



---

*Research article*

## **Fault-tolerant control for nonlinear systems with a dead zone: Reinforcement learning approach**

**Zichen Wang<sup>1,\*</sup> and Xin Wang<sup>2</sup>**

<sup>1</sup> College of Westa, Southwest University, Chongqing 400715, China

<sup>2</sup> College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China

\* **Correspondence:** Email: [swuzcwang@163.com](mailto:swuzcwang@163.com).

**Abstract:** This paper focuses on the adaptive reinforcement learning-based optimal control problem for standard nonstrict-feedback nonlinear systems with the actuator fault and an unknown dead zone. To simultaneously reduce the computational complexity and eliminate the local optimal problem, a novel neural network weight updated algorithm is presented to replace the classic gradient descent method. By utilizing the backstepping technique, the actor critic-based reinforcement learning control strategy is developed for high-order nonlinear nonstrict-feedback systems. In addition, two auxiliary parameters are presented to deal with the input dead zone and actuator fault respectively. All signals in the system are proven to be semi-globally uniformly ultimately bounded by Lyapunov theory analysis. At the end of the paper, some simulation results are shown to illustrate the remarkable effect of the proposed approach.

**Keywords:** fault-tolerant control; input dead zone; nonstrict-feedback; nonlinear system; reinforcement learning

---

### **1. Introduction**

Optimal control theory originated in the 1960s and has become an important part of automatic control theory primarily owing to its spirit of seeking the optimal solution from all possible control schemes [1, 2]. Optimal control problems for linear systems can generally be settled by solving Riccati equations. However, when it comes to nonlinear systems, there are quite a few effective methods since the Hamilton-Jacobi-Bellman (HJB) equation should be addressed. The HJB equation is generally difficult to solve analytically. To overcome this bottleneck, a lot of remarkable approaches have been developed, such as adaptive dynamic [3, 4], the use of actor-critic neural networks (ACNNs) [5, 6], policy iteration, and so on. Reinforcement learning (RL), as a method that can solve optimal control

problems in nonlinear systems to avoid solving the HJB equation, has received widespread attention in the past decades. In 1974, Werbos first applied the idea of RL to optimal control theory [7]. Since then, many outstanding outcomes have been subsequently discovered [8–10]. At present, RL theory is usually implemented by using ACNNs, where the critic neural network (CNN) provides policy evaluation and the actor neural network (ANN) updates the present policy. The RL algorithm can reduce energy consumption beyond that of other algorithms under the premise of achieving system stability; it became a significant method of modern control theory. In recent years, more and more approaches to RL have been presented in various fields, including online RL [11, 12], integral RL [13, 14], off-policy RL [15, 16], etc.

### *1.1. Motivation*

General RL strategies utilize the gradient descent method to obtain the ideal weights of neural networks, which often fall into the local optimal problem [17], resulting in neural network estimation errors that cannot easily meet the requirements. To overcome this bottleneck, Bai et al. proposed the multigradient recursive (MGR) algorithm in [17] to obtain the global optimum solution. By updating pseudo-gradients, this state-of-the-art technique can settle down the local optimal problem and accelerate the convergence rate of the neural network weight. However, the MGR algorithm suffers because of a heavy computational burden. Thus, it is necessary to mention the minimal learning parameter (MLP) scheme. It can reduce the number of updated laws without prominently reducing the estimation accuracy. Many studies have been proposed to validate the effectiveness of the MLP. Nevertheless, the aforementioned papers adopt only one of the two algorithms and fail to combine both the MLP and MGR algorithms to combine their advantages.

The optimal control problem of strict-feedback nonlinear systems has been widely studied. However, none of these strategies can be extended into the field of nonstrict-feedback systems [18, 19]. For this sake, some scholars have proposed their studies to overcome the bottleneck. For example, Tong et al. [20], presented a novel fuzzy tracking control design for a nonstrict-feedback SISO system. Bai et al. [21] utilized MLP-based RL theory to overcome optimal control problems for a class of nonstrict-feedback systems. However, all of the above results omitted the influence of actuator fault and dead zone input. These are common factors that affect the stability of the system. This negligence can lead to severe damage and must be taken seriously. Thus, many works are presented to offset their influence. However, no one has considered the situation that the dead zones and the actuator fault occur simultaneously. Thus, how to obtain an optimal controller for a nonstrict-feedback system with actuator faults and input dead zone with minimal computation and enough accuracy will be an important task, and it is the motivation for the current investigation.

### *1.2. Related work*

Combining with the back-stepping technique, the authors made a thorough investigation of the tracking control problem of the strict-feedback nonlinear systems [14–16]. At present, the back-stepping approach is introduced into the analysis for the high-order nonlinear systems. Li et al. [22], investigated the optimal control problem of a class of SISO strict-feedback systems via the fuzzy control method. Modares et al. [23] have developed an integral RL approach for the strict-feedback system with input constraints. Wang et al. [24] proposed an optimal fault-tolerant control strategy for a non-

linear strict-feedback system via adaptive critic design.

Besides, many papers have been proposed to study novel neural network weight updated algorithms. Li et al. [25] utilized the MLP technique to overcome the fault-tolerant problem of a class of multiagent systems. Liu et al. [26] designed an RL controller by applying a MLP scheme for classic MIMO systems with external disturbance. Bai et al. [27] developed an event trigger control scheme for the multiagent system based on the MLP technique.

Furthermore, many scholars are committed to investigating a tolerance strategy for the input dead zone and actuator fault. For instance, Wang and Yang [28] studied the fault detection problem for linear systems with disturbance. Tan et al. [29] developed a compensation control scheme for a class of discrete-time systems that has actuator failure. In addition, Na et al. [30] provided an adaptive dynamic control approach for a system with an unknown dead zone.

### 1.3. Paper contribution and organization

Based on the above discussion, an RL optimal controller is built in this paper to deal with the fault tolerant control problem for a class of nonstrict-feedback nonlinear systems in discrete time with an unknown dead zone input and actuator fault. To deal with the dead zone and actuator fault issues, we propose two auxiliary systems to offset the influence. The ANN and CNN are utilized to approximate the unknown terms and long-time utility functions, respectively. We propose a novel approach to update the neural networks' weight. The stability of all signals in the closed-loop is rigorously proved and tracking errors are converged to a small compact set. The novelties of this paper can be concluded to be as follows

- 1) We propose a novel neural networks weigh-updated algorithm to eliminate the local optimal problem and reduce computational burden. Besides, compared with the ordinary gradient descent algorithm [11, 26], the proposed approach can achieve a faster weight convergence rate.
- 2) We formulate a modified backstepping method with additional parameters to offset the influence of input dead zone, the actuator fault, and the algebraic loop problem. Besides that, the unified fault-tolerant control algorithms are developed based on the RL strategy.

The organization of this paper is given below. In Section 2, descriptions of the system and radial basis function neural network (RBF NN) theory are given. In Subsection 3.1, the CNN and our novel update law are presented. In Subsection 3.2, the design procedure of the adaptive RL controller is provided. In Section 4, we propose some simulation results to show the contributions of the scheme presented in this paper. The conclusion is provided in Section 5.

## 2. Problem formulation and preliminaries

### 2.1. Model description

The dynamics of a standard  $n$ -order strict-feedback nonlinear system [31–34] can be described as follows:

$$\begin{cases} x_i(k+1) = \varphi_i(\bar{x}_i(k)) + \phi_i(\bar{x}_i(k))x_{i+1}(k) \\ x_n(k+1) = \varphi_n(\bar{x}_n(k)) + \phi_n(\bar{x}_n(k))U(k) + d(k) \\ y(k) = x_1(k) \end{cases} \quad (2.1)$$

where  $x_i(k) \in R$  for  $i = 1, \dots, n$  represents the state variable of the system. The notation  $\bar{x}_n(k) = [x_1(k), x_2(k), \dots, x_n(k)]^\top \in R^n$  denotes the vectors of the states. The notations  $U(k) \in R$  and  $y(k) \in R$  are the input and output signals, respectively. Notation  $d(k)$  stands for the external disturbance. Notations  $\varphi_i(\cdot)$  and  $\phi_i(\cdot)$  represent unknown smooth nonlinear functions.

Motivated by the transformation proposed in [18, 19], the nonstrict-feedback system (2.1) can be further expressed as

$$\begin{cases} x_i(k+n-i+1) = \varphi_i(\bar{x}_n(k+n-i)) + \phi_i(\bar{x}_n(k+n-i))x_{i+1}(k+n-i) \\ x_n(k+1) = \varphi_n(\bar{x}_n(k)) + \phi_n(\bar{x}_n(k))U(k) + d(k) \\ y(k) = x_1(k). \end{cases} \quad (2.2)$$

To proceed smoothly, an assumption is introduced in the following sequel.

*Assumption 1:* According to the contributions in [34, 35], the functions  $\varphi_i(\bar{x}(k))$  and  $\phi_i(\bar{x}(k))$  satisfy  $0 < \underline{\varphi} < \varphi_i(\bar{x}(k)) < \bar{\varphi}$  and  $0 < \underline{\phi} < \phi_i(\bar{x}(k)) < \bar{\phi}$ , where  $\bar{\varphi}$  and  $\underline{\varphi}$  are the unknown upper bound and the unknown lower bound of  $\varphi_i(\bar{x}(k))$  and  $\bar{\phi}$  and  $\underline{\phi}$  are the unknown upper bound and unknown lower bound of  $\phi_i(\bar{x}(k))$ , respectively. The external disturbance is bounded and satisfies  $|d(k)| \leq \bar{d}$  with  $\bar{d}$  being an unknown positive constant.

The control signal with the actuator fault and input dead zone can be described as  $U(k) = \psi(k)u(k) + \delta(k)$ , where  $\psi(k)$  and  $\delta(k)$  denote the efficiency factor and the unknown drift fault of the actuator, respectively. We assume that  $\psi(k)$  is a positive constant with  $\psi(k) < \bar{\psi} < 1$ , where  $\bar{\psi}$  is an unknown constant. Further,  $\delta(k)$  satisfies  $\delta(k) < \bar{\delta}$  with  $\bar{\delta}$  being the upper bound. The dead zone can be defined as  $u(k) = D(v(k))$ , where  $v(k)$  represents the input of the dead zone and  $D(\cdot)$  is a function of  $v(k)$  which represents the output of the dead zone. According to propositions in [31], the dead zone is expressed as

$$D(v(k)) = \begin{cases} b_r(v(k) - f_r), & v(k) \geq f_r \\ 0, & -f_l < v(k) < f_r \\ b_l(v(k) + f_l), & v(k) \leq -f_l \end{cases} \quad (2.3)$$

where  $b_r$  and  $b_l$  denote the right slope of the dead zone and the left slope of the dead zone, respectively. The notations  $f_r$  and  $f_l$  are breakpoints of the input. For the purpose of simplifying the following calculation, the expression  $D(v(k))$  can be converted into a new form, as follows

$$D(v(k)) = b(k)v(k) + f(k) \quad (2.4)$$

where  $b(k)$  and  $f(k)$  can be described as

$$b(k) = \begin{cases} b_r, & v(k) > 0 \\ b_l, & v(k) \leq 0 \end{cases} \quad f(k) = \begin{cases} -b_r f_r, & v(k) \geq b_r \\ -b(k)v(k), & -b_r < v(k) < b_l \\ b_l f_l, & v(k) \leq b_l. \end{cases} \quad (2.5)$$

We suppose that  $b(k)$  and  $f(k)$  satisfy  $0 < \underline{b} < |b(k)| < \bar{b}$  and  $0 < \underline{f} < |f(k)| < \bar{f}$ , respectively. The control signal  $U(k)$  can be reorganized as

$$U(k) = \psi(k)(b(k)v(k) + f(k)) + \delta(k). \quad (2.6)$$

In this paper, the RL controller is developed for the nonstrict-feedback nonlinear system (2.2), ensuring the semi-globally uniformly ultimately bounded (SGUUB) capability of all signals in the closed-loop system. Based on the ACNNs, the tracking error  $\xi_1(k)$  is required to converge to the neighborhood of zero it will be specified subsequently.

