THE JOURNAL OF NAVIGATION (2017), **70**, 149–164.
doi:10.1017/S0373463316000448

Adaptive Sliding Mode Control for Depth Trajectory Tracking of Remotely Operated Vehicle with Thruster Nonlinearity

Zhenzhong Chu[1], Daqi Zhu[1], Simon X. Yang[2] and Gene Eu Jan[3]

[1](*Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Shanghai, 201306, China*)
[2](*The Advanced Robotics and Intelligent Systems Laboratory, School of Engineering, University of Guelph, Guelph, ON. N1G2W1, Canada*)
[3](*Department of Computer Science and Information Engineering, National Taipei University, Taipei County, 237, Taiwan, ROC*)
(E-mail: zdq367@aliyun.com)

This paper focuses on depth trajectory tracking control for a Remotely Operated Vehicle (ROV) with dead-zone nonlinearity and saturation nonlinearity of thruster; an adaptive sliding mode control method based on neural network is proposed. Through the analysis of dead-zone nonlinearity and saturation nonlinearity of thruster, the depth trajectory tracking control system model of a ROV which uses thruster control signals as system input has been established. According to the principle of sliding mode control, an adaptive sliding mode depth trajectory tracking controller is built by using three-layer feed-forward neural network for online identification of unknown items. The selection method and update laws of the control parameters are also given. The uniform ultimate boundedness of trajectory tracking error is analysed by Lyapunov theorem. Finally, the effectiveness of the proposed method is illustrated by simulations.

KEYWORDS

1. Remotely operated vehicle. 2. Trajectory tracking. 3. Adaptive control. 4. Sliding mode control. 5. Thruster nonlinearity.

Submitted: 14 July 2015. Accepted: 4 June 2016. First published online: 28 July 2016.

1. INTRODUCTION. A Remotely Operated Vehicle (ROV) is an important tool for marine resource exploitation and marine scientific research with the advantages of good economy, high flexibility and strong adaptability to the environment, among others (Chu et al., 2016a; Avila et al., 2013). The ROV control of horizontal position, orientation and depth is realised by operators relying on the real-time images transmitted by camera when an ROV works underwater (Wang et al., 2006). The precise synchronisation operation for various degrees of freedom is difficult to achieve, so

the ROV controller is essentially an open-loop control system. The operations of ROV are complex and difficult (Gao et al., 2015c), but if automatic depth control can be achieved during the operation, then manual operation is only for the motion in the horizontal plane and underwater operations will become relatively easy (Bessa et al., 2008; Gao et al., 2015a). Therefore the study of depth tracking control for ROV has a certain practical significance.

In the study of underwater vehicle location tracking control, many methods have been proposed (Pan and Xin, 2012; Feng and Allen, 2004; Sun et al., 2014; Gao et al., 2015b). Due to the complex uncertainties of underwater vehicle systems such as dynamics modelling error and external disturbances (Avila et al., 2012), the conventional controllers such as a Proportional-Integral-Derivative (PID) controller are unable to cope with these uncertainties and will cause poor performance. Hence the adaptive control methods have been researched extensively, such as adaptive sliding mode control (Zhang and Chu, 2012; Zhang et al., 2015), adaptive neural network control (Zhu et al., 2014), adaptive Proportional and Derivative (PD) control (Hoang and Kreuzer, 2007), etc. In most of these methods, the output of the controller is usually the force or moment acting on each degree of freedom. Since the actual thrust is unmeasurable, the desired thrust of each thruster needs to be determined by thruster distribution matrix first, and then the control signals which should be loaded on the thrusters can be calculated by the desired thrust according to the thrust model (Gan et al., 2004). But a complex mapping relationship between the thrust and the control signal is required in this method (Kim and Chung, 2006). It is difficult to obtain accurate control signals according to desired thrust, which may affect the quality of the tracking control system. Moreover, most of these control methods did not take the dead-zone nonlinearity and saturation nonlinearity of the thruster into consideration in the design process. For most thrusters, these two problems are encountered. The stability analysis of the control system shows that the stability of the closed-loop system may not be guaranteed by adaptive learning law when the thruster output is in a saturation zone or dead-zone (Chu et al., 2016b; Wu et al., 2012; Liu and Zhou, 2010; Tong et al., 2013). At present, the solutions to dead-zone nonlinearity of ROVs' tracking control are still rare. For the problem of saturation nonlinearity, the method based on the adjustment of command input signal is usually used (Gan et al., 2004). However, this method relies on an accurate thrust model and it is also very complex. Therefore, for the nonlinear problems of thrusters, this research focuses on how to design an adaptive controller to obtain thruster control signals directly and ensure the stability of the closed-loop control system.

Motivated by the aforementioned observations, an adaptive sliding mode control method based on three-layer feed-forward neural network is proposed for ROV depth trajectory tracking control. The Lyapunov stability theory is used for stability analysis to prove that the proposed controller can guarantee trajectory tracking error is uniformly ultimately bounded. The main contributions of this paper can be summarised as three points. First, the problem of dead-zone nonlinearity and saturation nonlinearity of thruster is considered in controller design. Second, the control signal loaded on the thruster can be obtained directly by the designed controller without a thrust model. The third contribution is that a three-layer feed-forward neural network is introduced for online learning for the unknown motion model of an ROV depth tracking system.

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

The paper is organised as follows. In Section 2, the dead-zone nonlinearity and saturation nonlinearity of thruster is analysed, and a depth trajectory tracking system model which uses thruster control signals as system input is established. In Section 3, an adaptive sliding mode controller for depth trajectory tracking based on neural network is given. In Section 4, Lyapunov stability theory is used to analyse the stability of the closed-loop system. Then in Section 5, the effectiveness of the proposed approach is verified by simulation experiments. Finally, a brief conclusion is included in Section 6 of this paper.

2. PROBLEM FORMULATION.

In underwater environments, an ROV is affected by hydrodynamic forces, cable drag forces and other factors, and its motion has strong nonlinearity and cross-coupling (Cao and Zhu, 2015). However, the distances between the centres of buoyancy and the centres of gravity of most ROVs are relatively far, which make the change of pitch and roll small, so that the influence of other degrees of freedom for vertical motion can be ignored (Bessa et al., 2010). Thus, the ROV vertical motion model can be described as:

$$M\ddot{z} + D(\dot{z})\dot{z} + g(z) + \tau_d = \tau(u) \tag{1}$$

where, z, $\dot{z}$, $\ddot{z}$ denote vertical depth, vertical velocity and vertical acceleration, respectively, M contains ROV mass and added mass, $D(\dot{z})$ is hydrodynamic drag, $g(z)$ is the difference between the gravity and buoyancy of the ROV, τ_d is external disturbance force, such as cable drag and current interference force, etc., $\tau(u)$ is the force acting on the vertical degree of freedom, u is control signal loaded on the thruster and it is usually expressed as a voltage value.

Remark 1: The assumption is that only one thruster is arranged in the vertical degree of freedom, and $\tau(u)$ is the thrust. But in the case that multiple identical thrusters are arranged uniformly in the vertical degree of freedom, the control signal of each thruster can be made the same and then $\tau(u) = n\tau_i(u_i)$. Where n is the number of thrusters, $\tau_i(u_i)$ is the thrust of each thruster. Therefore, $n\tau_i(u_i)$ can be used instead of $\tau(u)$ in Equation (1) in the controller design.

The desired trajectory of ROV depth motion is set as $[z_d, \dot{z}_d]^{\mathrm{T}}$. Then the trajectory tracking error vector $\xi = [\xi_1, \xi_2]^T$ can be defined as:

$$\begin{aligned} \xi_1 &= z - z_d \\ \xi_2 &= \dot{z} - \dot{z}_d \end{aligned} \tag{2}$$

where z_d, $\dot{z}_d$ are the desired vertical depth and the desired vertical velocity, respectively.

