MDP)

An MDP describes a learning environment in which objectives can be learned through continuous interaction between the agent and the environment. Furthermore, a 4-element tuple can be used to represent an MDP:

$$\mathrm{M} = [s, a, r, p], \tag{14}$$

where $s = s_1, s_2, \ldots, s_t, s_{t+1}$ represents a dynamic environment with s_t representing the state at time t. $a = a_1, a_2, \ldots, a_t, a_{t+1}$ indicates the action performed by the agent and a_t indicates the action taken at time t. $r = r_1, r_2, \ldots, r_t, r_{t+1}$ denotes the reward generated by the environment with r_t denoting the reward at time t. $\gamma \in [0, 1]$ is the discount factor for an accumulated discounted reward. p is the transition probability function defined as:

$$p_{ss'}^{a} = P\left[s_{t+1} = s' \mid s_t = s, a_t = a\right]. \tag{15}$$

Fig. 3 shows the interaction framework of an agent and an environment. The agent selects an action a_t according to observed environment state s_t. Then the environment updates its state to s_{t+1} and returns an reward r_{t+1} to the agent for responding the action a_t [37]. The selection of actions is constrained by a policy function ($\pi(a \mid s)$), which defines the probability mapping from state to action. The state-value function of state s and the action-value function of state s for a policy π are defined as:

$$v_\pi(s) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right]. \tag{16}$$

$$q_\pi(s, a) = E_\pi\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a\right]. \tag{17}$$

The goal of an RL problem is to find an optimal policy π^*, which can achieve a maximal accumulated discounted reward. That is to find either the maximum state-value function or the maximum action-value function as:

$$\pi^* = \underset{\pi}{\operatorname{argmax}} v_\pi(s) = \underset{\pi}{\operatorname{argmax}} q_\pi(s, a). \tag{18}$$

2.4.2. Policy gradient and deep deterministic actor-critic algorithm

The policy gradient based RL algorithm [38] was put forward to deal with situations involving continuous action space, in which the calculation of value function based RL algorithms such as Tabular Q learning [39] and Deep Q Network (DQN) becomes computationally expensive with a weak guarantee of convergence. In this subsection, policy gradient based RL algorithms have been well researched for the continuous control of ship path following. Similar to parameterizing a value function, a policy can be parameterized as:

$$\pi(a \mid s, \boldsymbol{\theta}) = \Pr\left\{a_t = a \mid s_t = s, \theta_t = \boldsymbol{\theta}\right\}, \tag{19}$$

where $\boldsymbol{\theta} \in \mathfrak{R}^{d'}$ is the parameter vector of policy in real domain $\mathfrak{R}$ with a dimension of d'. A scalar performance measure of the policy is defined as $J(\theta)$ and a gradient ascent can be defined to maximise $J(\theta)$ as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha \nabla J\left(\boldsymbol{\theta}_t\right), \tag{20}$$

where $\alpha \in [0, 1]$ is the learning rate. $\nabla J(\theta_t)$ is the approximated gradient of the performance measure $J(\theta_t)$. In an episodic learning case, $J(\theta)$ can be described as:

$$J(\boldsymbol{\theta}) = \sum_{s \in S} d^\pi(s) v^\pi(s) = \sum_{s \in S} d^\pi(s) \sum_{a \in A} \pi_{\boldsymbol{\theta}}(a \mid s) q^\pi(s, a), \tag{21}$$

where $d^\pi(s)$ is the stationary distribution of Markov chain for π_θ. Thus, $\nabla J(\theta)$ can be written as:

$$\nabla J(\boldsymbol{\theta}) \propto \sum_{s \in S} d^\pi(s) \sum_{a \in A} q^\pi(s, a) \nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(a \mid s). \tag{22}$$

The policy gradient method consists of two important components, which are the policy model ($\pi_\theta(a \mid s)$) and the value function ($q^\pi(s, a)$). Multiple methods, such as REINFORCE [40] and its variations, are introduced to use a sample return to measure the value function. In order to enhance the learning performance, the policy update is assisted with estimating the value function, which is the core of the policy gradient actor-critic method. By parameterizing the value function, the actor-critic method operates in a way that the value function parameters are updated by the critic whereas the policy function parameters are updated by the actor according to the direction suggested by the critic.

The policy can be either stochastic or deterministic. A stochastic policy represents a conditional probability distribution and is more applicable in a stochastic environment, where exploration is favored to find an optimal policy update result. However, this exploration always results in high computational complexity costs. To address this problem, the deterministic policy is proposed to explicitly depict a mapping (μ_θ, $\boldsymbol{\theta} \in \mathfrak{R}^m$) from state to action: $S \to A$. The deterministic policy gradient (DPG) [38] was proposed using the actor-critic method abiding by the same rule with stochastic policy gradient. Performance measure $J(\theta)$ of DPG can be described as:

$$J(\boldsymbol{\theta}) = \int_S \rho^{\mu}(s) q\left(s, \mu_\theta(s)\right) ds, \tag{23}$$

where $\rho^{\mu}(s)$ is the state distribution. The gradient of the performance measure $J(\theta)$ is defined as:

$$\begin{aligned} \nabla J(\boldsymbol{\theta}) &= \int_S \rho^{\mu}(s) \nabla_a q^{\mu}(s,a) \nabla_\theta \mu_\theta(s) \Big|_{a=\mu_\theta(s)} ds \\ &= E_{s\sim\rho^{\mu}} \left[\nabla_a q^{\mu}(s,a) \nabla_\theta \mu_\theta(s) \big|_{a=\mu_\theta(s)} \right]. \end{aligned} \tag{24}$$

To ensure a sufficient and satisfactory exploration, an off-policy learning strategy is adopted for the DPG so that a stochastic policy $\beta(a \mid s)$ is used to generate the training trajectories. The new performance measure and its gradient are expressed as:

$$J_\beta(\boldsymbol{\theta}) = \int_S \rho^{\beta}(s) q^{\mu}\left(s, \mu_\theta(s)\right) ds, \tag{25}$$

$$\nabla J(\boldsymbol{\theta}) = E_{s\sim\rho\beta} \left[\nabla_a q^{\mu}(s,a) \nabla_\theta \mu_\theta(s) \big|_{a=\mu_\theta(s)} \right]. \tag{26}$$

The deep deterministic policy gradient (DDPG) [41] is proposed by combining the DPG and DQN. It retains the feature of DQN that the Q-function is stably learned by experience replay. The pseudocode of the DDPG is presented in Algorithm 1. Besides, the DDPG extends the discrete action space to continuous action space with the help of the actor-critic framework and learns a deterministic policy [42]. Both the critic and actor have two networks with the same structure, which are the online critic/actor network and the target critic/actor network, respectively. The network parameters of DDPG are updated similar to that of DQN, i.e. by minimizing the loss between the estimated values and target values. Additionally, noise $\mathcal{N}$ is added when selecting an action from the policy $\mu_\theta(s)$, which enables a satisfactory exploration in state and action spaces for the DDPG shown as:

$$a = \mu_\theta(s) + \mathcal{N}. \tag{27}$$

Besides, different from the DQN, the target network of the DDPG is soft updated instead of staying frozen during some period of time, which stabilizes the learning process: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$ with $\tau \ll 1$.

3. The proposed approach

The framework of the proposed path following approach is shown in Fig. 5. This section details the main components of the approach. At first, a path deviation map and a corresponding perceptron are specially proposed and designed, respectively. Then, the MDP for path following considering path deviations is analyzed and put forward. Afterwards, the environment of path following used in this paper is constructed. Finally, the training process of the path following agent is detailed.

3.1. Path deviation map and perceptron

Path following is one of the major research contents of autonomous ships, and path deviation is an important evaluation index of path following. This subsection proposes a method to construct the path deviation map based on the FM method. After that, a path deviation perceptron is designed to sensor the path deviation boundary.

3.1.1. Path deviation map based on FM method

The proposed method is for tracking the planned path directly according to the path deviation. Therefore, a method to get the path deviation of the ship at any position is needed. The path deviation map needed can be a grid map. The value at each grid point in the grid map represents the minimum distance between the grid point and all path points. Therefore, using the map, the path deviation can be easily obtained regardless of the ship's position on the map. By sensing the path deviation, the ship can adjust its course to reduce the path deviation and ensure that it follows the path smoothly.

Remark 1. *To get the path deviation map, it is necessary to calculate the minimum distance between each grid point and all path points. Direct calculation of the distance is computationally expensive and is constrained by the number of grid points and the number of path points. Therefore, other solutions need to be found. Based on the research of the FM method in Section 2.3, we found that the arrival time matrix is exactly the path deviation map when the speed of the front expansion is set to 1 and all the path points are defined as Accepted points. By contrast, the FM method is more efficient and is only constrained by the number of grid points.*

There are two steps to calculate the path deviation map using the FM method. Step one is to round and project all the path points onto a grid map to get grid path points. Step two is running the FM method to generate an arrival time matrix by taking the grid path points as input. The arrival time matrix is precisely the path deviation map. The calculation process of path deviation map when using the FM method is illustrated in Fig. 4. In Fig. 4(a), path points are set as Accepted points and marked with green color. During the following iterations (Fig. 4(b)-(d)), the path deviations are calculated based on the FM method referring to [36]. In addition, as shown in Fig. 5, the path deviation map shows with color map and different colors indicate different path deviation values. It is easy to obtain the path deviation value of a ship from the map.

