sciendo

POLISH MARITIME RESEARCH 1 (117) 2023 Vol. 30; pp. 54-64
10.2478/pomr-2023-0006

ADPTIVE HEADING CONTROL OF UNDERACTUATED UNMANNED SURFACE VEHICLE BASED ON IMPROVED BACKPROPAGATION NEURAL NETWORK

Zaopeng Dong * [1, 2, 3]
Jiakang Li [1,3]
Wei Liu [4]
Haisheng Zhang [1, 3]
Shijie Qi [1,3]
Zhengqi Zhang [1, 3]

[1] Key Laboratory of High Performance Ship Technology (Wuhan University of Technology), Ministry of Education, Wuhan University of Technology, Wuhan, China
[2] Science and Technology on Underwater Vehicle Laboratory, Harbin Engineering University, Harbin, China
[3] School of Naval Architecture, Ocean and Energy Power Engineering, Wuhan University of Technology, Wuhan, China
[4] China Institute of Marine Technology&Economy, Beijing, China

* Corresponding author: *dongzaopeng@whut.edu.cn (Zaopeng Dong)*

ABSTRACT

Aiming at the challenges to the accurate and stable heading control of underactuated unmanned surface vehicles arising from the nonlinear interference caused by the overlay and the interaction of multi interference, and also the uncertainties of model parameters, a heading control algorithm for an underactuated unmanned surface vehicle based on an improved backpropagation neural network is proposed. Based on applying optimization theory to realize that the underactuated unmanned surface vehicle tracks the desired yaw angle and maintains it, the improved momentum of weight is combined with an improved tracking differentiator to improve the robustness of the system and the dynamic property of the control. A hyperbolic tangent function is used to establish the nonlinear mappings an approximate method is adopted to summarize the general mathematical expressions, and the gradient descent method is applied to ensure the convergence. The simulation results show that the proposed algorithm has the advantages of strong robustness, strong anti-interference and high control accuracy. Compared with two commonly used heading control algorithms, the accuracy of the heading control in the complex environment of the proposed algorithm is improved by more than 50%.

Keywords: underactuated unmanned surface vehicle; backpropagation neural network controller; heading control; hyperbolic tangent function

INTRODUCTION

With the development of the unmanned surface vehicle (USV), the intelligence of USV motion control has been gaining increasing attention. Thanks to its strongly autonomous navigation capability, environmental adaptability and modular design advantage, it has been widely used in hydrologic reconnaissance, maritime search and rescue, and navigation of formation and other fields [1-6]. To solve the nonlinear interference problem caused by the overlay and the interaction of multi interference, and the poor robustness of control methods due to the uncertainty of the model parameters [7-8], active disturbance rejection control [9-10] (ADRC), fuzzy control [11-12], backstepping control [13-15], sliding mode control [16-21] (SMC), and proportion integration differentiation (PID) control [22-24], combined with intelligent algorithms and artificial neural networks [25-27] (ANN), have been introduced by scholars.

A new robust model predictive control (MPC) algorithm for trajectory tracking of an autonomous surface vehicle is proposed by Esfahani [28]. A sliding mode control-based procedure for designing the MPC and a super-twisting term are adopted to fulfill the robustness property. A hierarchical control framework associated with control algorithms for the USV swarm is proposed by Zhuang [29]. The control framework is divided into three task

layers to implement the distributed control of the autonomous swarm. With the help of the Lyapunov method, the motion controller is proved asymptotically stable. Kula [30] presented a search operation support system, equipped with an IMC control system, which can provide highly accurate control. The simulation tests indicated that precise search work can be done using less time. Zhang [31] improved the PID control to an adaptive self-regulation PID (APID) scheme by using the Lyapunov direct method. The APID control scheme, which can ensure the boundedness of all signals in the ship course-keeping control system, is effective and robust. Besides, under the continuous development of intelligent control, intelligent control itself has gradually been attracting increasing interest among scholars.

Benefiting from the strongly robust and flexible controller design, neural network control has been applied in the field of motion control of USVs. A fuzzy dynamic surface control controller was designed by Wang [32], who merged the fuzzy method with the dynamic surface control to realize the ship's heading tracking. The Nussbaum function is applied to handle the unknown control direction without any prior knowledge. Combined ADRC and radial basis function (RBF) neural networks observed state values using an extended state observer, and the RBF neural networks were used to optimize the parameters in the ADRC control in a RBF-ADRC controller designed by Liu [33]. Yi [34] applied PID feedback control to the USV path following, and a control strategy containing heading control and velocity control is designed based on the line of sight algorithm. Separately from the heading control, the desired velocity is tracked.

However, those authors used neural networks to optimize the control parameters and did not consider the complex environmental disturbance. Rihem [35] integrated image sensing with motion control by introducing the result of perception into the navigation algorithm. A PID controller was used to accomplish the steering and change of speed. Du [36] proposed a safe Lyapunov boundary deep deterministic policy gradient algorithm of USV interception, pre-training the deep policy network by the proportional adaptive control combined with the line of sight algorithm. Comparing LQG control with adaptive PID control, a LQG controller combined with Kalman filtering was proposed by Asfihani [37]. It was concluded that the LQG controller combined with the Kalman filtering method is superior to the adaptive PID control when the PID control parameters are obtained by recursive least squares (RLS). Wang [38] presented an event-triggered adaptive control to handle the consensus problem when one-leader multi-agent systems were attacked. The RBF neural network was used to evaluate the actor which consisted of sliding mode control and adaptive reinforcement learning control. A formation control algorithm based on deep reinforcement learning and a leader follower that set a reward function related to the error of the follower's distance and the follower's velocity, together with a random braking mechanism, were designed by Zhao [39]. The local optimum was solved by a random barking mechanism, and the design of an efficient reward function sped up the convergence, enabling the formation to get out of disruption quickly. But those authors do not give the mathematical expression of the controlled variables, or examine the motion state of the ship.

With the complex environmental disturbance taken into consideration and adaptive control parameters, mathematical expressions of control are established, also the restriction of the motion state of the ship, and a backpropagation neural network (BPNN) controller is designed for heading control. Compared with existing studies, the main contributions of this paper can be summarized as follows: (i) we present a novel control algorithm for the heading control problem, in which the desired yaw angle of the USV is transformed into a minimized cost function which is related to the actual state and desired state, which can simplify the controller design; (ii) we propose an improved adaptive momentum of weight for the gradient descent, which helps the neural network get out of local optima and speeds up the convergence, guaranteeing that the controller can adapt online and finally converge stably; (iii) we devise an improved tracking differentiator for extracting state variables, weaken the noise amplification effect and implement the transition process; (iv) we design a novel controller for the heading control of the USV by combining the backpropagation neural network and optimization theory, applying the tanh function to build a nonlinear mapping, and an approximate strategy is adopted in the design process of the control algorithm to summarize the general mathematical expressions.

The remainder of this paper is organized as follows. In Motion Model, the motion model of the USV is described. In Controller Design, an adaptive control method is proposed and a backpropagation neural network controller is designed. To verify the effectiveness of the proposed control algorithm, numerical simulations, results and analysis are presented in Simulation Test Design, and Conclusion contains the conclusion.

MOTION MODEL

As the basis of motion control, the selection of the motion response model and model parameters is key. According to the literature [40], the second-order nonlinear KT equation can be described as follows:

$$T_1 T_2 \ddot{r} + (T_1 + T_2)\dot{r} + r + \alpha r^3 = K T_3 \dot{\delta} + K(\delta + \delta_r) \quad (1)$$

In Eq. (1), T_1, T_2, and T_3 are time parameters which are related to the rudder ability and heading stability of the USV, K is the rudder angle gain coefficient, which is related to the rotationality of the USV, α is a constant coefficient of the nonlinear part, δ_r is the critical rudder angle. Conventionally, ψ represents the yaw angle, r is the heading rate and δ is the rudder angle.

CONTROLLER DESIGN

In response to the heading control problems raised in the Introduction, the control algorithm is introduced based on the motion model described in Motion Model. The heading control system is set to obtain the status information and target

information to input into the BPNN controller, and the output of the BPNN controller, which is called the control variable, is the input of the USV motion control module. The block diagram of the system is shown in Fig. 1 below.

