Hindawi
Complexity
Volume 2018, Article ID 5950678, 15 pages
https://doi.org/10.1155/2018/5950678

Research Article

A Stable Distributed Neural Controller for Physically Coupled Networked Discrete-Time System via Online Reinforcement Learning

Jian Sun [1,2] **and Jie Li** [3]

[1] *School of Electronic and Information Engineering, Southwest University, Chongqing, China*
[2] *Chongqing University Key Laboratory of Networks and Cloud Computing Security, Chongqing, China*
[3] *State Grid Chongqing Electric Power Co. Electric Power Research Institute, Chongqing, China*

Correspondence should be addressed to Jian Sun; cq_jsun@163.com

Received 28 July 2017; Revised 21 November 2017; Accepted 21 December 2017; Published 7 February 2018

Academic Editor: Christopher P. Monterola

The large scale, time varying, and diversification of physically coupled networked infrastructures such as power grid and transportation system lead to the complexity of their controller design, implementation, and expansion. For tackling these challenges, we suggest an online distributed reinforcement learning control algorithm with the one-layer neural network for each subsystem or called agents to adapt the variation of the networked infrastructures. Each controller includes a critic network and action network for approximating strategy utility function and desired control law, respectively. For avoiding a large number of trials and improving the stability, the training of action network introduces supervised learning mechanisms into reduction of long-term cost. The stability of the control system with learning algorithm is analyzed; the upper bound of the tracking error and neural network weights are also estimated. The effectiveness of our proposed controller is illustrated in the simulation; the results indicate the stability under communication delay and disturbances as well.

1. Introduction

The increasing interconnection of physical systems through cybernetworks or physical networks has been observed in many infrastructures, such as power grid [1, 2], transportation networks, and unmanned systems. One critical issue of these called cyberphysical systems is complexity of the system when it grows very large, especially the control problem. Consequently, distributed schemes are suggested for reducing the communication and computational cost compared with centralized control scheme [3]. However, the coupling of subsystems and nonstatic environment in both cybernetworks and physics networks bring many challenges, such as physical interference among subsystems, time-varying plant parameters, communication delay, and expansibility of the cyberphysical system.

To increase expansibility of the cyberphysical system, the multiagent concept is usually introduced. The cyberphysical system can be divided into many agents. Each agent has its own control policy and a unified framework for pursuing its target [4]. The expansion of the cyberphysical system turns into simply duplicating agents without accommodating control policy. To deal with the physical coupling of networked system, one common approach is to decouple subsystems in control design [5–8]. Each subsystem may utilize state information of neighbored subsystems for mitigating their physical interference, or the designer treats their physical interference as random disturbance [9, 10]. On the other hand, for addressing nonstatic environment with time-varying plants, online supervised learning, adaptive control, and reinforcement learning algorithm are suggested; they all enable adaptively adjusting their control parameters online, while the combination of neural network and reinforcement learning usually leads to better control performance compared with conventional supervised learning and adaptive control scheme [11]. Reinforcement learning constructs a long-run cost-to-go function to predict the consequence cost;, each control action takes the estimated future result into account [12], while, compared with adaptive control, the adaptive ability is limited in the number of time-varying

parameters; the number of time-varying parameters of plant model may very large in practice.

Recently, many researches are focused on reinforcement learning with neural network. These researches are classified into two categories. The first category is to simply utilize neural network to approximate unknown part about system model or control strategy, such as cost-to-go function and optimal control law. Prokhorov and Wunsch discussed three families of reinforcement learning control design [13], Heuristic dynamic programming (HDP), dual heuristic programming (DHP), and globalized dual heuristic programming (GDHP) and their application in optimal control. Xu et al. focus on experimental studies of real-time online learning control for nonlinear systems using kernel-based ADP methods [14]. Lee et al. focus on a class of reinforcement learning (RL) algorithms, named integral RL (I-RL), that solve continuous-time (CT) nonlinear optimal control problems with input affine system dynamics [15]. The second category is to combine the approach in the first category with supervised learning algorithm for guaranteeing convergence of the learning system; the supervised reinforcement learning also reduces a large number of trials by employing the error signal with domain knowledge [16–18]. It generates instinct feedback for correcting the control actions. Xu et al. suggest a novel adaptive-critic-based neural network (NN) controller which is investigated for nonlinear pure-feedback systems [19]. Liu et al. were concerned with a reinforcement learning-based adaptive tracking control technique to tolerate faults for a class of unknown multiple-input multiple-output nonlinear discrete-time systems with less learning parameters [20]. Besides these, researchers try to employing multilayer/deep neural network for approximating the functions in control, so that the precision of model is enhanced and the performance can be improved in a consequence [21, 22]. However, it is hard to analyze its stability of learning algorithm. Moreover, the learning rate may be slow as the number of tuned parameters is very large in the deep neural network [23].

In this paper, we suggest a distributed neural controller for the physically coupled networked discrete-time system via online reinforcement learning. We model each subsystem as an agent; each agent can obtain its state and some physical neighbored subsystem state information to figure out optimal control action. One-layer adaptive critic neural network and action neural network are proposed for modeling the cost function and optimal action law. With deterministic learning algorithm, we incorporated supervised learning into our reinforcement learning algorithm for accelerating convergence rate. The stability of the learning algorithm is analyzed and the boundary of each parameter is also estimated. The contribution of this paper is two-fold.

(1) We propose a distributed online reinforcement learning algorithm for controlling physically coupled networked discrete-time system.

(2) Sufficient condition for guaranteeing learning algorithm stability and system stability are derived and the upper bound of parameters is estimated.

The rest of the paper is organized as follows: We model the physically coupled networked system and control system in a mathematical dynamic equation in Section 2, and some assumptions are made for simplifying the analysis; then, control system design via online reinforcement learning algorithm is depicted in Section 3; the stability analysis is detailedly discussed in Section 4; simulation results for illustrating the effectiveness and advantage of our algorithm are elaborated in Section 5. Section 6 is the conclusion part.

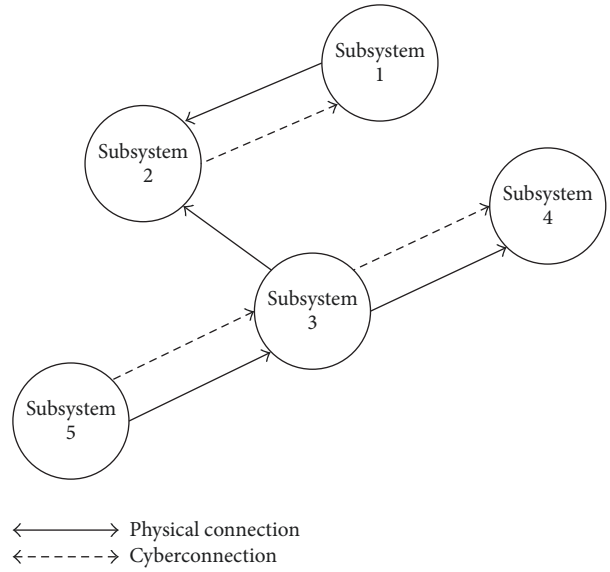

Figure 1: A physical-coupling networked system structure.

2. Physically Coupled Networked Control System and Problem Statement

In the physically coupled networked system, their subsystems may physically interfere with neighbored subsystems and change its state trajectory or dynamic. The structure is shown in Figure 1. In order to improve the control system performance, some cyberconnections of communication infrastructures are installed for exchanging the states of neighbored subsystems [3]. The topology of cyberconnections and physical connections may not be the same for probably practical constraints in cyberresources.

2.1. System Dynamic Equation. For a physically coupled networked system, consider that it consists of n nonlinear dynamic subsystems, which are given in the discrete-time form:

$$\begin{aligned} x_{i,1}(k+1) &= x_{i,1}(k), \\ x_{i,2}(k+1) &= x_{i,3}(k), \\ &\vdots \\ x_{i,p}(k+1) &= f_i(x_i(k)) + \sum_{j \in N_p(i)} f_{ji}\left(x_{j,p}(k)\right) \\ &\quad + g_i(x_i(k))\, u_i(k) + d_i(k), \end{aligned} \tag{1}$$

where $i = 1, \ldots, n$, $x_{i,l} \in R^{m\times 1}$, $l = 1, 2, \ldots, p$, and $x_i = (x_{i,1}{}^T, \ldots, x_{i,p}{}^T)^T \in R^{pm\times 1}$. $x_i, u_i \in R^{q\times 1}$ and $d_i \in R^{m\times 1}$ are system state vector, control input vector, and disturbance vector for subsystem i. $f_i(x_i(k)) \in R^{m\times 1}$, $f_{ji}(x_{j,p}(k)) \in R^{m\times 1}$, and $g_i(x_i(k)) \in R^{m\times q}$ are smooth vector function about local system dynamic, neighbor interference, and control input interference which are all unknown. $N_p(i)$ is physically connected neighbor set of subsystems i, which can interfere with the state trajectory of subsystem i. In order to simplify the analysis, the following reasonable assumptions are made [11].

Assumption 1. The disturbances are bounded $\|d_i\| \le d_{\max}$.

Assumption 2. g_i is an invertible matrix.

$d_{\max}$ is a positive real number, it means the magnitude of disturbances are bounded. Assumption 2 is made for simplifying the analysis of action network which will be discussed in next section.

The control objective is to track the state target vector y_i; then we have the error equation

$$e_{i,l}(k) = x_{i,l}(k) - y_{i,d}(k + l - p). \tag{2}$$

Therefore, the subsystem dynamic in a form of error is

$$\begin{aligned} e_{i,1}(k+1) &= e_{i,2}(k), \\ e_{i,2}(k+1) &= e_{i,3}(k), \\ &\vdots \\ e_{i,p}(k+1) &= f_i(x_i(k)) + \sum_{j\in N_p(i)} f_{ji}\left(x_{j,p}(k)\right) \\ &\quad + g_i(x_i(k))u_i(k) + d_i(k) - y_{i,p}(k+1). \end{aligned} \tag{3}$$

2.2. Distributed Control System and Control Objective. Distributed control system is more flexible and scalable than centralized control. Moreover, it divides a large system controller into many small subsystems controllers, which lead to the system state dimension reduction in a controller, so that much computational resource and time can be saved [24].

