

# Connectivity preserved nonlinear time-delayed multiagent systems using neural networks and event-based mechanism

Hongwen Ma<sup>1,2</sup> · Ding Wang<sup>1</sup>

Received: 24 June 2016 / Accepted: 16 September 2016  
© The Natural Computing Applications Forum 2016

**Abstract** This paper studies how to preserve connectivity for nonlinear time-delayed multiagent systems using event-based mechanism. By using the idea of divide-and-conquer, we divide the distributed controller into five parts to deal with different requirements of the time-delayed multiagent systems, such as eliminating the negative effects of time delays, preserving connectivity, learning the unknown dynamics and achieving consensus. To reduce the communication times among the agents, a centralized event-based protocol is introduced and an event-triggered function is devised to control the frequency of the communication without Zeno behavior. The technique of  $\sigma$ -functions is used to exclude the singularity of the established distributed controller. In the simulation example, the results demonstrate the validity of our developed methodology.

**Keywords** Connectivity preservation · Event-based control · Multiagent systems · Neural networks

## 1 Introduction

Multiagent systems have been a hot topic in the last few years [1, 7, 8, 10, 15, 16, 18, 21, 26, 28, 32–35, 37, 39, 41]. From a biological view, Reynolds proposed a distributed behavior model of flocks, herds and schools [25]. Then, Vicsek et al. [31] investigated a phase transition model of self-driven particles which is the derivation of nearest neighbor rules. Then, Jadbabaie et al. introduced the idea of nearest neighbor rules into the multiagent systems [13]. Recently, Werfel et al. had put this distributed algorithm into reality [38]. They established a multiagent construction system to automatically generate low-level rules based on limited neighboring information and to achieve specific designed goals. For more details, please refer to [2] and the references therein.

In physical implementations, the communication capability is constrained by the power of multiagent systems and the distance between any two agents. When the distance between two agents becomes too far, they will lose contact with each other [5, 40]. In [6], a leader-following rendezvous control protocol was proposed to maintain connectivity, and the position measurement was the only information accessible to the distributed controller. In [29], a distributed protocol for the double-integrator multiagent systems with connectivity preservation was proposed. Thus, preserving connectivity while reaching consensus is of great importance.

As the multiagent systems become complex, the dynamics are nonlinear and the exact models are hard to be obtained. Neural networks are powerful tools for learning the unknown dynamics of multiagent systems. In [12], a robust adaptive control was proposed to achieve consensus using neural networks. Subsequently, in [4], a neural-network-based leader-following control was

---

✉ Ding Wang  
ding.wang@ia.ac.cn

Hongwen Ma  
mahongwen2012@ia.ac.cn

<sup>1</sup> The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

<sup>2</sup> University of Chinese Academy of Sciences, Beijing 100190, China

proposed with external disturbances. However, due to the existence of data loss and channel congestion during the data transmission from sensors to CPUs, time delay is unavoidable which may deteriorate the performance of multiagent systems. In [3], an adaptive consensus control was proposed to deal with the nonlinear time-delayed multiagent systems by using neural networks. Then, in [22], external noises were taken into consideration and Lyapunov–Krasovskii functionals were introduced to solve the time-delay problems. Therefore, investigating nonlinear time-delayed multiagent systems by utilizing neural networks is meaningful for improving the autonomy and intelligence of the systems among the control community.

Reducing communication times between each pair of agents will increase the reliability of multiagent systems in an economical way and decrease power consumptions of on-chip systems, especially in wireless power-limited systems such as unmanned aerial vehicles (UAVs), autonomous underwater vehicles (AUVs) and attitude alignments of clusters of satellites. Event-based technique is a promising way to solve this problem [11, 17, 23, 27, 30]. In [20, 36], both centralized and decentralized event-triggered control were developed to achieve group consensus and the parameters of the event-triggered function were highly reduced. In [14], only partial states were available to the controller and an output feedback control was proposed based on event-triggered technique. In [42], taking input time delay into consideration, event-based leader-following consensus was reached. Based on all the papers mentioned above, studying how to combine the event-based technique with the distributed control algorithm for nonlinear time-delayed multiagent systems with connectivity preservation is of practical significance, and this motivates our investigation.

In [19], a neural-network-based distributed control algorithm for nonlinear time-delayed multiagent systems was established to guarantee the connectivity. We extend this problem to event-based case, and according to the idea of divide-and-conquer, the distributed controller is divided into five parts. The main contributions of this paper are listed as follows.

1. A Lyapunov–Krasovskii functional is borrowed from [3, 9] to eliminate the negative effects of time delays. Moreover, a  $\sigma$ -function is developed to avoid the singularity in the distributed controller.
2. A centralized event-based protocol is introduced, and an event-triggered function is designed to control the frequency of the communication. Furthermore, the Zeno behavior can be excluded.
3. Radial basis function neural networks (RBFNNs) are utilized to learn the unknown dynamics of multiagent

systems which can improve the intelligence and autonomy of multiagent systems.

The rest of this paper is organized as follows. In Sect. 2, fundamental preliminaries and problem statement are provided. In Sect. 3, an event-triggered function is devised and an event-based distributed control protocol is developed which guarantees the achievement of consensus. A simulation example and the concluding remarks are given in Sects. 4 and 5, respectively.

**Notations**  $(\cdot)^{\top}$  represents the transpose of a matrix.  $\text{tr}(\cdot)$  is the trace of a given matrix, and  $\|\cdot\|$  is the Frobenius norm or Euclidian norm.  $t^{-}$  denotes the time just before  $t$ .

## 2 Preliminaries

### 2.1 Graph theory

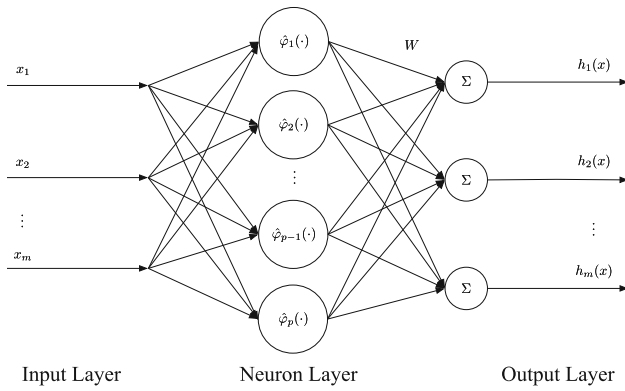
A triplet  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$  is called a weighted graph if  $\mathcal{V} = \{1, 2, \dots, N\}$  is the set of  $N$  nodes,  $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$  is the set of edges and  $\mathcal{A} = (\mathcal{A}_{ij}) \in \mathbb{R}^{N \times N}$  is the  $N \times N$  matrix of the weights of  $\mathcal{G}$ . Here, we denote  $\mathcal{A}_{ij}$  as the element of the  $i$ th row and  $j$ th column of matrix  $\mathcal{A}$ . The  $i$ th node in 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  $j$  can directly obtain the information from agent  $i$ . In addition,  $\mathcal{A}$  is called the adjacency matrix of graph  $\mathcal{G}$  and we use the notation  $\mathcal{G}(\mathcal{A}) : \mathcal{A}_{ij} \neq 0 \Leftrightarrow (j, i) \in \mathcal{E}$  to represent the graph  $\mathcal{G}$  corresponding to  $\mathcal{A}$ . We will focus on agent  $i$  if there is no confusion. Let

