Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 351 (2014) 4271–4286

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

Model-based event-triggered multi-vehicle coordinated tracking control using reduced order models

Eloy Garcia[a,b,*], Yongcan Cao[b], David W. Casbeer[b]

[a]Infoscitex Corporation, Dayton, OH 45431, USA

[b]Control Science Center of Excellence, Air Force Research Laboratory, Wright-Patterson AFB, OH 45433, USA

Received 13 August 2013; received in revised form 2 February 2014; accepted 17 May 2014
Available online 28 May 2014

Abstract

The coordinated tracking problem where a group of followers intercepts a dynamic leader is studied. It is shown in this paper that reduction of inter-agent communication is obtained and improved performance is achieved when each follower implements dynamical models of neighbors and by using an event-triggered control strategy that requires each agent to send measurement updates only when necessary. The results in this paper consider directed graphs and the possible existence of cycles. Performance bounds on the tracking error have been obtained, which are functions of the communication topology and the event thresholds. This approach is extended to consider measurement noise and similar bounds are presented for this case.

1. Introduction

Cooperative control problems have received significant attention in the last years. These problems typically require a group of mobile vehicles, also called agents, to perform a coordinated task by exchanging relevant information according to either a fixed or a time-varying communication topology. Most of the results concerning agents with continuous time dynamics require each agent to continuously transmit measurements to its neighbors [1,2]. However, in practical problems there exist computation and, especially, communication limitations that impose constraints into how frequently an agent is able to receive measurement updates from its neighbors.

*Corresponding author at: Infoscitex Corporation, Dayton, OH 45431, USA.
E-mail address: elgarcia@infoscitex.com (E. Garcia).

http://dx.doi.org/10.1016/j.jfranklin.2014.05.004
0016-0032/Published by Elsevier Ltd. on behalf of The Franklin Institute.

Recently, typical problems in cooperative control have been studied from a sampled-data perspective [3–10] where periodic samples of continuous time output signals are transmitted over a communication channel and are also used to compute sampled control inputs. The work in [3] considers a sampled-data consensus approach for double-integrator dynamics and for undirected graphs. The authors of [4] consider sampled-data consensus, and two communication cases: synchronous and asynchronous. Qin and Gao [5] studied the consensus problem with sampled information and time-varying communication delays. Their approach considers switching communication topologies and consensus is obtained for small enough sampling periods and if there exists a frequent enough directed spanning tree in the presence of delays. In [6] the sampled-data consensus problem was considered for agents with double-integrator dynamics. In the present paper, we extend the results on sampled-data cooperative tracking provided in [7], in order to reduce communication rates between agents and to improve performance as measured by the tracking errors.

We make use of the Model-Based Event-Triggered (MB-ET) control strategy [11–12,21–22],25 in which each agent implements a nominal dynamical model of other agents in the network. The states of the models provide an estimate of the real positions of neighbor vehicles and these estimated variables are used to compute the control inputs between measurement updates. Each agent generates its corresponding updates based on local error which measure the difference between real and estimated positions. The use of event-triggered control in cooperative control has been used by several authors [13–19]. The approach in [14,15] considered reduction of actuation updates assuming continuous communication while our purpose in this paper is to reduced communication updates. In the papers [13–19] a Zero-Order-Hold (ZOH) model is used, that is, the measurement updates received by each agent are held constant until new measurements arrive. In the approach shown in this paper, the updates received from neighbors are used by each agent to update an internal model of the neighbor and provide an estimate of the neighbor's position between measurement updates.

The paper is organized as follows: Section 2 states the problem. Section 3 establishes conditions to obtain bounded tracking errors using the MB-ET framework. Section 4 provides extensions to consider the case of measurement noise. Section 5 presents illustrative examples and Section 6 provides conclusions.

2. Problem statement

The problem studied in this paper is the dynamic leader coordinated tracking problem with limited communication where a group of n agents receive information, directly or indirectly, about the leader's position at some time instants and they are required to track the leader's position as close as possible. The approach we follow is a model-based approach in which each agent is equipped with dynamical models of other agents in the group. In the approach described in this paper each agent implements models of its neighbors only.

A similar approach was developed in our previous work [20]. In [20] we considered a different implementation that was restricted to graphs represented by directed trees with root at the leader. Each agent had to implement models of all nodes in the path from the leader to that particular agent. Since all agents implemented a model of the leader then updates from the leader had to be retransmitted through the entire network resulting in communication delays due to multi-hop communication. The approach in the present paper uses only models of neighbors, i.e. each agent implements models of those agents that directly affect its dynamics. In this case the events at each particular agent are triggered only by the size of its local error and it is not necessary to retransmit information originated at other nodes. The main advantage of this approach is that it allows us to consider more general communication topologies, other than directed trees. In this paper we consider general connected and directed graphs, and we allow for the presence of cycles in the communication topology. Since information is not retransmitted at any node, it is reasonable to assume that communication delays are negligible.

Consider a group of n agents, labeled as followers. Assume the dynamics of each agent are given by the following continuous time dynamics:

$$\dot{r}_i(t) = u_i(t), \quad i = 1, 2, ..., n. \tag{1}$$

The control inputs are given in sampled-data form as follows:

$$u_i(t) = u_i[k], \quad for \ \ kT \leq t < (k+1)T. \tag{2}$$

We use the sampling time $T > 0$ to discretize the continuous time dynamics (1) to obtain

$$r_i[k+1] = r_i[k] + Tu_i[k]. \tag{3}$$

Remark 1. The sampling time T is only used to obtain a discrete time equivalent of the continuous time agent dynamics but it is not used to establish a periodic communication between agents. At every time k each agent measures its position, computes models of positions of other agents and its local error, and updates its local input, but it does not transmit its measurement at every time k. Transmission instants do not occur at every sampling time k but only when the local state error becomes larger than a predefined threshold.

The discrete time models that will be implemented by each agent are represented by

$$\hat{r}_i^{(l)}[k+1] = \hat{r}_i^{(l)}[k] + T\hat{u}_i^{(l)}[k] \tag{4}$$

where $\hat{r}_i^{(l)}[k]$ represents the model state of agent i estimated by agent l and $\hat{u}_i^{(l)}[k]$ is the corresponding model control input.

