A randomized linear algorithm for clock synchronization in multi–agent systems

Bolognani, Saverio; Lovisari, Enrico; Carli, Ruggero; Zampieri, Sandro

Published in:
Proceedings of the 51st Conference on Decision and Control

2012

A randomized linear algorithm for clock synchronization in multi–agent systems

Saverio Bolognani Ruggero Carli Enrico Lovisari Sandro Zampieri

Abstract— **In this paper a randomized linear protocol for time synchronization of clocks in a multi–agent scenario is considered. Clocks are allowed to have different offsets and different rates, and they communicate through an asymmetric broadcast protocol. The contribution of this paper is twofold. It is first shown that, under very mild conditions on the communication graph, it is possible to tune a protocol parameter in such a way that synchronization is achieved in mean–square. Then, via numerical simulations, the proposed strategy is compared with other fully distributed strategies recently proposed in the literature. While being slightly slower to reach the asymptotic synchronization, the proposed strategy significantly outperforms the other strategies in terms of robustness against process and measurement noises and time-varying clock drifts.**

I. Introduction

In networked control systems and in multi–agent systems it is often needed to guarantee tight time synchronization among the different agents. For example, basic synchronization is needed in any sensor network, when different devices have to provide their measurements with proper time–stamping for subsequent data fusion and processing.

In some cases, however, the need for synchronization can be very demanding. This is the case, for example, when the collected data need to be interpreted according to a fast dynamical model for the system, like in distributed detection and localization of moving targets. In other scenarios, precise time synchronization is required in order to perform some specific measurements on the system: examples include the voltage phasor measurement in electric power networks, via synchronized phasor measurement units, and some time-of-flight-based (GPS-like) distance measurements. Also some ancillary services in networked control systems rely on correct time sync: notably, TDMA communication (where the use of a shared communication channel is regulated by precise slotting of the access times) and energy saving mechanism (when nodes remain idle for most of the time and must wake up all together in order to initiate a communication).

For all these applications, especially for large scale systems, extremely robust solutions must be designed, in order to guarantee synchronization also in case of partial failure of the system, communication faults, node appearance and disappearance, and also possible malicious attacks. Scalability is also an issue, as we want the performance of the synchronization algorithms to be practically independent from the size of the system. On the contrary we want the reconfiguration to be minimal every time a new node enters or leaves the network, or if two networks merge. Because of these reasons, the family of time synchronization algorithms that are based on the construction of a hierarchical coordination tree, as in [1], [2], are poorly suited for these applications. Maintaining such architecture may be unbearable in many scenarios, and these solutions exhibit little robustness against the failure of any node which is not a leaf of the tree.

Other algorithms available in the literature try to circumvent the main drawbacks of tree-based solutions by constructing different architectures, like clusters of nodes, each one headed by an elected master node [3]. Master nodes then synchronize among them, at a higher level of coordination. Unless the communication architecture is specifically designed, however, there is no guarantee that master nodes can communicate more reliably over the longer distances of the high level communication layer.

In this work, instead, we adopt a fully distributed (leaderless) approach, in which all the nodes communicate with a limited number of neighbors, and each node behaves in the same way. Existing algorithms in this sense include [4] and [5], which however suffer from specific drawbacks: the algorithm proposed in [4], inspired by the fireflies integrate-and-fire synchronization mechanism, can compensate for different clock offsets but not for different clock skews; on the other hand, the algorithm proposed in [5] compensates for the clock skews but not for the time offsets.

Fully distributed protocols that can compensate for both clock skews and offsets have been proposed in [6], [7]. For these algorithms, convergence has been proved by the authors, under reasonable assumptions. The main weakness of these solutions resides in their highly nonlinear dynamic behavior, which prevents the analysis of their robustness with respect to data losses in the communication, quantization noise, communication errors, and unmodeled dynamics of the clocks. On the other hand, in [8], a linear, PI-like, distributed algorithm has been proposed for the correction of both skew and offset clock errors. The performance and the robustness of this algorithm (which closely resembles high-order consensus algorithms) have been thoroughly analyzed via numerical simulations. However, providing a formal proof of its convergence proved to be a difficult task, except for some special cases in which either the communication graph was restricted to some special families, or some assumptions were made on the asynchronous activation of the nodes.

The research leading to these results has received funding from the EU 7th Framework Programme [FP7/2007-2013] under grant agreements no. 257462 HYCON2 and no. 223866 FeedNetBack.

The authors are with the Dept. of Information Engineering, University of Padova, Via Gradenigo 6/a, 35131 Padova, Italy. Email: {saverio.bolognani|carlirug|lovisari|zampi}@dei.unipd.it.

In this paper we prove that the algorithm convergence can be guaranteed, via proper tuning of a design parameter, independently from the communication graph. The proposed algorithm can indeed be specialized to different communication strategies. According to the adopted technology, it might be easier to perform symmetric vs. asymmetric communication, and point-to-point (gossip) vs. broadcast communication. Many of the most appealing applications of multi–agent systems that we have mentioned before, are provided with some inherently broadcast communication channel (namely, wireless communication for sensor networks and power-line communication in the electric grid). For this reason, we focus in the following on the broadcast protocol, even if the main result applies to generic communication protocols.

In Section II we introduce a model for the clocks and we described the proposed algorithm. In Section III we present the main theoretical result, proving and commenting the convergence properties of the algorithm. Finally, in Section IV, we simulate the algorithm behavior and we propose a numerical comparison with other fully distributed strategies. While being slightly slower in the asymptotic converge to synchronization, the strategy we propose in this paper significantly outperforms the other strategies in terms of robustness against process and measurement noises and time-varying clock drifts.

A. Mathematical preliminaries

Before proceeding, we collect some useful definitions and notations. In this paper, $\mathcal{G} = (V, \mathcal{E})$ denotes an undirected graph where $V = \{1, \dots, N\}$ is the set of vertices and $\mathcal{E}$ is the set of edges, i.e., $\mathcal{E} \subseteq V \times V$. Since $\mathcal{G}$ is undirected, if $(i, j) \in \mathcal{E}$ then also $(j, i) \in \mathcal{E}$. A path in $\mathcal{G}$ consists of a sequence of vertices $(i_1, i_2, \dots, i_r)$ such that $(i_h, i_h+1) \in \mathcal{E}$ for every $h \in \{1, \dots, r-1\}$. A graph $\mathcal{G}$ is connected if for any pair of vertices (i, j) there exists a path connecting i to j. Given a matrix $M \in \mathbb{R}^{N \times N}$, we define the associatedu graph $\mathcal{G}_M$ by taking N nodes and putting an edge (j, i) in $\mathcal{E}_M$ if $M_{ij} \neq 0$. Given a graph $\mathcal{G}$ on V, the matrix M is compatible with $\mathcal{G}$ if $\mathcal{E}_M \subseteq \mathcal{E}$. Given the node i, by $\mathcal{N}_i$ we denote the set of its neighbors, i.e., $N_i = \{j \in V | (i, j) \in \mathcal{E}\}$. With the symbol $\mathbb{1}$ we denote the N-dimensional vector having all the components equal to 1. Given the vector $v \in \mathbb{R}^N$, by $\mathrm{diag}(v)$ we denote the diagonal matrix having the components of v as diagonal elements. Superscript $*$ denote the transpose operation.

