Journal of
Marine Science and Engineering

MDPI

Article

Neural Network Non-Singular Terminal Sliding Mode Control for Target Tracking of Underactuated Underwater Robots with Prescribed Performance

Liwei Guo [1], Weidong Liu [1], Le Li [1,*], Yichao Lou [1], Xinliang Wang [2] and Zhi Liu [2]

[1] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China; 2019100504@mail.nwpu.edu.cn (L.G.); liuwd@nwpu.edu.cn (W.L.); liyl@mail.nwpu.edu.cn (Y.L.)

[2] The Second Wuhan Ship Design and Research Institute, Wuhan 430205, China; xinliang_w@sina.com (X.W.); zhyy5410@whu.edu.cn (Z.L.)

* Correspondence: leli@nwpu.edu.cn

Abstract: This paper proposes a neural network-based nonsingular terminal sliding mode controller with prescribed performances for the target tracking problem of underactuated underwater robots. Firstly, the mathematical formulation of the target tracking problem is presented with an underactuated underwater robot model and the corresponding control objectives. Then, the target tracking errors from the line-of-sight guidance law are transformed using the prescribed performance technique to achieve good dynamic performance and steady-state performance that meet the pre-set conditions. Meanwhile, considering the model's uncertainties and the external disturbances to the underwater robots, a target tracking controller is proposed based on the radial basis function (RBF) neural network and the non-singular terminal sliding mode control. Lyapunov stability analysis and homogeneity theory prove the tracking errors can converge on a small region that contains the origin with prescribed performance in finite time. In the simulation comparison, the controller proposed in this paper had better dynamic performance, steady-state performance and chattering supression. In particular, the steady-state error of the tracking error was lower, and the convergence time of the tracking error in the vertical distance was reduced by 19.1%.

Keywords: underwater robot; target tracking; neural network; non-singular terminal sliding mode; prescribed performance

check for updates

Citation: Guo, L.; Liu, W.; Li, L.; Lou, Y.; Wang, X.; Liu, Z. Neural Network Non-Singular Terminal Sliding Mode Control for Target Tracking of Underactuated Underwater Robots with Prescribed Performance. *J. Mar. Sci. Eng.* **2022**, *10*, 252. https://doi.org/10.3390/jmse10020252

Academic Editors: Alessandro Ridolfi and Sergei Chernyi

Received: 6 December 2021
Accepted: 8 February 2022
Published: 12 February 2022

1. Introduction

In recent years, underwater robots have been widely used in various underwater tasks. Typical applications include search and rescue, monitoring and surveillance, petroleum exploration, deep-sea archaeological research, ship hull maintenance industry [1–3] and so on. In the underwater applications mentioned above, underwater robots are often required to track the targets in a fast and accurate manner. As underwater robots generally have the characteristics of many uncertainties, high nonlinearity and strong coupling dynamics, and work in an environment with unknown external disturbances, it is typically difficult for traditional linear controllers to achieve good tracking control performance.

Many advanced control methods have been used in the control of underwater robots, such as backstepping control [4–6], model predictive control [7,8], neural network control [9–13], active disturbance rejection control [14–16], sliding mode control [17–23], adaptive and distributed control [24–26] and reinforcement learning [27–30]. In the above methods, sliding mode control has been widely studied or used because of its strong robustness to the external time-varying disturbance and unmodeled characteristics, fast responses and easy implementation. Yingkai Xia et al. designed of an improved line-of-sight-based adaptive sliding mode tracking controller for highly coupled dynamics, ocean current disturbances and input saturation of underactuated autonomous underwater vehicles

(AUVs) [21]. For the problems of the remote operated vehicle (ROV) movement instability and large tracking error caused by unknown disturbances, an adaptive sliding mode motion controller was developed by Zongsheng Wang et al. [20]. However, the standard sliding mode control can only realize the asymptotic stability of the system, and cannot guarantee the convergence time. As an effective method to speed up the convergence of tracking errors, terminal sliding mode control achieves finite-time stability by introducing fractional order terms when constructing a sliding hyperplane. For underactuated underwater vehicles, Taha Elmokadem et al. designed a target tracking controller based on terminal sliding mode control, and proved that the tracking errors can converge to zero within a specified finite time [18]. However, the negative fractional power terms contained in the controller may lead to a singularity when the errors converge. Cao Jian et al. designed a non-singular terminal sliding mode controller that evades the issue of a singularity in the target tracking problem of underwater robots [31]. Nevertheless, it is necessary to obtain the upper bound of uncertainty in advance. Strong robustness can be obtained by setting a larger switching gain. However, this will lead to a serious chattering problem, affect the control accuracy and increase the energy consumption, which are harmful to the tracking control of underwater robots. B.M. Patre et al. added a state observer to the sliding mode controller to reduce the switching gains [19], thereby attenuating the chattering. The precondition of applying this method is that the uncertainty changes slowly and the first derivative is almost zero, which is inconsistent with the actual working environment of underwater robots. In practice, underwater robots are always subject to various uncertainties, such as unknown parameters, unmodeled time-varying dynamics and measurement noise. A neural network can approximate nonlinear time-varying functions and has been becoming one of the most effective ways to observe uncertainties [9,12,13]. It is worth noting that a neural network can reduce the switching gain of a sliding mode controller by approximating unmodeled uncertainties. Therefore, a terminal sliding mode controller combined with RBF neural network technology improves the tracking control performance while attenuating chattering.

As underwater tasks are becoming more complex, it is important to ensure good target tracking of underwater robots. While improving the robustness to external disturbances and modeling uncertainties, the controller should also pay attention to the dynamic performance and steady-state performance of the tracking errors. The performance control parameters proposed by Charalampos P. Bechlioulis et al. include constraints to limit errors, which not only ensures that the steady-state errors are always less than a pre-set boundary, but also limits the dynamic performance of the system state errors, including convergence rate and overshoot, to meet the pre-set conditions [32]. Charalampos P. Bechlioulis et al. developed an approximation-free trajectory tracking controller for underactuated AUVs with prescribed performances [33]. Nevertheless, this controller ignored the existence of external disturbances and modeling uncertainties. Omid Elhaki et al. created a neural network-based target tracking controller for an underactuated AUV with a prescribed performance to overcome unmodeled dynamics and external disturbances [10]. The prescribed performance technique is applied to trajectory tracking to prevent collisions and improve control performance [34,35]. Furthermore, a finite-time performance function [36] and a nonlogarithmic piecewise error mapping function [37] have also been introduced to accelerate convergence. However, the finite-time convergence of the controllers has not been investigated in detail.

Differently from previous research, in order to achieve robustness to uncertainties and external disturbances, realize finite-time convergence, attenuate chattering and obtain the tracking error's prescribed performance simultaneously, this paper proposes a neural network nonsingular terminal sliding mode controller with prescribed performance for the target tracking problem of underactuated underwater robots. Compared with the nonsingular terminal sliding mode controller, the proposed controller has obvious advantages in dynamic performance and steady-state performance. Additionally, its chattering is weaker than that of a nonsingular terminal sliding mode controller with prescribed performances.

Firstly, the mathematical formulation of the target tracking problem is proposed for the underactuated underwater robot model and the corresponding control objectives. The target tracking guidance law is designed based on the line-of-sight. The range and bearing angles of the robot relative to the target are obtained as tracking errors. Then, the range and bearing angles are transformed into corresponding transformation errors with prescribed performances. The tracking errors converge to arbitrarily small limit bounds, and the dynamic performance is optimized with a prespecified maximum overshoot and the convergence speed. The non-singular terminal sliding mode controller was developed to ensure that the underwater robot is robust to external disturbances and modeling uncertainties, and guarantee finite-time convergence of the tracking errors. For improving the accuracy of tracking control and attenuating the chattering of sliding mode control, an RBF neural network estimator is integrated to approximate modeling uncertainties. Finally, a Lyapunov stability synthesis and homogeneity theory show that the tracking errors converge in finite time on a small region that contains zero with the prescribed performance. In the simulation comparison, the controller proposed in this paper had better dynamic performance, steady-state performance and chattering suppression.

The remainder of this paper is ordered as follows. In Section 2, the formulation of the target tracking problem for underwater robots is stated. In Section 3, the design of the neural network nonsingular terminal sliding mode controller with prescribed performance and its stability analysis are presented. Section 4 reports simulation experiments and performance comparisons. Finally, Section 5 draws the conclusions and proposes future research.

2. Problem

2.1. Underwater Robot Model

As shown in Figure 1, the motion of underwater robots is commonly described with the use of two coordinate systems. The first is the geotectonic inertial reference coordinate system {I}: the axes of the coordinate system are fixed to the earth and the origin is selected somewhere on the ground. The other is the body reference coordinate system {B}, whose origin coincides with the buoyancy center of the underwater robot. The robot is self-stable under roll, so the motion model does not need to consider this degree of freedom. The kinematic model of the underwater robot is described by the following equation:

$$\begin{bmatrix} \dot{\xi} \\ \dot{\eta} \\ \dot{\zeta} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi & \sin\theta\cos\psi & 0 & 0 \\ \sin\psi\cos\theta & \cos\psi & \sin\theta\sin\psi & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{\cos\theta} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ q \\ r \end{bmatrix} \tag{1}$$

where ξ, η and ζ indicate the position in {I}; θ represents the pitch angle in {I}; and ψ represents the yaw angle in {I}. u, v, w, q and r indicate the surge, sway, heave, pitch and yaw velocities in {B}.

By applying the Newton–Euler formulation, the dynamics of the underwater robot in {B} can be described as [38]:

$$M\dot{v} + C(v)v + D(v)v + G(\eta) \ = \ \tau_{ex} + \tau_v \tag{2}$$

where $M \in R^{5\times5}$ is the inertia matrix that includes body mass and added mass; $C \in R^{5\times5}$ is a matrix describing centripetal and Coriolis forces produced due to added inertia; $D \in R^{5\times5}$ represents the hydrodynamic damping matrix; $G(\eta) \in R^5$ represents the combined gravitational and buoyancy forces vector; $\tau_{ex} \in R^5$ is the unknown time-varying vector of external disturbance forces such as waves and ocean current; and $\tau_v \in R^5$ is s the vector of control moments and forces.

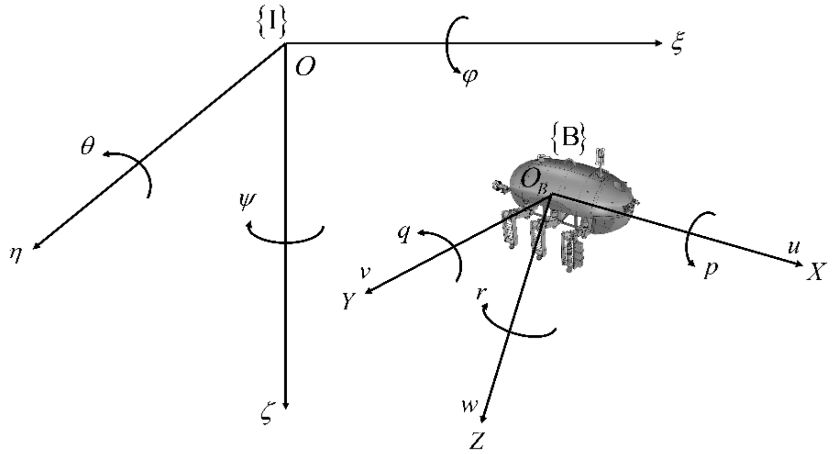


Figure 1. Inertial reference coordinate system {I} and body reference coordinate system {B}.

Due to the complexity and variability of the actual marine environment, it is difficult to ensure the accuracy of the hydrodynamic parameters obtained from an experiment, so the parameters are still uncertain. Therefore, M, $C(v)$ and $D(v)$ can be divided into two parts: the nominal dynamics M_0, $C_0(v)$ and $D_0(v)$; and dynamic uncertainties ΔM, ΔC and ΔD:

$$\begin{aligned} M &= M_0 + \Delta M \\ C(v) &= C_0(v) + \Delta C(v) \\ D(v) &= D_0(v) + \Delta D(v) \end{aligned} \tag{3}$$

Then, the dynamic equation can be rewritten as the following form:

$$M_0\dot{v} + C_0(v)v + D_0(v)v + G(\eta) + \Delta = \tau_{ex} + \tau_v \tag{4}$$

where $\Delta = \Delta M + \Delta C + \Delta D = \left[\Delta_u, \Delta_v, \Delta_w, \Delta_q, \Delta_r\right]^T$ is the total uncertainty of the dynamics.

