


Article

PID Controller Based on Improved DDPG for Trajectory Tracking Control of USV

Xing Wang ^{1,2}, Hong Yi ^{1,3}, Jia Xu ², Chuanyi Xu ⁴ and Lifei Song ^{4,*} 

¹ School of Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; wangxing786235493@163.com (X.W.); yihong@mail.sjtu.edu.cn (H.Y.)

² China Ship Development and Design Center, Wuhan 430064, China; jiaxu_china@163.com

³ Key Laboratory of Marine Intelligent Equipment and System, Shanghai Jiao Tong University, Ministry of Education, Shanghai 200240, China

⁴ Key Laboratory of High Performance Ship Technology, Wuhan University of Technology, Ministry of Education, Wuhan 430070, China; xcyzm@foxmail.com

* Correspondence: songlifei@whut.edu.cn

Abstract: When navigating dynamic ocean environments characterized by significant wave and wind disturbances, USVs encounter time-varying external interferences and underactuated limitations. This results in reduced navigational stability and increased difficulty in trajectory tracking. Controllers based on deterministic models or non-adaptive control parameters often fail to achieve the desired performance. To enhance the adaptability of USV motion controllers, this paper proposes a trajectory tracking control algorithm that calculates PID control parameters using an improved Deep Deterministic Policy Gradient (DDPG) algorithm. Firstly, the maneuvering motion model and parameters for USVs are introduced, along with the guidance law for path tracking and the PID control algorithm. Secondly, a detailed explanation of the proposed method is provided, including the state, action, and reward settings for training the Reinforcement Learning (RL) model. Thirdly, the simulations of various algorithms, including the proposed controller, are presented and analyzed for comparison, demonstrating the superiority of the proposed algorithm. Finally, a maneuvering experiment under wave conditions was conducted in a marine tank using the proposed algorithm, proving its feasibility and effectiveness. This research contributes to the intelligent navigation of USVs in real ocean environments and facilitates the execution of subsequent specific tasks.

Keywords: unmanned surface vehicles; PID controller; reinforcement learning; DDPG



Citation: Wang, X.; Yi, H.; Xu, J.; Xu, C.; Song, L. PID Controller Based on Improved DDPG for Trajectory Tracking Control of USV. *J. Mar. Sci. Eng.* **2024**, *12*, 1771. <https://doi.org/10.3390/jmse12101771>

Academic Editor: Sergei Chernyi

Received: 17 August 2024

Revised: 24 September 2024

Accepted: 2 October 2024

Published: 6 October 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Unmanned surface vehicles (USVs), also known as Autonomous Surface Vessels (ASVs), are capable of autonomous navigation and operations. They are often deployed in complex and harsh oceanic environments, which necessitate robust autonomous navigation capabilities. This autonomous navigation serves as the foundation for subsequent tasks such as collision avoidance, automatic docking, target tracking, and formation control. Among these capabilities, autonomous trajectory tracking is a critical aspect that highlights the USV's navigational proficiency. When navigating dynamic ocean environments with significant wave and wind disturbances, USVs face time-varying external interferences and limitations due to their underactuated nature, leading to reduced navigational stability and increased difficulty in trajectory tracking. Over the years, researchers have relentlessly explored the trajectory tracking control mechanisms of USVs under wave and wind conditions.

Do K. D. [1] investigated the trajectory tracking control of USVs in the presence of wind and wave disturbances using the Serret–Frenet frame. An adaptive robust trajectory tracking controller for underactuated USVs was developed by designing nonlinear

observers for the lateral velocity and yaw rate based on nonlinear backstepping and the Lyapunov stability theory [2].

Sun [3] designed an exponentially stable trajectory tracking controller for USVs based on the dynamic surface control theory. Aguiar A. P. [4] researched trajectory tracking control for USVs using switching control techniques, while Fahimi F. [5] proposed a robust sliding mode trajectory tracking control method. Harmouche M. [6] addressed the lack of speed measurement feedback in the trajectory tracking of USVs by proposing a control method based on observer technology and backstepping. Katayama H. [7] introduced a straight-line trajectory tracking control method for underactuated vessels with semi-global uniform asymptotic stability, utilizing nonlinear sampled-data control theory, state feedback, output feedback control techniques, and observer technology [8,9]. However, the aforementioned methods, including optimal control, feedback linearization, and backstepping, require precise modeling to achieve high control accuracy [10]. The motion model of a USV is affected by variables such as speed and load, making precise modeling challenging. Moreover, disturbances from wind, waves, and currents during navigation complicate the path-following control of USVs. Therefore, in real oceanic environments, control algorithms based on deterministic models or those with non-adaptive control parameters often fail to achieve the desired control performance.

The PID control algorithm remains dominant in ship autopilot systems because of its simplicity and reliability. However, when USVs experience substantial time-varying disturbances, the effectiveness of fixed-parameter PID algorithms is not satisfactory. Researchers have sought to enhance the effectiveness through improving the adaptability of PID parameters [11]. Most efforts are focused on combining PID with other theories such as fuzzy control and neural networks to achieve the adaptive tuning of PID parameters.

Adaptive PID controllers adjust PID parameters dynamically during the process of tracking a desired heading, significantly enhancing the algorithm's dynamic response. However, due to model uncertainties and external disturbances, a discrepancy between the estimated and actual system outputs often exists. Hu Zhiqiang [12] proposed an online self-optimizing PID heading control algorithm, which facilitates the online adjustment of control parameters and exhibits robust performance and interference resistance.

Genetic algorithms (GAs), known for their stable global optimization capabilities, are frequently used for parameter tuning in various USV path controllers. Liu [13] has demonstrated the use of GA for the online tuning of PID parameters to implement adaptive PID control. However, these controllers face challenges such as prolonged parameter optimization times, which can impact the real-time applicability on actual vessels. Designing crossover and mutation operators within the optimization process can shorten the optimization time and enhance the algorithm's real-time performance [14].

Fuzzy logic control translates expert knowledge into fuzzy rules; it can effectively address the impacts of model uncertainties and random disturbances on USV path-following control. In practical applications, fuzzy logic is often used for the parameter tuning of PID controllers and sliding mode controllers due to its rapid response and real-time performance [15,16]. Liu [15] has established fuzzy rules according to path point errors, heading errors, and error differentials to improve control smoothness. However, the accuracy of fuzzy controllers mainly depends on the complexity of the fuzzy rules, which is generally constructed based on expert knowledge and dynamic models, so complex rules may lead to computational challenges [17].

Peng Yan [18] researched the challenges faced by USV systems, such as large inertia, long time delays, nonlinearity, difficulty in precise modeling, and susceptibility to external disturbances like waves. The study revealed that traditional PID control often fails to achieve satisfactory trajectory tracking performance. Consequently, a PID cascade controller based on generalized predictive control (GPC-PID) was designed to separately control the heading and steering motions of USVs, providing enhanced resistance to external disturbances. Additionally, radial basis function (RBF) neural networks can approximate model uncertainties and external disturbances affecting PID parameters, thus improving the

robustness and interference resistance of the controller. RBF neural networks are frequently employed to model the impact of internal and external disturbances on PID parameters.

Reinforcement Learning (RL) has been extensively applied in control systems [19]. Its capability to operate without precise mathematical models and its self-learning abilities make it particularly effective in addressing challenges related to model uncertainties and unknown disturbances in USV trajectory tracking control [20,21]. Neural networks were utilized within sliding mode controllers to tune controller parameters, with RL employed to evaluate the tuning efficacy [22]. This approach enabled the self-learning of neural network parameters, addressing the low model accuracy requirement of sliding mode control while mitigating its chattering issue.

Sun [3] explored trajectory tracking control for underactuated USVs under unknown disturbances using neural network control technologies. Another study [23] employed Q-learning for PID parameter tuning, demonstrating that this controller can effectively resist external disturbances and facilitate the motion control of mobile robots through experiments. Bertaska [24] developed a multi-mode switching controller using Q-learning, which intelligently switches controllers based on the environment and the operational state of the USV, enhancing control performance across different conditions.

Magalhaes [25] proposed a RL control method based on Q-learning that mimics fish swimming motions to control the fins and tail of an unmanned underwater vehicle (UUV) for trajectory tracking. Bian [26] employed neural network controllers for ship trajectory tracking control; however, such learning methods require the pre-acquisition of reliable ship navigation sample data. Deep Q Networks (DQNs), which incorporate experience replay and fixed Q-targets, significantly enhance the stability and expressiveness of complex RL problems [27]. However, DQNs remain constrained to discrete action or state spaces based on Q-learning, making them challenging to apply to the continuous control problem of USV motion under dynamic wave conditions, where low control precision can lead to chattering.

Wu [28] modeled the trajectory tracking problem as a Markov Decision Process (MDP) based on the maneuvering characteristics and control requirements of ships. The DDPG algorithm was used as the controller, and offline learning methods were applied for controller training. Simulation tests showed promising results, although the simulation environment lacked environmental disturbances. Woo [29] introduced a trajectory tracking controller based on Deep Reinforcement Learning (DRL), allowing USVs to learn navigation experiences during voyages. In repetitive trajectory tracking simulations, the control strategy was trained to enable adaptive interaction with the environment, achieving trajectory following. However, this method directly employed DRL as the control strategy rather than adaptive parameter matching, which lacks the mature performance of classical control algorithms. Additionally, it did not leverage the dynamic performance of USVs within the controller, but treated it as a black box, which demands high-quality training data. Without comprehensive training data, the control performance may not meet expectations.

