Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2009, Article ID 623537, 12 pages
doi:10.1155/2009/623537

Research Article

Discrete-Time Second-Order Distributed Consensus Time Synchronization Algorithm for Wireless Sensor Networks

Gang Xiong and Shalinee Kishore

Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA

Correspondence should be addressed to Shalinee Kishore, skishore@lehigh.edu

Received 14 April 2008; Accepted 7 September 2008

Recommended by Erchin Serpedin

This paper proposes a novel discrete-time *second-order distributed consensus time synchronization* (SO-DCTS) algorithm for wireless sensor networks. The consensus properties and convergence rates of the SO-DCTS algorithm are analyzed for both directed and undirected networks. Additionally, the convergence region and optimal convergence rate of the SO-DCTS algorithm are determined for undirected networks and this convergence rate is shown to be superior to that of the *first-order* DCTS (FO-DCTS) algorithm under careful algorithm design. Furthermore, the asymptotic expectation and mean square synchronization error are investigated for the SO-DCTS algorithm when there is Gaussian delay between network nodes. Finally, simulation results are provided to verify these analytical results.

1. Introduction

Wireless sensor networks are typically comprised of inexpensive, small-sized, power-limited terminals. In a variety of applications, these sensor nodes are collectively required to maintain accurate time synchronization, for example, moving object acquisition and tracking, habitat monitoring, reconnaissance and surveillance, environmental monitoring, traffic control, and so forth [1]. This necessitates network algorithms that achieve and maintain *global time synchronization* at all network nodes, that is, algorithms that align all network nodes to a common notion of time.

Due to imperfections in low-cost hardware nodes and the decentralized nature of wireless sensor networks, global time synchronization has been recognized as a particularly challenging task. Conventional synchronization protocols such as time-synchronization protocol for sensor networks (TPSNs) [2], reference broadcast synchronization (RBS) [3], and flooding time synchronization protocol (FTSP) [4] aim to perform *centralized* global synchronization for all nodes in wireless sensor network [5]. These protocols achieve synchronization via time-stamped packet exchanges with a root node or a data fusion center and are thus vulnerable to failure of these central nodes.

Recently, several distributed time synchronization algorithms have been proposed. One important class of such algorithms is referred to as *distributed consensus time synchronization* (DCTS) [6]. In the DCTS approach, a global time consensus can be sufficiently reached within a connected network by averaging pair-wise local time information at network nodes. In [7], Olfati-Saber et al, established a theoretical framework for the analysis of consensus synchronization algorithms. Later, a fully distributed, asynchronous DCTS algorithm was proposed in [8]; this scheme was designed to reach agreement on time offset and skew offset between network nodes using media access control (MAC) layer time-stamped packet exchanges. As an alternative, a physical layer-based DCTS algorithm was introduced in [9] by modeling sensor nodes as coupled discrete-time oscillators. In particular, the algorithm adopts instantaneous received powers as weighted coefficients when updating local clocks.

To the best of our knowledge, existing literature on DCTS methods assumes that local timing update at each node is done using only current timing information, that is, via a *first-order* DCTS (FO-DCTS) approach. In contrast, a *second-order* DCTS (SO-DCTS) algorithm would utilize not only current timing information but also that available from the previous iteration of the algorithm to update local clocks. Such an extension to the basic consensus algorithm was first reported for a continuous time approach in [10]. Subsequent papers have analyzed this second-order continuous time consensus method assuming fixed network topologies [11], time delay [12], and switching topologies [13]. In this paper, we apply the principles of the second-order consensus approach to the distributed timing synchronization problem in wireless sensor networks. Specifically, we propose a novel *discrete-time* SO-DCTS algorithm for wireless sensor networks and examine its convergence properties.

The major contribution of this paper is the theoretical analysis of the convergence characteristics of the SO-DCTS algorithm for both directed and undirected networks. Moreover, we investigate the convergence region and optimal convergence rate of the SO-DCTS algorithm in undirected networks and claim that the optimal convergence rate of the SO-DCTS algorithm is superior to that of the FO-DCTS algorithm under an appropriate algorithm design. Finally, we present the asymptotic expectation and mean square synchronization error of the SO-DCTS algorithm when the timing offset between network nodes is Gaussian distributed.

This paper is outlined as follows. Section 2 describes the system model and background for the proposed SO-DCTS algorithm. Section 3 presents the convergence properties of the SO-DCTS algorithm in directed and undirected networks. Section 4 discusses the convergence region and optimal convergence rate of the SO-DCTS algorithm in undirected networks. In Section 5, we present some convergence results for the SO-DCTS method when network nodes have Gaussian delay between each other. Simulation results are presented in Section 6, and we conclude our discussion in Section 7.

2. Background and System Model

2.1. Proposed SO-DCTS Algorithm. Timing information between network nodes can be exchanged either using time-stamped packets at the MAC layer or by estimating arrival times of neighboring nodes' physical layer pulse signals. In the following, we describe the SO-DCTS method regardless of whether it is implemented at the MAC or physical layers. In each iteration of the SO-DCTS algorithm, each node processes and decodes the time-stamped message from its neighbors or estimates the arrival time of its neighbors' pulse signals. Each node then updates its local clock using the weighted average of the current time differences between itself and its neighboring nodes as well as stored time differences from the previous iteration of the algorithm. It should be noted that in the SO-DCTS algorithm, each node needs to store time information from its neighbor nodes for both the previous and current iterations; this is in contrast to the FO-DCTS approach where only the current time information is processed in the current iteration.

The timing update rule of the SO-DCTS algorithm at each node i is given as

$$\begin{aligned} t_i(k) = t_i(k-1) &+ \varepsilon \sum_{j\in\mathcal{N}_i} \left[t_j(k-1) - t_i(k-1)\right] \\ &- \gamma\varepsilon \sum_{j\in\mathcal{N}_i} \left[t_j(k-2) - t_i(k-2)\right], \end{aligned} \tag{1}$$

where $t_i(k)$ is the local time at node i during iteration k; $\mathcal{N}_i$ is the set of neighboring nodes that can communicate reliably with node i; ε is a constant step size; γ is a constant for each iteration. We assume that initial conditions of the SO-DCTS algorithm are $t_i(-1) = t_i(0) = z_i$, where z_i is initial time offset for node i. It is worth mentioning that when $\gamma = 0$, the SO-DCTS algorithm reduces to the FO-DCTS algorithm.

2.2. Network Model and Some Preliminaries. In the following, we model a wireless sensor network as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a set of n nodes $\mathcal{V} = \{1, 2, \dots, n\}$ and a set of edges $\mathcal{E}$. Each edge is denoted as $e = (i, j) \in \mathcal{E}$ where $i \in \mathcal{V}$ and $j \in \mathcal{V}$ are head and tail of the edge e, respectively. In a wireless sensor network, the presence of an edge (i, j) indicates that node i can communicate with node j reliably. We assume here a connected graph; that is, there exists a directed path connecting any pair of distinct nodes in the network. Throughout our discussion, we assume a time-invariant and connected network unless otherwise stated.

Given this network model, we denote $A = [a_{ij}]$ as the adjacency matrix of $\mathcal{G}$ such that

$$a_{ij} = \begin{cases} 1, & (i, j) \in \mathcal{E}, \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

Then, the in-degree and out-degree of a node i (denoted as c_i and d_i, resp.,) are given as $c_i = \sum_{j=1}^{n} a_{ji}$, and $d_i = \sum_{j=1}^{n} a_{ij}$. Specifically, d_i is also equal to the number of neighbors of node i from which it can receive information reliably, that is, $d_i = |\mathcal{N}_i|$.

Next, we let L be the graph Laplacian matrix of $\mathcal{G}$ which is defined as $L = D - A$, where $D = \operatorname{diag}(d_1, d_2, \dots, d_n)$ is the degree matrix of $\mathcal{G}$. Given this matrix L, we can show that $L\mathbf{1} = \mathbf{0}$, where $\mathbf{1} = [1, 1, \dots, 1]^{\mathrm{T}}$, and $\mathbf{0} = [0, 0, \dots, 0]^{\mathrm{T}}$. In particular, for a connected graph, the rank of L is $n - 1$. Furthermore, for a balanced directed network, the in-degree and out-degree of a node are equal, that is, $c_i = d_i$, thus we see that $\mathbf{1}^{\mathrm{T}}L = \mathbf{0}^{\mathrm{T}}$.

For an undirected network, the presence of an edge (i, j) indicates that nodes i and j can communicate with each other reliably. Under this condition, we can also show that $\mathbf{1}^{\mathrm{T}}L = \mathbf{0}^{\mathrm{T}}$. Additionally, in this case, L is a symmetric positive semidefinite matrix (implying that its eigenvalues are nonnegative), and its eigenvalues can be arranged in increasing order as $0 = \lambda_1(L) < \lambda_2(L) \le \cdots \le \lambda_n(L)$ [14].