II. Problem formulation and proposed solution

In this section we formulate the problem we aim at solving in this paper and we propose our solution, which is a modification of the one introduced in [9]. This section is divided into three parts. In subsection II-A we describe the adopted clock model. In subsection II-B we formulate the clock synchronization problem over a communication network. Finally, in subsection II-C we introduce the PI controller based on randomized asymmetric broadcast communications.

A. Mathematical modeling of a clock

Assume that each clock has an oscillator able to periodically increment a counter by one unit, commonly known as *tick*. Let Δ and $s(t)$ denote, respectively, the period of the oscillator and the evolution of the counter. Therefore

$$s(t) = \left\lfloor \frac{t - t_0}{\Delta} \right\rfloor$$

where t_0 is the time when the clock has been started and where $\lfloor a \rfloor$ denotes the largest integer smaller than or equal to a. Based on its own counter, each clock estimates the time.

The value Δ is assumed to be unknown; typically, only an estimate $\hat{\Delta}$ of it is available to the clock. Since only $\hat{\Delta}$ and s are known, a natural way to built an estimate $\hat{t}(t)$ of the absolute time t is given by

$$\hat{t}(t) = \hat{t}(t_0) + \hat{\Delta}\ (s(t) - s(t_0))\,, \tag{1}$$

where $\hat{t}(t_0)$ is an estimate of t_0 and denotes the initial offset.

Both $\hat{t}(t)$ and $\hat{\Delta}(t)$ can be modified if the clock obtains information allowing it to improve its time and oscillator period estimates. We denote by $T_{\mathrm{up}}(h)$, for $h = 0, 1, \dots$, these *updating time instants*. We can interpret the h-th update as an event which happens at time $T_{\mathrm{up}}(h)$, and such that[1]

$$\begin{cases} \hat{t}(T_{\mathrm{up}}^+(h)) = \hat{t}(T_{\mathrm{up}}(h)) + u'(h) \\ \hat{\Delta}(T_{\mathrm{up}}^+(h)) = \hat{\Delta}(T_{\mathrm{up}}(h)) + u''(h) \end{cases}$$

where u' and u'' denote the control inputs applied to $\hat{t}$ and $\hat{\Delta}$, respectively. In between consecutive updating times, the estimate $\hat{\Delta}$ is kept constant, while $\hat{t}(t)$ is updated according to (1). Thus, for $t \in \left(T_{\mathrm{up}}^+(h), T_{\mathrm{up}}(h+1)\right)$ the updating law can be written as

$$\begin{cases} \hat{t}(t) = \hat{t}(T_{\mathrm{up}}^+(h)) + \hat{\Delta}(T_{\mathrm{up}}^+(h))\,(s(t) - s(T_{\mathrm{up}}(h))) \\ \hat{\Delta}(t) = \hat{\Delta}(T_{\mathrm{up}}^+(h)) \end{cases} \tag{2}$$

We conclude this subsection by observing that $s(t) - s(T_{\mathrm{up}}(h)) = \frac{t - T_{\mathrm{up}}(h)}{\Delta} + r(h)$ where $-1 < r(h) < 1$ and so $r(h)$ can be neglected if $\Delta \ll 1$ which will be assumed in the sequel. Equation (2) can then be rewritten as

$$\begin{cases} \hat{t}(t) = \hat{t}(T_{\mathrm{up}}^+(h)) + \frac{\hat{\Delta}(T_{\mathrm{up}}^+(h))}{\Delta}\,(t - T_{\mathrm{up}}(h)) \\ \hat{\Delta}(t) = \hat{\Delta}(T_{\mathrm{up}}^+(h)) \end{cases} \tag{3}$$

B. Clock synchronization

Consider now a network composed by N clocks. For $i \in \{1, \dots, N\}$, let Δ_i be the period of the oscillator of clock i and let $x_i(t) = [x_i'(t)\ x_i''(t)]^* = [\hat{t}_i(t)\ \hat{\Delta}_i(t)]^*$ denote its local state.

Assume that the clocks can exchange their time readings $x_i'(t)$'s according to a graph of *admissible communications* $\mathcal{G} = (V, \mathcal{E})$, where $V = \{1, \dots, N\}$ and where $(i, j) \in \mathcal{E}$ whenever node i and node j can communicate.

For each clock i, $i \in \{1, \dots, N\}$, we denote by $T_{\mathrm{tx},i}(h)$, $h \in \mathbb{N}$ the time instants in which node i transmits its readings, and by $T_{\mathrm{up},i}(h')$, $h' \in \mathbb{N}$ the time instants in which it performs an update of its state based on the information

[1]Given the time t, the symbol t^+ denotes the instant just after time t.

received from its neighbors. More precisely, analogously to the case of a single clock in the previous subsection,

$$x_i(T^+_{\mathrm{up},i}(h)) = x_i(T_{\mathrm{up},i}(h)) + u_i(h), \tag{4}$$

where $u_i(h) = [u'_i(h)\ u''_i(h)]^*$ is the control action applied at time $T_{\mathrm{up},i}(h)$, while for $t \in \left(T^+_{\mathrm{up},i}(h), T_{\mathrm{up},i}(h+1)\right)$ we assume that the state x_i is updated according to (3), that is,

$$\begin{cases} x'_i(t) = x'_i(T^+_{\mathrm{up},i}(h)) + \frac{x''_i(T^+_{\mathrm{up},i}(h))}{\Delta_i}\,(t - T_{\mathrm{up},i}(h)) \\ x''_i(t) = x''_i(T^+_{\mathrm{up},i}(h)) \end{cases} \tag{5}$$

The goal is to find a control law that yields synchronization, i.e. such that there exist constants $a \in \mathbb{R}_{>0}$ and $b \in \mathbb{R}$ such that synchronization errors

$$e_i(t) := x'_i(t) - (at + b), \quad i = \{1, \dots, N\} \tag{6}$$

converge to zero or remain small.

C. A PI Controller based on randomized asymmetric broadcast communications

In this subsection we propose a control law to solve the synchronization problem stated in the previous subsection. To do so, we first need to define the data transmission and communication protocols used by the clocks to exchange information with each other. In this paper we adopt an *asymmetric broadcast communication model* where the transmission's time instants are the samples generated by N independent Poisson processes having all the same intensity. In formal terms, for each $i \in \{1, \dots, N\}$,

- the time instants $T_{\mathrm{tx},i}(h)$, $h \in \mathbb{N}$, are the sample times of a Poisson process of intensity $\lambda > 0$;
- at time $T_{\mathrm{tx},i}(h)$, $h \in \mathbb{N}$, node i transmits only the information related to the first component of its state, i.e., $x'_i(T_{\mathrm{tx},i}(h))$, to all its neighbors in the graph $\mathcal{G}$, namely, to any $j \in \mathcal{N}_i$.
- nodes $j \in \mathcal{N}_i$ receive $x'_i(T_{\mathrm{tx},i}(h))$ exactly at the time instant $T_{\mathrm{tx},i}(h)$ in which it has been transmitted (assuming negligible transmission delays)[2].