$$\mathcal{D} = \begin{bmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_N \end{bmatrix}$$

be an  $N \times N$  diagonal matrix where  $d_i = \sum_{j \in \mathcal{N}_i} \mathcal{A}_{ij}$  and  $\mathcal{N}_i = \{j \in \mathcal{V} | (j, i) \in \mathcal{E}\}$  is the set of neighbor nodes of node  $i$ . Then,  $\mathcal{D}$  is termed as the indegree matrix of  $\mathcal{G}$  and the Laplacian matrix is  $\mathcal{L} = \mathcal{D} - \mathcal{A}$  corresponding to  $\mathcal{G}$ . In addition, for a connected graph,  $\mathcal{L}$  has only one single zero eigenvalue [24].

### 2.2 Radial basis function neural network

In practice, we usually use neural networks as function approximators to model unknown functions. RBFNNs are potential candidates for learning the unknown dynamics of the multiagent systems in virtue of “linear-in-weight” property. In Fig. 1,  $h(x) = [h_1(x), h_2(x), \dots, h_m(x)]^{\top} : \mathbb{R}^m \rightarrow \mathbb{R}^m$  is a continuous unknown nonlinear function which can be approximated by an RBFNN:



**Fig. 1** Structure of the radial basis function neural network (RBFNN)

$$h(x) = W^T \Phi(x), \tag{1}$$

where  $x \in \Upsilon_x \subset \mathbb{R}^m$  is the input vector,  $W \in \mathbb{R}^{p \times m}$  is the weight matrix and  $p$  represents the number of neurons. Additionally,  $\Phi(x) = [\hat{\phi}_1(x), \hat{\phi}_2(x), \dots, \hat{\phi}_p(x)]^T$  is the activation function and

$$\hat{\phi}_i(x) = \exp \left[ \frac{-(x - \hat{\mu}_i)^T (x - \hat{\mu}_i)}{\hat{\sigma}_i^2} \right], \quad i = 1, 2, \dots, p, \tag{2}$$

where  $\hat{\sigma}_i$  is the width of the Gaussian function  $\hat{\phi}_i(x)$  and  $\hat{\mu}_i = [\hat{\mu}_{i1}, \hat{\mu}_{i2}, \dots, \hat{\mu}_{im}]^T$  is the center of the receptive field. RBFNNs can approximate any continuous function over a compact set  $\Upsilon_x \subset \mathbb{R}^m$  with arbitrary precision. Therefore, for a given positive constant  $\theta_N$ , there exists an ideal weight matrix  $W^*$  such that

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

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

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

$$\hat{h}(x) = \hat{W}^T \Phi(x), \tag{4}$$

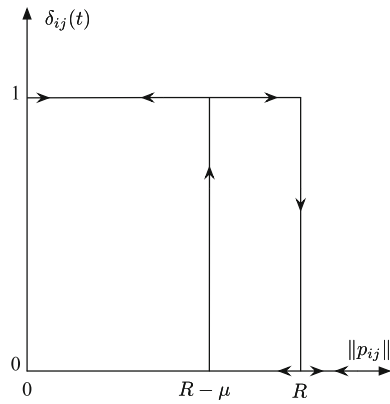
where  $\hat{W}$  can be updated online. The online updating laws will be given in Sect. 3.

### 2.3 Problem statement

The second-order nonlinear time-delayed multiagent system is modeled as follows:

$$\begin{aligned} \dot{p}_i &= v_i, \\ \dot{v}_i &= u_i + f_i(p_i(t), v_i(t)) + g_i(v_i(t - \tau_i)), \quad i = 1, 2, \dots, N, \end{aligned} \tag{5}$$

where  $i$  represents agent  $i$ ,  $p_i \in \mathbb{R}^2$  is the position,  $v_i \in \mathbb{R}^2$  is the velocity,  $\tau_i$  is the unknown time delay,  $f_i(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  and  $g_i(\cdot) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  are continuous but unknown nonlinear vector functions and  $u_i(\cdot) \in \mathbb{R}^2$  is the control input.



**Fig. 2** Indicator function  $\delta_{ij}(t)$

We assume that all the agents have a common sensing radius  $R$ . Due to the existence of measurement noises, we cannot detect the critical boundary precisely when an agent is losing connectivity. In Fig. 2,  $\mu \in (0, R)$  is the given constant which can generate the effect of hysteresis and  $\mu_0 \leq \mu$  is utilized to generate the initial set of edges  $\mathcal{E}(0)$ . The properties of the time-varying set of edges  $\mathcal{E}(t) = \{(i, j) | i, j \in \mathcal{V}\}$  can be described as follows [29]:

- The initial set of edges can be generated by  $\mathcal{E}(0) = \{(i, j) | i, j \in \mathcal{V}, \|p_i(0) - p_j(0)\| < R - \mu_0\}$ .
- When  $(i, j) \notin \mathcal{E}(t^-)$  and  $\|p_i(t) - p_j(t)\| < R - \mu$ , then we add a new edge  $(i, j)$  to  $\mathcal{E}(t)$ .
- When  $(i, j) \in \mathcal{E}(t^-)$  and  $\|p_i(t) - p_j(t)\| \geq R$ , then we delete the edge  $(i, j)$  from  $\mathcal{E}(t)$ .
- Otherwise,  $\mathcal{E}(t)$  keeps unchanged.

We introduce an indicator function  $\delta_{ij}(t) \in \{0, 1\}$  to show whether there exists an edge between agent  $i$  and agent  $j$  at time  $t$ . The corresponding definition is given as follows:

$$\delta_{ij}(t) = \begin{cases} 0, & \text{if } \{\delta_{ij}(t^-) = 0 \text{ and } \|p_i(t) - p_j(t)\| \geq R - \mu\} \\ & \text{or } \{\delta_{ij}(t^-) = 1 \text{ and } \|p_i(t) - p_j(t)\| \geq R\}; \\ 1, & \text{if } \{\delta_{ij}(t^-) = 0 \text{ and } \|p_i(t) - p_j(t)\| < R - \mu\} \\ & \text{or } \{\delta_{ij}(t^-) = 1 \text{ and } \|p_i(t) - p_j(t)\| < R\}. \end{cases}$$

Then, if the intensity of the communication is defined as a constant adjacency matrix  $\hat{\mathcal{A}}$ , the time-varying communication weights between each pair of agents are  $\mathcal{A}_{ij}(t) = \hat{\mathcal{A}}_{ij} \cdot \delta_{ij}(t)$ . Before proceeding, we introduce the following assumption and definition for demonstrating our main theorem.

