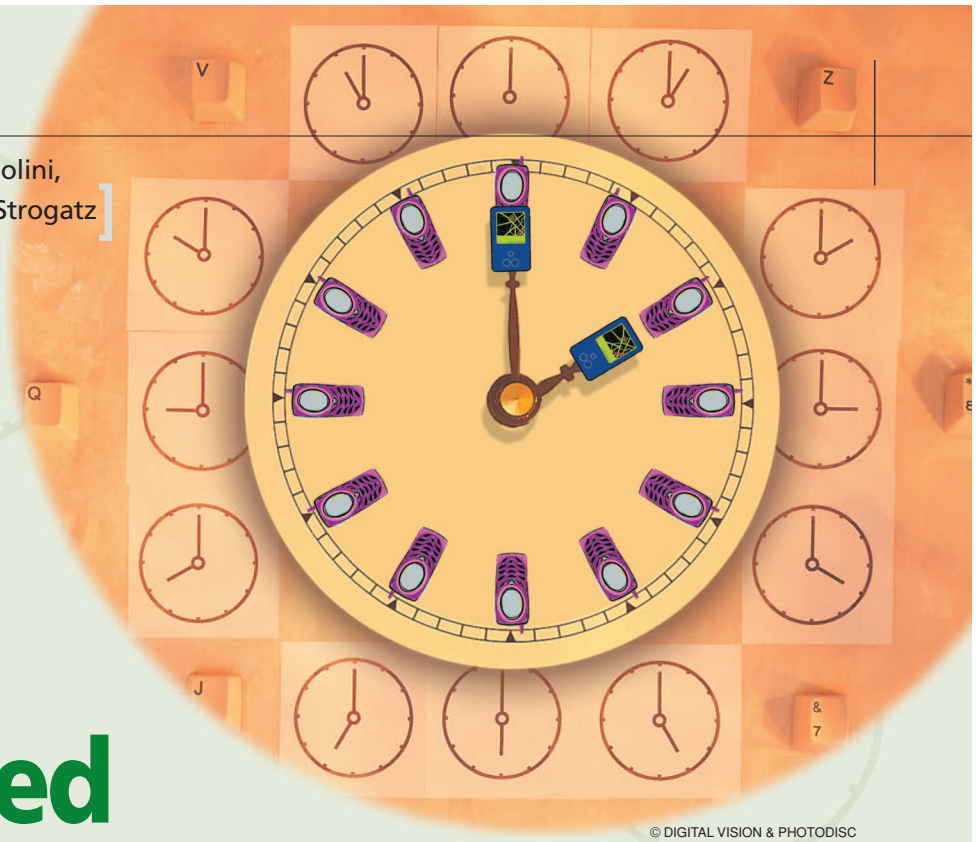
Osvaldo Simeone, Umberto Spagnolini,
Yeheskel Bar-Ness, and Steven H. Strogatz

Global synchronization via local connections



Distributed Synchronization in Wireless Networks

A large number of applications in distributed (sensor or ad hoc) wireless networks is enabled by, or benefit from, the availability of a common time scale among the participating nodes. Examples range from the tracking of moving objects via sensor networks to coordinated medium access control (MAC) or cooperative transmission. Achieving and maintaining synchronization in such scenarios poses new challenges in terms of scalability and energy efficiency and offers new opportunities through the interplay with specific distributed estimation/detection applications. In this context, an interesting solution, which is currently being investigated, is provided by distributed synchronization schemes based on the exchange of local time information among neighboring nodes at the physical layer (e.g., via transmission of a train of common waveforms that follows the local clock).

This article presents a survey of current research on distributed synchronization for decentralized wireless networks and illustrates the role of signal processing therein, with emphasis on physical layer-based synchronization schemes. The topic is introduced by tracing its origin in the parallel investigations carried out independently in mathematical biology and communication theory. Available models are discussed and compared. Open problems, such as the trade-off of complexity versus accuracy and fault tolerance, are outlined and some solutions are provided using tools from signal processing and algebraic graph theory. Available analytical results are also reported, along with numerical examples that corroborate the main conclusions, lending evidence to some interesting phenomena, such as small-world effects on distributed synchronization. The close relationship between distributed synchronization and distributed estimation/detection applications is discussed as well, showing the synergy between these two problems. Finally, synchronization of nonperiodic signals (chaos) is briefly touched upon for completeness and for its (debated) applicability to point-to-point wireless systems.

Digital Object Identifier 10.1109/MSP.2008.926661

INTRODUCTION

Synchronization refers to the process of achieving and maintaining coordination among independent local clocks via the exchange of local time information. Different synchronization schemes differ in the way such information is encoded, exchanged, and processed by the clocks toward the end of overcoming the unavoidable nuisance effects of inaccurate clocks and propagation/processing delays. Wireless communications provide the natural platform for the exchange of local time information between synchronizing clocks. Conversely, synchronization of local clocks enables a wealth of signal processing and communication applications in wireless networks. It is this mutual link between synchronization and wireless networks, with emphasis on decentralized structures such as ad hoc and sensor networks, that constitutes the main subject of this article.

A BRIEF HISTORY OF MUTUAL TIME SYNCHRONIZATION

By the end of the 19th century, synchronization of a distant clock to a reference time, also referred to as unidirectional or master-slave synchronization, became a standard engineering procedure thanks, first, to the advances in telegraphy and, later, wireless transmission. Railroad transportation, geodesy (measurement of longitude), and localization were just a few of the applications enabled by this pervasive new technology. Synchronization of a pair of distant clocks easily qualifies as an early signal processing problem in the context of wired or wireless communications: estimate the time offset between two clocks from measurements affected by propagation delays and random hardware (and human) imperfections [1]. The idea of synchronized time spurred an intense debate in physics and philosophy that eventually produced Einstein's theory of relativity [1]. In this regard, it is interesting to quote H. Poincaré: "Simultaneously is a convention, nothing more than the coordination of clocks by a cross exchange of electromagnetic signals taking into account the transit time of the signal."

In the years following these efforts, scientists wondered at the evidence of synchronization among distributed periodic events in a number of natural phenomena. As late as 1961, Joy Adamson wrote in awe of the observation of the synchronous flashing of fireflies [2]:

> a great belt of light, some ten feet wide, formed by thousands upon thousands of fireflies . . . the fluorescent band composed of these tiny organisms lights up and goes out with a precision that is perfectly synchronized, and one is left wondering what means of communications they possess which enables them to coordinated their shining as though controlled by a mechanical device.

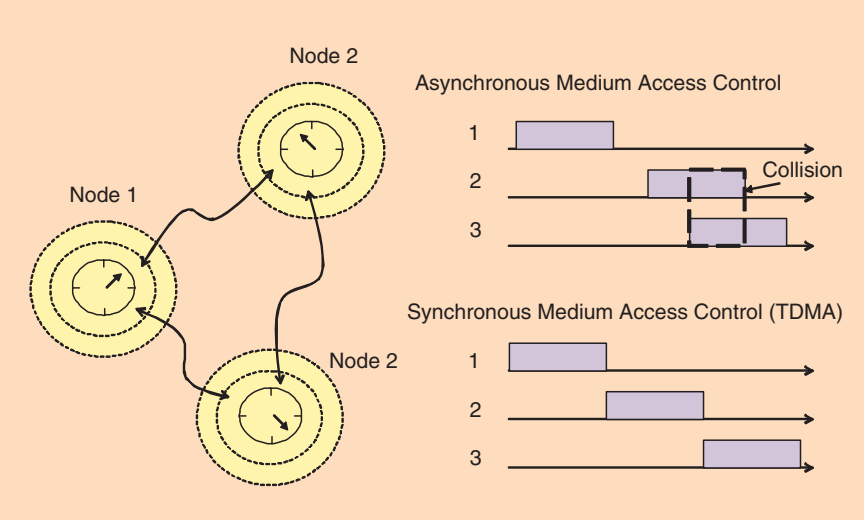
Other typical examples of spontaneous synchrony are the activity of individual fibers in heart muscles to produce the heartbeat or the synchronous hand clapping in a concert hall [2], [3]. Analytical modeling of the dynamic establishment of synchrony, even in the simplest abstractions of such scenarios, challenged mathematical biologists for decades and culminated in the landmark work by Winfree in 1967 [4] and later Kuramoto [5]. It was followed by more recent analyses of Crawford (see [6] and [7] for reviews and references), Mirollo, and Strogatz [8]. In parallel, the communications community started developing a theory of distributed synchronization for telecommunications networks in the 1960s in order to support the deployment of digital switching in the telephone network. This work led to the theory developed by Lindsey et al. [9].

DISTRIBUTED SYNCHRONIZATION IN WIRELESS NETWORKS

In the area of wireless networks, cellular telephony has monopolized the attention of researchers and industry for many years. In this traditional infrastructured scenario, synchronization of mobile stations can be achieved by exploiting a master-slave structure with the base station broadcasting a beacon or training signal. Applications encompass scheduling at the MAC layer and coherent transmission/reception at the physical layer. Distributed synchronization based on wireless communications plays a minor role in this context and has been considered for frame timing synchronization among base stations in [10].

More recently, distributed wireless network structures, such as ad hoc, sensor, or vehicular networks, have started to attract significant interest. In such scenarios, the availability of a common time scale, or of synchronized local oscillators, enables a number of unique functionalities at different layers of the protocol stack. Some representative examples are:

- *signal processing applications*: data fusion of time-sensitive measurements in distributed estimation and tracking for monitoring or surveillance based on sensor networks [11]



[FIG1] An application of synchronization in wireless networks: coordinated (synchronous) medium access control improves spectral/energy efficiency with respect to asynchronous solutions by avoiding collisions and idle periods.

■ *spectral and energy-efficient networking*: coordinated MAC schemes such as time division multiple access or variants, which overcome the shortcomings of collision-based schemes in terms of bandwidth efficiency [12] (see Figure 1); energy-efficient MAC that exploits sleep scheduling [13]

■ *cooperative transmission*: collaborative transmission through space-time coding [14], which requires mutual time synchronization (also referred to as distributed synchronization throughout the article) or distributed beamforming, which demands mutual carrier synchronization [15].

If a fixed or mobile access point is available whose transmission radius covers the entire network (e.g., a mobile fusion center in sensor networks [17]), then network synchronization can be achieved in principle by having the access point broadcast a beacon timing signal, as in cellular networks. This possibility is, for instance, enabled in the IEEE 802.15.4 standard for sensor networks (and associated commercial acticity in the ZigBee alliance) [12]. Moreover, in an outdoor environment with loose constraints on the energy consumption (such as vehicular networks), satellite-based synchronization can be employed. However, in this article we focus on fully distributed scenarios where no such possibilities exist, thus making distributed synchronization (as opposed to master-slave point-to-point synchronization via a central node) the only available solution [16].

As in the problem of synchronizing two distant clocks through electric signals tackled in the late 19th century, mutual synchronization in distributed wireless networks hinges on the exchange of local time information between pairs of nodes. Common complications of both problems are: i) the presence of random delays between transmission and reception of a timing signal, which depends not only on propagation but also on the inevitable processing latency at both sides of the link, and ii) hardware and clock inaccuracies. However, distributed synchronization in wireless networks provides a unique set of challenges for both design and analysis, which call for a variety of tools from signal processing, automatic control, and algebraic graph theory, just to mention a few.

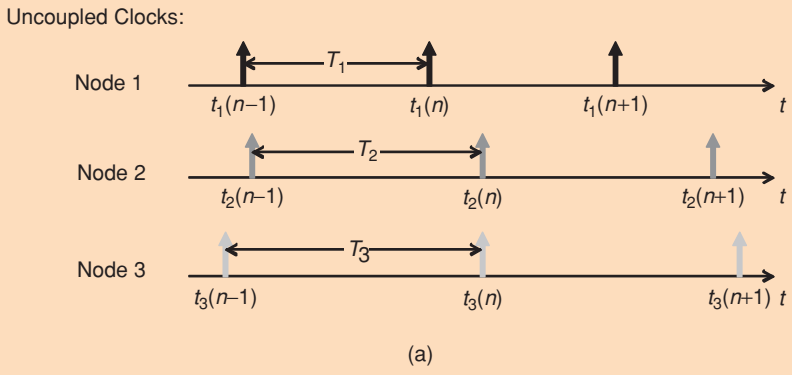
On the one hand, designing mutual synchronization in wireless networks requires to account for the following specific issues:

■ *Energy efficiency*: In the presence of battery-powered nodes, the trade-off between energy consumption and network performance becomes an essential merit criterion [12].

■ *Scalability*: Certain distributed networks, such as microsensor networks, are envisaged to be composed of a large number of nodes, in which case well-behaved scaling performance of synchronization is a critical issue [18].

■ *Application specificity*: In sensor networks, performance is defined in terms of application-specific criteria [19], thus rendering the design of mutual synchronization and the given signal processing functionality thoroughly intertwined [20]–[22].

