A PI consensus controller with gossip communication for clock synchronization in wireless sensors networks ⋆

Saverio Bolognani * **Ruggero Carli** ** **Sandro Zampieri** ***

* *Department of Information Engineering, University of Padova, Italy (e-mail: saverio.bolognani@dei.unipd.it).*
** *CCDC, University of California at Santa Barbara, CA 93106, USA (e-mail: carlirug@engr.ucsb.edu)*
*** *Department of Information Engineering, University of Padova, Italy (e-mail: zampi@dei.unipd.it).*

Abstract: In this paper a distributed clock synchronization algorithm is proposed. The algorithm requires gossip asynchronous communication between the nodes of the network, and because of its proportional-integral (PI) structure it is able to compensate both initial offsets and different clock speeds. Convergence of the algorithm is proved and analysed with respect to the controller parameter, while scalability issues are addressed by simulations.

Keywords: consensus, randomized algorithms, gossip algorithms, clock synchronization

1. INTRODUCTION

The field of sensor networks has been witnessing a large attention in the scientific community during the last decade. This is due in no small part to the remarkable advances which have been made in recent years in the development of small, relatively inexpensive sensor nodes with networking capabilities, ultimately intended for a wide variety of purposes such as search and rescue, environmental monitoring and surveillance just to mention few. Wireless sensor networks (WSNs) have the potential to truly revolutionize the way we monitor and control our environment. Several WSN applications either benefit from, or require, time synchronization. These applications include, for example, mobile target tracking using a large number of motion detection devices as in Oh et al. (2005), habitat monitoring by Szewczyk et al. (2004), power scheduling and TDMA communication schemes by Hohlt et al. (2004), and rapid synchronized coordination of powerlines nodes in electric power distribution networks for catastrophic power-outage prevention as in Amin and Schewe (2007).

One possible approach to the synchronization problem is electing a reference node and allowing every node to communicate with it. To deal with the problem that communication in a sensor network usually happens through multi-hop paths, these algorithms, for example the Time-synchronization Protocol for Sensor Networks (TPSN) by Ganeriwal et al. (2003) and the Flooding Time Synchronization Protocol (FTSP) by Maròti et al. (2004), require that first a spanning tree rooted in the clock reference is built, and then the offset of any node with the respect to the root is obtained simply by adding the offset of the edges in the unique path from each node to the root.

Although these strategies can be easily implemented and Maròti et al. (2004) showed that they exhibit remarkable performances, it suffers from two main problems. The first problem is robustness. In fact, if a node dies or a new node is added to the network, then it is necessary to rebuild the tree, at the price of additional implementation overhead and possibly long periods in which the network or part of it is poorly synchronized. The second problem is that, depending on how the tree is built, it might happen that two clocks, which are physically close and can communicate with each other, belong to two different branches of the tree or two different clusters, thus possibly having large clock differences.

Recently fully distributed algorithms for clocks synchronization have appeared. The Reachback Firefly Algorithm (RFA) by Werner-Allen et al. (2005), a protocol inspired by the fireflies integrate-and-fire synchronization mechanism, is able to compensate for different clock offsets but not for different clock skews. On the opposite, the algorithm proposed by Simeone and Spagnolini (2007), adopting a P-controller, is able to compensate for the clock skews but not for the offsets, obtaining constant time differences between the clocks.

Distributed protocols that can compensate for both clock skews and offsets are the Tiny-Sync Protocol by Yoon et al. (2007), the Distributed Time-Sync Protocol by Solis et al. (2006) and the Average Time-Sync Protocols by Schenato and Gamba (2007). The first one is based on a type of robust linear regression, the second on distributed least-square estimator, and the last on a cascade of two consensus algorithms. They are all proved to synchronize a network of clocks in the absence of noise and delivery time-delay and they also show good performance in experimental testbeds. However, in these protocols it is difficult to predict the effect of noise on the steady-state performance.

Differently, Carli et al. (2008) proposed a synchronization algorithm that can be formally analyzed not only in the noiseless scenario in terms of rate of convergence but also in a noisy setting in terms of the steady-state synchronization error. This algorithm compensates for both initial offsets and differences in internal clock speeds and is based on a Proportional-Integral (PI) controller. Both convergence guarantees as well optimal design using standard optimization tools when the underlaying communication

---

⋆ The research leading to these results has received part of its funding from the European Community's Seventh Framework Programme under agreement n. FP7-ICT-223866-FeedNetBack.

graph is known are provided. It is important to remark that the time-synchronization algorithm proposed by Carli et al. (2008) require each node to perform all the operations related to the k-th iteration of the algorithm, including transmitting messages, receiving messages and updating estimates, within a short time window. This pseudo-synchronous implementation might be very sensitive to packet losses, node and link failure.

The main contribution of this paper is to develop and analyze a far more practical version of the PI synchronization algorithm, assuming that clocks can communicate by a symmetric gossip protocol. That is, at each iteration one node can estabilish a bi-directional communication with only one of its neighbors. This applies very well to real sensor networks, and drastically reduces the network requirements in terms of reliability, bandwidth, and synchronization. Theoretical results are provided when the underlying communication topology is given by the complete graph, while more general families of graphs are considered by means of simulations. The outline of the paper is as follows. In section 2 the clock model for each node is presented, together with the communication capabilities of the network and with the proposed synchronization algorithm. In section 3 the algorithm is modified to introduce gossip communication between the nodes, while in section 4 its convergence is studied through the variance of the synchronization error in time. In section 5 the dependance of the stability of the protocol on the algorithm parameters and on the network size is addressed. Simulations in section 6 confirm the obtained results, and point out some scalability issues of the algorithm. These issues are addressed in section 7. Section 8 presents some directions of future investigation.

