Automatica 86 (2017) 199–204

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Distributed event-triggered control for asymptotic synchronization of dynamical networks☆

Tao Liu [a], Ming Cao [b], Claudio De Persis [b], Julien M. Hendrickx [c]

[a] Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong Special Administrative Region, China
[b] Faculty of Science and Engineering, University of Groningen, 9747 AG, Groningen, The Netherlands
[c] ICTEAM Institute, Université catholique de Louvain, Louvain-la-Neuve, Belgium

ARTICLE INFO

Article history:
Received 13 August 2015
Received in revised form 16 June 2017
Accepted 12 August 2017

Keywords:
Distributed event-triggered control
Asymptotic synchronization
Dynamical networks

ABSTRACT

This paper studies synchronization of dynamical networks with event-based communication. Firstly, two estimators are introduced into each node, one to estimate its own state, and the other to estimate the average state of its neighbours. Then, with these two estimators, a distributed event-triggering rule (ETR) with a dwell time is designed such that the network achieves synchronization asymptotically with no Zeno behaviours. The designed ETR only depends on the information that each node can obtain, and thus can be implemented in a decentralized way.

1. Introduction

Synchronization of dynamical networks, and its related problem—consensus of multi-agent systems, have attracted a lot of attention due to their extensive applications in various fields (see Arenas, Díaz-Guilera, Kurths, Moreno, and Zhou, 2008; Olfati-Saber, Fax, and Murray, 2007, Ren, Beard, and Atkins, 2007; Wu, 2007 for details). Motivated by the fact that connected nodes in some real-world networks share information over a digital platform, these problems have recently been investigated under the circumstance that nodes communicate to their neighbours only at certain discrete-time instants. To use the limited communication network resources effectively, event-triggered control (ETC) (see Heemels, Johansson, and Tabuada, 2012 and reference therein) introduced in networked control systems has been extensively used to synchronize networks. Under such a circumstance, each node can only get limited information, and the main issue becomes how to use these limited information to design an ETR for each node such that the network achieves synchronization asymptotically and meanwhile to prevent Zeno behaviours that are caused by the continuous/discrete-time hybrid nature of ETC, an undesirable in practice (Tabuada, 2007).

Early works in ETC focused on dynamical networks with simple node dynamics such as single-integrators and double-integrators. In Dimarogonas and Johansson (2009), distributed ETC was used to achieve consensus. To prevent Zeno behaviour, a decentralized ETR with a time-varying threshold was introduced to achieve consensus in Seyboth, Dimarogonas, and Johansson (2013). Self-triggered strategies were proposed in De Persis and Frasca (2013) and shown to be robust to skews of the local clocks, delays, and limited precision in the communication.

Most recently, attention has been increasingly paid to networks with generalized linear node dynamics. Different types of ETC have been developed to achieve either bounded or asymptotic synchronization for such networks (e.g., Demir and Lunze, 2012; Zhu, Jiang, and Feng, 2014; Liu, Cao, De Persis, and Hendrickx, 2013; Meng and Chen, 2013; Xiao, Meng, and Chen, 2015; Garcia, Cao, Wang, and Casbeer, 2015; Yang, Ren, Liu, and Chen, 2016; Hu, Liu, and Feng, 2016). In order to achieve asymptotic synchronization as well as to prevent Zeno behaviours, two main methods are developed in the literature. One uses bidirectional communication, i.e., at each event time, a node sends its sampled state to its neighbours and meanwhile asks for its neighbours' current states to update the control signal (e.g., Hu et al., 2016; Meng and Chen, 2013; Xiao et al., 2015). The other uses unidirectional communication, i.e., a node only sends its sampled information to its neighbours but does not require information from its neighbours (e.g., Liu et al., 2013; Garcia et al., 2015; Yang et al., 2016). However, the latter needs $d_i + 1 \geq 2$ estimators in each node and uses an exponential term in the ERT in order to prevent Zeno behaviours.

