Brief paper

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

CrossMark

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

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.

© 2016 Elsevier Ltd. All rights reserved.

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

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).

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:

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

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

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

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 $l^{ij}(t) = \arg \min_{k^{ij} \in \mathbb{N}: t \geq t_{k^{ij}}} \{t - t_{k^{ij}}\}$. 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*

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

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

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

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

its neighbours (line 10), then all the nodes $j \in \mathcal{N}_i$ can update their dynamic model of i taking into account the new input $u_i(t_{l^i})$ and the current state $x_i(t_{l^i})$ (line 3). So, they will always be able to evaluate the correct value of the flow $\varphi_f(\cdot)$ of node i. Notice also that, since the control input $u_i(t)$ is a piecewise constant function, node i does not need to transmit such information continuously in time, but only when there is a change in its current value.

The initialization of Algorithm 1 happens when at least one node sends the triplet $(t_{0^i}, x_i(t_{0^i}), u_i(t_{0^i}))$ to its neighbours, with t_{0^i} being the time instant when the generic node i broadcasts for the first time its triplet. Then, having received the value of the triplet, all the neighbours can start predicting its evolution and, at the same time, broadcasting their triplets to the transmitting node and to their neighbours. In this way all the nodes of the network can be connected in a finite time. Notice that condition (4) is always verified when a node joins for the first time the network since it computes the first synchronizing control input using the state information coming from its neighbours.

Algorithm 1 Event-triggered update

1: **loop**
2:     Integrate the dynamical model $\dot{x}_j = f(t, x_j(t)) + u_j(t_{l^j})$ from the initial condition $x_j(t_{l^j})$ for all the neighbouring nodes $j \in \mathcal{N}_i$, while listening to possible transmission from them;
3:     **if** a new value $u_h(t_{l^h})$, with $h \in \mathcal{N}_i$, is received **then** update the dynamical model $\dot{x}_h = f(t, x_h(t)) + u_h(t_{l^h})$;
4:     **end if**
5:     **if** condition (4) is violated for a node $h \in \mathcal{N}_i$ at a time instant t^* **then**
6:         $l^i \leftarrow l^i + 1$ and $t_{l^i} \leftarrow t^*$;
7:         $l^{ih} \leftarrow l^{ih} + 1$ and $t_{l^{ih}} \leftarrow t^*$;
8:         $e_{ih}(t_{l^{ih}}) \leftarrow e_{ih}(t = t^*)$;
9:         $\tilde{e}_{ih} \leftarrow 0$
10:     Update the control input to the value

$$u_i(t) = c \sum_{j=1}^{N} a_{ij} \Gamma e_{ij}(t_{l^{ij}}), \qquad t \in [t_{l^i}, t_{l^i+1}), \tag{5}$$

    and broadcast u_i to the neighbourhood $\mathcal{N}_i$;
11:     **end if**
12: **end loop**

Remark 2. If we choose $\varsigma_{ij}(t) = \varsigma_{ji}(t)$, Algorithm 1 guarantees that when node i triggers and updates its control, node j also triggers and so we have that $t_{l^{ij}} = t_{l^{ji}}$, which in turn implies the symmetry of the coupling strengths between any connected pair (i, j). This fact is a direct consequence of the choice of symmetric threshold functions together with the symmetry of the trigger condition expressed by (3)–(4).

Note that, when condition (4) in Algorithm 1 is violated for a certain node $h \in \mathcal{N}_i$ (line 5), lines 6 to 10 can be replaced by the following fragment of code:

6: $l^i \leftarrow l^i + 1$ and $t_{l^i} \leftarrow t^*$;
7: $l^{ij} \leftarrow l^{ij} + 1$ and $t_{l^{ij}} \leftarrow t^*$, $\forall j \in \mathcal{N}_i$;
8: $e_{ij}(t_{l^{ij}}) \leftarrow e_{ij}(t = t^*)$, $\forall j \in \mathcal{N}_i$;
9: $\tilde{e}_{ij} \leftarrow 0$, $\forall j \in \mathcal{N}_i$;
10: Update the control input to the value

$$u_i(t) = c \sum_{j=1}^{N} a_{ij} \Gamma e_{ij}(t_{l^i}), \qquad t \in [t_{l^i}, t_{l^i+1}). \tag{6}$$

    Basically, in this last case, once the first trigger occurs, say for $\tilde{e}_{ih}(t)$, then not only the current value e_{ih} is updated and the corresponding trigger error (3) resets, but also all other values e_{ij} with $j \in \mathcal{N}_i$. When the above choice is made, we denote the obtained algorithm as Algorithm 1′.

The control input (5) leads to a diffusively coupled event-triggered dynamical network given by

$$\dot{x}_i(t) = f(t, x_i(t)) + c \sum_{j=1}^{N} a_{ij} \Gamma e_{ij}(t_{l^{ij}}), \tag{7}$$

for $t \in [t_{l^i}, t_{l^i+1})$ and for $i = 1, \ldots, N$. A similar expression is obtained considering the control input (6).