2. PROBLEM FORMULATION

Assume that we have N clocks. Each clock has its own speed, denoted by d_i and its own initial offset, denoted by o_i. Let us denote with x_i the *local time* estimate of node i, $i \in \{1, \ldots, N\}$. The objective is to make all $x_i(t)$'s as close as possible to each other for all times t. This can be mathematically formulated by requiring that

$$\bar{x}_i(t) := x_i(t) - \frac{1}{N}\sum_{j=1}^{N} x_j(t)$$

goes to zero as t goes to infinity. The variable $\bar{x}_i(t)$ is called the *synchronization error* of the i-th node. Note that it is not needed that all clocks follow an *absolute time* but rather that they all agree on their estimate of time. If the nodes cannot communicate, then the local time $x_i(t)$ can be modeled as a discrete time integrator

$$x_i(t+1) = x_i(t) + d_i.$$

Clearly, different initial offset, $x_i(0)$, and different speeds d_i cause synchronization errors.

The solution proposed in this paper is inspired by Carli et al. (2008) where the authors propose a distributed clock synchronization strategy based on a Proportional-Integral (PI) controller that treats the different clock speeds as unknown constant disturbances and the different clock offsets as different initial conditions for the system dynamics. They assume that it is possible to control each clock by a local input $u_i(t)$ as follows

$$x_i(t+1) = x_i(t) + d_i + u_i(t).$$

To make the algorithm distributed, the control action $u_i(t)$ is only allowed to use *local* information. Precisely, at time instants $t \in \mathbb{N}$ nodes can exchange their local time according to a communication graph $\mathcal{G}$ having $\{1, \ldots, N\}$ as the set of vertices and in which there is an edge from j to i whenever the node j can send $x_j(t)$ to the node i. Moreover each node has in memory, other than the estimate $x_i(t)$, another variable denoted by $w_i(t)$; this variable plays an important role in compensating the different speeds d_i. If we define the N-dimensional vectors $x(t), u(t)$ and d as the vectors with components $x_i(t), u_i(t)$ and d_i respectively, the PI consensus controller proposed in Carli et al. (2008) has the following structure

$$\begin{aligned} w(t+1) &= w(t) - \alpha K x(t) \qquad w(0) = 0 \\ u(t) &= w(t) - K x(t) \end{aligned}$$

where α is a suitable positive real number in $[0, 1]$ and where $K \in \mathbb{R}^{N \times N}$ is such that (i) $I - K$ is an aperiodic and irreducible stochastic matrix (*consensus matrix*) and (ii) $K_{ij} \neq 0$ only if (i, j) is an edge of the graph $\mathcal{G}$.

Then the overall system becomes

$$\begin{bmatrix} x(t+1) \\ w(t+1) \end{bmatrix} = \begin{bmatrix} I - K & I \\ -\alpha K & I \end{bmatrix} \begin{bmatrix} x(t) \\ w(t) \end{bmatrix} + \begin{bmatrix} d \\ 0 \end{bmatrix}. \tag{1}$$

The analysis in Carli et al. (2008) is restricted to symmetric matrices K and shows that, under certain conditions on the parameter α and on the eigenvalues of K, asymptotic synchronization is achieved.

Note that the previous algorithm requires reliable and synchronized communication along all the edges of a given graph $\mathcal{G}$. Clearly this is an impracticable condition, as nodes do not have access to the *absolute time* t. To overcome these limitations the authors in Carli et al. (submitted) provide a realistic pseudo-synchronous implementation of the previous algorithm where each clock carries out all its operations related to the k-th iteration, including transmitting messages, receiving messages and updating its estimate, within a short time window. The effectiveness of this realistic implementation has been confirmed by simulations. However, certain conditions on the size of the time window and on the maximum distance between any pair $x_i(t), x_j(t)$ for any instant t are required. These conditions might be seriously affected by packet losses, node and link failures.

This paper aims to reduce the communication requirements of (1) by considering a different and more realistic, communication protocol between the clocks. Precisely we assume that the communication model is that of gossiping clocks, i.e. a model in which only a pair of clocks can communicate at any time. In the next section we will provide a formal description of the novel proposed algorithm.

3. A PI CONSENSUS CONTROLLER WITH GOSSIP COMMUNICATION

We start with the following assumption.

Assumption 1. The communication graph $\mathcal{G} = (V, \mathcal{E})$ is a undirected connected graph without any self-loop and, at every time instant t, each edge $(i, j) \in \mathcal{E}$ can be selected with a strictly positive probability $W^{(i,j)}$.

We then define the matrix $W \in \mathbb{R}^{N \times N}$ as the matrix having W^{ij} as element in the i-th row and j-th column and in the j-th row and i-th column, namely

$$W^{ij} = W^{ji} := W^{(i,j)}. \tag{2}$$

Similarly to (1), each node has in memory the two variables x_i and w_i. Assume that, at time t the edge (i, j) is selected. Then the updating equations for x and w are

$$
\begin{aligned}
x_i(t+1) &= \frac{1}{2}\big(x_i(t) + x_j(t)\big) + w_i(t) + d_i(t)\\
x_j(t+1) &= \frac{1}{2}\big(x_i(t) + x_j(t)\big) + w_j(t) + d_j(t)\\
x_h(t+1) &= x_h(t) + w_h(t) + d_h(t), \qquad h \neq i,j
\end{aligned}
$$

and

$$
\begin{aligned}
w_i(t+1) &= \frac{\alpha}{2}\big(-x_i(t) + x_j(t)\big) + w_i(t)\\
w_j(t+1) &= \frac{\alpha}{2}\big(-x_j(t) + x_i(t)\big) + w_j(t)\\
w_h(t+1) &= w_h(t), \qquad h \neq i,j
\end{aligned}
$$

