



Cooperative learning from adaptive neural control for a group of strict-feedback systems

Fei Gao¹ · Fengshan Bai¹ · Zhi Weng¹ · Xitai Na¹ · Jing Li²

Received: 15 July 2021 / Accepted: 29 March 2022 / Published online: 7 May 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

This paper investigates the distributed cooperative learning (DCL) from adaptive neural network (NN) control for a group of strict-feedback systems, where the structure of all strict-feedback systems is identical. In order to achieve DCL easily, the system transformation method is employed for the strict-feedback systems. For an agent, only one radial basis function NN is used to approximate the lumped uncertainty in control design. Then the output tracking performance of all strict-feedback systems is guaranteed. What's more, we prove that weights of all NNs in a multi-agent system converge to a small neighborhood around their common optimal value if the topology of the multi-agent system is connected and undirected. Thus, the approximation domain of all NNs is enlarged. Further, the previous learned NNs are used to improve the control performance. Finally, we provide two examples to demonstrate the effectiveness of the proposed scheme.

Keywords Strict-feedback systems · Neural networks · Distributed cooperative learning · System transformation

1 Introduction

Over the past two decades, the study on control of strict-feedback systems has received considerable attention (for instance, see [1–25]) due to such systems have wide applications, such as flexible joint robots, marine vehicles, and spacecrafts. In general, adaptive neural backstepping design as one of the most powerful techniques is often employed to address the control problem of strict-feedback systems, in which neural networks (NNs) are used to

approximate unknown nonlinear uncertainties over compact sets. However, in the early works [4, 5], to name just a few, the computational burden exponentially increases as the system order n grows, which is so-called the explosion of complexity. The main reason is that there exists the repeated differentiations of virtual controllers.

The problem of the explosion of complexity is overcome by proposing a dynamic surface control (DSC) technique [26]. At each backstepping step, a first-order filter is introduced to eliminate the repeated differentiation of virtual controllers and the related works on this technique are found in [27] and [28]. Combining the DSC technique and backstepping, adaptive neural control of nonlinear systems is extensively studied [18–25]. In [18], adaptive neural control design is presented for strict-feedback systems by applying the DSC technique. This work avoids the problem of differentiation of NNs in the traditional adaptive neural backstepping design. In [19], the authors study the robust adaptive NN tracking control by combining minimal learning parameter (MLP) methods and DSC, in which the number of parameters adjusted online is effectively reduced as well. In [21], a novel predictor-based neural DSC method is proposed, which systematically improves transient performance of closed-loop systems. Furthermore, the consideration of global adaptive neural DSC

✉ Fei Gao
mrgaofei@imu.edu.cn

Fengshan Bai
eefs@imu.edu.cn

Zhi Weng
wzhi@imu.edu.cn

Xitai Na
naxitai@163.com

Jing Li
xidianjing@126.com

¹ College of Electronic Information Engineering, Inner Mongolia University, Hohhot 010021, China

² School of Mathematics and Statistics, Xidian University, Xi'an 710071, China

[20], composite intelligent learning control [22], full state constraints [23], input saturation [24] and deterministic learning [25] is reported, respectively. Although the problem of explosion of complexity is eliminated in above studies, many approximators are still used in recursive backstepping design. The work in [29] proposes a new method that transforms the strict-feedback system into a normal form. Thus, the control problem of strict-feedback systems can be viewed as an output feedback control problem and only one NN is used to approximate the lumped uncertainty.

It should be noted that the stability analysis of the above works is mainly based on the Lyapunov methods. As approximators, the universal approximation ability of NNs is emphasized in adaptive neural control, but the learning ability is not revealed. NNs learn nothing after control process. As a result, the previously trained NNs cannot be reused to control the same or similar plants. In order to solve this problem, a deterministic learning mechanism is proposed in the pioneering work [30]. The authors prove that if the inputs of a radial basis function (RBF) NN are periodic or recurrent, then the regression vector satisfies the persistent excitation (PE) condition. Therefore, the RBF NN can store the knowledge after the control process and it can be reused without updating weights online. A lot of results on this mechanism are reported in recent years [25, 31–36]. In [25], with the help of the DSC technology, the authors study the learning problem of adaptive NN control of strict-feedback systems. Further, this mechanism is extended to affine systems [31], nonaffine systems [32] and pure-feedback nonlinear systems [33], respectively. In [34], dynamic learning with predefined control performance is investigated. In addition, several practical applications of the deterministic learning theory are explored in [34, 36], respectively.

Recently, inspired by the consensus theory of multi-agent systems, in [37], the authors propose a distributed cooperative learning (DCL) scheme and this approach enlarges the approximation domain of NNs compared with the deterministic learning mechanism. An important contribution in the work [37] is that the regression vectors of all agents in a multi-agent system satisfy the cooperative PE condition. Therefore, neural weights of all RBF NNs converge to a small neighborhood of their common optimal value. In [38], the output feedback control is considered for DCL if full states of plants are not measurable. With the help of the DCL scheme, in [39], the formation control is solved for a group of mechanical systems. It should be pointed out that the DCL scheme requires agents in multi-agent systems exchange the weight information with their neighboring agents. Thus, the DCL law with event-triggered communication is presented in [40] to overcome the continuous communication [37–39]. Although the DCL

scheme is widely studied in recent years, deciding how to achieve DCL for a group of high-order strict-feedback systems is not explored because there are some challenges: (i) the problem of explosion of complexity, (ii) it is difficult to verify whether the inputs of NNs are periodic, and (iii) the learning objective is difficult to be achieved.

Because strict-feedback systems exist widely in practice, it is necessary to study cooperative learning control of such systems to improve the control performance. In this paper, in order to achieve DCL easily, we transform a group of strict-feedback systems into the normal form. Due to only one RBF NN is used to approximate the lumped uncertainty in an agent, it is easy to verify the inputs of NNs are recurrent signals and easy to achieve cooperative learning. Further, control performance is improved and computational burden is reduced in experience control.

The main contributions of this paper are listed as follows:

- (i) The cooperative learning control is addressed for a group of strict-feedback systems. Most of works on adaptive neural control of strict-feedback systems only focus on control performance. However, in this paper, both control performance and NN approximation performance are considered.
- (ii) In contrast to the existing works [25, 34], the RBF NNs obtained in this paper have a large common approximation domain. Thus, the generalization ability of RBF NNs is improved.

The rest of this paper is organized as follows. In Sect. 2, some preliminary knowledge is provided. Section 3 presents problem formulation and the DCL control of strict-feedback systems. In Sect. 4, we discuss the NN control with past experience. Sect. 5 gives two examples to demonstrate the proposed scheme. Then the conclusions are reached in Sect. 6.

2 Preliminary

2.1 Kronecker product

Definition 1 [41]: Let $A \in R^{p \times q}$ and $B \in R^{m \times n}$. The Kronecker product of A and B is defined by

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1q}B \\ \vdots & \ddots & \vdots \\ a_{p1}B & \cdots & a_{pq}B \end{bmatrix} \in R^{pm \times qn}.$$

2.2 Graph theory

A graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V} = \{1, 2, \dots, N\}$ is a set of vertices, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges and the matrix \mathcal{A} is the adjacency matrix of the graph. The elements of \mathcal{A} are defined as $a_{ij} = 1$ in this paper if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. If for every $(j, i) \in \mathcal{E} \Leftrightarrow (i, j) \in \mathcal{E}$, the graph \mathcal{G} is called the undirected graph. If there is a path from i to j , then i and j are connected. If all pairs are connected, then \mathcal{G} is connected. The elements of the Laplacian matrix \mathcal{L} of a graph are defined as $l_{ii} = \sum_{j=1}^N a_{ij}$ and $l_{ij} = -a_{ij}$ if $i \neq j$. For an undirected graph, its Laplacian matrix \mathcal{L} is symmetric and positive semidefinite, and if $(j, i) \in \mathcal{E}$, it means that vertices i and j can exchange information. For a connected graph, only one of the eigenvalues of its Laplacian matrix \mathcal{L} is zero and the others are positive.

2.3 RBF NNs

In adaptive neural control, RBF NNs are used to approximate the continuous lumped uncertainties. Generally, for any continuous function $f(X)$, it can be approximated by an RBF NN over a compact set, which is described by

$$f(X) = S(X)^T W + \varepsilon(X), \tag{1}$$

where $X \in \Omega_X \subset R^q$ is the input vector of the RBF NN, $S(X) = [s_1(X), \dots, s_l(X)]^T$ is the regression vector, $W = [w_1, \dots, w_l]^T$ is the weight vector and $\varepsilon(X)$ represents the approximation error. $l > 1$ is the number of neurons and $s_j(X) = \exp\left[-\frac{\|(X-\theta_j)\|^2}{\tau^2}\right]$, $j \in \{1, \dots, l\}$, is the Gaussian function, where $\tau > 0$ and $\theta_j \in \Omega_X$ are the width and center of the Gaussian function, respectively. If l is sufficient large, then there exists a ideal weight W such that the approximation error $\varepsilon(X)$ can be made small enough [42, 43], i.e., $|\varepsilon(X)| < \varepsilon^*$, where ε^* is a small positive constant.

It is pointed out in [30] that along a bounded trajectory $\bar{X}(t) \subset \Omega_X$, the continuous function $f(\bar{X}(t))$ also can be approximated, which is described by

$$f(\bar{X}(t)) = S_{\zeta}(\bar{X}(t))^T W_{\zeta} + \varepsilon_{\zeta}(\bar{X}(t)), \tag{2}$$

where $(\cdot)_{\zeta}$ represents the region that is close to $\bar{X}(t)$, $W_{\zeta} = [w_{l_1}, \dots, w_{l_{\zeta}}]^T$ is a subvector of W with $l_{\zeta} < l$, $S_{\zeta}(\bar{X}(t)) = [s_{l_1}(\bar{X}(t)), \dots, s_{l_{\zeta}}(\bar{X}(t))]^T$ is a subvector of $S(\bar{X}(t))$, and $\varepsilon_{\zeta}(\bar{X}(t))$ is the new approximation error.

In a multi-agent system, assume that there are N agents (or N systems), we use $X_k(t)$ to denote the trajectory of the k th subsystem. Further, we use Ψ denote the union trajectory of all agents, i.e., $\Psi = X_1(t) \cup \dots \cup X_N(t) = \cup_{k=1}^N X_k(t)$. In [37], the cooperative PE condition is explored for a group of regression vectors of RBF NNs. This is a key property to achieve the convergence of NN weights.

Lemma 1 (Cooperative PE Condition of RBF NNs) [37]: Assume that $X_k(t), k = 1, \dots, N$ are periodic or recurrent. Let \mathcal{I} be a bounded μ -measurable subset of $[0, \infty)$ (take $\mathcal{I} = [t_0, t_0 + T_0]$), where T_0 is the period of Ψ . Then $S_{\zeta}(X_k(t)), k = 1, \dots, N$ satisfy the cooperative PE condition, that is, there exists a positive constant α such that

$$\int_{\mathcal{I}} \left[\sum_{k=1}^N S_{\zeta}(X_k(\tau)) S_{\zeta}(X_k(\tau))^T \right] d\mu(\tau) \geq \alpha I_{l_{\zeta}}, \tag{3}$$

where $(\cdot)_{\zeta}$ represents the region that is close to the trajectory $\Psi(t)$.