On the other hand, as discussed below, an analysis of the system often requires consideration of the dynamic behavior of a



[FIG2] Clocks $t_i(n)$ for $N = 3$ nodes in the case of: (a) uncoupled nodes; (b) frequency-synchronous nodes with common frequency $1/T$; (c) fully synchronized nodes.

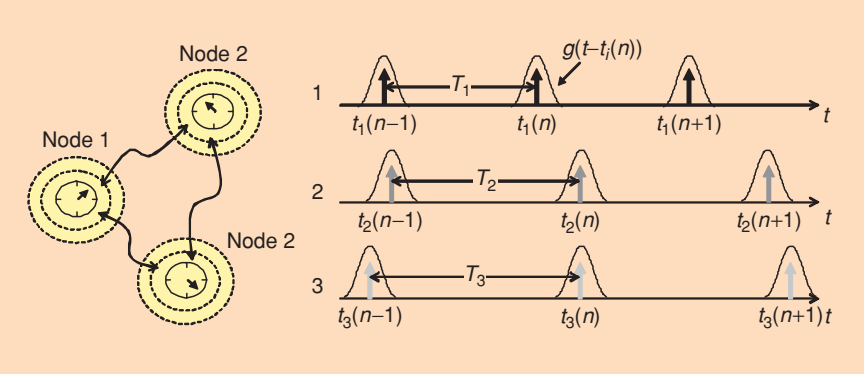
possibly large set of coupled oscillators, which calls for the stability analysis of a system of coupled linear or nonlinear equations. This is generally an involved task, especially in the presence of deterministic or random nuisance parameters.

PACKET-COUPLING VERSUS PULSE-COUPLING FOR MUTUAL SYNCHRONIZATION IN WIRELESS NETWORKS

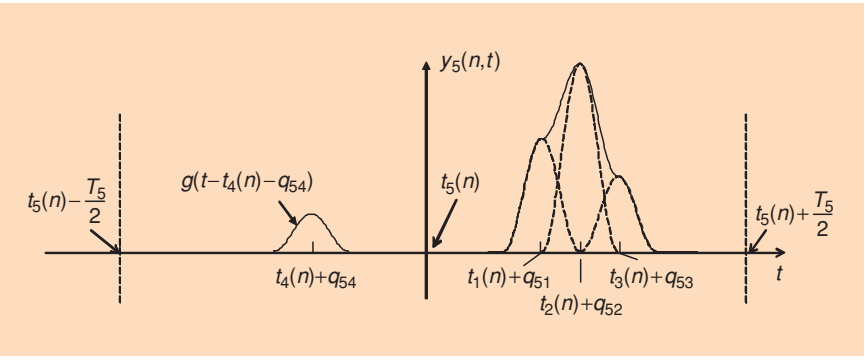
For the time being, we focus the discussion on time synchronization for its practicality and wide range of applications in distributed wireless networks, postponing the discussion on mutual carrier synchronization and its technological challenges to a later section. To illustrate the problem, we consider Figure 2 and define $t_i(n)$ as the time of the nth tick ($n = 0, 1, 2, \dots$) of the ith clock ($i = 1, 2, \dots, N$, where N is the total number of nodes). In Figure 2, the clock at each node is represented by a periodic train of pulses corresponding to time instants $t_i(n)$. In case nodes are uncoupled, i.e., no local timing information is exchanged, the clocks remain asynchronous with generally different local periods $t_i(n) - t_i(n-1) = T_i$, and phases $t_i(n)$ [Figure 2(a)]. On the contrary, if we allow each node, such as the ith, to gather information about the relative time offsets $t_j(n) - t_i(n)$ with respect to a subset of the other nodes ($j \neq i$), a synchronized state might be eventually achieved [Figure 2(b) and (c)]. Notice that the way this time offset information $t_j(n) - t_i(n)$ is exchanged and processed distinguishes different synchronization techniques. We say that a condition of frequency synchronization to a common frequency $1/T$ is achieved if the local periods $t_i(n) - t_i(n-1) = T$ are the same for all clocks [Figure 2(b)], whereas full synchronization is attained if clocks tick at the same times, i.e., $t_i(n) = t_j(n)$, $i \neq j$ [Figure 2(c)]. In the section "Clocks and Synchronization," we further specify and elaborate on these concepts.

Different approaches to mutual time synchronization are classified according to the mechanism adopted for computing and processing local time differences $t_j(n) - t_i(n)$ within the network. In particular, two main families of techniques have been considered. Traditional methods based on packet coupling prescribe the periodic exchange of packets carrying time stamps that contain the local time $t_j(n)$ at the sender, through either point-to-point or broadcast connections [16]. The main sources of errors for packet-based techniques are the random delays associated with the construction of a packet, queuing at the MAC layer, propagation, and processing of the packet at the receiver side. In fact, these delays imply that node i actually receives the timing packet from a node j at time $t_j(n) + q_{ij}$, where q_{ij} is the random delay between the two nodes, thus making the time information on $t_j(n)$ contained in the packet outdated. Different techniques have been designed to mitigate the effects of these random factors according to diverse principles, such as synchronization between receivers of the same packet rather than between transmitter and receiver. The state of the art in packet-based techniques reports synchronization accuracies of the order of milliseconds to microseconds [16]. Moreover, the need for exchanging of a large number of packets is common to all packet-based methods. This in turn entails large computational complexity, energy expenditure, and poor scalability.

To obviate to the drawbacks of packet-based solutions, more recently, there has been interest in physical layer-based schemes, where the local timing information in encoded directly in the transmission times of given waveforms $g(t)$. In particular, each node radiates a periodic train of waveforms $\sum_n g(t - t_i(n))$, according to its local clock, on either a dedicated bandwidth or on an overlay system such as ultra-wideband (UWB); see Figure 3. The update of each local clock is then carried out by processing the received signal, which is a combination of waveforms transmitted by neighboring nodes (see Figure 4). Possible processing techniques include time-of-arrival estimators but efficient synchronization techniques can be devised that do not need to explicitly perform such



[FIG3] A graphic illustration of the signal transmitted by $N = 3$ nodes for pulse-coupled clocks: each node sends a train of waveforms $g(t)$ (on a dedicated bandwidth or on an overlay system) for every tick of the local clock.



[FIG4] A sketch of the signal received by the fifth node out of a set of $N = 5$ pulse-coupled nodes in the nth period of its local clock.

operation [23], [24]. Pulse-coupled synchronization is naturally scalable, since the operations performed at each node are independent of the number of nodes available in the network and has limited complexity, requiring only simple processing at the baseband level (see the section "Trading Accuracy for Bandwith and Complexity").

REMARK 1

The waveform $g(t)$ can be a (possibly band-pass) pulse, as illustrated in Figure 3, but can also have a different shape, such as a pseudorandom sequence or the odd waveform investigated in [18] to study scalability of distributed synchronization. While bearing this in mind, in order to comply with the current literature (see [18] and [25]), we will refer to physical layer-based synchronization as pulse coupling (as opposed to packet coupling).

A landmark work in the area of pulse-coupled time synchronization for wireless networks is [25] by Hong and Scaglione, where the authors investigate a direct application of the model of integrate-and-fire pulse-coupled oscillators studied in the context of mathematical biology in [8]. The results of [25] have been extended in [26] by considering the convergence analysis in presence of more realistic nearest neighbor communications. Moreover, applications of pulse-coupled synchronization to signal processing problems have been discussed in [20] (change detection) and [22] (data fusion). A different approach to pulse-coupled synchronization is the use of linear processing at the baseband level according to the mechanism of discrete-time, phase-locked loops (PLLs). First-order, pulse-coupled linear discrete-time PLLs with frequency-synchronous clocks (i.e., $T_i = T$ for each $i = 1, \ldots, N$) have been proposed in [10] and then [23]. Moreover, [27] studies a mathematically equivalent method, where time information is exchanged via packets. A general model for pulse-coupled linear PLLs is proposed in [24] that can be seen as the discrete time counterpart of the continuously coupled (linearized) analog PLLs considered by Lindsey et al. [9].

REMARK 2

An important technological limitation is the half-duplex constraint imposed on wireless transceivers by the strong self-interference between transmit and receive paths (i.e., nodes cannot transmit and receive simulaneously). Notice, however, that there is some evidence that full-duplex transmission is technologically feasible [28].

Two solutions can be devised to implement pulse-coupled synchronization within such a constraint. The first approach is to choose an impulsive waveform $g(t)$ with a short duration and let nodes switch from transmit to receive mode (and vice versa) before and after transmission of a pulse, thus having a refractory time during transmission where nodes are not able to receive (see the section "Trading Accuracy for Bandwidth and Complexity" and [25] for further details). A second solution is to select any waveform with the desired resolution properties and to transmit a train $\sum_{n\in\mathcal{N}^*} g(t - t_i(n))$ where $\mathcal{N}^*$ is a (e.g., randomly selected) subset of clock periods: accordingly, each node transmits its synchronizing signal in some periods while in other listens to the signal received by other nodes.

CLOCKS AND SYNCHRONIZATION

In this article, we are concerned with a population of N clocks coupled through a wireless channel, as sketched in Figure 3. According to Albert Einstein [1],

> by clock we understand any thing characterized by a phenomenon passing periodically through identical phases so that we must assume, by virtue of the principle of sufficient reason, that all that happens in a given period is identical with all that happened in an arbitrary period.

A clock is then a time measurement device consisting of an oscillator and an accumulator, as detailed below for both analog and discrete time clocks. In this section, we first discuss the baseline scenario of uncoupled clocks [recall Figure 2(a)] and then introduce the basics of mutual synchronization via clock coupling [Figure 2(b) and (c)].

UNCOUPLED CLOCKS

Here we illustrate the behavior of uncoupled clocks, accounting for the reference case where each node runs its own local clock without exchanging timing information with the others.

ANALOG CLOCKS

An analog clock, such as the ith, is characterized by an oscillator

$$s_i(t) = \cos \Phi_i(t), \tag{1}$$

where $\Phi_i(t)$ is the total instantaneous phase (accumulator), which, in case of uncoupled nodes, evolve as

$$\Phi_i(t) = \Phi_i(0) + \frac{2\pi}{T_i} t + \zeta(t), \tag{2}$$

where i), the free-running oscillation period, reads $T_i = T_{\text{nom}} + \Delta T_i$, with T_{nom} being the nominal period and ΔT_i a random offset from the nominal value (related to the frequency offset or skew), that depends on hardware imperfections; ii) $\zeta(t)$ is a typically nonstationary random process modelling phase noise [9]. Moreover, we have selected an arbitrary initial time instant $t = 0$ for all the clocks and $\Phi_i(0)$ is the initial phase ($\Phi_i(t) = 0$ for $t < 0$). A more general model for (1) could be considered that accounts for frequency drifts [9]. In this article, for the sake of simplicity, we will not elaborate on this additional nuisance parameter.

DISCRETE-TIME CLOCKS

A discrete-time clock can be seen as a sequence $t_i(n)$ of significant time instants of an analog clock (e.g., upward zero crossing points: $\Phi_i(t_i(n)) = n \cdot 2\pi$), where index $n = 0, 1, 2, \ldots$ runs over the periods of the oscillator. In particular, from (2), an uncoupled discrete time clock evolves as

$$t_i(n) = t_i(0) + nT_i + \upsilon(t), \tag{3}$$

where $\upsilon(t)$ is the additive noise term that accounts for phase noise. As explained in the previous section, Figure 2(a) shows the behavior of N uncoupled clocks $\{t_i(n)\}_{i=1}^N$, assuming for simplicity no phase noise. It is apparent that the nodes, if isolated, remain asynchronous.