The analysis above highlights several key issues in the current trajectory tracking control methods for USVs: Firstly, many existing algorithms fail to consider the impact of time-varying environmental disturbances. Secondly, when employing machine learning algorithms for training USV motion controllers, acquiring sample data poses significant challenges, and the quality of these training samples greatly influences the performance of machine learning-based controllers. Thirdly, when using RL algorithms such as Q-learning for controller training, these approaches typically use a black box method to compensate for time-varying and difficult-to-quantify environmental disturbances. However, the motion and control models established by these methods are based on discrete mathematical models with low precision. Fourthly, directly using RL as a controller, rather than for adaptive parameter tuning, fails to leverage the superior performance of classical control algorithms and relies entirely on black box processes, which preclude the utilization of USV dynamics within the controller.

To address these challenges, this paper proposes a trajectory tracking control algorithm for USVs that calculates PID control parameters based on an improved Deep

Deterministic Policy Gradient (DDPG) algorithm. By integrating RL algorithms with classical control theory, this approach aims to enhance the adaptability of USV motion controllers. The subsequent sections of this paper are organized as follows: Section 2 introduces the maneuvering motion model and parameters for USVs, the guidance law for path tracking, and the PID control algorithm employed. Section 3 provides a detailed explanation of the proposed method for calculating PID parameters based on the improved DDPG algorithm and outlines the state, action, and reward settings for training the RL model. Section 4 presents the simulation results of various algorithms, demonstrating the superiority of the algorithm proposed in this study. In Section 5, maneuvering experiment in wave conditions was conducted in a marine tank using the proposed algorithm, proving its feasibility and effectiveness.

2. The Maneuvering Motion Model of USV and Path-Following Controller for USV

2.1. The Maneuvering Motion Model of USV

The 3-DOF Maneuvering Modeling Group (MMG) motion model of the USV is described as follows [30]:

$$\begin{cases} \dot{x} = u\cos\varphi - v\sin\varphi \\ \dot{y} = u\sin\varphi + v\cos\varphi \\ \dot{\varphi} = r \\ \dot{u} = \frac{1}{m+m_x} [(m+m_y)rv + X_H + X_P + X_R] \\ \dot{v} = \frac{1}{m+m_y} [-(m+m_x)rv + Y_H + Y_P + Y_R] \\ \dot{r} = \frac{1}{I_{zz}+J_{zz}} (N_H + N_P + N_R) \end{cases} \quad (1)$$

where X , Y , and N represent hydrodynamic forces and moments, while the subscripts H , P , and R represent the hull, propeller, and rudder, respectively. x and y are the coordinates of the USV in the geodetic coordinate system, φ is the angle of the bow, u represents the longitudinal velocity of the USV in the local coordinate system, v is the lateral velocity, and r denotes the angular velocity in the local coordinate system. m_x is the additional longitudinal mass, m_y is the additional transverse mass, and I_{zz} is the inertia moment of yaw. J_{zz} is the additional moment of inertia in yaw.

The model for calculating X_{calm} , Y_{calm} , N_{calm} was established using the MMG model.

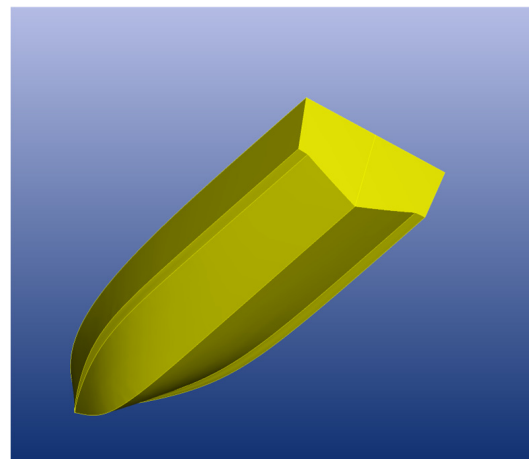
$$\begin{cases} X_{calm} = X_H + X_P + X_R \\ Y_{calm} = Y_H + Y_R \\ N_{calm} = N_H + N_R \end{cases} \quad (2)$$

The force acting on the hull is approximate to the hydrodynamic derivative, X_H , Y_H , N_H , and can be expressed as

$$\begin{cases} X_H = X_0 + X_u\Delta u + X_{uu}(\Delta u)^2 + X_{uuu}(\Delta u)^3 + X_{vv}v^2 + X_{rr}r^2 + X_{vr}vr \\ Y_H = Y_vv + Y_{vvv}v^3 + Y_{\dot{v}}\dot{v} + Y_r r + Y_{rrr}r^3 + Y_{\dot{r}}\dot{r} + Y_{vrr}vr^2 + Y_{vvr}v^2r \\ N_H = N_vv + N_{vvv}v^3 + N_{\dot{v}}\dot{v} + N_r r + N_{rrr}r^3 + N_{\dot{r}}\dot{r} + N_{vrr}vr^2 + N_{vvr}v^2r \end{cases} \quad (3)$$

where $\Delta u = u - u_0$ is the speed difference of the ship relative to the initial ship speed and $-X_0$ is the resistance when the speed of ship is u_0 . The geometric and physical models of the USV used in this study are depicted in Figure 1. Table 1 illustrates parameters of the USV.

Using Computational Fluid Dynamics (CFD) and regression analysis, the hydrodynamic derivatives for static water maneuvering have been obtained, as shown in Table 2.



(a) 3D model



(b) Entity model

Figure 1. Model of the USV.

Table 1. Parameters of the USV.

Part	Items	Definition	Value
Hull	L_{pp} [m]	Length between Perpendiculars	1.5
	B [m]	Breadth of Waterline	0.444
	d [m]	Depth	0.107
	C_B	Block Coefficient	0.395
	x_G [m]	X-position of Gravity Center relative to Midship	-0.12
	z_G [m]	Z-position of Gravity Center relative to Baseline	-0.3
	I_z [kg·m ²] I_x [kg·m ²]	Gyration about Z-axis Gyration about X-axis	3.947 0.35
Propeller	D_p [m]	Diameter	0.029
	Z_p	Number of Blades	5
Rudder	A_R [m ²]	Longitudinal Projection Area	0.001675
	H_R [m]	Rudder Height	0.05

Table 2. Hydrodynamic derivatives.

Hydrodynamic Derivative	Value
X'_0	-0.0026
X'_u	-0.000700
X'_v	-0.000238
X''_{uu}	0.002500
X''_{uuu}	-0.001800
X''_{vv}	0.002744
X''_{rr}	-0.002807
X''_{vr}	0.01247
Y'_v	-0.01471
Y'_v	-0.007049
Y''_{vvv}	0.112200
N'_v	-0.006399
N''_{vvv}	-0.006952
Y'_r	0.003013
Y'_r	-0.000777
Y''_{rrr}	-0.000467
N'_r	-0.001708
N'_v	-0.000283
N'_v	-0.000419
N''_{rrr}	-0.000261
Y''_{vvr}	-0.005687
Y''_{vrr}	-0.013020
N''_{vvr}	-0.015600
N''_{vrr}	-0.001047

The hydrodynamic interaction coefficients between the ship, propeller, and rudder are presented in Table 3.

Table 3. Ship–propeller–rudder hydrodynamic interaction coefficients.

Interaction Coefficients	Definition	Value
α_H	Rudder Force Correction Factor	0.5
t_p	Thrust Reduction Coefficient	0.01
w_{p0}	Wake Fraction Coefficient	0.01
ε		0.52
κ	Propeller–Rudder Interaction Coefficient	0.23
t_R		0.15
γ	Speed Coefficient	1.55
K_T	Open Water Propeller Thrust Coefficient	$0.5932 - 0.1971J_p - 0.0481J_p^2$

These hydrodynamic coefficients are used to account for the interactions between the ship, propeller, and rudder when calculating maneuvering forces.

2.2. The LOS Guidance Rule and PID Controller

The LOS strategy aims to guide the USV to track a virtual target along the desired trajectory, minimizing the position and heading deviations between the USV’s actual location and the target, ultimately ensuring the USV reaches the intended path. The trajectory tracking strategy using LOS is illustrated in Figure 2.

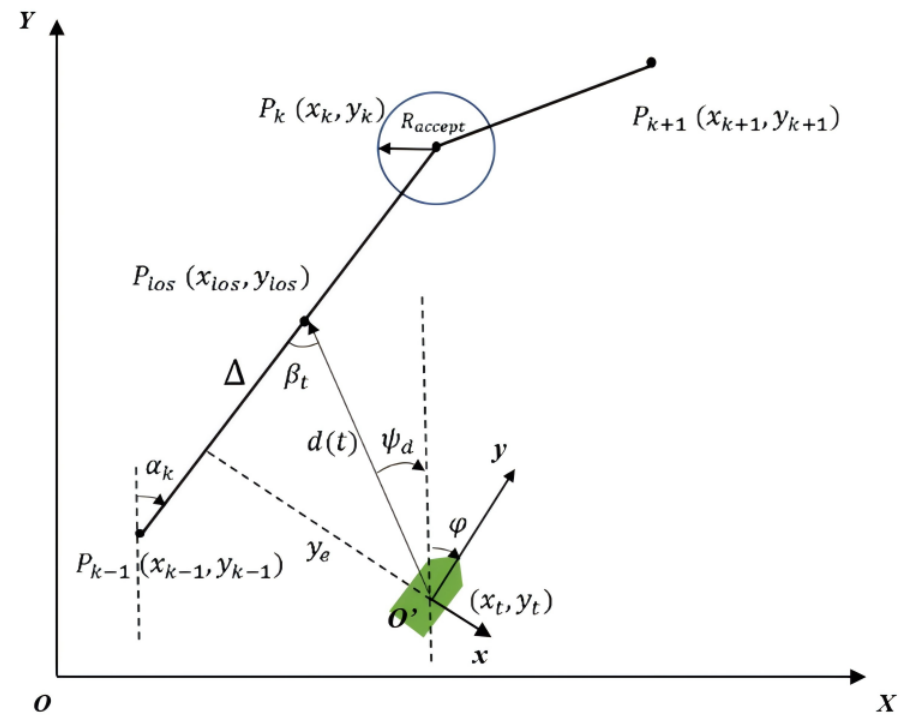


Figure 2. LOS guidance strategy.

