# Estimator-based adaptive neural network control of leader-follower high-order nonlinear multiagent systems with actuator faults

Zhou, Ning; Chen, Riqing; Xia, Yuanqing; Huang, Jie

*Published in:*
Concurrency and Computation: Practice and Experience

*DOI:*
10.1002/cpe.4225

Received: 3 October 2016 | Revised: 26 May 2017 | Accepted: 30 May 2017
DOI: 10.1002/cpe.4225

WILEY

SPECIAL ISSUE PAPER

# Estimator-based adaptive neural network control of leader-follower high-order nonlinear multiagent systems with actuator faults

Ning Zhou[1] | Riqing Chen[1] | Yuanqing Xia[2] | Jie Huang[3]

[1]College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China
[2]School of Automation, Key Laboratory of Intelligent Control and Decision of Complex Systems, Beijing Institute of Technology, Beijing 100081, China
[3]Faculty of Science and Engineering, University of Groningen, Groningen 9747 AG, The Netherlands

**Correspondence**
Riqing Chen, Institute of the Cloud Computing and Big Data for Smart Agriculture and Forestry, College of Computer and Information Sciences, Fujian Agriculture and Forestry University No. 15 Shang Xia Dian Road, Cangshan District, Fuzhou 350002, Fujian Province, China.
Email: riqing.chen@fafu.edu.cn

**Funding information**
Foundation of Cloud Computing and Big Data for Agriculture and Forestry, Grant/Award Number: 117-612014063; National Basic Research Program of China, Grant/Award Number: 2012CB720000; National Natural Science Foundation of China, Grant/Award Number: 61225015, 61105092, 61603094, 61603095; Foundation for Innovative Research Groups of the National Natural Science Foundation of China, Grant/Award Number: 61321002; Natural Science Foundation of Fujian Province of China, Grant/Award Number: 2016J01284, 2016J01286; Research Foundation for Outstanding Young Scholars in Fujian Agriculture and Forestry University, Grant/Award Number: xjq201612

**Summary**

The problem of distributed cooperative control for networked multiagent systems is investigated in this paper. Each agent is modeled as an uncertain nonlinear high-order system incorporating with model uncertainty, unknown external disturbance, and actuator fault. The communication network between followers can be an undirected or a directed graph, and only some of the follower agents can obtain the commands from the leader. To develop the distributed cooperative control algorithm, a prefilter is designed, which can derive the state-space representation to a newly constructed plant. Then, a set of distributed adaptive neural network controllers are designed by making certain modifications on traditional backstepping techniques with the aid of adaptive control, neural network control, and a second-order sliding mode estimator. Rigorous proving procedures are provided, which show that uniform ultimate boundedness of all the tracking errors can be achieved in a networked multiagent system. Finally, a numerical simulation is carried out to evaluate the theoretical results.

**KEYWORDS**

adaptive control, backstepping techniques, distributed cooperative control, multiagent systems, neural network control

## 1 | INTRODUCTION

In the past decade, cooperative control of multiagent systems has been receiving much attention because of the reason that multiple simpler agents show more beneficial comparing with single complicated agent. Research on higher-order cooperative control is motivated in part by the study of the flock behavior of birds. In some practical engineering applications, higher-order dynamics is used to model the specific systems, for example, the jerk system, which is a typical third-order dynamics.[1,2] Therefore, the extension from lower-order dynamics to higher-order ones becomes a necessary and significant field to study. But results about high-order nonlinear multiagent systems are still scarce.

Without considering the influence of actuator faults, some advanced control algorithms have been developed for high-order multiagent systems. By utilizing an encoding-decoding scheme and perturbation analysis of matrices, Qiu et al[3] investigated a leaderless and leader-follower quantized consensus for a kind of high-order systems with limited communication data rate. Sun et al[4] studied decentralised region tracking control for a group of high-order nonlinear agents and developed a set of decentralised adaptive neural network (NN) controllers by employing artificial potential functions, NN approximation, and adaptive backstepping techniques. Without employing any function approximators, a predefined performance design approach is proposed in Yoo[5] for distributed containment control of heterogeneous nonlinear strict-feedback systems; the developed algorithm guaranteed that the containment control errors can be preserved within certain given predefined bounds. The synchronization problem of identical linear high-order multiagent systems was studied in Xiang et al,[6] and a dynamical controller is constructed only depending on the weighted sum of relative output errors and the local measured output. Under the backstepping framework, two novel distributed adaptive fuzzy controllers are investigated for the cooperative control problem of two classes of high-order nonlinear multiagent systems in other works[7,8]; the proposed methods can overcome the effect of the unknown nonlinear dynamics and unknown disturbances, respectively. Hua et al[9] studied consensus control of high-order stochastic nonlinear agents with unknown nonlinear dead-zone under directed graph. The authors developed a distributed output tracking consensus controller based on backstepping method and dynamic surface control technique. Qi et al[10] considered leader-follower consensus of Lipschitz nonlinear dynamics. The finite-time coordinated tracking problem for a class of high-order uncertain nonlinear multiagent systems was studied in Fu and Wang[11] by using sliding mode control techniques. Regarding the directed and fixed graph condition, a distributed adaptive controller is designed in Shi and Shen[12] by employing Nussbaum-type gain technique and function approximation capability of neural networks. In Shen et al,[13] a distributed adaptive fuzzy control is investigated by using the approximate ability of the fuzzy logic systems to guarantee that each follower can asymptotically synchronize to the leader. But all the solutions mentioned above lack the capacity of fault tolerance due to the neglect of the impact of the actuator faults.

Fault-tolerant performance is a significant requirement for the coordinated control of practical high-order nonlinear systems. In the last few years, only limited solutions have been concerning high-order systems fault-tolerant control (FTC). For example, Shen et al[14] presented a novel FTC scheme that can guarantee all followers to synchronize a leader asymptotically and the tracking errors to converge to a small adjustable neighborhood of the origin. In the presence of actuator faults and network disconnections, Ma and Yang[15] discussed the active FTC problem for nonidentical high-order multiagent systems and presented a high-gain observer-like protocol and a cooperative FTC controller.

In this paper, the control objective is to design a distributed cooperative tracking control scheme such that each follower agent can track a time-varying leader under bidirectional or directional communication network environments. The main contributions are summarized as follows:

1. A novel estimator-based recursive approach is proposed for the distributed cooperative control of multiagent systems, and the key is redesigning errors variable $z_{nj}$ in the last step of the backstepping design.
2. A new distributed adaptive NN FTC algorithm is developed for the cooperative tracking of leader-follower high-order nonlinear systems, and rigorous proofs have been achieved step by step.
3. By adopting NN technology associated with adaptive control method, the proposed algorithm can provide robustness to eliminate the negative effectiveness of model uncertainty, unknown external disturbance, and actuator fault.

The rest of this paper is organized as follows. The graph theory, dynamics of the agent, and relative assumptions are given in Section 2. In Section3, the distributed adaptive NN cooperative controller control algorithms are developed. A simulation example is given in Section 4 to demonstrate the effectiveness of the proposed algorithm. Finally, the conclusions are given in Section 5.

## 2 | PRELIMINARIES AND PROBLEM FORMULATION

### 2.1 | Brief graph theory for multiagent system

In cooperative systems, many control laws are usually assumed to be distributed in the sense that it respects a communication network topology. The communication restrictions by topologies can severely limit the power of local distributed control algorithm at each individual agent. A communication network models the information flows over a multiagent network. A team of $m$ high-order nonlinear systems labeled as agent 1 to $m$ are considered here. The communication topology among the $m$ agents is assumed to be bidirectional or directional, and the interactions among the agents can be represented by an undirected or a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, A)$, where $\mathcal{V}$ is a set of the indices of the agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges that describe the communications between the agents. If $(i, j) \in \mathcal{E}$, then $i$ is neighboring to $j$, which means that agent $j$ can obtain information from agent $i$. $A = [a_{ij}] \in \mathbb{R}^{n\times n}$ is a weighted adjacency matrix with nonnegative adjacency elements $a_{ij}$. Moreover, it is assumed that $a_{ii} = 0$. If the state of agent $i$ is available to agent $j$, then agent $i$ is said to be a neighbor of agent $j$. The neighbor set of node $v_j$ is denoted by $\mathcal{N}_j$, where $j \notin \mathcal{N}_j$. The in-degree matrix of the weighted graph $\mathcal{G}$ is denoted by $D = \mathrm{diag}\{d_1, \ldots, d_n\}$ with $d_i = \sum_{j=1}^{n} a_{ij}$. The Laplacian matrix $L$ of $\mathcal{G}$ is defined as $L = D - A$. Note that $A$, $D$, and $L$ are all constant and bounded matrices. Define graph $\bar{\mathcal{G}}$ with $\mathcal{V} = \{0, i\}$. Denote $B = \mathrm{diag}\{b_1, \ldots, b_n\}$ be the leader adjacency matrix and $b_i > 0$ if agent $i$ has access to agent 0; otherwise, $b_i = 0$.