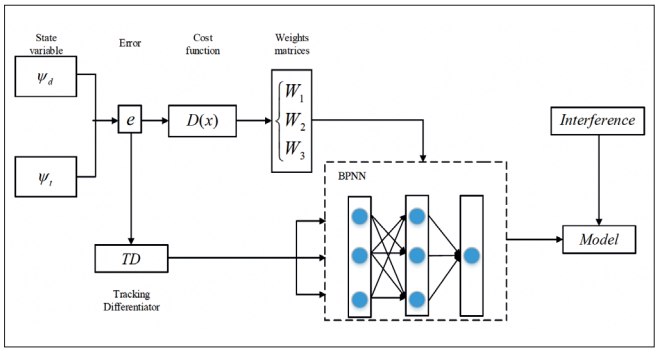

Fig. 1. Block diagram of control system

In the design of the controller, a tracking differentiator extracts the differential signal that is used in the controller, which is shown in part 1, and part 2 illustrates the main process of backpropagation neural network control, in which the structure of the controller and backpropagation algorithm are discussed. In part 3, the convergence of the proposed control strategy is analysed with the help of optimization theory and the Taylor formula.

A. TRACKING DIFFERENTIATOR

To solve the noise effect during differential calculations, the tracking differentiator is used to approximate differentiation. Consider a second-order integrator series system:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u, |u| \le \beta \end{cases} \tag{2}$$

According to motion theory, speed-up then slow-down is required for matching the motion state in the least time. In this way, the pivotal element of the fast optimal control function is when to accelerate and when to decelerate. For a problem that eventually stabilizes at the origin, based on Newton's second laws of motion, the motion state depends on $x_1+x_2|x_2|/(2\beta)$. Thus, its fast optimal control function can be concluded based on the literature [9]:

$$u(x_1, x_2) = \beta sign(x_1 + \frac{x_2|x_2|}{2\beta}) \tag{3}$$

Replacing x_1 with $x_1-v_0(t)$, so that x_1 can track $v_0(t)$, and x_2 can be used as the differential of $v_0(t)$. To smooth the transition process, the signum function is replaced by a hyperbolic tangent function, and the actual problems must be discontinuous, so the discretization of (2) can be expressed as:

$$\begin{cases} x_1(k+1) = x_1(k) + h_0 x_2(k) \\ x_2(k+1) = x_2(k) + h_0 f \\ f = -\beta \tanh\left(x_1(k) - v(k) + x_2(k)|x_2(k)|/(2\beta)\right) \end{cases} \tag{4}$$

However, this is only an approximate numerical solution for discrete systems, so the improved discrete fast optimal control function and tracking differentiator are introduced as follows:

$$\begin{cases} x_1(k+1) = x_1(k) + h_0 x_2(k) \\ x_2(k+1) = x_2(k) + h_0 fh \\ fh = fhan\left(x_1(k) - v(k), x_2(k), \beta, h\right) \end{cases} \rightarrow$$

$$\rightarrow \begin{cases} d = \beta, h \\ d_0 = dh \\ a_0 = hx_2 \\ y = x_1 + a_0 \\ a_1 = \sqrt{d(d+8|y|)} \\ a_2 = a_0 + \tanh(y)(a_1 - d)/2 \\ a = (a_0+y)fth(y, d) + a_2(1 - fth(y, d)) \\ fhan = -a\beta fth(a, d)/d - \beta \tanh(a)(1 - fth(a, d)) \end{cases} \tag{5}$$

In (5), $fth(x, y)=(\tanh(x+y)-\tanh(x-y)/2$. The hyperbolic tangent function can be described as follows:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{6}$$

So, the discretization of the improved tracking differentiator is established, h is a filter factor, h_0 is time-step, and β is a rapidity factor.

B. BPNN CONTROLLER

A BPNN with double hidden layers is used as the controller and a three-layer structure neural network is designed, including an input layer, a hidden layer which contains two layers of neurons, and an output layer. The input layer has three neurons, the hidden layer H1 has three neurons, the hidden layer H2 has three neurons, and the output layer has one neuron. The network structure [41] is shown in Fig. 2 below.

As Fig. 2 shows, the input vector is normalized to the input layer, the corresponding input of the input layer can be defined as I^i_{input}, and the corresponding input of the input layer can be defined as O^i_{input}.

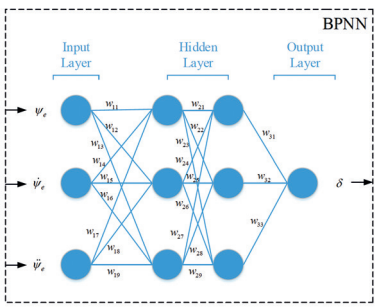

Fig. 2. Schematic diagram of structure of backpropagation neural network (BPNN)

The difference between the actual yaw angle at t and the target yaw angle can be described as $\psi_d - \psi$, represented by ψ_e, the rate of change of the difference between the actual yaw angle at t and the target yaw angle can be described as $\dot{\psi}_e$, and the derivation of the rate of change of the difference between the actual yaw angle at t and the target yaw angle can be described as $\ddot{\psi}_e$.

$$\begin{cases} I_j^1 = \mathrm{O}_j^1 = \psi_e \\ I_j^2 = \mathrm{O}_j^2 = \dot{\psi}_e \,, j = 1 \\ I_j^3 = \mathrm{O}_j^3 = \ddot{\psi}_e \end{cases} \tag{7}$$

Likewise, the input of hidden layer H1 can be defined as I^i_{layer1}, the output of hidden layer H1 can be defined as O^i_{layer1}, the input of hidden layer H2 can be defined as I^i_{layer2}, the output of hidden layer H2 can be defined as O^i_{layer2}, the input of the output layer can be defined as , and the output of the output layer can be defined as O_{out}.

The multiplication accumulation of the input layer's input and the corresponding weights are the input of the H1 layer of neurons. Generally, the input of neurons which located at the second and third layer can be expressed as:

$$\begin{cases} I_j^1 = \sum_{i=1}^{3} O_{j-1}^i \cdot \omega_{(j-1)(3i-2)} \\ I^2_j = \sum_{i=1}^{3} O_{j-1}^i \cdot \omega_{(j-1)(3i-1)}, j = (2, 3) \\ I^3_j = \sum_{i=1}^{3} O_{j-1}^i \cdot \omega_{(j-1)(3i)} \end{cases} \tag{8}$$

According to the controller design, in addition to the input layers, an activation function needs to be set in each layer. The hyperbolic tangent function is selected as the activation function, instead of the sigmoid function, to adapt to the range of values for the rudder angle, and ameliorate the vanishing gradient problem. Thus, the output of the H1, H2, and output layers is obtained by the activation function, which calculates the result of the function by treating all the inputs of a neuron in the layers as the function variable.

Meanwhile, the output of neurons can be expressed as:

$$O_j^i = \tanh(I_j^i) \tag{9}$$

In (9), $i = (1, 3)$, $j = (2, 3)$. As for the last layer, the rudder angle δ is its output.

$$\begin{cases} I_j^1 = \sum_{i=1}^{3} O_{j-1}^i \cdot \omega_{(j-1)(i)} \\ O_j^1 = \tanh(I_j^1) \end{cases}, j = 4 \tag{10}$$

The goal of the algorithm is to track the set yaw angle, thus it is necessary to design the cost function of the neural network to update the weights. The variable of the cost function is related with the target of control, and the cost function can be designed as:

$$D = \frac{1}{2} e^2 \tag{11}$$

In (11), e is the difference between the actual yaw angle at t and the target yaw angle, which is ψ_e. According to the gradient descent method, the updated weight can be obtained:

$$\begin{cases} \Delta\omega_{ji} = -\lambda \frac{\partial D}{\partial \omega_{ji}} \\ \omega_{ji} = \omega_{ji} + \Delta\omega_{ji} \end{cases} \tag{12}$$

In the equation, λ is the learning rate set initially based on experience, which is often taken as a number in the range of 0 to 1.

In the neural network, ω_{31} represents the first weight of the 3rd neural layer, that is, the weight of the first neuron of the H2 layer to the output layer. The $\partial D/\partial\omega$ can be deduced based on chain differentiation law, and the equation of $\partial D/\partial\omega_{31}$ is shown as follows:

$$\frac{\partial D}{\partial \omega_{31}} = \frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial \omega_{31}} \tag{13}$$

In (13), $\partial D/\partial u$ is the derivation of the cost function to the control output. According to (1), the result can be approximated as the difference between the actual yaw angle at t and the target yaw angle when ignoring the result's higher-order error, and the approximate errors can be compensated for by the adaptability of the network.