Let us define $\vec{\mathbf{t}}(k) = [t_1(k), t_2(k), \dots, t_n(k)]^{\mathrm{T}}$. The evolution of the SO-DCTS algorithm in (1) can be written as

$$\vec{\mathbf{t}}(k) = (I_n - \varepsilon L)\vec{\mathbf{t}}(k-1) + \gamma\varepsilon L\vec{\mathbf{t}}(k-2), \tag{3}$$

with the initial conditions $\vec{\mathbf{t}}(-1) = \vec{\mathbf{t}}(0) = \vec{\mathbf{z}}$, where $\vec{\mathbf{z}} = [z_1, z_2, \ldots, z_n]^{\mathrm{T}}$. Here, I_n denotes the $n \times n$ identity matrix.

3. Convergence Properties of the SO-DCTS Algorithm

In this section, we investigate the consensus properties of the SO-DCTS algorithm in directed and undirected networks. Additionally, we discuss the convergence rate of the SO-DCTS algorithm in such networks.

3.1. Consensus Analysis of the SO-DCTS Algorithm. The main result regarding the average consensus property of the SO-DCTS algorithm in directed networks is stated in the following theorem.

Theorem 1. *For a time-invariant, connected, directed network, consider the SO-DCTS algorithm,*

$$\vec{\mathbf{t}}(k) = (I_n - \varepsilon L)\vec{\mathbf{t}}(k-1) + \gamma\varepsilon L\vec{\mathbf{t}}(k-2), \tag{4}$$

with initial conditions $\vec{\mathbf{t}}(-1) = \vec{\mathbf{t}}(0) = \vec{\mathbf{z}}$. Define

$$H = \begin{bmatrix} I_n - \varepsilon L & \gamma\varepsilon L \\ I_n & \mathbf{0}_{n\times n} \end{bmatrix}, \qquad J = \begin{bmatrix} K & \mathbf{0}_{n\times n} \\ K & \mathbf{0}_{n\times n} \end{bmatrix}, \tag{5}$$

where $K = \mathbf{1}\vec{\beta}^{\mathrm{T}}/(\vec{\beta}^{\mathrm{T}}\mathbf{1})$ and $\vec{\beta}$ is the left eigenvector of L associated with $\lambda_1(L) = 0$. When $\rho(H - J) < 1$, a global consensus is achieved asymptotically, or equivalently,

$$\lim_{k\to\infty} t_i(k) = \frac{\vec{\beta}^{\mathrm{T}}\vec{\mathbf{z}}}{\vec{\beta}^{\mathrm{T}}\mathbf{1}}, \quad \forall i \in \mathcal{V}, \tag{6}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix.

Proof. The proof of this theorem is similar to [11, 15]. Here, we present a sketch proof. Let us define $\vec{\psi}(k) = [\vec{\mathbf{t}}(k)^{\mathrm{T}} \;\; \vec{\mathbf{t}}(k-1)^{\mathrm{T}}]^{\mathrm{T}}$. Then, the SO-DCTS algorithm in (4) can be rewritten as

$$\vec{\psi}(k) = H\vec{\psi}(k-1), \tag{7}$$

which implies $\vec{\psi}(k) = H^k\vec{\psi}(0)$. To calculate the eigenvalues of H, we have [16]

$$\begin{aligned} \mathbf{det}(H - \lambda I_{2n}) &= \mathbf{det}(\lambda^2 I_n + (\varepsilon L - I_n)\lambda - \gamma\varepsilon L) \\ &= \prod_{i=1}^{n}\left[\lambda^2 + (\varepsilon\lambda_i(L) - 1)\lambda - \gamma\varepsilon\lambda_i(L)\right] \\ &= 0. \end{aligned} \tag{8}$$

Thus, the eigenvalues of H are

$$\lambda_k(H) = \frac{1}{2}\left[1 - \varepsilon\lambda_i(L) \pm \sqrt{(1 - \varepsilon\lambda_i(L))^2 + 4\gamma\varepsilon\lambda_i(L)}\right]. \tag{9}$$

For a time-invariant, connected, directed network, L has only one eigenvalue $\lambda_1(L) = 0$. Then, we know that H has two eigenvalues $\lambda_1(H) = 1$ and $\lambda_2(H) = 0$. Additionally, the eigenvalues of $H - J$ agree with those of H except that $\lambda_1(H) = 1$ is replaced by $\lambda_1(H - J) = 0$ [16]. Since $\rho(H - J) < 1$, we see that the eigenvalues of H stay inside the unit circle except for $\lambda_1(H) = 1$. Thus, we have

$$\begin{aligned} \lim_{k\to\infty} H^k &= V \lim_{k\to\infty} \begin{bmatrix} 1 & \mathbf{0}_{1\times(2n-1)} \\ \mathbf{0}_{(2n-1)\times 1} & \Lambda^k \end{bmatrix} V^{-1} \\ &= V \begin{bmatrix} 1 & \mathbf{0}_{1\times(2n-1)} \\ \mathbf{0}_{(2n-1)\times 1} & \mathbf{0}_{(2n-1)\times(2n-1)} \end{bmatrix} V^{-1} \\ &= \vec{w}_r \vec{w}_l^{\mathrm{T}}, \end{aligned} \tag{10}$$

where Λ is the Jordan form matrix corresponding to eigenvalues $\lambda_i(H) \neq 1$ [16], $\vec{w}_l$ and $\vec{w}_r$ are left and right eigenvectors of H corresponding to $\lambda_1(H) = 1$, respectively, and $\vec{w}_r^{\mathrm{T}}\vec{w}_l = 1$. In particular, $\vec{w}_l = (1/\vec{\beta}^{\mathrm{T}}\mathbf{1})[\vec{\beta}^{\mathrm{T}} \;\; \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}}$ and $\vec{w}_r = [\mathbf{1}^{\mathrm{T}} \;\; \mathbf{1}^{\mathrm{T}}]^{\mathrm{T}}$. Plugging $\vec{w}_l$ and $\vec{w}_r$ into (10) and considering the SO-DCTS algorithm in (7), we have

$$\lim_{k\to\infty} \vec{\psi}(k) = \frac{1}{\vec{\beta}^{\mathrm{T}}\mathbf{1}} \begin{bmatrix} \mathbf{1}\vec{\beta}^{\mathrm{T}} & \mathbf{0}_{n\times n} \\ \mathbf{1}\vec{\beta}^{\mathrm{T}} & \mathbf{0}_{n\times n} \end{bmatrix} \begin{bmatrix} \vec{\mathbf{t}}(0) \\ \vec{\mathbf{t}}(-1) \end{bmatrix}, \tag{11}$$

which indicates that

$$\lim_{k\to\infty} t_i(k) = \frac{\vec{\beta}^{\mathrm{T}}\vec{\mathbf{z}}}{\vec{\beta}^{\mathrm{T}}\mathbf{1}}. \tag{12}$$

This completes the proof. □

According to Theorem 1, we see that in general, although average consensus is not achieved for directed networks, all nodes in the network can still reach a global agreement. By "average consensus" we mean that all nodes converge to the same timing which is determined by the average of the initial timing differences between the nodes. However, when the SO-DCTS algorithm is employed in either an undirected network or a *balanced* directed network, average consensus can be achieved asymptotically. We show this via the following theorem.

Theorem 2. *Consider the SO-DCTS algorithm in* (4) *in a time-invariant, connected, directed balanced network or a time-invariant, connected, undirected network, with initial conditions $\vec{\mathbf{t}}(-1) = \vec{\mathbf{t}}(0) = \vec{\mathbf{z}}$. When $\rho(H - J) < 1$, an average consensus is achieved asymptotically, or equivalently,*

$$\lim_{k\to\infty} t_i(k) = \frac{1}{n}\mathbf{1}^{\mathrm{T}}\vec{\mathbf{z}}, \quad \forall i \in \mathcal{V}. \tag{13}$$

We know that in a time-invariant, connected, directed balanced or undirected network, $\vec{\beta} = \mathbf{1}$ and $K = (1/n)\mathbf{1}\mathbf{1}^{\mathrm{T}}$. The rest of proof is similar to that of Theorem 1 and thus omitted here.

3.2. Convergence Rate for SO-DCTS Algorithm. One of the most important measures of any distributed iterative algorithm is its convergence speed. As we show next, the convergence speed of the SO-DCTS algorithm is determined by the spectral radius of $H - J$, which is similar to the FO-DCTS algorithm [17].