These equations can be rewritten as

$$
\begin{aligned}
&\begin{bmatrix} x(t+1)\\ w(t+1)\end{bmatrix} =\\
&= \begin{bmatrix} I - e_ie_i^* - e_je_j^* + (e_i+e_j)(e_i+e_j)^* & I\\ -\frac{\alpha}{2}(e_i+e_j)(e_i+e_j)^* - \alpha\left(e_ie_i^* + e_je_j^*\right) & I\end{bmatrix}\begin{bmatrix} x(t)\\ w(t)\end{bmatrix} +\\
&\quad + \begin{bmatrix} d(t)\\ 0\end{bmatrix} = \begin{bmatrix} I - \frac{1}{2}E^{(i,j)} & I\\ -\frac{\alpha}{2}E^{(i,j)} & I\end{bmatrix}\begin{bmatrix} x(t)\\ w(t)\end{bmatrix} + \begin{bmatrix} d(t)\\ 0\end{bmatrix}
\end{aligned}
\tag{3}
$$

where $E^{(i,j)} = (e_i - e_j)(e_i - e_j)^*$. Note that $I - E^{(i,j)}$ is a doubly stochastic matrix. Considering the variable $v(t) = w(t) + d(t)$, the above system can be rewritten as

$$
\begin{bmatrix} x(t+1)\\ v(t+1)\end{bmatrix} = \begin{bmatrix} I - \frac{1}{2}E^{(i,j)} & I\\ -\frac{\alpha}{2}E^{(i,j)} & I\end{bmatrix}\begin{bmatrix} x(t)\\ v(t)\end{bmatrix} := F^{(i,j)}\begin{bmatrix} x(t)\\ v(t)\end{bmatrix}
$$

To our aims it is convenient to consider the variables

$$
\bar{x}(t) = \Omega x(t) \text{ and } \bar{v}(t) = \Omega v(t)
$$

where

$$
\Omega = I - \frac{1}{N}\mathbf{1}\mathbf{1}^*
$$

with $\mathbf{1} = [1,\dots,1]^*$. Clearly the synchronization is asymptotically reached if and only if $\bar{x}(t) \to 0$. Straightforward calculations show that

$$
\begin{bmatrix} \bar{x}(t+1)\\ \bar{v}(t+1)\end{bmatrix} = F^{(i,j)}\begin{bmatrix} \bar{x}(t)\\ \bar{v}(t)\end{bmatrix}
\tag{4}
$$

i.e. $[\bar{x}\ \bar{w}]^*$ satisfies the same recursive equation of $[x\ w]^*$.

4. MEAN-SQUARE ANALYSIS

In this section we provide a mean-square analysis of (4). To do so, it is convenient to introduce the matrix

$$
P(t) = \mathbb{E}\left[\begin{bmatrix}\bar{x}(t)\\ \bar{v}(t)\end{bmatrix}[\bar{x}(t)^*\ \bar{v}(t)^*]\right] = \begin{bmatrix} P_{11} & P_{12}\\ P_{21}^* & P_{22}\end{bmatrix}
$$

where

$$
\begin{aligned}
P_{11}(t) &= \mathbb{E}\left[\bar{x}(t)\bar{x}^*(t)\right]\\
P_{12}(t) &= \mathbb{E}\left[\bar{x}(t)\bar{v}^*(t)\right]\\
P_{22}(t) &= \mathbb{E}\left[\bar{v}(t)\bar{v}^*(t)\right]
\end{aligned}
$$

We have that [1]

$$
\begin{bmatrix} P_{11} & P_{12}\\ P_{12}^* & P_{22}\end{bmatrix}^+ = \mathbb{E}\left\{F^{(i,j)}\begin{bmatrix} P_{11} & P_{12}\\ P_{12}^* & P_{22}\end{bmatrix}F^{(i,j)*}\right\}
$$

from which we obtain the following recursive equations

$$
\begin{aligned}
P_{11}^+ &= P_{11} - \frac{1}{2}\mathbb{E}\left[E^{(i,j)}P_{11}\right] + \frac{1}{4}\mathbb{E}\left[E^{(i,j)}P_{11}E^{(i,j)}\right]\\
&\quad - \frac{1}{2}\mathbb{E}\left[P_{11}E^{(i,j)}\right] + P_{12}^* - \frac{1}{2}\mathbb{E}\left[P_{12}^*E^{(i,j)}\right] + P_{12}\\
&\quad - \frac{1}{2}\mathbb{E}\left[E^{(i,j)}P_{12}\right] + P_{22}\\
P_{12}^+ &= \frac{\alpha}{2}\mathbb{E}\left[P_{11}E^{(i,j)}\right] - \frac{\alpha}{4}\mathbb{E}\left[E^{(i,j)}P_{11}E^{(i,j)}\right]\\
&\quad + \frac{\alpha}{2}\mathbb{E}\left[P_{12}^*E^{(i,j)}\right] + P_{12} - \frac{1}{2}\mathbb{E}\left[E^{(i,j)}P_{12}\right] + P_{22}\\
P_{22}^+ &= \frac{\alpha^2}{4}\mathbb{E}\left[E^{(i,j)}P_{11}E^{(i,j)}\right] + \frac{\alpha}{2}\mathbb{E}\left[P_{12}^*E^{(i,j)}\right]\\
&\quad + \frac{\alpha}{2}\mathbb{E}\left[E^{(i,j)}P_{12}\right] + P_{22}
\end{aligned}
$$

The covariance matrix P then updates according to a linear transformation

$$
P(t+1) = F\big[P(t)\big]
$$

defined by the recursive equations that we just computed, and whose initial conditions can be obtained once we state the following assumption on $x(0)$ and $d(0)$.

Assumption 2. The initial conditions $x(0)$ and $d(0)$ are random vectors such that $\mathbb{E}\left[x(0)\right] = 0$, $\mathbb{E}\left[x(0)x^*(0)\right] = \sigma_x^2 I$ and $\mathbb{E}\left[d(0)\right] = \mathbf{1}$, $\mathbb{E}\left[d(0)d^*(0)\right] = (\sigma_d^2+1)I$ for some $\sigma_x^2 > 0$ and $\sigma_d^2 > 0$. Moreover $w(0) = 0$.

