Observer-Based Adaptive Neural Network Trajectory Tracking Control for Remotely Operated Vehicle

Zhenzhong Chu, Daqi Zhu, and Simon X. Yang, *Senior Member, IEEE*

***Abstract*— This paper focuses on the adaptive trajectory tracking control for a remotely operated vehicle (ROV) with an unknown dynamic model and the unmeasured states. Unlike most previous trajectory tracking control approaches, in this paper, the velocity states and the angular velocity states in the body-fixed frame are unmeasured, and the thrust model is inaccurate. Obviously, it is more in line with the actual ROV systems. Since the dynamic model is unknown, a new local recurrent neural network (local RNN) structure with fast learning speed is proposed for online identification. To estimate the unmeasured states, an adaptive terminal sliding-mode state observer based on the local RNN is proposed, so that the finite-time convergence of the trajectory tracking error can be guaranteed. Considering the problem of inaccurate thrust model, an adaptive scale factor is introduced into thrust model, and the thruster control signal is considered as the input of the trajectory tracking system directly. Based on the local RNN output, the adaptive scale factor, and the state estimation values, an adaptive trajectory tracking control law is constructed. The stability of the trajectory tracking control system is analyzed by the Lyapunov theorem. The effectiveness of the proposed control scheme is illustrated by simulations.**

***Index Terms*— Adaptive control, observer, recurrent neural network (NN), remotely operated vehicle (ROV), trajectory tracking.**

I. Introduction

A REMOTELY operated vehicle (ROV) is an important tool for marine resources development, which can be used to perform a variety of tasks in a complex marine environment. When the ROV executes pipeline tracking, manipulator operation, and other tasks, the trajectory tracking control is usually required [1], [2]. Hence, the research on the trajectory tracking control method will be of important practical significance to make ROV have good tracking performance.

An ROV may be equipped with different operation equipment when performing different tasks, and it is difficult to establish an accurate dynamic model for the ROV system due to the complexity of external environment [3]. Therefore, ROV is a typical complex uncertain nonlinear system. For the uncertain nonlinear control systems, adaptive control methods based on neural network (NN) have been extensively researched due to the nonlinear mapping ability of NN [4]–[8]. In addition, ROV trajectory tracking control is also an important application field [9]–[12]. The existing adaptive trajectory tracking control methods for the ROV system mostly focus on the method to guarantee the stability of the closed-loop system and the method to improve the accuracy of trajectory tracking, and assume that all the states of ROV system can be directly measured by sensors. However, sometimes, this assumption is difficult to be satisfied for the actual ROV systems. The existing ROVs are usually only equipped with compass, ultra-short baseline (USBL), and depth gauge, and only the position state and the orientation state in the earth-fixed frame can be measured, and thus the velocity state and the angular velocity state in the body-fixed frame cannot be measured directly [3], [13], [14]. Although the velocity state and the angular velocity state can be obtained by using the difference processing and coordinate transformation for the position state and the orientation state in theory [15], these data cannot be used in the adaptive trajectory tracking controller due to the serious noise disturbance from the USBL and the depth gauge [16]. Therefore, it is necessary to study the method to design an ROV trajectory tracking controller in the case of that the velocity state and the angular velocity state cannot be measured.

Due to the fact that the position state and the orientation state in the earth-fixed frame can be measured by sensors, the ROV trajectory tracking system in essence is observable. If the dynamics model of the ROV system has been known in advance, the state estimation can be performed by constructing a state observer. However, as mentioned earlier, the dynamics model of the ROV system is difficult to be established accurately. Therefore, a high gain observer was used for state estimation, and then, the adaptive NN trajectory tracking controller was designed based on the estimated value [17], [18]. Stability analysis showed that the high-gain observer-based adaptive controller can make state estimation errors and trajectory tracking errors uniformly ultimately bounded or asymptotically convergent. However, in order to ensure the convergence of the state estimation errors, the gain coefficient is usually large, and this large gain coefficient will be

Manuscript received August 13, 2014; revised February 18, 2016 and March 9, 2016; accepted March 18, 2016. Date of publication April 12, 2016; date of current version June 15, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 51509150 and Grant 51575336 and in part by the Shanghai Municipal Natural Science Foundation under Grant 15ZR1419700.

Z. Chu and D. Zhu are with the College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China (e-mail: chu_zhenzhong@163.com; zdq367@aliyun.com).

S. X. Yang is with the Advanced Robotics and Intelligent Systems Laboratory, University of Guelph, Guelph, ON N1G2W1, Canada (e-mail: syang@uoguelph.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2016.2544786

introduced to the control law. It will result in the system output oscillation and affect the tracking performance. In addition, the finite-time convergence is a very attractive feature for the state observer, which can obviously improve the tracking performance. However, it is usually unable to be met in the high-gain observer.

Normally, the output command of the most existing trajectory tracking controllers of ROVs are the desired thrust of thrusters [9]–[12], [19]–[22]. Then, according to the thrust, the thruster control signal can be calculated by the thrust model [15]. However, there is complex nonlinear relationship between the thrust and the control signal [23]. From the thrust modeling test in the pool environment, it can be found that the thrust is related to not only the control signal but also the relative velocity between the thruster and the water. In other words, with the change of the ROV motion state, even if the control signal of thruster remains unchanged, the thrust will change constantly. Although experimental measurements, interpolation, fitting, and other steps can be used for the approximate description of the thrust model, the calculation of the thruster control signal by the thrust value is still relatively cumbersome.

Motivated by the aforementioned observations, in this paper, an observer-based adaptive NN trajectory tracking control approach is proposed for the ROV. The main contributions of this paper are as follows.

1) Since the velocity states and the angular velocity states in the body-fixed frame are unmeasured, an adaptive terminal sliding-mode state observer is proposed for state estimation. Different from high-gain observer, it can make the trajectory tracking errors converge to zero in a finite time.
2) For the unknown dynamic model, a local recurrent NN (local RNN) is proposed for online identification. Compared with a three-layer neural network (three-layer NN) and a recurrent NN, it has a faster learning speed and a good learning performance.
3) Unlike most control approaches of ROV, the modeling error of thrust is considered in this paper, and a thrust model with an adaptive scale factor is given. Based on this model, the thruster control signal is used as the output of the trajectory tracking controller directly.
4) According to the estimated values of observer and the output of local RNN, the adaptive trajectory tracking controller is designed. It can make the tracking errors of position state and orientation state converge in a finite time, which is an obvious advantage compared with other observer-based adaptive control methods.

This paper is organized as follows. The ROV's trajectory tracking control problem is described in Section II. In Section III, the designing method of the observer-based ROV adaptive NN trajectory tracking controller is given. In Section IV, the effectiveness of the proposed method is verified by simulation. Finally, a brief conclusion on this paper is provided in Section V.

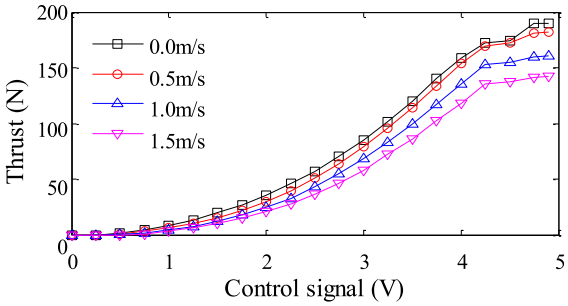

Fig. 1. Actual measured thrust curve.

II. Problem Formulation

ROV works in a complex marine environment. When it submerges to a certain depth, the extern disturbances mainly come from the ocean current. In this paper, it is assumed that the ocean current is constant and irrotational. The mathematical model of ROV with ocean current disturbance can be described as [24]

$$\begin{aligned}\dot{\eta} &= J(\eta)v \\ M_{\mathrm{RB}}\dot{v} + M_A\dot{v}_r + C_{\mathrm{RB}}(v)v + C_A(v_r)v_r &\\ + D(v_r)v_r + G(\eta) &= \bar{B}\tau \end{aligned} \quad (1)$$

where η denotes the vector of the position state and the orientation state in the earth-fixed frame, v is the vector of velocity state and the angular velocity state in the body-fixed frame, $v_r = v - J^{-1}(\eta)v_c$ is the relative velocity in the body-fixed frame, $J(\eta)$ is the kinematic transformation matrix, expressing the transformation from the body-fixed frame to the earth-fixed frame, $v_c = [v_{cx}, v_{cy}, v_{cz}, 0, 0, 0]^T$ is the velocity vector of ocean current in the earth-fixed frame, and $\dot{v}_c = 0$, M_{RB} is the rigid-body inertia matrix, $C_{\mathrm{RB}}(v)$ is the rigid-body centripetal and Coriolis matrix, M_A is the added inertia matrix, $C_A(v_r)$ is the matrix of hydrodynamic Coriolis and centripetal terms, $D(v_r)$ is the hydrodynamic damping term, $G(\eta)$ is the combined gravitational and buoyancy forces in the body-fixed frame, τ is the thruster thrust, and $\bar{B}$ is the distribution matrix of thrusters.

In [9]–[12] and [19]–[22], the mathematical model of the ROV is directly used to design the controller, in which the controller output is the thrust τ and the actual control signal of thruster is calculated by the thrust model. However, in practice, the thrust τ is a function about the thruster control signal and the velocity of the surrounding water, which has complex nonlinearity. Using a more frequently used thruster (Model 520) as an example, when the thruster moves at different relative velocities with the water, the actual relationship between the thrust and the control signal can be described as shown in Fig. 1.

In Fig. 1, there is complex nonlinear relationship between the thrust and the control signal. Meanwhile, for different thrusters, the thrust model may also be different. Therefore, it is difficult to use a simple mathematical equation to describe their relationship. According to the thrust curve as shown in Fig. 1, in this paper, a thrust model with a scale factor is proposed as

$$\tau_i = \gamma_i a \bar{u}_i |\bar{u}_i|, \quad i = 1, \ldots, n \quad (2)$$

where τ_i is the thrust of the ith thruster, $\bar{u}_i$ is the thruster control signal of the ith thruster, γ_i is a scale factor, n represents the number of thrusters, and $a > 0$ is a constant. Using the actual thrust data when the relative velocity of the thruster with the surrounding water is zero, a can be estimated by the least square method.