3 Cooperative learning from adaptive neural control

3.1 Problem formulation

Consider a group of strict-feedback systems

$$\begin{cases} \dot{x}_{k,i} = f_i(\bar{x}_{k,i}) + g_i(\bar{x}_{k,i})x_{k,i+1} & i = 1, \dots, n-1, \\ \dot{x}_{k,n} = f_n(\bar{x}_{k,n}) + g_n(\bar{x}_{k,n})u_k \\ y_k = x_{k,1}, & k = 1, \dots, N, \end{cases} \tag{4}$$

where N is the number of strict-feedback systems (or the number of agents), n is the order of each strict-feedback system, $\bar{x}_{k,n} = [x_{k,1}, \dots, x_{k,n}]^T \in R^n$ is the system state vector, $\bar{x}_{k,i} = [x_{k,1}, \dots, x_{k,i}]^T$, and y_k is the system output. Suppose that the full states of each system are measurable. $f_i(\dots)$ and $g_i(\dots)$ are unknown smooth nonlinear functions of $\bar{x}_{k,i}$. For further analysis, we need the following assumptions for systems (4).

Assumption 1 $g_i(\cdot), i = 1, \dots, n$, are bounded for all $\bar{x}_{k,n} \in \Omega$ and their signs are also known, where Ω is a compact set. Without losing generality, assume that $g_0 \leq g_i(\dots) \leq g_1$ for $i = 1, \dots, n$, where $0 < g_0 \leq g_1$ are positive constants.

Remark 1 Assumption 1 is standard and widely used in the literature, for example see [21, 25, 29, 34, 44]. It ensures

that control direction is known and system (4) is controllable. Assumption 1 is only used for stability and convergence analysis. The controller design is not required.

In this paper, each strict-feedback system, and its reference model serve as an agent in a multi-agent system. Next, we introduce the reference models.

Consider the following reference models

$$\begin{cases} \dot{y}_{k,i} = y_{k,i+1}, & i = 1, \dots, n - 1, \\ \dot{y}_{k,n} = f_{dk}(\bar{y}_{k,n}, t), \\ y_{dk} = y_{k,1} & k = 1, \dots, N, \end{cases} \quad (5)$$

where $\bar{y}_{k,n} = [y_{k,1}, \dots, y_{k,n}]^T$ is the state vector, $y_{dk} \in R$ is the output and $f_{dk}(\dots)$ is a known smooth function. For the k th reference model, we use φ_{dk} to denote its orbit starting from the initial state $\bar{y}_{k,n}(0)$. Let φ_d denote the union trajectory of all reference models, i.e., $\varphi_d = \cup_{k=1}^N \varphi_{dk}$.

Notice that for all the strict-feedback systems (4), their system functions $f_i(\dots)$ and $g_i(\dots)$ are identical. But their reference models are different. Therefore, the goals of this paper have three aspects: (i) guarantee the tracking performance and the stability of the closed-loop systems, (ii) approximate the lumped uncertainties by using the cooperative learning method, and what’s more, obtain a large common approximation domain in multi-agent systems compared with [25, 34], and (iii) design the controllers with the learned RBF NNs to improve control performance.

Remark 2 In a multiagent system, each agent can exchange information with its neighbouring agents. Thus, the agent can learn the knowledge not only from itself but also from its neighbouring agents by communication. This is called “cooperative learning”. The cooperative learning is originally proposed in [37]. For more details on cooperative learning, we refer the reader to [37].

3.2 Neural controller design and stability analysis

For any system of (4), most of adaptive neural control methods based on the traditional backstepping need to use n NNs to approximate lumped uncertainties. As pointed out in [25, 29] and [34], computational complexity will exponentially increase as the order n grows. To avoid this, the authors in [29] propose a system transformation method, i.e., (4) can be transformed into the following nonlinear systems in a normal form,

$$\begin{cases} \dot{z}_{k,i} = z_{k,i+1}, & i = 1, \dots, n - 1, \\ \dot{z}_{k,n} = F_n(\bar{x}_{k,n}) + G_n(\bar{x}_{k,n})u_k \\ y_k = z_{k,1}, & k = 1, \dots, N, \end{cases} \quad (6)$$

where

$$\begin{aligned} z_{k,1} &= y_k = x_{k,1}, \\ z_{k,i+1} &= \dot{z}_{k,i} = F_i(\bar{x}_{k,i}) + G_i(\bar{x}_{k,i})x_{k,i+1}, \quad i = 1, \dots, n - 1, \\ F_i(\bar{x}_{k,i}) &= \sum_{j=1}^{i-1} \left(\frac{\partial F_{i-1}}{\partial x_{k,j}} + \frac{\partial G_{i-1}}{\partial x_{k,j}} \right) x_{k,j} + G_{i-1}f_i(\bar{x}_i), \\ & \quad i = 2, \dots, n, \end{aligned}$$

and

$$\begin{aligned} G_i(\bar{x}_{k,i}) &= G_{i-1}(\bar{x}_{k,i-1})g_i(\bar{x}_{k,i}) = \prod_{j=1}^i g_j(\bar{x}_{k,j}), \\ & \quad i = 2, \dots, n. \end{aligned}$$

Remark 3 The above transformation requires that $f_i(\bar{x}_{k,i})$ and $g_i(\bar{x}_{k,i})$ in (4) are smooth functions of $\bar{x}_{k,i}$. This requirement is satisfied in many practical systems, such as robot manipulators [19] and unmanned surface vehicles [21, 44].

In what follows, we will control systems (6) such that y_k tracks the reference signal y_{dk} with a small error. Notice that the control target is the same as (4), which is not changed.

Although $z_{k,1}$ and the full states of (4) are available for measurement, $z_{k,i}, i = 2, \dots, n$, are not computable because the functions $F_i(\cdot), G_i(\cdot), i = 2, \dots, n$, are totally unknown. In this paper, we employ high-gain observers (HGOs) to estimate $z_{k,i}$ of (6).

Lemma 2 (HGO) [45, 46]: Consider the linear systems as follows

$$\begin{cases} \epsilon_k \dot{\zeta}_{k,1} = \zeta_{k,2}, \\ \vdots \\ \epsilon_k \dot{\zeta}_{k,n-1} = \zeta_{k,n}, \\ \epsilon_k \dot{\zeta}_{k,n} = -p_{k,1}\zeta_{k,n} - p_{k,2}\zeta_{k,n-1} - \dots - p_{k,n-1}\zeta_{k,2} \\ \quad - \zeta_{k,1} + y_k(t), \quad k = 1, \dots, N, \end{cases} \quad (7)$$

where ϵ_k is any small positive constant and $p_{k,1}, p_{k,2}, \dots, p_{k,n-1}$ are chosen such that the polynomial $s^n + p_{k,1}s^{n-1} + \dots + p_{k,n-1}s + 1$ is Hurwitz. Assume that the function

$y_k(t), k = 1, \dots, N$ and their first n derivatives are bounded. Then, there exist small positive constants $h_j, j = 2, \dots, n$, and T^* such that, for all $t > T^*$,

$$\frac{\zeta_{k,j+1}}{\epsilon_k^j} - y_k^{(j)} = -\epsilon_k \psi_k^{(j+1)} \quad j = 1, \dots, n - 1,$$

$$\left| \frac{\zeta_{k,j+1}}{\epsilon_k^j} - y_k^{(j)} \right| \leq \epsilon_k h_{j+1} \quad j = 1, \dots, n - 1,$$

where $\psi_k = \zeta_{k,n} + p_{k,1}\zeta_{k,n-1} + \dots + p_{k,n-1}\zeta_{k,1}$ and $\psi_k^{(j)}$ denotes the j th derivative of ψ_k , and $|\psi_k^{(j)}| \leq h_j$. \square

From (6), it can be seen that although the first n derivatives of $y_k(t)$ include the unknown continuous functions f_i, g_i, F_i and G_i , they are bounded if $\bar{x}_{k,n}$ is bounded. In the Theorem 1, which will be stated later, we prove that all signals are bounded. Therefore, $y_k(t)$ and its first n derivatives are bounded. By Lemma 2, we can use $\zeta_{k,j+1}/\epsilon_k^j$ to estimate $z_{k,i}$ of (6). Let $\hat{z}_{k,i}$ be the estimation of $z_{k,i}, i = 2, \dots, n$ and $\hat{z}_k = [\hat{z}_{k,1}, \dots, \hat{z}_{k,n}]^T$. Then,

$$\hat{z}_k = [\hat{z}_{k,1}, \dots, \hat{z}_{k,n}]^T = \left[z_{k,1}, \frac{\zeta_{k,2}}{1}, \frac{\zeta_{k,3}}{2}, \dots, \frac{\zeta_{k,n}}{n-1} \right]^T.$$

Based on [31, 34] and [38], we design the neural controllers of (6) as follows

$$u_k = -c_{k,n}e_{k,n} - S(Z_{k,n})^T \hat{W}_k, \quad k = 1, \dots, N, \tag{8}$$

where

$$e_{k,1} = z_{k,1} - y_{k,1}, \tag{9}$$

$$e_{k,j} = \hat{z}_{k,j} - \alpha_{k,j-1}, \quad j = 2, \dots, n \tag{10}$$

$$\alpha_{k,1} = y_{k,2} - c_{k,1}e_{k,1}, \tag{11}$$

$$\alpha_{k,j} = \dot{\alpha}_{k,j-1} - e_{k,j-1} - c_{k,j}e_{k,j}, \quad j = 2, \dots, n - 1, \tag{12}$$

with $c_{k,1}, \dots, c_{k,n} > 0$ being control gains. \hat{W}_k and $Z_{k,n}$ will be specified later.

Define the lumped uncertainty

$$H(Z_{k,n}) = \frac{F_n(\bar{x}_{k,n}) + \beta_k}{G_n(\bar{x}_{k,n})} \tag{13}$$

where $Z_{k,n} = [\bar{x}_{k,n}^T, \beta_k]^T$ and $\beta_k = e_{k,n-1} - \dot{\alpha}_{k,n-1}$. Since $H(Z_{k,n})$ is unknown, an RBF NN can approximate it, that is,

$$H(Z_{k,n}) = S^T(Z_{k,n})W + \varepsilon_k, \tag{14}$$

where $Z_{k,n}$ is the NN input, W is the common optimal weight and ε_k is the approximation error. In this paper, each agent only uses one RBF NN, and their width and centers of the Gaussian functions of all RBF NNs are identical. Thus, the dimensions of $S(\cdot)$ and W are identical for all agents although the inputs of $S(\cdot)$ are different. Therefore,

we define that W is the common optimal weight along the union orbit $Z = \cup_{k=1}^N Z_{k,n}$ rather than along the orbit $Z_{k,n}$.