It then follows that

$$
P = \begin{bmatrix} P_{11}(0) & P_{12}(0)\\ P_{12}^*(0) & P_{22}(0)\end{bmatrix} = \begin{bmatrix} \sigma_x^2\Omega & 0\\ 0 & (\sigma_d^2+1)\Omega.\end{bmatrix}
\tag{5}
$$

The analysis of the previous recursive equations is a challenging problem when $\mathcal{G}$ is an arbitrary graph. In the next section we provide a theoretical analysis for the complete graph. In Section 6 we consider also a more realistic family of graphs through numerical simulations.

4.1 Complete graph

Assume that the graph $\mathcal{G}$ describing the feasible communications between nodes is the complete graph. Moreover, assume that each edge has the same probability $\frac{2}{N(N-1)}$ of being selected. Hence

$$
W = \frac{2}{N(N-1)}(\mathbf{1}\mathbf{1}^* - I).
\tag{6}
$$

We have the following technical Lemma.

Lemma 1. Let W be defined as in (6). Then

$$
\begin{aligned}
\mathbb{E}\left[E^{(i,j)}\Omega\right] &= \mathbb{E}\left[\Omega E^{(i,j)}\right] = \frac{2}{N-1}\Omega\\
\mathbb{E}\left[E^{(i,j)}\Omega E^{(i,j)}\right] &= \frac{4}{N-1}\Omega
\end{aligned}
$$

Proof. The given expressions follow just by direct computation, that we omit here.

We can then state the two following results, that characterize the evolution of the matrices P_{11}, P_{12} and P_{22} when $\mathcal{G}$ is the complete graph and the matrix W has the expression given in (6).

Proposition 2. In the case of a complete graph and of equiprobability of edge selection, the set

$$
\mathcal{J} = \left\{P \middle| P = \begin{bmatrix} a & b\\ b & c\end{bmatrix} \otimes \Omega\right\}
$$

---

[1] Time dependance has been omitted in these recursive equations. The superscript plus sign indicates the value of the variables at time $t+1$.

is invariant under the transformation (1).

Proof. Let $P(t) = P$, with $P \in \mathcal{J}$. Then

$$\begin{aligned} P_{11}(t+1) &= \frac{N-2}{N-1} a\Omega + 2\frac{N-2}{N-1} b\Omega + c\Omega \\ P_{12}(t+1) &= \left(1 - \frac{\alpha+1}{N-1}\right) b\Omega + c\Omega \\ P_{22}(t+1) &= \frac{\alpha^2}{N-1} a\Omega - \frac{2\alpha}{N-1} b\Omega + c\Omega \end{aligned}$$

and therefore $P(t+1) \in \mathcal{J}$.

Proposition 3. In the case of a complete graph and equiprobability of edge selection, with assumption 2 holding, we have

$$P_{11}(t) = p_{11}(t), \Omega \quad P_{12}(t) = p_{12}(t), \Omega \quad P_{22}(t) = p_{22}(t), \Omega$$

where

$$\begin{bmatrix} p_{11}(t+1) \\ p_{12}(t+1) \\ p_{22}(t+1) \end{bmatrix} = \begin{bmatrix} \frac{N-2}{N-1} & 2\frac{N-2}{N-1} & 1 \\ 0 & 1 - \frac{\alpha+1}{N-1} & 1 \\ \frac{\alpha^2}{N-1} & -\frac{2\alpha}{N-1} & 1 \end{bmatrix} \begin{bmatrix} p_{11}(t) \\ p_{12}(t) \\ p_{22}(t) \end{bmatrix}$$

with initial conditions

$$\begin{bmatrix} p_{11}(0) \\ p_{12}(0) \\ p_{22}(0) \end{bmatrix} = \begin{bmatrix} \sigma_x^2 \\ 0 \\ 1 + \sigma_d^2 \end{bmatrix}$$

Proof. The result just follows from the fact that the initial condition (5) is in $\mathcal{J}$, and therefore we are allowed to parametrize the trajectories of the system in the form

$$P(t) = \begin{bmatrix} p_{11}(t) & p_{12}(t) \\ p_{12}(t) & p_{22}(t) \end{bmatrix} \otimes \Omega$$

because of the invariance stated in proposition 2. The same proposition gives also the update equations for the three parameters of the covariance matrix.

5. ALGORITHM CONVERGENCE

Theorem 4. Consider the network of clocks described in section 2, with a gossip communication protocol over a complete graph and a edge selection probability matrix W as in (6). Let the system be initialized according to assumption 2. Then the variance P of the synchronization error converges exponentially to zero if and only if

$$\alpha < \bar{\alpha} = \frac{3}{2} - N + \frac{1}{2}\sqrt{4N^2 - 12N + 17} \qquad (7)$$

Proof. The covariance matrix of the synchronization error evolves according to the linear update law and the initial conditions stated in proposition 3.

The stability of the update equation can then be studied by eigenvalue analysis. The characteristic polynomial of the update matrix in proposition 3 is

$$\begin{aligned} \Psi(z) = &- \frac{-N^2 + 2N - 1}{(N-1)^2} z^3 \\ &- \frac{-8N + \alpha - \alpha N + 3N^2 + 5}{(N-1)^2} z^2 \\ &- \frac{-8 - \alpha + 10N - 3N^2 - \alpha^2 + \alpha^2 N}{(N-1)^2} z \\ &- \frac{4 - 2\alpha - 4N + N^2 + \alpha N - 2\alpha^2 + \alpha^3 + \alpha^2 N}{(N-1)^2} \end{aligned} \qquad (8)$$

An efficient way to study the stability of $\Psi(z)$ is by applying Routh criterion to the continuous time version of $\Psi'(z)$, where the discrete time to continuous time conversion is performed by Tustin transformation, which preserves stability and maps the unit circle into the left hand semiplane.