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

**Definition 1** If there is no trajectory of the multiagent system with an infinite number of events within a finite period of time, that is  $\inf_k \{t_{k+1} - t_k\} > 0, k = 0, 1, 2, \dots$ , then the multiagent systems have no Zeno behavior.

### 3 Event-based control for multiagent systems

#### 3.1 Design of distributed control algorithm

Our aim is to make all the agents reach consensus while preserving connectivity based on event-triggered mechanism. That is,  $\forall i, j \in \mathcal{V}$ ,

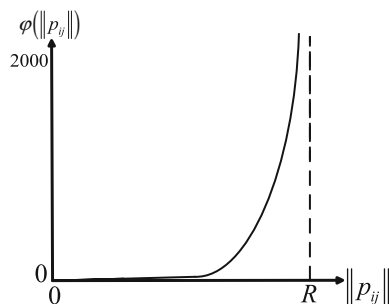
$$\begin{cases} \lim_{t \rightarrow \infty} \|p_i(t) - p_j(t)\| = 0, \\ \lim_{t \rightarrow \infty} v_i(t) = v_j(t) = 0, \end{cases} \quad (6)$$

and no agent will lose connection with its neighbors. Therefore, we adopt the hysteresis function mentioned in [41] to avoid this technical problem. We use the definition of the potential function in [29]:

$$\varphi(\|p_{ij}\|) = \frac{\|p_{ij}\|^2}{R - \|p_{ij}\| + \frac{R^2}{\hat{P}}}, \quad \|p_{ij}\| \in [0, R], \quad (7)$$

where  $R$  is the radius of the communication range,  $\|p_{ij}\| = \|p_i(t) - p_j(t_k)\|$ ,  $t_k$  represents the triggering time of (20) and  $\hat{P} > 0$  is a large constant. It should be noted that we utilize  $\mathcal{A}(t)$ ,  $\mathcal{N}(t)$  and  $\mathcal{L}(t)$  to represent the switching topology.

The schematic plot of the potential function  $\varphi(\|p_{ij}\|)$  is shown in Fig. 3. If we set  $\hat{P}$  large enough,  $\varphi(\|p_{ij}\|)$  can reach an extremely large value when  $\|p_{ij}\| = R$ , which is equal to the effect in Fig. 3. Thus,  $\varphi(\|p_{ij}\|)$  is nonnegative within  $[0, R]$ . Note that as the distance  $\|p_{ij}\|$  between each pair of agents increases, the value of  $\varphi(\|p_{ij}\|)$  will increase. Moreover, when  $\|p_{ij}\|$  approaches  $R$ ,  $\varphi(\|p_{ij}\|)$  will become extremely large to prevent the agent from losing connectivity. Additionally, calculating the derivative of  $\varphi(\|p_{ij}\|)$  with respect to  $\|p_{ij}\|$ , we can obtain that



**Fig. 3** Schematic plot of potential function  $\varphi(\|p_{ij}\|)$

$$\frac{d\varphi(\|p_{ij}\|)}{d(\|p_{ij}\|)} = \frac{\|p_{ij}\| \left( 2R - \|p_{ij}\| + \frac{2R^2}{\hat{P}} \right)}{\left( R - \|p_{ij}\| + \frac{R^2}{\hat{P}} \right)^2} > 0, \quad (8)$$

where  $\|p_{ij}\| \in [0, R]$ . A Lyapunov–Krasovskii functional is utilized as follows:

$$V_U(t) = \frac{1}{2} \sum_{i=1}^N \int_{t-\tau_i}^t U_i(v_i(\zeta)) d\zeta, \quad (9)$$

where  $U_i(v_i(t)) = \phi_i^2(v_i(t))$ . In order to avoid the singularity induced by the denominator of the developed distributed controller, we define a function  $\sigma(\cdot)$  as follows:

$$\sigma(v_i) = \begin{cases} 1, & \text{if } v_i = 0, \\ 0, & \text{if } v_i \neq 0. \end{cases} \quad (10)$$

Then, the distributed controller is divided into five parts:

$$u_i(t) = u_{i1}(t) + u_{i2}(t) + u_{i3}(t) + u_{i4}(t) + u_{i5}(t), \quad (11)$$

$$u_{i1}(t) = - \sum_{j \in \mathcal{N}_i(t)} \nabla_{p_i} \varphi(\|p_{ij}\|),$$

$$u_{i2}(t) = - \sum_{j \in \mathcal{N}_i(t)} \mathcal{A}_{ij}(t)(\hat{q}_i - \hat{q}_j),$$

$$u_{i3}(t) = - \frac{1}{2\|v_i\|^2 + \sigma(v_i)} \phi_i^2(v_i(t)), \quad (12)$$

$$u_{i4}(t) = - k_i(t)v_i,$$

$$u_{i5}(t) = - \hat{W}_i^\top \Phi_i(p_i, v_i),$$

$$k_i(t) = k_{i0} + 1 + \frac{0.5\theta_{N_i}^2}{\|v_i\|^2 + \sigma(v_i)},$$

where  $\hat{q}_i = \sum_{l \in \mathcal{N}_i(t)} \mathcal{A}_{il}(t)(\hat{v}_i - \hat{v}_l)$ ,  $\hat{v}_i = v_i(t_k)$ ,  $t_k (k = 0, 1, 2, \dots)$  are the triggering times,  $\theta_{N_i}$  is the upper bound of  $\theta_i$ ,  $k_{i0} > 0$  is the given feedback coefficient and  $\mathcal{N}_i(t)$  is the time-varying set of the neighbors of agent  $i$ . Specifically,  $u_{i1}(t)$  has the explicit expression based on potential function (7). That is

$$\nabla_{p_i} \varphi(\|p_{ij}\|) = \omega_{ij}(t)(p_i(t) - p_j(t_k)), \quad (13)$$

where

$$\omega_{ij}(t) = \begin{cases} \frac{\left( 2R - \|p_{ij}\| + \frac{2R^2}{\hat{P}} \right)}{\left( R - \|p_{ij}\| + \frac{R^2}{\hat{P}} \right)^2}, & (j, i) \in \mathcal{E}(t); \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

*Remark 1*  $u_{i1}(t)$  and  $u_{i2}(t)$  only need the information of agent  $i$  and its neighbor’s positions and velocities at triggering time  $t_k$ , respectively. Thus, compared with [19], the communication times among the multiagent system (5) are highly reduced.