In (2), γ_i represents the impact of the ROV motion state. Since the motion state is continuous, γ_i would also be continuous. Therefore, an adaptive method for γ_i can be used for online learning, so that the accurate thrust model is not needed. Taking into account the simplicity and clarity of the subsequent analysis and description, let $u = [u_1, \ldots, u_i, \ldots, u_n]$ as the output of controller, where $u_i = a\bar{u}_i|\bar{u}_i|$. If u_i is known, then the thruster control signal can be obtained as

$$\bar{u}_i = \begin{cases} \sqrt{u_i/a} & u_i > 0 \\ 0 & u_i = 0 \\ -\sqrt{-u_i/a} & u_i < 0. \end{cases} \tag{3}$$

Define $x = [x_1; x_2]$, where $x_1 = \eta, x_2 = J(\eta)v$. According to (1)–(3), one can obtain

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x) + B(x_1)\gamma u \end{aligned} \tag{4}$$

where $\gamma = \text{diag}([\gamma_1, \ldots, \gamma_n])$. $f(x)$ and $B(x_1)$ are shown in (5) and (6), respectively

$$\begin{aligned} f(x) = {} & J'(x_1)v + J(x_1)(M_{\text{RB}} + M_A)^{-1} \\ & \times [M_A(J^{-1}(x_1))'v_c - C_{\text{RB}}(J^{-1}(x_1)x_2)J^{-1}(x_1)x_2 \\ & - C_A(J^{-1}(x_1)x_2 - J^{-1}(x_1)v_c) \\ & \times (J^{-1}(x_1)x_2 - J^{-1}(x_1)v_c) \\ & - D(J^{-1}(x_1)x_2 - J^{-1}(x_1)v_c) \\ & \times (J^{-1}(x_1)x_2 - J^{-1}(x_1)v_c) - G(x_1)] \end{aligned} \tag{5}$$

$$B(x_1) = J(x_1)(M_{\text{RB}} + M_A)^{-1}\bar{B} \tag{6}$$

where $J'(\eta)$ and $(J^{-1}(\eta))'$ are the derivatives of $J(\eta)$ and $J^{-1}(\eta)$, respectively.

As can be seen from (4), by introducing the scale factor for the thrust model, an affine nonlinear system is obtained. Thus, the thruster control signal can be used as the output of the trajectory tracking controller directly, which is very convenient for designing the controller. Due to the external ocean current disturbance and the change of portable equipment of the ROV, it is difficult to establish an accurate dynamic model for the ROV. Thus, the nonlinear function $f(x)$ is difficult to be obtained, while $B(x_1)$ is a function about $J(\eta)$, M_{RB}, M_A, and $\bar{B}$, which are all easy to get. Therefore, $B(x_1)$ is known. Obviously, for a practical ROV system, the nonlinear function $f(x)$ is continuously differentiable. Similar to a lot of other studies, an NN will be used for online learning for $f(x)$. In addition, in trajectory tracking controller designing, it needs to get the velocity state and the angular velocity state in the body-fixed frame. These states can be measured by the Doppler velocity logger, gyroscope, and other sensors directly. However, these sensors with high precision are usually very expensive for the majority of small ROVs. In most cases, the depth gauge, compass, and USBL are used to measure the position state and the orientation states in the earth-fixed frame. That is to say, x_2 in model (4) is unknown. Therefore, a state observer will be designed to estimate the unmeasured states.

In this paper, the control objective is to design the trajectory tracking control law u under the situation that the thrust model is inaccurate, ROV dynamic model is unknown, and the state x_2 is unmeasured. In addition, the designed control law can make the trajectory tracking error converge to zero in a finite time.

III. Controller Designing

In this section, considering that the velocity state and the angular velocity state in the body-fixed frame cannot be directly measured and the dynamics model is unknown, an adaptive terminal sliding-mode state observer and the observer-based adaptive controller will be designed.

In order to obtain the actual dynamic of the unknown nonlinear function $f(x)$ in (4), adaptive NN learning is a more commonly used method. Among them, most NNs belong to forward NNs, such as radial basis function (RBF) NN [25], back propagation (BP) NN [26], fuzzy NN [27], and so on. An RBF NN has a better learning speed if the center and bandwidth of the RBF can be preselected by the teaching sample, but it is very difficult to obtain an adequate sample for the ROV because of complex marine environment. A fuzzy NN also has a better learning speed and fairly well capability of nonlinear mapping, but it excessively depends on the expert's knowledge. A BP NN is easy to be implemented, but the learning speed is much worse than the first two NNs. In addition, there are two problems that make forward NNs unsuitable for the ROV adaptive controller. The first problem is that the USBL and other sensors are affected by noise disturbance, and the collected data have large jitter. If these data are used as the input of NN, the output of NN will also appear jitter, and thus, the tracking error may be unable to converge. The other problem is that the NN weights need to be adjusted in real time according to the tracking error during the online learning. If the desired trajectory has a sudden change, the trajectory tracking error will also have a sudden change. It may cause the uncertainty of NN output, and it is more likely to cause the unpredictability of control law, and then affect the control performance.

In order to make NN have the ability to suppress noise and have a better transition when the desired trajectory changes suddenly, a viable solution is to use a recurrent NN [20]. In the recurrent NN, the sequence relationship between the input variable and the output variable is reflected by the regression structure from the hidden layer to the recurrent layer. It can reflect the connection between the last time states and the current time states. However, the traditional Elman NN has a lower learning efficiency. Because of the lower sampling frequency and fewer learning samples of the ROV system, the lower learning efficiency is difficult to meet the real-time requirement for trajectory tracking control.

The simulation for offline identification of recurrent NN shows that the number of recurrent layer neurons is an

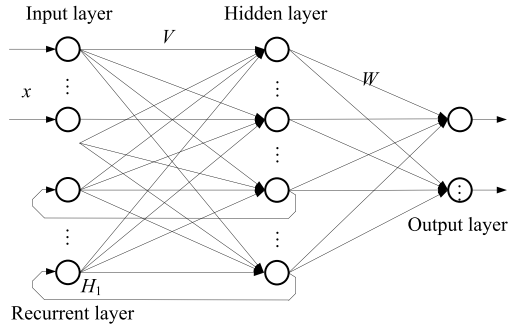

Fig. 2. Structure of recurrent NN.

TABLE I
LEARNING TIMES FOR DIFFERENT RECURRENT LAYER NEURON NUMBERS

Neuron number	2	4	6	8	10	12
Learning times	3483	300	134	197	287	318

important factor that affects the learning efficiency. To describe the influence of the number of recurrent layer neuron on learning speed, the function $y(t) = 15x|x| + 2x$, $x \in [-1, 1]$ has been trained. The structure of the recurrent NN is shown in Fig. 2, where H_1 is the output vector of the recurrent layer neurons. It is equal to the output of the hidden layer neurons with a recurrent structure. The number of hidden layer neurons is 12, and the number of recurrent layer neurons is chosen as 2, 4, 6, 8, 10, and 12. If the number of the recurrent layer neurons is 2, 4, 6, 8, or 10, then there are only 2, 4, 6, 8, or 10 hidden layer neurons regressing to recurrent layer, respectively. W is the weight matrix between the hidden layer neurons and the output layer neurons. V is the weight matrix among the input layer neurons, the recurrent layer neurons, and the hidden layer neurons. The size of both the teaching sample and the testing sample is 100. After 50 learning times, the training results are shown in Fig. 3. Set the desired error energy as SUM$(0.5e^2) = 1$, where e is learning error. The learning times for different number of recurrent layer neurons are shown in Table I.

Simulation results show that it is not needed to make all the hidden layer neurons regress to the recurrent layer, but only some of them, which can simplify the network structure and improve the learning efficiency. In addition, when the number of the recurrent layer neurons is 6, recurrent NN has the fastest learning speed by retaining a good learning performance. Thus, as shown in Fig. 2, one called local RNN [28] will be used for adaptive learning, and the number of recurrent layer neurons is chosen as half the number of hidden layer neurons.

For the local RNN, as shown in Fig. 2, because of the nonlinear mapping ability of NN, there are optimal network weights W and V, such that

$$f(x) = W\varphi(VH) + \varepsilon \tag{7}$$

where $H = [x;\ H_1]$ is the input of hidden layer neurons, $x = [x_1;\ x_2]$ is the input vector of the input layer neurons, ε is the estimation error of the optimal NN, and $\varphi(.)$ is the sigmoid function.

Using the local RNN, the designed adaptive terminal sliding-mode state observer is given as

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 - \alpha \\ \dot{\hat{x}}_2 &= \hat{f}(\hat{x}) + B(\hat{x}_1)\hat{\gamma}u - k_2\alpha^{q_2/p_2} - k_4\text{sgn}(\alpha)\end{aligned} \tag{8}$$

where

$$\alpha = \int_0^t \left[k_1\text{sgn}(s) + l^{-1}(q_1/p_1)\dot{\tilde{x}}_1^{2-p_1/q_1}\right]dt \tag{9}$$

$$s = \tilde{x}_1 + l\dot{\tilde{x}}_1^{p_1/q_1} \tag{10}$$

where $\hat{x} = [\hat{x}_1; \hat{x}_2]$ is the estimated value of x; $\tilde{x}_1 = \hat{x}_1 - x_1$ and $\tilde{x}_2 = \hat{x}_2 - x_2$ are the state estimation errors; k_2, k_3, and l are the positive constants; k_1 and k_4 are the adaptive parameters for some variables; and the update laws will be given later. p_1, q_1, p_2, and q_2 are the positive odd constants, and $p_1 > q_1$, $1 < p_1/q_1 < 2$, $p_2 > q_2$, and $1 < p_2/q_2 < 2$. $\hat{\gamma}$ is the estimated value of γ. $\hat{f}(\hat{x})$ is the output of the local RNN

$$\hat{f}(\hat{x}) = \hat{W}\varphi(\hat{V}\hat{H}) \tag{11}$$

where $\hat{H} = [\hat{x}; H_1]$, and $\hat{W}$ and $\hat{V}$ are the estimated values of W and V, respectively.

Remark 1: In (8), the sliding-mode manifold is designed as (10), which is often referred to as a nonsingular terminal sliding mode [29]. In (10), $\tilde{x}_1$ can be directly calculated by the measured value x_1 and the estimated value $\hat{x}_1$, and $\dot{\tilde{x}}_1$ can be obtained by taking the derivative of $\tilde{x}_1$.

