



# Estimator-based dynamic learning from neural control of discrete-time strict-feedback systems

Min Wang · Zheng Jiang · Haotian Shi

Received: 5 February 2023 / Accepted: 6 October 2023 / Published online: 7 November 2023  
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

**Abstract** The dynamic learning issue from adaptive neural control for a class of discrete-time strict-feedback nonlinear systems is the main topic of this paper. Different from the traditional control schemes, a new auxiliary error estimator is constructed in this paper to promote the solution of weight convergence. Subsequently, a new weight updating law is designed based on the estimation error rather than the conventional tracking error. Based on the variable substitution framework, a new adaptive neural control strategy is constructed to assure the stability of the considered system, neural accurate approximation of unknown dynamics as well as the exponential convergence of neural weights. Such convergent weights are shown and stored as constants, i.e., experience knowledge. In light of the experience knowledge, a static learning control strategy is constructed. Such a control strategy avoids time consumption caused by updating weights, facilitates the transient control performance and lessens space complexity. Simulations are fulfilled to demonstrate the availability of the presented strategy.

**Keywords** Learning control · Persistent excitation · Neural networks · Discrete-time nonlinear systems · Error estimator

## 1 Introduction

Nowadays, neural network (NN) is extensively adopted to tackle unknown dynamics on account of its omnipotent approximation and learning capabilities [1–3]. And adaptive neural control (ANC) is established for a major category of nonlinear systems satisfying matching conditions by means of the adaptive technology [4–7]. These schemes are effectively developed to more common nonlinear systems with different structures [9–14], including pure feedback and strict feedback, by employing the backstepping method [8]. Moreover, ANC has been also successfully applied to various industrial systems [15–18]. Notice that ANC is motivated from the learning ability of human beings. As such, ANC should have the ability: acquiring experience knowledge of a changing environment, then reusing the experience knowledge to facilitate their work efficiency [19]. However, the ANC methods mentioned above pay little attention to neural learning ability, thereby still requiring to online training neural networks for similar control objectives. The major cause lies in the great challenge on the testification of the persistent excitation (PE) condition for closed-loop nonlinear systems. Hereafter, the neural learning issue was gradually developed as reinforcement learning [20–

---

M. Wang (✉) · Z. Jiang  
School of Automation Science and Engineering, Guangdong Provincial Key Laboratory of Technique and Equipment for Macromolecular Advanced Manufacturing, South China University of Technology, Guangzhou 510641, China  
e-mail: auwangmin@scut.edu.cn

H. Shi  
School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China

22] and adaptive dynamic programming [23–25], in which the PE condition is often presupposed to be met. To remove this restriction, a dynamic learning (DL) approach has been presented in [26] to accomplish the neural learning ability for closed-loop unknown system dynamics based on any recurrent reference orbit. This method has been extensively developed in theory [27–31] and practical engineering applications [32–34].

Compared to plentiful results of the aforementioned nonlinear systems in continuous-time form, the discrete-time (DT) counterpart is relatively few even though digital technology has been generally applied in industrial production [35–40]. The main reason lies in the following challenges: (1) The difference of the DT Lyapunov function is nonlinear, which makes the stability analysis more severe; (2) the controller design may encounter noncausal problems for the strict-feedback form, and (3) the existing framework and stability results cannot achieve the experience knowledge learning of unknown dynamics in the system. Some exclusive efforts have been made to solve these challenges [41–45]. To be more specific, for strict-feedback nonlinear systems (SFNSs) in DT case, an  $n$ -step predictor technology has been cleverly proposed in [41] to work out tracking control. Such a technology has been further developed to pure-feedback systems [42,43] and multiinput-multioutput systems [44]. Based on a useful stability result of linear time-varying (LTV) systems in DT case subject to time delays, a novel DL strategy has been established in [46] to accomplish the learning control for DT SFNSs. It is worth pointing out that the  $n$ -step forward predictor technology brings about the  $n$ -step delay that exists in the NN weight updating law, which would lead to the complex weight storage, and the difficult knowledge reuse. Subsequently, a new ANC framework was proposed in [45] to avoid the time delay of weight updating laws by the variable substitution, instead of the  $n$ -step predictor. Unfortunately, this new framework cannot be directly used to solve neural DL because of the huge challenge on the exponentially convergent verification of neural weights. Particularly, unlike the continuous-time case, it is very difficult for DT SFNSs to construct the weight estimate error system as LTV system without delays, not to mention the exponential convergence of such a LTV system.

Inspired by the above discussion, this paper is to develop a new yet effective technology to handle the DL issue of SFNSs in DT form. To achieve such an

objective, we construct a new auxiliary error estimator in this paper, and then a novel weight updating law is constructed based on the variable substitution framework. The developed approach can ensure the expected control performance and the exponential convergence of weight estimates. Specially, every convergent weight can be stored as a constant, which is easily stored and reused. As a result, the presented learning strategy facilitates the transient tracking performance and reduces the time/space complexity. To further clarify, the following contributions are listed:

- (1) A new auxiliary error estimator is subtly designed to estimate the system tracking error by the constant characteristic of the ideal value of the estimated weight;
- (2) A novel neural weight update law is constructed by the estimator error, instead of the traditional tracking error, which not only avoids time delay [16,41,42,46,50], but also guarantees every weight converges exponentially to its unique ideal value;
- (3) A novel learning controller is designed based on the simple storage and usage rules of the converged weights, which achieves the good control performance with small overshoot, fast convergence and little computation burden.

The rest of the paper is organized as follows. Section 2 gives the problem formulation and preliminaries. In Sect. 3, a novel dynamic learning control strategy is designed for a class of DT SFNSs by a new error estimator, which guarantee the closed-loop stability, the exponential convergence of estimated NN weights and the precise identification of unknown system dynamics. And then, the learning constant weights are stored and reused to design a static neural learning controller, which achieves the improved tracking control performance with less space complexity, better transient performance and less energy consumption. These advantages are illustrated in Sect. 4 by a simulation comparison. The conclusions are shown in Sect. 5.

*Notation:* General notations are used throughout this paper. The estimated variable is denoted by  $\hat{(\cdot)}$ , and the estimation error is denoted by  $\tilde{(\cdot)}$ .  $I_m$  denotes the  $m \times m$  identity matrix. The  $m$ -dimensional Euclidean space is denoted by  $R^m$ . The recurrent orbit being made up of  $\Upsilon(k)$  is represented by  $\varphi(\Upsilon(k))$ .

## 2 Problem formulation

Consider the DT SFNSs as follows:

$$\begin{cases} \chi_i(k+1) = f_i(\bar{\chi}_i(k)) + g_i(\bar{\chi}_i(k))\chi_{i+1}(k) \\ \chi_n(k+1) = f_n(\bar{\chi}_n(k)) + g_n(\bar{\chi}_n(k))u(k) \\ y(k) = \chi_1(k) \end{cases} \quad (1)$$

where  $i = 1, 2, \dots, n - 1$ ,  $y(k) \in R$ ,  $\bar{\chi}_j(k) = [\chi_1(k), \dots, \chi_j(k)]^T \in R^j$  and  $u(k) \in R$  represent, severally, system output, state and input,  $f_j(\bar{\chi}_j(k)) \in R$  represent the unknown smooth nonlinear functions, and  $g_j(\bar{\chi}_j(k)) \in R$  stand for the known smooth functions, in which  $j = 1, 2, \dots, n$ .

**Assumption 1** [41]  $g_i(\bar{\chi}_i(k))$  satisfies  $0 < \underline{g}_i \leq |g_i(\bar{\chi}_i(k))| \leq \bar{g}_i$ , in which  $\underline{g}_i$  and  $\bar{g}_i$  are two constants with  $i = 1, 2, \dots, n$ .

