Ship collision avoidance based on deep deterministic policy gradients and beetle antenna search algorithm

Shuo Xie, Vittorio Garofano, Xiumin Chu, and Rudy R. Negenborn

***Abstract*—Continuous-action reinforcement learning methods (e.g., the deep deterministic policy gradient method (DDPG)) have attracted much attention in the ship collision avoidance area. To obtain potential flexible collision avoidance policies, the noise injection is widely used in the exploration of DDPG. In order to realize an adaptive exploration in DDPG for ship collision avoidance, an adaptive beetle antenna search-optimized DDPG (ABAS-DDPG) method is proposed in this study. The main idea of this method is to adapt the noise scale in the learning process simultaneously based on a beetle antenna search (BAS) algorithm, which utilizes the feedback from the critic in DDPG. The simulation results indicate that the proposed ABAS-DDPG method successfully adjusts the noise process automatically and obtains stabler and higher sum rewards than original DDPG in ship collision avoidance learning.**

***Index Terms*—Reinforcement learning, ship collision avoidance, deep deterministic policy gradients, beetle antenna search, noise scaling**

I. Introduction

SHIP collision avoidance has become a hot research topic in navigation safety [1], [2]. With the recent development of artificial intelligence, reinforcement learning (RL) methods [3], [4] have profound great effects on practical control issues [5], [6], [7], e.g., the ship collision avoidance problem [5]. Among plenty of RL methods, the policy gradient-based methods (e.g., deep deterministic policy gradient (DDPG) [8], asynchronous advantage actor-critic (A3C) [9], proximal policy optimization (PPO) [10], etc.) solve continuous action problems with a deterministic policy parameterized by neural networks, which attract more attention in ship collision avoidance recently. In this section, a brief survey on traditional ship collision avoidance methods and the related works on reinforcement learning-based ship collision avoidance methods are introduced. Besides, the related works on the main algorithms used in this study, i.e., the beetle antenna search (BAS) algorithm is also given.

S. Xie is with the Intelligent Transportation Systems Research Center, Wuhan University of Technology, 430063, Wuhan, Hubei Province, P.R. China.

X. Chu is with the Intelligent Transportation Systems Research Center, Wuhan University of Technology, 430063, Wuhan, Hubei Province, P.R. China.

V. Garofano and R. Negenborn are with the Department of Maritime and Transport Technology, Delft University of Technology,2628 CD Delft, The Netherlands.

A. A brief survey on traditional ship collision avoidance methods

Traditional ship collision avoidance methods can be briefly divided into two categories: path planning methods and optimization-based methods.

Path planning methods mainly include local path planning methods, e.g., artificial potential field (APF) [11] and global grid network-based methods, e.g., A* [12]. A* considers both the start position and the destination, which has global optimality but low efficiency in a large map [13]. Hierarchical Planning [14] is a commonly used approach to improve the efficiency of A*. APF uses artificial gravitational and repulsive field to model the navigation environment with a small computation [15], which can generate smoother paths than the gird-based methods (e.g., A*). Therefore, APF has been also applied for ship path planning [16] in past decades.

With the development of optimization algorithms, the quantitative optimization-based methods (e.g., fuzzy mathematics, neural networks, swarm intelligence and model predictive control, etc) have attracted more attention in ship collision avoidance area [17].

Fuzzy mathematics and neural networks adopt fuzzy membership functions and black-box approximated functions to model the collision risks, respectively. Classification and reasoning are the main issues in fuzzy mathematics. For classification, appropriate membership functions are required. In [18], the triangular and trapezoidal membership functions are used to represent different collision avoidance variables. In [19], the subjective feelings of the crew are also considered in fuzzy classification, which obtains better results. For fuzzy reasoning, the fuzzy maximum first (FMF) approach is widely used [20], and several representative achievements have been achieved [21]. Besides, the neural network is also an effective approach to model the uncertain factors in ship collision risk, which are commonly combined with fuzzy mathematics [22] and expert system (ES) [23] to realize collision avoidance.

Swarm intelligence optimization methods uses stochastic operator and swarm behaviors to achieve optimization, which have been also studied in ship collision avoidance. Ant colony optimization (ACO) [24] and particle swarm optimization (PSO) [25] are the most commonly used algorithms in ship collision avoidance, which can obtain good results with an appropriate fitness function considering the collision risk.

Recently, a typical model-based control method, i.e., model predictive control (MPC) [26], [1], [27], [28], [29], has been

also studied in ship collision avoidance area due to the ability of comprehensive considering the ship maneuverability model and the constraints. MPC has advantages of rolling optimization and state prediction, which can be extended to distributed MPC method for multi-ship collision avoidance problems [30], [1], [26], [31].

In summary, path planning and optimization-based methods have obtained rich achievements in ship collision avoidance. The similarity of existing traditional methods is to model an appropriate environment for searching or optimization (e.g., the grid map in A*, the objective function considering collision risks in MPC, etc.), which needs certain prior knowledge. In spite of this, existing traditional methods have difficulties in application in unknown environment, e.g., a limited perception caused by poor visibility, which needs further research.

B. Related works on ship collision avoidance based on reinforcement learning methods

At present, several reinforcement learning methods have been applied in ship collision avoidance, e.g., the Q-learning, Deep Q-network and DDPG, etc. By reasonable definitions of the state, action and reward function, the ship collision avoidance process can be regarded as a classical Markov decision process (MDP), of which the state transition probability is determined by the ship dynamics.

The conventional Q-learning method adopts a discrete Q-Table to map the state value and the action, which is capable of dealing with discrete state-action problems, e.g., the path planning for collision avoidance. In [32], the Q-learning algorithm is simply applied for path planning based on the kinematic model of the marine vehicle without considering obstacles. In [33], [34], the ship collision avoidance path planning problem is solved by Q-learning, which obtains more effective results than the traditional rapid-exploring random tree (RRT) and A* algorithms. By defining the discrete rudder actions and mapping the ship states to the grid map, a smoother path can be generated based on the established ship model and Q-learning.

Furthermore, for autonomous collision avoidance in dense regions, continuous states and actions are required to achieve flexible decision-making. Deep Q-network is firstly applied in ship collision avoidance with continuous state space by using function approximation. In [35], the distances measured by a set of fixed interval detection lines around the ship are defined as the states, and the discrete ship heading angle changes are defined as the actions in deep Q-network. This approach is validated in both numerical simulations [35] and model ship experiments [36]. In spite of this, high precision decision making with deep Q-network is still difficult due to the discrete actions. In [37], a so-called constrained deep Q-network is proposed to reduce the complexity of the action space by adding constrains based on international regulations for preventing collisions at sea (COLREGs), which also obtains good collision avoidance results. Besides, feature extraction in deep learning, e.g., the convolutional neural network (CNN), is an effective approach to train the value function in a high-dimensional state-action space. In [5], CNN is used in deep reinforcement learning to obtain a more reliable collision avoidance policy with a chain lumped state matrix including the perception information, the motion state and the ship's actions in a certain horizon.

To realize continuous action decisions and reduce the memory space for ship collision avoidance, deterministic policy gradient-based RL methods (e.g., DDPG [8], A3C [9], etc) adopt a network-approximated action policy that is regarded deterministic with respect to the current state, which performs better than Deep Q-network in continuous control problems [8]. In [38], the DDPG algorithm is applied for ship collision avoidance with the same state and action definitions as in [5], and the simulation results have indicated the effectiveness and advantages of DDPG algorithm in continuous ship behavior decision. In [39] the DDPG algorithm is applied with a simplified state-action space, of which the states and actions are defined by the relative parameters between own ship and target ship and the vertical distances away from the target course, respectively.

Actually, with the consideration of the ship dynamic model, the collision avoidance problem becomes very similar to the control problem, in which the RL methods have also been adopted recently [40], [41], [42]. Benefiting from no restrict requirements of the prior knowledge, RL methods can obtain effective collision avoidance policy for ships in different environments without large fine-tuning. However, due to the random exploration process for learning, the time cost of RL training becomes very high in complex environments. The exploration efficiency of RL needs to be improved for practical application.

C. Related works of the BAS algorithm

Stochastic optimization algorithms, such as ant colony optimization (ACO) algorithm [43], particle swarm optimization (PSO) algorithm [25], grey wolf optimization algorithm (GWO) [44], etc., are widely used in optimization tasks. Among them, the PSO algorithm is most commonly used due to its concise structure and fast convergence.

As a novel optimization algorithm similar to PSO, beetle antenna search (BAS) algorithm [45], and its swarm variation [46] are proposed based on the foraging behavior of beetles recently, which have simpler search strategies than PSO. The effectiveness of BAS-based algorithms have been validated in various optimization problems [47], [48], [49], [50], [2]. For fine-tuning problems, the BAS is capable of adjusting the hyper-parameters effectively, e.g., PID parameters [49], neural network parameters [48], etc. For direct optimization problems, the BAS has been proven to be an effective approach. In [51], the BAS algorithm is used to establish a portfolio model, which is combined with PSO. In [52], the 3-dimensional path planning problem is solved by the BAS algorithm and obtains a higher convergence rate than PSO. Due to the concise search strategy, the BSAS algorithm is considered to have great potential in solving optimization problems.

D. Motivations

Compared to traditional collision avoidance methods, the deterministic policy gradient-based reinforcement learning methods, e.g., DDPG, can obtain a continuous policy in uncertain environments by maximizing future rewards through interactions, which has the potentials under limited perception. In spite of this, a well-known drawback in existing reinforcement learning-based methods is the low exploring efficiency problem, especially the off-policy methods. Noise injection is a widely used approach, while it is difficult to obtain a stable policy with an inappropriate noise process. A reasonable noise adjustment strategy can not only obtain a stable learning process, but also eliminate the noise removal in parallel testing. Recent researches [53], [54] have shown that optimizing or learning an additional policy of the exploration process is an effective approach to improve the exploring performance. In [53], a meta-policy gradient is proposed by using another parametric network updated by the policy gradient for noise adaptation, which can scale the noise injection automatically.

The goal of this paper is to propose an efficient and adaptive DDPG based ship collision avoidance method. For adaptive exploration of DDPG, since the additional policy learning process requires more computation and brings additional problems like the gradient vanishing, alternative simper optimization techniques, e.g., the BAS algorithm, can be considered instead of the neural networks.

E. Contributions

In order to generate an adaptive exploration process for ship collision avoidance, an adaptive BAS optimized DDPG (ABAS-DDPG) algorithm is proposed by integrating a BAS optimizer into the actor-critic framework in DDPG. The main contribution of this paper is that the proposed ABAS-DDPG algorithm achieves adaptive exploration and better learning results by scaling the noise injection of the actor in DDPG algorithm [8], based on a beetle antenna search (BAS) algorithm, which is more suitable for direct application.

F. Outlines

The remainder of this article is organized as follows. In Section II, preliminaries including the widely used ship hydrodynamic model and collision risk model [56] are introduced. In Section III, the DDPG algorithm is applied to ship collision avoidance. In Section IV, the adaptive BAS-optimized DDPG framework is proposed for adaptive exploration. In Section V, ship collision avoidance simulations are conducted based on a model ship known as Tito-Neri tug ship [57] and a novel fully-actuated Delfia 1* ship developed by TU Delft [31]. In Section VI, conclusions and further research are presented.