Remark 2: For observer (8), a terminal sliding mode is used to let the estimation error $\tilde{x}_1$ converge in a finite time. This is obviously different from the high gain observer. If tracking error $e_1 = \hat{x}_1 - x_{1d}$ can converge in a finite time, the trajectory tracking error $\delta_1 = x_1 - x_{1d}$ will also converge in a finite time.

Next, the main conclusion will be presented as follows.

Proposition 1: Considering the mathematical model of the ROV expressed as (1), if the adaptive terminal sliding-mode state observer (8), the control law (12), and the update laws (13)–(17) are chosen and the control parameters satisfy (18), the trajectory tracking error $\delta_1 = x_1 - x_{1d}$ will converge to zero in a finite time. In addition, the state estimated error $\tilde{x}_2$ is asymptotically convergent

$$\begin{aligned}u = B_{\hat{\gamma}}^{-1}[&-e_1 - \hat{f}(\hat{x}) + k_2\alpha^{q_2/p_2} + k_4\text{sgn}(\alpha) \\ &+ (\beta e_2)^{q_3/p_3} - k_7\text{sgn}(e_2) - k_3e_2]\end{aligned} \tag{12}$$

$$\dot{k}_1 = \lambda_{k1}l(p_1/q_1)\left\|\dot{\tilde{x}}_1^{p_1/q_1-1}\right\|\|s\| \tag{13}$$

$$\dot{k}_4 = \lambda_{k4}\|\alpha\| \tag{14}$$

$$\dot{\hat{W}} = -\lambda_W\alpha\varphi(\hat{V}\hat{H})^T \tag{15}$$

$$\dot{\hat{V}} = -\lambda_V\varphi'(\hat{V}\hat{H})\hat{W}^T\alpha\hat{H}^T \tag{16}$$

$$\dot{\hat{\gamma}} = -\lambda_\gamma\,\text{diag}(\alpha^T B(\hat{x}_1))\text{diag}(u) \tag{17}$$

$$\begin{aligned}k_7 \geq\ &\|\varepsilon_0[\varepsilon_0 I + B(\hat{x}_1)\hat{\gamma}(B(\hat{x}_1)\hat{\gamma})^T]^{-1} \\ &\times[-e_1 - \hat{f}(\hat{x}) + k_2\alpha^{q_2/p_2} + k_4\text{sgn}(\alpha) \\ &+ (\beta e_2)^{q_3/p_3} - k_3e_2]\| + \theta_1\end{aligned} \tag{18}$$

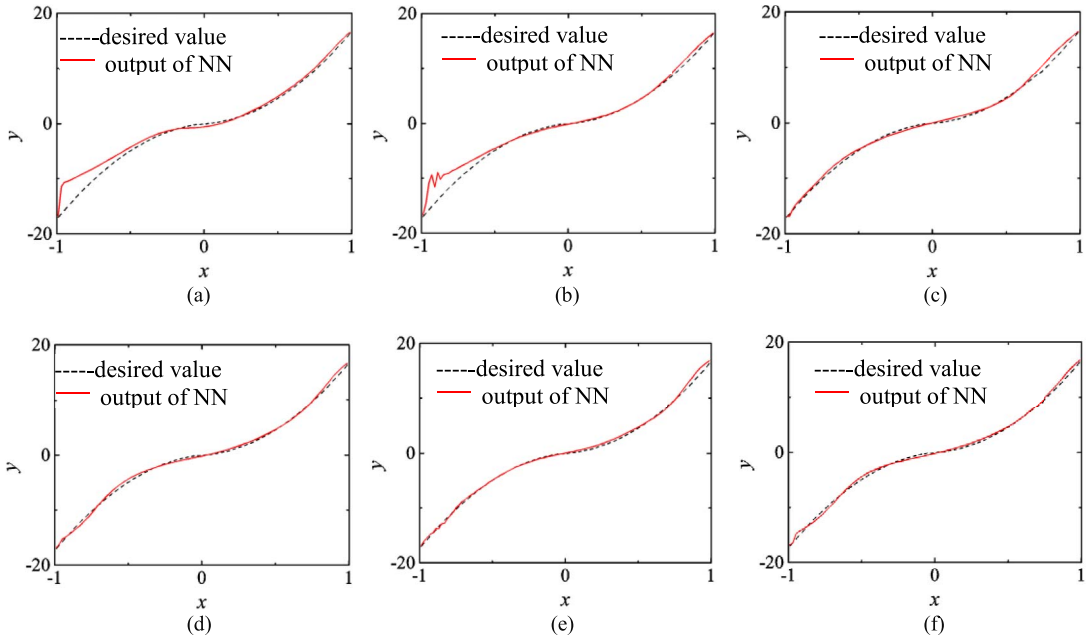

Fig. 3. Training results of recurrent NN. Neuron number of recurrent layer is (a) 2, (b) 4, (c) 6, (d) 8, (e) 10, and (f) 12.

where $e_1 = \hat{x}_1 - x_{1d}$, $e_2 = \hat{x}_2 - \hat{x}_{2d}$, x_{1d} is the desired trajectory, and the expression of variable $\hat{x}_{2d}$ will be given later. λ_W, λ_V, λ_γ, λ_{k1}, and λ_{k4} are all known positive constants. Both p_3 and q_3 are the positive odd constants, $p_3 > q_3$, and $1 < p_3/q_3 < 2$. ρ_0, θ_1, β, and ε_0 are the positive constants and diag(u) is a diagonal matrix with the elements of vector u. $B_{\hat{\gamma}}^{-1}$ is designed as

$$B_{\hat{\gamma}}^{-1} = (B(\hat{x}_1)\hat{\gamma})^T[\rho_0 I + B(\hat{x}_1)\hat{\gamma}(B(\hat{x}_1)\hat{\gamma})^T]^{-1}. \quad (19)$$

Remark 3: According to (1) and (6), $B(x_1)$ and γ are both full rank and greater than zero. However, the update law (17) cannot ensure $\gamma > 0$, that is $B(\hat{x}_1)\hat{\gamma}$ may be singular. For this problem, the regularized inverse is used as (19). Since ρ_0 is a very small positive constant, $B_{\hat{\gamma}}^{-1}$ is approximated to $(B(\hat{x}_1)\hat{\gamma})^{-1}$. Even if $B(\hat{x}_1)\hat{\gamma}$ is singular, the control law (12) is well defined [30].

Remark 4: The existence of (12) depends on the state $\hat{x}$, which can be estimated by observer (8). In (8), only the position state and the orientation state in the earth-fixed frame need to be known. For the parameters in (13)–(18), they can all be selected in advance or estimated online. Therefore, although the velocity state and the angular velocity state in the body-fixed frame cannot be measured directly, the control law (12) is still viable.

Proof: First, it will be proved that the state estimation error $\tilde{x}_1$ can converge to zero in a finite time.

According to (4) and (8), it can obtain

$$\dot{\tilde{x}}_1 = \tilde{x}_2 - \alpha. \quad (20)$$

Consider the Lyapunov function

$$V_1 = s^T s + \tilde{k}_1 \lambda_{k1}^{-1} \tilde{k}_1 \quad (21)$$

where $\tilde{k}_1 = k_1 - k_1^*$ and k_1^* is the upper bound of $\|\dot{\tilde{x}}_2\|$.

Remark 5: In (9), the sliding-mode item $k_1 \text{sgn}(s)$ is used to compensate the unknown item $\dot{\tilde{x}}_2$. Since the upper bound of $\|\dot{\tilde{x}}_2\|$ is unknown, we assume that there is an upper bound k_1^* for $\|\dot{\tilde{x}}_2\|$, and an adaptive method is used.

Differentiating V_1 with respect to time and using (8)–(10) and (20), one can obtain

$$\begin{aligned}\dot{V}_1 &= 2s^T\dot{s} + 2\tilde{k}_1\lambda_{k1}^{-1}\dot{\tilde{k}}_1 \\ &= 2s^T \text{diag}\big(l(p_1/q_1)\dot{\tilde{x}}_1^{p_1/q_1-1}\big) \\ &\quad \times\big[l^{-1}(q_1/p_1)\dot{\tilde{x}}_1^{2-p_1/q_1} + \dot{\tilde{x}}_2 - \dot{\alpha}\big] + 2\tilde{k}_1\lambda_{k1}^{-1}\dot{\tilde{k}}_1 \\ &= 2s^T \text{diag}\big(l(p_1/q_1)\dot{\tilde{x}}_1^{p_1/q_1-1}\big)[\dot{\tilde{x}}_2 - k_1\text{sgn}(s)] \\ &\quad + 2\tilde{k}_1\lambda_{k1}^{-1}\dot{\tilde{k}}_1. \end{aligned} \quad (22)$$

In (22), $\dot{\tilde{x}}_2$ is an unknown state. Here, $\dot{\tilde{x}}_2$ is supposed to be bounded as $\|\dot{\tilde{x}}_2\| \le k_1^*$. And the upper bound of $\dot{\tilde{x}}_2$ is adaptively learned by k_1. Substituting the update law (13) into (22), since $l(p_1/q_1)\dot{\tilde{x}}_1^{p_1/q_1-1} \ge 0$, we can get $\dot{V}_1 \le 0$. That is, s and $\tilde{k}_1$ are both bounded. Next, the finite-time convergence of s will be further proved. Here, the upper bound of k_1 is supposed to be k_{10}. Consider the Lyapunov function

$$V_2 = s^T s + (k_1 - k_{10})^2. \quad (23)$$

Differentiating V_2 with respect to time and using (8) and (9),

one can obtain

$$\begin{aligned}\dot{V}_2 &= 2s^T \operatorname{diag}\big(l(p_1/q_1)\dot{\tilde{x}}_1^{p_1/q_1-1}\big)[\dot{\tilde{x}}_2 - k_1\operatorname{sgn}(s)] \\ &\quad + 2(k_1-k_{10})\lambda_{k1}l(p_1/q_1)\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\| \\ &\le 2l(p_1/q_1)\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\|k_1^* - 2l(p_1/q_1)\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\|k_1 \\ &\quad + 2l(p_1/q_1)\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\|k_{10} - 2l(p_1/q_1) \\ &\quad \times \big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\|k_{10} \\ &\quad + 2(k_1-k_{10})\lambda_{k1}l(p_1/q_1)\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\| \\ &= -2l(p_1/q_1)\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\|(k_{10}-k_1^*) \\ &\quad - 2l(p_1/q_1)\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|\|s\|(\lambda_{k1}-1)(k_{10}-k_1). \end{aligned} \tag{24}$$

In (24), there always exists $k_{10} > 0$, such that $k_{10} \ge k_1$ and $k_{10} \ge k_1^*$ for all $t > 0$ [31]. If we choose $\lambda_{k1} > 1$, then

$$\begin{aligned}\dot{V}_2 \le &-2l\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|(p_1/q_1) \\ &\times \min((k_{10}-k_1^*), (\lambda_{k1}-1)\|s\|)V_2^{1/2}. \end{aligned} \tag{25}$$

Equation (25) shows that s can converge to zero in a finite time

$$t_1 = \frac{V_2^{1/2}(0)}{l\big\|\dot{\tilde{x}}_1^{p_1/q_1-1}\big\|(p_1/q_1)\min((k_{10}-k_1^*), (\lambda_{k1}-1)\|s\|)} \tag{26}$$

where $V_2(0)$ is the initial value of V_2.