The control objective is to decrease the error vector e_i as fast as possible and bound in a small region for a given bounded disturbance. For subsystem controller, usually, an exponential damping rate of error is expected with a form of

$$e_i(k+1) = \Gamma_i e_i(k), \tag{4}$$

where $\|\Gamma_i\| < 1$. Therefore, the desired control input of subsystem i can be in a form of

$$\begin{aligned} u_{d,i}(k) = -g_i(x_i(k))^{-1} \Bigg[&\sum_{j\in N_p(i)\cap N_c(i)} f_{ji}\left(x_{j,p}(k)\right) \\ &+ f_i(x_i(k)) - y_{i,p}(k+1) - \Gamma_i e_{i,p}(k) \Bigg]. \end{aligned} \tag{5}$$

$N_c(i)$ is the cyberconnected neighbor set of subsystems i, which means the controller of subsystem i utilizes the received state information from neighbored subsystems via communication network.

However, $f_i(x_i(k))$, $f_{ji}(x_{j,p}(k))$, and $g_i(x_i(k))$ are unknown. A reinforcement learning scheme with neural network is proposed for approximating the desired control strategy and strategy utility function about long-term cost.

3. Control System Design by Reinforcement Learning and Neural Network

The proposed distributed control scheme with reinforcement learning consists of three parts: the first part will introduce a strategy utility function (also called long-term cost function); the second part depicts the critic neural network and online training algorithm; the last part of this section elaborates the action neural network and parameter updating algorithm.

3.1. Strategy Utility Function. The utility function defined for subsystem i is based on the current filtered state error $e_i(k)$; it is formulated as

$$p_i^l(k) = \begin{cases} 0, & \text{if } e_i^l(k)^2 \le c_i^l \\ 1, & \text{otherwise,} \end{cases} \tag{6}$$

where $l = 1, 2, \ldots, mp$, $p_i^l(k) \in R$, and c_i^l is a given constant positive scalar threshold for lth element of state error vector e_i for subsystem i. $p_i^l(k)$ is also an indicator of current tracking performance; if $p_i^l(k)$ equals 1, it means the control system has a bad state, and the state deviates the desired value a lot. On the other hand, if $p_i^l(k)$ equals 0, it indicates well-tracking performance and the lth state error is in a small bounded region.

The long-term cost is the sum of utility function at each sampling time. Based on the utility function $p_i^l(k)$, strategic utility function is defined as

$$\begin{aligned} J_i(k) = \alpha^N p_i(k+1) + \alpha^{N-1} p_i(k+2) + \cdots \\ + \alpha^{k+1} p_i(N) + \cdots, \end{aligned} \tag{7}$$

where $0 < \alpha < 1$, $p_i(k) \in R^{mp}$, and N is stage number. If N is infinite or very large, the strategy utility function is defined in a rolling horizon with a fixed number of stages. It is obvious that the control objective is to minimize $J_i(k)$ which improve the control performance.

3.2. Critic Network Design. In our proposed scheme, one-layer neural network is considered for approximating strategy utility function J_i. For simplifying the stability analysis, only output layer weights of neural network are designed to be adjustable in online training. A one-layer network is suggested to approximate strategy utility function; it is

$$J_i(k) = W_{c,i}(k)\,\phi_{c,i}(k - \tau) + \varepsilon_{c,i}(k). \tag{8}$$

The basis function $\phi_{c,i}(k - \tau)$ is a Gaussian vector function which is defined as

$$\begin{aligned} \phi_{c,i}(k-\tau) &= \left[\phi_{c,i,1}, \phi_{c,i,2}, \ldots, \phi_{c,i,h}, \ldots, \phi_{c,i,q}\right], \\ \phi_{c,i,h}(k-\tau) &= \exp\left(-\frac{\left\|e_i(k-\tau) - c_{c,i,h}\right\|^2}{{\sigma_{c,i}}^2}\right), \end{aligned} \tag{9}$$

where τ is communication latency, $c_{c,i,h} \in R^{mp}$ is the Gaussian function center vector, and the centers should cover the system operation state region as much as possible. $\sigma_{c,i}$ is width of Gaussian function. The approximation error $\varepsilon_{c,i}$ would be very small if the dimension of basis function $\phi_{c,i} \in R^{q\times 1}$ is large enough [11]. The relation between kth and $(k+1)$th optimal control action is

$$J_i(k) = \min_{u_i(k)} \left\{\alpha J_i(k-1) - \alpha^{N+1} p_i(k)\right\}, \tag{10}$$

where $u_i(k) \in R^m$ is control action for subsystem i. We estimate the strategy utility function by

$$\widehat{J}_i(k) = \widehat{W}_{c,i}(k)\,\phi_{c,i}(k-\tau). \tag{11}$$

The prediction error of approximated strategy utility function $\widehat{J}_i$ for critic NN is

$$e_{c,i}(k) = \widehat{J}_i(k) - \alpha\left(\widehat{J}_i(k-1) - \alpha^N p_i(k)\right). \tag{12}$$

We define the objective function of critic NN for minimization at kth sampling as

$$E_{a,i}(k) = \frac{1}{2} e_{a,i}^T(k)\, e_{a,i}(k). \tag{13}$$

One common way to decrease the objective function is to update critic NN parameters along its gradient direction. Applying chain rule, partial derivative of objective function (13) with respect to $\widehat{W}_{c,i}(k)$ is

$$\begin{aligned}\frac{\partial E_{a,i}(k)}{\partial \widehat{W}_{c,i}(k)} &= e_{c,i}(k)\,\phi_{c,i}(k)^T \\ &= \left[\widehat{J}_i(k) - \alpha\left(\widehat{J}_i(k-1) - \alpha^N p_i(k)\right)\right]\phi_{c,i}(k-\tau)^T.\end{aligned} \tag{14}$$

Therefore, updating law for critic NN of subsystem i is

$$\begin{aligned}\widehat{W}_{c,i}(k+1) &= \widehat{W}_{c,i}(k) \\ &\quad - \delta_i\left[\widehat{J}_i(k) - \alpha\left(\widehat{J}_i(k-1) - \alpha^N p_i(k)\right)\right] \\ &\quad \cdot \phi_{c,i}(k-\tau)^T.\end{aligned} \tag{15}$$

δ_i is a given scalar, representing updating step size. The choice of δ_i is very important. If δ_i is too large, the online learning may diverge.

3.3. Action Neural Network Design. Our control objective is to minimize the tracking error e_i and also to minimize the long-term cost function/strategy utility function J_i. They depend on the control action in each step. The desired control action (5) is an expected strategy for approaching this objective, and an action neural network is suggested for approximating the desired control action. The desired control action $u_{d,i}$ can be equal to

$$u_{d,i}(k) = W_{a,i}(k)\,\phi_{a,i}(k-\tau) + \varepsilon_{a,i}(k), \tag{16}$$

where $W_{a,i}$ is the optimal weighting matrix for neural output which minimizes the residual $\varepsilon_{a,i}$; $\phi_{a,i} \in R^{l\times 1}$ is the basis function which has the same form as (9). $\varepsilon_{a,i}$ would be very small if the dimension of $\phi_{a,i}$ is very large. However, $u_{d,i}$ and $W_{a,i}$ are unknown; the desired control action is proposed to be estimated by

$$\widehat{u}_i(k) = \widehat{W}_{a,i}\phi_{a,i}(k-\tau), \tag{17}$$

where $\widehat{W}_{a,i}$ is the estimated weighting matrix for $W_{a,i}$. And we have the estimated error $\widetilde{u}_i$ for desired control action.

$$\widetilde{u}_i(k) = \widehat{u}_i(k) - u_{d,i} = \widetilde{W}_{a,i}\phi_{a,i}(k-\tau) - \varepsilon_{a,i}(k), \tag{18}$$

where $\widetilde{W}_{a,i} = \widehat{W}_{a,i} - W_{a,i}$, and we denote $\varsigma_i(k) = \widetilde{W}_{a,i}\phi_{a,i}(k-\tau)$, which causes dynamic (3) to be

$$\begin{aligned} e_{i,1}(k+1) &= e_{i,2}(k), \\ e_{i,2}(k+1) &= e_{i,3}(k), \\ &\vdots \\ e_{i,p}(k+1) &= \Gamma_i e_{i,p}(k) + g_i\varsigma_i(k) \\ &\quad + \sum_{j\in N_p(i)\setminus N_c(i)} f_{ji}\left(x_{j,p}(k)\right) - g_i\varepsilon_i(k) + d_i(k). \end{aligned} \tag{19}$$

$N_p(i)$ and $N_c(i)$ are the neighbor subsystem sets of subsystem i which are connected to subsystem i in physical way and cyberway. In our proposed scheme, supervised learning is incorporated into the action neural network training for accelerating the convergence rate of online updating. The objective of the policy is not only to minimize long-term cost J_i but also to approximate the desired control output $u_{d,i}$ with supervised learning. Thus, the error vector of action network is defined as

$$e_a(k) = \sqrt{g_i}\varsigma_i(k) + \sqrt{g_i}^{-1}\left(\widehat{J}_i(k) - J_{d,i}\right), \tag{20}$$

where $J_{d,i}$ is the desired utility function value for subsystem i, it can be set as 0 [20], and $\sqrt{g_i}$ is principal mean square root. The following cost function is defined for each step:

$$E_{a,i}(k) = \frac{1}{2} e_{a,i}(k)^T e_{a,i}(k). \tag{21}$$

Then, the partial derivative of (21) with respect to $\widetilde{W}_{a,i}$ is obtained by chain rule.

$$\begin{aligned}\frac{\partial E_{a,i}(k)}{\partial \widetilde{W}_{a,i}(k)} &= \frac{\partial E_{a,i}(k)}{\partial e_{a,i}(k)}\frac{\partial e_{a,i}(k)}{\partial \varsigma_i(k)}\frac{\partial \varsigma_i(k)}{\partial \widetilde{W}_{a,i}(k)} \\ &= \phi_{a,i}(k)\left[g_i\varsigma_i(k) + \widehat{J}_i(k)\right]^T.\end{aligned} \tag{22}$$

Therefore, with gradient descent principle, the action NN weight matrix is updated by

$$\begin{aligned}\widetilde{W}_{a,i}(k+1) &= \widetilde{W}_{a,i}(k) \\ &\quad - \beta_i\left[g_i\varsigma_i(k) + \widehat{J}_i(k)\right]\phi_{a,i}(k-\tau)^T.\end{aligned} \tag{23}$$

β_i is the updating step size for online learning of action neural network. The choice of β_i will be discussed in the next section, which is associated with the stability of the online learning algorithm.