In this paper, we study asymptotic synchronization of networks with generalized linear node dynamics by using the unidirectional communication method. The main differences from the existing

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Dimos V. Dimarogonas under the direction of Editor Christos G. Cassandras.

E-mail addresses: taoliu@eee.hku.hk (T. Liu), m.cao@rug.nl (M. Cao), c.de.persis@rug.nl (C. De Persis), julien.hendrickx@uclouvain.be (J.M. Hendrickx).

http://dx.doi.org/10.1016/j.automatica.2017.08.026

results are as follows. Firstly, a new sampling mechanism is used with which two estimators are introduced into each node, whereas most existing results need every node to estimate the states of all its neighbours. Secondly, inspired by the method proposed in Tallapragada and Chopra (2014), we replace the exponential term extensively used in the literature by a dwell time that was originally introduced in switched systems (Cao & Morse, 2010), which can simplify the implementation of the designed ETR. Thirdly, a distributed ETR for each node is designed based on the two estimators and dwell time, whereas most of the existing results use decentralized ETRs that only consist of local information of the node itself, i.e., the state error between the node and its own estimator and the time-dependent exponential term (e.g., Garcia et al., 2015; Yang et al., 2016). By introducing an estimation of the synchronization errors between neighbours using the neighbours' sampled information, the proposed ETR method can reduce the number of sampling times for each node significantly.

2. Network model and preliminaries

Notation: Denote the set of real numbers, non-negative real numbers, and non-negative integers by $\mathbb{R}$, $\mathbb{R}^+$, and $\mathbb{Z}^+$; the set of n-dimensional real vectors and $n \times m$ real matrices by $\mathbb{R}^n$ and $\mathbb{R}^{n\times m}$. I_n, 1_n and $1_{n\times m}$ are the n-dimensional identity matrix, n-dimensional vector and $n \times m$ matrix with all entries being 1, respectively. $\|\cdot\|$ represents the Euclidean norm for vectors and also the induced norm for matrices. The superscript $(\cdot)^\top$ is the transpose of vectors or matrices. $\otimes$ is the Kronecker product of matrices. For a single $\omega : \mathbb{R}^+ \to \mathbb{R}^n$, $\omega(t^-) = \lim_{s\uparrow t}\omega(s)$. Let $\mathcal{G}$ be an undirected graph consisting of a node set $\mathcal{V} = \{1, 2, \dots, N\}$ and a link set $\mathcal{E} = \{\bar{e}_1, \bar{e}_2, \dots, \bar{e}_M\}$. If there is a link $\bar{e}_k$ between nodes i and j, then we say node j is a neighbour of node i and vice versa. Let $A = (a_{ij}) \in \mathbb{R}^{N\times N}$ be the adjacency matrix of $\mathcal{G}$, where $a_{ii} = 0$ and $a_{ij} = a_{ji} > 0$, $i \neq j$, if node i and node j are neighbours, otherwise $a_{ij} = a_{ji} = 0$. The Laplacian matrix $L = (l_{ij}) \in \mathbb{R}^{N\times N}$ is defined by $l_{ij} = -a_{ij}$, if $j \neq i$ and $l_{ii} = \sum_{j=1}^N a_{ij}$.

We consider a dynamical network described by

$$\dot{x}_i(t) = Hx_i(t) + Bu_i(t), \quad \forall i \in \mathcal{V} \tag{1}$$

where $x_i = (x_{i1}, x_{i2}, \dots, x_{in})^\top \in \mathbb{R}^n$ is the state of node i. $H \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^n$, and $u_i \in \mathbb{R}$ are the node dynamics matrix, input matrix, and control input, respectively. Generally, continuous communication between neighbouring nodes is assumed, i.e., $u_i(t) = K\sum_{j=1}^N a_{ij}(x_j(t) - x_i(t))$. This yields the following network

$$\dot{x}_i(t) = Hx_i + BK\sum_{j=1}^N a_{ij}(x_j(t) - x_i(t)). \tag{2}$$

In this paper, we assume that connections in (1) are realized via discrete communication, i.e., each node only obtains information from its neighbours at certain discrete-time instants. We will present an event-triggered version of network (2), and study how to design an ETR for each node to achieve asymptotic synchronization. We suppose that the topological structure of the network is fixed, undirected and connected.