Before the control process, the common optimal weight W is unknown. In control law (8), \hat{W}_k denotes the estimation of W in the k th agent (or system) and $S(Z_{k,n})^T \hat{W}_k$ is employed to approximate the lumped uncertainty $H(Z_{k,n})$ in the control process. From the above analysis, it can be seen that only one RBF NN is used in an agent and computational complexity is reduced effectively.

In order to achieve cooperative learning, design the NN weight update laws as follows

$$\dot{\hat{W}}_k = \rho \left(e_{k,n} S(Z_{k,n}) - \sigma_k \hat{W}_k \right) - \gamma \sum_{j \in \mathcal{N}_k} a_{kj} \left(\hat{W}_k - \hat{W}_j \right),$$

$$k = 1, \dots, N, \tag{15}$$

where $\rho, \sigma_k, \gamma > 0$ are the design parameters, $a_{kj} > 0$ means that the k th agent can receive the NN weight \hat{W}_j from the j th agent and $a_{kj} = 0$ means that there is no communication between the agents k and j . Based on the above analysis, one of the results is summarized as follows.

Theorem 1 Consider a multi-agent system consisting of systems (4) with Assumption 1, the reference models (5), the HGOs (7), the controllers (8) and weight update laws (15). Assume that the communication topology of the multi-agent system is undirected and connected. For any initial conditions $\bar{x}_{k,n}(0) \in \Omega_k$ (where Ω_k is a compact set), it holds that

- (i) All the signals in the multi-agent system are still bounded;
- (ii) The output tracking errors $e_{k,1} = x_{k,1} - y_{k,1}, k = 1, \dots, N$, exponentially converges to a small neighborhood around zero by properly choosing design parameters. \square

Proof (i) In order to show all the signals in the multi-agent system are still bounded, consider the following Lyapunov candidate

$$V = \frac{1}{2} \sum_{k=1}^N \sum_{j=1}^n e_{k,j}^2 + \frac{1}{2\rho} \sum_{k=1}^N \tilde{W}_k^T \tilde{W}_k. \tag{16}$$

Let $\tilde{W}_k = \hat{W}_k - W$. From (8)–(14) and (15), we deduce the closed-loop error system is that

$$\left\{ \begin{aligned} \dot{e}_{k,1} &= -c_{k,1}e_{k,1} + e_{k,2} + \epsilon_k \psi_k^{(2)}, \\ \dot{e}_{k,j} &= -e_{k,j-1} - c_{k,j}e_{k,j} \\ &\quad + e_{k,j+1}, \quad j = 2, \dots, n-1, \\ \dot{e}_{k,n} &= G_n(\bar{x}_{k,n}) \left[\frac{F_n(\bar{x}_{k,n}) + e_{k,n-1} - \dot{\alpha}_{k,n-1}}{G_n(\bar{x}_{k,n})} + u_k \right] \\ &\quad - e_{k,n-1} - \epsilon_k \psi_k^{(n+1)} \\ &= G_n(\bar{x}_{k,n}) \left[S^T(Z_{k,n})W + \varepsilon_k + u_k \right] \\ &\quad - e_{k,n-1} - \epsilon_k \psi_k^{(n+1)} \\ &= G_n(\bar{x}_{k,n}) \left[-c_{k,n}e_{k,n} - S^T(Z_{k,n})\tilde{W}_k + \varepsilon_k \right] \\ &\quad - e_{k,n-1} - \epsilon_k \psi_k^{(n+1)}, \\ \dot{\tilde{W}}_k &= \rho \left(e_{k,n}S(Z_{k,n}) - \sigma_k \tilde{W}_k \right) \\ &\quad - \gamma \sum_{j \in \mathcal{N}_k} a_{kj} \left(\tilde{W}_k - \tilde{W}_j \right), \end{aligned} \right. \tag{17}$$

The time derivative of V along (17) is that

$$\begin{aligned} \dot{V} &= \sum_{k=1}^N \sum_{j=1}^n e_{k,j} \dot{e}_{k,j} + \frac{1}{\rho} \sum_{k=1}^N \tilde{W}_k^T \dot{\tilde{W}}_k \\ &= - \sum_{k=1}^N \sum_{j=1}^{n-1} c_{k,j} e_{k,j}^2 - \sum_{k=1}^N c_{k,n} G_n(\bar{x}_{k,n}) e_{k,n}^2 \\ &\quad + \sum_{k=1}^N \left(\epsilon_k e_{k,1} \psi_k^{(2)} - \epsilon_k e_{k,n} \psi_k^{(n+1)} + e_{k,n} G_n(\bar{x}_{k,n}) \varepsilon_k \right) \\ &\quad + \sum_{k=1}^N \left(e_{k,n} S^T(Z_{k,n}) \tilde{W}_k - e_{k,n} G_n(\bar{x}_{k,n}) S^T(Z_{k,n}) \tilde{W}_k \right. \\ &\quad \left. - \sigma_k \tilde{W}_k^T \tilde{W}_k \right) - \frac{\gamma}{\rho} \tilde{W}^T (\mathcal{L} \otimes I_l) \tilde{W}, \end{aligned} \tag{18}$$

where $\tilde{W} = [\tilde{W}_1^T, \dots, \tilde{W}_N^T]^T$. Since the communication topology is undirected and connected, then the term $\frac{\gamma}{\rho} \tilde{W}^T (\mathcal{L} \otimes I_l) \tilde{W}$ is always nonnegative. Notice that $g_0 \leq g_i(\cdot) \leq g_1$, then $\underline{g} \leq G_n(\bar{x}_{k,n}) \leq \bar{g}$, where \underline{g} and \bar{g} are positive constants. According to the Young’s inequality, we have the following inequalities,

$$\left\{ \begin{aligned} \epsilon_k e_{k,1} \psi_k^{(2)} &\leq \frac{\varrho_1 e_{k,1}^2}{2} + \frac{\epsilon_k^2 h_k^2}{2\varrho_1}, \\ -\epsilon_k e_{k,n} \psi_k^{(n+1)} &\leq \frac{\varrho_2 e_{k,n}^2}{2} + \frac{\epsilon_k^2 h_{n+1}^2}{2\varrho_2}, \\ e_{k,n} G_n(\bar{x}_{k,n}) \varepsilon_k &\leq \frac{\varrho_3 e_{k,n}^2}{2} + \frac{\bar{g}^2 \varepsilon^{*2}}{2\varrho_3}, \\ e_{k,n} S^T(Z_{k,n}) \tilde{W}_k &\leq \frac{s^2 e_{k,n}^2}{\sigma} + \frac{\sigma \tilde{W}_k^T \tilde{W}_k}{4}, \\ -e_{k,n} G_n(\bar{x}_{k,n}) S^T(Z_{k,n}) \tilde{W}_k &\leq \frac{s^2 \bar{g}^2 e_{k,n}^2}{\sigma} + \frac{\sigma \tilde{W}_k^T \tilde{W}_k}{4}, \\ -\sigma_k \tilde{W}_k^T \tilde{W}_k &\leq -\frac{3\sigma \tilde{W}_k^T \tilde{W}_k}{4} + \sigma_k \|W\|^2, \end{aligned} \right. \tag{19}$$

where $\|S(Z_{k,n})\| \leq s$, $\sigma = \min\{\sigma_1, \dots, \sigma_N\}$, and $s, \varrho_1, \varrho_2, \varrho_3$ are positive constants. By using (19), it follows that

$$\begin{aligned} \dot{V} &\leq - \sum_{k=1}^N \sum_{j=1}^{n-1} c_{k,j} e_{k,j}^2 - \sum_{k=1}^N c_{k,n} G_n(\bar{x}_{k,n}) e_{k,n}^2 \\ &\quad + \sum_{k=1}^N \left(\frac{\varrho_1 e_{k,1}^2}{2} + \frac{\varrho_2 e_{k,n}^2}{2} + \frac{\varrho_3 e_{k,n}^2}{2} + \frac{s^2 e_{k,n}^2}{\sigma} + \frac{s^2 \bar{g}^2 e_{k,n}^2}{\sigma} \right) \\ &\quad + \sum_{k=1}^N \left(\frac{\epsilon_k^2 h_k^2}{2\varrho_1} + \frac{\epsilon_k^2 h_{n+1}^2}{2\varrho_2} + \frac{\bar{g}^2 \varepsilon^{*2}}{2\varrho_3} + \sigma_k \|W\|^2 \right) \\ &\quad - \sum_{k=1}^N \frac{\sigma \tilde{W}_k^T \tilde{W}_k}{4}. \end{aligned} \tag{20}$$

Choosing the parameters $c_{k,1}$ and $c_{k,n}$ such that $c_{k,1} = c'_{k,1} + \varrho_1/2$ and $c_{k,n} \underline{g} = c'_{k,n} + \varrho_2/2 + \varrho_3/2 + s^2/\sigma + s^2 \bar{g}^2/\sigma$, then we obtain that

$$\begin{aligned} \dot{V} &\leq -c \sum_{k=1}^N \sum_{j=1}^n e_{k,j}^2 - \sum_{k=1}^N \frac{\sigma \tilde{W}_k^T \tilde{W}_k}{4} + \mu \\ &\leq -\delta V + \mu, \end{aligned} \tag{21}$$

where $c = \min\{c'_{k,1}, c_{k,2}, \dots, c_{k,n-1}, c'_{k,n}\}$, $k = 1, \dots, N$, $\delta = \min\{2c, \rho\sigma/2\}$ and $\mu = \sum_{k=1}^N \left(\epsilon_k^2 h_k^2/2\varrho_1 + \epsilon_k^2 h_{n+1}^2/2\varrho_2 + \bar{g}^2 \varepsilon^{*2}/2\varrho_3 + \sigma_k \|W\|^2 \right)$. From (21), we directly obtain

$$V(t) \leq V(0)e^{-\delta t} + \frac{\mu}{\delta}. \tag{22}$$

This implies that $e_{k,j}, \tilde{W}_k, j = 1, \dots, n, k = 1, \dots, N$ are bounded. Further, $\tilde{W}_k, \hat{z}_{k,j}, z_{k,j}, x_{k,j}, j = 1, \dots, n, k = 1, \dots, N$ are also bounded. The conclusion is reached.

(ii) In view of (16) and (22), one has

$$\sum_{k=1}^N \sum_{j=1}^n e_{k,j}^2 \leq 2V(t) \leq 2V(0)e^{-\delta t} + \frac{2\mu}{\delta}. \tag{23}$$

It is obvious that $\frac{2\mu}{\delta}$ can be made small enough by properly choosing large $c_{k,j}, j = 1, \dots, n, k = 1, \dots, N$ and small $\sigma_k, k = 1, \dots, N$. Thus $e_{k,j}, j = 1, \dots, n, k = 1, \dots, N$, exponentially converge to a small neighborhood around zero after a finite time T . This means that tracking errors $e_{k,1}, k = 1, \dots, N$, exponentially converge to a small neighborhood around zero after a finite time T . The proof is completed. \square

3.3 Cooperative learning

In Theorem 1, the control objective is achieved. In what follows, the learning ability of RBF NNs will be illustrated. Note that the NN weight update law $\dot{\hat{W}}_k$ contains the cooperative learning term $-\gamma \sum_{j \in \mathcal{N}_k} a_{kj}(\hat{W}_k - \hat{W}_j)$. This implies that any agent in the multi-agent system can learn the knowledge not only from itself but also from its neighboring agents. The following theorem shows that once the learning is completed, the common approximation domain is the union of all orbits of all agents.