Our main control objective is to construct a novel DL control strategy for DT SFNSs (1) with Assumption 1 such that: (1) the output signal  $y(k)$  can trace the recurrent tracking trajectory  $y_d(k)$ ; (2) NN weights can converge to their optimal values which is independent of the time sequence; (3) the converged weights can be easily expressed and reused for the amelioration of tracking control performance.

### 2.1 RBF NNs

#### 2.1.1 Omnipotent approximation ability

Radial basis function (RBF) NNs [47] are adopted to tackle the unknown function  $\varpi(\Upsilon(k))$  as

$$\varpi(\Upsilon(k)) = W^{*T} \Theta(\Upsilon(k)) + \epsilon(\Upsilon(k)), \forall \Upsilon(k) \in \Omega_\Upsilon \quad (2)$$

in which  $m > 1$  stands for NN node number,  $\Omega_\Upsilon$  denotes a compact set,  $W^* \in R^m$  stands for the desired weight vector, the DT NN input vector is denoted by  $\Upsilon(k) \subset R^q$ .  $\epsilon(\Upsilon(k))$  denotes the approximation error which satisfies  $|\epsilon(\Upsilon(k))| \leq \epsilon^*$ , in which  $\epsilon^* > 0$  denotes a tiny constant.  $\Theta(\Upsilon(k)) = [\Theta_1(\Upsilon(k)), \Theta_2(\Upsilon(k)), \dots, \Theta_m(\Upsilon(k))]^T \in R^m$  stands for the activation function vector with  $\Theta_i(\Upsilon(k)) = \exp[-(\Upsilon(k) - v_i)^T (\Upsilon(k) - v_i) / \ell_i^2]$ , where  $\ell_i$  and  $v_i = [v_{i1}, v_{i2}, \dots, v_{iq}]^T$  are, respectively, the width and the center of NN.

#### 2.1.2 Localized approximation ability

Reference [26] shows that the RBF NN owns the capacity of localized approximation. Particularly, for each recurrent signal  $\Upsilon(k) \in \Omega_\Upsilon$ , by employing the neurons near  $\Upsilon(k)$ , we have

$$\varpi(\Upsilon(k)) = W_\xi^{*T} \Theta_\xi(\Upsilon(k)) + \epsilon_\xi(\Upsilon(k)) \quad (3)$$

where  $W_\xi^* \in R^{m_\xi}$  with  $m_\xi < m$  stand for neural weights closing to the recurrent orbit  $\varphi(\Upsilon(k))$  which is composed of NN inputs  $\Upsilon(k)$ , the corresponding activation function and approximation error are, respectively,  $\Theta_\xi(\Upsilon(k)) = [\Theta_{1\xi}(\Upsilon(k)), \dots, \Theta_{j\xi}(\Upsilon(k))]^T \in R^{m_\xi}$ , and  $\epsilon_\xi(\Upsilon(k))$ . Due to the localized approximation ability, the term  $|\epsilon_\xi(\Upsilon(k))|$  is also arbitrarily small.

#### 2.1.3 PE and exponential stability

**Definition 1** [48] A DT sequence  $\Theta(\Upsilon(k))$  is regarded as PE if we can find positive constants  $k_0, k_1$  and  $k_2$  such that

$$\sum_{k=k_0}^{k_0+k_1-1} \Theta(\Upsilon(k))\Theta^T(\Upsilon(k)) \geq k_2 I. \quad (4)$$

**Lemma 1** [48] Suppose that the neural input  $\Upsilon(k)$  maintains in a compact set  $\Omega_\Upsilon$  and is recurrent. Then,  $\Theta_\xi(\Upsilon(k))$  meets the PE condition for the recurrent sub-vector  $\Theta_\xi(\Upsilon(k))$  being made up of the activation function vector  $\Theta(\Upsilon(k))$  near  $\varphi(\Upsilon(k))$ .

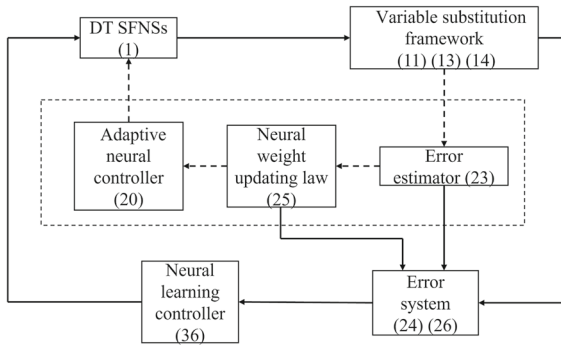
### 2.2 Exponential convergence of DT LTV systems

Consider the following DT LTV systems

$$\eta(k+1) = C(k)\eta(k) + \delta(k) \quad (5)$$

in which the system matrix is denoted by  $C(k) \in R^{m \times m}$ , the system state is denoted by  $\eta(k) \in R^m$ , and a bounded disturbance is  $\delta(k) \in R^m$ .

**Lemma 2** [48] Considering the LTV system in DT case (5), suppose that  $C(k) = I_n - \Gamma H(k)H^T(k)$ ,  $H^T(k)\Gamma H(k) < 2$ , and  $\|\delta(k)\|$  represents an adequately small value, where  $H(k) \in R^m$  stands for a time-varying vector, and  $\Gamma \in R^{m \times m}$  represents a constant positive definite symmetric matrix.  $\eta(k)$  can converge exponentially to a tiny vicinity around origin if  $H(k)$  satisfies the PE condition.



**Fig. 1** Overall scheme block diagram of the DL control strategy

### 3 DL control for DT strict-feedback system

A novel DL control strategy will be designed in this part for a class of DT SFNSs by using a new error estimator, so that NN weights can exponentially converge to a tiny vicinity of their desired values, and the unknown dynamics of the system can be precisely approximated by experienced neural networks with converged constant weights. To achieve such a goal and avoid the traditional causality contradiction, this paper firstly employs the variable substitution framework [45], and then the error estimator is cleverly designed to guarantee the exponential convergence of NN weights based on the exponential convergence theory of LTV systems in DT case. The overall scheme block diagram is shown in Fig. 1.

To begin with, define the following error variables:

$$z_i(k) = \chi_i(k) - \tau_{i-1}(k), \quad i = 1, 2, \dots, n \quad (6)$$

where  $y_d(k) \in R$  is the given reference trajectory,  $\tau_0(k) = y_d(k)$ , and  $\tau_i(k) \in R$  stands for the virtual controller.

Step 1: From (1) and (6), we have

$$\begin{aligned} z_1(k+1) &= f_1(\bar{\chi}_1(k)) + g_1(\bar{\chi}_1(k))\chi_2(k) - y_d(k+1) \\ &= g_1(\bar{\chi}_1(k)) \left[ \tau_1(k) + z_2(k) \right. \\ &\quad \left. + \frac{f_1(\bar{\chi}_1(k)) - y_d(k+1)}{g_1(\bar{\chi}_1(k))} \right]. \end{aligned} \quad (7)$$

The virtual controller  $\tau_1(k)$  is constructed as

$$\tau_1(k) = \rho_1(\bar{\chi}_1(k)) + \frac{y_d(k+1)}{\iota_1(\bar{\chi}_1(k))} \quad (8)$$

in which  $\rho_1(\bar{\chi}_1(k)) = -\frac{f_1(\bar{\chi}_1(k))}{g_1(\bar{\chi}_1(k))}$  and  $\iota_1(\bar{\chi}_1(k)) = g_1(\bar{\chi}_1(k))$ .