Let us define the global consensus value in each iteration as $m(k) = (1/\vec{\beta}^{\mathrm{T}}\mathbf{1})\vec{\beta}^{\mathrm{T}}\vec{\mathbf{t}}(k)$. In the SO-DCTS algorithm, this value remains invariant during each iteration since

$$\begin{aligned} m(k) &= (1/\vec{\beta}^{\mathrm{T}}\mathbf{1})\vec{\beta}^{\mathrm{T}}\left[(I_n - \varepsilon L)\vec{\mathbf{t}}(k-1) + \gamma\varepsilon L\vec{\mathbf{t}}(k-2)\right] \\ &= m(k-1) = \cdots = m(0). \end{aligned} \tag{14}$$

We now define the disagreement vector as $\vec{\delta}(k) = \vec{\mathbf{t}}(k) - m(k)\mathbf{1}$, which indicates the difference between the updated times and the global consensus times of the network nodes. Then, the evolution of the disagreement vector is obtained as

$$\vec{\delta}(k) = (I_n - \varepsilon L)\vec{\delta}(k-1) + \gamma\varepsilon L\vec{\delta}(k-2). \tag{15}$$

Given this dynamic of the disagreement vector, we note the following Lemma.

Lemma 1. *For the SO-DCTS algorithm in* (4) *in a time-invariant, connected network with initial conditions* $\vec{\mathbf{t}}(-1) = \vec{\mathbf{t}}(0) = \vec{\mathbf{z}}$ *and* $\alpha = \rho(H - J) < 1$, *a global consensus is exponentially reached in the following form:*

$$\frac{\|\vec{\delta}(k)\|^2 + \|\vec{\delta}(k-1)\|^2}{\|\vec{\delta}(0)\|^2} \le 2\alpha^{2k}, \tag{16}$$

where $\|\cdot\|$ *denotes the* ℓ_2 *norm of a vector.*

Proof. Let us define the error vector as $\vec{e}(k) = [\vec{\delta}^{\mathrm{T}}(k)\ \vec{\delta}^{\mathrm{T}}(k-1)]^{\mathrm{T}}$ which can be obtained from $\vec{e}(k) = \vec{\psi}(k) - J_1\vec{\psi}(k)$, where

$$J_1 = \begin{bmatrix} K & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & K \end{bmatrix}. \tag{17}$$

Based on this definition, we see that the error vector results in the following evolution:

$$\begin{aligned} \vec{e}(k) &= (H - J_1 H)\vec{\psi}(k-1) \\ &= (H - J)\left[\vec{\psi}(k-1) - J_1\vec{\psi}(k-1)\right] \\ &= (H - J)\vec{e}(k-1). \end{aligned} \tag{18}$$

The above equation is valid because $(H - J)J_1 = \mathbf{0}_{2n\times 2n}$ and $J_1 H = J$. Then, we have

$$\begin{aligned} \|\vec{e}(k)\|^2 &= \|(H - J)\vec{e}(k-1)\|^2 \le \alpha^2\|\vec{e}(k-1)\|^2 \\ &\le \cdots \le \alpha^{2k}\|\vec{e}(0)\|^2, \end{aligned} \tag{19}$$

which is equivalent to (16). This completes the proof. □

Therefore, we see that the convergence rate for the SO-DCTS algorithm in both directed and undirected networks is determined by the spectral radius of $H - J$.

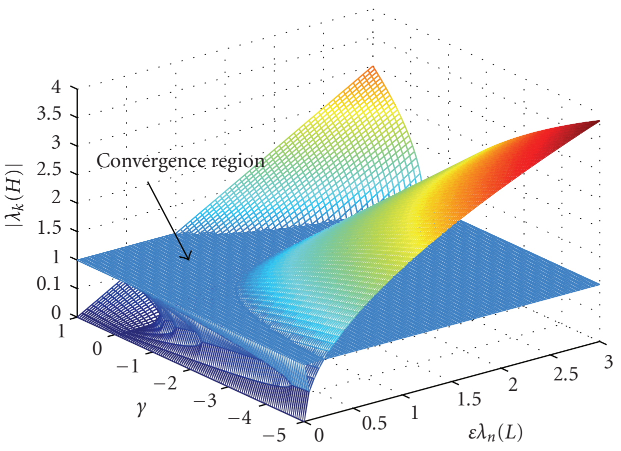

Figure 1: Convergence region for the SO-DCTS algorithm in undirected networks: three-dimensional view.

4. Convergence Region and Optimal Convergence Rate for SO-DCTS Algorithm in Undirected Networks

In this section, we investigate more specific convergence results (i.e., the convergence region and optimal convergence rate) for the SO-DCTS algorithm in undirected networks. Without loss of generality, we assume that ε and γ are real values, and $\varepsilon > 0$.

4.1. Convergence Region for SO-DCTS Algorithm in Undirected Networks

4.1. Convergence Region for SO-DCTS Algorithm in Undirected Networks. From Theorem 2, we know that when $\rho(H - J) < 1$, the SO-DCTS algorithm in an undirected network can achieve average consensus asymptotically. Let us define the convergence region $\mathcal{R}$ to satisfy $\rho(H - J) < 1$. After some algebraic derivations (outlined in Appendix A), the convergence region for the SO-DCTS algorithm in undirected networks is

$$\mathcal{R} = \mathcal{R}^{\dagger} \cup \mathcal{R}^{\ddagger}, \tag{20}$$

where $\mathcal{R}^{\dagger} = \{-1/(\varepsilon\lambda_n(L)) < \gamma < 1, 0 < \varepsilon < 1/\lambda_n(L)\}$, and $\mathcal{R}^{\ddagger} = \{-1/(\varepsilon\lambda_n(L)) < \gamma < 2/(\varepsilon\lambda_n(L)) - 1, 1/\lambda_n(L) \le \varepsilon < 3/\lambda_n(L)\}$.

The convergence region of the SO-DCTS algorithm in undirected networks is shown in Figures 1 and 2 using a three-dimensional and two-dimensional perspective, respectively. We see that compared to the FO-DCTS algorithm where the range of the step size ε is $(0, 2/\lambda_n(L))$, the range of ε in the SO-DCTS approach increases to $(0, 3/\lambda_n(L))$.

4.2. Optimal Convergence Rate for SO-DCTS Algorithm in Undirected Networks. Next, we investigate the fastest convergence rate of the SO-DCTS algorithm based on ε and γ. Recall that in the FO-DCTS algorithm, the constant step size $\varepsilon_{\mathrm{opt,FO}}$ which minimizes convergence time is given as [15]

$$\varepsilon_{\mathrm{opt,FO}} = \frac{2}{\lambda_n(L) + \lambda_2(L)}. \tag{21}$$

Additionally, the convergence rate for $\varepsilon_{\text{opt,FO}}$ is determined by the second largest absolute eigenvalue of the Perron matrix [18], that is,

$$\alpha_{\text{opt,FO}} = \frac{\lambda_n(L) - \lambda_2(L)}{\lambda_n(L) + \lambda_2(L)}. \tag{22}$$

As we show next, the convergence rate of the SO-DCTS algorithm can be superior to that of the FO-DCTS algorithm by choosing suitable ε and γ. However, as stated in the following lemma, the convergence rate of the FO-DCTS algorithm is faster under some circumstances.

Lemma 2. *For the SO-DCTS algorithm in* (4) *in a time-invariant, connected, undirected network with initial conditions* $\vec{\mathbf{t}}(-1) = \vec{\mathbf{t}}(0) = \vec{\mathbf{z}}$ *and* $(\varepsilon, \gamma) \in \mathcal{R}$ *in* (20), *if* $\gamma > 0$, *the convergence rate of the SO-DCTS algorithm is less than that of the FO-DCTS algorithm with the optimal constant step size in* (21).

The proof of this lemma is omitted here since it can be readily extended from the following result. Consider two real values a and b with $b > 0$, then $\max\{(1/2)|a + \sqrt{a^2 + b}|, (1/2)|a - \sqrt{a^2 + b}|\} > a$. Thus, we have $|\lambda_k(H)| > 1 - \varepsilon\lambda_i(L)$, which implies $|\lambda_k(H)| > \alpha_{\text{opt,FO}}$.

Based on the above lemma, we see that there may exist possible choices of ε and γ (e.g., when $\gamma < 0$) such that the convergence rate of the SO-DCTS method is faster than the FO-DCTS algorithm. To see this, we formulate the following spectral radius minimization problem to find the optimal ε and γ for the SO-DCTS algorithm:

$$\begin{aligned} &\text{minimize} \quad \rho(H - J) \\ &\text{subject to} \quad (\varepsilon, \gamma) \in \mathcal{R}, \ \gamma < 0. \end{aligned} \tag{23}$$

Using the steps outlined in Appendix B, the optimal ε and γ to minimize (23) can be obtained as

$$\begin{aligned} \varepsilon_{\text{opt,SO}} &= \frac{3\lambda_n(L) + \lambda_2(L)}{\lambda_n(L)[\lambda_n(L) + 3\lambda_2(L)]}, \\ \gamma_{\text{opt,SO}} &= -\frac{[\lambda_n(L) - \lambda_2(L)]^2}{[\lambda_n(L) + 3\lambda_2(L)][3\lambda_n(L) + \lambda_2(L)]}. \end{aligned} \tag{24}$$

It is worth noting that $(\varepsilon_{\text{opt,SO}}, \gamma_{\text{opt,SO}}) \in \mathcal{R}^{\ddagger}$. Recall that the convergence rate for the SO-DCTS algorithm in undirected networks is determined by the spectral radius of $H - J$, that is,

$$\alpha_{\text{opt,SO}} = \frac{\lambda_n(L) - \lambda_2(L)}{\lambda_n(L) + 3\lambda_2(L)}. \tag{25}$$

We see that $\alpha_{\text{opt,SO}} \le \alpha_{\text{opt,FO}}$ and $\alpha_{\text{opt,SO}} = \alpha_{\text{opt,FO}}$ only when $\lambda_2(L) = \lambda_n(L)$. Thus, we have the following theorem for the convergence rate of the SO-DCTS algorithm.