### 2.2 | Problem formulation

In this paper, we consider a network with $m + 1$ agents, in the sense that their states keep updated relaying on local information exchange. The flow of the information exchange among agents is described by a fixed and directed graph $\mathcal{G}$. The dynamics of the agents are described in the nonlinear

Brunovsky form[14,16]

$$\dot{x}_{ij} = x_{(i+1)j}, \tag{1}$$

$$\dot{x}_{nj} = u_j^f + f_j(x_j) + \zeta_j, \quad i = 1, \ldots, n-1, \quad j = 1, \ldots, m, \tag{2}$$

where $x_{ij} \in \mathbb{R}^q$ is the $i$-th state of the $j$-th agent; $x_j = [x_{1j}, \ldots, x_{nj}]^T \in \mathbb{R}^{n\times q}$ is the state matrix of the $j$-th agent with $q \geq 1$ being the dimension of the space; $f_j(x_j) : \mathbb{R}^{qn} \to \mathbb{R}^q$ is an unknown function that is also locally Lipschitz with $f_j(0) = 0$; $u_j^f$ is the control input; and $\zeta_j \in \mathbb{R}^q$ denotes an unknown external disturbance; it is assumed to be bounded. Specifically, when $n = 3$, $x_{1j}$, $x_{2j}$, and $x_{3j}$ are the position vector, velocity vector, and acceleration vector of the $j$-th agent, respectively.

In practical applications, actuator may become faulty. In this paper, the following fault is considered:

$$u_j^f = u_j + f_{ju}(t), \tag{3}$$

where $u_j \in \mathbb{R}^q$ is the control protocol of the $j$-th agent and $f_{ju}$ denotes an unknown and bounded time-varying signal.

The dynamics of the time-varying leader agent, labeled 0, is described by

$$\dot{x}_{i0} = x_{(i+1)0}, \tag{4}$$

$$\dot{x}_{n0} = f_0(t, \underline{x}_0), \tag{5}$$

for $i = 1, \ldots, n-1$, where $x_{i0} \in \mathbb{R}^q$ is the $i$-th state of the leader agent; $\underline{x}_0 = [x_{10}, x_{20}, \ldots, x_{n0}]^T \in \mathbb{R}^{qn}$ is the state vector with bounded $x_{10}, x_{20}, \ldots, x_{n0}$; $f_0(t, \underline{x}_0) : [0, \infty) \times \mathbb{R}^{qn} \to \mathbb{R}^q$ is piecewise continuous at $t$ and locally Lipschitz in $\underline{x}_0$ with $f_0(t, \mathbf{0}) = \mathbf{0}$ for all $t \geq 0$ and $\underline{x}_0 \in \mathbb{R}^{qn}$, and it is unknown to all the other agents. Assume that the leader is forward complete, ie, $\forall \underline{x}_0(0)$, the solution $\underline{x}_0(t)$ exists $\forall t \geq 0$. In other words, there is no finite escape time. The dynamics (4) to (5) can be identified as an exosystem that generates a desired command trajectory.

*Remark* 1. For the sake of simplicity in description, let $q = 1$ if not otherwise specified in the following analysis. However, it is worth noting that all the results hereafter can be directly extended to the higher dimensional case by using the Kronecker product.

In this paper, it is assumed that only the relative state information can be used for the controller design. More precisely, for the $j$-th agent, the only obtainable information is the neighborhood synchronization error. Then, the following assumptions are considered to facilitate the controller design:

**Assumption 1.** (Zhang and Frank[16]) Topology $\bar{\mathcal{G}}$ contains a spanning tree with the root node being agent 0.

*Remark* 2. It should be noted that if the preceding assumption does not hold and there is a set of $k(1 \leq k < N)$ agents $S_k = n_1, \ldots, n_k$ in $\mathcal{G}$ with $s_k \in \mathcal{N}$, which can not receive the commands from the leader agent, then it means that $S_k$ do not have access to information of agents $\bar{S}_k = \mathcal{N} \backslash S_k$. Obviously, there must exist some agent in $S_k$ in the following 2 cases: isolated or act as leaders, which do not receive information. It is obvious that both of the 2 cases cannot contribute to final synchronization to the leader agent.

**Assumption 2.** (Zhang and Frank[16]) The disturbance $\zeta_j$ is bounded such that $|\zeta_j| \leq \zeta_{Mj}$, where $\zeta_{Mj}$ is an unknown positive constant.

**Assumption 3.** (Shen et al[14]) The increased bias torque $f_{ju}$ is bounded, that is, $|f_{ju}| \leq \bar{f}_{ju}$, where $\bar{f}_{ju}$ is an unknown positive scalar.

**Assumption 4.** (Cao et al[17]) The time derivative of $f_0(t, \underline{x}_0)$ is bounded, ie, $\sup_t |\dot{f}_0(t, \underline{x}_0)| \leq \bar{f}_0$, where $\bar{f}_0$ is a positive constant.

## 3 | DISTRIBUTED ADAPTIVE NEURAL NETWORK COOPERATIVE CONTROL

In this section, we will show how to design the distributed adaptive NN fault-tolerant controllers for the follower agents such that the cooperative tracking problem can be solved.

### 3.1 | Definitions and Lemmas

Some necessary definitions and lemmas regarding systems (1) to (2) are needed in the following derivation.

**Lemma 1.** *(Huang et al[18]) Suppose n and m are 2 positive real numbers, and $a \geq 0$, $b \geq 0$, then, for any constant $c > 0$, $a^n b^m \leq c a^{n+m} + \frac{m}{n+m}\left[\frac{n}{c(n+m)}\right]^{\frac{n}{m}} b^{n+m}$.*

**Definition 1.** (Zou and Kumar[19]) It is assumed that only a subset of agents can receive commands from leader, and the communication network among the neighboring agents is undirected. To facilitate the further design, a reference tracking signal $\bar{x}_{ij}$ needs to be constructed for each individual follower agent. By using the nearest neighbor rule and the weighted average of its neighboring agents' states, we can compute $\bar{x}_{ij}$ for the $j$-th agent as follows:

$$\bar{x}_{ij} = \frac{\sum_{l \in \mathcal{N}_j} a_{jl} x_{il} + b_j x_{i0}}{\sum_{l \in \mathcal{N}_j} a_{jl} + b_j}, i = 1, \dots, n-1, \quad j = 1, \dots, m, \tag{6}$$

where $\mathcal{N}_j$ denotes the neighbor set of agent $j$ and $j \notin \mathcal{N}_j$, $l \in \mathcal{N}_j$ means that the $l$-th agent is a neighbor of the $j$-th agent. Since the matrix $L + B$ is symmetric and positive define, it implies that $\sum_{l \in \mathcal{N}_j} a_{jl} + b_j \neq 0$. Thus, the reference signal $\bar{x}_{ij}$ in Equation 6 is well defined.

**Definition 2.** (Lewis et al[26]) The solutions of the following dynamic system

$$\dot{x}(t) = f(x, t), \quad y = h(x, t), \quad x(t) \in \mathbb{R}^n, \quad t \geq t_0 \tag{7}$$

are said to be uniformly ultimately bounded (UUB) if there exist positive constants $\alpha$ and $\beta$, and $\forall \delta \in (0, \beta)$, there is a positive constant $T = T(\delta)$, such that $\|x(t_0)\| < \delta \Rightarrow \|x(t)\| < \alpha, \forall t \geq t_0 + T$.

Based on the preceding defined signal $\bar{x}_{ij}$, the following lemma is needed for the further design.