Substituting (8) into (7) yields

$$z_1(k+1) = g_1(\bar{\chi}_1(k))z_2(k). \quad (9)$$

Step  $i$  ( $2 \leq i \leq n-1$ ): It follows from (1) and (6) that

$$\begin{aligned} z_i(k+1) &= g_i(\bar{\chi}_i(k)) \left[ z_{i+1}(k) + \tau_i(k) \right. \\ &\quad \left. + \frac{f_i(\bar{\chi}_i(k))}{g_i(\bar{\chi}_i(k))} - \frac{\tau_{i-1}(k+1)}{g_i(\bar{\chi}_i(k))} \right]. \end{aligned} \quad (10)$$

Based on (1), we get that  $\chi_i(k+1)$  is a nonlinear function of  $\bar{\chi}_{i+1}(k)$ . Then, define the one-step predictor as

$$\chi_i(k+1) \doteq \vartheta_i(\bar{\chi}_{i+1}(k)), \quad i = 1, 2, \dots, n-1. \quad (11)$$

where  $\vartheta_i(\bar{\chi}_{i+1}(k)) = f_i(\bar{\chi}_i(k)) + g_i(\bar{\chi}_i(k))\chi_{i+1}(k)$ , and  $\bar{\chi}_{i+1}(k) = [\chi_1(k), \dots, \chi_j(k), \chi_{j+1}(k), \dots]^T \in R^{j+1}$ .

To show how to deal with the causal contradiction issue, the expression of  $\tau_1(k+1)$  is first given. According to (8) and (11), we have

$$\begin{aligned} \tau_1(k+1) &= \rho_1(\bar{\chi}_1(k+1)) + \frac{y_d(k+2)}{\iota_1(\bar{\chi}_1(k+1))} \\ &= \rho_1(\vartheta_1(\bar{\chi}_2(k))) + \frac{y_d(k+2)}{\iota_1(\vartheta_1(\bar{\chi}_2(k)))}. \end{aligned} \quad (12)$$

From (12), the problem of causal contradiction can be avoided by converting the future signal  $\chi_1(k+1)$  into a function of  $y_d(k+2)$  and  $\bar{\chi}_2(k)$ . Subsequently, by employing recursive design methods, it can be concluded that

$$\begin{aligned} \tau_{i-1}(k+1) &= \rho_{i-1}(\bar{\vartheta}_{i-1}(\bar{\chi}_i(k))) \\ &\quad + \frac{y_d(k+i)}{\iota_{i-1}(\bar{\vartheta}_{i-1}(\bar{\chi}_i(k)))} \end{aligned} \quad (13)$$

in which  $\bar{\vartheta}_{i-1}(\bar{\chi}_i(k)) = [\vartheta_1(\bar{\chi}_2(k)), \dots, \vartheta_{i-1}(\bar{\chi}_i(k))]^T \in R^{i-1}$ .

Based on (10) and (13), one can establish the virtual controller law  $\tau_i(k)$  as

$$\tau_i(k) = \rho_i(\bar{\chi}_i(k)) + \frac{y_d(k+i)}{\iota_i(\bar{\chi}_i(k))} \quad (14)$$

where

$$\begin{cases} \rho_i(\bar{\chi}_i(k)) = -\frac{f_i(\bar{\chi}_i(k))}{g_i(\bar{\chi}_i(k))} + \frac{\rho_{i-1}(\bar{\vartheta}_{i-1}(\bar{\chi}_i(k)))}{g_i(\bar{\chi}_i(k))} \\ \iota_i(\bar{\chi}_i(k)) = g_i(\bar{\chi}_i(k))\iota_{i-1}(\bar{\vartheta}_{i-1}(\bar{\chi}_i(k))). \end{cases} \quad (15)$$

By (10) and (14), we have

$$z_i(k+1) = g_i(\bar{\chi}_i(k))z_{i+1}(k). \quad (16)$$

Step  $n$ : Using the same analysis, we obtain

$$z_n(k + 1) = g_n(\bar{\chi}_n(k)) \times \left[ \frac{f_n(\bar{\chi}_n(k))}{g_n(\bar{\chi}_n(k))} - \frac{\tau_{n-1}(k + 1)}{g_n(\bar{\chi}_n(k))} + u(k) \right]. \tag{17}$$

From (6) and (14), we know that  $z_n(k)$  is unavailable because  $\tau_{n-1}(k)$  contains an unknown nonlinear function  $\rho_i(\bar{\chi}_i(k))$ . To construct an effective neural learning controller  $u(k)$ , the following recursive process is used in this paper by combining (9), (16) and (17)

$$z_1(k + n) = \iota(k + n - 1)[F(\Upsilon(k)) + u(k)] \tag{18}$$

where  $\iota(k + n - 1) = g_1(\bar{\chi}_1(k + n - 1))g_2(\bar{\chi}_2(k + n - 2)) \cdots g_n(\bar{\chi}_n(k)) \in R$  is the known smooth nonlinear function,  $\Upsilon(k) = [\chi_1(k), \chi_2(k), \dots, \chi_n(k), y_d(k + n)]^T \in R^{n+1}$ , and  $F(\Upsilon(k))$  is defined as

$$F(\Upsilon(k)) = -\rho_n(\bar{\chi}_n(k)) + \frac{y_d(k + n)}{\iota_n(\bar{\chi}_n(k))}$$

in which  $\rho_n(\bar{\chi}_n(k)) = -\frac{f_n(\bar{\chi}_n(k))}{g_n(\bar{\chi}_n(k))} + \frac{\rho_{n-1}(\bar{\vartheta}_{n-1}(\bar{\chi}_n(k)))}{g_n(\bar{\chi}_n(k))}$ ,  $\iota_n(\bar{\chi}_n(k)) = g_n(\bar{\chi}_n(k))\iota_{n-1}(\bar{\vartheta}_{n-1}(\bar{\chi}_n(k)))$ . It can be easily obtained from Assumption 1 that  $\iota(k + n - 1)$  is bounded.

Since  $F(\Upsilon(k))$  is unknown, then by adopting the RBF NN given in (2), one has

$$F(\Upsilon(k)) = W^{*T} \Theta(\Upsilon(k)) + \varepsilon(k). \tag{19}$$

Then, the NN controller is developed as

$$u(k) = -\hat{W}^T(k)\Theta(\Upsilon(k)) \tag{20}$$

in which  $\hat{W}(k)$  denotes the estimate of  $W^*$ . Substituting (20) into (18) yields

$$z_1(k + n) = \iota(k + n - 1)[- \tilde{W}^T(k)\Theta(\Upsilon(k)) + \varepsilon(k)] \tag{21}$$

where  $\tilde{W}(k) = \hat{W}(k) - W^*$ , which is the weight estimation error.

It can be seen from (21) that there exists  $n - 1$ -step delay between  $\tilde{W}(k)$  and  $z_1(k + n)$ . Such a phenomenon may cause the delay in the neural weight updating law, thereby preventing the exponential convergence of weight estimates. To solve such a problem, a new auxiliary error estimator is designed in this paper. Firstly, error dynamic equation (18) is rewritten as

$$z_1(k + 1) = \iota(k)[u(k_1) + F(\Upsilon(k_1))] \tag{22}$$

where  $k_1 = k - n + 1$ .