Remark 2: In order to realise the desired vertical depth trajectory, the desired vertical velocity $\dot{z}_d$ should be in the range of $(\dot{z}_{\min}, \dot{z}_{\max})$, where $\dot{z}_{\min}$, $\dot{z}_{\max}$ are the maximum diving velocity and the maximum rising velocity of the ROV, respectively.

Substituting Equation (1) into Equation (2), leads to:

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= f(z, \dot{z}) + M^{-1}\tau(u) \end{aligned} \tag{3}$$

where $f(z, \dot{z}) = -M^{-1}[D(\dot{z})\dot{z} + g(z) + \tau_d] - \ddot{z}_d$ are the un-modelled dynamics.

For most thrusters, the thrust model can be described approximately as $\tau(u) = a_1 u|u| - a_2 |u|v_a$, where a_1, a_2 are unknown positive parameters and v_a is the advance

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

speed of the propeller (Alessandri et al., 1999). However, the actual tested thrust curve of a Model 520 thruster is shown in Figure 1(a), which shows that the dead-zone nonlinearity problem exists. According to the test results, the theoretical thrust curve of the thruster can be inferred as shown in Figure 1(b). In Figure 1(b), when the thruster control signal is within the range of $[u_l, u_r]$, there is no thrust output. The function shown in Equation (4) can be used to describe the theoretical thrust curve of thruster in Figure 1(b).

$$\tau(u) = \begin{cases} \tau_+(u)(u - u_r) & u \geqslant u_r \\ 0 & u_l < u < u_r \\ \tau_-(u)(u - u_l) & u \leqslant u_l \end{cases} \tag{4}$$

From Figure 1(b) and Equation (4), the dead-zone of the thruster has the following characteristics:

Characteristic 1: $\tau(u)$ is unmeasurable.

Characteristic 2: The dead-zone parameters both u_r and u_l are unknown bounded constants.

Characteristic 3: $\tau_+(u)$ and $\tau_-(u)$ are smooth, and there exist positive constants k_r, k_l, $\dot{k}_r$ and $\dot{k}_l$, which make:

$$0 < k_r \leqslant \tau_+(u),\ 0 < k_l \leqslant \tau_-(u),\ \dot{k}_r \geqslant \dot{\tau}_+(u) > 0,\ \dot{k}_l \leqslant \dot{\tau}_-(u) < 0 \tag{5}$$

Characteristic 4: There is known positive constant β, so that $\beta \leqslant \min(k_r, k_l)$.

Characteristic 5: When $u < u_r$, then $\tau_+(u) = \tau_+(u_r)$ and when $u > u_l$, $\tau_-(u) = \tau_-(u_l)$. There exists an unknown positive constant ρ which makes $|d(t)| \leqslant \rho$.

According to the above characteristics, $\tau(u)$ can be re-expressed as:

$$\tau(u) = \boldsymbol{T}(t)\boldsymbol{\varphi}(t)u + d(t) \tag{6}$$

where $\boldsymbol{T}(t) = [\tau_+(u), \tau_-(u)]$, $\boldsymbol{\varphi}(t) = [\varphi_r, \varphi_l]^T$, and:

$$\varphi_r = \begin{cases} 1, & u > u_l \\ 0, & u \leqslant u_l \end{cases},\ \varphi_l = \begin{cases} 1, & u < u_r \\ 0, & u \geqslant u_r \end{cases},$$
$$d(t) = \begin{cases} -\tau_+(u)u_r & u \geqslant u_r \\ -[\tau_+(u) + \tau_-(u)]u & u_l < u < u_r \\ -\tau_-(u)u_l & u \leqslant u_l \end{cases} \tag{7}$$

Substituting Equation (6) into Equation (3), the depth trajectory tracking control system equation of the ROV can be expressed as:

$$\begin{aligned} \dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= f(z, \dot{z}) + M^{-1}\boldsymbol{T}(t)\boldsymbol{\varphi}(t)u + M^{-1}d(t) \end{aligned} \tag{8}$$

As can be seen from Equation (8), the control system is described in the form of affine nonlinear system which uses thruster control signal u as system input. This benefits the subsequent controller design to obtain thruster control signal u directly

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

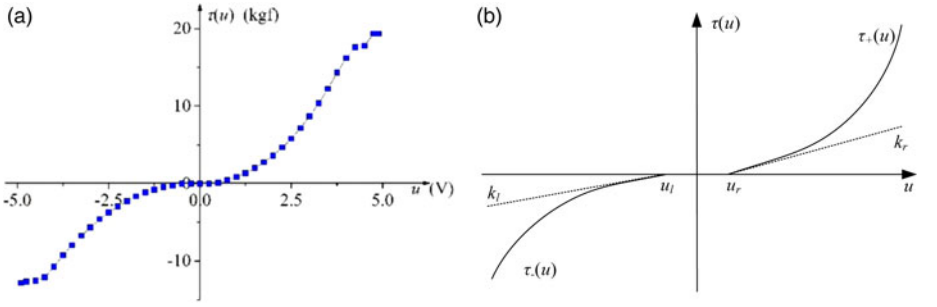

Figure 1. The thrust curve of thruster. (a) The actual thrust curve of Model 520 thruster. (b) The theoretical thrust curve of thruster.

without a thrust model, which is quite different from most existing ROV control methods.

Because of the finite thrust and the large ROV inertia in a practical thruster, the saturation nonlinearity of the thruster easily appears when the trajectory tracking error is large. So it must be considered during the process of controller design. When the output signal of the controller is in the saturation area, the control law as shown in Equation (9) is usually adopted (Chen et al., 2011).

$$u = \begin{cases} v_{\max} & v > v_{\max} \\ v & v_{\min} \leqslant v \leqslant v_{\max} \\ v_{\min} & v < v_{\min} \end{cases} \tag{9}$$

where v is the output of control law, $v_{\max}$ and $v_{\min}$ are the upper limit and lower limit of thruster control signal, respectively.

Remark 3: For an actual ROV system, the following must be satisfied: (1) if $u = v_{\max}$, then $\ddot{z} > 0$, and $\dot{z}$ will increase until $\dot{z}=\dot{z}_{\max}$; (2) if $u = v_{\min}$, then $\ddot{z} < 0$, and $\dot{z}$ will decrease until $\dot{z}=\dot{z}_{\min}$.

In summary, the main objective of this paper is to design the depth trajectory tracking control law v with thruster nonlinearity and un-modelled dynamics, which will finally make the depth trajectory tracking error uniformly ultimately bounded.

3. CONTROLLER DESIGN.

Since there are un-modelled dynamics, an adaptive sliding mode controller based on neural network will be given for ROV depth trajectory tracking in this Section. Without considering the thruster nonlinearity, many adaptive sliding mode control methods have been proposed (Zhang and Chu, 2012; Zhu et al., 2014; Hoang and Kreuzer, 2007). However, thruster nonlinearity is inevitable in an actual ROV system. That is, in these methods, when the control law reaches the saturated zone, the adaptive parameters are still adjusted although the actual control law does not change. If the control law stays in the saturated zone for a long time and then returns to a feasible region, control system instability might occur due to the adjusted parameters. In this paper, the proposed adaptive sliding mode controller will only adjust the adaptive parameters when the control law is in the feasible region, and the Lyapunov stability theory is used to prove that the depth trajectory tracking error is uniformly ultimately bounded. In this section, the sliding mode

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

dynamics will be analysed first. Then, the three-layer feed-forward neural network will be given and the estimation error of the neural network will be analysed. Finally, the theorem will be given.

According to the principle of sliding mode control, a sliding surface s is defined firstly as (Hussain and Po, 2004):

$$s = \xi_2 + l\xi_1 \tag{10}$$

where l is a positive constant.

Differentiating s and introducing Equation (8), we obtain:

$$\begin{aligned} \dot{s} &= f(z, \dot{z}) + M^{-1}\boldsymbol{T}(t)\boldsymbol{\varphi}(t)u + M^{-1}d(t) + l\dot{\xi}_1 \\ &= M^{-1}\boldsymbol{T}(t)\boldsymbol{\varphi}(t)\left[F(t) + u + \frac{d(t)}{\boldsymbol{T}(t)\boldsymbol{\varphi}(t)}\right] \end{aligned} \tag{11}$$

where $F(t) = M(\boldsymbol{T}(t)\boldsymbol{\varphi}(t))^{-1}\left(f(z, \dot{z}) + l\dot{\xi}_1\right)$.