Substituting $z = (1+s)/(1-s)$ into $\Psi(z)$ we get

$$\begin{aligned} \Psi_c(s) = &\frac{1}{(N-1)^2 (1-s)^3} \cdot \\ &\Big[\left[18 + \alpha^3 - \alpha^2 - 24N + 8N^2\right] s^3 \\ &+ \left[-3\alpha^3 + (5-2N)\alpha^2 + (6-4N)\alpha - 12 + 8N\right] s^2 \\ &+ \left[2 + 3\alpha^3 + (4N-7)\alpha^2 + (4N-8)\alpha\right] s \\ &- \alpha^3 + (3-2N)\alpha^2 + 2\alpha \Big] \end{aligned} \qquad (9)$$

We therefore apply Routh criterion to the numerator of $\Psi_c(s)$, which is a third order polynomial. The algorithm characterize the stability of this polynomial, and therefore of the original matrix A (a part from the three poles in 1), by studying the sign of four terms, namely

$$\begin{aligned} r_3 &= 18 + \alpha^3 - \alpha^2 - 24N + 8N^2 \\ r_2 &= -3\alpha^3 + (5-2N)\alpha^2 + (6-4N)\alpha - 12 + 8N \\ r_1 &= \Big[\alpha^6 + (2N-4)\alpha^5 \\ &\quad + \left(N^2 - N - 1\right)\alpha^4 + \left(2N^2 + 13 - 11N\right)\alpha^3 \\ &\quad + \left(1 - 7N - 2N^3 + 7N^2\right)\alpha^2 \\ &\quad + \left(-9 - 2N^2 + 9N\right)\alpha + 3 - 2N\Big] \\ &\quad \cdot \frac{8}{3\alpha^3 + (2N-5)\alpha^2 + (4N-6)\alpha + 12 - 8N} \\ r_0 &= -\alpha^3 + (3-2N)\alpha^2 + 2\alpha \end{aligned}$$

Routh's criterion states that to have stability of the polynomial, all the four terms r_i must have the same sign. We will now show that, under the technical assumption $N > 2$ (which is a necessary condition to have a properly randomized algorithm) the terms r_3, r_2, and r_1 are always positive, so stability only depends on the sign of r_0.

Sign of the term r_3

Let's first suppose that $\alpha = 0$. In this case term r_3 reduces to the second-order polynomial in N

$$r_3|_{\alpha=0} = 18 - 24N + 8N^2$$

which can easily proved to be greater than zero for all $N \in \mathbb{N}$, as it describes a convex parabola with its vertex in $N = 3/2$, where its value is zero. When $\alpha > 0$ is considered, it is still easy to see that, even if the parabola now crosses the N-axis, r_3 is still greater than zero for any $N > 2$ (more precisely, for $N > 3/2 + \alpha/4 \cdot \sqrt{2 - 2\alpha}$, which is less than 2. Therefore $r_3 > 0$ for all N's greater than 2.

Sign of the term r_2

All the positive powers of α in r_2 have negative coefficient, if $N > 2$. Therefore, it results

$$\begin{aligned} r_2 &= -3\alpha^3 + (5-2N)\alpha^2 + (6-4N)\alpha - 12 + 8N \\ &\geq (8 - 6N)\alpha - 12 + 8N. \end{aligned}$$

Therefore $r_2 > 0$ if

$$\alpha < \frac{12 - 8N}{8 - 6N}.$$

This is always true for $\alpha \in [0,1]$ and $N > 2$. Indeed

$$\left.\frac{12-8N}{8-6N}\right|_{N=3} = \frac{6}{5}$$

and it is increasing in N, as

$$\left.\frac{12-8N}{8-6N}\right|_{N+1} - \left.\frac{12-8N}{8-6N}\right|_{N} = \frac{2}{(3N-1)(3N-4)} > 0\,.$$

Therefore any $r_2 > 0$ for all the considered α's and N's.

Sign of the term r_1

Let's first study the denominator of r_1. As $\alpha \le 1$, we have

$$\mathrm{DEN}r_1 \le -2N + 4 < 0$$

once $N > 2$. As the denominator is then negative, we have to check negativity of the numerator to have $r_1 > 0$.

The coefficients of the four higher powers of α in the denominator are all positive, so as $\alpha < 1$ we can bound α^6, α^5 and α^4 with α^2, and α^3 with α, so that we have for the numerator NUM_{r_1}

$$\mathrm{NUM}_{r_1} < (-2N^3+8N^2-6N-3)\alpha^2+(4-2N)\alpha+3-2N\,.$$

This expression is negative, as all the coefficients of the polynomial are negative. This is trivially true for the coefficients of α and for the costant term. The coefficient of α^2 requires a little more analysis. We can easily check that it is negative for the smallest N, $N = 3$. Its discrete increment is

$$\begin{aligned}&\left(-2n^3 + 8n^2 - 6n - 3\right)_{n=N+1}\\ &\qquad - \left(-2n^3 + 8n^2 - 6n - 3\right)_{n=N} = -6N^2 + 10N\end{aligned}$$

therefore it is decreasing for $N \ge 2$. As both the numerator and the denominator of r_1 are negative, r_1 is positive.

Sign of the term r_0

Term r_0 is a third order polynomial but its sign can be easily studied as $\alpha = 0$ is one of its roots. As $\alpha > 0$ we then must have

$$-\alpha^2 + (3-2N)\alpha + 2 > 0\,. \tag{10}$$

The roots of (10) are

$$\alpha_{1,2} = \frac{3}{2} - N \pm \frac{1}{2}\sqrt{4N^2 - 12N + 17}$$

and therefore the stability condition results to be

$$0 < \alpha < \frac{3}{2} - N + \frac{1}{2}\sqrt{4N^2 - 12N + 17}. \tag{11}$$

Corollary 5. Under the same hypoteses of theorem 4, a sufficient condition on α for the variance P of the synchronization error to go to zero is that

$$0 < \alpha \le \frac{1}{N-1} \tag{12}$$

Proof. The proof follows exactly what have been done for theorem 4. When studying the positivity of the term r_0, though, one can use the fact that $\alpha^2 < \alpha$, obtaining

$$-\alpha^2 + (3-2N)\alpha + 2 > 2(1-N)\alpha + 2$$

which is greater than zero for

$$\alpha \le \frac{1}{N-1}\,.$$

It is easy to see that condition on α expressed in theorem 4 and in corollary 5 are very close (still being the bound in the corollary far more readable and meaningful). Figure 1 put the two conditions in comparison.