Finally, for convenience of controller design, the kinematics model and dynamics model of an underwater robot are presented in the standard form of a nonlinear system in Equation (5):

$$\begin{cases} \dot{x}_1 = Ax_2 \\ \dot{x}_2 = f(x,t) + g(t)u(t) + \Delta'(x,t) + d(t) \end{cases} \tag{5}$$

where $x = [x_1, x_2]^T$ is the state variable of the system, and its expression is:

$$\begin{aligned} x_1 &= [\xi, \eta, \zeta, \theta, \psi]^T \\ x_2 &= [u, v, w, q, r]^T \end{aligned} \tag{6}$$

The controller input $u(t)$ is expressed as:

$$u(t) = \tau_v = \left[T_u, T_v, T_w, T_q, T_r\right]^T \tag{7}$$

The limits of control moments and forces are −6000 to 6000 N(Nm). Since there is no thruster actuator in the Y direction, $T_v \equiv 0.f(x,t)$ is a smooth nonlinear term, and it can be expressed according to Equation (2):

$$f(x,t) = -M_0^{-1}[C_0(x_2)x_2 + D_0(x_2)x_2 + G(x_1)] = \left[f_u, f_v, f_w, f_q, f_r\right]^T \tag{8}$$

The invertible matrix $g(t)$ is the control gain function. $\Delta'(x,t)$ is the model uncertainty function; $d(t)$ is the external bounded disturbance function:

$$\begin{aligned} g(t) &= M_0^{-1} = diag\{g_u, g_v, g_w, g_q, g_r\} \\ \Delta'(x,t) &= -M_0^{-1}\Delta = \left[-g_u\Delta_u, -g_v\Delta_v, -g_w\Delta_w, -g_q\Delta_q, -g_r\Delta_r\right]^T \\ d(t) &= M_0^{-1}\tau_{ex} = \left[{\tau'}_{ex1}, {\tau'}_{ex2}, {\tau'}_{ex3}, {\tau'}_{ex4}, {\tau'}_{ex5}\right]^T \end{aligned} \tag{9}$$

2.2. Control Objectives

The position error between the underwater robot and the tracking target can be formulated as follows:

$$\begin{bmatrix} x_e \\ y_e \\ z_e \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & \sin\psi\cos\theta & -\sin\theta \\ -\sin\psi & \cos\psi & 0 \\ \sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix} \begin{bmatrix} \xi_d - \xi_B \\ \eta_d - \eta_B \\ \zeta_d - \zeta_B \end{bmatrix} \tag{10}$$

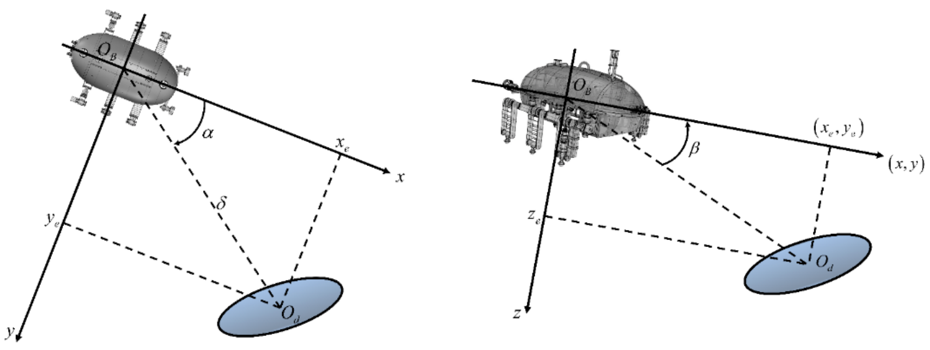
In Equation (10), (ξ_B, η_B, ζ_B) and (ξ_d, η_d, ζ_d) are the coordinates of the underwater robot and the tracking target in {I} respectively. The transformation relationship between the range and bearing angles $(\delta, \beta, \alpha, Z_e)$ and (x_e, y_e, z_e) is as follows:

$$\begin{cases} \delta &= \sqrt{{x_e}^2 + {y_e}^2} \\ \beta &= -\arctan(z_e / \sqrt{{x_e}^2 + {y_e}^2}) \\ \alpha &= \arctan(y_e / x_e) \\ Z_e &= z_e \end{cases} \tag{11}$$

and

$$\begin{cases} x_e &= \delta\cos\alpha \\ y_e &= \delta\sin\alpha \\ z_e &= Z_e \end{cases} \tag{12}$$

The range and bearing angles $(\delta, \beta, \alpha, Z_e)$ are considered to be four tracking errors. Figure 2a is the horizontal projection of the underwater robot and the target from the top view perspective in {B}. Figure 2b is the vertical plane projection of the underwater robot with the z axis coplanar with the target in {B}. It can be seen that δ is the distance from the projection of the target on the xO_By plane to the O_B point; α is the angle between the line projected by the target and the underwater robot on the xO_By plane and the x axis; Z_e is the distance from the xO_By plane to the target in {B}; and β is the angle between the line of the underwater robot to the target and the xO_By plane, as shown in Figure 2.



(**a**) (**b**)

Figure 2. The range and bearing angles $(\delta, \beta, \alpha, Z_e)$: (**a**) the horizontal projection of an underwater robot and tracking target; (**b**) the vertical plane projection of an underwater robot and tracking target.

Derivation of Equation (12) yields Equation (13):

$$\begin{cases} \dot{x}_e &= -u - qz_e + ry_e + \chi_{1d} \\ \dot{y}_e &= -v - r(x_e + z_e \tan\theta) + \chi_{2d} \\ \dot{z}_e &= -w + qx_e + ry_e \tan\theta + \chi_{3d} \end{cases} \tag{13}$$

where

$$\begin{cases} \chi_{1d} &= \dot{\xi}_d \cos\theta \cos\psi + \dot{\eta}_d \cos\theta \sin\psi - \dot{\zeta}_d \sin\theta \\ \chi_{2d} &= -\dot{\xi}_d \sin\psi + \dot{\eta}_d \cos\psi \\ \chi_{3d} &= \dot{\xi}_d \sin\theta \cos\psi + \dot{\eta}_d \sin\theta \sin\psi + \dot{\zeta}_d \cos\theta \end{cases} \tag{14}$$

By combining Equations (11)–(13), the dynamic error equation related to the $(\delta, \beta, \alpha, Z_e)$ can be obtained as follows:

$$\begin{cases} \dot{\delta} &= -u\cos\alpha - v\sin\alpha - qZ_e\cos\alpha - rZ_e\tan\theta\sin\alpha + \chi_{1d}\cos\alpha + \chi_{2d}\sin\alpha \\ \dot{\beta} &= -\left[x_e\delta/\left(\delta^2 + Z_e{}^2\right)\right]q + Z_e\dot{\delta}/\left(\delta^2 + Z_e{}^2\right) - \left[(-w + ry_e\tan\theta + \chi_{3d})/\left(\delta^2 + Z_e{}^2\right)\right]\delta \\ \dot{\alpha} &= -[(Z_e\tan\theta\cos\alpha + \delta)/\delta]r + (u\sin\alpha - v\cos\alpha + qZ_e\sin\alpha - \chi_{1d}\sin\alpha + \chi_{2d}\cos\alpha)/\delta \\ \dot{Z}_e &= -w + qx_e + ry_e\tan\theta + \chi_{3d} \end{cases} \tag{15}$$

The control objectives of this paper include the following. By designing the controller input $u(t)$, the target tracking errors $(\delta, \beta, \alpha, Z_e)$ stabilize near the origin in finite time with the pre-set dynamic responses and meet the steady-state error limit. To avoid collisions, a small positive threshold $\widetilde{\delta}$ is introduced as the safe distance between the underwater robot and the target. Therefore, δ should converge to $\widetilde{\delta}$, and the others should converge to zero. Moreover, the proposed controller is expected to be robust to the uncertainties of the dynamic model and unknown external disturbances, and can attenuate the chattering.

3. Controller Design

3.1. Prescribed Performance and Error Transformation

The prescribed performance is defined when the error can converge to a pre-assigned residual set; and the maximum overshoot, convergence rate and steady-state error in the convergent response strictly satisfy a bounded decreasing time function defined as the performance function [39]. The following formula is the mathematical expression of the prescribed performance.

$$L_1\eta_1 < \delta < H_1\eta_1 \tag{16}$$

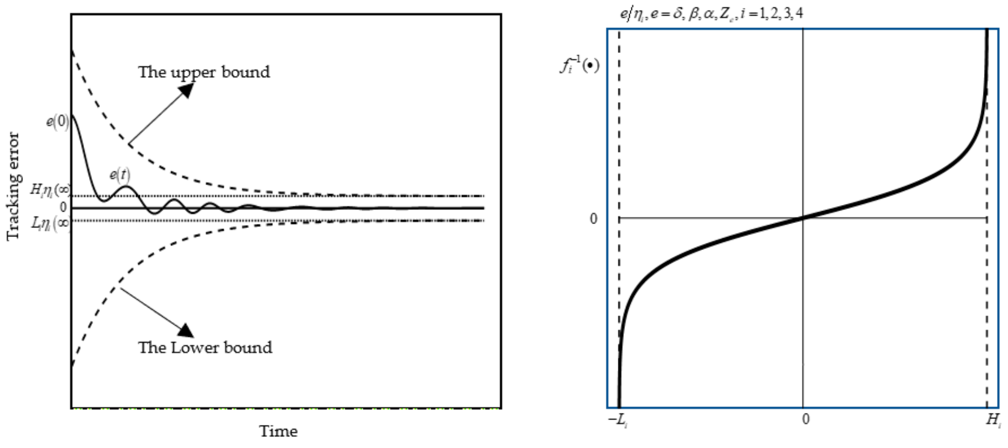
$$-L_2\eta_2 < \beta < H_2\eta_2 \tag{17}$$

$$-L_3\eta_3 < \alpha < H_3\eta_3 \tag{18}$$

$$-L_4\eta_4 < Z_e < H_4\eta_4 \tag{19}$$

where, $H_1, L_1, H_2, L_2, H_3, L_3, H_4$ and L_4 are positive real numbers, and $H_1 > L_1$ should be guaranteed. $\eta_i, i = 1,2,3,4$ is called the performance function, which could govern the dynamic and steady state performances of the errors. It is defined as $\eta_i = (\eta_{i0} - \eta_{i\infty})e^{-a_i t} + \eta_{i\infty}, i = 1,2,3,4$ and $\eta_{i0} > \eta_{i\infty} > 0$, $a_i > 0$, where $\eta_{i\infty}$ should be sufficiently small positive real numbers. $\eta_{i\infty}$ determines the final boundaries of the tracking error. a_i determines the convergence rate of the tracking errors, which is mainly related to the dynamic performance of the system. The prescribed performance and error transformation function are shown in Figure 3.

It can be seen in Equation (11) that when $\delta = 0, \beta = \pm\pi/2$, a singularity will occur in the system. If $\alpha = \pm\pi/2$, since there is no actuator for the lateral direction of the underwater robot, the system will lose control in this direction. Therefore, the above situations should be avoided. Since β, α and Z_e represent the vertical tracking error, horizontal orientation tracking error and vertical distance between the underwater robot and the target, respectively, they shall converge to zero as $t \to \infty$. If β, α, Ze satisfy the constraints of Equations (17)–(19), the possible singularity can be avoided and the system can always be under control. Further, when the bound of Equation (16) is strictly satisfied, the horizontal distance error δ converges to the minimum safe distance such that $\delta(\infty) = \widetilde{\delta} \neq 0$ as $t \to \infty$. This definition can not only eliminate the potential singularity of the system, but also maintain a safe distance between the robot and the tracking target.



(**a**) (**b**)

Figure 3. The prescribed performance and error transformation function: (**a**) a graphical illustration for the prescribed performance; (**b**) error transformation function f_i.