In order to determine the time instants at which a given agent needs to broadcast its current measured position we use an event-triggered strategy. In this case each agent i needs to implement a model of itself that is consistent with the models other agents have of agent i. Let us call this second model the 'q' model with dynamics

$$q_i^{(i)}[k+1] = q_i^{(i)}[k] + T\overline{u}_i^{(i)}[k]. \tag{5}$$

The difference between the inputs $u_i[k]$, $\hat{u}_i[k]$, and $\overline{u}_i[k]$ will become apparent in the following section. The events are triggered by the size of the local agent state error. The local state errors are given by

$$\varepsilon_i^{(i)}[k] = r_i[k] - q_i^{(i)}[k], \quad i = 0, 1, 2, ..., n \tag{6}$$

which measure the difference between the real agent position and the position estimated by its own local model. When agent i transmits a measurement update then we have

$$q_i^{(i)}[k_\mu^i] = r_i[k_\mu^i] \tag{7}$$

where k_μ^i, $\mu = 1, 2, \ldots$, represents the update instants for agent i, i.e. the time instants when agent i updates its model and transmits this update; we use this notation to emphasize the fact that the updates do not take place, in general, at every time k, but at some irregular instants k_μ^i indexed by μ. Note that if agent i is a neighbor of agent l then the transmitted measurement (7) will be received by agent l, who updates its 'r' model of agent i, i.e.

$$\hat{r}_i^{(l)}[k_\mu^i] = r_i[k_\mu^i]. \tag{8}$$

Each agent will transmit its current measured position if its local error is larger than a predefined threshold otherwise the agent will not attempt to send measurements since the current estimate is close to its real position. Each agent transmits its current measurement when the following is satisfied:

$$|\varepsilon_i^{(i)}[k]| > \alpha. \tag{9}$$

When an agent broadcasts its current measurement the error is equal to zero ($\varepsilon_i^{(i)}[k_\mu^i] = 0$) at that time instant because of the update (7). Then, because of the triggering condition (9), the following holds:

$$|\varepsilon_i^{(i)}[k]| \leq \alpha, \quad i = 0, 1, 2, ..., n. \tag{10}$$

3. Proportional-like controller

In the MB-ET framework the leader implements a model of its own dynamics. Suppose that the leader dynamics can be represented as in (3), that is, $r_0[k+1] = r_0[k] + Tu_0[k]$ for unknown $u_0[k]$. Since the input for agent 0 is completely unknown, a reasonable and simple model for the leader is a ZOH model, that is, the leader will hold its latest transmitted measurement. Note that the leader only implements its own model, the ZOH model based on its transmitted measurements. Since the leader does not receive updates from any other agent it does not need to implement models of other agents. Under these circumstances, the local state error corresponding to the leader is given by

$$\varepsilon_0^{(0)}[k] = r_0[k] - q_0^{(0)}[k] \tag{11}$$

where $q_0^{(0)}[k] = r_0[k_\mu^0]$ for $k_\mu^0 \leq k < k_{\mu+1}^0$ and k_μ^0, $\mu = 1, 2, ...$ represents the update instants corresponding to agent 0, that is, the time instants when the leader transmits a measurement.

The control input for agent i is given as a function of the agent's own position r_i and the estimates of the positions of its neighbors as given by its own models. The control inputs $u_i[k]$ are given by

$$u_i[k] = -\sum_{j=1}^{n} a_{ij}(r_i[k] - \hat{r}_j^{(i)}[k]) - a_{i0}(r_i[k] - \hat{r}_0^{(i)}[k]) \tag{12}$$

where a_{ij} is the (i,j) entry of the adjacency matrix associated with the follower's communication graph, $a_{i0} > 0$ if the leader is a neighbor of follower i and $a_{i0} = 0$ otherwise. Each agent computes its own control input since it has access to its real position $r_i[k]$ and to the model states of its neighbors $\hat{r}_j^{(i)}[k]$.

The model control inputs used by agent l for each one of the models of its neighbors can be described by

$$\hat{u}_i^{(l)}[k] = -\sum_{j=1}^{n} a_{ij}(\hat{r}_i^{(l)}[k] - \hat{r}_j^{(l)}[k]) - a_{i0}(\hat{r}_i^{(l)}[k] - \hat{r}_0^{(l)}[k]). \tag{13}$$

The model control inputs (13) can be computed locally by each agent since every agent implements the models (4) corresponding to its neighbors. Eq. (13) is only a function of model variables.

The control inputs $\overline{u}_i^{(i)}[k]$ for any given model $q_i^{(i)}[k]$ are given by

$$\overline{u}_i^{(i)}[k] = -\sum_{j=1}^{n} a_{ij}(q_i^{(i)}[k] - q_j^{(i)}[k]) - a_{i0}(q_i^{(i)}[k] - q_0^{(i)}[k]). \tag{14}$$

The model control inputs (14) are used by each agent only to estimate its own position in order to compute the local error (6) and to determine the transmission instants.

We consider fixed and directed communication topologies. The only constraint in the communication graph is to have a spanning tree with root at the leader. Directed graphs with cycles are also admitted as long as the previous condition is satisfied. In order to explain the type of models that are implemented by each agent let us consider a simple example of a group of agents transmitting information using a directed tree type of graph that is shown in Fig. 1.

The leader (agent 0) only implements a 'q' model, Eq. (11). The followers need to implement models of its neighbors called 'r' models, and models of itself called 'q' models. For instance, the 'r' model of agent 3 is given by

$$\hat{r}^{(3)}[k+1] = \begin{bmatrix} \hat{r}_0^{(3)}[k+1] \\ \hat{r}_1^{(3)}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Ta_{10} & 1 - Ta_{10} \end{bmatrix} \begin{bmatrix} \hat{r}_0^{(3)}[k] \\ \hat{r}_1^{(3)}[k] \end{bmatrix}. \tag{15}$$

Eq. (15) represents the model of agent 1 (and a ZOH model of agent 0) as seen by agent 3. The 'q' model of agent 3 is given by

$$q^{(3)}[k+1] = \begin{bmatrix} q_1^{(3)}[k+1] \\ q_3^{(3)}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Ta_{31} & 1 - Ta_{31} \end{bmatrix} \begin{bmatrix} q_1^{(3)}[k] \\ q_3^{(3)}[k] \end{bmatrix}. \tag{16}$$

The real dynamics of agent 3 are given by

$$r_3[k+1] = (1 - Ta_{31}) r_3[k] + Ta_{31} \hat{r}_1^{(3)}[k] \tag{17}$$

and its local error is computed as follows:

$$\varepsilon_3^{(3)}[k] = r_3[k] - q_3^{(3)}[k]. \tag{18}$$

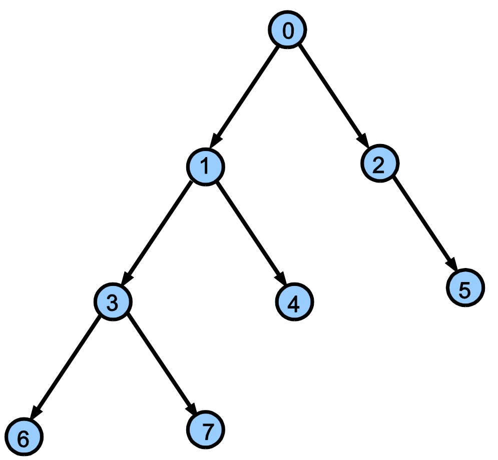

Fig. 1. A group of 7 followers tracking the position of a leader using a directed tree communication graph.