COUPLED CLOCKS

The goal of a coupling mechanism among the clocks is to drive the latter to synchronicity, possibly within a given tolerance. Before further elaborating on the basic ideas behind (either packet or pulse) clock coupling for analog and discrete time clocks, we formalize the notions of synchronized states, intuitively introduced through the discussion on Figure 2 in the previous section. For analog clocks, we have two conditions of interest:

- *Frequency synchronicity*: for t sufficiently large, there exists a common period of oscillation T for all the nodes so that

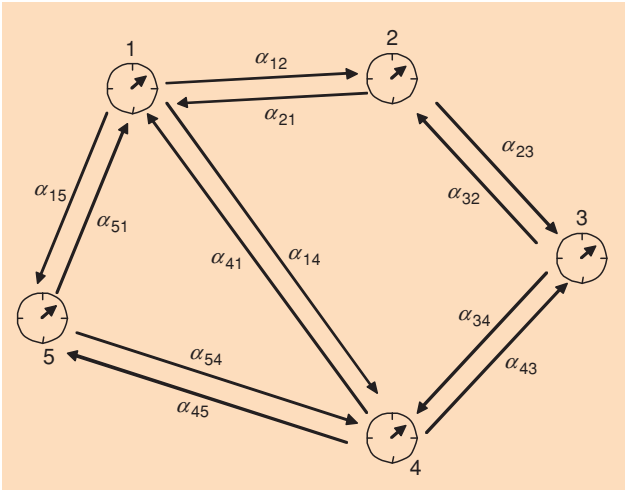
$$\Phi_i(t) = \Phi_i(t+T),\ i = 1, 2, \ldots, N. \tag{4}$$

- *Full (frequency and phase) synchronicity*: for t sufficiently large, we have

$$\Phi_i(t) = \Phi_j(t) \quad \text{for each } i \neq j. \tag{5}$$

Notice that for analog clocks, full synchronicity implies the existence of a common time scale at all times. On the other hand, for discrete time clocks, nodes are said to be synchronous if they agree on the time instants $t_i(n)$ corresponding to the ticks of the local clocks, which entails that a common notion of time does not exist for the period elapsed between two ticks. More specifically, for discrete time clocks, we have the following two conditions:

- *Frequency synchronicity* [Figure 2(b)]: for n sufficiently large, there exists a common period of oscillation T for all the nodes so that



[FIG5] Example of a connectivity graph $\mathcal{G}$: local time information is exchanged along the edges of the graph weighted by the coupling strengths α_{ij}. A key role in the analysis of distributed synchronizaton is played by the Laplacian matrix $\mathsf{L} = \mathsf{I} - \mathsf{A}$, where A is the adjacency matrix of the connectivity graph $\mathcal{G}$ ($[\mathsf{A}]_{ij} = \alpha_{ij}$ for $i \neq j$ and $[\mathsf{A}]_{ii} = 0$).

$$t_i(n+1) - t_i(n) = T, i = 1, \ldots, N. \tag{6}$$

- *Full (frequency and phase) synchronicity* [Figure 2(b)]: for n sufficiently large, we have

$$t_i(n) = t_j(n) \quad \text{for each } i \neq j. \tag{7}$$

In this article, we focus on diffusion protocols for the exchange of local time information. This class encompasses, among the others, the packet-coupling method of [27], pulse coupling (see [18], [25], and [26]), and the synchronization of analog clocks according to the standard Kuramoto model [5]–[7] or the analog PLLs in [9]. Moreover, as illustrated in the section "Distributed Consensus for Multiagent Coordination," diffusion synchronization protocols have strong connections with signal processing applications such as distributed estimation [21], [29], detection [30], and consensus [31] problems. The basic mechanism is as follows. Each node transmits (diffuses) its local time [either phase $\Phi_j(t)$ or clock tick $t_j(n)$] to its neighboring nodes, where the definition of neighbors usually identifies those nodes that receive a sufficiently large power from the sender. We recall that the timing information $t_j(n)$ can be encoded either as a time stamp in a packet (packet coupling) [27] or simply in the transmission time of a given waveform $g(t - t_j(n))$ (pulse coupling) in the case of discrete time clocks. For analog clocks, a signal proportional to the local oscillator $s_j(t)$ needs to be radiated by each node, as discussed in the section "Continuously Coupled Analog Clocks." The goal of each recipient, such as the ith, is to measure the phase or time differences between the local clock and the clocks of neighboring senders [$\Phi_j(t) - \Phi_i(t)$ or $t_j(n) - t_i(n)$, respectively], and to correct the local clock accordingly, despite the nuisance term due to propagation delays.

CONNECTIVITY GRAPH AND LAPLACIAN MATRIX

From the presentation above, it is clear that the achievement of a synchronized state strictly depends on the topology of the connections between clocks, since each node transmits its local time information only to neighbors. The standard way to represent this relationship between nodes is by means of a connectivity graph $\mathcal{G}$, as the one sketched in Figure 5 for $N = 5$. In particular, node i receives the synchronization signal from j (i.e., j is a neighbor of i) if there exists an edge directed from i to j. Moreover, this edge is weighted by a positive value α_{ij}, that represents the relative strength of the signal received by i from j with respect to the other neighbors of i (we have the normalization condition $\sum_j \alpha_{ij} = 1$). For instance, a typical choice for parameters α_{ij} is the following [10], [23]:

$$\alpha_{ij} = \frac{P_{ij}}{\sum_{j \in \mathcal{I}_i} P_{ij}}, \tag{8}$$

where P_{ij} is the power received by the ith node from the jth and $\mathcal{I}_i$ is the set of neighbors of i ($\mathcal{I}_i = \{j : P_{ij} > P_0\}$, with P_0 being a power threshold). Therefore, the edge weight α_{ij} depends on the distance between nodes i and j through path loss attenuation,

and on possibly random factors such as fading and shadowing. Notice that the graph is typically directed ($\alpha_{ij} \neq \alpha_{ji}$) and furthermore, it is bidirectional (i.e., $\alpha_{ij} > 0$ if and only if $\alpha_{ji} > 0$) unless different nodes have different power constraints (so that, given a pair of nodes, one node may be within the transmission radius of the other but not vice versa).

As illustrated in the rest of the article, diffusion synchronization protocols for both analog and discrete cases can be described by linear dynamic systems (see (11) and (17) for a preview) whose system matrix is linearly related to a key algebraic quantity that describes the connectivity graph $\mathcal{G}$, namely the Laplacian matrix $\mathbf{L}$ [32]. This is defined as $\mathbf{L} = \mathbf{I} - \mathbf{A}$, where $\mathbf{A}$ is the adjacency matrix of the graph ($[\mathrm{A}]_{ij} = \alpha_{ij}$ for $i \neq j$ and $[\mathrm{A}]_{ii} = 0$). It is then clear that the performance of mutual synchronization depends on the network topology (connectivity graph $\mathcal{G}$) through the eigenstructure of the Laplacian matrix $\mathbf{L}$. As elaborated in the following (and with some details in "Algebraic Graph Theory and Distributed Synchronization"), of particular relevance is the null space of matrix $\mathbf{L}$, that is sometimes referred to as the synchronization subspace. Specifically, the multiplicity of the zero-eigenvalue $\lambda(\mathbf{L}) = 0$ determines whether a synchronized state is eventually achieved or not, while the left eigenvector $\mathbf{v} = [v_1 \cdots v_N]^T$ corresponding to $\lambda(\mathbf{L}) = 0$ ($\mathbf{v}^T\mathbf{L} = 0$) yields the steady-state frequency and phases of the clocks (see (12), (14), and (19) for a preview).

As a final remark, in the discussion above, we have limited the scope to time-invarying and deterministic topologies, but the analysis can be extended to both time-varying [31], [33] and random [34] topologies. We will provide some comments on these important cases in the following, and we point to references for further details.

REMARK 3

The (average) accuracy of different clocks is sometimes measured in parts-per-million (PPM) by calculating the average (absolute value of) the clock error after one second. There exists a clear trade-off between accuracy and power consumption. For instance, accuracies of typical clocks range between around 10^{-4} and 10^{-11} PPM with corresponding power consumptions on the order of 1μW and hundreds of megawatts, respectively [12].

CONTINUOUSLY COUPLED ANALOG CLOCKS

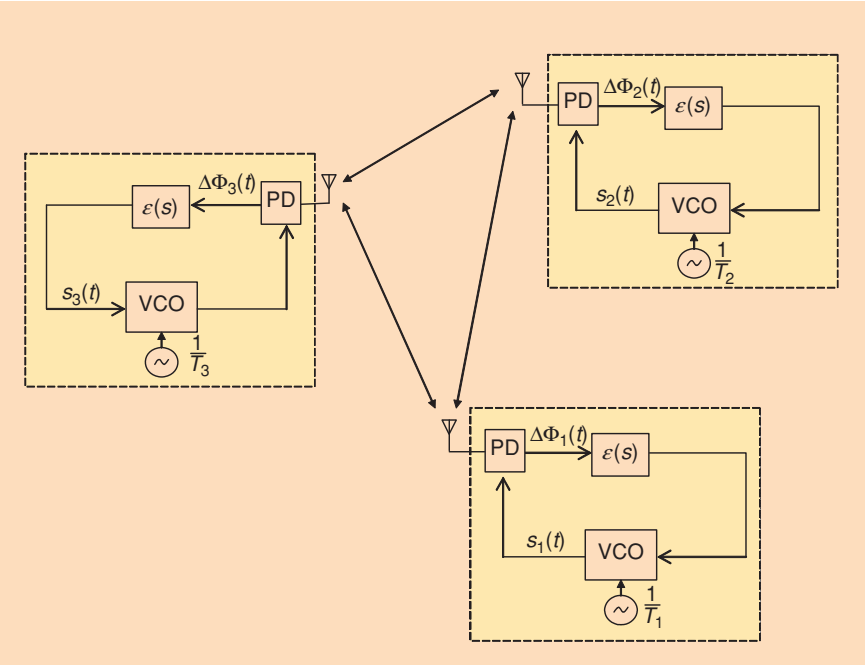
In this section, we study the problem of distributed synchronization of coupled analog clocks. The interest of such problem for wireless communications is related to applications such as, e.g., cooperative beamforming or frequency division multiple access in ad hoc networks. Moreover, it is historically the first studied model of distributed synchronization, and sets the ground for the discussion on discrete-time clocks in the next section.

With coupled analog clocks, each node transmits a signal proportional to its local oscillator $s_i(t)$ in (1) and updates the instantaneous phase $\Phi_i(t)$ based on the signal received from other nodes. Notice that this procedure assumes that each node is able to transmit and receive continuously and at the same time (full duplex, see Remark 2). The basic mechanism of continuously coupled clocks is phase locking (see Figure 6). Each node, say the ith, measures through its phase detector (PD) the convex combination of phase differences

$$\Delta\Phi_i(t) = \sum_{j=1, j\neq i}^{N} \alpha_{ij} \cdot f(\Phi_j(t) - \Phi_i(t)), \tag{9}$$

where $\Phi_j(t) - \Phi_i(t)$ is the phase difference with respect to node j, and $f(\cdot)$ and α_{ij} are phase detector-specific features, namely a nonlinear function and convex combination weights (i.e., $\sum_{j=1}^{N} \alpha_{ij} = 1$ and $\alpha_{ij} \geq 0$), respectively (recall the discussion in the previous section). Notice that the choice of a convex combination in (9) ensures that the output of the phase detector $\Delta\Phi_i(t)$ takes values in the range between the minimum and the maximum of phase differences $f(\Phi_j(t) - \Phi_i(t))$. Finally, the difference $\Delta\Phi_i(t)$ (9) is fed to a loop filter $\varepsilon(s)$, whose output drives the voltage controlled oscillator (VCO), which updates the local phase as

$$\dot{\Phi}_i(t) = \frac{2\pi}{T_i} + \varepsilon_0 \sum_{j=1, j\neq i}^{N} \alpha_{ij} \cdot f(\Phi_j(t) - \Phi_i(t)), \tag{10}$$



[FIG6] Block diagram of $N = 3$ continuously coupled oscillators (PD: phase detector; VCO: voltage controlled oscillator).

where we have considered for simplicity a simple loop filter $\varepsilon(s) = \varepsilon_0$. Moreover, in our presentation above, we have assumed absence of phase noise [see (2)], instantaneous coupling among the clocks (i.e., irrelevant propagation delays: $q_{ij} = 0$) and time-invariant network topology (constant coefficient α_{ij}). These assumptions will be assumed throughout the paper unless otherwise stated (see the section "Signal Processing Aspects of Distributed Time Synchronization" for a discussion on more general and realistic models).

KURAMOTO'S MODEL

The first model of coupled analog oscillators was proposed by Kuramoto in the context of mathematical biology [5] and has attracted considerable attention since its definition because of the many challenges it poses to mathematicians (see the survey papers [6] and [7]). The basic Kuramoto model corresponds to the general system (10) with a sinusoidal phase detector $f(x) = \sin(x)$, all-to-all connectivity, i.e., $\alpha_{ij} = 1/N$ for $i, j = 1, \ldots, N$ (fully meshed connectivity graph), and a simple loop filter $\varepsilon(s) = \varepsilon_0$ (first-order PLLs). The analysis in [5] is concerned with the assessment of the steady state (equilibrium point) of the system. In particular, the author discovered that (assuming unimodal local frequency distribution) there exists a critical value of the loop gain ε_0, say ε_0^*, such that if $\varepsilon_0 > \varepsilon_0^*$ the population of clocks attains a state of partial (frequency and phase) synchronization in which part of the oscillators is phase locked and part is out of synchrony (full synchronization is eventually achieved for $\varepsilon_0 \to \infty$), whereas if $\varepsilon_0 < \varepsilon_0^*$ the clocks remain in an incoherent state. A thorough understanding of the stability properties of the system has proved to be elusive for many years and a few questions are still open, see [6] for a recent review. As final remarks, we refer the reader to [3] for an application of Kuramoto's model to the study of the dynamics of hand clapping in a concert hall.