Theorem 3. *For the SO-DCTS algorithm in* (4) *in a time-invariant, connected, undirected network with initial conditions* $\vec{\mathbf{t}}(-1) = \vec{\mathbf{t}}(0) = \vec{\mathbf{z}}$ *and* $(\varepsilon, \gamma) \in \mathcal{R}$ *in* (20), *there exists a pair of* ε *and* γ *such that the convergence rate of the SO-DCTS algorithm is greater than or equal to that of the FO-DCTS algorithm with the optimal constant step size in* (21).

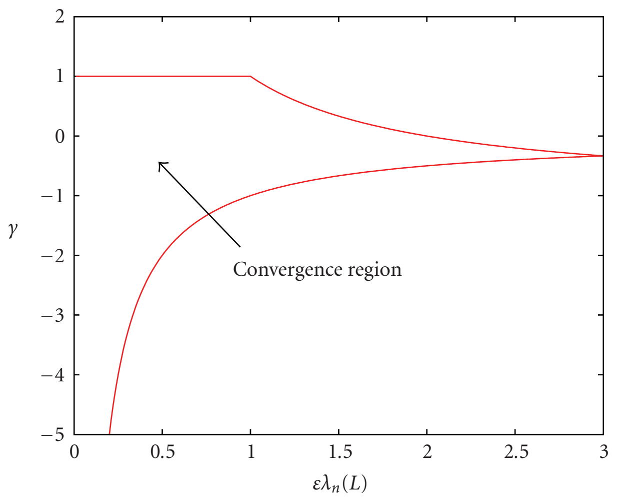

Figure 2: Convergence region for the SO-DCTS algorithm in undirected networks: two-dimensional view.

5. SO-DCTS Algorithm with Gaussian Delay in Undirected Networks

In this section, we investigate the convergence properties of the SO-DCTS algorithm in undirected networks when there is both deterministic and random (Gaussian) delay between network nodes during local time information exchange. In [19], we motivate why the Gaussian assumption is appropriate to model the undeterministic timing differences between nodes exchanging either MAC layer or physical layer timing information. We do not reiterate those arguments here but rather present convergence results for the SO-DCTS algorithm when such timing differences exist. We have separately examined the performance of the SO-DCTS algorithm considering alternate delay distributions, for example, exponential delay distribution [20]. Results show similar performance bounds as those presented in this paper for the Gaussian assumption. For this reason, we constrain our discussion here to the more common Gaussian delay model.

With Gaussian delay, the timing update rule of the SO-DCTS algorithm at each node i is given as

$$\begin{aligned} t_i(k) = t_i(k-1) &+ \varepsilon \sum_{j \in \mathcal{N}_i} [\hat{t}_j(k-1) - t_i(k-1)] \\ &- \gamma\varepsilon \sum_{j \in \mathcal{N}_i} [\hat{t}_j(k-2) - t_i(k-2)], \end{aligned} \tag{26}$$

where $\hat{t}_j(k) = t_j(k) + T_{\text{delay}} = t_j(k) + T_c + L_{ij}/c + \nu_j(k)$; T_c is a constant (deterministic) delay; L_{ij} is the distance between node i and j; c is light speed (thus, L_{ij}/c is the propagation delay between nodes i and j); $\nu_j(k)$ are independent identical distributed (i.i.d) Gaussian random variables, with zero mean and variance σ^2. Local time information exchange between node i and j under this delay model is shown in Figure 3.

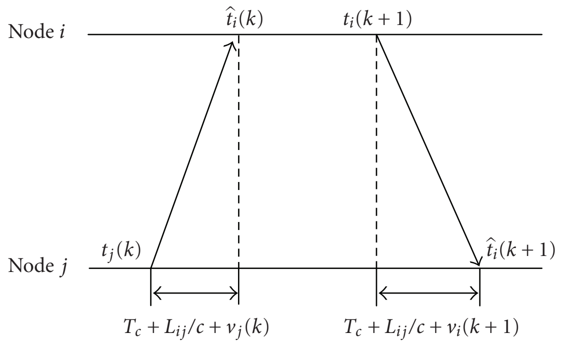

FIGURE 3: SO-DCTS algorithm with Gaussian delay during local time information exchange.

The SO-DCTS algorithm in (26) can be rearranged as

$$t_i(k) = t_i(k-1) + \varepsilon \sum_{j\in\mathcal{N}_i} \left[t_j(k-1) - t_i(k-1)\right] - \gamma\varepsilon \sum_{j\in\mathcal{N}_i} \left[t_j(k-2) - t_i(k-2)\right] + n_i(k-1), \quad (27)$$

where $n_i(k-1) = (1-\gamma)\varepsilon\sum_{j\in\mathcal{N}_i}(T_c + L_{ij}/c) + \varepsilon\sum_{j\in\mathcal{N}_i}[v_j(k-1) - \gamma v_j(k-2)]$.

Let us define the noise vector $\vec{\mathbf{n}}(k) = [n_1(k), n_2(k), \ldots, n_n(k)]^{\mathrm{T}}$. Based on this definition, the evolution of SO-DCTS algorithm in (27) can be written as

$$\vec{\mathbf{t}}(k) = (I_n - \varepsilon L)\vec{\mathbf{t}}(k-1) + \gamma\varepsilon L\vec{\mathbf{t}}(k-2) + \vec{\mathbf{n}}(k-1). \quad (28)$$

We now define $\vec{\mathbf{v}}(k) = [v_1(k), v_2(k), \ldots, v_n(k)]^{\mathrm{T}}$ and $\vec{\mathbf{u}} = [u_1, u_2, \ldots, u_n]^{\mathrm{T}}$, where $u_i = \sum_{j\in\mathcal{N}_i}(T_c + L_{ij}/c)$. Then, the noise vector in (28) is given as $\vec{\mathbf{n}}(k-1) = \varepsilon[(1-\gamma)\vec{\mathbf{u}} + A(\vec{\mathbf{v}}(k-1) - \gamma\vec{\mathbf{v}}(k-2))]$.

Let us additionally define $\vec{\zeta}(k) = [\vec{\mathbf{n}}(k)^{\mathrm{T}} \ \ \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}}$. Then, (28) can be rewritten as

$$\vec{\psi}(k) = H\vec{\psi}(k-1) + \vec{\zeta}(k-1). \quad (29)$$

Recall that for undirected networks, the average value in each iteration is $m(k) = (1/n)\mathbf{1}^{\mathrm{T}}\vec{\mathbf{t}}(k)$. Thus, the mean and variance of the average value $m(k)$ are given in the following lemma.

Lemma 3. *For the SO-DCTS algorithm in* (28)*, the mean and variance of the average value $m(k)$ are given as*

$$\mathbb{E}[m(k)] = m(0) + \frac{k}{n}\sum_{i=1}^{n} u_i, \qquad \mathbf{var}\,[m(k)] = \frac{k\varepsilon^2\sigma^2(1+\gamma^2)}{n^2}\sum_{i=1}^{n} d_i^2. \quad (30)$$

The proof of this lemma is straightforward and thus omitted from this paper. It can be seen that as iteration time increases, both mean and variance in (30) increase linearly with the time index k, that is, as the algorithm evolves. Furthermore, the variance of $m(k)$ increases linearly with the variance of the random Gaussian delay, σ^2. As we will see in our numerical results, although the average value $m(k)$ grows linearly with iteration time when there is Gaussian delay in the network, an average consensus may still be achievable under certain network topologies.

5.1. Expectation and Second Central Moment of Error Vector.

In order to understand the convergence property of SO-DCTS algorithm with Gaussian delay, we first quantify the overall impact of uncertainty by computing the first two moments of the disagreement vector.

With Gaussian delay, we see that the error vector $\vec{e}(k)$ results in the following evolution:

$$\vec{e}(k) = P\vec{e}(k-1) + Q\vec{\zeta}(k-1), \quad (31)$$

where $P = H - J$ and $Q = I_{2n} - J_1$. Then, we have the following lemma.