II. Preliminaries

In the ship collision avoidance problem, a normalized indicator is needed to measure the risk of collisions between the encountering ships, i.e., the collision risk index (CRI). To calculate the CRI, a collision risk model [56] is widely applied to map the relative motion states, e.g., the distance of the closed point of approach (DCPA), the time of the closed point of approach (TCPA), to the CRI value. Then the risk model can be used to design the collision avoidance algorithm.

In this section, the mathematic models used for collision avoidance in this study are introduced, which include the ship hydrodynamic model and the ship collision risk model.

A. Ship hydrodynamic model

Abkowitz model [58] and MMG (math model group) model [59] have been commonly used for the modeling of ship motion. Both of them can be simplified to 3 degree-of-freedom (DOF) for surface ships as shown in Fig. 1(a). In Fig. 1(a), $O_o - x_o y_o$ is the inertial coordinate system of the vessel; $O - xy$ is the co-rotational coordinate system of the vessel; u, v and r are the velocities in surge (body-fixed x), sway (body-fixed y) and yaw directions, respectively; δ and ψ are the rudder and heading angle of the vessel, respectively; β is the drift angle. Then the ship hydrodynamic model can be denoted as the following 3-DOF form [59], [60]:

$$\begin{aligned} \dot{\boldsymbol{\eta}}\left(\boldsymbol{t}\right) &= \boldsymbol{R}\left(\psi\left(t\right)\right)\boldsymbol{\upsilon}\left(\boldsymbol{t}\right), \\ \boldsymbol{M}\dot{\boldsymbol{\upsilon}}\left(\boldsymbol{t}\right) &= \boldsymbol{f}\left(u\left(t\right), v\left(t\right), r\left(t\right), \delta\left(t\right), n\left(t\right)\right), \end{aligned} \tag{1}$$

where $\boldsymbol{\eta}\left(\boldsymbol{t}\right) = \left[\begin{array}{ccc} x\left(t\right) & y\left(t\right) & \psi\left(t\right) \end{array}\right]^{\mathrm{T}}$ and $\boldsymbol{\upsilon}\left(\boldsymbol{t}\right) = \left[\begin{array}{ccc} u\left(t\right) & v\left(t\right) & r\left(t\right) \end{array}\right]^{\mathrm{T}}$ are the position and velocity vectors of the ship at time t, respectively; $x\left(t\right)$ and $y\left(t\right)$ are the surge and sway positions, m; $\psi\left(t\right)$ is the heading, rad; $\delta\left(t\right)$ is the rudder angle, rad; $n\left(t\right)$ is the engine speed, r/min; $\boldsymbol{f}$ is the nonlinear lumped force and moment matrix of the ship with respect to $\boldsymbol{\upsilon}, \delta$ and n. $\boldsymbol{R}\left(\psi\right)$ is the rotation matrix between $\dot{\boldsymbol{\eta}}$ and $\boldsymbol{\upsilon}$. $\boldsymbol{M}$ is the inertia matrix of the ship:

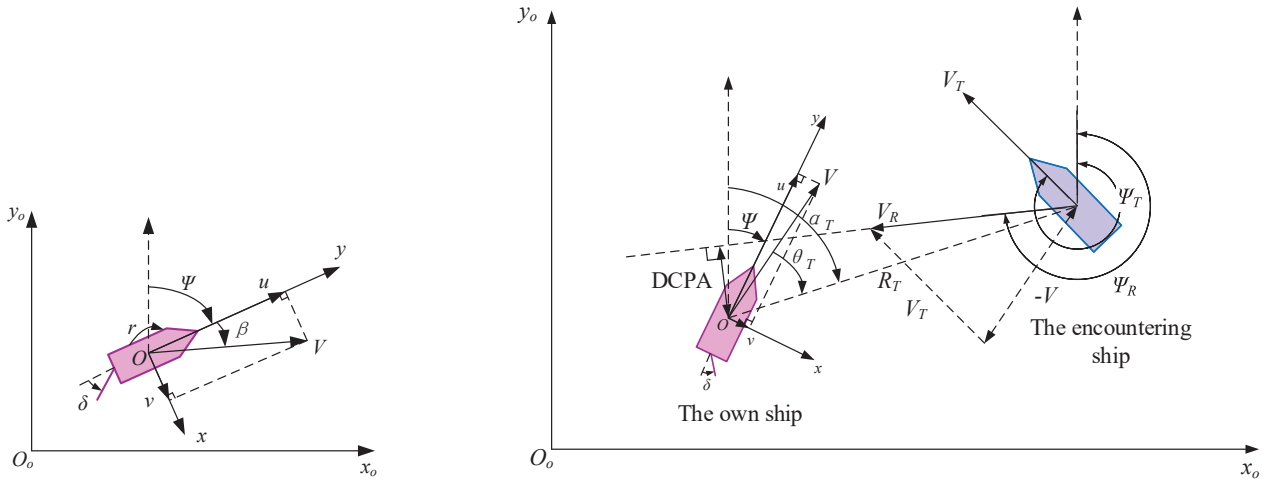
$$\begin{aligned} \boldsymbol{R}\left(\psi\left(t\right)\right) &= \left[\begin{array}{ccc} \sin\left(\psi\left(t\right)\right) & \cos\left(\psi\left(t\right)\right) & 0 \\ -\cos\left(\psi\left(t\right)\right) & \sin\left(\psi\left(t\right)\right) & 0 \\ 0 & 0 & 1 \end{array}\right], \\ \boldsymbol{M} &= \left[\begin{array}{ccc} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_G - Y_{\dot{r}} \\ 0 & mx_G - N_{\dot{v}} & I_z - N \end{array}\right], \end{aligned} \tag{2}$$

where m is the total mass of the vessel, kg; x_G is the longitudinal coordinate of the gravity center of the vessel in surge direction, m; I_Z is the moment of the inertia, kg·m^2; $X_{\dot{u}}, Y_{\dot{v}}, Y_{\dot{r}}, N_{\dot{v}}$ and $N_{\dot{r}}$ are the inertia coefficients.

B. Collision risk index model

In typical encounters, the CRI (collision risk index) between two ships can be evaluated based on the relative motion parameters, e.g., DCPA and TCPA. In this study, we consider only one ship as the control object, which is defined as the own ship. The other ship which causes the encounter situation with the own ship is defined as the encountering ship.

Assuming that the lumped state matrix of the own ship is $\boldsymbol{X} = [\boldsymbol{\eta}^{\mathrm{T}}, \boldsymbol{\upsilon}^{\mathrm{T}}]^{\mathrm{T}}$, and the state matrix of the encountering ship is $\boldsymbol{X_T} = [\boldsymbol{\eta}_T^{\mathrm{T}}, \boldsymbol{\upsilon}_T^{\mathrm{T}}]^{\mathrm{T}} = [x_T, y_T, \psi_T, u_T, v_T, r_T]^{\mathrm{T}}$. As shown in



(a) Ship motion coordinate system.

(b) Typical two-ship encounter.

Fig. 1. Ship motion coordinate system and typical encounter.

Fig. 1(b), the relative motion parameters can be obtained [56] as in (3):

$$\begin{aligned} DCPA &= R_T \sin\left(\psi_R - \alpha_T - \pi\right), \\ TCPA &= R_T \cos\left(\psi_R - \alpha_T - \pi\right)/V_R, \\ R_T &= \sqrt{\left(x_T - x\right)^2 + \left(y_T - y\right)^2}, \\ \theta_T &= \alpha_T - \psi \pm 2\pi, \end{aligned} \tag{3}$$

where R_T is the relative distance between two ships; ψ_R is the relative course direction of the obstacle ship; α_T is the true relative position direction of the obstacle ship, which can be obtained by (4); θ_T is converted from α_T in body-fixed coordinate system of the own ship.

In (4), $V_T = \sqrt{u_T^2 + v_T^2}$ and $V = \sqrt{u^2 + v^2}$ are the speeds of the obstacle ship and the own ship, respectively; V_R is the relative speed of the obstacle ship; v_{x_R} and v_{y_R} are the relative speed components of the obstacle ship on the X and Y axis, respectively. Then, to establish the CRI model, the fuzzy logic method [22] is adopted by using membership functions, which are general indicators of the degree of truth. The membership functions of DCPA, TCPA, R_T, θ_T and velocity ratio $K = V_T/V$ can be calculated [61], [22] as follows:

$$\begin{aligned} v_{x_R} &= u_T \sin(\psi_T) + v_T \cos(\psi_T) - (u \cdot \sin(\psi) + v \cos(\psi)), \\ v_{y_R} &= u_T \cos(\psi_T) - v_T \sin(\psi_T) - (u \cos(\psi) - v \sin(\psi)), \\ V_R &= \sqrt{{v_{x_R}}^2 + {v_{y_R}}^2}, \\ \psi_R &= \arctan\frac{v_{x_R}}{v_{y_R}} + \begin{cases} 0 & v_{x_R} \geq 0 \cup v_{y_R} \geq 0, \\ \pi & (v_{x_R} < 0 \cup v_{y_R} < 0) || (v_{x_R} \geq 0 \cup v_{y_R} < 0), \\ 2\pi & (v_{x_R} < 0 \cup v_{y_R} \geq 0), \end{cases} \\ \alpha_T &= \arctan\frac{(x_T - x)}{(y_T - y)} + \begin{cases} 0 & (x_T - x) \geq 0 \cup (y_T - y) \geq 0, \\ \pi & ((x_T - x) < 0 \cup (y_T - y) < 0) || ((x_T - x) \geq 0 \cup (y_T - y) < 0), \\ 2\pi & ((x_T - x) < 0 \cup (y_T - y) \geq 0), \end{cases} \end{aligned} \tag{4}$$

1. The membership of DCPA:

$$\begin{aligned} u_{DCPA} &= \begin{cases} 1 & |DCPA| \leq d_1, \\ \frac{1}{2} - \frac{1}{2} c_{DCPA} & d_1 < |DCPA| \leq d_2, \\ 0 & |DCPA| > d_2, \end{cases} \\ c_{DCPA} &= \sin\left[\frac{\pi\left(|DCPA| - (d_1 + d_2)/2\right)}{d_2 - d_1}\right], \end{aligned} \tag{5}$$

where $d_2 = 2d_1$ and d_1 is the closest safety distance of the two ships, which varies with θ_T as:

$$d_1 = \begin{cases} 1.1 - \frac{\theta_T}{180^\circ} \times 0.2 & 0^\circ \leq \theta_T < 112.5^\circ, \\ 1.0 - \frac{270^\circ - \theta_T}{180^\circ} \times 0.8 & 112.5^\circ \leq \theta_T < 247.5^\circ, \\ 1.1 - \frac{360^\circ - \theta_T}{180^\circ} \times 0.4 & 247.5^\circ \leq \theta_T < 360^\circ. \end{cases} \tag{6}$$

2. The membership of R_T :

$$u_{R_T} = \begin{cases} 1 & R_T \leq r_1, \\ \frac{1}{2} - \frac{1}{2}\sin\left[\frac{\pi}{r_2 - r_1}\left(R_T - \frac{r_1 + r_2}{2}\right)\right] & r_1 < R_T \leq r_2, \\ 0 & R_T > r_2, \end{cases} \tag{7}$$

where, r_1 and r_2 are the distance values of the last action (DLA) and Arena of the own ship, respectively. The Arena is proposed by David [62] to describe the area for which entering of a ship should trigger a collision avoidance action so as to avoid violating the actual domain [56]. The DLA indicates the closest distance of taking action by the own ship to avoid collision [61].