Remark 3. When using Algorithm 1′, all triggers related to pair (i, j), with $j \in \mathcal{N}_i$, are forced to be synchronous and, moreover, $t_{l^{ij}} = t_{l^{ih}}$ for all $j, h \in \mathcal{N}_i$. Conversely, at a generic time instant t, we have $e_{ij}(t) \neq e_{ji}(t)$. So, as all e_{ij} are updated at the same time, the symmetry is lost of the control actions between coupled pairs (i, j).

A convergence result for the considered event-triggered control scheme is now given. Before this, let us define

$$\epsilon_{\varsigma} := \frac{c\sqrt{N}N_{\max}\|\Gamma\|_2}{c\lambda_2(L \otimes \Gamma) - L_f}, \tag{8}$$

and for any constant $\delta > 0$,

$$\alpha := \frac{\delta}{1+\delta}\left[c\lambda_2(L \otimes \Gamma) - L_f\right], \tag{9}$$

where L represents the Laplacian network of the graph and $\lambda_2(L \otimes \Gamma)$ indicates the smallest nonzero eigenvalue of the positive semidefinite matrix $L \otimes \Gamma$.

Theorem 1. *Let us consider the event-triggered connected network* (7), *where the function $f(t, x)$ is Lipschitz continuous with respect to x with Lipschitz constant L_f and let us choose a coupling gain c such that*

$$L_f - c\lambda_2(L \otimes \Gamma) < 0. \tag{10}$$

Let us consider some constants k_{ς} such that

$$k_{\varsigma} \geq \frac{\|e(0)\|_2}{\epsilon_{\varsigma}}, \tag{11}$$

and λ_{ς} such that

$$0 < \lambda_{\varsigma} < \alpha, \tag{12}$$

where ϵ_{ς} and α are defined in (8) *and* (9), *respectively. We have the following results:*

i. *If $\lim_{t \to \infty} \varsigma_{ij}(t) = \bar{\varsigma}_{ij}$, with $\bar{\varsigma}_{ij} > 0$ for all i, j such that $a_{ij} \neq 0$, then both Algorithm 1 and Algorithm 1′ guarantee bounded synchronization of the network;*

ii. *If $\varsigma_{ij}(t) = k_{\varsigma}e^{-\lambda_{\varsigma}t}$ for all pairs (i, j) such that $a_{ij} \neq 0$, then Algorithm 1 and Algorithm 1′ guarantee complete synchronization of the network with exponential rate λ_{ς}.*

Furthermore, no Zeno behaviour occurs.

Proof. The proof is split into two steps. Firstly it is proven that (Step 1) synchronization occurs and then (Step 2) that no Zeno behaviour occurs. Eq. (7) can be rewritten as

$$\dot{x}_i = f(t, x_i) + c \sum_{j=1}^{N} a_{ij} \Gamma e_{ij}(t)$$

$$+ c \sum_{j=1}^{N} a_{ij} \Gamma \tilde{e}_{ij}(t), \quad \forall i = 1, \ldots, N. \tag{13}$$

Step 1. Let us consider the candidate Lyapunov function $V(e(t)) = \frac{1}{2}e^T e$ defined in the error space. We obtain

$$\dot{V}(e(t)) = \sum_{i=1}^{N} e_i^T \dot{e}_i = \sum_{i=1}^{N} e_i^T f(t, x_i) - \sum_{i=1}^{N} e_i^T \dot{\bar{x}}$$

$$- ce^T (L \otimes \Gamma) e + c \sum_{i=1}^{N} e_i^T \sum_{j=1}^{N} a_{ij} \Gamma \tilde{e}_{ij}.$$

Now, from condition (4), taking into account that $\sum_{i=1}^N e_i^T \dot{\bar{x}} = 0$, adding and subtracting $\sum_{i=1}^N e_i^T f(t, \bar{x})$, where $\sum_{i=1}^N e_i^T f(t, \bar{x}) = 0$ since $\sum_{i=1}^N e_i^T = 0$, the following inequality holds using the one-sided Lipschitz property (Agarwal & Lakshmikantham, 1993)

$$\dot{V} \le L_f e^T e - c e^T (L \otimes \Gamma) e + c\|e\|_2 \sqrt{N} N_{\max} \|\Gamma\|_2 \varsigma(t),$$

where L_f is the Lipschitz constant of the function f and $\varsigma(t) = \max_{i,j} \varsigma_{ij}(t)$. To obtain the previous inequality we have exploited the fact that $\|c\sum_{i=1}^N e_i^T \sum_{j=1}^N a_{ij}\Gamma\tilde{e}_{ij}\|_2 \le \|e\|_2 \cdot \|\xi\|_2$, with $\xi = \left(c\sum_{j=1}^N a_{1j}\Gamma\tilde{e}_{1j}, \ldots, c\sum_{j=1}^N a_{Nj}\Gamma\tilde{e}_{Nj}\right)^T$ and that, since $\|\tilde{e}_{ij}\|_2 \le \varsigma(t)\ \forall i,j$, we can use the bound $\|\xi\|_2 \le c\sqrt{N}N_{\max}\|\Gamma\|_2\varsigma(t)$.