After s converges to zero, $\tilde{x}_1$ and $\dot{\tilde{x}}_1$ will converge to zero in a finite time

$$t_2 = t_1 + p_1/(p_1-q_1)l^{q_1/p_1}\|\tilde{x}_1(0)\|^{(p_1-q_1)/p_1} \tag{27}$$

where $\tilde{x}_1(0)$ is the initial value of $\tilde{x}_1$.

After $\tilde{x}_1$ and $\dot{\tilde{x}}_1$ converge to zero, $\tilde{x}_2 = \alpha$ can be obtained by (20). If the controller tracking error $e_1 = \hat{x}_1 - x_{1d}$ can converge in a finite time, the trajectory tracking error $\delta_1 = x_1 - x_{1d}$ will also converge in a finite time. Therefore, the convergence of e_1 will be analyzed next.

For the desired trajectory x_{1d}, according to (8), one can get

$$\dot{e}_1 = \hat{x}_2 - \alpha - \dot{x}_{1d}. \tag{28}$$

Consider the Lyapunov function

$$V_3 = e_1^T e_1. \tag{29}$$

Differentiating V_3 and using (28), we can get

$$\dot{V}_3 = 2e_1^T\dot{e}_1 = 2e_1^T(\hat{x}_2 - \alpha - \dot{x}_{1d}). \tag{30}$$

Choose virtual control law

$$\hat{x}_{2d} = \alpha + \dot{x}_{1d} - k_5e_1 - k_6\operatorname{sgn}(e_1) \tag{31}$$

where both k_5 and k_6 are the positive constants.

Substitute (31) into (30), then

$$\dot{V}_3 = -2e_1^Tk_5e_1 - 2e_1^Tk_6\operatorname{sgn}(e_1) \le -2k_6V_3^{1/2}. \tag{32}$$

From (32), if $\hat{x}_2 = \hat{x}_{2d}$, e_1 will converge to zero within a limited time

$$t_3 = V_3^{1/2}(0)/k_6 \tag{33}$$

where $V_3(0)$ is the initial value of V_3.

However, $\hat{x}_{2d}$ is only a virtual control law, and there is tracking error $e_2 = \hat{x}_2 - \hat{x}_{2d}$. In order to make e_1 converge in a finite time, e_2 should converge first. Therefore, the real control law (12) will be analyzed later to prove that e_2 can converge in a finite time.

In control law designing, in order to avoid the emergence of higher order derivative terms, the first-order low-pass filter is usually used for processing [32]. However, the low-pass filter has asymptotical convergence, so that $\hat{x}_2$ cannot converge to the true $\hat{x}_{2d}$ immediately and (31) cannot be guaranteed perfectly. In this paper, the fractional order filter (34) is used to make $\hat{x}_2$ converge in a finite time

$$\hat{x}_2 = \hat{x}_{2d} + \frac{1}{\beta}\dot{\hat{x}}_{2d}^{p_3/q_3}. \tag{34}$$

According to (8) and (34), one can obtain

$$\dot{e}_2 = \hat{f}(\hat{x}) + B(\hat{x}_1)\hat{\gamma}u - k_2\alpha^{q_2/p_2} - k_4\operatorname{sgn}(\alpha) - (\beta e_2)^{q_3/p_3}. \tag{35}$$

Consider the Lyapunov function

$$V_4 = e_1^Te_1 + e_2^Te_2. \tag{36}$$

Differentiating V_4 with respect to time and using (28) and (35), one can obtain

$$\begin{aligned}\dot{V}_4 &= 2e_1^T\dot{e}_1 + 2e_2^T\dot{e}_2 \\ &\le -2e_1^Tk_5e_1 - 2e_1^Tk_6\operatorname{sgn}(e_1) + 2e_2^T[-\theta_1\operatorname{sgn}(e_2) - k_3e_2] \\ &\le -2\min(k_6, \theta_1)V_4^{1/2}. \end{aligned} \tag{37}$$

From (37), it can be known that e_1 and e_2 will converge to zero in a finite time

$$t_4 = V_4^{1/2}(0)/\min(k_6, \theta_1) \tag{38}$$

where $V_4(0)$ is the initial value of V_4.

Since $\tilde{x}_1$ and e_1 both can converge in a finite time, the trajectory tracking error $\delta_1 = x_1 - x_{1d}$ will also converge in a finite time. Next, the convergence of $\tilde{x}_2$ will be analyzed.

Base on the Lyapunov analysis of V_2, it has been known that $\tilde{x}_1$ can converge to zero in a finite time. Then, $B(\hat{x}_1) = B(x_1)$ and $\tilde{x}_2 = \alpha$ can be obtained. According to (4) and (8), the state estimation error of x_2 can be expressed as

$$\dot{\tilde{x}}_2 = \hat{f}(\hat{x}) - f(x) + B(\hat{x}_1)\tilde{\gamma}u - k_2\tilde{x}_2^{q_2/p_2} - k_4\operatorname{sgn}(\tilde{x}_2) \tag{39}$$

where $\tilde{\gamma} = \hat{\gamma} - \gamma$ is the estimation error.

According to the output of NN (7) and (11), the estimation error of the NN can be expressed as

$$\begin{aligned}\hat{f}(\hat{x}) - f(x) = &\tilde{W}\varphi(\hat{V}\hat{H}) + W\varphi(\hat{V}\hat{H}) - W\varphi(V\hat{H}) \\ &+ W\varphi(V\hat{H}) - W\varphi(VH) - \varepsilon \end{aligned} \tag{40}$$

where $\tilde{W} = \hat{W} - W$ and $\tilde{V} = \hat{V} - V$ are the estimation errors.

Take the Taylor expansion of $\varphi(V\hat{H})$ to about $\hat{V}\hat{H}$

$$\varphi(V\hat{H}) = \varphi(\hat{V}\hat{H}) - \varphi'(\hat{V}\hat{H})\tilde{V}\hat{H} + o^2(\tilde{V}\hat{H}) \tag{41}$$

where $\varphi'(\hat{V}\hat{H}) = \partial\varphi(V\hat{H})/\partial V\hat{H}|_{V\hat{H}=\hat{V}\hat{H}}$. $o^2(\tilde{V}\hat{H})$ is the high-order item of the Taylor expansion.

Substitute (41) into (40), then

$$\hat{f}(\hat{x}) - f(x) = \tilde{W}\varphi(\hat{V}\hat{H}) + \hat{W}\varphi'(\hat{V}\hat{H})\tilde{V}\hat{H} + \omega_1 \tag{42}$$

where

$$\omega_1 = \tilde{W}\varphi'(\hat{V}\hat{H})\tilde{V}\hat{H} - Wo^2(\tilde{V}\hat{H}) + W\varphi(V\hat{H}) - W\varphi(VH) - \varepsilon. \quad (43)$$

Consider the Lyapunov function

$$V_5 = \tilde{x}_2^T\tilde{x}_2 + \mathrm{tr}\big[\tilde{W}^T\lambda_W^{-1}\tilde{W}\big] + \mathrm{tr}\big[\tilde{V}^T\lambda_V^{-1}\tilde{V}\big] + \mathrm{tr}\big[\tilde{\gamma}^T\lambda_\gamma^{-1}\tilde{\gamma}\big] + \tilde{k}_4\lambda_{k4}^{-1}\tilde{k}_4 \quad (44)$$

where $\tilde{k}_4 = k_4 - k_4^*$ and k_4^* is the upper bound of $\|\omega_1\|$.

Remark 6: In most papers, they usually assume that the upper bound of $\|\omega_1\|$ is known. Obviously, this assumption is not reasonable. In this paper, we just assume that there is an upper bound for $\|\omega_1\|$, and an adaptive method is used to estimate the upper bound of $\|\omega_1\|$.