3. The membership of TCPA:

$$u_{TCPA} = \begin{cases} 1 & |TCPA| \le t_1, \\ \left(\frac{t_2 - |TCPA|}{t_2 - t_1}\right) & t_1 < |TCPA| \le t_2, \\ 0 & |TCPA| > t_2, \end{cases} \tag{8}$$

where, t_1 and t_2 represent the time limits for collision avoidance, which can be determined by r_1 and r_2:

$$\begin{aligned} t_1 &= \begin{cases} \frac{1}{V_R}\sqrt{{r_1}^2 - DCPA^2} & DCPA \le r_1, \\ \frac{1}{V_R}\left(r_1 - DCPA\right) & DCPA > r_1, \end{cases} \\ t_2 &= \begin{cases} \frac{1}{V_R}\sqrt{{r_2}^2 - DCPA^2} & DCPA \le r_2, \\ \frac{1}{V_R}\left(r_2 - DCPA\right) & DCPA > r_2. \end{cases} \end{aligned} \tag{9}$$

4. The membership of θ_T :

$$u_{\theta_T} = \frac{1}{2}\left[\cos\left(\theta_T - 19\right) - \frac{5}{17} + \sqrt{\frac{440}{289} + \cos^2\left(\theta_T - 19\right)}\right]. \tag{10}$$

5. The membership of K :

$$u_K = \frac{1}{1 + \frac{2}{K\sqrt{K^2+1+2K\sin C}}}, \tag{11}$$

where $C \in [0, 180]$ is a constant coefficient.

Therefore, the following CRI model is established:

$$\begin{aligned} &f_{CRI} = \boldsymbol{\lambda}_{CRI}\boldsymbol{u}_{CRI}, \\ &\boldsymbol{\lambda}_{CRI} = \begin{bmatrix} \lambda_{DCPA} & \lambda_{TCPA} & \lambda_{R_T} & \lambda_{\theta_T} & \lambda_K \end{bmatrix}, \\ &\boldsymbol{u}_{CRI} = \begin{bmatrix} u_{DCPA} & u_{TCPA} & u_{R_T} & u_{\theta_T} & u_K \end{bmatrix}^{\mathrm{T}}, \end{aligned} \tag{12}$$

where λ_{DCPA} , λ_{TCPA} ,λ_{R_T}, λ_{θ_T} and λ_K are the set weights of u_{DCPA} , u_{TCPA} ,u_{R_T}, u_{θ_T} and u_K, respectively. It can be seen that the CRI model outputs the CRI values with respect to the relative ship states, which can be used to design the state-action space and reward function in reinforcement learning.

III. DDPG BASED SHIP COLLISION AVOIDANCE METHOD

Reinforcement learning attracts much attention in ship collision avoidance recently. The DDPG algorithm is capable of learning good policies in unknown environments, e.g., the limited perception environment caused by low visibility or sensor problems.

In this section, the DDPG algorithm is applied to ship collision avoidance problem with limited perception by regarding the collision avoidance process as a typical Markov decision-making process (MDP), in which the definitions of the state-action space and reward function are the key issues.

A. State and action design

Reinforcement learning tasks are usually described by discrete finite Markov decision processes (MDP), in which the time space is divided in discrete steps. At each discrete time step t, the own ship observes the states of the encountering ship $\boldsymbol{s}_t \in \mathbb{R}^{N_s}$ in the perception range and takes a rudder action $a_t \in \mathbb{R}$ based on a collision avoidance policy π. Then the own ship receives a new observation $\boldsymbol{s}_{t+1} \in \mathbb{R}^{N_s}$ and a reward $r_t \in \mathbb{R}$ of the action, where N_s are the dimensions of the state space. The final goal of the reinforcement learning method in ship collision avoidance is to learn the optimal policy π which generate rudder actions with the maximum future rewards.

1) Definition of the action a_t

Since the optimal rudder action with repect to different encountering ships may be different, the multi-ship collision avoidance process is divided into a set of sub-MDP processes with each encountering ship to learn a general collision avoidance policy. At each time step t, the own ship takes a combined rudder action $a = \delta$ to avoid collisions with all encountering ship, while receives different reward r^i with respect to the ith encountering ship from the artificial reward function. Therefore, different actions are generated in sub-MDP processes, the final combined action is obtained by weighting the generated actions based on the collision risks to prioritize the most dangerous encountering ship:

$$a_t = \frac{f_{CRI}^i}{\sum_{i=1}^{n} f_{CRI}^i}\pi\left({\boldsymbol{s}^i}_t\right), \tag{13}$$

where n is the number of the encountering ships, a_t is the final decision rudder result for collision avoidance, f_{CRI}^i is the collision risk between the own ship and the ith encountering ship, ${\boldsymbol{s}^i}_t$ is the observed state of the ith encountering ship, $\pi\left({\boldsymbol{s}^i}_t\right)$ is the collision avoidance policy. After the own ship takes the action a_t, the states with different encountering ships are updated for learning.

2) Definition of the state ${\boldsymbol{s}^i}_t$

In typical two-ship encounters, a set of the relative collision states (i.e., DCPA, TCPA, relative distance R_T, relative position direction θ_T, relative heading direction C_T, relative speed ratio K) between the own ship and the encountering ship can represent multiple sets of different original motion states ($\boldsymbol{X}$, $\boldsymbol{X}_T$) [2]. Therefore, the relative motion states with respect to the ith encountering ship are defined as the state ${\boldsymbol{s}^i}_t$ in the sub-MDP:

$${\boldsymbol{s}^i}_t = \left[DCPA_t^i, TCPA_t^i, R_{Tt}^i, \theta_{Tt}^i, f_{CRIt}^i, C_{Tt}^i, K_t^i\right]^{\mathrm{T}}. \tag{14}$$

B. Reward design

A reasonable reward function considering safety and economy is very important in reinforcement learning for ship collision avoidance. An immediate continuous reward function is considered to be more suitable than a discrete reward function because of the large inertia and continuity of ship motion [38].

1) safety

The encountering ships and the navigation boundary should be both considered for navigation safety.

With respect to the encountering ships, the same avoidance action may lead to different collision risks with different encountering ships at the same time. Therefore, the collision risks with all the detected ships are considered for safety reward for the encountering ships. The safety reward can be defined as the sum of squares of collision risks, which can automatically increase the weight of the reward for higher risks, so as to give priority to the most dangerous encountering ship:

$$r_t^{es} = -\sum_{i=1}^{n} {f^i_{CRI}}_t{}^2, \tag{15}$$

where r_t^{es} is the reward for safety with encountering ships; n is the number of the detected encountering ships. Besides, without losing generality, the static obstacle can be also regarded as a special dynamic obstacle from the perspective of the own ship, of which the speed is zero and the r_1 and r_2 in (7) are set based on the scale of the obstacle for risk calculation.

With respect to the navigation boundary, assuming that the boundary can be represented by a binary function $f_{nb}(x_t, y_t)$, where $f_{nb}(x_t, y_t) = 0$ represent that the position $[x_t, y_t]$ at time t is inside the boundary and $f_{nb}(x_t, y_t) = 1$ represent that the position $[x_t, y_t]$ is outside the boundary. Therefore, the safety reward for the navigation boundary is obtained as:

$$r_t^{nb} = -f_{nb}(x_t, y_t), \tag{16}$$

where $[x_t, y_t]$ is the position of the own ship. Then, the final safety reward is $r_t^s = r_t^{es} + r_t^{nb}$.

2) economy

During the voyage, steering with a large rudder angle will reduce the ship surge speed and increase the time for resumption. Generally, the squares of the rudder angle, the sway and yaw velocities are used to construct the economic reward with respect to the energy loss caused by large rudder steering:

$$r_t^{\delta} = -\left(\delta_t^2 + v_t^2 + r_t^2\right). \tag{17}$$

In addition, to make the own ship moves towards the destination as soon as possible, the distance to the destination is used to construct part of the economic reward:

$$r_t^{dis} = -\frac{(x_t - x_{dst})^2 + (y_t - y_{dst})^2}{(x_{srt} - x_{dst})^2 + (y_{srt} - y_{dst})^2}, \tag{18}$$

where $[x_{srt}, y_{srt}]$ and $[x_{dst}, y_{dst}]$ are the starting point and destination, respectively.

Then, the economy reward is obtained as $r_t^e = r_t^{\delta} + r_t^{dis}$, and the final reward function for collision avoidance r_t can be obtained by weighting the safety reward r_t^s and economy reward r_t^e:

$$r_t = \lambda_s r_t^s + \lambda_e r_t^e, \tag{19}$$

where λ_s and λ_e are the setting weights for r_t^s and r_t^e, respectively.

C. DDPG reinforcement learning

After designing the state-action space and the reward function, the DDPG algorithm can be applied for learning the optimal collision avoidance policy $\pi\left({s^i}_t\right)$. In reinforcement learning, the expected reward of the rudder action a_t with state ${s^i}_t$ is described as the well known state-action value function $Q\left({s^i}_t, \boldsymbol{a}_t\right)$ as follows:

$$Q\left({s^i}_t, \boldsymbol{a}_t\right) = \mathbb{E}_{r, \boldsymbol{s}\sim\boldsymbol{E}, \boldsymbol{a}\sim\pi}\left[\left(\sum_{j=t}^{T}\gamma^{j-t} r_j\right)\middle|\left({s^i}_t, \boldsymbol{a}_t\right)\right], \tag{20}$$

where $\gamma \in [0, 1]$ is a discount factor, $\boldsymbol{E}$ is the set of all expected future states under the rudder action $\boldsymbol{a}_t$, $\sum_{i=t}^{T}\gamma^{i-t} r_i$ represents the discounted sum of the future rewards. Then, the Bellman equation is used to calculate the state-action value function as:

$$\begin{aligned} Q\left({s^i}_t, \boldsymbol{a}_t\right) = \mathbb{E}_{r_t, \boldsymbol{s}_{t+1}\sim\boldsymbol{E}, \boldsymbol{a}_t\sim\pi}\big[\big(r_t\left({s^i}_t, \boldsymbol{a}_t\right) \\ +\gamma\mathbb{E}_{\boldsymbol{a}_{t+1}\sim\pi}\left[Q\left(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1}\right)\right]\big)\big]. \end{aligned} \tag{21}$$

Referring to DDPG [8], a deterministic collision avoidance policy $\boldsymbol{a}_t = \pi({s^i}_t)$ parametrized by $\boldsymbol{\theta}_\pi$ is adopted as an actor to generate deterministic rudder actions which maximize the state-action value. Therefore, the inner expectation can be obtained as:

$$\begin{aligned} Q\left({s^i}_t, \boldsymbol{a}_t\right) = \mathbb{E}_{r_t, \boldsymbol{s}_{t+1}\sim\boldsymbol{E}, \boldsymbol{a}_t\sim\pi}\big[\big(r_t\left({s^i}_t, \boldsymbol{a}_t\right) \\ +\gamma Q\left(\boldsymbol{s}_{t+1}, \pi\left(\boldsymbol{s}_{t+1}\right)\right)\big)\big]. \end{aligned} \tag{22}$$

With the deterministic collision avoidance policy, a neural network parametrized by $\boldsymbol{\theta}_Q$ is used as a critic to approximate the state-action value function as in deep Q-learning [3], which is updated by the following gradient descent:

$$\begin{aligned} L\left(\boldsymbol{\theta}^Q\right) = \mathbb{E}_{{\boldsymbol{s}^i}_t\sim\boldsymbol{B}}\left[\left(Q\left({s^i}_t, a_t\middle|\boldsymbol{\theta}^Q\right) - y_t\right)^2\right], \\ \boldsymbol{\theta}^Q \leftarrow \boldsymbol{\theta}^Q + \eta^Q \nabla_{\boldsymbol{\theta}^Q} L\left(\boldsymbol{\theta}^Q\right), \end{aligned} \tag{23}$$

where $\boldsymbol{B}$ denotes the replay buffer B, $y_t = r_t + \gamma Q\left(\boldsymbol{s}_{t+1}, \pi\left(\boldsymbol{s}_{t+1}\right)\middle|\boldsymbol{\theta}^Q\right)$ is the actual calculated state-action value, η^Q is the learning rate of the critic.