Writing $e = a\hat{e}$, where $a = \|e\|_2$ is the norm of the error and $\hat{e} = \frac{1}{\|e\|_2} e$ is the unitary vector associated to e and considering that, due to the *Rayleigh–Ritz* theorem (Horn & Johnson, 1987), $\lambda_2(L \otimes \Gamma)e^T e \le e^T(L \otimes \Gamma)e$, the above inequality can be rewritten as

$$\dot{V}(e) \le \left(L_f - c\lambda_2(L \otimes \Gamma)\right)a^2 + c\sqrt{N}N_{\max}\|\Gamma\|_2\varsigma(t)a. \tag{14}$$

Now, as c is chosen in order to fulfil inequality (10), then the error trajectory $e(t)$ converges to the invariant region $\|e(t)\|_2 \le \epsilon$, where

$$\epsilon = \frac{c\sqrt{N}N_{\max}\|\Gamma\|_2\varsigma(t)}{c\lambda_2(L \otimes \Gamma) - L_f}, \tag{15}$$

or, using (8), equivalently $\epsilon = \epsilon_\varsigma \varsigma(t)$. So, if $\lim_{t\to\infty}\varsigma_{ij}(t) = \bar{\varsigma}_{ij}$ is verified, then $\lim_{t\to+\infty}\varsigma(t) = \bar{\varsigma} > 0$ and so bounded synchronization is ensured. Conversely, if $\varsigma_{ij}(t) = k_\varsigma e^{-\lambda_\varsigma t}$ holds, then $\lim_{t\to+\infty}\varsigma(t) = 0$ and so complete synchronization is achieved since the invariant region given by ϵ shrinks with exponential rate λ_ς.

Step 2. We prove next that no Zeno behaviour occurs. The more complicated case (item ii.) of complete synchronization will be firstly analysed, while a simpler reasoning will be later used for the case of bounded synchronization (item i.).

Let us define the strictly decreasing function

$$b(t) = (1+\delta)\epsilon_\varsigma\varsigma(t), \tag{16}$$

where $\delta > 0$ is an arbitrary constant value.

In order to prove that no Zeno behaviour occurs, we first show that for any time instant, inequality

$$\|e(t)\|_2 \le b(t) \quad \forall t \ge 0, \tag{17}$$

holds. In order to do so, let us note that $\|e(0)\|_2 < b(0)$. Now, since both $e(t)$ and $b(t)$ are continuous, if there is no time instant $\bar{t}$ such that $b(\bar{t}) = \|e(\bar{t})\|_2$, then relation (17) is trivially true. So, let us suppose that such time instant $\bar{t}$ exists. Now, for all $t \ge \bar{t}$ we evaluate the value of $\dot{V}(e)$ when e is such that $\|e\|_2 = b$. More precisely we have that

$$\dot{V}(e)\big|_{\|e\|_2=b} \le -\delta(1+\delta)\frac{[c\sqrt{N}N_{\max}\|\Gamma\|_2]^2}{c\lambda_2(L \otimes \Gamma) - L_f}\varsigma^2(t)$$

where the above formula has been obtained substituting a with expression (16) in (14). Multiplying and dividing the above relation by $(1+\delta)[c\lambda_2(L \otimes \Gamma) - L_f]$ the following expression is obtained

$$\dot{V}(e)\big|_{\|e\|_2=b} \le -\alpha b^2, \tag{18}$$

where α has been defined as in (9). Now, since

$$\dot{V}(e)\big|_{\|e\|_2=b} = \frac{d}{dt}\frac{1}{2}\|e\|_2^2\bigg|_{\|e\|_2=b} = b\frac{d}{dt}\|e\|_2\bigg|_{\|e\|_2=b}, \tag{19}$$

comparing (18) and (19) we get

$$\frac{d}{dt}\|e\|_2\bigg|_{\|e\|_2=b} \le -\alpha b. \tag{20}$$

Moreover, considering the decreasing function $B(t) = \frac{1}{2}b^2$ and remembering that $\varsigma(t) = k_\varsigma e^{-\lambda_\varsigma t}$, we have $\dot{B} = -\lambda_\varsigma b^2$. So, using (12) we get $\dot{V}(e)\big|_{\|e\|_2=b} \le \dot{B} < 0$ or, equivalently

$$\frac{d}{dt}\|e\|_2\bigg|_{\|e\|_2=b} \le -\alpha b \le -\lambda_\varsigma b. \tag{21}$$

Since expression (21) holds for all values $b \in [0, b(0)]$, (17) can be obtained by integrating both sides of (21) with respect to time.