Differentiating V_5 with respect to time and using (39), one can obtain

$$\begin{aligned}\dot{V}_5 &= 2\tilde{x}_2^T\dot{\tilde{x}}_2 + 2\mathrm{tr}\big[\tilde{W}^T\lambda_W^{-1}\dot{\tilde{W}}\big] + 2\mathrm{tr}\big[\tilde{V}^T\lambda_V^{-1}\dot{\tilde{V}}\big] \\ &\quad + 2\mathrm{tr}\big[\tilde{\gamma}^T\lambda_\gamma^{-1}\dot{\tilde{\gamma}}\big] + 2\tilde{k}_4\lambda_{k4}^{-1}\dot{\tilde{k}}_4 \\ &= 2\tilde{x}_2^T\big(\hat{f}(\hat{x}) - f(x) + B(\hat{x}_1)\tilde{\gamma}u - k_2\tilde{x}_2^{q_2/p_2} - k_4\mathrm{sgn}(\tilde{x}_2)\big) \\ &\quad + 2\mathrm{tr}\big[\tilde{W}^T\lambda_W^{-1}\dot{\tilde{W}}\big] + 2\mathrm{tr}\big[\tilde{V}^T\lambda_V^{-1}\dot{\tilde{V}}\big] \\ &\quad + 2\mathrm{tr}\big[\tilde{\gamma}^T\lambda_\gamma^{-1}\dot{\tilde{\gamma}}\big] + 2\tilde{k}_4\lambda_{k4}^{-1}\dot{\tilde{k}}_4. \end{aligned} \quad (45)$$

Substituting (42) into (45)

$$\begin{aligned}\dot{V}_5 &= 2\tilde{x}_2^T\tilde{W}\varphi(\hat{V}\hat{H}) + 2\tilde{x}_2^T\hat{W}\varphi'(\hat{V}\hat{H})\tilde{V}\hat{H} \\ &\quad + 2\tilde{x}_2^T B(\hat{x}_1)\tilde{\gamma}u - 2\tilde{x}_2^T\big(\omega_1 - k_2\tilde{x}_2^{q_2/p_2} - k_4\mathrm{sgn}(\tilde{x}_2)\big) \\ &\quad + 2\mathrm{tr}\big[\tilde{W}^T\lambda_W^{-1}\dot{\tilde{W}}\big] + 2\mathrm{tr}\big[\tilde{V}^T\lambda_V^{-1}\dot{\tilde{V}}\big] \\ &\quad + 2\mathrm{tr}\big[\tilde{\gamma}^T\lambda_\gamma^{-1}\dot{\tilde{\gamma}}\big] + 2\tilde{k}_4\lambda_{k4}^{-1}\dot{\tilde{k}}_4. \end{aligned} \quad (46)$$

Substituting (15)–(17) into (46) and since $\tilde{x}_2 = \alpha$

$$\dot{V}_5 = 2\tilde{x}_2^T\big(\omega_1 - k_2\tilde{x}_2^{q_2/p_2} - k_4\mathrm{sgn}(\tilde{x}_2)\big) + 2\tilde{k}_4\lambda_{k4}^{-1}\dot{\tilde{k}}_4. \quad (47)$$

Substituting (14) into (47)

$$\dot{V}_5 \le -2\tilde{x}_2^T k_2\tilde{x}_2^{q_2/p_2} \le 0. \quad (48)$$

According to (48), it can be known that $\tilde{x}_2$ is asymptotical convergent, and $\tilde{W}$, $\tilde{V}$, $\tilde{\gamma}$, and $\tilde{k}_4$ are all uniformly ultimately bounded.

Above proofs show that e_1, e_2, and $\tilde{x}_1$ can converge to zero in a finite time and $\tilde{x}_2$ is asymptotically convergent. Therefore, the trajectory tracking error $\delta_1 = x_1 - x_{1d}$ can converge to zero in a finite time. It shows that Proposition 1 is true.

Remark 7: In (8) and (12), k_2, k_3, k_5, k_6, and θ_1 are all positive constants. Above stability analysis shows that the values of these parameters do not affect the stability of the closed-loop system but only affect the convergence time. However, for a practical trajectory tracking control problem, it is not true. The desired convergence time cannot be very short. If the desired convergence time is very short, the closed-loop system may be instable because of the saturation constraint of the thruster. In addition, even though the desired convergence time is feasible, the transient performance is also needed to be considered. Therefore, a suitable desired convergence time should be selected according to maximum maneuver capability, and then k_2, k_3, k_5, k_6, and θ_1 can be calculated based on (37) and (48). In Section IV, a more detailed calculation process of these parameters will be given for the specific problem.

IV. Simulation Results

In order to verify the effectiveness of the proposed method, the simulations are carried out. For most ROV systems, the distance between the gravity and the buoyancy is usually far and the changes of the roll angle and the pitch angle are small. Thus, the trajectory tracking control of the ROV can be decomposed into two independent parts: trajectory tracking control in horizontal surface and trajectory tracking control in vertical surface. In this paper, only horizontal surface trajectory tracking control is carried out, namely, the trajectory tracking control of position X, position Y, and yaw angle in the earth-fixed frame. The parameters of dynamic model of surge degree of freedom (DOF), sway DOF, and yaw DOF in the body-fixed frame are chosen as [33]: $M_{\mathrm{RB}} = \mathrm{diag}([125, 125, 4.63])$, $M_A = \mathrm{diag}([62.4, 62.4, 0])$, $G(\eta) = 0$, $v_c = [0.2, 0.1, 0]$, $u_i = 20\bar{u}_i|\bar{u}_i|$, $\tau = \mathrm{diag}([1 - v_{1r}/8,\ 1 - v_{1r}/8,\ 1 - v_{2r}/8,\ 1 - v_{2r}/8])u$, $C_{\mathrm{RB}}(v) = [0, 0, -125v_2;\ 0,\ 0,\ 125v_1;\ 125v_2,\ -125v_1, 0]$, $C_A(v_r) = [0, 0, -125v_{2r};\ 0,\ 0,\ 125v_{1r};\ 125v_{2r},\ -125v_{1r}, 0]$, $D_r(v_r) = \mathrm{diag}([148|v_{1r}| + 100,\ 148|v_{2r}| + 100,\ 280|v_{3r}| + 230])$, and $\bar{B} = [1, 1, 0, 0;\ 0, 0, 1, 1;\ 0.38, -0.38, 0.38, -0.38]$, where v_1, v_2, and v_3 represent the surge speed, sway speed, and yaw angular velocity in the body-fixed frame, respectively. v_{1r}, v_{2r}, and v_{3r} represent the relative velocity in the body-fixed frame. The maximum value and minimum value of u_i are 100 and -100, respectively.

The initial position state and orientation state of ROV are $x_1 = [x_{11}, x_{12}, x_{13}]^T = [10, 0, 0]^T$, and the initial estimation values are $\hat{x}_1 = [\hat{x}_{11}, \hat{x}_{12}, \hat{x}_{13}]^T = [5, 2, 0]^T$, where x_{11}, x_{12}, and x_{13} are position X, position Y, and yaw angle in the earth-fixed frame, respectively. The desired trajectory is chosen as

$$x_{11d} = \begin{cases} 3\sin(2\pi t/40) & t \le 40s \\ 3 & 40s < t \le 60s \\ 5 & 60s < t \le 100s \end{cases}$$

$$x_{12d} = \begin{cases} 3\cos(2\pi t/40) & t \le 40s \\ 5 & 40s < t \le 60s \\ 3 & 60s < t \le 100s \end{cases}$$

$$x_{13d} = \begin{cases} 0 & t \le 30s \\ -0.1 & 30s < t \le 80s \\ 0.2\sin(2\pi t/40) & 80s < t \le 100s. \end{cases}$$

A. Controller Parameters

In control law (12), there are some controller parameters that should be chosen first according to (13)–(18), so that the computational burden may be heavy in each control period. However, the control frequency of ROV is usually set from 1 to 10 Hz, and the frequency of the ROV's computer $i >$ 1 GHz. Hence, the processing of this computation is more

than sufficient. Next, according to the controlled system, the selecting methods of controller parameters will be given.

1) First, $p_1 = p_2 = p_3 = 5$ and $q_1 = q_2 = q_3 = 3$ are chosen, and these values have been frequently used in other terminal sliding-mode methods.
2) l determines the convergence time of $\tilde{x}_1$ after $s = 0$. The convergence time t_2 can be calculated as (27). At the initial time, the maximum value of $\tilde{x}_1(0)$ is 5. If $t_2 < 2$ s needs to be satisfied, l will be chosen as $l = 0.24$.
3) It can be seen from (48) that the convergence speed of $\tilde{x}_2$ is in connection with k_2. If $\tilde{W}$, $\tilde{V}$, $\tilde{\gamma}$, and $\tilde{k}_4$ are all supposed to be zero, then $\dot{\tilde{x}}_2 \leq -k_2\tilde{x}_2^{q_2/p_2}$, so that $\tilde{x}_2$ will converge to zero in $t_5 = k_2^{-1}p_2/(p_2 - q_2)\|\tilde{x}_2(0)\|^{(p_2-q_2)/p_2}$. If $t_5 < 0.5$ s needs to be satisfied when $\|\tilde{x}_2(0)\| = 1$, k_2 will be chosen as $k_2 = 5$.
4) k_3, k_5, k_6, and θ_1 are all in connection with the convergence speed of e_1 and e_2. According to (37), supposing $e_1 = 0$, $e_2(0) = 1$, and $k_5 = k_6 = \theta_1 = 0$, $\dot{e}_2 \leq -k_3e_2$. If e_2 needs to converge to the 5% of initial value in 4s, k_3 should be chosen as $k_3 = 0.75$. Similarly, k_5 can be chosen as $k_5 = 0.75$. Supposing $e_2 = 0$ and $k_3 = k_5 = \theta_1 = 0$, $e_1\dot{e}_1 \leq -k_6\|e_1\|$. If $e_1(0)$ needs to converge to zero in 2 s when $e_1(0) = 1$, then k_6 should be chosen as $k_6 = 0.5$. Similarly, θ_1 can be chosen as $\theta_1 = 0.5$.
5) λ_W, λ_V, λ_γ, λ_{k1}, and λ_{k4} are all in connection with the learning rate of $\tilde{W}$, $\tilde{V}$, $\tilde{\gamma}$, $\tilde{k}_1$, and $\tilde{k}_4$, respectively, where λ_{k1} should be >1. In simulation, $\lambda_{k1} = 1.5$. Through several attempts, $\lambda_W = \lambda_V = \lambda_\gamma = \lambda_{k4} = 0.05$ are chosen.
6) ρ_0 is a small positive constant, which is chosen as $\rho_0 = 0.0001$.
7) For the local RNN, the number of hidden layer neurons and recurrent layer neurons is set as 12 and 6, respectively. The initial value of weights $\hat{W}$ and $\hat{V}$ are both selected randomly in the range from -1 to 1.
8) $\hat{\gamma}$ is the estimation value of γ, whose initial value diag([1, 1, 1, 1]) is suitable. The initial values of k_1 and k_4 are set as 0, and they will be adjusted according to the update laws (13) and (14), respectively.