4. Stability Analysis

This section discusses the stability of online learning algorithm and the tracking performance. It is necessary for control design. The upper bound of error and weight parameter of neural networks are analyzed. Firstly, a theorem about the stability of this scheme is proposed.

Theorem 3. *For a given networked control system described in (3) and the parameter updating algorithm in (15) (23), if* $A_1 \leq 0$, $A_2 \leq 0$, $A_3 \leq 0$, *and* $A_4 \leq 0$, *where*

$$
\begin{aligned}
A_1 &= -\frac{\mu_{i,1}}{2}\left[1-\kappa_i\Gamma_i^{max}\right]+\sum_{i\in N_p(j)\backslash N_c(j)}\frac{\mu_{j,1}\kappa_j f_{ij}^{max}}{2},\\
A_2 &= -\frac{\mu_{i,2}}{2}\beta_i\left[2g_i^{min}-\frac{1}{g_i^{max}}-2\beta_i g_i^{max}\left\|\phi_{a,i}(k-\tau)\right\|^2\right]\\
&\quad+\frac{\mu_{i,1}}{2}\kappa_i g_i^{max},\\
A_3 &= \frac{\mu_{i,2}}{2}\left[\beta_i g_i^{max}+2\beta_i^2\left\|\phi_{a,i}(k-\tau)\right\|^2\right]\\
&\quad-\frac{\mu_{i,3}}{2}\delta_i\left[1-\alpha^2-\alpha^{2(N+1)}-3\delta_i\left\|\phi_{c,i}(k-\tau)\right\|^2\right]\\
&\quad+\frac{\mu_{i,4}}{2},\\
A_4 &= \frac{\mu_{i,3}}{2}\left[\delta_i+3\alpha\delta_i^2\left\|\phi_{c,i}(k-\tau)\right\|^2\right]-\frac{\mu_{i,4}}{2},\\
A_5 &= \frac{\mu_{i,3}}{2}\left[3\delta_i^2\left\|\phi_{c,i}(k-\tau)\right\|^2+1\right],\\
A_6 &= \frac{\kappa_i\mu_{i,1}}{2},
\end{aligned}
\tag{24}
$$

where $\kappa_i = 3+|N_p(i)\backslash N_c(i)|$, $g_i^{max}=\lambda_{max}(g_i^T g_i)$, $\Gamma_i^{max}=\lambda_{max}(\Gamma_i)^2$, $g_i^{min}=\lambda_{max}(g_i{}^T g_i)^{1/2}$, *and* $f_i^{max}=\lambda_{max}(\partial f_{ji}^T/\partial e_{j,p}\partial f_{ji}/\partial e_{j,p})$. *Then, there exist upper bounds for* $\|e_i(k)\|^2$ $\|\varsigma_i(k)\|^2$ $\|\widetilde{J}_i(k)\|^2$, *when* $t\to+\infty$, *and they are*

$$
\begin{aligned}
\left\|e_i(k)\right\|^2 &\leq -\frac{A_5+A_6 {d_{max}}^2}{A_1},\\
\left\|\varsigma_i(k)\right\|^2 &\leq -\frac{A_5+A_6 {d_{max}}^2}{A_2},\\
\left\|\widetilde{J}_i(k)\right\|^2 &\leq -\frac{A_5+A_6 {d_{max}}^2}{A_3}.
\end{aligned}
\tag{25}
$$

And the system is stable.

Proof of Theorem 3. For the dynamic system described in (3), (15), and (23), we first define a Lyapunov function which consisted of quadratic of tracking error, action network weight error, and the error of critic neural network. It is

$$
V(k)=\sum_{i\in S}V_i(k),
\tag{26}
$$

where

$$
\begin{aligned}
V_i(k) &= V_{i,1}(k)+V_{i,2}(k)+V_{i,3}(k)+V_{i,4}(k),\\
V_{i,1}(k) &= \frac{\mu_{i,1}}{2}e_{i,p}(k)^T e_{i,p}(k),\\
V_{i,2}(k) &= \frac{\mu_{i,2}}{2}\operatorname{tr}\left[\widetilde{W}_{a,i}(k)^T\widetilde{W}_{a,i}(k)\right],\\
V_{i,3}(k) &= \frac{\mu_{i,4}}{2}\operatorname{tr}\left[\widetilde{W}_{c,i}(k)^T\widetilde{W}_{c,i}(k)\right],\\
V_{i,4}(k) &= \frac{\mu_{i,5}}{2}\operatorname{tr}\left[\widetilde{J}_i(k-1)^T\widetilde{J}_i(k-1)\right],
\end{aligned}
\tag{27}
$$

where $\widetilde{W}_{c,i}=\widehat{W}_{c,i}-W_{c,i}$ and $\widetilde{J}_{c,i}=\widetilde{W}_{c,i}\phi_{c,i}$. For a subsystem i, we have

$$
\begin{aligned}
\Delta V_{i,1} &= \frac{\mu_{i,1}}{2}\left[e_i(k+1)^T e_i(k+1)-e_i(k)^T e_i(k)\right]\\
&= \frac{\mu_{i,1}}{2}\left[\left\|\Gamma_i e_{i,p}(k)+g_i\varsigma_i(k)\right.\right.\\
&\quad+\sum_{j\in N_p(i)\backslash N_c(i)}f_{ji}\left(e_{j,p}(k)+y_{i,d}(k)\right)-g_i\varepsilon_i(k)\\
&\quad\left.\left.+d_i(k)\right\|^2-\left\|e_i(k)\right\|^2\right]\leq\frac{\mu_{i,1}}{2}\left\{-\left[1-\kappa_i\Gamma_i^{\max}\right]\right.\\
&\quad\cdot\left\|e_{i,p}(k)\right\|^2+\sum_{j\in N_p(i)\backslash N_c(i)}\kappa_i f_{ji}^{\max}\left\|e_{j,p}(k)\right\|^2\\
&\quad\left.+\kappa_i g_i^{\max}\left(\left\|\varepsilon_i(k)\right\|^2+\left\|\varsigma_i(k)\right\|^2\right)+\kappa_i d_i(k)\right\},
\end{aligned}
\tag{28}
$$

where $\kappa_i = 3+|N_p(i)\backslash N_c(i)|$, $g_i^{\max}=\lambda_{\max}(g_i^T g_i)$, $f_i^{\max}=\lambda_{\max}((\partial f_{ji}^T/\partial e_{j,p})(\partial f_{ji}/\partial e_{j,p}))$, $\Gamma_i^{\max}=\lambda_{\max}(\Gamma_i)^2$, and

$$
\begin{aligned}
\Delta V_{i,2} &= \frac{\mu_{i,2}}{2}\left\{\operatorname{tr}\left[\widetilde{W}_{a,i}(k+1)^T\widetilde{W}_{a,i}(k+1)\right]\right.\\
&\quad\left.-\operatorname{tr}\left[\widetilde{W}_{a,i}(k)^T\widetilde{W}_{a,i}(k)\right]\right\}\leq\frac{\mu_{i,2}}{2}\left\{-2\right.\\
&\quad\cdot\operatorname{tr}\left[\beta_i\varsigma_i(k)^T\left(g_i\varsigma_i(k)+\widehat{J}_i(k)\right)\right]\\
&\quad\left.+\beta_i^2\left\|\phi_{a,i}(k-\tau)\right\|^2\left\|g_i\varsigma_i(k)+\widehat{J}_i(k)\right\|^2\right\}\\
&\leq\frac{\mu_{i,2}}{2}\left\{-\beta_i\left[2g_i^{\min}-\frac{1}{g_i^{\max}}\right.\right.\\
&\quad\left.\left.-2\beta_i g_i^{\max}\left\|\phi_{a,i}(k-\tau)\right\|^2\right]\left\|\varsigma_i(k)\right\|^2+\left[\beta_i g_i^{\max}\right.\right.\\
&\quad\left.\left.+2\beta_i^2\left\|\phi_{a,i}(k-\tau)\right\|^2\right]\left\|\widehat{J}_i(k)\right\|^2\right\},
\end{aligned}
\tag{29}
$$

where $g_i^{\max} = \lambda_{\max}(g_i{}^T g_i)$ and $g_i^{\min} = \lambda_{\max}(g_i{}^T g_i)^{1/2}$. For strategy utility function, (10) leads to

$$J_i(k) = \alpha J_i(k-1) - \alpha^{N+1} p_i{}^*(k), \tag{30}$$

and $p_i{}^*$ is the utility function under the optimal strategy.