We can now show that no Zeno behaviour occurs. Let us consider the dynamics of the error between a generic connected pair of nodes $(i, h) \in \mathcal{E}$. Such dynamics can be expressed as $\dot{e}_{ih}(t) = \dot{x}_h(t) - \dot{x}_i(t)$ thus,

$$\dot{e}_{ih} = f(t, x_h) + c\sum_{j=1}^N a_{hj}\Gamma e_{hj}(t) + c\sum_{j=1}^N a_{hj}\Gamma\tilde{e}_{hj}(t)$$
$$-f(t, x_i) - c\sum_{j=1}^N a_{ij}\Gamma e_{ij}(t) - c\sum_{j=1}^N a_{ij}\Gamma\tilde{e}_{ij}(t).$$

Now, taking the norm of both sides of the above equation into account together with the fact that f is Lipschitz and that $\|e_{ih}(t)\|_2 = \|x_h(t) - x_i(t) + \bar{x}(t) - \bar{x}(t)\|_2 \le \|e_h(t)\|_2 + \|e_i(t)\|_2 \le 2\|e(t)\|_2$ and recalling relation (17), we obtain

$$\|\dot{e}_{ih}(t)\|_2 \le 2\left[L_f + c\|\Gamma\|_2(N_h + N_i)\right]b(t)$$
$$+ c\|\Gamma\|_2(N_h + N_i)\varsigma(t), \tag{22}$$

where N_i and N_h are the degrees of nodes i and h, respectively, and where we have bounded $\|\tilde{e}_{ij}(t)\|_2$ and $\|\tilde{e}_{hj}(t)\|_2$ with the maximum admissible value of the threshold according to condition (4). Let $q_1 = 2\left[L_f + c\|\Gamma\|_2(N_h + N_i)\right]$ and $q_2 = c\|\Gamma\|_2(N_h + N_i)$. Then, at the last trigger event $t = t_{l^{ih}}$, from (22) we obtain

$$\|\dot{e}_{ih}(t)\|_2 \le q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}} + q_2 k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}}, \tag{23}$$

where, we have considered the choice $\varsigma_{ij}(t) = \varsigma(t) = k_\varsigma e^{-\lambda_\varsigma t}$ for all the pairs (i, j) (item ii. in the Theorem statement), with $a_{ij} \ne 0$. Now, in order to prove that Zeno behaviours do not occur in the network, we show that for all triggering instants $t_{k^{ih}}$ there exists a nonzero lower bound $\tau_m > 0$ such that the next event $t_{k^{ih}+1}$ will satisfy the condition $t_{k^{ih}+1} - t_{k^{ih}} \ge \tau_m$. To do so, let us consider the dynamics of the triggering error $\tilde{e}_{ih}(t)$ at time instants $t > t_l^{ih}$. Clearly, the following considerations will be valid not only for the last event instant $t_{l^{ih}}$ but for all instants $t_{k^{ih}}$, since the sequence $\{t_{k^{ih}}\}_{k^{ih}=0}^\infty$ is implicitly defined by the sequence of the last events. It is possible to write

$$\|\tilde{e}_{ih}(t)\|_2 \le \int_{t_{l^{ih}}}^t \|\dot{\tilde{e}}_{ih}(s)\|_2\,ds = \int_{t_{l^{ih}}}^t \|\dot{e}_{ih}(s)\|_2\,ds.$$

Taking into account inequality (23) and considering $t = t_{l^{ih}} + \tau$ from the above formula, we can write

$$\|\tilde{e}_{ih}(t_{l^{ih}} + \tau)\|_2 \le \left(q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}} + q_2 k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}}\right)\tau. \tag{24}$$

Referring to the trigger function (4) with the considered threshold $\varsigma(t_{l^{ih}} + \tau) = k_\varsigma e^{-\lambda_\varsigma(t_{l^{ih}}+\tau)}$, we have that τ_m solves the equation

$$k_\varsigma e^{-\lambda_\varsigma(t_{l^{ih}}+\tau_m)} = \left(q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}} + q_2 k_\varsigma e^{-\lambda_\varsigma t_{l^{ih}}}\right)\tau_m.$$

Multiplying both sides of the previous equation by $\frac{1}{k_\varsigma} e^{\lambda_\varsigma t_{lih}}$ we finally obtain

$$e^{-\lambda_\varsigma \tau_m} = \left(q_1(1+\delta)\epsilon_\varsigma + q_2\right)\tau_m, \qquad (25)$$

which implicitly defines τ_m as a non-zero lower bound between any two consecutive triggering instants.

The case of bounded synchronization (item i. of the Theorem statement) is, instead, easier than the case of complete synchronization. Indeed, note that

$$\|e_{ih}(t)\|_2 \le 2\|e(t)\|_2 \le 2 \sup_{t' \in [t, +\infty)} \|e(t')\|_2 \le 2\tilde{b}(t), \qquad (26)$$

where $\tilde{b}(t)$ is the nonincreasing piecewise smooth continuous function

$$\tilde{b}(t) = \begin{cases} \|e(t)\|_2 & \text{if } \|e(t)\|_2 > \epsilon_\varsigma \varsigma(t) \\ \epsilon_\varsigma \varsigma(t) & \text{if } \|e(t)\|_2 \le \epsilon_\varsigma \varsigma(t). \end{cases}$$

So, considering a generic triggering event at $t = t_{lih}$, inequality (22) can be bounded as

$$\|\dot{e}_{ih}(t)\|_2 \le q_1 \tilde{b}(t_{lih}) + q_2 \varsigma(t_{lih}), \quad \forall t \ge t_{lih}, \qquad (27)$$

where the same positions of q_1 and q_2 as done in equation (23) have been used in order to simplify the notation.