3.1.2. Path deviation perceptron

Path following studied in this paper requires that the path deviation does not exceed the set value during the entire path following process. In other words, the ship can only appear in specific areas and cannot exceed it during the path following process after the maximum path deviation e_p is set. For example, as shown in Fig. 6, the path deviation boundaries are shown as dashed lines, and the ship is not allowed to exceed the dashed lines surrounding the set path. Therefore, it is necessary to design a path deviation perceptron to sense

Algorithm 1: Deep deterministic policy gradient (DDPG) [41].

```
Randomly initialize critic network $Q\left(s, a \mid \theta^{Q}\right)$ and actor network $\mu\left(s \mid \theta^{\mu}\right)$ with weights $\theta^{Q}$ and $\theta^{\mu}$.
initialize target networks $Q'$ and $\mu'$ with weights $\theta^{Q'} \leftarrow \theta^{Q}$ and $\theta^{\mu'} \leftarrow \theta^{\mu}$.
initialize replay buffer $R_b$ with size $N_b$.
for $episode = 1, \cdots, M_e$ do
    initialize a random process $\mathcal{N}$ for action exploration.
    Receive initial observation state $s_1$.
    for $t = 1, \cdots, N_s$ do
        Select action $a_t = \mu\left(s_t \mid \theta^{\mu}\right) + \mathcal{N}_t$ according to the current policy and the exploration noise.
        Execute action $a_t$ and observe reward $r_t$ and observe new state $s_{t+1}$.
        Store transition $\left(s_t, a_t, r_t, s_{t+1}\right)$ in $R_b$.
        Sample a random minibatch of $N_m$ transitions $\left(s_i, a_i, r_i, s_{i+1}\right)$ from $R_b$.
        Set $y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1} \mid \theta^{\mu'}\right) \mid \theta^{e'}\right)$.
        Update critic by minimising the loss: $L = \frac{1}{N}\sum_{i=1}\left(y_i - Q\left(s_i, a_i \mid \theta^{Q}\right)\right)^2$.
        Update the actor policy using the sampled policy gradient:
          $\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_i \nabla_a Q\left(s, a \mid \theta^{Q}\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu\left(s \mid \theta^{\mu}\right)\Big|_{s_i}$.
        Update the target networks: $\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'}$, $\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'}$.
    end
end
```

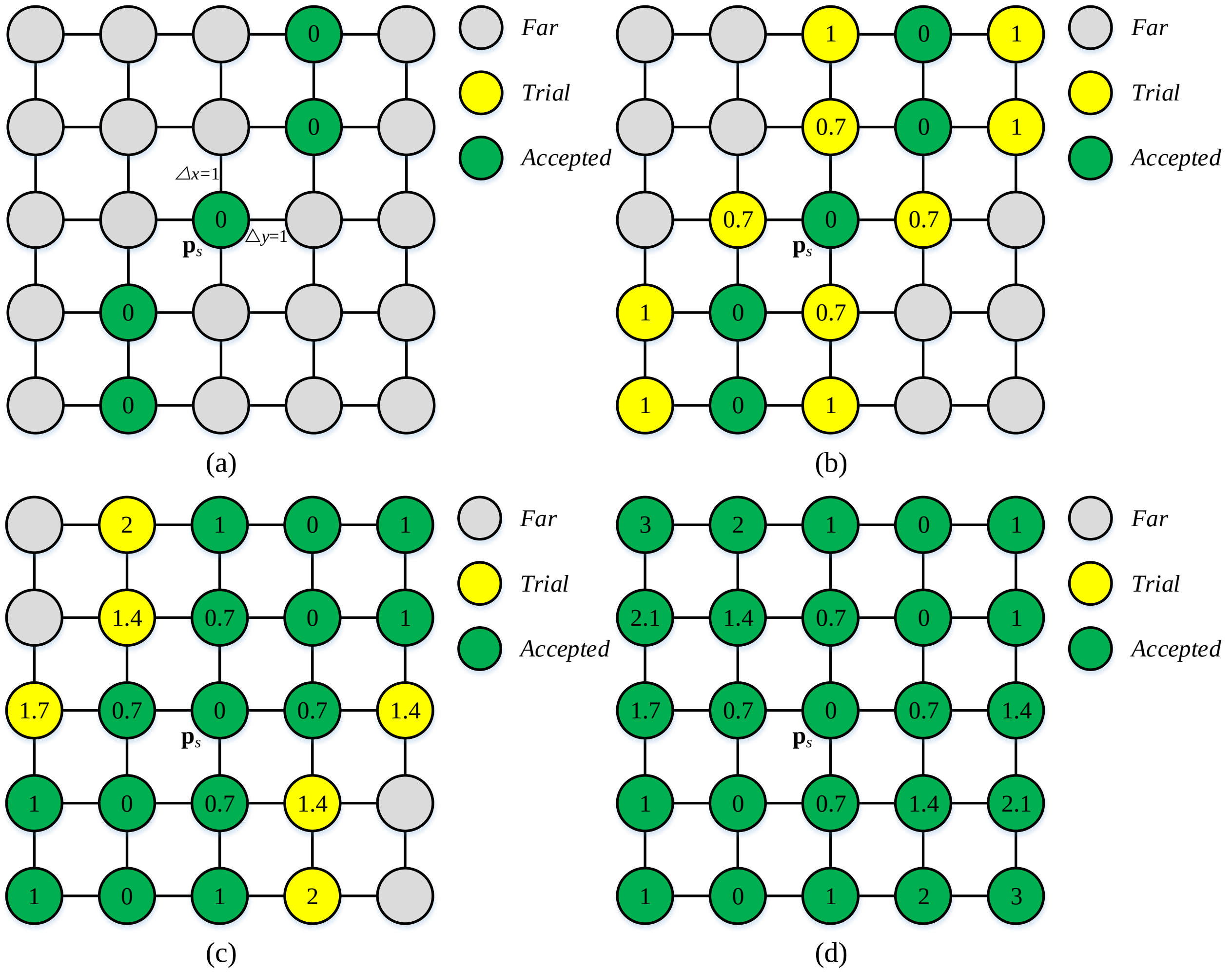

Figure 4: The calculation process of path deviation map when using the FM method.

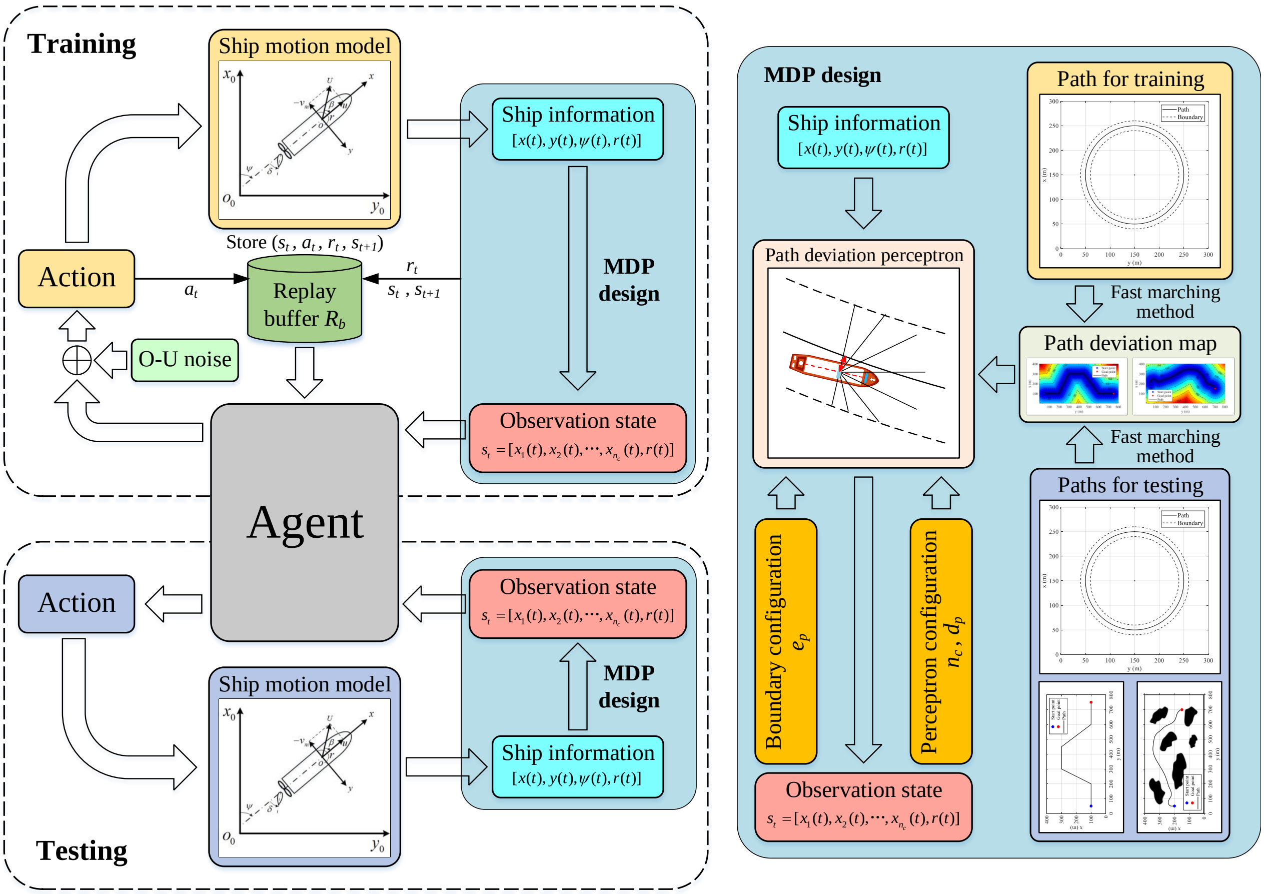
Training
Ship motion model
Action
Store (s_t, a_t, r_t, s_{t+1})
a_t
Replay buffer R_b
r_t
s_t, s_{t+1}
O-U noise
Ship information
$[x(t), y(t), \psi(t), r(t)]$
MDP design
Observation state
$s_t = [x_1(t), x_2(t), \cdots, x_{n_c}(t), r(t)]$
Agent
Action
Observation state
$s_t = [x_1(t), x_2(t), \cdots, x_{n_c}(t), r(t)]$
MDP design
Ship motion model
Ship information
$[x(t), y(t), \psi(t), r(t)]$
Testing
MDP design
Ship information
$[x(t), y(t), \psi(t), r(t)]$
Path for training
Path deviation perceptron
Fast marching method
Path deviation map
Fast marching method
Paths for testing
Boundary configuration e_p
Perceptron configuration n_c, d_p
Observation state
$s_t = [x_1(t), x_2(t), \cdots, x_{n_c}(t), r(t)]$