$$\begin{aligned} e_{c,i}(k) &= \widehat{J}_i(k) - \alpha \widehat{J}_i(k-1) + \alpha^{N+1} p_i(k) \\ &= \widehat{J}_i(k) - \alpha \widehat{J}_i(k-1) + \alpha^{N+1} p_i(k) - J_i(k) \\ &\quad + \alpha J_i(k-1) - \alpha^{N+1} p_i(k) \\ &= \widetilde{J}_i(k) - \alpha \widetilde{J}_i(k-1) + \alpha^{N+1} \widetilde{p}_i(k), \end{aligned} \tag{31}$$

and $\widetilde{p}_i(k) = p_i(k) - p_i{}^*(k)$, and $\|\widetilde{p}_i(k)\| \le 1$. The updating equation (15) yields

$$\begin{aligned} \Delta V_{i,3} &= \frac{\mu_{i,3}}{2} \left\{ \operatorname{tr}\left[\widetilde{W}_{c,i}(k+1)^T \widetilde{W}_{c,i}(k+1) \right] \right. \\ &\quad - \operatorname{tr}\left[\widetilde{W}_{c,i}(k)^T \widetilde{W}_{c,i}(k) \right] \Big\} \le \frac{\mu_{i,3}}{2} \left\{ -2\delta_i \operatorname{tr}\left[\widetilde{W}_{c,i}(k) \right. \right. \\ &\quad \cdot \left(\widetilde{J}_i(k) - \alpha \widetilde{J}_i(k-1) + \alpha^{N+1} \widetilde{p}_i(k) \right) \phi_{c,i}(k-\tau)^T \Big] \\ &\quad + \delta_i^2 \left\| \phi_{c,i}(k-\tau) \right\|^2 \left\| \widetilde{J}_i(k) - \alpha \widetilde{J}_i(k-1) \right. \\ &\quad \left. + \alpha^{N+1} \widetilde{p}_i(k) \right\|^2 \Big\} \le \frac{\mu_{i,3}}{2} \left\{ -\delta_i \left[1 - \alpha^2 - \alpha^{2(N+1)} \right. \right. \\ &\quad - 3\delta_i \left\| \phi_{c,i}(k-\tau) \right\|^2 \Big] \left\| \widetilde{J}_i(k) \right\|^2 + \left[\delta_i \right. \\ &\quad \left. \left. + 3\alpha\delta_i^2 \left\| \phi_{c,i}(k-\tau) \right\|^2 \right] \left\| \widetilde{J}_i(k-1) \right\|^2 \right. \\ &\quad \left. + \left[3\delta_i^2 \left\| \phi_{c,i}(k-\tau) \right\|^2 + 1 \right] \right\}. \end{aligned} \tag{32}$$

The last part of the variation of V is

$$\Delta V_{i,4} = \frac{\mu_{i,4}}{2} \left\{ \widetilde{J}_i(k)^T \widetilde{J}_i(k) - \widetilde{J}_i(k-1)^T \widetilde{J}_i(k-1) \right\}. \tag{33}$$

With the sum of all the above variations, we get

$$\begin{aligned} \Delta V &= \Delta V_{i,1} + \Delta V_{i,2} + \Delta V_{i,3} + \Delta V_{i,4} \\ &= A_1 \left\| e_{i,p}(k) \right\|^2 + A_2 \left\| \varsigma_i(k) \right\|^2 + A_3 \left\| \widetilde{J}_i(k) \right\|^2 \\ &\quad + A_4 \left\| \widetilde{J}_i(k-1) \right\|^2 + A_5 \\ &\quad + A_6 \left(\left\| \varepsilon_i(k) \right\|^2 + {d_{\max}}^2 \right), \end{aligned} \tag{34}$$

where A_1, A_2, A_3, A_4, A_5, and A_6 are given in Theorem 3. Therefore, if $A_1 \le 0$, $A_2 \le 0, A_3 \le 0$, and $A_4 \le 0$, the upper boundary of $\|x_i(k)\|^2 \|\varsigma_i(k)\|^2 \|\widehat{J}_i(k)\|^2$ can be estimated, when $t \to +\infty$, $\|\varepsilon_i(k)\|^2$ is very small which can be neglected if the dimension of basis function ϕ_i is large enough. The upper bound can be estimated by

$$\begin{aligned} \left\| x_i(k) \right\|^2 &\le -\frac{A_5 + A_6 d_{\max}^2}{A_1}, \\ \left\| \varsigma_i(k) \right\|^2 &\le -\frac{A_5 + A_6 d_{\max}^2}{A_2}, \\ \left\| \widehat{J}_i(k) \right\|^2 &\le -\frac{A_5 + A_6 d_{\max}^2}{A_3}. \end{aligned} \tag{35}$$

□

Remark 4. The stability of this system depends on the control parameters Γ_i, β_i, δ_i, and α, system functions g_i and f_{ji}, and the communication networks which affect parameter κ_i. It is obvious that if subsystem can obtain all state information from physically connected neighbors, the parameter κ_i would be smaller, it improves the system performance because the absolute value of A_1 and A_2 will be larger, and it decreases the upper bound of $e_i(k)$ and ς_i. Moreover, the sign of A_5 and A_6 cannot be necessarily definite, as they are not the coefficients of the estimated variable in the following Lyapunov function variation expression (34).

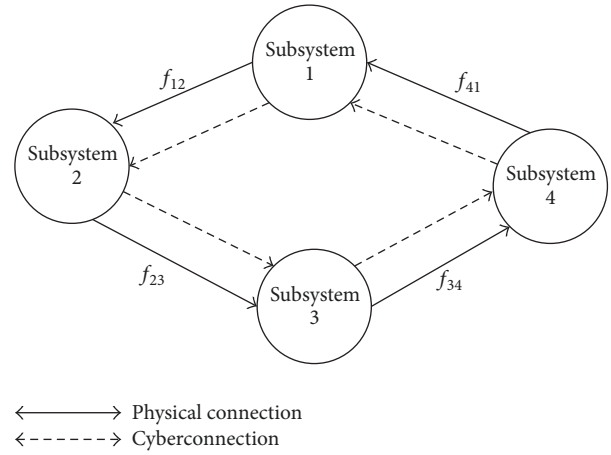

Figure 2: The structure of networked system I.

5. Simulation Results

This simulation illustrates the effectiveness and advantage of our proposed control scheme in four aspects: (1) The effectiveness of our proposed control scheme of physical coupling networked control system in tracking sine wave signal with disturbances; (2) its effectiveness with communication delay; (3) its advantages compared with conventional reinforcement learning; (4) its effectiveness in multicontrol input system.

The first simulation considers a networked system called system I as shown in Figure 2. System I consisted of four subsystems, each subsystem physically coupled with other subsystems. Each subsystem is a nonlinear system. Their equations are

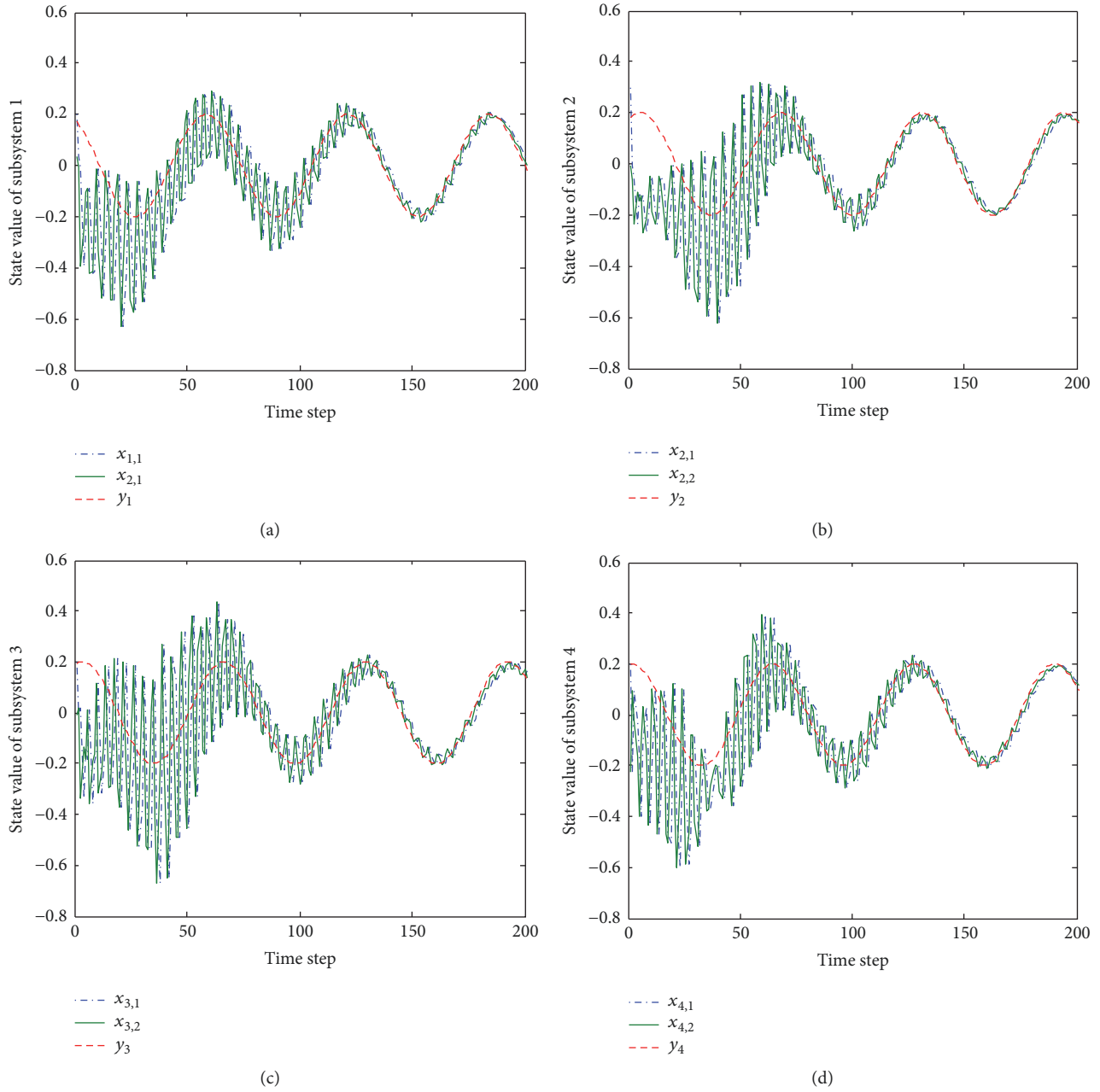

Figure 3: The state curves of system I with the proposed control scheme.

$$x_{i,1}(k+1) = x_{i,1}(k),$$
$$x_{i,2}(k+1) = -\frac{5}{8}\left[x_{i,1}(k) + x_{i,2}(k)^2\right] + 0.3x_{i,2}(k) + u(k) + \sum_{j\in N_p(i)} f_{ji}(x_i(k)) + d_i(k). \quad (36)$$

$i = 1, 2, 3, 4$. The initial value of states for this simulation are $x_1 = [0.1 \;\; 0.2]^T$, $x_2 = [0.1 \;\; 0.3]^T$, $x_3 = [0.1 \;\; 0.2]^T$, and $x_4 = [0.5 \;\; 0.2]^T$. The target signals for $x_{i,2}(k)$ $(i = 1, 2, 3, 4)$ are

$$y_1(k) = 0.2\sin(0.1k + 2),$$
$$y_2(k) = 0.2\sin(0.1k + 1),$$
$$y_3(k) = 0.2\sin(0.1k + 1.2),$$
$$y_4(k) = 0.2\sin(0.1k + 1.4). \quad (37)$$

The details of other functions and variables are listed in Table 1.