Kuramoto's model is hardly directly applicable to wireless networks, for two main reasons: i) the assumption of continuous coupling among the clocks, which requires full-duplex transceivers (see discussion above) and ii) the assumption on all-to-all (mesh) connectivity (but see [7] for extensions of the basic Kuramoto model to more general scenarios).

CONTINUOUSLY COUPLED LINEAR PLLs

In the context of synchronization for (wired) telecommunication networks, Lindsey et al. [9] studied the general model of coupled analog clocks (10) for linear phase detectors ($f(x) = x$), arbitrary connections α_{ij} (under the constraint of convexity), and loop filters $\varepsilon(s) = \varepsilon_0/(1 - s/\mu)$ (second-order PLLs). Notice that other types of second-order PLLs that include also a zero in the loop filter (e.g., proportional-plus-integral loop filters) are possible and may bring significant benefits, especially in terms of stability (see [35]). The model in [9] then alleviates the problem ii) mentioned above by allowing arbitrary connectivity, while simplifying the analysis through linearization. Linearization allows to readily use tools from algebraic graph theory in order to study convergence, and makes the analysis flexible enough to enable assessment of the effects of nuisance parameters such as possible communications delays and clock imperfections.

To elaborate on this point, let us focus on the case of continuously coupled linear PLLs with $\varepsilon(s) = \varepsilon_0$ (first-order PLLs). Then, the set of PLLs (10) can be cast as a vector linear time-invariant differential equation

$$\dot{\mathbf{\Phi}}(t) = \boldsymbol{\omega} - \varepsilon_0 \cdot \mathbf{L}\mathbf{\Phi}(t), \tag{11}$$

where we have defined the vectors $\mathbf{\Phi}(t) = [\Phi_1(t) \cdots \Phi_N(t)]^T$, $\boldsymbol{\omega} = [2\pi/T_1 \cdots 2\pi/T_N]^T$ and matrix $\mathbf{L}$ is the graph Laplacian associated with the connectivity graph that describes the network (recall the section "Connectivity Graph and Laplacian Matrix" and Figure 5). As detailed in "Algebraic Graph Theory and Distributed Synchronization," the convergence (synchronization) properties of system (11) depend on the network topology (connectivity graph) through the eigenvalues of the Laplacian matrix $\mathbf{L}$. In particular, a sufficient condition for convergence (asymptotic stability) is that the connectivity graph is strongly connected (i.e., there exists at least one path between every pair of nodes).

Under the condition that (11) is asymptotically stable, [9] finds that the steady-state solution of (11) is characterized by frequency synchronization (4) with common frequency given by the weighted combination

$$\frac{1}{T} = \sum_{i=1}^{N} v_i \frac{1}{T_i}, \tag{12}$$

where vector $\mathbf{v} = [v_1 \cdots v_N]^T$ plays a central role and is the normalized left eigenvector of $\mathbf{L}$ corresponding to the zero eigenvalue: $\mathbf{L}^T\mathbf{v} = \mathbf{0}$ and $\sum_{i=1}^{N} v_i = 1$ (recall the section "Connectivity Graph and Laplacian Matrix"). Even if the formalism of the algebraic graph theory was not employed in [9], reinterpreting the results of [9] in this light allows a unified presentation of diffusion-based synchronization schemes, including synchronization discrete-time clocks (see the section "Pulse-Coupled Discrete-Time Clocks") and distributed estimation/detection and consensus (see the section "Distributed Consensus for Multiagent Coordination"). Notice that the entries of vector $\mathbf{v}$ are real and positive if the graph is strongly connected by virtue of the Perron-Frobenius theorem (see also "Algebraic Graph Theory and Distributed Synchronization"), so that the common frequency $1/T$ (12) is a convex combination of all the local frequencies $\{1/T_i\}_{i=1}^N$. While frequency synchronization is attained, full frequency and phase synchronization (5) is generally not achieved, and the steady-state phases are mismatched by an amount related to the deviations between local and common frequency $\Delta\boldsymbol{\omega} = \boldsymbol{\omega} - 2\pi/T \cdot \mathbf{1}$, where $\mathbf{1} = [1\ 1 \cdots 1]^T$

$$\mathbf{\Phi}(t) \to \mathbf{1} \cdot \frac{2\pi}{T} t + \mathbf{1} \cdot \mathbf{v}^T \left(\mathbf{\Phi}(0) - \frac{\mathbf{L}^\dagger \Delta\boldsymbol{\omega}}{\varepsilon_0} \right) + \frac{\mathbf{L}^\dagger \Delta\boldsymbol{\omega}}{\varepsilon_0}. \tag{13}$$

Notice that the second term in the right hand side of (13) is a phase common to all clocks and the third represents the phase mismatch (see "Algebraic Graph Theory and Distributed Synchronization" for details). As a special case of these results, if no deviation among local frequency exists ($T_i = T_{\text{nom}}$), then from (12) the common frequency is $1/T = 1/T_{\text{nom}}$, and, from (13), full frequency and phase synchronization is achieved with (recall that $\boldsymbol{\Delta\omega} = 0$)

$$\Phi_i(t) \to \frac{2\pi}{T}t + \sum_{j=1}^{N} v_j \Phi_j(0). \tag{14}$$

The results summarized above are extended in [9] to more complex scenarios with loop filters $\varepsilon(s) = \varepsilon_0/(1 - s/\mu)$, delays and phase noise. In particular, similarly to conventional PLLs, it is shown that adding a pole μ in the loop filter $\varepsilon(s)$ (second-order PLLs) reduces the steady-state phase error [see (13)] but, at the same time, reduces the stability margin. These results can be seen as the natural extension of known conclusions in the context of classical (master-slave point-to-point) PLLs [35]. Moreover, propagation delays are shown to cause steady-state phase mismatch. Further discussion on the latter topic is provided in the section "The Impact of Propagation and Processing Delays and Phase Noise" for the case of discrete time PLLs.

PULSE-COUPLED DISCRETE TIME CLOCKS

In this section, we review the two approaches proposed for pulse-coupled discrete time clocks (Figure 3): integrate-and-fire oscillators [8], [25], and distributed discrete time PLLs [10], [23], [24] (see also [27]). As in the previous section, in order to simplify the presentation, we focus on a scenario with absence of phase noise and delays ($q_{ij} = 0$). Moreover, we limit the scope to infinite-resolution time detectors: that is, we assume that each node is able to detect the time of arrival $t_j(n)$ of any pulse received from its neighboring nodes. Clearly, in practice, there exists a trade-off between resolution on one hand, and bandwidth and complexity on the other. More general models with phase noise, delays, and finite-resolution time detectors will be discussed in the section "Signal Processing Aspects of Distributed Time Synchronization."

PULSE-COUPLED INTEGRATE-AND-FIRE OSCILLATORS

This model was first studied in the context of mathematical biology in [8] and then applied by [25] to wireless networks. In order to enable the analysis, it is assumed that no frequency mismatch among different nodes is present ($T_i = T_{\text{nom}}$). The impact of a frequency mismatch has been investigated via numerical simulations in [25]. Moreover, according to the model, each node is equipped with an integrate-and-fire oscillator, as sketched in Figure 7(a). Adapting the notation of [8] to fit our overview, this oscillator is described, when isolated, by a state variable $x_i(t) = g(\Phi_i(t))$, where $g(\cdot)$ is a periodic function (with period 2π) such that in each period it is smooth, monotonically increasing from zero to one, and concave. As before, the ticks $t_i(n)$ of the clock correspond to the time instants when the phase returns, after one period, to 2π, or equivalently when the state variables charges up to its maximum value $x_i(t_i(n)) = 1$ and then returns to zero.

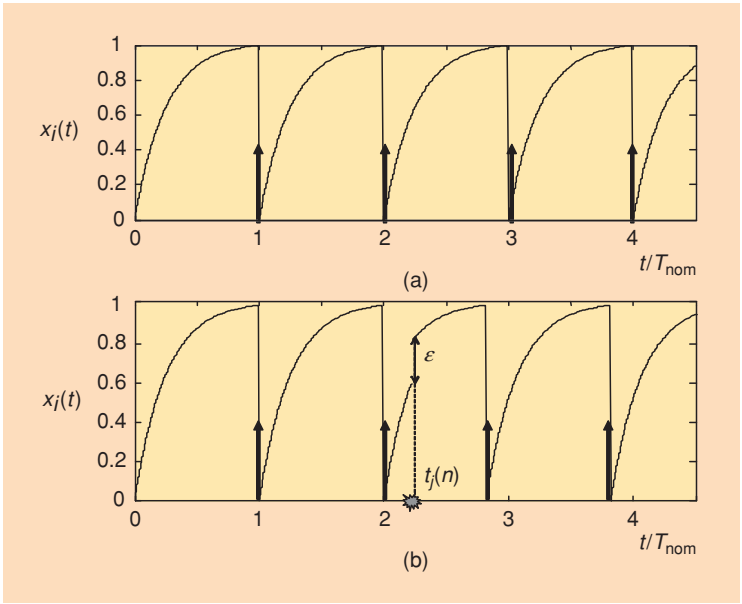
The model of integrate-and-fire oscillators prescribes the following coupling mechanism among clocks, illustrated in Figure 7(b). Upon detection of the pulse sent by any node j at time $t_j(n)$ (propagation delays are neglected in this model), the ith clock modifies the state function by adding a value ε towards the goal of selecting a firing instant that is closer to that of clock j

$$x_i(t_j(n)^+) = \begin{cases} x_i(t_j(n)^-) + \varepsilon & \text{if } x_i(t_j(n)^-) + \varepsilon < 1 \\ 0 & \text{otherwise} \end{cases}$$

and adjusts the phase $\Phi_i(t)$ accordingly.

Convergence of pulse-coupled integrate-and-fire clocks can be evaluated for arbitrary connections α_{ij} by casting the problem as the study of asymptotic stability of a system of differential equations [26]. Using Lyapunov stability theory, convergence is shown to depend on the properties of the graph Laplacian $\mathbf{L}$ (see the section "Coupled Clocks" and "Algebraic Graph Theory and Distributed Synchronization,") similarly to the case of analog oscillators.

The main drawbacks of the model of integrate-and-fire oscillators when applied to wireless networks are: i) it is hard to extend the analysis to realistic and complex scenarios with inaccurate clocks, propagation delays, or time-varying channels; ii) the system design is not flexible enough to grant



[FIG7] Pulse-coupled integrate-and-fire clocks: (a) State function $x_i(t)$ for isolated clocks; (b) State function $x_i(t)$ behavior in presence of a received pulse.

degrees of freedom for the achievement of additional relevant goals, such as trading complexity for accuracy, security, etc.

PULSE-COUPLED DISCRETE-TIME PLLs

An alternative model for pulse-coupled clocks was proposed in [10], [23], and [24] based on distributed discrete-time PLLs. The approach can be seen as the discrete counterpart of the system of coupled analog PLLs illustrated in the section "Continuously Coupled Linear PLLs." From its analog predecessor, the system of pulse-coupled discrete time PLLs inherits the linear nature that enables analysis and flexible system design using standard tools from algebraic graph theory and signal processing (see also the section "Signal Processing Aspects of Distributed Time Synchronization").