B. Simulation

1) Comparison I: In order to illustrate the effectiveness of the proposed adaptive terminal sliding-mode state observer, a high-gain observer-based proportion-integral-derivative (PID) controller is used for comparison. For the latter, the observer is used to estimate the state first, and then, the PID controller is designed based on the estimated values. The high-gain observer and the PID controller are

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 - k_{11}\tilde{x}_1 \\ \dot{\hat{x}}_2 &= B(\hat{x}_1)\hat{\gamma}u - k_{12}\tilde{x}_1 \end{aligned} \tag{49}$$

$$\begin{aligned} u = {} & B^T(\hat{x}_1)[\rho_0 I + B(\hat{x}_1)B^T(\hat{x}_1)]^{-1} \\ & \times \left(-k_p e_1 - k_d\dot{e}_1 - k_i\int_0^t e_1 dt\right) \end{aligned} \tag{50}$$

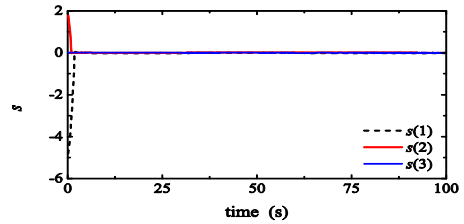

Fig. 4. Value of sliding mode in the proposed method.

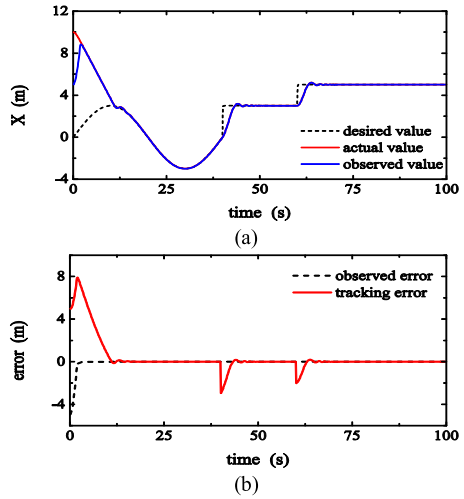

Fig. 5. Simulation result of position X in the proposed method. (a) Actual trajectory and observer estimated values of position X. (b) Tracking error and observer estimation error of position X.

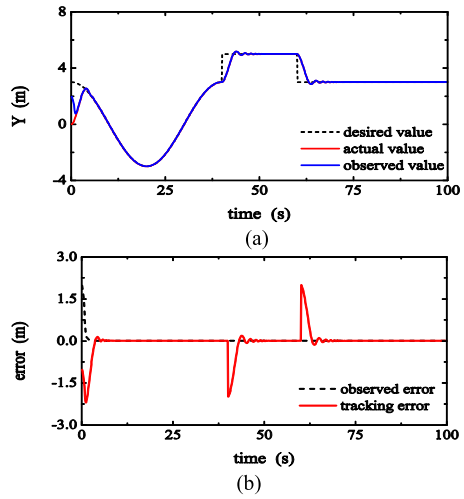

Fig. 6. Simulation result of position Y in the proposed method. (a) Actual trajectory and observer estimated values of position Y. (b) Tracking error and observer estimation error of position Y.

where $k_{11} = \text{diag}([2, 2, 40])$, $k_{12} = \text{diag}([2, 2, 40])$, $k_p = \text{diag}([5, 5, 40])$, $k_d = \text{diag}([0.5, 0.5, 0.5])$, and $k_i = \text{diag}([0.2, 0.2, 0.8])$.

The simulation results of the proposed method are shown in Figs. 4–7, and the results of the high-gain observer-based PID controller are shown in Figs. 8–10.

According to Figs. 4–7, the convergence times of the sliding mode and the estimation errors are shown in Table II.

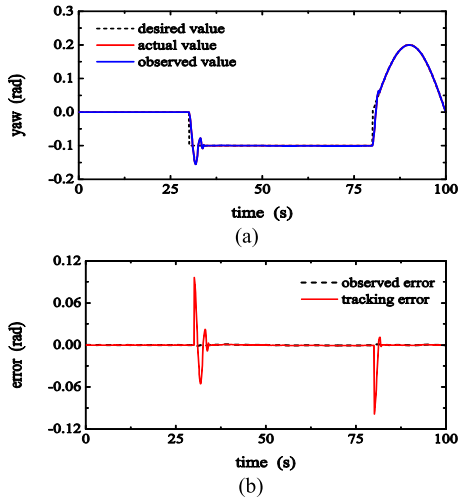

Fig. 7. Simulation result of yaw angle in the proposed method. (a) Actual trajectory and observer estimated values of yaw angle. (b) Tracking error and observer estimation error of yaw angle.

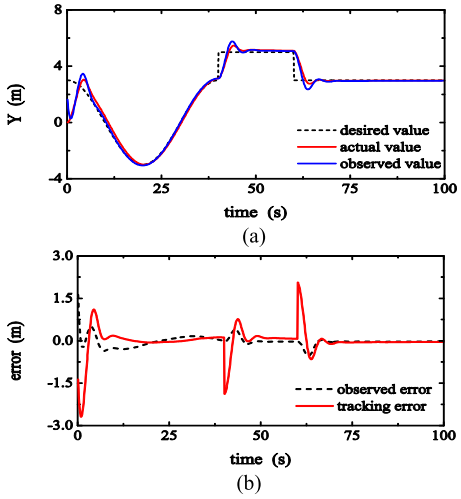

Fig. 9. Simulation result of position Y in high gain observer-based PID controller. (a) Actual trajectory and observer estimated values of position Y. (b) Tracking error and observer estimation error of position Y.

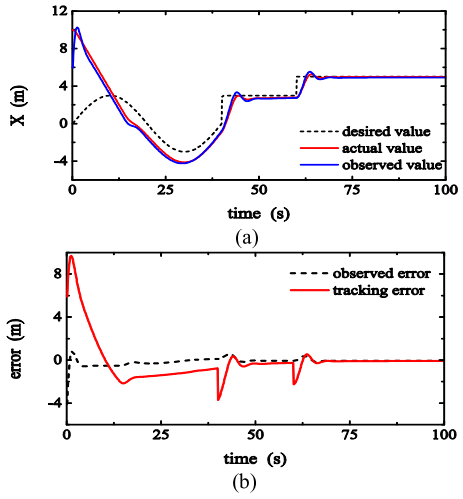

Fig. 8. Simulation result of position X in high gain observer-based PID controller. (a) Actual trajectory and observer estimated values of position X. (b) Tracking error and observer estimation error of position X.

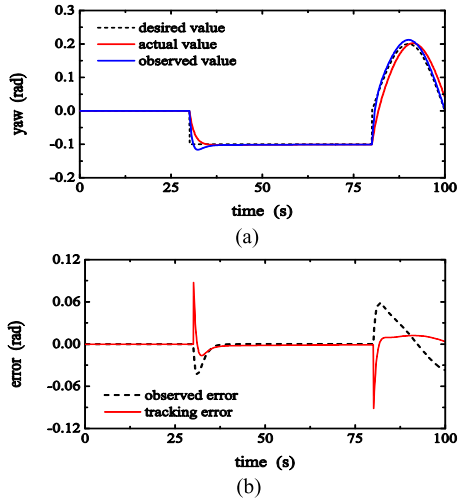

Fig. 10. Simulation result of yaw angle in the high gain observer-based PID controller. (a) Actual trajectory and observer estimated values of yaw angle. (b) Tracking error and observer estimation error of yaw angle.

In Table II, $s(1)$, $s(2)$, and $s(3)$ can all converge in 2.1 s, even though the desired trajectory has a mutation. And then, $\tilde{x}_{11}$, $\tilde{x}_{12}$, and $\tilde{x}_{13}$ also converge to zero in 4.7 s. The maximum convergence times of tracking errors of the proposed method are shown in Table III. After 18.2 s, the tracking errors all converge to zero. In Table IV, the average values and mean variances of tracking errors from 18.2 to 30 s are analyzed. These indicators are all very small and close to zero. It shows that the proposed adaptive terminal sliding-mode state observer is feasible.

According to Figs. 8–10, the average values and mean variances of the estimation errors from 18.2 to 30 s are shown in Table V. It shows that the estimation errors of the PID controller cannot converge to zero quickly. The average values and mean variances of the tracking errors from 18.2 to 30 s are shown in Table VI. It shows that the performance of a high-gain observer-based PID controller is worse than the proposed controller. Although the estimation error and tracking error of yaw angle are equal to zero, this is because both the initial value of the estimation error and the tracking error are near to zero. By comparing the simulation curves from 80 to 100 s, the high-gain observer-based PID controller is worse than the proposed controller. Although there may be some optimal control parameters that can make the performance of the high-gain observer-based PID controller better, the estimation errors are still unable to converge to zero in a finite time. The results show that the proposed controller is more effective for ROV trajectory tracking than the high-gain observer-based PID controller.

2) Comparison II: In order to illustrate the effectiveness of the NN-based adaptive controller for ROV trajectory tracking, the controller based on fixed-structure RBF NN is used for comparison simulation. For the latter, according to the modeling method in [14], the fixed-structure RBF NN is used for

TABLE II
CONVERGENCE TIMES OF SLIDING MODE AND ESTIMATION ERRORS

	$s(1)$	$s(2)$	$s(3)$	$\tilde{x}_{11}$	$\tilde{x}_{12}$	$\tilde{x}_{13}$
convergence time	2.1s	1.1s	0.2s	4.7s	5.3s	2.8s

TABLE III
CONVERGENCE TIMES OF TRACKING ERRORS

	e_{11}	e_{12}	e_{13}
convergence time	18.2s	11.3s	5.6s

TABLE IV
AVERAGE VALUES AND MEAN VARIANCES OF TRACKING ERRORS FROM 18.2 TO 30 s

	e_{11}	e_{12}	e_{13}
average value	-0.0080m	0.0085m	0rad
mean variance	0.0017m	7.9e-4m	0rad

TABLE V
AVERAGE VALUES AND MEAN VARIANCES OF ESTIMATION ERRORS FROM 18.2 TO 30 s BY PID CONTROLLER

	$\tilde{x}_{11}$	$\tilde{x}_{12}$	$\tilde{x}_{13}$
average value	-0.253m	0.025m	0 rad
mean variance	0.065m	0.085m	0 rad

TABLE VI
AVERAGE VALUES AND MEAN VARIANCES OF TRACKING ERRORS FROM 18.2 TO 30 s BY PID CONTROLLER

	e_{11}	e_{12}	e_{13}
average value	-1.430m	-0.041m	0 rad
mean variance	0.108m	0.022m	0 rad

the offline identification of unknown nonlinear function $f(x)$. In the selection of teaching sample, the ocean current v_c is considered as 0, and scale factor γ is considered as 1. After training, the control law is designed as

$$u = B^{-1}(\hat{x}_1)[-e_1 - F(\hat{x}) + k_2\alpha^{q_2/p_2} + k_4\text{sgn}(\alpha) + (\beta e_2)^{q_3/p_3} - k_7\text{sgn}(e_2) - k_3 e_2] \quad (51)$$

where $F(\hat{x})$ is the output of RBF NN. $\hat{x}$ is the state estimation, which is obtained by

$$\dot{\hat{x}}_1 = \hat{x}_2 - \alpha$$
$$\dot{\hat{x}}_2 = F(\hat{x}) + B(\hat{x}_1)u - k_2\alpha^{q_2/p_2} - k_4\text{sgn}(\alpha). \quad (52)$$

In (51) and (52), the controller parameters are the same as the proposed controller. The simulation of the controller based on the fixed structure of RBF NN is shown in Figs. 11–14.