Figure 2 illustrates both the physical connection and the cyberconnection of system I. The communication network can send state information from subsystems 1 to 2, 2 to 3, 3 to 4, and 4 to 1. The parameters of the proposed controller are illustrated in Table 2.

The simulation results are shown in Figures 3 and 4. From Figure 3, it is obvious that all of the subsystems converge to

Table 1: Function and parameters in networked system I.

Function/parameter name	Description	Mathematical expression
f_{12}	The interference function on subsystem 2 from subsystem 1	$0.1x_{1,2}$
f_{23}	The interference function on subsystem 3 from subsystem 1	$0.01x_{2,1} + 0.05x_{2,2}$
f_{34}	The interference function on subsystem 3 from subsystem 1	$0.03x_{3,1} + 0.1x_{3,2}$
f_{41}	The interference function on subsystem 1 from subsystem 4	$0.1x_{4,1} + 0.002x_{4,2}$
d_1	The disturbance on subsystem 1	Gaussian noise with magnitude of 0.01
d_2	The disturbance on subsystem 2	Gaussian noise with magnitude of 0.01
d_3	The disturbance on subsystem 3	Gaussian noise with magnitude of 0.01
d_4	The disturbance on subsystem 4	Gaussian noise with magnitude of 0.01

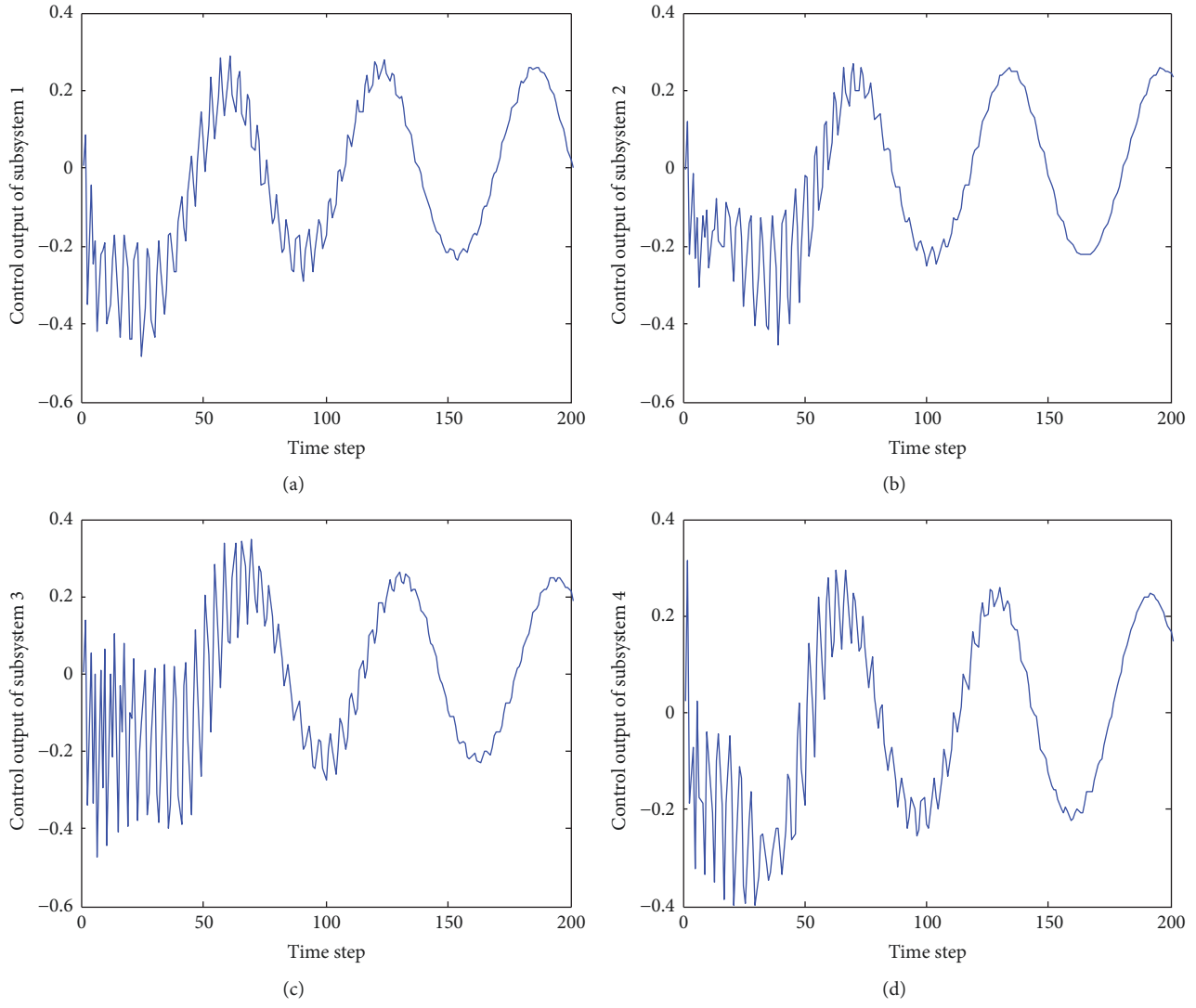

Figure 4: Control outputs of subsystem neural controllers with the proposed control scheme.

the target state with small errors. The curves converge to the target curves at about 125th control actions, which mean the online learning algorithm successfully obtained the desired action network and critic network. From Figure 4, it can be seen that the fluctuant of control output is decreased along the time during the online learning process. They also illustrate the effectiveness of our proposed control scheme.

In order to present the advantage of our suggested control scheme, we select conventional reinforcement learning without supervised learning scheme; the updating of action network solely depends on the backpropagation of critic network with the objective of minimizing the output of critic network [12]. The result is shown in Figure 5. The results explicitly indicate the divergence of the learning algorithm

Table 2: The parameters of controllers.

Parameter name	Description	Value
q	Dimension of basis vector function $\varphi_{c,i}$ for critic network. $i = 1, 2, 3, 4$.	40
l	Dimension of basis vector function $\varphi_{a,i}$ for action network. $i = 1, 2, 3, 4$.	40
$\sigma_{c,i}$	The width of radial basis function for critic network. $i = 1, 2, 3, 4$.	1.414
$\sigma_{a,i}$	The width of radial basis function for action network. $i = 1, 2, 3, 4$.	1.414
$c_{c,i,h}, c_{a,i,h'}$	The center vector for basis function. $i = 1, 2, 3, 4$. $h = 1, \ldots, q$. $h' = 1, \ldots, l$.	Element distributed in $[-1, +1]$
N	The horizon length of strategy utility function.	100
δ_i	The update rate for critic network weight matrix. $i = 1, 2, 3, 4$.	0.05
β_i	The update rate for action network weight matrix. $i = 1, 2, 3, 4$.	0.05
Γ_i	The damping rate of tracking error. $i = 1, 2, 3, 4$.	0.05