Figure 5: The framework of the proposed path following approach.

the path deviation boundary at any time during the entire path following process.

Remark 2. *Based on the path deviation map obtained by the method described in Section 3.1.1, it is easy to determine the location of any set path deviation boundary. After that, a special perceptron was designed to sense the path deviation boundary. As shown in Fig. 6, the perceptron simulates a range sensor and emits laser beams to detect the path deviation boundary. When the beams detect the boundary, the distance between the ship and the boundary in a certain azimuth can be obtained. Besides, if the distance exceeds the maximum detection range of the perceptron, the distance will be set to the maximum value d_p.*

The parameters of the perceptron should meet the following conditions:

$$\phi = \frac{\pi}{(n_c - 1)}, \tag{28}$$

$$d_i \leq d_p, i = 1, 2, ..., n_c, \tag{29}$$

$$x_i = \frac{d_i}{d_p}, x_i \in [0, 1], \tag{30}$$

where ϕ is the angle between two neighbor beams and n_c is the number of laser beams. The maximum detection range of the beam is d_p. And d_i is the distance value obtained by laser beam i. x_i is the normalized distance value of d_i.

3.2. MDP design for path following

Based on the special design of the path deviation map and perceptron, a path following MDP can be defined. In this subsection, the state space, action space and reward are detailed.

3.2.1. State space and action space design

In the path following problem, the ship observation state represents the current ship status according to the ship information as shown in Fig. 5. The ship information used in this study includes ship coordinates (x, y), ship heading ψ and ship yaw rate r.

Remark 3. *The ship observation state is characterized by the normalized distance values $x_i \in [0, 1]$ of laser beam i. i.e. $x_i = 0$ when the ship exceeds the path deviation boundary and $x_i = 1$ when the laser beam i does not detect the boundary. In other cases, x_i is between 0 and 1. Besides, taking into account the inertia of the ship, the observation state is also characterized by the ship yaw rate r.*

At time step t, the actual observation state of the ship is

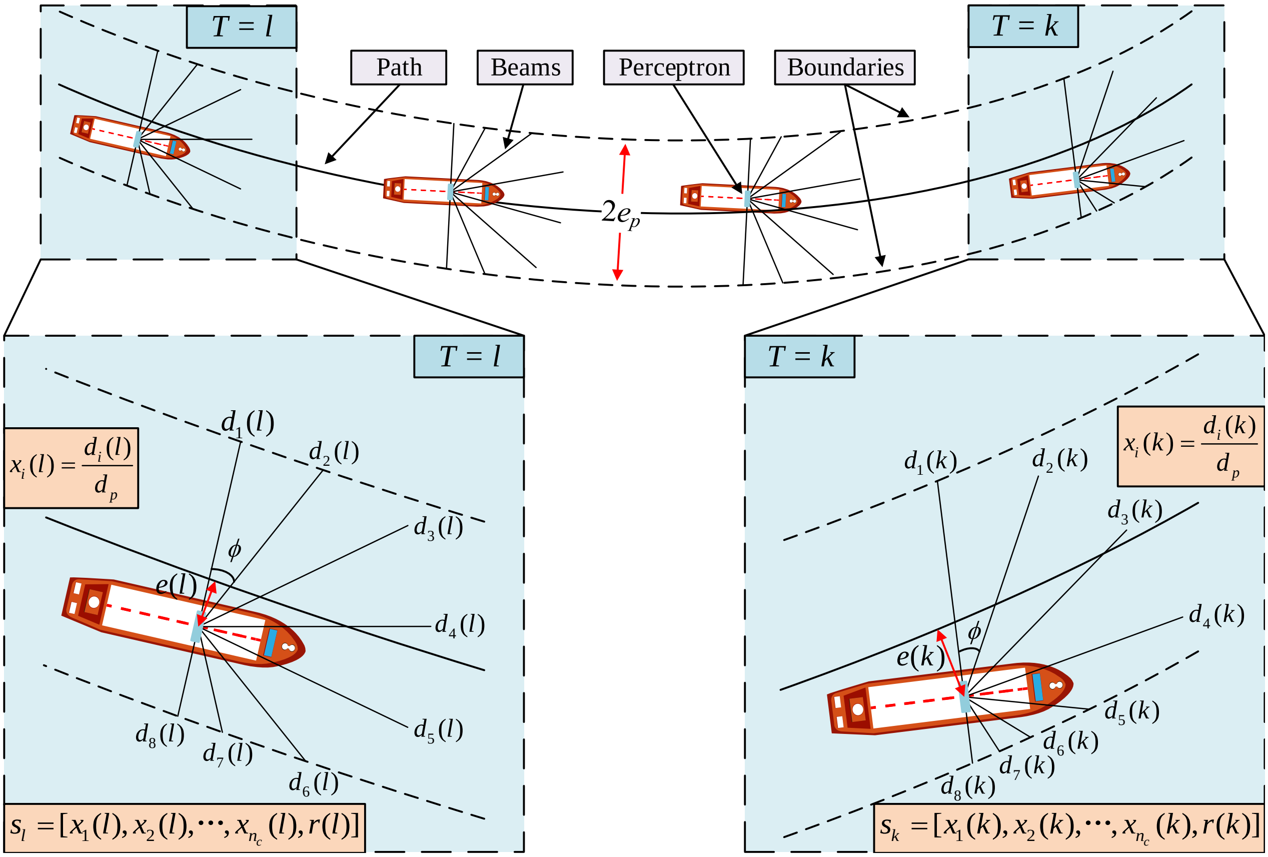

Figure 6: Overview of the path deviation perceptron.

shown as follow:

$$s_t = \left[x_1(t), x_2(t), \dots, x_{n_c}(t), r(t)\right], \tag{31}$$

where $x_i(t)$ is the normalized distance value at time step t. s_t is the actual state of ship at time step t. Note that the actual state s_{t+1} is resulted from state s_t, taking action a_t.

In RL, the agent interacts with the environment by taking actions in various systems states. The action taken by the agent is the rudder angle in this study. The action space is defined as follow:

$$a_t = [\delta(t)], \tag{32}$$

where $\delta(t)$ is the command of rudder angle at time step t.

Usually, the action of the DDPG is oscillating or fluctuating during the decision process, which will be difficult to apply to the actual ship control system directly. To address this issue, the simple moving average (SMA) method was adopted to smooth the action serials.

$$a_{k,t+1} = a_{k,t} + \frac{1}{k}\left(a_{t+1} - a_{t-k+1}\right), \tag{33}$$

where $a_{k,t}$ and $a_{k,t+1}$ are the SMA values with sampling width k at time step t and $t+1$, respectively. a_{1-k}, a_{2-k}, …, a_{-1} and $a_{k,0}$ are equal to a_0. In this research, sampling width k was set to 6.