## 2.2. RBF NN

Note that, the RBF NN can approximate any smooth nonlinear functions over a compact set. That is to say, considering an unknown nonlinear function  $F(N)$ , there exists an RBF NN  $W^{*\top}S(N)$  such that  $F(N) = W^{*\top}S(N) + \sigma(N)$ , where  $W^* = [w_1, \dots, w_l]^\top \in R^l$  denotes the ideal weight vector,  $l$  represents the node number in the hidden layer and  $\sigma(N)$  is the estimation error. Both  $W^*$  and  $\sigma(N)$  satisfy  $\|W^*\| < \bar{W}$  and  $\|\sigma(N)\| < \bar{\sigma}$  with  $\bar{W}$  and  $\bar{\sigma}$  as unknown upper bounds. The notation  $S(N) = [s_1(N), \dots, s_l(N)]^\top$  is the vector of the basis function and  $s_i(N)$  is applied as Gaussian form  $s_i(N) = \exp\left[-\frac{(N-c_i)^\top(N-c_i)}{\eta_i^2}\right]$ , where  $c_i$  represents the kernel of the receptive field,  $\eta_i(k)$  denotes the width of the function. Because  $0 < s_i(N) < 1$ , we can further derive that  $0 < \sum_{i=1}^l s_i(N)s_i(N) = S(N)^\top S(N) < l$ .

## 3. Design of adaptive RL controller

### 3.1. CNN

The utility function [35] can be chosen as

$$\rho(k) = \begin{cases} 0, & |\xi_1(k)| \geq \varpi \\ 1, & |\xi_1(k)| < \varpi \end{cases} \quad (3.1)$$

where  $\varpi$  is a positive constant that denotes the threshold value of tracking performance. The tracking error is written as  $\xi_1(k) = y(k) - x_d(k)$   $x_d(k)$  indicates the reference signal. The long term strategic utility function [21, 27] is given by

$$M(k) = \rho(k+1) + a\rho(k+2) + a^2\rho(k+3) + \dots, \quad (3.2)$$

where  $a$  is a predefined positive parameter satisfying  $a < 1$ . According to RBF NN theory, the long term utility function  $M(k)$  is defined below

$$M(k) = W_M^\top S_M(k) + \delta_M(k) \quad (3.3)$$

where  $W_M$  and  $\delta_M(k)$  indicate the ideal weight vector and the error of the approximation, respectively. Let  $S_M(k)$  be the RBF NN basis function. We define  $\hat{M}(k) = \hat{W}_M^\top(k)S_M(k)$ ; it denotes the estimation of function  $M(k)$  with  $\hat{W}_M(k)$  being the estimation of the ideal weight  $W_M$ .

On the basis of the MLP scheme,  $\hat{M}(k)$  can be written in the form

$$\hat{M}(k) = \hat{\Psi}_M(k) \|S_M(k)\| \quad (3.4)$$

where  $\|\cdot\|$  indicates the Euclidean norm,  $\hat{\Psi}_M(k) = \|\hat{W}_M(k)\|$  is true for  $\hat{\Psi} \leq \bar{\Psi}$  and  $\bar{\Psi}$  is a positive unknown constant.

According to the scheme in [36], the equation of the Bellman error is designed as

$$E_M(k) = a\hat{M}(k) - [\hat{M}(k-1) - \rho(k)]. \quad (3.5)$$

Adopting the cost function in its quadratic form  $\beta_M(k) = (1/2)(E_M(k))^2$ , the gradient of  $\hat{W}_M$  is obtained

$$\Delta \hat{W}_M(k) = a \|S_M(k)\| \left[ a\hat{M}(k) - \hat{M}(k-1) + \rho(k) \right]. \quad (3.6)$$

Defining  $\omega_M(k-j+1) = \hat{\Psi}_M(k) \|S_M(k-j+1)\|$ , we can further get

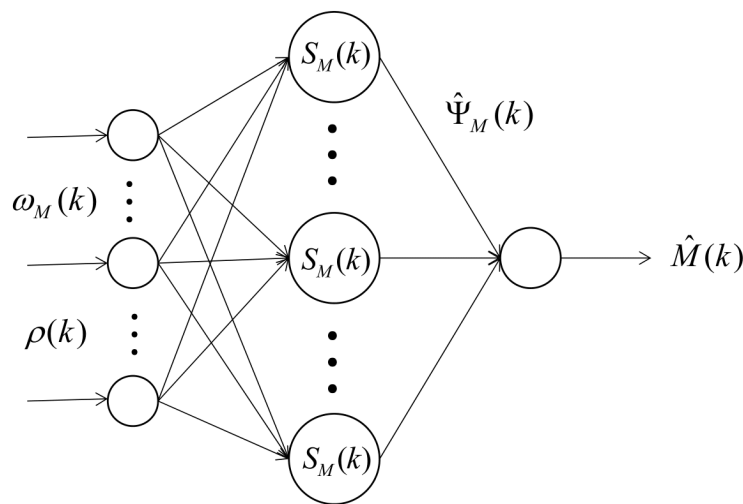
$$g(\iota, \beta_M(k)) = \sum_{j=1}^{\iota} a \|S_M(k-j+1)\| \left[ a\omega_M(k-j+1) - \omega_M(k-j) + \rho(k-j+1) \right] \quad (3.7)$$

where  $\iota \geq 1$  is a positive predefined constant that indicates the step length of the gradient.

Together with (3.7), the updated law of  $\hat{\Psi}_M$  can be obtained

$$\hat{\Psi}_M(k+1) = \hat{\Psi}_M(k) - \mu_M \sum_{j=1}^{\iota} a \|S_M(k-j+1)\| \left[ a\omega_M(k-j+1) - \omega_M(k-j) + \rho(k-j+1) \right] \quad (3.8)$$

where  $\mu_M$  is the selected learning rate. The structure of the CNN is shown in Figure 1.



**Figure 1.** Structure of the CNN.

*Remark 1:* The neural networks in this paper are updated by our weight-updated algorithm. As compare to the classic gradient descent method, our updated algorithm has the following advantages: 1) Reduces the computational complexity; 2) Eliminates the local optimal problem; 3) Accelerates neural networks weight convergence speed;

### 3.2. Design of $n$ -step adaptive neural network controller

In this section, an ANN will be utilized to implement the  $n$ -step backstepping RL control strategy. Specifically, two auxiliary signals are introduced in the  $n$  step to eliminate the impact of the dead zone and actuator fault.

*Step 1:* The tracking error can be defined as  $\xi_1(k+n) = x_1(k+n) - x_d(k+n)$  and  $\xi_2(k+n-1) = x_2(k+n-1) - \alpha_1(k)$ . According to System (2.2), the tracking error  $\xi_1(k+n)$  can be further deduced as

$$\begin{aligned} \xi_1(k+n) = & \phi_1(\hat{x}_n(k+n-1)) \left[ \left( \frac{\varphi_1(\bar{x}_n(k+n-1))}{\phi_1(\bar{x}_n(k+n-1))} - \frac{x_d(k+n)}{\phi_1(\bar{x}_n(k+n-1))} + x_d(k+n) \right) \right. \\ & \left. + \alpha_1(k) - x_d(k+n) + \xi_2(k+n-1) \right] \end{aligned}$$

where  $\alpha_1(k)$  denotes the virtual controller. Let

$$\gamma_1(k) = -\left(\frac{\varphi_1(\bar{x}_n(k+n-1))}{\phi_1(\bar{x}_n(k+n-1))} + x_d(k+n) - \frac{x_d(k+n)}{\phi_1(\bar{x}_n(k+n-1))}\right). \quad (3.9)$$

With the universal approximation capability of the RBF NN,  $\gamma_1(k) = W_1^T S_1(N_1(k)) + \sigma_1(k)$  can be approximated, where  $W_1$  represents the ideal weight vector. We define  $N_1(k) = [\bar{x}_n(k+n-1), x_d(k+n)]$  and  $\sigma_1(k)$  indicates the approximation error. Suppose that  $W_1$  and  $\sigma_1(k)$  satisfy  $\|W_1\| < \bar{W}_1$  and  $\|\sigma_1(k)\| < \bar{\sigma}_1$ , respectively. Both  $\bar{W}_1$  and  $\bar{\sigma}_1$  are corresponding upper bounds.

Combining (3.9) and  $\gamma_1(k)$ ,  $\xi_1(k+n)$  can be further expressed as

$$\xi_1(k+n) = \phi_1(\hat{x}_n(k+n-1)) \left[ -W_1^T S_1(N_1(k)) - \sigma_1(k) + \alpha_1(k) + \xi_2(k+n-1) - x_d(k+n) \right]. \quad (3.10)$$

For the purpose of solving the algebraic loop problem which will be mentioned later, the term  $o_1$  is proposed

$$o_1(k) = \Psi_1 \|S_1(\epsilon_1(k))\| \quad (3.11)$$

where  $\epsilon_1(k) = [\bar{x}_1(k+n-1), x_d(k+n)]$  and  $\Psi_1 = \|W_1(k)\|$ .

Adding and subtracting (3.11) into (3.10), one can easily derive

$$\begin{aligned} \xi_1(k+n) = & \phi_1(\bar{x}_n(k+n-1)) \left[ -W_1^T S_1(N_1(k)) - \sigma_1(k) + \alpha_1(k) \right. \\ & \left. + \xi_2(k+n-1) - x_d(k+n) + \Psi_1 \|S_1(\epsilon_1(k))\| - \Psi_1 \|S_1(\epsilon_1(k))\| \right]. \end{aligned} \quad (3.12)$$

In order to further simplify (3.12), the virtual controller is designed as

$$\alpha_1(k) = -\hat{\Psi}_1(k) \|S_1(\epsilon_1(k))\| + x_d(k+n) \quad (3.13)$$

where  $\hat{\Psi}_1(k) = \|\hat{W}_1(k)\|$  and  $\hat{W}_1$  is the estimation of  $W_1$ .

Substituting (3.13) into (3.12), we get

$$\begin{aligned} \xi_1(k+n) = & \phi_1(\bar{x}_n(k+n-1)) \left[ -W_1^T S_1(N_1(k)) - \sigma_1(k) + \xi_2(k+n-1) \right. \\ & \left. - \tilde{\Psi}_1(k) \|S_1(\epsilon_1(k))\| - \Psi_1 \|S_1(\epsilon_1(k))\| \right] \end{aligned} \quad (3.14)$$

where  $\tilde{\Psi}_1(k) = \hat{\Psi}_1(k) - \Psi_1$ .

Transforming (3.14) with the  $k+1$  time instant, one has

$$\xi_1(k+1) = \phi_1(\bar{x}_n(k)) \left[ -W_1^T S_1(N_1(k_1)) - \sigma_1(k_1) + \xi_2(k) - \tilde{\Psi}_1(k_1) \|S_1(\epsilon_1(k_1))\| - \Psi_1 \|S_1(\epsilon_1(k_1))\| \right] \quad (3.15)$$

where  $k_1 = k+n-1$  represents the time instant.