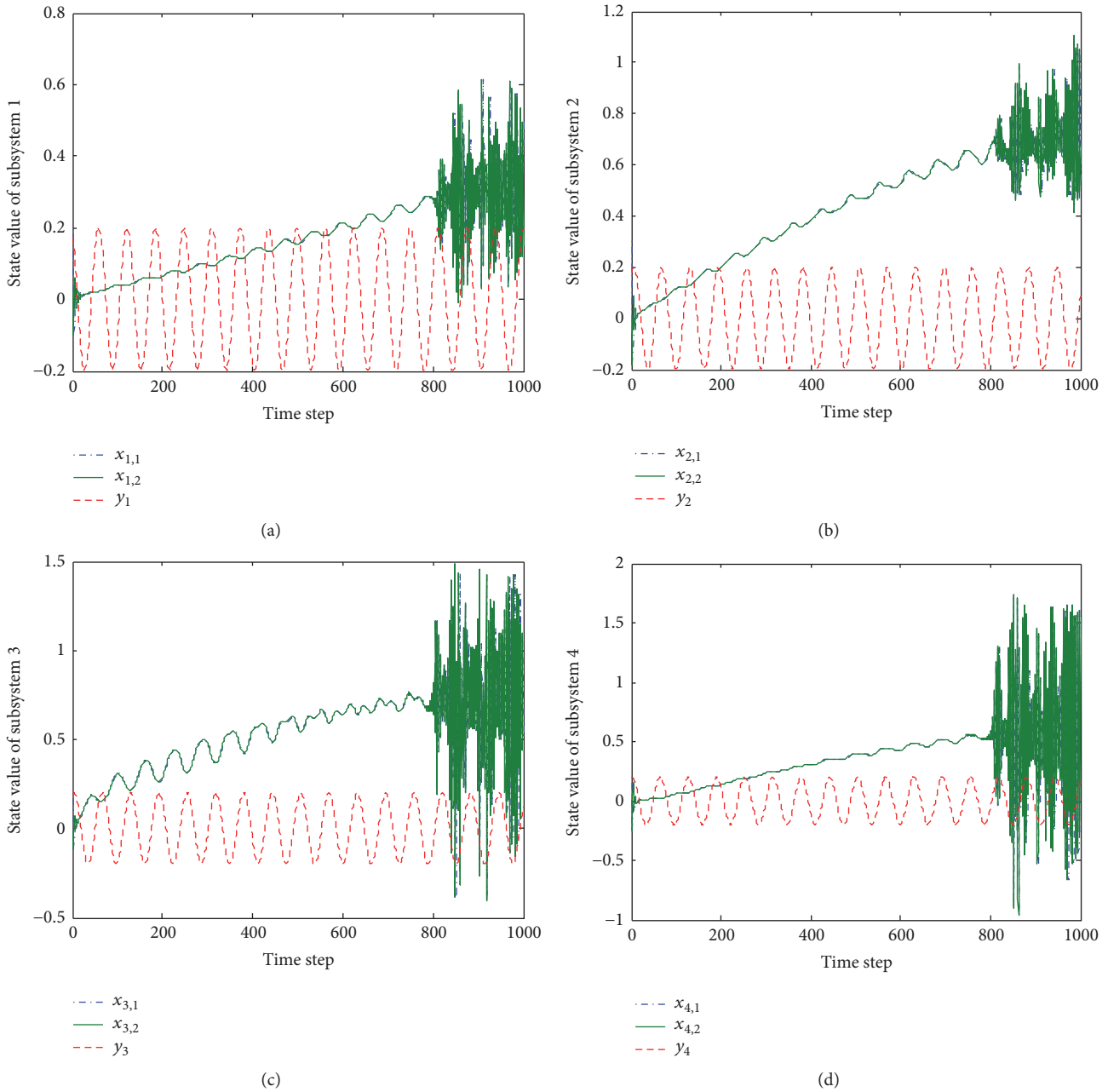

Figure 5: The state curve of subsystem with conventional reinforcement learning.

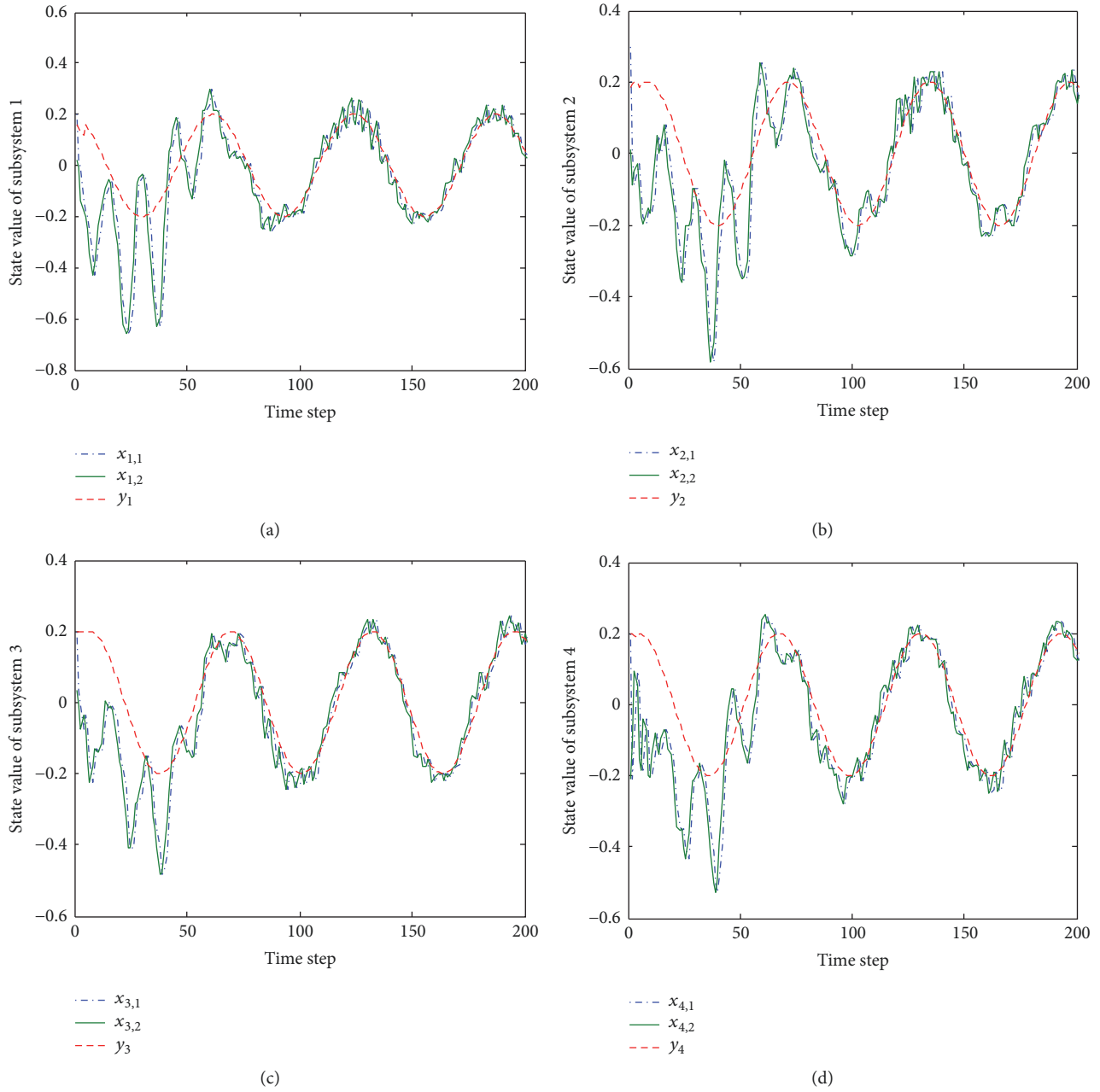

FIGURE 6: The state curves of subsystems with control delay $\tau = 3$ under the proposed control scheme.

because of the fast changing of target signal. And the conventional reinforcement learning may need off-line learning in advance. The results illustrate our proposed control scheme is more stable and has more powerful online learning ability than the conventional method.

In practice, the controller usually encounters action delay or communication delay. It is also modeled in our suggested model. To illustrate the effectiveness of our proposed control scheme under communication delay, we chose three communication delay values $\tau = 3, 5, 10$ to carry out the simulation. The simulation results are shown in Figures 6–8. These results show our proposed control scheme is stable under communication delay. However, static error increases with the communication delay. It is clear that the error of simulation with $\tau = 3$ is relatively smallest and the error of simulation with $\tau = 10$ is largest in the results.

For further demonstrating the effectiveness of our suggested scheme with multiple control input, we choose another system called system II for simulation. The model of system II is

$$x_{i,1}(k+1) = x_{i,2}(k),$$
$$x_{i,2}(k+1) = x_{i,3}(k),$$
$$x_{i,3}(k+1) = -\frac{5}{8}\left(x_{i,1}(k) + x_{i,3}(k)^2\right) + x_{i,1}(k) + \frac{1}{10 + x_{i,1}(k)^2 + x_{i,3}(k)^2} u_{i,1}(k)$$

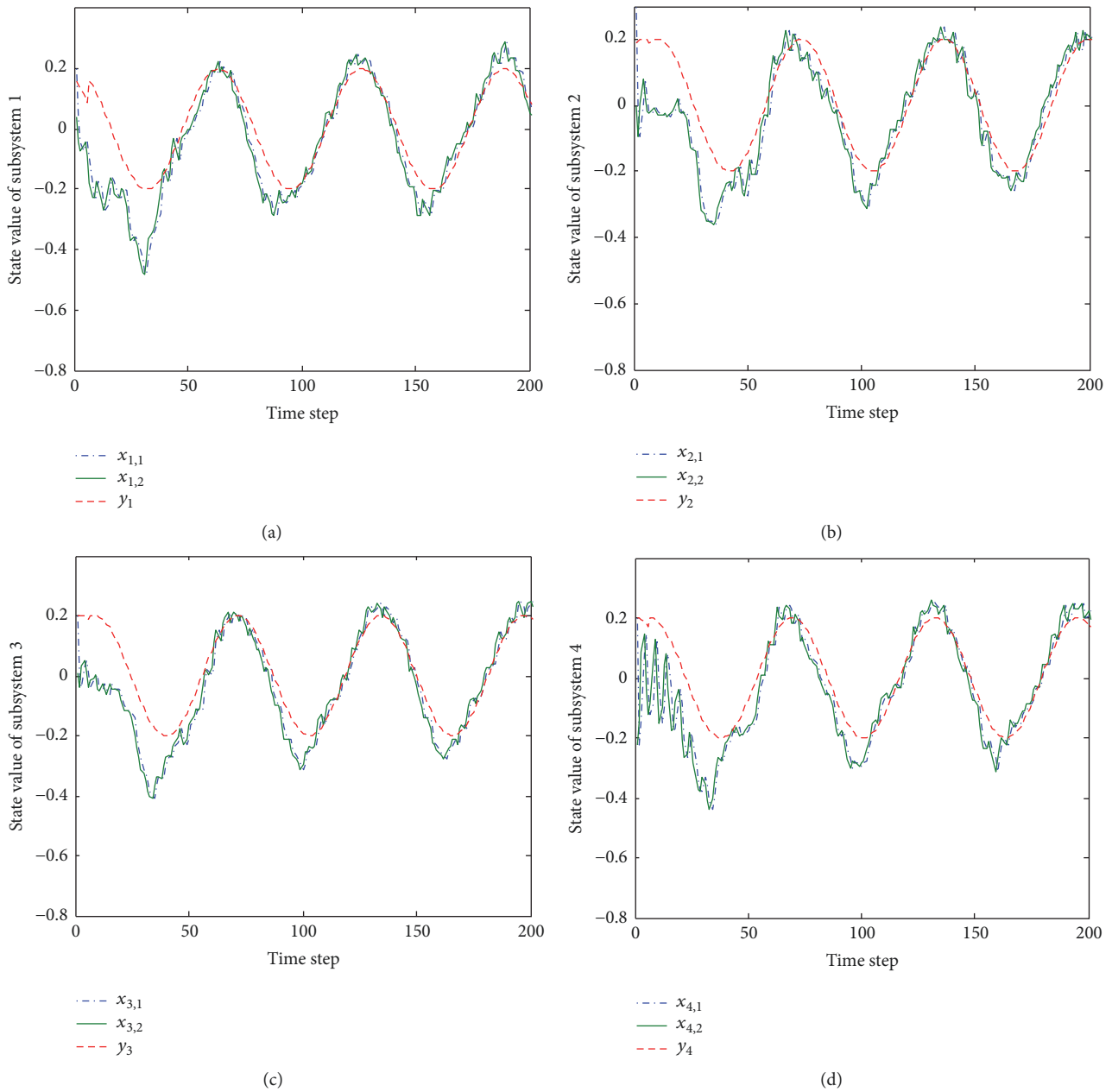

FIGURE 7: The state curves of subsystems with control delay $\tau = 5$ under the proposed control scheme.