Lemma 4. *For the SO-DCTS algorithm in* (28)*, the expectation of the error vector $\vec{e}(k)$ is given by*

$$\vec{e}(k) = P^k\vec{e}(0) + (1-\gamma)\varepsilon\sum_{l=0}^{k-1} P^l Q\vec{\mathbf{u}}_1, \quad (k \ge 1), \quad (32)$$

where $\vec{\mathbf{u}}_1 = [\vec{\mathbf{u}}^{\mathrm{T}} \ \ \mathbf{0}^{\mathrm{T}}]^{\mathrm{T}}$.

The proof of this lemma is straightforward and thus omitted from this paper.

Let us define the second central moment of the error vector as $\kappa_e(k) = \mathbb{E}\{(\vec{e}(k) - \mathbb{E}[\vec{e}(k)])^{\mathrm{T}}(\vec{e}(k) - \mathbb{E}[\vec{e}(k)])\}$ and the covariance matrix of the error vector as $\Sigma_e(k) = \mathbb{E}\{(\vec{e}(k) - \mathbb{E}[\vec{e}(k)])(\vec{e}(k) - \mathbb{E}[\vec{e}(k)])^{\mathrm{T}}\}$. It is worth mentioning that $\kappa_e(k) = \mathbf{tr}(\Sigma_e(k))$, where $\mathbf{tr}(\cdot)$ denotes the trace of a matrix. Additionally, let us denote the covariance matrix of $\vec{\zeta}(k)$ as $\Sigma_\zeta = \mathbb{E}\{(\vec{\zeta}(k) - \mathbb{E}[\vec{\zeta}(k)])(\vec{\zeta}(k) - \mathbb{E}[\vec{\zeta}(k)])^{\mathrm{T}}\}$ which is given as

$$\Sigma_\zeta = \varepsilon^2(1+\gamma^2)\sigma^2\begin{bmatrix} A^2 & \mathbf{0}_{n\times n} \\ \mathbf{0}_{n\times n} & \mathbf{0}_{n\times n}\end{bmatrix}. \quad (33)$$

Given these definitions, we next note Lemma 5.

Lemma 5. *For the SO-DCTS algorithm in* (28)*, the covariance matrix of the error vector $\vec{e}(k)$ is given as*

$$\Sigma_e(k) = P^k\vec{e}(0)\vec{e}(0)^{\mathrm{T}}(P^{\mathrm{T}})^k + \sum_{l=0}^{k-1} P^l Q\Sigma_\zeta Q(P^{\mathrm{T}})^l, \quad (k \ge 1), \quad (34)$$

and the second central moment of the error vector $\vec{e}(k)$ is given as

$$\kappa_e(k) = \vec{e}(0)^{\mathrm{T}}(P^{\mathrm{T}})^k P^k\vec{e}(0) + \mathbf{tr}\left(Q\sum_{l=0}^{k-1}(P^{\mathrm{T}})^l P^l Q\Sigma_\zeta\right), \quad (k \ge 1). \quad (35)$$

The proof of this lemma is similar to [19] and thus omitted from the paper.

5.2. Asymptotic Expectation of Global Synchronization Error. Using Lemma 4, we see that the steady state of expectation of the error vector $\vec{e}(k)$ is

$$\lim_{k\to\infty}\vec{e}(k) = (1-\gamma)\varepsilon(I_{2n}-P)^{-1}Q\vec{\mathbf{u}}_1. \tag{36}$$

The above equation holds because $\lim_{k\to\infty}P^k = \lim_{k\to\infty}(H^k - J) = \mathbf{0}$. Before we investigate the convergence property of SO-DCTS algorithm with Gaussian delay, we give the following lemma for block matrix inversion.

Lemma 6. *Consider $n\times n$ matrices A_1, A_2, A_3, and A_4, when A_4 and $C = A_1 - A_2A_4^{-1}A_3$ are nonsingular, then [16]*

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}^{-1} = \begin{bmatrix} C^{-1} & -C^{-1}A_2A_4^{-1} \\ -A_4^{-1}A_3C^{-1} & A_4^{-1}+A_4^{-1}A_3C^{-1}A_2A_4^{-1} \end{bmatrix}. \tag{37}$$

Based on this lemma, the steady state of error vector $\vec{e}(k)$ is

$$\begin{aligned}\lim_{k\to\infty}\vec{e}(k) &= (1-\gamma)\varepsilon\begin{bmatrix} W_1 & \gamma\varepsilon W_1 L \\ GW_1 & I_n+\gamma\varepsilon GW_1L \end{bmatrix}\begin{bmatrix} G\vec{\mathbf{u}} \\ \mathbf{0}\end{bmatrix} \\ &= (1-\gamma)\varepsilon\begin{bmatrix} W_1G\vec{\mathbf{u}} \\ W_1G\vec{\mathbf{u}}\end{bmatrix},\end{aligned} \tag{38}$$

where $G = I_n - K$ and $W_1 = [(1-\gamma)\varepsilon L + K]^{-1}$. The above equation is valid because $KW_1 = K$, which implies $KW_1G = \mathbf{0}_{n\times n}$, which in turn implies $GW_1G = W_1G$. Specifically, we see that the eigenvalues of W_1 are $\lambda_1(W_1) = 1$ and $\lambda_i(W_1) = 1/[(1-\gamma)\varepsilon\lambda_i(L)], i = 2,\ldots,n$. Additionally, the steady state of the disagreement vector $\vec{\delta}(k)$ is upper half of the vector $\lim_{k\to\infty}\vec{e}(k)$, that is,

$$\vec{\mu}(\infty) \triangleq \lim_{k\to\infty}\vec{\delta}(k) = (1-\gamma)\varepsilon W_1G\vec{\mathbf{u}}. \tag{39}$$

For this $\vec{\mu}(\infty)$, we can show the following theorem.

Theorem 4. *In an undirected network with fixed connected topology, $\vec{\mu}(\infty)$ in* (39) *is a constant vector independent of the constant values of ε and γ.*

Proof. Let us denote the eigenvectors of W_1 as w_i. It is easy to check that the eigenvector corresponding to $\lambda_1(W_1) = 1$ is $w_1 = \mathbf{1}$. $\vec{\mu}(\infty)$ in (39) can thus be rewritten as

$$\begin{aligned}\vec{\mu}(\infty) &= (1-\gamma)\varepsilon\mathbf{1}\mathbf{1}^{\mathrm{T}}G\vec{\mathbf{u}} + (1-\gamma)\varepsilon\left[\sum_{i=2}^{n}\lambda_i(W_1)w_iw_i^{\mathrm{T}}\right]G\vec{\mathbf{u}} \\ &= (L+K)^{-1}G\vec{\mathbf{u}}.\end{aligned} \tag{40}$$

Therefore, $\vec{\mu}(\infty)$ does not depend on ε and γ. This completes the proof. □

Thus, for constants ε and γ, the steady state of the expectation of the disagreement vector is a constant vector regardless of ε and γ. In other words, in an undirected network with fixed topology, the expectation of global synchronization error is the same regardless of the speed of synchronization. We observed the same phenomena in the FO-DCTS algorithm with Gaussian delay [19]. Let us now define the asymptotic expectation of pair-wise synchronization error as

$$\Delta t_{i,j} = \lim_{k\to\infty}\mathbb{E}[t_i(k)-t_j(k)] = \mu_i(\infty)-\mu_j(\infty), \quad \forall i,j\in\mathcal{V}. \tag{41}$$

Hence, the maximum asymptotic expectation of the global synchronization error between any two nodes is $\Delta t_{\max} = \max\{|\Delta t_{i,j}|\}$. Then, we have the following definition.

Definition 1. A connected network is called "average consensus achievable with tolerable synchronization error" if the maximum asymptotic expectation of the global time synchronization error is less than a predefined threshold Δt_{Th} when applying the SO-DCTS algorithm in (28), that is, when $\Delta t_{\max} < \Delta t_{\mathrm{Th}}$.

Similar to [19], we have Definition 2.

Definition 2. A network is called "time delay balanced network" if the delay

$$\sum_{j\in\mathcal{N}_i}(T_c+L_{ij}/c) = \sum_{m\in\mathcal{N}_k}(T_c+L_{km}/c), \quad (i,j)\in\mathcal{E},(k,m)\in\mathcal{E}, \tag{42}$$

or equivalently, $\Delta t_{\max} = 0$.