On the basis of the RL control scheme, the strategic utility function can be defined as

$$E_1(k) = \hat{\Psi}_1(k_1) \|S_1(\epsilon_1(k_1))\| + (\hat{M}(k) - M_d(k)) \quad (3.16)$$

where  $M_d(k)$  represents the ideal strategic utility function and it is usually defined as "0" [37].

The cost function is derived as  $\beta_1(k) = (1/2)(E_1(k))^2$  and the gradient of  $\hat{\Psi}_1(k)$  is deduced as

$$\Delta \hat{\Psi}_1(k) = \frac{\partial \beta_1(k)}{\partial \hat{\Psi}_1(k_1)} = \|S_1(\epsilon_1(k_1))\| \left[ \hat{\Psi}_1(k) \|S_1(\epsilon_1(k_1))\| + \hat{M}(k) \right]. \quad (3.17)$$

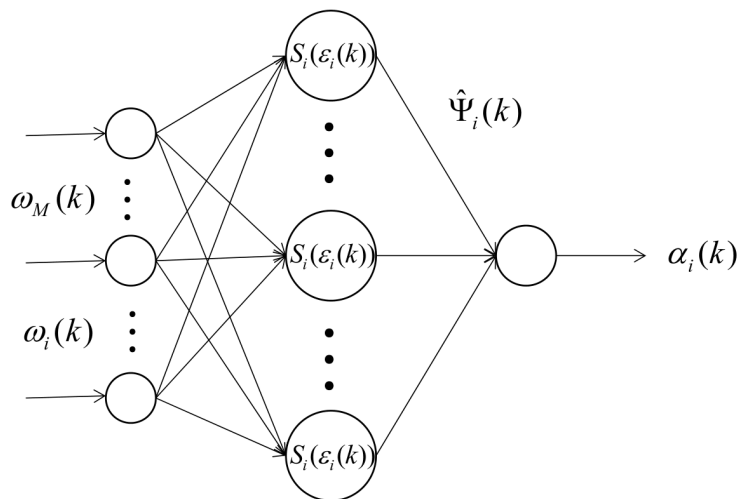
The multigradient can be further obtained as

$$g(l, \beta_1(k)) = \sum_{j=1}^l \|S_1(\epsilon_1(k_1 - j + 1))\| \left[ \omega_1(k_1 - j + 1) + \omega_M(k - j + 1) \right]. \quad (3.18)$$

Define  $\omega_1(k_1 - j + 1) = \hat{\Psi}_1(k_1) \|S_1(\epsilon_1(k_1 - j + 1))\|$ . Similar to step (3.8), the MGR updated law of  $\hat{\Psi}_1(k)$  is derived

$$\begin{aligned} \hat{\Psi}_1(k+1) &= \hat{\Psi}_1(k_1) - \mu_1 g(l, \beta_1(k)) \\ &= \hat{\Psi}_1(k_1) - \mu_1 \sum_{j=1}^l \|S_1(\epsilon_1(k_1 - j + 1))\| \left[ \omega_1(k_1 - j + 1) + \omega_M(k - j + 1) \right] \end{aligned} \quad (3.19)$$

where  $\mu_1$  stands for the chosen learning rate. The structure of the ANN is proposed in Figure 2.



**Figure 2.** Structure of the ANN.

*Remark 2:* It is necessary to emphasize that previous works usually designed the ANN basis function to have the form  $S_1(N_1(k))$ , which is not the function of  $x_1(k)$ . According to this design,  $\alpha_1(k)$  and  $\hat{\Psi}_1(k+1)$  are all built up as functions of  $N_1(k) = [\bar{x}_n(k+n-1), x_d(k+n)]^T$ . This results in the algebraic loop problem proposed in [38]. To settle this conundrum, the term  $\alpha_1(k)$  is presented in this paper and we adapt  $\alpha_1(k)$  and  $\hat{\Psi}_1(k+1)$  as functions of  $\epsilon_1(k) = [\bar{x}_1(k+n-1), x_d(k+n)]^T$ .

*Step i:* Define the tracking error  $\xi_i(k+n-i+1) = x_i(k+n-i+1) - \alpha_{i-1}(k)$  and  $\xi_{i+1}(k+n-i) = x_{i+1}(k+n-i) - \alpha_i(k)$ . Notation  $\alpha_{i-1}(k)$  and  $\alpha_i(k)$  indicate the virtual controller at Step  $i-1$  and Step  $i$ , respectively. Similar to the process in (3.9), one has

$$\begin{aligned} \xi_i(k+n-i+1) &= \phi_i(\bar{x}_n(k+n-i)) \left[ \left( \frac{\varphi_i(\bar{x}_n(k+n-i))}{\phi_i(\bar{x}_n(k+n-i))} - \frac{\alpha_{i-1}(k)}{\phi_i(\bar{x}_n(k+n-i))} + \alpha_{i-1}(k) \right) \right. \\ &\quad \left. + \alpha_i(k) + \xi_{i+1}(k+n-i) - \alpha_{i-1}(k) \right]. \end{aligned} \quad (3.20)$$



According to the definition of  $\gamma_1(k)$ , one has  $\gamma_i(k) = -\left(\frac{\varphi_i(\bar{x}_n(k+n-i))}{\phi_i(\bar{x}_n(k+n-i))} + \alpha_{i-1}(k) - \frac{\alpha_{i-1}(k)}{\phi_i(\bar{x}_n(k+n-i))}\right)$ . The unknown function can be approximated by the RBF NN which is given as  $\gamma_i(k) = W_i^T S_i(N_i(k)) + \sigma_i(k)$ , where  $W_i$  and  $\sigma_i(k)$  are defined as the ideal weight vector and the approximation error, respectively. Furthermore, we let  $N_i(k) = [x_1(k+n-i), \dots, x_n(k+n-i), x_d(k+n)]^T$ .

Substituting  $\gamma_i(k)$  into (3.20), one has

$$\xi_i(k+n-i+1) = \phi_i(\bar{x}_n(k+n-i)) \left[ -W_i^T S_i(N_i(k)) - \sigma_i(k) + \alpha_i(k) + \xi_{i+1}(k+n-i) - \alpha_{i-1}(k) \right]. \quad (3.21)$$

The term  $o_i(k)$  is given in the form below

$$o_i(k) = \Psi_i \|S_i(\epsilon_i(k))\| \quad (3.22)$$

where  $\epsilon_i(k) = [\bar{x}_i(x+n-i), x_d(k+n)]^T$  and  $\Psi_i$  denotes the Euclidean norm of the weight vector  $W_i$ .

Substituting (3.22) into (3.21) yields

$$\begin{aligned} \xi_i(k+n-i+1) = & \phi_i(\bar{x}_n(k+n-i)) \left[ -W_i^T S_i(N_i(k)) - \sigma_i(k) + \alpha_i(k) \right. \\ & \left. + \xi_{i+1}(k+n-1) - \alpha_{i-1}(k) + \Psi_i \|S_i(\epsilon_i(k))\| - \Psi_i \|S_i(\epsilon_i(k))\| \right]. \end{aligned} \quad (3.23)$$

The same as the previous process, the virtual controller is designed as

$$\alpha_i(k) = -\hat{\Psi}_i(k) \|S_i(\epsilon_i(k))\| + \alpha_{i-1}(k) \quad (3.24)$$

where  $\hat{W}_i(k)$  is the estimation of  $W_i$  and  $\hat{\Psi}_i(k) = \|\hat{W}_i(k)\|$ .

Substituting (3.24) into (3.23),  $\xi_i(k+n-i+1)$  expresses

$$\begin{aligned} \xi_i(k+n-i+1) = & \phi_i(\bar{x}_n(k+n-i)) \left[ -W_i^T S_i(N_i(k)) - \sigma_i(k) \right. \\ & \left. + \xi_{i+1}(k+n-1) - \tilde{\Psi}_i(k) \|S_i(\epsilon_i(k))\| - \Psi_i \|S_i(\epsilon_i(k))\| \right]. \end{aligned} \quad (3.25)$$

Resembling Step (3.15), (3.25) can be further described as

$$\xi_i(k+1) = \phi_i(\bar{x}_n(k)) \left[ -W_i^T S_i(N_i(k_i)) - \sigma_i(k_i) + \xi_{i+1}(k) - \tilde{\Psi}_i(k_i) \|S_i(\epsilon_i(k_i))\| - \Psi_i \|S_i(\epsilon_i(k_i))\| \right] \quad (3.26)$$

where  $k_i = k - n + i$ .

Let the prediction error  $E_i(k) = \hat{\Psi}_i(k) \|S_i(\epsilon_i(k_i))\| + \hat{M}(k)$ . According to  $E_i(k)$ , the cost function is described in its quadratic form  $\beta_i(k) = (1/2)(E_i(k))^2$ . The gradient of  $\hat{\Psi}_i$  is obtained according to the derivation:

$$\Delta \hat{\Psi}_i(k) = \frac{\partial \beta_i(k)}{\partial \hat{\Psi}_i(k_i)} = \|S_i(\epsilon_i(k_i))\| \left[ \hat{\Psi}_i(k) \|S_i(\epsilon_i(k_i))\| + \hat{M}(k) \right]. \quad (3.27)$$

On the basis of the MGR algorithm definition, the multigradient is expressed as

$$g(\iota, \beta_i(k)) = \sum_{j=1}^{\iota} \|S_i(\epsilon_i(k_i - j + 1))\| \left[ \omega_i(k_i - j + 1) + \omega_M(k - j + 1) \right]. \quad (3.28)$$

The updated law of  $\hat{\Psi}_i(k)$  is deduced according to (3.28):

$$\begin{aligned}\hat{\Psi}_i(k+1) &= \hat{\Psi}_i(k_i) - \mu_i g(\iota, \beta_i(k)) \\ &= \hat{\Psi}_i(k_i) - \mu_i \sum_{j=1}^l \|S_i(\epsilon_i(k_i - j + 1))\| \left[ \omega_i(k_1 - j + 1) + \omega_M(k - j + 1) \right]\end{aligned}\quad (3.29)$$

where  $\mu_i$  is the  $i$  step learning rate.

*Step n:* The tracking error in the  $n$ -th subsystem can be described as  $\xi_n(k+1) = x_n(k+1) - \alpha_{n-1}(k)$ . Substitute (2.2) and (2.6) into the  $n$ -th tracking error equation:

$$\xi_n(k+1) = \varphi_n(\bar{x}_n(k)) + \phi_n(\bar{x}_n(k)) \left( \psi(k) (b(k)v(k) + f(k)) + \delta(k) \right) + d(k) - \alpha_{n-1}(k). \quad (3.30)$$

For the purpose of simplifying 3.30,  $\pi(k)$  is defined

$$\pi(k) = - \frac{1}{\phi_n(\bar{x}_n(k))\psi(k)b(k)} \left( \varphi_n(\bar{x}_n(k)) - \alpha_{n-1}(k) \right). \quad (3.31)$$

Using the theory of the RBF NN to approximate (3.31), one gets

$$\pi(k) = W_n^T S_n(N_n(k)) + \sigma_n(k) \quad (3.32)$$

where the definitions of  $W_n$  and  $\sigma_n(k)$  are the same as those for the steps from 1 to  $n-1$  and  $N_n(k) = [\bar{x}_n(k), x_d(k+n)]^T$ .