Theorem 2 Consider a multi-agent system consisting of systems (4) with Assumptions 1, the reference models (5), the HGOs (7), the controllers (8) and weight update laws (15). Assume that the communication topology of the multi-agent system is undirected and connected. For any recurrent orbits $\varphi_{dk}, k = 1, \dots, N$ and any bounded initial conditions $\bar{x}_{k,n} \in \Omega_k$ (where Ω_k is a compact set), and $\dot{W}_k = 0$, the weight estimates $\hat{W}_k, k = 1, \dots, N$ converge to a small neighborhood of W , and the lumped uncertainty $H(Z_{k,n})$ can be approximated by $S^T(Z)\bar{W}_k$ along the union orbit $Z = \cup_{k=1}^N Z_{k,n}$, where

$$\bar{W}_k = \text{mean}_{t \in [t_1, t_2]} \hat{W}_k,$$

$[t_1, t_2] (t_2 > t_1 > T_1)$ represents a time segment after the transient process and $\text{mean}_{t \in [t_1, t_2]}$ stands for calculating the mean value on the time segment $[t_1, t_2]$. \square

Proof According to the localized approximation property of RBF NNs and noting that $\tilde{W}_{k_\zeta} = \hat{W}_{k_\zeta} - W_\zeta$, (15) and (17) can be expressed as

$$\begin{aligned} \dot{e}_{k,n} &= G(\bar{x}_{k,n}) \left[-c_{k,n} e_{k,n} - S_\zeta^T(Z_{k,n}) \tilde{W}_{k_\zeta} \right] + \varepsilon_{k_\zeta} G(\bar{x}_{k,n}) \\ &\quad + \eta_k, \end{aligned} \tag{24}$$

$$\dot{\hat{W}}_{k_\zeta} = \rho \left(e_{k,n} S_\zeta(Z_{k,n}) - \sigma_k \hat{W}_{k_\zeta} \right) - \gamma \sum_{j \in \mathcal{N}_k} a_{kj} (\hat{W}_{k_\zeta} - \hat{W}_{j_\zeta}) \tag{25}$$

and

$$\dot{\hat{W}}_{k_\zeta} = \rho \left(e_{k,n} S_\zeta(Z_{k,n}) - \sigma_k \hat{W}_{k_\zeta} \right) - \gamma \sum_{j \in \mathcal{N}_k} a_{kj} (\hat{W}_{k_\zeta} - \hat{W}_{j_\zeta}), \tag{26}$$

where $(\cdot)_\zeta$ and $(\cdot)_{\bar{\zeta}}$ represent the region close to and away from the union orbit $Z = \cup_{k=1}^N Z_{k,n}$, respectively. $\eta_k = -e_{k,n-1} - \epsilon_k \psi_k^{(n+1)}$. $S_\zeta^T(Z_{k,n}), \tilde{W}_{k_\zeta}$ and \tilde{W}_{j_ζ} are the subvectors of $S^T(Z_{k,n}), \tilde{W}_k$ and \tilde{W}_j , respectively, and $\varepsilon_{k_\zeta} = \varepsilon_k - S_\zeta^T(Z_{k,n}) \tilde{W}_{k_\zeta}$ is the new approximation error along the orbit Z . Because $S_{\bar{\zeta}}(Z_{k,n})$ is away from Z , then $S_{\bar{\zeta}}(Z_{k,n})$ is very small. Choosing small σ_k and noting that $\hat{W}_{k_\zeta}(0) = \hat{W}_{j_\zeta}(0) = 0$, it is clear that \hat{W}_{k_ζ} is slightly updated. Based on the above analysis, $S_\zeta^T(Z_{k,n}) \tilde{W}_{k_\zeta}$ is very small such that $\varepsilon_{k_\zeta} = O(\varepsilon_k)$.

It is worth noting that $\varepsilon_{k_\zeta} G(\bar{x}_{k,n})$ will be a perturbation term if we use (24) and (25) to construct a closed-loop error system. However, $\varepsilon_{k_\zeta} G(\bar{x}_{k,n})$ may be not a small value owing to $G_n(\bar{x}_{k,n}) \geq \underline{g}$. To avoid this problem, similar to [31, 34], a linear transformation $\zeta_{k,n} = e_{k,n}/\bar{g}$ is employed. It can be deduced from (24) and (25) that

$$\dot{\zeta}_{k,n} = -c_{k,n} G(\bar{x}_{k,n}) \zeta_{k,n} - \frac{G(\bar{x}_{k,n}) S_\zeta^T(Z_{k,n})}{\bar{g}} \tilde{W}_{k_\zeta} + \eta'_k, \tag{27}$$

$$\dot{\tilde{W}}_{k_\zeta} = \rho \left(\bar{g} S_\zeta(Z_{k,n}) \zeta_{k,n} - \sigma_k \hat{W}_{k_\zeta} \right) - \gamma \sum_{j \in \mathcal{N}_k} a_{kj} (\tilde{W}_{k_\zeta} - \tilde{W}_{j_\zeta}), \tag{28}$$

where $\eta'_k = (G(\bar{x}_{k,n})/\bar{g})\varepsilon_{k_\zeta} + \eta_k/\bar{g}$. Due to $G(\bar{x}_{k,n})/\bar{g} \leq 1$, thus it is clear that η'_k is a small value. Combining (27) and (28), the overall closed-loop system of the multi-agent system can be expressed by

$$\begin{aligned} \begin{bmatrix} \dot{\zeta} \\ \dot{\tilde{W}}_\zeta \end{bmatrix} &= \begin{bmatrix} A(t) & -\Psi(t)^T \\ \Gamma(t)\Psi(t) & -\gamma(\mathcal{L} \otimes I_\zeta) \end{bmatrix} \begin{bmatrix} \zeta \\ \tilde{W}_\zeta \end{bmatrix} \\ &\quad + \begin{bmatrix} \eta' \\ A\hat{W}_\zeta \end{bmatrix}, \end{aligned} \tag{29}$$

where

$$\begin{aligned} \zeta &= [\zeta_{1,n}, \dots, \zeta_{N,n}]^T \\ \tilde{W}_\zeta &= [\tilde{W}_{1_\zeta}^T, \dots, \tilde{W}_{N_\zeta}^T]^T \\ A(t) &= \text{diag}\{-c_{1,n}G(\bar{x}_{1,n}), \dots, -c_{N,n}G(\bar{x}_{N,n})\} \\ \Psi(t) &= \text{diag}\{G(\bar{x}_{1,n})S_\zeta(Z_{1,n})/\bar{g}, \dots, G(\bar{x}_{N,n})S_\zeta(Z_{N,n})/\bar{g}\} \\ \Gamma(t) &= \text{diag}\{\rho\bar{g}^2/G(\bar{x}_{1,n}), \dots, \rho\bar{g}^2/G(\bar{x}_{N,n})\} \\ \eta' &= [\eta'_1, \dots, \eta'_N]^T \\ A &= \text{diag}\{\rho\sigma_1, \dots, \rho\sigma_N\}. \end{aligned}$$

Note that $\Gamma(t)$ is a symmetric positive definite matrix. Let $P(t) = \Gamma(t)$, and then

$$\begin{aligned} & \dot{P}(t) + P(t)A(t) + A^T(t)P(t) \\ &= -\text{diag}\{2\rho\bar{g}^2c_{1,n} + \rho\bar{g}^2\dot{G}(\bar{x}_{1,n})/G^2(\bar{x}_{1,n}), \dots, \\ & \dots, 2\rho\bar{g}^2c_{N,n} + \rho\bar{g}^2\dot{G}(\bar{x}_{N,n})/G^2(\bar{x}_{N,n})\} \end{aligned} \tag{30}$$

where ρ and \bar{g}^2 are positive. Since $g_i(\cdot) \geq g_0, i = 1, \dots, n$, are smooth nonlinear functions, we can easily obtain that $G(\bar{x}_{k,n}) \geq 0$ and $\dot{G}(\bar{x}_{k,n})$ are the continuous function. Furthermore, according to the boundness of $\bar{x}_{k,n}$, $\dot{G}(\bar{x}_{k,n})$ is also bounded, but its direction is also unknown. Consequently, we can properly choose large $c_{k,n}$ such that $2\rho\bar{g}^2c_{k,n} + \rho\bar{g}^2\dot{G}(\bar{x}_{k,n})/G^2(\bar{x}_{k,n}) > 0, k = 1, \dots, N$, as well as there exist a symmetric positive definite matrix $Q(t)$ such that $\dot{P}(t) + P(t)A(t) + A^T(t)P(t) = -Q(t)$, where $Q(t) = \text{diag}\{2\rho\bar{g}^2c_{1,n} + \rho\bar{g}^2\dot{G}(\bar{x}_{1,n})/G^2(\bar{x}_{1,n}), \dots, 2\rho\bar{g}^2c_{N,n} + \rho\bar{g}^2\dot{G}(\bar{x}_{N,n})/G^2(\bar{x}_{N,n})\}$.

On the other hand, Theorem 1 shows that $e_{k,j}, j = 1, \dots, n, k = 1, \dots, N$, converge to a small neighborhood around zero after the finite time T . Thus, \hat{z}_k converges closely to $\bar{y}_{k,n}$ after the time T . It follows from Lemma 2 that $z_k = [z_{k,1}, \dots, z_{k,n}]^T$ is the recurrent signal. Moreover, $f_i(\cdot)$ and $g_i(\cdot)$ are smooth functions. We can conclude that $\bar{x}_{k,n}$ is recurrent. Using (9)–(12) and (17) and with simple mathematical operation, we have

$$\begin{aligned} \dot{\alpha}_{k,n-1} &= L_{k1}(e_{k,1}, \dots, e_{k,n}) + L_{k2}(\epsilon_k\psi_k^{(2)}, \dots, \epsilon_k\psi_k^{(n)}) \\ &+ f_{dk}(\bar{y}_{k,n}, t), \end{aligned} \tag{31}$$

where $L_{k1}(\cdot)$ and $L_{k2}(\cdot)$ are linear combinations. Because $L_{k1}(e_{k,1}, \dots, e_{k,n})$ and $L_{k2}(\epsilon_k\psi_k^{(2)}, \dots, \epsilon_k\psi_k^{(n)})$ are very small. Therefore, $\dot{\alpha}_{k,n-1}$ is the recurrent signal. Further, the NN input $Z_{k,n}$ is also recurrent. By Lemma 1, $S_\zeta(Z_{k,n}), k = 1, \dots, N$ are cooperative PE, then $G(\bar{x}_{k,n})S_\zeta(Z_{k,n})/\bar{g}, k = 1, \dots, N$, satisfy the cooperative PE condition.