5.3. Asymptotic Mean Square Time Synchronization Error. Using Lemma 5, the steady state of the second central moment of the error vector is

$$\kappa_e(\infty) \triangleq \lim_{k\to\infty}\kappa_e(k) = \mathbf{tr}(QW_2Q\Sigma_\zeta), \tag{43}$$

where $W_2 = \sum_{l=0}^{\infty}(P^{\mathrm{T}})^lP^l$. Note that W_2 satisfies the following condition:

$$I + P^{\mathrm{T}}W_2P = W_2. \tag{44}$$

Let us denote the covariance matrix and second central moment of the disagreement vector as $\Sigma_\delta(k)$ and $\kappa_\delta(k)$, respectively. We see that

$$\mathbf{tr}(\Sigma_e(k)) = \mathbf{tr}(\Sigma_\delta(k)) + \mathbf{tr}(\Sigma_\delta(k-1)). \tag{45}$$

Therefore, as $k\to\infty$, the steady state of second central moment of disagreement vector is

$$\kappa_\delta(\infty) \triangleq \lim_{k\to\infty}\kappa_\delta(k) = \frac{\kappa_e(\infty)}{2} = \frac{\mathbf{tr}(QW_2Q\Sigma_\zeta)}{2}. \tag{46}$$

We now define the asymptotic mean square time synchronization error as

$$\sigma_{\Delta t}^2 = \lim_{k\to\infty}\sum_{i=1}^{n}\mathbb{E}[|t_i(k)-m(k)|^2], \tag{47}$$

which indicates the amount of error by which the updated time at each node differs from the average value over all n nodes. We see that

$$\sigma_{\Delta t}^2 = \vec{\mathbf{u}}^{\mathrm{T}} Q(L+K)^{-2} Q\vec{\mathbf{u}} + \frac{\mathbf{tr}(QW_2Q\Sigma_\zeta)}{2}. \tag{48}$$

6. Simulation Results

In the following simulation results, we assume that the initial time offset of node i is $(i-1/2)T/n, i = 1,\ldots,n$, where $T = 1000$ microseconds unless otherwise stated (trends similar to the ones noted below were observed when initial time offsets between nodes were arbitrary (e.g., when they were uniformly distributed over $[0,T]$). We use this fixed offset assumption here for comparison purposes).

6.1. Structured Networks. In our simulations, we examine the convergence performance of the FO-DCTS and SO-DCTS algorithms for several structured, undirected networks. Specifically, we study the following network topologies.

Definition 3. "A Ring Network with Equal Distance (R_n)": A ring network is a network that consists of a single cycle. The ring network with equal distance is a ring network that has n nodes, n edges, and $L_c = L_{ij} = L_{km}$ for $(i,j) \in \mathcal{E}$ and $(k,m) \in \mathcal{E}$.

Definition 4. "A Path Network with Equal Distance (P_n)": A path network is a network that consists of edge set $\{(i,i+1), 1 \le i < n\}$. The path network with equal distance is a path network that has n nodes, $n-1$ edges and $L_c = L_{ij} = L_{km}$ for $(i,j) \in \mathcal{E}$ and $(k,m) \in \mathcal{E}$.

Definition 5. "A Star Network with Equal Distance (S_n)": A star network is a network that consists of edge set$\{(i,n), 1 \le i < n\}$. The star network with equal distance is a star network that has n nodes, $n-1$ edges, and $L_c = L_{ij} = L_{km}$ for $(i,j) \in \mathcal{E}$ and $(k,m) \in \mathcal{E}$.

Figure 4 shows examples of these networks: a ring network R_8, a path network P_5, and a star network S_8. Based on Definition 2, we see that R_n is a "*time delay balanced network*" and $\Delta t_{\max} = 0$. We now explore the convergence properties of the SO-DCTS algorithm for these structured networks via simulation.

Optimal Convergence Rate. First we compare the convergence speeds of the SO-DCTS and FO-DCTS algorithms for the above structured networks assuming that the convergence rate is defined as $\nu = -\log(\alpha)$, and there is no Gaussian delay between nodes. Table 1 gives the numerical values of the optimal convergence rate for the SO-DCTS and FO-DCTS algorithms under the R_{16}, P_{16}, and S_{16} topologies. As expected, the SO-DCTS algorithm converges faster than the FO-DCTS algorithm in all three cases. Specifically, we see that the optimal convergence rate of the SO-DCTS algorithm is nearly twice as that of the FO-DCTS algorithm for all three types of networks.

Table 1: Numerical results comparing convergence rates of FO-DCTS and SO-DCTS algorithms in R_{16}, P_{16}, S_{16}.

	R_{16}		P_{16}		S_{16}	
	α_{opt}	ν_{opt}	α_{opt}	ν_{opt}	α_{opt}	ν_{opt}
FO-DCTS Alg.	0.9267	0.0762	0.9808	0.0194	0.8824	0.1252
SO-DCTS Alg.	0.8634	0.1469	0.9623	0.0384	0.7895	0.2364

Table 2: Asymptotic results for the SO-DCTS algorithm in structured networks with Gaussian delay.

	R_{16}	P_{16}	S_{16}
$\Delta t_{\max}$ (μs)	0	35	8.75
$\sigma_{\Delta t}^2$	305.8075	13329	84.2996

Convergence Properties of SO-DCTS Algorithm with Gaussian Delay. In our simulations of the SO-DCTS algorithm with Gaussian delay, we assume $T_c + L_c/c = 10$ microseconds and the optimal values of $\varepsilon_{\text{opt,SO}}$ and $\gamma_{\text{opt,SO}}$ from (24). The simulation results and the asymptotic mean square time synchronization errors for the R_{16}, P_{16}, and S_{16} networks are shown in Figure 5. For each network topology, the asymptotic mean square time synchronization error $\sigma_{\Delta t}^2$ is calculated from (48). It can be seen that as time index increases, mean square time synchronization error approaches the steady-state value when utilizing SO-DCTS algorithm with Gaussian delay. Additionally, we see that the SO-DCTS algorithm performs poorest in a path network where it has the largest value of $\sigma_{\Delta t}^2$ and the slowest convergence speed. This is not surprising since in such networks information flow from node 1 to node n requires $n-1$ hops.

Table 2 summarizes the asymptotic results of the SO-DCTS algorithm for structured networks. As expected, the maximum asymptotic expectation of global time synchronization error for R_n is 0 since R_n is a *time delay balanced network*. Furthermore, the SO-DCTS algorithm in P_n has the largest $\Delta t_{\max}$ because of its highly unbalanced time delay structure. It is worth mentioning that the SO-DCTS algorithm in star networks S_n has relatively small values of $\Delta t_{\max}$ and $\sigma_{\Delta t}^2$. In fact, the SO-DCTS algorithm for a star network can be seen as a type of centralized time synchronization algorithm in which a root node determines and propagates the average of local time information of all other nodes in the network.

In Figure 6, we show the asymptotic value of $\sigma_{\Delta t}^2$ as a function of the number of nodes in these structured networks. It can be seen that when using the optimal $\varepsilon_{\text{opt,SO}}$ and $\gamma_{\text{opt,SO}}$, the asymptotic mean square time synchronization error for a star network is nearly constant as the number of nodes increases. However, $\sigma_{\Delta t}^2$ is an increasing function of the number of nodes for both path and ring networks.

6.2. Random Networks. We also present here simulation results for a random network comprised of n nodes that were randomly generated with uniform distribution over a unit square kilometer; two nodes were assumed connected if the distance between them was less than η, a predefined threshold. One realization of such a network with 16 nodes

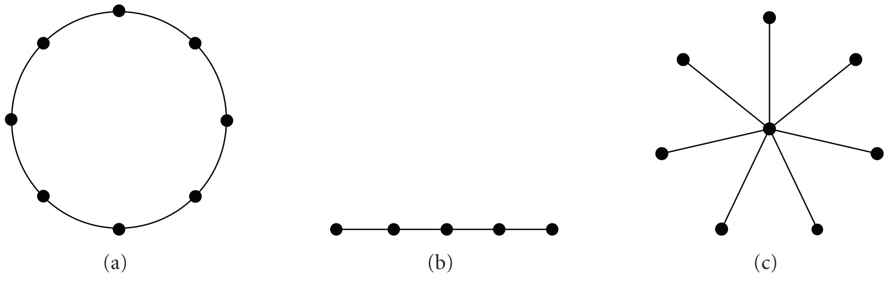

FIGURE 4: Structured networks: (a) R_8, (b) P_5, (c) S_8.

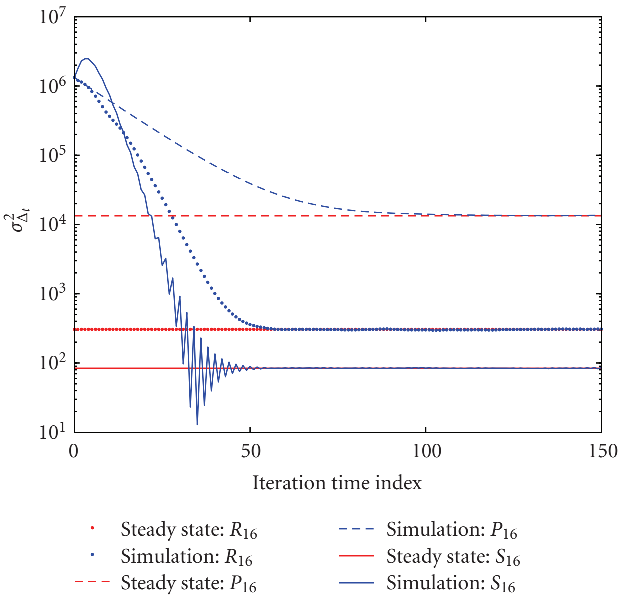

FIGURE 5: $\sigma^2_{\Delta t}$ as a function of the iteration time index for the SO-DCTS algorithm in structured networks (R_{16}, P_{16}, S_{16}) with Gaussian delay.