Integrating both sides of (27) with respect to time, a nonzero lower bound $\tau_{ih}(t_{lih})$ for the inter-event time between the last trigger event t_l^{ih} and the next one t_{lih+1} for the generic pair (i, h) is

$$\tau_{ih}(t_{lih}) = \frac{\bar{\varsigma}_{ih}}{q_1 \tilde{b}(t_{lih}) + q_2 \varsigma(t_{lih})}. \qquad (28)$$

This completes the proof.

Remark 4. Notice that choosing a high value of λ_ς allows to speed up the convergence rate. However, (25) shows that a faster synchronization reduces the value of the inter-event bound and so increases the frequency of the triggers.

Remark 5. Note that Theorem 1 holds for both Algorithm 1 and Algorithm 1′ since the proof is independent on the updating criterion of e_{ij}. Since in Algorithm 1′ all e_{ij} with $j \in \mathcal{N}_i$ are updated at the same time instant t_{li} and the corresponding errors $\tilde{e}_{ij}$ are reset, both for bounded and complete synchronization there implicitly exists a non-zero lower bound between any two consecutive updating events of the control law. For this reason, Algorithm 1′ can be implemented in all applications where constraints on actuators do not allow to change the control input arbitrarily fast.

4. Periodic event-triggered synchronization

The scheme presented in the previous section can be easily modified in order to derive a periodic event-triggered synchronization setup, where the agents communicate in an asynchronous way and with clocks of possibly different periods. Such scheme exploits the advantages of having a nonzero lower bound for the inter-event times in Algorithm 1′, which turns to be useful for a periodic event detection of a trigger condition. Specifically, we no longer require a model based approach and we consider here for each agent i a periodic checking sequence $\{T_{p^i}\}_{p^i=0}^{\infty}$, where the agent obtains its neighbours' state value. Such sequence is computed considering the sampled times $T_{p^i} = t_{0^i} + p^i \tau_i$, with $\tau_i \le \tau_m$, where τ_m is the solution of (25). For the sake of brevity and without loss of generality, we present here the case of a constant period τ_i, but the same setup and the same analysis can be adopted in the case of a time-varying $\tau_i(T_{p^i}) \le \tau_m$, thus resulting in a cyclic event-triggered

detection, instead of a periodic one. More specifically, at each T_{p^i}, agent i obtains the measurement of its own state and the one of its neighbours and evaluates, for all $h \in \mathcal{N}_i$, the value $\tilde{e}_{ih}(T_{p^i})$ given in (3), deciding whether or not to trigger an update of its piecewise constant control input (6) according to a suitable trigger condition. So, the resulting updating sequence $\{t_{k^i}\}_{k^i=0}^{\infty}$ is a subsequence of the periodic checking sequence, namely $\{t_{k^i}\}_{k^i=0}^{\infty} \subseteq \{T_{p^i}\}_{p^i=0}^{\infty}$. Considering the same constants given in Section 3, the periodic event-triggered synchronization algorithm is reported below as Algorithm 2.

Algorithm 2 Periodic event-triggered update

1: **for** $p^i = 0, 1, \ldots$ **do**
2: At each instant $T_{p^i} = t_{0^i} + p^i \tau_i$, obtain neighbours' state information and evaluate $\tilde{e}_{ih}(T_{p^i})$, with $h \in \mathcal{N}_i$.
3: Compute $\bar{\tau}_i(T_{p^i}) = \min_{h \in \mathcal{N}_i} \tau_{ih}^*(T_{p^i})$, with $\tau_{ih}^*(T_{p^i})$ the solution of the implicit equation
$$e^{-\lambda_\varsigma \tau_{ih}^*(T_{p^i})} = \frac{1}{k_\varsigma}\|\tilde{e}_{ih}(T_{p^i})\|_2 e^{\lambda_\varsigma T_{p^i}} + \left(q_1(1+\delta)\epsilon_\varsigma + q_2\right)\tau_{ih}^*(T_{p^i}) \qquad (29)$$
4: **if** $\bar{\tau}_i(T_{p^i}) < \tau_i$ **then**
5:   $l^i \leftarrow l^i + 1$ and $t_{l^i} \leftarrow T_{p^i}$;
6:   $\tilde{e}_{ih} \leftarrow 0$;
7:   Update the control input $u_i(t_{l^i})$ to the value in (6);
8: **end if**
9: **end for**

For the periodic event-triggered scheme we can give the following result.