According to the Theorem 1 of [37], the solution of (29) converges to a small neighborhood of zero. As a result, the weight estimation \hat{W}_{k_ζ} converges to a small neighborhood of W_{k_ζ} , that is, $\hat{W}_{1_\zeta} \cong \dots \cong \hat{W}_{N_\zeta}$, as well as $\bar{W}_1 \cong \dots \cong \bar{W}_N$. This means that the lumped uncertainty $H(\cdot)$ can be approximated by RBF NNs along the union orbit Z , that is,

$$\begin{aligned} H(Z) &= S_\zeta(Z)^T W_\zeta + \varepsilon_\zeta \\ &= S_\zeta(Z)^T \bar{W}_{k_\zeta} + \varepsilon_{k_\zeta 1} \\ &= S_\zeta(Z)^T \bar{W}_{k_\zeta} + S_\zeta(Z)^T \bar{W}_{k_\zeta} + \varepsilon_{k_\zeta 1} - S_\zeta(Z)^T \bar{W}_{k_\zeta} \\ &= S(Z)^T \bar{W}_k + \varepsilon_{k2}, \end{aligned}$$

where $\varepsilon_{k2} = \varepsilon_{k_\zeta 1} - S_\zeta(Z)^T \bar{W}_{k_\zeta}$ and $\varepsilon_{k_\zeta 1}$ are approximation errors. They are very small. Thus, the approximation of $H(\cdot)$ along the union orbit $Z = \cup_{k=1}^N Z_{k,n}$ is obtained. The proof is completed. \square

Remark 4 It should be pointed out that the NN weight update law $\dot{\hat{W}}_k$ in this paper is different from that in [25, 34]. Due to the existence of the cooperative learning term $-\gamma \sum_{j \in \mathcal{N}_k} a_{kj} (\hat{W}_k - \hat{W}_j)$, any agent in the multi-agent system can learn the knowledge from its neighbor in a distributed manner. The benefits add this term is that the approximation domain of RBF NNs obtained in this paper is the union orbit of all agent, which is larger than that in [25] and [34]. But it is necessary to transmit weight information between agents.

Remark 5 The proposed solution in this paper can be used in many practical strict-feedback systems, such as robot manipulators [19] and unmanned surface vehicles [21, 44], because these systems satisfy Assumption 1. In contrast to the existing works, the main factor affecting practical applicability of the proposed solution is that we add the cooperative learning term $-\gamma \sum_{j \in \mathcal{N}_k} a_{kj} (\hat{W}_k - \hat{W}_j)$ in the NN weight update law (15). It requires the communication between the neighbouring agents and a lot of weight information is transmitted over the communication network. Thus, high-performance communication networks, such as Ethernet, should be used in practice.

4 Control with experience

In Sect. 3, the NN approximation of the lumped uncertainty is obtained along the union orbit. In this section, we will use the obtained NNs to design the controller to control the same plant. Compared with adaptive neural control, the neural weight \bar{W}_k obtained in Theorem 2 can be directly used to design controllers, which does not adapt online. Therefore, the control performance, such as the convergence rate and transient performance, will be improved effectively. To show this, let us consider the same plant as (4)

$$\begin{cases} \dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1} & i = 1, \dots, n, \\ \dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u \\ y = x_1, \end{cases} \tag{32}$$

where $\bar{x}_i = [x_1, \dots, x_i]^T, \bar{x}_n = [x_1, \dots, x_n]^T \in R^n$ is the state vector, and y is the output. First of all, we transform (32) into the normal form of (6)

$$\begin{cases} \dot{z}_i = z_{i+1}, & i = 1, \dots, n, \\ \dot{z}_n = F_n(\bar{x}_n) + G_n(\bar{x}_n)u \\ y = z_1 = x_1, \end{cases} \tag{33}$$

where $z = [z_1, \dots, z_n]^T$ is the state vector. The HGO (7) is still used to estimate the unknown state z of (33). Denote the estimation of z as $\hat{z} = [\hat{z}_1, \hat{z}_2, \dots, \hat{z}_n]^T$.

For the k th reference model of (5), similar to (8)–(13), we directly give the controller of (32)

$$u = -c_n e_n - S(Z_n)^T \bar{W}_k, \quad k \in \{1, \dots, N\}. \tag{34}$$

where \bar{W}_k is the previously learned NN weight.

Theorem 3 Consider the closed-loop system consisting of the plant (32), the HGO (7), the k th reference model of (5), and the controller (34). For the initial condition $\bar{y}_{k,n}(0)$ generates the same periodic reference trajectory φ_{dk} as in Theorem 2, and with the initial condition $\bar{x}_n(0)$ of (32) in a small neighborhood of $\Upsilon_x(0)$, where $\Upsilon_x(0)$ represents the value of $x_n(t)$ when $\bar{y}_{k,n}(t) = \bar{y}_{k,n}(0)$, then the following statements are true,

- (i) All the signals in the closed-loop system are still bounded;
- (ii) The system output $y = x_1$ exponentially converges to a small neighborhood of $y_{k,1}$. \square

Proof The proof is similar to the proof of Theorem 2 of [40]. Thus, it is omitted here. \square

5 Simulation

In this section, two examples are provided to validate the effectiveness of the proposed mechanism.

5.1 Numerical example

Consider a multi-agent system consisting of the following three strict-feedback systems [29]

$$\begin{cases} \dot{x}_{k,1} = 0.1x_{k,1}^2 + (1 + 0.1 \sin(x_{k,1}))x_{k,2} \\ \dot{x}_{k,2} = 0.2e^{-x_{k,2}^2} + x_{k,1} \sin(x_{k,2}) + u_k \\ y_k = x_{k,1}, \quad k = 1, 2, 3, \end{cases} \tag{35}$$

where $x_{k,1}$ and $x_{k,2}$ are the system states, and y_k is the output. The communication topology of the multi-agent system is shown in Fig. 1. We assume that $f_1(\bar{x}_{k,1}) = 0.1x_{k,1}^2$, $f_2(\bar{x}_{k,2}) = 0.2e^{-x_{k,2}^2} + x_{k,1} \sin(x_{k,2})$ and $g_1(\bar{x}_{k,1}) = 1 + 0.1 \sin(x_{k,1})$ are totally unknown, but they



Fig. 1 Numerical example: communication topology \mathcal{G}

satisfy Assumption 1. The initial state values of (35) are $\bar{x}_{1,2} = [0.2, 0.4]^T$, $\bar{x}_{2,2} = [0.3, 0.4]^T$ and $\bar{x}_{3,2} = [0.1, 0.3]^T$.

Denote $z_{k,1} = x_{k,1}$ and $z_{k,2} = \dot{x}_{k,1}$. Then we transform (35) into the normal form

$$\begin{cases} \dot{z}_{k,1} = z_{k,2} \\ \dot{z}_{k,2} = F_2(\bar{x}_{k,2}) + G_2(\bar{x}_{k,2})u_k \\ y_k = z_{k,1} = x_{k,1}, \quad k = 1, 2, 3, \end{cases} \tag{36}$$

where

$$F_2(\bar{x}_{k,2}) = [0.2x_{k,1} + 0.1x_{k,2} \cos(x_{k,1})][0.1x_{k,1}^2 + (1 + 0.1 \sin(x_{k,1}))x_{k,2}] + (1 + 0.1 \sin(x_{k,1}))[0.2e^{-x_{k,2}^2} + x_{k,1} \sin(x_{k,2})]$$

and $G_2(\bar{x}_{k,2}) = 1 + 0.1 \sin(x_{k,1})$. Next, the following three HGOs are designed to estimate the states of (36)

$$\begin{cases} \epsilon_k \dot{\zeta}_{k,1} = \zeta_{k,2} \\ \epsilon_k \dot{\zeta}_{k,2} = -p_{k,1} \zeta_{k,2} - \zeta_{k,1} + y_k(t), \quad k = 1, 2, 3, \end{cases} \tag{37}$$

where $p_{1,1} = p_{2,1} = p_{3,1} = 1$ and $\epsilon_1 = \epsilon_2 = \epsilon_3 = 0.0008$.

In this example, Duffing oscillators [30] are used as the reference models

$$\begin{cases} \dot{y}_{k,1} = y_{k,2} \\ \dot{y}_{k,2} = -\lambda_{k,1}y_{k,1} - \lambda_{k,2}y_{k,1}^3 - \lambda_{k,3}y_{k,2} + q_k \cos(w_k t) \\ y_{dk} = y_{k,1}, \quad k = 1, 2, 3, \end{cases} \tag{38}$$

where the parameters $[\lambda_{1,1}, \lambda_{2,1}, \lambda_{3,1}] = [-1.1, -0.8, -0.6]$, $\lambda_{1,3} = \lambda_{2,3} = \lambda_{3,3} = 0.55$, $\lambda_{1,2} = \lambda_{2,2} = \lambda_{3,2} = 1$, $[q_1, q_2, q_3] = [1.4, 1.2, 1]$, and $w_1 = w_2 = w_3 = 1$. The initial conditions are set to be $\bar{y}_{1,2} = [1.5, 0.4]^T$, $\bar{y}_{2,2} = [1.2, 0.8]^T$ and $\bar{y}_{3,2} = [1.6, 0.5]^T$.

From Sect. 3, we know that after the system transformation, an agent only use one RBF NN to approximate the lumped uncertainty $H(Z)$ along the union orbit $Z = \cup_{k=1}^3 Z_{k,2}$. This means that only three RNF NNs are used in this simulation. Note that the parameters of the three RBF NNs are the same. Each RBF NN contains $11 \times 11 \times 11$ centers and these centers are spaced in $[-2.5, 2.5] \times [-2.5, 2.5] \times [-5, 5]$. The width of RBF NNs is $\iota = 0.8$. Other design parameters are set to be $\rho = 2$, $\gamma = 1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0.00001$, $c_{1,1} = c_{2,1} = c_{3,1} = 6$ and $c_{1,2} = c_{2,2} = c_{3,2} = 10$.