Subsequently, design the following error estimator

$$\hat{z}_1(k + 1) = \iota(k)[u(k_1) + \hat{W}^T(k)\Theta(\Upsilon(k_1))]. \tag{23}$$

Define the estimation error  $\tilde{z}_1(k) = \hat{z}_1(k) - z_1(k)$ . From (22) and (23), one has

$$\tilde{z}_1(k + 1) = \iota(k)[\tilde{W}^T(k)\Theta(\Upsilon(k_1)) - \varepsilon(k_1)]. \tag{24}$$

It is easily seen from (24), we have canceled the time delay between  $\tilde{z}_1(k + 1)$  and  $\tilde{W}(k)$ . Based on the auxiliary estimation error, the neural weight updating law is skillfully constructed as

$$\hat{W}(k + 1) = \hat{W}(k) - \Gamma\Theta(\Upsilon(k_1))\tilde{z}_1(k + 1) \tag{25}$$

where the design parameter  $\Gamma$  satisfies  $\Gamma = \Gamma^T > 0$ , and  $\lambda_{\max}(\Gamma) = r$ . Noting that  $\tilde{W}(k) = \hat{W}(k) - W^*$ , one has

$$\tilde{W}(k + 1) = \tilde{W}(k) - \Gamma\Theta(\Upsilon(k_1))\tilde{z}_1(k + 1). \tag{26}$$

*Remark 1* By using the constant characteristic of the ideal value of the estimated weight, a new auxiliary error estimator is subtly designed to estimate the system tracking error. Notice that two principles should be satisfied to achieve the exponential convergence of NN weights  $\hat{W}(k)$ : (1) the neural weight updating law is implementable; (2) the error system  $\tilde{W}(k + 1)$  can be converted into LTV system (5) in DT case. It is observed that  $z_n(k)$  is unavailable in this paper due to the unknown virtual control  $\tau_{n-1}$ . Although  $z_1(k)$  is available, there is no connection between  $z_1(k)$  and  $\tilde{W}(k)$ . As such, the use of  $z_1(k)$  cannot derive the LTV form of the weight estimate error system. To solve such a difficulty, auxiliary error estimator (23) is cleverly designed. Subsequently, estimation error system (24) is designed to construct the relationship between  $\tilde{z}_1(k + 1)$  and  $\tilde{W}(k)$ . Moreover, since  $\tilde{z}_1(k + 1)$  is available, this paper creatively uses  $\tilde{z}_1(k + 1)$ , instead of the traditional  $z_1(k + 1)$ , to design weight updating law  $\hat{W}(k + 1)$ , which makes two principles mentioned above be met. In addition, if we use  $\tilde{z}_1(k + 1)$  to design a neural weight updating law, each weight component of the estimated weight vector converges to one value instead of multiple values. Therefore, the convergent weights can be easily stored as constant weights.

**Theorem 1** Consider the closed-loop system being made up of SFNSs (1), actual control input (20), error estimator (23) as well as weight update law (25). Then, we can prove that: (1)  $z_1(k)$  converges to the desired tiny vicinity around origin; (2) the estimated weights



vector  $\hat{W}(k)$  converges exponentially to the tiny vicinity of the desired weights vector  $W^*$ ; (3) the unknown dynamics  $F(\Upsilon(k))$  is precisely approximated by the constant neural network  $\tilde{W}^T \Theta(\Upsilon(k))$  with the converged weights  $\tilde{W}$  expressed by

$$\tilde{W} = \text{mean}_{k \in [k_i, k_j]} \hat{W}(k) \tag{27}$$

where  $\text{mean}_{k \in [k_i, k_j]} \hat{W}(k) = \frac{1}{k_j - k_i + 1} \sum_{k=k_i}^{k_j} \hat{W}(k)$ , and  $[k_i, k_j]$  with  $k_j > k_i$  represents a time segment of steady-state process.

**Proof:** This proof will be divided into two parts. The first part will be used to complete the goal (1) of Theorem 1, and the goals (2) and (3) will be achieved in the second part.

(1) Let the Lyapunov function as

$$V(k) = \tilde{W}^T(k) \Gamma^{-1} \tilde{W}(k) + \frac{1}{\bar{\iota}} \tilde{z}_1^2(k) \tag{28}$$

in which  $\bar{\iota} > 0$  is the upper bound of  $\iota(k)$ . Subsequently, we have

$$\begin{aligned} \Delta V(k) &= \tilde{W}^T(k+1) \Gamma^{-1} \tilde{W}(k+1) - \tilde{W}^T(k) \Gamma^{-1} \\ &\quad \times \tilde{W}(k) + \frac{1}{\bar{\iota}} \tilde{z}_1^2(k+1) - \frac{1}{\bar{\iota}} \tilde{z}_1^2(k). \end{aligned} \tag{29}$$

For convenience, let  $\Theta(k) = \Theta(\Upsilon(k))$ . Substituting (26) into (29), we can get

$$\begin{aligned} \Delta V(k) &= \frac{1}{\bar{\iota}} \tilde{z}_1^2(k+1) - \frac{1}{\bar{\iota}} \tilde{z}_1^2(k) - 2\tilde{W}^T(k) \Theta(k_1) \tilde{z}_1 \\ &\quad (k+1) + \Theta^T(k_1) \Gamma \Theta(k_1) \tilde{z}_1^2(k+1) \\ &= \frac{1}{\bar{\iota}} \tilde{z}_1^2(k+1) - 2 \left[ \frac{\tilde{z}_1(k+1)}{\iota(k)} + \varepsilon(k_1) \right] \tilde{z}_1 \\ &\quad (k+1) - \frac{1}{\bar{\iota}} \tilde{z}_1^2(k) + \Theta^T(k_1) \Gamma \Theta(k_1) \tilde{z}_1^2(k+1) \\ &\leq -\frac{1}{\bar{\iota}} \tilde{z}_1^2(k+1) - \frac{1}{\bar{\iota}} \tilde{z}_1^2(k) - 2\varepsilon(k_1) \tilde{z}_1(k+1) \\ &\quad + \Theta^T(k_1) \Gamma \Theta(k_1) \tilde{z}_1^2(k+1). \end{aligned}$$

Based on this fact  $\|\Theta(k)\|^2 \leq l$  [49] and Young's inequality, one obtains

$$\begin{aligned} \Delta V(k) &\leq -\frac{1}{\bar{\iota}} \tilde{z}_1^2(k+1) - \frac{1}{\bar{\iota}} \tilde{z}_1^2(k) + \frac{r \tilde{z}_1^2(k+1)}{\bar{\iota}} \\ &\quad + \frac{\bar{\iota} \varepsilon^2(k_1)}{r} + r l \tilde{z}_1^2(k+1) \\ &= -\frac{1-r-r\bar{\iota}}{\bar{\iota}} \tilde{z}_1^2(k+1) - \frac{1}{\bar{\iota}} \tilde{z}_1^2(k) \\ &\quad + \frac{\bar{\iota} \varepsilon^2(k_1)}{r} \end{aligned}$$

where  $r$  is a positive constant. By selecting  $r < \frac{1}{1+\bar{\iota}}$  and using  $|\varepsilon(k)| \leq \bar{\varepsilon}$  yields

$$\Delta V(k) \leq -\frac{1}{\bar{\iota}} \tilde{z}_1^2(k) + \frac{\bar{\iota} \bar{\varepsilon}^2}{r}. \tag{30}$$

Formula (30) implies  $\Delta V(k) \leq 0$ , if

$$|\tilde{z}_1(k)| > \frac{\bar{\iota} \bar{\varepsilon}}{\sqrt{r}}. \tag{31}$$

From (30), it can be seen that for a small constant  $\mu_1 > \frac{\bar{\iota} \bar{\varepsilon}}{\sqrt{r}}$ , we can find a integer  $M > 0$ , so that for each  $k > M$ ,  $|\tilde{z}_1(k)| < \mu_1$  holds. Further, we have  $|\tilde{z}_1(k+1)| < \mu_1$  when  $k > M$ .

According to (24), we have that  $\tilde{W}^T(k) \Theta(k_1) = \frac{\tilde{z}_1(k+1)}{\iota(k)} + \varepsilon(k_1)$ . Because  $\iota(k)$  is bounded, there is a small constant  $\mu_2 = O_2(\bar{\varepsilon})$ , so that for each  $k > M$ ,  $|\tilde{W}^T(k) \Theta(k_1)| < \mu_2$ , i.e.,  $|\tilde{W}^T(k+n) \Theta(k)| < \mu_2$  holds.