The online updating algorithm for the weight matrix of the RBFNN is given as follows:

$$\dot{\hat{W}}_i = \begin{cases} \chi_i \Phi_i(p_i, v_i) v_i^T, & \text{if } \text{tr}(\hat{W}_i^T \hat{W}_i) < W_i^{\max}, \text{ or} \\ \text{if } \text{tr}(\hat{W}_i^T \hat{W}_i) = W_i^{\max} \text{ and } v_i^T \hat{W}_i^T \Phi_i(p_i, v_i) < 0, \\ \chi_i \Phi_i(p_i, v_i) v_i^T - \chi_i \frac{v_i^T \hat{W}_i^T \Phi_i(p_i, v_i)}{\text{tr}(\hat{W}_i^T \hat{W}_i)} \hat{W}_i, & \text{otherwise,} \end{cases} \quad (15)$$

where  $i = 1, 2, \dots, N$ ,  $\hat{W}_i$  is the estimation of the ideal weight matrix  $W^*$  and  $\chi_i$  is the updating rate. Moreover, the initial values of  $\hat{W}_i$  should satisfy  $\text{tr}(\hat{W}_i^T(0) \hat{W}_i(0)) \leq W_i^{\max}$ .

### 3.2 Main theorem

Before proceeding, we define the potential energy function for agent  $i$  as follows:

$$P_i(t) = \sum_{j \in \mathcal{N}_i(t)} \varphi(\|p_{ij}\|) + \frac{1}{2} v_i^T v_i + \frac{1}{2} \int_{t-\tau_i}^t U_i(v_i(\zeta)) d\zeta + \frac{1}{2} \text{tr} \left( \frac{1}{\chi_i} \tilde{W}_i^T \tilde{W}_i \right), \quad (16)$$

where  $\tilde{W}_i = W_i^* - \hat{W}_i$ . Then, the total potential energy function is  $P(t) = \sum_{i=1}^N P_i(t)$ .

**Theorem 1** *The multiagent system (5) consists of  $N$  agents and the distributed controller is designed as (11) with the triggering function (20). With Assumption 1, if the initial energy  $P(0)$  is finite, and the initial undirected topology  $\mathcal{G}(0)$  is connected, the consensus of the multiagent system (5) can be reached with preserving connectivity.*

*Proof* The derivative of  $P(t)$  is

$$\begin{aligned} \dot{P}(t) &= \sum_{i=1}^N \sum_{j \in \mathcal{N}_i(t)} \dot{\varphi}(\|p_{ij}\|) + \sum_{i=1}^N v_i^T \dot{v}_i + \dot{V}_v(t) - \sum_{i=1}^N \text{tr} \left( \frac{1}{\chi_i} \tilde{W}_i^T \dot{\tilde{W}}_i \right) \\ &= \sum_{i=1}^N v_i^T \sum_{j \in \mathcal{N}_i(t)} \nabla_{p_i} \varphi(\|p_{ij}\|) + \frac{1}{2} \sum_{i=1}^N (\phi_i^2(v_i(t)) - \phi_i^2(v_i(t - \tau_i))) \\ &\quad + \sum_{i=1}^N v_i^T \left( -k_i(t) v_i - \sum_{j \in \mathcal{N}_i(t)} \nabla_{p_i} \varphi(\|p_{ij}\|) - \sum_{j \in \mathcal{N}_i(t)} \mathcal{A}_{ij}(t) (\hat{q}_i - \hat{q}_j) \right. \\ &\quad \left. - \frac{1}{2\|v_i\|^2 + \sigma(v_i)} \phi_i^2(v_i(t)) - \hat{W}_i^T \Phi_i(\cdot) + g_i(v_i(t - \tau_i)) \right. \\ &\quad \left. + W_i^{*T} \Phi_i(\cdot) + \theta_i \right) - \sum_{i=1}^N \text{tr} \left( \frac{1}{\chi_i} \tilde{W}_i^T \dot{\tilde{W}}_i \right), \end{aligned} \quad (17)$$

where  $\Phi_i(\cdot)$  is short for  $\Phi_i(p_i, v_i)$ . If  $\text{tr}(\hat{W}_i^T(0) \hat{W}_i(0)) \leq W_i^{\max}$ , it is easy to demonstrate that

$\text{tr}(\hat{W}_i^T(t) \hat{W}_i(t)) \leq W_i^{\max}$  [22]. Thus, according to the updating algorithm (15), the inequality  $\text{tr}(\tilde{W}_i^T \left( \frac{1}{\chi_i} \dot{\tilde{W}}_i - \Phi_i(\cdot) v_i^T \right)) \geq 0$  holds. If  $v_i(t) = 0$ ,  $\dot{P}_i(t) = -\frac{1}{2} \phi_i^2(v_i(t - \tau_i)) < 0$ . Thus, we need to focus on the case where  $v_i(t) \neq 0$ . Merging the polynomial (17), we can obtain

$$\begin{aligned} \dot{P}(t) &= \frac{1}{2} \sum_{i=1}^N (\phi_i^2(v_i(t)) - \phi_i^2(v_i(t - \tau_i))) - \sum_{i=1}^N \text{tr} \left( \frac{1}{\chi_i} \tilde{W}_i^T \dot{\tilde{W}}_i \right) \\ &\quad + \sum_{i=1}^N v_i^T \left( - \sum_{j \in \mathcal{N}_i(t)} \mathcal{A}_{ij}(t) (\hat{q}_i - \hat{q}_j) - \frac{1}{2\|v_i\|^2} \phi_i^2(v_i(t)) \right. \\ &\quad \left. - k_i(t) v_i + \tilde{W}_i^T \Phi_i(\cdot) + g_i(v_i(t - \tau_i)) + \theta_i \right) \\ &= - \sum_{i=1}^N k_i(t) \|v_i\|^2 - \sum_{i=1}^N v_i^T \sum_{j \in \mathcal{N}_i(t)} \mathcal{A}_{ij}(t) (\hat{q}_i - \hat{q}_j) \\ &\quad - \sum_{i=1}^N \text{tr} \left( \tilde{W}_i^T \left( \frac{1}{\chi_i} \dot{\tilde{W}}_i - \Phi_i(\cdot) v_i^T \right) \right) \\ &\quad - \frac{1}{2} \sum_{i=1}^N \phi_i^2(v_i(t - \tau_i)) + \sum_{i=1}^N v_i^T (g_i(v_i(t - \tau_i)) + \theta_i) \\ &\leq - \sum_{i=1}^N k_i(t) \|v_i\|^2 - \frac{1}{2} \sum_{i=1}^N \phi_i^2(v_i(t - \tau_i)) \\ &\quad + \frac{1}{2} \sum_{i=1}^N (2\|v_i\|^2 + \|g_i(v_i(t - \tau_i))\|^2 + \|\theta_i\|^2) \\ &\quad - \sum_{i=1}^N v_i^T \sum_{j \in \mathcal{N}_i(t)} \mathcal{A}_{ij}(t) (\hat{q}_i - \hat{q}_j) \text{ (with Assumption 1)} \\ &\leq - \sum_{i=1}^N (k_i(t) - 1) \|v_i\|^2 + \frac{1}{2} \sum_{i=1}^N \theta_{N_i}^2 \\ &\quad - \sum_{i=1}^N v_i^T \sum_{j \in \mathcal{N}_i(t)} \mathcal{A}_{ij}(t) (\hat{q}_i - \hat{q}_j), \end{aligned} \quad (18)$$

