

Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2007, Article ID 57054, 13 pages
doi:10.1155/2007/57054

Research Article

Distributed Time Synchronization in Wireless Sensor Networks with Coupled Discrete-Time Oscillators

O. Simeone¹ and U. Spagnolini²

¹ Center for Wireless Communications and Signal Processing Research, New Jersey Institute of Technology, University Heights, Newark, NJ 07102-1982, USA

² Dipartimento di Elettronica e Informazione, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Received 25 September 2006; Accepted 30 March 2007

Recommended by Mischa Dohler

In wireless sensor networks, distributed timing synchronization based on pulse-coupled oscillators at the physical layer is currently being investigated as an interesting alternative to packet synchronization. In this paper, the convergence properties of such a system are studied through algebraic graph theory, by modeling the nodes as discrete-time clocks. A general scenario where clocks may have different free-oscillation frequencies is considered, and both time-invariant and time-variant network topologies (or fading channels) are discussed. Furthermore, it is shown that the system of oscillators can be studied as a set of coupled discrete-time PLLs. Based on this observation, a generalized system design is discussed, and it is proved that known results in the context of conventional PLLs for carrier acquisition have a counterpart in distributed systems. Finally, practical details of the implementation of the distributed synchronization algorithm over a bandlimited noisy channel are covered.

Copyright © 2007 O. Simeone and U. Spagnolini. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Distributed timing synchronization refers to a decentralized procedure that ensures the achievement and maintenance of a common time-scale (frequency and phase) for all the nodes of the network [1]. This condition enables a wide range of applications and functionalities of a sensor networks, including complex sensing tasks (distributed detection/estimation, data fusion), power saving (all nodes sleep and wake-up at coordinate times), and medium access control for communication (e.g., time division multiple access and cooperative communications).

Conventional design of distributed algorithms for timing synchronization prescribes the exchange of local time information through packets carrying a time-stamp to be appropriately elaborated by the transmitting and receiving nodes [1]. Packet-based synchronization has been widely studied, especially in the context of wireline networks. However, the specific features and requirements of wireless sensor networks call for alternative methods that improve both the computational complexity (and therefore energy efficiency) and scalability. Toward this goal, physical layer-based synchronization protocols are currently being investigated that

exploit the broadcast nature of radio propagation. The idea is to build distributed algorithms based on the exchange of pulses at the physical layer, thus avoiding the need to perform complex processing at the packet level.

Physical layer-based synchronization was studied in [2] using a mathematical framework developed in [3] in order to model the spontaneous establishment of synchronous periodic activities in biological systems, such as the flashing of fireflies. In [2, 3], nodes are modeled as integrate-and-fire oscillators coupled through the transmission of pulses. Convergence is proved under the assumption of an all-to-all interconnection among the nodes. The model was later extended in [4], by explicitly including constraints on the transmission range of each node. In particular, the authors derived a bound on the velocity of convergence by using algebraic graph theory [5]. An implementation of distributed synchronization on a real sensor network testbed was reported in [6]. A related work is [7], where a generalization of the model in [3] is proposed and the regime of an asymptotically dense network is investigated. As a final remark, it should be noted that the framework of physical layer-based timing synchronization has been recently interpreted as a means to achieve distributed estimation/detection [8, 9] or data fusion [10].

In this paper, we reconsider physical layer-based synchronization by modeling the sensors as coupled discrete-time oscillators. Basically, each node modifies its current clock based on a weighted average of the residual differences of timing phases as measured with respect to other nodes. The synchronization algorithms proposed in [11] in the context of interbase station communication and [12] for intervehicle transmission can be seen as instances of this general model. The analytical framework is at the same time a generalization and an application of the literature on discrete-time consensus problems for networks of agents (see, e.g., [13]). In particular, differently from [13], here we address the case of clocks with generally different free-oscillation frequencies, and account for the specific features of a wireless network, namely channel reciprocity and randomness (fading). Analysis of convergence of the synchronization process is carried out by algebraic graph theory as in [4], allowing to relate global convergence properties to the local connectivity of the network. The results are first derived for a time-invariant scenario, and then extended to the case where the network topology (or fading) varies with time, building on the results presented in [14].

A central contribution of this paper is the observation that the distributed synchronization system at hand can be modeled as a set of coupled discrete-time phase locked loops (PLLs). The system can thus be seen as a discrete-time version of the network synchronization scheme of [15], that is based on continuously-coupled analog PLLs. This fact allows us to generalize the system design by introducing the concept of loop order. Moreover, we prove that known results about the convergence of conventional PLLs for carrier acquisition have a counterpart in distributed systems. In particular, it is shown that, under appropriate conditions on the interconnections between sensors, (i) a system of first-order distributed PLLs is able to recover perfectly a phase mismatch among the clocks; (ii) in case of a frequency error, first-order loops are able to recover the frequency gap, but at the expense of an asymptotic phase mismatch; (iii) this asymptotic phase mismatch can be reduced by considering second-order loops.

Finally, the analysis is complemented by addressing the issue of a practical implementation of the distributed synchronization algorithm over a bandlimited Gaussian channels.

2. SYSTEM MODEL AND MAIN ASSUMPTIONS

Let the wireless network be composed of K sensors, where each node, say the k th, has a discrete-time clock with period T_k . If the nodes are left isolated, the timing clock of the k th sensor evolves as $t_k(n) = nT_k + \tau_k(0)$, where $0 \leq \tau_k(0) < T_k$ is an initial arbitrary phase and $n = 1, 2, \dots$ runs over the periods of the timing signal. Two synchronization conditions are of interest. We say the K clocks are *frequency* synchronized if

$$t_k(n+1) - t_k(n) = T \quad (1)$$

for each k and for sufficiently large n , where $1/T$ is the common frequency. A more strict condition requires full *frequency and phase* synchronization¹:

$$t_1(n) = t_2(n) = \dots = t_K(n) \quad \text{for } n \rightarrow \infty. \quad (2)$$

We remark that the network is said to fractionate into, say, two clusters of synchronization if there exist a permutation function on the nodes' labels, $\pi(i) : [1, \dots, n] \rightarrow [1, \dots, n]$ such that for n large enough

$$\begin{aligned} t_{\pi(1)}(n) &= \dots = t_{\pi(r)}(n), \\ t_{\pi(r+1)}(n) &= \dots = t_{\pi(K)}(n), \end{aligned} \quad (3)$$

where the number of nodes in the two clusters is r and $K - r$, respectively. The definition above generalizes naturally to more than two clusters.

Towards the goal of achieving synchronization, the clocks of different sensors can be coupled by letting any node radiate a timing signal as the one sketched in Figure 1. A pulse² is transmitted at times $t_k(n)$ by the k th node and received through independent flat fading channels by the other sensors. It is assumed that all the nodes transmit with the same power, and that the power P_{ki} received on the wireless link between the i th and the k th user reads

$$P_{ki}(n) = \frac{C}{d_{ki}^\gamma(n)} \cdot G_{ki}(n), \quad (4)$$

where C is an appropriate constant that depends on the transmitted power (assumed here to be the same for all nodes), $d_{ki}(n) = d_{ik}(n)$ is the distance between node i and node k at the n th period, $G_{ki}(n)$ is a random variable accounting for the fading process, and γ is the path loss exponent ($\gamma = 2 \div 4$). Notice that the fading channel is reciprocal (all transmissions use the same carrier frequency), which implies that $G_{ik}(n) = G_{ki}(n)$ and $P_{ik}(n) = P_{ki}(n)$ for $i \neq k$ [16]. As detailed in the following, each node (at any period n) processes the received signal in order to estimate the time difference between its own clock $t_k(n)$ and the corresponding "firing" instant of other nodes, that is, $t_i(n) - t_k(n)$, $i \neq k$, and, based on this measure, it updates its own clock.

2.1. The synchronization algorithm

In this section, we consider the synchronization procedure under the ideal assumptions that any node, say the k th, is able to measure exactly the time differences $t_i(n) - t_k(n)$ and the powers $P_{ki}(n)$ of other nodes ($i \neq k$) based on the received signal. This model is elaborated upon in the first part of the

¹ In [6], a distinction is made between synchronization (the state where nodes of the network have a common notion of time) and synchronicity (nodes agree on "firing" period and phase). In this paper, as in most part of the literature, we focus on the latter, and refer to it as either synchronization or synchronicity.

² The temporal width of the transmitted pulse (or equivalently the employed bandwidth) has to be selected so as to guarantee the desired resolution of timing synchronization (see Section 7).