Combining (3.32) and (3.29) we derive

$$\xi_n(k+1) = \phi_n(\bar{x}_n(k))\psi(k)b(k) \times \left( v(k) + \frac{f(k)}{b(k)} + \frac{\delta(k)}{\psi(k)b(k)} - \pi(k) \right) + d(k). \quad (3.33)$$

From (3.33) we can acquire the dynamics of the actuator fault and dead-zone shown in (2.3) and (2.6), separately. It is easy to deduce that they have the following properties

$$\frac{f(k)}{b(k)} \leq \frac{\bar{f}}{\underline{b}} = \tau \quad \frac{\delta(k)}{\psi(k)b(k)} \leq \frac{\bar{\delta}}{\underline{\psi}\underline{b}} = \vartheta \quad (3.34)$$

where  $\vartheta$  and  $\tau$  are both unknown parameters. Define the estimation of both parameters as  $\hat{\vartheta}$  and  $\hat{\tau}$ ; so the estimation error  $\tilde{\vartheta}$  and  $\tilde{\tau}$  are given by

$$\tilde{\vartheta}(k) = \hat{\vartheta}(k) - \vartheta \quad \tilde{\tau}(k) = \hat{\tau}(k) - \tau. \quad (3.35)$$

Based on estimation errors of (3.35), the actual controller is designed as

$$v(k) = \hat{\Psi}_n(k_n) \|S_n(N_n(k_n))\| + \hat{\tau}(k) + \hat{\vartheta}(k) \quad (3.36)$$

where  $\hat{\Psi}_n(k) = \|\hat{W}_n(k)\|$  and  $\hat{W}_n(k)$  stands for the estimation of  $W_n$ . The time index  $k_n = k$ .

With (3.35) and (3.36), the estimation fault (3.33) can be further written as

$$\begin{aligned}\xi_n(k+1) &= \phi_n(\bar{x}_n(k))\psi(k)b(k_n)\tilde{\Psi}_n(k) \|S_n(N_n(k_n))\| \phi_n(\bar{x}_n(k))\psi(k)b(k) \\ &\times \left( \tilde{\vartheta}(k) + \vartheta + \tilde{\tau}(k) + \tau + \frac{f(k)}{b(k)} + \frac{\delta(k)}{\psi(k)b(k)} \right) + \phi_n(\bar{x}_n(k))\psi(k)b(k)q(k_n) + d(k)\end{aligned}\quad (3.37)$$

where  $\tilde{\Psi}_n(k) = \hat{\Psi}_n(k) - \Psi_n$  and  $q(k) = \Psi_n \|S_n(N_n(k_n))\| - W_n^T S_n(N_n(k)) - \sigma_n(k)$ . Similar to Step  $i$ , the strategic utility function is defined

$$E_n(k) = \hat{\Psi}_n(k_n) \|S_n(N_n(k_n))\| + \hat{M}(k). \quad (3.38)$$

Further define the cost function as  $\beta_n(k) = (1/2)(E_n(k))^2$ ; the gradient of  $\hat{\Psi}_n(k)$  yields

$$\Delta \hat{\Psi}_n(k) = \frac{\partial \beta_n(k)}{\partial \hat{\Psi}_n(k_n)} = \|S_n(N_n(k_n))\| \left[ \hat{\Psi}_n(k) \|S_n(N_n(k_n))\| + \hat{M}(k) \right]. \quad (3.39)$$

The multigradient yields

$$g(\iota, \beta_n(k)) = \sum_{j=1}^{\iota} \|S_n(N_n(k_n - j + 1))\| \left[ \omega_n(k_n - j + 1) + \omega_M(k - j + 1) \right] \quad (3.40)$$

where  $\omega_n(k_n - j + 1) = \hat{\Psi}_n(k_n) \|S_n(N_n(k_n - j + 1))\|$ .

The weight updated law for  $\hat{\Psi}_n(k)$  can be further obtained as

$$\begin{aligned} \hat{\Psi}_n(k+1) &= \hat{\Psi}_n(k_n) - g(\iota, \beta_n(k)) \\ &= \hat{\Psi}_n(k_n) - \mu_n \sum_{j=1}^{\iota} \|S_n(N_n(k_n - j + 1))\| \left[ \omega_n(k_n - j + 1) + \omega_M(k - j + 1) \right] \end{aligned} \quad (3.41)$$

where  $\mu_n$  is a chosen positive learning rate. Let

$$E_\vartheta(k) = \hat{\vartheta}(k) + \hat{M}(k) \quad E_\tau(k) = \hat{\tau}(k) + \hat{M}(k). \quad (3.42)$$

The cost functions of two auxiliary signals are chosen to have a form that is the same as  $\beta_i(k)$  form

$$\beta_\vartheta(k) = (1/2)(E_\vartheta(k))^2 \quad \beta_\tau(k) = (1/2)(E_\tau(k))^2. \quad (3.43)$$

The gradients are deduced

$$\begin{aligned} \Delta \hat{\vartheta}(k) &= \frac{\partial \beta_\vartheta(k)}{\partial \hat{\vartheta}(k)} = \hat{\vartheta}(k) + \hat{M}(k) \\ \Delta \hat{\tau}(k) &= \frac{\partial \beta_\tau(k)}{\partial \hat{\tau}(k)} = \hat{\tau}(k) + \hat{M}(k). \end{aligned} \quad (3.44)$$

Two MGR updated laws are obtained

$$\begin{aligned} \hat{\vartheta}(k+1) &= \hat{\vartheta}(k) - \mu_\vartheta \sum_{j=1}^{\iota} \left( \hat{\vartheta}(k - j + 1) + \hat{M}(k - j + 1) \right) \\ \hat{\tau}(k+1) &= \hat{\tau}(k) - \mu_\tau \sum_{j=1}^{\iota} \left( \hat{\tau}(k - j + 1) + \hat{M}(k - j + 1) \right) \end{aligned} \quad (3.45)$$

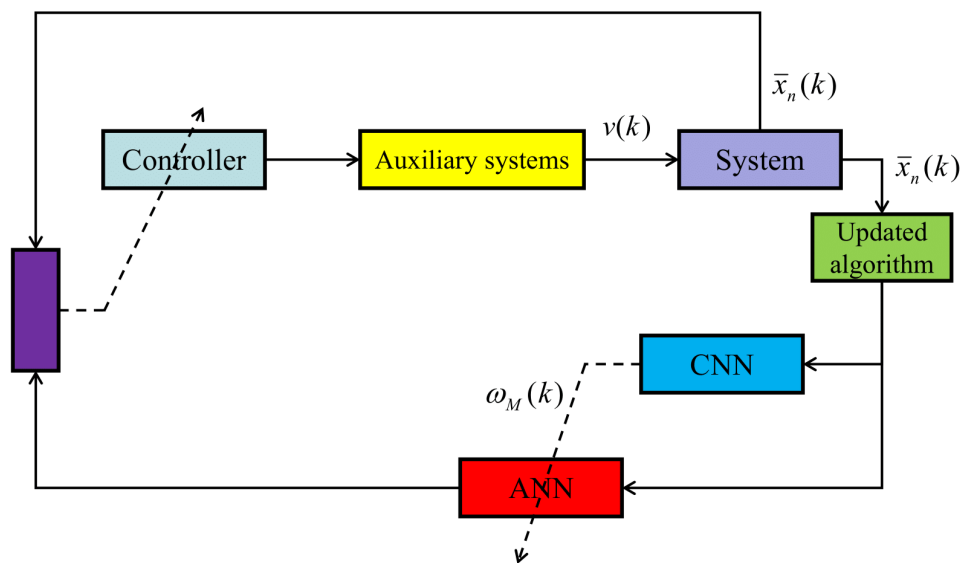
where  $\mu_\vartheta$  and  $\mu_\tau$  are positive learning factors.

Figure 3 shows the structure of the proposed control strategy. The analysis of the stability and tracking performance are mentioned in the following theorem.

*Theorem 1:* Consider the nonstrict feedback nonlinear system (2.2). The adaptive RL strategy includes the updated laws (3.8), (3.19), (3.29) and (3.41), the virtual controllers (3.13) and (3.24) and the actual controller (3.36). If the selected parameters that

$$0 < \mu_M < 1/(la^2), 0 < \mu_i < 1/l, 0 < \mu_\theta < 1/l, 0 < \mu_\tau < 1/l \quad (3.46)$$

and Assumption 1 holds, our control strategy can ensure that all signals are SGUUB and the tracking error is tolerated. The proof of Theorem 1 is shown in the appendix.



**Figure 3.** Control system structure for the proposed strategy.

#### 4. Simulation

In this section, some simulation results are presented to illustrate the effectiveness of the proposed approach.

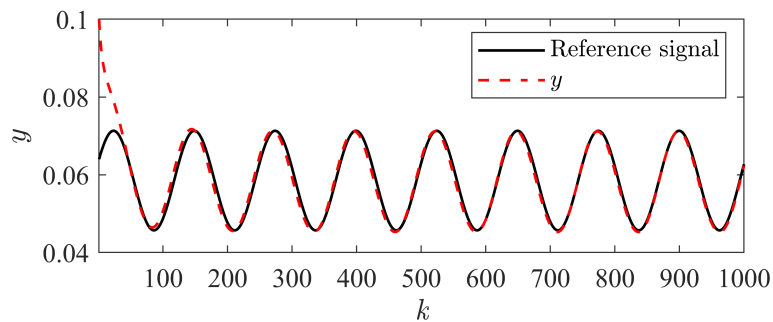
The nonstrict-feedback nonlinear discrete time system is chosen as

$$\begin{cases} x_1(k+1) = \varphi_1(\bar{x}_2(k)) + \phi_1(\bar{x}_2(k))x_2(k) \\ x_2(k+1) = \varphi_2(\bar{x}_2(k)) + \phi_2(\bar{x}_2(k))U(k) + d(k) \\ y(k) = x_1(k) \end{cases} \quad (4.1)$$

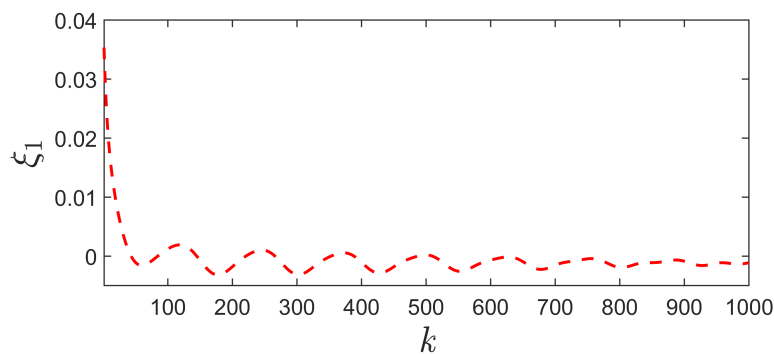
where  $x_1(k)$  and  $x_2(k)$  are the states,  $U(k)$  is the input and  $y(k)$  is the output. The functions  $\varphi_1(\bar{x}_n(k))$  and  $\varphi_2(\bar{x}_n(k))$  are chosen as  $x_1(k)$  and  $x_2(k)$ , respectively. We choose  $\phi_1(\bar{x}_n(k))$  and  $\phi_2(\bar{x}_n(k))$  as  $[-x_1(k) + 0.019(1.5 - x_1(k))\exp(4x_2(k)/(3.4 + x_2(k)))]/20$  and  $[-x_2(k) + 3.1(0.4 - x_1(k))\exp(1.5x_2(k)/(3.4 + x_2(k))) - 4(x_2(k) - U(k))]$ , respectively. The external disturbance  $d(k)$  is chosen as  $0.1\cos(0.05k)\cos(x_1(k))$ . The desired signal  $x_d(k)$  can be described as  $x_d(k) = 0.013\sin(\pi/8 + 0.6k\pi/38)$ .