The derivation of the hyperbolic tangent function can be expressed as:

$$f'(x)=\left(\frac{e^x-e^{-x}}{e^x+e^{-x}}\right)'=1-\frac{(e^x-e^{-x})^2}{(e^x+e^{-x})^2}=1-\left(\tanh(x)\right)^2 \tag{14}$$

Similarly, the weight ω_{3i} $(i = 2, 3)$ is updated:

$$\frac{\partial D}{\partial \omega_{3i}} = \frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial \omega_{3i}} \tag{15}$$

Then, the weight ω_{21}, which is the weight between the hidden layer H2 and the hidden layer H1, is updated. The update of ω_{21} should be calculated as:

$$\frac{\partial D}{\partial \omega_{21}} = \frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial O_3^1} \cdot \frac{\partial O_3^1}{\partial I_3^1} \cdot \frac{\partial I_3^1}{\partial \omega_{21}} \tag{16}$$

Similarly, the weights $\omega_{2m}(m = 1, 4, 7)$, $\omega_{2q}(q = 2, 5, 8)$, and $\omega_{2k}(k = 3, 6, 9)$ are updated:

$$\begin{cases} \frac{\partial D}{\partial \omega_{2m}} = \frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial O_3^1} \cdot \frac{\partial O_3^1}{\partial I_3^1} \cdot \frac{\partial I_3^1}{\partial \omega_{2m}} \\ \frac{\partial D}{\partial \omega_{2q}} = \frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial O_3^2} \cdot \frac{\partial O_3^2}{\partial I_3^2} \cdot \frac{\partial I_3^2}{\partial \omega_{2q}} \\ \frac{\partial D}{\partial \omega_{2k}} = \frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial O_3^3} \cdot \frac{\partial O_3^3}{\partial I_3^3} \cdot \frac{\partial I_3^3}{\partial \omega_{2k}} \end{cases} \tag{17}$$

Then, the weight ω_{11}, which is the weight between the hidden layer H1 and the input layer, is updated. All errors in the backpropagation should be fully considered. Thus, the update of ω_{11} should be calculated as:

$$\frac{\partial D}{\partial \omega_{11}} = \frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial O_3^1} \cdot \frac{\partial O_3^1}{\partial I_3^1} \cdot \frac{\partial I_3^1}{\partial O_2^1} \cdot \frac{\partial I_2^1}{\partial \omega_{11}} +$$
$$\frac{\partial D}{\partial u} \cdot \frac{\partial u}{\partial O_4^1} \cdot \frac{\partial O_4^1}{\partial I_4^1} \cdot \frac{\partial I_4^1}{\partial O_3^2} \cdot \frac{\partial O_3^2}{\partial I_3^2} \cdot \frac{\partial I_3^2}{\partial O_2^1} \cdot \frac{\partial O_2^1}{\partial I_2^1} \cdot \frac{\partial I_2^1}{\partial \omega_{11}} +$$

$$\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^1}\cdot\frac{\partial O_3^3}{\partial I_3^3}\cdot\frac{\partial I_3^3}{\partial O_2^1}\cdot\frac{\partial O_2^1}{\partial I_2^1}\cdot\frac{\partial I_2^1}{\partial \omega_{11}} \tag{18}$$

Similarly, the weights $\omega_{1m}(m = 1, 4, 7)$, $\omega_{1q}(q = 2, 5, 8)$, and $\omega_{1k}(k = 3, 6, 9)$ are updated.

$$\begin{cases}
\frac{\partial D}{\partial \omega_{1m}}=\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^1}\cdot\frac{\partial O_3^1}{\partial I_3^1}\cdot\frac{\partial I_3^1}{\partial O_2^1}\cdot\frac{\partial O_2^1}{\partial I_2^1}\cdot\frac{\partial I_2^1}{\partial \omega_{1m}}+\\
\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^2}\cdot\frac{\partial O_3^2}{\partial I_3^2}\cdot\frac{\partial I_3^2}{\partial O_2^1}\cdot\frac{\partial O_2^1}{\partial I_2^1}\cdot\frac{\partial I_2^1}{\partial \omega_{1m}}+\\
\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^3}\cdot\frac{\partial O_3^3}{\partial I_3^3}\cdot\frac{\partial I_3^3}{\partial O_2^1}\cdot\frac{\partial O_2^1}{\partial I_2^1}\cdot\frac{\partial I_2^1}{\partial \omega_{1m}}\\
\frac{\partial D}{\partial \omega_{1q}}=\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^1}\cdot\frac{\partial O_3^1}{\partial I_3^1}\cdot\frac{\partial I_3^1}{\partial O_2^2}\cdot\frac{\partial O_2^2}{\partial I_2^2}\cdot\frac{\partial I_2^2}{\partial \omega_{1q}}+\\
\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^2}\cdot\frac{\partial O_3^2}{\partial I_3^2}\cdot\frac{\partial I_3^2}{\partial O_2^2}\cdot\frac{\partial O_2^2}{\partial I_2^2}\cdot\frac{\partial I_2^2}{\partial \omega_{1q}}+\\
\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^3}\cdot\frac{\partial O_3^3}{\partial I_3^3}\cdot\frac{\partial I_3^3}{\partial O_2^2}\cdot\frac{\partial O_2^2}{\partial I_2^2}\cdot\frac{\partial I_2^2}{\partial \omega_{1q}}\\
\frac{\partial D}{\partial \omega_{1k}}=\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^1}\cdot\frac{\partial O_3^1}{\partial I_3^1}\cdot\frac{\partial I_3^1}{\partial O_2^3}\cdot\frac{\partial O_2^3}{\partial I_2^3}\cdot\frac{\partial I_2^3}{\partial \omega_{1k}}+\\
\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^2}\cdot\frac{\partial O_3^2}{\partial I_3^2}\cdot\frac{\partial I_3^2}{\partial O_2^3}\cdot\frac{\partial O_2^3}{\partial I_2^3}\cdot\frac{\partial I_2^3}{\partial \omega_{1k}}+\\
\frac{\partial D}{\partial u}\cdot\frac{\partial u}{\partial O_4^1}\cdot\frac{\partial O_4^1}{\partial I_4^1}\cdot\frac{\partial I_4^1}{\partial O_3^3}\cdot\frac{\partial O_3^3}{\partial I_3^3}\cdot\frac{\partial I_3^3}{\partial O_2^3}\cdot\frac{\partial O_2^3}{\partial I_2^3}\cdot\frac{\partial I_2^3}{\partial \omega_{1k}}
\end{cases} \tag{19}$$

In the updating of the weight ω_{2i} and $\omega_{1i}(i = 1, 9)$, the weights of the latter layer are used to update the current weight. There are two ideas here: one is to use the updated $\omega_{3i}(i = 1, 3)$, the other is to use the $\omega_{3i}(i = 1, 3)$, which has not been updated. The former is a unified update and the latter is to update the weights with the updated weights, which can theoretically accelerate the convergence. Therefore, the iterative update method is adopted.

On the basis of the above weight algorithm, the control variables of the controller can be represented mathematically. The related matrices are defined as follows:

$$E = [e, \dot{e}, \ddot{e}] \tag{20}$$

The matrix E is the error, representing the difference between the desired state and actual state.

$$[T, C]^T = [1, T_1 + T_2, T_1T_2, K, KT_3] \tag{21}$$

The matrix T, which contains three rows and one column, is the time parameters of the ship model, and the matrix C, which contains two rows and one column, is the coefficient of the controlled variable.

$$\Lambda = [\delta, \delta_r, \dot{\delta}] \tag{22}$$

The matrix Λ is the controlled variable. According to the equations above, (1) can be converted to (23), shown as follows:

$$[-\dot{E}, r^3]\cdot[T, \alpha]^T = \Lambda\cdot C \tag{23}$$

To calculate the controlled variable, the following matrices are introduced:

$$W_1=\begin{bmatrix} w_{11} & w_{14} & w_{17}\\ w_{12} & w_{15} & w_{18}\\ w_{13} & w_{16} & w_{19}\end{bmatrix},\ W_2=\begin{bmatrix} w_{21} & w_{24} & w_{27}\\ w_{22} & w_{25} & w_{28}\\ w_{23} & w_{26} & w_{29}\end{bmatrix},$$

$$W_3 = [w_{31}, w_{32}, w_{33}]^T \tag{24}$$

In addition, considering that the traditional gradient descent method easily falls into the local optima, an adaptive momentum was adopted when updating the weights. The adaptive momentum was designed as:

$$\varepsilon_\zeta = \zeta\tanh(e^{\frac{\rho}{x}}) \tag{25}$$

In (25), ρ is a constant coefficient, x is the error, and ζ is an adaptive coefficient. The specific equation of ζ is given as follows:

$$\zeta_{(t+1)} = \Delta\omega_{ji}(t)G(x) \tag{26}$$

In (26), $\Delta\omega_{ji}(t)$ is the former change of weight, and $G(x)$ is a piecewise function as follows:

$$G(x)=\begin{cases}0.8, & x>\sigma\\ 0, & x<\sigma\end{cases} \tag{27}$$

In (27), σ is the maximum permissible error. The stability of the convergence is guaranteed by the piecewise function, and the adaptive momentum is zero when the accuracy is satisfied.