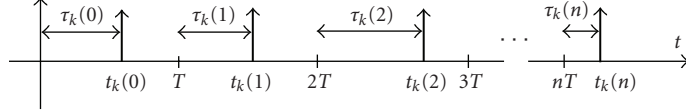


FIGURE 1: Clock $t_k(n)$ of the k th node. $\tau_k(n)$ is the timing phase in the n th period of the clock.

paper. A practical implementation of the system that alleviates the said assumptions (and in particular, does not require estimation of time of arrivals) is then discussed in Section 7. At the n th period, the k th node updates its clock $t_k(n)$ according to a weighted sum of timing differences $\Delta t_k(n+1)$ ³:

$$t_k(n+1) = t_k(n) + \varepsilon \cdot \Delta t_k(n+1) + T_k, \quad (5a)$$

$$\Delta t_k(n+1) = \sum_{i=1, i \neq k}^K \alpha_{ki}(n)(t_i(n) - t_k(n)), \quad (5b)$$

where ε is the step-size ($0 < \varepsilon < 1$) and the coefficients $\alpha_{ki}(n)$ are selected so that $\alpha_{ki}(n) \geq 0$ and $\sum_{i=1, i \neq k}^K \alpha_{ki}(n) = 1$. The updating rule (5) generalizes the algorithms of [11, 12] (and the consensus algorithms, see, e.g., [13]) to a frequency-asynchronous scenario. In this paper, we focus on the following choice for the coefficients $\alpha_{ki}(n)$:

$$\alpha_{ki}(n) = \frac{P_{ki}(n)}{\sum_{j=1, j \neq k}^K P_{kj}(n)}. \quad (6)$$

The selection of the weighting coefficients (6) is inspired by the algorithms proposed in [11, 12]. The rationale of this design is that time differences measured over more unreliable (i.e., low-power) channels should be weighted less when updating the clock, thus rendering the algorithm robust against measurement errors over the fading channels (see also Section 7). Notice that by using (5b) we are implicitly neglecting the propagation delays among nodes, that are assumed to be smaller than the timing resolution. A method to handle propagation delays is described in [11]. As a final remark, we notice that the dynamic system (5) updates the clock $t_k(n+1)$ as a convex combination of the times $\{t_i(n)\}_{i=1}^K$ [14].

By defining the vector containing the clocks of all nodes as $\mathbf{t}(n) = [t_1(n) \cdots t_K(n)]^T$ and the vector of clock periods $\mathbf{T} = [T_1 \cdots T_K]^T$, we can express (5) as the difference vector equation

$$\mathbf{t}(n+1) = \mathbf{A}(n) \cdot \mathbf{t}(n) + \mathbf{T}, \quad (7)$$

where $\mathbf{A}(n)$ is a $K \times K$ matrix such that we have $[\mathbf{A}(n)]_{ii} = 1 - \varepsilon$ on the main diagonal and $[\mathbf{A}(n)]_{ij} = \varepsilon \cdot \alpha_{ij}(n)$ for $i \neq j$.

Notice that *even though we assume channel reciprocity, matrix $\mathbf{A}(n)$ is not symmetric*. Moreover, by construction, matrix $\mathbf{A}(n)$ is nonnegative and stochastic since the sum of the elements on each row sums to one, or equivalently

$$\mathbf{A}(n) \cdot \mathbf{1} = \mathbf{1}. \quad (8)$$

3. TIME-INVARIANT FREQUENCY-SYNCHRONOUS NETWORK

In this section, we study the convergence properties of the distributed synchronization algorithm (5) under the following assumptions: (i) frequency-synchronous network, that is, all the clocks share the same period $T = T_1 = \cdots = T_K$; (ii) the network is time-invariant, that is, $P_{ki}(n) = P_{ki}$ for any n and $k \neq i$. From assumption (i), the clock of the k th node can be expressed as

$$t_k(n) = nT + \tau_k(n), \quad (9)$$

where $\tau_k(n)$ is the timing phase $0 \leq \tau_k(n) < T$ of the k th node in the n th period (see Figure 1). Moreover, by substituting (9) into (5a) and using assumption (ii), it easily follows that the synchronization algorithm (5) can be written in terms of the phases $\tau_k(n)$ as

$$\tau_k(n+1) = \tau_k(n) + \varepsilon \cdot \Delta \tau_k(n+1), \quad (10a)$$

$$\Delta \tau_k(n+1) = \sum_{i=1, i \neq k}^K \alpha_{ki}(\tau_i(n) - \tau_k(n)) \quad (10b)$$

with coefficients α_{ki} :

$$\alpha_{ki} = \frac{P_{ki}}{\sum_{j=1, j \neq k}^K P_{kj}}. \quad (11)$$

Finally, by defining the vector containing the timings of all nodes as $\boldsymbol{\tau}(n) = [\tau_1(n) \cdots \tau_K(n)]^T$, the vector model (7) becomes

$$\boldsymbol{\tau}(n+1) = \mathbf{A} \cdot \boldsymbol{\tau}(n), \quad (12)$$

where \mathbf{A} is a $K \times K$ matrix such that we have $[\mathbf{A}]_{ii} = 1 - \varepsilon$ on the main diagonal and $[\mathbf{A}]_{ij} = \varepsilon \cdot \alpha_{ij}$ for $i \neq j$.

Model (12) resembles the one considered in the literature on multiagent coordination (see, e.g., [13]). The goal of this section is to determine the conditions under which the system (12) converges to a unique cluster or to multiple clusters of synchronization for a fixed realization of the fading variables G_{ki} in (4), that is, matrix \mathbf{A} is assumed to be deterministic. We will define the conditions of convergence in terms of the properties of the graph associated to the wireless network under study, or equivalently in terms of the system matrix \mathbf{A} .

³ A scenario with additive noise in the update rule, that models jitter in the local clocks, could be treated by using the theory developed in [17]. This issue is outside the scope of this paper and will not be further pursued here.

3.1. The associated graph and useful definitions

The synchronization algorithm defines a weighted directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ of order K on the sensor network, where $\mathcal{V} = \{1, \dots, K\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges weighted by the off-diagonal elements of the $K \times K$ adjacency matrix $[\mathcal{A}]_{ij} = \alpha_{ij}$. The edge connecting the i th and the j th nodes, $i \neq j$, belongs to \mathcal{E} if and only if $\alpha_{ij} > 0$. Notice that the graph is directed ($\alpha_{ij} \neq \alpha_{ji}$ for $i \neq j$), even though fading links are reciprocal ($P_{ij} = P_{ji}$ for $i \neq j$). Moreover, notice that the system matrix reads

$$\mathbf{A} = \mathbf{I} - \epsilon \mathbf{L}, \quad (13)$$

where \mathbf{L} is the graph Laplacian of the network that is defined as [13]: $[\mathbf{L}]_{ii} = 1$ (which is the degree of node i : $\sum_{j \neq i} \alpha_{ij}$) and $[\mathbf{L}]_{ij} = -\alpha_{ij}$ for $i \neq j$. The main result of this section (Theorem 1) relates the convergence properties of the distributed synchronization procedure in (10) with the connectivity of the graph \mathcal{G} associated to the sensor network. We need the following definitions.

Definition 1. A graph \mathcal{G} is said to be strongly connected if there exists a path (i.e., a collection of edges in \mathcal{E}) that links every pair of nodes.

It can be proved that strong connectivity of graph \mathcal{G} is equivalent to the irreducibility of matrix \mathbf{A} [18].

Definition 2. A $K \times K$ matrix \mathbf{A} is said to be *reducible* if there exists a $K \times K$ permutation matrix \mathbf{P} and an integer $r > 0$ such that

$$\mathbf{P}^T \mathbf{A} \mathbf{P} = \begin{bmatrix} \mathbf{B} & \mathbf{C} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}, \quad (14)$$

where \mathbf{B} is $r \times r$, \mathbf{D} is $K - r \times K - r$, \mathbf{C} is $r \times K - r$, and the zero matrix $\mathbf{0}$ is $K - r \times r$. A matrix \mathbf{A} is called irreducible if it is not reducible.

The degree of irreducibility of a matrix \mathbf{A} , or equivalently of strong connectivity of the associated graph \mathcal{G} , can be measured by the following quantity

$$\sigma = \min_{\mathcal{V}_1, \mathcal{V}_2} \left(\sum_{i \in \mathcal{V}_1, \notin \mathcal{V}_2} \alpha_{ij} + \sum_{i \in \mathcal{V}_2, \notin \mathcal{V}_1} \alpha_{ij} \right), \quad (15)$$

where the minimum is taken over all nonempty proper subsets of \mathcal{V} , $\mathcal{V}_1 \cap \mathcal{V}_2 = \emptyset$ ($\mathcal{V}_1 \cup \mathcal{V}_2 = \mathcal{V}$). It can be shown that $\sigma = 0$ if and only if the matrix \mathbf{A} is reducible, or the associated graph \mathcal{G} is not strongly connected [19].⁴

3.2. Convergence properties

The main result of this section can be now stated as follows.

Theorem 1. (i) The distributed synchronization (10) converges to a unique cluster of synchronized nodes, $\tau_1(n) = \dots = \tau_K(n) = \tau^*$ for $n \rightarrow \infty$, if and only if the associated weighted directed graph \mathcal{G} is strongly connected, or equivalently if system matrix \mathbf{A} is irreducible. (ii) In this case, the system (12) converges to (for $n \rightarrow \infty$)

$$\boldsymbol{\tau}(n) \rightarrow \boldsymbol{\tau}^* = \mathbf{1} \cdot \mathbf{v}^T \boldsymbol{\tau}(0), \quad (16)$$

or equivalently $\tau_k(n) \rightarrow \tau_k^* = \mathbf{v}^T \boldsymbol{\tau}(0)$ for $k = 1, \dots, K$, where \mathbf{v} is the normalized left eigenvector of matrix \mathbf{A} corresponding to eigenvalue 1: $\mathbf{A}^T \mathbf{v} = \mathbf{v}$ with $\mathbf{1}^T \mathbf{v} = 1$.

An immediate consequence of Theorem 1 is that the timing vectors converge to the average of their initial values $\boldsymbol{\tau}(0)$ if and only if the system matrix \mathbf{A} is doubly stochastic (i.e., if \mathbf{A}^T is stochastic as well). In fact, in this case $\mathbf{A}^T \mathbf{1} = \mathbf{1}$ and vector \mathbf{v} in (16) reads $\mathbf{v} = 1/K \cdot \mathbf{1}$. This condition occurs in balanced networks [13], where $\sum_{i \neq j} \alpha_{ij} = 1 = \sum_{i \neq j} \alpha_{ji}$. In sensor networks, this result is of interest in applications where the steady state value of synchronization is used in order to infer the status of the process monitored by the sensor [8, 9, 20].