$$
\begin{aligned}
&+ d_{i,1}(k) + \sum_{j \in N_p(i)} f_{ji,1}\left(x_j(k)\right), \\
x_{i,4}(k+1) &= -\frac{5}{8}\left(x_{i,2}(k) + x_{i,4}(k)^2\right) + x_{i,1}(k) \\
&+ \frac{1}{10 + x_{i,2}(k)^2 + x_{i,4}(k)^2} u_{i,2}(k) \\
&+ d_{i,2}(k) + \sum_{j \in N_p(i)} f_{ji,2}\left(x_j(k)\right).
\end{aligned} \tag{38}
$$

$i = 1, 2, 3, 4$. The system structure is the same as shown in Figure 2. The target signals for $x_{i,3}$ and $x_{i,4}$ are

$$
\begin{aligned}
y_{1,3}(k) &= 0.2 \sin(0.01k + 2), \\
y_{1,4}(k) &= 0.2 \sin(0.01k + 2), \\
y_{2,3}(k) &= 0.2 \sin(0.01k + 1), \\
y_{2,4}(k) &= 0.2 \sin(0.01k + 2.2), \\
y_{3,3}(k) &= 0.2 \sin(0.01k + 1.2), \\
y_{3,4}(k) &= 0.2 \sin(0.01k + 2.5), \\
y_{4,3}(k) &= 0.2 \sin(0.01k + 1.4), \\
y_{4,4}(k) &= 0.2 \sin(0.01k + 1).
\end{aligned} \tag{39}
$$

Other model parameters are illustrated in Table 3.

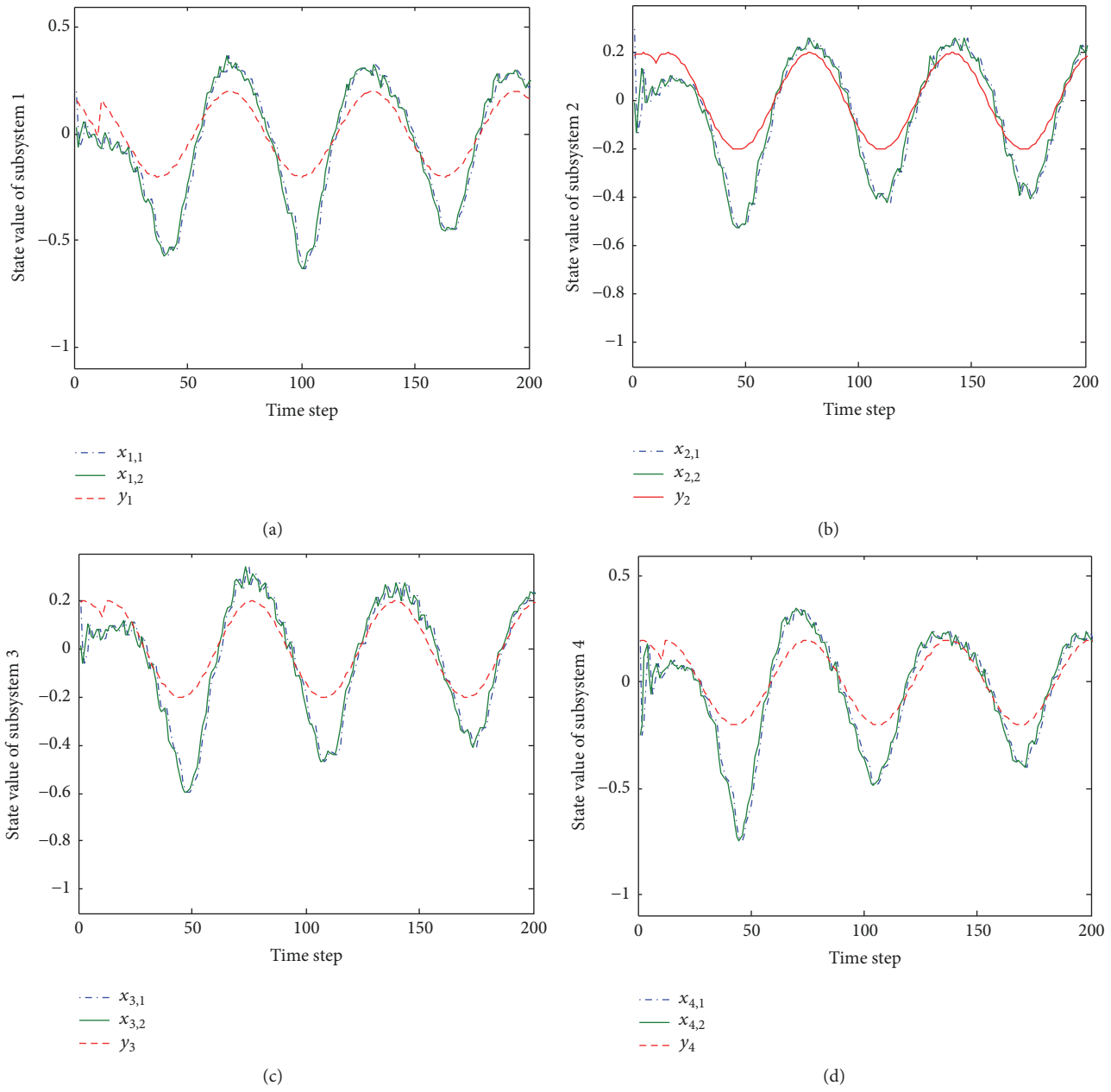

FIGURE 8: The state curves of subsystems with control delay $\tau = 10$ under the proposed control scheme.

The controller parameters are set as shown in Table 4.

The simulation results are shown in Figure 9. They show that all the subsystem states converge to the target signals within a small number of time steps (it is about 120). The tracking errors are small, and each of the state variables converges to its corresponding target signal. It illustrates the effectiveness of our suggested control scheme in application of multicontrol input systems with a relatively larger dimension compared with the previous simulation.

6. Conclusion

This paper suggests online reinforcement learning with one-layer neural network for controlling physically coupled networked system. It is a distributed learning control scheme. The networked system is divided into many subsystems; each system is an individual agent with controller and reinforcement learning algorithm. The reinforcement learning algorithm consists of the learning of critic network and action network. The critic network approximates the strategy utility function and the action network approximate the defined desired optimal controller. The action network weights updating decreases long-term cost with supervised learning mechanism by incorporating the desired control error ς with long-term cost function J. The effectiveness of our proposed controller is illustrated in the simulation part. The simulation results also indicate that the proposed control scheme improves the tracking performance compared with

Table 3: Model parameters of system II.

Function/parameter name	Description	Mathematical expression
f_{12}	The interference function on subsystem 2 from subsystem 1	$\begin{pmatrix} f_{12,3} \\ f_{12,4} \end{pmatrix} = \begin{pmatrix} 0.05 & 0.1 & 0.2 & 0.3 \\ 0.33 & 0.2 & 0.01 & 0.07 \end{pmatrix} x_1$
f_{23}	The interference function on subsystem 3 from subsystem 1	$\begin{pmatrix} f_{23,3} \\ f_{23,4} \end{pmatrix} = \begin{pmatrix} 0.01 & 0.05 & 0.01 & 0.07 \\ 0.04 & 0.04 & 0.1 & 0.3 \end{pmatrix} x_2$
f_{34}	The interference function on subsystem 3 from subsystem 1	$\begin{pmatrix} f_{34,3} \\ f_{34,4} \end{pmatrix} = \begin{pmatrix} 0.03 & 0.1 & 0.1 & 0.3 \\ 0.02 & 0.07 & 0.2 & 0.04 \end{pmatrix} x_3$
f_{41}	The interference function on subsystem 1 from subsystem 4	$\begin{pmatrix} f_{41,3} \\ f_{41,4} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.002 & 0.2 & 0.043 \\ 0.01 & 0.3 & 0.2 & 0.1 \end{pmatrix} x_4$
d_1	The disturbance on subsystem 1	Gaussian noise with magnitude of 0.05
d_2	The disturbance on subsystem 2	Gaussian noise with magnitude of 0.05
d_3	The disturbance on subsystem 3	Gaussian noise with magnitude of 0.05
d_4	The disturbance on subsystem 4	Gaussian noise with magnitude of 0.05