3.2.2. *Reward design*

The reward design plays a critical role in RL. The environment returns reward signal r_{t+1} to the agent when action a_t is taken by the agent. The value of r_{r+1} represents the immediate reward after transition from state s_t to state s_{t+1} with action a_t. Path following aims to maintain a low path deviation during the whole procedure. Therefore, the reward function is defined as:

$$r_{t+1} = \begin{cases} -1, & \text{if the ship exceeds the boundary.} \\ 1 - \frac{2e(t+1)}{e_p}, & \text{otherwise.} \end{cases} \tag{34}$$

where $e(t+1)$ is the path deviation of the ship at time step $t+1$. e_p is the set maximum deviation value. The negative reward of -1 means the agent is penalized if the next state is not feasible or the ship exceeds the boundary. The $1-\frac{2e(t+1)}{e_p}$ function normalises the path deviation e_{t+1} to a reward signal in the range of [-1, 1]. Note that the *termination* will be set to $True$ if the ship exceeds the path deviation boundary or the episode completes at this step.

3.3. Path following control using deep deterministic policy gradient

In this work, a path following control was achieved using the DDPG algorithm. A training strategy has been designed as shown in Fig. 5. The observation state designed for the path following environment allows a ship to learn an optimal path following policy by interacting with the environment using a rudder action signal. Besides, such an observation state and the training strategy provide an extendibility for ship path following in different scenarios.

Remark 4. *Once the policy network's parameters have been successfully updated, the trained policy network can be easily applied in a new path following task. Also, new path deviation requirements can be easily adapted by adopting new maximum detection ranges of beams. A circular path was involved in the training procedure. During the testing phase, different cases were designed for validating the adaptability of the proposed path following approach. Benefiting from the specific observation state and reward designs, the ship in different path following cases can reuse the policy learned from the training procedure.*

As shown in Fig. 7, a structure with two hidden layers (each layer containing 400 and 300 neurons respectively) has been adopted for both actor and critic networks in the DDPG algorithm. Besides, the Ornstein-Uhlenbeck (O-U) process $\mathcal{N}$ was added for action selection to fully explore state and action space. The O-U process can be defined as:

$$\mathcal{N}_{t+1} = \mathcal{N}_t + \theta_{\mathrm{OU}} \left(\mu_{\mathrm{OU}} - \mathcal{N}_t \right) + \sigma_{\mathrm{OU}} \mathcal{N}(0, 1), \quad (35)$$

where $\mathcal{N}_t$ and $\mathcal{N}_{t+1}$ are the values of O-U process at times t and $t+1$, respectively. $\mathcal{N}_1$ is set to zero. $\mathcal{N}(0, 1)$ is a Gaussian process with a mean of 0 and a standard deviation of 1. In this work, the parameters of O-U process are specified as $\mu_{OU} = 0$, $\theta_{OU} = 0.15$, $\delta_{OU} = 0.2$. Other parameters for DDPG based path following control can be viewed in Table 2.

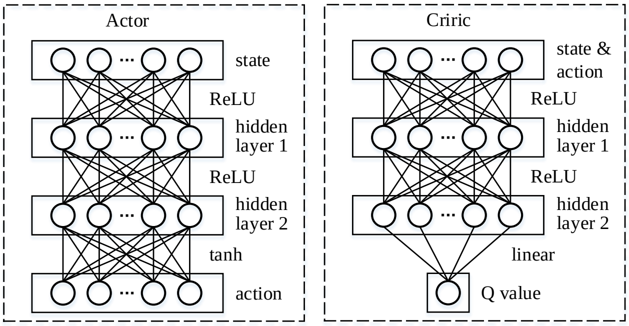

Figure 7: The neural network design for the actor-critic framework.

3.4. Environment of path following

The environment of path following for the RL training consists of two parts, i.e. the ship dynamic model (see Section 2.1) and the MDP definition of path following (see Section 3.2). Algorithm 2 depicts how the path following environment is formulated. All parameters of the ship should be set before the start of the first episode. And the parameters of the path following should also be configured. In each episode, the environment randomly selects the start position and heading of the ship in a set range. At each time step of an episode, action a_t generated by the agent is determined at state s_t. Then, taking action a_t as input, the position and heading of the ship are updated using equation (2)-(5). Afterwards, state s_{t+1} and immediate reward r_t can be calculated with equation (30), (31) and (34). This process repeats until the $Termination$ is equal to $True$. Note that the environment would carry out the termination of an episode when the ship exceeds the boundary or the maximum time step has been reached.

Table 2
List of parameters used for deep deterministic actor-critic algorithm in this study.

No.	Parameter	Value
1	Actor learning rate r_a	0.0001
2	Critic learning rate r_c	0.001
3	Soft update τ	0.01
4	Discount rate γ	0.9
5	Memory size N_b	2000
6	Minibatch size N_m	16
7	Hidden layer 1	400 units
8	Hidden layer 2	300 units

3.5. Agent training

The target of the path following task is to minimize the overall travel cost from the start point to the goal point. The overall travel cost in this study is a cumulative value which is the sum of the reward defined in equation (34). Such a path-following task is intended to train an agent for future path following control in different path tasks. The training process is an episodic task. In each step of an episode, a set of historical path following experiences are randomly sampled from the replay buffer for the agent to learn policies by minimizing the travel cost. This training process repeats until the average episode reward converges. The historical path following experience is collected in each step during the whole training procedure.

The RL agent training and policy application follow the procedure presented in Fig. 8. At the beginning of the procedure, the RL training parameters such as the learning rate and the discount rate, and the path following parameters such as the path deviation and the perception beam count should be configured. During the training period, the agent interacts with the path-following environment for storing transition in the replay buffer. Then, the agent is trained with experience replay. Once the training is converged, the learned path following policy needs to be validated in different path following cases. In the application phase, the policy can be saved for future path-following tasks when it is feasible in all testing cases. Otherwise, more episodes will be needed for the training procedure.

As with most reinforcement learning algorithms, the use of non-linear function approximators nullifies any convergence guarantees. However, it is still possible to ensure a stable learning process with a special design of the MDP

Algorithm 2: Environment of the path following problem.

```
Set the parameters of the ship.
Configure the parameters of path planning.
for $episode = 1, \cdots, M_e$ do
    Randomly select the start position and heading of the ship according to a set range.
    Set $Termination = False$.
    Calculate the state $s_1$ with equation (30) and (31).
    for $t = 1, \cdots, N_s$ do
        Generate action input $a_t$ from the agent according to state $s_t$.
        Calculate the new positions and headings of the ship with equation (2)-(5).
        Update the state $s_{t+1}$ with equation (30) and (31).
        Get the immediate reward $r_t$ with equation (34).
        if the ship exceeds the boundary then
            Set $Termination = True$.
            break.
        end
    end
    Set $Termination = True$.
end
```

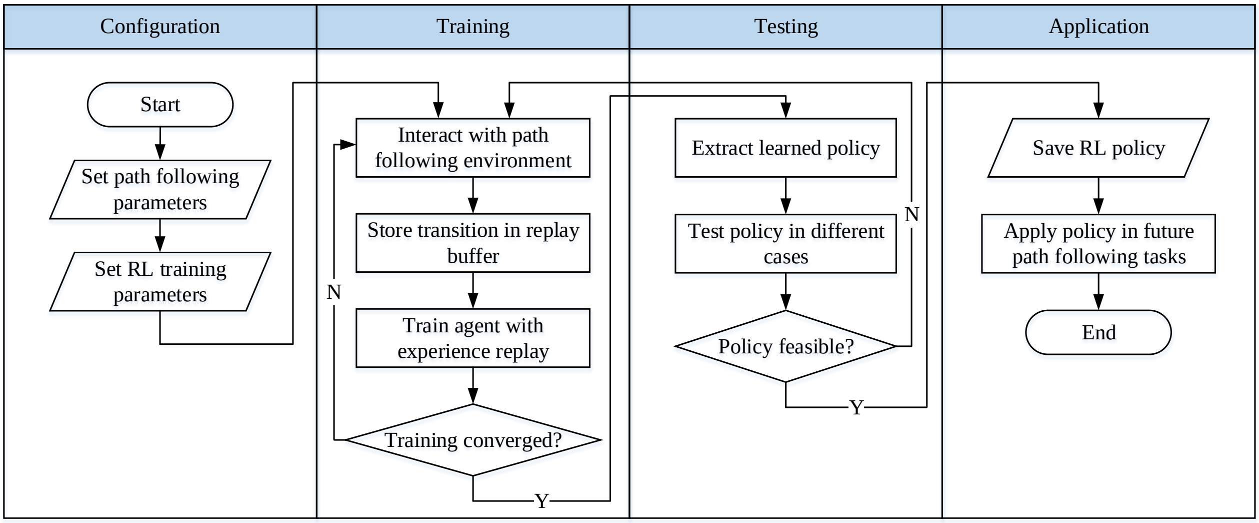
Configuration
Training
Testing
Application
Start
Set path following parameters
Set RL training parameters
Interact with path following environment
Store transition in replay buffer
Train agent with experience replay
Training converged?
N
Y
Extract learned policy
Test policy in different cases
Policy feasible?
N
Y
Save RL policy
Apply policy in future path following tasks
End

Figure 8: The agent training and policy application procedure of the reinforcement learning (RL).

[41]. In this way, the convergence of the path following errors can be guaranteed.