Proof. The proof of part (i) of Theorem 1 is available in the literature for applications where the graph \mathcal{G} associated to the dynamic system (12) is undirected [5]. In the case of a directed graph, strong connectivity can generally be proved to be only a sufficient condition for synchronization. However, in a wireless fading case with reciprocal channels, the result can be proved as shown in the following. The second part (ii) of Theorem 1 follows from a result derived, among the others, in [13].

As explained above, in order to prove Theorem 1, we only need to show that strong connectivity is also a necessary condition for synchronization. As a by-product, the proposed proof brings insight into the formation of multiple clusters of synchronization (3). Let us assume that \mathbf{A} is reducible (or equivalently the associated graph \mathcal{G} is not strongly connected). Then, by definition, there exists a permutation matrix \mathbf{P} and an integer $r > 0$ such that (14) holds. But if $\alpha_{ij} = 0$ in \mathbf{A} , then for reciprocity $P_{ij} = P_{ji} = 0$ and then $\alpha_{ji} = 0$ ($i \neq j$). This property is sometimes referred to as *bidirectionality* of the graph (i.e., $\alpha_{ij} = 0$ if and only if $\alpha_{ji} = 0$ but α_{ij} and α_{ji} need not to be equal [14]). Therefore, the $r \times K - r$ matrix \mathbf{C} in (14) has all zero entries. Since the permuted matrix $\mathbf{P}^T \mathbf{A} \mathbf{P}$ is nonnegative and stochastic, so are submatrices \mathbf{B} and \mathbf{D} . By applying the permutation function $\pi(k) = \mathbf{P}_k [1 \dots K]^T$, where \mathbf{P}_k is the k th row of matrix \mathbf{P} , to the nodes' labels, we can write the system (12) as

$$\tilde{\boldsymbol{\tau}}(n+1) = \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \tilde{\boldsymbol{\tau}}(n), \quad (17)$$

where $\tilde{\boldsymbol{\tau}}(n) = \mathbf{P} \boldsymbol{\tau}(n)$. Therefore, the set of r nodes $\{\pi(1), \dots, \pi(r)\}$ evolves independently from the remaining nodes $\{\pi(r+1), \dots, \pi(K)\}$. Now, if either \mathbf{B} or \mathbf{D} are reducible, the reasoning above can be iterated bringing to the formation of multiple independent set of nodes evolving separately. At the end of this procedure, the system matrix can be

⁴ Equation (15) provides an upper bound on the second largest eigenvalue of the system matrix \mathbf{A} (see also Appendix A).

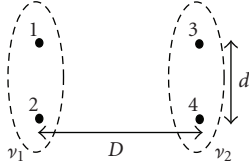


FIGURE 2: The rectangular topology considered in the example in Section 3.3.

written as a block matrix with irreducible stochastic blocks on the diagonal. Without loss of generality, let us then assume that \mathbf{B} and \mathbf{D} are irreducible. From the first part of the proof (see also Appendix A), it follows the two cluster of r and $(K - r)$ nodes synchronize among themselves according to (3). Moreover, the steady state values of the timing vectors depend on the left eigenvectors of \mathbf{B} and \mathbf{D} according to (16):

$$\tau_{\pi(i)}(n) \longrightarrow \mathbf{v}_B^T \tilde{\boldsymbol{\tau}}_r(0), \quad i = 1, \dots, r, \quad (18a)$$

$$\tau_{\pi(i)}(n) = \mathbf{v}_D^T \tilde{\boldsymbol{\tau}}_{K-r}(0), \quad i = r + 1, \dots, K - r, \quad (18b)$$

where $\mathbf{B}^T \mathbf{v}_B = \mathbf{v}_B$, $\mathbf{D}^T \mathbf{v}_D = \mathbf{v}_D$, $\tilde{\boldsymbol{\tau}}_r(n) = [\tau_{\pi(1)}(n) \cdots \tau_{\pi(r)}(n)]$ is the $r \times 1$ vector collecting the first r entries of $\tilde{\boldsymbol{\tau}}(n)$ and $\tilde{\boldsymbol{\tau}}_{K-r}(n) = [\tau_{\pi(r+1)}(n) \cdots \tau_{\pi(K)}(n)]$ is the $K - r \times 1$ vector collecting the remaining entries. \square

The convergence of the dynamic system at hand could be also studied in terms of the subdominant eigenvalue of matrix \mathbf{A} , similarly to approach commonly adopted in the context of the analysis of Markov chains [21]. In particular, the following results can be proved relating convergence to the multiplicity of eigenvalue 1.

Theorem 2. *The distributed synchronization (10) converges to a unique cluster of synchronized nodes as in (2) if and only if the subdominant eigenvalue $\lambda_2 \neq 1$.*

Proof. By recalling Theorem 1, it is enough to prove that: (i) if $\lambda_2 = 1$, then the graph is not strongly connected; (ii) if the graph is not strongly connected then, $\lambda_2 = 1$. Part (i) can be proved similarly to [13]; however, in Appendix A we give an alternative proof based on the measure σ in (15) of irreducibility of \mathbf{A} . Part (ii) does not hold in general for problems with directed graphs but it is easily shown under the reciprocity assumption similarly to Theorem 1. \square

3.3. Numerical results

Here, we present a numerical example to corroborate the analysis discussed above. A network of $K = 4$ nodes is considered where the nodes are divided into two groups, $\mathcal{V}_1 = \{1, 2\}$ and $\mathcal{V}_2 = \{3, 4\}$, as in Figure 2. The initial phases $\tau_k(0)$ are set to $\boldsymbol{\tau}(0)/T = [0.1 \ 0.4 \ 0.6 \ 0.8]^T$. Fading variables G_{ki} are equal to 1, the path loss exponent is $\gamma = 3$, $D/d = 2$, and $\epsilon = 0.3$. Notice that, given the definition (11), the performance is not affected by the value of C in (4) and it only depends on relative distances. Figure 3 shows the timing vector

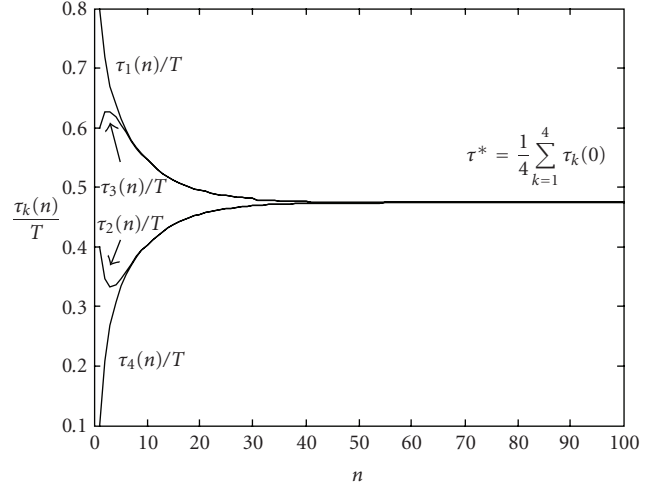


FIGURE 3: Timing phases $\{\tau_k(n)\}_{k=1}^K$ versus the period n for the rectangular topology in Figure 2 with $D/d = 2$ ($\epsilon = 0.3$, $\gamma = 3$, $K = 4$).

$\boldsymbol{\tau}(n)$ versus n . After a transient where the nodes tend to synchronize in pairs within the two groups, the system reaches the steady state to the average value $\tau^*/T = 0.475$, as stated in Theorem 1, since the system matrix is easily shown to be doubly stochastic for this specific example.

In order to quantify the rate of convergence, from Theorem 2, we notice that the convergence of the synchronization protocol (10) depends on the subdominant eigenvalue λ_2 . In particular, as it is well known from the theory of linear difference equations, the rate of convergence is ruled by a term proportional to $|\lambda_2|^n$ (see, e.g., [22]). If we define a threshold λ_o , we could say that the protocol reaches the steady state condition at the time instant n_o for which $|\lambda_2|^{n_o} = \lambda_o$: $n_o = \log \lambda_o / \log |\lambda_2|$. Therefore, we can take

$$v = -\log |\lambda_2| \quad (19)$$

as a measure of the rate of convergence of the algorithm. Figure 4 shows the rate of convergence v versus the normalized distance D/d for $\epsilon = 0.3, 0.7$. As expected the rate v decreases with increasing D/d and decreasing ϵ . Along with v , Figure 4 shows the measure of irreducibility (or strong connectivity) σ (15) as dashed lines. It is interesting to note that the rate of convergence v and the measure of irreducibility σ have the same behavior as a function of D/d and ϵ . This confirms that convergence is strictly related to the connectivity properties of the associated graph, as proved in Theorem 1.

