Automatica 73 (2016) 1–7

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Distributed model based event-triggered control for synchronization of multi-agent systems ☆

Davide Liuzza [a], Dimos V. Dimarogonas [a], Mario di Bernardo [b,c], Karl H. Johansson [a]

[a] ACCESS Linnaeus Center and School of Electrical Engineering, Royal Institute of Technology (KTH), Stockholm, Sweden

[b] Department of Engineering Mathematics, University of Bristol, UK

[c] Department of Electrical Engineering and Information Technology, University of Naples Federico II, Italy

ARTICLE INFO

Article history:
Received 9 October 2013
Received in revised form
19 February 2016
Accepted 31 May 2016

Keywords:
Event-triggered control
Synchronization
Multi-agent systems

ABSTRACT

This paper investigates the problem of event-triggered control for the synchronization of networks of nonlinear dynamical agents; distributed model-based approaches able to guarantee the synchronization of the overall system are derived. In these control schemes all the agents use a model of their neighbourhood in order to generate triggering instants in which the local controller is updated and, if needed, local information based on the adopted control input is broadcasted to neighbouring agents. Synchronization of the network is proved and the existence of Zeno behaviour is excluded; an event-triggered strategy able to guarantee the existence of a minimum lower bound between inter-event times for broadcasted information and for control signal updating is proposed, thus allowing applications where both the communication bandwidth and the maximum updating frequency of actuators are critical. This idea is further extended in an asynchronous periodic event-triggered schemes where the agents check a trigger condition via a periodic distributed communication without requiring a model based computation.

1. Introduction

The problem of controlling a multi-agent system to reach some coordinated behaviour has been widely exploited in the literature. Specifically, synchronization of dynamical systems has been investigated as a paradigm for more specific behaviours like consensus algorithms and platooning and formation control (Arcak, 2007; Olfati-Saber, Fax, & Murray, 2007).

Distributed control algorithms for multi-agent systems have often been realized in continuous time. However, continuous time control laws for such kind of networked systems are not easy or even impossible to implement in real applications where a wireless medium is often exploited to enact the communication.

In order to save the bandwidth and avoid unnecessary updating, the case of event-triggered communication (Tabuada, 2007) among single and double networked integrators has been studied in the recent literature, e.g. Dimarogonas, Frazzoli, and Johansson (2012) and Seyboth, Dimarogonas, and Johansson (2013).

Studies on synchronization of linear systems under an event-triggered framework can be found in Guinaldo, Dimarogonas, Johansson, Sánchez, and Dormido (2011) and Liu, Cao, Persis, and Hendrickx (2013) where the control signals are continuous in time and are generated via a model based approach while the communication signals are piecewise constant and based on the error between the real state and the uncoupled model state. Synchronization of linear systems has also been investigated in Liu, Hill, and Liu (2013), although the absence of Johansson, Egerstedt, Lygeros, and Sastry (1999) is not proved, while in De Persis (2013) a self-triggered approach is exploited in order to compute the next triggering instant.

In this paper we study a novel scheme for distributed event-triggered control able to guarantee synchronization of nonlinear multi-agent systems by using distributed information related to each pair of connected agents. The relative information on the state mismatch between each pair of connected agents will be considered, in order to generate local events and update the control law. The proposed idea follows a model-based approach, where

☆ This work was supported by the Swedish Research Council (VR) and the Knut and Alice Wallenberg Foundation. The material in this paper was partially presented at the 9th IFAC Symposium on Nonlinear Control Systems, September 4–6, 2013, Toulouse, France. This paper was recommended for publication in revised form by Associate Editor Claudio De Persis under the direction of Editor Christos G. Cassandras. The second author is also affiliated with the KTH Centre for Autonomous Systems.