Therefore, the updated weights can be represented as:

$$\begin{cases}\Delta\omega_{ji}(t+1) = -\lambda\frac{\partial D}{\partial\omega_{ji}} + \varepsilon_\zeta\\ \omega_{ji}(t+1) = \omega_{ji}(t) + \Delta\omega_{ji}(t+1)\end{cases} \tag{28}$$

When the algorithm falls into the local optima, $\Delta\omega_{ji}(t+1)\approx\Delta\omega_{ji}(t)$, thus:

$$\Delta\omega_{ji}(t+1)=-\lambda\frac{\partial D}{\partial\omega_{ji}}+\varepsilon_\zeta\approx-\lambda\frac{\partial D}{\partial\omega_{ji}}+\Delta\omega_{ji}(t+1)G(x)\tanh(e^{\frac{\rho}{x}})=$$
$$-\frac{\lambda}{1-G(x)\tanh(e^{\frac{\rho}{x}})}\frac{\partial D}{\partial\omega_{ji}} \tag{29}$$

It can be concluded that the learning rate is amplified to shift away from the local optima, and the convergence is accelerated. In summary, the control variable of the controller can be obtained:

$$\mathrm{u} = \tanh\{\tanh[\tanh(EW_1)W_2]W_3\} \tag{30}$$

C. CONVERGENCE ANALYSIS

In order to better describe the convergence analysis process of the control algorithm presented here, the following theorems are applied.

Theorem 1 [42]:

Suppose the function $f(x)$ is Lipschitz continuous, for the constant $L>0$ and any x, y on the domain, it is concluded that:

$$||\nabla f(x) - \nabla f(y)|| \le L||x - y|| \tag{31}$$

Theorem 2 [42]:

Suppose the function $f(x)$ obeys Theorem 1, the Taylor second expansion of function $f(x)$ can be obtained:

$$f(y)=f\big(x+(y-x)\big)\le f(x)+\nabla f(x)^T(y-x)+\frac{1}{2}\nabla^2 f(x)||y-x||^2$$
$$\le f(x)+\nabla f(x)^T(y-x)+\frac{1}{2}\nabla L||y-x||^2 \quad (32)$$

Considering (11): $f(x)=x^2/2$, its domain is $[-\pi, \pi]$, and it is easy to verify that $||\nabla f(x)-\nabla f(y)|| \le L||x-y||$ when $L\ge\pi$, which means that the function $f(x)$ obeys Theorem 1 and Theorem 2.

According to (12), we obtain:

$$y = x_{t+1} = x_t - \lambda\nabla f(x_t) \quad (33)$$

Thus, combining (32), we obtain:

$$f(x_{t+1}) \le f(x_t)+\nabla f(x_t)^T(x_{t+1}-x_t)+\frac{1}{2}L||x_{t+1}-x_t||^2$$
$$=f(x_t)+\nabla f(x_t)^T\big(x_t-\lambda\nabla f(x_t)-x_t\big)+\frac{1}{2}L||x_t-\lambda\nabla f(x_t)-x_t||^2$$
$$=f(x_t)-\nabla f(x_t)^T\lambda\nabla f(x_t)+\frac{1}{2}L||\lambda\nabla f(x_t)||^2$$
$$=f(x_t)-\lambda||\nabla f(x_t)||^2+\frac{1}{2}L\lambda^2||\nabla f(x_t)||^2 \quad (34)$$
$$=f(x_t)-(1-\frac{1}{2}L\lambda)\lambda||\nabla f(x_t)||^2$$

while $0<\lambda\le 1/L$ and $L>0$, $-(1-L\lambda/2)=L\lambda/2-1\le L(1/L)/2-1=1/2-1=-1/2$, thus:

$$f(x_{t+1}) \le f(x_t) -\frac{1}{2}\lambda||\nabla f(x_t)||^2 \quad (35)$$

Now, assuming that $f(x^*)$ is the optimal solution, the Taylor first expansion of $f(x^*)$ can be obtained:

$$\begin{cases} f(x^*) \ge f(x_t) + \nabla f(x_t)^T(x^*-x_t) \\ f(x_t) \le f(x^*) + \nabla f(x_t)^T(x_t-x^*) \end{cases} \quad (36)$$

Plugging this into (34), we obtain:

$$f(x_{t+1}) \le f(x^*) + \nabla f(x_t)^T(x_t-x^*)-\frac{\lambda}{2}||\nabla f(x_t)||^2 \quad (37)$$

$$f(x_{t+1})-f(x^*)\le\frac{1}{2\lambda}\big(2\lambda\nabla f(x_t)^T(x_t-x^*)-\lambda^2||\nabla f(x_t)||^2\big)$$
$$=\frac{1}{2\lambda}\big(2\lambda\nabla f(x_t)^T(x_t-x^*)-\lambda^2||\nabla f(x_t)||^2-||x_t-x^*||^2+||x_t-x^*||^2\big)$$
$$=\frac{1}{2\lambda}\big(||x_t-x^*||^2-||x_t-\lambda\nabla f(x_t)-x^*||^2\big) \quad (38)$$

and plugging (33) into (38),

$$f(x_{t+1}) - f(x^*) \le\frac{1}{2\lambda}\big(||x_t-x^*||^2-||x_{t+1}-x^*||^2\big) \quad (39)$$

Summing over iterations, we get:

$$\sum_{t=1}^{n}\big(f(x_t)-f(x^*)\big) \le \sum_{t=1}^{n}\frac{1}{2\lambda}\big(||x_0-x^*||^2-||x_t-x^*||^2\big)$$
$$= \frac{1}{2\lambda}\big(||x_0-x^*||^2-||x_n-x^*||^2\big) \le \frac{1}{2\lambda}\big(||x_0-x^*||^2\big) \quad (40)$$

$$\sum_{t=1}^{n}f(x_t) \le nf(x^*) + \frac{1}{2\lambda}||x_0-x^*||^2 \quad (41)$$

According to (35), we obtain:

$$nf(x_n) \le nf(x^*) + \frac{1}{2\lambda}||x_0-x^*||^2 \quad (42)$$

$$f(x_n) \le f(x^*) + \frac{1}{2n\lambda}||x_0-x^*||^2 \quad (43)$$

For arbitrary infinitesimals γ, in order for $f(x_n)-f(x^*)\le\gamma$ to be satisfied, it must be guaranteed that $n\ge||x_0-x^*||^2/(2\lambda\gamma)$. The adaptive momentum is adopted, and the learning rate is enlarged, which means that convergence can be achieved in smaller number of iterations.

The design of the controller is depicted above, and it is demonstrated that the algorithm takes the optimal solution when n moves towards positive infinity. The convergence of the proposed algorithm has been proved mathematically to be reasonable and stable.

SIMULATION TEST DESIGN

To verify the effectiveness of the proposed algorithm and the superiority of the improvement, simulation experiments were carried out.

To demonstrate the advantage of the improved TD and adaptive momentum, the simulation tests are displayed.