3.4. Effect of fading: an example

In this section, the effect of fading on the rate of convergence v is investigated via simulation for linear, ring, and star topologies (see Figure 5). Rayleigh fading is assumed, that is, the fading amplitude G_{ki} in (4) accounts for Rayleigh fading with unit average power. Fading is assumed to be constant for any n during the evolution of the algorithm. Figure 6 plots the average rate of convergence $E[v]$ (where the average $E[\cdot]$ is taken with respect to the distribution of fading) for the

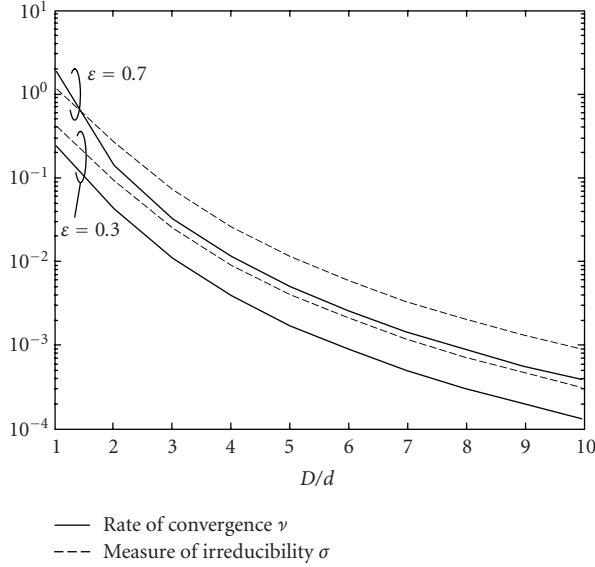


FIGURE 4: Rate of convergence ν (19) and the measure of irreducibility σ (15) versus D/d for the rectangular topology in Figure 2 ($\varepsilon = 0.3, 0.7, \gamma = 3, K = 4$).

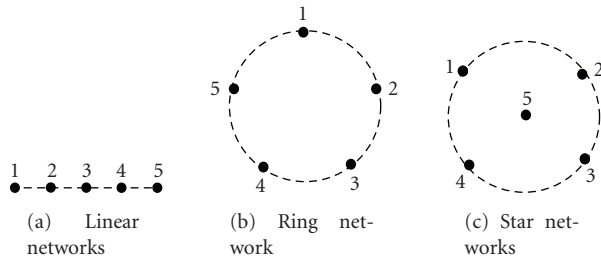


FIGURE 5: The linear, ring, and star networks ($K = 5$).

three networks versus the number of nodes K ($\varepsilon = 0.3$). Notice that for $K = 2$ the three networks coincide and recall that only relative distances are of concern for the behavior of the system (10). As it is expected, the star topology has the largest rate of convergences whereas the linear network yields the slowest convergence.

4. TIME-VARYING FREQUENCY-SYNCHRONOUS NETWORK

Here, we reconsider the performance of the synchronization procedure (5) by removing the assumption of time-invariance underlying the analysis of the previous section. However, we still assume a frequency-synchronous network. Overall, the system (5) can be written in vector form in terms of phases as (recall (12))

$$\boldsymbol{\tau}(n+1) = \mathbf{A}(n)\boldsymbol{\tau}(n), \quad (20)$$

with the definition of the system matrix $\mathbf{A}(n)$ in Section 2. In this case, the sensor network can be described by a sequence of directed graphs $\mathcal{G}(n) = (\mathcal{V}, \mathcal{E}(n), \mathcal{A}(n))$ of order

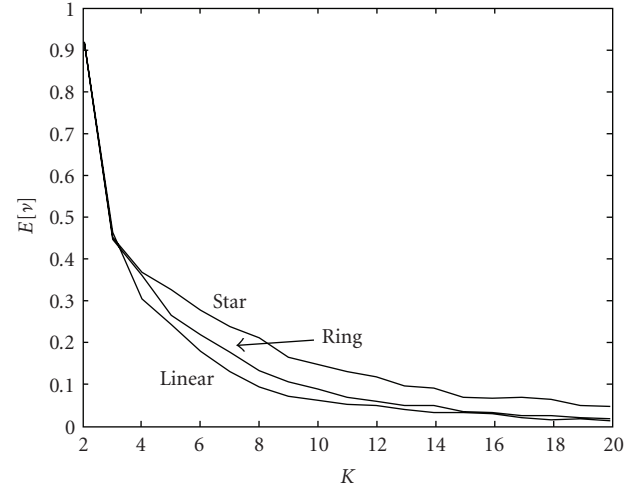


FIGURE 6: Average rate of convergence $E[\nu]$ for the linear, ring, and star networks in Figure 5 versus the number of nodes K ($\varepsilon = 0.3, \gamma = 3$).

K , defined similarly to Section 3. In particular, $\mathcal{A}(n)$ is the adjacency matrix $[\mathcal{A}(n)]_{ij} = \alpha_{ij}(n)$ and the edge connecting the i th and the j th nodes, $i \neq j$, belongs to $\mathcal{E}(n)$ if and only if $\alpha_{ij}(n) > 0$. At each time instant n , the dynamic system describing the synchronization process evolves as where $\mathbf{A}(n) = \mathbf{I} - \varepsilon \mathbf{L}(n)$ with $\mathbf{L}(n)$ being the graph Laplacian at time n (see Section 3). Study of convergence of a family of algorithms encompassing (10) has been recently attempted in a few works (see [13, 14] and references therein). In particular, adapting a result first presented in [14] to our case, we are able to relate the convergence of dynamic system (20) to the connectivity properties of the associated sequence of graphs $\mathcal{G}(n)$. We need the following definition.

Definition 3. A sequence of graphs $\mathcal{G}(n)$ is said to be strongly connected across an interval $I \subseteq \{0, 1, 2, \dots\}$ if the directed graph $(\mathcal{V}, \bigcup_{n \in I} \mathcal{E}(n), \bigcup_{n \in I} \mathcal{A}(n))$ is strongly connected (see Definition 1).

Theorem 3. The distributed synchronization (20) in a time-varying topology converges to a unique cluster of synchronized nodes, $\tau_1(n) = \tau_2(n) = \dots = \tau_k(n)$ for $n \rightarrow \infty$, if and only if the associated sequence of graphs $\mathcal{G}(n)$ is strongly connected across $[n_0, \infty)$ for any $n_0 = 0, 1, 2, \dots$.

Proof. Theorem 3 can be proved by specializing the proof of [14, Theorem 3] to our scenario. The basic idea is to exploit the convexity, or the contractive property, of transformation (10). An interesting remark is that reciprocity of fading plays here a key role as it did in the proof of Theorem 1 for the case of fixed topology. Reciprocity of fading translates into bidirectionality of the associated graphs (see Section 3). As proved in [14], in presence of unidirectional communication among nodes (i.e., nonreciprocal fading in our scenario), convergence of synchronization is not necessarily guaranteed if each sensor communicates to every other sensor (either

directly or via intermediate nodes) in an interval $[n_0, \infty)$. On the contrary, in order to guarantee convergence in a unidirectional graph, a limit should be imposed on the time it takes for the graph to become strongly connected, that is, the interval in Theorem 3 should be modified as $[n_0, n_0 + T]$ where $T \geq 0$ finite. \square

5. FREQUENCY-ASYNCHRONOUS NETWORK

In the previous sections, it was assumed that all the nodes have the same clock period T (frequency synchronous network). However, in practice, different nodes might have different frequencies $\{1/T_k\}_{k=1}^K$, and the question arises of whether or not the physical layer-based scheme (5) is still able to achieve synchronization on a strictly connected graph. For a time-invariant scenario (i.e., $P_{ki}(n) = P_{ki}$ for any n and $k \neq i$), it will be shown below that, in presence of a frequency mismatch, the scheme (10) is able to synchronize the clock periods of the nodes (recall (1)), but not their timing phases, so that the full synchronization condition (2) is not achieved. In this regard, it should be noted that, while perfect synchronization (2) is necessary for many applications, in other scenarios having nodes with synchronized frequency is the only requirement (i.e., to ensure equal sensor duty cycles).

For a frequency-asynchronous time-invariant network, the considered synchronization scheme (5) reads

$$t_k(n+1) = t_k(n) + \varepsilon \cdot \Delta t_k(n+1) + T_k, \quad (21a)$$

$$\Delta t_k(n+1) = \sum_{i=1, i \neq k}^K \alpha_{ki}(t_i(n) - t_k(n)), \quad (21b)$$

or, in vector form (see (7)):

$$\mathbf{t}(n+1) = \mathbf{A} \cdot \mathbf{t}(n) + \mathbf{T}. \quad (22)$$

Let us now denote a possible common value for the clock period of all nodes as T (to be determined) as in (1). It follows that the clock of the k th sensor can then be written for sufficiently large n as

$$t_k(n) = nT + \tau_k(n), \quad (23)$$

or equivalently, in vector form, as $\mathbf{t}(n) = nT \cdot \mathbf{1} + \boldsymbol{\tau}(n)$ with $\boldsymbol{\tau}(n) = [\tau_1(n) \cdots \tau_K(n)]^T$. We are interested in determining if such common frequency $1/T$ exists and, if so, whether eventually the phases $\boldsymbol{\tau}(n)$ converge to the same value for $n \rightarrow \infty$. The main conclusion is summarized in the theorem below, whose proof is inspired by the analysis of the convergence of coupled analog oscillators in [23].