Considering the saturation nonlinearity of the thruster, the control signal is loaded on the thruster according to Equation (9). When the control law output v is beyond the range of $[v_{\min}, v_{\max}]$, v is different from thruster control signal u. Therefore, the variable δ shown in Equation (12) is defined to indicate the difference between v and u.

$$\delta = v - u \tag{12}$$

From Equation (9), when $v_{\min} \leqslant v \leqslant v_{\max,}$ then $\delta = 0$. When $v > v_{\max}$, then $\delta > 0$. When $v < v_{\min}$, then $\delta < 0$.

Substituting Equation (12) into Equation (11) leads to:

$$\dot{s} = M^{-1}\boldsymbol{T}(t)\boldsymbol{\varphi}(t)\left[F(t) + v - \delta + \frac{d(t)}{\boldsymbol{T}(t)\boldsymbol{\varphi}(t)}\right] \tag{13}$$

In Equation (13), $F(t)$ contains information such as hydrodynamic force, external disturbance force and dead-zone nonlinearity, and it is usually difficult to obtain accurately by dynamic modelling. Therefore, the method which uses neural network to learn the unknown items in the model online is widely adopted, such as radial basis function neural network (Gao et al., 2014) and fuzzy neural network (Zhang et al., 2009), etc. A radial basis function neural network has the advantage of fast learning speed, but on the other hand it usually needs to select the radial basis function centre and the bandwidth offline according to the experimental data. Fuzzy neural network has good nonlinear identification ability, but the fuzzy field division depends on the designer's experience and the network structure is relatively complex. Based on the above consideration, three-layer feed-forward neural network is used for online identification in this paper. The weights from hidden layer to output layer and the weights from input layer to hidden layer in the network are obtained through online learning. The structure of the three-layer feed-forward neural network is shown in Figure 2.

According to the nonlinear mapping ability of the neural network, there are the optimal network weights $\boldsymbol{W}$ and $\boldsymbol{V}$ (Peng and Duba, 2012), which make:

$$F(t) = \boldsymbol{W}\boldsymbol{\sigma}(\boldsymbol{V}\boldsymbol{x}) + \varepsilon \tag{14}$$

where $\boldsymbol{W}$ is the weight vector from hidden layer to output layer, $\boldsymbol{V}$ is the weight matrix from input layer to hidden layer, $\boldsymbol{x} = [u, z, \dot{z}]^T$ is the input vector of neural network, $\boldsymbol{\varepsilon}$ is the approximation error of neural network, $\boldsymbol{\sigma}(\bullet)$ is a sigmoid function.

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

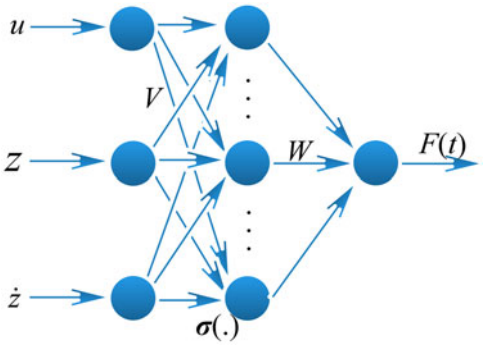

Figure 2. The structure of the three-layer feed-forward neural network.

According to Equation (14), the estimated output of the neural network can be expressed as:

$$\hat{F}(t) = \hat{\boldsymbol{W}}\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right) \tag{15}$$

where $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{V}}$ are the estimates of $\boldsymbol{W}$ and $\boldsymbol{V}$, respectively.

The estimation error of neural network identification should be analysed first. From Equations (14) and (15), the estimation error of neural network can be obtained as:

$$\tilde{F}(t) = F(t) - \hat{F}(t) = \tilde{\boldsymbol{W}}\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right) + \boldsymbol{W}\boldsymbol{\sigma}(\boldsymbol{V}\boldsymbol{x}) - \boldsymbol{W}\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right) + \varepsilon \tag{16}$$

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

For the adaptive law design of weights, we need to obtain the explicit expressions of $\tilde{\boldsymbol{W}}$ and $\tilde{\boldsymbol{V}}$. Therefore, the Taylor expansion of $\boldsymbol{\sigma}(\boldsymbol{V}\boldsymbol{x})$ about $\hat{\boldsymbol{V}}\boldsymbol{x}$ gives:

$$\boldsymbol{\sigma}(\boldsymbol{V}\boldsymbol{x}) = \boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right) + \boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\tilde{\boldsymbol{V}}\boldsymbol{x} + \boldsymbol{o}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)^2 \tag{17}$$

where $\boldsymbol{o}(\hat{\boldsymbol{V}}\boldsymbol{x})^2$ is the higher order item of Taylor expansion.

Introducing Equation (17) into Equation (16), the estimation error of the neural network can be rewritten as:

$$\tilde{F}(t) = \tilde{\boldsymbol{W}}\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right) + \hat{\boldsymbol{W}}\boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\tilde{\boldsymbol{V}}\boldsymbol{x} + \omega \tag{18}$$

where $\omega = \tilde{\boldsymbol{W}}\boldsymbol{\sigma}'(\hat{\boldsymbol{V}}\boldsymbol{x})\tilde{\boldsymbol{V}}\boldsymbol{x} + \boldsymbol{W}\boldsymbol{o}(\hat{\boldsymbol{V}}\boldsymbol{x})^2 + \varepsilon$.

It is assumed that the approximation error of the neural network is unknown and bounded, and the upper bound is known (Gao et al., 2015c; Yang and Wang, 2007), thus the stability of the closed-loop system can be ensured by using the sliding mode strategy to compensate. Although this hypothesis is tenable, the suitable upper bound value is difficult to select. A larger upper bound estimation can easily cause system chattering. To relax the qualifications, only $\tilde{F}(t)$ is to be assumed bounded and its upper bound is obtained by adaptive learning of an estimation item in this paper. Through introducing an adaptive estimation item into control law, the system chattering caused by the sliding mode strategy item can be avoided.

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

Assumption 1 (Zhang et al., 2009):

$$\left|\tilde{F}(t)\right| \leqslant C\psi(\boldsymbol{x}) \tag{19}$$

where C is an unknown positive constant, $\psi(\boldsymbol{x})$ is a known function about $\boldsymbol{x}$.

After the analysis of the sliding mode dynamics and estimation error of the neural network, the main conclusion of this paper will be given as follows:

Theorem 1: Consider the vertical dynamics of an ROV expressed as Equation (1) with thruster nonlinearity and un-modelled dynamics. If the control law in Equation (20) is introduced with the thruster control signal in Equation (9), and the control parameters in Equation (25), and the parameters' adaptive laws in Equations (21)–(24), the depth trajectory tracking error ξ_1 will be uniformly ultimately bounded.