A system of pulse-coupled discrete-time PLLs is exemplified by Figure 8. Similarly to the analog case (see Figure 6), based on the received signal, each node calculates a convex combination of the time differences (time difference detector)

$$\Delta t_i(n) = \sum_{j=1, i\neq j}^{N} \alpha_{ij} \cdot (t_j(n) - t_i(n)), \tag{15}$$

that is fed to a loop filter $\varepsilon(z)$. Considering for simplicity loop filters $\varepsilon(z) = \varepsilon_0$ (first-order PLLs), we have

$$t_i(n+1) = t_i(n) + T_i + \varepsilon_0 \cdot \sum_{j=1,\, i\neq j}^{N} \alpha_{ij} \cdot (t_j(n) - t_i(n)). \tag{16}$$

To further analyze the system, let us cast the system (16) as a vector time invariant difference equation

$$\mathbf{t}(n+1) - \mathbf{t}(n) = \mathbf{T} - \varepsilon_0 \mathbf{L} \cdot \mathbf{t}(n), \tag{17}$$

where we defined the vectors $\mathbf{t}(n) = [t_1(n) \cdots t_N(n)]$ and $\mathbf{T} = [T_1 \cdots T_N]^T$ (notice that the case of PLLs with frequency-synchronous clocks studied in [10], [23], and [27] corresponds to (16) with $\mathbf{T} = \mathbf{0}$). From comparison of (17) and (11), it is apparent that the same tools and results derived in the continuous case can be applied to the pulse-coupled discrete time case. In particular, convergence is guaranteed under the same conditions (see "Algebraic Graph Theory and Distributed Synchronization"), and the steady-state solutions are characterized by frequency synchronization (6) with common frequency given by the weighted combination of local frequencies (12), but generally mismatched phases [i.e., absence of full synchrony (7)] with

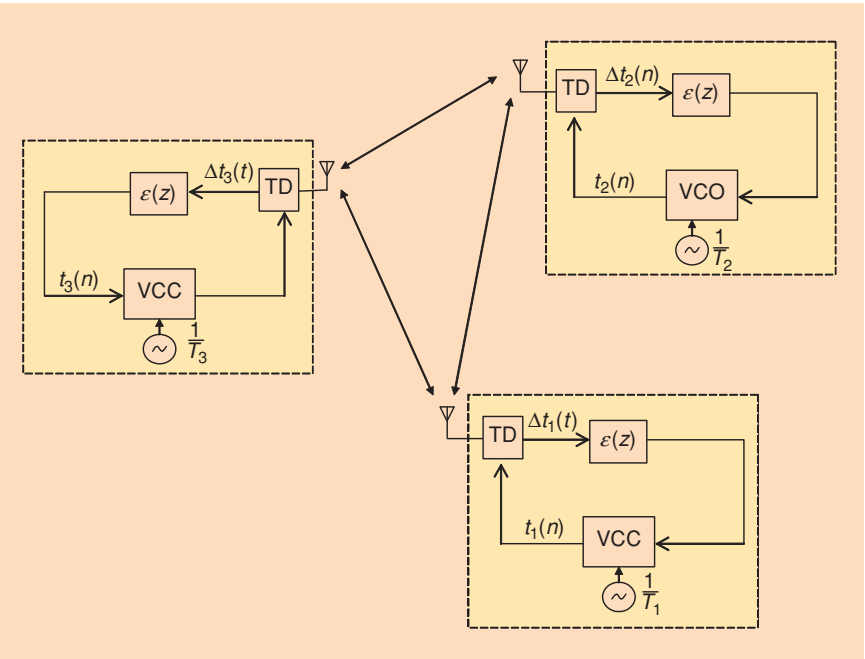
$$\mathbf{t}(n) \to nT \cdot \mathbf{1} + \mathbf{1} \cdot \mathbf{v}^T \left(\mathbf{t}(0) - \frac{\mathbf{L}^\dagger \boldsymbol{\Delta}\mathbf{T}}{\varepsilon_0} \right) + \frac{\mathbf{L}^\dagger \boldsymbol{\Delta}\mathbf{T}}{\varepsilon_0}, \tag{18}$$

where $\boldsymbol{\Delta}\mathbf{T} = \mathbf{T} - T \cdot \mathbf{1}$ [similar to (13)]. As previously discussed, an important special case of these results occurs when there is no frequency mismatch between the clocks ($T_i = T_{\text{nom}}$), in which case the common frequency equals the nominal local frequency $1/T = 1/T_{\text{nom}}$, and, from (18), full frequency and phase synchronization is achieved [similarly to (14)]

$$t_i(n) \to nT + \sum_{j=1}^{N} v_j t_j(0). \tag{19}$$

Moreover, adding a pole μ in the loop filter $\varepsilon(z) = \varepsilon_0/(1 - \mu z^{-1})$ (second-order PLLs) can be shown to reduce the steady-state phase error in (18) (by a factor $1 - \mu$) at the expenses of a reduced stability margin [24], while full synchronization can be in principle achieved with proportional-plus-integral loop filters (see [35]).

Here we consider a simple numerical example for the network of $N = 4$ nodes illustrated in the box of Figure 9. We assume the weights (8) (with $P_0 = 0$), a path loss model $P_{ij} = 1/d_{ij}^3$ (d_{ij} is the distance between node i and j), and frequency synchronous clocks with $T = 1$, and plot the evolution of the phases $t_i(n) - nT$ for $\varepsilon_0 = 0.3$ and $\mu = 0$. After a brief transient where the nodes tend to synchronize in pairs between neighbors, the system reaches the steady state to the condition (19), where $\mathbf{v} = 1/4 \cdot \mathbf{1}$ for this specific topology.



[FIG8] Block diagram of $N = 3$ pulse-coupled discrete time clocks (TD: time difference detector; VCC: voltage controlled clock).

REMARK 4

For both analog (section "Continuously Coupled Analog Clocks") and discrete-time (section "Pulse-Coupled Discrete-Time PLLs") coupled clocks, we have assumed a homogeneous scenario where all the nodes use identical loop filters. Extension of the analysis to a heterogeneous setup calls for substitution of the loop gains $\varepsilon(s)$, and $\varepsilon(z)$, with diagonal matrices containing the local loop filters at the N nodes. This model requires further study.

IMPACT OF TOPOLOGY AND SMALL-WORLD EFFECTS OF SHADOWING

The convergence properties of distributed synchronization depend on the topology of the network, which is in turn defined by the weighting factors α_{ij} (recall the section "Connectivity Graph and Laplacian Matrix" and Figure 5). Here we illustrate the performance of pulse-coupled PLLs for a network of randomly located nodes with weights α_{ij} (8) ($P_0 = 4$) and log-normal shadowing. More specifically, the power received over distance d_{ij} is $P_{ij} = 10^{\frac{\nu}{10}}/d_{ij}^3$, where ν a zero-mean Gaussian random with standard deviation σ. As a performance measure, we evaluate the standard deviation $\xi(n)$ of the clocks, where

$$\xi^2(n) = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(t_i(n) - \frac{1}{N} \sum_{k=1}^{N} t_k(n) \right)^2, \qquad (20)$$

versus time n, averaged over random location of nodes and shadowing. The initial phases $\mathbf{t}(0)$ are selected randomly in the set $(0, 1)$ (and $\mathbf{t}(n) = \mathbf{0}$ for $n < 0$), while the local free-oscillation periods are selected independently in the interval 1 ± 0.01. The dashed lines in Figure 10 are obtained from the asymptotic result (18). It can be seen that increasing the amount of shadowing in the model (i.e., the standard deviation σ) improves both the convergence speed and the asymptotic phase error of the system of distributed PLLs. The beneficial impact of shadowing can be interpreted as an instance of the fact, reported in [2] and [36], that distributed agreement on a graph improves if the graph has the features of a small-world network. A small-world network is characterized by the existence of paths made of a small number of edges between any two nodes. In fact, shadowing breaks a few close connections and, due to the long tails of the log-normal distribution, creates a few long links, thus enhancing the small-world properties of the connectivity graph [2].

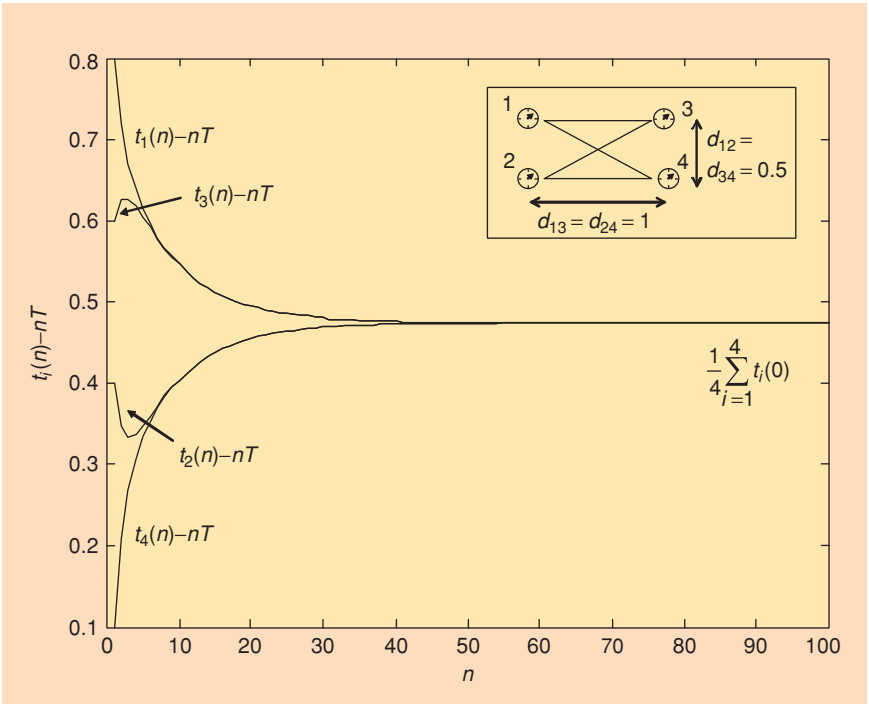
SIGNAL PROCESSING ASPECTS OF DISTRIBUTED TIME SYNCHRONIZATION

In the previous sections, we have shown that basic analysis of distributed synchronization relies on linear algebraic, and more specifically graph algebraic, concepts (namely, on the eigenstructure of the Laplacian matrix associated with the connectivity graph $\mathcal{G}$). In this section, we elaborate on various further aspects of the analysis and design of distributed time synchronization, where signal processing tools play a major role. We focus on pulse-coupled discrete-time PLLs for their practical relevance in wireless networks. At first, we remove the assumption of infinite-resolution time error detectors in the section "Trading Accuracy for Bandwidth and Complexity." Next, we address the issue of fault tolerance and security. Finally, we discuss the impact of propagation and processing delays and phase noise.

TRADING ACCURACY FOR BANDWIDTH AND COMPLEXITY

In this section, we remove the assumption made in the previous section of infinite resolution of the time difference detectors for discrete-time PLLs. The main goal of the section is to illustrate the trade-offs available in the system design between accuracy and complexity of the receiver (which in turns translates into power consumption). To start, we recall that any ith node receives, in an interval of duration T_i around its local clock tick $t_i(n)$ a combination of the waveforms transmitted by other nodes, as sketched in Figure 4. Based on this signal, the node needs to estimate the time differences $t_j(n) - t_i(n)$ in order to correct the local clock [i.e., to decide the next clock tick $t_i(n+1)$] according to the PLL mechanism of Figure 8.

A first approach to evaluate the time differences $t_j(n) - t_i(n)$ would be to perform an estimation of time-of-arrivals $t_j(n)$ (recall that we are neglecting the delays q_{ij} at this



[FIG9] Phases of the $N = 4$ pulse-coupled discrete-time clocks shown in the box versus period n ($T_i = T$, $\varepsilon_0 = 0.3$, $\mu = 0$ and $t(0) = [0.8\ 0.6\ 0.4\ 0]^T$).

stage) based on the knowledge of the transmitted waveform $g(t)$. However, this choice would entail a large computation complexity. A more efficient approach that avoids explicit estimate of the time of arrival is the center of mass time-detector proposed in [23]. Let us assume a square root Nyquist waveforms $g(t)$ with given roll-off. According to the proposal in [23], the receiver performs baseband filtering matched to the transmitted waveform $g(t)$ (or an approximation thereof) and then samples the received signal at some multiple $L \geq 1$ of $1/T_s$, where T_s is the peak-to-first-zero time for the autocorrelation of $g(t)$. Based on the samples (indexed by m) received in the nth observation window, $\{y_i(n, m)\}$, the ith node does not explicitly calculates the single time differences $t_j(n) - t_i(n)$. Instead, it directly estimates the convex combination of time differences $\Delta t_i(n)$ (15), using the definition (8) for the convex weights α_{ij}, as the center of mass of the received signal

$$\Delta t_i(n) = \sum_{m \in \mathcal{I}} \tilde{\alpha}_{im} \cdot \frac{m T_s}{L}, \tag{21}$$

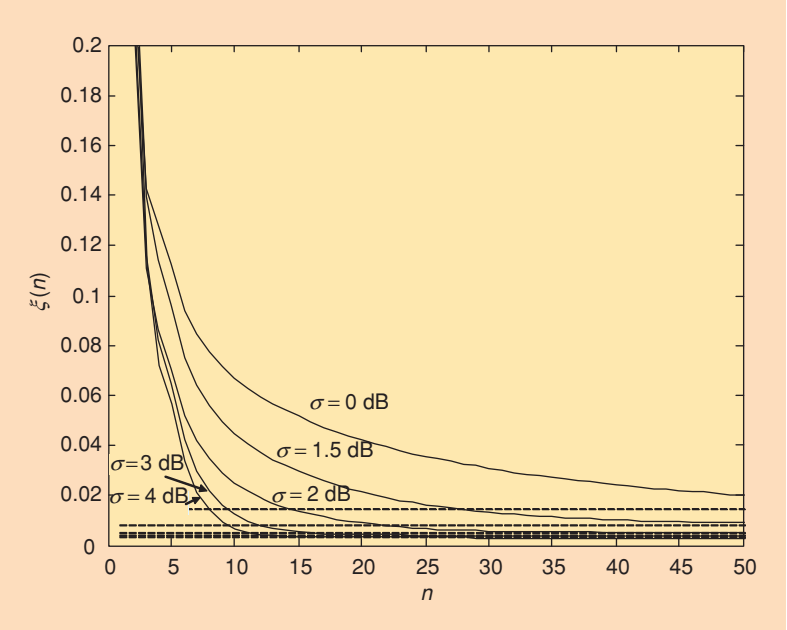
$$\tilde{\alpha}_{im} = \frac{|y_i(n, m)|^2}{\sum_{k \in \mathcal{I}} |y_i(n, k)|^2}, \tag{22}$$

where set $\mathcal{I}$ excludes the samples in the possible refractory period around the firing time $t_i(n)$ due to the half-duplex constraint (recall Remark 2). Notice that this method does not require knowledge of the received powers, and that its complexity (i.e., number of operations) is independent of the number of nodes in the network.