By (26), we can get

$$\begin{aligned} \tilde{W}(k+n) &= \tilde{W}(k+n-1) - \Gamma \Theta(k) \tilde{z}_1(k+n) \\ &= \tilde{W}(k) - \Gamma \sum_{i=1}^n \Theta(k-n+i) \tilde{z}_1(k+i). \end{aligned} \tag{32}$$

When  $k > M$ ,  $\tilde{W}(k+n)$  is in the close vicinity of  $\tilde{W}(k)$ , because  $|\tilde{z}_1(k)| < \mu_1$ , for  $\forall k > M$ , where  $n$  is a finite real number. Then, according to  $|\tilde{W}^T(k+n) \Theta(k)| < \mu_2, \forall k > M$  and (21), we know that there is a tiny constant  $\mu_3 = O_3(\bar{\varepsilon})$ , so that for  $k > M$ ,  $|z_1(k+n)| < \mu_3$ . Hence,  $z_1(k)$  converges to a tiny vicinity of origin.

Based on (9), one has  $z_2(k+n-1) = \frac{1}{g_1(\bar{\chi}_1(k+n-1))} z_1(k+n)$ . Noting that  $g_1(\bar{\chi}_1(k+n-1))$  is bounded, there exists a small constant  $\mu_4 = O_4(\bar{\varepsilon})$ , so that for each  $k > M$ ,  $|z_2(k+n-1)| < \mu_4$  holds. As a result,  $z_2(k)$  is tiny small. Similarly, on the basis of (16), it is easy to conclude that  $z_i(k), i = 3, \dots, n$ , is also tiny small.

(2) In light of Theorem 1,  $z_1(k)$  at time  $M$  is extremely small. Since  $y_d(k)$  is recurrent and  $z_1(k) = \chi_1(k) - y_d(k)$ , it is obvious that  $\chi_1(k)$  is recurrent. By (1), one has  $\chi_2(k) = \frac{\chi_1(k+1) - f_1(\bar{\chi}_1(k))}{g_1(\bar{\chi}_1(k))}$ , which indicates that  $\chi_2(k)$  is recurrent. In the same way, from (1), it is not difficult to deduce that for  $k > M$ ,  $\chi_i(k)$  are also recurrent signals with  $i = 1, 2, \dots, n$ .

$\varphi(\Upsilon(k))$  is the trajectory of NN input variables  $\Upsilon(k)$  starting from time  $M$ . From Lemma 1, the recurrent sub-vector  $\Theta_\xi(\Upsilon(k))$  meets the PE condition along the

recurrent orbit  $\varphi(\Upsilon(k))$ . Moreover, in light of (26), we get

$$\tilde{W}_\xi(k+1) = A_\xi(k)\tilde{W}_\xi(k) + \delta_\xi(k) \tag{33}$$

where

$$A_\xi(k) = I_\xi - \Gamma l(k)\Theta_\xi(k_1)\Theta_\xi^T(k_1),$$

$$\delta_\xi(k) = \Gamma l(k)S_\xi^T(k_1)\varepsilon_\xi(k_1)$$

By choosing an appropriate parameter matrix  $\Gamma$ , we have  $\iota(k)\Theta_\xi(k_1)\Gamma\Theta_\xi^T(k_1) < 2I_\xi$  holds. Combining the small perturbation  $\delta_\xi(k)$  and the PE condition of  $\Theta_\xi(k_1)$ , we can prove from Lemma 2 that  $\tilde{W}_\xi(k)$  exponentially converges to a tiny vicinity around origin. This means that along  $\varphi(\Upsilon(k))$ ,  $\hat{W}_\xi(k)$  can converge exponentially to a tiny vicinity of the desired weight  $W^*$ . On the other hand, the weight estimates  $\hat{W}_\xi(k)$  distant from the recurrent trajectory, remain nearly unchanged due to the smaller value  $\Theta_\xi(k)$ . As such, when we select the initial values  $\hat{W}(0) = \dots = \hat{W}(n) = 0$ , the weight estimates  $\hat{W}_\xi(k)$  will remain at zero. Based on the above analysis, every weight estimate can converge to a constant, which is expressed as

$$\bar{W} = \text{mean}_{k \in [k_i, k_j]} \hat{W}(k) \tag{34}$$

where  $[k_i, k_j]$ ,  $k_j > k_i > M$  denotes the time quantum after the transient process.

Subsequently, along  $\varphi(\Upsilon(k))$ , we can obtain

$$\begin{aligned} F(\Upsilon(k)) &= \hat{W}^T(k)\Theta(\Upsilon(k)) + \varepsilon_1(k) \\ &= \bar{W}^T\Theta(\Upsilon(k)) + \varepsilon_2(k) \end{aligned} \tag{35}$$

where  $|\varepsilon_2(k)| < \varepsilon$  is very small.

*Remark 2* It should be pointed out that the traditional  $n$ -step predictor scheme is widely employed in the existing results [16,41,42,46,50] to convert system (1) into the  $n$ -step predictor system. Such a method results in weight updating law with  $n$ -step delays, i.e.,  $\hat{W}(k+1) = \hat{W}(k-n+1) - \Gamma\Theta(\Upsilon(k-n+1))z_1(k+1)$ , which may make  $\hat{W}(k)$  converge to multiple different values based on time sequence, and further increases the difficulty of knowledge storage and reuse [46]. To solve this problem, the auxiliary error estimator  $\hat{z}_1(k+1)$  in (23) is designed by replacing  $\hat{W}^T(k_1)\Theta(\Upsilon(k_1))$  with  $\hat{W}^T(k)\Theta(\Upsilon(k_1))$ . Subsequently, the weight updating law  $\hat{W}(k+1)$  in (25) is designed on the basis of the estimation error  $\tilde{z}_1(k+1)$  instead of the traditional tracking error  $z_1(k+1)$ , which is free from time delay. Based on the new weight updating law, it is easy to verify from Lemma 2 that  $\hat{W}(k)$

converges to a constant value  $\bar{W}$ , which reduces the space complexity and simplifies knowledge reuse.

By adopting the trained RBF NN  $\bar{W}^T\Theta(\Upsilon(k))$ , we can develop the following static neural controller

$$u(k) = -\bar{W}^T\Theta(\Upsilon(k)). \tag{36}$$

Controller (36) is employed to obtain the elevated tracking performance for the similar tracking objective.

**Corollary 1** Consider the closed-loop system being made up of SFNSs (1) and actual control input (36) constructed using the constant weights shown in (34). The tracking error can converge to the expected tiny vicinity around origin for the similar reference trajectory  $y_d(k)$ .

**Proof:** Let the Lyapunov function as:

$$V_1(k) = \frac{1}{\bar{l}^2}z_1^2(k) \tag{37}$$

By (22), (35), and (36), we get

$$\begin{aligned} \Delta V_1(k) &= \frac{1}{\bar{l}^2}z_1^2(k+1) - \frac{1}{\bar{l}^2}z_1^2(k) \\ &= -\frac{1}{\bar{l}^2}z_1^2(k) + \frac{1}{\bar{l}^2}l^2(k)\varepsilon_2^2(k_1) \\ &\leq -\frac{1}{\bar{l}^2}z_1^2(k) + \varepsilon^2. \end{aligned} \tag{38}$$

From (38), it can be seen that for a small constant  $\mu_5 > \frac{\varepsilon}{\bar{l}}$ , there exists a positive integer  $K_1$ , so that for any  $k > K_1$ ,  $|z_1(k)| < \mu_5$  holds. As a result, it is easy to demonstrate that  $z_1(k)$  converges to a tiny vicinity around origin.

