



ELSEVIER

Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

A proportional integral estimator-based clock synchronization protocol for wireless sensor networks

Wenlun Yang^{a,*}, Minyue Fu^{a,b}^a College of Control Science and Engineering and State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China^b School of Electrical Engineering and Computer Science, University of Newcastle, NSW 2308, Australia

ARTICLE INFO

Article history:

Received 1 October 2016

Received in revised form

28 February 2017

Accepted 26 March 2017

Available online 12 April 2017

Keywords:

Wireless sensor networks

Clock synchronization

Proportional–integral estimator

ABSTRACT

Clock synchronization is an issue of vital importance in applications of WSNs. This paper proposes a proportional integral estimator-based protocol (EBP) to achieve clock synchronization for wireless sensor networks. As each local clock skew gradually drifts, synchronization accuracy will decline over time. Compared with existing consensus-based approaches, the proposed synchronization protocol improves synchronization accuracy under time-varying clock skews. Moreover, by restricting synchronization error of clock skew into a relative small quantity, it could reduce periodic re-synchronization frequencies. At last, a pseudo-synchronous implementation for skew compensation is introduced as synchronous protocol is unrealistic in practice. Numerical simulations are shown to illustrate the performance of the proposed protocol.

© 2017 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Recent years have witnessed great advancement in smaller, cheaper and low-power sensors which are capable of sensing, collecting, processing data and communication through wireless channel [1]. Sensor networks are mainly used for data fusion [2], which highlights the necessity for a synchronized clock of time among sensors, that is, all local sensors should share the same global reference time. For example, in distributed data fusion process, sensor readings and time-stamps are grouped into packages and then pass along to their neighbours so that fusion of such information will be used to calculate a precise estimate. Indeed, the fusion of individual sensor readings is meaningful only with packets that are time-stamped by each sensor's local clock. High accuracy of local clocks is also essential for energy-saving purposes [3], as sensor nodes need to spend most of the time in the sleeping mode with only occasional interactions with neighbouring nodes. Furthermore, most common services in WSN, including coordination, communication, object tracking or distributed logging also depend on the existence of global time [4,5].

To develop successful clock synchronization protocols for WSNs, several issues need to be considered carefully. First, WSNs have wide deployment of sensors which increase the complexity of the network. This leads to scalability requirements for the

synchronization protocols. Additionally, wireless communication is unreliable and may suffer from severe interference. Hence the synchronization protocol need to enhance the robustness in order to avoid node failures and packet losses. Furthermore, the energy conservation becomes a significant concern due to the fact that the smaller size sensors are almost battery-based with limited power supply. To avoid this restriction, it is required to optimize energy use in software levels. Effective protocol with low overhead in both communication and computation still remains to be studied further.

There are two kinds of clock synchronization protocols: structure-based and distributed. In structure-based protocols a hierarchical topology is created within the WSNs. Initially, one node is chosen to be the root node which is treated as the global clock reference, then a spanning tree based on this root node is created. Afterwards, each node synchronizes both its clock skew and its offset with respect to its parent node. Typical examples are listed as follows. Timing-sync Protocol for Sensor Networks (TPSN) [6] establishes a hierarchical structure in the network and then a pairwise synchronization is performed to construct a global timescale throughout the network. Flooding Time Synchronization Protocol (FTSP) [7] initially elects the root of the network which maintains the global time and all other nodes synchronize their clocks to that of the root with periodic flooding packets. Reference Broadcast Synchronization (RBS) [8] is proposed as one-hop time synchronization, where a node is selected as reference node and then broadcasts a sequence of synchronization messages to other receivers in order to estimate both clock skew and offset of local clocks relative to each other. Sari et al. [9] further apply the joint

* Corresponding author.

E-mail addresses: yangwenlun@zju.edu.cn (W. Yang), minyue.fu@newcastle.edu.au (M. Fu).

maximum likelihood (JML) estimator of clock offset and skew under exponential noise model in RBS protocol. Besides, [10] develops a receiver-only synchronization which can synchronize a series of sensor nodes by receiving time stamps of pair-wise references while it could reduce the energy consumption of the whole network. To deal with a time-varying nature of the clock offset, a novel Bayesian approach to the clock offset estimation is proposed in [11]. In most cases, structure-based protocols suffer from computational overhead if a new root needs to be elected under the circumstance of dynamic changes of communication topology. To our best knowledge, they do not satisfactorily handle node failures or packet losses.