We can see that the 'r' model is used to compute the control input of the agent while the 'q' model is used to compute its error. Model (15) is updated with information received from the neighbors, in this case, from agent 1. Model (16) is updated when the local error is larger than the corresponding threshold and the agent, in this case agent 3, needs to broadcast a measurement which is received by agents 6 and 7 in this example. In both models, the secondary variables are modeled as ZOH, e.g. $\hat{r}_0^{(3)}$ in (15), and $q_1^{(3)}$ in (16).

The detailed updates for each agent are as follows: when the local error is larger than the threshold then the 'q' model is updated as follows. The first elements of $q^{(i)}$ are updated using the corresponding elements of $\hat{r}^{(i)}$. The last element of $q^{(i)}$ corresponds to the local measurement, therefore it is updated using the real measurement of agent i. The new state of the 'q' model is transmitted after these updates take place. Returning to the example of agent 3 in Fig. 1, at any time k_μ^3 we have the update $[q_1^{(3)}[k_\mu^3] \quad q_3^{(3)}[k_\mu^3]]^T = [\hat{r}_1^{(3)}[k_\mu^3] \quad r_3[k_\mu^3]]^T$ and this update is transmitted to agents 6 and 7 in the example in Fig. 1. Both agents use this information to update their 'r' models, for instance, agent 6 'r' model is given by

$$\hat{r}^{(6)}[k+1] = \begin{bmatrix} \hat{r}_1^{(6)}[k+1] \\ \hat{r}_3^{(6)}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ Ta_{31} & 1 - Ta_{31} \end{bmatrix} \begin{bmatrix} \hat{r}_1^{(6)}[k] \\ \hat{r}_3^{(6)}[k] \end{bmatrix} \tag{19}$$

and it is updated using the information received from agent 3, i.e. $[\hat{r}_1^{(6)}[k_\mu^3] \quad \hat{r}_3^{(6)}[k_\mu^3]]^T = [q_1^{(3)}[k_\mu^3] \quad q_3^{(3)}[k_\mu^3]]^T$.

Remark 2. Note that the 'r' model of every follower is equivalent to the 'q' model of its parent and if they are updated at similar times using the same information then they produce the same estimated variables.

Remark 3. When an agent receives an update from its neighbor that agent does not necessarily generate an event by itself at the same time instant since its local error may not be larger than the threshold. The overall communication policy is asynchronous and each agent transmits information to its neighbors at some time instants based only on local information.

This method of implementing models of neighbors only can be easily extended to cases where the communication graph is not a simple directed tree but it is a more general connected graph which can also contain cycles. The main difference is that each agent may need to implement more than one 'r' model depending on the number of neighbors. However, each agent only needs one 'q' model since there is only one local error to compute. The order of each of these models may increase with respect to the directed tree graph case due to the fact that each agent may have more than one neighbor.

The following is a simple set of rules to determine the number of models and the order of each one of those models for every agent.

a) Number of models 'r' implemented by agent $i = |N_i|$.
b) Order of each model $j \in N_i = 1 + |N_j|$.
c) Order of 'q' model of agent $i = 1 + |N_i|$.

where $|N_i|$ represents the number of neighbors of agent i. Let us consider an example of a more general graph with cycles. Consider a group of $N = 7$ followers labeled $i = 1, \ldots, 7$ plus a dynamic leader, agent 0. The adjacency graph corresponding to the interaction among the followers is given by

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 6 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 10 & 8 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 5 & 6 & 0 & 0 \end{bmatrix}. \tag{20}$$

In addition, the leader is a neighbor of followers 1 and 2, $a_{10}=4, \quad a_{20}=3$. In order to illustrate the selection of models following the rules established above, let us consider the dynamics of agent 1:

$$r_1[k+1] = r_1[k] - T(a_{15}(r_1[k] - \hat{r}_5^{(1)}[k]) + a_{10}(r_1[k] - \hat{r}_0^{(1)}[k])). \tag{21}$$

Agent 1 has two neighbors. It will implement two 'r' models. The neighbors of agent 1 are the leader and agent 5. The order of the first 'r' model is $1+|N_0|=1$, since the leader has no neighbors, i.e. the leader does not receive updates from any other vehicle in the network. The order of the second model is $1+|N_5|=3$. The models are given by

$$\hat{r}_0^{(1)}[k+1] = \hat{r}_0^{(1)}[k] \tag{22}$$

$$\begin{bmatrix} \hat{r}_1^{(1)}[k+1] \\ \hat{r}_2^{(1)}[k+1] \\ \hat{r}_5^{(1)}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Ta_{51} & Ta_{52} & 1 - Ta_{51} - Ta_{52} \end{bmatrix} \begin{bmatrix} \hat{r}_1^{(1)}[k] \\ \hat{r}_2^{(1)}[k] \\ \hat{r}_5^{(1)}[k] \end{bmatrix}. \tag{23}$$

Model (22) is updated when a measurement from the leader is received by agent 1. Similarly, model (23) is updated when agent 1 receives an update from agent 5. At first sight, it seems odd to have the estimate $\hat{r}_1^{(1)}[k]$ in (23) but we should remember that this helps to obtain a consistency of the model variables by different agents. Finally, the 'q' model of agent 1 is given by

$$\begin{bmatrix} q_0^{(1)}[k+1] \\ q_5^{(1)}[k+1] \\ q_1^{(1)}[k+1] \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ Ta_{10} & Ta_{15} & 1 - Ta_{10} - Ta_{15} \end{bmatrix} \begin{bmatrix} q_0^{(1)}[k] \\ q_5^{(1)}[k] \\ q_1^{(1)}[k] \end{bmatrix}. \tag{24}$$

Remark 4. This method to construct the models is adaptive in the sense that initially each agent only needs to know its own weights. From this information, it can immediately build its 'q' model. The agent will broadcast all elements of its 'q' model at any event time. The receiving agents will immediately know the number of neighbors of its parent from this update. If, in addition to transmitting the elements of 'q' each agent transmits its weights at the first time instant then every agent will be able to construct all 'r' models it needs. The structure of the matrices of all models is simple in the sense that only the last row contains specific information to each agent and its neighbors.