E-mail addresses: liuzza@kth.se (D. Liuzza), dimos@kth.se (D.V. Dimarogonas), m.dibernardo@bristol.ac.uk (M. di Bernardo), kallej@kth.se (K.H. Johansson).

http://dx.doi.org/10.1016/j.automatica.2016.06.011
0005-1098/

each agent is equipped with its own embedded processor and it is assumed to know the dynamical model of its neighbours, and to predict their state evolutions between any two consecutive triggering events. Both the control and the communication signals will be piecewise constant and, specifically, neighbouring nodes will exchange information about their current (piecewise constant) control input. Such information will allow each node to predict the evolution of its neighbours and evaluate a trigger condition. The proposed scheme solves the problem of achieving synchronization of the interconnected nonlinear systems while guaranteeing a nonzero lower bound for the inter-event time. The existence of such a bound is a stronger result than proving simply the absence of Zeno behaviour, which only excludes accumulation point over a finite time, but does not prevent triggers to get infinitesimally close in time. This advantage allows applications where both the communication bandwidth and the maximum updating frequency of actuators are critical. Furthermore, it also allows the development of an asynchronous periodic event-triggered strategy, where the agents check periodically a trigger condition and decide whether or not to update their control input. In this case, no computations based on the model are needed. Such periodic event-triggered scheme represents the other major contribution of this work.

For the sake of brevity, we omit a background section on algebraic graph theory. For more details we refer the reader to Godsil and Royle (2001).

2. Model-based event-triggered control

Consider N identical dynamical agents of the form:

$$\dot{x}_i = f(t, x_i) + u_i, \quad x_i, u_i \in \mathbb{R}^n,\ t \geq 0,\ \forall i = 1, \ldots, N. \tag{1}$$

The aim is to guarantee the emergence of coordinated collective motion (synchronization) of all the agents by considering a distributed event-triggered control law. More precisely, the average trajectory is defined as

$$\bar{x}(t) = \frac{1}{N}\sum_{j=1}^{N} x_j(t), \tag{2}$$

and the synchronization errors as $e_i(t) = x_i(t) - \bar{x}(t)$, which in stack vector form corresponds to $e(t) = \left(e_1^T(t), \ldots, e_N^T(t)\right)^T \in \mathbb{R}^{nN}$. We want to achieve either one of the following two objectives:

Bounded synchronization. There exists an arbitrarily small $\epsilon > 0$ such that $\lim_{t\to\infty} \sup \|e(t)\|_2 \leq \epsilon$;
Complete synchronization. $\lim_{t\to\infty} \|e(t)\|_2 = 0$.

The setup upon which the synchronization analysis will be conducted in Section 3 is now described. Specifically, we assume that each agent is able to exchange information with a subset of the other agents. The resulting communication network, which for the sake of simplicity is assumed to be bidirectional, can be described by an undirected adjacency matrix $A = [a_{ij}]$ defined in the usual way. Furthermore, we assume that each agent is equipped with its own embedded processor able to execute a local control law based on the prediction of the evolution of its neighbours. Thanks to this local information, each node will execute an event-triggered update of its controller. In particular, at each node i we associate:

(1) a time sequence, $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty} : \mathbb{N} \mapsto [0, +\infty)$, of events corresponding to node i receiving information from node j, where $a_{ij} \neq 0$ and k^{ij} is the index of the sequence related to the pair (i, j);
(2) a time sequence, $\{t_{k^i}\}_{k^i=0}^{\infty} : \mathbb{N} \mapsto [0, +\infty)$, of instants when node i updates its control input $u_i(t)$, with k^i being the index of the sequence related to the updating of $u_i(t)$.

For any index $k^{ij} \in \mathbb{N}$ (or $k^i \in \mathbb{N}$) we have that $t_{k^{ij}} \leq t_{k^{ij}+1}$ (or $t_{k^i} \leq t_{k^i+1}$).

For each sequence $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ we introduce the *last function* $l^{ij}(t) : [0, +\infty) \mapsto \mathbb{N}$ defined as $l^{ij}(t) = \arg\min_{k^{ij}\in\mathbb{N}: t\geq t_{k^{ij}}} \left\{t - t_{k^{ij}}\right\}$. So, for each time instant t, $t_{l^{ij}(t)}$ is the most recent event occurred to i with respect to j, while with $t_{l^{ij}(t)+1}$ we indicate the next event.