It is difficult to deal with inequality constraints Equations (16)–(19) directly. Therefore, inequality constraints are firstly transformed into equality constraints, and four smooth and strictly increasing functions are defined as error transformation functions $f_1(\varepsilon_\delta), f_2(\varepsilon_\beta), f_3(\varepsilon_\alpha), f_4(\varepsilon_{Z_e})$ with the following properties:

$$\delta_{(t)} = \eta_1 f_1(\varepsilon_\delta),\ \beta_{(t)} = \eta_2 f_2(\varepsilon_\beta),\ \alpha_{(t)} = \eta_3 f_3(\varepsilon_\alpha),\ Z_{e(t)} = \eta_4 f_4(\varepsilon_{Z_e}), \tag{20}$$

$$L_1 < f_1(\varepsilon_\delta) < H_1,\ -L_2 < f_2\left(\varepsilon_\beta\right) < H_2,\ -L_3 < f_3(\varepsilon_\alpha) < H_3,\ -L_4 < f_4(\varepsilon_{Z_e}) < H_4, \tag{21}$$

$$\lim_{\varepsilon_\delta \to +\infty} f_1(\varepsilon_\delta) = H_1,\ \lim_{\varepsilon_\beta \to +\infty} f_2\left(\varepsilon_\beta\right) = H_2,\ \lim_{\varepsilon_\alpha \to +\infty} f_3(\varepsilon_\alpha) = H_3,\ \lim_{\varepsilon_{Z_e} \to +\infty} f_4(\varepsilon_\delta) = H_4, \tag{22}$$

$$\lim_{\varepsilon_\delta \to -\infty} f_1(\varepsilon_\delta) = L_1,\ \lim_{\varepsilon_\beta \to -\infty} f_2\left(\varepsilon_\beta\right) = -L_2,\ \lim_{\varepsilon_\alpha \to -\infty} f_3(\varepsilon_\alpha) = -L_3,\ \lim_{\varepsilon_{Z_e} \to -\infty} f_4(\varepsilon_\delta) = -L_4, \tag{23}$$

$$f_1(0) = L_1 + \varepsilon_1,\ f_2(0) = 0,\ f_3(0) = 0,\ f_4(0) = 0, \tag{24}$$

$$\lim_{\varepsilon_\delta \to 0} \delta = \eta_1(L_1 + \varepsilon_1),\ \lim_{\varepsilon_\beta \to 0} \beta = 0,\ \lim_{\varepsilon_\alpha \to 0} \alpha = 0,\ \lim_{\varepsilon_{Z_e} \to 0} Z_e = 0, \tag{25}$$

where $\varepsilon_\delta, \varepsilon_\beta, \varepsilon_\alpha$ and ε_{Z_e} are defined as the transformation errors corresponding to tracking errors $\delta, \beta, \alpha, Z_e$ respectively. When the transformation errors converge to zero, the tracking errors converge with prescribed dynamic and steady-state performances. β, α, Z_e converge to zero, and δ converges to the minimum safe distance $\widetilde{\delta} = \lim\limits_{\substack{\varepsilon_\delta \to 0 \\ t \to \infty}} \delta = \eta_{1\infty}(L_1 + \varepsilon_1)$.

According to the above properties, $f_1(\varepsilon_\delta), f_2(\varepsilon_\beta), f_3(\varepsilon_\alpha), f_4(\varepsilon_{Z_e})$ is designed as follows:

$$\begin{cases} f_1(\varepsilon_\delta) = \frac{H_1 e^{(p_1\varepsilon_\delta + v_1)} + L_1 e^{-(p_1\varepsilon_\delta + v_1)}}{e^{(p_1\varepsilon_\delta + v_1)} + e^{-(p_1\varepsilon_\delta + v_1)}} \\ f_2(\varepsilon_\beta) = \frac{H_2 e^{(p_2\varepsilon_\beta + v_2)} - L_2 e^{-(p_2\varepsilon_\beta + v_2)}}{e^{(p_2\varepsilon_\beta + v_2)} + e^{-(p_2\varepsilon_\beta + v_2)}} \\ f_3(\varepsilon_\alpha) = \frac{H_3 e^{(p_3\varepsilon_\alpha + v_3)} - L_3 e^{-(p_3\varepsilon_\alpha + v_3)}}{e^{(p_3\varepsilon_\alpha + v_3)} + e^{-(p_3\varepsilon_\alpha + v_3)}} \\ f_4(\varepsilon_{Z_e}) = \frac{H_4 e^{(p_4\varepsilon_{Z_e} + v_4)} - L_4 e^{-(p_4\varepsilon_{Z_e} + v_4)}}{e^{(p_4\varepsilon_{Z_e} + v_4)} + e^{-(p_4\varepsilon_{Z_e} + v_4)}} \end{cases} \tag{26}$$

where $v_1 = 0.5\ln(\varepsilon_1/(H_1 - L_1 - \varepsilon_1))$, $v_2 = 0.5\ln(H_2/L_2)$, $v_3 = 0.5\ln(H_3/L_3)$, $v_4 = 0.5\ln(H_4/L_4)$ and $\varepsilon_1 > 0, p_i > 0, i = 1,2,3,4$.

The tracking errors should initially satisfy $L_1\eta_{10} < \delta(0) < H_1\eta_{10}$, $-L_2\eta_{20} < \beta(0) < H_2\eta_{20}$, $-L_3\eta_{30} < \alpha(0) < H_3\eta_{30}$ and $-L_4\eta_{40} < Z_e(0) < H_4\eta_{40}$; that is, H_1,H_2,H_3,H_4,L_2,L_3,L_4 should be sufficiently large positive numbers, L_1 should be a small enough positive number and $\varepsilon_\delta, \varepsilon_\beta, \varepsilon_\alpha, \varepsilon_{Z_e} \in L_\infty$. Thus, the tracking errors of the system can be guaranteed to meet the prescribed performance. All the signals of the closed-loop system are bounded and non-singular, and the tracking errors converge with the specified dynamic and steady state performance (i.e., maximum overshoot, convergence rate and final accuracy). Finally, the tracking errors would be stabilized in the pre-set boundary: $L_1\eta_{1\infty} < \delta(\infty) < H_1\eta_{1\infty}$, $-L_2\eta_{2\infty} < \beta(\infty) < H_2\eta_{2\infty}$, $-L_3\eta_{3\infty} < \alpha(\infty) < H_3\eta_{3\infty}$ and $-L_4\eta_{4\infty} < Z_e(\infty) < H_4\eta_{4\infty}$. The accuracy of the tracking errors can be improved by appropriately selecting parameters. Since the error transformation functions $f_1(\varepsilon_\delta), f_2(\varepsilon_\beta), f_3(\varepsilon_\alpha), f_4(\varepsilon_{Z_e})$ are strictly increasing and the performance function $\eta_i \neq 0, i = 1,2,3,4$, the transformation errors can be obtained through the inverse transformation:

$$\left\{\begin{array}{l} \varepsilon_\delta = f_1^{-1}\left(\delta_{(t)}/\eta_1\right) = 0.5{p_1}^{-1}\ln(\delta/\eta_1 - L_1) + 0.5{p_1}^{-1}\ln(H_1 - L_1 - \varepsilon_1) - 0.5{p_1}^{-1}\ln(H_1\varepsilon_1 - (\delta/\eta_1)\varepsilon_1) \\ \varepsilon_\beta = f_2^{-1}\left(\beta_{(t)}/\eta_2\right) = 0.5{p_2}^{-1}\ln((\beta/\eta_2)H_2 + L_2H_2) - 0.5{p_2}^{-1}\ln(L_2H_2 - (\beta/\eta_2)L_2) \\ \varepsilon_\alpha = f_3^{-1}\left(\alpha_{(t)}/\eta_3\right) = 0.5{p_3}^{-1}\ln((\alpha/\eta_3)H_3 + L_3H_3) - 0.5{p_3}^{-1}\ln(L_3H_3 - (\alpha/\eta_3)L_3) \\ \varepsilon_{Ze} = f_4^{-1}\left(Ze_{(t)}/\eta_4\right) = 0.5{p_4}^{-1}\ln((Ze/\eta_4)H_4 + L_4H_4) - 0.5{p_4}^{-1}\ln(L_4H_4 - (Ze/\eta_4)L_4) \end{array}\right. \tag{27}$$

By continuing to derive Equation (28),

$$\left\{\begin{array}{l} \dot{\varepsilon}_\delta = \zeta_1\left(\dot{\delta} - \delta\dot{\eta}_1/\eta_1\right) \\ \dot{\varepsilon}_\beta = \zeta_2\left(\dot{\beta} - \beta\dot{\eta}_2/\eta_2\right) \\ \dot{\varepsilon}_\alpha = \zeta_3(\dot{\alpha} - \alpha\dot{\eta}_3/\eta_3) \\ \dot{\varepsilon}_{Z_e} = \zeta_4\left(\dot{Z}_e - Z_e\dot{\eta}_4/\eta_4\right) \end{array}\right. \tag{28}$$

where $\left\{\begin{array}{l} \zeta_1 = [1/(\delta/\eta_1 - L_1) - 1/(\delta/\eta_1 - H_1)]/2p_1\eta_1 \\ \zeta_2 = [1/(\beta/\eta_2 + L_2) - 1/(\beta/\eta_2 - H_2)]/2p_2\eta_2 \\ \zeta_3 = [1/(\alpha/\eta_3 + L_3) - 1/(\alpha/\eta_3 - H_3)]/2p_3\eta_3 \\ \zeta_4 = [1/(Z_e/\eta_4 + L_4) - 1/(Z_e/\eta_4 - H_4)]/2p_4\eta_4 \end{array}\right.$.

By substituting Equation (15) into Equation (28), we can get:

$$\left\{\begin{array}{l} \dot{\varepsilon}_\delta = -\zeta_1 u\cos\alpha + Y_1 \\ \dot{\varepsilon}_\beta = -\zeta_2\left[x_e\delta/\left(\delta^2 + {z_e}^2\right)\right]q + Y_2 \\ \dot{\varepsilon}_\alpha = (-\zeta_3 z_e\tan\theta\cos\alpha/\delta - \zeta_3)r + Y_3 \\ \dot{\varepsilon}_{Z_e} = -\zeta_4 w + Y_4 \end{array}\right. \tag{29}$$

where $\left\{\begin{array}{l} Y_1 = -\zeta_1 q z_e\cos\alpha + \zeta_1\chi_{1d}\cos\alpha - \zeta_1 v\sin\alpha - \zeta_1 r z_e\tan\theta\sin\alpha + \zeta_1\chi_{2d}\sin\alpha - \zeta_1\delta\dot{\eta}_1/\eta_1 \\ Y_2 = \zeta_2 z_e\dot{\delta}/\left(\delta^2 + {z_e}^2\right) - \zeta_2\left[(-w + r y_e\tan\theta + \chi_{3d})/\left(\delta^2 + {z_e}^2\right)\right]\delta - \zeta_2\beta\dot{\eta}_2/\eta_2 \\ Y_3 = \zeta_3(-v\cos\alpha + u\sin\alpha + q z_e\sin\alpha + \chi_{2d}\cos\alpha - \chi_{1d}\sin\alpha)/\delta - \zeta_3\alpha\dot{\eta}_3/\eta_3 \\ Y_4 = \zeta_4(q x_e + r y_e\tan\theta + \chi_{3d}) - \zeta_4 z_e\dot{\eta}_4/\eta_4 \end{array}\right.$.

3.2. Dynamic Controller Design

In this section, a neural network nonsingular terminal sliding mode controller is proposed under bounded external disturbances and modeling uncertainties as shown in Figure 4.

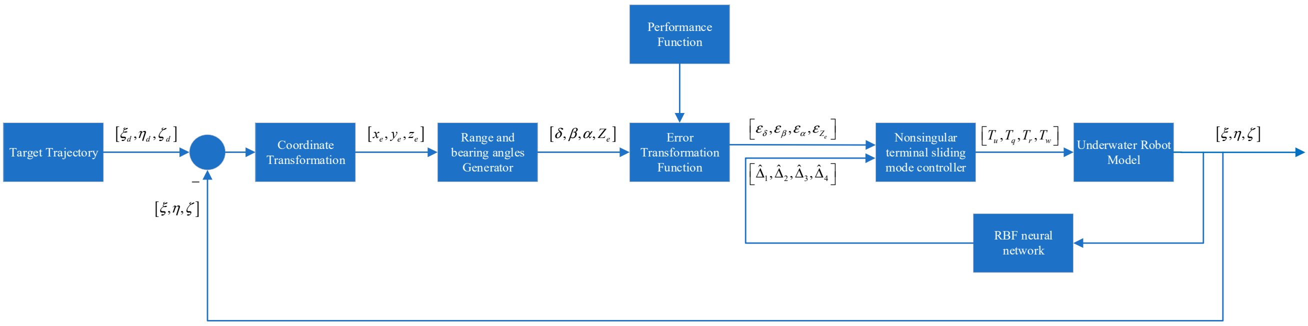

Performance Function
Target Trajectory
$[\xi_d, \eta_d, \zeta_d]$
$-$
$[\xi, \eta, \zeta]$
Coordinate Transformation
$[x_e, y_e, z_e]$
Range and bearing angles Generator
$[\delta, \beta, \alpha, Z_e]$
Error Transformation Function
$[\varepsilon_\delta, \varepsilon_\beta, \varepsilon_\alpha, \varepsilon_{Z_e}]$
$[\hat{\Delta}_1, \hat{\Delta}_2, \hat{\Delta}_3, \hat{\Delta}_4]$
Nonsingular terminal sliding mode controller
$[T_u, T_q, T_r, T_w]$
Underwater Robot Model
$[\xi, \eta, \zeta]$
RBF neural network

Figure 4. The block diagram of the proposed controller.

