



Distributed control algorithm for bipartite consensus of the nonlinear time-delayed multi-agent systems with neural networks[☆]



Ding Wang^a, Hongwen Ma^a, Derong Liu^{b,*}

^a The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

^b School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

ARTICLE INFO

Article history:

Received 6 June 2015

Received in revised form

24 August 2015

Accepted 1 October 2015

Communicated by Weiming Xiang

Available online 23 October 2015

Keywords:

Bipartite consensus

Distributed control algorithm

Multi-agent systems

Neural networks

Time delays

ABSTRACT

A neural-network-based distributed control algorithm is established for bipartite consensus of the nonlinear multi-agent systems with time delays. By using a backstepping technique, a desired reference signal is introduced. Then, neural networks are used to learn the unknown nonlinear dynamics of the multi-agent systems. In order to eliminate the effects of time delays, the information of a constructed Lyapunov–Krasovskii functional is included in the distributed control algorithm. However, it can induce singularities in the distributed control algorithm. Therefore, a σ -function is utilized to circumvent this problem. With the developed distributed control algorithm, bipartite consensus can be reached if the communication graph is structurally balanced. Finally, simulation examples are conducted to demonstrate the validity of the main theorem.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Multi-agent systems have attracted a lot of attention during the last decade. Variety of problems studied includes optimal control problems [1–5], output-based control problems [6–9], containment problems [10,11], formation problems [12–14], event-triggered problems [15,16] and consensus problems [17–30]. For more details, refer to the survey papers [31–35] and the references therein. However, the communication weights of multi-agent systems in all the papers above are nonnegative and they have been fully investigated. Due to the existence of negative communication weights, bipartite consensus is a new branch of traditional consensus problems. Therefore, it is worthy of investigating how to design distributed control algorithms for bipartite consensus problems.

In many physical scenarios, it is reasonable to assume that some of the agents are cooperative while the rest are competitive. For example, one community can be divided into two clusters holding the opposite opinions as shown in Fig. 1. In [36], negative weights were introduced to the communication topology and bipartite consensus can be reached in the presence of antagonistic

interactions. However, it only dealt with the simplest situation where the first-order dynamics of each agent was equal to the control input. Subsequently, bipartite consensus problems were extended to formation control [13] and directed signed networks [37] with the same dynamics. In [38], the dynamics of the multi-agent systems were high-order and bipartite consensus can be reached under the stabilizability assumption with an equilibrium between two fully competing groups. However, none of them takes time delays into consideration. Due to the limit of the communication capability, time delays are ubiquitous in physical implementations and they will induce instability. Furthermore, the unknown nonlinear dynamics are considered in this paper to generalize bipartite consensus problems to a complex external environment. Therefore, it is important to investigate bipartite consensus of nonlinear time-delayed multi-agent systems.

In [39], adaptive neural control was introduced to solve the uncertain MIMO nonlinear systems. In [17] and [18], a decentralized adaptive control with neural networks (NNs) was proposed for multi-agent systems with unknown dynamics, which made great contributions to the studies of nonlinear multi-agent systems. A neural network technique is a powerful tool for learning the unknown dynamics [40]. In [41], an adaptive neural control protocol was utilized for a class of strict-feedback nonlinear systems with unknown time delays. In [42], a Lyapunov–Krasovskii functional and Young's inequality were used for the consensus of time-delayed multi-agent systems. We borrow the technique of Lyapunov–Krasovskii functional from [41,42] to eliminate the negative effects of time delays. However, this technique will induce singularities in the distributed control algorithm.

[☆]This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61233001, 61273140, 61304086, and 61374105, in part by Beijing Natural Science Foundation under Grant no. 4132078, and in part by the Early Career Development Award of SKLMCCS.

* Corresponding author. Tel.: +86 10 62332905.

E-mail addresses: ding.wang@ia.ac.cn (D. Wang), mahongwen2012@ia.ac.cn (H. Ma), derong@ustb.edu.cn (D. Liu).

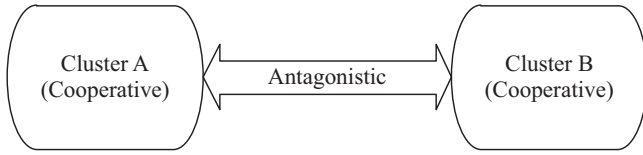


Fig. 1. Two clusters with cooperative behaviors inside and antagonistic behaviors between each other.

Thus, we utilize a σ -function to deal with this problem. Furthermore, to the best of authors' knowledge, it is the first time to investigate bipartite consensus of the time-delayed nonlinear multi-agent systems of second order with the σ -function developed. The main contributions of this paper are listed as follows:

- (1) A distributed control algorithm with neural network technique is developed to achieve bipartite consensus of the nonlinear time-delayed multi-agent systems.
- (2) A σ -function is introduced to circumvent the singularities in the distributed control algorithm and the backstepping technique is utilized to design a reference signal which can reduce the difficulty of achieving bipartite consensus.
- (3) A Lyapunov–Krasovskii functional is introduced to eliminate the negative effects of time delays and enhance the reliability of the learning capability of NNs.

The rest of this paper is organized as follows. Basic definitions of bipartite consensus and radial basis function neural networks (RBFNNs) are given in Section 2. The distributed control algorithm with NNs is developed for bipartite consensus in Section 3. Implementations of bipartite consensus are conducted to demonstrate the effectiveness of the developed algorithm in Section 4. Conclusion is given in Section 5.

Notations: $(\cdot)^T$ denotes the transpose of a given matrix. $\text{tr}(\cdot)$ is the trace of a given matrix. $\|\cdot\|$ is the Frobenius norm or Euclidian norm. \otimes stands for the Kronecker product. $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the smallest nonzero eigenvalue and the largest eigenvalue of a given matrix, respectively. $\text{diag}(\cdot)$ represents a diagonal matrix.

2. Preliminaries

2.1. Signed graph and bipartite consensus

A triplet $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ is called a signed graph if $\mathcal{V} = \{1, 2, \dots, N\}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\mathcal{A} = (A_{ij}) \in \mathbb{R}^{N \times N}$ is the adjacency matrix of \mathcal{G} . Denote A_{ij} as the element of the i th row and j th column of the matrix \mathcal{A} . The i th node in a signed graph \mathcal{G} represents the i th agent, and a directed path from node i to node j is denoted as an ordered pair $(i, j) \in \mathcal{E}$ which means that agent i can directly transfer its information to agent j and $A_{ji} \neq 0 \Leftrightarrow (i, j) \in \mathcal{E}$. The interaction between the i th and the j th agent is cooperative if $A_{ij} > 0$. It is competitive if $A_{ij} < 0$ and there is no interaction if $A_{ij} = 0$. Note that self-loops will not be considered in this paper, i.e., $A_{ii} = 0$, $i = 1, 2, \dots, N$. The Laplacian matrix of the signed graph is given as follows:

$$\mathcal{L}_{ij} = \begin{cases} \sum_{k \in \mathcal{N}_i} |A_{ik}| & \text{if } i = j; \\ -A_{ij} & \text{if } i \neq j. \end{cases} \quad (1)$$

The following two definitions are important concepts in this paper.

Definition 1 (Structurally balanced, cf. Altafini [36]). In this paper, a signed graph $\mathcal{G}(\mathcal{A})$ is said to be structurally balanced if it contains a bipartition of the sets of nodes \mathcal{V}_1 and \mathcal{V}_2 , where $\mathcal{V} = \mathcal{V}_1 \cup$

\mathcal{V}_2 , $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ such that $A_{ij} \geq 0, \forall i, j \in \mathcal{V}_p (p \in \{1, 2\}); A_{ij} \leq 0, \forall i \in \mathcal{V}_p, j \in \mathcal{V}_q, p \neq q (p, q \in \{1, 2\})$. Otherwise, it is called structurally unbalanced.

Definition 2 (Bipartite consensus). If for any initial conditions $x_i(0)$, $i \in \mathcal{V}$, the distributed control algorithm will make the following conditions hold:

$$\begin{cases} \lim_{t \rightarrow \infty} \|x_j(t) - x_i(t)\| = 0, & \forall i, j \in \mathcal{V}_1 \text{ or } \forall i, j \in \mathcal{V}_2; \\ \lim_{t \rightarrow \infty} \|x_j(t) + x_i(t)\| = 0, & \forall i \in \mathcal{V}_1 \text{ and } \forall j \in \mathcal{V}_2, \end{cases} \quad (2)$$

where \mathcal{V}_1 and \mathcal{V}_2 are the distinct sets defined in Definition 1. Then, we say that the multi-agent systems reach bipartite consensus.