The convergence result that we derived by analyzing the covariance matrix of the synchronization error is even more meaningful once it allows to prove the convergence of the system with probability 1, as the following corollary states.

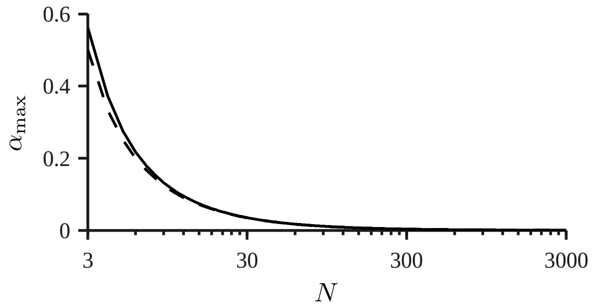

Fig. 1. Comparison of the conservative bound (12) on α (dashed) and the exact bound (7) (continuous).

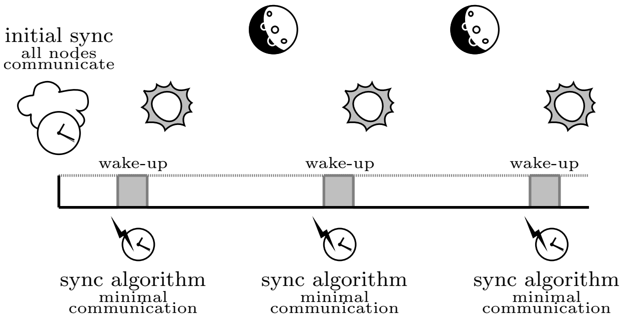

Fig. 2. The testbed that inspires the choice of the numerical parameters for the presented simulations. Agents wake up daily for a small time window to perform their tasks (sensing, computing,...). In this window they are also allowed to perform few communications to keep their clocks synchronized.

Corollary 6. Under the same hypoteses of theorem 4, if $\alpha < \bar{\alpha}$, then there exist with probability 1 a positive real number η such that, eventually,

$$\left\|\begin{matrix}\bar{x}(t)\\ \bar{v}(t)\end{matrix}\right\|_2^2 \le e^{-\eta t}.$$

Proof. From theorem 4 we can state that any norm of $P(t)$ converges exponentially to zero if $\alpha < \bar{\alpha}$. Therefore for the trace norm we have

$$\mathrm{Tr}(P) = \mathrm{Tr}\left(\mathbb{E}\left[\begin{bmatrix}\bar{x}\\ \bar{v}\end{bmatrix}\left[\bar{x}(t)^* \ \ \bar{v}(t)^*\right]\right]\right) = \mathbb{E}\left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 \le Ce^{-2\eta t}$$

for some positive C and η. We also have that

$$\int \left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 \mathrm{d}\mu \ge \epsilon(t) \int \chi_{\left\|\begin{smallmatrix}\bar{x}\\ \bar{v}\end{smallmatrix}\right\|_2^2 > \epsilon(t)} \mathrm{d}\mu = \epsilon(t)\mathbb{P}\left[\left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 \ge \epsilon(t)\right]$$

where $\mathrm{d}\mu$ is the probability measure and $\epsilon(t)$ is a positive function of t (dependance on t has been omitted in the vector norms). Let's take $\epsilon(t) = e^{-\eta t}$, obtaining

$$\mathbb{P}\left[\left\|\begin{matrix}\bar{x}(t)\\ \bar{v}(t)\end{matrix}\right\|_2^2 \ge e^{-\eta t}\right] \le Ce^{-\eta t}.$$

Therefore, by Borell-Cantelli's lemma, with probability 1 it will eventually hold $\left\|\begin{matrix}\bar{x}\\ \bar{v}\end{matrix}\right\|_2^2 \le e^{-\eta t}$.

6. SIMULATION RESULTS

Consider a network of N agents, in which any agent can communicate with any other agent in a bi-directional fashion. Suppose that at regular time intervals the agents

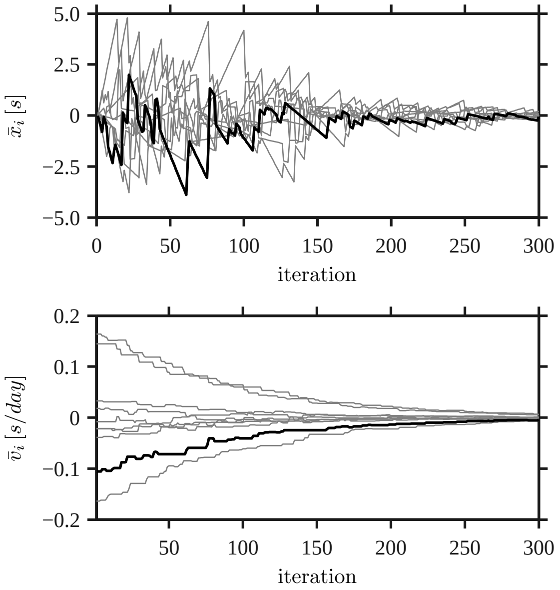

Fig. 3. Single run of the algorithm with $N = 10$ and $\alpha = 0.01$. The clock offset and the skew of one specific node has been highlighted.

wake up and the communication between a randomly choosen node and one of its neighbors takes place. Based on the received data, the two nodes run the algorithm described in section 2 (see figure 2).