By continuing to take the second derivative of the transformation errors $\varepsilon_\delta, \varepsilon_\beta, \varepsilon_\alpha, \varepsilon_{Z_e}$ with respect to time, Equation (30) can be obtained:

$$\begin{aligned}
\begin{bmatrix} \ddot{\varepsilon}_\delta \\ \ddot{\varepsilon}_\beta \\ \ddot{\varepsilon}_\alpha \\ \ddot{\varepsilon}_{Z_e} \end{bmatrix} = {} & \begin{bmatrix} -\dot{\zeta}_1 u \cos\alpha + \zeta_1 u \sin\alpha\dot{\alpha} \\ \left\{ \frac{-\dot{\zeta}_2 x_e \delta}{\delta^2 + z_e{}^2} - \zeta_2 \left[\frac{\left(\dot{x}_e\delta + x_e\dot{\delta}\right)\left(\delta^2 + z_e{}^2\right) - x_e\delta\left(2\delta\dot{\delta} + 2z_e\dot{z}_e\right)}{\left(\delta^2 + z_e{}^2\right)^2} \right] \right\} q \\ \left\{ -\left[\frac{\left(\dot{\zeta}_3 z_e \tan\theta\cos\alpha + \zeta_3 \dot{z}_e \tan\theta\cos\alpha + \zeta_3 z_e \dot{\theta}\cos\alpha/\cos^2\theta - \zeta_3 z_e \tan\theta\sin\alpha\dot{\alpha}\right)\delta - \zeta_3 z_e \tan\theta\cos\alpha\dot{\delta}}{\delta^2} \right] - \dot{\zeta}_3 \right\} r \\ -\dot{\zeta}_4 w \end{bmatrix} \\
& + \begin{bmatrix} -\zeta_1 \cos\alpha & 0 & 0 & 0 \\ 0 & -\frac{\zeta_2 x_e \delta}{\delta^2 + z_e{}^2} & 0 & 0 \\ 0 & 0 & \left(-\frac{\zeta_3 z_e \tan\theta\cos\alpha}{\delta} - \zeta_3 \right) & 0 \\ 0 & 0 & 0 & -\zeta_4 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{r} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \\ \dot{Y}_3 \\ \dot{Y}_4 \end{bmatrix} \\
= {} & \begin{bmatrix} f_\delta \\ f_\beta \\ f_\alpha \\ f_{Z_e} \end{bmatrix} + \begin{bmatrix} g_\delta & 0 & 0 & 0 \\ 0 & g_\beta & 0 & 0 \\ 0 & 0 & g_\alpha & 0 \\ 0 & 0 & 0 & g_{Z_e} \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{q} \\ \dot{r} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} \dot{Y}_1 \\ \dot{Y}_2 \\ \dot{Y}_3 \\ \dot{Y}_4 \end{bmatrix}
\end{aligned} \tag{30}$$

Substituting Equation (5) into Equation (30) yields

$$\begin{cases} \ddot{\varepsilon}_\delta = g_\delta g_u T_u + g_\delta f_u - g_\delta g_u \Delta_u + g_\delta \tau'_{ex1} + f_\delta + \dot{Y}_1 \\ \ddot{\varepsilon}_\beta = g_\beta g_q T_q + g_\beta f_q - g_\beta g_q \Delta_q + g_\beta \tau'_{ex4} + f_\beta + \dot{Y}_2 \\ \ddot{\varepsilon}_\alpha = g_\alpha g_r T_r + g_\alpha f_r - g_\alpha g_r \Delta_r + g_\alpha \tau'_{ex5} + f_\alpha + \dot{Y}_3 \\ \ddot{\varepsilon}_{Z_e} = g_{Z_e} g_w T_w + g_{Z_e} f_w - g_{Z_e} g_w \Delta_w + g_{Z_e} \tau'_{ex3} + f_{Z_e} + \dot{Y}_4 \end{cases} \tag{31}$$

Choose the following nonsingular terminal sliding mode surface:

$$\begin{cases} S_1 = \varepsilon_\delta + \frac{1}{k_\delta} \dot{\varepsilon}_\delta{}^{\frac{p_\delta}{q_\delta}} \\ S_2 = \varepsilon_\beta + \frac{1}{k_\beta} \dot{\varepsilon}_\beta{}^{\frac{p_\beta}{q_\beta}} \\ S_3 = \varepsilon_\alpha + \frac{1}{k_\alpha} \dot{\varepsilon}_\alpha{}^{\frac{p_\alpha}{q_\alpha}} \\ S_4 = \varepsilon_{Z_e} + \frac{1}{k_{Z_e}} \dot{\varepsilon}_{Z_e}{}^{\frac{p_{Z_e}}{q_{Z_e}}} \end{cases} \tag{32}$$

where $k_i > 0$, p_i, q_i is positive and odd and $1 < \frac{p_i}{q_i} < 2, i = \delta, \beta, \alpha, Z_e$.

The nonsingular terminal sliding mode controller is designed as follows:

$$\begin{cases} T_u = -g_\delta{}^{-1} g_u{}^{-1} \left(\frac{k_\delta q_\delta}{p_\delta} \dot{\varepsilon}_\delta{}^{(2 - \frac{p_\delta}{q_\delta})} + C_1 S_1 + \kappa_1 \mathrm{sgn}(S_1) + f_\delta + g_\delta f_u \right) \\ T_q = -g_\beta{}^{-1} g_q{}^{-1} \left(\frac{k_\beta q_\beta}{p_\beta} \dot{\varepsilon}_q{}^{(2 - \frac{p_\beta}{q_\beta})} + C_2 S_2 + \kappa_2 \mathrm{sgn}(S_2) + f_\beta + g_\beta f_q \right) \\ T_r = -g_\alpha{}^{-1} g_r{}^{-1} \left(\frac{k_\alpha q_\alpha}{p_\alpha} \dot{\varepsilon}_r{}^{(2 - \frac{p_\alpha}{q_\alpha})} + C_3 S_3 + \kappa_3 \mathrm{sgn}(S_3) + f_\alpha + g_\alpha f_r \right) \\ T_w = -g_{Z_e}{}^{-1} g_w{}^{-1} \left(\frac{k_{Z_e} q_{Z_e}}{p_{Z_e}} \dot{\varepsilon}_w{}^{(2 - \frac{p_{Z_e}}{q_{Z_e}})} + C_4 S_4 + \kappa_4 \mathrm{sgn}(S_4) + f_{Z_e} + g_{Z_e} f_w \right) \end{cases} \tag{33}$$

In Equation (33), $C_i > 0, i = 1,2,3,4$, $\kappa_1 > g_\delta \tau'_{ex1} + \dot{Y}_1 - g_\delta g_u \Delta_u$, $\kappa_2 > g_\beta \tau'_{ex4} + \dot{Y}_2 - g_\beta g_q \Delta_q$, $\kappa_3 > g_\alpha \tau'_{ex5} + \dot{Y}_3 - g_\alpha g_r \Delta_r$ and $\kappa_4 > g_{Z_e} \tau'_{ex3} + \dot{Y}_4 - g_{Z_e} g_w \Delta_w$.

Consider the following Lyapunov function:

$$V_1 = \frac{1}{2}\left({S_1}^2 + {S_2}^2 + {S_3}^2 + {S_4}^2\right) \tag{34}$$

Take the differential of Equation (34) and substitute Equation (33) into Equation (34) to obtain

$$\begin{aligned}
\dot{V}_1 = & -\frac{p_\delta \dot{\varepsilon}_\delta^{(\frac{p_\delta}{q_\delta}-1)}}{k_\delta q_\delta}\left[C_1 S_1 + \kappa_1 \mathrm{sgn}(S_1) - g_\delta \tau'_{ex1} - \dot{Y}_1 + g_\delta g_u \Delta_u\right] S_1 - \frac{p_\beta \dot{\varepsilon}_\beta^{(\frac{p_\beta}{q_\beta}-1)}}{k_\beta q_\beta}\left[C_2 S_2 + \kappa_2 \mathrm{sgn}(S_2) - g_\beta \tau'_{ex4} - \dot{Y}_2 + g_\beta g_q \Delta_q\right] S_2 \\
& -\frac{p_\alpha \dot{\varepsilon}_\alpha^{(\frac{p_\alpha}{q_\alpha}-1)}}{k_\alpha q_\alpha}\left[C_3 S_3 + \kappa_3 \mathrm{sgn}(S_3) - g_\alpha \tau'_{ex5} - \dot{Y}_3 + g_\alpha g_r \Delta_r\right] S_3 - \frac{p_{Z_e} \dot{\varepsilon}_{Z_e}^{(\frac{p_{Z_e}}{q_{Z_e}}-1)}}{k_{Z_e} q_{Z_e}}\left[C_4 S_4 + \kappa_4 \mathrm{sgn}(S_4) - g_{Z_e} \tau'_{ex3} - \dot{Y}_4 + g_{Z_e} g_w \Delta_w\right] S_4 \\
= & -\frac{p_\delta \dot{\varepsilon}_\delta^{\frac{p_\delta}{q_\delta}-1}}{k_\delta q_\delta}\left[C_1|S_1| + \kappa_1 - \left(g_\delta \tau'_{ex1} + \dot{Y}_1 - g_\delta g_u \Delta_u\right)\mathrm{sgn}(S_1)\right]|S_1| - \frac{p_\beta \dot{\varepsilon}_\beta^{\frac{p_\beta}{q_\beta}-1}}{k_\beta q_\beta}\left[C_2|S_2| + \kappa_2 - \left(g_\beta \tau'_{ex4} + \dot{Y}_2 - g_\beta g_q \Delta_q\right)\mathrm{sgn}(S_2)\right]|S_2| \\
& \frac{-p_\alpha \dot{\varepsilon}_\alpha^{\frac{p_\alpha}{q_\alpha}-1}}{k_\alpha q_\alpha}\left[C_3|S_3| + \kappa_3 - \left(g_\alpha \tau'_{ex5} + \dot{Y}_3 - g_\alpha g_r \Delta_r\right)\mathrm{sgn}(S_3)\right]|S_3| - \frac{p_{Z_e} \dot{\varepsilon}_{Z_e}^{\frac{p_{Z_e}}{q_{Z_e}}-1}}{k_{Z_e} q_{Z_e}}\left[C_4|S_4| + \kappa_4 - \left(g_{Z_e} \tau'_{ex3} + \dot{Y}_4 - g_{Z_e} g_w \Delta_w\right)\mathrm{sgn}(S_4)\right]|S_4| \\
= & -C_\delta|S_1| - C_\beta|S_2| - C_\alpha|S_3| - C_{Z_e}|S_4|
\end{aligned} \tag{35}$$

Obviously, $C_j > 0, j = \delta, \beta, \alpha, Z_e$. When $C_{\min} = \min\left(C_j\right)$, Equation (36) is obtained as follows:

$$\begin{aligned}
\dot{V}_1 & \leq -C_{\min}(|S_1| + |S_2| + |S_3| + |S_4|) \\
& \leq -\sqrt{2} C_{\min} V_1^{\frac{1}{2}}
\end{aligned} \tag{36}$$

According to Lyapunov finite time stability proof, the sliding mode variables S_1, S_2, S_3, S_4 will converge to zero in finite time. When $S_i = 0, i = 1,2,3,4$, the dynamic equation of the terminal sliding surface is $\dot{\varepsilon}_j = -k_j^{\frac{q_j}{p_j}} \varepsilon_j^{\frac{q_j}{p_j}} = h\left(\varepsilon_j\right), j = \delta, \beta, \alpha, Z_e$. According to the sufficient and necessary conditions for homogeneous global finite-time stability:

1. $\varepsilon_j h\left(\varepsilon_j\right) \in \mathbf{R}$ and $\varepsilon_j h\left(\varepsilon_j\right) \leq 0$, if and only if $\varepsilon_j = 0$, $\varepsilon_j h\left(\varepsilon_j\right) = 0$;
2. ε_{j0} is the initial value at $S_i = 0$, $\forall \varepsilon_{j0} \in \mathbf{R}$, $\int_{\varepsilon_{j0}}^{0} \frac{d\varepsilon_j}{h\left(\varepsilon_j\right)} = \left(\frac{k_j^{\frac{-q_j}{p_j}}}{1-\frac{q_j}{p_j}}\right) \varepsilon_{j0}^{1-\frac{q_j}{p_j}} < \infty$.