*Remark 3* It should be pointed out that the variable substitution approach has been put forward in [45] to tackle the causality contradiction. By combining this approach and a new auxiliary error estimator, a novel weight updating law is constructed in our paper to guarantee the exponential convergence of NN weights. Compared with the existing result [45], the proposed scheme can fulfil knowledge acquirement and storage of NN weights and reuse the stored constant weights for the high-performance control with less space complexity, better transient performance and less energy consumption.

### 4 Simulation study

Consider the following DT SFNS [45]:

$$\begin{aligned} \chi_1(k+1) &= g_1(\chi_1(k))\chi_2(k) + f_1(\chi_1(k)) \\ \chi_2(k+1) &= g_2(\bar{\chi}_2(k))u(k) + f_2(\bar{\chi}_2(k)) \end{aligned} \tag{39}$$

**Table 1** Comparisons between PLC in [46] and the new LC

Simulation comparison	MAE	Computation time
LC	0.0125	3.0051s
PLC with correct time series	0.0209	5.3536s
PLC with incorrect time series	0.0221	5.5273s

“MAE” represents the mean absolute error

where  $f_1(\chi_1(k)) = \frac{\chi_1^2(k)}{\chi_1^2(k)+1}$ ,  $f_2(\bar{\chi}_2(k)) = \frac{0.2\chi_1(k)-0.6\chi_2(k)}{1+\chi_1^2(k)+\chi_2^2(k)} + \chi_2(k)$ ,  $g_1(\chi_1(k)) = 0.2 \sin(\chi_1(k)) + 0.5$ , and  $g_2(\bar{\chi}_2(k)) = 0.8 \cos(\chi_1(k)) + 1$  with  $\bar{\chi}_2(k) = [\chi_1(k), \chi_2(k)]^T$ .

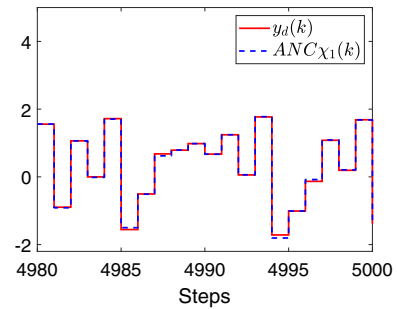
The tracking trajectory  $y_d(k)$  is derived from the following Henon system shown in (40):

$$\begin{aligned} \chi_{d,1}(k+1) &= \chi_{d,2}(k) \\ \chi_{d,2}(k+1) &= 1.4p - \frac{\chi_{d,2}^2(k)}{p} + 0.3\chi_{d,1}(k) \end{aligned} \tag{40}$$

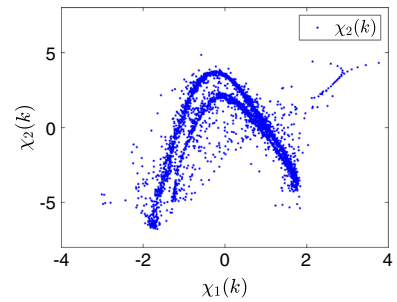
where  $y_d(k) = \chi_{d,1}(k)$ , and  $p = 1$ .

First, in the simulation, select initial states and design parameters as follows:  $\bar{\chi}_2(0) = [\chi_1(0), \chi_2(0)]^T = [0, 0]^T$ ,  $\chi_d(0) = [\chi_{d,1}(0), \chi_{d,2}(0)]^T = [0, 0]^T$ ,  $\hat{W}(0) = 0$ ,  $\Gamma = 0.3I$ . Moreover, the center  $v_i$  is equally spaced over  $[-2.1, 2.1] \times [-7, 4.5] \times [-2.1, 2.1]$ , and the corresponding widths  $\ell_i = [0.375, 0.625, 0.375]^T$ , the number of nodes is 5400. Besides,  $\bar{W}$  is designed as  $\bar{W} = \frac{1}{100} \sum_{k=4901}^{5000} \hat{W}(k)$ . Simulation results are displayed in Figs. 2, 3, 4, 5, 6, and 7. The tracking error is exhibited in Fig. 2 by using ANC strategy (20). Figures 3 and 4 display the recurrent feature of  $\chi_2(k)$  and the boundedness of controller  $u(k)$ , respectively. Figure 5 shows that  $\hat{W}(k)$  indeed converges exponentially to a tiny vicinity of their desired value. From Fig. 6, we know that the RBF NN can accurately approximate  $F(\Upsilon(k))$ . The simulation comparison of tracking performance is given in Fig. 7 between ANC (20) and static neural control (SNC) (36). By Fig. 7, we can see that the transient tracking performance is vastly improved by employing the stored experience weights  $\bar{W}$ .

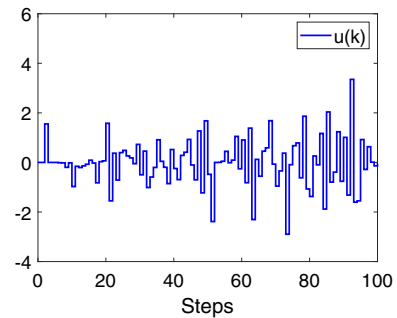
Then, simulation experiments are carried out compared with the existing schemes. For same DT SFNS (39) and the desired tracking signal in (40), simulation shows the results of previous learning control (PLC) scheme, which refers to the literature [46]. Such a simulation is to exhibit the advantages of the estimator-based leaning control (LC) scheme proposed in this paper by comparing it with PLC scheme [46].



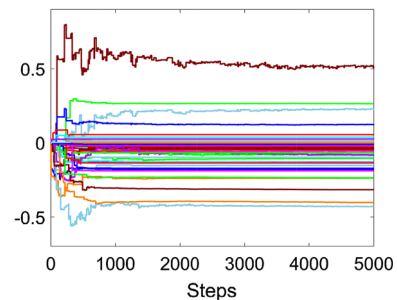
**Fig. 2** Tracking performance by using adaptive neural controller



**Fig. 3** System state  $\chi_2(k)$

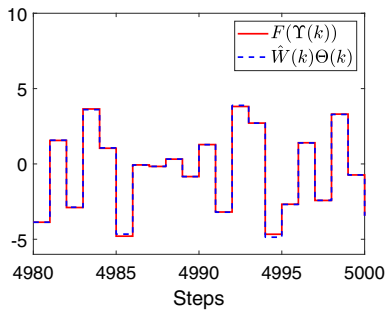


**Fig. 4** System control input  $u(k)$

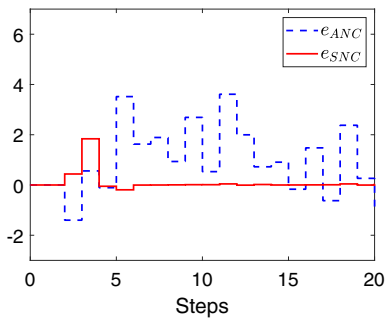


**Fig. 5** Partial NN weights convergence of  $\hat{W}(k)$

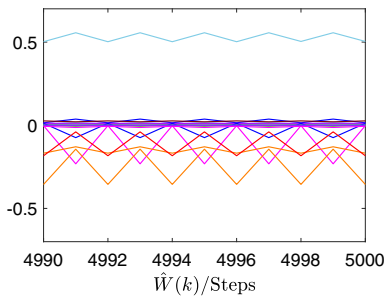




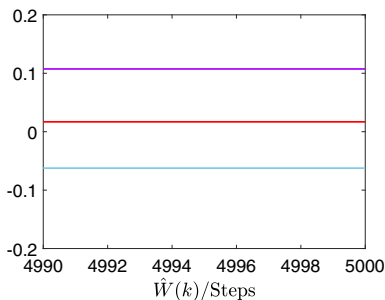
**Fig. 6** System dynamics  $F(\Upsilon(k))$  and RBF NN  $\hat{W}^T(k)\Theta(k)$