The parameters can be selected as follows:  $\Psi_2(0) = 0.001$ ,  $\mu_M = 0.005$ ,  $\mu_1 = 0.008$ ,  $\mu_2 = 0.006$ ,  $\mu_\tau = 0.05$ ,  $\mu_\theta = 0.05$ ,  $a = 0.00001$ ,  $\hat{\tau}(0) = 0.2$  and  $\hat{\nu}(0) = 0.3$ . The hidden layer node numbers of the

ACNNs have been set as  $l_{W_1} = 10$ ,  $l_{W_2} = 10$  and  $l_{W_M} = 10$ . The pace of the multigradient is chosen as  $\iota = 10$ .



**Figure 4.** Tracking trajectory for the proposed scheme when  $b_r = 0.5$ ,  $b_l = -0.5$ ,  $f_r = 0.3$ , and  $f_l = -0.3$ .

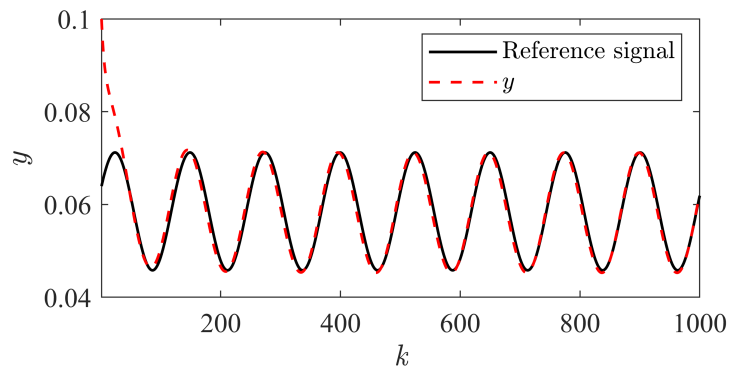


**Figure 5.** Tracking error for the proposed scheme when  $b_r = 0.5$ ,  $b_l = -0.5$ ,  $f_r = 0.3$ , and  $f_l = -0.3$ .

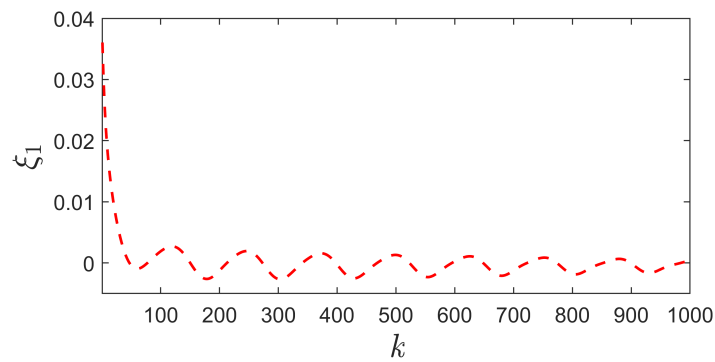
Figures 4 and 5 illustrate the trajectories of signals and the tracking error when  $b_r = 0.5$ ,  $b_l = -0.5$ ,  $f_r = 0.3$ , and  $f_l = -0.3$ , respectively. The control output achieved precise tracking of the reference signal and the tracking error was close to zero (almost 0.001). Figures 6 and 7 illustrate the trajectories of signals and the tracking error when  $b_r = 0.4$ ,  $b_l = -0.4$ ,  $f_r = 0.25$  and  $f_l = -0.25$ , respectively. The tracking performance in this scenario was also great (the tracking error was almost 0.002). According to our results, it is true that our fault-tolerant approach can offset the influence of dead zones with different parameters.

Figure 8 describes the trajectories of the ANN weight, the CNN weight, and the control input. We make a comparison between the proposed scheme and the MLP-based strategy. According to the results, it is clear that the proposed scheme achieved a faster convergence rate with the weight parameters than the ordinary MLP scheme. Figures 9 and 10 show the tracking trajectory and the tracking error without utilizing two auxiliary systems. Affected by the input dead zone and the actuator fault, tracking became extremely inaccurate and the tracking error was up to 0.01. Comparing Figures 4 and 9, it is obvious that our the approach can successfully offset the influence of the input dead zone and the actuator fault.

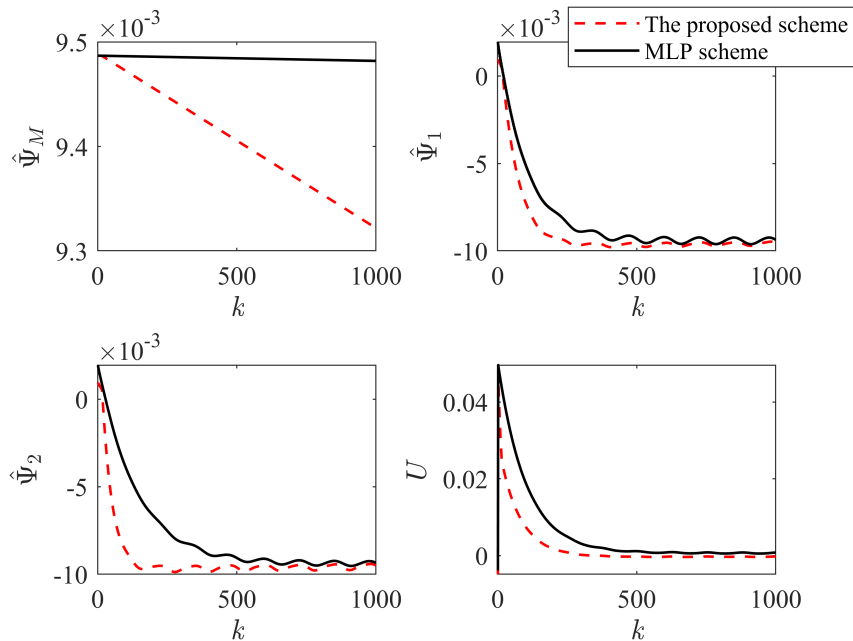
To verify our novel updated algorithm can reduce the computational burden, an experiment was



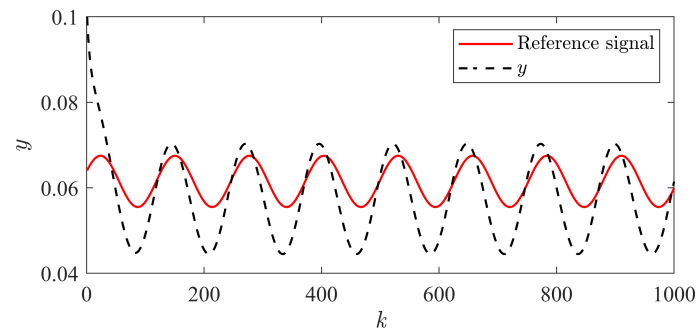
**Figure 6.** Tracking trajectory for the proposed scheme when  $b_r = 0.4$ ,  $b_l = -0.4$ ,  $f_r = 0.25$ , and  $f_l = -0.25$ .



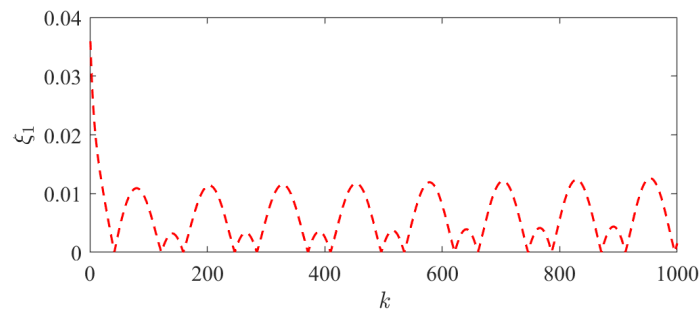
**Figure 7.** Tracking error utilizing proposed scheme when  $b_r = 0.4$ ,  $b_l = -0.4$ ,  $f_r = 0.25$ , and  $f_l = -0.25$ .



**Figure 8.** Comparison of the MLP-based control strategy and the proposed scheme.



**Figure 9.** Tracking performance for the scheme proposed in [21].



**Figure 10.** Tracking error for the scheme proposed in [21].

conducted as follows. We consider that the longer the computation the greater the computational burden. The total sample number was 1000. All of results were derived in the same environment by a computer with a 3.6 GHz CPU and 16 GB RAM.

**Table 1.** Simulation results.

approach	computational time/s
Gradient descent algorithm in [17]	0.5445
The proposed approach	0.2862

From Table 1, we can derive that the computational time of the ordinary gradient descent algorithm is 0.5445 s but the proposed updated algorithm only needs 0.2862 s. The computational time is reduced by 47.44%. Combined with the previous analysis, our approach not only alleviates the computational burden of the MGR algorithm but it also achieves a faster convergence rate for the weight parameters.

## 5. Conclusions

The aim of this paper was to build up a fault-tolerant controller for a class of nonstrict feedback systems with input dead zone. We proposed a novel neural network-updated algorithm to achieve a faster computational speed and eliminate the local optimal problem. Two auxiliary parameters were presented to offset the influence of the dead zone and actuator fault. To eliminate the occurrence of the algebraic loop problem, an auxiliary term was introduced to overcome this difficulty. According to the Lyapunov theory, all signals in the closed-loop system were proven to be SGUUB and the tracking error

converged to the neighborhood of zero. Finally, some simulation results were presented to illustrate the effectiveness of our approach.

There are still some other difficulties in the control area; for instance, the tracking control for stochastic systems via RL will be our future topic based on our current investigation.

## Acknowledgments

This work was supported by the Natural Science Foundation Project of Chongqing under Grant cstc2019jcyj-msxmX036, and also in part by the open funding project of the Guangdong Provincial Key Laboratory of Intelligent Decision and Coordination Control under Grant F2021098.

## Conflict of interest

The authors declare that they have no conflict of interest.