**Lemma 2.** *(Zou and Kumar[19]) Let $y_{ij} = x_{ij} - \bar{x}_{ij} (i = 1, \dots, n-1; \quad j = 1, \dots, m)$ be the state error between $x_{ij}$ and the reference signal $\bar{x}_{ij}$. If $y_{i*} = [y_{i1}, \dots, y_{im}]^T = 0$, then $x_{ij} = x_{i0}$.*

**Lemma 3.** *(Zou and Kumar[20]) Define a continuous function $V(t) \geq 0, \forall t \in \mathbb{R}^+$, if $\dot{V}(t) \leq -\gamma V + \varepsilon$, where $\gamma > 0$ and $\varepsilon$ are constants, then*

$$V(t) \leq \kappa + (V(0) - \kappa)e^{-\gamma t} \tag{8}$$

*with $\kappa = \varepsilon / \gamma$.*

**Lemma 4.** *(Khalil[21]) Boundedness theorem: If $I = [a, b]$ is a closed and bounded interval, and $f : I \to \mathbb{R}$ is a continuous function on $I$, then $f$ is bounded on $I$.*

**Definition 3.** Define the errors variables $z_{*j} = [z_{1j}, z_{2j}, \dots, z_{nj}]^T$ with the aid of backstepping techniques:

$$z_{1j} = x_{1j} - \bar{x}_{1j}, \tag{9}$$

$$z_{ij} = x_{ij} - \bar{x}_{ij} - \alpha_{ij}, \quad \text{for} \quad 2 \leq i \leq n-1, \tag{10}$$

$$z_{nj} = x_{nj} - \hat{x}_{nj} - \alpha_{nj}, \tag{11}$$

where $\hat{x}_{nj}$ is an estimator designed in Equations 32 to 33. It is obvious that if $z_{i*} = x_{i*} - \bar{x}_{i*} = D^{-1}(L + B)(x_{i*} - I_n \underline{x}_0)$, then $z_{i*} = 0$ means that $x_{i*} - I_n \underline{x}_0 = 0$.

## 3.2 | The NN-based approximation theory

In the following design process, NN will be utilized to eliminate the adverse effect from the unknown nonlinear function $f_j(x_j)$ in Equation 2. Based on the approximation property of NN,[22] a nonlinear continuous function can be approximated to any desired accuracy over a compact set $\Omega_z$. Taking $f(z)$: $\mathbb{R}^l \to \mathbb{R}^m$ as an example, an ideal weights matrix $W^*$ should exist such that the ideal NN can approximate $f(z)$ as accurately as possible on $\Omega_z \subset \mathbb{R}^l$:

$$f(z) = W^{*T} \phi(z) + \epsilon_z, \tag{12}$$

where $W^* \in \mathbb{R}^{p \times m}$ is the optimal weight matrix of NN, $p$ is the number of neuron, and $\phi(z) = [\phi_1(z), \dots, \phi_p(z)]^T$ represents a basis function vector that can be chosen as sigmoid, Gaussian, etc. In general, $W^*$ is only used for analytical purposes, which needs to be estimated. $\epsilon_z \in \mathbb{R}^m$ denotes the minimum possible deviation between $W^{*T} \phi(z)$ and $f(z)$, which is bounded with $\|\epsilon_z\| \leq \epsilon_N$, where $\epsilon_N$ is an unknown positive constant. For all $z \in \Omega_z \subset \mathbb{R}^l$, $W^*$ is defined as

$$W^* \triangleq \arg \min_{W \in \mathbb{R}^{p \times m}} \left\{ \sup_{z \in \Omega_z} \left\| f(z) - W^T \phi(z) \right\| \right\}. \tag{13}$$

By employing NN, $f(z) \in \mathbb{R}^m$ can be approximated on $\Omega_z$ as follows:

$$f_{NN}(W, z) = \hat{W}^T \phi(z), \tag{14}$$

where $\hat{W} \in \mathbb{R}^{p \times m}$ denotes an adjustable weight matrix.

*Remark* 3. It is worthy to note that the NN approximation error $\epsilon_z$ would be decreased sufficiently small as $p$ increases. Furthermore, $\|\epsilon_z\|$ can be reduced to arbitrary small if $p$ is large enough.[23,24]

### 3.3 | Recursive controller design procedure

By employing the recursive method in a distributed manner, a sequence of virtual controllers $\alpha_{ij}$ will be designed in this subsection only relaying on the neighboring information. Moreover, the actual controllers $u_j$ will be constructed by utilizing adaptive control, neural network control, backstepping method associated with a predesigned second-order sliding mode estimator. Detailed recursive procedures are given as follows.

**Step 1**: Let $\alpha_{2j}$ be the first virtual controller of the $j$-th agent. Using Equation 1 for Equation 9, it has

$$\dot{z}_{1j} = \dot{x}_{1j} - \dot{\bar{x}}_{1j} = x_{2j} - \bar{x}_{2j} = z_{2j} + \alpha_{2j}. \tag{15}$$

Consider $z_{1j} = x_{1j} - \bar{x}_{1j}$ of the first-order subsystem in Equation 1, and select the following Lyapunov function $V_1$:

$$V_1 = \frac{1}{2} z_{1*}^T z_{1*}, \tag{16}$$

where $z_{1*} = [z_{11}, z_{12}, \dots, z_{1m}]^T$.

Using Equations 15 to 16, one can derive the time derivative of $V_1$ as

$$\dot{V}_1 = z_{1*}^T \dot{z}_{1*} = \sum_{j=1}^{m} z_{1j}(z_{2j} + \alpha_{2j}). \tag{17}$$

In order to ensure that the time derivative of the Lyapunov function $V_1$ is negative definite, an appropriate distributed virtual controller $\alpha_{2j}$ should be designed, which is given by

$$\alpha_{2j} = -c_{1j} z_{1j}, \tag{18}$$

where $c_{1j}$ is the design parameter, satisfying $c_{1j} > 0$. By using Equations 10, 15, and 18, $\dot{z}_{1j}$ becomes

$$\dot{z}_{1j} = -c_{1j} z_{1j} + z_{2j}. \tag{19}$$

From Equation 9, it has $z_{1*} = D^{-1}(L + B)(x_{1*} - I_n \underline{x}_0)$, then we get $Dz_{1*} = (L + B)(x_{1*} - I_n x_0)$. When $z_{1*} \to 0$, it means that $x_{1*} - I_n \underline{x}_0 \to 0$.

Therefore, by substituting Equation 18 into Equation 17, it yields

$$\begin{aligned} \dot{V}_1 &= -z_{1*}^T \operatorname{diag}(c_{1*}) z_{1*} + \sum_{j=1}^{m} z_{1j} z_{2j}, \\ &= -c_1 z_{1*}^T z_{1*} + \sum_{j=1}^{m} z_{1j} z_{2j}, \end{aligned} \tag{20}$$

where $c_1 = \min_j \{c_{1j}\}$.

**Step 2**: Using Equations 1 and 10, the following equation can be obtained.

$$\begin{aligned} \dot{z}_{2j} &= x_{3j} - \bar{x}_{3j} - \dot{\alpha}_{2j}, \\ &= z_{3j} + \alpha_{3j} - \frac{\partial \alpha_{2j}}{\partial x_{1j}} x_{2j} - \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{2j}}{\partial x_{1l}} x_{2l} - \frac{\partial \alpha_{2j}}{\partial x_{10}} x_{20}, \end{aligned} \tag{21}$$

where $\alpha_{3j}$ is a virtual controller. Consider $z_{2j} = x_{2j} - \bar{x}_{2j}$ of the second-order subsystem in Equation 1, and redesign the second Lyapunov function $V_2$ as

$$V_2 = V_1 + \frac{1}{2} z_{2*}^T z_{2*}, \tag{22}$$

where $z_{2*} = [z_{21}, z_{22}, \dots, z_{2m}]^T$. Utilizing Equations 20 and 21, we can obtain the time derivative of $V_2$ as follows:

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + \sum_{j=1}^{m} z_{2j} \dot{z}_{2j}, \\ &= -c_1 z_{1*}^T z_{1*} + \sum_{j=1}^{m} z_{1j} z_{2j} + \sum_{j=1}^{m} z_{2j} \\ &\quad \left( z_{3j} + \alpha_{3j} - \frac{\partial \alpha_{2j}}{\partial x_{1j}} x_{2j} - \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{2j}}{\partial x_{1l}} x_{2l} - \frac{\partial \alpha_{2j}}{\partial x_{10}} x_{20} \right), \end{aligned} \tag{23}$$

To guarantee $\dot{V}_2 < 0$, the appropriate distributed virtual control $\alpha_{3j}$ is designed as

$$\alpha_{3j} = -z_{1j} - c_{2j} z_{2j} + \frac{\partial \alpha_{2j}}{\partial x_{1j}} x_{2j} + \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{2j}}{\partial x_{1l}} x_{2l} + \frac{\partial \alpha_{2j}}{\partial x_{10}} x_{20}, \tag{24}$$

where $c_{2j} > 0$ is a design parameter. By using Equations 10, 21, and 24, $\dot{z}_{2j}$ becomes

$$\dot{z}_{2j} = -z_{1j} - c_{2j} z_{2j} + z_{3j}. \tag{25}$$

Using Equation 24 for Equation 23, we have

$$\dot{V}_2 = -\sum_{k=1}^{i} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{ij} z_{(i+1)j}, \tag{26}$$

where $c_2 = \min_j\{c_{2j}\}$.