4. Simulation results and analysis

To verify and validate the performances of the proposed path following approach, a set of simulations have been carried out in this section. This work aims to develop a path following policy by employing the DDPG algorithm with the help of the path deviation map and perceptron. More importantly, the trained policy network can be directly applied to new path following tasks without further training. It should be noted that a steady wind ($U_W = 2.0\,\mathrm{m/s}$, $\theta_W = 30°$) and a constant current ($U_C = 0.08\,\mathrm{m/s}$, $\theta_C = 60°$) as not observed external disturbances were introduced into all the testing cases but not into the training procedure. The simulation results in this section are presented in the following ways:

1) Demonstrating the path following performance of the designed DRL structure. The training results of the developed DDPG algorithm on a circular path are presented in Section 4.1;

2) Validating the effectiveness of the proposed path following approach. Using the trained policy network in Section 4.1, simulations of path following results demonstrating on the same circular path are presented in Section 4.2;

3) Validating the adaptability and advancement of the proposed approach by using different paths to follow. The path following policy on a polyline path and a curved line path are tested in Section 4.3 and Section 4.4, respectively.

4.1. Training results

The training of the proposed algorithm was carried out on a circular path with a 7-meter KVLCC2 ship [24]. It is an-

Table 3
List of parameters used for training

No.	Parameter	Value
1	Maximum step N_s	600
2	Maximum path deviation e_p	10 m
3	Control period T_s	1 s
4	Initial position x	$[245, 255]$ m
5	Initial position y	150 m
6	Initial heading ψ	90 or 270 °
7	Initial speed u	1.179 m/s
8	Initial speed v	0 m/s
9	Initial rudder angle δ_0	0 °
10	Rudder angle δ	$[-10, 10]$ °
11	RPS n_r	10.4 Hz
12	Perception distance d_p	40 m
13	Perception beam count n_c	8

ticipated that the ship can learn a generalized strategy by performing training in a simple scenario and apply the learned strategies to accommodate more complicated paths.

The circular path with a radius of 100 m used for training in this work is shown in Fig. 9. The solid line indicates the path that needs to be followed, and the dashed line indicates the boundary of the path deviation (maximum path deviation). The parameters used in training are listed in Table 3.

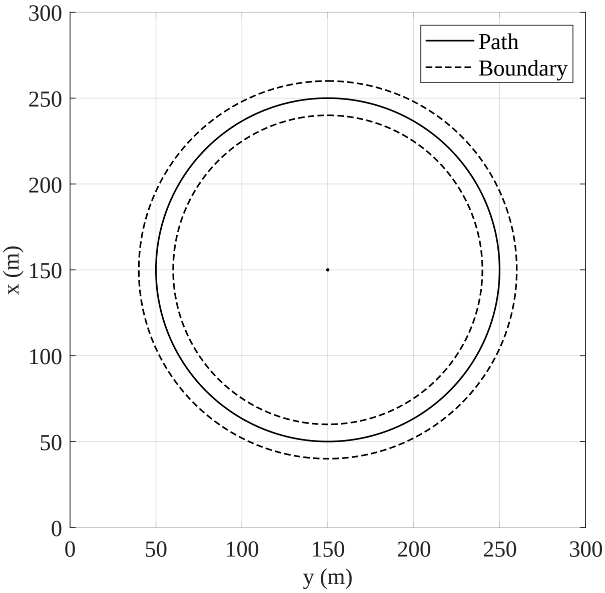

Figure 9: The training environment for path following task.

A training environment with dimension of 300 m $*$ 300 m was adopted. The maximum steps N_s for a training episode was set to 600 and the maximum path deviation e_p for path following was set to 10 m in this study. In addition, the control period T_s was set to one second. The parameters of the ship used in this study are also listed in Table 3. It should be noted that the initial speed was set to 1.179 m/s, which is the designed speed of the 7 meters KVLCC2. To improve the adaptability of training strategies, the initial position x and initial heading ψ of the ship during training were randomly selected within a certain range. The rudder angle δ used in this study was limited to a range between $-10°$ and $10°$. Other important parameters for training such as the perception distance d_p and the beam count n_c are also shown in Table 3. During training, an episode will be terminated if the number of steps reaches the maximum step N_s or the ship exceeds the path deviation boundary.

In order to have full visibility of the designed DRL based path following approach, the randomness of the training process has been taken into account by adopting a random seed across the parameters within the built neural networks to achieve reproducibility. The results are shown in Fig. 10 and Fig. 11. It can be observed that the steps of path following can fast converge to the maximum value (Fig. 10). The cumulative reward value of the path following (Fig. 11) can also converge, which confirms that the training has been successful.

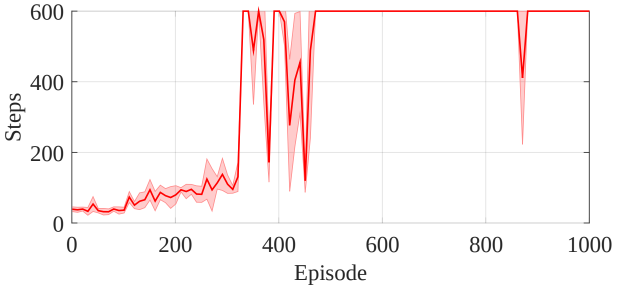

Figure 10: The cumulative steps of the training.

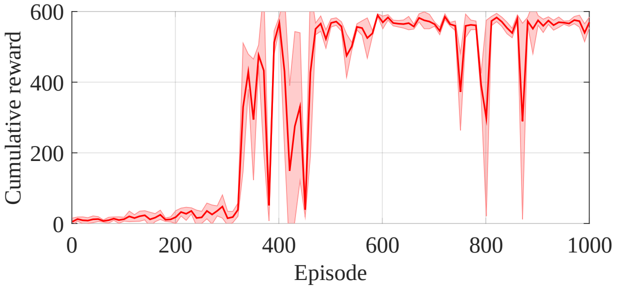

Figure 11: The cumulative reward value of the training.

4.2. Testing case 1

The learned policy was directly used to test the performance of path-following control by tracking the same circular path in this testing case. As shown in Fig. 12, a 7 m initial position error has been configured to verify if the ship can take the correct manoeuvring actions and keep following the path. Special parameters for the path following used in this case are listed in Table 4. Other parameters can be viewed in Table 3.

As shown in Fig. 12, the tracking trajectory of the path following has been displayed with the red dash-dot line. Al-

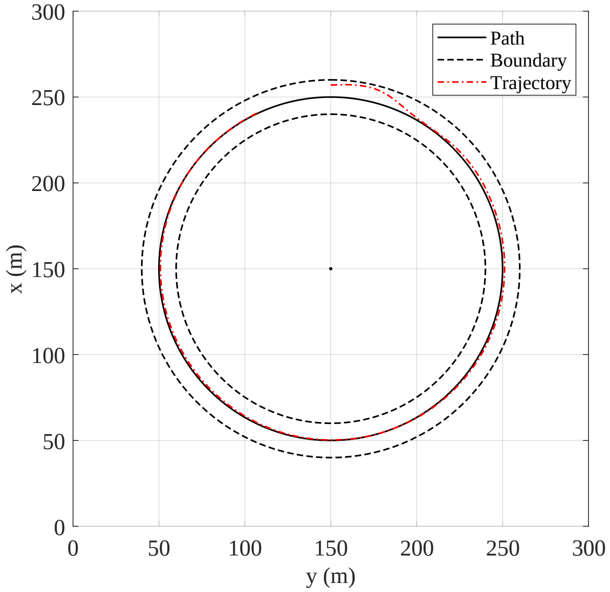

Figure 12: The trajectory of path following in testing case 1.

Table 4
List of special parameters used in path following testing case 1.

No.	Parameter	Value
1	Start point p_s	(257, 150) m
2	Initial heading ψ	90 °

though the initial position of the ship slightly deviates from the path and environmental disturbances (wind and current) are introduced, the ship is capable of reducing and maintaining its path deviation by employing the trained policy. Such a performance can be further validated by the quantities evaluation results shown in Fig. 13. The results show that the ship has a good performance of keeping the path deviation at a low level. Fig. 14 shows the rudder angle during the path following procedure. It can be found that the ship is capable of following the path with a small rudder angle changing.

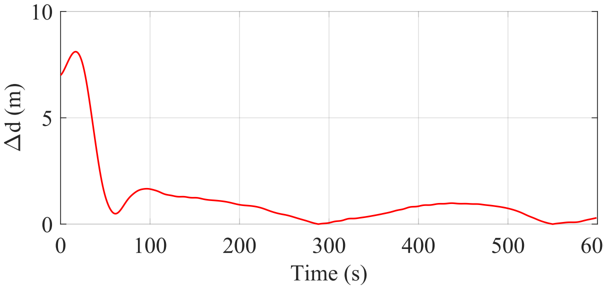

Figure 13: The path deviation of path following in testing case 1.