## References

1. J. B. Du, W. J. Cheng, G. Y. Lu, H. T. Gao, X. L. Chu, Z. C. Zhang, et al., Resource pricing and allocation in MEC enabled blockchain systems: An A3C deep reinforcement learning approach, *IEEE Trans. Network Sci. Eng.*, **9** (2022), 33–44. <https://10.1109/TNSE.2021.3068340>
2. H. X. Peng, X. M. Shen, Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks, *IEEE Trans. Network Sci. Eng.*, **7** (2021), 2416–2428. <https://10.1109/TNSE.2020.2978856>
3. D. C. Chen, X. L. Liu, W. W. Yu, Finite-time fuzzy adaptive consensus for heterogeneous nonlinear multi-agent systems, *IEEE Trans. Network Sci. Eng.*, **7** (2021), 3057–3066. <https://10.1109/TNSE.2020.3013528>
4. J. Wang, Q. Wang, H. Wu, T. Huang, Finite-time consensus and finite-time  $H_\infty$  consensus of multi-agent systems under directed topology, *IEEE Trans. Network Sci. Eng.*, **7** (2020), 1619–1632. <https://10.1109/TNSE.2019.2943023>
5. T. Gao, T. Li, Y. J. Liu, S. Tong, IBLF-based adaptive neural control of state-constrained uncertain stochastic nonlinear systems, *IEEE Trans. Neural Networks Learn. Syst.*, **33** (2022), 7345–7356. <https://10.1109/TNNLS.2021.3084820>
6. T. T. Gao, Y. J. Liu, D. P. Li, S. C. Tong, T. S. Li, Adaptive neural control using tangent time-varying BLFs for a class of uncertain stochastic nonlinear systems with full state constraints, *IEEE Trans. Cybern.*, **51** (2021), 1943–1953. <https://10.1109/TCYB.2019.2906118>
7. P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*, Ph. D. dissertation, Harvard University, 1974.
8. Y. Tang, D. D. Zhang, P. Shi, W. B. Zhang, F. Qian, Event-based formation control for nonlinear multiagent systems under DOS attacks, *IEEE Trans. Autom. Control.*, **66** (2021) 452–459. <https://10.1109/TAC.2020.2979936>
9. Y. Tang, X. T. Wu, P. Shi, F. Qian, Input-to-state stability for nonlinear systems with stochastic impulses, *Automatica*, **113** (2020), 0005–1098. <https://doi.org/10.1016/j.automatica.2019.108766>



10. X. T. Wu, Y. Tang, J. D. Cao, X. R. Mao, Stability analysis for continuous-time switched systems with stochastic switching signals, *IEEE Trans. Autom. Control.*, **63** (2018), 3083–3090. <https://10.1109/TAC.2017.2779882>
11. B. Kiumarsi, K. G. Vamvoudakis, H. Modares, F. L. Lewis, Optimal and autonomous control using reinforcement learning: A survey, *IEEE Trans. Neural Networks Learn. Syst.*, **29** (2018), 2042–2062. <https://10.1109/TNNLS.2017.2773458>
12. V. Narayanan, S. Jagannathan, Event-triggered distributed control of nonlinear interconnected systems using online reinforcement learning with exploration, *IEEE Trans. Cybern.*, **48** (2018), 2510–2519. <https://10.1109/TCYB.2017.2741342>
13. B. Luo, H. N. Wu, T. Huang, Off-policy reinforcement learning for  $H_\infty$  control design, *IEEE Trans. Cybern.*, **45** (2015), 65–76. <https://10.1109/TCYB.2014.2319577>
14. R. Song, F. L. Lewis, Q. Wei, Off-policy integral reinforcement learning method to solve nonlinear continuous-time multiplayer nonzero sum games, *IEEE Trans. Neural Networks Learn. Syst.*, **28** (2017), 704–713. <https://10.1109/TNNLS.2016.2582849>
15. X. Yang, D. Liu, B. Luo, C. Li, Data-based robust adaptive control for a class of unknown nonlinear constrained-input systems via integral reinforcement learning, *Inf. Sci.*, **369** (2016), 731–747. <https://doi.org/10.1016/j.ins.2016.07.051>
16. H. Zhang, K. Zhang, Y. Cai, J. Han, Adaptive fuzzy fault-tolerant tracking control for partially unknown systems with actuator faults via integral reinforcement learning method, *IEEE Trans. Fuzzy Syst.*, **27** (2019), 1986–1998. <https://10.1109/TFUZZ.2019.2893211>
17. W. Bai, Q. Zhou, T. Li, H. Li, Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation, *IEEE Trans. Cybern.*, **50** (2020), 3433–3443. <https://10.1109/TCYB.2019.2921057>
18. Y. Li, S. Tong, Adaptive neural networks decentralized FTC design for nonstrict-feedback nonlinear interconnected large-scale systems against actuator faults, *IEEE Trans. Neural Networks Learn. Syst.*, **28** (2017), 2541–2554. <https://10.1109/TNNLS.2016.2598580>
19. Q. Chen, H. Shi, M. Sun, Echo state network-based backstepping adaptive iterative learning control for strict-feedback systems: An error-tracking approach, *IEEE Trans. Cybern.*, **50** (2020), 3009–3022. <https://10.1109/TCYB.2019.2931877>
20. S. Tong, Y. Li, S. Sui, Adaptive fuzzy tracking control design for SISO uncertain nonstrict feedback nonlinear systems, *IEEE Trans. Fuzzy Syst.*, **24** (2016), 1441–1454. <https://10.1109/TFUZZ.2016.2540058>
21. W. Bai, T. Li, S. Tong, NN reinforcement learning adaptive control for a class of nonstrict-feedback discrete-time systems, *IEEE Trans. Cybern.*, **50** (2020), 4573–4584. <https://10.1109/TCYB.2020.2963849>
22. Y. Li, K. Sun, S. Tong, Observer-based adaptive fuzzy fault-tolerant optimal control for SISO nonlinear systems, *IEEE Trans. Cybern.*, **49** (2019), 649–661. <https://10.1109/TCYB.2017.2785801>
23. H. Modares, F. L. Lewis, M. B. Naghibi-Sistani, Integral reinforcement learning and experience replay for adaptive optimal control of partially-unknown constrained-input continuous-time systems, *Automatica*, **50** (2014), 193–202. <https://doi.org/10.1016/j.automatica.2013.09.043>

24. Z. Wang, L. Liu, Y. Wu, H. Zhang, Optimal fault-tolerant control for discrete-time nonlinear strict-feedback systems based on adaptive critic design, *IEEE Trans. Neural Networks Learn. Syst.*, **29** (2018), 2179–2191. <https://doi.org/10.1109/TNNLS.2018.2810138>
25. H. Li, Y. Wu, M. Chen, Adaptive fault-tolerant tracking control for discrete-time multiagent systems via reinforcement learning algorithm, *IEEE Trans. Cybern.*, **51** (2021), 1163–1174. <https://doi.org/10.1109/TCYB.2020.2982168>
26. Y. J. Liu, L. Tang, S. Tong, C. L. P. Chen, D. J. Li, Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems, *IEEE Trans. Neural Networks Learn. Syst.*, **26** (2015), 165–176. <https://doi.org/10.1109/TNNLS.2014.2360724>
27. W. Bai, T. Li, Y. Long, C. L. P. Chen, Event-triggered multigradient recursive reinforcement learning tracking control for multiagent systems, *IEEE Trans. Neural Networks Learn. Syst.*, **34** (2023), 355–379. <https://doi.org/10.1109/TNNLS.2021.3094901>
28. H. Wang, G. H. Yang, A finite frequency domain approach to fault detection for linear discrete-time systems, *Int. J. Control.*, **81** (2008), 1162–1171. <https://doi.org/10.1080/00207170701691513>
29. C. Tan, G. Tao, R. Qi, A discrete-time parameter estimation based adaptive actuator failure compensation control scheme, *Int. J. Control.*, **86** (2013), 276–289. <https://doi.org/10.1080/00207179.2012.723828>
30. J. Na, X. Ren, G. Herrmann, Z. Qiao, Adaptive neural dynamic surface control for servo systems with unknown dead-zone, *Control Eng. Pract.*, **19** (2011), 1328–1343. <https://doi.org/10.1016/j.conengprac.2011.07.005>
31. Y. J. Liu, S. Li, S. Tong, C. L. P. Chen, Adaptive reinforcement learning control based on neural approximation for nonlinear discrete-time systems with unknown nonaffine dead-zone input, *IEEE Trans. Neural Networks Learn. Syst.*, **30** (2019), 295–305. <https://doi.org/10.1109/TNNLS.2018.2844165>
32. S. S. Ge, J. Zhang, T. H. Lee, Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete time, *IEEE Trans. Syst., Man Cybern. B, Cybern.*, **34** (2004), 1630–1645. <https://doi.org/10.1109/TSMCB.2004.826827>
33. Y. J. Liu, Y. Gao, S. Tong, Y. Li, Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete time systems with dead-zone, *IEEE Trans. Fuzzy Syst.*, **24** (2016), 16–28. <https://doi.org/10.1109/TFUZZ.2015.2418000>
34. S. S. Ge, G. Y. Li, T. H. Lee, Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems, *Automatica*, **39** (2003), 807–819. [https://doi.org/10.1016/S0005-1098\(03\)00032-3](https://doi.org/10.1016/S0005-1098(03)00032-3)
35. Q. Yang, S. Jagannathan, Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators, *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, **42** (2012), 377–390. <https://doi.org/10.1109/TSMCB.2011.2166384>
36. Y. J. Liu, Y. Gao, S. Tong, Y. Li, Fuzzy approximation-based adaptive backstepping optimal control for a class of nonlinear discrete time systems with dead-zone, *IEEE Trans. Fuzzy Syst.*, **24** (2016), 16–28. <https://doi.org/10.1109/TFUZZ.2015.2418000>
37. S. Ferrari, J. E. Steck, R. Chandramohan, Adaptive feedback control by constrained approximate dynamic programming, *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, **38** (2008), 982–987. <https://doi.org/10.1109/TSMCB.2008.924140>

38. S. Tong, Y. Li, S. Sui, Adaptive fuzzy tracking control design for SISO uncertain nonstrict feedback nonlinear systems, *IEEE Trans. Fuzzy Syst.*, **24** (2016), 1441–1454. <https://10.1109/TFUZZ.2016.2540058>

## Appendix

*Step 1:* Defining  $\theta_{\xi_1}$ ,  $\theta_{\Psi_1}$  and  $\theta_M$  as positive constants. The Lyapunov function is chosen as

$$V_1(k) = V_{11}(k) + V_{12}(k) + V_{13}(k) + V_{14}(k) \quad (\text{A.1})$$

where  $V_{11}(k) = (\theta_{\xi_1}/4)\xi_1^2(k)$ ,  $V_{12}(k) = (\theta_{\Psi_1}/\mu_1) \sum_{s=0}^{n-1} \tilde{\Psi}_1(k_1 + i)^2$ ,  $V_{13}(k) = (\theta_M/\mu_M)\tilde{\Psi}_M(k)^2$  and  $V_{14}(k) = 2\theta_M \sum_{j=1}^l [\tilde{\Psi}_M(k)\|S_M(k-j)\|]^2$ .

The Cauchy-Schwarz inequality is expressed as

$$(a_1 + a_2 + \dots + a_n)^2 \leq n(a_1^2 + a_2^2 + \dots + a_n^2). \quad (\text{A.2})$$

Young's inequality is also given below

$$\tilde{a}^T \tilde{b} \leq (1/2)\tilde{a}^T \tilde{a} + (1/2)\tilde{b}^T \tilde{b} \quad (\text{A.3})$$

where  $\tilde{a}$  and  $\tilde{b}$  are arbitrary vectors.

By utilizing the inequality (A.3) and property that  $0 < S(N)^T S(N) < I$ , the two terms shown in (3.12) have the following quality

$$\begin{aligned} & -W_i^T S_i(N_i(k_i)) - \Psi_i \|S_i(\epsilon_i(k_i))\| \\ & \leq |W_i^T S_i(N_i(k_i))| + \Psi_i \|S_i(\epsilon_i(k_i))\| \\ & \leq \frac{1}{2} W_i^T W_i + \frac{1}{2} S_i(N_i(k_i))^T S_i(N_i(k_i)) + \frac{1}{2} \Psi_i^2 + \frac{1}{2} \|S_i(\epsilon_i(k_i))\|^2 \\ & \leq \tilde{\Psi}_i^2 + l \end{aligned} \quad (\text{A.4})$$

where  $i = 1, \dots, n-1$ .