Theorem 2. *Let us consider network* (7)*, where the function $f(t, x)$ is Lipschitz continuous with respect to x with Lipschitz constant L_f and let us choose a coupling gain c and a function $\varsigma(t) = k_\varsigma e^{-\lambda_\varsigma t}$ such that the inequalities* (10)–(12) *of Theorem 1 hold. Then, the periodic event-triggered control scheme given in Algorithm 2 guarantees complete synchronization of the network with exponential rate λ_ς.*

Proof. The proof can be obtained following similar steps of those in the proof of Theorem 1. In particular, the key point is to prove that the trigger condition $\bar{\tau}_i(T_{p^i}) < \tau_i$ guarantees $\|\tilde{e}_{ih}(t)\|_2 \le \varsigma(t)$ for all $t \ge 0$ and for all the connected pairs (i, h). To do so, let us consider an induction argument starting from a generic update time $t_{l^i} = T_{l^i}$. Following the same steps leading to (22), we obtain

$$\|\dot{e}_{ih}(t)\|_2 \le q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma T_{l^i}} + q_2 k_\varsigma e^{-\lambda_\varsigma T_{l^i}}, \qquad (30)$$

which is the same inequality of (23) where we have substituted t_{lih} with T_{l^i}. Now, considering that $\tilde{e}_{ih}(t) = \tilde{e}_{ih}(T_{l^i}) + \int_{T_{l^i}}^{t} \dot{\tilde{e}}_{ih}(s)ds$, analogously to what was done for Theorem 1, evaluating the norm of both sides and substituting (30), we obtain $\|\tilde{e}_{ih}(T_{l^i} + \tau_{ih})\|_2 \le \|\tilde{e}_{ih}(T_{l^i})\|_2 + \left(q_1(1+\delta)\epsilon_\varsigma k_\varsigma e^{-\lambda_\varsigma T_{l^i}} + q_2 k_\varsigma e^{-\lambda_\varsigma T_{l^i}}\right)\tau_{ih}$. Following similar steps to those taken to obtain Eq. (25), we get the implicit equation (29) in the variable $\tau_{ih}^*(T_{p^i})$. Since at $t_{l^i} = T_{l^i}$ we have $\|\tilde{e}_{ih}(t)\|_2 = 0$ due to line 6 in Algorithm 2, Eqs. (25) and (29) are identical, providing the same solution τ_m. So, at the following checking instant $\|\tilde{e}_{ih}(T_{p^i+1})\|_2 \le \varsigma(T_{p^i+1})$. Solving again Eq. (29) at T_{p^i+1}, if $\bar{\tau}_i(T_{p^i+1}) \ge \tau_i$ then at the checking instant T_{p^i+2} we will satisfy again $\|\tilde{e}_{ih}(T_{p^i+2})\|_2 \le \varsigma(T_{p^i+2})$, otherwise a trigger is generated and the control input is updated. By induction, the reasoning can be iterated for all the future instants, while the stability proof follows exactly the same steps as in Step 1 of Theorem 1.

5. Numerical examples

We consider a network of identical Chua circuits, a paradigmatic nonlinear example of chaotic behaviours which has been

Table 1
Number of triggers in unitary intervals for the network of Chua systems: static thresholds.

	[0, 1) s	[1, 2) s	[2, 3) s	[3, 4) s	[4, 5)
Node 1	32	33	26	25	22
Node 2	35	25	27	39	33
Node 3	29	25	19	22	23
Node 4	32	24	22	27	36
Node 5	15	15	14	21	23

Table 2
Number of triggers in unitary intervals for the network of Chua systems: exponential thresholds.

	[0, 1) s	[1, 2) s	[2, 3) s	[3, 4) s	[4, 5)
Node 1	16	14	17	16	17
Node 2	19	25	26	26	22
Node 3	7	12	18	13	13
Node 4	0	0	0	0	20
Node 5	0	0	0	0	10

Table 3
Number of triggers in unitary intervals for the network of Chua systems: periodic event-triggered.

	[0, 1) s	[1, 2) s	[2, 3) s	[3, 4) s	[4, 5)
Node 1	5882	5883	5882	5882	5883
Node 2	6667	6666	6667	6667	6666
Node 3	1000	1000	1000	1000	1000
Node 4	8333	8333	8334	8333	8333
Node 5	7143	7142	7143	7143	7143

considered as a testbed for both numerical and experimental analyses of theoretical synchronization strategies and for applications to communication (de Magistris, di Bernardo, Tucci, & Manfredi, 2012; Gámez-Guzmán, Cruz-Hernández, López-Gutiérrez, & García-Guerrero, 2009). The dynamical model of a single system is $\dot{x}_i = f(x_i)$ given by $\dot{x}_{i1} = \alpha\left[x_{i2} - x_{i1} - \varphi(x_{i1})\right]$; $\dot{x}_{i2} = x_{i1} - x_{i2} + x_{i3}$; $\dot{x}_{i3} = -\beta x_{i2}$, with $\alpha = 10$, $\beta = 17.30$, and $\varphi(x_{i1}) = bx_{i1} + (a - b)(|x_{i1} + 1| - |x_{i1} - 1|)/2$, with $a = -1.34$, $b = -0.73$. For this vector field it is possible to evaluate an upper bound for the Lipschitz constant $L_f = 34.2$; a network of five Chua circuits over a connected random graph is simulated and the matrix Γ is the identity matrix for the sake of simplicity; a value of the minimum coupling guaranteeing inequality (10) is $c = 13.7$. Simulations have been performed applying Algorithm 1 and setting an identical static threshold $\varsigma_{ij}(t) = \bar{\varsigma}$ for all connected pairs (i, j), with $\bar{\varsigma} = 0.1$. The synchronization of the chaotic trajectories is obtained within the first 2 s (figures are omitted here for the sake of brevity). Also, simulations have been carried out for Algorithm 1′ considering identical exponential threshold functions $\varsigma_{ij} = k_\varsigma e^{-\lambda_\varsigma t}$ with $k_\varsigma = 1$ and $\lambda_\varsigma = 0.5$ (again figures are omitted for the sake of brevity). In this case, the exponential synchronization of the network is obtained within 5 s. The number of triggers for each node in time intervals of unitary length for the first 5 s of simulation is reported in Tables 1 and 2 for the case of static threshold with Algorithm 1 and for the case of exponential threshold with Algorithm 1′, respectively. Observe how the first approach generates a higher number of triggers than the second one. Simulations have also been carried out for the case of the same static threshold with Algorithm 1′, showing better performance than the case with Algorithm 1.