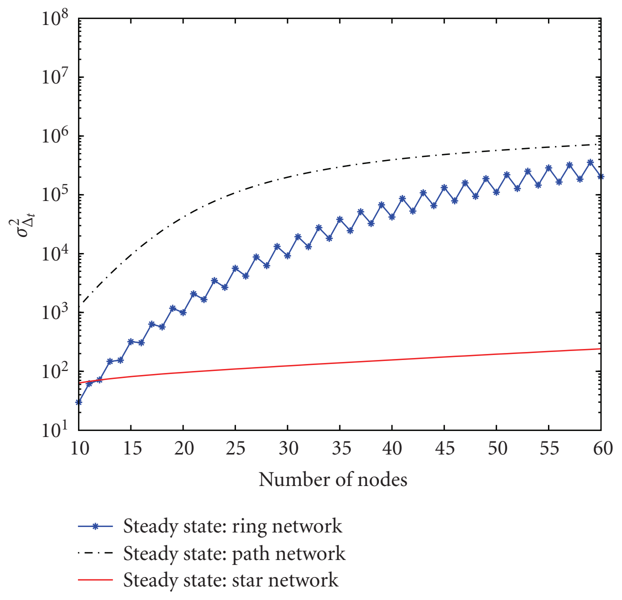

FIGURE 6: $\sigma^2_{\Delta t}$ as a function of the number of nodes for the SO-DCTS algorithm in structured networks with Gaussian delay.

is shown in Figure 7. We assume that the average distance between two nodes is 0.5 km.

Figure 8 shows the simulation results for the convergence rates of the FO-DCTS and SO-DCTS algorithms in random networks with 256 nodes when $\eta = 0.25$ without Gaussian delay between network nodes. Specifically, we plot the mean square time synchronization error (defined as $(1/n)\|\vec{\delta}(k)\|^2$). In simulating random networks, we average results over 5000 network realizations. To obtain these results, we chose $\varepsilon_{\text{opt,FO}}$ for the FO-DCTS algorithm and $\varepsilon_{\text{opt,SO}}$ and $\gamma_{\text{opt,SO}}$ for the SO-DCTS algorithm. In Figure 8, we observe that the optimal convergence rate of the SO-DCTS algorithm is faster than that of the FO-DCTS algorithm. In addition to the results shown here, we ran this simulation setup for various realizations of random networks, assuming both $n = 256$ and a smaller network with $n = 16$. Overall, the results show a similar trend, that is, the convergence rate of the SO-DCTS algorithm exceeds the FO-DCTS algorithm.

Figure 9 shows the simulation results when the SO-DCTS algorithm is implemented in a random network

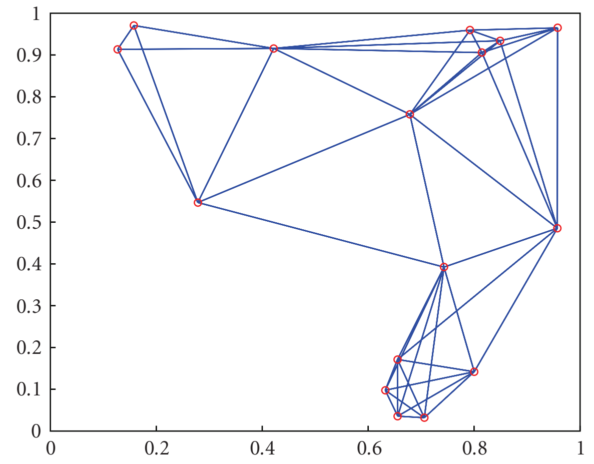

FIGURE 7: Random network with 16 nodes used to obtain simulation results in Figure 9.

of Figure 7 assuming Gaussian delay between network nodes. As expected, we see here that an asymptotic global

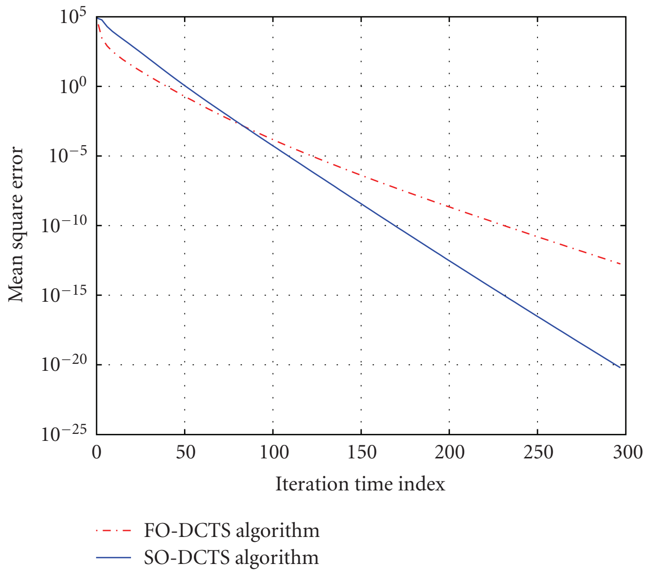

Figure 8: Evolutions of the FO-DCTS and SO-DTS algorithms in random network with 256 nodes when $\eta = 0.25$ without Gaussian delay between network nodes.

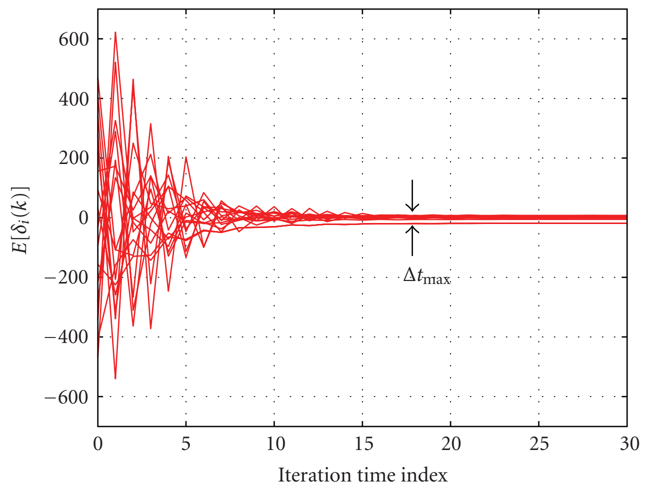

Figure 9: Evolution of the average disagreement of the SO-DCTS algorithm in random network (see Figure 7) with Gaussian delay between network nodes.

synchronization error persists between some pairs of nodes, that is, $\Delta t_{\max} = 26.4130$ microseconds for this random network. If we specify a threshold Δt_{Th} to be greater than or equal to this $\Delta t_{\max}$, we call this network as "*average consensus achievable with tolerable synchronization error*" as described in Definition 1.

7. Conclusions

In this paper, we propose a novel discrete-time SO-DCTS algorithm to address the global timing synchronization problem in wireless sensor networks. We analyze several important convergence characteristics of the SO-DCTS algorithm for directed and undirected networks. Additionally, we investigate the convergence region and optimal convergence rate of the SO-DCTS algorithm in undirected networks and claim that the optimal convergence rate of the SO-DCTS algorithm is superior to that of the FO-DCTS algorithm under an appropriate algorithm design. Furthermore, we investigate the asymptotic expectation and mean square synchronization error of the SO-DCTS algorithm when there is Gaussian delay between network nodes. In the future, we intend to investigate the effects of skew, link failure, and other practical conditions when utilizing the SO-DCTS algorithm in wireless sensor networks.

Appendices

A. Convergence Region for SO-DCTS Algorithm in Undirected Networks

Let us denote the pairs of eigenvalues of H corresponding to $\lambda_i(L)$ as $\lambda_{i'}(H)$ and $\lambda_{i''}(H)$, that is,

$$\lambda_{i'}(H) = \frac{1}{2}\left[1 - \varepsilon\lambda_i(L) + \sqrt{(1-\varepsilon\lambda_i(L))^2 + 4\gamma\varepsilon\lambda_i(L)}\right],$$
$$\lambda_{i''}(H) = \frac{1}{2}\left[1 - \varepsilon\lambda_i(L) - \sqrt{(1-\varepsilon\lambda_i(L))^2 + 4\gamma\varepsilon\lambda_i(L)}\right]. \tag{A.1}$$

Now, we examine the convergence region for the SO-DCTS algorithm based on conditions $|\lambda_{i'}(H)| < 1, 1 < i' \leq n$, and $|\lambda_{i''}(H)| < 1, 1 < i'' \leq n$.