Therefore, the transformation error ε_j will converge to zero for finite time on the sliding surface. The tracking errors $\left(\delta - \widetilde{\delta}\right), \beta, \alpha, Z_e$ also converge to zero. By solving differential equation $\dot{\varepsilon}_j = h\left(\varepsilon_j\right)$, the convergence time of sliding mode is $t_c = \frac{p_j}{k_j^{\frac{q_j}{p_j}}\left(p_j - q_j\right)}\left|\varepsilon_{j0}\right|^{1-\frac{q_j}{p_j}}$.

This paper introduces an RBF neural network as shown in Figure 5 to approximate the total modeling uncertainties. The RBF neural network has three layers: an input layer, a hidden layer and an output layer. The neuron activation function of the hidden layer consists of a radial basis function. It has good generalization ability and simple structure. Meanwhile, it can avoid unnecessary and lengthy calculations and can approximate any nonlinear function with arbitrary precision in a compact set [40]. In the RBF neural network, the uncertainty terms are shown by Equation (37):

$$\Delta_j = \boldsymbol{W}_i^{*\mathrm{T}} \boldsymbol{h}(\boldsymbol{x}) + \varepsilon_i (i = 1,2,3,4; j = u, q, r, w) \tag{37}$$

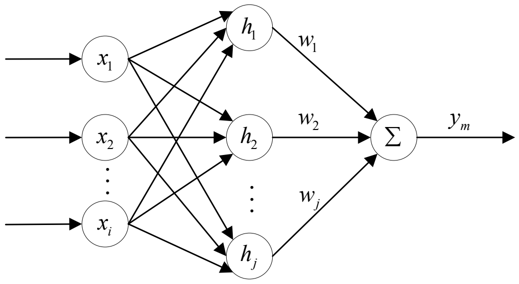


Figure 5. RBF neural network structure.

In Equation (37), $\boldsymbol{x}$ represents the input to the network, the hidden layer output of the network is $\boldsymbol{h}(\boldsymbol{x})$, $h_j(x) = \exp\left(-\frac{\|\boldsymbol{x}-\boldsymbol{c}_j\|^2}{2b_j^2}\right)$ is the output of the jth neuron in the hidden layer, $\boldsymbol{c}_j$ is the center point vector of the Gaussian basis function in the hidden layer, b_j is the width of the Gaussian basis function in the hidden layer, $\boldsymbol{W}_i{}^*$ is the ideal weight of the neural network, ε_i is the network approximation error and $|\varepsilon_i| \le \varepsilon_{Mi}$. The larger b_j is, the larger the non-zero output region of the radial basis function is, indicating a stronger mapping ability to the input. The closer the input is to the center $\boldsymbol{c}_j$, the larger its output value will be, indicating that the radial basis function is more sensitive to the input. The neural network input vector is set as $\boldsymbol{x} = [u, v, w, q, r]^{\mathrm{T}}$, so the output of the radial basis network can be written as follows:

$$\hat{\Delta}_i = \hat{\boldsymbol{W}}_i{}^T\boldsymbol{h}(\boldsymbol{x}) \tag{38}$$

where $\hat{\boldsymbol{W}}_i$ is the actual estimate of the ideal weight, which is adaptively updated online based on the following Lyapunov stability analysis. Then, Equation (33) can be rewritten as follows:

$$\begin{cases} T_u = -g_\delta{}^{-1}g_u{}^{-1}\left(\frac{k_\delta q_\delta}{p_\delta}\dot{\varepsilon}_\delta{}^{(2-\frac{p_\delta}{q_\delta})} + C_1S_1 + \kappa_1\mathrm{sgn}(S_1) + f_\delta + g_\delta f_u - g_\delta g_u\hat{\Delta}_u\right) \\ T_q = -g_\beta{}^{-1}g_q{}^{-1}\left(\frac{k_\beta q_\beta}{p_\beta}\dot{\varepsilon}_q{}^{(2-\frac{p_\beta}{q_\beta})} + C_2S_2 + \kappa_2\mathrm{sgn}(S_2) + f_\beta + g_\beta f_q - g_\beta g_q\hat{\Delta}_q\right) \\ T_r = -g_\alpha{}^{-1}g_r{}^{-1}\left(\frac{k_\alpha q_\alpha}{p_\alpha}\dot{\varepsilon}_r{}^{(2-\frac{p_\alpha}{q_\alpha})} + C_3S_3 + \kappa_3\mathrm{sgn}(S_3) + f_\alpha + g_\alpha f_r - g_\alpha g_r\hat{\Delta}_r\right) \\ T_w = -g_{Z_e}{}^{-1}g_w{}^{-1}\left(\frac{k_{Z_e}q_{Z_e}}{p_{Z_e}}\dot{\varepsilon}_w{}^{(2-\frac{p_{Z_e}}{q_{Z_e}})} + C_4S_4 + \kappa_4\mathrm{sgn}(S_4) + f_{Z_e} + g_{Z_e}f_w - g_{Z_e}g_w\hat{\Delta}_w\right) \end{cases} \tag{39}$$

Substituting Equation (39) into the Lyapunov function V_1 yields:

$$\begin{aligned} \dot{V}_1 = & -\frac{p_\delta}{k_\delta q_\delta}\dot{\varepsilon}_\delta{}^{(p_\delta/q_\delta-1)}\left[C_1S_1 + \kappa_1\mathrm{sgn}(S_1) - g_\delta\tau'_{ex1} - \dot{Y}_1 + g_\delta g_u\widetilde{\Delta}_u\right]S_1 \\ & -\frac{p_\beta}{k_\beta q_\beta}\dot{\varepsilon}_\beta{}^{(p_\beta/q_\beta-1)}\left[C_2S_2 + \kappa_2\mathrm{sgn}(S_2) - g_\beta\tau'_{ex4} - \dot{Y}_2 + g_\beta g_q\widetilde{\Delta}_q\right]S_2 \\ & -\frac{p_\alpha}{k_\alpha q_\alpha}\dot{\varepsilon}_\alpha{}^{(p_\alpha/q_\alpha-1)}\left[C_3S_3 + \kappa_3\mathrm{sgn}(S_3) - g_\alpha\tau'_{ex5} - \dot{Y}_3 + g_\alpha g_r\widetilde{\Delta}_r\right]S_3 \\ & -\frac{p_{Z_e}}{k_{Z_e}q_{Z_e}}\dot{\varepsilon}_{Z_e}{}^{(p_{Z_e}/q_{Z_e}-1)}\left[C_4S_4 + \kappa_4\mathrm{sgn}(S_4) - g_{Z_e}\tau'_{ex3} - \dot{Y}_4 + g_{Z_e}g_w\widetilde{\Delta}_w\right]S_4 \end{aligned} \tag{40}$$

where $\widetilde{\Delta}_i = \Delta_i - \hat{\Delta}_i = \boldsymbol{W_i}^{*\boldsymbol{T}}\boldsymbol{h}(\boldsymbol{x}) + \varepsilon_i - \hat{\boldsymbol{W}}_i{}^{\boldsymbol{T}}\boldsymbol{h}(\boldsymbol{x}) = \widetilde{\boldsymbol{W}}_i{}^{\boldsymbol{T}}\boldsymbol{h}(\boldsymbol{x}) + \varepsilon_i$, and $\widetilde{\boldsymbol{W}}_i = \hat{\boldsymbol{W}}_i{}^* - \hat{\boldsymbol{W}}_i$.

The Lyapunov function can be modified as:

$$V_2 = V_1 + \frac{1}{2}\sum_{i=1}^{4}\gamma_i\widetilde{\boldsymbol{W}}_i{}^T\widetilde{\boldsymbol{W}}_i \tag{41}$$

where $\gamma_i > 0$. Differentiating V_2 and substituting Equation (39) into Equation (41) yields

$$\begin{aligned} \dot{V}_2 = & -\frac{p_\delta}{k_\delta q_\delta}\dot{\varepsilon}_\delta{}^{(p_\delta/q_\delta-1)}\left[C_1S_1{}^2 + \kappa_1|S_1| - g_\delta\tau'_{ex1}S_1 - \dot{Y}_1S_1 - \varepsilon_1S_1\right] - \widetilde{\boldsymbol{W}}_1{}^{\boldsymbol{T}}\left(\frac{p_\delta}{k_\delta q_\delta}\dot{\varepsilon}_\delta{}^{(p_\delta/q_\delta-1)}g_\delta g_uS_1\boldsymbol{h}(\boldsymbol{x}) + \gamma_1\dot{\hat{\boldsymbol{W}}}_1\right) \\ & -\frac{p_\beta}{k_\beta q_\beta}\dot{\varepsilon}_\beta{}^{(p_\beta/q_\beta-1)}\left[C_2S_2{}^2 + \kappa_2|S_2| - g_\beta\tau'_{ex4}S_2 - \dot{Y}_2S_2 - \varepsilon_2S_2\right] - \widetilde{\boldsymbol{W}}_2{}^{\boldsymbol{T}}\left(\frac{p_\beta}{k_\beta q_\beta}\dot{\varepsilon}_\beta{}^{(p_\beta/q_\beta-1)}g_\beta g_qS_2\boldsymbol{h}(\boldsymbol{x}) + \gamma_2\dot{\hat{\boldsymbol{W}}}_2\right) \\ & -\frac{p_\alpha}{k_\alpha q_\alpha}\dot{\varepsilon}_\alpha{}^{(p_\alpha/q_\alpha-1)}\left[C_3S_3{}^2 + \kappa_3|S_3| - g_\alpha\tau'_{ex5}S_3 - \dot{Y}_3S_3 - \varepsilon_3S_3\right] - \widetilde{\boldsymbol{W}}_3{}^{\boldsymbol{T}}\left(\frac{p_\alpha}{k_\alpha q_\alpha}\dot{\varepsilon}_\alpha{}^{(p_\alpha/q_\alpha-1)}g_\alpha g_rS_3\boldsymbol{h}(\boldsymbol{x}) + \gamma_3\dot{\hat{\boldsymbol{W}}}_3\right) \\ & -\frac{p_{Z_e}}{k_{Z_e}q_{Z_e}}\dot{\varepsilon}_{Z_e}{}^{(p_{Z_e}/q_{Z_e}-1)}\left[C_4S_4{}^2 + \kappa_4|S_4| - g_{Z_e}\tau'_{ex3}S_4 - \dot{Y}_4S_4 - \varepsilon_4S_4\right] - \widetilde{\boldsymbol{W}}_4{}^{\boldsymbol{T}}\left(\frac{p_{Z_e}}{k_{Z_e}q_{Z_e}}\dot{\varepsilon}_{Z_e}{}^{(p_{Z_e}/q_{Z_e}-1)}g_{Z_e}g_wS_4\boldsymbol{h}(\boldsymbol{x}) + \gamma_4\dot{\hat{\boldsymbol{W}}}_4\right) \end{aligned} \tag{42}$$

Consider the following adaptive law:

$$\dot{\hat{\boldsymbol{W}}}_i = -\frac{1}{\gamma_i}\frac{p_j}{k_j q_j}\dot{\varepsilon}_j{}^{(p_j/q_j-1)}g_j g_k S_i \boldsymbol{h}(\boldsymbol{x})(i = 1,2,3,4; j = \delta,\beta,\alpha,Z_e; k = u,q,r,w) \tag{43}$$