$$v = -\hat{F}(t) - (K_1+K_3)s - \hat{C}\psi(\boldsymbol{x}) - \frac{\hat{\rho}}{\beta}\mathrm{sgn}(s) - K_2\mathrm{sgn}(s) \tag{20}$$

$$\dot{\hat{\boldsymbol{W}}} = \begin{cases} \Gamma_W\left[s\boldsymbol{\sigma}^T\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right) - \lambda_W\left(\hat{\boldsymbol{W}} - \boldsymbol{W}_o\right)\right] & \delta = 0 \text{ or } (\delta \geqslant 0 \text{ and } s \geqslant 0) \text{ or} \\ & (\delta \leqslant 0 \text{ and } s \leqslant 0) \\ \boldsymbol{0} & (\delta < 0 \text{ and } s > 0) \text{ or} \\ & (\delta > 0 \text{ and } s < 0) \end{cases} \tag{21}$$

$$\dot{\hat{\boldsymbol{V}}} = \begin{cases} \Gamma_V\left[s\left(\hat{\boldsymbol{W}}\boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right)^T\boldsymbol{x}^T - \lambda_V(\boldsymbol{V} - \boldsymbol{V}_o)\right] & \delta = 0, (\delta \geqslant 0 \text{ and } s \geqslant 0) \text{ or} \\ & (\delta \leqslant 0 \text{ and } s \leqslant 0) \\ 0 & (\delta < 0 \text{ and } s > 0) \text{ or} \\ & (\delta > 0 \text{ and } s < 0) \end{cases} \tag{22}$$

$$\dot{\hat{\rho}} = \begin{cases} \Gamma_\rho\left[|s|/\beta - \lambda_\rho\left(\hat{\rho} - \rho_o\right)\right] & \delta = 0 \text{ or } (\delta \geqslant 0 \text{ and } s \geqslant 0) \text{ or } (\delta \leqslant 0 \text{ and } s \leqslant 0) \\ 0 & (\delta < 0 \text{ and } s > 0) \text{ or } (\delta > 0 \text{ and } s < 0) \end{cases} \tag{23}$$

$$\dot{\hat{C}} = \begin{cases} \Gamma_C\left[|s|\psi(\boldsymbol{x}) - \lambda_C\left(\hat{C} - C_o\right)\right] & \delta = 0 \text{ or } (\delta \geqslant 0 \text{ and } s \geqslant 0) \text{ or } (\delta \leqslant 0 \text{ and } s \leqslant 0) \\ 0 & (\delta < 0 \text{ and } s > 0) \text{ or } (\delta > 0 \text{ and } s < 0) \end{cases} \tag{24}$$

$$K_1 = \begin{cases} \dfrac{\left\|\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|^2}{\lambda_W} + \dfrac{\left\|\hat{\boldsymbol{W}}\right\|^2\left\|\boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|^2\|\boldsymbol{x}\|^2}{\lambda_V} + \gamma & \delta = 0 \text{ or } (\delta \geqslant 0 \text{ and } s \geqslant 0) \text{ or} \\ & (\delta \leqslant 0 \text{ and } s \leqslant 0) \\ 0 & (\delta > 0 \text{ and } s < 0) \text{ or} \\ & (\delta < 0 \text{ and } s > 0) \end{cases}$$

$$K_2 = \begin{cases} \hat{C}\psi(\boldsymbol{x}) & \delta = 0 \text{ or } (\delta \geqslant 0 \text{ and } s \geqslant 0) \text{ or } (\delta \leqslant 0 \text{ and } s \leqslant 0) \\ 0 & (\delta > 0 \text{ and } s < 0) \text{ or } (\delta < 0 \text{ and } s > 0) \end{cases}$$

$$K_3 = \begin{cases} -M\dot{k}_l/\left(2\beta^2\right) & \delta = 0 \text{ or } (\delta \geqslant 0 \text{ and } s \geqslant 0) \text{ or } (\delta \leqslant 0 \text{ and } s \leqslant 0) \\ 0 & (\delta > 0 \text{ and } s < 0) \text{ or } (\delta < 0 \text{ and } s > 0) \end{cases} \tag{25}$$

where $\hat{C}$ is the estimated value of C, $\hat{\rho}$ is the estimated value of ρ. Γ_W, Γ_V, Γ_C, Γ_ρ, λ_W, λ_V, λ_C, λ_ρ are known positive constants, $\boldsymbol{W}_o$, $\boldsymbol{V}_o$, C_o, ρ_o are offline estimates of $\boldsymbol{W}$, $\boldsymbol{V}$, C, ρ, respectively.

Remark 4: For the control law Equation (20), there are several parameters which should be adjusted in the process of control, so a large amount of calculation will be needed. But this is not a problem for the control system of an ROV, because the

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

control cycle is usually set as 0·1 s to 1 s (Ma and Zeng, 2015) which makes enough time to calculate these parameters.

4. STABILITY ANALYSIS. In the stability analysis of most sliding mode controllers, the Lyapunov function usually contains s^2, and $s\dot{s} < 0$ will be acquired ultimately to prove that the control system is stable. However, since the saturation nonlinearity is considered, δ may take different values as different controller output signal v. When $\delta s \geqslant 0$, $-\delta s$ can be ignored by the stability theory. When $\delta s < 0$, its impact on the stability of the system must be analysed. Therefore, the situation aiming at different δ and s will be discussed separately.

Situation 1: $\delta = 0$ or ($\delta \geqslant 0$ and $s \geqslant 0$) or ($\delta \leqslant 0$ and $s \leqslant 0$)

Consider the Lyapunov function:

$$V_1 = \frac{1}{2}M(\boldsymbol{T}(t)\boldsymbol{\varphi}(t))^{-1}s^2 + \frac{1}{2}\Gamma_W^{-1}\tilde{\boldsymbol{W}}\tilde{W}^T + \frac{1}{2}tr\left[\tilde{\boldsymbol{V}}\Gamma_V^{-1}\tilde{\boldsymbol{V}}^T\right] + \frac{1}{2}\Gamma_C^{-1}\tilde{C}^2 + \frac{1}{2}\Gamma_\rho^{-1}\tilde{\rho}^2 \quad (26)$$

where $\tilde{C} = C - \hat{C}$, $\tilde{\rho} = \rho - \hat{\rho}$.

Differentiating V_1 and substituting Equation (13) into it, we have:

$$\begin{aligned}\dot{V}_1 = {} & \frac{1}{2}M\frac{d(\boldsymbol{T}(t)\boldsymbol{\varphi}(t))^{-1}}{dt}s^2 + s\left[F(t) + v - \delta + \frac{d(t)}{\boldsymbol{T}(t)\boldsymbol{\varphi}(t)}\right] \\ & + \Gamma_W^{-1}\tilde{\boldsymbol{W}}\dot{\tilde{W}}^T + tr\left[\tilde{\boldsymbol{V}}\Gamma_V^{-1}\dot{\tilde{\boldsymbol{V}}}^T\right] + \Gamma_C^{-1}\tilde{C}\dot{\tilde{C}} + \Gamma_\rho^{-1}\tilde{\rho}\dot{\tilde{\rho}}\end{aligned} \quad (27)$$

According to the definition and characteristics Equations (4) ~ (7), we can obtain:

$$\frac{d(\boldsymbol{T}(t)\boldsymbol{\varphi}(t))^{-1}}{dt} = \begin{cases} -\dfrac{\dot{\tau}_+(u)}{(\boldsymbol{T}(t)\boldsymbol{\varphi}(t))^2} & u \geqslant u_r \\ -\dfrac{\dot{\tau}_+(u) + \dot{\tau}_-(u)}{(\boldsymbol{T}(t)\boldsymbol{\varphi}(t))^2} & u_l < u < u_r \leqslant -\dot{k}_l/\beta^2 \\ -\dfrac{\dot{\tau}_-(u)}{(\boldsymbol{T}(t)\boldsymbol{\varphi}(t))^2} & u \leqslant u_l \end{cases} \quad (28)$$