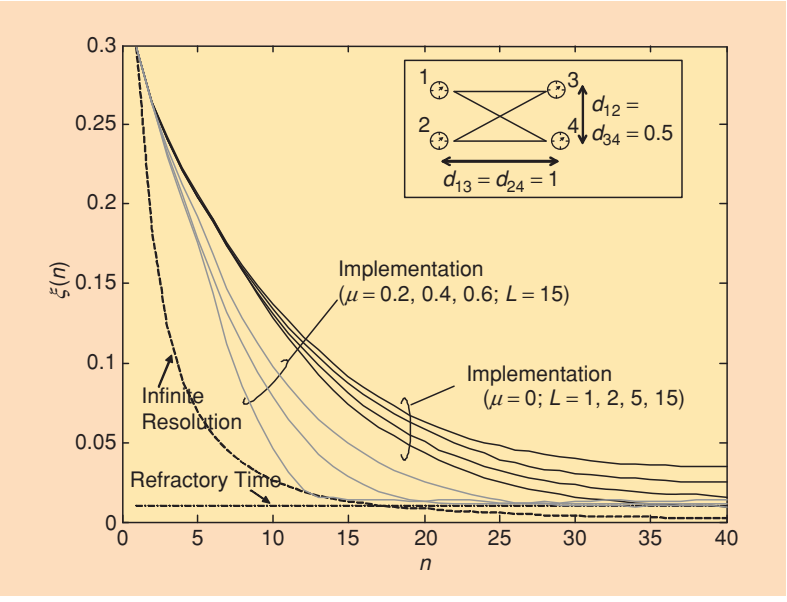
From the discussion above, we have identified three degrees of freedom in the system design for trading accuracy with complexity:

1) the finite switching time from transmit to receive mode that defines the refractory time interval, which depends on the RF hardware employed
2) the oversampling factor L, which can be increased at the expense of computational and hardware complexity at the baseband level
3) the possible presence of a loop filter with pole μ, which increases the number of operations to be performed and thus the computational complexity (see the section "Pulse-Coupled Discrete-Time PLLs").

Here we evaluate the impact of these parameters on the performance (accuracy) of the synchronization scheme. Let us consider the simple network with $N = 4$ nodes shown in the box in Figure 11. Figure 11 shows the standard deviation $\xi(n)$ (20) of the timing vector $\mathbf{t}(n)$ versus n averaged with respect to noise at the receiver side. 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 synchronization accuracy is set by the refractory period. This bound is reached for n and L sufficiently large. Adding a pole in the loop can increase the convergence speed as shown in Figure 11 for $\mu = 0.2,\ 0.4,\ 0.6$. Finally, Figure 11 shows that an upper bound on



[FIG10] Small-world effects of shadowing: standard deviation of the clocks for discrete-time PLLs versus time n for different values of the standard deviation of shadowing σ. Dashed lines correspond to the analytical result (18) ($\varepsilon_0 = 0.6$, $\mu = 0.4$).



[FIG11] Trade-off accuracy versus complexity: Standard deviation of the clocks (20) versus time n for different values of the oversampling factor L and of the pole μ. Also shown for reference is the ideal case of infinite-resolution time difference detectors ($T_i = T$ for $i = 1, \ldots, N$, $\varepsilon_0 = 0.9$, $T_s = 0.01$ and refractory time due to the half-duplex constraint equal to T_s).

the performance of the practical implementation discussed here is set by the performance of the ideal system with infinite resolution studied in the previous sections [see (17)].

FAULT-TOLERANCE AND SECURITY

Distributed wireless networks, such as ad hoc and sensor, are often designed under the assumption that all the nodes in the network are benign and well functioning, and thus comply with the preestablished behavior envisaged by the network designer. However, in many applications, such an assumption is easily too optimistic, and robust design has to account for the possible activity of either faulty or malicious nodes. Here we discuss how simple signal processing techniques can enhance resilience of distributed synchronization to such phenomena.

A simple approach to secure mutual synchronization follows the idea of [37]. In a basic discrete-time PLL, the time difference detector evaluates the convex combination $\Delta t_i(n)$ (15) of the clock errors $t_j(n) - t_i(n)$. Using only this measure, it is not possible for the nodes to recognize outliers that may disrupt the synchronization process. This goal calls for robust approaches that evaluate the dispersion of the clock errors $t_j(n) - t_i(n)$ around the weighted average $\Delta t_i(n)$, by, e.g., computing the variance $\sigma_i^2(n) = \sum_{j=1, j\neq i}^{N} \alpha_{ij} \cdot (t_j(n) - t_i(n) - \Delta t_i(n))^2$, and then update the local clock by considering only the set of clock differences $t_j(n) - t_i(n)$ that are within a given fraction $\beta\sigma_i(n)$ from the average $\Delta t_i(n)$.

An example of the performance of such a scheme is shown in Figure 12. Consider a network of $N = 20$ randomly distributed nodes in a square region of unit area with frequency-synchronous clocks ($T_i = T_{\text{nom}} = T$), out of which four nodes are malfunctioning or malicious, having clocks running as $t_i(n) = nT + \theta_i(n)$, where phases $\theta_i(n)$ are selected independently and uniformly distributed in the set $(0, 1)$. The dashed line corresponds to the performance of the system with no malicious nodes, which, as expected, leads to an asymptotically vanishing error $\xi(n)$. On the contrary, in a scenario with malicious nodes, the timing error of the basic scheme (evaluated only on the sixteen nonmalicious nodes) increases linearly, thus showing that the network is not able to reach even frequency synchronization. However, the secure scheme, for threshold β small enough, manages to maintain a constant error $\xi(n)$ over n, thus showing that it is able to approximately achieve full synchronization within a limited (here 5%) timing error.

THE IMPACT OF PROPAGATION AND PROCESSING DELAYS AND PHASE NOISE

In this section, we remove two further assumptions that have been made throughout the article, namely negligible propagation delays and absence of phase noise (see the section "Clocks and Synchronization"). In order to simplify the analysis, we tackle the two problems separately.

PROPAGATION AND PROCESSING DELAYS

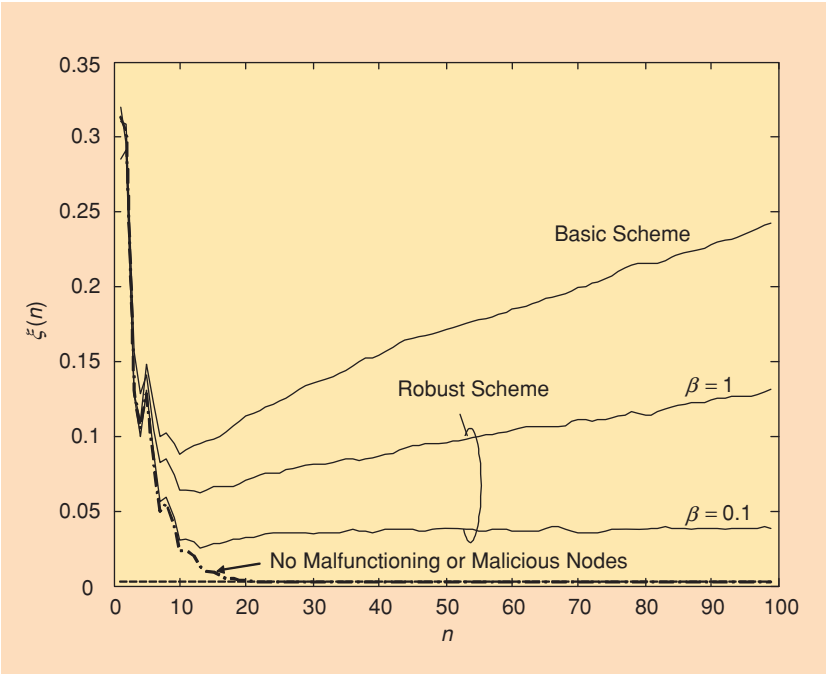
Here we study the impact of delays on the performance of pulse-coupled PLLs but similar conclusions hold also for the continuous case as shown in [9]. Assume (for simplicity) a frequency synchronous network with common local frequency $1/T$ and first-order PLLs. Moreover, let q_{ij} be the (finite) propagation delay between the ith and the jth node (by symmetry, we have $q_{ij} = q_{ji}$). The time at which the nth pulse emitted by node j [at time $t_j(n)$] is recorded by the ith is $t_j(n) + q_{ij}$, so that the timing detection error of the ith PLL

$$\sum_{j=1,\, j\neq i}^{N} \alpha_{ij} \cdot (t_j(n) + q_{ij} - t_i(n)) = \Delta t_i(n) + \sum_{j=1,\, j\neq i}^{N} \alpha_{ij} q_{ij}, \tag{23}$$

contains an additive term to the timing error $\Delta t_i(n)$ defined in (15). Now, introducing the effective local frequency $1/T_i$ as

$$T_i = T + \varepsilon_0 \sum_{j=1, j\neq i}^{N} \alpha_{ij} q_{ij}, \tag{24}$$

it turns out that the model that account for propagation delays boils down to the frequency-asynchronous model (17). In other words, propagation delays have the effect of introducing an equivalent frequency offset between the local clocks,



[FIG12] Security in distributed synchronization: Standard deviation of the clocks (20) versus time n for the basic and secure scheme in case we have four malfunctioning or malicious nodes out of a total number of $N = 20$ nodes ($\varepsilon_0 = 0.6$ and $\mu = 0$).

which eventually leads to the static phase error discussed in the section "Distributed Consensus for Multiagent Coordination" [see (18)], [9]. Precompensation is clearly possible if the propagation delays, or if an estimate of the aggregate measure $\sum_{j=1,\, j\neq i}^{N} \alpha_{ij} q_{ij}$, is available locally (see [10] for a discussion on this point).

PHASE NOISE

Adding phase noise, the basic model (17) reduces to the stochastic difference equation

$$\mathbf{t}(n+1) - \mathbf{t}(n) = \mathbf{T} - \epsilon_0 \mathbf{L} \cdot \mathbf{t}(n) + \boldsymbol{\upsilon}(n), \tag{25}$$

where $\boldsymbol{\upsilon}(n)$ models phase noise, and is assumed here to be a vector random process with independent identically distributed components along the dimensions of nodes and time, with zero mean and given variance. The expectation $E[\mathbf{t}(n)]$ behaves as the deterministic (phase noise-free) model discussed in the section "Pulse-Coupled Discrete-Time PLLs" since it satisfies the difference equation $E[\mathbf{t}(n+1)] - E[\mathbf{t}(n)] = \mathbf{T} - \varepsilon_0 \mathbf{L} \cdot E[\mathbf{t}(n)]$, which coincides with (17). To get further insight, let us assume that the clocks are frequency-synchronous ($\mathbf{T} = T\mathbf{1}$) and that the the Laplacian $\mathbf{L}$ is symmetric, where the latter condition requires a regular topology such as the one considered in Figures 9 and 11. Under these assumptions, model (25) has been investigated, after the change of variable $\tilde{\mathbf{t}}(n) = \mathbf{t}(n) - nT \cdot \mathbf{1}$, in the context of consensus problems in [38] (see next sections for further details on consensus). It is therein proved that, while the clocks $\mathbf{t}(n)$ do not converge in any useful sense, the average of their relative deviation (20) $E[\xi^2(t)]$ indeed converges to a finite steady-state values that is a function of the network topology through the graph Laplacian $\mathbf{L}$.

DISTRIBUTED SYNCHRONIZATION-BASED SIGNAL PROCESSING AND CONTROL APPLICATIONS

The basic concepts and mathematical framework of mutual synchronization have recently found application in the literature on distributed processing and control for networks of multiple wireless nodes, or more generally defined agents. In this section, we briefly review two such applications: distributed consensus and distributed estimation/detection.

DISTRIBUTED CONSENSUS FOR MULTIAGENT COORDINATION

Multiagent systems consist of nodes, such as mobile robots or unmanned air vehicles, that need to coordinate their behavior towards the achievement of some collective goal. This task is accomplished via the exchange of messages along the edges of the connectivity graph $\mathcal{G}$ that describes the links (wireless or of other nature) among the agents. One of the basic problems in multiagent coordination is achieving consensus or agreement on a given quantity, e.g., to yield a common decision. This problem was at first studied in computer science [44] and is now an active area of research in control [31].