Then

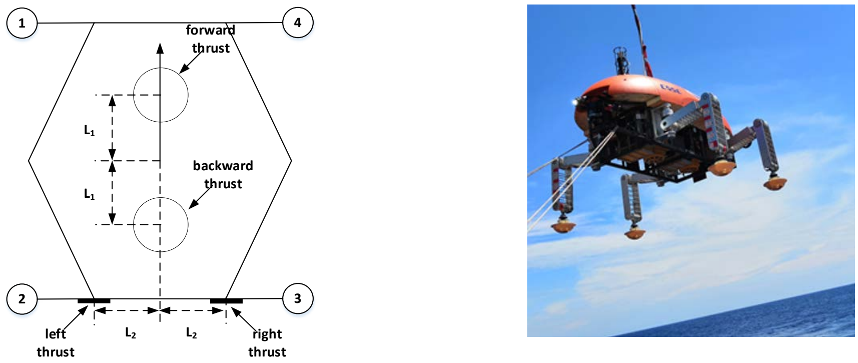
$$\begin{aligned}\dot{V}_2 \quad &= -\tfrac{p_\delta}{k_\delta q_\delta}\dot{\varepsilon}_\delta{}^{(p_\delta/q_\delta-1)}\left[C_1 {S_1}^2 + \kappa_1|S_1| - g_\delta \tau'_{ex1}S_1 - \dot{Y}_1 S_1 - \varepsilon_1 S_1\right] \\ &\quad -\tfrac{p_\beta}{k_\beta q_\beta}\dot{\varepsilon}_\beta{}^{(p_\beta/q_\beta-1)}\left[C_2 {S_2}^2 + \kappa_2|S_2| - g_\beta \tau'_{ex4}S_2 - \dot{Y}_2 S_2 - \varepsilon_2 S_2\right] \\ &\quad -\tfrac{p_\alpha}{k_\alpha q_\alpha}\dot{\varepsilon}_\alpha{}^{(p_\alpha/q_\alpha-1)}\left[C_3 {S_3}^2 + \kappa_3|S_3| - g_\alpha \tau'_{ex5}S_3 - \dot{Y}_3 S_3 - \varepsilon_3 S_3\right] \\ &\quad -\tfrac{p_{Z_e}}{k_{Z_e} q_{Z_e}}\dot{\varepsilon}_{Z_e}{}^{(p_{Z_e}/q_{Z_e}-1)}\left[C_4 {S_4}^2 + \kappa_4|S_4| - g_{Z_e} \tau'_{ex3}S_4 - \dot{Y}_4 S_4 - \varepsilon_4 S_4\right]\end{aligned} \tag{44}$$

Since ε_i can be limited to a small size and τ'_{exi} is bounded, when $\kappa_1 > g_\delta \tau'_{ex1} + \dot{Y}_1 + \varepsilon_1, \kappa_2 > g_\beta \tau'_{ex4} + \dot{Y}_2 + \varepsilon_2, \kappa_3 > g_\alpha \tau'_{ex5} + \dot{Y}_3 + \varepsilon_3, \kappa_4 > g_{Z_e} \tau'_{ex3} + \dot{Y}_4 + \varepsilon_4$, then $\dot{V}_2 < 0$. Thus, the adaptive update rate of $\hat{\boldsymbol{W}}_i$ is also given completely.

The above stability analysis proves that the tracking errors $\left(\delta - \widetilde{\delta}\right), \beta, \alpha, Z_e$ can converge to the neighborhood of the zero without any singularity in finite time with the prescribed performance, and the proposed controller can solve the problem of underwater robot target tracking with the external disturbances and modeling uncertainties.

4. Numerical Simulation Example

To verify the effectiveness and robustness of the target tracking controller proposed in this paper, a numerical simulation has been performed on the "Qilin" underwater robot using MATLAB/Simulink®. The "Qilin" underwater robot is a new prototype deep-sea work platform that can cruise in the deep sea and crawl on the bottom of the sea. The thruster layout and physical prototype of the robot are shown in Figure 6. Among them, $L_1 = L_2 = 1$ m; the thrust output of the four thrusters is −3000 to 3000 N. Thus, the limit of control moments and forces was −6000 to 6000 N(Nm) in simulation.



(**a**) (**b**)

Figure 6. The underwater robot: (**a**) the thruster layout; (**b**) the physical prototype.

For this simulation, the control parameters are given in Table 1. The initial position and orientation of the underwater robot were $(\xi_0, \eta_0, \zeta_0) = (1.25\text{ m}, 1.25\text{ m}, 100.1\text{ m})$ and $(\psi_0, \theta_0) = \left(\frac{\pi}{2}rad, 0rad\right)$; the initial velocity of the underwater robot was $(u_0, v_0, w_0) = (0\text{ m/s}, 0\text{ m/s},$ The initial position of the target was $(\xi_{d0}, \eta_{d0}, \zeta_{d0}) = (2\text{ m}, 2\text{ m}, 100\text{ m})$, and the desired trajectory was generated by following typical timing laws:

$$\left\{\begin{array}{lcl}\xi_d & = & 5\cos(\pi t/10) - 3 \\ \eta_d & = & 5\sin(\pi t/10) + 2 \\ \zeta_d & = & -0.05t + 100\end{array}\right. \tag{45}$$

Table 1. The control parameters.

Controller Function	Control Parameters
Non-singular terminal sliding mode function	$k_\delta = 0.01,\ k_\beta = 0.1,\ k_\alpha = 0.01,\ k_{Z_e} = 0.1,$ $p_\delta = 5,\ p_\beta = 5,\ p_\alpha = 5,\ p_{Z_e} = 5,$ $q_\delta = 3,\ q_\beta = 3,\ q_\alpha = 3,\ q_{Z_e} = 3,$ $C_1 = 1,\ C_2 = 10,\ C_3 = 10,\ C_4 = 10,$ $\kappa_1 = 0.5,\ \kappa_2 = 0.5,\ \kappa_3 = 0.005,\ \kappa_4 = 0.001.$
Prescribed performance function	$\eta_{10} = 1,\ \eta_{1\infty} = 0.2,\ \eta_{20} = 1,\ \eta_{2\infty} = 0.1,$ $\eta_{30} = 1,\ \eta_{3\infty} = 0.1,\ \eta_{40} = 1,\ \eta_{4\infty} = 0.1,$ $a_1 = 0.2,\ a_2 = 0.2,\ a_3 = 0.2,\ a_4 = 0.2.$
Error transformation function	$H_1 = 2,\ L_1 = 0.01,\ H_2 = 0.4,\ L_2 = 4,$ $H_3 = 1.2,\ L_3 = 1.2,\ H_4 = 0.2,\ L_4 = 0.2,$ $p_1 = 1,\ p_2 = 1,\ p_3 = 1,\ p_4 = 1,\ \varepsilon_1 = 0.5.$
RBF neural network function	$\boldsymbol{c}_1 = linspace(0, 1.8, 100), \boldsymbol{c}_2 = linspace(-0.5, 0.7, 100),$ $\boldsymbol{c}_3 = linspace(-0.1, 0.7, 100), \boldsymbol{c}_4 = linspace(-0.5, 1.6, 100),$ $b_1 = 0.5,\ b_2 = 1.5,\ b_3 = 0.4,\ b_4 = 1,$ $\gamma_1 = 0.7,\ \gamma_2 = 0.01,\ \gamma_3 = 0.8,\ \gamma_4 = 0.08.$

Equation (46) was introduced to simulate the bounded external disturbances.

$$\begin{cases} \tau_{ex1} = 0.25sign(u) + 0.5\sin(0.1t) \\ \tau_{ex2} = 0.25sign(v) + 0.5\sin(0.1t) \\ \tau_{ex3} = 0.25sign(w) + 0.5\sin(0.1t) \\ \tau_{ex4} = 0.25sign(q) + 0.5\sin(0.1t) \\ \tau_{ex5} = 0.25sign(r) + 0.5\sin(0.1t) \end{cases} \tag{46}$$

The uncertainties of robot were as follows in the simulation:

$$\Delta_i = 5i + 2.5|i|i + 1.5i^3,\ i = u, v, w, q, r \tag{47}$$

Equation (46) and Equation (47) refer to [10].

To better verify the performance of the proposed controller, the proposed controller (PNTSMC) is compared with the nonsingular terminal sliding mode controller (TSMC) and the nonsingular terminal sliding mode controller with prescribed performance (PTSMC). The initial conditions for these controllers were the same.

Simulation results are illustrated in Figures 7–11.

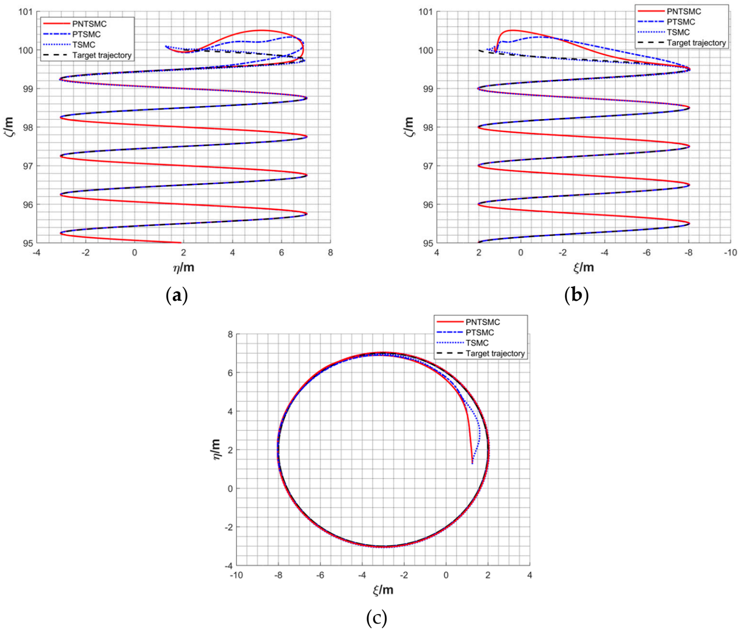
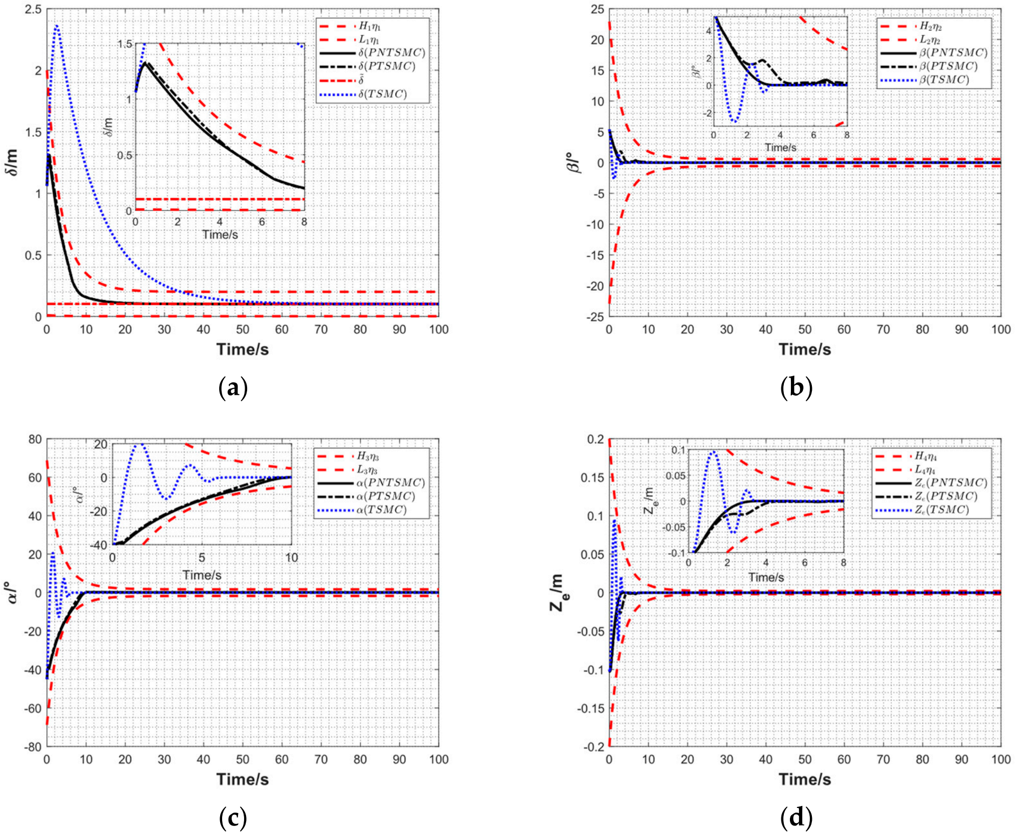


Figure 7. The underwater robot trajectory and target trajectory: (**a**) $\eta - \zeta$ plane; (**b**) $\xi - \zeta$ plane; (**c**) $\xi - \eta$ plane.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. The tracking errors with their performance bounds: (**a**) $\delta_{(t)}$; (**b**) $\beta_{(t)}$; (**c**) $\alpha_{(t)}$; (**d**) $Ze_{(t)}$.