**Fig. 7** Tracking error comparison between ANC and SNC



**Fig. 8** Partial NN weights convergence of PLC in [46]



**Fig. 9** Partial NN weights convergence of LC

The constant NN weights are calculated as

$$\begin{aligned} \bar{W}_{\text{PLC},1} &= \frac{1}{50} \sum_{k^1=4991}^{4999} \hat{W}_{\text{PLC}}(k^1) \\ \bar{W}_{\text{PLC},2} &= \frac{1}{50} \sum_{k^2=4992}^{5000} \hat{W}_{\text{PLC}}(k^2) \end{aligned} \tag{41}$$

where time sequences  $k^1$  and  $k^2$  are chosen, respectively, as

$$\{k^1 | k^1 = 4991, 4993, 4995, \dots\}$$

and

$$\{k^2 | k^2 = 4992, 4994, 4996, \dots\}.$$

In the simulation, choose initial states and design parameters as follows:  $\bar{\chi}_{\text{PLC},2}(0) = [\chi_{\text{PLC},1}(0), \chi_{\text{PLC},2}(0)]^T = [0, 0]^T$ ,  $\chi_{d,\text{PLC}}(0) = [\chi_{d,\text{PLC}1}(0), \chi_{d,\text{PLC}2}(0)]^T = [0, 0]^T$ ,  $\hat{W}_{\text{PLC}}(0) = 0$ ,  $\Gamma_{\text{PLC}} = 0.2I$ . The centers  $v_{i,j}$ ,  $j = 1, 2$  of NNs are evenly spaced on  $[-3, 3] \times [-8, 5] \times [-3, 3]$  and  $[-3, 3] \times [-8, 5] \times [-8, 5]$ . The corresponding widths are

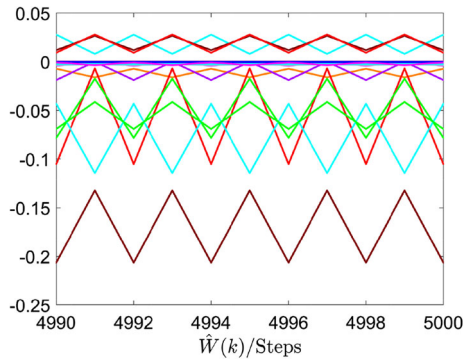
$$\ell_{i,j} = [0.625, 0.625, 0.625]^T, j = 1$$

and

$$\ell_{i,j} = [0.625, 0.625, 1]^T, j = 2.$$

And the numbers of nodes are 5239 and 6851, respectively. Compared to the PLC, it is demonstrated from Figs. 8 and 9 that in the LC scheme proposed in this paper, the NN weights converge to a fixed value. For the same NN weight, the LC scheme only needs to store one value, while the PLC scheme needs to store two values. Therefore, LC scheme reduces the overhead of system storage space, especially when the system order is higher, the cost saving is more obvious. Moreover, Table 1 shows that LC scheme reduces the computation time with smaller mean absolute error.

In addition, the scheme proposed in this paper is compared with that in the existing work [51]. The simulation parameters of PLC scheme in [51] are set to be consistent with the scheme in this paper, and the simulation results are shown in Fig. 10 and Table 2. Figure 10 shows that each weight of the NN weight vector in PLC scheme [51] converges to two values instead of one. According to Table 2, although the computation time of our LC scheme is slightly larger, its mean absolute error is reduced significantly. Moreover, for the same



**Fig. 10** Partial NN weights convergence of PLC in [51]

**Table 2** Comparisons between PLC in [51] and the new LC

Simulation comparison	MAE	Computation time
LC	0.0125	3.0051s
PLC with correct time series	0.0301	2.2450s
PLC with incorrect time series	0.0310	2.7515s

number of neural nodes, our LC scheme reduces the system storage space by 50%.

## 5 Conclusions

A new neural DL approach was proposed for a class of DT SFNSs in this article. By combining the variable substitution method and the new auxiliary error dynamic estimator, a novel NN updating law was proposed to guarantee that neural weight estimates exponentially converged to a tiny vicinity of their desired values, which avoids  $n$ -step delays of weight update and simplifies weight knowledge reuse. The converged constant weights were stored as experience knowledge and reemployed to establish a static LC strategy. Such a controller achieved the improvement of control performance due to the avoidance of online update of NN weights.

**Acknowledgements** This work was partially supported by the National Natural Science Foundation of China (Nos. 62273156 and 61890922) and the Guangdong Natural Science Foundation under Grants 2019B151502 058.

**Funding** The authors have not disclosed any funding.

**Data availability** Enquiries about data availability should be directed to the authors.

## Declarations

**Conflict of interest** The authors have not disclosed any competing interests.

## References

- Luo, C., Lei, H., Li, J., Zhou, C.: A new adaptive neural control scheme for hypersonic vehicle with actuators multiple constraints. *Nonlinear Dyn.* **100**, 3529–3553 (2020)
- Ding, Y., Fu, M., Luo, P., Wu, F.X.: Network learning for biomarker discovery. *Int. J. Netw. Dyn. Intell.* **2**, 51–65 (2023)
- Zhao, G., Li, Y., Xu, Q.: From emotion AI to cognitive AI. *Int. J. Netw. Dyn. Intell.* **1**, 65–72 (2022)
- Lewis, F.L., Jagannathan, S., Yesildirek, A.: *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor Francis, London, UK (1999)
- Huang, C., Liu, Z., Chen, C.L.P., Zhang, Y.: Adaptive neural asymptotic control for uncertain nonlinear multiagent systems with a fuzzy dead zone constraint. *Fuzzy Sets Syst.* **432**, 152–167 (2022)
- Liang, H., Liu, G., Zhang, H., Huang, T.: Neural-network-based event-triggered adaptive control of nonaffine nonlinear multiagent systems with dynamic uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(5), 2239–2250 (2021)
- Sun, X., Wang, G., Fan, Y.: Adaptive trajectory tracking control of vector propulsion unmanned surface vehicle with disturbances and input saturation. *Nonlinear Dyn.* **106**, 2277–2291 (2021)
- Krstic, M., Kanellakopoulos, I., Kokotovic, P.K.: *Nonlinear and Adaptive Control Design*. Wiley, New York (1995)
- Polycarpou, M.M.: Stable adaptive neural scheme for nonlinear systems. *IEEE Trans. Autom. Control* **41**, 447–451 (1996)
- Cao, L., Cheng, Z., Liu, Y., Li, H.: Event-based adaptive NN fixed-time cooperative formation for multiagent systems. *IEEE Trans. Neural Netw. Learn. Syst.* <https://doi.org/10.1109/TNNLS.2022.3210269>
- Zheng, X., Li, H., Ahn, C.K., Yao, D.: NN-based fixed-time attitude tracking control for multiple unmanned aerial vehicles with nonlinear faults. *IEEE Trans. Aerosp. Electron. Syst.* <https://doi.org/10.1109/TAES.2022.3205566>
- Wang, M., Wang, L., Huang, R., Yang, C.: Event-based disturbance compensation control for discrete-time SPMSM with mismatched disturbances. *Int. J. Syst. Sci.* **52**(4), 785–804 (2021)
- Huang, T., Li, T.: Attitude tracking control of a quadrotor UAV subject to external disturbance with L2 performance. *Nonlinear Dyn.* **111**, 10183–10200 (2023)
- Ding, J., Zhang, W.: Prescribed performance tracking control for high-order uncertain nonlinear systems. *Nonlinear Dyn.* **111**, 14199–14212 (2023)