Define the tracking error for each follower as follows:

$$\xi_i[k] = r_i[k] - r_0[k]. \tag{25}$$

Let $\xi[k] = [\xi_1[k], ..., \xi_n[k]]^T$ denote the tracking errors in vector form.

Proposition 1. *The tracking error vector response is given by*

$$\xi[k] = Q^k \xi[0] - \sum_{l=1}^{k} Q^{k-l} T A_{aug} \varepsilon[l-1] + \sum_{l=1}^{k} Q^{k-l} x_p^r[l-1] \tag{26}$$

where $Q = (I_n - TL - T \cdot diag\{a_{10}, ..., a_{n0}\})$, $\varepsilon[k] = [\varepsilon_0^{(0)}[k], ..., \varepsilon_n^{(n)}[k]]^T$, $A_{aug} = [A_0 \ A]$, $A_0 = [a_{10}, ..., a_{n0}]^T$, $x_p^r[k] = (-r_0[k+1] + r_0[k])1_n$, *A and L are, respectively, the adjacency and the Laplacian matrices associated with the followers communication graph.*

Proof. The tracking error dynamics for follower i can be described by

$$\begin{aligned} \xi_i[k+1] &= r_i[k+1] - r_0[k+1] \\ &= T\left(- \sum_{j=1}^{n} a_{ij}(r_i[k] - \hat{r}_j^{(i)}[k]) - a_{i0}(r_i[k] - \hat{r}_0^{(i)}[k]) \right) + r_i[k] - r_0[k+1]. \end{aligned} \tag{27}$$

Note that using the MB-ET scheme proposed above we have that $\hat{r}_j^{(i)}[k] = q_j^{(j)}[k]$ and by using (6) we obtain

$$\begin{aligned} \xi_i[k+1] = \xi_i[k] - T \sum_{j=1}^{n} a_{ij}(\xi_i[k] - \xi_j[k] + \varepsilon_j^{(j)}[k]) - T a_{i0}(\xi_i[k] \\ + \varepsilon_0^{(0)}[k]) - (r_0[k+1] - r_0[k]). \end{aligned} \tag{28}$$

The tracking errors can be expressed in compact form as follows:

$$\begin{aligned} \xi[k+1] &= \begin{bmatrix} 1 - T\sum_{j=0}^{n} a_{1j} & T a_{12} & \dots & T a_{1n} \\ T a_{21} & 1 - T\sum_{j=0}^{n} a_{2j} & & \\ \vdots & & \ddots & \\ T a_{n1} & & & \end{bmatrix} \xi[k] - T \begin{bmatrix} a_{10} & 0 & a_{12} & \dots & a_{1n} \\ a_{20} & a_{21} & 0 & & \\ a_{30} & a_{31} & a_{32} & & \\ \vdots & & & & \\ a_{n0} & & & & \end{bmatrix} \varepsilon[k] + x_p^r[k] \\ &= (I_n - TL - T \cdot diag\{a_{10}, ..., a_{n0}\})\xi[k] - T A_{aug}\varepsilon[k] + x_p^r[k]. \end{aligned} \tag{29}$$

The response of (29) with initial conditions $\xi[0]$ and with inputs $\varepsilon[k]$, $x_p^r[k]$ can be directly described as in (26). ■

The multi-agent event-triggered approach described in this section can also be seen as a dynamically changing topology since communication links are used intermittently, that is, communication links appear and disappear. The main differences of our event-triggered communication approach with respect to frameworks such as the ones described in [23] and [24] are given by the time-based versus the event-based nature of the problems and also by the control approach. In general, results dealing with dynamically changing topologies make the control weights $a_{ij} = 0$ when agent i does not receive information from agent j. In contrast, we use an estimate of neighbors' states to compute the control inputs when the real state is not being transmitted. For instance, if we consider a leader and the 7 followers described by (20) then the

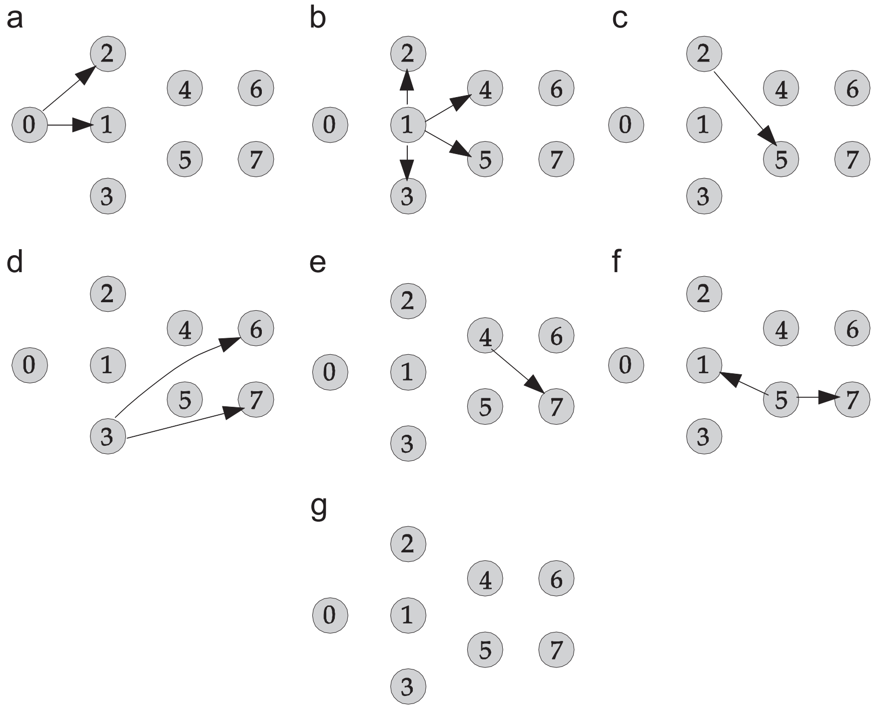

Fig. 2. Individual topologies of a Leader and 7 Followers.

individual topologies are shown in Fig. 2. We define the individual topologies as those topologies that contain links that begin from the same agent. The topology with zero active links (case ‘g’) is also defined to be an individual topology.