where  $\|\theta_i\| < \theta_{N_i}$ . Then,

$$\begin{aligned} \dot{P}(t) &< - \sum_{i=1}^N v_i^T \sum_{j \in \mathcal{N}_i(t)} \mathcal{A}_{ij}(t) (\hat{q}_i - \hat{q}_j) \\ &\leq - v^T (\mathcal{L}(t) \otimes I_2) \hat{q} \\ &\leq - v^T (\mathcal{L}(t) \otimes I_2) (\mathcal{L}(t) \otimes I_2) \hat{v}, \end{aligned}$$

where

$$\begin{aligned} v &= (v_1^T(t), v_2^T(t), \dots, v_N^T(t))^T, \\ \hat{v} &= (\hat{v}_1^T, \hat{v}_2^T, \dots, \hat{v}_N^T)^T, \\ \hat{q} &= (\hat{q}_1^T, \hat{q}_2^T, \dots, \hat{q}_N^T)^T. \end{aligned}$$

Let  $e_i(t) = \hat{v}_i - v_i(t)$ , for  $\mathcal{L}(t)$  is symmetric, then

$$\begin{aligned} \dot{P}(t) &< -v^T(\mathcal{L}(t) \otimes I_2)(\mathcal{L}(t) \otimes I_2)(v + e) \\ &= -\|(\mathcal{L}(t) \otimes I_2)v\|^2 - v^T(\mathcal{L}(t) \otimes I_2)(\mathcal{L}(t) \otimes I_2)e \\ &\leq -\|(\mathcal{L}(t) \otimes I_2)v\|^2 + \|(\mathcal{L}(t) \otimes I_2)v\| \| \mathcal{L}(t) \otimes I_2 \| \|e\|, \end{aligned}$$

where  $e = (e_1^T(t), e_2^T(t), \dots, e_N^T(t))^T$ . In order to guarantee  $\dot{P}(t) < 0$ , we enforce  $e$  to satisfy

$$\|e\| \leq \beta \frac{\|(\mathcal{L}(t) \otimes I_2)v\|}{\|\mathcal{L}(t) \otimes I_2\|}, \tag{19}$$

where  $\beta \in (0, 1)$ . Then,  $\dot{P}(t) < (\beta - 1)\|(\mathcal{L}(t) \otimes I_2)v\|^2 \leq 0$ . Therefore, the triggering function is

$$z(v) = \|e\| - \beta \frac{\|(\mathcal{L}(t) \otimes I_2)v\|}{\|\mathcal{L}(t) \otimes I_2\|}. \tag{20}$$

According to the similar proof steps in [20], we can prove that the inter-event times  $\inf_k \{t_{k+1} - t_k\}$ ,  $k = 0, 1, 2, \dots$ , are strictly positive by a lower bounded time  $\hat{t} = \frac{\beta}{\|\mathcal{L}(t)\|(1+\beta)}$ . Therefore, the Zeno behavior is excluded.

$\dot{P}(t) < 0$  when  $t \in [t_{c0}, t_{c1})$ , and this means that  $P(t) \leq P_0 < P_{\max}, \forall t \in [t_{c0}, t_{c1})$ . For  $\varphi(R) > P_{\max} > P_0$ , the distance between each pair of the existing edges cannot reach  $R$  when  $t \in [t_{c0}, t_{c1})$ . Therefore, no edge will lose connectivity at time  $t_{c1}$ . Due to the decrease in the total potential energy function  $P(t)$ , there must be new edges which are added to the network at switching time  $t_{c1}$ . Suppose that at time  $t_{c1}$ ,  $s_{c1}$  new edges are added to the network, where  $0 < s_{c1} \leq S_c$  and  $S_c = \frac{(N-1)(N-2)}{2}$ . Thus, we can imply that  $P(t_{c1}) \leq P_0 + s_{c1}\varphi(\|R - \mu\|) \leq P_{\max}$ .

Following the similar proof steps in the above analysis, when  $t \in [t_{c,k-1}, t_{ck}), k = 2, 3, \dots$ , the derivative of the total potential energy function  $P(t)$  is  $\dot{P}(t) < (\beta - 1)\|(\mathcal{L}(t) \otimes I_2)v\|^2 \leq 0$ . Thus,  $P(t) \leq P(t_{c,k-1}) < P_{\max}, t \in [t_{c,k-1}, t_{ck}), k = 1, 2, \dots$ . Therefore, no edge will lose connectivity at time  $t_{ck}, k = 1, 2, \dots$ . In summary, if the initial energy  $P(0)$  is finite and the initial undirected topology  $\mathcal{G}(0)$  is connected, the connectivity can be preserved.

There are at most  $S_c$  new edges to be added to the network, and thus, the switching times of the topology are finite. Specifically, the following discussion is based on the fixed topology. Since every term in  $P(t)$  is positive and  $\dot{P}(t) < 0$  when  $v_i \neq 0$ , all the terms such as  $\varphi(\|p_{ij}\|), v_i^T v_i, V_U(t)$  and  $\text{tr}(\frac{1}{\hat{\mu}_i} \tilde{W}_i^T \tilde{W}_i)$  will tend to be zero. That is,  $p_1 = p_2 = \dots = p_N$  and  $v_1 = v_2 = \dots = v_N = 0$ . Therefore, all the agents can asymptotically reach the same position and velocity satisfied with (6). Furthermore,  $\lim_{t \rightarrow \infty} \|W_i^* - \hat{W}_i\| = 0$  shows that RBFNNs can learn the unknown dynamics of the multiagent system. The proof is complete.  $\square$

### 4 Simulation example

The multiagent system includes seven agents which move on the two-dimensional plane. We set the sensing radius  $R = 4$  m and choose the initial positions and velocities randomly from  $[0, 8 \text{ m}] \times [0, 8 \text{ m}]$  and  $[0, 2 \text{ m/s}] \times [0, 2 \text{ m/s}]$ , respectively. For simplicity, here we suppose that all the existing communication weights are 1 and the initial network is undirected and connected. In addition,  $\mu = \mu_0 = 1$  m,  $\hat{P} = 1000$  and the time-delay parameters are in Table 1.