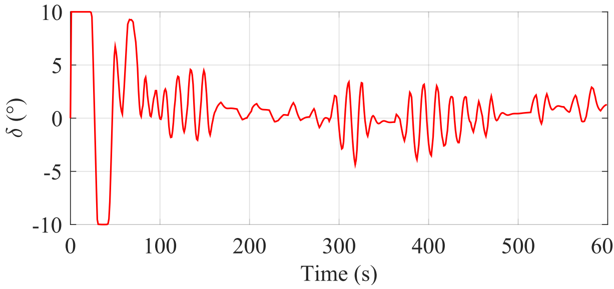

Figure 14: The rudder angle of path following in testing case 1.

Table 5
List of special parameters used in path following testing case 2

No.	Parameter	Value
1	Start point p_s	(107, 50) m
2	Goal point p_g	(100, 750) m
3	Initial position p_i	(100, 50) m
4	Initial heading ψ	90 °
5	Perception distance d_p	60 m
6	Maximum path deviation e_p	15 m

Table 6
List of waypoints and corresponding acceptance circle radiuses used in path following testing case 2

No.	Waypoints	Radiuses
1	(100.00, 50.00) m	-
2	(100.00, 100.00) m	3.50 m
3	(100.00, 200.00) m	30.45 m
4	(300.00, 300.00) m	30.45 m
5	(300.00 450.00) m	19.46 m
6	(100.00 600.00) m	19.46 m
7	(100.00 750.00) m	5.00 m

4.3. Testing case 2

In this test, the adaptability of the proposed method has been verified by following a polyline path. The polyline path consists of six waypoints and five straight lines. The color map with contours of the path deviation corresponding to the polyline is shown in Fig. 15. Special parameters such as start point, goal point, initial position and initial heading of the ship used in this test are listed in Table 5. Different from the testing case 1, larger perception distance d_p (60 m) and maximum path deviation e_p (15 m) were used for more challenging path following tasks. In addition, the same environmental disturbances were considered in this testing case. Other parameters are outlined in Table 3.

For comparisons, a guidance and control system was used to follow the path in the same condition. The ALOS guidance law acted as the guidance module. Correspondingly, a PD and a PID heading tracking controllers were used as the control module. The radius of the acceptance circles used in the ALOS guidance law was calculated by equation (12)-

Table 7
List of performance indexes of path following in testing case 2

Method	Length (m)	Time (s)	Avg Speed (m/s)	Avg deviation (m)	Max deviation (m)
Proposed	906.17	879	1.031	1.56	7.20
ALOS+PD	919.12	909	1.011	2.13	9.31
ALOS+PID	919.37	915	1.005	2.02	7.80

Table 8
List of special parameters used in path following testing case 3

No.	Parameter	Value
1	Start point p_s	(200, 50) m
2	Goal point p_g	(150, 700) m
3	Initial position p_i	(200, 50) m
4	Initial heading ψ	0 °
5	Perception distance d_p	60 m
6	Maximum path deviation e_p	15 m

Table 9
List of waypoints and corresponding acceptance circle radiuses used in path following testing case 3

No.	Waypoints	Radiuses
1	(200.00, 50.00) m	-
2	(232.40, 53.59) m	43.40 m
3	(239.37, 86.48) m	8.38 m
4	(220.74, 132.88) m	11.65 m
5	(237.54, 180.08) m	3.50 m
6	(253.24, 227.58) m	3.55 m
7	(272.41, 273.88) m	3.67 m
8	(297.42, 317.18) m	3.50 m
9	(322.87, 360.28) m	3.70 m
10	(342.02 406.58) m	4.88 m
11	(344.39 456.58) m	3.93 m
12	(336.64 505.98) m	5.68 m
13	(309.36 547.95) m	4.37 m
14	(271.56 580.76) m	4.07 m
15	(227.26 604.12) m	4.63 m
16	(192.39 640.00) m	5.16 m
17	(173.02 686.20) m	9.31 m
18	(150.00 700.00) m	5.00 m

Table 10
List of parameters of PID controllers in testing case 3

Method	Kp	Ki	Kd
ALOS+PD	1.80	0.00	10.00
ALOS+PID	2.32	0.15	11.58
ALOS+PID1	1.83	0.11	9.86
ALOS+PID2	1.68	0.10	9.72

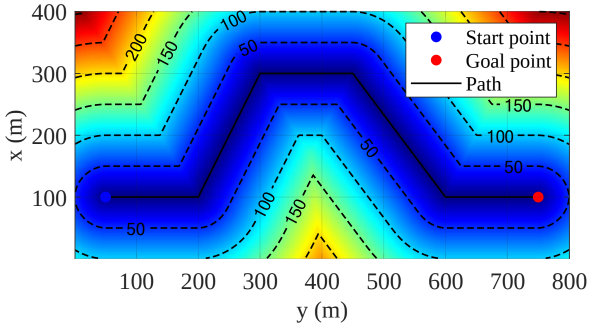

Figure 15: The path deviation color map with contours in testing case 2.

(13) and the tolerance for ships arriving at the goal point was equal to 5 m. In the PD controller, the proportional and derivative values were 1.80 and 10.00, respectively. And in the PID controller, the proportional, integral and derivative values were 2.32, 0.15 and 11.58, respectively. It should be noted that all the parameters of the PD and PID controllers have been tuned (manual and automatic) [43].

Using the same path following policy in Testing case 1, the results revealing the tracking trajectories of the ship are shown in Fig. 16. The start point is located at the bottom left of the map shown as a blue dot and the goal point shown as a red dot is located at the bottom right of the map. The planned path is displayed with a black line. And the black circles along the path are the waypoints. The coordinates and the corresponding acceptance circle radiuses of the waypoints are listed in Table 6. It should be noted that a 7 m initial position deviation was configured to simulate a step function line and waypoint 2 was added to accelerate the convergence rate of ALOS-based path following methods. There are three kinds of trajectories shown in Fig. 16. The red dash-dot, blue dotted and green dashed trajectories are generated by proposed, ALOS PD and ALOS PID methods, respectively. It can be observed that the ship is able to keep following the set path, although the path deviations are slightly larger near waypoint 5 owing to the environmental disturbances. In fact, it is a challenge for ships to follow a sharp turn path with disturbances.

Quantitative assessment of the path following methods in this test is presented as shown in Fig. 17 and Fig. 18. It can be seen that the ship has learned to adjust its rudder angle to perform the path following control and reduce the path deviation to a small range. The curves of path deviation and rudder angle with time are indicated with different colors. The red solid, blue dotted and green dashed curves represent the results produced by the proposed, ALOS PD and ALOS PID methods, respectively. According to Fig. 17, there are 3, 3 and 5 path deviation peaks (exclude the first peak at the

Table 11
List of performance indexes of path following in testing case 3

Method	Length (m)	Time (s)	Avg Speed (m/s)	Avg deviation (m)	Max deviation (m)
Proposed	795.41	775	1.026	1.36	5.38
ALOS+PD	805.64	799	1.008	1.48	7.24
ALOS+PID	807.27	811	0.995	1.69	5.58
ALOS+PID1	809.59	813	0.996	2.16	13.13
ALOS+PID2	805.22	811	0.993	1.82	5.17

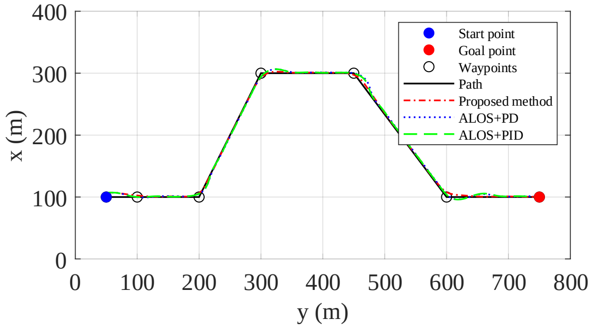

Figure 16: The trajectory of path following in testing case 2.

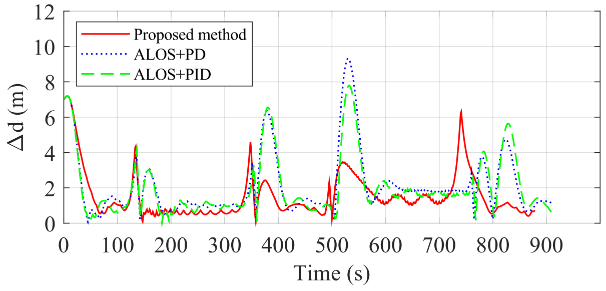

Figure 17: The path deviation of path following in testing case 2.

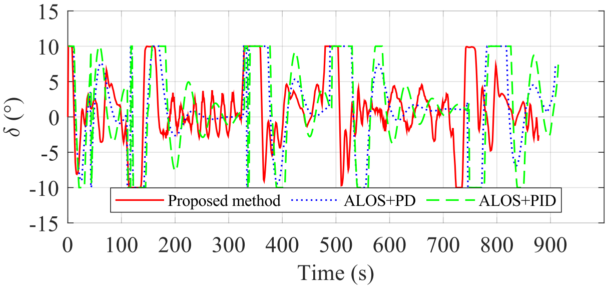

Figure 18: The rudder angle of path following in testing case 2.

start point) of the proposed, ALOS PD and ALOS PID methods over 4 m, respectively. At the same time, the counts of peaks become 0, 1 and 1 when over 7 m. What's more, the path deviation of the proposed method is always at a lower level compared with other methods. Every time the ship passes a turning waypoint, the path deviation will quickly converge after a local peak. It is obvious that the proposed method shows great ability in low path deviation path following, which indicates that the method has the potential to be applied to the path following of autonomous ships. On the basis of Fig. 18, the rudder angle changes of the above three path following methods are not smooth enough during the whole path following procedure. All three methods produce large rudder angle changes near the waypoints. In addition, the rudder angle of the proposed method fluctuates slightly during the entire path-following procedure for keeping a low path deviation. Therefore, a measure or a punishment could be added to limit the rudder angle change during the strategy learning procedure when the rudder angle change is required to be smoother.