The finite set of topologies is composed of all individual topologies plus any possible combination of those topologies, since at any given time the possibility exists that more than one agent is able to transmit. Therefore, at each time k we can have different communication links appear and disappear, which effectively makes the communication graph a time-varying one. The variation in the communication graph is not based on time but based on events. For large thresholds we expect case ‘g’ in the example shown in Fig. 2 to appear very often (no communication links are established).

Another difference between the approach in this paper and [23], [24] is the use of neighbor's estimated states. When a physical link disappears, agents still make use of some (uncertain) information about neighbors to compute their local input. Another way to think about this issue is when a physical link disappears, an “hypothetical” link is created locally for an agent to use its neighbors uncertain and outdated state information. Overall, the practical advantage of the multi-agent event-triggered approach is that it saves communication resources by specifying a way to establish (directed) communication links only when it is necessary.

Theorem 2. *Assume that the union of individual topologies has a directed spanning tree with root at the leader. Also assume that the continuous time leader's position* $r_0(t)$ *satisfies the following:*

$$|\dot{r}_0(t)| \leq \bar{r} \tag{30}$$

for any $T>0$, $|r_0(t+T)-r_0(t)| \le \bar{r}T$ *holds, which is equivalent to (when considering T as the discretization sampling interval)*

$$\left|\frac{r_0[k]-r_0[k-1]}{T}\right| \le \bar{r} \tag{31}$$

then, the maximum tracking error of the n followers is ultimately bounded by

$$(\alpha||A_{aug}||_\infty + \bar{r})||(L + diag\{a_{10}, ..., a_{n0}\})^{-1}||_\infty \tag{32}$$

when T is designed to satisfy

$$T < \min_{i=1,...,n} 1 / \sum_{j=0}^{n} a_{ij}. \tag{33}$$

Proof. From (26) the norm of the tracking error can be expressed as

$$||\xi[k]||_\infty \le ||Q^k||_\infty ||\xi[0]||_\infty + \alpha T ||A_{aug}||_\infty || \sum_{l=0}^{k-1} Q^l ||_\infty + T\bar{r} || \sum_{l=0}^{k-1} Q^l ||_\infty \tag{34}$$

where (10) and (31) have been used. Since the leader has directed paths to all followers and $0 < T < \min_{i=1,...,n} 1/\sum_{j=0}^{n} a_{ij}$ then, by Lemma 8.3 in [8], Q has all its eigenvalues within the unit circle and $\lim_{k\to\infty} Q^k = 0$. Additionally, from Lemma 1.26 and Lemma 1.28 in [8], we have that

$$\lim_{k\to\infty} || \sum_{l=0}^{k-1} Q^l ||_\infty = ||(I_n - Q)^{-1}||_\infty \tag{35}$$

and the maximum tracking error of the n followers, $||\xi[k]||_\infty$, is ultimately bounded by (32). ■

Remark 5. In Theorem 2 and similar to the results in [7,8] T has to meet certain conditions for stability, i.e. it has to be small enough so the overall system does not become unstable. Using the MB-ET framework we are able to reduce T to obtain a stable matrix Q but we do not need to communicate every T seconds. We are now able to avoid frequent communication by choosing a desired error threshold α while maintaining a stable overall system.

Remark 6. An additional advantage of the framework presented in this paper compared to the sampled-data approach [7,8] is that the agents are not required to transmit measurements to neighbors at the same time instants, i.e. at the sampling instants kT, $k=0,1,\ldots$ In the sampled-data approach the agents are not required to communicate continuously but all of them use the same period T to sample and transmit their measurements. The communication scheme in the present paper allows for asynchronous and decentralized communication in the sense that each agent determines its own time instants at which it sends a measurement update.

Remark 7. For fixed control weights a_{ij} and a fixed period T, the tradeoff between reduction of communication and performance as measured by the tracking error is a function of $\alpha>0$. By increasing α we are able to reduce communication between agents but the tracking error will increase in general. Smaller tracking errors can be obtained by decreasing α at the cost of increased communication. The case when α is the same for all agents simplifies the analysis. However, each agent can have a different value $\alpha_i>0$ and boundedness of the tracking error can be easily shown by using $\alpha = \max_i(\alpha_i)$.

4. Leader tracking with noisy measurements

It was assumed in the previous section that every agent is able to measure its own position with infinite precision. When considering the control and position measurement of unmanned vehicles it is typical that the true position of a vehicle can be determined only within a precision range which is usually described by $y_i[k] = r_i[k] \pm a$, where a is estimated based on the measurement method and quality of the measurement devices. The objective in this section is to provide a bound on the tracking error considering the presence of this measurement uncertainty. In particular we use the following representation:

$$y_i[k] = r_i[k] + w_i[k] \tag{36}$$

where $w_i[k]$ is a measurement noise with uniform distribution in $[-a,a]$. Since $y_i[k]$ is the available measurement to be used when a measurement update is triggered and also to be used for local control we have that

$$q_i^{(i)}[k_\mu^i] = y_i[k_\mu^i] = r_i[k_\mu^i] + w_i[k_\mu^i] \tag{37}$$

and

$$u_i[k] = -\sum_{j=1}^{n} a_{ij}(y_i[k] - \hat{r}_j^{(i)}[k]) - a_{i0}(y_i[k] - \hat{r}_0^{(i)}[k]). \tag{38}$$

The local state errors are now defined by

$$\overline{\varepsilon}_i^{(i)}[k] = y_i[k] - q_i^{(i)}[k], \quad i = 0, 1, 2, ..., n. \tag{39}$$

Since Eq. (39) is the error that can be monitored, events are triggered when the following is satisfied:

$$|\overline{\varepsilon}_i^{(i)}[k]| > \alpha. \tag{40}$$

When an agent broadcasts its current measurement the error is equal to zero ($\overline{\varepsilon}_i^{(i)}[k_\mu^i] = 0$) at that time instant because of the update (37). Then, because of the triggering condition (40), the following holds:

$$|\overline{\varepsilon}_i^{(i)}[k]| \le \alpha, \quad i = 0, 1, 2, ..., n. \tag{41}$$

Using the definition for the tracking error as in Eq. (25) we have the following.