According to Equation (28) and substituting control law Equations (20) and (26) into Equation (27), we obtain:

$$\begin{aligned}\dot{V}_1 \leqslant {} & s\left[F(t) - \hat{F}(t) - K_1 s - \hat{C}\psi(\boldsymbol{x}) - \frac{\hat{\rho}}{\beta}\mathrm{sgn}(s) - K_2\mathrm{sgn}(s) - \delta + \frac{d(t)}{\boldsymbol{T}(t)\boldsymbol{\varphi}(t)}\right] \\ & + \Gamma_W^{-1}\tilde{\boldsymbol{W}}\dot{\tilde{\boldsymbol{W}}}^T + tr\left[\tilde{\boldsymbol{V}}\Gamma_V^{-1}\dot{\tilde{\boldsymbol{V}}}^T\right] + \Gamma_C^{-1}\tilde{C}\dot{\tilde{C}} + \Gamma_\rho^{-1}\tilde{\rho}\dot{\tilde{\rho}} \\ = {} & s\left[\tilde{\boldsymbol{W}}\boldsymbol{\sigma}(\hat{\boldsymbol{V}}\boldsymbol{x}) + \hat{\boldsymbol{W}}\boldsymbol{\sigma}'(\hat{\boldsymbol{V}}\boldsymbol{x})\tilde{\boldsymbol{V}}\boldsymbol{x} + \omega - K_1 s - \hat{C}\psi(x)\right. \\ & \left. - \frac{\hat{\rho}}{\beta}\mathrm{sgn}(s) - K_2\mathrm{sgn}(s) - \delta + \frac{d(t)}{\boldsymbol{T}(t)\boldsymbol{\varphi}(t)}\right] \\ & + \Gamma_W^{-1}\tilde{\boldsymbol{W}}\dot{\tilde{\boldsymbol{W}}}^T + tr\left[\tilde{\boldsymbol{V}}\Gamma_V^{-1}\dot{\tilde{\boldsymbol{V}}}^T\right] + \Gamma_C^{-1}\tilde{C}\dot{\tilde{C}} + \Gamma_\rho^{-1}\tilde{\rho}\dot{\tilde{\rho}}\end{aligned} \quad (29)$$

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

Since $\delta s \geqslant 0$, Equation (29) can be expressed as:

$$\dot{V}_1 \leqslant s\left[\tilde{\boldsymbol{W}}\boldsymbol{\sigma}(\hat{\boldsymbol{V}}\boldsymbol{x})+\hat{\boldsymbol{W}}\boldsymbol{\sigma}'(\hat{\boldsymbol{V}}\boldsymbol{x})\tilde{\boldsymbol{V}}\boldsymbol{x}+\omega-K_1 s-\hat{C}\psi(\boldsymbol{x})-\frac{\hat{\rho}}{\beta}\mathrm{sgn}(s)-K_2\mathrm{sgn}(s)+\frac{d(t)}{\boldsymbol{T}(t)\boldsymbol{\varphi}(t)}\right] + \Gamma_W^{-1}\tilde{\boldsymbol{W}}\dot{\tilde{\boldsymbol{W}}}^T + tr\left[\tilde{\boldsymbol{V}}\Gamma_V^{-1}\dot{\tilde{\boldsymbol{V}}}^T\right] + \Gamma_C^{-1}\tilde{C}\dot{\tilde{C}} + \Gamma_\rho^{-1}\tilde{\rho}\dot{\tilde{\rho}} \tag{30}$$

Introducing adaptive law Equations (21) and (22) into Equation (30), we obtain:

$$\begin{aligned}\dot{V}_1 \leqslant & -K_1 s^2 + s\left[\omega - \hat{C}\psi(\boldsymbol{x})\right] - K_2|s| + \frac{\tilde{\rho}}{\beta}|s| \\ & + \lambda_W\tilde{\boldsymbol{W}}(\hat{\boldsymbol{W}}-\boldsymbol{W}_o)^T + tr\left[\lambda_V\tilde{\boldsymbol{V}}(\hat{\boldsymbol{V}}-\boldsymbol{V}_o)^T\right] + \Gamma_C^{-1}\tilde{C}\dot{\tilde{C}} + \Gamma_\rho^{-1}\tilde{\rho}\dot{\tilde{\rho}}\end{aligned} \tag{31}$$

Introducing adaptive law Equations (23) into (31), and $K_2 = 0$, we obtain:

$$\begin{aligned}\dot{V}_1 \leqslant & -K_1 s^2 + s\left[\tilde{F}(t) - \hat{C}\psi(\boldsymbol{x})\right] \\ & + s\left[-\tilde{\boldsymbol{W}}\boldsymbol{\sigma}(\hat{\boldsymbol{V}}\boldsymbol{x}) - \hat{\boldsymbol{W}}\boldsymbol{\sigma}'(\hat{\boldsymbol{V}}\boldsymbol{x})\tilde{\boldsymbol{V}}\boldsymbol{x}\right] + \lambda_W\tilde{\boldsymbol{W}}(\hat{\boldsymbol{W}}-\boldsymbol{W}_o)^T \\ & + tr\left[\lambda_V\tilde{\boldsymbol{V}}(\hat{\boldsymbol{V}}-\boldsymbol{V}_o)^T\right] + \lambda_\rho\tilde{\rho}(\hat{\rho}-\rho_o) + \Gamma_C^{-1}\tilde{C}\dot{\tilde{C}} \\ \leqslant & -K_1 s^2 + |s|\tilde{C}\psi(\boldsymbol{x}) + |s|\left\|\tilde{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}(\hat{\boldsymbol{V}}\boldsymbol{x})\right\| + |s|\left\|\hat{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}'(\hat{\boldsymbol{V}}\boldsymbol{x})\right\|\left\|\tilde{\boldsymbol{V}}\right\|\|\boldsymbol{x}\| \\ & + \lambda_W\tilde{\boldsymbol{W}}(\hat{\boldsymbol{W}}-\boldsymbol{W}_o)^T + tr\left[\lambda_V\tilde{\boldsymbol{V}}(\hat{\boldsymbol{V}}-\boldsymbol{V}_o)^T\right] + \lambda_\rho\tilde{\rho}(\hat{\rho}-\rho_o) + \Gamma_C^{-1}\tilde{C}\dot{\tilde{C}}\end{aligned} \tag{32}$$

Introducing adaptive law Equations (24) into (32), we obtain:

$$\begin{aligned}\dot{V}_1 \leqslant & -K_1 s^2 + |s|\left\|\tilde{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}(\hat{\boldsymbol{V}}\boldsymbol{x})\right\| + |s|\left\|\hat{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}'(\hat{\boldsymbol{V}}\boldsymbol{x})\right\|\left\|\tilde{\boldsymbol{V}}\right\|\|\boldsymbol{x}\| + \lambda_W\tilde{\boldsymbol{W}}(\hat{\boldsymbol{W}}-\boldsymbol{W}_o)^T \\ & + tr\left[\lambda_V\tilde{\boldsymbol{V}}(\hat{\boldsymbol{V}}-\boldsymbol{V}_o)^T\right] + \lambda_\rho\tilde{\rho}(\hat{\rho}-\rho_o) + \lambda_C\tilde{C}\left(\hat{C}-C_o\right)\end{aligned} \tag{33}$$

Since:

$$\lambda_W\tilde{\boldsymbol{W}}(\hat{\boldsymbol{W}}-\boldsymbol{W}_o)^T = \lambda_W\tilde{\boldsymbol{W}}(\boldsymbol{W}-\tilde{\boldsymbol{W}}-\boldsymbol{W}_o)^T \leqslant -\frac{\lambda_W}{2}\left\|\tilde{\boldsymbol{W}}\right\|^2 + \frac{\lambda_W}{2}\|\boldsymbol{W}-\boldsymbol{W}_o\|^2 \tag{34}$$

$$tr\left[\lambda_V\tilde{\boldsymbol{V}}(\hat{\boldsymbol{V}}-\boldsymbol{V}_o)^T\right] = tr\left[\lambda_V\tilde{\boldsymbol{V}}(\boldsymbol{V}-\tilde{\boldsymbol{V}}-\boldsymbol{V}_o)^T\right] \leqslant -\frac{\lambda_V}{2}\left\|\tilde{\boldsymbol{V}}\right\|^2 + \frac{\lambda_V}{2}\|\boldsymbol{V}-\boldsymbol{V}_o\|^2 \tag{35}$$

$$\lambda_\rho\tilde{\rho}(\hat{\rho}-\rho_o) = \lambda_\rho\tilde{\rho}(\rho-\tilde{\rho}-\rho_o) \leqslant -\frac{\lambda_\rho}{2}\tilde{\rho}^2 + \frac{\lambda_\rho}{2}(\rho-\rho_o)^2 \tag{36}$$

$$\lambda_C\tilde{C}\left(\hat{C}-C_o\right) = \lambda_C\tilde{C}(C-\tilde{C}-C_o) \leqslant -\frac{\lambda_C}{2}\tilde{C}^2 + \frac{\lambda_C}{2}(C-C_o)^2 \tag{37}$$