For further quantitatively analyzing the performance of the above three path following approaches, the trajectory length, sailing time and average speed of the ship are calculated. At the same time, the average and maximum deviations of the ship during the path following procedure are also calculated. The results are given in Table 7. It can be easily found that the proposed method performs the best and the ALOS PD method performs the worst among the three methods. The proposed method has the shortest trajectory length and sailing time which are 906.17 m and 879 s, respectively. In addition, it has the highest average speed of 1.031 m/s, in spite of the frequent changing of rudder commands. The above indexes indicate that the proposed method has the highest overall efficiency. As for the deviations, the proposed method has the smallest average deviation (1.56 m) and maximum deviation (7.02 m), respectively, which shows outstanding safety performance during the whole sailing.

4.4. Testing case 3

The adaptability of the proposed method is further verified in this test by following a more complex curved. A particularly complex environment used for this test is shown in Fig. 19. And a collision-free path is generated by the angle-guidance fast matching square (AFMS) method [44] for this test. Besides, the color map with contours of the path deviation in this test is shown in Fig. 20. Based on the above two figures, it can be found that ship collision avoidance can be achieved by following the planned path with low path deviations. Parameters for the path following used in this test are detailed in Table 3 and Table 8.

In this case, the parameters of path-following methods used for comparisons are the same as used in Testing case 2

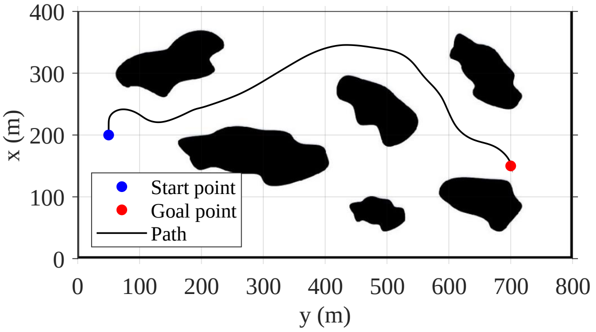

Figure 19: The planned path based on the angle-guidance fast matching square (AFMS) method for testing case 3.

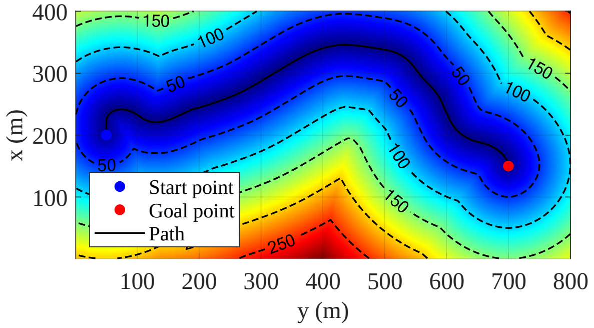

Figure 20: The path deviation color map with contours in testing case 3.

including the environmental disturbances described in Section 2.1. Meanwhile, the path following policy learned in Section 4.1 is reused again. Fig. 21 shows the results revealing the trajectories of the ship. The blue dot and red dot located at the two sides of the map are the start point and goal point, respectively. The waypoints, whose coordinates and corresponding acceptance circle radiuses are listed in Table 9, are shown as a series of black circles along the path (black solid line). Besides, the trajectories generated by the three methods are shown with red dash-dot, blue dotted and green dashed lines, respectively. It is obvious that all three methods perform good during the whole path following procedure.

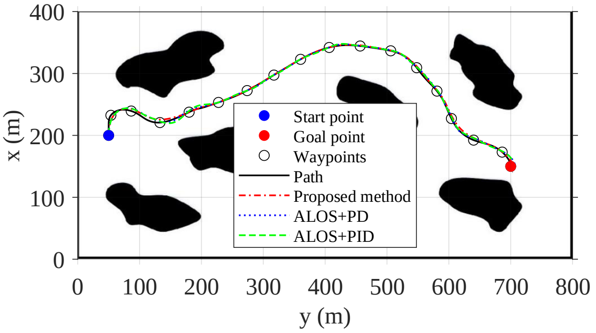

Figure 21: The trajectory of path following in testing case 3.

In Fig. 22 and Fig. 23, the path deviation and rudder angle curves of different methods are shown with different colors. The results obtained by the proposed, ALOS PD and ALOS PID methods are shown with red solid, blue dotted and green dashed lines, respectively. The quantitative results further confirm that the proposed method shows great adaptability in path following among the three methods. However, through the analysis of Fig. 21 and Fig. 22, it can be found that when the path curvature is very large, for example during the initial stage of the path following, the proposed method has a comparatively large path deviation. To address this issue, it is expected to eliminate the above shortcomings by reducing the radius of the circular path and increasing the range of the rudder angle during training. In addition, near the end of this path-following task, the proposed method does not maintain the path deviation at a low level very well owning to the environmental disturbances. Fortunately, this situation does not always happen and can be mitigated by introducing environmental disturbances into the training procedure. Besides, the rudder angle of the proposed method in this test also fluctuates as in testing case 2 as shown in Fig. 23. In order to get smoother rudder commands in real ship control, it needs to take more effort into reward function optimization.

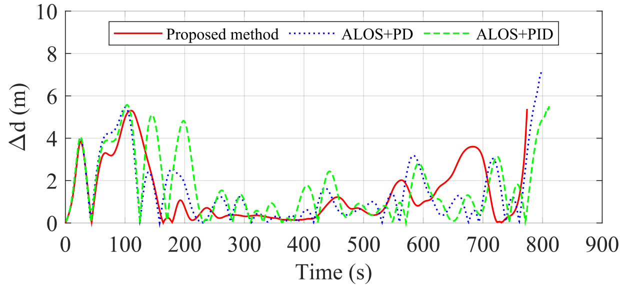

Figure 22: The path deviation of path following in testing case 3.

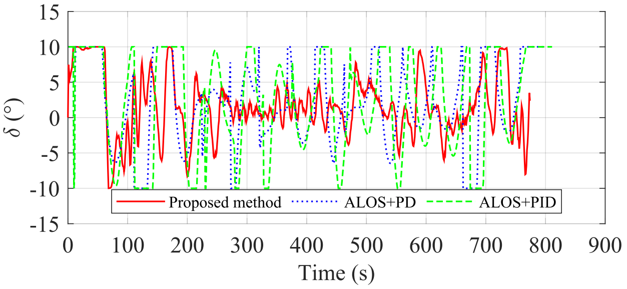

Figure 23: The rudder angle of path following in testing case 3.

For further comparison, another two ALOS PID controllers are introduced in testing case 3. All the parameters of PID controllers used in this case are listed in Table 10.

Indexes used in Testing case 2 are also calculated in this test for further quantitative evaluation of the above path following approaches. The values of the above indexes are given in Table 11. According to the average deviation, the ALOS PD controller performs better than the ALOS PID controllers. In addition, the ALOS PID2 shows the best competitiveness among the four traditional methods due to the shortest trajectory length and the smallest maximum deviation. It should be noted that the ALOS PID2 also has a smaller maximum deviation than the proposed method. However, the proposed method shows the highest overall efficiency with the shortest trajectory length (795.41 m), the shortest sailing time (775 s), the highest average speed of 1.026 m/s and the smallest average deviation (1.36 m). Based on the above comparison, it is evident that the proposed method shows better effectiveness and adaptability in the path following task.

5. Conclusions and future work

In this work, a novel path following approach for autonomous ships has been developed using the FM method and the DRL. The FM method is used to generate the path deviation map. And the DDPG algorithm is employed as the underpinning strategy to achieve a path following control in continuous action spaces. The special design of the path deviation map and perceptron for sensing the path deviation boundary makes the learned strategy can be easily reused in different path following tasks without any further training. The proposed method shows well competitiveness compared to traditional methods (ALOS PD and ALOS PID) in simulations.

In further research, significant external disturbances and large rudder angles need to be researched. It is necessary to design an environmental disturbance observer for better performance of DRL. The training of the DRL algorithm can also be improved. On the one hand, to get better performance in different path following tasks, more circular paths with different radiuses can be introduced into the training process. On the other hand, more advanced RL algorithms such as Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) can guarantee non-decreasing long-term reward by introducing a surrogate objective function and a Kullback-Leibler divergence (KLD) constraint. Using these new RL algorithms, the training performance, as well as the complexity of implementation and computation, can be improved.