Based on the information received, nodes $j \in \mathcal{N}_i$ instantaneously update their current state $x_j(T_{\mathrm{tx},i}(h))$ with the correction

$$u_j = \begin{bmatrix} u'_j \\ u''_j \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 1 \\ \alpha \end{bmatrix}\left(x'_i(T_{\mathrm{tx},i}(h)) - x'_j(T_{\mathrm{tx},i}(h))\right)$$

where $\alpha > 0$ is a control parameter. Notice that, since there are no deliver delays, $T_{\mathrm{tx},i}(h) = T_{\mathrm{up},j}(h')$ for some $h' \geq h$. From (4), it follows that, for all $j \in \mathcal{N}_i$,

$$\begin{aligned} x'_j(T^+_{\mathrm{tx},i}(h)) &= \frac{1}{2}\left(x'_j(T_{\mathrm{tx},i}(h)) + x'_i(T_{\mathrm{tx},i}(h))\right) \\ x''_j(T^+_{\mathrm{tx},i}(h)) &= x''_j(T_{\mathrm{tx},i}(h)) + \\ &\quad + \frac{\alpha}{2}\left(x'_i(T_{\mathrm{tx},i}(h)) - x'_j(T_{\mathrm{tx},i}(h))\right). \end{aligned} \tag{7}$$

Notice that the above control law can be seen as a PI controller where $u'_j = x'_i(T_{\mathrm{tx},i}(h)) - x'_j(T_{\mathrm{tx},i}(h))$ and $u''_j = \alpha\left(x'_i(T_{\mathrm{tx},i}(h)) - x'_j(T_{\mathrm{tx},i}(h))\right)$ represent the proportional and the integral part, respectively. We refer to the control law described above as a *PI controller based on randomized asymmetric broadcast communications*.

Remark 2.1: The proposed strategy is similar to the one introduced in [9]. However, in [9], the authors adopted a randomized asymmetric gossip communication model, i.e., the information $x'_i(T_{\mathrm{tx},i}(h))$, is sent by node i to only one of its neighbors, randomly selected in $\mathcal{N}_i$ with probability $1/|\mathcal{N}_i|$.

III. Analysis

The goal of this section is to provide some theoretical insights on the convergence properties of the control strategy presented in the previous section. We start our analysis by introducing a convenient vector-form description of the evolution of the clocks' network. To do so, we need some auxiliary definitions. First of all, we stack the state variables in vectors as follows

$$x' = \begin{bmatrix} x'_1 \\ \vdots \\ x'_N \end{bmatrix} \in \mathbb{R}^N,\ x'' = \begin{bmatrix} x''_1 \\ \vdots \\ x''_N \end{bmatrix} \in \mathbb{R}^N,\ x = \begin{bmatrix} x' \\ x'' \end{bmatrix} \in \mathbb{R}^{2N}$$

and similarly for $u' \in \mathbb{R}^N$, $u'' \in \mathbb{R}^N$ and $u \in \mathbb{R}^{2N}$.

Let the matrix $E_i \in \mathbb{R}^{N \times N}$, for $i \in \{1, \dots, N\}$, be defined as

$$E_i := \sum_{j \in \mathcal{N}_i} e_j e_j^* - e_j e_i^*,$$

where e_k is defined as the vector whose value is 1 in position k and 0 elsewhere. The control action at time $T_{\mathrm{tx},i}(h)$ can therefore be rewritten as

$$u(T_{\mathrm{tx},i}(h)) = \begin{bmatrix} -\frac{1}{2}E_i & 0 \\ -\frac{\alpha}{2}E_i & 0 \end{bmatrix} x(T_{\mathrm{tx},i}(h)),$$

i.e. E_i is the "control matrix" for the broadcast of agent i. It is also clear that $E_i \mathbb{1} = 0$, which intuitively means that if the clocks are synchronized, no control is needed.

We let the matrix $D \in \mathbb{R}^{N \times N}$ be defined as

$$D = \mathrm{diag}\{d_1, \dots, d_N\}$$

where for simplicity $d_i := 1/\Delta_i$. To conclude, we let $\{T_{\mathrm{up}}(h), h \in \mathbb{N}\}$ be the set of all the updating time instants of the clocks' network, i.e.,

$$\{T_{\mathrm{up}}(h), h \in \mathbb{N}\} = \bigcup_{i=1}^{N} \{T_{\mathrm{up},i}(h), h \in \mathbb{N}\},$$

where, without loss of generality, we assume $T_{\mathrm{up}}(h) \leq T_{\mathrm{up}}(h+1)$. Notice that, for any $h \in \mathbb{N}$, there exist $i \in \{1, \dots, N\}$, $j \in \mathcal{N}_i$, and $h', h''_j \in \mathbb{N}$ with $h' \leq h$, $h''_j \leq h$, such that

$$T_{\mathrm{up}}(h) := T_{\mathrm{up},j}(h'_j) = T_{\mathrm{tx},i}(h''),\ \forall j \in \mathcal{N}_i.$$

Moreover observe that, since the N Poisson processes generating the transmission time instants are independent one from another, the updating time instants $\{T_{\mathrm{up}}(h), h \in \mathbb{N}\}$ can be seen as the sample times of a Poisson process of

[2] In general, the information $x_i(T_{\mathrm{tx},i}(h))$ is received by clock $j \in \mathcal{N}_i$ at a delayed time $T_{\mathrm{rx},i,j}(h) = T_{\mathrm{tx},i}(h) + \gamma_{i,j}(h)$, where $\gamma_{i,j}(h)$ is a nonnegative real number representing the deliver delay between i and j.

intensity $N\lambda$. We denote by $\delta(h) := T_{\rm up}(h+1) - T_{\rm up}(h)$ the interarrival time between two subsequent updates. By the properties of Poisson processes, and being $\delta(h)$ i.i.d., we have

$$\mathbb{E}\left[\delta(h)\right] = \mu = \frac{1}{N\lambda}, \qquad \mathbb{E}\left[\delta(h)^2\right] = \sigma^2 = \frac{2}{N^2\lambda^2}$$

We now consider the discrete time evolution of the state x at the update time instants $T_{\rm up}(h), h \in \mathbb{N}$. By combining (3) with (7), we obtain

$$x(h+1) = \begin{bmatrix} I & \delta(h)D \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} - \begin{bmatrix} \frac{1}{2}E(h) & 0 \\ \frac{\alpha}{2}E(h) & 0 \end{bmatrix} \right) x(h) \tag{8}$$

where for simplicity we denote $x(h) := x(T_{up}(h))$, and we set $E(h) = E_i$ if, during the h-th iteration, node i is the transmitting node. The sampled system is a stochastic time–varying system, and if we prove that it achieves synchronization, then also the original continuous-time system synchronizes, since it is autonomous (i.e., no control is applied) when no update takes place.