The dynamics of time-delay terms are described as follows:

$$g_i(v_i(t)) = \begin{bmatrix} m_{i1} v_{i1}(t) \cos(v_{i2}(t)) \\ m_{i2} v_{i2}(t) \sin(v_{i1}(t)) \end{bmatrix}, \tag{21}$$

where  $m_{i1}$  and  $m_{i2}, i = 1, 2, \dots, 7$ , are the corresponding constant coefficients given in Table 2, and the feedback coefficients are chosen to be the same as  $k_{i0} = 5, i = 1, 2, \dots, 7$ . The parameter of the event-triggered function is  $\beta = 0.9$ . Then,

$$\phi_i(v_i(t)) = \sqrt{(m_{i1} v_{i1}(t))^2 + (m_{i2} v_{i2}(t))^2}, \quad i = 1, 2, \dots, 7.$$

The unknown dynamics are chosen to be

$$f_i(p_i(t), v_i(t)) = \begin{bmatrix} l_{i1} p_{i1}(t) \sin(p_{i2}(t)) v_{i1} v_{i2} \\ l_{i2} p_{i2}(t) \cos(p_{i1}(t)) \sin(v_{i1} v_{i2}) \end{bmatrix}, \tag{22}$$

where  $l_{i1}$  and  $l_{i2}, i = 1, 2, \dots, 7$ , are the corresponding constant coefficients given in Table 3. The parameters of RBFNNs are  $\theta_{N_i} = 0.1, \hat{\sigma}_i = 2, W_i^{\max} = 100$  and  $\chi_i = 100$ . The number of neurons for each RBFNN is 16.  $\hat{\mu}_i$  is distributed uniformly among the range  $[0, 10] \times [0, 10]$ .

**Table 1** Time delay for each agent

$i$	1	2	3	4	5	6	7
$\tau_i$	0.1	0.05	0.15	0.08	0.18	0.1	0.04

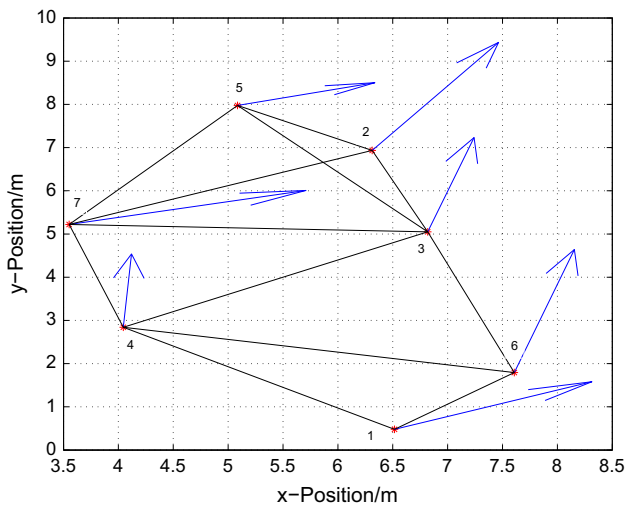
**Table 2** Coefficients of time-delay terms

$i$	1	2	3	4	5	6	7
$m_{i1}$	0.4	-0.65	0.5	-0.75	0.1	0.4	0.2
$m_{i2}$	0.5	0.45	-0.6	0.4	1	0.1	0.3

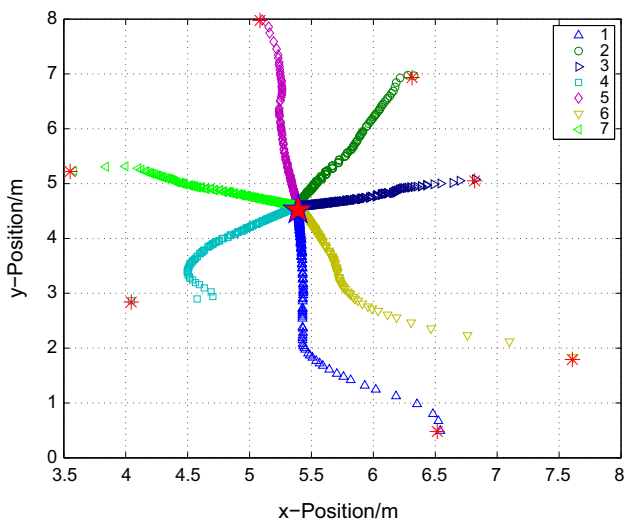
**Table 3** Coefficients of nonlinear terms

$i$	1	2	3	4	5	6	7
$l_{i1}$	0.9	1.2	-1.1	-0.7	0.6	0.3	0.5
$l_{i2}$	1.2	0.8	0.6	0.3	0.8	0.4	-0.6