In Figure 2, XOY denotes the global coordinate system, $xO'y$ denotes the local coordinate system, and the USV’s position is denoted as P , with heading angle φ , longitudinal velocity u , lateral velocity v , and yaw rate r . The target trajectory is segmented into several approximate straight-line segments, with the current segment being tracked lying between points P_{k-1} and P_k . The lateral tracking error is denoted as y_e , the look-ahead distance as Δ , and the desired heading angle as ψ_d . During the curve tracking process, the sub-target along the trajectory needs to be switched. The switch occurs from P_{k-1} to P_k when $\vec{P}_{k-1} \cdot \vec{P}_{k-1} P_k < 0$.

The PID controller is a linear regulator that compares the desired heading angle $\psi_d(k)$ with the actual heading angle $\varphi(k)$ to form the heading angle deviation $e(k)$:

$$e(k) = \psi_d(k) - \varphi(k) \tag{4}$$

The desired rudder angle can be expressed as Equation (5):

$$\delta(k) = K_p e(k) + K_i \sum_{i=0}^k e(i) dt - K_d (e(k) - e(k-1)) / dt \tag{5}$$

Considering the integral saturation condition of the PID controller, the PD parameters are adjusted to ensure the USV quickly tend to the desired track and keep the USV navigating within the error range. Therefore, Equation (5) can also be expressed as Equation (6):

$$\delta = K_p e + K_d (e - e') \tag{6}$$

where e' is the error at the previous moment. The neural network is performed to produce the appropriate PD parameters.

3. PID Parameter Calculating Model Based on an Improved DDPG Algorithm

3.1. Algorithm Description

The tracking control process of USVs is a time-sequential decision-making problem characterized by MDPs. Thus, a USV can be regarded as an agent, with its motion control expressed in the form of $\{s, A, P(s'|s, a), R\}$. Herein, the details are as follows:

- a. $P(s'|s, a)$ is the state transition matrix and refers to the probability of transitioning to the next state s' after applying action a in the current state s .
- b. s , the state set encompasses all states and exhibits the Markov property; that is, future states depend only on the current state and are independent of past states.
- c. A , the action set, comprises all possible actions that the agent can select. State transitions depend not only on the environment but also on the agent's ability to guide state transitions by selecting different actions.
- d. R , the reward function, maps states and actions to rewards, reflecting preferences for different states.

Executing actions transitions the system from an initial state to a terminal state, forming a trajectory τ , represented as

$$\tau = (s_0, a_0, s_1, a_1, \dots, s_t, a_t) \tag{7}$$

where s_0 is the initial state, s_t is the terminal state, and a_i is the action chosen at time step i .

The aim is to maximize the cumulative reward $R(\tau)$ along the trajectory, expressed as

$$R(\tau) = \sum_{t=0}^T \gamma^t r_t \tag{8}$$

where T is the terminal time, r_t is the reward at time t , and γ is the discount factor, with values ranging between 0 and 1.

To maximize the cumulative reward $R(\tau)$, the policy π must be continuously optimized. The policy π maps states s to actions a , and the agent chooses actions based on the observed state according to the policy. The strategy that maximizes the cumulative reward is called the optimal policy, denoted as π^* .

$$\pi^* = \operatorname{argmax}_{\pi} E_{\tau \sim \pi} [R(\tau)] \tag{9}$$

where E is the expectation, $\tau \sim \pi$ is the trajectory τ based on policy π .

The goal of RL is to find the optimal policy π^* .

$$Q^*(s, a) = \max_{\pi}(E_{\tau \sim \pi}[R(\tau)|s_t = s, a_t = a]) \tag{10}$$

$Q^*(s, a)$ denotes the optimal action value function, can be used to provide the best action, and can be recursively expressed based on the Bellman equation:

$$Q^*(s, a) = E_{s' \sim P}[r(s, a, s') + \max_{a'}[\gamma Q^*(s', a')]] \tag{11}$$

where $s' \sim P$ denotes the next state derived from the environment transition probability P , and a' represents the subsequent action obtained from the policy π .

$$a^*(s) = \operatorname{argmax}_a Q^*(s, a) \tag{12}$$

$a^*(s)$, the best actions, can be recursively expressed by the Bellman equation.

In policy-based methods, the policy is denoted as π_{θ} , where θ represents a set of parameters constituting the policy. The goal is to maximize the expected return $J(\pi_{\theta}) = E_{\tau \sim \pi_{\theta}}[R(\tau)]$, and gradient ascent is employed to update the parameters θ and optimize the policy.

$$\theta_{k+1} = \theta_k + \alpha \nabla J(\pi_{\theta})|_{\theta_k} \tag{13}$$

where $\nabla J(\pi_{\theta})$ is the policy gradient.

The improved DDPG algorithm in this study utilizes an Actor–Critic architecture, comprising four networks, detailed as follows:

(1) Current Actor Network

The input to the current Actor is the state space s , and the output is the action a . In this study, the state space is defined as

$$s = [u, v, r, \varphi, e_{psi}, y_e, d, \alpha_k, \dot{d}, \dot{e}_{psi}] \tag{14}$$

where u and v are the longitudinal and lateral velocities, r is the yaw rate, φ is the heading angle, e_{psi} is the lateral deviation from the target trajectory, d is the rudder angle, and α_k is the inclination angle of the target trajectory. The action a is defined as

$$a = [k_p, k_d] \tag{15}$$

The objective of updating the Actor is to maximize the Q-value evaluated by the current Critic network. Thus, the gradient of the Actor is updated via backpropagation through the Critic’s gradient.

The Actor network architecture, as shown in Figure 3, consists of an input layer, two hidden layers, and an output layer. The input layer has a dimension of 10, the two hidden layers contain 400 and 300 neurons, and the output layer has a dimension of 2. To extract features effectively, and also to prevent gradient saturation and vanishing problems, ReLU is used in the hidden layers and Tanh is used in the output layer.

Figure 4 illustrated the relationship between the Actor network and USV motion controller. During training in the simulation system, the Actor outputs actions based on the input state through its neural network. These actions, representing PID control parameters, are utilized for USV motion control. The control deviation of the USV is used as input to obtain the USV control output variables, such as steering actions in heading control. Consequently, the USV executes steering actions, which are resolved through the rudder force model in the simulation environment, resulting in the USV’s subsequent motion posture.

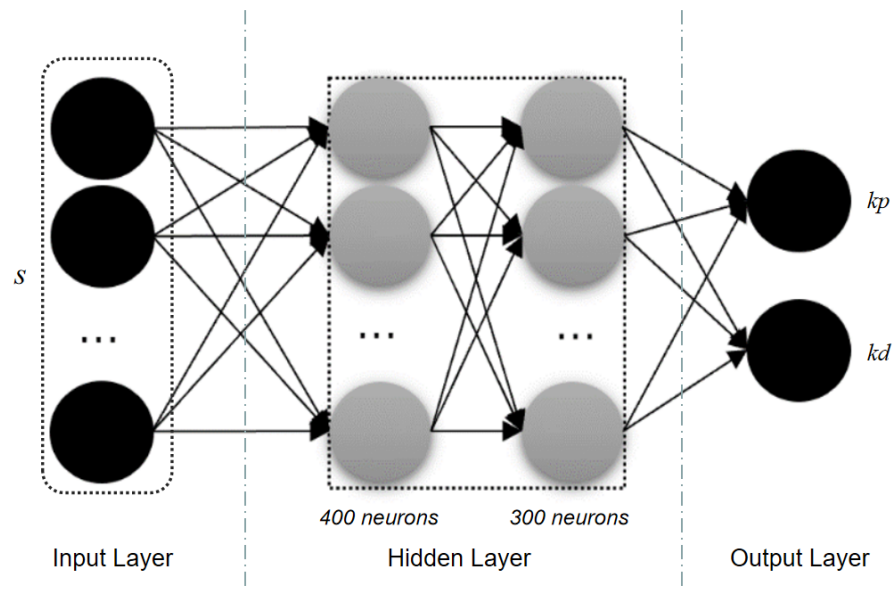


Figure 3. The Actor network.

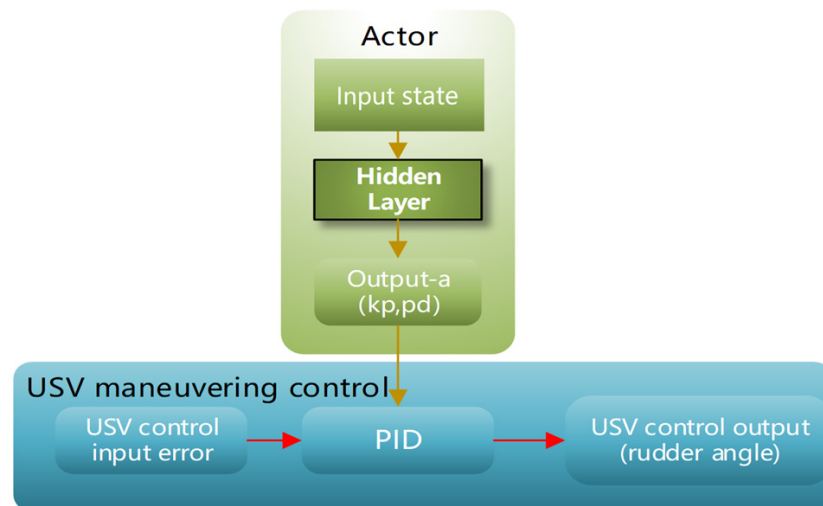


Figure 4. Relationship between the Actor network and USV motion controller.