Proposition 3. *The tracking error vector response is given by*

$$\xi[k] = Q^k \xi[0] - \sum_{l=1}^{k} Q^{k-l} T(A_{aug}\overline{\varepsilon}[l-1] + A_w w[l-1]) + \sum_{l=1}^{k} Q^{k-l} x_p^r[l-1] \tag{42}$$

where

$$Q = (I_n - TL - T \bullet diag\{a_{10}, ..., a_{n0}\}), \;\; A_{aug} = [A_0 \;\; A],$$
$$A_w = A_{aug} + \left[0_n \quad diag\left\{ \sum_{j=0}^{n} a_{1j}, ..., \sum_{j=0}^{n} a_{nj} \right\} \right],$$
$$A_0 = [a_{10}, ..., a_{n0}]^T, \; \overline{\varepsilon}[k] = [\overline{\varepsilon}_0^{(0)}[k], ..., \overline{\varepsilon}_n^{(n)}[k]]^T, \; w[k] = [w_0[k], ..., w_n[k]]^T,$$
$$x_p^r[k] = (-r_0[k+1] + r_0[k])1_n$$

A and L are, respectively, the adjacency and the Laplacian matrices associated with the followers communication graph.

Proof. The tracking error dynamics for follower i can be described by

$$\xi_i[k+1] = r_i[k+1] - r_0[k+1]$$
$$= T\left(-\sum_{j=1}^{n} a_{ij}(r_i[k] - \hat{r}_j^{(i)}[k] + w_i[k]) - a_{i0}(r_i[k] - \hat{r}_0^{(i)}[k] + w_i[k]) \right) + r_i[k] - r_0[k+1]. \tag{43}$$

Similar to the noise free case in the previous section we have $\hat{r}_j^{(i)}[k] = q_j^{(j)}[k]$ and by using Eq. (39) we obtain

$$\xi_i[k+1] = \xi_i[k] - T\sum_{j=1}^{n} a_{ij}(\xi_i[k] - \xi_j[k] + \bar{\varepsilon}_j^{(j)}[k] + w_i[k] + w_j[k])$$
$$- Ta_{i0}(\xi_i[k] + \bar{\varepsilon}_0^{(0)}[k] + w_i[k] + w_0[k]) - (r_0[k+1] - r_0[k]) \tag{44}$$

The tracking errors can be expressed in compact form as follows:

$$\xi[k+1] = Q\xi[k] - T(A_{aug}\bar{\varepsilon}[k] + A_w w[k]) + x_p^r[k]. \tag{45}$$

The response of Eq. (45) with initial conditions $\xi[0]$ and with inputs $\bar{\varepsilon}[k], w[k], \ x_p^r[k]$ can be directly described as in (42). ■

Corollary 4. *Assume that the union of individual topologies has a directed spanning tree with root at the leader. Also assume that the continuous time leader's position $r_0(t)$ satisfies (30).*

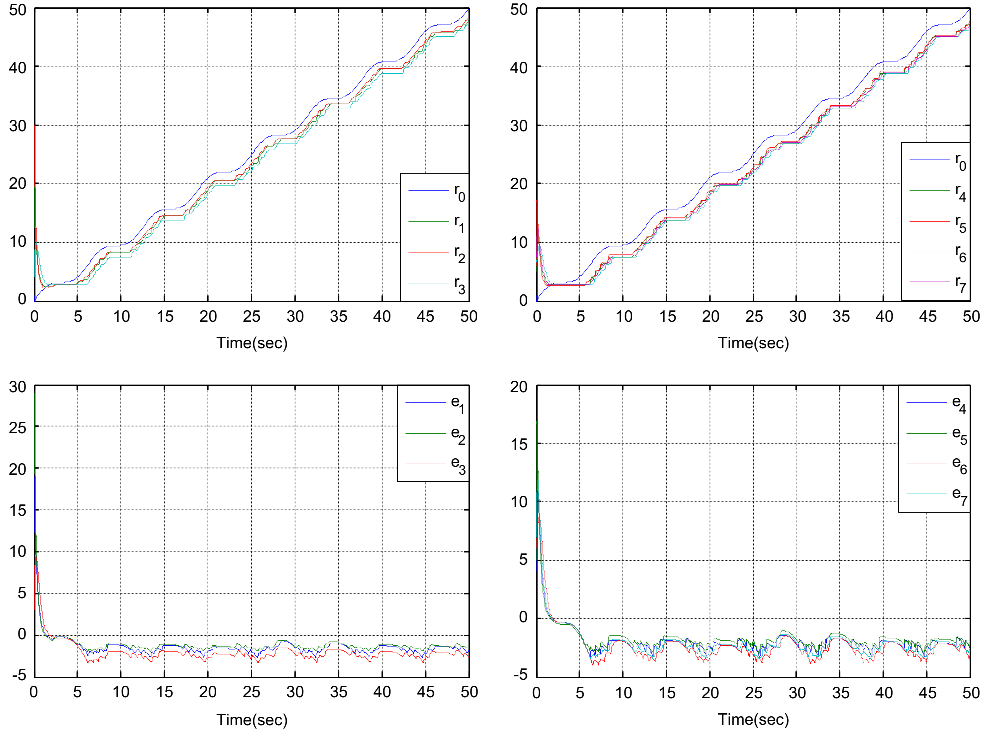

Fig. 3. Positions and tracking errors for MB-ET leader follower with time-varying leader velocity and using a connected communication graph with cycles.

Then, the maximum tracking error of the n followers is ultimately bounded by

$$(\alpha||A_{aug}||_\infty + a||A_w||_\infty + \overline{r})||(L + diag\{a_{10}, ..., a_{n0}\})^{-1}||_\infty \tag{46}$$

when T is designed to satisfy (33).

Proof. The proof is similar to the proof of Theorem 2 by including the new term containing the measurement noise. ■

5. Examples

Example 1. Consider a group of $n=7$ vehicles, $i=1,\ldots,7$, following the position of a leader (Agent 0) and transmitting information according to the communication graph specified by (20). The sampling time is $T=0.01$ s and the threshold is $\alpha=1$. Simulation results are shown in Fig. 3.