Synchronously, the actor π is updated by $\max_{\boldsymbol{\theta}^\pi} J\left(\boldsymbol{\theta}^\pi\right) = Q\left({s^i}_t, \pi\left({s^i}_t\right)\middle|\boldsymbol{\theta}^\pi\right)$. To solve this maximizing problem, a policy gradient is applied with the chain rule to update the parameter $\boldsymbol{\theta}^\pi$ as:

$$\begin{aligned} \nabla_{\boldsymbol{\theta}^\pi} J\left(\boldsymbol{\theta}^\pi\right) = \nabla_{\boldsymbol{\theta}^\pi}\mathbb{E}_{{\boldsymbol{s}^i}_t\sim\boldsymbol{B}}\left[\nabla_{\boldsymbol{a}} Q\left(\boldsymbol{s}, \boldsymbol{a}\middle|\boldsymbol{\theta}^Q\right)\middle|_{\boldsymbol{s}={\boldsymbol{s}^i}_t, a=\pi({\boldsymbol{s}^i}_t)}\right. \\ \left.\nabla_{\boldsymbol{\theta}^\pi}\pi\left(\boldsymbol{s}\middle|\boldsymbol{\theta}^\pi\right)\middle|_{\boldsymbol{s}={\boldsymbol{s}^i}_t}\right], \\ \boldsymbol{\theta}^\pi \leftarrow \boldsymbol{\theta}^\pi + \eta^\pi \nabla_{\boldsymbol{\theta}^\pi} J\left(\boldsymbol{\theta}^\pi\right), \end{aligned} \tag{24}$$

where η^π is the learning rate of the actor, $\nabla_{\boldsymbol{a}} Q\left(\boldsymbol{s}, \boldsymbol{a}\middle|\boldsymbol{\theta}^Q\right)$ and $\nabla_{\boldsymbol{\theta}^\pi}\pi\left(\boldsymbol{s}\middle|\boldsymbol{\theta}^\pi\right)$ are the gradients of the state-action value function and the deterministic policy with respects to the rudder action and the parameter $\boldsymbol{\theta}^\pi$, respectively. Besides, the experience buffer reply technique in deep Q-learning is also used to update the actor $\boldsymbol{\theta}^\pi$ and critic $\boldsymbol{\theta}^Q$ in DDPG.

Generally speaking, the main idea of the DDPG based collision avoidance method is to use deep neural networks to learn the optimal collision avoidance policy (i.e., the actor)

through interactions with the environment based on policy gradient, which can generate rudder actions that maximize the discounted future rewards.

IV. ABAS-DDPG FOR ADAPTIVE EXPLORATION

Although the DDPG algorithm can obtain an optimal collision avoidance policy, the independent exploration efficiency of DDPG without noises would be very low in complex environment due to the exploration-exploitation dilemma. The baseline in [8] uses a stable OU noise process for exploration in DDPG and learns good policies in many tasks in OpenAI environments.

In spite of this, the action noise is also considered to vary over time since the exploration requirement changes in the learning process. Reasonable adaptive scaling of the noise becomes the key challenge in the exploration, of which the purpose is to prevent premature converge to a local optimum under the premise of stable learning. In order to realize adaptive exploration in DDPG, a simple beetle antenna search algorithm is used to scale the action noise adaptively in this section.

A. The optimization strategy for DDPG

A common practice in exploration is injecting an uncorrelated action noise (e.g., Gaussian process) or a correlated action noise (e.g., Ornstein-Uhlenbeck (OU) process) [63] selected by the deterministic actor policy:

$$\boldsymbol{a}_t = \pi\left(\boldsymbol{s}^i{}_t\right) + \mathcal{N}\left(0, \sigma^2 \boldsymbol{I}\right), \tag{25}$$

where $\mathcal{N}$ is the injected noise and σ is the scale parameter with respect to the standard deviation. Since DDPG is an off-policy learning method, the noise $\mathcal{N}$ can be treated independently without much influences on the learning process. Therefore, learning an exploration policy additionally is an effective approach to obtain an adaptive noise process [53]. To reduce the additional computations, we attempt to realize the adaptive scaling of the noise by establishing an simpler optimization strategy instead of the parametric neural networks in [53].

Generally, a noise attenuation technique is adopted as:

$$\begin{aligned} \hat{\sigma}_{t+1} &= \eta \hat{\sigma}_t, \\ \sigma_t &= \hat{\sigma}_t + \sigma_{\min}, \end{aligned} \tag{26}$$

where $\hat{\sigma}_t$ is the varying term in σ at t step, $\sigma_{\min}$ is the set minimum σ for exploration, η is the attenuation factor for convergence.

Note that the noise scale is considered to be enlarged for comprehensive exploration when the actor $\boldsymbol{a}_t$ has not yet learned a good policy; On the contrary, the noise scale can be attenuated when the actor policy $\boldsymbol{a}_t$ can already output actions which can obtain the current maximum reward. Since the role of the critic in DDPG is to approximate the Q-function, the performance of the actor's output actions can be judged according to the output of the critic. Therefore, the following optimization-based strategy is proposed to make real-time noise scaling:

$$\begin{aligned} &\eta_t = \eta_{\min} + \left(\eta_{\max} - \eta_{\min}\right) \operatorname{sgn}\left(Q\left(s_t, \boldsymbol{a}_t^o\right) - Q\left(s_t, \pi\left(s_t\right)\right)\right), \\ &\boldsymbol{a}_t^o = \arg\max Q\left(s_t, \boldsymbol{a}_t\right), \\ &subject\ to,\ \boldsymbol{a}_{\min} \le \boldsymbol{a}_t \le \boldsymbol{a}_{\max}, \end{aligned} \tag{27}$$

where $\boldsymbol{a}_{\min}$ and $\boldsymbol{a}_{\max}$ are the setting minimum and maximum constraints for the actions, $0 < \eta_{\min} < 1$ and $\eta_{\max} > 1$ are two scaling factors which are set for reducing and increasing the scale of the noise based on the Q-values of the current action output $\pi\left(s_t\right)$ and the optimal action $\boldsymbol{a}_t^o$.

By solving the optimization problem in (27), the adaptive scaling of the noise can be realized by a finite n-step optimization at each learning step, i.e., solving (27) in n iterations starting from the current action output $\pi\left(s_t\right)$. If a better action $\boldsymbol{a}_t^o$ can be searched with larger Q-value, the attenuation will be $\eta_t = \eta_{\max}$ to increase the noise scale, otherwise the policy of the actor is considered good enough and the attenuation will be $\eta_t = \eta_{\min}$ to reduce the noise scale.

B. Adaptive BAS-optimized DDPG

In order to solve the optimization problem (27) in real-time, we integrate a beetle antenna search (BAS) optimization algorithm [45] for adaptive exploration in this study, i.e., the adaptive BAS-optimized DDPG (ABAS-DDPG) algorithm. The main idea of ABAS-DDPG is to use a beetle represented by two antennas and a centroid for optimization, and the centroid of the beetle is updated by moving in the direction of the antenna which obtains better fitness than the opposite one.

For (27), the centroid and antennas of the beetle are defined as the potential actions $\hat{\boldsymbol{a}}, \hat{\boldsymbol{a}}_l$ and $\hat{\boldsymbol{a}}_r$. The fitness with respect to an action $\hat{\boldsymbol{a}}$ is defined as the opposite of the state-action value $f_t\left(\hat{\boldsymbol{a}}\right) = -Q\left(s_t, \hat{\boldsymbol{a}}\right)$. Therefore, the updating strategy of the original BAS is denoted as:

$$\begin{aligned} \hat{\boldsymbol{a}}_{lk} &= \hat{\boldsymbol{a}}_k + l_k \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \\ \hat{\boldsymbol{a}}_{rk} &= \hat{\boldsymbol{a}}_k - l_k \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \\ \hat{\boldsymbol{a}}_{k+1} &= \hat{\boldsymbol{a}}_k - c l_k \operatorname{sgn}\left(f_t\left(\hat{\boldsymbol{a}}_{lk}\right) - f_t\left(\hat{\boldsymbol{a}}_{rk}\right)\right) \\ &= \hat{\boldsymbol{a}}_k - c l_k \operatorname{sgn}\left(Q\left(\boldsymbol{s}^i{}_t, \hat{\boldsymbol{a}}_{rk}\right) - Q\left(\boldsymbol{s}^i{}_t, \hat{\boldsymbol{a}}_{lk}\right)\right), \end{aligned} \tag{28}$$

where $\hat{\boldsymbol{a}}_{lk}$, $\hat{\boldsymbol{a}}_{rk}$ and $\hat{\boldsymbol{a}}_k$ are actions of the left, right antennas and the centroid of the beetle at k iteration respectively, l_k is the exploring step of the antennas at k iteration, c is a setting constant value which represents the ratio between the step of the beetle and the exploration scale l_k, $\operatorname{sgn}(\cdot)$ is a sign-function. At each time step t, the beetle generates two actions $\hat{\boldsymbol{a}}_{lk}$ and $\hat{\boldsymbol{a}}_{rk}$ by the left and right antennas, and updates the centroid from $\hat{\boldsymbol{a}}_0 = \boldsymbol{a}_t$ to $\hat{\boldsymbol{a}}_n$ in n iterations. The optimal action $\hat{\boldsymbol{a}}_t^o$ is updated based on the following greedy strategy:

$$\hat{\boldsymbol{a}}_t^o = \hat{\boldsymbol{a}}_k, f_t\left(\hat{\boldsymbol{a}}_t^o\right) = f_t\left(\hat{\boldsymbol{a}}_k\right)\ \ if\left\{f_t\left(\hat{\boldsymbol{a}}_k\right) < f_t\left(\hat{\boldsymbol{a}}_t^o\right)\right\}, \tag{29}$$

and the exploration scale l_k is updated by the following attenuation strategy [64]:

$$\begin{aligned} l_{k+1} &= \eta_{BAS} l_k \quad if\left\{f_t\left(\hat{\boldsymbol{a}}_k\right) \ge f_t\left(\hat{\boldsymbol{a}}_t^o\right)\right\}, \\ l_{k+1} &= l_k \qquad otherwise, \end{aligned} \tag{30}$$