Confronted with the above problems, distributed protocols have been proposed for time synchronization in WSNs. These protocols work in a distributed way and do not require a specific tree topology or a root node, thus have the advantages of being scalable and robust to node failure and packet losses. Typical examples include [12–19]. Among distributed protocols consensus-based ones serve as the most popular designing methods. Existing consensus-based algorithms can be classified into two main categories according to their ways of implementation, synchronous [20,21] and asynchronous protocols, also known as gossip [22]. For asynchronous protocols, Schenato proposed an Average TimeSync (ATS) protocol [23] which is based on a cascade of two consensus algorithms to make all the nodes converge to a virtual reference clock by tuning compensation parameters for each node. CCS [24] reduces the clock errors between nodes whose locations are geographically close and achieves long lasting synchronization by converging all nodes to a common skew. He proposed a novel maximum time synchronization algorithm (MTS) [25,26], for delay-free case and a weighted maximum time synchronization algorithm for random delay case. Other work includes [27], etc. For synchronous implementation, see [28–31]. Initially, synchronous implementation seems unrealistic as it requires each node to update its information simultaneously, which implicitly requires a common clock, which contradicts the fact that they do not share a common global clock. Carli et al. [29] proposed a synchronization algorithm that is based on a proportional–integral (PI) consensus-based controller. A similar approach, based on the second-order consensus algorithm, has been proposed in [30] to deal with the synchronization of networks of non-identical double integrators. Based on [30], Carli and Zampieri [31] further develop a pseudo-synchronous implementation way for synchronous protocols and it is proved to have the same performance. Since then researchers can focus on designing synchronous synchronization protocols but implementing them using a pseudo-synchronous implementation.

In this paper, we consider a distributed approach and develop a proportional–integral estimator-based protocol (EBP) for clock synchronization over WSNs. As each local clock skew may experience slow drift due to external environmental conditions such as ambient temperature or battery voltage and on oscillator aging, even if all clocks are perfectly synchronized at a certain time instant, they will slowly diverge from each other. In the case of slow changes of clock skews, tracking is a preferable choice. Compared with the existing protocols, the concrete technical merits of the proposed algorithm can be summarized as follows:

1. Theoretical contribution to tackle with clock synchronization time-varying clock skew. Most existing synchronization algorithms either ignore the drifted clock skew or ideally assume the change of clock skew as a zero-mean noise [32]. Ahmad [11] proposed a novel Bayesian approach to deal with time-varying clock offset estimation by using a factor graph representation of

the posterior density but only in scenarios of pairwise synchronization. In spite of realizing convergence of clock parameters, they fail to take time-varying clock skew into consideration when giving their theoretical analysis. We aim to develop a consensus-based synchronization protocol which could theoretically prove the convergent result under time-varying clock skew. The proposed protocol generally assumes that each physical clock skew has a relatively small change bounded by a constant quantity. By applying EBP, the synchronization error of virtual clock skews can be bounded by a relative small steady state error bound when physical clock skews are gradually changing within certain limits.

2. Higher synchronization accuracy under time-varying clock skew. Our work focuses on improving synchronization accuracy. The comparison between other two consensus-based algorithms indicates that the proposed algorithm could gain better synchronization accuracy especially under time-varying clock skew.

The proposed protocol also deals with random delay case and shows that the convergence of virtual clock skew is in mean square sense. After the clock skew compensation, an asynchronous clock offset compensation protocol is presented. Inspired by [31], a pseudo-synchronous implementation for EBP is presented as synchronous implementation for clock skew compensation is unrealistic in practice. Moreover, as pseudo-synchronous implementation requires no simultaneous action of each sensor node at a global time instant, EBP with a pseudo-synchronous implementation could support both half-duplex and full-duplex systems.

The remainder of this paper is organized as follows. In Section 2, the wireless sensor network model and a time-varying clock skew model for WSNs are introduced as the preliminary knowledge. Section 3 introduces the proportional integral estimator-based protocol. Filtering-based algorithms under both delay-free and random delay cases are presented. Then a proportional integral estimator-based protocol including both clock skew and offset compensation is proposed, where the convergent results are shown in the main theorem and other two corollaries. In Section 4, analysis of pseudo-synchronous implementation is presented for handling the unrealistic synchronous implementation. Simulation results are shown in Section 5. Conclusion of our work and several open problems are given in Section 6. The proof of main theorem is in Appendix.