**Step i, $i = 3, \dots, n-1$**: Follow the preceding design procedure, it has

$$\begin{aligned} z_{ij} &= x_{(i+1)j} - \bar{x}_{(x+1)j} - \dot{\alpha}_{ij}, \\ &= z_{(i+1)j} + \alpha_{(i+1)j} - \sum_{k=1}^{i-1} \frac{\partial \alpha_{ij}}{\partial x_{kj}} x_{(k+1)j} - \sum_{k=1}^{i-1} \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{ij}}{\partial x_{kl}} x_{(k+1)l} - \sum_{k=1}^{i-1} \frac{\partial \alpha_{ij}}{\partial x_{k0}} x_{(k+1)0}. \end{aligned} \tag{27}$$

Choose the $i$-th Lyapunov function candidate $V_i$ as

$$V_i = V_{i-1} + \frac{1}{2} z_{i*}^T z_{i*}, \tag{28}$$

where $z_{i*} = [z_{i1}, z_{i2}, \dots, z_{im}]^T$. Taking the time derivative of $V_i$, it has

$$\begin{aligned} \dot{V}_i &= \dot{V}_{i-1} + \sum_{j=1}^{m} z_{ij} \dot{z}_{ij}, \\ &= -\sum_{k=1}^{i-1} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{i-1j} z_{ij} + \sum_{j=1}^{m} z_{ij} \left[ z_{(i+1)j} + \alpha_{(i+1)j} - \sum_{k=1}^{i-1} \frac{\partial \alpha_{ij}}{\partial x_{kj}} x_{(k+1)j} - \sum_{k=1}^{i-1} \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{ij}}{\partial x_{kl}} x_{(k+1)l} - \sum_{k=1}^{i-1} \frac{\partial \alpha_{ij}}{\partial x_{k0}} x_{(k+1)0} \right], \end{aligned} \tag{29}$$

where $c_k = \min_j\{c_{kj}\}, k = 1, \dots, n-1$. Designing the intermediate controller $\alpha_{(i+1)j}$ as

$$\alpha_{(i+1)j} = -z_{(i-1)j} - c_{ij} z_{ij} + \sum_{k=1}^{i-1} \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{ij}}{\partial x_{kl}} x_{(k+1)l} + \sum_{k=1}^{i-1} \frac{\partial \alpha_{ij}}{\partial x_{kj}} x_{(k+1)j} + \sum_{k=1}^{i-1} \frac{\partial \alpha_{ij}}{\partial x_{k0}} x_{(k+1)0}, \tag{30}$$

where $c_{ij} >$ is a design parameter. By substituting Equation 30 into Equation 29, $\dot{V}_i$ can be rewritten as

$$\dot{V}_i = -\sum_{k=1}^{i} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{ij} z_{(i+1)j}. \tag{31}$$

**Step n**: Denote $\hat{x}_{nj}$ and $\hat{f}_{0j}$ be the estimate of $x_{n0}$ and $f_0$ for the $j$-th agent, respectively. Inspired by Cao et al,[17] a second-order sliding mode estimator (Equations 32-33) is proposed for each agent to guarantee that $\hat{x}_{nj} \to x_{n0}$ in finite time $\bar{T}_s$, for $j = 1, \dots, m$. $\bar{T}_s$ can be calculated according to Theorem 4.1 in Cao et al.[17] When $t \geq \bar{T}_s$, it has $\hat{x}_{nj} \equiv x_{n0}$, then we can get $z_{nj} \equiv x_{nj} - x_{n0} - \alpha_{nj}$.

$$\dot{\hat{x}}_{nj} = \hat{f}_{0j} - \gamma_1 \text{sign}\left[ \sum_{l \in \mathcal{N}_j} a_{jl}(\hat{x}_{nj} - \hat{x}_{nl}) + b_j(\hat{x}_{nj} - x_{n0}) \right], \tag{32}$$

$$\dot{\hat{f}}_{0j} = -\gamma_2 \text{sign}\left[ \sum_{l \in \mathcal{N}_j} a_{jl}(\hat{f}_{0j} - \hat{f}_{0l}) + b_j \left( \hat{f}_{0j} - f_0(t, \underline{x}_0) \right) \right], \tag{33}$$

where $\gamma_1 > 0$ and $\gamma_2 > \bar{f}_0$ are constants.

Differentiating $z_{nj} = x_{nj} - \hat{x}_{nj} - \alpha_{nj}$, it can be obtained that

$$\begin{aligned} \dot{z}_{nj} &= \dot{x}_{nj} - \dot{\hat{x}}_{n0} - \dot{\alpha}_{nj}, \\ &= u_j + f_{ju}(t) + f_j(x_j) + \zeta_j(t) - \dot{\alpha}_{nj} - \hat{f}_{0j} + \gamma_1 \text{sign}\left[ \sum_{l \in \mathcal{N}_j} a_{jl}(\hat{x}_{nj} - \hat{x}_{nl}) + b_j(\hat{x}_{nj} - x_{n0}) \right], \\ &= u_j + f_{ju}(t) + f_j(x_j) + \zeta_j(t) - \sum_{k=1}^{n-1} \frac{\partial \alpha_{nj}}{\partial x_{kj}} x_{(x+1)j} - \sum_{k=1}^{n-1} \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{nj}}{\partial x_{kl}} x_{(k+1)l} - \sum_{k=1}^{n-1} \frac{\partial \alpha_{nj}}{\partial x_{k0}} x_{(x+1)0} - \hat{f}_{0j} + \gamma_1 \text{sign}\left[ \sum_{l \in \mathcal{N}_j} a_{jl}(\hat{x}_{nj} - \hat{x}_{nl}) + b_j(\hat{x}_{nj} - x_{n0}) \right]. \end{aligned} \tag{34}$$

In the following analysis, in order to derive the final algorithm for the control objective, a discussion on how to design a model-dependant control approach (ie, for the case that the unknown function $f_j(x_j)$ is available) will be given in **Step n(a)** in priority to deduce the design of the distributed adaptive neural network controller in **Step n(b)**.

**Step n(a)** Consider the case that the unknown function $f_j(x_j)$ in Equation 2 is available.

From Assumptions 2 and 3, it is easy to assume that there exist a positive constant $\xi_j$ such that $|\zeta_j + f_{ju}| \leq \xi_j$. Denote $\hat{\xi}_j$ be the estimate of $\xi_j$, and choose the $n$-th Lyapunov function candidate $V_n$ as

$$V_n = V_{n-1} + \frac{1}{2} z_{n*}^T z_{n*} + \frac{1}{2} \sum_{j=1}^{m} \eta_{2j}^{-1} \tilde{\xi}_j^2, \tag{35}$$

where $\eta_{2j} > 0$ is a design parameter.