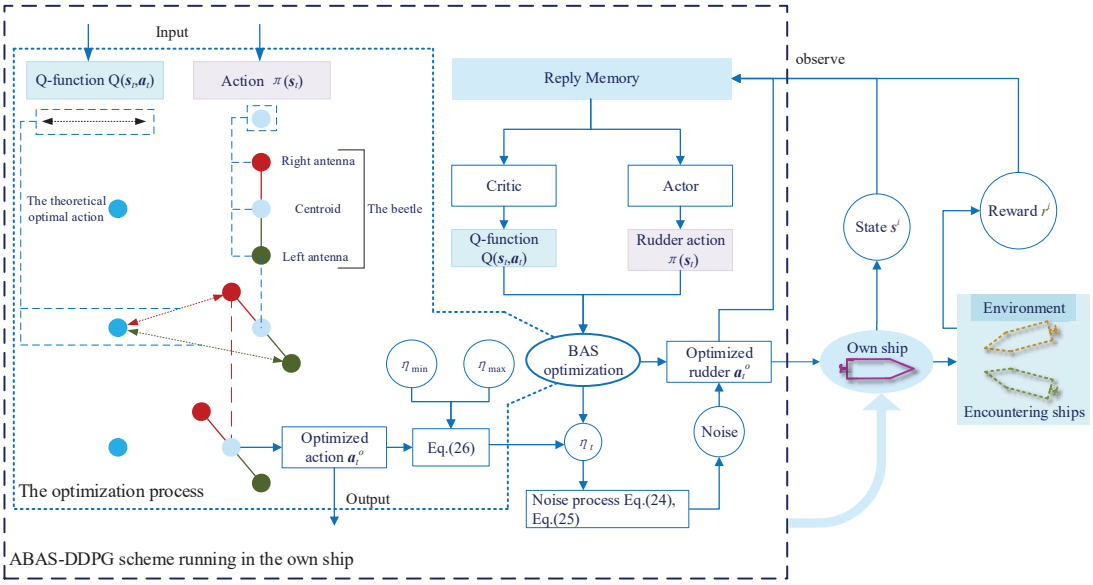


Fig. 2. The ABAS-DDPG flow chart.

where $0 < \eta_{BAS} < 1$ is the exploration attenuation factor of the beetle. After n iterations, the final optimal action is adopted as $\hat{\boldsymbol{a}}_t^o = \boldsymbol{a}_t^o$ in (27). Fig. 2 shows a flow chart of the proposed ABAS-DDPG scheme with the original BAS in 2-DOF searching space for example. As can be seen that the centroid of the beetle becomes closer to the theoretical optimal action following the better antenna, which shows the optimization process of BAS for DDPG.

Note that the rudder actions of the ship are bounded by a continuous cube $[\boldsymbol{a}_{\min}, \boldsymbol{a}_{\max}]$, thus the initial exploring step of original BAS l_0 is set based on the length of the search space [64] as $l_0 = \|\boldsymbol{a}_{\max} - \boldsymbol{a}_{\min}\|$ for global optimality. In the initial learning stage, the large exploring step may lead to the oscillation of the centroid since the historical trajectories of two antennas are abandoned after exploring in each iteration. To defect this problem, a historical optimum-based strategy [65] in (31) is used to exploit the historical optimums of two antennas instead of the original strategy in (28):

$$\begin{cases} \hat{\boldsymbol{a}}_{lk} = \hat{\boldsymbol{a}}_{lk}^o + l_k \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \\ \hat{\boldsymbol{a}}_{rk} = \hat{\boldsymbol{a}}_{rk}^o - l_k \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \\ \hat{\boldsymbol{a}}_{k+1} = \hat{\boldsymbol{a}}_k + r_d c_l \left(\hat{\boldsymbol{a}}_{lk}^o - \hat{\boldsymbol{a}}_k\right) + r_d c_r \left(\hat{\boldsymbol{a}}_{rk}^o - \hat{\boldsymbol{a}}_k\right), \end{cases} \tag{31}$$

where $\hat{\boldsymbol{a}}_{lk}^o$ and $\hat{\boldsymbol{a}}_{rk}^o$ are the historical optimums of two antennas at iteration k, $r_d \in [0,1]$ is a random value, c_l and c_r are two learning factors which generate movements from the current centroid $\hat{\boldsymbol{a}}_k$ to the left and right antenna historical optimums, respectively. Particularly, we consider that the movement to the antenna with better historical fitness should be larger than that to the other antenna in probability. Then, we design the learning factors based on the historical optimums as:

$$\begin{aligned} c_l &= \frac{e^{Q\left(s^i{}_t, \hat{\boldsymbol{a}}_{lk}^o\right)}}{e^{Q\left(s^i{}_t, \hat{\boldsymbol{a}}_{lk}^o\right)} + e^{Q\left(s^i{}_t, \hat{\boldsymbol{a}}_{rk}^o\right)}}, \\ c_r &= 1 - c_l. \end{aligned} \tag{32}$$

Since the fitness function in BAS is defined by the approximate Q-value function of the critic instead of the actual rewards, the n-iterations optimization can be carried out in the current step without repeated interactions with the environments. After the optimization is finished at each step, the optimized action $\boldsymbol{a}_t^o$ is used instead of the original action output of the actor $\pi(\boldsymbol{s}_t)$ for better exploration.

V. Case Studies

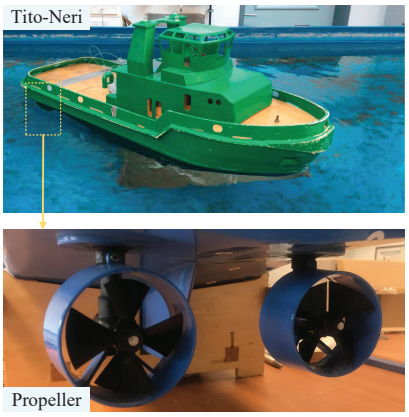
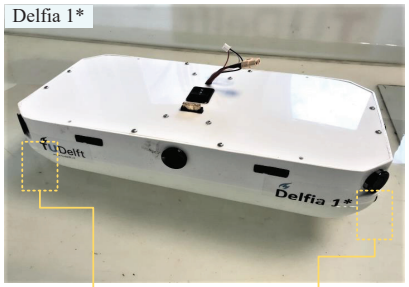
In this section, multi-ship collision avoidance simulations based on different ship models are conducted to verify the effectiveness of the ABAS-DDPG algorithm. Throughout all the experiments, the proposed method and the original DDPG are running on a computer with a 3.2 GHz 4 core CPU and 8 GB of RAM. The encountering ships are labeled as ES and the own ship is labeled as OS in the following figures. To verify the effectiveness of the BAS optimization, we conduct the comparisons between the proposed ABAS-DDPG and the original DDPG.

The remainder of this section is organized as follows. In Section V-A, the scenarios and two ship models are introduced. In Section V-B and Section V-C, two different ship models are used for reinforcement learning to verify the proposed ABAS-DDPG.

A. Simulation scenario

Ship collision avoidance simulations are conducted in the scenario inbound an experimental pool in Delft University of Technology (52°00'08.1"N, 4°22'17.2"E) as shown in Fig. 3. The origin $[0,0]$ represents the GPS position (52°00'07.6499739"N, 4°22'16.4941528"E) and the relative coordination is transferred from GPS position by GAUSS projection. A navigation boundary is set as $x, y \in [0, 20]$ in this scenario.

TABLE I
THE PARAMETERS OF DELFIA 1* SHIP AND TITO-NERI TUG SHIP.

Tito Neri		Delfia 1*
		
		 
Scale/(-)	1:30	1:16
Ship length/(m)	0.97	0.38
Moulded width/(m)	0.3	0.185
Moulded depth/(m)	0.18	0.07
Mass of the ship/(kg)	16.9	3.345
Inertia moment/(kg.m2)	0.51	0.031

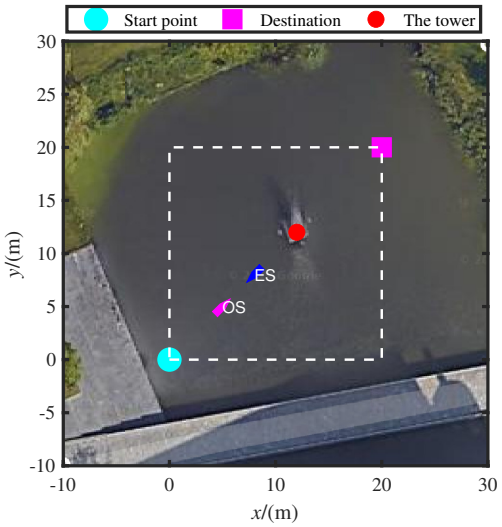


Fig. 3. The simulation scenario.

The goal of the own ship (OS) is to avoid collisions with the encountering ships (ES) and get close to the destination under the boundary constraint. In addition, the tower in the experimental pool is considered as a static obstacle. Since DDPG is a model-free learning algorithm, we consider two ships with different dynamics, i.e., a model ship known as Tito-Neri tug ship [57] and a novel fully-actuated Delfia 1* ship developed by TU Delft [31], to fully verify the learning performance of the proposed method. The basic parameters of Delfia 1* ship and Tito-Neri tug ship are shown in Table. I. More hydrodynamic parameters are given in [57], [31].

B. Case study 1: Collision avoidance learning of Tito-Neri ship

1) Set up: The original DDPG baseline in [8] with OU noise $\mathcal{N}\left(0, 0.2^2\boldsymbol{I}\right)$ is used for comparisons. Referring to [66], the two scaling factors in ABAS-DDPG are set as:

$$\eta_{\min} = \alpha^{\frac{1}{n_E}}, \eta_{\max} = \frac{1}{\alpha}^{\frac{1}{\beta n_E}}, \tag{33}$$

where $0 < \alpha < 1$ and $\frac{1}{\alpha} > 1$ are two constants which represent the expected scaling results of $\eta_{\min}$ and $\eta_{\max}$ after continuous n_E and βn_E steps, respectively. $\beta > 1$ is the ratio between the steps required by $\eta_{\min}$ and $\eta_{\max}$ to obtain the same scaling effect. Then, the hyper-parameters of DDPG and BAS, e.g., the learning rates and discount factors, are set as Table II.

To ensure collisions if no measures are taken, the initial state of different encountering ships are set to form the multi-ship encounter situation as follows:

$$\begin{aligned} x^i_{T0} &= x_0 + R^i\sin(\psi_0) + R^i\sin(\psi_0+\theta^i),\\ y^i_{T0} &= y_0 + R^i\cos(\psi_0) + R^i\cos(\psi_0+\theta^i),\\ \psi^i_{T0} &= \psi_0 + \theta^i + \pi,\\ u^i_{T0} &= u_0 = \hat{U}, v^i_{T0} = v_0 = 0, r^i_{T0} = r_0 = 0, \end{aligned} \tag{34}$$

where $\hat{U}$ is the setting average speed of the ship, $[x^i_{T0}, y^i_{T0}, \psi^i_{T0}, u^i_{T0}, v^i_{T0}, r^i_{T0}]$ is the initial state of the ith obstacle ship, R^i is the set distance before the collision between the own ship and the ith obstacle ship, and θ^i is the set collision angle of the ith obstacle ship. By setting different R^i and θ^i, the encountering ships will have collisions with the own ship in sequences if no measures are taken. In this study, A scenario including 3 encountering ships (ES) and the static tower with different R^i and θ^i are set in Table III for collision avoidance. The initial state of the own ship (OS) is set as $[0, 0, 45, \hat{U}, 0, 0]$, and the destination is set as $[x_d, y_d] = [20, 20]$.