(2) Target Actor Network

The Target Actor network generates target actions a' , which are fed into the Target Critic network. The structure of the Target Actor mirror that of the current Actor network. Its weights are obtained via soft updates from the current Actor’s weights using Polyak averaging:

$$\theta_{target} \leftarrow \tau\theta_{current} + (1 - \tau)\theta_{target} \tag{16}$$

where θ_{target} represents the target network parameters, $\theta_{current}$ is the current network parameters, and τ is a small coefficient. The arrow symbol used here is used to update θ_{target} . This slow update process maintains the stability and relative independence of the target network, facilitating smoother changes that enhance algorithm stability and convergence.

(3) Current Critic network

The current Critic network evaluates the value of action a in state s . Training the current Critic network requires target values, which are computed using the Target Critic and Target Actor networks. The current Critic is a feedforward neural network; its structure

is depicted in Figure 5. The input is a concatenation of the environmental state s and action a , outputting the state-action value $Q(s, a)$. The hidden layers contain 400 and 300 neurons, and the output layer dimension is 1, representing the state-action value.

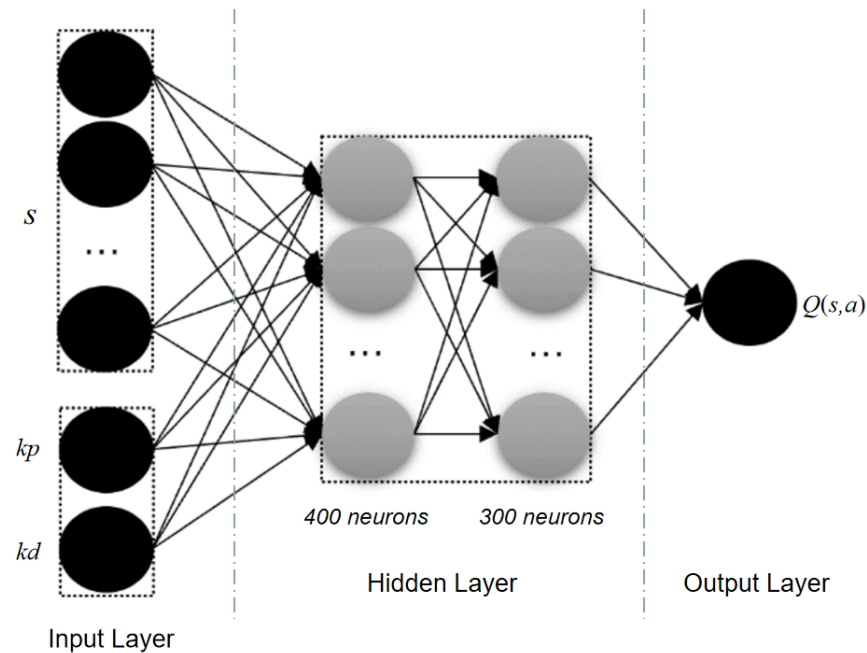


Figure 5. Critic network.

(4) Target Value Network (Target Critic)

The Target Critic network calculates the target Q-value $Q'(s, a)$ and, combined with the actual reward r , computes the TD error, which is used to update the weights of the current Critic network. The structure of the Target Critic network mirrors that of the current Critic but is used for target updates. Its weights are also obtained through soft updates from the current Critic’s weights. The input consists of the environmental state s and action a' , and the output is the state-action value $Q'(s, a)$.

3.2. Ornstein–Uhlenbeck (OU) Noise

OU noise is a time-correlated stochastic process initially used to describe Brownian motion in physics. In RL, it is employed to generate smooth and orderly noise sequences that aid in exploring the action space. The mathematical expression for the OU process is

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t \tag{17}$$

where x_t is the value of the noise at time t , θ is the parameter controlling the speed of noise regression to the mean, μ is the long-term mean of the noise, σ is the intensity of noise fluctuations, and dW_t is the standard Brownian motion.

At each time step, the noise is updated according to the following OU process:

$$x_{t+1} = x_t + \theta(\mu - x_t)\Delta t + \sigma\sqrt{\Delta t}\mathcal{N}(0, 1) \tag{18}$$

where $\mathcal{N}(0, 1)$ is a normal distribution with mean 0 and variance 1. When executing the strategy, the generated OU noise is added to the actions derived from the deterministic strategy.

3.3. Binary Experience Pool Based on Adaptive Batching

To address the slow training speed of the DDPG algorithm, the experience pool is divided into a success experience pool and a failure experience pool. To eliminate correlations between data, an adaptive batch size function is designed, where B experiences

are sampled from each of the success and failure pools. New experience is gained from the environment (including the current state, action taken, reward received, next state, and termination status), and it is added to the buffer. During each current network training session, a random batch of experiences is sampled from the buffer, effectively breaking the temporal correlation between experiences and ensuring that training data are more independently and identically distributed.

$$\sigma = 1 + \frac{n_e}{n_{max}} \tag{19}$$

$$B = \lfloor \sigma n \rfloor \tag{20}$$

In this context, n_e represents the number of training iterations, and n_{max} denotes the total number of training iterations set.

The overall algorithmic process is as follows:

1. The Actor generates a set of actions, to which OU noise is added.
2. The agent obtains the next state based on the current action and inputs it into the reward function. The data from (s_t, a_t, r, s_{t+1}) is categorized into the successful experience pool or the failed experience pool based on success or failure.
3. A sample of n experiences is drawn from the experience pool, and both s_t and s_{t+1} are input into the Actor network, while (s_t, a_t, r, s_{t+1}) are fed into the Critic network, where iterative updates are performed.
4. The Target Actor network receives s_t and s_{t+1} ; inputs action a and random noise into the agent, which interacts with the environment to obtain the next action a_{t+1} ; and outputs it to the Target Critic network. The action values Q are then received to update the network.
5. The Target Critic network receives s_{t+1} and a_{t+1} and calculates the Q-values. These values are then combined with the rewards to compute the labels used for iterative network updates. See Figure 6.

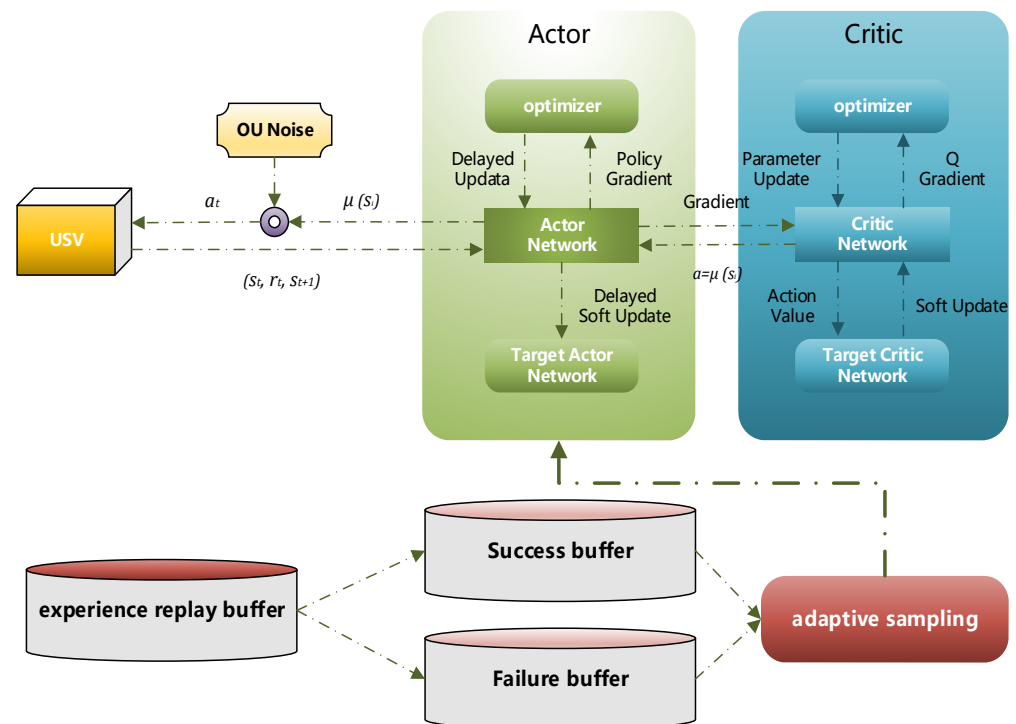


Figure 6. The flowchart of the improved DDPG.