Example 2. Consider the same group of vehicles as in Example 1. In this example we would like to analyze the maximum tracking error for different threshold values. In this case we let the leader move using a constant maximum velocity, i.e. (31) holds with equality and $\overline{r}=1$. The sampling time is still $T=0.01$ s. Fig. 4 shows the absolute value of the maximum tracking error for four different choices of α. It can be seen that the steady state maximum tracking error is bounded by the corresponding theoretical bound obtained from Theorem 2, Eq. (32), using the particular choice of α. The theoretical bound for each threshold value is shown in Table 1 and 2. In addition, Table 1 shows information related to the number of measurements transmitted by each agent for each one of the four simulations in Fig. 4. This information is shown as a

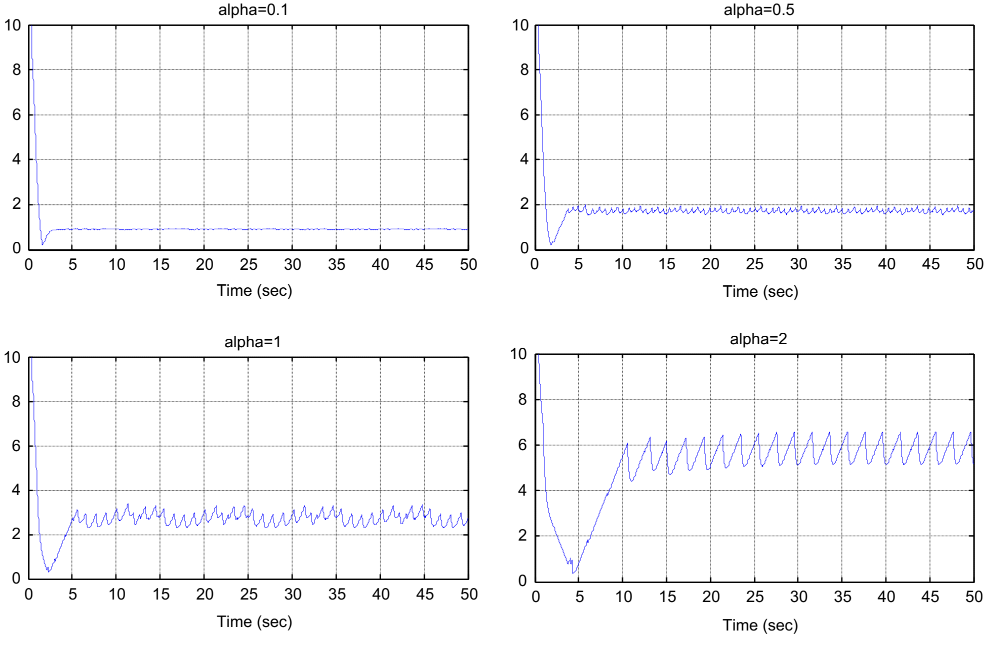

Fig. 4. Maximum tracking error for MB-ET leader follower for different threshold values.

Table 1
Theoretical bound for different threshold values and percentage of transmissions (with respect to the periodic implementation) by each agent.

α	Bound	Percentage of transmissions by each agent							
		0	1	2	3	4	5	6	7
0.1	1.96	9.38	5.88	4.08	4.06	5.84	6.08	4.12	6.02
0.5	7	2.00	1.64	1.52	1.58	1.60	1.70	1.54	1.66
1	13.3	0.98	0.96	0.92	0.92	0.94	1.00	0.88	0.98
2	25.9	0.48	0.48	0.50	0.48	0.50	0.52	0.42	0.46

Table 2
Theoretical bound for MB-ET leader follower with noisy measurements for different threshold values and percentage of transmissions (with respect to the periodic implementation) by each agent.

α	Bound	Percentage of transmissions by each agent							
		0	1	2	3	4	5	6	7
0.1	9.52	69.02	68.98	70.56	69.16	69.74	69.32	69.88	69.24
0.5	14.56	2.70	3.22	2.82	3.28	3.38	3.36	2.52	3.28
1	20.86	0.94	1.06	0.98	1.02	1.04	1.14	1.04	1.10
2	33.46	0.48	0.50	0.50	0.50	0.52	0.54	0.44	0.50

percentage of the total number of transmissions that a sampled-data periodic scheme will produce using the same sampling time T, i.e. a communication policy that requires every agent to transmit its measurement every time k.

In this example, every simulation is executed for 50 s which results, using $T=0.01$ s, in 5000 transmissions (100%) by each agent. An important reduction of transmissions can be obtained using the MB-ET approach. It can be seen from Fig. 4 and Table 1 that as we increase the threshold the ultimate bound is increased and the number of transmissions is reduced resulting in a tradeoff between desired tracking performance and use of communication resources.

Note that using the periodic sampled-data approach with no models the system becomes unstable as we increase the sampling time in order to reduce communication among agents. From [7] a sufficient condition for bounded tracking error is given by (33). In this example, condition (33) results in $T<0.0556$ s. From simulations we obtain that the system is stable for approximately $T<0.09$ s, which means that the model-based approach described in this paper is able to obtain a greater reduction of messages transmitted while keeping the tracking error bounded.

Example 3. A similar simulation analysis to Example 2 is performed. The difference in this case is the presence of measurement noise. We consider uniform noise with $\alpha=0.3$. Fig. 5 shows the maximum tracking error for different threshold values. The steady state maximum tracking error is bounded by the theoretical bound obtained in (46). It can be seen that, in general, this is a conservative bound.

An important difference compared to the previous example is the reduction of transmissions. For the first case we have that $\alpha=0.1$ is a small threshold compared to the magnitude of the noise. The results in this paper hold and the tracking error is bounded as predicted but the

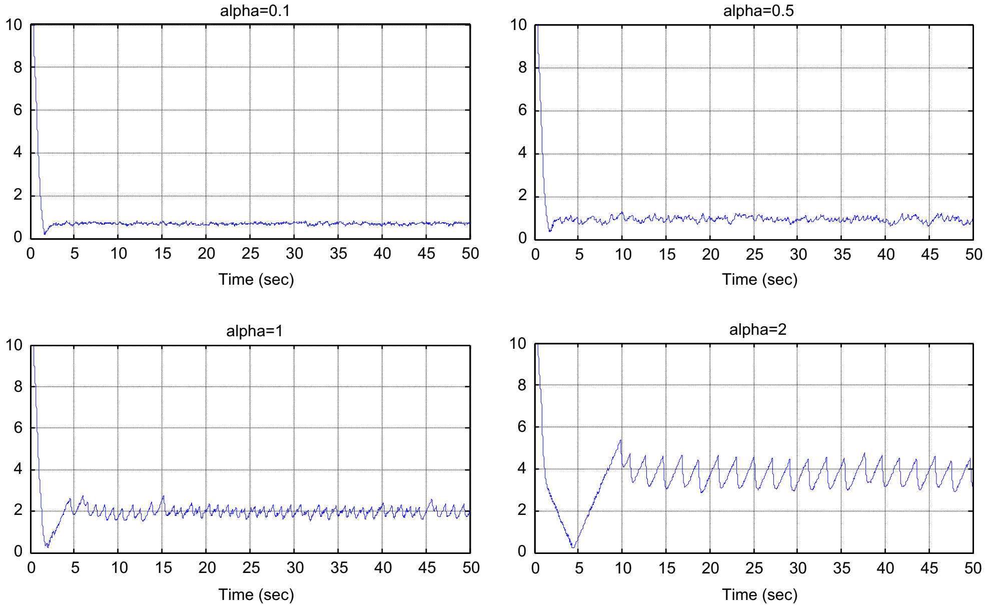

Fig. 5. Maximum tracking error for MB-ET leader follower with noisy measurements for different threshold values.

reduction of messages transmitted compared to the periodic case is small. When the threshold α is greater than the bound on the measurement noise the reduction of communication is similar to the case when noise free measurements are transmitted.