Analogously, we define the function $l^i(t)$ for the sequence $\{t_{k^i}\}_{k^i=0}^{\infty}$.

As will be clear in what follows, the last indices $l^{ij}(t)$ and $l^i(t)$ will be used to generate iteratively the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^i}\}_{k^i=0}^{\infty}$.

Note that, although the communication graph is undirected, events related to coupled pairs (i, j) are, in general, not synchronous, so $t_{l^{ij}(t)} \neq t_{l^{ji}(t)}$. For this reason, the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^{ji}}\}_{k^{ji}=0}^{\infty}$ are generally different. For the sake of brevity, in what follows we will often omit the explicit dependence of l^{ij} and l^i on time.

The updating law of the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^i}\}_{k^i=0}^{\infty}$ will be described in detail in Section 3. Here we anticipate that, for each node i, the control u_i is updated (and so a new event in the sequence $\{t_{k^i}\}_{k^i=0}^{\infty}$ is generated) any time a new event on a connected pair (i, j) happens, i.e., every time there is a new event on one of the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$, with $j \in \mathcal{N}_i$. So, the latter are subsequences of $\{t_{k^i}\}_{k^i=0}^{\infty}$.

3. Event-triggered synchronization

In the setup we introduced, each node knows the dynamical model and the value of the initial conditions of its neighbours (or the value of their state at a specific time instant, for example at the first trigger). Therefore, each node i can compute from any event at time $t_{k^{ij}}$ the flow $\varphi_f(t - t_{k^{ij}}, t_{k^{ij}}, x_j(t_{k^{ij}}))$, $\forall j \in \mathcal{N}_i$. Note that in order to evaluate it, node i must also have information on the current control input $u_j(t)$ acting on each of its neighbours. Later, an algorithm able to guarantee that this information is shared among nodes will be presented. However, we firstly focus on the triggering events occurring at a generic node i.

For all pairs $(i, j) \in \mathcal{E}$ we define the *trigger error*

$$\tilde{e}_{ij}(t) := e_{ij}(t_{l^{ij}}) - e_{ij}(t), \quad t \in [t_{l^{ij}}, t_{l^{ij}+1}), \tag{3}$$

where $e_{ij}(t) = x_j(t) - x_i(t)$.

The error in (3) is referred to the last and the future trigger instants and is used, as will be clear in what follows, to compute the future trigger instant $t_{l^{ij}+1}$. Similarly $\tilde{e}_{ji}(t)$ is defined for the pair (j, i). Note that, as mentioned earlier, events referred to node i with respect to j are, in general, not synchronous with the events referred to j with respect to i. Indeed, as will be clear in what follows, in general $t_{l^{ij}} \neq t_{l^{ji}}$ since such time instants depend on the whole neighbourhood of node i and j respectively. For this reason, the pair (i, j) is treated here as a directed link and, in general, $\tilde{e}_{ij}(t) \neq \tilde{e}_{ji}(t)$. For all pairs (i, j), we also define the *trigger function* as $\Xi_{ij}(t, \tilde{e}_{ij}(t)) = \|\tilde{e}_{ij}(t)\|_2 - \varsigma_{ij}(t)$, where $\varsigma_{ij}(t)$ is a continuous-time non-increasing *threshold function* (particular choices of such function will be later considered and analysed). Then, an event occurs when the following condition is violated

$$\Xi_{ij}(t, \tilde{e}_{ij}(t), \varsigma_{ij}(t)) < 0. \tag{4}$$

For a generic agent i, the sequences $\{t_{k^{ij}}\}_{k^{ij}=0}^{\infty}$ and $\{t_{k^i}\}_{k^i=0}^{\infty}$ are generated by Algorithm 1 given below, as well as the piecewise constant control input $u_i(i)$, whose value at each update is computed as in (5), with $c > 0$ being a *coupling gain* and $\Gamma = \Gamma^T > 0$ being the *inner coupling matrix*. Such algorithm is run independently at each node of the network. Note that, as every node that triggers changes its control input and broadcasts it to