3.4. Reward Definition and Analysis

Based on the maneuvering characteristics of USVs and the features of path tracking control, the reward function is defined as follows:

$$r_{psi} = \begin{cases} 0, & e_{psi} \leq 0.1 \\ -e_{psi} - 0.1e_{psi_last}, & e_{psi} > 0.1 \end{cases} \quad (21)$$

$$r_{ye} = \begin{cases} 0, & y_e \leq 0.1 \\ -0.1, & y_e > 0.1 \end{cases} \quad (22)$$

$$r = r_{ye} + r_{psi} \quad (23)$$

The reward function comprehensively considers both the lateral deviation and heading angle deviation in trajectory tracking, with the expectation that both deviations remain minimal. According to the reward setting, if the heading angle deviation is less than 0.1, the reward is 0. Conversely, if the heading angle deviation exceeds 0.1, a negative reward is generated, with the magnitude of the negative reward positively correlated with the consecutive heading angle deviations. This design aims to use RL to reduce the heading angle deviation. For lateral deviation, the requirement is to keep it within 1 m. Within this range, the lateral deviation reward is 0, but if it exceeds 1 m, a negative reward of -0.1 is applied, aiming to reduce lateral deviation through RL.

Below is an analysis of typical scenarios based on this reward setting (see Figure 7):

- (a) $y_e > 1$ [m], $e_{psi} > 0.1$ [rad]: As illustrated in the scenario, both the lateral and heading angle deviations receive significant negative rewards (on the order of -0.1). During training, actions receiving nearly 0 rewards will be emphasized, guiding the USV to alter its heading angle toward the LOS target point and reduce lateral error. According to the reward setting, the priority between altering the motion direction toward the LOS target point and approaching the tracking trajectory to reduce lateral deviation is dynamically adjusted.
- (b) $y_e > 1$ [m], $e_{psi} < 0.1$ [rad]: In this scenario, the lateral deviation receives a significant negative reward (on the order of -0.1), while the negative reward for heading angle deviation is minor (close to 0). During training, actions with nearly 0 rewards are reinforced, guiding the USV to approach the tracking trajectory to reduce lateral error without excessively adjusting the heading angle, which could lead to increased lateral deviation and negative rewards for heading angle deviation. According to the reward setting, the priority between turning toward the LOS target point and approaching the tracking trajectory is dynamically adjusted.
- (c) $y_e < 1$ [m], $e_{psi} > 0.1$ [rad]: Here, the lateral deviation receives a minor negative reward (close to 0), while the heading angle deviation incurs a significant negative reward (on the order of -0.1). During training, actions yielding nearly 0 rewards are emphasized, guiding the USV to adjust its heading angle toward the LOS target to reduce the heading angle deviation. However, due to the inertia of USV motion and the narrow 1 m reward boundary for lateral deviation, if the heading angle deviation is substantial, the USV might overshoot the 1 m boundary during adjustment, resulting in oscillation around the tracking trajectory. To avoid this, the USV should not have excessive heading angle deviation when entering this scenario.
- (d) $y_e < 1$ [m], $e_{psi} < 0.1$ [rad]: In this case, both lateral and heading angle deviations receive minor negative rewards (close to 0). The USV's trajectory tracking control tends to stabilize, meeting the tracking requirements and maintaining this condition despite wave disturbances. However, if the USV enters this scenario and maintains stability, there may exist a steady-state lateral error of less than 1 m. With 0 rewards for both lateral and heading angle deviations, this steady-state error might not be corrected. This issue can be addressed by dynamically changing the LOS target point, but this requires a longer training time.

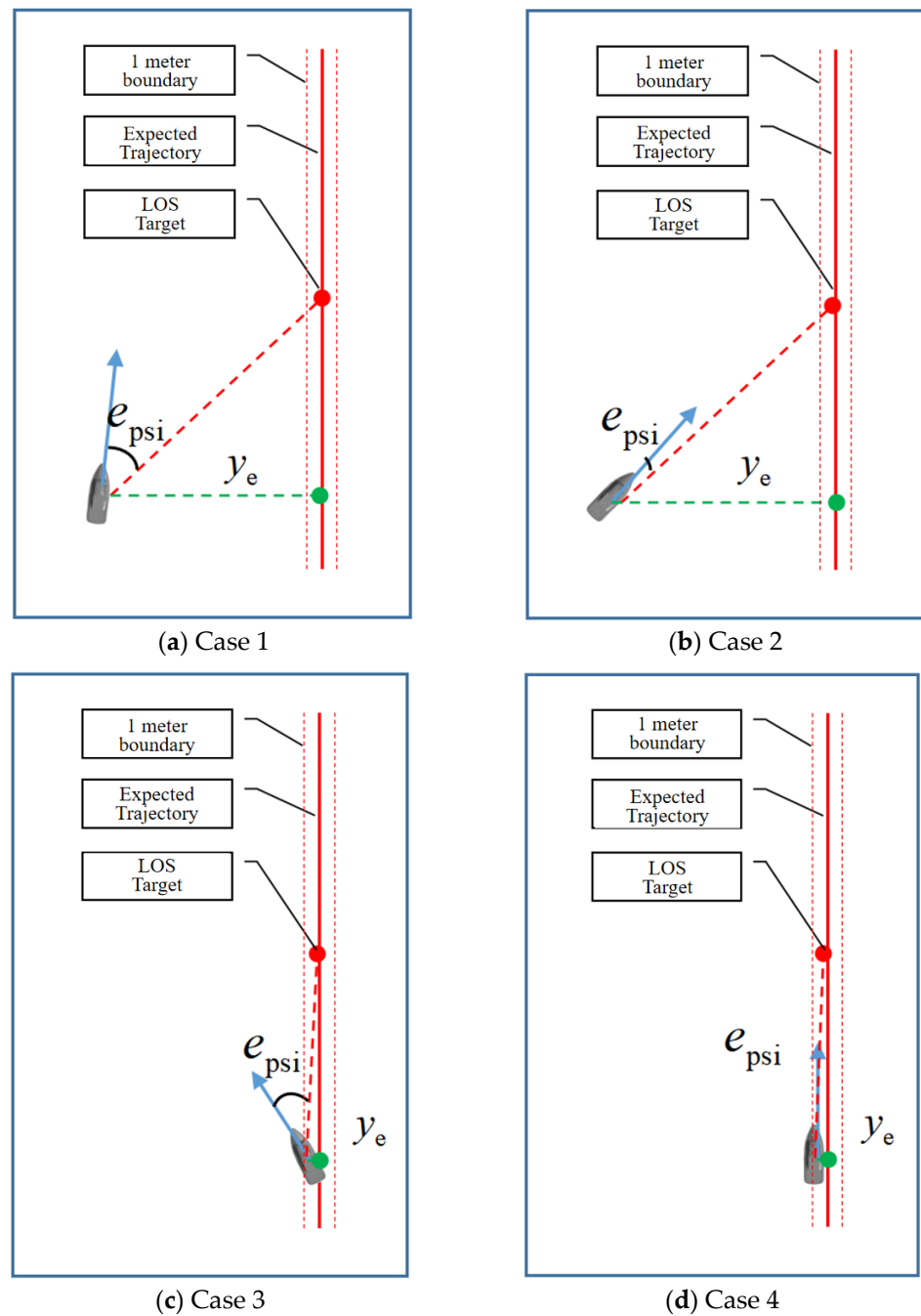


Figure 7. Typical scenarios based on this reward setting.

4. Simulation Results and Analysis

To validate the effectiveness of the proposed improved DDPG-based PID control algorithm for USV trajectory tracking, this study conducts a comprehensive comparison with both an adaptive PID control algorithm and a standard DDPG-based PID control algorithm. The hardware used for model training includes a desktop computer equipped with an Intel Core i5-12700 CPU (Intel, Santa Clara, CA, USA) and 8 GB of RAM. The software environment consists of the Windows 11 operating system, TensorFlow version 2.1.19 (GPU), and Python version 3.9.1. During the adaptive navigation control, the RL parameter tuner was trained for a total of 1500 steps per episode, with a sampling time of 0.1 s. The desired trajectory was set as a straight line parallel to the y -axis, and a total of 2000 episodes were trained. Each episode terminated either upon reaching the specified number of training steps or upon arriving at the designated end point on the straight line.

The simulation environment was designed to include scenarios with wave disturbances. To assess the system’s stability and resilience against external disturbances, a transverse force of $[-0.2 \times 10^3, 0.2 \times 10^3]$ [N] and a turning moment of $[-0.2 \times 10^3, 0.2 \times 10^3]$ [N·m] were introduced as perturbations. This is modeled as $f(x) = 0.2U(t)$, where $U(t)$ represents an independent random variable at each time point, following a uniformly distributed random function $U(t) \sim f_u(-1, 1)$, with $f_u()$ denoting the random distribution. The applied transverse disturbance force is depicted in Figure 8, while the turning disturbance moment is illustrated in Figure 9.

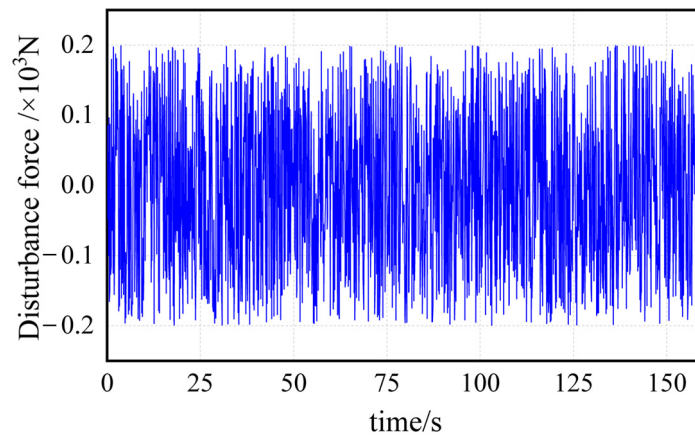


Figure 8. Transverse disturbing force.