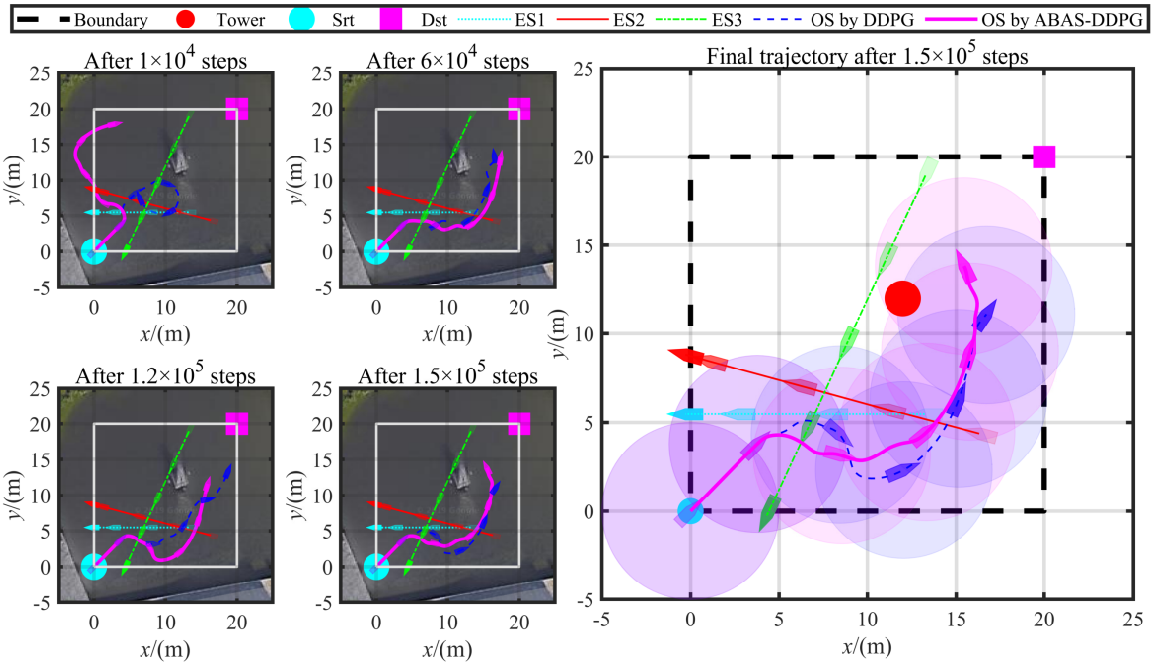


Fig. 4. The learning results of DDPG and ABAS-DDPG of Tito-Neri.

TABLE II
PARAMETERS OF DDPG AND BAS.

Paremeters	DDPG								BAS			
	η^{π}	η^{Q}	n_{hidden}	N_B	n_B	γ	α	β	n_E	c	λ	η_l
Value	10^{-3}	10^{-3}	64	10^4	32	0.9	0.1	10	1.5×10^4	1	1.1	0.9
Significance	Learning rates of the actor and critic		Nerual numbers in hidden layer	Experience buffer size	Batch size	Discount factor	Scaling parameters			Constant values		Attenuation factor

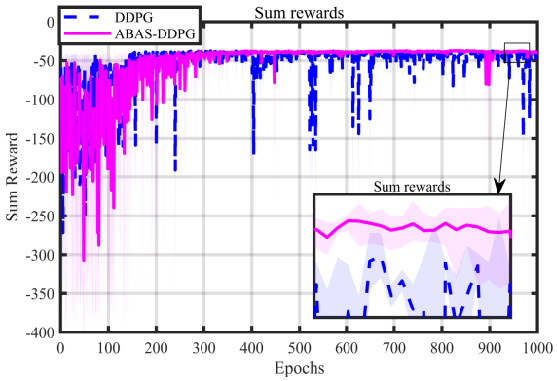


Fig. 5. The mean sum rewards of the proposed ABAS-DDPG and original DDPG of Tito-Neri.

TABLE III
INITIAL R^i AND θ^i FOR DIFFERENT ENCOUNTERING SHIPS.

ES	1	2	3	The tower
θ^i/°	45	60	340	0
R^i/m	8	10	12	17

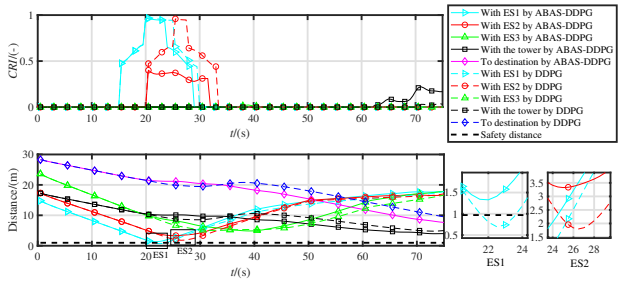


Fig. 6. The CRIs and distances results of DDPG and ABAS-DDPG of Tito-Neri.

A maximum perception range of the Tito-Neri ship is set as $R_{p\max}$ = 5m, i.e., the own ship cannot observe the states of the encountering ships with the relative distance $R_T > R_{p\max} = 5$m. Besides, the weight of the safety reward λ_s is set larger than that of the economic reward λ_e for navigation safety. Referring to the basic reward function in [5], we set $\lambda_s = 0.85$, $\lambda_e = 0.15$ for collision avoidance in this study. Referring to [31], the safety distance of Tito-Neri is set based on the ship length, i.e., 0.97m.

2) Results: The final trajectories of Tito-Neri after 1.5×10^5 learning steps are shown in Fig. 4. The light pink and blue

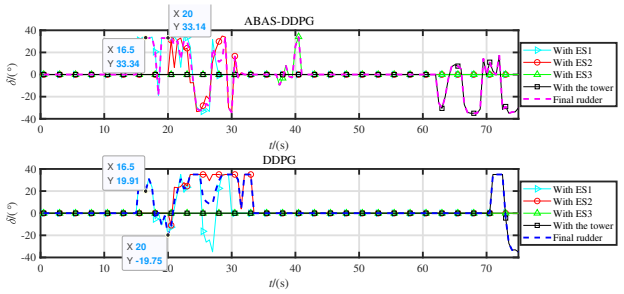


Fig. 7. The rudder actions of DDPG and ABAS-DDPG of Tito-Neri.

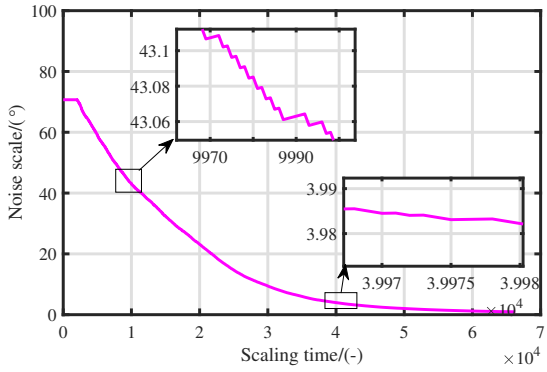


Fig. 8. The noise scaling results of the proposed ABAS-DDPG on Tito-Neri.

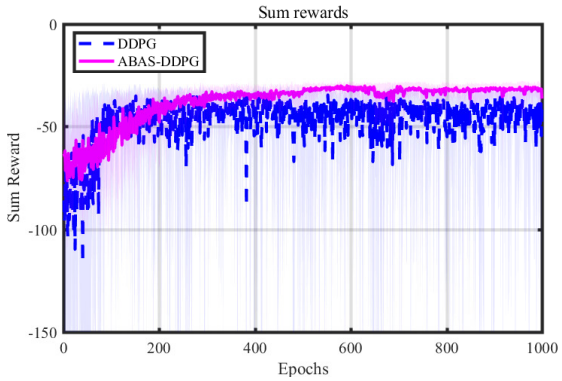


Fig. 9. The mean sum rewards of the proposed ABAS-DDPG and original DDPG of Delfia 1*

solid circles around the own ship show the perception ranges at different time steps. The sum rewards $\sum_{j=1}^{T} r_j$ of the proposed ABAS-DDPG and original DDPG of Tito-Neri are shown in Fig. 5, where T is the maximum steps in an episode and r_j is the obtained reward in j step. The details of the CRIs and relative distances between different encountering ships are shown in Fig. 6. The rudder actions of Tito-Neri for different encountering ships and the final rudders are shown in Fig. 7. Besides, the indicators including the minimum relative distances between different ships $R_{T_{\min}}$ and the corresponding maximum CRIs $f_{CRI_{\max}}$, the final distance to the destination $D_{\min}$, the duration when the own ship outside the boundary T_{ob}, the final reward are both calculated in Table IV to analyze the learning performance.

It can be seen from Fig. 4∼Fig. 6 that the collision risks between the own ship and the encountering ships are perceived as zero when $t < 15s$ since all the encountering ships are out of the perception range at that time. After the own ship has detected the collision risks, the algorithms are able to generate avoiding actions. As can be seen from Fig. 7, the initial trajectories generated by the ABAS-DDPG and the original DDPG are still stochastic at 1×10^4 step, e.g., running a circle or outside the boundary. After 6×10^4 steps, the learned policies can already generate more stable trajectories, which indicates the effectiveness of reinforcement learning.

Compared with the original DDPG, the proposed ABAS-DDPG can obtain higher and stabler rewards in the learning process, and the trend of policy convergence is more obvious. From Fig. 5, it can be seen that an inappropriate noise scale may lead to unstable learning results in the convergence stage. For example, it can be seen from Fig. 6, Fig. 7 and Table IV that the original DDPG with a constant noise scale fails to generate stable rudders with ES1 (at 16.5s) and ES2 (at 20s), which results in higher collision risks $f_{CRI_{\max}}$ and smaller minimum distances R_T with ES1 and ES2 than ABAS-DDPG. Particularly, the original DDPG fails to avoid ES1, i.e., the minimum distance R_T is smaller than the safety distance. While the proposed ABAS-DDPG with the adaptive noise successfully avoids all encountering ships and gets closer to the destination than the original DDPG in the final stage.

Fig. 8 shows the noise scaling process of the proposed ABAS-DDPG on Tito-Neri ship. As can be seen in the detailed views, the integrating BAS is adjusting the noise scale in the entire learning process with a global soft decreasing trend. Moreover, the BAS adopts less noise increasing behaviors adaptively as the learning progresses since the learned policy becomes better and the probability of $\eta_t = \eta_{\min}$ becomes larger than that of $\eta_t = \eta_{\max}$ in (27).

C. Case study 2: Collision avoidance learning of Delfia 1 ship*

1) Set up: For the convenience of comparison, the parameters of the algorithms and the scenarios are set the same as in Table II and Table III, respectively. The safety distance of the Delfia 1* ship is set based on the length of the ship, i.e., 0.38m.

2) Results: Similarly as the results of Tito-Neri, the mean sum rewards of the proposed ABAS-DDPG and original DDPG of Delfia 1* are shown in Fig. 9. The final trajectories of Delfia 1* are shown in Fig. 10. The details of the CRIs and relative distances are shown in Fig. 11. The rudder actions of Delfia 1* are shown in Fig. 12. The indicators are calculated in Table V. Fig. 13 shows the noise scaling process of the proposed ABAS-DDPG on Delfia 1*. Results similar to those in Fig. 8 can be seen that the ABAS is adjusting the noise scale effectively with Delfia 1*.