Case 1. When $\lambda_{i'}(H)$ and $\lambda_{i''}(H)$ are real values: in this case, we have $(1-\varepsilon\lambda_i(L))^2 + 4\gamma\varepsilon\lambda_i(L) \geq 0$, that is,

$$\gamma \geq -\frac{[1-\varepsilon\lambda_i(L)]^2}{4\varepsilon\lambda_i(L)}, \quad 1 < i \leq n. \tag{A.2}$$

In the following, we assume that $1 < i, i', i'' \leq n$ unless otherwise stated.

(1) First, we consider the convergence region for $\lambda_{i'}(H)$. After some manipulations, we can show that the convergence region is

$$\left\{\gamma < 1,\ 0 < \varepsilon < \frac{3}{\lambda_i(L)}\right\}$$
$$\cup \left\{\frac{2-\varepsilon\lambda_i(L)}{\varepsilon\lambda_i(L)} < \gamma < 1,\ \varepsilon > \frac{3}{\lambda_i(L)}\right\}. \tag{A.3}$$

(2) Then, we consider the convergence region for $|\lambda_{i''}(H)| < 1$ which is given as

$$\left\{\gamma < \frac{2-\varepsilon\lambda_i(L)}{\varepsilon\lambda_i(L)},\ 0 < \varepsilon < \frac{3}{\lambda_i(L)}\right\}. \tag{A.4}$$

Combining the convergence region for $\lambda_{i'}(H)$ and $\lambda_{i''}(H)$ with (A.2), the convergence region $\mathcal{R}_1$ for this case is

$$\mathcal{R}_1 = \left\{ -\frac{[1-\varepsilon\lambda_i(L)]^2}{4\varepsilon\lambda_i(L)} \le \gamma < 1,\ 0 < \varepsilon < \frac{1}{\lambda_i(L)} \right\} \cup \left\{ -\frac{[1-\varepsilon\lambda_i(L)]^2}{4\varepsilon\lambda_i(L)} \le \gamma < \frac{2-\varepsilon\lambda_i(L)}{\varepsilon\lambda_i(L)},\ \frac{1}{\lambda_i(L)} \le \varepsilon < \frac{3}{\lambda_i(L)} \right\} \tag{A.5}$$

Case 2. When $\lambda_{i'}(H)$ and $\lambda_{i''}(H)$ are complex values: In this case, we have $(1-\varepsilon\lambda_i(L))^2 + 4\gamma\varepsilon\lambda_i(L) < 0$, that is,

$$\gamma < -\frac{[1-\varepsilon\lambda_i(L)]^2}{4\varepsilon\lambda_i(L)}. \tag{A.6}$$

Here, $\Re\{\lambda_{i'}(H)\} = \Re\{\lambda_{i''}(H)\}$ and $\Im\{\lambda_{i'}(H)\} = -\Im\{\lambda_{i''}(H)\}$. Thus, we only need to examine the convergence region for $|\lambda_{i'}(H)|$. In order to satisfy the conditions, we have

(1) the real part of $\lambda_{i'}(H)$ should be less than 1, that is, $|\Re\{\lambda_{i'}(H)\}| < 1$, then we have

$$0 < \varepsilon < \frac{3}{\lambda_i(L)}; \tag{A.7}$$

(2) the imaginary part of $\lambda_{i'}(H)$ should be less than 1, that is, $|\Im\{\lambda_{i'}(H)\}| < 1$, then we have

$$-\frac{4+[1-\varepsilon\lambda_i(L)]^2}{4\varepsilon\lambda_i(L)} < \gamma < -\frac{[1-\varepsilon\lambda_i(L)]^2}{4\varepsilon\lambda_i(L)}; \tag{A.8}$$

(3) the absolute value of $\lambda_{i'}(H)$ should be less than 1, that is, $\Re^2\{\lambda_{i'}(H)\} + \Im^2\{\lambda_{i'}(H)\} < 1$, then we have

$$\gamma > -\frac{1}{\varepsilon\lambda_i(L)}. \tag{A.9}$$

Combining the above results, the convergence region $\mathcal{R}_2$ for this case is

$$\mathcal{R}_2 = \left\{ -\frac{1}{\varepsilon\lambda_i(L)} < \gamma < -\frac{[1-\varepsilon\lambda_i(L)]^2}{4\varepsilon\lambda_i(L)},\ 0 < \varepsilon < \frac{3}{\lambda_i(L)} \right\}. \tag{A.10}$$

By taking the union of $\mathcal{R}_1$ in (A.5) and $\mathcal{R}_2$ in (A.10) and considering the increasing order of $\lambda_i(L)$, the convergence region for the SO-DCTS algorithm in (20) is obtained.

B. Solution for Minimization Problem

Here, we give a sketch solution to the spectral radius minimization problem in (23). Since $\lambda_2(L) \le \cdots \le \lambda_n(L)$, the optimization problem is equivalent to minimize

$$\max\{|\lambda_{2'}(H)|, |\lambda_{2''}(H)|, |\lambda_{n'}(H)|, |\lambda_{n''}(H)|\}. \tag{B.1}$$

(1) First, we find the optimal γ given ε to minimize (B.1). Here, we consider four different cases depending on whether $\lambda_{2'}(H), \lambda_{2''}(H), \lambda_{n'}(H), \lambda_{n''}(H)$ are real values or complex values. After algebraic derivations, we can show that the minimum of (B.1) given ε can be achieved when $\lambda_{2'}(H)$ and $\lambda_{2''}(H)$ are real values and $\lambda_{n'}(H)$ and $\lambda_{n''}(H)$ are complex values. Additionally, the following equation should be satisfied:

$$|\lambda_{2'}(H)| = |\lambda_{n'}(H)| = |\lambda_{n''}(H)|. \tag{B.2}$$

Thus, we have

$$\gamma = -\frac{\lambda_n(L)[1-\varepsilon\lambda_2(L)]^2}{\varepsilon[\lambda_2(L)+\lambda_n(L)]^2}. \tag{B.3}$$

(2) Next, we find the optimal ε given γ to minimize (B.1). Again, this can be achieved by taking $\Im\{\lambda_{n'}(H)\} = 0$. Then, we have the following relationship between ε and γ:

$$\gamma = -\frac{[1-\varepsilon\lambda_n(L)]^2}{4\varepsilon\lambda_n(L)}. \tag{B.4}$$

Combining (B.3) with (B.4), we get (24).

Acknowledgment

This work was supported in part by research grants from Thales Communications, Inc., Md, USA, and the National Science Foundation.

References

[1] D. Culler, D. Estrin, and M. Srivastava, "Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.

[2] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03)*, pp. 138–149, Los Angeles, Calif, USA, November 2003.

[3] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proceedings of the 5th Symposium on Operating Systems Design and Implementation (OSDI '02)*, pp. 147–163, Boston, Mass, USA, December 2002.

[4] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The flooding time synchronization protocol," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 39–49, Baltimore, Md, USA, November 2004.

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

[6] A. Giridhar and P. R. Kumar, "Distributed clock synchronization over wireless networks: algorithms and analysis," in *Proceedings of the 45th IEEE Conference on Decision and Control (CDC '06)*, pp. 4915–4920, San Diego, Calif, USA, December 2006.

[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[8] L. Schenato and G. Gamba, "A distributed consensus protocol for clock synchronization in wireless sensor network," in *Proceedings of the 46th IEEE Conference on Decision and Control (CDC '07)*, pp. 2289–2294, New Orleans, La, USA, December 2007.

[9] O. Simeone and U. Spagnolini, "Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, Article ID 57054, 13 pages, 2007.

[10] H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.

[11] W. Ren and E. Atkins, "Second-order consensus protocols in multiple vehicle systems with local interactions," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference (GN&C '05)*, pp. 3689–3701, San Francisco, Calif, USA, August 2005.

[12] P. Lin, Y. Jia, J. Du, and S. Yuan, "Distributed consensus control for second-order agents with fixed topology and time-delay," in *Proceedings of the 26th Chinese Control Conference (CCC '07)*, pp. 577–581, Zhangjiajie, China, July-June 2007.

[13] W. Ren, "Second-order consensus algorithm with extensions to switching topologies and reference models," in *Proceedings of the American Control Conference (ACC '07)*, pp. 1431–1436, New York, NY, USA, July 2007.

[14] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1985.

[15] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, vol. 5, pp. 4997–5002, Maui, Hawaii, USA, December 2003.

[16] C. D. Meyer, *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2001.

[17] 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.

[18] S. Kar and J. M. F. Moura, "Topology for global average consensus," in *Proceedings of the 40th Asilomar Conference on Signals, Systems and Computers (ACSSC '06)*, pp. 276–280, Pacific Grove, Calif, USA, October-November 2006.

[19] G. Xiong and S. Kishore, "Analysis of distributed consensus time synchronization with Gaussian delay over wireless sensor networks," submitted to *EURASIP Journal on Wireless Communications and Networking*.

[20] H. S. Abdel-Ghaffar, "Analysis of synchronization algorithms with time-out control over networks with exponentially symmetric delays," *IEEE Transactions on Communications*, vol. 50, no. 10, pp. 1652–1661, 2002.