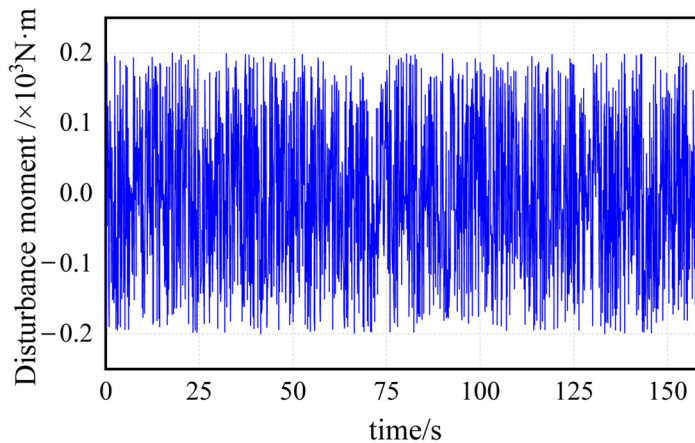


Figure 9. Turning disturbing moment.

4.1. Comparative Analysis of Zigzag Trajectory Tracking

Figure 10 presents the results of Zigzag trajectory tracking under wave disturbance using adaptive PID algorithms, PID parameters calculated by the DDPG algorithm, and PID parameters determined by the improved DDPG algorithm. The trajectory path points are set as polylines between the coordinates (0, 0), (45, 45), (85, 5), and (125, 45). The figures indicate that, due to disturbances, all three algorithms exhibit significant overshoot in the initial stage. However, the USV is eventually able to adjust to the desired trajectory. During target point transitions, each algorithm demonstrates effective control performance, with lateral deviation errors fluctuating within acceptable limits and heading angle deviations rapidly diminishing until stabilization; the rudder angle changes are also smooth. Notably, both RL algorithms outperform the adaptive PID controller in managing lateral deviations and heading tracking. Moreover, the method employing PID parameters calculated using the improved DDPG algorithm results in a smoother desired heading angle, indicating superior tracking performance. Its steering curve is also more stable. See also Table 4.

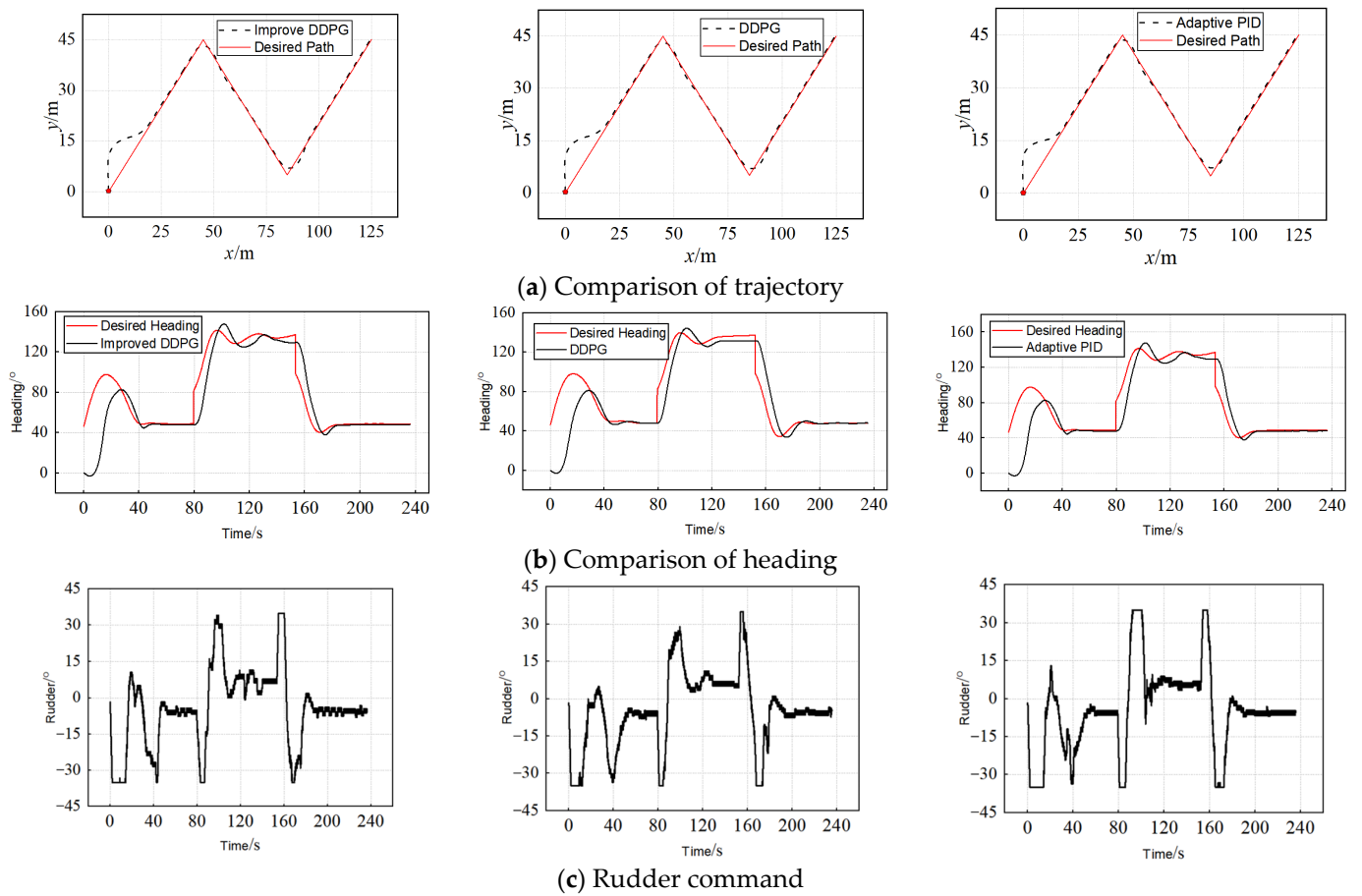


Figure 10. Comparison of Zigzag tracking with PID parameter adjustment using different algorithms under disturbance of wave.

Table 4. Comparison of Zigzag trajectory tracking metrics.

Controller	Stable Mean Lateral Error/m	Relative Cross-Track Error	Mean Heading Angle Tracking Deviation/ ^o
Improved DDPG and PID	0.109	0.245	3.601
DDPG and PID	0.388	0.874	3.952
Adaptive PID	0.985	2.218	6.584

4.2. Comparative Analysis of Curved Trajectory Tracking

Figure 11 illustrates the comparison of tracking a circular trajectory with a radius of 10 m under sea state 4 with a 90 degree wave, using the three different aforementioned algorithms for PID parameter calculating. It is evident that the USV can quickly adjust to the desired trajectory, demonstrating effective control performance. The lateral deviation errors fluctuate within acceptable limits, and the heading angle deviations rapidly decrease and stabilize. The changes in rudder angle are also smooth. Both RL algorithms outperform the adaptive PID controller in managing lateral deviation and heading tracking. When using the adaptive PID controller and the controller based on DDPG-calculated PID parameters, significant initial mistakes can lead to difficulties in quickly eliminating errors. Combined with the inertia of the USV and wave disturbances, this can result in oscillations. In contrast, the improved DDPG algorithm for PID control dynamically adjusts control parameters according to the trend in tracking error changes, reducing oscillations near the trajectory and achieving better control performance. The resulting desired heading angle and steering curve are smoother, indicating that this controller is highly adaptable and robust, providing superior tracking performance. See also Table 5.