Theorem 4. *With reference to (23), under the assumption that the graph \mathcal{G} is strictly connected, the system (21) synchronizes the clocks of the K nodes to the common period*

$$T = \mathbf{v}^T \mathbf{T}, \quad (24)$$

where \mathbf{v} is the normalized left eigenvector of matrix \mathbf{A} corresponding to eigenvalue 1: $\mathbf{A}^T \mathbf{v} = \mathbf{v}$ with $\mathbf{1}^T \mathbf{v} = 1$. However, the

timing phases $\boldsymbol{\tau}(n)$ remain generally mismatched and given for $n \rightarrow \infty$ by

$$\boldsymbol{\tau}(n) \longrightarrow \boldsymbol{\tau}^* = \mathbf{1} \cdot \eta + \frac{\mathbf{L}^\dagger}{\varepsilon} \Delta \mathbf{T}, \quad (25)$$

with $(\cdot)^\dagger$ denoting the pseudoinverse and the definitions

$$\eta = \mathbf{v}^T \left(\boldsymbol{\tau}(0) - \frac{\mathbf{L}^\dagger}{\varepsilon} \Delta \mathbf{T} \right), \quad (26)$$

$$[\Delta \mathbf{T}]_k = \Delta T_k = T_k - T. \quad (27)$$

The theorem above states that, in presence of a frequency mismatch, the algorithm (21) is able to synchronize the frequencies of different nodes to the common clock period T in (24). However, the system does not lead to phase-synchronous clocks, and the phase error is determined by the frequency (period) mismatch $\Delta \mathbf{T}$ according to (25). Notice that, if the network is such that the system matrix \mathbf{A} is doubly stochastic (as in the example of Section 3.3), the eigenvector \mathbf{v} reads $1/K$ and the common period T is in this case the average $T = 1/K \sum_{k=1}^K T_k$. Moreover, with doubly stochastic matrix \mathbf{A} , condition (25) simplifies as $\eta = 1/K \sum_{k=1}^K \tau_k(0)$ since $\mathbf{1}^T \mathbf{L}^\dagger = \mathbf{0}$ (see proof below for further details). Finally, we remark that if the frequency mismatch is $\Delta \mathbf{T} = \mathbf{0}$ (or equivalently $T_k = T$), Theorem 4 follows from Theorem 1.

Proof. Under the assumption of a connected graph (or irreducible matrix \mathbf{A}), according to Theorem 2 or [13], the Laplacian \mathbf{L} is easily shown to have rank $K - 1$, where the one-dimensional null subspaces are defined by the relationships

$$\mathbf{v}^T \mathbf{L} = \mathbf{0}^T, \quad \mathbf{L} \mathbf{1} = \mathbf{0}. \quad (28)$$

Using the latter equality, recalling (13) and the definition of common clock period T and phases $\boldsymbol{\tau}(n)$ in (23), the vector difference equation (22) can be written as

$$\boldsymbol{\tau}(n+1) - \boldsymbol{\tau}(n) = -\varepsilon \mathbf{L} \cdot \boldsymbol{\tau}(n) + \Delta \mathbf{T}. \quad (29)$$

An equilibrium state $\boldsymbol{\tau}(n+1) = \boldsymbol{\tau}(n) = \boldsymbol{\tau}^*$ for the difference equation (29) satisfies $\boldsymbol{\tau}(n+1) - \boldsymbol{\tau}(n) = \mathbf{0}$, which yields the condition

$$\mathbf{L} \cdot \boldsymbol{\tau}^* = \frac{\Delta \mathbf{T}}{\varepsilon}. \quad (30)$$

From (28), it follows that (i) in order for (30) to be feasible (i.e., for an equilibrium point to exist), the common clock period T must satisfy $\mathbf{v}^T \Delta \mathbf{T} = \mathbf{0}$ or equivalently (24); (ii) an equilibrium phase vector $\boldsymbol{\tau}^*$ must read $\boldsymbol{\tau}^* = (\mathbf{L}^\dagger / \varepsilon) \Delta \mathbf{T} + \eta \mathbf{1}$ where η is an arbitrary constant. It remains to show that the system actually converges for $n \rightarrow \infty$ to the equilibrium point $\boldsymbol{\tau}^*$ determined above, and to evaluate the constant η .

Toward the goal of studying convergence, let us define $\boldsymbol{\tau}'(n) = \boldsymbol{\tau}(n) - (\mathbf{L}^\dagger / \varepsilon) \Delta \mathbf{T}$. With this change of variables, the difference equation (29) boils down to

$$\boldsymbol{\tau}'(n+1) = \mathbf{A} \cdot \boldsymbol{\tau}'(n), \quad (31)$$

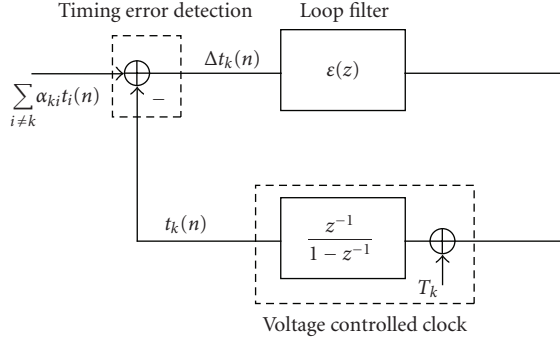


FIGURE 7: Synchronization algorithm (5) as a linear dynamical feedback system: analogy with a discrete-time PLL.

where we used the relationship $\mathbf{L}\mathbf{L}^\dagger\Delta\mathbf{T} = \Delta\mathbf{T}$, which easily follows from the definition of pseudoinverse and (24). As a consequence of (31), as per Theorem 1, we have $\mathbf{r}'(n) \rightarrow \mathbf{v}^T \mathbf{r}'(0)$. This expression is equivalent to (25), thus proving the theorem. Notice that from (31) the rate of convergence is the same as in the case of no frequency mismatch. \square

As a final remark, we notice that the study of time-varying frequency-asynchronous networks is a challenging task and is left for future work.

6. DISTRIBUTED TIME SYNCHRONIZATION AS COUPLED DISCRETE-TIME PLLS

The purpose of this section is to discuss the distributed synchronization algorithm investigated throughout the paper by casting it into the framework of discrete-time phase locked loops (PLLs) [24]. The discussion is not only beneficial for a better understanding of the system, but it also allows us to generalize the system design. In order to appreciate the similarity with a discrete-time PLL, Figure 7 depicts the synchronization procedure (5) carried out at each sensor as a linear dynamic feedback system. Similarly to a discrete-time PLL, the adder at the input evaluates the timing error $\Delta t_k(n)$, that is then multiplied by ϵ and then fed to a voltage controlled clock (VCC) that updates the clock according to (5a). The constant ϵ plays the role of the loop filter in a discrete-time PLL. The procedure (5) can then be interpreted as a first-order discrete-time PLL since the loop filter is a trivial pure gain [25].

From the discussion above, it is clear that the second or third order discrete PLLs⁵ can be obtained by introducing a loop filter $\epsilon(z)$, with one or two poles respectively, instead of the constant ϵ in the synchronization system of Figure 7. For instance, in the case of a second-order loop, we can introduce a pole μ in the loop by setting $\epsilon(z) = \epsilon/(1 - \mu z^{-1})$ with

$0 < \mu < 1$. The corresponding update rule (5a) modifies as

$$t_k(n+1) = t_k(n) + \epsilon \cdot \Delta t_k(n+1) + \mu(t_k(n) - t_k(n-1)) + (1-\mu)T_k. \quad (32)$$

The updating rule (32) essentially corrects the local period T_k by the estimate of the common clock period $t_k(n) - t_k(n-1)$. The convergence analysis of the second-order loop (32) can be carried out similarly to the previous section where a first-order loop was considered. In particular, the following results hold.

Theorem 5. *If the network of distributed PLL is strictly connected and the system (32) converges, then it synchronizes the clocks of the K nodes to the common period (24). However, under the same conditions, the timing phases $\boldsymbol{\tau}(n)$ remain generally mismatched and given for $n \rightarrow \infty$ by*

$$\boldsymbol{\tau}(n) \rightarrow \boldsymbol{\tau}^* = \mathbf{1} \cdot \eta + (1-\mu) \frac{\mathbf{L}^\dagger}{\epsilon} \Delta\mathbf{T}, \quad (33)$$

with $(\cdot)^\dagger$ denoting the pseudoinverse, and with definitions

$$\eta = \mathbf{v}^T \left(\boldsymbol{\tau}(0) - (1-\mu) \frac{\mathbf{L}^\dagger}{\epsilon} \Delta\mathbf{T} \right) \quad (34)$$

and (27).

Proof. The proof is along the lines of the proof of Theorem 4 (see Appendix B for details). \square

Comparing the statement of the previous theorem with the results derived for a first-order loop (Theorem 4), it can be seen that introducing a pole μ in the loop causes a reduction in the steady state phase error by a factor $1-\mu$. However, this reduction comes at the expense of decreased margins of stability. In fact, convergence cannot be guaranteed for all values of $0 < \epsilon < 1$ and $0 < \mu < 1$. We refer to Appendix C for further analysis on this point. Here, we illustrate this issue by means of an example. Consider a network with two nodes. In this case, we have $\alpha_{12} = \alpha_{21} = 1$ and the graph is connected. Figure 8 shows the four eigenvalues of the system matrix associated with (32) (see Appendices B and C for further details):

$$\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A} + \mu \mathbf{I} & -\mu \mathbf{I} \\ \mathbf{I} & \mathbf{0} \end{bmatrix}, \quad (35)$$

for different values of the pole μ and $\epsilon = 0.9$. Notice that the system matrix (35) is 4×4 since (32) is a system of two second order difference equations [22]. Moreover, one eigenvalue of $\tilde{\mathbf{A}}$ is 1 irrespective of the value of μ . The absolute value of the remaining eigenvalues tends to one for $\mu \rightarrow 1$, showing that increasing the value of the pole leads to lack of stability of the equilibrium point (33). Moreover, the value of μ at which a couple of eigenvalues acquire a nonzero imaginary part can be calculated exactly as a function of the spectrum of matrix \mathbf{A} , as shown in Appendix C (see (C.5)).

In order to corroborate the conclusions above, Figure 9 shows the standard deviation $\xi(n)$ of the timing vector $\mathbf{t}(n)$

⁵ As discussed in [25], loops are never built with order larger than three.