2. Preliminaries

This section introduces some notations, preliminaries of graph theory, wireless sensor network model and a time-varying clock skew model.

2.1. Notations

\mathbb{R} denotes the set of real numbers and \mathbb{R}^+ denotes the set of positive real numbers. $\mathbf{1}$ represents n -dimensional vector of ones while $\mathbf{0}$ represents vector of zeros with an appropriate dimension. \mathbb{R}^n represents an n -dimensional vector while $\mathbb{R}^{n \times n}$ denotes an $n \times n$ square matrix composed of real numbers. I^n indicates identity matrix with order n while $\mathbf{0}^n$ indicates zero matrix with order n . \mathbb{Z} denotes the set of nonnegative integer numbers.

2.2. Graph theory

A weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a non-empty node set $\mathcal{V} = \{1, 2, \dots, n\}$ and an edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ where an edge of \mathcal{G} is a pair of unordered nodes. The neighbourhood $\mathcal{N}_i \in \mathcal{V}$ of the vertex v_i will be understood as the set $\{v_j \in \mathcal{V} | v_i v_j \in \mathcal{E}\}$, that is, the set of all vertices that are adjacent to v_i . If $v_j \in \mathcal{N}_i$, it follows that $v_i \in \mathcal{N}_j$, since the edge set in a (undirected) graph consists of unordered vertex pairs. d_i denotes the cardinality of \mathcal{N}_i . The notion of adjacency in the graph can be used to move around along the edges of the graph. Thus, a path of length m in \mathcal{G} is given by a sequence of distinct vertices:

$$v_{i_0}, v_{i_1}, \dots, v_{i_m}, \quad (1)$$

such that for $k = 0, 1, \dots, m-1$, the vertices v_{i_k} and $v_{i_{k+1}}$ are adjacent. In this case, v_{i_0} and v_{i_m} are called the end vertices of the path; the vertices $v_{i_1}, \dots, v_{i_{m-1}}$ are the inner vertices.

An undirected graph is called connected if for every pair of vertices in \mathcal{G} , there is a path that has them as its end vertices. Associated with each edge $(i, j) \in \mathcal{E}$ there exists a positive weight ξ_{ij} . For an undirected graph \mathcal{G} , the degree matrix $D(\mathcal{G})$ is defined as follows:

$$D(\mathcal{G})_{ij} = \begin{cases} d_i & i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The adjacency matrix $A(\mathcal{G})$ is defined as follows:

$$A(\mathcal{G})_{ij} = \begin{cases} 1 & i \neq j, j \in \mathcal{N}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The associated Laplacian matrix $L(\mathcal{G})$ is defined as follows:

$$L(\mathcal{G})_{ij} = \begin{cases} -1 & i \neq j, j \in \mathcal{N}_i, \\ 0 & i \neq j, j \notin \mathcal{N}_i, \\ d_i & i = j. \end{cases} \quad (4)$$

In this paper we consider a wireless sensor network model represented by a weighted undirected connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is composed of n sensor nodes and \mathcal{E} stands for contact between two neighbour nodes. Communication delay in WSNs needs to be taken into account since they can be much larger compared with the required synchronization accuracy [5]. We mainly deal with two cases as follows:

1. There are no transmission or computational delays in WSNs. Specifically, the transmission time of node $i \in \mathcal{V}$ and the receiving time of node $j \in \mathcal{N}_i$ are considered to be instantaneous.
2. The communication delays at different time instants are assumed to be positive random variables with constant mean and variance and they are mutually identically and independently distributed.

Finally, two important lemmas are introduced.