For the convergence analysis it is convenient to introduce the quantities $y(h) \in \mathbb{R}^{N-1}$ and $z(h) \in \mathbb{R}^{N-1}$ defined as

$$y(h) := V^* x'(h) \qquad z(h) := V^* D x''(h)$$

where $V \in \mathbb{R}^{N \times N-1}$ is a matrix whose columns form an orthonormal basis for $\mathrm{span}\{\mathbb{1}\}$, i.e. $V^*\mathbb{1} = 0$ and $V^*V = I_{N-1}$, begin I_{N-1} the $(N-1) \times (N-1)$ identity matrix. In words, $y(h)$ is an error vector which is zero only if $x'(h)$ belongs to $\mathrm{span}\{\mathbb{1}\}$, namely when the clocks are synchronized. Analogously, $z(h)$ is zero only if the vector of the estimates $x''(h)$ belongs to $\mathrm{span}\{D^{-1}\mathbb{1}\}$, namely when the slope of all the clocks is the same, i.e. $\frac{\hat{\Delta}_1}{\Delta_1} = \cdots = \frac{\hat{\Delta}_N}{\Delta_N}$. This argument shows that the synchronization error defined in (6) is zero, or asymptotically approaches this value, if both $y(h)$ and $z(h)$ vanish in time.

Our aim is to perform a mean–square analysis of the process $\begin{bmatrix} y(h) \\ z(h) \end{bmatrix}$, and thus we introduce

$$\Sigma(h) = \mathbb{E} \begin{bmatrix} y(h) \\ z(h) \end{bmatrix} \begin{bmatrix} y(h)^* & z(h)^* \end{bmatrix}$$

We say that *mean–square synchronization* is achieved if $\begin{bmatrix} y(h)^* & z(h)^* \end{bmatrix}^* \overset{t\to\infty}{\longrightarrow} 0$ in mean square, i.e. if $\Sigma(h) \overset{t\to\infty}{\longrightarrow} 0$.

In order to state our main result, we need some additional notations. Let $E = \mathbb{E}[E(h)]$, $h \in \mathbb{N}$, be the expected value of the communication matrices. Notice that since $E(h)$ is i.i.d. and uniform, $E = \mathbb{E}[E_i] = \frac{1}{N}\sum_{i \in V} E_i$.

It is easy to see that the admissible communication graph $\mathcal{G} = (V, \mathcal{E})$ is the graph induced by E, namely, an edge (i,j) exists in $\mathcal{E}$ if and only if namely if $[E]_{ij} \neq 0$.

We can now state the following result, which gives sufficient conditions for mean–square synchronization to take place. The proof is postponed to the next section.

Theorem 3.1: Assume that the admissible communications graph $\mathcal{G} = (V, \mathcal{E})$ is connected. Assume moreover that the matrix inequality

$$I_{N-1} \otimes \bar{E}^{-1}\bar{F} + \bar{E}^{-1}\bar{F} \otimes I_{N-1} > 0 \tag{9}$$

is satisfied, where $\bar{E} = V^*EV$ and $\bar{F} = V^*DEV$. Then there exists a value $\alpha^* > 0$ such that, for any $\alpha \in (0, \alpha^*)$, mean–square synchronization is achieved.

Remark 3.2: The result stated in Theorem 3.1 holds true not only when the asymmetric broadcast communication protocol is adopted but also for any other communication protocol, like the symmetric gossip [10] and the asymmetric gossip [9].

Among the hypoteses of Theorem 3.1, connectivity of $\mathcal{G}$ is clearly a necessary condition, as otherwise the graph would be divided into two or more components which do not communicate, and thus cannot, in general, synchronize. Condition (9) is a bit more involving, however it is always verified in some notable cases, as the following corollary states.

Corollary 3.3: Assume that the admissible communications graph $\mathcal{G} = (V, \mathcal{E})$ is connected. Assume moreover $E = E^T$. Then there exists a value $\alpha^* > 0$ such that for any $\alpha \in (0, \alpha^*)$ mean–square synchronization is achieved.

The result of Corollary 3.3 is quite remarkable since, provided that $E = E^T$, it ensures that existence of $\alpha^* > 0$ for any matrix $D > 0$ (i.e. for any difference in the oscillators' frequency). It can be shown that the condition $E = E^T$ is satisfied in particular in the asymmetric broadcast protocol adopted in this paper. Furthermore, $E = E^T$ is also true in some other notable scenarios. These include the particular case of highly regular graphs, like Cayley graphs [11], and the case of symmetric protocols, like the *symmetric gossip* [10], in which the matrix $E(h)$ is extracted from a family of symmetric matrices.

Remark 3.4: Condition (9) *is automatically verified if $D = I$, for any (possibly asymmetric) communication protocol. Since the dynamics of Σ are ruled by an operator whose eigenvalues depend continuously on the matrix D, the existance of $\alpha^* > 0$ that achieves mean–square synchronization is guaranteed even in the case where D is a small enough perturbation of the identity matrix I.*

This is the approach followed by the authors in [9], where they considered the same control law analyzed in this paper, based however on the asymmetric gossip protocol. They performed a convergence study of the evolution of Σ assuming that $\mathcal{G}$ is the complete graph and that $D = I$. Under these assumptions they found that synchronization is achieved if and only if $\alpha < \alpha^* = \lambda/2$.

In general, for a given matrix D, one needs to check condition (9) numerically, thus performing a robustness analysis on the values of oscillator periods $\hat{\Delta}_i$, $i \in \{1, \ldots, N\}$, for which the PI controller yields the synchronization.

Remark 3.5: Theorem 3.1 offers an answer to the problem of mean–square synchronization, since it ensures the existence of a consensus–like scheme capable of achieving synchronization, under minimal assumptions on $\mathcal{G}$. It remains to study how the amplitude of the maximum α^* depends on the adopted protocol and how it scales with the size of the network. This issue is important since it is in quite intuitive that the smaller is α, the slower the clocks reach the synchronization.

A. Proof of Theorem 3.1 and Corollary 3.3

In order to prove our results, we first need to write in a suitable way the evolution of $\Sigma(h)$. In order to do this, notice that $\Omega := VV^* = I_N - \frac{1}{N}\mathbb{1}\mathbb{1}^T$. In this section I will denote the $(N-1)\times(N-1)$ identity. Using the fact that for any $i \in \{1,\dots,N\}$, $E_i\Omega = E_i$, by easy computations one can see that the evolution of y and z is described by the following iteration

$$\begin{bmatrix} y(h+1) \\ z(h+1) \end{bmatrix} = \begin{bmatrix} I - \frac{1}{2}\tilde{E}(h) - \frac{\alpha\delta(h)}{2}\tilde{F}(h) & \delta(h) \\ -\frac{\alpha}{2}\tilde{F}(h) & I \end{bmatrix} \begin{bmatrix} y(h) \\ z(h) \end{bmatrix}$$

where $\tilde{E}(h) = V^*E(h)V$ and $\tilde{F}(h) = V^*DE(h)V$.