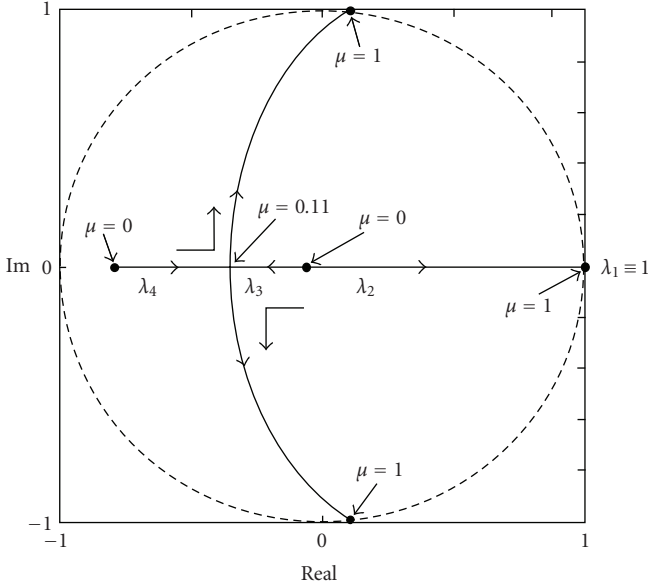


FIGURE 8: Eigenvalues of the second-order loop system (32) in the case of two users with μ increasing from 0 to 1.

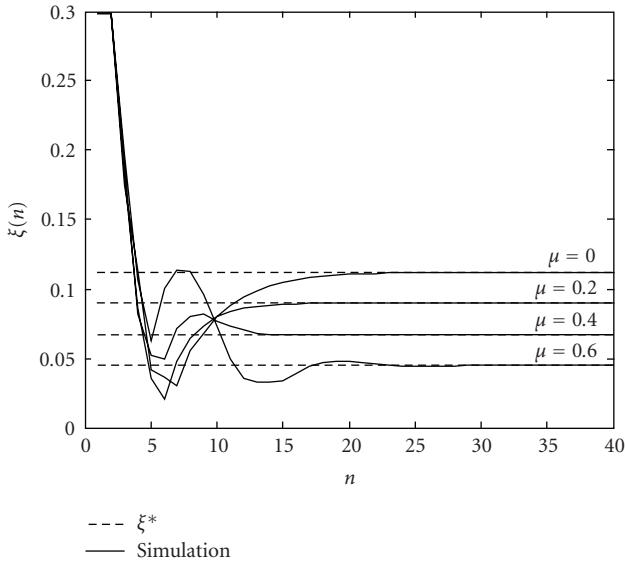


FIGURE 9: Standard deviation $\xi(n)$ of the timing vector $\mathbf{t}(n)$ versus n for the network in Figure 2 for different values of the pole μ ($\gamma = 3$, $D/d = 2$, $\varepsilon = 0.9$). Dashed lines correspond to the analytical result (33).

versus n , where $\xi^2(n) = 1/4 \cdot \sum_{k=1}^4 (t_k(n) - 1/4 \sum_{k=1}^4 t_k(n))^2$, for the network in Figure 2 with parameters $\gamma = 3$, $D/d = 2$, $\varepsilon = 0.9$ and $\Delta T_1 = \Delta T_4 = 0$, $\Delta T_2 = 0.05$, $\Delta T_3 = -0.05$ with $T = 1$. Recall that the graph associated to this network is symmetric. Different values of the pole μ are considered showing the reduction in steady state synchronization error with increasing μ . Dashed lines correspond to the analytical result (33).

To conclude, it is interesting to revisit the results of Theorems 1, 4, and 5 in the light of the analogy with conventional PLLs drawn above. It has been shown that in a strictly connected network: (i) a phase error is perfectly recovered for $n \rightarrow \infty$ by the distributed synchronization algorithm (5) (Theorem 1); (ii) a frequency error is perfectly recovered at the expense of a phase mismatch for $n \rightarrow \infty$ (Theorem 4); (iii) the residual phase mismatch caused by a frequency error can be reduced by introducing a pole in the control loop (Theorem 5). All these results can be read as the counterpart of known facts in the analysis of linearized PLLs, which assert that a first-order loop is indeed able to (i) recover phase errors and (ii) to achieve a constant phase error (referred to as static phase error) in case of a frequency mismatch [25]. Moreover, in this second case, it is interesting to notice how the phase errors (25) and (33) depend on the frequency mismatch ΔT exactly as the static phase error of a PLL [25]. Further results on large-scale randomly deployed networks can be found in [26].

7. IMPLEMENTATION OF DISTRIBUTED COUPLED DISCRETE-TIME OSCILLATORS

In the previous sections, it was assumed that each node is able to measure time differences and powers of other nodes so as to calculate the phase update $\Delta t_k(n)$. Here, we remove this assumption by presenting a practical scheme to implement the phase detector over a bandlimited noisy channel. Since the algorithm is based only on instantaneous power measurements by different nodes, it applies to both a time-invariant and time-variant scenario. The scheme is inspired by the proposal in [12]. A carrier frequency is dedicated to the synchronization channel, where each node, say the k th, transmits a bandlimited waveform $g(t)$ (say a square-root raised cosine pulse) centered at times $t_k(n)$ with symbol period $1/F_s$ (i.e., the time between peak and first zero). The symbol period $1/F_s$ defines the timing resolution of the system.

Each node works in an half-duplex mode and measures the received signal on a interval of duration T_k around the current timing instant $t_k(n)$. Due to the half duplex constraint and the finite switching time between transmitting and receiving mode, each sensor is not able to measure the received signal in an interval of (unilateral) size θ around the firing instant $t_k(n)$. It follows that the observation window reads $t \in (t_k(n) - T_k/2, t_k(n) - \theta) \cup (t_k(n) + \theta, t_k(n) + T_k/2]$. Figure 10(a) illustrates a block diagram of the operations performed at the receiver side by each sensor. The receiver performs baseband filtering matched to the transmitted waveform and then samples the received signal at some multiple L of the symbol frequency F_s , that is, LF_s with $L \geq 1$. Based on the $N = LF_s T$ samples received in the n th observation window, the k th node computes the update (21a) in case a first-order loop is employed, or (32) if a second-order loop is considered. Not knowing the exact timings and powers of other nodes, $t_i(n)$ and P_{ki} with $i \neq k$, the k th sensor cannot directly calculate the updating term $\Delta t_k(n)$ from (5b) and (6). Instead, it estimates these quantities from the received samples, as explained below.

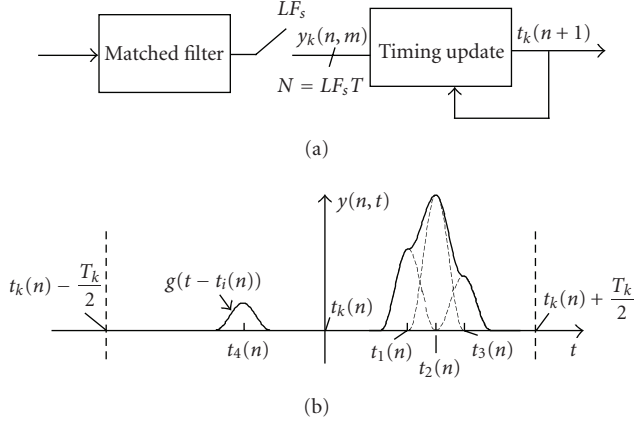


FIGURE 10: (a) Block diagram of the practical implementation of the distributed synchronization scheme discussed in Section 7; (b) a sketch of the received signal (36) in the n th observation window.

After matched filtering and sampling, the discrete-time baseband signal received by the k th node in the n th time period reads (sampling index m ranges within $-N/2 < m \leq N/2$ with $m = 0$ corresponding to the firing time $t_k(n)$ of the k th node):

$$y_k(n, m) = \sum_{i=1, i \neq k}^K \sqrt{E_{ki}} \cdot \beta_{ki} \cdot g\left(\frac{m}{LF_s} - (t_i(n) - t_k(n))\right) + w(n, m), \quad (36)$$

where the average energy per symbol reads $E_{ki} = C/(d_{ki}^y \cdot F_s)$ (recall (4)); β_{ki} denotes the Rayleigh fading coefficient, that is a zero-mean and unit-power complex (circularly symmetric) Gaussian random variable with $|\beta_{ki}|^2 = G_{ki}$; and $w(n, m)$ is the additive Gaussian noise with zero mean and power N_0 . Notice that the sample in the interval $-\Delta LF_s \leq m \leq \Delta LF_s$ is not measured (i.e., zero) due to the half-duplex constraint and the switching time between receive and transmit mode of node k . A sketch of a possible realization of the received signal (36) is provided in Figure 10(b) using an arbitrary waveform $g(t)$.

A simple estimate of $\Delta T_k(n)$ can then be obtained as

$$\Delta t_k(n+1) = \sum_{m \in \mathcal{J}} \bar{\alpha}_{km} \cdot \frac{m}{LF_s}, \quad (37a)$$

$$\bar{\alpha}_{km} = \frac{|y_k(n, m)|^2}{\sum_{i \in \mathcal{J}} |y_k(n, i)|^2}, \quad (37b)$$

where \mathcal{J} is the subset of time instants $m \in (-N/2, -\theta LF_s) \cup (\theta LF_s, N/2)$ for which the received signal $|y_k(n, m)|^2$ is above a given threshold as in [12]. The threshold is a system parameter that can be optimized. Being based solely on instantaneous power measurements (i.e., on samples $|y_k(n, m)|^2$), the practical scheme (37) has the advantage that it does not require any a priori knowledge at the nodes about network topology or average received powers.