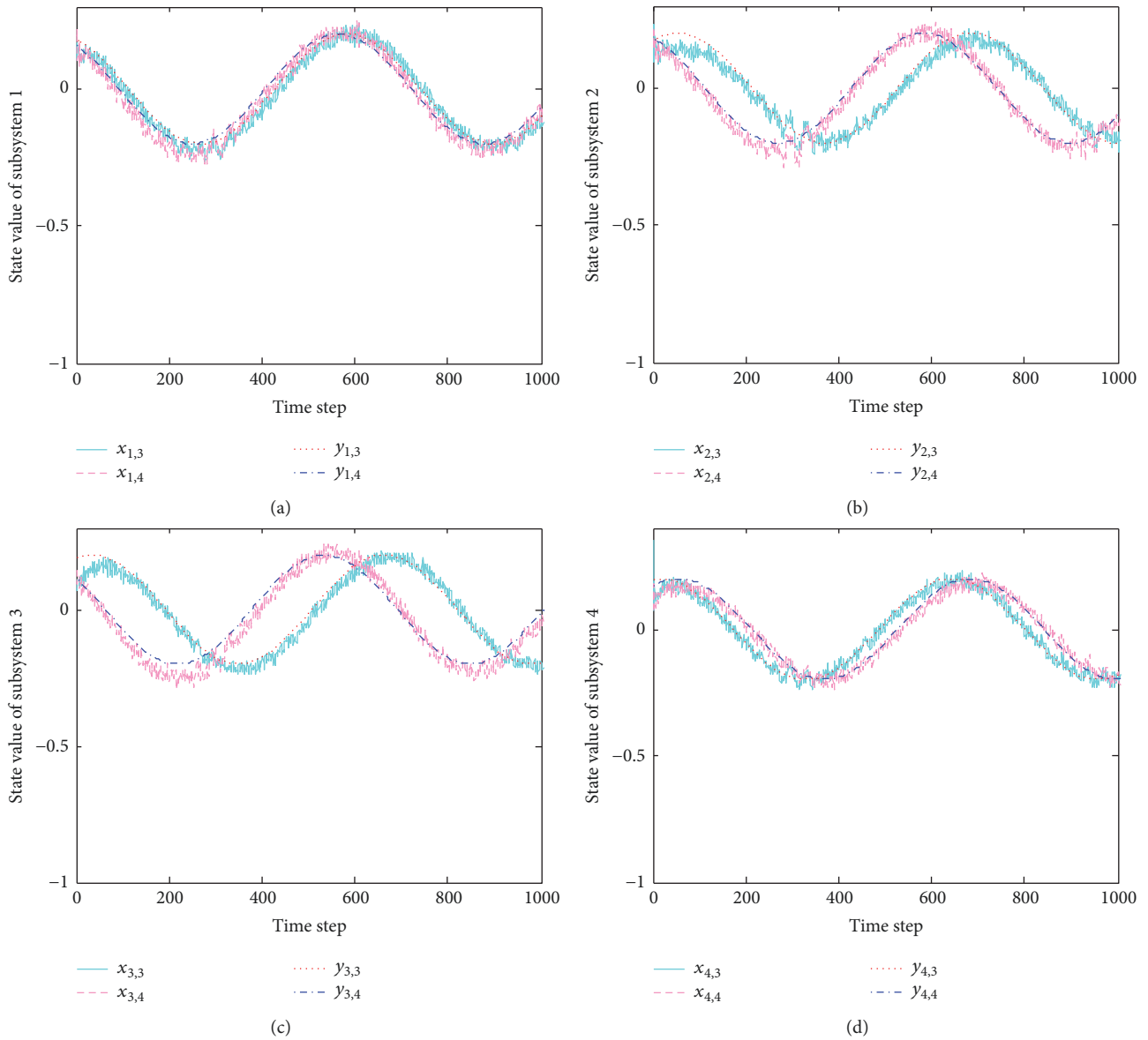

Figure 9: The state curves of system II with the proposed control scheme.

Table 4: Controller parameters of system II.

Parameter name	Description	Value
q	Dimension of basis vector function $\varphi_{c,i}$ for critic network. $i = 1, 2, 3, 4$.	100
l	Dimension of basis vector function $\varphi_{a,i}$ for action network. $i = 1, 2, 3, 4$.	100
$\sigma_{c,i}$	The width of radial basis function for critic network. $i = 1, 2, 3, 4$.	1.414
$\sigma_{a,i}$	The width of radial basis function for action network. $i = 1, 2, 3, 4$.	1.414
$c_{c,i,h}$, $c_{a,i,h'}$	The center vector for basis function. $i = 1, 2, 3, 4$. $h = 1, \ldots, q$. $h' = 1, \ldots, l$.	Element distributed in $[-1, +1]$
N	The horizon length of strategy utility function.	100
δ_i	The update rate for critic network weight matrix. $i = 1, 2, 3, 5$.	0.001
β_i	The update rate for action network weight matrix. $i = 1, 2, 3, 5$.	0.03
Γ_i	The damping rate of tracking error. $i = 1, 2, 3, 4$.	0.01

conventional reinforcement learning with only objective of long-term cost (critic network). In the future research, we will investigate the application of our proposed control scheme in a large cyberphysical system such as smart grid.

Conflicts of Interest

The authors declare that they have no conflicts of interest with regard to the publication of this manuscript.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61703347. This research is also supported by Fundamental Research Funds for the Central Universities Grant XDJK2017C071 and Chongqing Natural Science Foundation Grant cstc2016jcyjA0428.

References

[1] J. Sun, H. Zheng, Y. Chai, Y. Hu, K. Zhang, and Z. Zhu, "A direct method for power system corrective control to relieve current violation in transient with UPFCs by barrier functions," *International Journal of Electrical Power & Energy Systems*, vol. 78, pp. 626–636, 2016.

[2] J. Sun, Y. Chai, Y. Hu, H. Zheng, R. Ling, and K. Zhang, "UPFCs control design for avoiding generator trip of electric power grid with barrier function," *International Journal of Electrical Power & Energy Systems*, vol. 68, pp. 150–158, 2015.

[3] J. Sun, Y. Hu, Y. Chai et al., "L-infinity event-triggered networked control under time-varying communication delay with communication cost reduction," *Journal of The Franklin Institute*, vol. 352, no. 11, pp. 4776–4800, 2015.

[4] H. Li, G. Chen, T. Huang, and Z. Dong, "High-performance consensus control in networked systems with limited bandwidth communication and time-varying directed topologies," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 5, pp. 1043–1054, 2017.

[5] Z. Peng, D. Wang, H. Zhang, G. Sun, and H. Wang, "Distributed model reference adaptive control for cooperative tracking of uncertain dynamical multi-agent systems," *IET Control Theory & Applications*, vol. 7, no. 8, pp. 1079–1087, 2013.

[6] H. Chu, J. Yuan, and W. Zhang, "Observer-based adaptive consensus tracking for linear multi-agent systems with input saturation," *IET Control Theory & Applications*, vol. 9, no. 14, pp. 2124–2131, 2015.

[7] H. Su, M. Z. Q. Chen, X. Wang, H. Wang, and N. V. Valeyev, "Adaptive cluster synchronisation of coupled harmonic oscillators with multiple leaders," *IET Control Theory & Applications*, vol. 7, no. 5, pp. 765–772, 2013.

[8] Y. Feng, Y. Lv, and Z. Duan, "Distributed adaptive consensus protocols for linearly coupled Lur'e systems over a directed topology," *IET Control Theory Applications*, vol. 11, no. 15, pp. 2465–2474, 2017.

[9] A. Bemporad, M. Heemels, and M. Johansson, *Networked Control Systems*, Lecture Notes in Control and Information Sciences, Springer, London, UK, 2010.

[10] J. Qin, Q. Ma, W. X. Zheng, H. Gao, and Y. Kang, "Robust H_∞ group consensus for interacting clusters of integrator agents," *Institute of Electrical and Electronics Engineers Transactions on Automatic Control*, vol. 62, no. 7, pp. 3559–3566, 2017.

[11] J. Sarangapani, *Neural network control of nonlinear discrete-time systems*, CRC press, 2006.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, MIT press Cambridge, UK, 1998.

[13] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 8, no. 5, pp. 997–1007, 1997.

[14] X. Xu, C. Lian, L. Zuo, and H. He, "Kernel-based approximate dynamic programming for real-time online learning control: an experimental study," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 1, pp. 146–156, 2014.

[15] J. Y. Lee, J. B. Park, and Y. H. Choi, "Integral reinforcement learning for continuous-time input-affine nonlinear systems with simultaneous invariant explorations," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 5, pp. 916–932, 2015.

[16] J. Sarangapani, *Neural Network Control of Nonlinear Discrete-Time Systems (Public Administration and Public Policy)*, CRC/Taylor & Francis, 2006.

[17] C. Zhang, S. Abdallah, and V. Lesser, "Efficient multi-agent reinforcement learning through automated supervision," 1365-1370.

[18] S. Kumar Jilledi, "Comparison of multi-line power flow control using unified power flow controller (UPFC) and interline power flow controller (IPFC) in power transmission systems," *International Journal of Engineering Science & Technology*, vol. 3, no. 4, pp. 3229–3235, 2011.

[19] B. Xu, C. Yang, and Z. Shi, "Reinforcement learning output feedback NN control using deterministic learning technique,"

IEEE Transactions on Neural Networks and Learning Systems, vol. 25, no. 3, pp. 635–641, 2014.

[20] L. Liu, Z. Wang, and H. Zhang, "Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 299–313, 2017.

[21] R. Cui, C. Yang, Y. Li, and S. Sharma, "Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, 2017.

[22] C. W. Anderson, P. M. Young, M. R. Buehner, J. N. Knight, K. A. Bush, and D. C. Hittle, "Robust reinforcement learning control using integral quadratic constraints for recurrent neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 18, no. 4, pp. 993–1002, 2007.

[23] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: training deep neural networks with binary weights during propagations," in *Proceedings of the 29th Annual Conference on Neural Information Processing Systems, NIPS*, pp. 3123–3131, 2015.

[24] Z. Wang, F. Liu, S. H. Low, C. Zhao, and S. Mei, "Distributed frequency control with operational constraints, part II: network power balance," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1-1, 2017.