Figures 2, 3, 4 and 5 show the simulation results. In Fig. 2, we can see that the output of each agent can follow its reference signal effectively. Figures 3 and 4 show the approximation performance of RBF NNs, where $\bar{W}_k = \text{mean}_{t \in [330s, 350s]} \hat{W}_k, k = 1, 2, 3$. To verify the approximation domain of each RBF NN is the union orbit

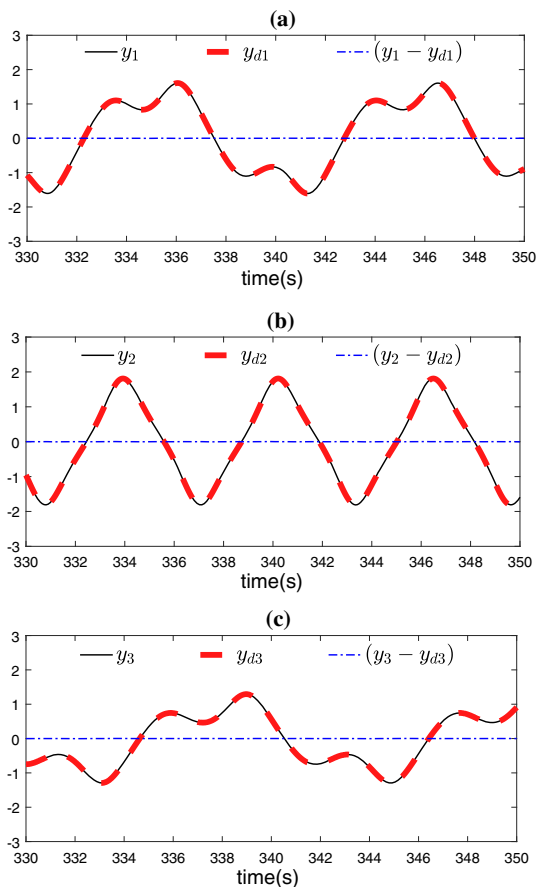


Fig. 2 Numerical example: control performance

$Z = \cup_{k=1}^3 Z_{k,2}$ rather than its own orbit $Z_{k,2}$, we exchange the inputs of the RBF NNs and Fig. 4 shows the corresponding function approximation performance. From Fig. 4, it is clear that the approximation performance is still very good. Fig. 5 shows that the NN weights of all agents converge to a small neighborhood of their common optimal value, which also implies the approximation domain is the union orbit. For the purpose of comparison, let $\gamma = 0$, and then the weight update law is same as that in [25, 34]. Fig. 6 depicts the convergence of the NN weights, where only $\gamma = 0$ and other parameters are same as before. It is evident from Fig. 6 that convergence values of their weights are different. Further, we exchange the inputs of RBF NNs. The approximation performance is shown in Fig. 7. Obviously, the approximation performance is worse than that in Fig. 4. This implies that the approximation domain of each agent in its own orbit rather than the union orbit. Thus, the approximation domain of NNs that are obtained in this paper is larger than that in [25, 34].

After the above cooperative learning control process, we obtain the NNs, such as $S(\cdot)\bar{W}_1$, $S(\cdot)\bar{W}_2$ and $S(\cdot)\bar{W}_3$. They can be used to design the controller to control the same plant, see (34). Compared with the adaptive neural control,

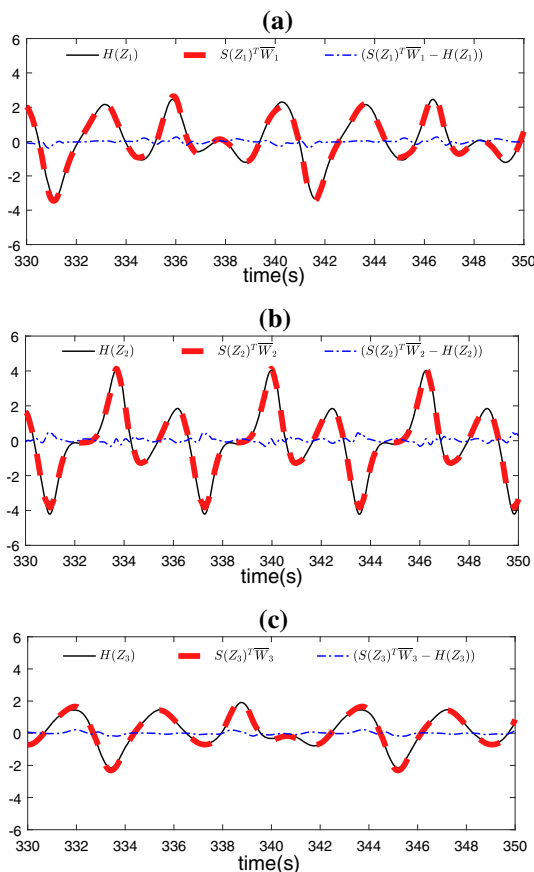


Fig. 3 Numerical example: function approximation

the performance of neural control with the learned NNs is improved. The effectiveness can be found in [25, 30, 34, 36] and [37]. Thus, the simulation of neural control with past experience is omitted here.

5.2 Inverted pendulum example

Consider the following five inverted pendulum systems [29, 47],

$$\begin{cases} \dot{x}_{k,1} = x_{k,2} \\ \dot{x}_{k,2} = \frac{g \sin(x_{k,1}) - \frac{m l x_{k,2}^2 \cos(x_{k,1}) \sin(x_{k,1})}{m_c + m}}{l(\frac{4}{3} - \frac{m \cos^2(x_{k,1})}{m_c + m})} + \frac{\cos(x_{k,1})}{m_c + m} u_k \\ y_k = x_{k,1}, \quad k = 1, 2, 3, 4, 5, \end{cases}$$

where y_k and $x_{k,2}$ are the angle and angular velocity of the k th pendulum, respectively, u_k is the input force. m_c is the mass of cart, m is the mass of a pole, l is the half length of a pole and $g = 9.8 m/s^2$ is the gravity acceleration. The parameters are given by $m_c = 1 \text{ Kg}$, $m = 0.1 \text{ Kg}$ and

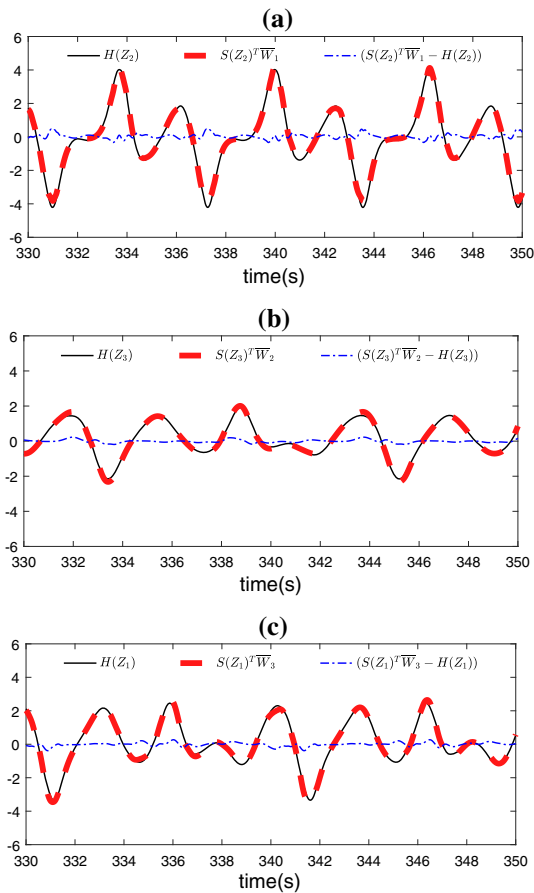


Fig. 4 Numerical example: function approximation with exchanging inputs

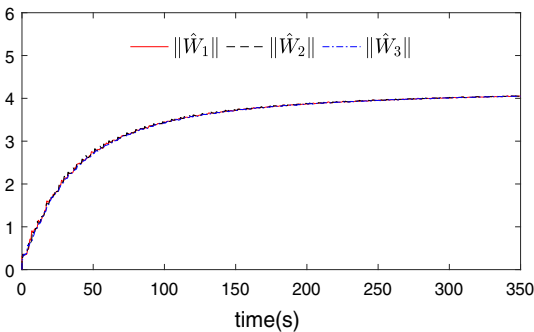


Fig. 5 Numerical example: the convergence of NN weight

$l = 0.5$ m. Notice that $f_1(\bar{x}_{k,1}) = 0$ and $g_1(\bar{x}_{k,1}) = 1$. Thus, there is no need to use HGOs to estimate the unknown states.

The reference models are given as follows

$$\begin{cases} \dot{y}_{k,1} = y_{k,2} \\ \dot{y}_{k,2} = -0.0025x_{k,1} - 0.2x_{k,2} + q'_k \cos(2t) \\ y_{dk} = y_{k,1}, \quad k = 1, 2, 3, 4, 5, \end{cases} \quad (39)$$

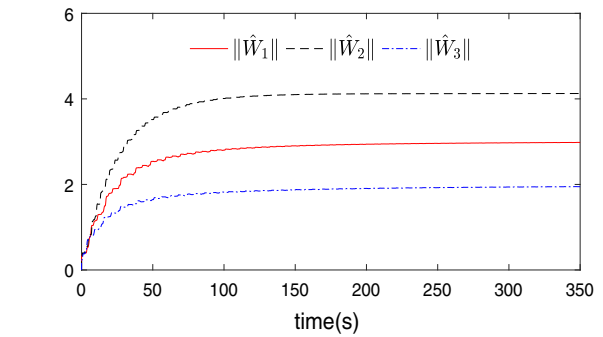


Fig. 6 Numerical example: the convergence of NN weight in [25, 34]

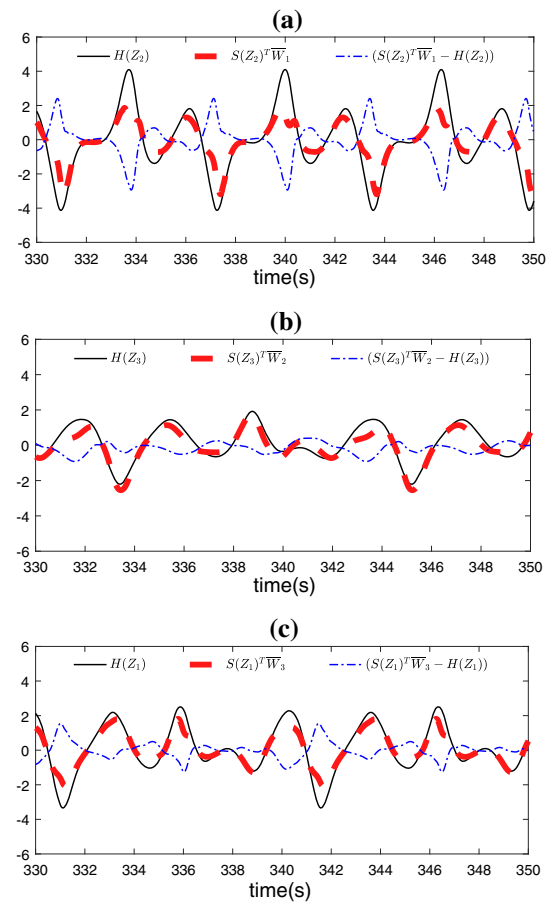
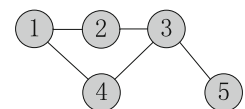


Fig. 7 Numerical example: function approximation using the schemes in [25, 34] with exchanging inputs

Fig. 8 Inverted pendulum systems: communication topology \mathcal{G}



where $[q'_1, q'_2, q'_3, q'_4, q'_5] = [0.05, 0.08, 0.1, 0.13, 0.16]$. The initial states of (39) are set to be $\bar{y}_{k,2} = [0, 0]^T, k = 1, 2, 3, 4, 5$.