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

Then:

$$\begin{aligned}\dot{V}_1 \leqslant & -K_1 s^2 + |s|\left\|\tilde{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\| + |s|\left\|\hat{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|\left\|\tilde{\boldsymbol{V}}\right\|\|\boldsymbol{x}\| \\ & -\frac{\lambda_W}{2}\left\|\tilde{\boldsymbol{W}}\right\|^2+\frac{\lambda_W}{2}\|\boldsymbol{W}-\boldsymbol{W}_o\|^2-\frac{\lambda_V}{2}\left\|\tilde{\boldsymbol{V}}\right\|^2 \\ & +\frac{\lambda_V}{2}\|\boldsymbol{V}-\boldsymbol{V}_o\|^2-\frac{\lambda_\rho}{2}\tilde{\rho}^2+\frac{\lambda_\rho}{2}\left(\rho-\rho_o\right)^2-\frac{\lambda_C}{2}\tilde{C}^2+\frac{\lambda_C}{2}(C-C_o)^2\end{aligned} \tag{38}$$

Because of:

$$|s|\left\|\tilde{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\| - \frac{\lambda_W}{4}\left\|\tilde{\boldsymbol{W}}\right\|^2 \leqslant \frac{s^2\left\|\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|^2}{\lambda_W} \tag{39}$$

$$|s|\left\|\hat{\boldsymbol{W}}\right\|\left\|\boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|\left\|\tilde{\boldsymbol{V}}\right\|\|\boldsymbol{x}\| - \frac{\lambda_V}{4}\left\|\tilde{\boldsymbol{V}}\right\|^2 \leqslant \frac{s^2\left\|\hat{\boldsymbol{W}}\right\|^2\left\|\boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|^2\|\boldsymbol{x}\|^2}{\lambda_V} \tag{40}$$

Introducing Equations (39) and (40) and K_1 into Equation (32), we obtain:

$$\begin{aligned}\dot{V}_1 \leqslant & -K_1 s^2 - \frac{\lambda_\rho}{2}\tilde{\rho}^2 - \frac{\lambda_C}{2}\tilde{C}^2 - \frac{\lambda_W}{4}\left\|\tilde{\boldsymbol{W}}\right\|^2-\frac{\lambda_V}{4}\left\|\tilde{\boldsymbol{V}}\right\|^2 \\ & +\frac{s^2\left\|\boldsymbol{\sigma}\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|^2}{\lambda_W}+\frac{s^2\left\|\hat{\boldsymbol{W}}\right\|^2\left\|\boldsymbol{\sigma}'\left(\hat{\boldsymbol{V}}\boldsymbol{x}\right)\right\|^2\|\boldsymbol{x}\|^2}{\lambda_V} \\ & +\frac{\lambda_W}{2}\|\boldsymbol{W}-\boldsymbol{W}_o\|^2+\frac{\lambda_V}{2}\|\boldsymbol{V}-\boldsymbol{V}_o\|^2+\frac{\lambda_\rho}{2}\left(\rho-\rho_o\right)^2+\frac{\lambda_C}{2}(C-C_o)^2 \\ \leqslant & -\chi_1 V+\lambda_1\end{aligned} \tag{41}$$

where $\chi_1 = \min\{2\gamma k_r M^{-1}, 2\gamma k_l M^{-1}, \Gamma_\rho\lambda_\rho, \Gamma_C\lambda_C, \Gamma_W\lambda_W/2, \Gamma_V\lambda_V/2\}$ which is a positive constant, λ_1 is shown in Equation (42).

$$\lambda_1 = \frac{\lambda_W}{2}\|\boldsymbol{W}-\boldsymbol{W}_o\|^2+\frac{\lambda_V}{2}\|\boldsymbol{V}-\boldsymbol{V}_o\|^2+\frac{\lambda_\rho}{2}\left(\rho-\rho_o\right)^2+\frac{\lambda_C}{2}(C-C_o)^2 \tag{42}$$

From Equation (41), we can obtain:

$$0 \leqslant V_1 \leqslant \frac{\lambda_1}{\chi_1} + \left[V_1(0) - \frac{\lambda_1}{\chi_1}\right]\exp\left(-\chi_1 t\right) \tag{43}$$

where $V_1(0)$ is the initial value of V_1 at $t = 0$.

From Equation (43), when $t \to \infty$, $V_1 \to \lambda_1/\chi_1$, namely that s, $\tilde{\boldsymbol{V}}$, $\tilde{C}$, $\tilde{\rho}$ are uniformly ultimately bounded, which means that trajectory tracking error ξ_1 is uniformly ultimately bounded.

Situation 2: $\delta < 0$ and $s > 0$

When $\delta < 0$, then $v < v_{\min}$ and $u = v_{\min}$. For this situation, the adaptive laws Equations (21)-(24) are set as zero, so only the convergence of trajectory tracking error ξ_1 will be analysed.

According to the definition of s, when $s > 0$, then $\xi_1 > 0$ and $\xi_2 < 0$, or $\xi_1 < 0$ and $\xi_2 > 0$, or $\xi_1 > 0$ and $\xi_2 > 0$. Obviously, ξ_1 will converge to zero for the previous two cases, so

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

that the control system is stable. Therefore, only the situations of $\xi_1 > 0$ and $\xi_2 > 0$ need to be considered.

When $u = v_{\min}$, $\ddot{z}$ will be less than zero and $\dot{z}$ will decrease until $\dot{z}=\dot{z}_{\min}$, which makes $\xi_2 = \dot{z}_{\min} - \dot{z}_d < 0$. Therefore, $\xi_2 > 0$ is just for a short time and its value will be less than zero eventually, so that ξ_1 will also converge to zero.

If $u = v_{\min}$ continues being used, both ξ_1 and ξ_2 will be less than zero, that is $s < 0$. That satisfies *Situation 1*.

Situation 3: $\delta > 0$ and $s < 0$

When $\delta > 0$, then $v > v_{\max}$ and $u = v_{\max}$. For this situation, the adaptive laws Equation (21)-(24) are also set as zero, so only the convergence of trajectory tracking error ξ_1 like *Situation 2* is to be analysed.

When $s < 0$, then $\xi_1 > 0$ and $\xi_2 < 0$, or $\xi_1 < 0$ and $\xi_2 > 0$, or $\xi_1 < 0$ and $\xi_2 < 0$. Obviously, ξ_1 will converge to zero in the previous two cases, so that the control system is stable. Therefore, only the situations of $\xi_1 < 0$ and $\xi_2 < 0$ need to be considered.

When $u = v_{\max}$, $\ddot{z}$ will be larger than zero and then $\dot{z}$ will increase until $\dot{z}_{\max}$. That makes $\xi_2 = \dot{z}_{\max} - \dot{z}_d > 0$. Therefore, $\xi_2 < 0$ is just for a short time and its value will be larger than zero eventually, so that ξ_1 will also converge to zero.

If $u = v_{\max}$ continues being used, both ξ_1 and ξ_2 will be larger than zero, that is $s > 0$. That satisfies *Situation 1*.

Based on the above analysis, it can be concluded that the designed control law Equation (20) can ensure the stability of ROV depth trajectory tracking control and that the depth trajectory tracking error is uniformly ultimately bounded.