Since the Delfia 1* is more flexible than the Tito-Neri, it is more difficult for Delfia 1* to learn a good collision avoidance policy within the same number of epochs. Therefore, the trajectories of Delfia 1* are more stochastic than those of Tito-Neri during the learning process, which can be seen from

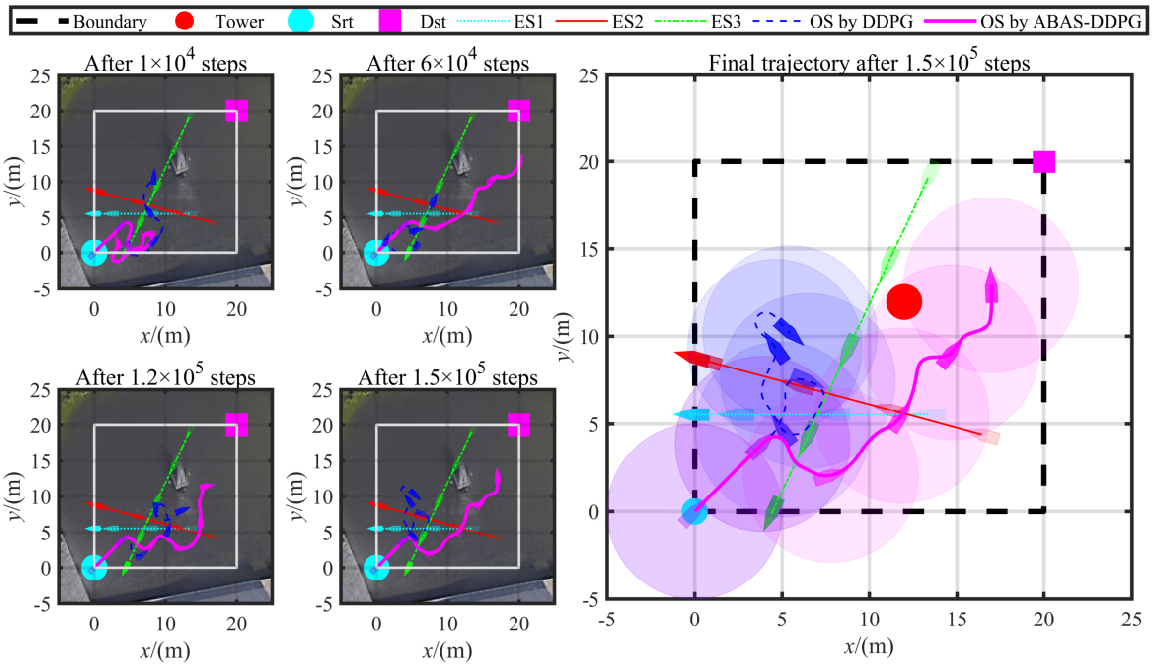


Fig. 10. The learning results of DDPG and ABAS-DDPG of Delfia 1*.

TABLE IV
THE COLLISION AVOIDANCE INDICATORS OF TITO-NERI OF THE PROPOSED ABAS-DDPG AND ORIGINAL DDPG.

Method	ES1		ES2		ES3		Tower		$D_{\min}$	T_{ob}	Reward
	$R_{T_{\min}}$	$f_{CRI_{\max}}$	$R_{T_{\min}}$	$f_{CRI_{\max}}$	$R_{T_{\min}}$	$f_{CRI_{\max}}$	$R_{T_{\min}}$	$f_{CRI_{\max}}$			
ABAS-DDPG	1.329	0.965	3.326	0.400	4.859	0.023	3.951	0.221	7.663	0	-37.572
DDPG	0.695	0.970	1.798	0.959	5.047	0.000	4.791	0.038	9.489	0	-50.202

TABLE V
THE COLLISION AVOIDANCE INDICATORS OF DELFIA 1* OF THE PROPOSED ABAS-DDPG AND ORIGINAL DDPG.

Method	ES1		ES2		ES3		Tower		$D_{\min}$	T_{ob}	Reward
	$R_{T_{\min}}$	$f_{CRI_{\max}}$	$R_{T_{\min}}$	$f_{CRI_{\max}}$	$R_{T_{\min}}$	$f_{CRI_{\max}}$	$R_{T_{\min}}$	$f_{CRI_{\max}}$			
ABAS-DDPG	1.821	0.558	4.675	0.040	2.851	0.380	3.681	0.471	12.158	0	-36.843
DDPG	0.912	0.948	3.579	0.421	1.883	0.544	9.285	0.000	19.908	0	-56.028

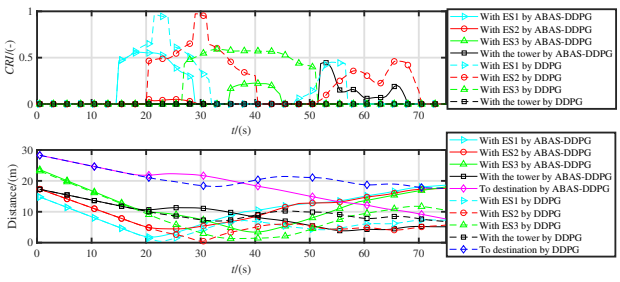


Fig. 11. The CRIs and distances results of DDPG and ABAS-DDPG of Delfia 1*.

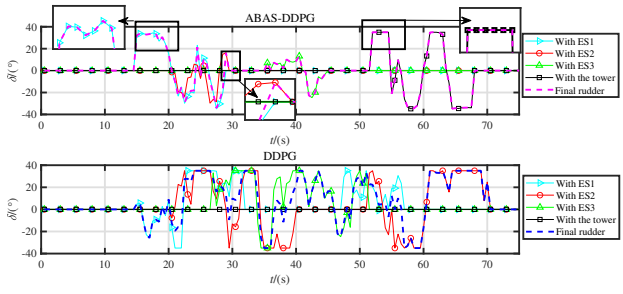


Fig. 12. The rudder actions of DDPG and ABAS-DDPG of Delfia 1*.

Fig. 10 and Fig. 4. The obtained rewards of the proposed ABAS-DDPG are also obviously higher than those of the original DDPG, which can be seen in Fig. 9. The influence of the inappropriate noise scale on the learning of Delfia 1* are more obvious than that of Tito-neri. E.g., from the detailed results in Fig. 11 and Table V, it can be seen that the proposed ABAS-DDPG is already able to generate continuous starboard steering actions to avoid ES1 and ES2, as well as ES3 and the tower after 1.2×10^5 learning steps. While the original DDPG with a constant noise scale still fails to learn a effective policy to generate stable trajectories after 1.5×10^5 steps, which results in much higher collision risks $f_{CRI_{\max}}$ with ES1, ES2

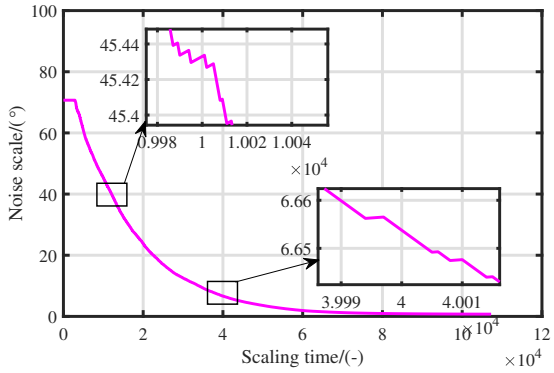


Fig. 13. TThe noise scaling results of the proposed ABAS-DDPG on Delfia 1*.

and ES3 and larger distance to the destination $D_{\min}$ than the proposed method. Moreover, as can be seen from Fig. 9 and Table V, the final obtained sum reward of the proposed method is also larger and stabler than that of the original DDPG, which is similar to the result of Tito-Neri ship.

VI. Conclusions and Future Research

Setting a constant noise scale in DDPG is an easy approach, while it may be insufficient at the beginning stage or too large in the convergence stage of learning. In order to realize adaptive exploration of DDPG reinforcement learning method for ship collision avoidance, an ABAS-DDPG algorithm is proposed that integrates an adaptive beetle antenna search (ABAS) optimizer for adaptive scaling of the noise injection in DDPG. The main originality of the proposed method is to use the Q-value estimated by the critic in DDPG as the fitness in BAS for concise noise scaling, which avoids large computation and repeated interactions. From the simulation results of a Tito-Neri tug model ship and a fully-actuated Delfia 1* ship, it can be concluded that

In summary, benefiting from the adaptive scaling of the BAS, the proposed ABAS-DDPG could be more suitable than the original DDPG for direct application without parallel testing by noise removal, at least, the time of policy convergence can be determined more clearly.

Future works should be carried out on the following aspects:

1) Only the own ship is taken as the agent for the ship collision avoidance learning. Since the cooperative multi-vessel systems (CMVSs) becomes the trend in ship collision avoidance, multi-agent reinforcement learning will be considered in future research.

2) Practical experiments with real ships or larger scales will be considered to verify the learned policy in further research.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (2018YFB1600400), the Fundamental Research Funds for the Central Universities (WUT:203144003), the National Natural Science Foundation of China (No. 51709220), the Open Project Program of Fujian University Engineering Research Center of Marine Intelligent Ship Equipment (No. 322031010602), the ResearchLab Autonomous Shipping (RAS) of Delft University of Technology, and the Joint WUT (Wuhan University of Technology) - TUDelft (Delft University of Technology) Cooperation. The authors wish to thank Ir. Vittorio Garofano for providing the hydrodynamics model of Delfia 1* and Tito-Neri.

References

[1] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Robust distributed predictive control of waterborne AGVs—A cooperative and cost-effective approach," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2449–2461, 2018.

[2] S. Xie, V. Garofano, X. Chu, and R. R. Negenborn, "Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks," *Ocean Engineering*, vol. 193, p. 106609, 2019.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of Go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[5] Y. Cheng and W. Zhang, "Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels," *Neurocomputing*, vol. 272, pp. 63–73, 2018.

[6] T. Tan, F. Bao, Y. Deng, A. Jin, Q. Dai, and J. Wang, "Cooperative deep reinforcement learning for large-scale traffic grid signal control," *IEEE transactions on cybernetics*, 2019.

[7] Z. Zhang, J. Chen, Z. Chen, and W. Li, "Asynchronous episodic deep deterministic policy gradient: Toward continuous control in computationally complex environments," *IEEE Transactions on Cybernetics*, 2019.

[8] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[9] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proceddings of the International conference on machine learning*, 2016, pp. 1928–1937.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[11] Y. Z. Xue, Y. Wei, and Y. Qiao, "The research on ship intelligence navigation in confined waters," *Advanced Materials Research*, vol. 442, pp. 398–401, 2012.

[12] Y. Singh, S. Sharma, R. Sutton, D. Hatton, and A. Khan, "A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents," *Ocean Engineering*, vol. 169, pp. 187–201, 2018.

[13] C. Liu, Q. Mao, X. Chu, and S. Xie, "An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning," *Applied Sciences*, vol. 9, no. 6, pp. 1057–1074, 2019.

[14] L. Cheng, C. Liu, and B. Yan, "Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot," in *2014 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2014, pp. 695–698.

[15] M.-H. Kim, J.-H. Heo, Y. Wei, and M.-C. Lee, "A path planning algorithm using artificial potential field based on probability map," in *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. IEEE, 2011, pp. 41–43.

[16] A. Lazarowska, "A new potential field inspired path planning algorithm for ships," in *2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR)*. IEEE, 2018, pp. 166–170.