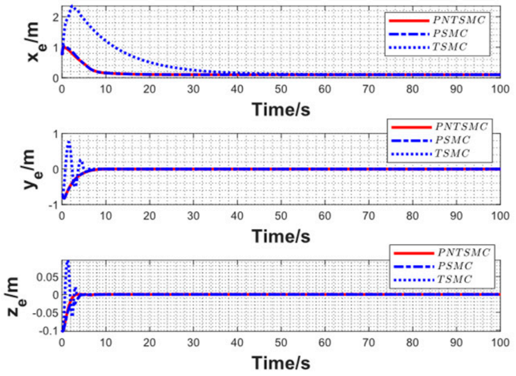
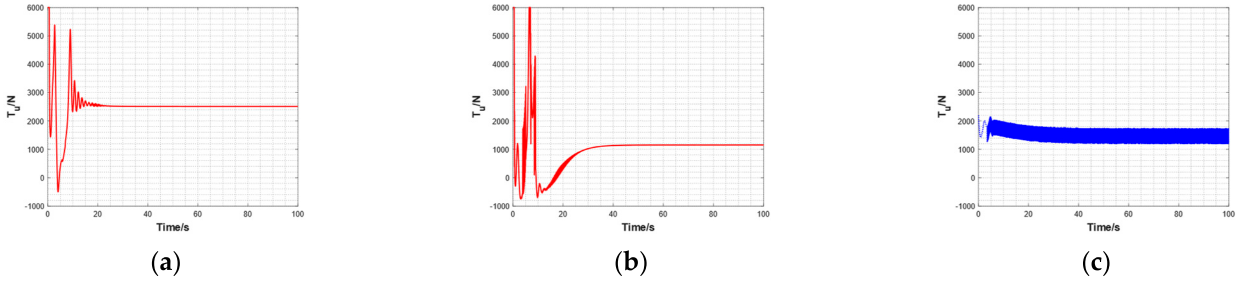
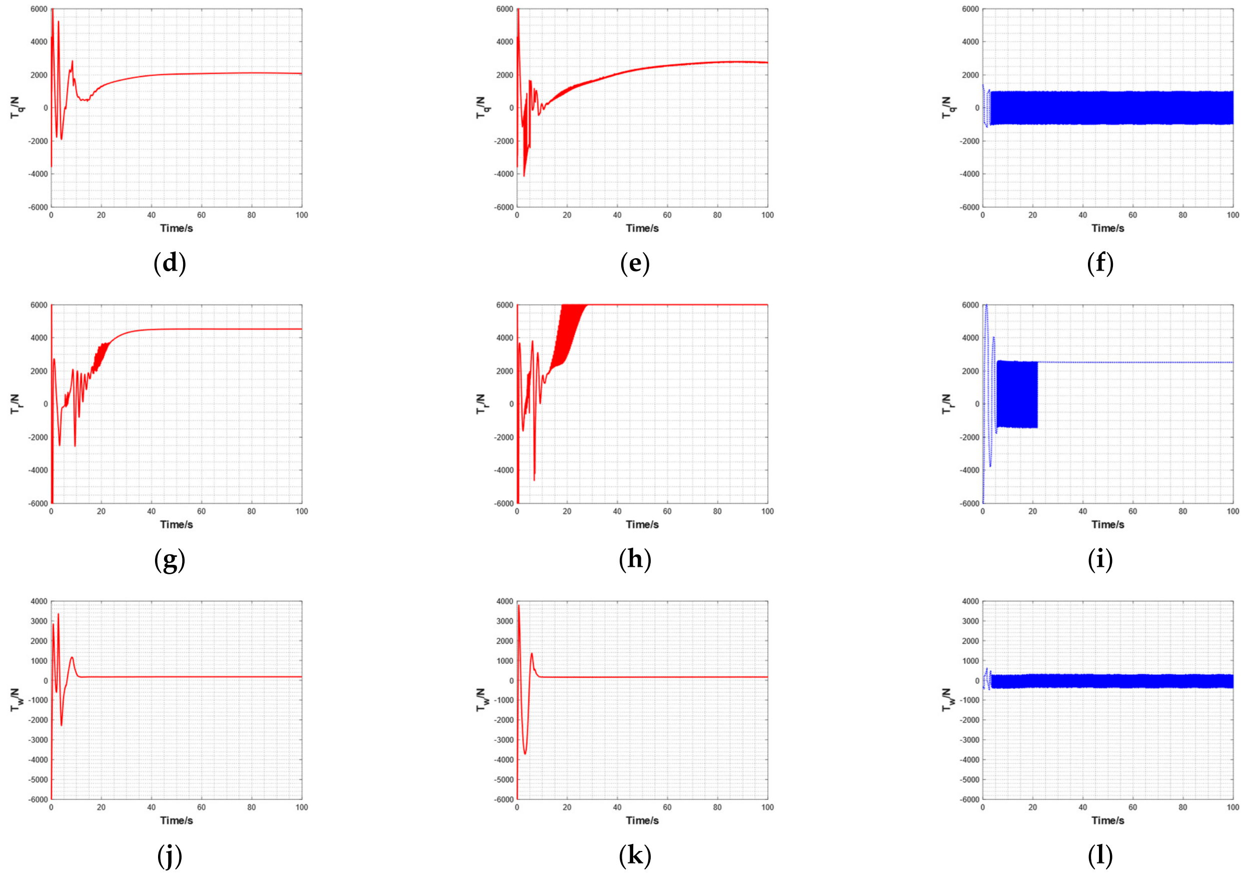


Figure 9. The positional errors between the robot and the tracking target: x_e, y_e and z_e.



(**a**) (**b**) (**c**)

Figure 10. *Cont.*

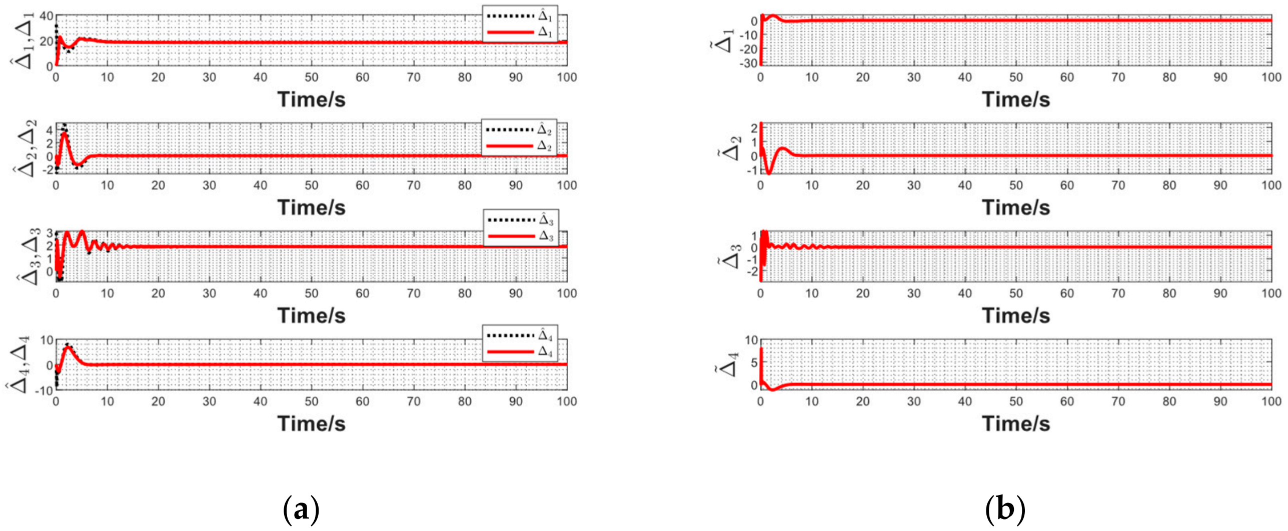


(**d**) (**e**) (**f**)

(**g**) (**h**) (**i**)

(**j**) (**k**) (**l**)

Figure 10. The control moments and forces T_u, T_q, T_r and T_w. (**a**) T_u of PNTSMC; (**b**) T_u of PTSMC; (**c**) T_u of TSMC; (**d**) T_q of PNTSMC; (**e**) T_q of PTSMC; (**f**) T_q of TSMC; (**g**) T_r of PNTSMC; (**h**) T_r of PTSMC; (**i**) T_r of TSMC; (**j**) T_w of PNTSMC; (**k**) T_w of PTSMC; (**l**) T_w of TSMC.



(**a**) (**b**)

Figure 11. The total model uncertainty estimation: (**a**) the total uncertainties term Δ_i; the estimates of the total uncertainties term $\hat{\Delta}_i$; (**b**) the errors of the above term $\widetilde{\Delta}_i$.

Figure 7 shows the trajectories of the underwater robot and the target. After cruising for a short distance, the trajectory of the underwater robot overlaps with that of the target. The results show that the three controllers can achieve accurate trajectory tracking in the presence of external disturbances and modeling uncertainties. In Figure 8, the tracking errors of PNTSMC and PTSMC could converge to zero with the prescribed performance and converge faster with less overshooting than those of TSMC. It can be observed in Figure 8 that the convergence process of PNTSMC is smoother than that of PTSMC. Figure 9 shows the posture errors. As y_e and z_e approach 0 m, x_e approaches the safe distance $\widetilde{\delta} = 0.102$ m. It not only achieved convergence in the underactuated direction, but also avoided a collision, which shows the good performance of the tracking error and the success of the tracking guidance law. Figure 10 shows the control moments and forces generated by the three controllers. TSMC had the most dramatic chattering. Since the RBF neural network can approximate the unmodeled uncertainties, the gain of the sliding mode controller can be reduced to attenuate chattering. Simulation results show that chattering phenomenon of PNTSMC is weaker than that of PTSMC. From Figure 11, we can see that the RBF neural network effectively approximates the uncertainties of the underwater robot.

Remark 1. *If the chattering in Figure 10 cannot be tolerated by the thrusters, continuous saturation functions or hyperbolic tangent functions can be used to replace the sign functions to further attenuate chattering. However, this approach comes at the cost of losing control accuracy and reducing robustness.*

In order to better understand the performance of different controllers, typical criteria such as steady-state error, convergence time and root mean square error are used for a quantitative comparison in Table 2. The steady-state error reflects the control accuracy and anti-disturbance ability of the system. The convergence time reflects the dynamic time of the system. In addition, root mean square error is reported to describe the controller's control performance during the whole tracking process. Firstly, the results of the simulation for PNTSMC and TSMC are compared. It can be seen that PNTSMC has advantages over TSMC in steady-state error, which shows that PNTSMC can achieve more accurate tracking. As for the convergence time, the convergence times of tracking errors $\delta_{(t)}$, $\alpha_{(t)}$ and $Z_{e(t)}$ were shortened by 30.74, 0.152 and 0.621 s, respectively—reduced by 74.4%, 3.2% and 19.1%. The convergence time of tracking error $\beta_{(t)}$ increased by 1.171 s, that is, 36.8%. In general, PNTSMC accelerated the convergence process. The root-mean-square errors of the tracking errors of the PNTSMC are all smaller, corresponding to $\delta_{(t)}$, $\beta_{(t)}$, $\alpha_{(t)}$ and $Z_{e(t)}$, which were reduced by 0.4238, 0.1279, 3.1862 and 0.0012 m, respectively—70.5%, 25.2%, 50.1% and 11.0%. Secondly, for the comparison between PNTSMC and PTSMC, the gap between the two is very small by most criteria. The performance of PNTSMC was slightly better. In conclusion, we demonstrated the advantages of PNTSMC in rapid acquisition of stability and accurate trajectory tracking due to the prescribed performance technique.

Table 2. Comparison of the performances of different controllers.

Quantitative Comparison	**Control Scheme**	$\delta_{(t)}$	$\beta_{(t)}$	$\alpha_{(t)}$	$Z_{e(t)}$
Steady-state error	PNTSMC	0.0001 m	$-1.6155 \times 10^{-5\circ}$	$7.4915 \times 10^{-5\circ}$	-1.8646×10^{-5} m
	PTSMC	0.0001 m	$7.2886 \times 10^{-5\circ}$	0.0012°	-7.355×10^{-6} m
	TSMC	0.0030 m	$1.848 \times 10^{-4\circ}$	0.0019°	1.6467×10^{-6} m
Convergence Time	PNTSMC	10.59 s	4.351 s	4.639 s	2.625 s
	PTSMC	10.7 s	4.146 s	4.121 s	3.951 s
	TSMC	41.33 s	3.18 s	4.797 s	3.246 s
Root-mean-square error	PNTSMC	0.1870 m	0.3791°	3.1786°	0.0097 m
	PTSMC	0.1942 m	0.4598°	6.1865°	0.0104 m
	TSMC	0.6008 m	0.5070°	6.3648°	0.0109 m

According to the above results, the proposed controller can force an underwater robot to accurately track the desired target with prescribed steady-state and dynamic performances in the presence of modeling uncertainties and external disturbances. Overall, PNTSMC provides better dynamic performance, steady performance and chattering repression than the other two controllers.