2.2. Radial basis function neural networks

In practice, we usually employ a neural network as the function approximator to model an unknown function. RBFNN is a potential candidate for approximating the unknown dynamics of the multi-agent systems in virtue of “linear-in-weight” property. In Fig. 2, a continuous unknown nonlinear function vector $h(x) = [h_1(x), h_2(x), \dots, h_m(x)]^T: \mathbb{R}^m \rightarrow \mathbb{R}^m$ can be approximated by RBFNNs:

$$h(x) = W^T \Phi(x), \quad (3)$$

where $x = [x_1, x_2, \dots, x_m]^T \in \mathbb{R}^m$ is the input vector, $W \in \mathbb{R}^{p \times m}$ is the weight matrix and p represents the number of neurons. $\Phi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_p(x)]^T$ is the activation function vector and

$$\varphi_i(x) = \exp \left[\frac{-(x - \mu_i)^T (x - \mu_i)}{\delta_i^2} \right], \quad i = 1, 2, \dots, p, \quad (4)$$

where $\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{im}]^T$ is the center of receptive field and δ_i is the width of Gaussian function. RBFNNs can approximate any continuous function over a compact set with a given precision. Therefore, for a given positive constant θ_N , there exists an ideal weight matrix W^* such that

$$h(x) = W^{*T} \Phi(x) + \theta, \quad (5)$$

where $\theta \in \mathbb{R}^m$ is the approximating error with $\|\theta\| < \theta_N$.

However, it is difficult to obtain W^* in real applications. Thus, we denote \hat{W} as the estimation of the ideal weight matrix W^* . The estimation of $h(x)$ can be written as

$$\hat{h}(x) = \hat{W}^T \Phi(x), \quad (6)$$

where \hat{W} can be updated online. The online updating law will be given in Section 3.

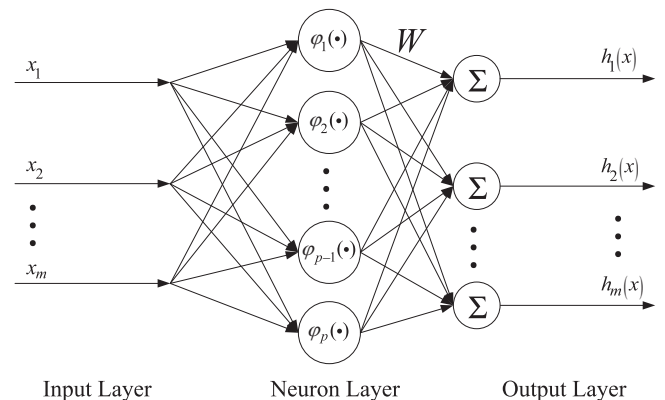


Fig. 2. Structure of the radial basis function neural networks.

3. Distributed control algorithm for bipartite consensus

We discuss the second-order case where multi-manipulator systems are one of its physical applications. It can be described as follows:

$$\ddot{x}_i(t) = f_i(x_i(t), \dot{x}_i(t)) + g_i(\dot{x}_i(t - \tau_i)) + u_i(t), \quad i = 1, 2, \dots, N, \quad (7)$$

where $x_i(\cdot) \in \mathbb{R}^m$ is the state vector, τ_i is the unknown time delay, $u_i(\cdot) \in \mathbb{R}^m$ is the control vector, $f_i(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ and $g_i(\cdot) : \mathbb{R}^m \rightarrow \mathbb{R}^m$ are continuous but unknown nonlinear vector functions. For simplicity, in the sequel we will ignore time expression t in case there is no confusion.

Remark 1. In practice, x_i , \dot{x}_i and \ddot{x}_i can represent position, velocity and acceleration of manipulator i , respectively. Furthermore, $g_i(\cdot)$ is the unknown nonlinear time-delay term which can be seen as the frictional impact on manipulator i .

The following three assumptions and one lemma are helpful for demonstrating Theorem 1.

Assumption 1. $g_i(\dot{x}_i(t - \tau_i))$, $i = 1, 2, \dots, N$, are unknown smooth nonlinear functions. The inequalities $\|g_i(\dot{x}_i(t))\| \leq \phi_i(\dot{x}_i(t))$, $i = 1, 2, \dots, N$, hold and $\phi_i(\cdot)$, $i = 1, 2, \dots, N$, are known positive smooth scalar functions. Furthermore, $g_i(0) = 0$ and $\phi_i(0) = 0$, $i = 1, 2, \dots, N$.

Assumption 2. The unknown time delays τ_i , $i = 1, 2, \dots, N$, are bounded by a known constant τ_{\max} , i.e., $\tau_i \leq \tau_{\max}$, $i = 1, 2, \dots, N$.

Assumption 3. θ_i , $i = 1, 2, \dots, N$, are bounded approximation errors of RBFNNs, i.e.,

$$\|\theta_i\| \leq \theta_{N_i}, \quad i = 1, 2, \dots, N, \quad (8)$$

where θ_{N_i} , $i = 1, 2, \dots, N$, are positive constants.

Lemma 1 (cf. Ge and Wang [39]). $V(t) \geq 0$ denotes a continuous function and $V(0)$ is bounded. If $\forall t \geq 0$, $\dot{V}(t) \leq -b_1 V(t) + b_2$, where b_1 and b_2 are positive constants, then the following inequality holds

$$V(t) \leq V(0)e^{-b_1 t} + \frac{b_2}{b_1}(1 - e^{-b_1 t}). \quad (9)$$

Our aim is to design a distributed control algorithm to drive the nonlinear time-delayed multi-agent systems toward bipartite consensus. The distributed control algorithm is divided into four parts and they are linear feedback term, neural network term, time-delay eliminated term and second-order information term. Before proceeding, we introduce a Lyapunov–Krasovskii functional as follows:

$$L_Q(t) = \frac{1}{2} \sum_{i=1}^N \int_{t-\tau_i}^t Q_i(\dot{x}_i(\zeta)) d\zeta, \quad (10)$$

where $Q_i(\dot{x}_i(\zeta)) = \phi_i^2(\dot{x}_i(\zeta))$. The time derivative of $L_Q(t)$ is

$$\dot{L}_Q(t) = \frac{1}{2} \sum_{i=1}^N (\phi_i^2(\dot{x}_i(t)) - \phi_i^2(\dot{x}_i(t - \tau_i))). \quad (11)$$

Let $y_{i1} = x_i$ and $y_{i2} = \dot{x}_i$. Then, Eq. (7) can be rewritten in the following form:

$$\begin{cases} \dot{y}_{i1} = y_{i2}, \\ \dot{y}_{i2} = f_i(y_{i1}, y_{i2}) + g_i(y_{i2}(t - \tau_i)) + u_i, \end{cases} \quad i = 1, 2, \dots, N. \quad (12)$$

In the sequel, for convenient analysis, we will ignore the declaration that $i = 1, 2, \dots, N$ and concentrate on agent i . We employ the backstepping technique and suppose that

$$y_{i2d} = -k_i \sum_{j \in \mathcal{N}_i} |A_{ij}| (y_{i1} - \text{sgn}(A_{ij}) y_{j1}), \quad (13)$$

where $\text{sgn}(\cdot)$ is a sign function given as follows:

$$\text{sgn}(A_{ij}) = \begin{cases} 1, & A_{ij} > 0, \\ 0, & A_{ij} = 0, \\ -1, & A_{ij} < 0. \end{cases} \quad (14)$$

Then, we can obtain an error signal between the real state y_{i2} and the virtual state y_{i2d} , i.e., $v_{ei} = y_{i2} - y_{i2d}$. Consequently, the time derivative of v_{ei} is

$$\dot{v}_{ei} = \dot{y}_{i2} - \dot{y}_{i2d} = f_i(y_{i1}, y_{i2}) + g_i(y_{i2}(t - \tau_i)) + u_i + k_i \sum_{j \in \mathcal{N}_i} |A_{ij}| (y_{i2} - \text{sgn}(A_{ij}) y_{j2}). \quad (15)$$

We utilize RBFNNs to approximate $f_i(y_{i1}, y_{i2})$. The distributed control algorithm is designed as follows:

$$\begin{aligned} u_i = & -\rho_i(t)v_{ei} - \hat{W}_i^T \Phi_i(y_i) - \frac{1}{2} \frac{v_{ei}}{\|v_{ei}\|^2 + \sigma(v_{ei})} \phi_i^2(y_{i2}) \\ & - k_i \sum_{j \in \mathcal{N}_i} |A_{ij}| (y_{i2} - \text{sgn}(A_{ij}) y_{j2}), \end{aligned} \quad (16)$$

where

$$\begin{aligned} \rho_i(t) = & k_{i0} + \frac{1}{2} + \frac{1}{2\omega_i} \left(1 + \frac{1}{\|v_{ei}\|^2 + \sigma(v_{ei})} h_i \right), \\ h_i = & \int_{t-\tau_{\max}}^t Q_i(y_{i2}(\zeta)) d\zeta + \omega_i \|y_{i2}\|^2 + (\omega_i + \lambda_{\max}(M)) \|y_{ei}\|^2, \\ y_{ei} = & \sum_{j \in \mathcal{N}_i} |A_{ij}| (y_{i1} - \text{sgn}(A_{ij}) y_{j1}), \\ y_i = & [y_{i1}^T, y_{i2}^T]^T, \\ \sigma(v_{ei}) = & \begin{cases} 1, & \text{if } \|v_{ei}\| < C, \\ 0, & \text{if } \|v_{ei}\| \geq C, \end{cases} \end{aligned} \quad (17)$$

and C is a predetermined threshold (e.g., $C=0.01$). Furthermore, $\omega_i > 0$ and M are defined in (21). Next, we discuss the structure of the distributed control algorithm.