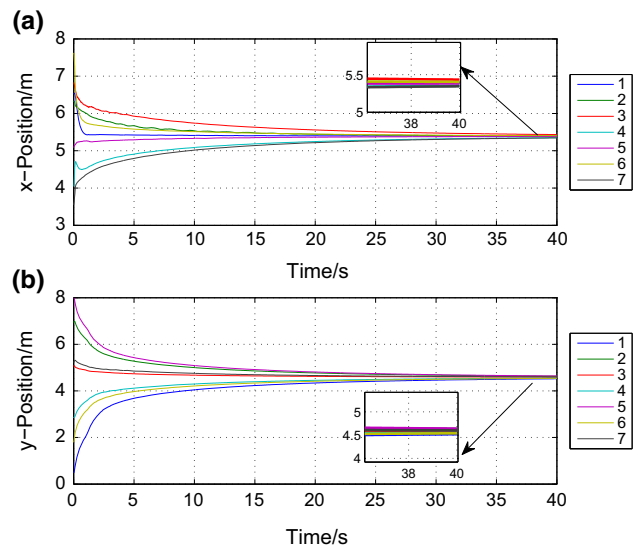


**Fig. 4** Initial positions of the multiagent system

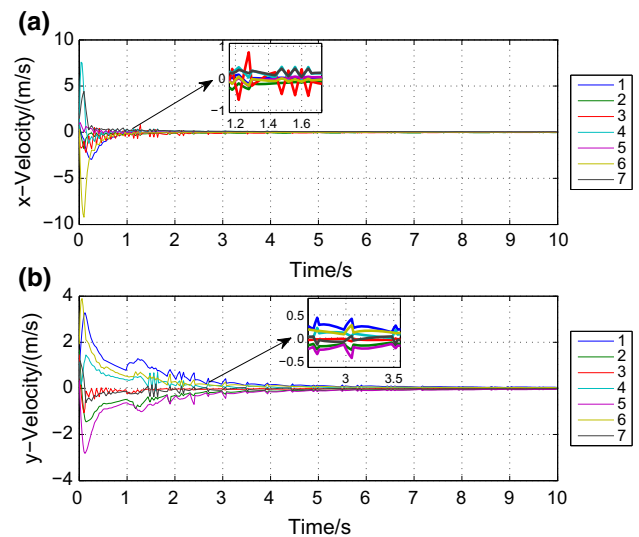


**Fig. 5** Paths of the multiagent system on  $x$ - $y$  plane for  $\mu = 1$  m with event-triggered mechanism

Figure 4 illustrates the initial positions of the seven agents, where the red asterisks, the black solid lines and the blue arrows represent the agents, the edges and the directions of velocities, respectively. In Fig. 5, the red asterisks represent the starting points of the multiagent system and the trajectories are consisted of a sequence of points with seven different symbols. Each point means that the function (20) is triggered at that point. The red star represents the consensus point of the seven agents, and it shows that under the centralized event-based control algorithm, all the agents can reach consensus. This demonstrates Theorem 1. In addition, when  $\mu = 1$  m, Fig. 6 shows that the final positions of the seven agents are the same which are near (5.5, 4.6 m).

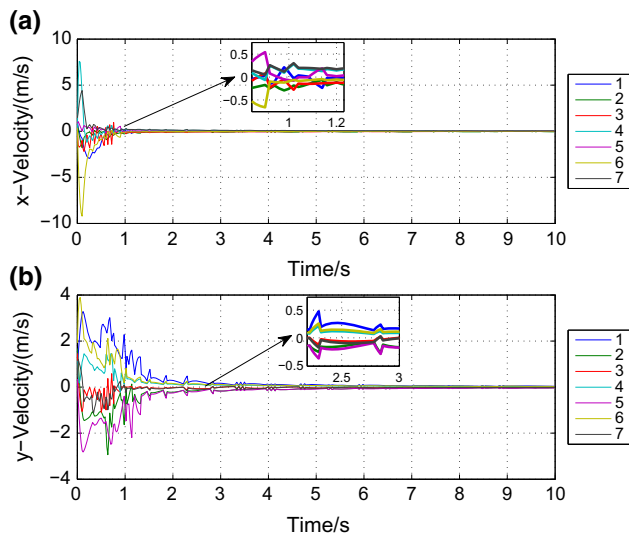


**Fig. 6** Position trajectories of seven agents for  $\mu = 1$  m. **a**  $x$ -Position. **b**  $y$ -Position



**Fig. 7** Velocity trajectories of seven agents for  $\mu = 1$  m. **a**  $x$ -Velocity. **b**  $y$ -Velocity

Figures 7 and 8 illustrate the trajectories of velocities in two dimensions when  $\mu = 1$  m and  $\mu = 0.5$  m, respectively. From the extracting windows in Fig. 7, all the velocities change at the same time and this in turn shows the characteristics of the centralized event-based control. In addition, due to the existence of frictions in the dynamics of (5), all the velocities tend to be zero. Furthermore, comparing with Figs. 7 and 8, we can see that when  $\mu = 0.5$  m the velocities vary more sharply than  $\mu = 1$  m. This can be interpreted that the smaller the hysteresis parameter  $\mu$  is, the more sensitive the indicator function  $\delta_{ij}(t)$  will be. The percentage in Table 4 is defined as follows:



**Fig. 8** Velocity trajectories of seven agents for  $\mu = 0.5$  m. **a**  $x$ -Velocity. **b**  $y$ -Velocity

**Table 4** Percentage of event-triggered case compared with time-triggered case

Case	$\mu$ (m)	$\beta$	Percentage (%)
1	1	0.9	15.15
2	1	0.1	29.95
3	0.5	0.9	14.55
4	0.5	0.1	29.60

$$\text{Percentage (\%)} = \frac{\text{Times of self-triggered events}}{\text{Times of time-triggered events}} \times 100\%.$$

In Table 4, we set all the parameters unchanged except  $\beta$  and  $\mu$  to make comparisons of the performance affected by these two parameters. From Table 4, we can obtain that when the hysteresis parameter  $\mu$  decreases, new edges are easier to be added. Then, agents can get more information from their neighbors at one triggering time. Thus, the event-triggered times are reduced. However, the key factor of the performance is the parameter  $\beta$  of the triggering function (20). As  $\beta$  increases, the threshold of (20) increases and the average time between two triggered events also increases. Thus, the triggered times are highly reduced which can save communication resources and power consumptions. And this demonstrates the advantages of the event-triggered control over the time-triggered control.

## 5 Conclusion

This paper investigates nonlinear time-delayed multiagent systems based on event-based mechanism with connectivity preservation. The distributed controller is divided into five parts with the idea of divide-and-conquer. By utilizing

radial basis function neural networks, the distributed controller can learn the unknown dynamics online. Moreover, by using Lyapunov–Krasovskii functionals, the negative effects of time delays can be eliminated. The event-based mechanism is introduced to reduce communication times among the multiagent systems and save power consumptions. By devising a proper event-triggered function, the Zeno behavior is excluded which can improve the stability of the multiagent systems. Connectivity preservation can be guaranteed by utilizing a high-threshold potential function. Simulation results demonstrate the validity of the developed methodology. In future work, we will investigate how to use distributed event-triggered mechanism and design self-triggered functions to make the algorithm more applicable to physical implementations.

**Acknowledgments** This work was supported in part by the National Natural Science Foundation of China under Grants 61233001, 61273140, 61304086, 61533017, 61503379 and U1501251, in part by China Scholarship Council under the State Scholarship Fund, in part by Beijing Natural Science Foundation under Grant 4162065 and in part by the Early Career Development Award of SKLMCCS.