Taking the time derivative of $V_n$ with respect to Equations 31 to 34, we obtain

$$\begin{aligned}\dot{V}_n &= -\sum_{k=1}^{n-1} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{(n-1)j} z_{nj} + \sum_{j=1}^{m} z_{nj}\dot{z}_{nj} - \sum_{j=1}^{m} \eta_{2j}^{-1}\tilde{\xi}_j\dot{\hat{\xi}}_j, \\ &= -\sum_{k=1}^{n-1} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{(n-1)j} z_{nj} + \sum_{j=1}^{m} z_{nj} \\ &\left[-\sum_{k=1}^{n-1}\frac{\partial\alpha_{nj}}{\partial x_{kj}} x_{(k+1)j} + u_j + f_{ju}(t) + f_j(x_j) + \zeta_j(t) - \dot{\hat{x}}_{n0} - \sum_{k=1}^{n-1}\sum_{l\in\mathcal{N}_j}\frac{\partial\alpha_{nj}}{\partial x_{kl}} x_{(k+1)l} - \sum_{k=1}^{n-1}\frac{\partial\alpha_{nj}}{\partial x_{k0}} x_{(k+1)0}\right] - \sum_{j=1}^{m}\eta_{2j}^{-1}\tilde{\xi}_j\dot{\hat{\xi}}_j.\end{aligned} \tag{36}$$

Construct the distributed adaptive controller $u_j$ and the following adaptation law $\hat{\xi}_j$ as follows:

$$u_j = -z_{(n-1)j} - c_{nj} z_{nj} + \sum_{k=1}^{n-1}\frac{\partial\alpha_{nj}}{\partial x_{kj}} x_{(k+1)j} + \sum_{k=1}^{n-1}\sum_{l\in\mathcal{N}_j}\frac{\partial\alpha_{nj}}{\partial x_{kl}} x_{(k+1)l} + \hat{f}_{0j} + \sum_{k=1}^{n-1}\frac{\partial\alpha_{nj}}{\partial x_{k0}} x_{(k+1)0} - f_j(x_j) - \hat{\xi}_j\mathrm{sign}(z_{nj}) - \gamma_1\mathrm{sign}\left[\sum_{l\in\mathcal{N}_j} a_{jl}(\hat{x}_{nj} - \hat{x}_{nl}) + b_j(\hat{x}_{nj} - x_{n0})\right], \tag{37}$$

$$\dot{\hat{\xi}}_j = \eta_{2j}\left(|z_{nj}| - \beta_{2j}\hat{\xi}_j\right), \tag{38}$$

where $\beta_{2j} > 0$ is a design parameter.

Substituting Equations 37 to 38 into Equation 36, we obtain

$$\dot{V}_n = -\sum_{k=1}^{n-1} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{(n-1)j} z_{nj} + \sum_{j=1}^{m} z_{nj}\left[-z_{(n-1)j} - c_{nj} z_{nj} + \zeta_j + f_{ju} - \hat{\xi}_j\mathrm{sign}(z_{nj})\right] - \sum_{j=1}^{m}\tilde{\xi}_j|z_{nj}| + \beta_{2j}\tilde{\xi}_j\hat{\xi}_j. \tag{39}$$

With the aid of Lemma 1, the following inequality is achieved for the further proof

$$\beta_{2j}\tilde{\xi}_j\hat{\xi}_j \le -\frac{1}{2}\beta_{2j}\tilde{\xi}_j^2 + \frac{1}{2}\beta_{2j}\xi_j^2. \tag{40}$$

Using Equation 40 for Equation 39, we can derive

$$\begin{aligned}\dot{V}_n &\le -\sum_{k=1}^{n} c_k z_{k*}^T z_{k*} - \sum_{j=1}^{m}\frac{1}{2}\beta_{2j}\tilde{\xi}_j^2 + \mu_1 \\ &\le -\mu_2 V_n + \mu_1,\end{aligned} \tag{41}$$

where $\mu_1 = \sum_{j=1}^{m}\mu_{1j}, \mu_{1j} = \frac{1}{2}\beta_{2j}\xi_j^2, \mu_2 = \min\{\min_k\{2c_k\}, \min_j\{\beta_{2j}\eta_{2j}\}\}$. Denote $\bar{\mu}_2 = \frac{\mu_1}{\mu_2}$, by employing Lemma 3, then Equation 41 satisfies

$$0 \le V_n(t) \le \bar{\mu}_2 + (V_n(0) - \bar{\mu}_2)e^{-\mu_2 t}. \tag{42}$$

From Equation 42 and Lemma 4 (ie, boundedness theorem in Khalil[21]), we can further achieve that $z_{j*}$ is bounded for $i = 1, \ldots, n$; moreover, $x_{i*} - I_n \underline{x}_0$ is bounded from Definition 3 and Theorem 3.2 in Zhou and Xia,[25] then it derives the boundedness of $x_{ij} - x_{i0}$.

**Step n(b)** Please note that the proposed controller (Equation 37) will not work when the function $f_j(x_j)$ in Equation 2 is completely unknown. Thus, we further consider the case that when $f_j(x_j)$ is unavailable.

By employing the NN, $f_j(x_j)$ can be approximated to an arbitrary accuracy in the following form.

$$f_j(x_j) = W_j^{*T}\phi_j(x_j) + \epsilon_j, \tag{43}$$

where $W_j^* \in \mathbb{R}^{p_j}$ is the optimal weight matrix of NN, $\epsilon_j \in \mathbb{R}$ is the approximation error, and $\phi_j(x_j) \in \mathbb{R}^{p_j}$ are the basis function vector.

Before moving on, the following assumption is needed.

**Assumption 5.** There exist unknown constants $\bar{\epsilon}_j$ such that $|\epsilon_j| \le \bar{\epsilon}_j$, for $j = 1, \ldots, n$.

Denote $\hat{W}_j$ be the estimation of the ideal NN weight $W_j^*$, $\hat{\xi}_j$ be the estimate of $\xi_j$ with $|\epsilon_j + \zeta_j + f_{ju}| \le \xi_j$, and reselect the $n$-th Lyapunov function candidate $V_n$ as

$$V_n = V_{n-1} + \frac{1}{2} z_{n*}^T z_{n*} + \frac{1}{2}\sum_{j=1}^{m}\tilde{W}_j^T\Gamma_j^{-1}\tilde{W}_j + \frac{1}{2}\sum_{j=1}^{m}\eta_{2j}^{-1}\tilde{\xi}_j^2, \tag{44}$$

where $\Gamma_j \in \mathbb{R}^{p_j \times p_j}$ is a positive definite matrix and $\eta_{2j} > 0$ is a design parameter.

From Equations 31 to 34, we can get the time derivative of $V_n$ as follows:

$$\begin{aligned}\dot{V}_n &= -\sum_{k=1}^{n-1} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{(n-1)j} z_{nj} + \sum_{j=1}^{m} z_{nj}\dot{z}_{nj} - \sum_{j=1}^{m}\tilde{W}_j^T\Gamma_j^{-1}\dot{\hat{W}}_j - \sum_{j=1}^{m}\eta_{2j}^{-1}\tilde{\xi}_j\dot{\hat{\xi}}_j, \\ &= -\sum_{k=1}^{n-1} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{(n-1)j} z_{nj} + \sum_{j=1}^{m} z_{nj}\left[-\sum_{k=1}^{n-1}\frac{\partial\alpha_{nj}}{\partial x_{kj}} x_{(k+1)j} - \sum_{k=1}^{n-1}\sum_{l\in\mathcal{N}_j}\frac{\partial\alpha_{nj}}{\partial x_{kl}} x_{(k+1)l} + u_j + f_{ju}(t) + f_j(x_j) + \zeta_j(t) - \dot{\hat{x}}_{n0} - \sum_{k=1}^{n-1}\frac{\partial\alpha_{nj}}{\partial x_{k0}} x_{(k+1)0}\right] \\ &\quad - \sum_{j=1}^{m}\tilde{W}_j\Gamma_j^{-1}\dot{\hat{W}}_j - \sum_{j=1}^{m}\eta_{2j}^{-1}\tilde{\xi}_j\dot{\hat{\xi}}_j.\end{aligned} \tag{45}$$

Design the distributed adaptive NN controller $u_j$ and the following adaptation laws $\hat{\xi}_j$ and $\hat{W}_j$ as follows.

$$
\begin{aligned}
u_j = & -z_{(n-1)j} - c_{nj} z_{nj} + \sum_{k=1}^{n-1} \frac{\partial \alpha_{nj}}{\partial x_{kj}} x_{(k+1)j} + \sum_{k=1}^{n-1} \sum_{l \in \mathcal{N}_j} \frac{\partial \alpha_{nj}}{\partial x_{kl}} x_{(k+1)l} + \hat{f}_{0j} \\
& + \sum_{k=1}^{n-1} \frac{\partial \alpha_{nj}}{\partial x_{k0}} x_{(k+1)0} - \hat{W}_j^T \phi_j(x_j) - \hat{\xi}_j \mathrm{sign}(z_{nj}) \\
& - \gamma_1 \mathrm{sign}\left[ \sum_{l \in \mathcal{N}_j} a_{jl}(\hat{x}_{nj} - \hat{x}_{nl}) + b_j(\hat{x}_{nj} - x_{n0}) \right],
\end{aligned} \tag{46}
$$

$$
\dot{\hat{\xi}}_j = \eta_{2j} \left( |z_{nj}| - \beta_{2j} \hat{\xi}_j \right), \tag{47}
$$

$$
\dot{\hat{W}}_j = \Gamma_j \left( z_{nj} \phi_j(x_j) - \beta_{3j} \hat{W}_j \right), \tag{48}
$$

where $\beta_{2j} > 0$ and $\beta_{3j} > 0$ are design parameters.