We can thus write

$$\Sigma(h) = \mathbb{E}\left[\begin{bmatrix} y(h) \\ z(h) \end{bmatrix} \begin{bmatrix} y^*(h) & z^*(h) \end{bmatrix} \right] = \begin{bmatrix} \Sigma_{yy}(h) & \Sigma_{yz}(h) \\ \Sigma^*_{yz}(h) & \Sigma_{zz}(h) \end{bmatrix},$$

where

$$\begin{aligned} \Sigma_{yy}(h) &:= \mathbb{E}[y(h)y^*(h)], \quad & \Sigma_{yz}(h) &:= \mathbb{E}[y(h)z^*(h)], \\ \Sigma_{zy}(h) &:= \mathbb{E}[z(h)y^*(h)], \quad & \Sigma_{zz}(h) &:= \mathbb{E}[z(h)z^*(h)]. \end{aligned}$$

The assumption on statistical independence on the choice of $E(h)$ and of the updating times allows to write

$$\Sigma(h+1) = \mathbb{E}[A(h)\Sigma(h)A^*(h)] \tag{10}$$

where

$$A(h) := \begin{bmatrix} I - \frac{1}{2}\tilde{E}(h) - \frac{\alpha\delta(h)}{2}\tilde{F}(h) & \delta(h) \\ -\frac{\alpha}{2}\tilde{F}(h) & I \end{bmatrix}.$$

Simple manipulation yields then

$$\begin{aligned} \Sigma^+_{yy} = \mathbb{E}\Bigg[& \left(I - \frac{1}{2}\tilde{E}(h) - \frac{\alpha\delta(h)}{2}\tilde{F}(h) \right) \Sigma_{yy} \times \\ & \times \left(I - \frac{1}{2}\tilde{E}(h)^* - \frac{\alpha\delta(h)}{2}\tilde{F}(h)^* \right) \\ & + \lambda\Sigma_{zy} \left(I - \frac{1}{2}\tilde{E}(h)^* - \frac{\alpha\delta(h)}{2}\tilde{F}(h)^* \right) \\ & + \lambda \left(I - \frac{1}{2}\tilde{E}(h) - \frac{\alpha\delta(h)}{2}\tilde{F}(h) \right) \Sigma_{yz} \\ & + \delta(h)^2\Sigma_{zz} \Bigg] \end{aligned}$$

$$\begin{aligned} \Sigma^+_{yz} = \mathbb{E}\Bigg[& -\frac{\alpha}{2} \left(I - \frac{1}{2}\tilde{E}(h) - \frac{\alpha\delta(h)}{2}\tilde{F}(h) \right) \Sigma_{yy}\tilde{F}(h)^* \\ & + \left(I - \frac{1}{2}\tilde{E}(h) - \frac{\alpha\delta(h)}{2}\tilde{F}(h) \right) \Sigma_{yz} \\ & - \frac{\alpha\delta(h)}{2}\Sigma_{zy}\tilde{F}(h)^* + \delta(h)\Sigma_{zz} \Bigg] \end{aligned}$$

$$\begin{aligned} \Sigma^+_{zy} = \mathbb{E}\Bigg[& -\frac{\alpha}{2}\tilde{F}(h)\Sigma_{yy} \left(I - \frac{1}{2}\tilde{E}(h)^* - \frac{\alpha\delta(h)}{2}\tilde{F}(h)^* \right) \\ & + \Sigma_{zy} \left(I - \frac{1}{2}\tilde{E}(h)^* - \frac{\alpha\delta(h)}{2}\tilde{F}(h)^* \right) \\ & - \frac{\alpha\delta(h)}{2}\tilde{F}(h)\Sigma_{yz} + \delta(h)\Sigma_{zz} \Bigg] \end{aligned}$$

$$\begin{aligned} \Sigma^+_{zz} = \mathbb{E}\Bigg[& \frac{\alpha^2}{4}\tilde{F}(h)\Sigma_{yy}\tilde{F}(h)^* - \frac{\alpha}{2}\tilde{F}(h)\Sigma_{yz} \\ & - \frac{\alpha}{2}\Sigma_{zy}\tilde{F}(h)^* + \Sigma_{zz} \Bigg] \end{aligned}$$

Define now $\bar{E} = \mathbb{E}[\tilde{E}(h)]$, $\bar{F} = \mathbb{E}[\tilde{F}(h)]$ and

$$\mathbb{E}_{QR} = \mathbb{E}[Q \otimes R]$$

where $Q, R \in \{E, F\}$.

Once we set

$$\begin{aligned} Y &= \operatorname{vec} \Sigma_{yy} & W &= \operatorname{vec} \Sigma_{yz} \\ W' &= \operatorname{vec} \Sigma_{zy} & Z &= \operatorname{vec} \Sigma_{zz} \end{aligned}$$

it is easy, making use of the properties of the Kronecker product and sum[3], to obtain the following iteration rule

$$\begin{bmatrix} Y \\ W \\ W' \\ Z \end{bmatrix}^+ = \left(M_0 + \alpha M_1 + \alpha^2 M_2 \right) \begin{bmatrix} Y \\ W \\ W' \\ Z \end{bmatrix}$$

where, called $\bar{A} = -\frac{1}{2}\bar{E} \oplus \bar{E} + \frac{1}{4}\mathbb{E}_{EE}$,

$$M_0 = \begin{bmatrix} I + \bar{A} & \mu(I - \frac{1}{2}I \otimes \bar{E}) & \mu(I - \frac{1}{2}\bar{E} \otimes I) & \sigma^2 I \\ 0 & I - \frac{1}{2}I \otimes \bar{E} & 0 & \mu I \\ 0 & 0 & I - \frac{1}{2}\bar{E} \otimes I & \mu I \\ 0 & 0 & 0 & I \end{bmatrix}$$

and, called $\bar{B} = -\frac{\mu}{2}(\bar{F} \oplus \bar{F} - \frac{1}{2}\mathbb{E}_{EF} - \frac{1}{2}\mathbb{E}_{FE})$,

$$M_1 = \begin{bmatrix} \bar{B} & -\frac{\sigma^2}{2}I \otimes \bar{F} & -\frac{\sigma^2}{2}\bar{F} \otimes I & 0 \\ -\frac{1}{2}(\bar{F} \otimes I - \frac{1}{2}\mathbb{E}_{FE}) & -\frac{\mu}{2}I \otimes \bar{F} & -\frac{\mu}{2}\bar{F} \otimes I & 0 \\ -\frac{1}{2}(I \otimes \bar{F} - \frac{1}{2}\mathbb{E}_{EF}) & -\frac{\mu}{2}I \otimes \bar{F} & -\frac{\mu}{2}\bar{F} \otimes I & 0 \\ 0 & -\frac{1}{2}I \otimes \bar{F} & -\frac{1}{2}\bar{F} \otimes I & 0 \end{bmatrix}$$

and finally

$$M_2 = \begin{bmatrix} \frac{\sigma^2}{4}\mathbb{E}_{FF} & 0 & 0 & 0 \\ \frac{\mu}{4}\mathbb{E}_{FF} & 0 & 0 & 0 \\ \frac{\mu}{4}\mathbb{E}_{FF} & 0 & 0 & 0 \\ \frac{1}{4}\mathbb{E}_{FF} & 0 & 0 & 0 \end{bmatrix}$$

We have thus rewritten the evolution of the matrix $\Sigma(h)$ as a linear system governed by a matrix $M(\alpha)$ dependent on the design parameter α.