According to (A.2) and (3.15), the first order difference of  $V_{11}(k)$  is described as

$$\Delta V_{11}(k) \leq \theta_{\xi_1} \bar{\phi}_1^2 (\tilde{\Psi}_1(k_1) \|S_1(\epsilon_1(k_1))\|)^2 + \theta_{\xi_1} \bar{\phi}_1^2 (\tilde{\Psi}_1^2 + l)^2 + \theta_{\xi_1} \bar{\phi}_1^2 \xi_2(k)^2 + \theta_{\xi_1} \bar{\phi}_1^2 \bar{\sigma}_1^2 - \frac{1}{4} \theta_{\xi_1} \xi_1^2(k). \quad (\text{A.5})$$

Based on (A.2) and (3.19), the first order difference of  $\Delta V_{12}(k)$  can be derived

$$\begin{aligned} \Delta V_{12}(k) & \leq -\theta_{\Psi_1} (1 - l\mu_1) \sum_{j=1}^l [\omega_1(k_1 - j + 1) + \omega_M(k - j + 1)]^2 + 2\theta_{\Psi_1} l (\tilde{\Psi}_1 + \tilde{\Psi}_M)^2 \\ & \quad + 2\theta_{\Psi_1} \sum_{j=1}^l [\tilde{\Psi}_M(k_1) \|S_M(k - j + 1)\|]^2 - \theta_{\Psi_1} \sum_{j=1}^l [\tilde{\Psi}_1(k_1) \|S_1(\epsilon_1(k_1 - j + 1))\|]^2. \end{aligned} \quad (\text{A.6})$$

Similar to the processes in (A.6), the first order difference of  $\Delta V_{13}(k)$  is calculated as

$$\begin{aligned} \Delta V_{13}(k) \leq & -\theta_M(1 - \iota\mu_M a^2) \sum_{j=1}^l \left[ a\omega_M(k-j+1) + \rho(k-j+1) - \omega_M(k-1) \right]^2 \\ & - \theta_M a^2 \sum_{j=1}^l \left[ \tilde{\Psi}_M(k_1) \|S_M(k-j+1)\| \right]^2 + 2\theta_M \sum_{j=1}^l \left[ \tilde{\Psi}_M(k_1) \|S_M(k-j)\| \right]^2 \\ & + 2\theta_M \iota \left[ \tilde{\Psi}_M(1+a) + 1 \right]^2. \end{aligned} \quad (\text{A.7})$$

Then, consider  $\Delta V_{14}(k)$ ; we can obtain

$$\Delta V_{14}(k) = 2\theta_M \sum_{j=1}^l \left[ \tilde{\Psi}_M(k) \|S_M(k-j+1)\| \right]^2 - 2\theta_M \sum_{j=1}^l \left[ \tilde{\Psi}_M(k) \|S_M(k-j)\| \right]^2. \quad (\text{A.8})$$

Combining (A.5)–(A.8), the first-order difference of  $\Delta V_1(k)$  is derived

$$\begin{aligned} \Delta V_1(k) \leq & -\theta_{\Psi_1}(1 - \iota\mu_1) \sum_{j=1}^l \left[ \omega_1(k_1-j+1) + \omega_M(k-j+1) \right]^2 \\ & - \theta_M(1 - \iota\mu_M a^2) \sum_{j=1}^l \left[ a\omega_M(k-j+1) + \rho(k-j+1) - \omega_M(k-1) \right]^2 \\ & - (\theta_{\Psi_1} - \theta_{\xi_1} \bar{\phi}_1^2) \left[ \tilde{\Psi}_1(k_1) \|S_1(\epsilon_1(k_1))\| \right]^2 + \theta_{\Psi_1} \bar{\phi}_1^2 \xi_2^2(k) \\ & - \left( \theta_M a^2 - 2\theta_{\Psi_1} - 2\theta_M \right) \sum_{j=1}^l \left[ \tilde{\Psi}_M(k) \|S_M(k-j+1)\| \right]^2 \\ & - \theta_{\Psi_1} \sum_{j=2}^l \left[ \tilde{\Psi}_1(k_1) \|S_1(\epsilon_1(k_1-j+1))\| \right]^2 - \frac{\theta_{\xi_1}}{4} \xi_1^2(k) + B_1 \end{aligned} \quad (\text{A.9})$$

where  $B_1 = 2\theta_{\Psi_1} \iota (\bar{W}_1 + \bar{W}_M)^2 + \theta_{\xi_1} \bar{\Psi}_1^2 \bar{\sigma}_1^2 + \theta_{\xi_1} \bar{\phi}_1^2 (\bar{\Psi}_1^2 + l)^2 + 2\theta_M \iota (\bar{\Psi}_M(1+a) + 1)^2$ .

*Step i:* The Lyapunov function in Steps 2 to  $n-1$  is designed as

$$V_i(k) = V_{i1}(k) + V_{i2}(k) \quad (\text{A.10})$$

where  $V_{i1}(k) = (\theta_{\xi_i}/4) \xi_i^2(k)$ ,  $V_{i2}(k) = (\theta_{\Psi_i}/\mu_i) \sum_{s=0}^{n-i} \tilde{\Psi}_i(k_i+s)^2$  and  $\theta_{\xi_i}$  and  $\theta_{\Psi_i}$  are both positive constants.

According to (3.26), the first order difference of  $V_{i1}$  can be deduced

$$\Delta V_{i1}(k) \leq \theta_{\xi_i} \bar{\phi}_i^2 \left[ \tilde{\Psi}_i(k_i) \|S_i(\epsilon_i(k_i))\| \right]^2 + \theta_{\xi_i} \bar{\phi}_i^2 \xi_{i+1}^2 + \theta_{\xi_i} \bar{\phi}_i^2 \bar{\sigma}_i^2 + \theta_{\xi_i} \bar{\phi}_i^2 (\bar{\Psi}_i^2 + l)^2 - \frac{1}{4} \theta_{\xi_i} \xi_i(k)^2. \quad (\text{A.11})$$

Similar to (A.6), one deduces  $\Delta V_{i2}(k)$  as

$$\begin{aligned} \Delta V_{i2}(k) \leq & -\theta_{\Psi_i}(1 - \iota\mu_i) \sum_{j=1}^l \left[ \omega_i(k_i-j+1) + \omega_M(k-j+1) \right]^2 + 2\theta_{\Psi_i} \iota (\bar{\Psi}_i + \bar{\Psi}_M)^2 \\ & + 2\theta_{\Psi_i} \sum_{j=1}^l \left[ \tilde{\Psi}_M(k) \|S_M(k-j+1)\| \right]^2 - \theta_{\Psi_i} \sum_{j=1}^l \left[ \tilde{\Psi}_i(k_i) \|S_i(\epsilon_i(k_i-j+1))\| \right]^2. \end{aligned} \quad (\text{A.12})$$

Combining (A.11) and (A.12),  $\Delta V_i(k)$  is derived, which shows that

$$\begin{aligned} \Delta V_i(k) \leq & -\theta_{\Psi_i}(1 - \iota\mu_i) \sum_{j=1}^l \left[ \omega_i(k_i - j + 1) + \omega_M(k - j + 1) \right]^2 - \frac{1}{4}\theta_{\xi_i}(k)\xi_i(k)^2 + \theta_{\xi_i}\bar{\phi}_i^2\xi_{i+1}^2 \\ & + 2\theta_{\Psi_i} \sum_{j=1}^l \left[ \tilde{\Psi}_M(k)\|S_M(k - j + 1)\| \right]^2 - \left( \theta_{\Psi_i} - \theta_{\xi_i}\bar{\phi}_i^2 \right) \left[ \tilde{\Psi}_i(k_i)\|S_i(\epsilon_i(k_i))\| \right]^2 \\ & - \theta_{\Psi_i} \sum_{j=2}^l \left[ \tilde{\Psi}_i(k_i)\|S_i(\epsilon_i(k_i - j + 1))\| \right]^2 + B_i \end{aligned} \quad (\text{A.13})$$

where  $B_i = \theta_{\xi_i}\bar{\phi}_i^2(\bar{\Psi}_i^2 + l)^2 + 2\theta_{\Psi_i}\iota(\bar{\Psi}_i + \bar{\Psi}_M)^2 + \theta_{\xi_i}\bar{\phi}_i^2\bar{\sigma}_i^2$ .

*Step n:* The Lyapunov function in the  $n$ -th step is

$$V_n(k) = V_{n1}(k) + V_{n2}(k) + V_{n3}(k) + V_{n4}(k) \quad (\text{A.14})$$

where  $V_{n1}(k) = (\theta_{\xi_n}/2)\xi_n^2(k)$ ,  $V_{n2}(k) = (\theta_{\Psi_n}/\mu_n)\bar{\Psi}_n^2(k)$ ,  $V_{n3}(k) = (\theta_{\vartheta}/\mu_{\vartheta})\bar{\vartheta}^2(k)$  and  $V_{n4}(k) = (\theta_{\tau}/\mu_{\tau})\bar{\tau}^2(k)$ .

Using the above inequality, (3.37) follows

$$\begin{aligned} \Delta V_{n1}(k) \leq & \frac{16}{3}\theta_{\xi_n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \left[ 2\tau^2 + \bar{\tau}(k)^2 + 2\vartheta^2 + \bar{\vartheta}(k)^2 \right. \\ & \left. + \left( \tilde{\Psi}_n(k)\|S_n(N_n(k_n))\| \right)^2 + \bar{q} \right] + \frac{2}{3}\theta_{\xi_n}\bar{d}^2 - (\theta_{\xi_n}/3)\xi_n(k)^2 \end{aligned} \quad (\text{A.15})$$

where  $q(k_n)^2 < (\bar{\sigma}_n + 2l^{1/2}\bar{\Psi}_n) = \bar{q}$ .

Based on (3.41) and (A.2),  $\Delta V_{n2}(k)$  is given as

$$\begin{aligned} \Delta V_{n2}(k) \leq & -\theta_{\Psi_n}(1 - \iota\mu_n) \sum_{j=1}^l \left[ \omega_n(k_n - j + 1) + \omega_M(k - j + 1) \right]^2 + 2\theta_{\Psi_n}\iota(\bar{\Psi}_n + \bar{\Psi}_M)^2 \\ & + 2\theta_{\Psi_n} \sum_{j=1}^l \left[ \tilde{\Psi}_M(k)\|S_M(k - j + 1)\| \right]^2 - \theta_{\Psi_n} \sum_{j=1}^l \left[ \tilde{\Psi}_n(k_n)\|S_n(\epsilon_n(k_n - j + 1))\| \right]^2. \end{aligned} \quad (\text{A.16})$$

Obviously, similar to (A.6), the first order difference of  $V_{n3}(k)$  and  $V_{n4}(k)$  is:

$$\begin{aligned} \Delta V_{n3}(k) \leq & -\theta_{\vartheta}(1 - \mu_{\vartheta}\iota) \sum_{j=1}^l \left[ \hat{\vartheta}(k - j + 1) + \omega_M(k - j + 1) \right]^2 - \theta_{\vartheta} \sum_{j=1}^l \bar{\vartheta}(k - j + 1)^2 \\ & + 2\theta_{\vartheta} \sum_{j=1}^l \left[ \tilde{\Psi}_M(k)\|S_M(k - j + 1)\| \right]^2 + 2\theta_{\vartheta}\iota(\vartheta + \bar{\Psi}_M)^2 \\ \Delta V_{n4}(k) \leq & -\theta_{\tau}(1 - \mu_{\tau}\iota) \sum_{j=1}^l \left[ \hat{\tau}(k - j + 1) + \omega_M(k - j + 1) \right]^2 - \theta_{\tau} \sum_{j=1}^l \bar{\tau}(k - j + 1)^2 \\ & + 2\theta_{\tau} \sum_{j=1}^l \left[ \tilde{\Psi}_M(k)\|S_M(k - j + 1)\| \right]^2 + 2\theta_{\tau}\iota(\tau + \bar{\Psi}_M)^2. \end{aligned} \quad (\text{A.17})$$