This way, for example, a network of battery-powered sensors can keep their clocks synchronized and reduce significantly the fraction of the time they have to be active for communication. The more precise the synchronization is, the thinner the active time window can be and the longer their batteries will last. Moreover, it is critical that the communication is kept to a minimum, to reduce energy consumption and to save communication resouces.

The numerical values for the parameters of the model come from a hypotetical case study of commercially available wireless sensors waking up once a day, initialized at deployment time with a synchronization error in the order of 1 second, and with relative clock speed drifts in the order few seconds a month (characteristic of quartz oscillators).

Table 1. Simulation clock parameters

time interval between updates	T	1 day = 86400 s
clock offset standard deviation	σ_x	$10^{-5}T$
clock skew standard deviation	σ_v	10^{-6}

Figure 3 shows how the algorithm works for a small network of $N = 10$ clocks of this type.

Let us verify that the bound on α that we obtained for stability is meaningful. In figure 4 the mean square error of the clocks has been plotted for different values of α. One can see as the $\alpha_{\max}$ obtained in theorem 4 is critical in determining the convergence of the error. Just by choosing a value of α smaller or larger than $\alpha_{\max}$ by 10%, the behavior of the system is qualitatively different and corresponds to what the analysis says.

In this plot and in all the other plots of the mean square error, the curve represents an average curve over 100 realizations of the algorithm.

In figure 5 the mean square error is plotted for different values of α, in a network of size $N = 50$. Note that there is a first transient in which the drift errors make the synchronization error to increase. After this transient the error start decreasing at a rate that depends on α (and an optimal α could then be searched). Due to the absence of quantization, noise, and time-variance of the drifts, the error then decreases to zero.

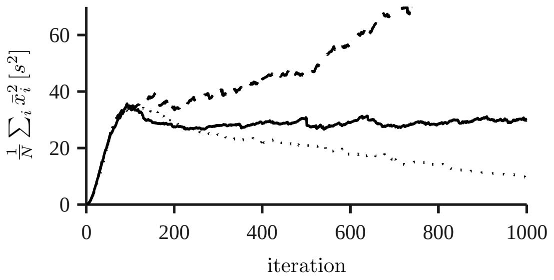

Fig. 4. Average of the squared clock errors for $\alpha = 0.9\alpha_{\max}$ (dotted), $\alpha = 1.1\alpha_{\max}$ (dashed), and $\alpha = \alpha_{\max}$ (continuous). $N = 50$.

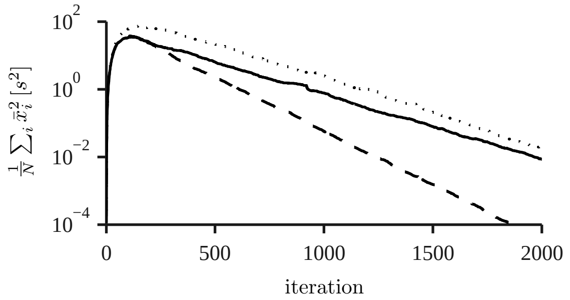

Fig. 5. Average of the squared clock errors for $\alpha = 0.1\alpha_{\max}$ (dotted), $\alpha = 0.5\alpha_{\max}$ (dashed), and $\alpha = 0.7\alpha_{\max}$ (continuous).

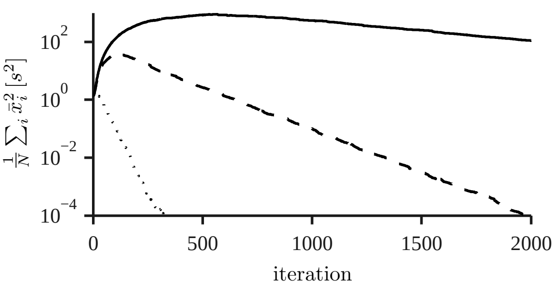

Fig. 6. Average of the squared clock errors for networks of size $N = 10$ (dotted), $N = 50$ (dashed), and $N = 250$ (continuous). In all cases $\alpha = 0.5\alpha_{\max}$.

In figure 6 the behavior of networks of different sized is illustrated. $\alpha = 0.5\alpha_{\max}$ has been choosen for all the cases. It's easy to see that the bound we found on α makes the algorithm not scalable. That is, as the number of agents increases, the rate of convergence becomes slower (more than linearly in the network size). Moreover, the transient becomes unacceptable if our algorithm relies on synchronization to be implementable. Indeed, if the clock error becomes larger than the wake-up time window, communication becomes impossible for that node. This scalability issue is addressed in the next section.

6.1 Geometric graphs

In many real networks the communication graph cannot be considered complete. For example when agents are spa-

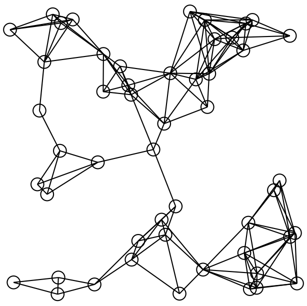

Fig. 7. Geometric graph with $N = 50$ agents. The average degree of the nodes is 5.7.

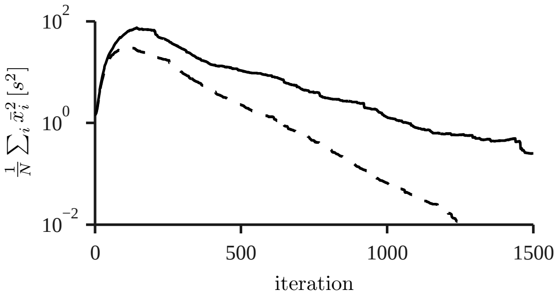

Fig. 8. Average of the squared clock errors for the communication graph of figure 7 (continuous) and for the complete graph (dashed). $N = 50$ and $\alpha = 0.01$.

tially deployed and signal strength decreases with distance, it is reasonable to assume that the communication graph is a *geometric graph*, that is a bidirectional edge exists between two nodes when their distance is smaller than a certain threshold. One example of a geometric graph is represented in figure 7, while in figure 8 it is plotted the mean square error obtained by running the synchronization algorithm on that graph.