- (1) The linear feedback term $-\rho_i(t)v_{ei}$ is utilized to drive the i th agent to the final bipartite consensus state. It contains all the information which can be used by agent i to guide its direction towards bipartite consensus. Moreover, if bipartite consensus can be reached, then $-\rho_i(t)v_{ei}$ has no impact on the multi-agent system (7).
- (2) In order to model the unknown dynamics in (15), the neural network term $-\hat{W}_i^T \Phi_i(y_i)$ is used to learn the characteristics of $f_i(y_i)$ online. \hat{W}_i represents the estimation of RBFNN weight matrix of agent i . Motivated by the projection algorithm, we can derive the adaptive updating law as follows:

$$\dot{\hat{W}}_i = \begin{cases} a_i \Phi_i(y_i) v_{ei}^T, & \text{if } \text{tr}(\hat{W}_i^T \hat{W}_i) < W_i^{\max} \text{ or} \\ & \text{if } \text{tr}(\hat{W}_i^T \hat{W}_i) = W_i^{\max} \text{ and } v_{ei}^T \hat{W}_i^T \Phi_i(y_i) < 0; \\ a_i \Phi_i(y_i) v_{ei}^T - a_i \frac{v_{ei}^T \hat{W}_i^T \Phi_i(y_i)}{\text{tr}(\hat{W}_i^T \hat{W}_i)} \hat{W}_i, & \\ & \text{if } \text{tr}(\hat{W}_i^T \hat{W}_i) = W_i^{\max} \text{ and } v_{ei}^T \hat{W}_i^T \Phi_i(y_i) \geq 0, \end{cases} \quad (18)$$

where $a_i > 0$ has impact on the updating rate of \hat{W}_i and $W_i^{\max} > 0$ is utilized to constrain the value of \hat{W}_i . It is noted that the initial value $\hat{W}_i(0)$ should satisfy

$$\text{tr}(\hat{W}_i^T(0) \hat{W}_i(0)) \leq W_i^{\max}. \quad (19)$$

Thus, we let $\hat{W}_i(0)$ be a zero matrix. Furthermore, according to Lemma 2 in [17], if the updating law is expressed as (18), then $\text{tr} \hat{W}_i^T(t) \hat{W}_i(t) \leq W_i^{\max}$, $\forall t \geq 0$. Note that in (18), the final aim is to regulate the angle between v_{ei} and $\hat{W}_i^T \Phi_i(y_i)$ to 90° .

- (3) $-(1/2)v_{ei}/(\|v_{ei}\|^2 + \sigma(v_{ei}))\phi_i^2(y_{i2})$ is the time-delay elimination term which is introduced to eliminate the effects of time delays. Note that if the σ -function is removed from the denominator, it has singularity at $\|v_{ei}\| = 0$. In physical applications, it is nonsense because of the infinite control induced by $\|v_{ei}\| = 0$. Therefore, we should exclude zero case and σ -function is a wise choice for solving this problem.
- (4) $-k_i \sum_{j \in \mathcal{N}_i} |A_{ij}|(y_{i2} - \text{sgn}(A_{ij})y_{j2})$ is the second-order information term and it includes the information of velocities that agent i can obtain.

Theorem 1. The nonlinear time-delayed multi-agent system is described by (7) with Assumptions 1–3. If the distributed control algorithm is expressed as (16), the updating law of RBFNN weight matrix is given in (18), the communication topology is structurally balanced, and the initial condition is satisfied with (19), then the multi-agent system (7) can reach bipartite consensus.

Proof. We construct a Lyapunov function containing error signal v_{ei} , $i = 1, 2, \dots, N$, as follows:

$$\begin{aligned} V(t) &= V_{y_1}(t) + L_Q(t) + \frac{1}{2} \sum_{i=1}^N \text{tr} \left(\frac{1}{a_i} \tilde{W}_i^T \tilde{W}_i \right) + \frac{1}{2} v_e^T v_e \\ &= \frac{1}{2} \hat{y}_1^T (\mathcal{L} \otimes I_m) \hat{y}_1 + \frac{1}{2} \sum_{i=1}^N \int_{t-\tau_i}^t Q_i(y_{i2}(\zeta)) d\zeta \\ &\quad + \frac{1}{2} \sum_{i=1}^N \text{tr} \left(\frac{1}{a_i} \tilde{W}_i^T \tilde{W}_i \right) + \frac{1}{2} v_e^T v_e, \end{aligned} \quad (20)$$

where $V_{y_1}(t) = (1/2) \hat{y}_1^T (\mathcal{L} \otimes I_m) \hat{y}_1$, $\hat{y}_1 = [y_{11}^T, y_{21}^T, \dots, y_{N1}^T]^T$ and $v_e = [v_{e1}^T, v_{e2}^T, \dots, v_{eN}^T]^T$. Note that if $\|v_{ei}\| = 0$, then $y_{i2} = y_{i2d}$, i.e., $\hat{y}_{i1} = y_{i2d} = -k_i \sum_{j \in \mathcal{N}_i} |A_{ij}|(y_{i1} - \text{sgn}(A_{ij})y_{j1})$. It is obvious that this is a traditional distributed control algorithm for bipartite consensus in [36]. Therefore, in the sequel we will focus on the case where $\|v_{ei}\| \neq 0$. Then, we can infer that

$$\begin{aligned} \frac{dV(t)}{dt} &= y_e^T \dot{y}_1 + \frac{1}{2} \sum_{i=1}^N \left(\phi_i^2(y_{i2}(t)) - \phi_i^2(y_{i2}(t-\tau_i)) \right) - \sum_{i=1}^N \text{tr} \left(\frac{1}{a_i} \tilde{W}_i^T \dot{\tilde{W}}_i \right) + v_e^T \dot{v}_e \\ &= y_e^T \dot{y}_2 + \frac{1}{2} \sum_{i=1}^N \left(\phi_i^2(y_{i2}(t)) - \phi_i^2(y_{i2}(t-\tau_i)) \right) - \sum_{i=1}^N \text{tr} \left(\frac{1}{a_i} \tilde{W}_i^T \dot{\tilde{W}}_i \right) \\ &\quad + \sum_{i=1}^N v_{ei}^T (u_i - \dot{y}_{i2d}) + \sum_{i=1}^N v_{ei}^T (f_i(y_{i1}, y_{i2}) + g_i(y_{i2}(t-\tau_i))), \end{aligned}$$

where $\tilde{W}_i = W_i^* - \hat{W}_i$, $\hat{y}_2 = [y_{12}^T, y_{22}^T, \dots, y_{N2}^T]^T$ and $y_e = [y_{e1}^T, y_{e2}^T, \dots, y_{eN}^T]^T$. Furthermore, the communication topology is connected and structurally balanced. Thus, according to Lemma 1 in [36], zero is an m -multiplicity eigenvalue of $\mathcal{L} \otimes I_m$ and T contains eigenvectors of $\mathcal{L} \otimes I_m$ corresponding to the eigenvalue matrix $\Lambda = \text{diag}(0I_m, \lambda_2 I_m, \lambda_3 I_m, \dots, \lambda_n I_m)$, where $TT^T = T^T T = I_{mN}$ and $T^{-1} = T^T$. Hence,

$$\begin{aligned} \hat{y}_1^T (\mathcal{L} \otimes I_m) \hat{y}_1 &= \hat{y}_1^T T^T \Lambda T \hat{y}_1 \\ &= \hat{y}_1^T T^T \sqrt{\Lambda} \sqrt{\Lambda} T \hat{y}_1 \\ &= \hat{y}_1^T T^T \sqrt{\Lambda} \sqrt{\Lambda} \sqrt{\Lambda^{-1}} \sqrt{\Lambda^{-1}} \sqrt{\Lambda} \sqrt{\Lambda} T \hat{y}_1 \\ &= \hat{y}_1^T T^T \Lambda T T^T \Lambda^{-1} T T^T \Lambda T \hat{y}_1 \\ &= \hat{y}_1^T (\mathcal{L} \otimes I_m)^T M (\mathcal{L} \otimes I_m) \hat{y}_1 \\ &= y_e^T M y_e, \end{aligned} \quad (21)$$