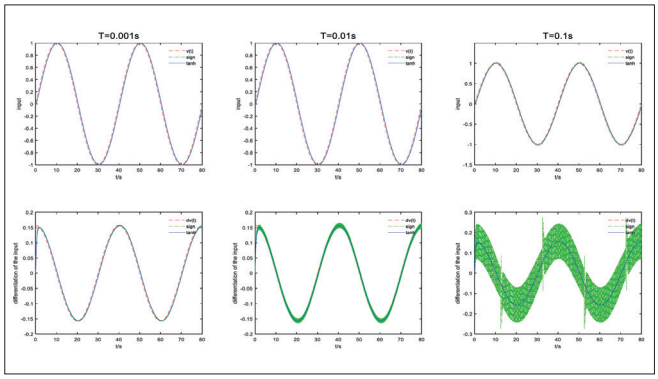

Fig. 3. Curves of input and differentiation under TD and improved TD with various time-steps

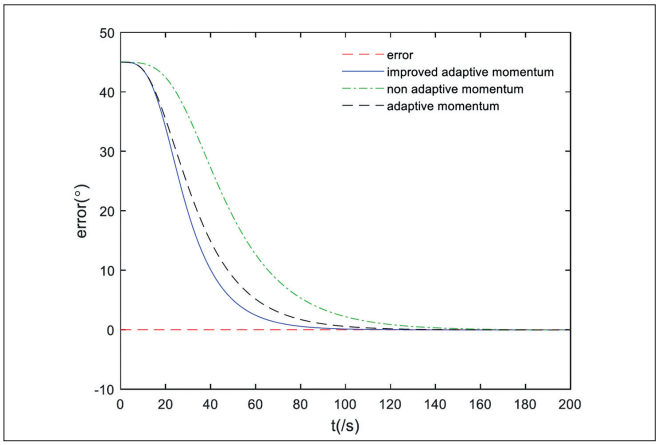

Fig. 4. Curve of convergence of error under various momentum strategies

Fig. 3 shows the simulation tests under several time-steps, and it can be seen that the classic tracking differentiator got worse when the time-step was greater than 0.001s. When the time-step is large, the error caused by the signum function is amplified, resulting in oscillations. In contrast, the improved tracking differentiator is stable and precise under all time-steps.

Fig. 4 shows the simulation tests under different momentum strategies, and it can be concluded that the longest convergence times occurred in the algorithm without adaptive momentum. The convergence time of the adaptive momentum algorithm is 30% less than the former, whereas the convergence time of the improved adaptive momentum algorithm is 15% less than the adaptive momentum algorithm.

The superiority of the improvement has been demonstrated, so the control algorithm then needs to be tested. For comprehensive results, simulation experiments under several sets of working conditions were carried out to verify the convergence of the error and the accuracy of the BPNN control algorithm.

The second-order nonlinear model of the USV in the literature [40] was selected. The parameters of the USV are shown in Table 1, and the Runge–Kutta method is selected for the numerical simulation experiments.

Tab. 1. Parameters of USV

Parameter	T_1/s	T_2/s	T_3/s	K/s	α/(s^2·rad^{-2})	δ_f/rad
Value	11.646	12.753	0.968	0.135	−0.002	0.006

Considering the limitation of the actuator and the limitation of the change of rudder angle in the steering, the maximum rudder angle is chosen as 0.5 rad, and the maximum rate of change is chosen as 0.06 rad/s. The simulated time-step is set to 0.1s, and the simulation experiments are set up as follows.

Case 1:

The initial state of the USV is , the environmental interference is taken from the literature, the interference formation is , and the desired yaw angle is 2pi/3, pi/4. The simulation results are presented in Figures 5 and 6.

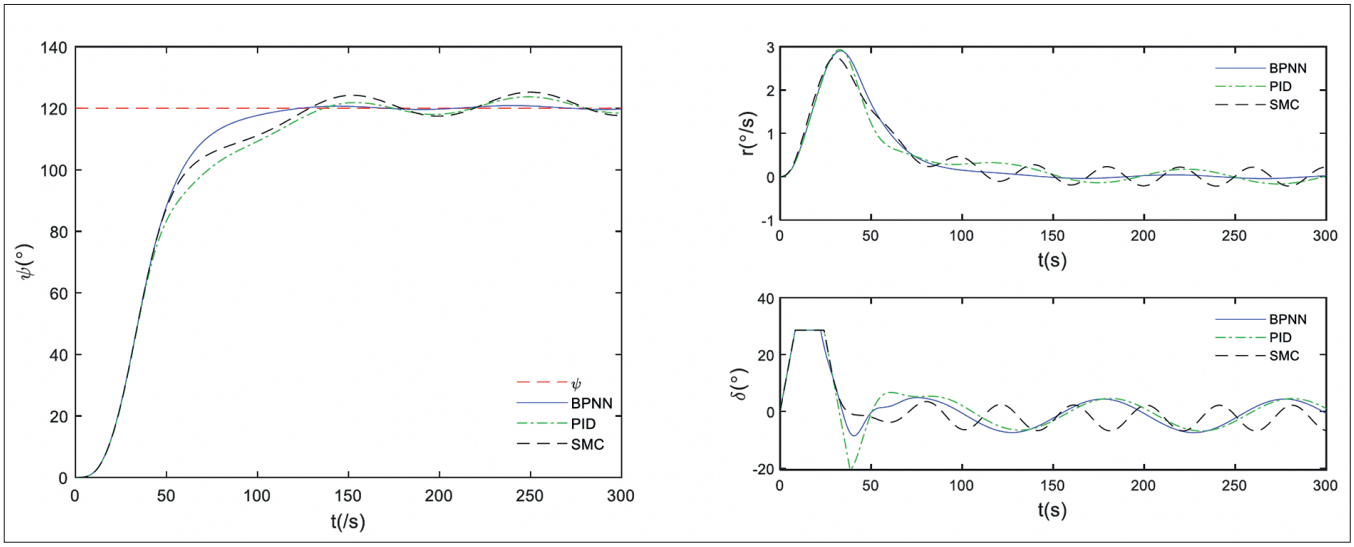

Fig. 5. Curve of change of yaw, rudder and heading rate under BPNN, SMC and PID in the presence of disturbance

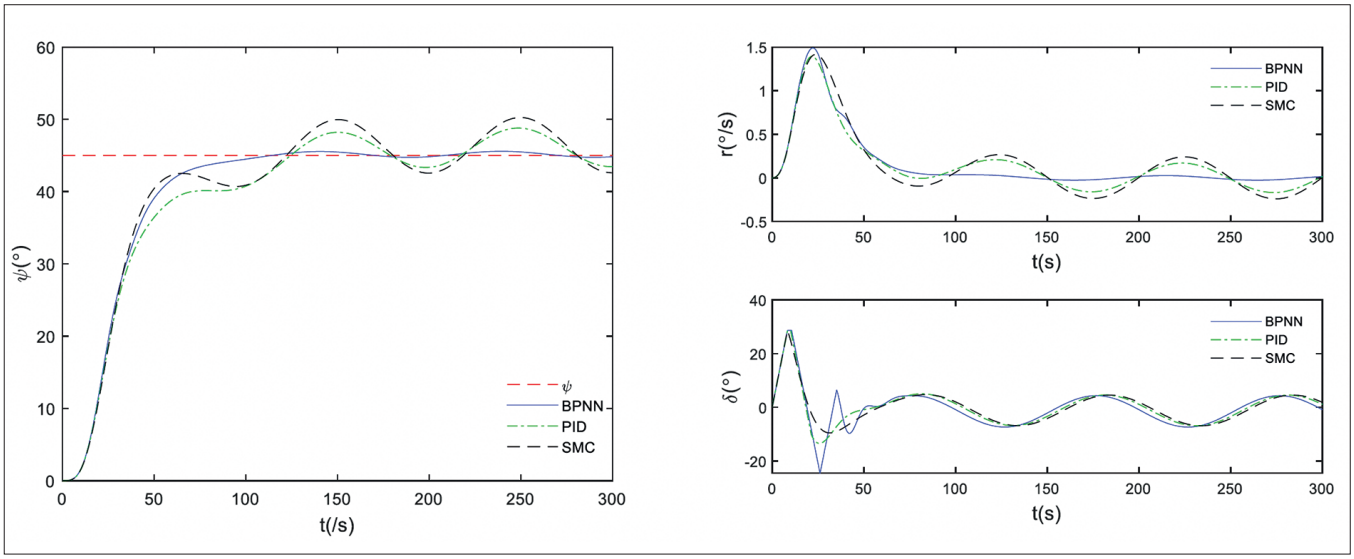

Fig. 6. Curve of change of yaw, rudder and heading rate under BPNN, SMC and PID in the presence of disturbance