Substituting Equations 46 to 48 into Equation 45, it yields

$$
\begin{aligned}
\dot{V}_n = & -\sum_{k=1}^{n-1} c_k z_{k*}^T z_{k*} + \sum_{j=1}^{m} z_{(n-1)j} z_{nj} + \sum_{j=1}^{m} z_{nj} \left[ -z_{(n-1)j} - c_{nj} z_{nj} + \tilde{W}_j^T \phi_j(x_j) + \epsilon_j + \zeta_j + f_{ju} - \hat{\xi}_j \mathrm{sign}(z_{nj}) \right] \\
& - \sum_{j=1}^{m} z_{nj} \tilde{W}_j^T \phi_j(x_j) - \sum_{j=1}^{m} \tilde{\xi}_j |z_{nj}| + \beta_{2j} \tilde{\xi}_j \hat{\xi}_j + \beta_{3j} \tilde{W}_j^T \hat{W}_j.
\end{aligned} \tag{49}
$$

With the aid of Lemma 1, we have

$$
\begin{aligned}
\beta_{3j} \tilde{W}_j^T \hat{W}_j &= -\beta_{3j} \tilde{W}_j^T \tilde{W}_j + \beta_{3j} \tilde{W}_j^T W_j^* \\
&\leq -\frac{1}{2} \beta_{3j} \tilde{W}_j^T \tilde{W}_j + \frac{1}{2} \beta_{3j} ||W_j^*||^2.
\end{aligned} \tag{50}
$$

Using Equations 40 and 50 for Equation 49, we can derive

$$
\begin{aligned}
\dot{V}_n &\leq -\sum_{k=1}^{n} c_k z_{k*}^T z_{k*} - \sum_{j=1}^{m} \frac{1}{2} \beta_{2j} \tilde{\xi}_j^2 - \sum_{j=1}^{m} \frac{1}{2} \beta_{3j} \tilde{W}_j^T \tilde{W}_j + \mu_3, \\
&\leq -\mu_4 V_n + \mu_3,
\end{aligned} \tag{51}
$$

where $\mu_{3j} = \frac{1}{2} \beta_{2j} \xi_j^2 + \frac{1}{2} \beta_{3j} ||W_j^*||^2$, $\mu_3 = \sum_{j=1}^{m} \mu_{1j}$, $\mu_4 = \min\{\min_k\{2c_k\}, \min_j\{\beta_{2j}\eta_{2j}, \beta_{3j}\lambda_{\min}(\Gamma_j)\}\}$. Denote $\mu_5 = \frac{\mu_3}{\mu_4}$, by employing Lemma 3, then Equation 51 satisfies

$$
0 \leq V_n(t) \leq \mu_5 + (V_n(0) - \mu_5) e^{-\mu_4 t}. \tag{52}
$$

From the above discussion, we are ready to present the results.

**Theorem 1.** *A group of m high-order nonlinear multiagent systems with a leader (Equations 4-5) is described by Equations 1 to 2, and Assumptions 1 to 5 are satisfied. If the control laws are provided by Equation 46 associated with a series of intermediate controllers (Equations 18, 24, and 30), and the adaptive laws are given by Equations 47 to 48, then* $z_{j*}$ *is UUB; furthermore,* $x_{i*} - I_n \underline{x}_0$ *are UUB, for* $t \geq \bar{T}_s$, $i = 1, \dots, n$.

*Proof.* From Equation 52 in **Step n(b)** and Lemma 4 (ie, boundedness theorem in Khalil[21]), it can be shown that $z_{j*}$ is bounded for $i = 1, \dots, n$; furthermore, $x_{i*} - I_n \underline{x}_0$ is bounded according to Definition 3 and the Theorem 3.2 in Zhou and Xia,[25] which means that $x_{ij} - x_{i0}$ is bounded.

This completes the proof. □

*Remark* 4. The role of parameter $c_{ij} (i = 1, \dots, n, j = 1, \dots, m)$ designed in every virtual controller $\alpha_{ij} (2 \leq i \leq n)$ and control law $u_j$ is to construct the structure of Equations 41 and 51 for the further theoretical analysis. From Cao et al,[17] the parameters $\gamma_1 > 0$ and $\gamma_2 > \bar{f}_0$ designed in Equations 32 and 33 are utilized to guarantee that $\hat{x}_{nj} \to x_{n0}$ and $\hat{f}_{0j} \to f_0(t, \underline{x}_0)$ in finite time. The adaptive parameter $\hat{\xi}_j$ in Equation 47 is designed to reject the adverse effect from the external disturbance $\zeta_j$ and approximation error $\epsilon_j$. The adaptive parameter matrix $\hat{W}_j$ in Equation 48 is utilized to estimate the NN optimal weight matrix $W_j^*$ to further overcome the modeling uncertainty from $f_{j(x_j)}$.

*Remark* 5. From Equations 35 and 42 in **Step n(a)**, and Equations 44 and 51 in **Step n(b)**, it has $\dot{V}_n < 0$, as long as the following conditions hold, respectively.

$$
||z_{i*}|| \geq \sqrt{\frac{\mu_1}{c_i}}, \quad ||z_{i*}|| \geq \sqrt{\frac{\mu_3}{c_i}}.
$$

Therefore, the parameters $c_{ij}$, for $i = 1, \dots, n, j = 1, \dots, m$, determine the final accuracy of $z_{j*}$, ie, the smaller the desired $z_{j*}$, the bigger the controller parameter $c_{ij}$, for $i = 1, \dots, n, j = 1, \dots, m$, are required.

*Remark* 6. Using Equations 44 and 52, it holds that $||z_{i*}|| \leq \sqrt{2\mu_5 + 2(V_n(0) - \mu_5)e^{-\mu_4 t}}$. If $V_n(0) = \mu_5$, taking $\mu_6 = \sqrt{2\mu_5}$ yields $||z_{i*}|| \leq \mu_6$; if $V_n(0) \neq \mu_5$, there must exist $T$ such that $\forall t > T$, $\lim_{t \to T} e^{-\mu_4 t} = 0$, ie, $||z_{i*}|| \leq \mu_6$. Therefore, we conclude that $||z_{i*}|| \leq \mu_6$, which means that $|x_{ij} - x_{i0}| \leq \mu_6$.

## 4 | NUMERICAL EXAMPLE

In this section, we take an example to show the effectiveness of the proposed distributed control algorithm Equation 46. Firstly, a 6-node directed graph is considered, which can be seen in Figure 1. Let 0 be a leader agent. It is obvious that the communication graph $\mathcal{G}$ satisfies Assumption 1. For simplicity, the corresponding adjacent weights $a_{ij}$ between the networked agents are assumed to be 1, and all the others are 0.