7. SCALABILITY ISSUES

The result stated in theorem 4 seems to say that, to preserve stability of the whole systems, the parameter α has to decrease as the inverse of the number of nodes. Simulations show that the rate of convergence of the clocks to a consensus value depends also on the value of the parameter α, and therefore having smaller α's correspond to slower convergence of the system and unacceptable transients that can make the algorithm unimplementable. In this sense the system is therefore not scalable.

7.1 Multiple symmetric gossip

The approach we propose to tackle this problem consists in implementing *multiple gossip communications*, i.e. allowing for more than one edges to be activated at each time step. Indeed, the fact that the clock synchronization algorithm is not scalable mainly depends on the fact that at each iteration only two nodes update their state, while the others keep integrating their local skew error. There is no way to overcome this problem by choosing an appropriate α, on the contrary theorem 4 states that α has to go to zero when N grows, to preserve stability.

Consider then the following modification of the algorithm. Suppose that at every iteration each node decides whether to communicate or not with probability p. If it opts for communication, then one of its neighbors is randomly

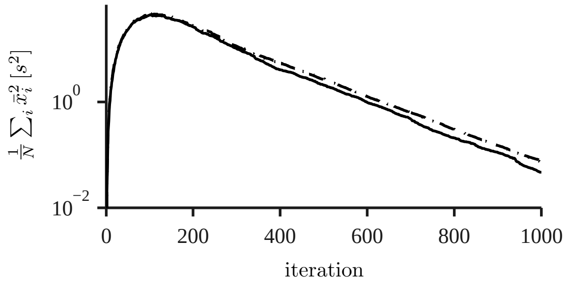

Fig. 9. Average of the squared clock errors for networks of size $N = 50$ (continuous), $N = 500$ (dashed), and $N = 5000$ (dotted). $\alpha = 0.01$ and 2% of the total edges are activated at the same time.

choosen and a symmetric gossip communication takes place. This way, at every iteration, the expected number of communications that take place is $p * N$. Note moreover that this solution is easier to implement of a large network than the original solution of activating one only edge in the whole system. The resulting process is a Bernoulli process, and the time between successive communication of a certain node is now described by a geometric random variable independent from N.

Figure 9 shows the behavior of the synchronization error for networks of different sizes, with $p = 0.02$ (that is, having in expectation 2% of the nodes starting a communication at each iteration). $\alpha = 0.5\bar{\alpha}_{\max}$ has been choosen, where $\bar{\alpha}_{\max}$ is the bound for a stabilizing α for a network of size $1/p$, that is a network that would have one only edge communicating (in expectation) with the given p. The plot clearly shows how the behavior of the synchronization error is independent from the number of agents.

8. CONCLUSIONS

We developed a version of the PI synchronization algorithm that relies on symmetric gossip communications to achieve synchronization of a network of clocks and compensation of different clock speeds. In the case of a complete graph, the stability of the algorithm is studied with respect to the parameter α, showing that the algorithm badly scales with the number of nodes. This scalability issue is tackled by allowing for more than one node to estabilish a connection with a neighbor at the same time. This not only makes the algorithm performance independent from the size of the network, but is easier to implement in a completely distributed fashion.

Future direction of investigation include studying different communication graphs and protocols (for example broadcast communication), and modeling some of the most common non-idealities like packet drops, quantization of the states, delays and time-varying speed of the clocks.

REFERENCES

Amin, M. and Schewe, P. (2007). Preventing blackouts. *Scientific American*, 60–67.

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (2008). A PI consensus controller for networked clocks synchronization. In *IFAC Congress*. Seoul, Corea.

Carli, R., Chiuso, A., Schenato, L., and Zampieri, S. (submitted). A PI consensus controller for networked clocks synchronization. *Automatica*.

Ganeriwal, S., Kumar, R., and Srivastava, M. (2003). Timingsync protocol for sensor networks. In *Proceed-*

ings of the first international conference on Embedded networked sensor systems (SenSys'03).

Hohlt, B., Doherty, L., and Brewer, E. (2004). Flexible power scheduling for sensor networks. In *IEEE and ACM International Symposium on Information Processing in Sensor Networks (IPSN'04).*

Maròti, M., Kusy, B., Simon, G., and Àkos Ldeczi (2004). The flooding time synchronization protocol. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys'04)*, 39–49.

Oh, S., Schenato, L., and Sastry, S. (2005). A hierarchical multiple-target tracking algorithm for sensor networks. In *IEEE International Conference on Robotics and Automation*, 2197 – 2202. Barcelona, Spain.

Schenato, L. and Gamba, G. (2007). A distributed consensus protocol for clock synchronization in wireless sensor network. In *Proceedings of IEEE Conference on Decision and Control*, 2289–2294. New Orleans, U.S.A. doi:10.1109/CDC.2007.4434671.

Simeone, O. and Spagnolini, U. (2007). Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. *EURASIP Journal on Wireless Communications and Networking*, 2007, Article ID 57054, 13 pages. Doi:10.1155/2007/57054.

Solis, R., Borkar, V., and Kumar, P.R. (2006). A new distributed time synchronization protocol for multihop wireless networks. In *To appear in 45th IEEE Conference on Decision and Control (CDC'06)*. San Diego.

Szewczyk, R., Osterweil, E., Polastre, J., Hamilton, M., Mainwaring, A.M., and Estrin, D. (2004). Habitat monitoring with sensor networks. *Communication of the ACM*, 46, 34–40.

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

Yoon, S., Veerarittiphan, C., and Sichitiu, M.L. (2007). Tiny-sync: Tight time synchronization for wireless sensor networks. *ACM Journal of Sensor Networks*, 3(2).