5. SIMULATION RESULTS.

To verify the effectiveness of the proposed adaptive sliding mode controller in this paper, the ODIN vertical motion model is used for simulation experiment. The ODIN vertical motion model is shown in Equation (44) (Podder and Sarkar, 2001).

$$187{\cdot}4\ddot{z} + 148|\dot{z}|\dot{z} + 100\dot{z} - 2{\cdot}7 + 10\sin(0{\cdot}2\pi t) + \text{rand}(-10, 10) = \tau(u) \tag{44}$$

where rand(−10,10) is the random number within [−10,10]. It is assumed that only one thruster works in the ODIN vertical direction. Thrust $\tau(u)$ is calculated by:

$$\tau(u) = \begin{cases} -1{\cdot}27u^3 + 10{\cdot}35u^2 - 8{\cdot}40u + 1{\cdot}77 & 0{\cdot}5 \leqslant u \leqslant 5{\cdot}0 \\ 0 & -0{\cdot}5 < u < 0{\cdot}5 \\ -1{\cdot}27u^3 - 10{\cdot}35u^2 - 8{\cdot}40u - 1{\cdot}77 & -5{\cdot}0 \leqslant u \leqslant -0{\cdot}5 \end{cases} \tag{45}$$

According to the thrust model Equation (45), it can be seen that $u_l = -0{\cdot}5$ V, $u_r = 0{\cdot}5$ V, $v_{\min} = -5$ V, $v_{\max} = 5$ V. The maximum forward and reverse thrust are both 60N.

According to the given motion model Equation (44) and thrust model Equation (45), the relevant parameters in adaptive sliding mode controller Equation (20) are selected as: $\beta = 1$, $l = 5$, $\gamma = 3$, $\psi(\boldsymbol{x}) = 1$, $m_0 = 188$, $\Gamma_W = \Gamma_V = \Gamma_C = \Gamma_\rho = 1$, $\lambda_W = \lambda_V = \lambda_C = \lambda_\rho = 0{\cdot}5$. The initial values of $\hat{\boldsymbol{W}}$, $\hat{V}$, $\hat{C}$ and $\hat{\rho}$ are randomly selected within the range [−0·1,0·1], $\boldsymbol{W_o} = \hat{\boldsymbol{W}}$, $\boldsymbol{V}_o = \hat{\boldsymbol{V}}$, $C_o = \hat{C}$, $\rho_o = \hat{\rho}$. The number of hidden layer units of three-layer feed-forward neural network is 10. The initial depth of the ROV is 1·0 m and the control cycle is 0·1 s. Two simulations of sine trajectory tracking control and depth setting tracking control are performed, respectively. In them, the

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

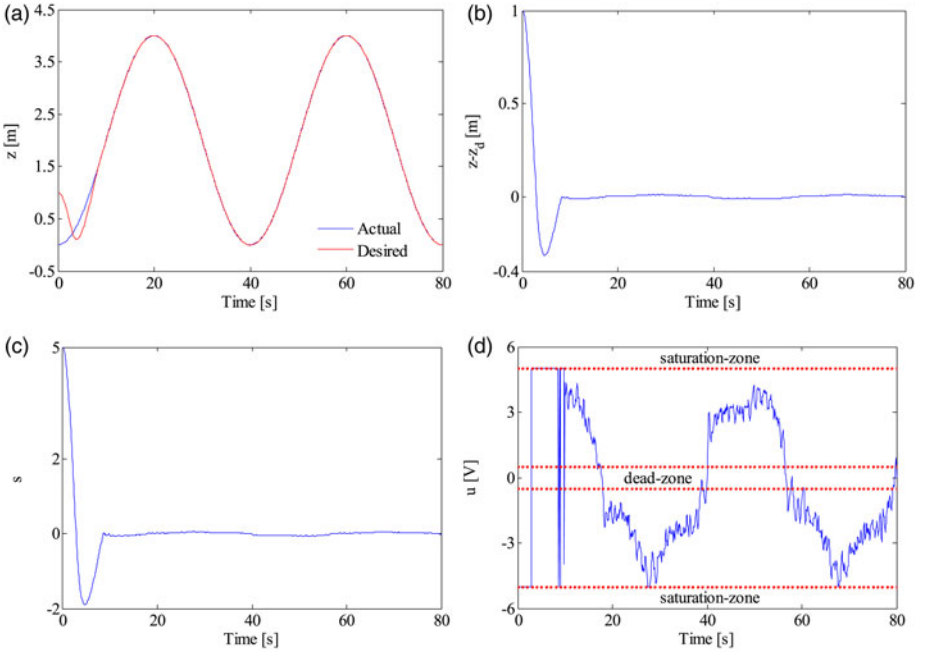

Figure 3. Simulation curve of sine depth trajectory tracking control. (a) Depth z. (b) Depth tracking error $\xi_1 = z\text{-}z_d$. (c) Sliding mode s. (d) Thruster control signal u.

desired depth of the ROV is $z_d = 2(1 - \cos(0{\cdot}05\pi t))$ in the simulation of sine trajectory tracking control, and the desired depth of ROV in the simulation of depth setting tracking control is shown in Equation (46). Two simulation curves are shown in Figures 3 and 4, respectively.

$$z_d = \begin{cases} 0{\cdot}5 & t \leqslant 20s \\ 2{\cdot}0 & 20s < t \leqslant 50s \\ 1{\cdot}0 & 50s < t \leqslant 80s \end{cases} \tag{46}$$

As can be observed from Figures 3 and 4, the depth output follows the desired trajectory well and the depth tracking error ξ_1 and the sliding mode s are close to zero. In Figure 3(b), the depth tracking error converges to zero at 9·6 s. From 9·7 s to 80 s, the average value of depth tracking error is 0·001 m. In Figure 4(b), the depth tracking error converges to zero at 4·7 s. From 4·8 s to 20 s, the average value of depth tracking error is −0·016 m. This indicates that the trajectory tracking error is uniformly ultimately bounded. From Figure 3(d) and Figure 4(d), it can be observed that the thruster works in the dead zone and saturation zone for a certain time, and the closed-loop system is stable in this period. Simulation results show that the proposed controller is well capable of the depth trajectory tracking and the proposed method is effective for the dead-zone nonlinearity and saturation nonlinearity of the thruster.

https://doi.org/10.1017/S0373463316000448 Published online by Cambridge University Press

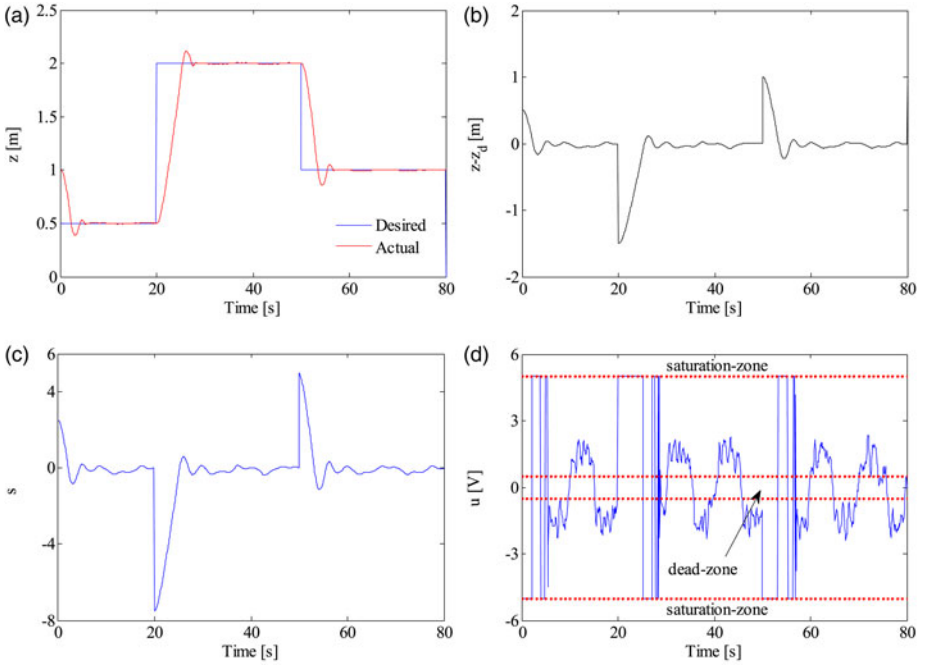

Figure 4. Simulation curve of depth setting tracking control. (a) Depth z. (b) Depth tracking error $\xi_1 = z\text{-}z_d$. (c) Sliding mode *s*. (d) Thruster control signal *u*.