where

$$\begin{aligned} \sqrt{\Lambda} &= \text{diag}(0I_m, \sqrt{\lambda_2} I_m, \sqrt{\lambda_3} I_m, \dots, \sqrt{\lambda_n} I_m), \\ \bar{\Lambda} &= \text{diag}(\lambda_2 I_m, \lambda_2 I_m, \lambda_3 I_m, \dots, \lambda_n I_m), \\ \sqrt{\bar{\Lambda}} &= \text{diag}(\sqrt{\lambda_2} I_m, \sqrt{\lambda_2} I_m, \sqrt{\lambda_3} I_m, \dots, \sqrt{\lambda_n} I_m), \end{aligned}$$

and $M = T^T \bar{\Lambda}^{-1} T$. Then, with Assumptions 1 and 3, we substitute (16) into dV/dt to obtain

$$\begin{aligned} \frac{dV(t)}{dt} &\leq \frac{1}{2} \sum_{i=1}^N (\|y_{ei}\|^2 + \|y_{i2}\|^2) - \sum_{i=1}^N \text{tr} \left(\frac{1}{a_i} \tilde{W}_i^T \dot{\tilde{W}}_i \right) \\ &\quad + \frac{1}{2} \sum_{i=1}^N \left(\phi_i^2(y_{i2}(t)) - \phi_i^2(y_{i2}(t-\tau_i)) \right) + \sum_{i=1}^N \theta_{N_i} \\ &\quad + \sum_{i=1}^N v_{ei}^T \tilde{W}_i^T \Phi_i(y_i) - \sum_{i=1}^N \left(\rho_i(t) - \frac{1}{2} \right) \|v_{ei}\|^2 \\ &\quad + \frac{1}{2} \sum_{i=1}^N \left(\phi_i^2(y_{i2}(t-\tau_i)) - \phi_i^2(y_{i2}(t)) \right). \end{aligned}$$

According to (18), we discuss the following two cases.

(a) When $\dot{\tilde{W}}_i = a_i \Phi_i(y_i) v_{ei}^T$, we have

$$\text{tr} \left(\tilde{W}_i^T \left(\frac{1}{a_i} \dot{\tilde{W}}_i - \Phi_i(y_i) v_{ei}^T \right) \right) = 0.$$

(b) When $\dot{\tilde{W}}_i = a_i \Phi_i(y_i) v_{ei}^T - a_i (v_{ei}^T \tilde{W}_i^T \Phi_i(y_i) / \text{tr}(\tilde{W}_i^T \tilde{W}_i)) \tilde{W}_i$, we have

$$\text{tr} \left(\tilde{W}_i^T \left(\frac{1}{a_i} \dot{\tilde{W}}_i - \Phi_i(y_i) v_{ei}^T \right) \right) = - \frac{v_{ei}^T \tilde{W}_i^T \Phi_i(y_i)}{\text{tr}(\tilde{W}_i^T \tilde{W}_i)} \text{tr}(\tilde{W}_i^T \tilde{W}_i).$$

Furthermore, we can obtain

$$\begin{aligned} \text{tr}(\tilde{W}_i^T \dot{\tilde{W}}_i) &= \text{tr}(\tilde{W}_i^T W_i^*) - \text{tr}(\tilde{W}_i^T \tilde{W}_i) \\ &= \frac{1}{2} [\text{tr}(\tilde{W}_i^T W_i^*) + \text{tr}(W_i^{*T} \tilde{W}_i)] - \text{tr}(\tilde{W}_i^T \tilde{W}_i) \\ &= \frac{1}{2} [\text{tr}(W_i^{*T} W_i^*) - \text{tr}(\hat{W}_i^T W_i^*)] + \frac{1}{2} [\text{tr}(\hat{W}_i^T \tilde{W}_i) \\ &\quad + \text{tr}(\tilde{W}_i^T \tilde{W}_i)] - \text{tr}(\tilde{W}_i^T \tilde{W}_i) \\ &= \frac{1}{2} \text{tr}(W_i^{*T} W_i^*) - \frac{1}{2} \text{tr}(\hat{W}_i^T \tilde{W}_i) - \frac{1}{2} \text{tr}(\tilde{W}_i^T \tilde{W}_i) \\ &\leq 0 \end{aligned}$$

with

$$\text{tr}(\tilde{W}_i^T \tilde{W}_i) \geq 0$$

and

$$\text{tr}(\hat{W}_i^T \tilde{W}_i) = W_i^{\max} \geq \text{tr}(W_i^{*T} W_i^*).$$

Then, if

$$\text{tr}(\hat{W}_i^T \tilde{W}_i) = W_i^{\max} > 0 \quad \text{and} \quad v_{ei}^T \tilde{W}_i^T \Phi_i(y_i) \geq 0,$$

we can obtain

$$\text{tr} \left(\tilde{W}_i^T \left(\frac{1}{a_i} \dot{\tilde{W}}_i - \Phi_i(y_i) v_{ei}^T \right) \right) \geq 0. \quad (22)$$

Therefore, in the above two cases, the inequality (22) always holds.

In terms of Assumption 2, we have

$$\frac{1}{2} \sum_{i=1}^N \int_{t-\tau_i}^t Q_i(y_{i2}(\zeta)) d\zeta \leq \frac{1}{2} \sum_{i=1}^N \int_{t-\tau_{\max}}^t Q_i(y_{i2}(\zeta)) d\zeta.$$

Thus, with (17) we can obtain

$$\begin{aligned} \frac{dV(t)}{dt} &\leq \sum_{i=1}^N \left(-k_{i0} \|v_{ei}\|^2 - \frac{1}{2\omega_i} \|v_{ei}\|^2 - \frac{\lambda_{\max}(M)}{2\omega_i} \|y_{ei}\|^2 \right) \\ &\quad - \sum_{i=1}^N \frac{2W_i^{\max}}{\omega_s a_i} + \sum_{i=1}^N \frac{2W_i^{\max}}{\omega_s a_i} + \sum_{i=1}^N \theta_{N_i} \end{aligned}$$

$$\begin{aligned}
& -\frac{1}{2\omega_i} \sum_{i=1}^N \int_{t-\tau_{\max}}^t Q_i(y_{i2}(\zeta)) d\zeta - \text{tr} \left(\hat{W}_i^T \left(\frac{1}{a_i} \dot{W}_i - \Phi_i(y_i) v_{ei}^T \right) \right) \\
& \leq -\frac{1}{\omega_s} V_{\hat{y}_1}(t) - \frac{1}{\omega_s} L_Q(t) - \frac{1}{2\omega_s} v_e^T v_e - \frac{1}{2\omega_s} \sum_{i=1}^N \text{tr} \left(\frac{1}{a_i} \dot{W}_i^T \hat{W}_i \right) \\
& \quad + \sum_{i=1}^N \frac{2W_i^{\max}}{\omega_s a_i} + \theta_s \leq -\frac{1}{\omega_s} V(t) + \sum_{i=1}^N \frac{2W_i^{\max}}{\omega_s a_i} + \theta_s,
\end{aligned}$$

where $\omega_s = \max_{i \in \mathcal{V}} \omega_i$ and $\theta_s = \sum_{i=1}^N \theta_{N_i}$.

On the basis of Lemma 1, we have

$$V(t) \leq V(0)e^{-(1/\omega_s)t} + \nu_s \left(1 - e^{-(1/\omega_s)t} \right), \quad (23)$$

where $\nu_s = \sum_{i=1}^N 2W_i^{\max}/a_i + \omega_s \theta_s$. Since all the terms in (20) are nonnegative, as $t \rightarrow \infty$ we can obtain that $V_{\hat{y}_1}(t) \leq \nu_s$. That is, $\sum_{(i,j) \in \mathcal{E}} |\mathcal{A}_{ij}| (y_{i1} - \text{sgn}(\mathcal{A}_{ij}) y_{j1})^2 \leq \nu_s$. By choosing the parameters W_i^{\max} , a_i , ω_i , θ_i and \mathcal{A}_{ij} properly, we can eventually derive that

$$\|y_{i1} - \text{sgn}(\mathcal{A}_{ij}) y_{j1}\| \leq \sqrt{\frac{\nu_s}{|\mathcal{A}_{ij}|}}, \quad (i, j) \in \mathcal{E}, \quad (24)$$

where $\sqrt{\nu_s/|\mathcal{A}_{ij}|}$ can be set small enough to meet the requirements in Definition 2. Therefore, bipartite consensus can be achieved. \square

Algorithm 1. Distributed control algorithm for bipartite consensus of nonlinear time-delayed multi-agent systems.