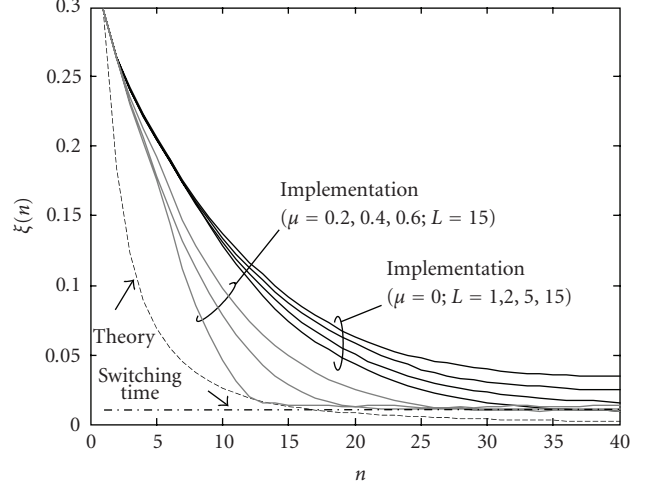


FIGURE 11: Standard deviation of the timing vectors $\xi(n)$ for the synchronization algorithm over a bandlimited Gaussian channel (37) and for the dynamic system (10) (network in Figure 2, SNR = 15 dB, $D/d = 2$, $\varepsilon = 0.9$, $\gamma = 3$, $K = 4$).

For the example of Section 3.3 with no fading ($\beta_{ki} = 1$ for every i and k), Figure 11 shows the standard deviation $\xi(n)$ of the timing vector $\mathbf{t}(n)$ versus n , where $\xi^2(n) = 1/4 \cdot E[\sum_{k=1}^4 (t_k(n) - 1/4 \sum_{k=1}^4 t_k(n))^2]$ and expectation $E[\cdot]$ is taken with respect to noise. We are considering equal clock periods $T_k = T = 1$ for $k = 1, \dots, K$. Moreover, it is assumed that all nodes transmit the same power and the signal-to-noise ratio for transmission to the closest node (e.g., from 2 to 1) is set to SNR = $E_{12}/N_0 = 15$ dB. Other parameters are as follows: $\varepsilon = 0.9$; the threshold is set for simplicity to zero (see discussion below); distances satisfy $D/d = 2$; the normalized timing resolution is $1/F_s = 0.01$; the waveform $g(t)$ is a raised cosine with roll-off $\delta = 0.2$; the switching time is set to $\theta = 1/F_s$.⁶ We first consider the first-order loop (21a) (or equivalently (32) with pole $\mu = 0$) with different oversampling factors $L = 1, 2, 5, 10, 15$. It can be seen that the finite resolution of the system produces a performance floor for increasing n , that can be lowered by increasing the oversampling factor L . In any case, an upper bound on the accuracy of synchronization is set by the finite switching time $\theta = 0.01$. This bound is reached for n and L sufficiently large.⁷ Adding a pole in the loop as in (32) can increase the convergence speed as shown in Figure 11 for $\mu = 0.2, 0.4, 0.6$. Notice that convergence speed could also be improved by setting an appropriately chosen threshold in (37) (not shown). Finally, Figure 11 shows that an upper bound on the performance of the practical implementation discussed here is set by the

⁶ Since we employ a raised cosine waveform, a more realistic choice would be $\theta \approx 3 \div 4 \cdot 1/F_s$. However, this would make the visualization of the performance as a function of the system parameters less clear.

⁷ This performance limit could be improved by allowing the nodes to remain in the receiving mode for an entire period T_k at some (e.g., randomly selected) time-instant.

performance of the dynamic system (10), where the performance gap is due to noisy observations, finite resolution and finite switching time.

8. CONCLUSIONS

An increasing number of applications of sensor networks requires the availability of a common time reference to all the nodes. Aiming at scalability and complexity reduction, physical-layer based synchronization qualifies as a valid alternative to conventional packet-based synchronization. In this work, the convergence properties of such an approach have been investigated by modeling the network as a set of discrete-time coupled oscillators and relying on the analytical framework of algebraic graph theory. It has been shown that the system can be equivalently studied as a set of distributed discrete-time PLLs. This observation allowed to generalize the design of the synchronization process, and prove that well-established results on conventional PLLs extend naturally to distributed systems. Finally, practical details of the implementation of the synchronization algorithm have been discussed on a bandlimited Gaussian channel.

APPENDICES

A. PROOF OF THEOREM 2

We need to prove that if $\lambda_2 = 1$, then the graph is not strongly connected. Toward this goal, we note that we have the following bound on the measure of irreducibility σ (15) [19]:

$$|1 - \lambda_2| \geq \sigma \frac{8}{2K^2 + (-1)^K - 1}, \quad (\text{A.1})$$

from which it easily follows that if $\lambda_2 = 1$, $\sigma = 0$ and thus the graph is not strongly connected.

B. PROOF OF THEOREM 5

The system of difference equation (32) ($k = 1, \dots, K$) can be stated, similarly to (22) in vector form as

$$\mathbf{t}(n+1) = \mathbf{A} \cdot \mathbf{t}(n) + \mu(\mathbf{t}(n) - \mathbf{t}(n-1)) + (1 - \mu)\mathbf{T}. \quad (\text{B.1})$$

Following the same steps as in the proof of Theorem 4, the system (22) can be written in terms of phases $\boldsymbol{\tau}(n)$ relative to the common period T as

$$\boldsymbol{\tau}(n+1) - \boldsymbol{\tau}(n) = -\varepsilon \mathbf{L} \boldsymbol{\tau}(n) + \mu(\boldsymbol{\tau}(n) - \boldsymbol{\tau}(n-1)) + (1 - \mu)\Delta \mathbf{T}. \quad (\text{B.2})$$

An equilibrium state $\boldsymbol{\tau}^*$ for the difference equation (B.2) satisfies $\boldsymbol{\tau}(n+1) = \boldsymbol{\tau}(n) = \boldsymbol{\tau}(n-1) = \boldsymbol{\tau}^*$, which yields the condition

$$\mathbf{L} \boldsymbol{\tau}^* = (1 - \mu) \frac{\Delta \mathbf{T}}{\varepsilon}. \quad (\text{B.3})$$

From (B.3), it follows that: (i) in order for (B.3) to be feasible (i.e., for an equilibrium point to exist), the common clock period T must satisfy $\mathbf{v}^T \Delta \mathbf{T} = \mathbf{0}$ or equivalently (24);

(ii) an equilibrium phase vector $\boldsymbol{\tau}^*$ must read $\boldsymbol{\tau}^* = (1 - \mu)(\mathbf{L}^\dagger / \varepsilon) \Delta \mathbf{T} + \eta \mathbf{1}$ where η is an arbitrary constant.

Let us define $\boldsymbol{\tau}'(n) = \boldsymbol{\tau}(n) - (1 - \mu)(\mathbf{L}^\dagger \Delta \mathbf{T} / \varepsilon)$. With this change of variables, the difference equation (B.2) boils down to

$$\boldsymbol{\tau}'(n+1) = \mathbf{A} \cdot \boldsymbol{\tau}'(n) + \mu(\boldsymbol{\tau}'(n) - \boldsymbol{\tau}'(n-1)). \quad (\text{B.4})$$

The system (B.4) is a second-order vector difference equation, that can be studied by recasting it as a first-order vector difference equation in terms of vector $\tilde{\boldsymbol{\tau}}(n) = [\boldsymbol{\tau}'(n)^T \quad \boldsymbol{\tau}'(n-1)^T]^T$ with system matrix $\tilde{\mathbf{A}}$ (35) (see, e.g., [22]). Convergence of the corresponding system $\tilde{\boldsymbol{\tau}}(n) = \tilde{\mathbf{A}} \tilde{\boldsymbol{\tau}}(n-1)$ depends on the eigenvalues of $\tilde{\mathbf{A}}$. It is easy to see that $\tilde{\mathbf{A}}$ has an eigenvalue equal to one, with left and (normalized) right eigenvectors $\mathbf{z}_\ell = \mathbf{1}$ and $\mathbf{z}_r = 1/(1 - \mu) \cdot [\mathbf{v}^T \quad -\mu \mathbf{v}^T]^T$ (recall that \mathbf{v} is the right eigenvector of \mathbf{A} corresponding to the eigenvalue $\lambda = 1$). Moreover, it will be shown in Appendix C that this eigenvalue is unique. Therefore, the system (B.4) is stable if and only if all the remaining $2K - 1$ eigenvalues of $\tilde{\mathbf{A}}$ have absolute value less than one (see, e.g., [13]). This point is further investigated in Appendix C. Assuming that the stability condition mentioned above holds, then we have $\tilde{\mathbf{A}}^n \rightarrow \mathbf{z}_\ell \mathbf{z}_r^T$ for $n \rightarrow \infty$ (see, e.g., [13]) and the phases $\boldsymbol{\tau}'(n)$ converge as $\boldsymbol{\tau}'(n) \rightarrow \mathbf{1} \mathbf{v}^T \boldsymbol{\tau}'(0)$ (having set $\boldsymbol{\tau}(-1) = \boldsymbol{\tau}(0)$), which implies that the constant η in (33) is (34).

C. DISCUSSION ON THE STABILITY OF SECOND-ORDER DISTRIBUTED PLLs (32)

In this section, we further investigate the stability of system (32) by studying the eigenvalues of the system matrix $\tilde{\mathbf{A}}$ (35). As discussed in Appendix B, stability is only achieved if and only if all the $2K - 1$ eigenvalues of $\tilde{\mathbf{A}}$, apart from the eigenvalue equal to one, have absolute value less than one. In the following, we show how to relate the eigenvalues of $\tilde{\mathbf{A}}$ with those of the system matrix \mathbf{A} .