5. Conclusions

To achieve good target tracking motion control performance in an environment with uncertainties and external disturbances, we designed a neural network non-singular terminal sliding mode controller for underactuated underwater robots with prescribed performances. By using non-singular terminal sliding mode and the RBF neural network, the controller achieves strong robustness against the modeling uncertainties and external disturbances. The prescribed performance technique ensures that the underwater robot has excellent target tracking control performance. Numerical simulations showed that the proposed controller has better dynamic performance, steady-state performance and chattering suppression, and can accomplish the target tracking task accurately and reliably in the presence of modeling uncertainties and external disturbances. The controller proposed in this paper is unique in its ability to achieve robustness against modeling uncertainties and the external disturbances, finite-time convergence, attenuating chattering and prescribed performances, simultaneously. It provides a new research topic for the target tracking control of underwater robots.

There are still some open problems to improve the target tracking control of underwater robots which are for future work:

1. A hardware implementation of the proposed controller will be realized in a practical robot control system, and the possible concentration degree in the actual deployment would be discussed. Non-singular terminal sliding mode control and RBF neural networks have been used on a variety of platforms, and the prescribed performance technique only adds some logarithmic operations. In mainstream embedded computers, the computational load of the controller proposed in this paper is affordable. We will put this controller to the test in a computer with Intel® Atom™ N455 as the core.
2. When there is a large deviation in the tracking error, or when the underwater robot encounters a large disturbance, the prescribed performance technique may produce singular values. It is necessary to adaptively adjust the relevant parameters according to the real environment. At the same time, a finite-time performance function will be considered to improve the control performance.
3. After comparing with PTSMC and TSMC, the method proposed in this paper should also be compared with other state-of-the-art positioning error and tracking error methods. This is one of the directions for further extending and improving the proposed controller.

Author Contributions: Conceptualization, L.G., W.L. and L.L.; methodology, L.G., W.L. and L.L.; software, L.G.; validation, L.G.; formal analysis, L.G., Y.L., X.W. and Z.L.; resources, L.G., X.W. and Z.L.; data curation, L.G. and Y.L.; writing—original draft preparation, L.G.; writing—review and editing, L.G., W.L. and L.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded in part by the National Science Foundation of China (project number 61903304), in part by the Fundamental Research Funds for the Central Universities (project number 3102020HHZY030010), in part by Science and Technology Program of Xi'an (project number 2020KJRC0119) and in part by the 111 Project under grant number B18041.

Institutional Review Board Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest with respect to the research, authorship, and publication of this article.

References

1. Muthugala, M.A.V.J.; Samarakoon, S.M.B.P.; Elara, M.R. Toward energy-efficient online Complete Coverage Path Planning of a ship hull maintenance robot based on Glasius Bio-inspired Neural Network. *Expert Syst. Appl.* **2022**, *187*, 115940. [CrossRef]
2. Chi, X.; Zhan, Q. Design and Modelling of an Amphibious Spherical Robot Attached with Assistant Fins. *Appl. Sci.* **2021**, *11*, 3739. [CrossRef]
3. Song, C.; Cui, W. Review of Underwater Ship Hull Cleaning Technologies. *J. Mar. Sci. Appl.* **2020**, *19*, 415–429. [CrossRef]

4. Yu, C.; Xiang, X.; Wilson, P.A.; Zhang, Q. Guidance-Error-Based Robust Fuzzy Adaptive Control for Bottom Following of a Flight-Style AUV With Saturated Actuator Dynamics. *IEEE Trans. Cybern.* **2020**, *50*, 1887–1899. [CrossRef] [PubMed]
5. Tran, H.N.; Pham, T.N.N.; Choi, S.H. Robust depth control of a hybrid autonomous underwater vehicle with propeller torque's effect and model uncertainty. *Ocean Eng.* **2021**, *220*, 108257. [CrossRef]
6. Cho, G.R.; Li, J.H.; Park, D.; Jung, J.H. Robust trajectory tracking of autonomous underwater vehicles using back-stepping control and time delay estimation. *Ocean Eng.* **2020**, *201*, 107131. [CrossRef]
7. Yan, Z.; Gong, P.; Zhang, W.; Wu, W. Model predictive control of autonomous underwater vehicles for trajectory tracking with external disturbances. *Ocean Eng.* **2020**, *217*, 107884. [CrossRef]
8. Gan, W.; Zhu, D.; Ji, D. QPSO-model predictive control-based approach to dynamic trajectory tracking control for unmanned underwater vehicles. *Ocean Eng.* **2018**, *158*, 208–220. [CrossRef]
9. Che, G.; Yu, Z. Neural-network estimators based fault-tolerant tracking control for AUV via ADP with rudders faults and ocean current disturbance. *Neurocomputing* **2020**, *411*, 442–454. [CrossRef]
10. Elhaki, O.; Shojaei, K. Neural network-based target tracking control of underactuated autonomous underwater vehicles with a prescribed performance. *Ocean Eng.* **2018**, *167*, 239–256. [CrossRef]
11. Guo, J.; Li, D.; He, B. Intelligent Collaborative Navigation and Control for AUV Tracking. *IEEE Trans. Ind. Inform.* **2021**, *17*, 1732–1741. [CrossRef]
12. Guo, Y.; Qin, H.; Xu, B.; Han, Y.; Fan, Q.Y.; Zhang, P. Composite learning adaptive sliding mode control for AUV target tracking. *Neurocomputing* **2019**, *351*, 180–186. [CrossRef]
13. Yan, Z.; Wang, M.; Xu, J. Global Adaptive Neural Network Control of Underactuated Autonomous Underwater Vehicles with Parametric Modeling Uncertainty. *Asian J. Control.* **2019**, *21*, 1342–1354. [CrossRef]
14. Miao, J.; Wang, S.; Zhao, Z.; Li, Y.; Tomovic, M.M. Spatial curvilinear path following control of underactuated AUV with multiple uncertainties. *ISA Trans.* **2017**, *67*, 107–130. [CrossRef] [PubMed]
15. Lamraoui, H.C.; Qidan, Z. Path following control of fully-actuated autonomous underwater vehicle in presence of fast-varying disturbances. *Appl. Ocean Res.* **2019**, *86*, 40–46. [CrossRef]
16. Zhang, Z.; Liu, B.; Wang, L. Autonomous underwater vehicle depth control based on an improved active disturbance rejection controller. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 172988141989153. [CrossRef]
17. Ali, N.; Tawiah, I.; Zhang, W. Finite-time extended state observer based nonsingular fast terminal sliding mode control of autonomous underwater vehicles. *Ocean Eng.* **2020**, *218*, 108179. [CrossRef]
18. Elmokadem, T.; Zribi, M.; Youcef-Toumi, K. Terminal sliding mode control for the trajectory tracking of underactuated Autonomous Underwater Vehicles. *Ocean. Eng.* **2017**, *129*, 613–625. [CrossRef]
19. Patre, B.M.; Londhe, P.S.; Waghmare, L.M.; Mohan, S. Disturbance estimator based non-singular fast fuzzy terminal sliding mode control of an autonomous underwater vehicle. *Ocean Eng.* **2018**, *159*, 372–387. [CrossRef]
20. Wang, Z.; Liu, Y.; Guan, Z.; Zhang, Y. An Adaptive Sliding Mode Motion Control Method of Remote Operated Vehicle. *IEEE Access* **2021**, *9*, 22447–22454. [CrossRef]
21. Wang, Z.; Liu, Y.; Guan, Z.; Zhang, Y. Improved line-of-sight trajectory tracking control of under-actuated AUV subjects to ocean currents and input saturation. *Ocean Eng.* **2019**, *174*, 14–30. [CrossRef]
22. Yan, Z.; Wang, M.; Xu, J. Robust adaptive sliding mode control of underactuated autonomous underwater vehicles with uncertain dynamics. *Ocean Eng.* **2019**, *173*, 802–809. [CrossRef]
23. Mu, W.; Wang, Y.; Sun, H.; Liu, G. Double-Loop Sliding Mode Controller with An Ocean Current Observer for the Trajectory Tracking of ROV. *J. Mar. Sci. Eng.* **2021**, *9*, 1000. [CrossRef]
24. Zhang, Z.; Yan, W.; Li, H. Distributed Optimal Control for Linear Multiagent Systems on General Digraphs. *IEEE Trans. Autom. Control.* **2021**, *66*, 322–328. [CrossRef]
25. Zhang, Z.; Li, H.; Yan, W. Fully Distributed Control of Linear Systems with Optimal Cost on Directed Topologies. *IEEE Trans. Circuits Systems. II Express Briefs* **2021**, *68*, 336–340. [CrossRef]
26. Li, Z.; Liu, W.; Li, L.; Guo, L.; Zhang, W. Modeling and adaptive controlling of cable-drogue docking system for autonomous underwater vehicles. *Int. J. Adapt. Control. Signal Processing* **2021**, *36*, 354–372. [CrossRef]
27. Wu, H.; Song, S.; You, K.; Wu, C. Depth Control of Model-Free AUVs via Reinforcement Learning. *IEEE Trans. Syst. Man Cybern.-Syst.* **2017**, *49*, 2499–2510. [CrossRef]
28. Carlucho, I.; De Paula, M.; Wang, S.; Petillot, Y.; Acosta, G.G. Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning. *Robot. Auton. Syst.* **2018**, *107*, 71–86. [CrossRef]
29. Anderlini, E.; Parker, G.G.; Thomas, G. Docking Control of an Autonomous Underwater Vehicle Using Reinforcement Learning. *Appl. Sci.* **2019**, *9*, 3456. [CrossRef]
30. Sun, Y.; Zhang, C.; Zhang, G.; Xu, H.; Ran, X. Three-Dimensional Path Tracking Control of Autonomous Underwater Vehicle Based on Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2019**, *7*, 443. [CrossRef]
31. Cao, J.; Sun, Y.; Zhang, G.; Jiao, W.; Wang, X.; Liu, Z. Target tracking control of underactuated autonomous underwater vehicle based on adaptive nonsingular terminal sliding mode control. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 172988142091994. [CrossRef]
32. Bechlioulis, C.P.; Rovithakis, G.A. Robust Adaptive Control of Feedback Linearizable MIMO Nonlinear Systems with Prescribed Performance. *IEEE Trans. Autom. Control.* **2008**, *53*, 2090–2099. [CrossRef]

33. Bechlioulis, C.P.; Karras, G.C.; Heshmati-Alamdari, S.; Kyriakopoulos, K.J. Trajectory Tracking with Prescribed Performance for Underactuated Underwater Vehicles Under Model Uncertainties and External Disturbances. *IEEE Trans. Control. Syst. Technol.* **2017**, *25*, 429–440. [CrossRef]
34. Liang, H.; Fu, Y.; Gao, J.; Cao, H. Finite-time velocity-observed based adaptive output-feedback trajectory tracking formation control for underactuated unmanned underwater vehicles with prescribed transient performance. *Ocean Eng.* **2021**, *233*, 109071. [CrossRef]
35. Shojaei, K.; Chatraei, A. Robust platoon control of underactuated autonomous underwater vehicles subjected to nonlinearities, uncertainties and range and angle constraints. *Appl. Ocean Res.* **2021**, *110*, 102594. [CrossRef]
36. Wang, Y.; Wang, H.; Li, M.; Wang, D.; Fu, M. Adaptive fuzzy controller design for dynamic positioning ship integrating prescribed performance. *Ocean. Eng.* **2021**, *219*, 107956. [CrossRef]
37. Li, J.; Du, J.; Hu, X. Robust adaptive prescribed performance control for dynamic positioning of ships under unknown disturbances and input constraints. *Ocean Eng.* **2020**, *206*, 107254. [CrossRef]
38. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
39. Wang, W.; Huang, J.; Wen, C. Prescribed performance bound-based adaptive path-following control of uncertain nonholonomic mobile robots. *Int. J. Adapt. Control Signal Processing* **2017**, *31*, 805–822. [CrossRef]
40. Park, J.; Sandberg, I.W. Universal Approximation Using Radial-Basis-Function Networks. *Neural Comput.* **1991**, *3*, 246–257. [CrossRef]