1. **In:** Parameters of the multi-agent system (7), W_i^{\max} , θ_{N_i} , k_{i0} , a_i , $W_i(0)$, \mathcal{A}_{ij} , τ_{\max} , k_i , μ_i , δ_i , and the initial conditions $x_i(0)$ and $\dot{x}_i(0)$
2. **Out:** $x_i(t)$, $\dot{x}_i(t)$, $x_i^e = \|y_{ei}\|$.
3. **If** topology \mathcal{G} is structurally unbalanced:
4. Terminate the algorithm.
5. **While** bipartite consensus is not reached:
6. Calculate feedback coefficient $\rho_i(t)$;
7. Obtain linear feedback signal $\rho_i(t) v_{ei}$;
8. Calculate the output of RBFNN $\hat{W}_i^T \Phi_i(y_i)$;
9. Update \hat{W}_i by (18);
10. Calculate $-\frac{1}{2} \frac{v_{ei}}{\|v_{ei}\|^2 + \sigma(v_{ei})} \phi_i^2(y_{i2})$;
11. Calculate $-k_i \sum_{j \in \mathcal{N}_i} |\mathcal{A}_{ij}| (y_{i2} - \text{sgn}(\mathcal{A}_{ij}) y_{j2})$;
12. Use input signal u_i to control agent i ;
13. Return $x_i(t)$, $\dot{x}_i(t)$, x_i^e .
14. Terminate the algorithm.

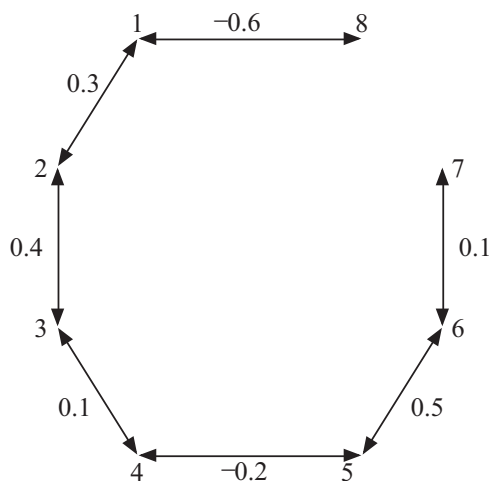


Fig. 3. Topology of Example 1 with eight agents.

4. Implementations of bipartite consensus

Example 1. In this example, the multi-agent system contains eight agents shown in Fig. 3. Each agent can represent a robot that moves on the plane. $x_i = [x_{i1}, x_{i2}]^T$ is the position of agent i . The adjacency matrix \mathcal{A}_1 is given as follows:

$$\mathcal{A}_1 = \begin{bmatrix} 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & -0.6 \\ 0.3 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & -0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ -0.6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

The dynamics of the multi-agent system are described by the

Table 1

Coefficients of the unknown dynamics in Example 1.

i	1	2	3	4	5	6	7	8
p_{i1}	0.4	-0.65	8	1	-10	1.5	0.5	-1
p_{i2}	0.5	0.45	-6	11	11	9	2	5

Table 2

Coefficients of time-delay terms in Example 1.

i	1	2	3	4	5	6	7	8
s_{i1}	0.9	1.2	-1.1	-0.7	0.6	0.3	0.2	0.6
s_{i2}	1.2	0.8	0.6	0.3	0.8	0.4	-0.1	-0.4

Table 3

Time delays of eight agents in Example 1.

i	1	2	3	4	5	6	7	8
τ_i	0.1	0.05	0.15	0.08	0.18	0.1	0.11	0.02

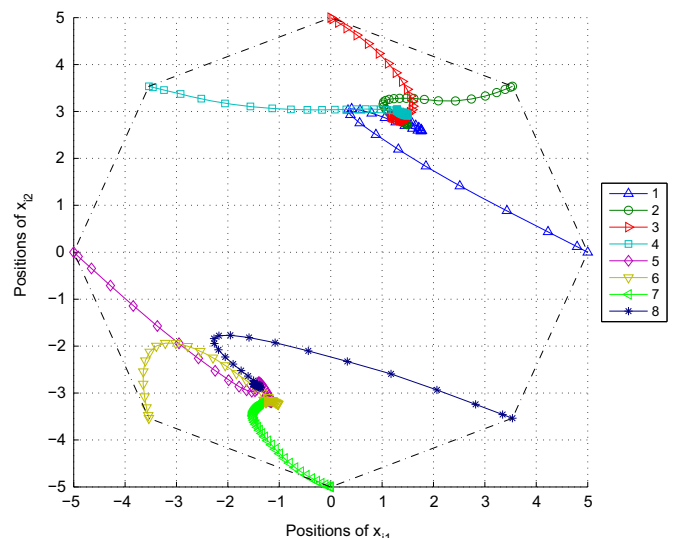


Fig. 4. Trajectories of eight agents on the plane.

following equations:

$$\begin{bmatrix} \ddot{x}_{i1}(t) \\ \ddot{x}_{i2}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_{i2}(t) \sin(p_{i1}x_{i1}(t)) \\ \dot{x}_{i1}(t) \cos(p_{i2}x_{i2}(t)) \end{bmatrix} + u_i + \begin{bmatrix} s_{i1}\dot{x}_{i1}(t-\tau_i) \cos(\dot{x}_{i2}(t-\tau_i)) \\ s_{i2}\dot{x}_{i2}(t-\tau_i) \sin(\dot{x}_{i1}(t-\tau_i)) \end{bmatrix},$$

$$i = 1, 2, \dots, 8, \quad (25)$$

where p_{i1} , p_{i2} , s_{i1} and s_{i2} are the corresponding positive coefficients given in Tables 1 and 2. We choose $\phi_i(\dot{x}_i) = \sqrt{(s_{i1}\dot{x}_{i1})^2 + (s_{i2}\dot{x}_{i2})^2}$. τ_i , $i = 1, 2, \dots, 8$, are time delays shown in Table 3. We suppose that the initial states of the multi-agent system are on a circle with radius 5 and the initial velocities are zeros. All the eight agents have the same parameters, $\tau_{\max} = 0.2$, $k_{i0} = 50$, $k_i = 30$, $\omega_i = 50$, $W_i^{\max} = 100$, $a_i = 100$ and $\theta_{N_i} = 0.01$. The number of neurons for each RBFNN is 16 and $\delta_i^2 = 2$. μ_i , $i = 1, 2, \dots, 8$, are distributed uniformly in the range $[-5, 5] \times [-5, 5]$. Our control objective is to

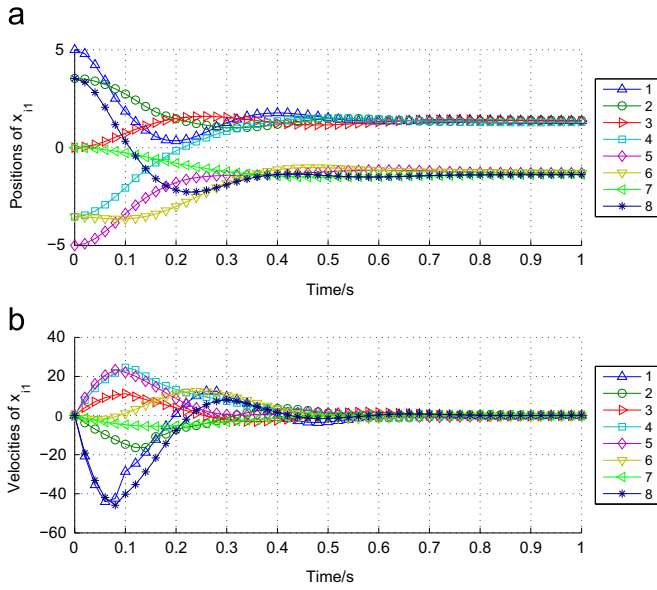


Fig. 5. Trajectories of positions and velocities in first dimension for eight agents: (a) trajectories of positions in first dimension, (b) trajectories of velocities in first dimension.