Simulations for the case of periodic event-triggered synchronization developed in Algorithm 2 have been conducted, considering for (29) the same parameters of the case of identical exponential threshold. This led to a $\tau_m = 0.17$ ms and so, $\tau_i \leq \tau_m$ have been randomly assigned accordingly for each node. As illustrated in Table 3, a significant higher number of triggers is generated in the periodic case due to the conservativeness of the approach, that does not rely on model-based computations. Although, a faster convergence is obtained (within 0.5 s) due to the reduced error mismatch on the connected pairs of agents.

Finally, for the sake of comparison, a time-triggered control protocol where all the nodes update their control law following a centralized sampling of period $T_s = 60$ ms has been carried out. Such sampling period corresponds to the average of all the inter-event intervals obtained for the case of exponential thresholds and leads the network to instability.

6. Conclusions

A model-based approach where connected agents broadcast input information has been considered and results have been given for bounded synchronization and for exponential synchronization. The absence of Zeno behaviour has been proven guaranteeing a lower bound for the inter-event times between consecutive updates. The results were then extended to an asynchronous periodic event-triggered setup, where the agents periodically gather neighbours' state information and check for a trigger condition in order to decide whether or not update their control input. In this latter scheme, a model-based information is no longer required.

References

Agarwal, R. P., & Lakshmikantham, V. (1993). *Uniqueness and nonuniqueness criteria for ordinary differential equations*. Singapore: World Scentific.

Arcak, M. (2007). Passivity as a design tool for group coordination. *IEEE Transactions on Automatic Control*, 52(8), 1380–1390.

de Magistris, M., di Bernardo, M., Tucci, E. D., & Manfredi, S. (2012). Synchronization of networks of non-identical Chua circuits: analysis and experiments. *IEEE Transactions on Circuits and Systems*, 59, 1029–1041.

Dimarogonas, D. V., Frazzoli, E., & Johansson, K. H. (2012). Distributed event-triggered control for multi-agent systems. *IEEE Transactions on Automatic Control*, 57, 1291–1297.

Gámez-Guzmán, L., Cruz-Hernández, C., López-Gutiérrez, R. M., & García-Guerrero, E. E. (2009). Synchronization of Chua's circuits with multi-scroll attractors: Application to communication. *Communications in Nonlinear Science and Numerical Simulation*, 14, 2765–2775.

Godsil, C. D., & Royle, G. (2001). *Algebraic graph theory*. Springer.

Guinaldo, M., Dimarogonas, D.V., Johansson, K.H., Sánchez, J., & Dormido, S. (2011). Distributed event-based control for interconnected linear systems. In *50th IEEE conference on decision and control and European control conference*.

Horn, R. A., & Johnson, C. R. (1987). *Matrix analisis*. Cambridge University Press.

Johansson, K. H., Egerstedt, M., Lygeros, J., & Sastry, S. S. (1999). On the regularization of Zeno hybrid automata. *Systems & Control Letters*, 38, 141–150.

Liu, T., Cao, M., Persis, C.D., & Hendrickx, J. (2013). Distributed event-triggered control for synchronization of dynamical networks with estimators. In *Proceeding of IFAC NecSys 2013*.

Liu, T., Hill, D.J., & Liu, B. (2013). Synchronization of dynamical networks with distributed event-based communication. In *51st IEEE conference on decision and control* (pp. 7199–7204).

Olfati-Saber, R., A., Fax, J., & Murray, R. (2007). Consensus and cooperation in networked multi-agent system. In *Proceedings of the IEEE, Vol. 95* (pp. 215–233).

De Persis, C. (2013). On self-triggered synchronization of linear systems. In *4th IFAC workshop on distributed estimation and control in networked systems, NecSys*.

Seyboth, G. S., Dimarogonas, D. V., & Johansson, K. H. (2013). Event-based broadcasting for multi-agent average consensus. *Automatica*, 49, 245–252.

Tabuada, P. (2007). Event-triggered real-time scheduling of stabilizing control tasks. *IEEE Transactions on Automatic Control*, 52, 1680–1684.