Both continuous-time or discrete-time models for message passing among the agents have been considered. Let us denote as vectors $\boldsymbol{\Phi}(t)$ (continuous time) and $\mathbf{t}(n)$ (discrete time) the instantaneous values of the quantity different agents are trying to achieve consensus on. It turns out that the basic models for multiagent consensus coincide with (11) and (17), respectively, having set zero frequencies $\boldsymbol{\omega} = \mathbf{0}$ and $\mathbf{T} = \mathbf{0}$ [31].

We concentrate here on the discrete-time case, since similar results are straightforwardly extended to the continuous-time case. From the discussion above, recalling the section "Pulse-Coupled Discrete-Time PLLs," the basic signal model for multiagent consensus reads

$$\mathbf{t}(n+1) = (\mathbf{I} - \varepsilon_0 \mathbf{L}) \cdot \mathbf{t}(n), \tag{26}$$

and the steady-state solution depends on the initial conditions as $t_i(n) \to \sum_{j=1}^{N} v_j t_j(0)$, where we recall that $\mathbf{v}$ is the normalized left eigenvector of the graph Laplacian $\mathbf{L}$. The common value on which the agents achieve consensus is then a convex combination of the initial values $\mathbf{t}(0)$ with weights given by $\mathbf{v}$ (we are assuming a strictly connected graph, so as to be able to apply Perron-Frobenius theorem). Controlling the weights α_{ij} allows to obtain appropriate desirable convex combination of the initial values, e.g., average or maximum likelihood estimate of a given parameter. This possibility is discussed in the next section.

DISTRIBUTED ESTIMATION/DETECTION IN WIRELESS SENSOR NETWORKS

Consider a wireless sensor networks without a fusion center, whose goal is to achieve a global estimation/detection task through localized processing so that the final estimation/decision value is available to each node. This approach is being strongly advocated for its robustness and ease of implementation. Typical examples of applications are monitoring, tracking, and localization. In many of these cases, a basic operation performed by the network is the calculation of the global (convex) weighted average of some (possibly vector-valued) local measurements [45]. For instance, calculation of distributed plain averages (i.e., with equal weights $\mathbf{v} = \mathbf{1}/N$) is used in [29] in order to perform distributed Kalman filtering and in [30] as a means to achieve distributed hypothesis testing through exchange of belief information. This approach is usually referred to as average consensus. Weighted sums are instead considered in [46] towards to goal of achieving localization through energy measurements and in [21] for distributed maximum-likelihood estimators.

According to our discussion in the previous sections, global averages can be computed through local message passing on the connectivity graph. Moreover, controlling the coefficients α_{ij} enables the choice of the vector of weights $\mathbf{v}$. For instance, it can be easily proved that, in the case of average consensus, vector $\mathbf{v}$ equals the desired $\mathbf{v} = \mathbf{1}/N$ if and only if the graph is balanced, i.e., if we have $\sum_{i=1}^{N} \alpha_{ij} = 1$ for any j [47]. In [47], the remaining degrees of freedom in the choice of coefficients α_{ij} are leveraged to maximize the convergence rate by formulating an appropriate optimization problem. Moreover, two distinct approaches can be followed to obtain global averaging of local measures:

1) Similarly to the consensus problem discussed in the previous section, local measurements are mapped into the initial values ($\Phi_i(0)$ or $t_i(0)$) of local oscillators (11) or (17), respectively, with zero frequencies ($\omega_i = 0$ or $T_i = 0$). The final outcome is given at each node by the steady-state value of the phase of the local oscillator ($\Phi_i(t)$ in (14) or $t_i(n)$ in (19)).
2) Alternatively, one could map the local measurement into the free-running frequencies ω_i or T_i of the local oscillators (11) or (17), using the fact that the frequencies eventually synchronize to the common value (12), which is exactly the desired convex sum [21]. Notice that [21] considers a general nonlinear model as in (10) with $f(\cdot)$ being continuously differentiable, odd, and increasing.

As a final remark, we notice that distributed estimation based on consensus has been studied in [48] as the solution of a decentralized optimization problem through decomposition techniques.

SYNCHRONIZATION OF NONPERIODIC SIGNALS (CHAOS)

Since the early 1990s, there has been evidence that synchronization can be achieved not only among periodic signals (clocks), which is also referred to as limit-cycle oscillator in the jargon of nonlinear dynamics [39], as discussed throughout the article, but also among nonperiodic (chaotic) systems. Envisaged applications to wireless communications are spread spectrum modulation and secure transmission. In this section, we provide a glimpse of this complex subject and refer the reader to [39]–[41] and references therein for further information. We focus on unidirectional (master-slave) synchronization of two chaotic systems, both for its simplicity and because it is the model of interest for the above mentioned applications to wireless communications (mutual synchronization does not seem to have found significant applications to this field so far [40]). Consider, for instance, two Lorenz chaotic systems, indexed by $i = 1, 2$ and described in a three-dimensional phase space by trajectories $\mathbf{x}_i(t) = [x_i^{(1)}(t)\ x_i^{(2)}(t)\ x_i^{(3)}(t)]^T$, satisfying the non-linear system $\dot{\mathbf{x}}_i = \mathbf{f}(x_i^{(1)}, x_i^{(2)}, x_i^{(3)})$ (we drop the temporal dependence for the sake of a simpler notation) [39], [41]:

$$\begin{aligned} \dot{x}_i^{(1)} &= a\left(x_i^{(2)} - x_i^{(1)}\right) \\ \dot{x}_i^{(2)} &= b x_i^{(1)} - x_i^{(2)} - x_i^{(1)} x_i^{(3)} \\ \dot{x}_i^{(3)} &= x_i^{(1)} x_i^{(2)} - b x_i^{(3)}, \end{aligned} \tag{27}$$

where $a, b, c > 0$ are parameters. According to their chaotic behavior, if the two systems $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ are initialized with even slightly different initial conditions, they eventually diverge from one another (even though they still retain the same attractor pattern). However, surprisingly, it is possible to lock the two system in synchrony by appropriate coupling. In particular, exchange one signal from, say, the first system ($x_1^{(1)}$) to the second by setting $x_2^{(1)} = x_1^{(1)}$. The system of resulting five differential equation (six from the Lorenz systems minus the constraint imposed by coupling) can be proved to converge to $x_1^{(2)} = x_2^{(2)}$ and $x_1^{(3)} = x_2^{(3)}$ for any initial condition. Through coupling via transmission of one message from the master (first system) to the slave (the second system), we are then able to obtain synchronization of chaotic systems. Extensions of this basic model to mutual synchronization of multiple chaotic systems coupled through a connectivity graph have been considered, showing that the convergence properties strictly depend on the algebraic graph characteristics of the network, similarly to the cases studied in this article [40].

CONCLUSIONS

This article has explored history, recent advances, and challenges in distributed synchronization for distributed wireless systems. We have focused on synchronization schemes based on exchange of signals at the physical layer and corresponding baseband processing, wherein analysis and design can be performed using known tools from signal processing. Emphasis has also been given on the synergy between distributed synchronization and distributed estimation/detection problems. Finally, we have touched upon synchronization of nonperiodic (chaotic) signals. Overall, we hope to have conveyed the relevance of the subject and to have provided insight on the open issues and available analytical tools that could inspire further research within the signal processing community.

AUTHORS

Osvaldo Simeone (osvaldo.simeone@njit.edu) received his Ph.D. degree in information engineering from Politecnico di Milano, Milan, Italy, in 2005. He is currently with the Center for Wireless Communications and Signal Processing Research, New Jersey Institute of Technology, Newark, New Jersey, where he is an assistant professor. His current research interests concern the cross-layer analysis and design of wireless networks with emphasis on information-theoretic, signal processing and queuing aspects. He is an editor for *IEEE Transactions on Wireless Communication*.

Umberto Spagnolini (spagnoli@elet.polimi.it) received the D.I.E.E. degree (cum laude) from the Politecnico di Milano, Milan, Italy, in 1988. Since 1988, he has been with the Dipartimento di Elettronica e Informazione, Politecnico di Milano, where he is a full professor of telecommunications. His current research interests include the area of statistical signal processing. He was an associate editor for *IEEE Transactions on Geoscience and Remote Sensing* and currently serves as an editor for *IEEE Transactions on Wireless Communications*. He is the cofounder of WiSyTech (Wireless System Technology), a spinoff company of Politecnico di Milano on Software Defined Radio.

Yeheskel Bar-Ness (yeheskel.barness@njit.edu) received the B.Sc and M.Sc degrees in electrical engineering from the Technion, Israel, and Ph.D. degree in applied mathematics from Brown University, Providence, Rhode Island. He is a distinguished professor of ECE, foundation chair of Communication and Signal Processing Research, and executive director of the Center for Wireless Communication and Signal Processing

Research at the New Jersey Institute of Technology, Newark. After working in the private sector, he joined the School of Engineering, Tel-Aviv University in 1973. He came to NJIT from AT&T Bell Laboratories in 1985. Current research interests include design of MIMO-OFDM, and MC-CDMA, adaptive array and spatial interference cancellation and signal separation for

ALGEBRAIC GRAPH THEORY AND DISTRIBUTED SYNCHRONIZATION

Asymptotic stability of mutual synchronization strictly depends on the topology of the graph $\mathcal{G}$ describing the connections among the local oscillators (see Figure 5). The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ is generally weighted and directed, with the N vertices $\mathcal{V}$ corresponding to the nodes of the network and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ describing the coupling between different nodes as defined by the $N \times N$ adjacency matrix $\mathbf{A}$. The Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{A}$ of the graph has the following basic properties:

i) existence of a zero eigenvalue, $\lambda_1(\mathbf{L}) = 0$ (follows from the fact that $\mathbf{L}$ has zero row sums, i.e., $\mathbf{L1} = 0$)

ii) all the eigenvalues $\lambda_k(\mathbf{L})$ satisfy the condition $|\lambda_k(\mathbf{L}) - 1| \leq 1$ (follows from the Gershgorin theorem [49])

iii) if the weights are symmetric, i.e., $\alpha_{ij} = \alpha_{ji}$, matrix $\mathbf{L}$ is symmetric and positive semidefinite so that $\lambda_k(\mathbf{L})$ are real and satisfy $0 \leq \lambda_k(\mathbf{L}) \leq 2$ (in this case, we can order the eigenvaluesas $0 = \lambda_1(\mathbf{L}) \leq \lambda_2(\mathbf{L}) \leq \cdots \leq \lambda_N(\mathbf{L})$).

Conditions for Mutual Synchronization

Here we limit the treatment to continuously coupled or pulse-coupled first-order (linear) PLLs studied in the sections "Continuously Coupled Linear PLLs" [see (11)] and "Pulse-Coupled Discrete-Time PLLs" [see (17)], respectively, referring the reader to appropriate references for more general scenarios, where similar synchronization conditions can be derived. It can be easily shown that for continuously coupled first-order PLLs, synchronization occurs if and only if $\mathrm{Re}\{\lambda_k(\mathbf{L})\} > 0$ for $k \neq 1$, while for pulse-coupled first-order discrete-time PLLs, the condition is $|\lambda_k(\mathbf{L})| > 0$ for $k \neq 1$ (as another example, in the case of nonlinearly coupled oscillators with symmetric weights, as in [21], the loop gain ε_0 necessary to reach convergence is inversely proportional to the largest nonzero eigenvalue $\lambda_2(\mathbf{L})$, and similar conditions can be found for chaotic sysems [40]). Notice that both conditions, given property ii), amount to requiring that the multiplicity of the unitary eigenvalue $\lambda_1(\mathbf{L}) = 0$ is one (simple eigenvalue). Moreover, it can be shown that the exponential rate of convergence of the synchronization depend on the "smallest" (in terms of real part for the continuous case and absolute value for the discrete) nonzero eigenvalue: the "smaller" the eigenvalue the slower the convergence. This particular eigenvalue is generally referred to as algebraic connectivity (see [40] for the exact analytical expression of the algebraic connectivity for some simple network topologies and general bounds).

A first sufficient condition for synchronization follows directly from the well-known Perron-Frobenius theorem [49] and states that, if the graph is strongly connected, i.e., if there exist a (directed) path (possibly composed of multiple edges) between any pair of nodes, then the multiplicity of $\lambda_1(\mathbf{L}) = 0$ is one, and thus synchronization is achieved (see, e.g., [42] and [24]). This conclusion complies with the intuition that synchronization is achieved if every node sees, possibly through the intermediation of other nodes, the time of all the clocks in the network. Notice, however, that a fully meshed configuration in which we have direct exchange of time between all pairs of nodes is not necessary. A second, more recent, result proves that a necessary and sufficient condition for synchronization is the existence of a spanning tree directed tree on the graph obtained from $\mathcal{G}$ by reversing the direction of all the links [31], [43]. This second conclusion shows that strong connectivity is not necessary; what is necessary is that there exists at least one clock that is heard by all the nodes, possibly not directly heard, but through the intermediation of other nodes (see also [51]).