We introduce two estimators $\mathcal{O}_i$ and $\mathcal{O}_{\mathcal{V}_i}$ into each node i, where $\mathcal{O}_i$ is used to estimate its own state, and $\mathcal{O}_{\mathcal{V}_i}$ is used to estimate the average state of its neighbours. We adopt the following control input

$$u_i(t) = K\left(\hat{x}_{\mathcal{V}_i}(t) - l_{ii}\hat{x}_i(t)\right) \tag{3}$$

where $K \in \mathbb{R}^{1\times n}$ is the control gain to be designed, $\hat{x}_i \in \mathbb{R}^n$ and $\hat{x}_{\mathcal{V}_i} \in \mathbb{R}^n$ are states of $\mathcal{O}_i$ and $\mathcal{O}_{\mathcal{V}_i}$, respectively. The state equations of $\mathcal{O}_i$ and $\mathcal{O}_{\mathcal{V}_i}$ are given by

$$\mathcal{O}_i : \begin{array}{ll} \dot{\hat{x}}_i(t) = H\hat{x}_i(t), & t \in [t_{k_i}, t_{k_i+1}) \\ \hat{x}_i(t) = x_i(t), & t = t_{k_i} \end{array} \tag{4}$$

$$\mathcal{O}_{\mathcal{V}_i} : \begin{array}{ll} \dot{\hat{x}}_{\mathcal{V}_i}(t) = H\hat{x}_{\mathcal{V}_i}(t), & t \in [t_{\bar{k}_i}, t_{\bar{k}_i+1}) \\ \hat{x}_{\mathcal{V}_i}(t) = \hat{x}_{\mathcal{V}_i}(t^-) - \sum_{j\in\mathcal{J}_i} e_j(t^-), & t = t_{\bar{k}_i}. \end{array} \tag{5}$$

The increasing time sequences $\{t_{k_i}\}$ and $\{t_{\bar{k}_i}\}$, $k_i, \bar{k}_i \in \mathbb{Z}^+$ represent time instants that node i sends updates to its neighbours and that it receives updates from one or more of its neighbours, respectively. We assume that: there is no time delay for computation and execution, i.e., t_{k_i} represents both the k_ith sampling time and the k_ith time when node i broadcasts updates; and the communication network is under an ideal circumstance, i.e., there are no time delays or data dropouts in communication. Therefore, the set $\mathcal{J}_i = \mathcal{J}_i(t_{\bar{k}_i}) = \left\{j \mid t_{k_j} = t_{\bar{k}_i}, j \in \mathcal{V}_i\right\}$ is a subset of $\mathcal{V}_i$, from which node i receives updated information at $t = t_{\bar{k}_i}$, and $\mathcal{V}_i = \{j \mid a_{ij} > 0, j \in \mathcal{V}\}$ is the index set of the neighbours for node i. The vector $e_i(t) = \hat{x}_i(t) - x_i(t)$ represents the deviation between the state of estimator $\mathcal{O}_i$ and its own, and which node i can easily compute.

The time sequence $\{t_{k_i}\}$ is decided by the ETR

$$t_{k_i+1} = \inf\left\{t > t_{k_i} \mid r_i(t, x_i, \hat{x}_i, \hat{x}_{\mathcal{V}_i}) > 0\right\} \tag{6}$$