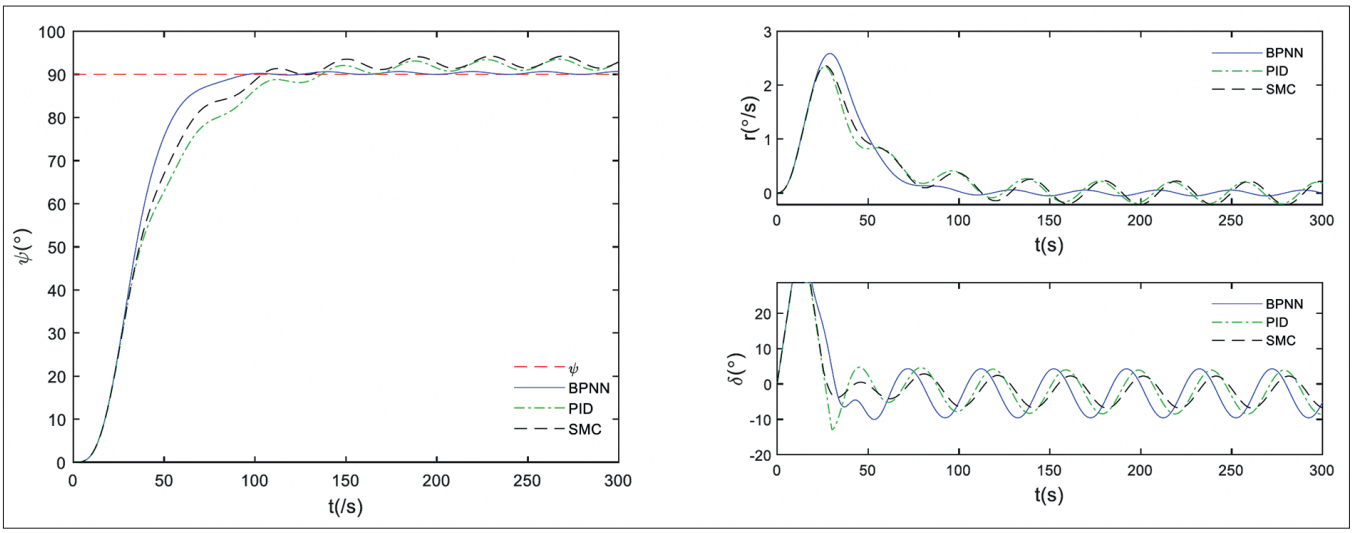

Fig. 7. Curve of change of yaw, rudder and heading rate under BPNN, SMC and PID in the presence of disturbance

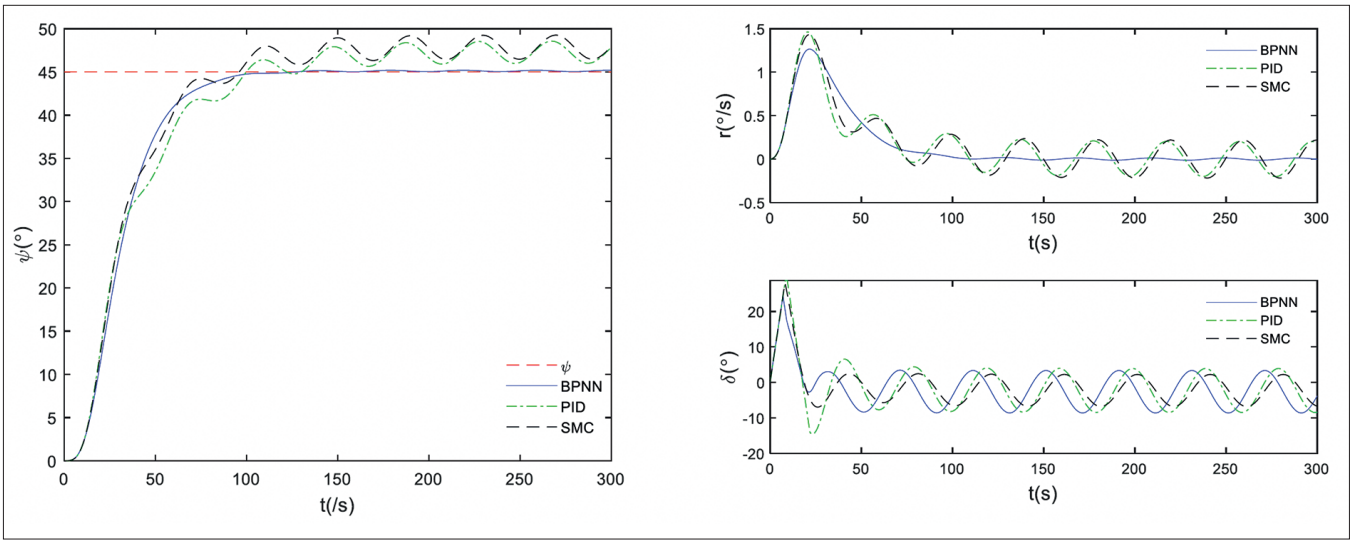

Fig. 8. Curve of change of yaw, rudder and heading rate under BPNN, SMC and PID in the presence of disturbance

Fig. 5 is the yaw tracking diagram under the slow time-varying interference when the desired yaw angle is 2pi/3, and Fig. 6 is the yaw tracking diagram under the slow time-varying interference when the desired yaw angle is pi/4.

Case 2:

The initial state of the USV is (ψ, δ), the environmental interference is taken from the literature, the interference formation is d_r=0.04+0.1sin(0.5πt), and the desired yaw angle is pi/2, pi/4. The simulation results are presented in Figures 7 and 8.

Fig. 7 is the yaw tracking diagram under the time-varying interference when the desired yaw angle is pi/2, while Fig. 8 is under the pi/4 desired yaw angle.

Case 3:

The initial state of the USV is , the environmental interference is taken from the literature, the interference formation is , the desired yaw angle is pi/3, pi/4. The simulation results are presented in Figures 9 and 10.

Fig. 9 is the yaw tracking diagram under the multi time-varying interference interference when the desired yaw angle is pi/2, while Fig. 10 is under the pi/4 desired yaw angle.

Compared with the literature [20, 23], the rapidity of control has improved by more than 20%, and the accuracy of control has improved by more than 50%. The strong anti-interference has been proved under multi-interference, and the algorithm realized maintained the deviation at under 2% without a disturbance observer. In conclusion, according to the figures, BPNN is more rapid and accurate than SMC and PID. Furthermore, unlike the SMC and PID that have feedforward compensation to the critical rudder angle, adaptive compensation was implemented in the BPNN controller, which shows strong robustness.

CONCLUSION

Utilizing optimization of the cost function, combined with adaptive backpropagation neural network control, a heading control method of a USV based on the improved BPNN is proposed. A novel control algorithm for the heading control problem is presented, and the desired yaw angle of the USV

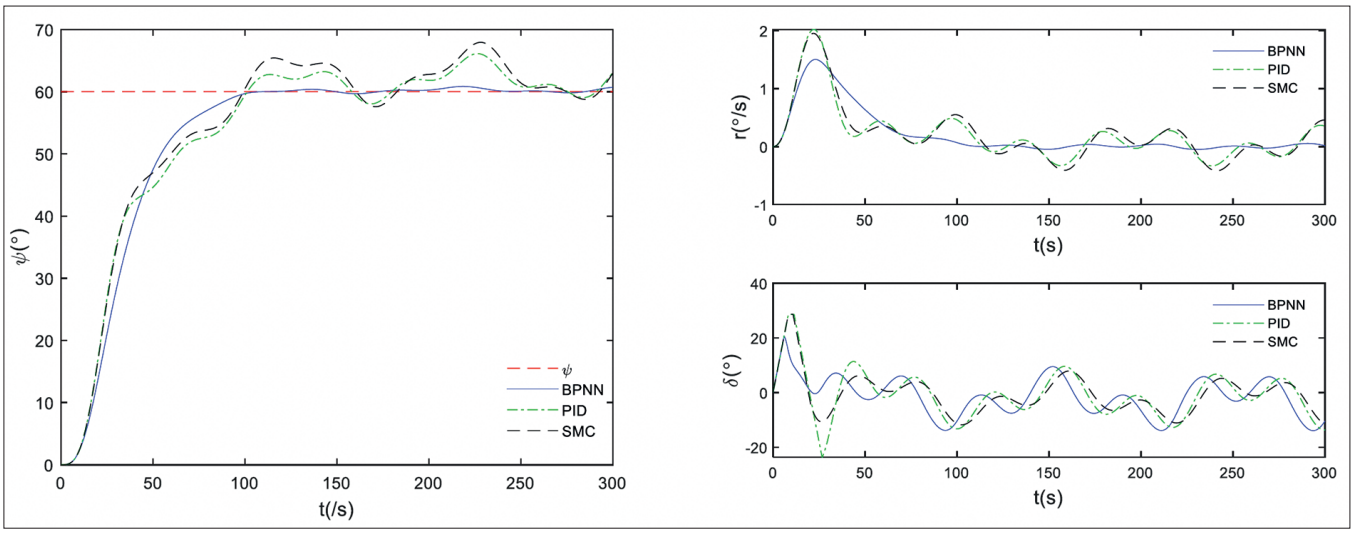

Fig. 9. Curve of change of yaw, rudder and heading rate under BPNN, SMC and PID in the presence of disturbance