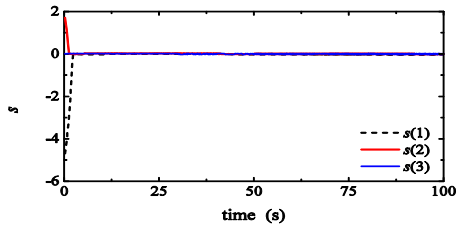

Fig. 11. Value of sliding mode in the controller based on fixed structure of RBF NN.

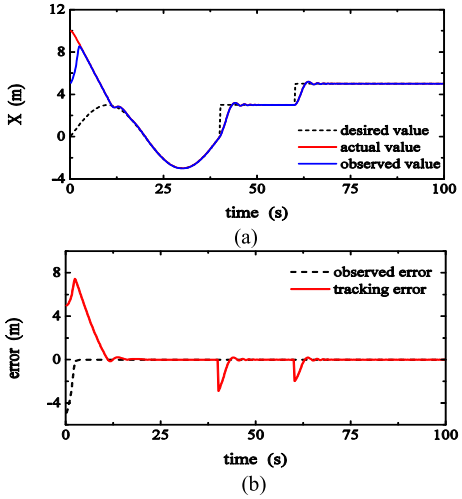

Fig. 12. Simulation result of position X in the fixed structure RBF NN controller. (a) Actual trajectory and observer estimated values of position X. (b) Tracking error and observer estimation error of position X.

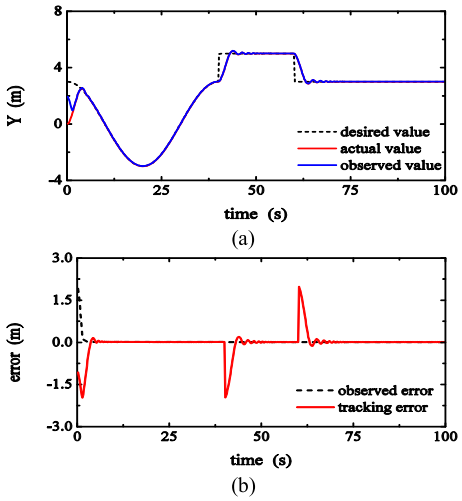

Fig. 13. Simulation result of position Y in the fixed structure RBF NN controller. (a) Actual trajectory and observer estimated values of position Y. (b) Tracking error and observer estimation error of position Y.

According to the simulation curves, the convergence times of the sliding mode are shown in Table VII, the convergence times of the tracking errors are shown in Table VIII, and the average values and mean variances of tracking errors from 18.2 to 30 s are shown in Table IX. From the simulation results, it can be found that both the controllers are

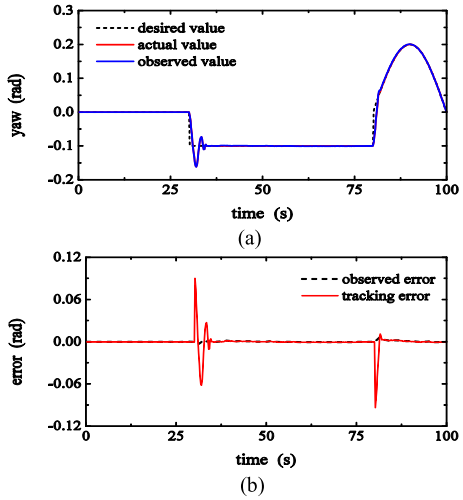

Fig. 14. Simulation result of yaw angle in fixed structure RBF NN controller. (a) Actual trajectory and observer estimated values of yaw angle. (b) Tracking error and observer estimation error of yaw angle.

TABLE VII
CONVERGENCE TIMES OF SLIDING MODE AND ESTIMATION ERRORS BY FIXED STRUCTURE RBF NN

	$s(1)$	$s(2)$	$s(3)$	$\tilde{x}_{11}$	$\tilde{x}_{12}$	$\tilde{x}_{13}$
convergence time	2.3s	1.3s	1.2s	4.4s	3.2s	3.6s

TABLE VIII
CONVERGENCE TIMES OF TRACKING ERRORS BY FIXED-STRUCTURE RBF NN

	e_{11}	e_{12}	e_{13}
convergence time	21s	6.8s	5.4s

useful for trajectory tracking control. In the fixed structure NN-based controller, the modeling error of offline identification cannot be avoided, but it can be compensated by the sliding-mode item. Therefore, the performance of two controllers is similar. However, comparing the realization processes of two controllers, it is obvious that the controller based on the fixed-structure RBF NN is more complex, because the nonlinear function $f(x)$ needs to be learned offline first. ROVs usually need to be equipped with the operation equipment for different tasks. Therefore, the dynamic model will be changed, and then, the model needs to be learned again, but the learning process would take a lot of time. Simulation results show that the proposed adaptive control approach is effective and more suitable for the actual ROV system.

3) Comparison III: In order to illustrate the effectiveness of the local RNN in the adaptive controller, the three-layer NN-based adaptive controller is used for comparison simulation, and the sensor noise is also considered in simulation. For the latter, the control law is also chosen as (12) while using the three-layer NN instead of the local RNN. The structure of the three-layer NN is chosen according to [34],

TABLE IX
AVERAGE VALUES AND MEAN VARIANCES OF TRACKING ERRORS FROM 18.2 TO 30 s BY FIXED-STRUCTURE RBF NN

	e_{11}	e_{12}	e_{13}
average value	0.0131m	0m	0rad
mean variance	0.0209m	0.0016m	0rad

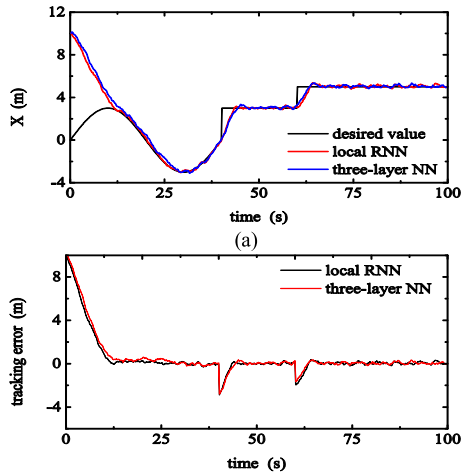

Fig. 15. Tracking results of position X. (a) Actual trajectory of position X. (b) Trajectory tracking error of position X.

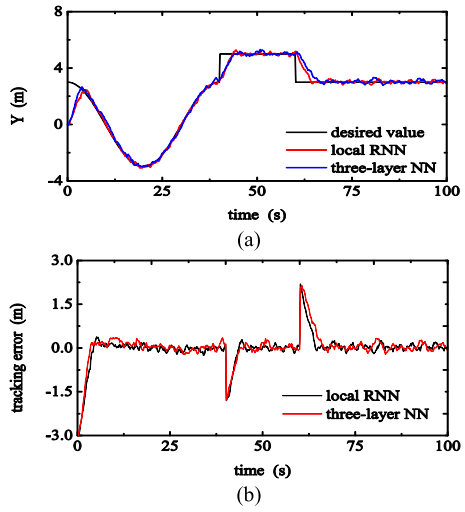

Fig. 16. Tracking results of position Y. (a) Actual trajectory of position Y. (b) Trajectory tracking error of position Y.

which does not have the recurrent layer. The weights of the three-layer NN are adjusted as update laws (15) and (16). The sensor noise of state η is selected as diag([0.01, 0.01, 0.001]) $(2 \times \text{rand}(3, 1) - 1)$. The simulation results of tracking performance are shown in Figs. 15–17.

According to the results of Figs. 15(b), 16(b), and 17(b), the maximum convergence time of trajectory tracking errors of two different controllers is shown in Table X. It shows that the adaptive controllers based on the local RNN and the

TABLE X
CONVERGENCE TIME OF TRAJECTORY TRACKING ERROR

	δ_{11}		δ_{12}		δ_{13}	
	local RNN	three-layer NN	local RNN	three-layer NN	local RNN	three-layer NN
convergence time	5.0s	7.4s	4.4s	6.8s	6.4s	11.4s

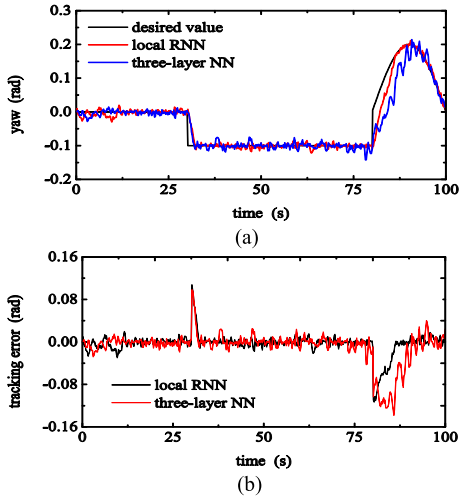

Fig. 17. Tracking results of yaw angle. (a) Actual trajectory of position yaw angle. (b) Trajectory tracking error of yaw angle.