where $r_i(\cdot,\cdot,\cdot,\cdot) : \mathbb{R}^+ \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is the event-triggering function to be designed. For $t > t_{k_i}$, if $r_i > 0$ at $t = t^-_{k_i+1}$, then node i samples $x_i(t^-_{k_i+1})$, $\hat{x}_i(t^-_{k_i+1})$, calculates $e_i(t^-_{k_i+1})$, sends $e_i(t^-_{k_i+1})$ to its neighbours, and reinitialize the estimator $\mathcal{O}_i$ at $t = t_{k_i+1}$ by $x_i(t_{k_i+1})$. In addition, node i will reinitialize the estimator $\mathcal{O}_{\mathcal{V}_i}$ by $\hat{x}_{\mathcal{V}_i}(t_{\bar{k}_i+1}) = \hat{x}_{\mathcal{V}_i}(t^-_{\bar{k}_i+1}) - \sum_{j\in\mathcal{J}_i} e_j(t^-_{\bar{k}_i+1})$ each time when it receives updates from its neighbours. We further assume the network is well initialized at $t = t_0$, i.e., $\hat{x}_i(t_0) = 0$ and each node samples and sends $e_i(t_0)$ to its neighbours. Therefore, we have $\hat{x}_i(t_0) = x_i(t_0)$, $\hat{x}_{\mathcal{V}_i}(t_0) = \sum_{j\in\mathcal{V}_i} x_j(t_0)$ and $\mathcal{J}_i(t_0) = \mathcal{V}_i$ for all $i \in \mathcal{V}$. Then, the problem is with the given network topology, to design a proper ETR (6) such that network (1) achieves synchronization asymptotically without Zeno behaviours.

To simplify the analysis, we will show that network (1) with controller (3) and estimators (4), (5) is equivalent to the following system where each node maintains an estimator of the state of each of its neighbours.

$$\dot{x}_i(t) = Hx_i(t) - BK\sum_{j=1}^N l_{ij}\hat{x}_j(t), \forall i \in \mathcal{V} \tag{7a}$$

$$\mathcal{O}_i : \begin{array}{ll} \dot{\hat{x}}_i(t) = H\hat{x}_i(t), & t \in [t_{k_i}, t_{k_i+1}) \\ \hat{x}_i(t) = x_i(t), & t = t_{k_i}. \end{array} \tag{7b}$$

Defining $\bar{z}_i = \sum_{j\in\mathcal{V}_i}\hat{x}_j$ gives $\dot{\bar{z}}_i(t) = \sum_{j\in\mathcal{V}_i}\dot{\hat{x}}_j(t) = H\bar{z}_i(t)$, $t \in [t_{\bar{k}_i}, t_{\bar{k}_i+1})$, which has the same dynamics as $\hat{x}_{\mathcal{V}_i}$ defined in (5). Moreover, at $t = t_{\bar{k}_i}$, we have

$$\begin{aligned}\bar{z}_i(t) &= \sum_{j\in\mathcal{V}_i/\mathcal{J}_i(t)}\hat{x}_j(t^-) + \sum_{j\in\mathcal{J}_i(t)} x_j(t) \\ &= \sum_{j\in\mathcal{V}_i/\mathcal{J}_i(t)}\hat{x}_j(t^-) + \sum_{j\in\mathcal{J}_i(t)}\left(\hat{x}_j(t^-) - e_j(t^-)\right) \\ &= \hat{x}_{\mathcal{V}_i}(t).\end{aligned} \tag{8}$$

Thus, we have $\bar{z}_i(t) = \hat{x}_{\mathcal{V}_i}(t)$ for all $t \geq t_0$. Then, controller (3) becomes

$$u_i = K\left(\bar{z}_i - l_{ii}\hat{x}_i\right) = K\left(\hat{x}_{\mathcal{V}_i} - l_{ii}\hat{x}_i\right). \tag{9}$$

Substituting (9) into (1) gives that network (1) with (3), (4), and (5) is equivalent to (7).

Moreover, let $\hat{z}_i = \sum_{j\in\mathcal{V}_i}(\hat{x}_j - \hat{x}_i)$. We have $\hat{x}_{\mathcal{V}_i} = \bar{z}_i = \hat{z}_i + l_{ii}\hat{x}_i$. Then, ETR (6) can be reformulated as

$$t_{k_i+1} = \inf\left\{t > t_{k_i} \mid r_i(t, x_i, \hat{x}_i, \hat{z}_i) > 0\right\}. \tag{10}$$