In order to prove Theorem 3.1, we make use of the following perturbation result, taken from [12], in which we call an eigenvalue semi–simple if its algebraic and geometric multiplicities coincide.

Theorem 3.6: Let be $M(\alpha) \in \mathbb{R}^{N\times N}$ be a matrix dependent on the parameter α in a sufficiently smooth way so that the first derivative $\dot{M}(\alpha)|_{\alpha=0}$ exists. Let moreover $\mu_1,\dots,\mu_m$ be semi–simple eigenvalues of $M(\alpha)$ with associated right eigenvectors $r_1,\dots,r_m$ and left eigenvectors

[3]The Kronecker sum of A and B is defined as $A \oplus B = A \otimes I + I \otimes B$, where the identities are of suitable dimensions.

$l_1^T, \dots, l_m^T$. Assume that these families of eigenvectors are chosen such that if

$$\mathcal{R} = \begin{bmatrix} r_1 & \dots & r_m \end{bmatrix} \qquad \mathcal{L} = \begin{bmatrix} l_1 & \dots & l_m \end{bmatrix}^*$$

then $\mathcal{L}\mathcal{R} = I_m$, where I_m is the $m \times m$ identity. Then the derivative of μ_i w.r.t. α, for $\alpha = 0$, exists and is the i-th eigenvalue of the matrix $\mathcal{L}M'\mathcal{R}$ where $M' = \dot{M}(\alpha)|_{\alpha=0}$.

Our scope here is to use this theorem in order to study the eigenvalues of the matrix $M(\alpha)$ for α small and positive.

First of all, we need to introduce some other notations and a technical result. Given $E_i \in \mathcal{S}$, we let $P_i = I - \frac{1}{2}E_i$ and $P = I - \frac{1}{2}E$. Notice that each P_i is a row–stochastic matrix, and P is a primitive matrix under the assumption $\mathcal{G}$ to be connected. By Frobenius–Perron theorem $Pv = v$ if and only if $v = \beta\mathbb{1}$. Moreover, the left eigenvalue of P associated with 1, which is usually denoted by π^T, has only strictly positive entries. We normalize π^T so that $\pi^T\mathbb{1} = 1$.

The following fact holds.

Lemma 3.7: Assume that $\mathcal{G} = (V, \mathcal{E})$ is strongly connected and for any i, $P_{ii} > 0$ with nonzero probability. Then the matrix M_0 has exactly $(N-1)^2$ semi–simple eigenvalues in 1, and the other eigenvalues are stable, namely, in absolute value less than 1.

Proof: As it is clear from the upper-block-triangular structure of M_0, this matrix has at least $(N-1)^2$ eigenvalues in 1, so first of all we need to check that the other blocks have only stable eigenvalues. First of all,

$$I - \frac{1}{2}\bar{E} = V^*(I - \frac{1}{2}\mathbb{E}E_i)V = V^*PV$$

and since $\bar{\mathcal{G}}$ is strongly connected, P has only stable eigenvalues and a unique eigenvalue in 1 associated with $\mathbb{1}$. It is an easy exercise to see that V^*PV has all the eigenvalues of P apart that in 1. This proves that $(I - \frac{1}{2}\bar{E}) \otimes I$ is stable, and analogously $I \otimes (I - \frac{1}{2}\bar{E})$, due to the properties of Kronecker product.

It remains to analyze the first block of M_0, which is

$$(V^* \otimes V^*) \left(\mathbb{E}(I - \frac{1}{2}E_i) \otimes (I - \frac{1}{2}E_i) \right) (V \otimes V)$$

Applying Proposition 4.3 in [13], we know that the middle matrix is row–stochastic and primitive. Analogously to the previous case, we conclude for the stability of the block. ■

As a side–consequence of the Lemma, the matrix $A = -\frac{1}{2}\bar{E} \oplus \bar{E} + \frac{1}{4}\mathbb{E}_{EE}$ turns out to be invertible.

Consider now the following matrices

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 & 2\bar{E}^{-1} \otimes \bar{E}^{-1} \end{bmatrix}$$

$$\mathcal{R} = \begin{bmatrix} \bar{A}^{-1} \left(\mu^2 \bar{E} \oplus \bar{E} + \frac{1}{2}(\sigma^2 - 2\mu^2)\bar{E} \otimes \bar{E} \right) \\ \mu\bar{E} \otimes I \\ \mu I \otimes \bar{E} \\ \frac{1}{2}\bar{E} \otimes \bar{E} \end{bmatrix}$$

It is easy to check the following equalities

$$\mathcal{L}M_0 = \mathcal{L},\ M_0\mathcal{R} = \mathcal{R},\ \mathcal{L}\mathcal{R} = I_{N-1}$$

and that both $\mathcal{L}$ and $\mathcal{R}$ are (row– and column–, respectively) full–rank matrices. Thus the $N-1$ eigenvalues in 1 of M_0 are semi–simple. We can now prove our result.

Proof: [of Theorem 3.1] By assumption $\mathcal{G}$ is connected, thus the previously defined matrices exist and the eigenvalues in 1 are semi–simple. And we can thus apply Theorem 3.6, which in our case reads

$$\mathcal{L}M_0\mathcal{R} = -\mu^2(I \otimes \bar{E}^{-1}\bar{F} + \bar{E}^{-1}\bar{F} \otimes I)$$

Since $\lambda > 0$, the derivative of all the eigenvalues in 1 of M_0 is strictly negative for $\alpha > 0$ small enough. Thus there exists α^* such that if $\alpha \in (0, \alpha^*)$, $M(\alpha)$ is a stable matrix, and then $\Sigma(h) \overset{t\to\infty}{\longrightarrow} 0$, i.e. we achieve mean–square synchronization. ■

If we assume $E = E^T$ we can prove our second result.

Proof: [of Corollary 3.3] If $E = E^T$ is symmetric, then not only $E = E\Omega$, but also $E = \Omega E$. We can thus write $\bar{F} = V^*DEV = V^*DVV^*EV = \bar{D}\bar{E}$, where $\bar{D} = V^*DV$. Thus

$$I \otimes \bar{E}^{-1}\bar{F} + \bar{E}^{-1}\bar{F} \otimes I = \left(E^{-1} \otimes E^{-1}\right)\left(I \otimes \bar{D} + \bar{D} \otimes I\right)(E \otimes E)$$

so that $I \otimes \bar{E}^{-1}\bar{F} + \bar{E}^{-1}\bar{F} \otimes I > 0$ if and only if $I \otimes \bar{D} + \bar{D} \otimes I > 0$. Since the generic eigenvalue of this last sum is the sum of any pair of eigenvalues of $I \otimes \bar{D}$ and $\bar{D} \otimes I$, we only have to check that the eigenvalues of $\bar{D} = V^*DV$ are strictly positive, which is trivial if $D > 0$. ■

IV. Numerical Examples

A. Implementation examples of the PI consensus controller algorithm

In this section we provide a numerical example illustrating the PI consensus controller algorithm proposed in this paper.