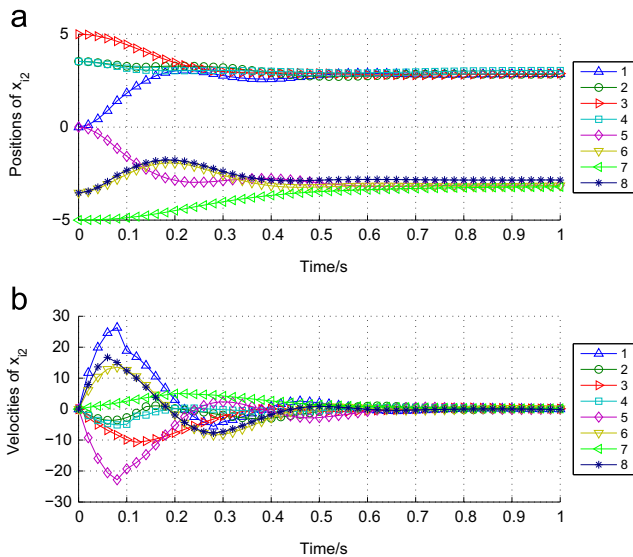


Fig. 6. Trajectories of positions and velocities in second dimension for eight agents: (a) trajectories of positions in second dimension, (b) trajectories of velocities in second dimension.

drive the eight agents on the plane to the bipartite consensus state. Fig. 4 illustrates that the eight agents can reach bipartite consensus with the distributed control algorithm in Algorithm 1. Additionally, Figs. 5 and 6 show the trajectories of positions and

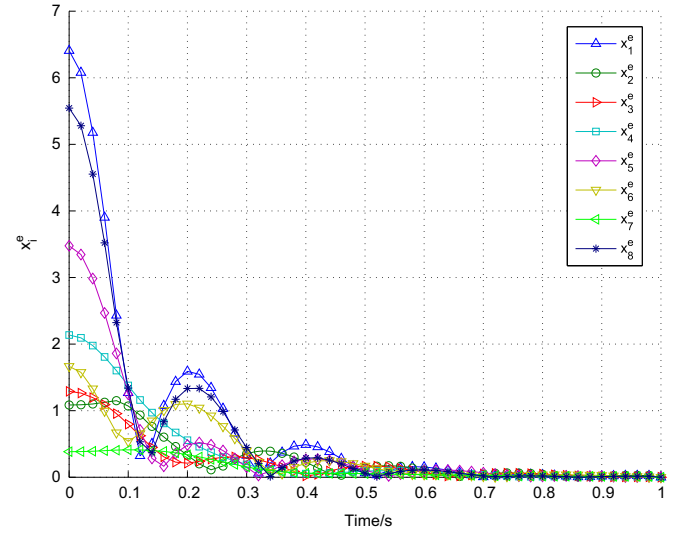


Fig. 7. Error trajectories of bipartite consensus with eight agents.

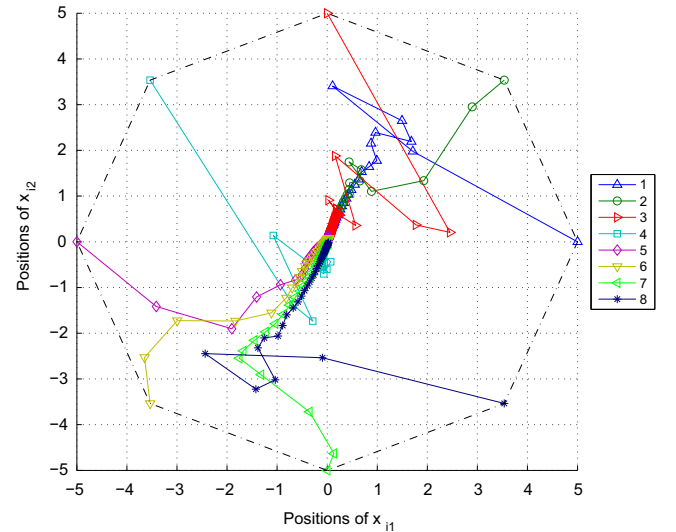


Fig. 8. Trajectories of eight agents with structurally unbalanced graph.

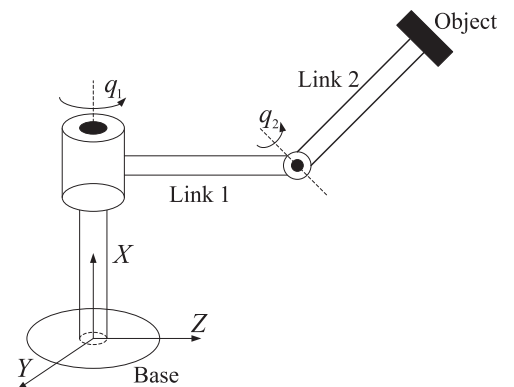


Fig. 9. Two-link revolve manipulator.

velocities in two dimensions. In Fig. 5(b), the velocities eventually become zeros and the eight agents are divided into two different groups with opposite signs. In order to describe whether bipartite consensus has been achieved, we define the measurement of error of bipartite consensus for each agent:

$$x_i^e = \|y_{ei}\|, \quad i = 1, 2, \dots, 8. \quad (26)$$

In Fig. 7, it shows that the error of each agent can gradually approach zero. Therefore, bipartite consensus for multi-agent system (25) can be achieved. In Fig. 8, we just change the topology of multi-agent system (25) and the adjacency matrix is modified as

$$\mathcal{A}_1^u = \begin{bmatrix} 0 & 0.3 & 0 & 0 & 0 & 0 & 0 & -0.6 \\ 0.3 & 0 & 0.4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & -0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.1 & 0 & -0.2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.2 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0.2 \\ -0.6 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0 \end{bmatrix}.$$

It is obvious that the corresponding communication topology of \mathcal{A}_1^u is structurally unbalanced. Fig. 8 shows that bipartite consensus cannot be reached and this in turn demonstrates the importance of structural balance for bipartite consensus problems.

Example 2. We utilize a multi-manipulator system [43] to verify the validity of the distributed control algorithm (16) in Section 3. The profile of the two-link manipulator is shown in Fig. 9 and each manipulator holds a component which is used to assemble the industrial product. The concept of bipartite consensus can be applied to the tasks which need to assemble the product in a symmetrical way. The dynamics of the multi-manipulator system

Table 4
Parameters of each manipulator in Example 2.

\tilde{g}	l_{i1}	l_{i2}	m_{i1}	m_{i2}
9.8 m/s ²	1.5 m	1 m	2 kg	1 kg

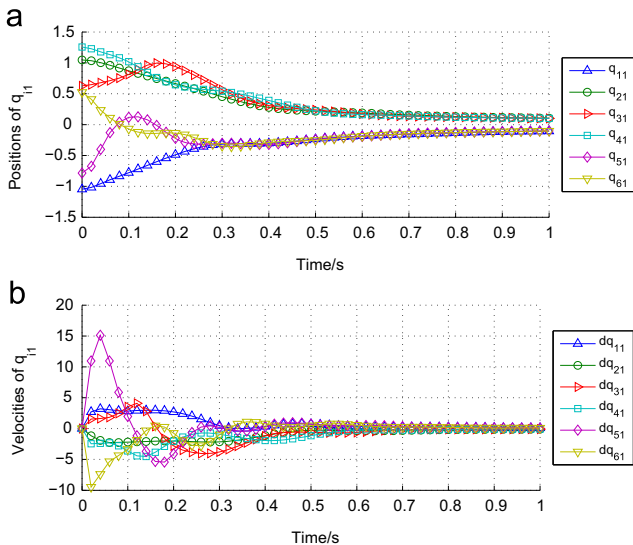


Fig. 10. Positions and velocities of link 1 with six manipulators: (a) trajectories of positions of link 1, (b) trajectories of velocities of link 1.

are described as follows:

$$M_i(q_i)\ddot{q}_i + V_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i) + g_i(\dot{q}_i(t - \tau_i)) = \Gamma_i, \quad i = 1, 2, \dots, 6, \quad (27)$$

where $q_i = [q_{i1}, q_{i2}]^T \in \mathbb{R}^2$, \dot{q}_i and \ddot{q}_i are the position, velocity and acceleration vector of the i th manipulator, respectively. $M_i(q_i) \in \mathbb{R}^{2 \times 2}$ is the inertia matrix of manipulator i , $V_i(q_i, \dot{q}_i) \in \mathbb{R}^{2 \times 2}$ is the centripetal-Coriolis matrix of manipulator i , $G_i(q_i) \in \mathbb{R}^2$ is the gravitational vector of manipulator i and $\Gamma_i \in \mathbb{R}^2$ is the torque vector of manipulator i . We give the detailed parameters of each manipulator as follows:

$$V_i(q_i, \dot{q}_i) = \begin{bmatrix} V_{i11} & V_{i12} \\ V_{i21} & V_{i22} \end{bmatrix},$$

$$G_i(q_i) = [G_{i1} \ G_{i2}],$$

$$M_i = I,$$

$$V_{i11} = -m_{i2}l_{i1}l_{i2} \sin(q_{i2})\dot{q}_{i2},$$

$$V_{i12} = -m_{i2}l_{i1}l_{i2} \sin(q_{i2})\dot{q}_{i2} - m_{i2}l_{i1}l_{i2} \sin(q_{i2})\dot{q}_{i1},$$

$$V_{i21} = m_{i2}l_{i1}l_{i2} \sin(q_{i2})\dot{q}_{i1},$$

$$V_{i22} = 0,$$

$$G_{i1} = (m_{i1} + m_{i2})\tilde{g}l_{i1} \sin(q_{i1}) + m_{i2}\tilde{g}l_{i2} \sin(q_{i1} + q_{i2}),$$

$$G_{i2} = m_{i2}\tilde{g}l_{i2} \sin(q_{i1} + q_{i2}).$$

\tilde{g} , l_{i1} , l_{i2} , m_{i1} and m_{i2} are given in Table 4. For simplicity, we set $M_i = I$. $g_i(\dot{q}_i(t - \tau_i))$ represents the frictional force vector where

$$g_i(\dot{q}_i(t - \tau_i)) = \begin{bmatrix} s_{i1}\dot{q}_{i1}(t - \tau_i) \cos(\dot{q}_{i2}(t - \tau_i)) \\ s_{i2}\dot{q}_{i2}(t - \tau_i) \sin(\dot{q}_{i1}(t - \tau_i)) \end{bmatrix}. \quad (28)$$