## References

- Butt MA, Akram M (2015) A novel fuzzy decision-making system for CPU scheduling algorithm. *Neural Comput Appl*. doi:10.1007/s00521-015-1987-8
- Cao Y, Yu W, Ren W, Chen G (2013) An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Trans Ind Inf* 9(1):427–438
- Chen C, Wen GX, Liu YJ, Wang FY (2014) Adaptive consensus control for a class of nonlinear multiagent time-delay systems using neural networks. *IEEE Trans Neural Netw Learn Syst* 25(6):1217–1226
- Cheng L, Hou ZG, Tan M, Lin Y, Zhang W (2010) Neural-network-based adaptive leader-following control for multiagent systems with uncertainties. *IEEE Trans Neural Netw* 21(8):1351–1358
- De La Torre G, Yucelen T, Johnson E (2014) Bounded hybrid connectivity control of networked multiagent systems. *IEEE Trans Autom Control* 59(9):2480–2485
- Dong Y, Huang J (2014) Leader-following connectivity preservation rendezvous of multiple double integrator systems based on position measurement only. *IEEE Trans Autom Control* 59(9):2598–2603
- Gao W, Jiang ZP (2016) Nonlinear and adaptive suboptimal control of connected vehicles: a global adaptive dynamic programming approach. *J Intell Robot Syst*. doi:10.1007/s10846-016-0395-3
- Gao W, Jiang ZP, Ozbay K (2016) Data-driven adaptive optimal control of connected vehicles. *IEEE Trans Intell Transp Syst*. doi:10.1109/TITS.2016.2597279
- Ge S, Hong F, Lee T (2004) Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients. *IEEE Trans Syst Man Cybern Part B Cybern* 34(1):499–516
- He W, Ge S (2016) Cooperative control of a nonuniform gantry crane with constrained tension. *Automatica* 66(4):146–154
- Heemels W, Johansson KH, Tabuada P (2012) An introduction to event-triggered and self-triggered control. In: *Proceedings of the*



- conference on decision and control, Maui, Hawaii, USA, pp 3270–3285
12. Hou ZG, Cheng L, Tan M (2009) Decentralized robust adaptive control for the multiagent system consensus problem using neural networks. *IEEE Trans Syst Man Cybern Part B Cybern* 39(3):636–647
  13. Jadbabaie A, Lin J, Morse AS (2003) Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans Autom Control* 48(6):988–1001
  14. Jiang ZP, Liu TF (2015) A survey of recent results in quantized and event-based nonlinear control. *Int J Autom Comput* 12(5):455–466
  15. Liang H, Zhang H, Wang Z, Wang J (2014) Consensus robust output regulation of discrete-time linear multi-agent systems. *IEEE/CAA J Autom Sin* 1(2):204–209
  16. Liu D, Wang D, Li H (2014) Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach. *IEEE Trans Neural Netw Learn Syst* 25(2):418–428
  17. Liu T, Jiang ZP (2015) Event-based control of nonlinear systems with partial state and output feedback. *Automatica* 53:10–22
  18. Luo X, Feng L, Yan J, Guan X (2015) Dynamic coverage with wireless sensor and actor networks in underwater environment. *IEEE/CAA J Autom Sin* 2(3):274–281
  19. Ma H, Liu D, Wang D (2015) Distributed control for nonlinear time-delayed multi-agent systems with connectivity preservation using neural networks. In: Proceedings of the 22nd international conference on neural information processing, Istanbul, Turkey, pp 34–42
  20. Ma H, Liu D, Wang D, Tan F, Li C (2015) Centralized and decentralized event-triggered control for group consensus with fixed topology in continuous time. *Neurocomputing* 161:267–276
  21. Ma H, Liu D, Wang D, Luo B (2016) Bipartite output consensus in networked multi-agent systems of high-order power integrators with signed digraph and input noises. *Int J Syst Sci* 47(13):3116–3131
  22. Ma H, Wang Z, Wang D, Liu D, Yan P, Wei Q (2016) Neural-network-based distributed adaptive robust control for a class of nonlinear multiagent systems with time delays and external noises. *IEEE Trans Syst Man Cybern Syst* 46(6):750–758
  23. Mu C, Ni Z, Sun C, He H (2016) Data-driven tracking control with adaptive dynamic programming for a class of continuous-time nonlinear systems. *IEEE Trans Cybern*. doi:10.1109/TCYB.2016.2548941
  24. Olfati-Saber R, Murray RM (2004) Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans Autom Control* 49(9):1520–1533
  25. Reynolds CW (1987) Flocks, herds, and schools: a distributed behavioral model. *Comput Graph* 21(4):25–34
  26. Schneider MO, Rosa JLG (2009) Application and development of biologically plausible neural networks in a multiagent artificial life system. *Neural Comput Appl* 18(1):65–75
  27. Seyboth GS, Dimarogonas DV, Johansson KH (2013) Event-based broadcasting for multi-agent average consensus. *Automatica* 49(1):245–252
  28. Shen J, Tan H, Wang J, Wang J, Lee S (2015) A novel routing protocol providing good transmission reliability in underwater sensor networks. *J Internet Technol* 16(1):171–178
  29. Su H, Wang X, Chen G (2010) Rendezvous of multiple mobile agents with preserved network connectivity. *Syst Control Lett* 59(5):313–322
  30. Tabuada P (2007) Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Trans Autom Control* 52(9):1680–1685
  31. Vicsek T, Czirók A, Ben-Jacob E, Cohen I, Shochet O (1995) Novel type of phase transition in a system of self-driven particles. *Phys Rev Lett* 75(6):1226–1229
  32. Wang D, Liu D, Zhao D, Huang Y, Zhang D (2013) A neural-network-based iterative GDHP approach for solving a class of nonlinear optimal control problems with control constraints. *Neural Comput Appl* 22(2):219–227
  33. Wang D, Liu D, Li H, Ma H, Li C (2016) A neural-network-based online optimal control approach for nonlinear robust decentralized stabilization. *Soft Comput* 20(2):707–716
  34. Wang D, Liu D, Mu C, Ma H (2016) Decentralized guaranteed cost control of interconnected systems with uncertainties: a learning-based optimal control strategy. *Neurocomputing*. doi:10.1016/j.neucom.2016.06.020
  35. Wang D, Ma H, Liu D (2016) Distributed control algorithm for bipartite consensus of the nonlinear time-delayed multi-agent systems with neural networks. *Neurocomputing* 174:928–936
  36. Wang D, Mu C, He H, Liu D (2016) Event-driven adaptive robust control of nonlinear systems with uncertainties through ndp strategy. *IEEE Trans Syst Man Cybern Syst*. doi:10.1109/TSMC.2016.2592682
  37. Wang D, Mu C, Liu D (2016) Data-driven nonlinear near-optimal regulation based on iterative neural dynamic programming. *Acta Autom Sin* (accepted)
  38. Werfel J, Petersen K, Nagpal R (2014) Designing collective behavior in a termite-inspired robot construction team. *Science* 343(6172):754–758
  39. Xie S, Wang Y (2014) Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. *Wirel Pers Commun* 78(1):231–246
  40. Yu H, Antsaklis P (2012) Formation control of multi-agent systems with connectivity preservation by using both event-driven and time-driven communication. In: Proceedings of the IEEE conference on decision control, Maui, Hawaii, USA, pp 7218–7223
  41. Zavlanos M, Tanner H, Jadbabaie A, Pappas G (2009) Hybrid control for connectivity preserving flocking. *IEEE Trans Autom Control* 54(12):2869–2875
  42. Zhu W, Jiang ZP (2015) Event-based leader-following consensus of multi-agent systems with input time delay. *IEEE Trans Autom Control* 60(5):1362–1367