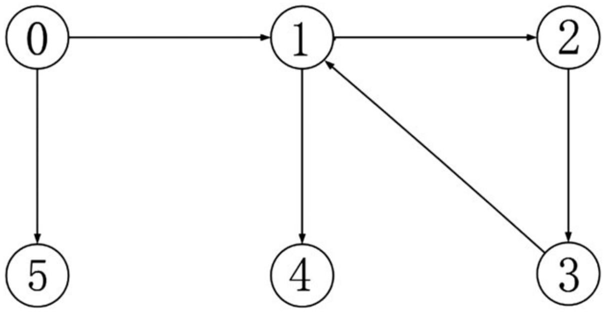


**FIGURE 1** Communication graph $\mathcal{G}$

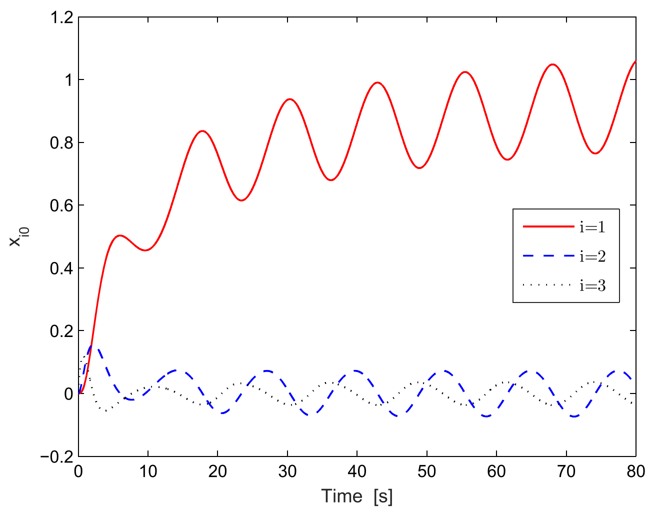


**FIGURE 2** State of the agent 0

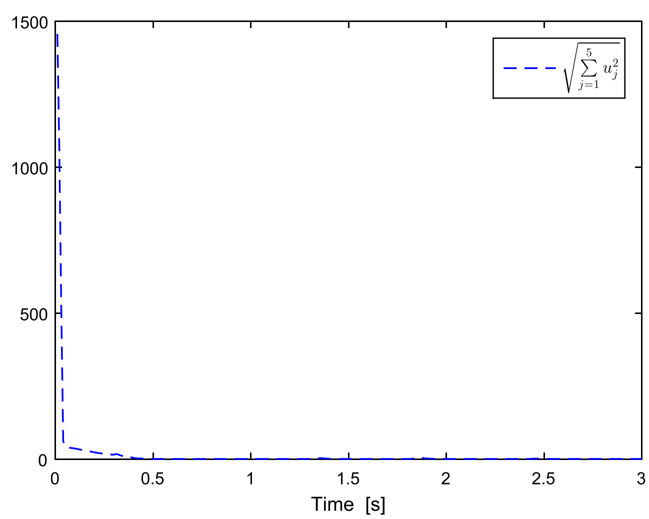


**FIGURE 3** Response of the CPI

Consider the following third-order uncertain nonlinear dynamics:

$$\dot{x}_{1j}(t) = x_{2j},$$
$$\dot{x}_{2j}(t) = x_{3j},$$
$$\dot{x}_{3j}(t) = u_j + f_j(x_{1j}, x_{2j}, x_{3j}) + \zeta_j(t),$$

with

$$\dot{x}_{31}(t) = u_1 + 0.2(x_{11} + x_{31}) + d_1,$$
$$\dot{x}_{32}(t) = u_2 + 0.3(x_{12} + x_{22} - 1) + d_2,$$
$$\dot{x}_{33}(t) = u_3 + 0.3\cos(x_{13} + x_{23}) + d_3,$$
$$\dot{x}_{34}(t) = u_4 + 0.2\sin(x_{14} + x_{24}) + d_4,$$
$$\dot{x}_{35}(t) = u_5 + 0.2\sin(x_{15}) + d_5,$$

where $d_j$ is assumed to be random and bounded by $|d_j| \leq 1$ for $j = 1, \ldots, 5$.

The dynamics of the leader node is given by

$$\dot{x}_{10}(t) = x_{20},$$
$$\dot{x}_{20}(t) = x_{30},$$
$$\dot{x}_{30}(t) = -x_{20} - x_{30} + 0.03\sin(t/2) + 0.06\cos(t/2)$$
$$- (x_{10} + x_{20} - 1)^2(x_{10} + 4x_{20} + 3x_{30} - 1)/3.$$

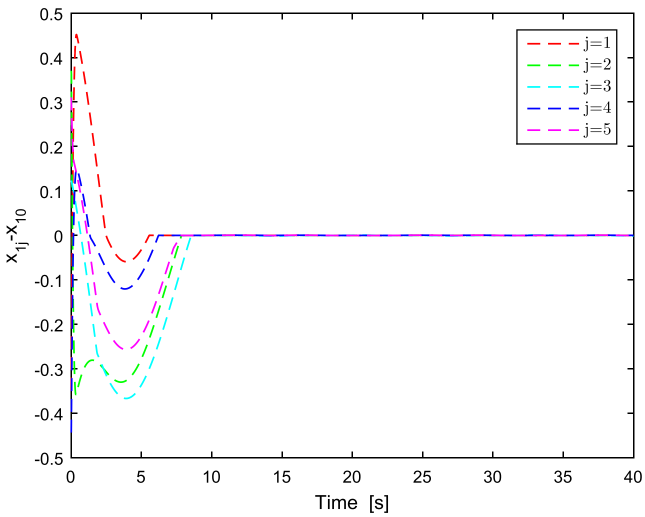


**FIGURE 4** Response of the tracking error $x_{1j} - x_{10}$

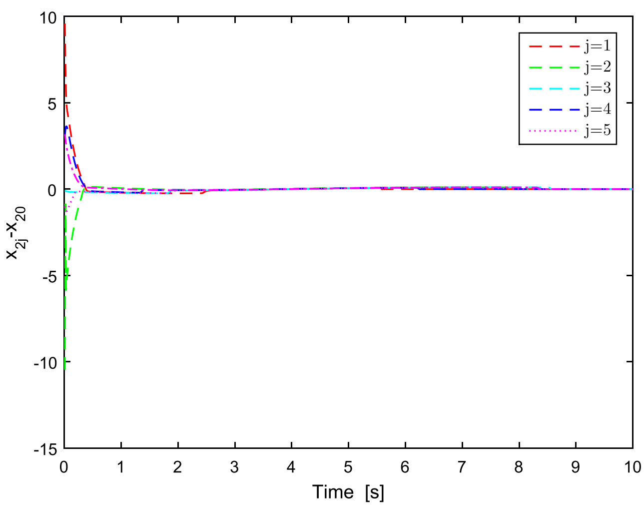


**FIGURE 5** Response of the tracking error $x_{2j} - x_{20}$

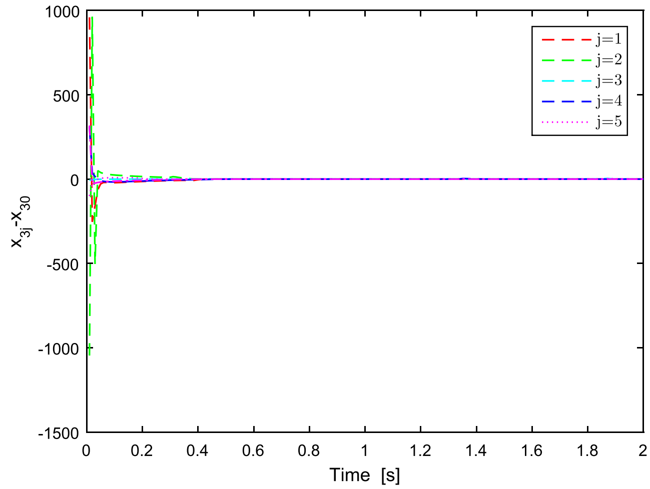


**FIGURE 6** Response of the tracking error $x_{3j} - x_{30}$

**TABLE 1** The number of neurons $p_i$and the running time

| Time $k$ | $k = 500$ | $k = 500$ | $k = 500$ | $k = 500$ | $k = 500$ |
|---|---|---|---|---|---|
| The number of neurons $p_i$ | 3 | 6 | 12 | 24 | 36 |
| Running time, s | 1.289 | 1.305 | 1.319 | 1.356 | 1.409 |

In this simulation, only the actuators in the 1st, 4th, and 5th agents become faulty when $t \geq 15$ seconds. The faulty actuators are described as

$$f_{1u}(t) = \begin{cases} 0 & t < 15 \text{ s} \\ 0.1\text{rand}(t_i) + 0.055\sin(0.5\pi t) & t \geq 15 \text{ s} \end{cases}, \tag{53}$$

$$f_{4u}(t) = \begin{cases} 0 & t < 15 \text{ s} \\ 0.1\text{rand}(t_i) + 0.05\cos(0.5\pi t) & t \geq 15 \text{ s} \end{cases}, \tag{54}$$

$$f_{5u}(t) = \begin{cases} 0 & t < 15 \text{ s} \\ 0.1\text{rand}(t_i) + 0.06\sin(0.5\pi t - \pi/3) & t \geq 15 \text{ s} \end{cases}. \tag{55}$$

In this example, we use 6 neurons for each NN, and the sigmoid basis functions are used. The initial value of NN weights are assumed to be $\hat{W}_j(0) = [0, 0, 0, 0, 0, 0]^T$ for $j = 1, \dots, 5$.