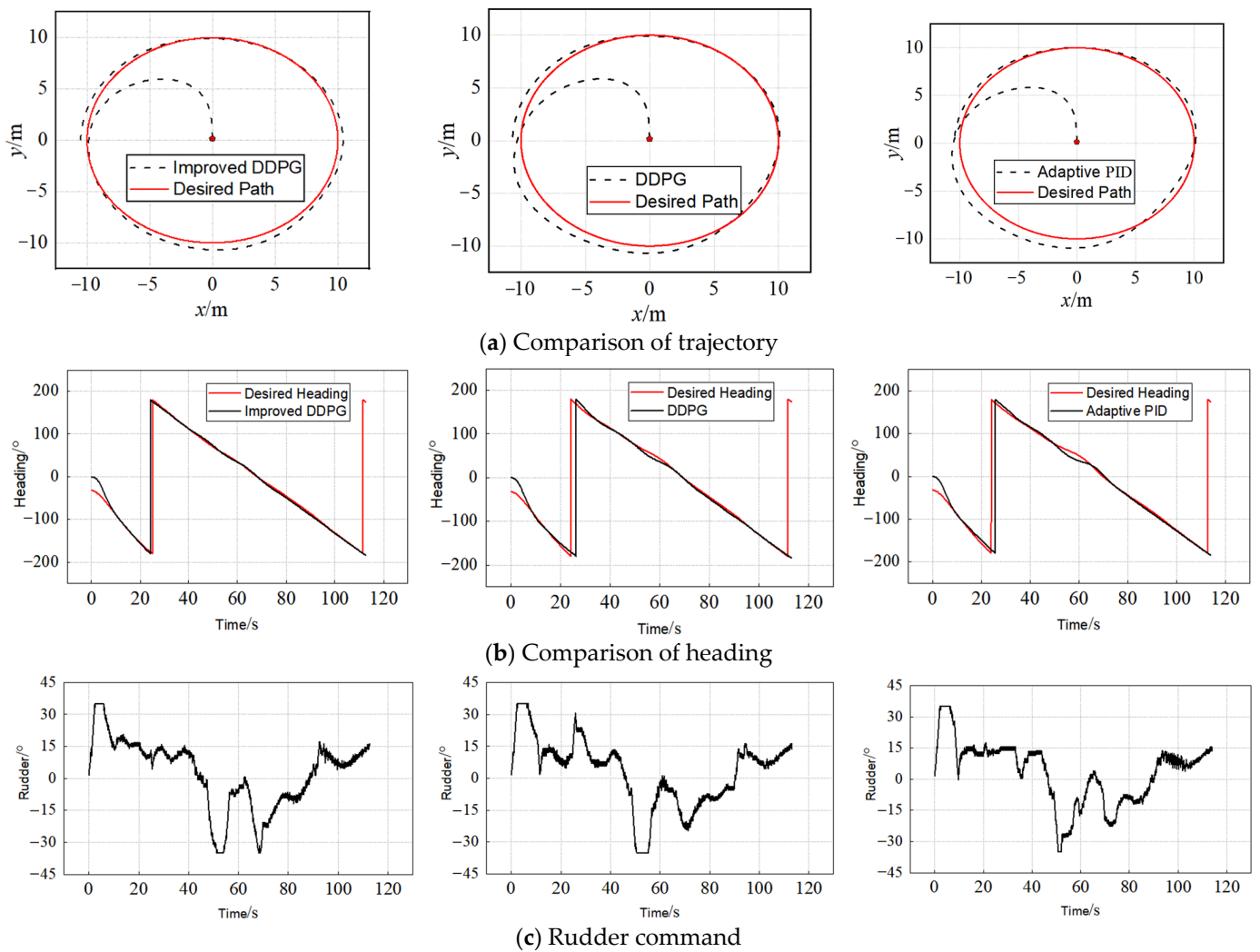


Figure 11. Comparison of circle trajectory tracking with PID parameter adjustment using different algorithms under wave disturbance.

Table 5. Comparison of circle trajectory tracking metrics.

Controller	Stable Mean Lateral Error [m]	Relative Cross-Track Error	Mean Heading Angle Tracking Deviation [deg]
Improved DDPG and PID	0.402	0.905	1.98
DDPG and PID	0.568	1.279	2.56
Adaptive PID	0.652	1.468	2.98

5. Experimental Results and Analysis

It is challenging to conduct curve tracking experiments due to the wave maker located in the ship towing tank, which has a limited width. Therefore, this study conducted Zigzag trajectory tracking control experiments under wave conditions only. The positioning equipment used was an indoor UWB system, and the USV was equipped with a gyroscope for heading, a servo motor for propeller adjustment, and a rudder servo for steering. The main control board was an STM32, facilitating wireless serial communication with the host computer. The host computer sent commands to the model ship, and the USV transmitted real-time sensor data back to the host for display. Simultaneously, the host computer performed control calculations based on the collected data, utilizing the proposed improved DDPG algorithm for calculating PID parameters.

The experiment involved six irregular trajectory points. Initially, the propeller speed was adjusted in still water to achieve a USV speed of 0.8 [m/s], corresponding to a Froude number of 0.21. The USV's heading was adjusted based on its real-time position relative to the first target point of the trajectory. Once the wave-making machine established a stable waveform with the desired amplitude and wavelength, intelligent control was activated on the host computer. The USV dynamically adjusted its rudder angle according to the algorithm, ensuring movement along the trajectory points. The aforementioned method was used to measure and record the motion data of the USV. The experimental process is depicted in Figure 12, and the six trajectory points along with the USV's motion trajectory are shown in Figure 13.

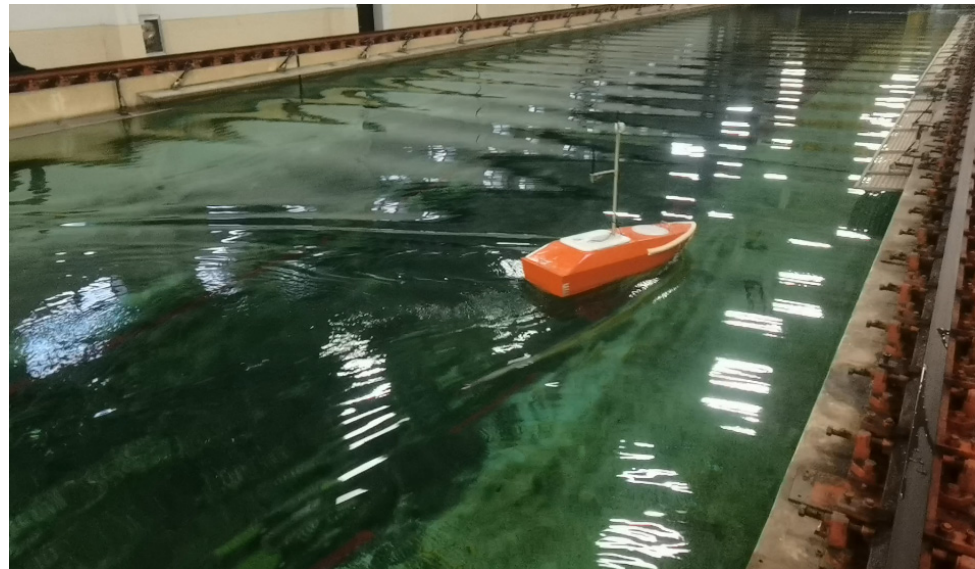


Figure 12. Path tracking control experiment of USV in waves.

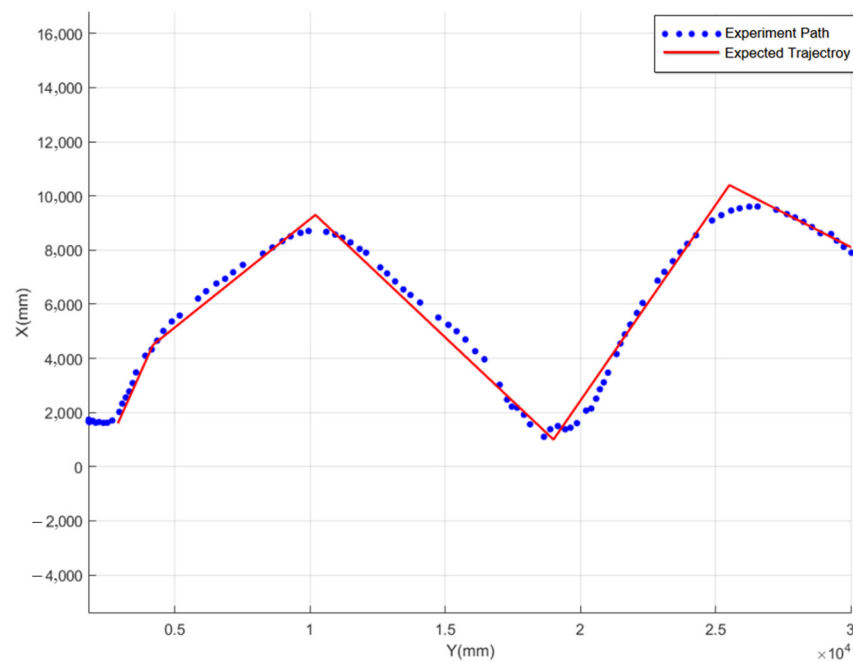


Figure 13. Motion trajectory of USV path tracking control in waves.

The experiment demonstrated that, despite wave disturbances, the USV effectively tracked all trajectory points with a smooth path, achieving a maximum trajectory error

of approximately 200 [mm]. This error is small and is primarily attributed to deviations in trajectory point acquisition caused by the UWB positioning accuracy and the resulting control deviations. Under the current hardware conditions and operating modes, the positioning accuracy was optimized using filtering techniques; however, it is still affected by equipment precision, UWB positioning methods, and electromagnetic interference from water waves and metal tracks in the tank. Despite these challenges, the experiment and error analysis indicate that the intelligent matching technique for control parameters based on the proposed method achieves an excellent control accuracy under existing conditions.

Figure 14 illustrates the real-time feedback of the rudder angle and the measured heading angle during the experiment. The changes in heading reveal five major trend adjustments required for path tracking, corresponding to the five segments in the actual trajectory. The heading changes were smooth, with a maximum rudder angle of 50 degrees at sharp turns along the target trajectory. The adaptive control yielded favorable results.

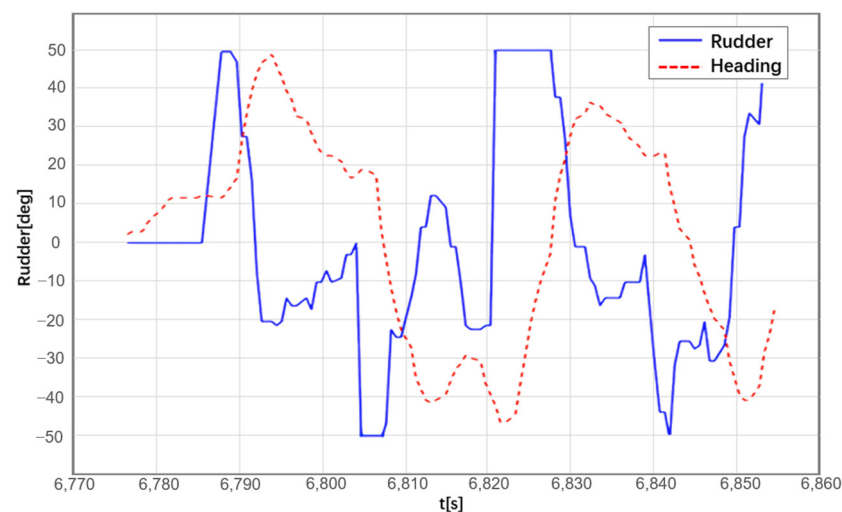


Figure 14. Rudder angle and heading angle curves in USV path tracking control experiment in waves.