TABLE XI
AVERAGE VALUES AND MEAN VARIANCES OF TRAJECTORY TRACKING ERROR

	δ_{11}		δ_{12}		δ_{13}	
	local RNN	three-layer NN	local RNN	three-layer NN	local RNN	three-layer NN
average value	0.471m	0.567m	-0.031m	0.021m	-0.004rad	-0.009rad
mean variance	1.803m	1.899m	0.505m	0.523m	0.020rad	0.030rad

three-layer NN are both feasible for trajectory tracking control. The average values and mean variances of trajectory tracking errors are shown in Table XI. It shows that the proposed adaptive controller has a higher accuracy, especially when the desired trajectory has a sudden change. Simulation results show that the local RNN-based adaptive NN controller is more effective for ROV trajectory tracking.

V. CONCLUSION

In this paper, the adaptive NN trajectory tracking control method based on the terminal sliding-mode observer is researched. The state of the ROV system can be estimated by constructing an adaptive terminal sliding-mode state observer, and a local RNN is used for the online learning of the unknown function. Based on the output of local RNN and the estimated values of observer, the adaptive trajectory tracking control law is designed to obtain the control signal of thrusters directly. Simulation results show that the proposed observer-based adaptive local NN trajectory tracking controller can complete the ROV trajectory tracking control. It has a better control performance than the high-gain observer-based PID controller, fixed-structure RBF NN-based controller, and three-layer NN-based adaptive controller. The trajectory tracking error can converge in a finite time in the proposed method. Although the proposed method has been proven to be feasible and effective, the problem of finite-time convergence for the adaptive local RNN estimation error should be paid more attention in the future work. If it can be guaranteed, the selecting method for parameter λ_W and λ_V can be obtained, which are very important to the learning rate and the convergence of local RNN.

REFERENCES

[1] S. A. Woods, R. J. Bauer, and M. L. Seto, "Automated ballast tank control system for autonomous underwater vehicles," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 727–739, Oct. 2012.
[2] H. Shim, B.-H. Jun, P.-M. Lee, H. Baek, and J. Lee, "Workspace control system of underwater tele-operated manipulators on an ROV," *Ocean Eng.*, vol. 37, nos. 11–12, pp. 1036–1047, 2010.
[3] E. C. De Souza and N. Maruyama, "Intelligent UUVs: Some issues on ROV dynamic positioning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 214–226, Jan. 2007.
[4] L. Cheng, Y. Lin, Z.-G. Hou, M. Tan, J. Huang, and W. J. Zhang, "Adaptive tracking control of hybrid machines: A closed-chain five-bar mechanism case," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 6, pp. 1155–1163, Dec. 2010.
[5] G. Chen and F. L. Lewis, "Distributed adaptive tracking control for synchronization of unknown networked Lagrangian systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 805–816, Jun. 2011.
[6] D. Chwa, "Fuzzy adaptive tracking control of wheeled mobile robots with state-dependent kinematic and dynamic disturbances," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 3, pp. 587–593, Jun. 2012.
[7] D. Wang, D. Liu, Q. Zhang, and D. Zhao, "Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, doi: 10.1109/TSMC.2015.2492941.
[8] C. Mu, Z. Ni, C. Sun, and H. He, "Air-breathing hypersonic vehicle tracking control based on adaptive dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, doi: 10.1109/TNNLS.2016.2516948.
[9] B. Miao, T. Li, and W. Luo, "A DSC and MLP based robust adaptive NN tracking control for underwater vehicle," *Neurocomputing*, vol. 111, pp. 184–189, Jul. 2013.
[10] B. Xu, S. R. Pandian, N. Sakagami, and F. Petry, "Neuro-fuzzy control of underwater vehicle-manipulator systems," *J. Franklin Inst.*, vol. 349, no. 3, pp. 1125–1138, 2012.
[11] L. Lapierre and B. Jouvencel, "Robust nonlinear path-following control of an AUV," *IEEE J. Ocean. Eng.*, vol. 33, no. 2, pp. 89–102, Apr. 2008.
[12] A. Bagheri, T. Karimi, and N. Amanifard, "Tracking performance control of a cable communicated underwater vehicle using adaptive neural network controllers," *Appl. Soft Comput.*, vol. 10, no. 3, pp. 908–918, 2010.
[13] N. Q. Hoang and E. Kreuzer, "Adaptive PD-controller for positioning of a remotely operated vehicle close to an underwater structure: Theory and experiments," *Control Eng. Pract.*, vol. 15, no. 4, pp. 411–419, 2007.
[14] Z.-Z. Chu and M.-J. Zhang, "Fault reconstruction of thruster for autonomous underwater vehicle based on terminal sliding mode observer," *Ocean Eng.*, vol. 88, no. 1, pp. 426–434, Sep. 2014.
[15] Y. Gan, L.-R. Wang, J.-C. Liu, and Y.-R. Xu, "The embedded basic motion control system of autonomous underwater vehicle," *Robot*, vol. 26, no. 3, pp. 246–253, 2004.
[16] Z. Li, C. Zheng, and D. Sun, "Track analysis and design for ultra short baseline installation error calibration," in *Proc. Oceans*, San Diego, CA, USA, 2013, pp. 1–5.
[17] J. S. Gao, Z. W. Xing, and H. B. Zhang, "Observer-based neural network adaptive control of underwater vehicles," *Robot*, vol. 26, no. 6, pp. 515–518, 2004.
[18] L.-J. Zhang, X. Qi, and Y.-J. Pang, "Adaptive output feedback control based on DRFNN for AUV," *Ocean Eng.*, vol. 36, nos. 9–10, pp. 716–722, 2009.

[19] J.-C. Yu, Q. Li, A.-Q. Zhang, and X.-H. Wang, "Neural network adaptive control for underwater vehicles," *Control Theory Appl.*, vol. 25, no. 1, pp. 9–13, 2008.
[20] N. Fischer, D. Hughes, P. Walters, E. M. Schwartz, and W. E. Dixon, "Nonlinear RISE-based control of an autonomous underwater vehicle," *IEEE Trans. Robot.*, vol. 30, no. 4, pp. 845–852, Aug. 2014.
[21] P. Millán, L. Orihuela, I. Jurado, and F. R. Rubio, "Formation control of autonomous underwater vehicles subject to communication delays," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 2, pp. 770–777, Mar. 2014.
[22] J. Biggs and W. Holderbaum, "Optimal kinematic control of an autonomous underwater vehicle," *IEEE Trans. Autom. Control*, vol. 54, no. 7, pp. 1623–1626, Jul. 2009.
[23] J. Kim and W. K. Chung, "Accurate and practical thruster modeling for underwater vehicles," *Ocean Eng.*, vol. 33, nos. 5–6, pp. 566–586, 2006.
[24] G. Antonelli, "Modelling of underwater robots," in *Underwater Robots: Motion and Force Control of Vehicle-Manipulator Systems*, 2nd ed. Berlin, Germany: Springer Berlin Heidelberg, 2006.
[25] L. Ma, K. Schilling, and C. Schmid, "Adaptive backstepping sliding mode control with Gaussian networks for a class of nonlinear systems with mismatched uncertainties," in *Proc. 44th IEEE Conf. Decision Control, Eur. Control Conf.*, Seville, Spain, Dec. 2005, pp. 5504–5509.
[26] M. Tan, X.-X. Li, and C. Lu, "Self-adaptive neural PID and its application to temperature control system of plastic machine," *Control Eng. China*, vol. 13, no. 3, pp. 250–251, 2006.
[27] H. A. Talebi, K. Khorasani, and S. Tafazoli, "A recurrent neural-network-based sensor and actuator fault detection and isolation for nonlinear systems with application to the satellite's attitude control subsystem," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 45–60, Jan. 2009.
[28] M.-J. Zhang and Z.-Z. Chu, "Adaptive sliding mode control based on local recurrent neural networks for underwater robot," *Ocean Eng.*, vol. 45, no. 1, pp. 56–62, May 2012.
[29] S. Li, M. Zhou, and X. Yu, "Design and implementation of terminal sliding mode control method for PMSM speed regulation system," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 1879–1891, Nov. 2013.
[30] V. Nekoukar and A. Erfanian, "Adaptive fuzzy terminal sliding mode control for a class of MIMO uncertain nonlinear systems," *Fuzzy Sets Syst.*, vol. 179, no. 1, pp. 34–49, 2011.
[31] P. Li and Z.-Q. Zheng, "Robust adaptive second-order sliding-mode control with fast transient performance," *IET Control Theory Appl.*, vol. 6, no. 2, pp. 305–312, Jan. 2012.
[32] S. C. Tong, Y. M. Li, G. Feng, and T. S. Li, "Observer-based adaptive fuzzy backstepping dynamic surface control for a class of MIMO nonlinear systems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 4, pp. 1124–1135, Aug. 2011.
[33] T. K. Podder and N. Sarkar, "Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy," *Robot. Auto. Syst.*, vol. 34, no. 1, pp. 39–52, 2001.
[34] L. Cheng, Z.-G. Hou, and M. Tan, "Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model," *Automatica*, vol. 45, no. 10, pp. 2312–2318, 2009.

Zhenzhong Chu was born in Suzhou, China. He received the B.Sc. degree in mechanical design, manufacturing and automation and the Ph.D. degree in mechanical electronic engineering from Harbin Engineering University, Harbin, China, in 2007 and 2013, respectively.

He is currently a Lecturer with the Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Shanghai, China. His current research interests include adaptive control and fault diagnosis of AUV.

Daqi Zhu was born in Anhui, China. He received the B.Sc. degree in physics from the Huazhong University of Science and Technology, Wuhan, China, in 1992, and the Ph.D. degree in electrical engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2002.

He is currently a Professor with Information Engineering College and the Head of the Laboratory of Underwater Vehicles and Intelligent Systems with Shanghai Maritime University, Shanghai, China. His current research interests include neural networks, fault diagnosis, and control of AUV.

Simon X. Yang (S'97–M'99–SM'08) received the B.Sc. degree in engineering physics from Peking University, Beijing, China, in 1987, the M.Sc. degree in biophysics from the Chinese Academy of Sciences, Beijing, in 1990, the M.Sc. degree in electrical engineering from the University of Houston, Houston, TX, USA, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Alberta, Edmonton, AB, Canada, in 1999.

He is currently a Professor and the Head of the Advanced Robotics and Intelligent Systems Laboratory with the University of Guelph, Guelph, ON, Canada. His current research interests include intelligent systems, robotics, sensors and multisensory fusion, wireless sensor networks, control systems, and computational neuroscience.