Remark 5

The discussion throughout the article assumes a time-invariant topology, i.e., the coefficient α_{ij} does not depend on time. Distributed synchronization (consensus) in presence of time-varying connections has been studied in the literature following two main approaches. One is to exploit the linearity of the considered model by using known properties of infinite product of matrices [50], while the other leverages nonlinear tools, such as Lyapunov theory, in order to exploit the contractive properties of the models at hand [33].

Steady-State Analysis

Here we present a brief derivation of the steady-state solution for the case of continuously coupled first-order (linear) PLLs (11) treated in the section "Continuously Coupled Analog Clocks." Results for the discrete-time counterpart covered in the section "Pulse-Coupled Discrete-Time PLLs" follow using the same arguments. To start, let us write the phases $\boldsymbol{\Phi}(t)$ of the N clocks in terms of a possible common frequency $1/T$ (to be determined) and of corresponding instantaneous phases $\boldsymbol{\theta}(t)$ as

$$\boldsymbol{\Phi}(t) = \frac{2\pi}{T} t \cdot \mathbf{1} + \boldsymbol{\theta}(t). \tag{28}$$

With this change of variables in (11), we are interested in finding the steady-state equilibrium (if any) of phases $\boldsymbol{\theta}(t)$ say $\boldsymbol{\theta}^*$, and the value of the common frequency $1/T$ (if any). Plugging (28) in (11), we easily get

$$\dot{\boldsymbol{\theta}}(t) = -\varepsilon_0 \mathbf{L} \cdot \boldsymbol{\theta}(t) + \boldsymbol{\Delta\omega}, \tag{29}$$

where we recall that $\boldsymbol{\Delta\omega} = \boldsymbol{\omega} - 2\pi/T \cdot \mathbf{1}$. Solving for the equilibrium point $\boldsymbol{\theta}^*$, we set $\dot{\boldsymbol{\theta}}(t) = 0$, obtaining the condition

$$\mathbf{L}\boldsymbol{\theta}^* = \frac{\boldsymbol{\Delta\omega}}{\varepsilon_0}. \tag{30}$$

Since by definition of vector $\mathbf{v}$ (see the section "Continuously Coupled Analog Clocks"), we have $\mathbf{v}^T \mathbf{L} = \mathbf{0}$, it follows that $\mathbf{v}^T \boldsymbol{\Delta\omega} = \mathbf{0}$, from which the value of the common frequency (12) easily follows. Moreover, through the further change of variables $\boldsymbol{\varphi}(t) = \boldsymbol{\theta}(t) - \mathbf{L}^\dagger \boldsymbol{\Delta\omega}/\varepsilon_0$ in (29), we obtain $\dot{\boldsymbol{\varphi}}(t) = -\varepsilon_0 \mathbf{L} \cdot \boldsymbol{\varphi}(t)$. Under the condition that $\lambda_1(\mathbf{L}) = 0$ is a simple eigenvalue of $\mathbf{L}$ we have that $\boldsymbol{\varphi}(t) \to \mathbf{1}\mathbf{v}^T \boldsymbol{\varphi}(0)$ (see [31]), which finally leads to (13).

multi-user communications, and modulation classification. He published numerous papers in these areas. He is on the editorial board of *WIRED* Magazine, was the founding cditor-in-chief of *IEEE Communication Letters*, and was associate and area editor for *IEEE Transactions on Communications*. He is on the editorial board of the *Journal of Communication Networks*. He has been the technical chair of several major conferences and symposiums and was the recipient of the 1973 Kaplan Prize. He is a Fellow of the IEEE and received the IEEE Communication Society's Exemplary Service Award and was selected the NJ 2006 Inventor of the Year.

Steven H. Strogatz (strogatz@cornell.edu) received a Ph.D. in applied mathematics from Harvard University. He joined the Cornell faculty in 1994, where he is the Jacob Gould Schurman Professor of Applied Mathematics and professor of theoretical and applied mechanics. His research interests include nonlinear dynamics, complex networks, mathematical biology, and synchronization. He is the author of the textbook *Nonlinear Dynamics and Chaos* (Perseus, 1994) and the trade book *Sync* (Hyperion, 2003).

REFERENCES

[1] P. Gallison, *Einstein's Clocks, Poincare's Maps: Empires of Time*. New York: Norton, 2003.

[2] S. Strogatz, *Sync: The Emerging Science of Spontaneous Order*. New York: Hyperion, 2003.

[3] A.L. Barabasi, *Linked: How Everything is Connected to Everything Else and What it Means*. Plume, 2003.

[4] A.T. Winfree, "Biological rhythms and the behavior of populations of coupled oscillators," *J. Theor. Biol.*, vol. 16, pp.15-42, 1967.

[5] Y. Kuramoto, *Chemical Oscillations, Waves and Turbulence*. Berlin: Springer-Verlag, 1984.

[6] S.H. Strogatz, "From Kuramoto to Crawford: Exploring the onset of synchronization in populations of coupled oscillators," *Phys. D*, no. 143, pp. 1–20, 2000.

[7] J.A. Acebron, L.L. Bonilla, C.J. Perez Vincente, F. Ritort, and R. Spigler, "The Kuramoto model: A simple paradigm for synchronization phenomena," *Rev. Mod. Phys.*, vol. 77, pp. 137–185, Jan. 2005.

[8] R.E. Mirollo and S.H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math.*, vol. 50, no. 6, pp. 1645–1662, Dec. 1990.

[9] W.C. Lindsey, F. Ghazvinian, W.C. Hagmann, and K. Desseouky, "Network synchronization," *Proc. IEEE*, vol. 73, no. 10, pp. 1445–1467, Oct. 1985.

[10] F. Tong and Y. Akaiwa, "Theoretical analysis of interbase-station synchronization systems," *IEEE Trans. Commun.*, vol. 45, no. 5, pp. 590–594, 1998.

[11] D. Li, K.D. Wong, Y.H. Hu, and A.M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 17–29, Mar. 2002.

[12] B.M. Sadler, "Fundamentals of energy-constrained sensor network systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 20, no. 8, pp. 17–35, Aug. 2005.

[13] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Networking*, vol. 12, no. 3, pp. 493–506, June 2004.

[14] J.N. Lanemann and G.W. Wornell, "Distributed space-time coded protocols for exploiting cooperative diversity in wireless networks," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2415–2425, Oct. 2003.

[15] R. Mudumbai, J. Hespanha, and U. Madhow, "Scalable feedback control for distributed beamforming in sensor networks," in *Proc. IEEE Int. Symp. Information Theory (ISIT 2005)*, Sept. 2005, pp. 137–141.

[16] F. Sivrikaya and B.F. Yener, "Time synchronization in sensor networks: A survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July–Aug. 2004.

[17] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in *Proc. MILCOM*, Oct. 2003.

[18] A.-S. Hu and S.D. Servetto, "On the scalability of cooperative time synchronization in pulse-connected networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2725–2748, June 2006.

[19] Y. Sung, S. Misra, L. Tong, and A. Ephremides, "Signal processing for application-specific ad hoc networks," *IEEE Signal Processing Mag., (Special Issue on Signal Processing for Wireless Ad hoc Communication Networks),* vol. 23, no. 5, pp. 74–83, Sept. 2006.

[20] Y.-W. Hong and A. Scaglione, "Distributed change detection in large scale sensor networks through the synchronization of pulse-coupled oscillators," in *Proc. ICASSP*, July 2004, pp. III-869–872.

[21] S. Barbarossa and G. Scutari, "Decentralized maximum likelihood estimation for sensor networks composed of nonlinearly coupled dynamical systems," *IEEE Trans. Signal Processing,* vol. 55, no. 7, pp. 3456–3470, July. 2007.

[22] N. Wakamiya and M. Murata, "Scalable and robust scheme for data fusion in sensor networks," in *Proc. Int. Workshop Biologically Inspired Approaches to Advanced Information Technology*, Jan. 2004.

[23] E. Sourour and M. Nakagawa, "Mutual decentralized synchronization for intervehicle communications." *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 2015–2027, Nov. 1999.

[24] O. Simeone and U. Spagnolini, "Distributedsynchronization for wireless sensor networks with coupled discrete-time oscillators," *Eurasip J. Wireless Commun. Networking*, #57054, 2007.

[25] Y.-W. Hong, and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE J. Select. Areas Commun.*, vol. 23, no. 5, pp. 1085–1099, May 2005.

[26] D. Lucarelli and I-J. Wang, "Decentralized synchronization protocols with nearest neighbor communication," in *Proc. ACM SenSys 2004*, Baltimore, MD, Nov. 2004.

[27] Q. Li and D. Rus, "Global clock synchronization in sensor networks," *IEEE Trans. Comput.*, vol. 55, no. 2, pp. 214–226, Feb. 2006.

[28] S. Chen, M.A. Beach, and J.P. McGeehan, "Division-free duplex for wireless applications," *Electronics Lett.*, vol. 34, no. 2, pp. 147–148, Jan. 1998.

[29] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proc. IEEE Conf. Decision and Control*, 2005, pp. 8179–8184.

[30] R. Olfati-Saber, E. Franco, E. Frazzoli, and J.S. Shamma, "Belief consensus and distributed hypothesis testing in sensor networks," *Lecture Notes in Control and Information Sciences*, vol. 331, pp. 169–182, July 2006.

[31] W. Ren, R.W. Beard, and E.M. Atkins, "A survey of consensus problems in multi-agent coordination," in *Proc. American Control Conf.*, vol. 3, June 2005, pp. 1859–1864.

[32] C. Godsil and G. Royle, *Algebraic Graph Theory*. New York: Springer-Verlag, 2001.

[33] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Trans. Automat. Contr.*, vol. 5, no. 2, pp. 169–181, Feb. 2005.

[34] A. Tahbaz Salehi and A. Jadbabaie, "A necessary and sufficient condition for consensus over random networks," *IEEE Trans. Automat. Contr.*, vol. 53, no. 3, pp. 791–795, Apr. 2008.

[35] F.M. Gardner, *Phaselock Techniques*. New York: Wiley, 1966.

[36] X.F. Wang and G. Chen, "Complex networks: Small-world, scale-free and beyond," *IEEE Circuits Syst. Mag.*, vol. 3, no. 1, pp. 6–20, 2003.

[37] L. Lamport and P.M. Melliar-Smith, "Synchronization in the presence of faults," *J. Assoc. Computing Machinery*, vol. 32, no. 1, pp. 52–78, Jan. 1985.

[38] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distributed Comput.,* vol. 67, no. 1, pp. 33–46, Jan. 2007.

[39] S.H. Strogatz, *Nonlinear Dynamics and Chaos*. Boulder, CO: Westview, 2000.

[40] C.W. Wu, *Synchronization in Coupled Chaotic Circuits and Systems*. Singapore: World Scientific, 2002.

[41] L.M. Pecora, T.L. Carroll, G.A. Johnson, D.J. Mar, and J.F. Heagy, "Fundamentals of synchronization in chaotic systems, concepts, and applications," *Chaos: An Interdisciplinary J. Nonlinear Sci.*, vol. 7, no. 4, pp. 520–543, Dec. 1997.

[42] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Automat. Contr.*, vol. 49, no. 9, pp. 1520–1533, Sept. 2004.

[43] C.W. Wu, "On Rayleigh-Ritz ratios of a generalized Laplacian matrix of directed graphs," *Linear Algebra and Its Applicat.*, vol. 402, pp. 207–227, 2005.

[44] D. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[45] D.S. Scherber, H.C. Papadopoulos, "Distributed computation of averages over ad hoc networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 4, pp. 776–787, Apr. 2005.

[46] M. Rabbat, R. Nowak, and J. Bucklew, "Robust decentralized source localization via averaging," in *Proc. ICASSP*, vol. 5, 2005, pp. 1057–1060.

[47] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proc. 42nd IEEE Conf. Decision and Control*, Dec. 2003, pp. 4997–5002.

[48] I. Schizas, A. Ribeiro, and G.B. Giannakis, "Consensus in ad hoc WSNs with noisy links—Part I: Distributed estimation of deterministic signals," *IEEE Trans. Signal Processing,* vol. 56, no. 1, pp. 350–364, Jan. 2008.

[49] S.U. Pillai, T. Suel, and S. Cha, "The Perron-Frobenius theorem," *IEEE Signal Processing Mag.*, vol. 22, pp. 62–75, Mar. 2005.

[50] J. Wolfowitz, "Product of indecomposable, aperiodic, stochastic matrices," *Proc. American Math. Soc.*, vol. 14, no. 5, pp. 733–736, Oct. 1963.

[51] O. Simeone, U. Spagnolini, G. Scutari, and Y. Bar-Ness, "Physical-layer distributed synchronization in wireless networks and applications," *Physical Commun.*, vol. 1, no. 1, pp. 67–83, Mar. 2008.

SP