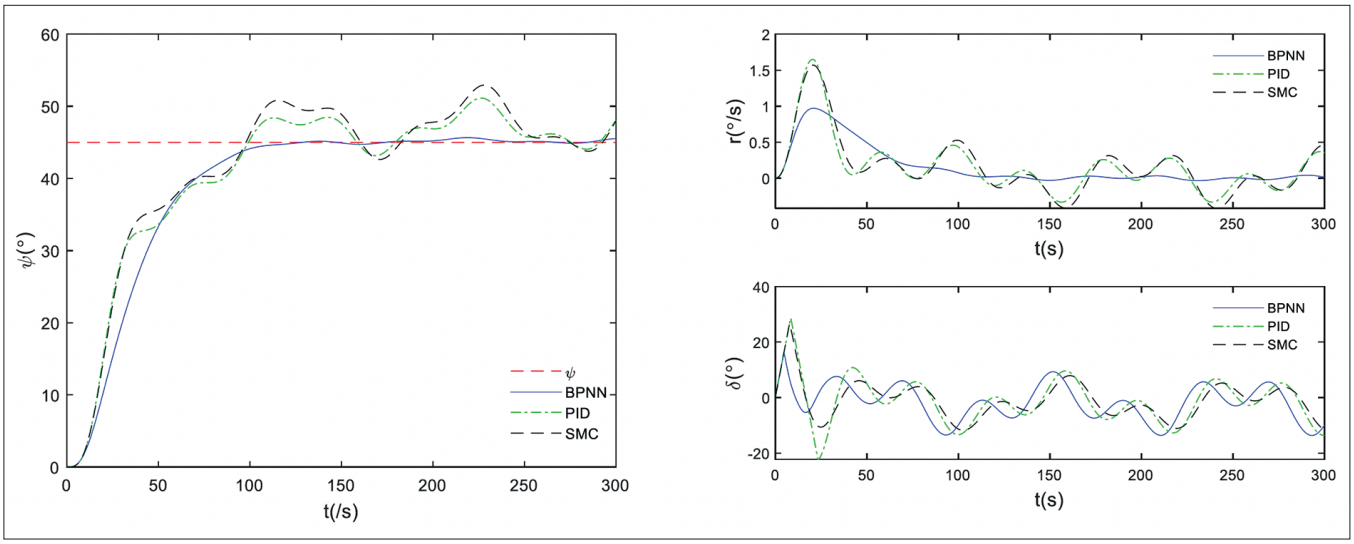

Fig. 10. Curve of change of yaw, rudder and heading rate under BPNN, SMC and PID in the presence of disturbance

is transformed into a minimized cost function that contains the actual state and desired state, which is convenient for the controller design. An improved gradient descent strategy that adopts an improved adaptive momentum facilitates the neural network to search for the global optimum, ensuring rapid and stable convergence and implementing adaptive control parameters. Theoretically, it is proved that the improved gradient descent strategy can solve the local optima problem and reduce the number of iterations, and its effectiveness and accuracy are confirmed by the experiments. An improved tracking differentiator is devised for extracting state variables and weakening the amplification effect of noise, and a transition process is implemented to smooth the input signal. The improved tracking differentiator enables signal tracking to be stable and effective in all time-steps, supplemented by tests to verify it. By combining the backpropagation neural network and optimization theory, applying the tanh function to build a nonlinear mapping, and using an approximate strategy to summarize the general mathematical expressions in the design process of the control algorithm, a novel controller for heading control of the USV is designed. A purely adaptive controller design that is based on adaptive control has theoretical significance, which means that accurate model parameters are not required for precise control. Relying on the adaptive capabilities, disturbance caused by the assumptions and approximate strategy in the algorithm can be compensated, which is of engineering significance. With the advantages of strong robustness, strong anti-interference and highly accurate control, as shown in a variety of simulation experiments that have been carried out, the advantages of the algorithm have been proved compared with conventional algorithms.

ACKNOWLEDGEMENTS

This work is supported by the Natural Science Foundation of China (Grant numbers 51709214, 51779052, 51809203); Stable Supporting Fund of Science and Technology on Underwater Vehicle Technology (Grant number JCKYS2022SXJQR-04); Innovative Research Foundation of Ship General Performance (Grant number 21822214).

REFERENCES

1. Y. L. Liao, Y. Li, K. W. Pan, et al., 'Adaptive Multi-body Yawing Filter of Wave Driven Robot with Dynamic Response Amending,' Ocean Engineering, vol. 242, pp. 1-11(110167), December 2021. doi: 10.1016/j.oceaneng.2021.110167.

2. G. Shao, Y. Ma, R. Malekian, et al., 'A novel cooperative platform design for coupled USV-UAV systems,' IEEE Transactions on Industrial Informatics, vol. 15, no. 9, pp. 4913-4922, April 2019. doi: 10.1109/TII.2019.2912024.

3. G. Q. Zhang, W. Yu, W. D. Zhang, et al., 'Robust adaptive formation control of underactuated surface vehicles with the desired-heading amendment,' Journal of Marine Science and Technology, vol. 27, pp. 138-150, June 2021. doi: 10.1007/s00773-021-00820-2.

4. Y. Peng, Y. Yang, J. X. Cui, et al., 'Development of the USV 'JingHai-I' and sea trials in the southern Yellow Sea,' Ocean Engineering, vol. 131, pp. 186-196, February 2017. doi: 10.1016/j.oceaneng.2016.09.001.

5. Y. L. Liao, Z. H. Jia, W. B. Zhang, et al., 'Layered berthing method and experiment of unmanned surface vehicle based on multiple constraints analysis,' Applied Ocean Research, vol. 86, no. 5, pp. 47-60, May 2019. doi: 10.1016/j.apor.2019.02.003.

6. A. Stateczny, P. Burdziakowski, 'Universal Autonomous Control and Management System for Multipurpose Unmanned Surface Vessel,' Polish Maritime Research, vol. 26, no. 1, pp. 30-39, March 2019. doi: 10.2478/pomr-2019-0004.

7. J. Kim, 'Target following and close monitoring using an unmanned surface vehicle,' IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 11, pp. 4233-4242, November 2020. doi: 10.1109/TSMC.2018.2846602.

8. Y. L. Liao, Q. Q. Jiang, T. P. Du, et al., 'Redefined output model-free adaptive control method and unmanned surface vehicle heading control,' IEEE Journal of Oceanic Engineering, vol. 45, no. 3, pp. 714-723, July 2020. doi: 10.1109/JOE.2019.2896397.

9. J. X. Hu, Y. Ge, X. Zhou, et al., 'Research on the course control of USV based on improved ADRC,' Systems Science & Control Engineering, vol. 9, no. 1, pp. 44-51, January 2021. doi: 10.1080/21642583.2020.1865216.

10. M. R. Neria, R. Madonski, S. Shao, et al., 'Robust tracking in underactuated systems using flatness-based ADRC with cascade observers,' Journal of Dynamic Systems Measurement and Control - Transactions of the ASME, vol. 142, no. 9, pp. 1-9, September 2020. doi: 10.1115/1.4046799.

11. M. H. Khooban, N. Vafamand, T. Dragicevic, et al., 'Polynomial fuzzy model-based approach for underactuated surface vessels,' IET Control Theory and Applications, vol. 12, no. 7, pp. 914-921, February 2018. doi: 10.1049/iet-cta.2017.1106.

12. Y. G. Deng, X. K. Zhang, N. Im, et al., 'Adaptive fuzzy tracking control for underactuated surface vessels with unmodeled dynamics and input saturation,' ISA Transactions, vol. 103, pp. 52-62, August 2020. doi: 10.1016/j.isatra.2020.04.010.