We set the same parameters for all the six manipulators, $k_{i0} = 15$, $\omega_i = 30$, $\sigma_i^2 = 1.6$ and the number of neurons for each RBFNN is 16. μ_i , $i = 1, 2, \dots, 6$, are distributed uniformly in the range $[-3, 3] \times [-3, 3]$. The coefficients of time-delay terms and time delays are chosen from numbers 1 to 6 in Tables 2 and 3. The

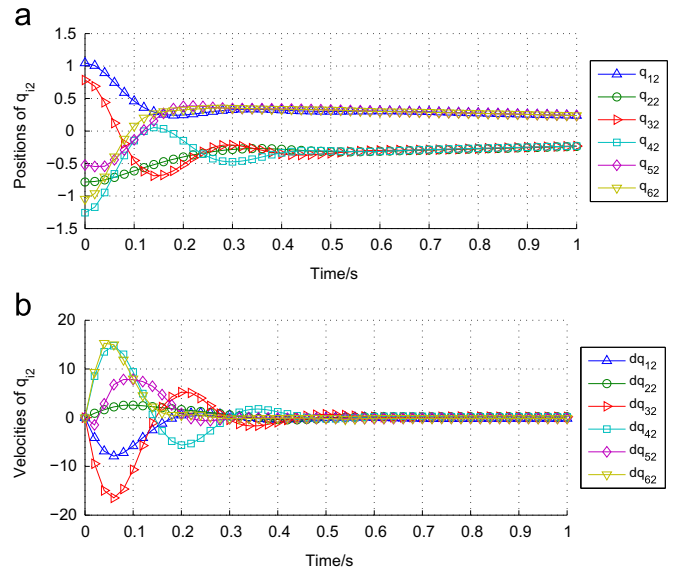


Fig. 11. Positions and velocities of link 2 with six manipulators: (a) trajectories of positions of link 2, (b) trajectories of velocities of link 2.

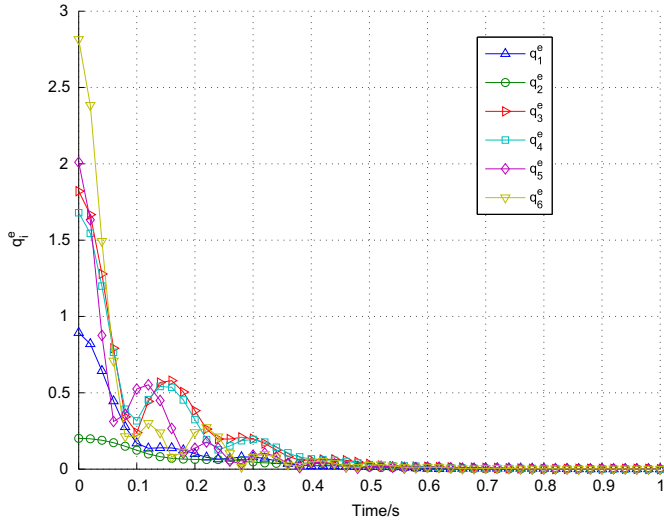


Fig. 12. Error trajectories of bipartite consensus for six manipulators.

initial states of the multi-manipulator system are

$$\begin{bmatrix} q_1^T(0) & q_1^T(0) \\ q_2^T(0) & q_2^T(0) \\ q_3^T(0) & q_3^T(0) \\ q_4^T(0) & q_4^T(0) \\ q_5^T(0) & q_5^T(0) \\ q_6^T(0) & q_6^T(0) \end{bmatrix} = \begin{bmatrix} [-\pi/3, \pi/3] & [0, 0] \\ [\pi/3, -\pi/4] & [0, 0] \\ [\pi/5, \pi/4] & [0, 0] \\ [2\pi/5, -2\pi/5] & [0, 0] \\ [-\pi/4, -\pi/6] & [0, 0] \\ [\pi/6, -\pi/3] & [0, 0] \end{bmatrix}.$$

The adjacency matrix A_2 is given as follows:

$$A_2 = \begin{bmatrix} 0 & -0.5 & 0 & 0 & 0 & 0.3 \\ -0.5 & 0 & 0.2 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.7 & 0 & 0 \\ 0 & 0 & 0.7 & 0 & -0.1 & 0 \\ 0 & 0 & 0 & -0.1 & 0 & 1.5 \\ 0.3 & 0 & 0 & 0 & 1.5 & 0 \end{bmatrix}.$$

Other parameters are the same in Example 1.

From Figs. 10 and 11, we can infer that the multi-manipulator system can reach the opposite positions in a symmetrical manner, where dq_{i1} and dq_{i2} represent the velocities of Link 1 and Link 2, respectively. In Fig. 12, all the errors of bipartite consensus q_i^e , $i = 1, 2, \dots, 6$, approach zeros, which are similar to the definition of x_i^e in Example 1. This further demonstrates the effectiveness of the developed control algorithm in physical applications.

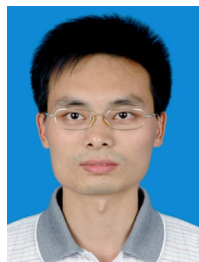
5. Conclusion

Bipartite consensus of the unknown nonlinear time-delayed multi-agent systems is investigated in this paper. RBFNNs are utilized to learn the unknown dynamics online and a Lyapunov–Krasovskii functional is introduced to deal with time delays. However, in order to avoid singularities in the distributed control algorithm, a σ -function is applied to the time-delay elimination part of the algorithm. If the communication topology is structurally balanced, then the distributed control algorithm can make the multi-agent systems achieve bipartite consensus, which can be applied to a multi-manipulator system. In the future, to adapt to more complex environment, we will concentrate on the cases where external noises are added to the control input and connectivity is preserved.