The design parameters are selected as $\gamma_1 = 3, \gamma_2 = 4, c_1 = c_2 = c_3 = 5, \eta_{2j} = \beta_{2j} = \beta_{3j} = 1, \Gamma_j = I_6$. The initial values are chosen as $\hat{x}_{31} = 0.5$, $\hat{x}_{32} = -0.4, \hat{x}_{33} = 0.1, \hat{x}_{34} = 0.2, \hat{x}_{35} = 0.2, \hat{f}_{01} = 0.1, \hat{f}_{02} = -0.2, \hat{f}_{03} = 0.3, \hat{f}_{04} = -0.2, \hat{f}_{05} = -0.3, \hat{\xi}_j = 0$. To examine the overall control effort, we define a comprehensive performance index (CPI) as follows:

$$\textbf{CPI} = \left( \sum_{j=1}^{5} u_j^2 \right)^{1/2}. \tag{56}$$

The simulation results are presented in Figures 2 to 5. The state of the leader node 0 is bounded, which can be seen from Figure 2. Figure 3 shows the response of CPI using Equation 46. Figures 4 to 6 show the time histories of tracking errors $x_{1j} - x_{10}, x_{2j} - x_{20}$ and $x_{3j} - x_{30}$ for $j = 1, \dots, 5$, respectively. From Figure 3, it can be seen that, under the control torque, which are shown in Figure 1, the control objective is achieved. These figures demonstrate the efficiency of the proposed algorithm in guaranteeing distributed tracking despite the presence of complex unknown dynamics, external disturbances, and actuator faults. Therefore, the distributed cooperative control laws in Theorem 1 are effective. Furthermore, Table 1 shows the results of computational load by using different number of neurons $p_i$. The comparisons concerning $p_i$ and the running time of simulation program are implemented in the same work environment (ie, the same computer, programming way, and timekeeping method). The results imply that the computational burden increases as the number of neurons goes up.

## 5 | CONCLUSIONS

With a predesigned second-order sliding mode estimator, a new backstepping-based distributed adaptive NN control scheme was presented in this paper for a group of uncertain nonlinear high-order multiagent systems with actuator faults. The proposed algorithm can overcome the affections of the external disturbances and modeling uncertainties while guaranteeing the convergence of the tracking errors. Furthermore, the stability of the closed-loop systems were ensured step-by-step in the sense of the Lyapunov stability.

## ACKNOWLEDGMENTS

The authors would like to thank the referees for their valuable and helpful comments, which have improved the presentation. The work was supported in part by the Foundation of Cloud Computing and Big Data for Agriculture and Forestry (grant no. 117-612014063), the National Basic Research Program of China (973 Program) (grant no. 2012CB720000), the National Natural Science Foundation of China (grant nos. 61225015, 61105092, 61603094, and 61603095), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (grant no. 61321002), the Natural Science Foundation of Fujian Province of China (grant no. 2016J01284 and 2016J01286), the Research Foundation for Outstanding Young Scholars in the University of Fujian Province of China, and the Research Foundation for Outstanding Young Scholars in Fujian Agriculture and Forestry University (grant no. xjq201612).

## REFERENCES

1. Huang J, Chen C, Fang H, Chen J, Dou L. On consensus of multiple high-order uncertain systems based on distributed backstepping framework. *Int J Adv Robot Syst*. 2014;11(1):1-9.
2. Huang J, Fang H, Dou L, Chen J. An overview of distributed high-order multi-agent coordination. *IEEE/CAA J Autom Sin*. 2014;1(1):1-9.
3. Qiu ZR, Xie LH, Hong YG. Quantized leaderless and leader-following consensus of high-order multi-agent systems with limited data rate. *IEEE Trans Autom Control*. 2016;61(9):2432-2447.
4. Sun X, Ge SS, Zhang J, Cao XB. Region tracking control for high-order multi-agent systems in restricted space. *IET Control Theory Appl*. 2016;10(4):396-406.
5. Yoo SJ. Distributed containment control with predefined performance of high-order multi-agent systems with unknown heterogeneous non-linearities. *IET Control Theory Appl*. 2015;9(10):1571-1578.
6. Xiang J, Li Y, Wei W. Synchronisation of linear high-order multi-agent systems: an internal model approach. *IET Control Theory Appl*. 2013;7(17):2110-2116.
7. Huang J, Dou L, Fang H, Chen J, Yang Q. Distributed backstepping-based adaptive fuzzy control of multiple high-order nonlinear dynamics. *Nonlinear Dynamics*. 2015;81(1-2):63-75.
8. Huang J, Fang H, Chen J, Dou L, Deng F. Cooperative adaptive fuzzy control of high-order nonlinear multi-agent systems with unknown dynamics. *IFAC Proceedings Volumes*. 2014;47(3):9774-9779.
9. Hua C, Zhang L, Guan X. Distributed adaptive neural network output tracking of leader-following high-order stochastic nonlinear multiagent systems with unknown dead-zone input. *IEEE Trans Cybern*. 2017;47(1):177-185.
10. Wang Q, Fu J, Wang J. Leader-follower consensus for high-order integrator systems with Lipschitz nonlinear dynamics. Paper presented at: The 34th Chinese Control Conference. IEEE, Hangzhou, China; 2015.
11. Fu J, Wang J. Finite-time coordinated tracking for high-order uncertain nonlinear multi-agent systems with directed communication graphs. Paper presented at: The 33th Chinese Control Conference. IEEE; 2014.
12. Shi P, Shen Q. Cooperative control of multi-agent systems with unknown state-dependent controlling effects. *IEEE Trans Autom Sci Eng*. 2015;12(3):827-834.
13. Shen Q, Shi P, Shi Y. Distributed adaptive fuzzy control for nonlinear multiagent systems via sliding mode observers. *IEEE Trans Cybern*. 2016;46(12):3086-3097.
14. Shen Q, Jiang B, Shi P, Zhao J. Cooperative adaptive fuzzy tracking control for networked unknown nonlinear multiagent systems with time-varying actuator faults. *IEEE Trans Fuzzy Syst*. 2014;22(3):494-504.
15. Ma HJ, Yang GH. Adaptive fault tolerant control of cooperative heterogeneous systems with actuator faults and unreliable interconnections. *IEEE Trans Autom Control*. 2016;61(11):3240-3255.
16. Zhang H, Frank LL. Adaptive cooperative tracking control of higher-order nonlinear systems with unknown dynamics. *Automatica*. 2012;48(7):1432-1439.
17. Cao Y, Ren W, Meng Z. Decentralized finite-time sliding mode estimators and their applications in decentralized finite-time formation tracking. *Syst Control Lett*. 2010;59(9):522-529.
18. Huang X, Lin W, Yang B. Global finite-time stabilization of a class of uncertain nonlinear systems. *Automatica*. 2005;41(5):881-888.
19. Zou AM, Kumar KD. Robust attitude coordination control for spacecraft formation flying under actuator failures. *J Guid Control Dyn*. 2012;35(35):1247-1255.
20. Zou AM, Kumar KD. Adaptive attitude control of spacecraft without velocity measurements using Chebyshev neural network. *Acta Astronaut*. 2010;66(5):769-779.
21. Khalil HK. *Nonlinear Systems*. 3rd ed. Upper Saddle River, NJ: Prentice-Hall; 2002.
22. Chen CLP, Wen GX, Liu YJ, Wang FY. Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks. *IEEE Trans Neural Netw Learn Syst*. 2014;25(6):1217-1226.
23. Ge SS, Hang CC, Zhang T. Adaptive neural network control of nonlinear systems by state and output feedback. *IEEE Trans Syst Man Cybern, B Cybern*. 1999;29(6):818-828.
24. Liu YJ, Tong S. Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems. *IEEE Trans Cybern*. 2016;46(11):2670-2680.

25. Zhou N, Xia Y. Distributed fault-tolerant control design for spacecraft finite-time attitude synchronization. *Int J Robust Nonlinear Control*. 2016;26:2994-3017.

26. Lewis FL, Jaganathan S, Yesidirek A. *Neural Network Control of Robot Manipulatorsand Nonlinear Systems*. New York: Taylor & Francis; 1998.

**How to cite this article:** Zhou N, Chen R, Xia Y, Huang J. Estimator-based adaptive neural network control of leader-follower high-order nonlinear multiagent systems with actuator faults. *Concurrency Computat: Pract Exper*. 2017;29:e4225. https://doi.org/10.1002/cpe.4225