Combining (A.15)–(A.17), one has

$$\begin{aligned}
 \Delta V_n(k) &= V_{n1}(k) + V_{n2}(k) + V_{n3}(k) + V_{n4}(k) \\
 &\leq -\theta_{\Psi_n}(1 - \iota\mu_n) \sum_{j=1}^l \left[ \omega_n(k_n - j + 1) + \omega_M(k - j + 1) \right]^2 \\
 &\quad - \theta_{\vartheta}(1 - \mu_{\vartheta}\iota) \sum_{j=1}^l \left[ \hat{\vartheta}(k - j + 1) + \omega_M(k - j + 1) \right]^2 \\
 &\quad - \theta_{\tau}(1 - \mu_{\tau}\iota) \sum_{j=1}^l \left[ \hat{\tau}(k - j + 1) + \omega_M(k - j + 1) \right]^2 \\
 &\quad - \left( \theta_{\Psi_n} - \frac{16}{3}\theta_{\xi_n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \left[ \tilde{\Psi}_n(k) \|S_n(\epsilon_n(k_n))\| \right]^2 \\
 &\quad - \theta_{\xi_n} \sum_{j=2}^l \left[ \tilde{\Psi}_n(k) \|S_n(\epsilon_n(k_n - j + 1))\| \right]^2 - (\theta_{\xi_n}/3)\xi_n(k)^2 \\
 &\quad + \left( 2\theta_{\Psi_n} + 2\theta_{\vartheta} + 2\theta_{\tau} \right) \left[ \tilde{\Psi}_M(k) \|S_M(k - j + 1)\| \right]^2 \\
 &\quad - \left( \theta_{\vartheta} - \frac{16}{3}\theta_{\xi_n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \tilde{\vartheta}(k)^2 - \left( \theta_{\tau} - \frac{16}{3}\theta_{\xi_n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \tilde{\tau}(k)^2 \\
 &\quad - \theta_{\vartheta} \sum_{j=2}^l \tilde{\vartheta}(k - j + 1)^2 - \theta_{\tau} \sum_{j=2}^l \tilde{\tau}(k - j + 1)^2 + B_n
 \end{aligned} \tag{A.18}$$

where  $B_n = \frac{16}{3}\theta_{\xi_n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2(2\tau^2 + 2\vartheta^2 + \bar{q}) + \frac{2}{3}\theta_{\xi_n}\bar{d}^2 + 2\theta_{\tau}\iota(\tau + \bar{\Psi}_M)^2 + 2\theta_{\Psi_n}\iota(\bar{\Psi}_n + \bar{\Psi}_M)^2 + 2\theta_{\vartheta}\iota(\vartheta + \bar{\Psi}_M)^2$ .

Combining the Lyapunov function from Step 1 to Step  $n$ , we can obtain

$$\begin{aligned}
 V(k) &= \sum_{i=1}^n V_i(k) \\
 &= \sum_{i=1}^n \frac{\theta_{\Psi_i}}{\mu_i} \sum_{s=0}^{n-i} \tilde{\Psi}_i(k_i + s)^2 + \sum_{i=1}^{n-1} \frac{1}{4}\theta_{\xi_i}\xi_i^2(k) + \frac{1}{3}\theta_{\xi_n}\xi_n^2(k) + \frac{\theta_M}{\mu_M}\tilde{\Psi}_M(k)^2 \\
 &\quad + 2\theta_M \sum_{j=1}^l \Psi_M(k - j)^2 + (\theta_{\vartheta}/\mu_{\vartheta})\tilde{\vartheta}(k)^2 + (\theta_{\tau}/\mu_{\tau})\tilde{\tau}(k)^2.
 \end{aligned} \tag{A.19}$$

Combining (A.9), (A.13) and (A.18), we finally get

$$\begin{aligned}
\Delta V(k) \leq & -\left(\frac{\theta_{\xi i}}{3} - \theta_{\xi(n-1)}\bar{\phi}_{n-1}^2\right)\xi_n(k)^2 \\
& - \sum_{i=1}^n \theta_{\Psi_i}(1 - \iota\mu_i) \sum_{j=1}^l \left[ \omega_i(k_i - j + 1) + \omega_M(k - j + 1) \right]^2 \\
& - \theta_{\vartheta}(1 - \mu_{\vartheta}\iota) \sum_{j=1}^l \left[ \hat{\vartheta}(k - j + 1) - \omega_M(k - j + 1) \right] \\
& - \theta_{\tau}(1 - \mu_{\tau}\iota) \sum_{j=1}^l \left[ \hat{\tau}(k - j + 1) + \omega_M(k - j + 1) \right] \\
& - \theta_M(1 - \iota\mu_M a^2) \sum_{j=1}^l \left[ a\omega_M(k - j + 1) + \rho(k - j + 1) - \omega_M(k - 1) \right]^2 \\
& - \left( \theta_{\vartheta} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \tilde{\vartheta}(k)^2 - \left( \theta_{\tau} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \tilde{\tau}(k)^2 \\
& - \left( \theta_M a^2 - 2\theta_M - 2 \sum_{i=1}^n \theta_{\Psi_i} - 2\theta_{\vartheta} - 2\theta_{\tau} \right) \sum_{j=1}^l \left[ \tilde{\Psi}_M \| S_M(k - j + 1) \right]^2 \\
& - \sum_{i=2}^{n-1} \left( \frac{\theta_{\xi i}}{4} - \theta_{\xi(i-1)}\bar{\phi}_{i-1}^2 \right) \xi_i(k)^2 - \frac{1}{4}\theta_{\xi 1}\xi_1(k)^2 - \sum_{i=1}^n \theta_{\Psi_i} \sum_{j=2}^l \left[ \tilde{\Psi}_i(k_i) \| S_i(\epsilon_i(k_i - j + 1)) \right]^2 \\
& - \sum_{i=1}^{n-1} (\theta_{\Psi_i} - \theta_{\Psi_i}\bar{\phi}_i^2) \left[ \tilde{\Psi}_i(k_i) \| S_i(\epsilon_i(k_i)) \right]^2 - \left( \theta_{\Psi_n} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \left[ \tilde{\Psi}_n(k_n) \| S_n(\epsilon_n(k_n)) \right]^2 + B
\end{aligned} \tag{A.20}$$

where  $B = \sum_{i=1}^n B_i$ .

Selecting the parameters as  $0 < \mu_M < 1/(\iota a^2)$ ,  $0 < \mu_i < 1/\iota$ ,  $0 < \mu_{\vartheta} < 1/\iota$  and  $0 < \mu_{\tau} < 1/\iota$ . Thus, the first order difference of  $V(k)$  can be simplified as

$$\begin{aligned}
\Delta V(k) \leq & - \left( \theta_{\vartheta} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \tilde{\vartheta}(k)^2 - \left( \theta_{\tau} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \tilde{\tau}(k)^2 \\
& - \left( \theta_M a^2 - 2\theta_M - 2 \sum_{i=1}^n \theta_{\Psi_i} - 2\theta_{\vartheta} - 2\theta_{\tau} \right) \sum_{j=1}^l \left[ \tilde{\Psi}_M \| S_M(k - j + 1) \right]^2 \\
& - \frac{1}{4}\theta_{\xi 1}\xi_1(k)^2 + B - \sum_{i=2}^{n-1} \left( \frac{\theta_{\xi i}}{4} - \theta_{\xi(i-1)}\bar{\phi}_{i-1}^2 \right) \xi_i(k)^2 \\
& - \left( \frac{\theta_{\xi i}}{3} - \theta_{\xi(n-1)}\bar{\phi}_{n-1}^2 \right) \xi_n(k)^2 - \sum_{i=1}^n \theta_{\Psi_i} \sum_{j=2}^l \left[ \tilde{\Psi}_i(k_i) \| S_i(\epsilon_i(k_i - j + 1)) \right]^2 \\
& - \sum_{i=1}^{n-1} (\theta_{\Psi_i} - \theta_{\Psi_i}\bar{\phi}_i^2) \left[ \tilde{\Psi}_i(k_i) \| S_i(\epsilon_i(k_i)) \right]^2 - \left( \theta_{\Psi_n} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2 \right) \left[ \tilde{\Psi}_n(k_n) \| S_n(\epsilon_n(k_n)) \right]^2.
\end{aligned} \tag{A.21}$$

In this paper, the parameters  $\theta_{\Psi_i}$ ,  $\theta_{\Phi_n}$ ,  $\theta_{\xi_i}$ ,  $\theta_{\xi_n}$ ,  $\theta_M$ ,  $\theta_{\vartheta}$  and  $\theta_{\tau}$  are respectively designed as  $\theta_{\vartheta} > \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2$ ,  $\theta_{\tau} > \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2$ ,  $\theta_M > \frac{1}{a^2}(2\theta_M + 2 \sum_{i=1}^n \theta_{\Psi_i} + 2\theta_{\vartheta} + 2\theta_{\tau})$ ,  $\theta_{\Psi_i} > \theta_{\Psi_i}\bar{\phi}_i^2$ ,  $\theta_{\Psi_n} > \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2$ ,

$\theta_{\xi i} > 4\theta_{\xi(i-1)}\bar{\phi}_{i-1}^2$  and  $\theta_{\xi n} > 3\theta_{\xi(n-1)}\bar{\phi}_{n-1}^2$ . On the basis of the method, we have that  $\Delta V(k) < 0$  if the following inequalities hold

$$\begin{aligned}
 |\tilde{\theta}(k)| &> \frac{\sqrt{B}}{\sqrt{\theta_{\theta} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2}}, & |\tilde{\tau}(k)| &> \frac{\sqrt{B}}{\sqrt{\theta_{\tau} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2}}, \\
 \left| \sum_{j=1}^l (\tilde{\Psi}_M \|S_M(k-j+1)\|) \right| &> \frac{\sqrt{B}}{\sqrt{\theta_M a^2 - 2\theta_M - 2\sum_{i=1}^n \theta_{\Psi_i} - 2\theta_{\theta} - 2\theta_{\tau}}}, \\
 |\xi_1(k)| &> \frac{\sqrt{B}}{\sqrt{\frac{1}{4}\theta_{\xi 1}}}, & |\xi_i(k)| &> \frac{\sqrt{B}}{\sqrt{\frac{\theta_{\xi i}}{4} - \theta_{\xi(i-1)}\bar{\phi}_{i-1}^2}}, \\
 |\xi_n(k)| &> \frac{\sqrt{B}}{\sqrt{\frac{\theta_{\xi i}}{3} - \theta_{\xi(n-1)}\bar{\phi}_{n-1}^2}}, & \left| \sum_{j=2}^l (\tilde{\Psi}_i(k_i) \|S_i(\epsilon_i(k_i-j+1))\|) \right| &> \frac{\sqrt{B}}{\sqrt{\theta_{\Psi_i}}}, \\
 \left| \tilde{\Psi}_i(k_i) \|S_i(\epsilon_i(k_i))\| \right| &> \frac{\sqrt{B}}{\sqrt{\theta_{\Psi_i} - \theta_{\Psi_i}\bar{\phi}_i^2}}, \\
 \left| \tilde{\Psi}_n(k_n) \|S_n(\epsilon_n(k_n))\| \right| &> \frac{\sqrt{B}}{\sqrt{\theta_{\Psi_n} - \frac{16}{3}\theta_{\xi n}\bar{\phi}_n^2\bar{\psi}^2\bar{b}^2}}.
 \end{aligned} \tag{A.22}$$

In this way, all signals in the closed-loop system are proven to be SGUUB.



AIMS Press

© 2023 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>)