We consider a connected random geometric graph generated by choosing $N = 100$ points uniformly distributed in the unit square, and then placing an edge between each pair of points at distance less than 0.1. In Figure 1 we plot the trajectories of the quantity $\log N^{-1/2}\|e(h)\|$ for three different values of α, precisely $\alpha = \lambda, \lambda/5, \lambda/10$, where we set $\lambda = 0.01$. In all the simulations we run we assume the initial condition $x_i'(0)$ uniformly distributed in $[0, 10]$. Moreover the plots reported are the result of the average over 1000 Monte Carlo runs, randomized with respect to both the graph and the initial conditions.

Observe that the all the trajectories converge to zero exponentially and that the speed of convergence depends on the value of the control parmeter α.

B. Comparison with other distributed strategies

In this section we provide a comparison between the approach we propose in this paper and the one pursued in [7], based on the cascade of two consensus algorithms. Precisely, the goal is to compare the performance, in terms of robustness to both noisy transmission data and time-varying oscillator periods, between the algorithm described in Section II-C and the *Average TimeSync* algorithm (denoted hereafter with the shorthand ATS), introduced in [7].

For the sake of clearness, we start by briefly reviewing the ATS algorithm. As in previous sections, let $\mathcal{G}$ be a connected undirected graph. The ATS algorithm is the following.

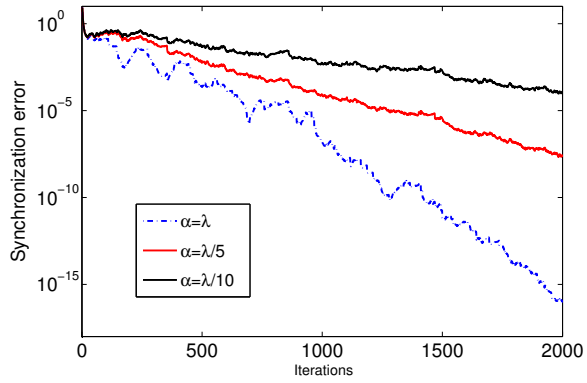

Fig. 1. Trajectories of the synchronization error generated by the *PI consensus strategy* algorithm for different values of α.

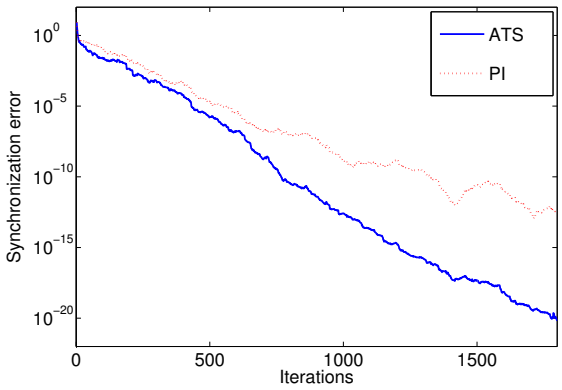

Fig. 2. Comparison in terms of speed of convergence between the approach proposed in this paper and the ATS algorithm.

Processor states: Recall that, for $i \in \{1, \dots, N\}$, the i-th node has a local clock that, according to the notation used in [14], we denote in this subsection as $\tau_i(t)$, $t \in \mathbb{R}_{\geq 0}$; specifically, $\tau_i(t) = d_i t + x'_i(0)$ where d_i and $x'_i(0)$ are as in Section II.

At any time instant t, the i-th node keeps in memory the values $\alpha_i(t)$, $\gamma_i(t)$, $\bar{\tau}_i(t)$ and $\{\eta_{ij}(t)\}_{j \in \mathcal{N}_i}$, where α_i, γ_i, $\bar{\tau}_i$ and η_{ij}, $j \in \mathcal{N}_i$, are auxiliary variables. Moreover the i-th node stores in memory also the value $\tau_i(T_{tx,i}(h_{s_i}))$ and the values $\tau_i(T_{tx,j}(h_{s_j}))$ where, for $j \in \mathcal{N}_i \cup \{i\}$, $T_{tx,j}(h_{s_j})$ denotes the instant in which node j performed its last transmission before time t, i.e., $t \in (T_{tx,j}(h_{s_j}), T_{tx,j}(h_{s_j}+1))$

Transmission and Updating step: At time $T_{tx,i}(h)$ node i performs its h-th transmission broadcasting to all its neighbors the data $\tau_i(T_{tx,i}(h-1))$, $\tau_i(T_{tx,i}(h))$, $\alpha_i(T_{tx,i}(h))$ and $\bar{\tau}_i(T_{tx,i}(h))$. For $j \in \mathcal{N}_i$, node j istantaneously performs the following actions in order

1) it receives the data $\tau_i(T_{tx,i}(h-1))$, $\tau_i(T_{tx,i}(h))$, $\alpha_i(T_{tx,i}(h))$ and $\bar{\tau}_i(T_{tx,i}(h))$;
2) it estimates the relative clock skew $\eta_{ji} := d_i/d_j$ by computing

$$\eta_{ji}(T_{tx,i}(h)^+) = \rho\, \eta_{ji}(T_{tx,i}(h)) + (1-\rho) \frac{\tau_i(T_{tx,i}(h)) - \tau_i(T_{tx,i}(h-1))}{\tau_j(T_{tx,i}(h)) - \tau_j(T_{tx,i}(h-1))}$$

where ρ is a filtering parameter;
3) it updates the variable α_j according to

$$\alpha_j(T_{tx,i}(h)^+) = \frac{1}{2}\,\alpha_j(T_{tx,i}(h)) + \frac{1}{2}\,\eta_{ji}(T_{tx,i}(h)^+)\,\alpha_i(T_{tx,i}(h))$$

4) it updates the variable γ_j according to

$$\gamma_j(T_{tx,i}(h)^+) = \gamma_j(T_{tx,i}(h)) + \frac{1}{2}\left(\bar{\tau}_i(T_{tx,i}(h)) - \bar{\tau}_j(T_{tx,i}(h))\right)$$

5) it updates the variable $\bar{\tau}_j$ according to

$$\bar{\tau}_j(T_{tx,i}(h)^+) = \alpha_j(T_{tx,i}(h)^+)\tau_j(T_{tx,i}(h)) + \gamma_j(T_{tx,i}(h)^+)$$

Test 4.1 (Speed of convergence to synchronization): We provide here a comparison of the speed of convergence of the proposed algorithms and of the ATS algorithm. We consider a random geometric graph $\mathcal{G}$, where vertices are 30 points uniformly distributed in the unit square, and nodes whose distance is smaller than 0.4 are connected.

In Figure 2, we depict for both strategies the behavior of $\log N^{-1/2}\|e\|$ for both strategies being e the synchronization error defined as $e = \Omega x'$ for the PI strategy and $e = \Omega\bar{\tau}$ for the ATS algorithm, where $\Omega = I - \frac{1}{N}\mathbb{1}\mathbb{1}^T$. The dashed red curve refers to the PI strategy, while the solid blue curve to the ATS algorithm. The two strategies have been simulated using the same transmission times. Moreover the control parameters of both algorithms have been experimentally designed in order to maximize the speed of convergence.

