Neurocomputing 174 (2016) 928–936

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

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.

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).

http://dx.doi.org/10.1016/j.neucom.2015.10.013
0925-2312/

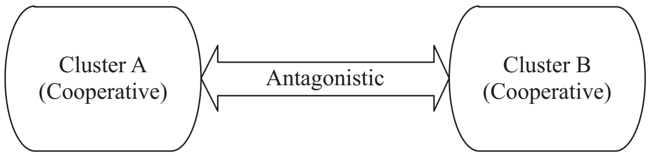

Fig. 1. Two clusters with cooperative behaviors inside and antagonistic behaviors between each other.

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: $(\cdot)^{\mathsf{T}}$ denotes the transpose of a given matrix. $(\cdot)$ 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. $\mathrm{diag}(\cdot)$ 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 $\mathcal{A}=(\mathcal{A}_{ij})\in\mathbb{R}^{N\times N}$ is the adjacency matrix of $\mathcal{G}$. Denote $\mathcal{A}_{ij}$ as the element of the ith row and jth column of the matrix $\mathcal{A}$. The ith node in a signed graph $\mathcal{G}$ represents the ith 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 $\mathcal{A}_{ji}\neq 0\Leftrightarrow(i,j)\in\mathcal{E}$. The interaction between the ith and the jth agent is cooperative if $\mathcal{A}_{ij}>0$. It is competitive if $\mathcal{A}_{ij}<0$ and there is no interaction if $\mathcal{A}_{ij}=0$. Note that self-loops will not be considered in this paper, i.e., $\mathcal{A}_{ii}=0,\ i=1,2,\dots,N$. The Laplacian matrix of the signed graph is given as follows:

$$\mathcal{L}_{ij}=\begin{cases}\sum\limits_{k\in\mathcal{N}_i}|\mathcal{A}_{ik}| & \text{if } i=j;\\ -\mathcal{A}_{ij} & \text{if } i\neq j.\end{cases} \tag{1}$$

The following two definitions are important concepts in this paper.

Definition 1 (*Structurally balanced, cf. Altafini [36]*). In this paper, a signed graph $\mathcal{G}(A)$ is said to be structurally balanced if it contains a bipartition of the sets of nodes $\mathcal{V}_1$ and $\mathcal{V}_2$, where $\mathcal{V}=\mathcal{V}_1\cup\mathcal{V}_2,\mathcal{V}_1\cap\mathcal{V}_2=\varnothing$ such that $\mathcal{A}_{ij}\geq 0,\forall i,j\in\mathcal{V}_p(p\in\{1,2\});\mathcal{A}_{ij}\leq 0$, $\forall i\in\mathcal{V}_p,j\in\mathcal{V}_q,p\neq q(p,q\in\{1,2\})$. 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:

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

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 $h(x)=[h_1(x),h_2(x),\dots,h_m(x)]^{\mathsf{T}}:\mathbb{R}^m\to\mathbb{R}^m$ can be approximated by RBFNNs:

$$h(x)=W^{\mathsf{T}}\Phi(x), \tag{3}$$

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

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

where $\mu_i=[\mu_{i1},\mu_{i2},\dots,\mu_{im}]^{\mathsf{T}}$ 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

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

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

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

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

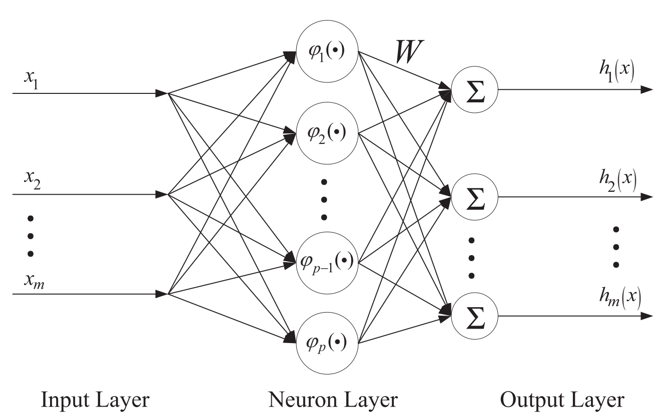

Fig. 2. Structure of the radial basis function neural networks.