6. Conclusions

In this study, we present a novel algorithm that integrates DDPG with a PID controller to achieve path following in the presence of complex conditions, particularly wave interference. The algorithm begins by leveraging a 3-DOF MMG maneuvering motion model, which serves as the foundation for subsequent DDPG training. CFD simulations and regression analysis are employed to extract hydrodynamic derivatives and interaction coefficients necessary for precise motion prediction. We further detail the LOS guidance strategy, which directs the USV to follow a virtual target along the desired trajectory, with PID parameters adjusted dynamically via the DDPG framework.

The design of both the Actor and Critic networks is carefully structured, and to address the issue of slow training speeds in DDPG, we implement a dual-experience pool, separating the successful and failed experiences. Additionally, an adaptive batch size function is introduced to minimize data correlations, further enhancing the training efficiency. The reward function is rigorously formulated to account for both the lateral deviation and heading angle deviation, and its effectiveness is thoroughly examined under various operating conditions.

Simulations and experimental trials involving path following in Zigzag and turning maneuvers, conducted under wave disturbance, demonstrate the algorithm's robust performance. The USV maintained superior tracking accuracy, even under continuously varying wave directions, highlighting the algorithm's strong generalization capabilities and robustness.

Two key practical insights emerge from this work: First, the desired trajectory must align with the USV's maneuvering limits. Trajectory points that exceed the USV's steer-

ing capabilities, especially under full rudder conditions, will render the target trajectory unattainable in the presence of external disturbances. This not only compromises tracking performance but also hinders the Reinforcement Learning process. Second, while improving tracking accuracy, setting excessively small thresholds for error in the reward function can lead to instability in the neural network. Lowering the static error threshold makes successful experiences more difficult to achieve, thereby slowing the agent's learning rate and significantly increasing the training time.

Author Contributions: Data curation, C.X.; funding acquisition, J.X. and L.S.; methodology, L.S.; software, C.X. and H.Y.; visualization, C.X.; writing—original draft, C.X. and X.W.; writing—review and editing, X.W. and L.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant numbers 51809203.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Nomenclature

USV	Unmanned Surface Vehicles
DDPG	Deep Deterministic Policy Gradient
PID	Proportion Integration Differentiation
RL	Reinforcement Learning
ASVs	Autonomous Surface Vessels
GA	Genetic Algorithms
GPC-PID	PID cascade controller based on Generalized Predictive Control
RBF	Radial Basis Function
UUV	Unmanned Underwater Vehicle
DQN	Deep Q Networks
MDP	Markov Decision Process
DRL	CFD: Computational Fluid Dynamics
MMG	Maneuvering Modeling Group
OU	Ornstein–Uhlenbeck
LOS	Line Of Sight

References

- Do, K.D. Practical control of underactuated ships. *Ocean Eng.* **2010**, *37*, 1111–1119. [\[CrossRef\]](#)
- Wei, L.; Hui, Y. Super-Twisting Sliding Mode Control for the Trajectory Tracking of Underactuated USVs with Disturbances. *J. Mar. Sci. Eng.* **2023**, *11*, 636. [\[CrossRef\]](#)
- Sun, J.; Wang, N.; Meng, J.E.; Liu, Y. Extreme learning control of surface vehicles with unknown dynamics and disturbances. *Neurocomputing* **2015**, *167*, 535–542. [\[CrossRef\]](#)
- Aguiar, A.P.; Hespanha, J.P. Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty. *IEEE Trans. Autom. Control* **2007**, *52*, 1362–1379. [\[CrossRef\]](#)
- Fahimi, F.; Van Kleeck, C. Alternative trajectory-tracking control approach for marine surface vessels with experimental verification. *Robotica* **2013**, *31*, 25–33. [\[CrossRef\]](#)
- Harmouche, M.; Laghrouche, S.; Chitour, Y. Global tracking for underactuated ships with bounded feedback controllers. *Int. J. Control* **2014**, *87*, 2035–2043. [\[CrossRef\]](#)
- Katayama, H.; Aoki, H. Straight-line trajectory tracking control for sampled-data underactuated ships. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 1638–1645.
- Awad, N.; Lasheen, A.; Elnaggar, M.; Kamel, A. Model predictive control with fuzzy logic switching for path tracking of autonomous vehicles. *ISA Trans.* **2022**, *129*, 193–205. [\[CrossRef\]](#)
- Chen, H.; Tang, G.; Wang, S.; Guo, W.; Huang, H. Adaptive fixed-time backstepping control for three-dimensional trajectory tracking of underactuated autonomous underwater vehicles. *Ocean Eng.* **2023**, *275*, 114109. [\[CrossRef\]](#)

10. Li, Z.F.; Lei, K. Robust Fixed-Time Fault-Tolerant Control for USV with Prescribed Tracking Performance. *J. Mar. Sci. Eng.* **2024**, *12*, 799. [[CrossRef](#)]
11. Zhao, B.; Zhang, X.; Liang, C.; Han, X. An improved model predictive control for path-following of USV based on global course constraint and event-triggered mechanism. *IEEE Access* **2021**, *9*, 79725–79734. [[CrossRef](#)]
12. Hu, Z.; Zhou, H. The course control based on an on-line self-adjusted PID control algorithm for unmanned surface vehicles. *Robot* **2013**, *35*, 263–268. [[CrossRef](#)]
13. Liu, S.; Fang, L.; Ge, Y.M.; Fu, H.X. GA-PID adaptive control research for ship course-keeping system. *J. Syst. Simul.* **2007**, *19*, 3783–3786.
14. Ouyang, Z.; Yu, W. PID Control with Improved Genetic Algorithm for Ship Steering. *Navig. China* **2017**, *40*, 13–16.
15. Liu, S.; Xing, B.; Zhu, W. A fusion fuzzy PID controller with real-time implementation on a ship course control system. In Proceedings of the 2015 23th Mediterranean Conference on Control and Automation, Torremolinos, Spain, 16–19 June 2015.
16. Fan, Y.; Li, C.; Wang, G.; Guo, C.; Zhao, Y. Design and validation of course tracking controller for unmanned surface vehicle. *J. Dalian Marit. Univ.* **2017**, *43*, 1–7.
17. Sharma, A.; Zheng, Q.; Noel, M.M. Active disturbance rejection control for cargo ship steering. In Proceedings of the American Control Conference, Chicago, IL, USA, 1–3 July 2015; pp. 3956–3961.
18. Peng, Y.; Wu, W. USV Tracking Control Based on Cascade GPC-PID. *Control Eng. China* **2014**, *21*, 245–248.
19. Huang, H.; Gong, M.; Zhuang, Y.; Sharma, S.; Xu, D. A new guidance law for trajectory tracking of an underactuated unmanned surface vehicle with parameter perturbations. *Ocean Eng.* **2019**, *175*, 217–222. [[CrossRef](#)]
20. Wang, N.; Gao, Y.; Yang, C.; Zhang, X. Reinforcement learning-based finite-time tracking control of an unknown unmanned surface vehicle with input constraints. *Neurocomputing* **2022**, *484*, 26–37. [[CrossRef](#)]
21. Wu, C.; Yu, W.; Li, G.; Liao, W. Deep reinforcement learning with dynamic window approach based collision avoidance path planning for maritime autonomous surface ships. *Ocean Eng.* **2023**, *284*, 115208. [[CrossRef](#)]
22. Wang, R.; Shen, Z. Fuzzy adaptive iterative sliding mode control for sail-assisted ship trajectory tracking. In Proceedings of the 2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS), Dalian, China, 24–26 July 2017.
23. Yang, X.; Yan, X.; Liu, W.; Ye, H.; Du, Z.; Zhong, W. An improved stanley guidance law for large curvature path following of unmanned surface vehicle. *Ocean Eng.* **2022**, *266*, 112809. [[CrossRef](#)]
24. Bertaska, I.R.; Von, E. Experimental evaluation of supervisory switching control for unmanned surface vehicles. *IEEE J. Ocean Eng.* **2018**, *44*, 7–28. [[CrossRef](#)]
25. Magalhães, J.; Damas, B.; Lobo, V. Reinforcement learning: The application to autonomous biomimetic underwater vehicles control. *IOP Conf. Ser. Earth Environ. Sci.* **2018**, *172*, 12–19. [[CrossRef](#)]
26. Bian, X.Q. Adaptive Neural Network Control System of Path Following for AUVs. In Proceedings of the 2012 Proceedings of IEEE Southeastcon, Orlando, FL, USA, 15–18 March 2012.
27. Xiaofei, Y.; Yilun, S.; Wei, L.; Hui, Y.; Weibo, Z.; Zhengrong, X. Global path planning algorithm based on double DQN for multi-tasks amphibious unmanned surface vehicle. *Ocean Eng.* **2022**, *266*, 112809. [[CrossRef](#)]
28. Wu, C.; Yu, W. Deep reinforcement learning with intrinsic curiosity module based trajectory tracking control for USV. *Ocean Eng.* **2024**, *308*, 118342. [[CrossRef](#)]
29. Woo, J.; Yu, C.; Kim, N. Deep reinforcement learning based controller for path following of an unmanned surface vehicle. *Ocean Eng.* **2019**, *183*, 155–166. [[CrossRef](#)]
30. Song, L.; Xu, C.; Hao, L.; Yao, J.; Guo, R. Research on PID Parameter Tuning and Optimization Based on SAC-Auto for USV Path Following. *J. Mar. Sci. Eng.* **2022**, *10*, 1847. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.