6. Conclusions

Coordinated multi-vehicle leader tracking has been studied using sampled-data approaches. Typical results require a small enough sampling time for bounded tracking error otherwise the overall system becomes unstable. The approach in this paper extends previous work by using event-triggered control and a model-based approach that provides estimates of positions of other agents. The overall framework is able to reduce transmission of information among agents while maintaining a bounded tracking error. Ultimate bounds were obtained for the cases of noise free and noisy measurements. Simulation examples show the effectiveness of this approach for the two cases described in the paper. In this work we studied the P-like controller; future research will consider a Proportional-Derivative controller that may reduce the tracking error while keeping a low usage of communication resources.

References

[1] L. Moreau, Stability of continuous-time distributed consensus algorithms, in: Proceedings of the 43rd IEEE Conference on Decision and Control, 2004, pp. 3998–4003.

[2] W. Ren, R.W. Beard, E.M. Atkins, Information consensus in multivehicle cooperative control, IEEE Control Syst. Mag. 27 (2) (2007) 71–82.

[3] T. Hayakawa, T. Matsuzawa, S. Hara, Formation control of multiagent systems with sampled information, in: Proceedings of the 45th IEEE Conference on Decision and Control, 2006, pp. 4333–4338.

[4] Y. Gao, L. Wang, Consensus of multiple double integrator agents with intermittent measurements, Int. J. Robust Nonlinear Control 20 (10) (2010) 1140–1155.
[5] J. Qin, H. Gao, A sufficient condition for convergence of sampled-data consensus for double-integrator dynamics with nonuniform and time-varying communication delays, IEEE Trans. Autom. Control 57 (9) (2012) 2417–2422.
[6] H. Liu, G. Xie, L. Wang, Necessary and sufficient conditions for solving consensus of double integrator dynamics via sample control, Int. J. Robust Nonlinear Control 20 (15) (2010) 1706–1722.
[7] Y. Cao, W. Ren, Y. Li, Distributed discrete-time coordinated tracking with time-varying reference state and limited communication, Automatica 45 (2009) 1299–1305.
[8] W. Ren, Y. Cao, Distributed Coordination of Multi-Agent Networks: Emergent Problems, Models, and Issues, Springer-Verlag, London, 2011.
[9] Y. Cao, W. Ren, Sample-data formation control under dynamic directed interaction, in: Proceedings of the 2009 American Control Conference, 2009, pp. 5186–5191.
[10] Y. Cao, W. Ren, Multi-vehicle coordination for double-integrator dynamics under fixed undirected/directed interaction in a sampled data setting, Int. J. Robust Nonlinear Control 20 (9) (2010) 987–1000.
[11] E. Garcia, P.J. Antsaklis, Model-based event-triggered control with time-varying network delays, in: Proceedings of the 50th IEEE Conference on Decision and Control-European Control Conference, 2011, pp. 1650–1655.
[12] E. Garcia, P.J. Antsaklis, Model-based event-triggered control for systems with quantization and time-varying network delays, IEEE Trans. Autom. Control 58 (2) (2013) 422–434.
[13] X. Yin, D. Yue, S. Hu, Distributed event-triggered control of discrete-time heterogeneous multi-agent systems, J. Frankl. Inst. 350 (3) (2013) 651–669.
[14] D.V. Dimarogonas, E. Frazzoli, K.H. Johansson, Distributed event-triggered control for multi-agent systems, IEEE Trans. Autom. Control 57 (5) (2012) 1291–1297.
[15] D.V. Dimarogonas, K.H. Johansson, Event-triggered control of multi-agent systems, in: Joint IEEE Conference on Decision and Control and 28th Chinese Control Conference, 2009, pp. 7131–7136.
[16] M. Guo, D.V. Dimarogonas, Nonlinear consensus via continuous, sampled, and aperiodic updates, Int. J. Control 86 (4) (2013) 567–578.
[17] X. Chen, F. Hao, Event-triggered average consensus control for discrete-time multi-agent systems, IET Control Theory Appl. 6 (16) (2012) 2493–2498.
[18] H. Yu, P.J. Antsaklis, Quantized output synchronization of networked passive systems with event-driven communication, in: Proceedings of the 2012 American Control Conference, 2012, pp. 5706–5711.
[19] E. Garcia, Y. Cao, H. Yu, P.J. Antsaklis, D.W. Casbeer, Decentralized event-triggered cooperative control with limited communication, Int. J. Control 86 (9) (2013) 1479–1488.
[20] E. Garcia, Y. Cao, D.W. Casbeer, Model-Based Event-Triggered Multi-Vehicle Coordinated Tracking with Communication Delays, No. AFRL-RQ-WP-TP-2013-0084, Air Force Research Laboratory, Wright Patterson AFB, OH, Aerospace Systems Directorate, 2013.
[21] E. Garcia, P.J. Antsaklis, Decentralized model-based event-triggered control of networked systems, in: Proceedings of the 2012 American Control Conference, 2012, pp. 6485–6490.
[22] Y. Sun, N.H. El-Farra, Dynamic quasi-decentralized control of networked process systems with limited measurements, in: Proceedings of the 2011 American Control Conference, 2011, pp. 3152–3157.
[23] Y. Cao, W. Ren, Sampled-data discrete-time coordination algorithms for double-integrator dynamics under dynamic directed interaction, Int. J. Control 83 (3) (2010) 506–515.
[24] J. Qin, H. Gao, C. Yu, On discrete-time convergence for general linear multi-agent systems under dynamic topology, IEEE Trans. Autom. Control 59 (4) (2014) 1054–1059.
[25] E. Garcia, Y. Cao, D.W. Casbeer, Cooperative control with general linear dynamics and limited communication: centralized and decentralized event-triggered control strategies, in: Proceedings of the American Control Conference, 2014.