6. CONCLUSION. The ROV adaptive sliding mode depth trajectory tracking control method is researched, and the problem of dead-zone nonlinearity and saturation nonlinearity of thruster is considered during the process of controller design. For an unknown thrust model, the proposed controller is able to obtain the control signal directly. Under the condition of un-modelled dynamics, three-layer feed-forward neural network is used for online adaptive identification. The proposed controller can ensure that the depth trajectory tracking error is uniformly ultimately bounded. Simulation results indicate that the proposed method is effective for ROV depth trajectory tracking control with the dead-zone nonlinearity and saturation nonlinearity of thruster. However, we need to realise that only depth tracking has been considered in this paper. In future work, the problem of trajectory tracking control of multiple degrees of ROV with dead-zone nonlinearity and saturation nonlinearity should be researched, so that automatic control of an ROV can be achieved. This will be more challenging because the transformation matrix which expresses the transformation from the body-fixed frame to earth-fixed frame will be introduced and the transformation matrix is very difficult to be processed in controller design based on the proposed method.

ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China under Grant 51509150, 51575336; Shanghai Municipal Natural Science Foundation under Grant 15ZR1419700.

REFERENCES

Alessandri, A., Caccia, M. and Veruggio, G. (1999). Fault detection of actuator faults in unmanned underwater vehicles. *Control Engineering Practice*, **7**, 357–368.

Avila, J.P.J, Adamowski, J.C., Maruyama, N. and Takase, F.K. (2012). Modeling and identification of an open-frame underwater vehicle: the yaw motion dynamics. *Journal of Intelligent & Robot Systems*, **66**, 37–56.

Avila, J.P.J., Donha, D.C. and Adamowski, J.C. (2013). Experimental model identification of open-frame underwater vehicles. *Ocean Engineering*, **60**, 81–94.

Bessa, W.M., Dutra, M.S. and Kreuzer, E. (2008). Depth control of remotely operated underwater vehicles using an adaptive fuzzy sliding mode controller. *Robotics and Autonomous Systems*, **56**, 670–677.

Bessa, W.M., Dutra, M.S. and Kreuzer, E. (2010). An adaptive fuzzy sliding mode controller for remotely operated underwater vehicles. *Robotics and Autonomous Systems*, **58**, 16–26.

Cao, X. and Zhu, D. (2015). Multi-AUV underwater cooperative search algorithm based on biological inspired neurodynamics model and velocity synthesis. *The Journal of Navigation*, **68**, 1075–1087.

Chen, M., Ge, S.S. and Ren, B. (2011). Adaptive tracking control of uncertain MIMO nonlinear systems with input constraints. *Automatica*, **47**, 452–465.

Chu, Z., Zhu, D. and Yang, S.X. (2016a). Observer-based adaptive neural network trajectory tracking control for remotely operated vehicle. *IEEE Transaction on Neural Networks and Learning Systems*, DOI: 10.1109/TNNLS.2016.2544786.

Chu, Z., Zhu, D. and Yang, S.X. (2016b). Adaptive terminal sliding mode based sensorless speed control for underwater thruster. *International Journal of Robotics and Automation*, DOI: 10.2316/Journal.206.2016.3.206–4428.

Feng, Z. and Allen, R. (2004). Reduced order H∞ control of an autonomous underwater vehicle. *Control Engineering Practice*, **12**, 1511–1520.

Gan, Y., Wang, L.R., Liu, J.C. and Xu, Y.R. (2004). The embedded basic motion control system of autonomous underwater vehicle. *Robot*, **26**, 246–250.

Gao, D.X., Wang, S.X. and Zhang, H.J. (2014). A singularly perturbed system approach to adaptive neural back-stepping control design of hypersonic vehicles. *Journal of Intelligent & Robotic Systems*, **73**, 249–259.

Gao., J., Proctor, A. and Bradley, C. (2015a). Adaptive neural network visual servo control for dynamic positioning of underwater vehicles. *Neurocomputing*, **167**, 604–613.

Gao, J., Proctor, A., Shi, Y. and Bradley, C. (2015b). Hierarchical model predictive image-based visual servoing of underwater vehicles with adaptive neural network dynamic control. *IEEE Transactions on Cybernetics*, DOI: 10.1109/TCYB.2015.2475376.

Gao, W., Yang, J., Liu, J., Shi, H.Y. and Xu, B. (2015c). Moving horizon estimation for cooperative localization with communication delay. *Journal of Navigation*, **68**, 493–510.

Hoang, N.Q. and Kreuzer, E. (2007). Adaptive PD-controller for positioning of a remotely operated vehicle close to an underwater structure: Theory and experiments. *Control Engineering Practice*, **15**, 411–419.

Hussain, M.A., Ho, P.Y. (2004). Adaptive sliding mode control with neural network based hybrid models. *Journal of Process Control*, **14**, 157–176.

Kim, J. and Chung, W.K. (2006). Accurate and practical thruster modelling for underwater vehicles, *Ocean Engineering*, **33**, 566–586.

Liu, Y.J. and Zhou, N. (2010). Observer-based adaptive fuzzy-neural control for a class of uncertain nonlinear systems with unknown dead-zone input. *ISA Transactions*, **49**, 462–469.

Ma, C. and Zeng, Q. (2015). Distributed formation control of 6-DOF autonomous underwater vehicles networked by sampled-data information under directed topology. *Neurocomputing*, **154**, 33–40.

Pan, H. and Xin, M. (2012). Depth control of autonomous underwater vehicle using indirect robust control method. *International Journal of Control*, **85**, 98–113.

Peng, J. and Duba, R. (2012). Nonlinear inversion-based control with adaptive neural network compensation for uncertain MIMO systems, *Expert Systems with Applications*, **39**, 8162–8171.

Podder, T.P. and Sarkar, N. (2001). Fault-tolerant control of an autonomous underwater vehicle under thruster redundancy. *Robotics and Autonomous Systems*, **34**, 39–52.

Sun, B., Zhu, D. and Yang, X. (2014). A bioinspired filtered backstepping tracking control of 7000-m manned submarine vehicle. *IEEE Transactions on Industrial Electronics*, **61**, 3682–3693.

Tong, S., Sui, S. and Li, Y. (2013). Adaptive fuzzy decentralized control for stochastic large-scale nonlinear systems with unknown dead-zone and unmodeled dynamics. *Neurocomputing*, **135**, 367–377.

Wang, Z.H., Ge, T. and Zhu, J.M. (2006). Timing sequence and logicality design of ROV's dynamic positioning system. *Ocean Engineering*, **24**, 61–66.

Wu, W., Gao, L., Mei, D. and Zhou, S. (2012). L1 adaptive controller for aircraft attitude with input constraints. *Journal of Nanjing University of Aeronautics & Astronautics*, **44**, 809–817.

Yang, Y.S. and Wang, X.F. (2007). Adaptive H∞ tracking control for a class of uncertain nonlinear systems using radial-basis-function neural networks. *Neurocomputing*, **70**, 932–941.

Zhang, L.J., Qi, X. and Pang, Y.J. (2009). Adaptive output feedback control based on DRFNN for AUV. *Ocean Engineering*, **36**, 716–722.

Zhang, M.J. and Chu, Z.Z. (2012). Adaptive sliding mode control based on local recurrent neural networks for underwater robot. *Ocean Engineering*, **45**, 56–62.

Zhang, M., Liu, X., Yin, B. and Liu, W. (2015). Adaptive terminal sliding mode based thruster fault tolerant control for underwater vehicle in time-varying ocean currents. *Journal of the Franklin Institute*, **352**, 4935–4961.

Zhu, D.Q., Hua, X. and Sun, B. (2014). A neurodynamics control strategy for real-time tracking control of autonomous underwater vehicle. *Journal of Navigation*, **67**, 113–127.