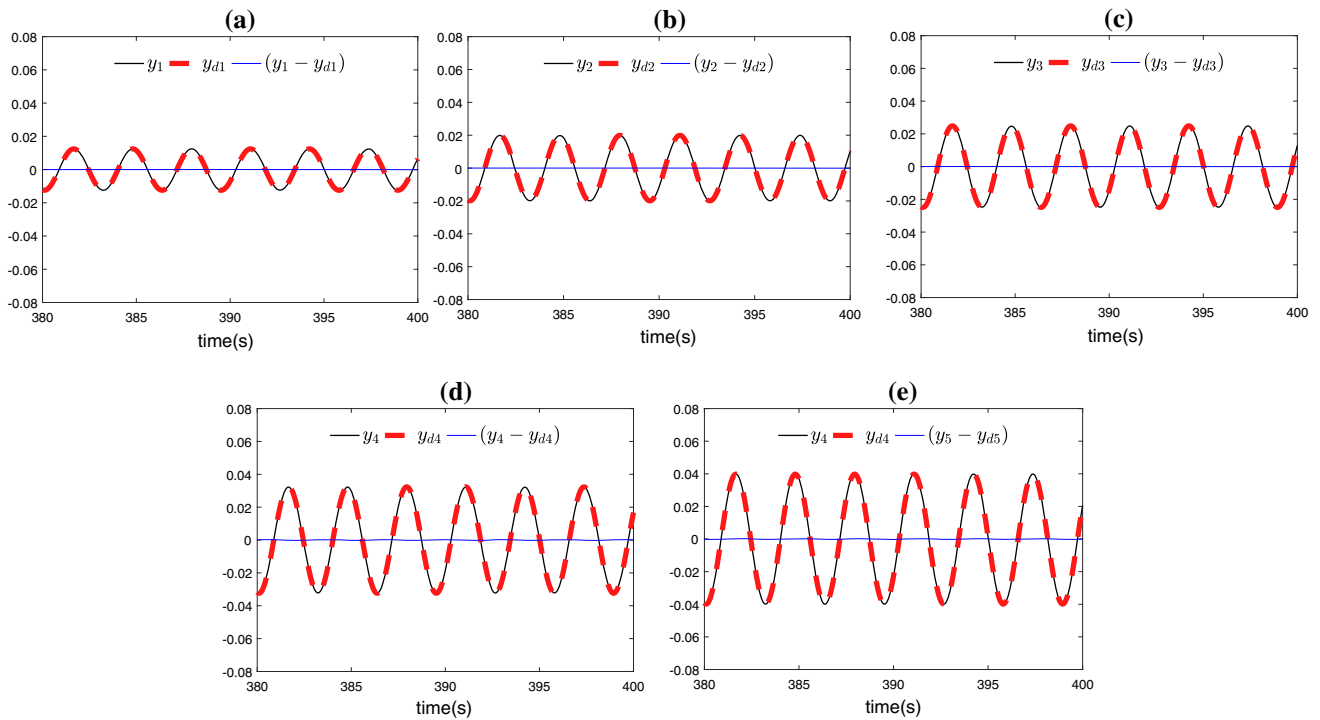


Fig. 9 Inverted pendulum systems: control performance

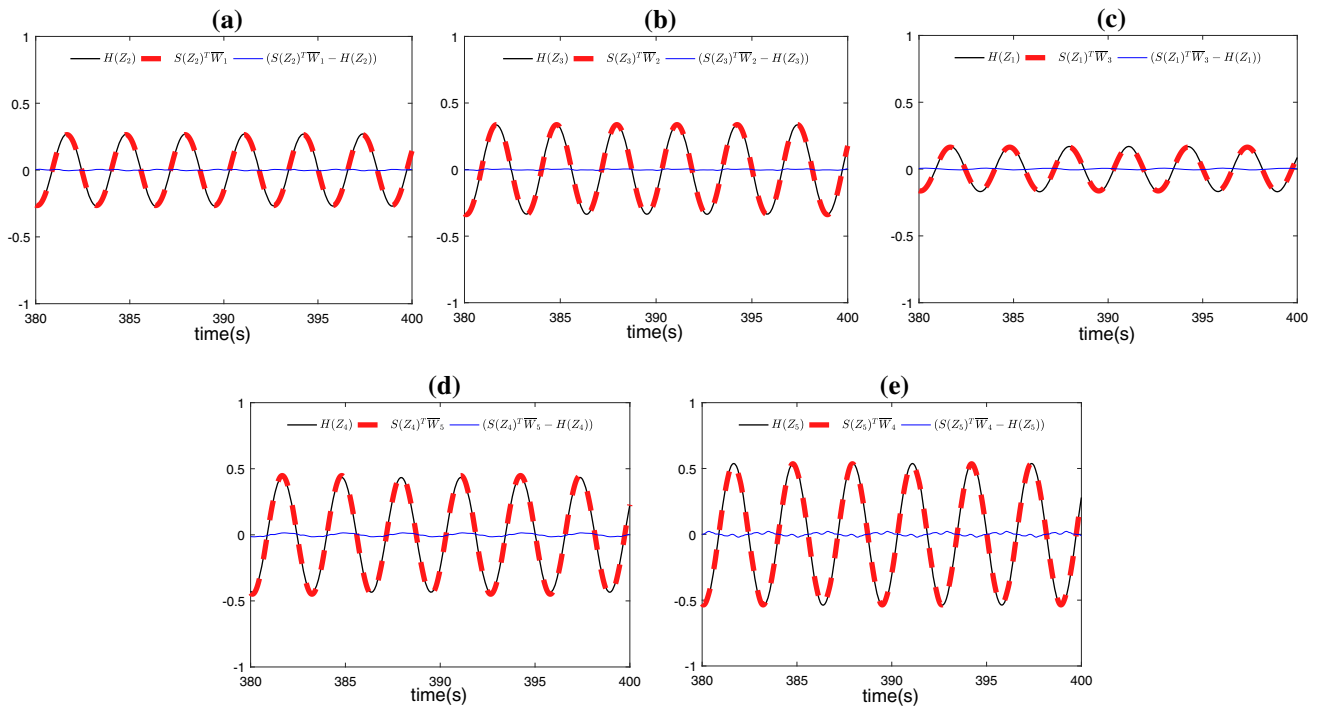


Fig. 10 Inverted pendulum systems: function approximation with exchanging inputs

In this example, five RBF NNs are used to approximate $H(Z)$ along the union orbit $Z = \cup_{k=1}^5 Z_{k,2}$. Each RBF NN contains $11 \times 11 \times 11$ centers and these centers are spaced in $[-0.1, 0.1] \times [-0.2, 0.2] \times [-0.3, 0.3]$. The width of

RBF NNs is $\iota = 0.05$. The design parameters are selected as $\rho = 2$, $\gamma = 1$, $\sigma_1 = \sigma_2 = \sigma_3 = 0.00001$, $c_{1,1} = c_{2,1} = c_{3,1} = 4$ and $c_{1,2} = c_{2,2} = c_{3,2} = 10$. Figure 8 is the topology of the multiagent system.

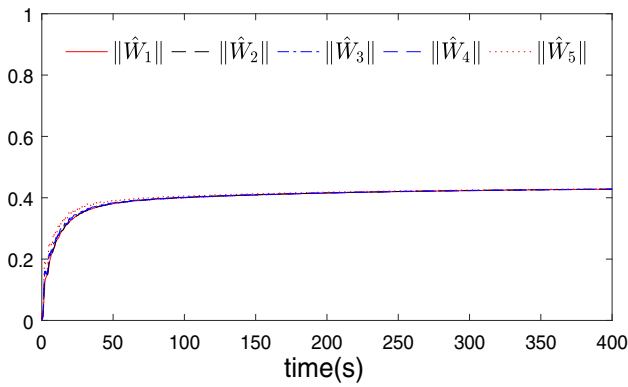


Fig. 11 Inverted pendulum systems: the convergence of NN weight

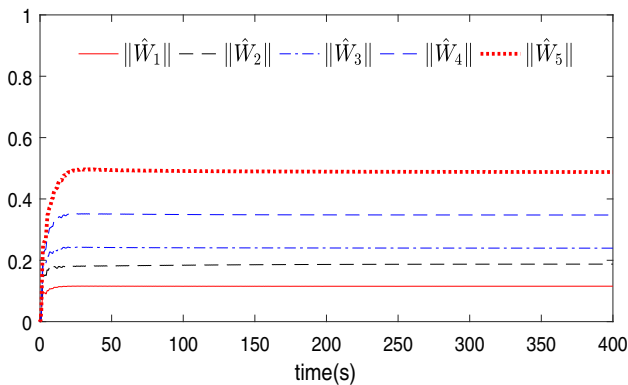


Fig. 12 Inverted pendulum systems: the convergence of NN weight in [25, 34]

Simulation results are shown in Figs. 9, 10 and 11. Fig. 9 shows that tracking performance. Figure 10 shows the function approximation with exchanging inputs. It indicates that approximation performance is still very good. Figure 11 shows that the NN weights of all agents converge to a small neighborhood of their common optimal value. Figures 12 and 13 are obtained by using the scheme in [25, 34]. They show that the function approximation with exchanging inputs is very bad because NN weights do not converge to a small neighborhood of their common optimal value.