The plots reported have been obtained averaging over 1000 simulations; a different random geometric graph and initial conditions are independently generated for each simulation. From Figure 2, one can see that the ATS algorithm outperforms, with the respect to the speed of convergence, the performance of the PI consensus strategy.

Test 4.2: (*Robustness with the respect to communication noise and time-varying oscillator frequencies*):
We now assume that

1) the information exchanged by the nodes is affected by communication noise; and
2) the oscillator periods are time-varying.

Specifically, we assume that if $x'_j(T_{tx,j}(h))$ is any information transmitted by node j to node i at time $T_{tx,j}(h)$, then node i receives the information $x'_j(T_{tx,j}(h)) + n_{j\to i}(h)$ where $n_{j\to i}(h)$ is a white noise of bounded support. As far as the oscillator periods are concerned, we assume that they are modeled as saturated random walks, namely, for each $i \in \{1, \dots, N\}$, the value of Δ_i is always within the interval $[\Delta - \epsilon, \Delta + \epsilon]$ for some ϵ such that $0 < \epsilon < \Delta$, where we

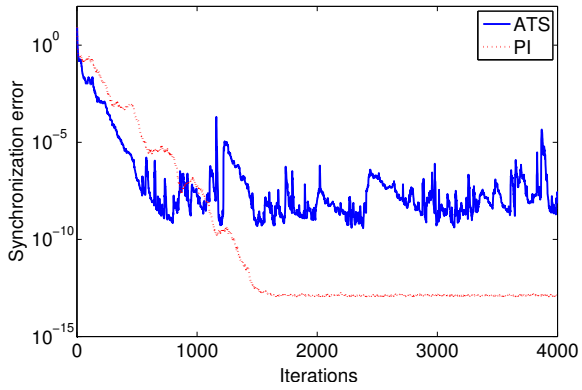

Fig. 3. Comparison in terms of robustness to communication noises and time-varying oscillator periods between the PI consensus strategy proposed in this paper and the ATS algorithm.

recall Δ represents the nominal value of the oscillator period. Precisely, for $i \in \{1, \dots, N\}$,

$$\Delta_i(T_{up}(h+1)) = \mathrm{Sat}_{\Delta,\epsilon}\left[\Delta_i(T_{up}(h)) + n_i(T_{up}(h))\right]$$

where $n_i(T_{up}(h))$ is a white noise of bounded support and, for $\Delta, \epsilon \in \mathbb{R}$ such that $0 < \epsilon < \Delta$,

$$\mathrm{Sat}_{\Delta,\epsilon}(x) = \begin{cases} x & \text{if} \quad \Delta - \epsilon \leq x \leq \Delta + \epsilon \\ \Delta - \epsilon & \text{if} \quad x < \Delta - \epsilon \\ \Delta + \epsilon & \text{if} \quad x > \Delta + \epsilon \end{cases}$$

The behavior of $\log N^{-1/2}\|e\|$ is plotted in Figure 3 for both strategies.

The control parameters for both algorithms have been chosen experimentally in such a way to minimize the steady state value of $\|e\|$. Data have been obtained averaging over 1000 simulations; a different random geometric graph and initial conditions are generated for each simulation. From Figure 3, one can see that the PI consensus strategy outperforms, with respect to robustness to communication noises and time-varying oscillator periods, the ATS algorithm.

Remark 4.3: Besides the comparison provided in the previous two examples, it is worth stressing also the fact that the ATS algorithm requires, in general, higher computational and memory capabilities than the PI strategy. In the ATS algorithm, each node has to perform non-linear updates and has to keep in memory a number of variables which is proportional to the number of its neighbors.

Remark 4.4: The Distributed Time-Sync Protocol is an another fully distributed algorithm recently proposed in the literature to solve the clock synchronization problem, see [6]. The authors of [6], for simplicity, restrict themselves to the case where all the clocks have exactly the same constant oscillator period, but have different initial offsets. The offsets compensation is posed as a least-squares problem which is solved in a distributed way through a gradient-based method. The authors mention that a similar least-squares approach could be used also to solve the case of different clocks frequencies. We have run a number of simulations implementing this cascade of two least-squares solvers and we have observed that the performance of this protocol is comparable with the performance of the ATS algorithm with the respect to both the speed of convergence and robustness.

V. Conclusions

In this paper we have considered a recently proposed randomized strategy for time synchronization of a network of clocks, adopting in particular an asymmetric broadcast communication protocol. Our main result shows that under mild conditions in the admissible communications, it is always possible to tune the control parameter α in order to achieve (robust) mean–square synchronization. A comparison with other distributed strategies has also been performed, showing slower convergence but higher resilience to noise and uncertainties. Future work includes the study of the maximum allowed α and a more extensive comparison of the proposed strategy versus its competitors.

References

[1] S. Ganeriwal, R. Kumar, and M. Srivastava, "Timingsync protocol for sensor networks," in *Proceedings of the first international conference on Embedded networked sensor systems (SenSys'03)*, 2003.

[2] M. Maròti, B. Kusy, G. Simon, and Àkos Lèdeczi, "The flooding time synchronization protocol," in *Proc. of the 2nd international conf. on Embedded networked sensor systems (SenSys'04)*, 2004, pp. 39–49.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th symposium on Operating systems design and implementation (OSDI'02)*, 2002, pp. 147–163.

[4] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *ACM Conference on Embedded Networked Sensor Systems (SenSys'05)*, San Diego, November 2005.

[5] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP Journal on wireless sensor networks*, 2007.

[6] R. Solis, V. Borkar, and P. R. Kumar, "A new distributed time synchronization protocol for multihop wireless networks," in *45th IEEE Conference on Decision and Control (CDC'06)*, San Diego, December 2006, pp. 2734–2739.

[7] L. Schenato and F. Fiorentin, "Average timesynch: a consensus-based protocol for clock synchronization in wireless sensor networks," *Automatica*, vol. 47, no. 9, pp. 1878–1886, 2011.

[8] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Optimal synchronization for networks of noisy double integrators," *IEEE Transactions on Automatic Control*, vol. 56, no. 5, pp. 1146–1152, May 2011.

[9] R. Carli, E. D'Elia, and S. Zampieri., "A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks," in *Proceedings of the 50th IEEE Conference on Decision and Control. CDC'11*, 2011.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508 – 2530, june 2006.

[11] L. Babai, "Spectra of cayley graphs," *Journal of Combinatorial Theory, Series B.*, pp. 27:180–189, 1979.

[12] K. Cai and H. Ishii, "Average consensus on general digraphs," 1988, available Online.

[13] F. Fagnani and S. Zampieri, "Randomized consensus algorithms over large scale networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, pp. 634–649, 2008.

[14] L. Schenato and F. Fiorentin, "Average timesync: A consensus-based protocol for time synchronization in wireless sensor networks," in *Proceedings of 1st IFAC Workshop on Estimation and Control of Networked Systems (NecSys09)*, September 2009.