13. Z. P. Dong, Y. Liu, H. Wang, et al., 'Method of cooperative formation control for underactuated USVs based on nonlinear backstepping and cascade system theory,' Polish Maritime Research, vol. 28, no. 1, pp. 149-162, March 2021. doi: 10.2478/pomr-2021-0014.

14. T. Lin, Y. W. Huang., 'Course regulation for USV by prescribed performance backstepping control,' in Proceedings of 2021 Chinese Intelligent Systems Conference, pp. 680-689, January 2022. doi: 10.1007/978-981-16-6328-4_69.

15. N. Wang, H. K. He, 'Dynamics-level finite-time fuzzy monocular visual servo of an unmanned surface vehicle,' IEEE Transactions on Industrial Electronics, vol. 67, no. 11, pp. 9648-9658, November 2020. doi: 10.1109/tie.2019.2952786.

16. Z. Q. Liu, 'Adaptive Sliding Mode Control for Ship Autopilot with Speed Keeping,' Polish Maritime Research, vol. 25, no. 4, pp. 21-29, December 2018. doi: 10.2478/pomr-2018-0128.

17. C. H. Cheng, L. Li, Q. Han, et al., 'Adaptive sliding mode ADRC for attitude tracking with actuator saturation and uncertainties,' International Journal of Robotics & Automation, vol. 36, no. 5, pp. 337-344, 2021. doi:10.2316/j.2021.206-0560.

18. S. S. Wang, Y. L. Tuo, 'Robust Trajectory Tracking Control of Underactuated Surface Vehicles with Prescribed Performance,' Polish Maritime Research, vol. 27, no. 4, pp. 148-156, December 2020. doi: 10.2478/pomr-2020-0075.

19. H. B. Wang, J. Dong, Z. K. Liu, et al., 'Control algorithm for trajectory tracking of an underactuated USV under multiple constraints,' Mathematical Problems in Engineering, vol. 1, no. 8, pp. 1-12, May 2022. doi: 10.1155/2022/5274452.

20. L. G. Li, Z. Y. Pei, J. C. Jin, et al., 'Control of unmanned surface vehicle along the desired trajectory using improved line of sight and estimated sideslip angle,' Polish Maritime Research, vol. 28, no. 2, pp. 18-26, June 2021. doi: 10.2478/pomr-2021-0017.

21. Z. P. Dong, S. J. Qi, M. Yu, et al., 'An improved dynamic surface sliding mode method for autonomous cooperative formation control of underactuated USVs with complex marine environment disturbances,' Polish Maritime Research, vol. 29, no. 3, pp. 47-60, December 2022. doi: 10.2478/pomr-2022-0025.

22. N. K. Gupta, M. K. Kar, A. K. Singh, 'Design of a 2-DOF-PID controller using an improved sine-cosine algorithm for load frequency control of a three-area system with nonlinearities,' Protection and Control of Modern Power Systems, vol. 7, no. 1, p. 33, September 2022. doi: 10.1186/s41601-022-00255-w.

23. M. M. Ozyetkin, 'An approximation method and PID controller tuning for systems having integer order and non-integer order delay,' Alexandria Engineering Journal, vol. 61, no. 12, pp. 11365-11375, December 2022. doi: 10.1016/j.aej.2022.05.015.

24. R. L. Miao, Z. P. Dong, L. Wan, et al., 'Heading control system design for a micro-USV based on an adaptive expert S-PID algorithm,' Polish Maritime Research, vol. 25, no. 2, pp. 6-13, June 2018. doi: 10.2478/pomr-2018-0049.

25. C. Ke, H. F. Chen, 'Cooperative path planning for air-sea heterogeneous unmanned vehicles using search-and-tracking mission,' Ocean Engineering, vol. 262, p. 112020, October 2022. doi: 10.1016/j.oceaneng.2022.112020.

26. G. Q. Zhang, S. J. Chu, W. D. Zhang, et al., 'Adaptive neural fault-tolerant control for USV with the output-based triggering approach,' IEEE Transactions on Vehicular Technology, vol. 71, no. 7, pp. 6948-6957, July 2022. doi: 10.1109/TVT.2022.3167038.

27. A. G. Garcia, H. Castaneda, L. Garrido, 'USV path-following control based on deep reinforcement learning and adaptive control,' in Global Oceans 2020, pp. 1-7, August 2020. doi: 10.1109/IEEECONF38699.2020.9389360.

28. H. N. Esfahani, R. Szlapczynski, 'Model predictive super-twisting sliding mode control for an autonomous surface vehicle,' Polish Maritime Research, vol. 26, no. 3, pp. 163-171, October 2019. doi: 10.2478/pomr-2019-0057.

29. J. Y. Zhuang, L. Zhang, Z. H. Qin, et al., 'Motion Control and Collision Avoidance Algorithms for Unmanned Surface Vehicle Swarm in Practical Maritime Environment,' Polish Maritime Research, vol. 26, no. 1, pp. 163-171, March 2019. doi: 10.2478/pomr-2019-0012.

30. K. S. Kula, 'Automatic Control of Ship Motion Conducting Search in Open Waters,' Polish Maritime Research, vol. 27, no. 4, pp. 157-169, December 2020. doi: 10.2478/pomr-2020-0076.

31. Q. Zhang, Z. Y. Ding, M. J. Zhang, 'Adaptive Self-Regulation PID Control of Course-Keeping for Ships,' Polish Maritime Research, vol. 27, no. 1, pp. 39-45, March 2020. doi: 10.2478/pomr-2020-0004.

32. S. S. Wang, M. Y. Fu, Y. H. Wang, 'Robust adaptive steering control for unmanned surface vehicle with unknown control direction and input saturation,' International Journal of Adaptive Control and Signal Processing, vol. 33, no. 7, pp. 1212-1224, 2023. doi: 10.32604/iasc.2023.027614.

33. L. Liu, Y. S. Fan, 'Active disturbance rejection course control for USV based on RBF neural network,' in 2020 39th Chinese Control Conference (CCC), pp. 3344-3351, July 2020. doi: 10.23919/CCC50068.2020.9188760.

34. G. Yi, Z. Liu, J. Q. Zhang, et al., 'Research on underactuated USV path following algorithm,' in 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), pp. 2141-2145, June 2020. doi: 10.1109/ITNEC48623.2020.9085222.

35. R. Farkh, K. Aljaloud, 'Vision navigation based PID control for line tracking robot,' Intelligent Automation & Soft Computing, vol. 35, no. 1, pp. 901-911, 2023. doi: 10.32604/iasc.2023.027614.

36. B. Du, B. Lin, C. M. Zhang, et al., 'Safe deep reinforcement learning-based adaptive control for USV interception mission,' Ocean Engineering, vol. 246, p. 110477, February 2022. doi: 10.1016/j.oceaneng.2021.110477.

37. T. Asfihani, D. K. Arif, Subchan, et al., 'Comparison of LQG and adaptive PID controller for USV heading control,' Journal of Physics: Conference Series, vol. 1218, p. 012058, May 2019. doi: 10.1088/1742-6596/1218/1/012058.

38. S. T. Wang, X. H. Yin, P. Li, et al., 'Consensus control of multi-agent systems with deception attacks using event-triggered adaptive cognitive control,' Communications in Nonlinear Science and Numerical Simulation, vol. 114, p. 106675, November 2022. doi: 10.1016/j.cnsns.2022.106675.

39. Y. J. Zhao, Y. Ma, S. L. Hu, 'USV formation and path-following control via deep reinforcement learning with random braking,' IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 12, pp. 5468-5478, April 2021. doi: 10.1109/TNNLS.2021.3068762.

40. S. Xie, X. M. Chu, C. G. Liu, 'Parameter identification of ship motion model based on multi-innovation methods,' Journal of Marine Science and Technology, vol. 25, pp. 162-184, March 2022. doi: 10.1007/s00773-019-00639-y.

41. Y. T. Gui, D. Q. Li, R. Y. Fang, 'A fast adaptive algorithm for training deep neural networks,' Applied Intelligence, vol. 53, no. 12, pp. 1-10, June 2022. doi: 10.1007/s10489-022-03629-7.

42. Y. H. Kang, Y. Kuang, J. Cheng, et al., 'Robust leaderless time-varying formation control for unmanned aerial vehicle swarm system with Lipschitz nonlinear dynamics and directed switching topologies,' Chinese Journal of Aeronautics, vol. 35, no. 1, pp. 124-136, July 2021. doi: 10.1016/j.cja.2021.05.017.