6 Conclusion

This paper explores the DCL from adaptive NN control for a group of strict-feedback systems. First, we transform the strict-feedback systems into a normal form. With the help of this transformation, the number of RBF NNs is reduced in an agent, and it is easy to verify that the inputs of NNs are recurrent. Then we prove that all the signals in the multi-agent system are still bounded and the system output of each agent tracks its reference signal with a small error. Furthermore, the convergence of NN weights is achieved and the lumped uncertainty is approximated along the union orbit by RBF NNs. This implies that the RBF NNs have a large common approximation domain compared with the existing works. Finally, the learned NNs can be

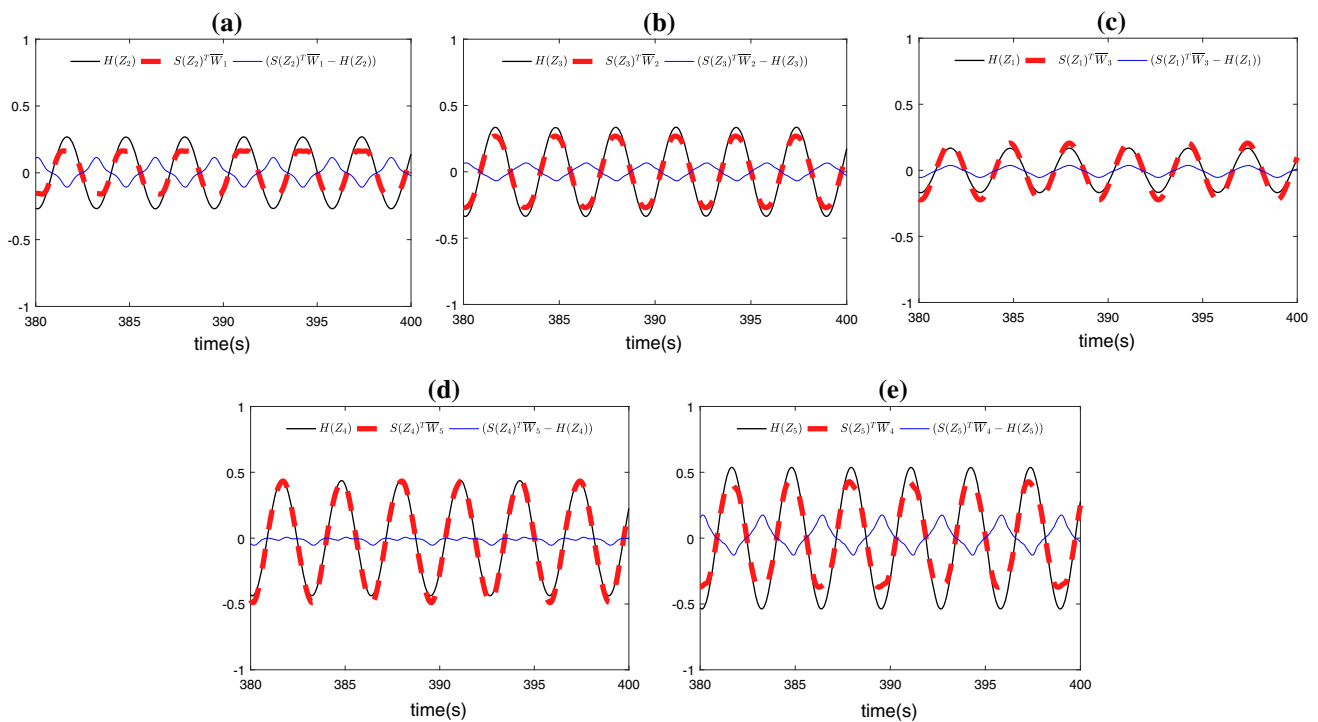


Fig. 13 Inverted pendulum systems: function approximation using the schemes in [25, 34] with exchanging inputs

reused to design controller to improve the control performance.

It is found that NN weights are transmitted over the communication network. In other words, a large amount of data should be transmitted. This may cause the channel congestion. In the future work, we can study intermittent communication and quantification to overcome this problem.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grants 61966026, 61941304 and 61673014, 62163030, and in part by Natural Science Foundation of Inner Mongolia under Grants 2020BS06004 and 2019BS06006.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

- Krstic M, Kanellakopoulos I, Kokotovic P (1995) Nonlinear and adaptive control design. Wiley, New York
- Shao X, Si H, Zhang W (2021) Event-triggered neural intelligent control for uncertain nonlinear systems with specified-time guaranteed behaviors. *Neural Comput Appl* 33:5771–5791
- Yan H, Li Y (2017) Adaptive NN prescribed performance control for nonlinear systems with output dead zone. *Neural Comput Appl* 28:145–153
- Zhang T, Ge S, Hang C (2000) Adaptive neural network control for strict-feedback nonlinear systems using backstepping design. *Automatica* 6(12):1835–1846
- Ge S, Wang C (2002) Direct adaptive NN control of a class of nonlinear systems. *IEEE Trans Neural Netw* 13(1):214–221
- Li J, Chen W, Li J (2011) Adaptive NN output-feedback decentralized stabilization for a class of large-scale stochastic nonlinear strict-feedback systems. *Int J Robust Nonlinear Control* 21:452–472
- Tong S, Li Y (2012) Adaptive fuzzy output feedback tracking backstepping control of strict-feedback nonlinear systems with unknown dead zones. *IEEE Trans Fuzzy Syst* 20(1):168–180
- Liu Y, Tong S (2017) Barrier Lyapunov functions for Nussbaum gain adaptive control of full state constrained nonlinear systems. *Automatica* 76:143–152
- Liu J (2014) Observer-based backstepping dynamic surface control for stochastic nonlinear strict-feedback systems. *Neural Comput Appl* 24:1067–1077
- Wang W, Li Y (2019) Observer-based event-triggered adaptive fuzzy control for leader-following consensus of nonlinear strict-feedback systems. *IEEE Trans Cybern* 51(4):2131–2141
- Wang Z, Liu L, Wu Y, Zhang H (2018) Optimal fault-tolerant control for discrete-time nonlinear strict-feedback systems based on adaptive critic design. *IEEE Trans Neural Netw Learn Syst* 29(6):2179–2191
- Zhang J, Yang G (2019) Low-complexity tracking control of strict-feedback systems with unknown control directions. *IEEE Trans Autom Control* 64(12):5175–5182
- Song S, Zhang B, Song X, Zhang Z (2019) Neuro-fuzzy-based adaptive dynamic surface control for fractional-order nonlinear strict-feedback systems with input constraint. *IEEE Trans Syst Man Cybern Syst* 51(6):3575–3586
- Li H, Zhao S, He W, Lu R (2019) Adaptive finite-time tracking control of full state constrained nonlinear systems with dead-zone. *Automatica* 100:99–107
- Xu B, Shou Y, Luo J, Pu H, Shi Z (2019) Neural learning control of strict-feedback systems using disturbance observer. *IEEE Trans Neural Netw Learn Syst* 30(5):1296–1307
- Cui G, Jiao T, Wei Y, Song G, Chu Y (2014) Adaptive neural control of stochastic nonlinear systems with multiple time-varying delays and input saturation. *Neural Comput Appl* 25:779–791
- Liu L, Liu Y, Tong S (2019) Neural networks-based adaptive finite-time fault-tolerant control for a class of strict-feedback switched nonlinear systems. *IEEE Trans Cybern* 49(7):2536–2545
- Wang D, Huang J (2005) Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form. *IEEE Trans Neural Netw* 16(1):195–202
- Li T, Wang D, Feng G, Tong S (2010) A DSC approach to robust adaptive NN tracking control for strict-feedback nonlinear systems. *IEEE Trans Syst Man Cybern Part B Cybern* 40(3):915–927
- Huang J (2015) Global adaptive neural dynamic surface control of strict-feedback systems. *Neurocomputing* 165:403–413
- Peng Z, Wang D, Wang J (2017) Predictor-based neural dynamic surface control for uncertain nonlinear systems in strict-feedback form. *IEEE Trans Neural Netw Learn Syst* 28(9):2156–2167
- Xu B, Sun F (2018) Composite intelligent learning control of strict-feedback systems with disturbance. *IEEE Trans Cybern* 48(2):730–741
- Zhang T, Xia M, Yi Y (2017) Adaptive neural dynamic surface control of strict-feedback nonlinear systems with full state constraints and unmodeled dynamics. *Automatica* 81:232–239
- Chen M, Tao G, Jiang B (2015) Dynamic surface control using neural networks for a class of uncertain nonlinear systems with input saturation. *IEEE Trans Neural Netw Learn Syst* 26(9):2086–2097
- Wang M, Wang C (2015) Learning from adaptive neural dynamic surface control of strict-feedback systems. *IEEE Trans Neural Netw Learn Syst* 26(6):1247–1259
- Swaroop D, Hedrick J, Yip P, Gerdes J (2000) Dynamic surface control for a class of nonlinear systems. *IEEE Trans Autom Control* 45(10):1893–1899
- Yip P, Hedrick J (1998) Adaptive dynamic surface control: a simplified algorithm for adaptive backstepping control of nonlinear systems. *Int J Control* 71(5):959–979
- Song B, Hedrick J (2004) Observer-based dynamic surface control for a class of nonlinear systems: an LMI approach. *IEEE Trans Autom Control* 49(11):1995–2001
- Park J, Kim S, Moon C (2009) Adaptive neural control for strict-feedback nonlinear systems without backstepping. *IEEE Trans Neural Netw* 20(7):1204–1209
- Wang C, Hill D (2006) Learning from neural control. *IEEE Trans Neural Netw* 17(1):130–146
- Liu T, Wang C, Hill DJ (2009) Learning from neural control of nonlinear systems in normal form. *Syst Control Lett* 58(9):633–638
- Dai S, Wang C, Wang M (2014) Dynamic learning from adaptive neural network control of a class of nonaffine nonlinear systems. *IEEE Trans Neural Netw Learn Syst* 25(1):111–123
- Wang M, Wang C (2015) Neural learning control of pure-feedback nonlinear systems. *Nonlinear Dyn* 79:2589–2608
- Wang M, Wang C, Shi P, Liu X (2016) Dynamic learning from neural control for strict-feedback systems with guaranteed predefined performance. *IEEE Trans Neural Netw Learn Syst* 27(12):2564–2576

35. Wang M, Yang A (2017) Dynamic learning from adaptive neural control of robot manipulators with prescribed performance. *IEEE Trans Syst Man Cybern Syst* 47(8):2244–2255
36. Dai S, He S, Wang M, Yuan C (2019) Adaptive neural control of underactuated surface vessels with prescribed performance guarantees. *IEEE Trans Neural Netw Learn Syst* 30(12):3686–3698
37. Chen W, Hua S, Zhang H (2015) Consensus-based distributed cooperative learning from closed-loop neural control systems. *IEEE Trans Neural Netw Learn Syst* 26(2):331–345
38. Ai W, Chen W, Hua S (2018) Distributed cooperative learning for a group of uncertain systems via output feedback and neural networks. *J Frankl Inst* 355(5):2536–2561
39. Yuan C, He H, Wang C (2019) Cooperative deterministic learning-based formation control for a group of nonlinear uncertain mechanical systems. *IEEE Trans Ind Inform* 15(1):319–333
40. Gao F, Chen W, Li Z, Li J, Xu B (2020) Neural network-based distributed cooperative learning control for multiagent systems via event-triggered communication. *IEEE Trans Neural Netw Learn Syst* 31(2):407–419
41. Laub A (2005) *Matrix analysis for scientists and engineers*. SIAM, Philadelphia
42. Park J, Sandberg I (1991) Universal approximation using radial-basis-function networks. *Neural Comput* 3(2):246–257
43. Sanner R, Slotine J (1992) Gaussian networks for direct adaptive control. *IEEE Trans Neural Netw* 3(6):837–863
44. Cui B, Xia Y, Liu K, Shen G (2020) Finite-time tracking control for a class of uncertain strict-feedback nonlinear systems with state constraints: a smooth control approach. *IEEE Trans Neural Netw Learn Syst* 31(11):4920–4932
45. Bthtash S (1990) Robust output tracking for non-linear systems. *Int J Control* 51(6):1381–1407
46. Ge SS, Hang CC, Lee T, Zhang T (2001) *Stable adaptive neural network control*. Kluwer, Norwell
47. Yang YS, Ren JS (2003) Adaptive fuzzy robust tracking controller design via small gain approach and its application. *IEEE Trans Fuzzy Syst* 11(6):783–795

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