[17] Y. Huang, L. Chen, P. Chen, R. R. Negenborn, and P. van Gelder, "Ship collision avoidance methods: State-of-the-art," *Safety Science*, vol. 121, pp. 451–473, 2020.

[18] K. Hasegawa, A. Kouzuki, T. Muramatsu, H. Komine, and Y. Watabe, "Ship auto-navigation fuzzy expert system (safes)," *Journal of the Society of Naval Architects of Japan*, vol. 1989, no. 166, pp. 445–452, 1989.

[19] K. Hara and A. Hammer, "A safe way of collision avoidance maneuver based on maneuvering standard using fuzzy reasoning model," in *Proceedings of the 1993 International Conference on Marine Simulation & Ship Manoeuvrability*, 1993, pp. 163–170.

[20] L. P. Perera, J. P. Carvalho, and C. G. Soares, "Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions," *Journal of Marine Science & Technology*, vol. 16, no. 1, pp. 84–99, 2011.

[21] L. P. Perera, V. Ferrari, F. P. Santos, M. A. Hinostroza, and C. G. Soares, "Experimental evaluations on ship autonomous navigation and collision avoidance by intelligent guidance," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 2, pp. 374–387, 2014.

[22] J. H. Ahn, K. P. Rhee, and Y. J. You, "A study on the collision avoidance of a ship using neural networks and fuzzy logic," *Applied Ocean Research*, vol. 37, no. 4, pp. 162–173, 2012.

[23] U. Simsir, M. Bal, and S. Ertugrul, "Decision support system for collision avoidance of vessels," *Applied Soft Computing Journal*, vol. 25, no. C, pp. 369–378, 2014.

[24] A. Lazarowska, "Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation," *Journal of Navigation*, vol. 68, no. 2, pp. 291–307, 2015.

[25] Y. Ma, M. Hu, and X. Yan, "Multi-objective path planning for unmanned surface vehicle with currents effects," *ISA transactions*, vol. 75, pp. 137–156, 2018.

[26] L. Chen, J. J. Hopman, and R. R. Negenborn, "Distributed model predictive control for vessel train formations of cooperative multi-vessel systems," *Transportation Research Part C: Emerging Technologies*, vol. 92, pp. 101–118, 2018.

[27] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Fast ADMM for distributed model predictive control of cooperative waterborne AGVs," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1406 – 1413, 2017.

[28] ——, "Closed-loop scheduling and control of waterborne AGVs for energy-efficient inter terminal transport," *Transportation Research Part E: Logistics and Transportation Review*, vol. 105, pp. 261–278, 2017.

[29] ——, "Predictive path following with arrival time awareness for waterborne AGVs," *Transportation Research Part C: Emerging Technologies*, vol. 70, pp. 214–237, 2016.

[30] R. R. Negenborn and J. M. Maestre, "Distributed model predictive control: An overview and roadmap of future research opportunities," *IEEE Control Systems Magazine*, vol. 34, no. 4, pp. 87–97, 2014.

[31] L. Chen, Y. Huang, H. Zheng, J. J. Hopman, and R. R. Negenborn, "Cooperative multi-vessel systems in urban waterway networks," *IEEE Transactions on Intelligent Transportation Systems*, no. 1, pp. 1–14, 2019.

[32] B. Yoo and J. Kim, "Path optimization for marine vehicles in ocean currents using reinforcement learning," *Journal of Marine Science and Technology*, vol. 21, no. 2, pp. 334–343, 2016.

[33] C. Chen, X.-Q. Chen, F. Ma, X.-J. Zeng, and J. Wang, "A knowledge-free path planning approach for smart ships based on reinforcement learning," *Ocean Engineering*, vol. 189, p. 106299, 2019.

[34] C. Chen, F. Ma, J.-L. Liu, X.-P. Yan, and X.-Q. Chen, "A novel path planning approach for unmanned ships based on deep reinforcement learning," *Data Science and Knowledge Engineering for Sensing Decision Support*, pp. 626–633, 2019.

[35] H. Shen, C. Guo, and T. Li, "An intelligent collision avoidance and navigation approach of unmanned surface vessel considering navigation experience and rules," *Journal of Harbin Engineering University*, vol. 39, no. 6, pp. 1–7, 2017.

[36] H. Shen, H. Hashimoto, A. Matsuda, Y. Taniguchi, D. Terada, and C. Guo, "Automatic collision avoidance of multiple ships based on deep Q-learning," *Applied Ocean Research*, vol. 86, pp. 268–288, 2019.

[37] R. Zhang, X. Wang, K. Liu, X. Wu, T. Lu, and C. Zhaohui, "Ship collision avoidance using constrained deep reinforcement learning," in *Proceedings of the IEEE 2018 5th International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC)*. IEEE, 2019, pp. 115–120.

[38] H. Xu, N. Wang, H. Zhao, and Z. Zheng, "Deep reinforcement learning-based path planning of underactuated surface vessels," *Cyber-Physical Systems*, vol. 5, no. 1, pp. 1–17, 2019.

[39] D.-H. Kim, S.-U. Lee, J.-H. Nam, and Y. Furukawa, "Determination of ship collision avoidance path using deep deterministic policy gradient algorithm," *Journal of the Society of Naval Architects of Korea*, vol. 56, no. 1, pp. 58–65, 2019.

[40] I. R. Bertaska, *Intelligent supervisory switching control of unmanned surface vehicles*. Florida Atlantic University, 2016.

[41] J. Woo, C. Yu, and N. Kim, "Deep reinforcement learning-based controller for path following of an unmanned surface vehicle," *Ocean Engineering*, vol. 183, pp. 155–166, 2019.

[42] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.

[43] M. Dorigo and G. Di Caro, "Ant colony optimization: a new meta-heuristic," in *Proceedings of the 1999 congress on evolutionary computation-CEC99 (Cat. No. 99TH8406)*, vol. 2. IEEE, 1999, pp. 1470–1477.

[44] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[45] X. Jiang and S. Li, "BAS: Beetle antennae search algorithm for optimization problems," *International Journal of Robotics and Control*, vol. 1, no. 1, pp. 1–5, 2018.

[46] J. Wang and H. Chen, "BSAS: Beetle swarm antennae search algorithm for optimization problems," *arXiv preprint arXiv:1808.00206*, 2018.

[47] T. Wang, L. Yang, and Q. Liu, "Beetle swarm optimization algorithm: Theory and application," *arXiv preprint arXiv:1808.00206*, 2018.

[48] Y. Sun, J. Zhang, G. Li, Y. Wang, J. Sun, and C. Jiang, "Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes," *International Journal for Numerical and Analytical Methods in Geomechanics*, 2019.

[49] X. Lin, Y. Liu, and Y. Wang, "Design and research of DC motor speed control system based on improved BAS," in *2018 Chinese Automation Congress (CAC)*. IEEE, 2018, pp. 3701–3705.

[50] M. Lin and Q. Li, "A hybrid optimization method of beetle antennae search algorithm and particle swarm optimization," *DEStech Transactions on Engineering and Technology Research*, no. ecar, 2018.

[51] T. Chen, Y. Zhu, and J. Teng, "Beetle swarm optimisation for solving investment portfolio problems," *The Journal of Engineering*, vol. 2018, no. 16, pp. 1600–1605, 2018.

[52] Y. Mu, B. Li, D. An, and Y. Wei, "Three-dimensional route planning based on the beetle swarm optimization algorithm," *IEEE Access*, vol. 7, pp. 117 804–117 813, 2019.

[53] T. Xu, Q. Liu, L. Zhao, and J. Peng, "Learning to explore with meta-policy gradient," *arXiv preprint arXiv:1803.05044*, 2018.

[54] A. Sharaf and I. Daumé, Hal, "Meta-Learning for Contextual Bandit Exploration," *arXiv e-prints*, p. arXiv:1901.08159, Jan 2019.

[55] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[56] R. Szlapczynski and J. Szlapczynska, "Review of ship safety domains: Models and applications," *Ocean Engineering*, vol. 145, pp. 277–289, 2017.

[57] A. Haseltalab and R. R. Negenborn, "Model predictive maneuvering control and energy management for all-electric autonomous ships," *Applied Energy*, vol. 251, p. 113308, 2019.

[58] M. A. Abkowitz, "Measurement of hydrodynamic characteristics from ship maneuvering trials by system identification," *Maneuverability*, 1980.

[59] H. Yasukawa and Y. Yoshimura, "Introduction of mmg standard method for ship maneuvering predictions," *Journal of Marine Science and Technology*, vol. 20, no. 1, pp. 37–52, 2015.

[60] W. Luo and X. Li, "Measures to diminish the parameter drift in the modeling of ship manoeuvring using system identification," *Applied Ocean Research*, vol. 67, pp. 9 – 20, 2017.

[61] D. Chen, C. Dai, X. Wan, and J. Mou, "A research on AIS-based embedded system for ship collision avoidance," in *2015 International Conference on Transportation Information and Safety (ICTIS)*. IEEE, 2015, pp. 512–517.

[62] P. Davis, M. Dove, and C. Stockel, "A computer simulation of multi-ship encounters," *Journal of Navigation*, vol. 35, no. 2, pp. 347–352, 1982.

[63] G. E. Uhlenbeck and L. S. Ornstein, "On the theory of the brownian motion," *Physical review*, vol. 36, no. 5, p. 823, 1930.

[64] X. Jiang and S. Li, "Beetle antennae search without parameter tuning (BAS-WPT) for multi-objective optimization," *arXiv preprint arXiv:1807.10470*, 2017.

[65] S. Xie, X. Chu, M. Zheng, and C. Liu, "Ship predictive collision avoidance method based on an improved beetle antennae search algorithm," *Ocean Engineering*, vol. 192, p. 106542, 2019.

[66] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, and M. Andrychowicz, "Parameter space noise for exploration," *arXiv preprint arXiv:1706.01905*, 2017.

Shuo Xie received the M.Sc. degree and Ph.D. degree in transportation engineering from the Wuhan University of Technology in 2017 and 2020, respectively.

His research interests include ship collision avoidance and ship model identification.

Vittorio Garofano received the B.Sc. degree in control and automation system engineering from the University of Rome, Italy, in 2012, and the M.Sc. degree in mechatronic engineering from the Polytechnic University of Turin, Italy, in 2015. He is currently pursuing the Ph.D. degree with the Department of Maritime and Transport Technology, Delft University of Technology, Delft, The Netherlands.

His research interests include autonomous ship systems, multi-agent systems and intelligent ships.

Xiumin Chu received his PhD degree (2002) and M.S. degree (1998) majoring in automobile application engineering in Jilin University. He is currently a professor in the National Engineering Research Center for Water Transport Safety, Wuhan University of Technology, Wuhan, China. He is also the Secretary General of the intelligent transportation professional committee of China Artificial Intelligence Society.

His research interests include waterway transportation intelligence, smart ship, and ship motion simulation.

Rudy R. Negenborn received the M.Sc. degree in computer science/intelligent systems from Utrecht University in 1998 and the Ph.D. degree in distributed control for networked systems from the Delft University of Technology, Delft, The Netherlands, in 2007. He is currently a Full Professor (Multi-Machine Operations and Logistics) with the Department of Maritime and Transport Technology, Delft University of Technology.

His research interests are in the areas of distributed control, multi-agent systems, model predictive control, and optimization. He applies the developed theories to address control problems in large-scale transportation and logistic systems.