Acknowledgements

This research was supported by High-tech Ship Research Projects Sponsored by MIIT- Green Intelligent Inland Ship Innovation Programme, National Key R&D Program of China [2018YFB1601503] and Ministry of Industry and Information Technology of the People's Republic of China [2018473].

References

[1] Cunlong Fan, Krzysztof Wróbel, Jakub Montewka, Mateusz Gil, Chengpeng Wan, and Di Zhang. A framework to identify factors influencing navigational risk for maritime autonomous surface ships. *Ocean Engineering*, 202:107188, 2020.

[2] Pengfei Chen, Yamin Huang, Junmin Mou, and PHAJM van Gelder. Probabilistic risk analysis for ship-ship collision: state-of-the-art. *Safety science*, 117:108–122, 2019.

[3] Yamin Huang, Linying Chen, Pengfei Chen, Rudy R Negenborn, and PHAJM van Gelder. Ship collision avoidance methods: State-of-the-art. *Safety science*, 121:451–473, 2020.

[4] Mingyang Zhang, Di Zhang, Houjie Yao, and Kai Zhang. A probabilistic model of human error assessment for autonomous cargo ships focusing on human–autonomy collaboration. *Safety science*, 130:104838, 2020.

[5] Tengfei Wang, Qing Wu, Jinfen Zhang, Bing Wu, and Yang Wang. Autonomous decision-making scheme for multi-ship collision avoidance with iterative observation and inference. *Ocean Engineering*, 197:106873, 2020.

[6] Bing Wu, Tingting Cheng, Tsz Leung Yip, and Yang Wang. Fuzzy logic based dynamic decision-making system for intelligent navigation strategy within inland traffic separation schemes. *Ocean Engineering*, 197:106909, 2020.

[7] Xinping Yan, Chao Wu, and Feng Ma. Conceptual design of navigation brain system for in-telligent cargo ship. *Navigation of China*, 40(4):95–98, 2017.

[8] Xinping Yan, Feng Ma, Jialun Liu, and Xuming Wang. Applying the navigation brain system to inland ferries. In *18th international conference on computer and IT applications in the maritime industries (COMPIT 2019), Tullamore, Ireland*, pages 25–27, 2019.

[9] Zhilin Liu, Guosheng Li, Jun Zhang, and Linhe Zheng. Path following for underactuated surface vessels with disturbance compensating predictive control. *International Journal of Advanced Robotic Systems*, 17(2):1729881420920039, 2020.

[10] Thor I Fossen. Guidance and control of ocean vehicles. *University of Trondheim, Norway, Printed by John Wiley & Sons, Chichester, England, ISBN: 0 471 94113 1, Doctors Thesis*, 1999.

[11] Thor I Fossen, Kristin Y Pettersen, and Roberto Galeazzi. Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. *IEEE Transactions on Control Systems Technology*, 23(2):820–827, 2014.

[12] Thor I Fossen, Morten Breivik, and Roger Skjetne. Line-of-sight path following of underactuated marine craft. *IFAC proceedings volumes*, 36(21):211–216, 2003.

[13] Thor I Fossen and Kristin Y Pettersen. On uniform semiglobal exponential stability (usges) of proportional line-of-sight guidance laws. *Automatica*, 50(11):2912–2917, 2014.

[14] Walter Caharija, Kristin Y Pettersen, Marco Bibuli, Pedro Calado, Enrica Zereik, José Braga, Jan Tommy Gravdahl, Asgeir J Sørensen, Milan Milovanović, and Gabriele Bruzzone. Integral line-of-sight guidance and control of underactuated marine vehicles: Theory, simulations, and experiments. *IEEE Transactions on Control Systems Technology*, 24(5):1623–1642, 2016.

[15] Thor I Fossen and Anastasios M Lekkas. Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing*, 31(4):445–463, 2017.

[16] Marco Bibuli, Gabriele Bruzzone, Massimo Caccia, and Lionel Lapierre. Path-following algorithms and experiments for an unmanned surface vehicle. *Journal of Field Robotics*, 26(8):669–688, 2009.

[17] Wei Meng, Chen Guo, Yang Liu, Yang Yang, and Zhengling Lei. Global sliding mode based adaptive neural network path following control for underactuated surface vessels with uncertain dynamics. In *2012 Third International Conference on Intelligent Control and Information Processing*, pages 40–45. IEEE, 2012.

[18] Christian R Sonnenburg and Craig A Woolsey. Modeling, identification, and control of an unmanned surface vehicle. *Journal of Field Robotics*, 30(3):371–398, 2013.

[19] Cheng Liu, Daiyi Wang, Yuxi Zhang, and Xiannan Meng. Model predictive control for path following and roll stabilization of marine vessels based on neurodynamic optimization. *Ocean Engineering*, 217:107524, 2020.

[20] Jun Morimoto and Kenji Doya. Robust reinforcement learning. *Neural computation*, 17(2):335–359, 2005.

[21] Lixing Zhang, Lei Qiao, Jianliang Chen, and Weidong Zhang. Neural-network-based reinforcement learning control for path following of underactuated ships. In *2016 35th Chinese Control Conference (CCC)*, pages 5786–5791. IEEE, 2016.

[22] Joohyun Woo, Chanwoo Yu, and Nakwan Kim. Deep reinforcement learning-based controller for path following of an unmanned surface vehicle. *Ocean Engineering*, 183:155–166, 2019.

[23] Andreas B Martinsen and Anastasios M Lekkas. Straight-path following for underactuated marine vessels using deep reinforcement learning. *IFAC-PapersOnLine*, 51(29):329–334, 2018.

[24] Hironori Yasukawa and Y Yoshimura. Introduction of mmg standard method for ship maneuvering predictions. *Journal of Marine Science and Technology*, 20(1):37–52, 2015.

[25] Dongdong Mu, Guofeng Wang, Yunsheng Fan, Xiaojie Sun, and Bingbing Qiu. Modeling and identification for vector propulsion of an unmanned surface vehicle: Three degrees of freedom model and response model. *Sensors*, 18(6):1889, 2018.

[26] Jialun Liu, Robert Hekkenberg, Erik Rotteveel, and Hans Hopman. Hydrodynamic characteristics of multiple-rudder configurations. *Ships and Offshore Structures*, 12(6):818–836, 2017.

[27] Jialun Liu, Robert Hekkenberg, Frans Quadvlieg, Hans Hopman, and Bingqian Zhao. An integrated empirical manoeuvring model for inland vessels. *Ocean Engineering*, 137:287–308, 2017.

[28] Shuo Xie, Deshan Chen, Xiumin Chu, and Chenguang Liu. Identification of ship response model based on improved multi-innovation extended kalman filter. *J. Harbin Eng. Univ*, 39(2):282–289, 2018.

[29] Yin Jian-Chuan, Zou Zao-Jian, and Xu Feng. Parametric identification of abkowitz model for ship maneuvering motion by using partial least squares regression. *Journal of Offshore Mechanics and Arctic Engineering*, 137(3), 2015.

[30] H Yasukawa, M Zaky, I Yonemasu, and R Miyake. Effect of engine output on maneuverability of a vlcc in still water and adverse weather conditions. *Journal of Marine Science and Technology*, 22(3):574–586, 2017.

[31] Flavia Benetazzo, Gianluca Ippoliti, Sauro Longhi, and Paolo Raspa. Advanced control for fault-tolerant dynamic positioning of an offshore supply vessel. *Ocean Engineering*, 106:472–484, 2015.

[32] Cheng Liu, Jing Sun, and Zaojian Zou. Integrated line of sight and model predictive control for path following and roll motion control using rudder. *Journal of Ship Research*, 59(02):99–112, 2015.

[33] Chenguang Liu. *Motion control of unmanned surface vehicles based on model predictive control*. PhD thesis, Wuhan: Wuhan University of Technology, 2017.

[34] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[35] Yuanchang Liu, Richard Bucknall, and Xinyu Zhang. The fast marching method based intelligent navigation of an unmanned surface vehicle. *Ocean Engineering*, 142:363–376, 2017.

[36] Xin-ping Yan, Shu-wu Wang, Feng Ma, Yuan-chang Liu, and Jin Wang. A novel path planning approach for smart cargo ships based on anisotropic fast marching. *Expert Systems with Applications*, 159:113558, 2020.

[37] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

[38] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[39] Chen Chen, Xian-Qiao Chen, Feng Ma, Xiao-Jun Zeng, and Jin Wang. A knowledge-free path planning approach for smart ships based on reinforcement learning. *Ocean Engineering*, 189:106299, 2019.

[40] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[41] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[42] Weiye Wang, Feng Ma, and Jialun Liu. Course tracking control for smart ships based on a deep deterministic policy gradient-based algorithm. In *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, pages 1400–1404. IEEE, 2019.

[43] F Isdaryani, F Feriyonika, and R Ferdiansyah. Comparison of ziegler-nichols and cohen coon tuning method for magnetic levitation control system. In *Journal of Physics: Conference Series*, volume 1450, page 012033. IOP Publishing, 2020.

[44] Yuanchang Liu and Richard Bucknall. The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method. *Applied Ocean Research*, 59:327–344, 2016.