15. Zhou, Q., Zhao, S., Li, H., Lu, R., Wu, C.: Adaptive neural network tracking control for robotic manipulators with dead zone. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 3611–3620 (2018)
16. Liu, Y.-J., Zeng, Q., Tong, S., Chen, C.L.P., Liu, L.: Adaptive neural network control for active suspension systems with time-varying vertical displacement and speed constraints. *IEEE Trans. Ind. Electron.* **66**, 9458–9466 (2019)
17. Ding, R., Ding, C., Xu, Y., Yang, X.: Neural-network-based adaptive robust precision motion control of linear motors with asymptotic tracking performance. *Nonlinear Dyn.* **108**, 1339–1356 (2022)
18. Abdelatti, M., Yuan, C.Z., Zeng, W., Wang, C.: Cooperative deterministic learning control for a group of homogeneous nonlinear uncertain robot manipulators. *Sci. China Inf. Sci.* **61**, 112201 (2018)
19. Fu, K.S.: Learning control systems and intelligent control systems: an intersection of artificial intelligence and automatic control. *IEEE Trans. Autom. Control* **16**, 70–72 (1971)
20. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. *J. Artif. Intell. Res.* **4**, 237–285 (1996)
21. Liu, L., Wang, Z.S., Zhang, H.G.: Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters. *IEEE Trans. Autom. Sci. Eng.* **14**, 299–313 (2017)
22. Guo, X., Bi, Z., Wang, J., Qin, S., Liu, S., Qi, L.: Reinforcement learning for disassembly system optimization problems: a survey. *Int. J. Netw. Dyn. Intell.* **2**, 1–14 (2023)
23. Zhang, H., Wei, Q., Liu, D.: An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. *Automatica* **47**, 207–214 (2011)
24. Wang, D., Liu, D., Mu, C., Zhang, Y.: Neural network learning and robust stabilization of nonlinear systems with dynamic uncertainties. *IEEE Trans. Neural Netw. Learn. Syst.* **29**, 1342–1351 (2017)
25. Wang, X., Sun, Y., Ding, D.: Adaptive dynamic programming for networked control systems under communication constraints: a survey of trends and techniques. *Int. J. Netw. Dyn. Intell.* **1**, 85–98 (2022)
26. Wang, C., Hill, D.J.: Learning from neural control. *IEEE Trans. Neural Netw.* **17**, 130–146 (2006)
27. Liu, T., Wang, C., Hill, D.J.: Learning from neural control of nonlinear systems in normal form. *Syst. Control Lett.* **58**, 633–638 (2009)
28. Dai, S.-L., Wang, C., Wang, M.: Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems. *IEEE Trans. Neural Netw. Learn. Syst.* **25**, 111–123 (2014)
29. Wang, M., Wang, C.: Neural learning control of pure-feedback nonlinear systems. *Nonlinear Dyn.* **79**, 2589–2608 (2015)
30. Wang, M., Wang, C.: Learning from adaptive neural dynamic surface control of strict-feedback systems. *IEEE Trans. Neural Netw. Learn. Syst.* **26**, 1247–1259 (2015)
31. Wang, Q., Wang, C., Sun, Q.: A model-based time-to-failure prediction scheme for nonlinear systems via deterministic learning. *J. Frankl. Inst.* **357**, 3771–3791 (2020)
32. Dai, S.-L., Wang, M., Wang, C.: Neural learning control of marine surface vessels with guaranteed transient tracking performance. *IEEE Trans. Ind. Electron.* **63**, 1717–1727 (2016)
33. Chen, T., Hill, D.J., Wang, C.: Distributed fast fault diagnosis for multimachine power systems via deterministic learning. *IEEE Trans. Ind. Electron.* **67**, 4152–4162 (2020)
34. Shi, H., Wang, M., Wang, C.: Pattern-based autonomous smooth switching control for constrained flexible joint manipulator. *Neurocomputing* **492**, 162–173 (2022)
35. Chen, W., Wang, Z., Ding, D., Dong, H.: Consensusability of discrete-time multi-agent systems under binary encoding with bit errors. *Automatica* **133**, 109867 (2021)
36. Possieri, C., Incremona, G.P., Calafiore, G.C., Ferrara, A.: An iterative data-driven linear quadratic method to solve nonlinear discrete-time tracking problems. *IEEE Trans. Autom. Control* **66**, 5514–5521 (2021)
37. Mu, C., Liao, K., Wang, K.: Event-triggered design for discrete-time nonlinear systems with control constraints. *Nonlinear Dyn.* **103**, 2645–2657 (2021)
38. Yu, L., Cui, Y., Liu, Y., Alotaibi, N.D., Alsaadi, F.E.: Sampled-based consensus of multi-agent systems with bounded distributed time-delays and dynamic quantisation effects. *Int. J. Syst. Sci.* **53**(11), 2390–2406 (2022)
39. Ji, D., Wang, C., Li, J., Dong, H.: A review: data driven-based fault diagnosis and RUL prediction of petroleum machinery and equipment. *Syst. Sci. Control Eng.* **9**(1), 724–747 (2021)
40. Wu, W., Hu, J., Zhang, F., Wang, C.: New results on rapid dynamical pattern recognition via deterministic learning from sampling sequences. *IEEE Trans. Neural Netw. Learn. Syst.* **59**(2), 1738–1748 (2023)
41. Ge, S.S., Li, G.Y., Lee, T.H.: Adaptive NN control for a class of strict-feedback discrete-time nonlinear systems. *Automatica* **39**, 807–819 (2003)
42. Ge, S.S., Yang, C., Lee, T.H.: Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time. *IEEE Trans. Neural Netw.* **19**, 1599–1614 (2008)
43. Wang, M., Shi, H., Wang, C., Fu, J.: Neural learning control for discrete-time nonlinear systems in pure-feedback form. *Sci. China Inf. Sci.* **65**(122206), 1–122206 (2022)
44. Ge, S.S., Zhang, J., Lee, T.H.: Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time. *IEEE Trans. Syst. Man Cybern. Syst.* **34**, 1630–1645 (2004)
45. Wang, M., Wang, Z., Dong, H., Han, Q.-L.: A novel framework for backstepping-based control of discrete-time strict-feedback nonlinear systems with multiplicative noises. *IEEE Trans. Autom. Control* **66**, 1484–1496 (2020)
46. Wang, M., Shi, H., Wang, C., Fu, J.: Dynamic learning from adaptive neural control for discrete-time strict-feedback systems. *IEEE Trans. Neural Netw. Learn. Syst.* **33**, 3700–3712 (2022)

47. Ding, D., Wang, Z., Han, Q.-L.: Neural-network-based outputfeedback control with stochastic communication protocols. *Automatica* **106**, 221–229 (2019)
48. Chen, W., Hua, S., Ge, S.S.: Consensus-based distributed cooperative learning control for a group of discrete-time nonlinear multi-agent systems using neural networks. *Automatica* **50**, 2254–2268 (2014)
49. Kurdila, A.J., Narcowich, F.J., Ward, J.D.: Persistency of excitation in identification using radial basis function approximants. *SIAM J. Control Optim.* **33**, 625–642 (1995)
50. Liu, Y.-J., Li, S., Tong, S., Chen, C.L.P.: Adaptive reinforcement learning control based on neural approximation for nonlinear discrete-time systems with unknown nonaffine dead-zone input. *IEEE Trans. Neural Netw. Learn. Syst.* **30**, 295–305 (2019)
51. Shi, H., Wang, M., Wang, C.: Leader-follower formation learning control of discrete-time nonlinear multiagent systems. *IEEE Trans. Cybern.* **53**, 1184–1194 (2023)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.