References

- [1] H. Zhang, T. Feng, G.-H. Yang, H. Liang, Distributed cooperative optimal control for multiagent systems on directed graphs: an inverse optimal approach, *IEEE Trans. Cybern.* 45 (7) (2015) 1315–1326.
- [2] H. Zhang, J. Zhang, G.-H. Yang, Y. Luo, Leader-based optimal coordination control for the consensus problem of multiagent differential games via fuzzy adaptive dynamic programming, *IEEE Trans. Fuzzy Syst.* 23 (1) (2015) 152–163.
- [3] D. Liu, D. Wang, H. Li, Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach, *IEEE Trans. Fuzzy Syst.* 25 (2) (2014) 418–428.
- [4] D. Wang, D. Liu, H. Li, H. Ma, C. Li, A neural-network-based online optimal control approach for nonlinear robust decentralized stabilization, *Soft Comput.* <http://dx.doi.org/10.1007/s00500-014-1534-z>.
- [5] Z. Zhang, D. Zhao, Clique-based cooperative multiagent reinforcement learning using factor graphs, *IEEE/CAA J. Autom. Sinica* 1 (3) (2014) 248–256.
- [6] T. Liu, Z.-P. Jiang, Distributed output-feedback control of nonlinear multi-agent systems, *IEEE Trans. Autom. Control* 58 (11) (2013) 2912–2917.
- [7] H. Ma, D. Liu, D. Wang, H. Li, Output-based high-order bipartite consensus under directed antagonistic networks, in: *Proceedings of the 2014 IEEE Symposium on Intelligent Agents*, Orlando, FL, USA, 2014, pp. 16–21.
- [8] H. Liang, H. Zhang, Z. Wang, J. Wang, Consensus robust output regulation of discrete-time linear multi-agent systems, *IEEE/CAA J. Autom. Sinica* 1 (2) (2014) 204–209.
- [9] T. Liu, Z.-P. Jiang, Distributed control of nonlinear uncertain systems: a cyclic-small-gain approach, *IEEE/CAA J. Autom. Sinica* 1 (1) (2014) 46–53.
- [10] H. Liu, L. Cheng, M. Tan, Z.-G. Hou, Containment control of continuous-time linear multi-agent systems with aperiodic sampling, *Automatica* 51 (7) (2015) 78–84.
- [11] J. Hu, J. Yu, J. Cao, Distributed containment control for nonlinear multi-agent systems with time-delayed protocol, *Asian J. Control.* <http://dx.doi.org/10.1002/asjc.1131>.
- [12] T. Liu, Z.-P. Jiang, Distributed formation control of nonholonomic mobile robots without global position measurements, *Automatica* 49 (2) (2013) 592–600.
- [13] J. Hu, Z. Xiao, Y. Zhou, J. Yu, Formation control over antagonistic networks, in: *Proceedings of the Chinese Control Conference*, Xi'an, China, 2013, pp. 6879–6884.
- [14] D. Shen, Z. Sun, W. Sun, Leader–follower formation control without leader's velocity information, *Sci. China Inf. Sci.* 57 (9) (2014) 1–12.
- [15] H. Ma, D. Wang, D. Liu, C. Li, Continuous-time group consensus using distributed event-triggered control, in: *Proceedings of the 7th International Conference on Advanced Computational Intelligence*, Wuyi, China, 2015, pp. 174–178.
- [16] H. Ma, D. Liu, D. Wang, F. Tan, C. Li, Centralized and decentralized event-triggered control for group consensus with fixed topology in continuous time, *Neurocomputing* 161 (2015) 267–276.
- [17] Z.-G. Hou, L. Cheng, M. Tan, Decentralized robust adaptive control for the multiagent system consensus problem using neural networks, *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* 39 (3) (2009) 636–647.
- [18] L. Cheng, Z.-G. Hou, M. Tan, Y. Lin, W. Zhang, Neural-network-based adaptive leader-following control for multiagent systems with uncertainties, *IEEE Trans. Neural Netw.* 21 (8) (2010) 1351–1358.
- [19] A. Das, F.L. Lewis, Distributed adaptive control for synchronization of unknown nonlinear networked systems, *Automatica* 46 (12) (2010) 2014–2021.
- [20] L. Cheng, Z.-G. Hou, Y. Lin, M. Tan, W. Zhang, Solving a modified consensus problem of linear multi-agent systems, *Automatica* 47 (10) (2011) 2218–2223.
- [21] W. Chen, X. Li, W. Ren, C. Wen, Adaptive consensus of multi-agent systems with unknown identical control directions based on a novel Nussbaum-type function, *IEEE Trans. Autom. Control* 59 (7) (2014) 1887–1892.
- [22] A. Hu, J. Cao, M. Hu, L. Guo, Distributed control of cluster synchronisation in networks with randomly occurring non-linearities, *Int. J. Syst. Sci.* <http://dx.doi.org/10.1080/00207172.2014.1002026>.
- [23] L. Cheng, H. Wang, Z.-G. Hou, M. Tan, Reaching a consensus in networks of high-order integral agents under switching directed topologies, *Int. J. Syst. Sci.* <http://dx.doi.org/10.1080/00207172.2014.966281>.
- [24] Y. Wang, L. Cheng, W. Ren, Z.-G. Hou, M. Tan, Seeking consensus in networks of linear agents: communication noises and Markovian switching topologies, *IEEE Trans. Autom. Control* 60 (5) (2015) 1374–1379.
- [25] H. Liang, H. Zhang, Z. Wang, J. Wang, Cooperative robust output regulation for heterogeneous second-order discrete-time multi-agent systems, *Neurocomputing* 162 (2015) 41–47.
- [26] C. Song, J. Cao, Y. Liu, Robust consensus of fractional-order multi-agent systems with positive real uncertainty via second-order neighbors information, *Neurocomputing* 165 (2015) 293–299.
- [27] C. Xu, Y. Zheng, H. Su, Z. Hongbin, Containment for linear multi-agent systems with exogenous disturbances, *Neurocomputing* 160 (2015) 206–212.
- [28] W. Chen, S. Hua, H. Zhang, Consensus-based distributed cooperative learning from closed-loop neural control systems, *IEEE Trans. Fuzzy Syst.* 26 (2) (2015) 331–345.
- [29] Y. Wan, G. Wen, J. Cao, W. Yu, Distributed node-to-node consensus of multi-agent systems with stochastic sampling, *Int. J. Robust Nonlinear Control.* <http://dx.doi.org/10.1002/rnc.3302>.

- [30] Q. Wang, Y. Wang, Cluster synchronization of a class of multi-agent systems with a bipartite graph topology, *Sci. China Inf. Sci.* 57 (1) (2014) 1–11.
- [31] A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Trans. Autom. Control* 48 (6) (2003) 988–1001.
- [32] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, *IEEE Trans. Autom. Control* 49 (9) (2004) 1520–1533.
- [33] W. Ren, R. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, *IEEE Trans. Autom. Control* 50 (5) (2005) 655–661.
- [34] Y. Cao, W. Yu, W. Ren, G. Chen, An overview of recent progress in the study of distributed multi-agent coordination, *IEEE Trans. Ind. Inform.* 9 (1) (2013) 427–438.
- [35] R. Olfati-Saber, J.A. Fax, R.M. Murray, Consensus and cooperation in networked multi-agent systems, *Proc. IEEE* 95 (1) (2007) 215–233.
- [36] C. Altafini, Consensus problems on networks with antagonistic interactions, *IEEE Trans. Autom. Control* 58 (4) (2013) 935–946.
- [37] J. Hu, W. Zheng, Bipartite consensus for multi-agent systems on directed signed networks, in: *Proceedings of the IEEE Conference on Decision and Control*, Florence, Italy, 2013, pp. 3451–3456.
- [38] M.E. Valcher, P. Misra, On the consensus and bipartite consensus in high-order multi-agent dynamical systems with antagonistic interactions, *Syst. Control Lett.* 66 (2014) 94–103.
- [39] S.S. Ge, C. Wang, Adaptive neural control of uncertain MIMO nonlinear systems, *IEEE Trans. Neural Netw.* 15 (3) (2004) 674–692.
- [40] M.M. Polycarpou, Stable adaptive neural control scheme for nonlinear systems, *IEEE Trans. Autom. Control* 41 (3) (1996) 447–451.
- [41] S. Ge, F. Hong, T. Lee, Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients, *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* 34 (1) (2004) 499–516.
- [42] C.L.P. Chen, G.-X. Wen, Y.-J. Liu, F.-Y. Wang, Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks, *IEEE Trans. Fuzzy Syst.* 25 (6) (2014) 1217–1226.
- [43] L. Cheng, Z.-G. Hou, M. Tan, Decentralized adaptive consensus control for multi-manipulator system with uncertain dynamics, in: *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, Singapore, 2008, pp. 2712–2717.



Ding Wang received the B.S. degree in mathematics from Zhengzhou University of Light Industry, Zhengzhou, China, the M.S. degree in operations research and cybernetics from Northeastern University, Shenyang, China, and the Ph.D. degree in control theory and control engineering from Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2007, 2009, and 2012, respectively. He is currently an associate professor with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences. His research interests include neural networks, learning systems, and complex systems and intelligent control.



Hongwen Ma received the B.S. degree in electric engineering and automation from Nanjing University of Science and Technology in 2012. He is currently working toward the Ph.D. degree in The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is also with the University of Chinese Academy of Sciences, Beijing. He is the Merit Student of the University of Chinese Academy of Sciences. He has won IEEE Student Travel Grants and IEEE CIS Graduate Student Research Grants. His research interests include neural networks, distributed algorithm, networked control systems, and multi-agent systems.



Derong Liu received the Ph.D. degree in electrical engineering from the University of Notre Dame in 1994. He was a Staff Fellow with General Motors Research and Development Center, from 1993 to 1995. He was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, from 1995 to 1999. He joined the University of Illinois at Chicago in 1999, and became a Full Professor of Electrical and Computer Engineering and of Computer Science in 2006. He was selected for the “100 Talents Program” by the Chinese Academy of Sciences in 2008, and he served as the Associate Director of The State Key Laboratory of Management and Control for Complex Systems at the Institute of Automation, from 2010 to 2015. He is now a Full Professor with the School of Automation and Electrical Engineering, University of Science and Technology Beijing. He has published 15 books (six research monographs and nine edited volumes). He is an elected AdCom member of the IEEE Computational Intelligence Society and he is the Editor-in-Chief of the *IEEE Transactions on Fuzzy Systems*. He was the General Chair of 2014 IEEE World Congress on Computational Intelligence and is the General Chair of 2016 World Congress on Intelligent Control and Automation. He received the Faculty Early Career Development Award from the National Science Foundation in 1999, the University Scholar Award from University of Illinois from 2006 to 2009, the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China in 2008, and the Outstanding Achievement Award from Asia Pacific Neural Network Assembly in 2014. He is a Fellow of the IEEE and a Fellow of the International Neural Network Society.