In order to evaluate the eigenvalues of $\tilde{\mathbf{A}}$, let us consider that for $\tilde{\lambda}$ to be an eigenvalue of $\tilde{\mathbf{A}}$ there must exist a vector $\mathbf{w} = [\mathbf{w}_1^T \quad \mathbf{w}_2^T]^T$ such that $\tilde{\mathbf{A}} \mathbf{w} = \tilde{\lambda} \mathbf{w}$. After some straightforward algebra, the latter equation becomes

$$\mathbf{A} \mathbf{w}_2 = \frac{\tilde{\lambda}^2 - \mu \tilde{\lambda} + \mu}{\tilde{\lambda}} \mathbf{w}_2. \quad (\text{C.1})$$

Therefore, denoting as λ any of the K eigenvalues of \mathbf{A} , we have the following relationship between the spectrum of matrices \mathbf{A} and $\tilde{\mathbf{A}}$:

$$\tilde{\lambda}^2 - (\mu + \lambda) \tilde{\lambda} + \mu = 0, \quad (\text{C.2})$$

or equivalently

$$\tilde{\lambda}(\lambda, \mu) = \frac{\mu + \lambda \pm \sqrt{\mu^2 + 2\mu(\lambda - 2) + \lambda^2}}{2}. \quad (\text{C.3})$$

It follows that for every eigenvalue λ of \mathbf{A} , we have two eigenvalues of $\tilde{\mathbf{A}}$, given by (C.3). In order to proceed, we need to recall the following result on the eigenvalues of \mathbf{A} .

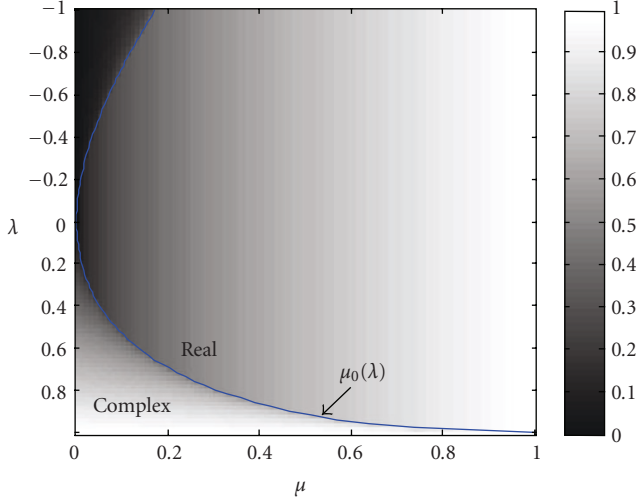


FIGURE 12: Absolute value of the eigenvalue $\tilde{\lambda}(\lambda, \mu)$ (C.3) (with positive sign) of matrix $\tilde{\mathbf{A}}$ (35) versus the eigenvalue λ of matrix \mathbf{A} and the pole μ . Also shown is the threshold value $\mu_0(\lambda)$ (C.5) that discriminates the region where the eigenvalues $\tilde{\lambda}(\lambda, \mu)$ are real or complex conjugate.

Lemma 1. Every eigenvalue of \mathbf{A} satisfies

$$|\lambda - (1 - \varepsilon)| \leq \varepsilon. \quad (\text{C.4})$$

Moreover, if \mathcal{G} is connected, only one eigenvalue $\lambda = 1$.

Proof. See, for example, [13]. \square

From (C.3) and (C.4), it can be seen that the eigenvalue $\tilde{\lambda}$ of $\tilde{\mathbf{A}}$ (with the positive sign in (C.3)) corresponding to $\lambda = 1$ equals one. Moreover, by substituting $\tilde{\lambda} = 1$ in (C.3), this eigenvalue is unique. In order to prove stability of the system at hand, we then need to show that all the other $2K - 1$ eigenvalues $\tilde{\lambda}(\lambda, \mu)$ different from $\tilde{\lambda} = 1$ are such that $|\tilde{\lambda}(\lambda, \mu)| < 1$. In general, it is not easy to find general conditions on ε and μ that guarantee the condition $|\tilde{\lambda}(\lambda, \mu)| < 1$ for any λ in (C.4) (except $\lambda = 1$). However, a numerical study can be easily carried out from (C.3) and (C.4). For illustration, consider an undirected graph \mathcal{G} . In this case, the eigenvalues of matrix \mathbf{A} are real and from (C.4) satisfy $-1 < \lambda < 1$ (recalling that $0 < \varepsilon < 1$). The absolute value $|\tilde{\lambda}(\lambda, \mu)|$ (C.3) (with positive sign, similar result holds for the negative sign) is plotted versus μ and λ in Figure 12. It can be seen that unless $\mu \rightarrow 1$, the stability condition is satisfied. Moreover, by studying the sign of the polynomial in (C.3), it is also possible to evaluate the value of the pole μ , say $\mu_0(\lambda)$, such that for $\mu \leq \mu_0(\lambda)$ the eigenvalues $\tilde{\lambda}$ corresponding to λ are real and for $\mu > \mu_0(\lambda)$ are complex conjugate. This value is easily found to be

$$\mu_0(\lambda) = 2 - \lambda - 2\sqrt{1 - \lambda}, \quad (\text{C.5})$$

which is shown in Figure 12.

As a final remark, we notice that, for $\mu = 1$, the absolute value $|\tilde{\lambda}(\lambda, \mu)|$ equals one irrespective of λ , leading to an unstable system.

ACKNOWLEDGMENT

The authors would like to acknowledge Matteo Cremaschi for validating some of the numerical results in Figures 3, 4, and 6.

REFERENCES

- [1] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, 2004.
- [2] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 5, pp. 1085–1099, 2005.
- [3] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal on Applied Mathematics*, vol. 50, no. 6, pp. 1645–1662, 1990.
- [4] D. Lucarelli and I.-J. Wang, "Decentralized synchronization protocols with nearest neighbor communication," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 62–68, Baltimore, Md, USA, November 2004.
- [5] C. Godsil and G. Royle, *Algebraic Graph Theory*, Springer, New York, NY, USA, 2001.
- [6] G. Werner-Allen, G. Tewari, A. Patel, M. Welsh, and R. Nagpal, "Firefly-inspired sensor network synchronicity with realistic radio effects," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems (SenSys '05)*, pp. 142–153, San Diego, Calif, USA, November 2005.
- [7] A.-S. Hu and S. D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2725–2748, 2006.
- [8] Y.-W. Hong and A. Scaglione, "Distributed change detection in large scale sensor networks through the synchronization of pulse-coupled oscillators," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, vol. 3, pp. 869–872, Montreal, Quebec, Canada, May 2004.
- [9] S. Barbarossa and F. Celano, "Self-organizing sensor networks designed as a population of mutually coupled oscillators," in *Proceedings of the 6th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC '05)*, pp. 475–479, New York, NY, USA, June 2005.
- [10] N. Wakamiya and M. Murata, "Scalable and robust scheme for data fusion in sensor networks," in *Proceedings of the 1st International Workshop on Biologically Inspired Approaches to Advanced Information Technology (Bio-ADIT '04)*, pp. 412–427, Lausanne, Switzerland, January 2004.
- [11] F. Tong and Y. Akaiwa, "Theoretical analysis of interbase-station synchronization systems," *IEEE Transactions on Communications*, vol. 46, no. 5, pp. 590–594, 1998.
- [12] E. Sourour and M. Nakagawa, "Mutual decentralized synchronization for intervehicle communications," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 6, pp. 2015–2027, 1999.
- [13] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

- [14] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, 2005.
- [15] W. C. Lindsey, F. Ghazvinian, W. C. Haggmann, and K. Dessouky, "Network synchronization," *Proceedings of the IEEE*, vol. 73, no. 10, pp. 1445–1467, 1985.
- [16] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice-Hall, Upper Saddle River, NJ, USA, 2001.
- [17] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *Journal of Parallel and Distributed Computing*, vol. 67, no. 1, pp. 33–46, 2007.
- [18] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.
- [19] X.-D. Zhang and R. Luo, "Upper bound for the non-maximal eigenvalues of irreducible nonnegative matrices," *Czechoslovak Mathematical Journal*, vol. 52, no. 3, pp. 537–544, 2002.
- [20] V. V. Veeravalli, "Decentralized quickest change detection," *IEEE Transactions on Information Theory*, vol. 47, no. 4, pp. 1657–1665, 2001.
- [21] J. R. Norris, *Markov Chains*, Cambridge University Press, Cambridge, UK, 1998.
- [22] S. N. Elaydi, *An Introduction to Difference Equations*, Springer, New York, NY, USA, 1999.
- [23] G. Scutari, S. Barbarossa, and L. Pescosolido, "Optimal decentralized estimation through self-synchronizing networks in the presence of propagation delays," in *Proceedings of the 7th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC '06)*, pp. 1–5, Cannes, France, July 2006.
- [24] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Digital Communication Receivers*, John Wiley & Sons, New York, NY, USA, 1998.
- [25] F. M. Gardner, *Phaselock Techniques*, John Wiley & Sons, New York, NY, USA, 1966.
- [26] O. Simeone, U. Spagnolini, and Y. Bar-Ness, "“Small-world” effect of shadowing in pulse-couple distributed time synchronization," *IEEE Communications Letters*, vol. 11, no. 3, pp. 282–284, 2007.