Davide Liuzza received his Ph.D. degree in Automation Engineering from the University of Naples Federico II, in Apr. 2013. He has been visiting Ph.D. student at the Department of Applied Mathematics, University of Bristol (UK) from Jan. 2012 to Feb. 2012 and at the ACCESS Linnaeus Centre, Royal Institute of Technology (KTH), Stockholm, Sweden, from Jul. 2012 to Jan. 2013. From Sep. 2013 to Sep. 2015 he had a Post-doctoral scholarship at the Automatic Control Laboratory in KTH. Currently, he is a post doc at the department of Engineering, University of Sannio, Benevento, Italy.

His research interests are in networked control systems, coordination of multi-agent systems, incremental stability of nonlinear systems, nonlinear control and hybrid systems.

Dimos V. Dimarogonas was born in Athens, Greece, in 1978. He received the Diploma in Electrical and Computer Engineering in 2001 and the Ph.D. in Mechanical Engineering in 2007, both from the National Technical University of Athens (NTUA), Greece. Between May 2007 and February 2009, he was a Postdoctoral Researcher at the Automatic Control Laboratory, School of Electrical Engineering, ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden. Between February 2009 and March 2010, he was a Postdoctoral Associate at the Laboratory for Information and Decision Systems (LIDS) at the Massachusetts Institute of Technology (MIT), Boston, MA, USA. He is currently an Associate Professor at the Automatic Control Laboratory, ACCESS Linnaeus Center, KTH Royal Institute of Technology, Stockholm, Sweden. His current research interests include Multi-Agent Systems, Hybrid Systems and Control, Robot Navigation and Networked Control. He was awarded a Docent in Automatic Control from KTH in 2012. He serves in the Editorial Board of Automatica, the IEEE Transactions on Automation Science and Engineering and the IET Control Theory and Applications and is a member of IEEE and the Technical Chamber of Greece. He received an ERC Starting Grant from the European Commission for the proposal BUCOPHSYS in 2014 and was awarded a Wallenberg Academy Fellow grant in 2015.

Mario di Bernardo (SMIEEE'06, FIEEE 2012) is currently Full Professor of Automatic Control at the University of Naples Federico II, Italy. He is also Professor of Nonlinear Systems and Control at the University of Bristol, UK

On 28th February 2007 he was bestowed the title of "Cavaliere" of the Order of Merit of the Italian Republic for scientific merits from the President of Italy. In January 2012 he was elevated to the grade of Fellow of the IEEE for his contributions to the analysis, control and applications of nonlinear systems and complex networks. In 2009, he was elected President of the Italian Society for Chaos and Complexity for the term 2010–2013. He was re-elected in 2010 for the term 2014–2017. In 2006 and again in 2009 he was elected to the Board of Governors

of the IEEE Circuits and Systems Society. From 2011 to 2014 he was Vice President for Financial Activities of the IEEE Circuits and Systems Society. In 2015 he served as appointed member of the Board of Governors of the IEEE Control Systems Society. He is Distinguished Lecturer of the IEEE Circuits and Systems Society for the term 2016–2017.

His research interests include the analysis, synchronization and control of complex network systems; the analysis and control of hybrid and piecewise-smooth dynamical systems; nonlinear dynamics, nonlinear control theory and applications to engineering and synthetic biology.

He serves on the Editorial Board of several international scientific journals and conferences. From 1st January 2014 till 31st December 2015 he was Deputy Editor-in-Chief of the IEEE Transactions on Circuits and Systems: Regular Papers. He is Associate Editor of the IEEE Transactions on Control of Network Systems, Nonlinear Analysis: Hybrid Systems, the Conference Editorial Board of the IEEE Control System Society and the European Control Association (EUCA).

Karl H. Johansson is Director of the ACCESS Linnaeus Centre and Professor at the School of Electrical Engineering, KTH Royal Institute of Technology, Sweden. He heads the Stockholm Strategic Research Area ICT The Next Generation. He received M.Sc. and Ph.D. degrees in Electrical Engineering from Lund University. He has held visiting positions at UC Berkeley, California Institute of Technology, Nanyang Technological University, and Institute of Advanced Studies Hong Kong University of Science and Technology. His research interests are in networked control systems, cyber–physical systems, and applications in transportation, energy, and automation systems. He is a member of the IEEE Control Systems Society Board of Governors and the European Control Association Council. He is past Chair of the IFAC Technical Committee on Networked Systems. He has been on the Editorial Boards of Automatica, IEEE Transactions on Automatic Control, and IET Control Theory and Applications. He is currently a Senior Editor of IEEE Transactions on Control of Network Systems and Associate Editor of European Journal of Control. In 2009 he was awarded Wallenberg Scholar, as one of the first ten scholars from all sciences, by the Knut and Alice Wallenberg Foundation. He was awarded Future Research Leader from the Swedish Foundation for Strategic Research in 2005. He received the triennial Young Author Prize from IFAC in 1996 and the Peccei Award from the International Institute of System Analysis, Austria, in 1993. He received Young Researcher Awards from Scania in 1996 and from Ericsson in 1998 and 1999. He is a Fellow of the IEEE.