Lemma 2.1 ([33]). Consider a block matrix

$$C = \begin{bmatrix} A & -B \\ B & 0 \end{bmatrix} \quad (5)$$

where $A, B \in \mathbb{R}^{n \times n}$ are real, symmetric and positive definite (i.e. A and B have positive real eigenvalues) and A and B can commute ($AB = BA$). Let $\lambda_{A,max}, \lambda_{B,max}$ be the maximum eigenvalues of A and B and $\lambda_{A,min}, \lambda_{B,min}$ be the minimum eigenvalues of A and B . Any eigenvalue of C

satisfies the following bounds:

1. $\text{Re}(\lambda) \geq \min_{\lambda_A} \text{Re} \left[\frac{1}{2} \left(\lambda_A - \sqrt{\lambda_A^2 - 4\lambda_{B,min}^2} \right) \right]$, where $\lambda_A \in \{\lambda_{A,min}, \lambda_{A,max}\}$,
2. $\text{Re}(\lambda) \leq \text{Re} \left[\frac{1}{2} \left(\lambda_{A,max} + \sqrt{\lambda_{A,max}^2 - 4\lambda_{B,min}^2} \right) \right]$, $\text{Im}(\lambda) \geq \left| \text{Im} \left[\frac{1}{2} \left(\lambda_A - \sqrt{\lambda_A^2 - 4\lambda_{B,max}^2} \right) \right] \right|$

Lemma 2.2 (Input-to-state stability [34]). For the linear time-invariant system $x(k+1) := Ax(k) + Bu(k)$ with a Schur-stable matrix A , the zero-input response decays to zero exponentially fast, while the zero-state response is bounded for every bounded input.

2.3. Clock model

Environmental conditions such as temperature, pressure, or even humidity may affect the behaviour of the oscillator, causing clock skews to speed up or slow down gradually. A time-varying clock skew model is presented in [13] as

$$\tau_i(t) = \int_0^t \alpha_i(t') dt' + \beta_i, \quad \tau_i(0) = \beta_i, \quad (6)$$

where $\tau_i(t)$ is the local clock reading of node i ; $\alpha_i(t)$ is the physical clock skew which determines the local clock speed; β_i is the physical clock offset and t indicates absolute reference time. For each node i at t , it cannot acquire its physical clock skew $\alpha_i(t)$. The only information known by node i is $\tau_i(t)$. Explicit notation of t indicates $\alpha_i(t)$ is a slowly time-varying variable. Suppose this slow change satisfies the following assumption.

Assumption 2.1. Slow change of physical clock skew ensures the uniform boundedness of $\alpha_i(t)$ at any time instant t

$$1 - \rho_1 \leq \alpha_i(t) \leq 1 + \rho_1, \quad (7)$$

where $0 < \rho_1 \ll 1$ indicates the maximum drift. Crystal oscillators used in sensor nodes normally have a drift between 30 and 100 ppm.

As the synchronization protocol is in discrete-time form, we denote one sampling period as T . For the sake of simplicity, we express $\alpha_i(kT) = \alpha_i(k)$, $\forall k \in \mathbb{Z}$, i.e., we assume $T=1$. Then one-sampling-period drift during $[kT, (k+1)T]$ for node i is defined as

$$\Delta\alpha_i(k) = \alpha_i(k+1) - \alpha_i(k), \quad k \in \mathbb{Z}. \quad (8)$$

Another assumption concerning with the change rate of $\alpha_i(t)$ is made as follows.

Assumption 2.2. Slow change also ensures the uniform boundedness of $\Delta\alpha_i(k)$ during any sampling period $[kT, (k+1)T]$

$$|\Delta\alpha_i(k)| \leq \rho_2, \quad (9)$$

where $0 < \rho_2 \ll 1$ is the bound on the change of i 's clock skew in one sampling period.

The objective of a distributed clock synchronization protocol is to synchronize all the nodes with respect to a common virtual reference clock as close as possible, namely

$$\bar{\tau}(t) = \int_0^t \bar{\alpha}(t') dt' + \bar{\beta}, \quad (10)$$

where $\bar{\alpha}(t) = \frac{\sum_{i=1}^n \alpha_i(t)}{n}$ is the average value of clock skew at time t .

Every local clock i keeps an update of its virtual clock reading $\hat{\tau}_i(t)$ as follows:

$$\hat{\tau}_i(t) = F_i(\tau_i(t), \tau_j(t), j \in \mathcal{N}_i), \quad (11)$$

where $F_i(\cdot)$ is a compensator depending on the information available at node i and its neighbour nodes $j \in \mathcal{N}_i$. More specifically, we consider $F_i(\cdot)$ as a linear updating rule, that is

$$\hat{\tau}_i(t) = G_i(\hat{\alpha}_i(t)\tau_i(t), \tau_j(t), j \in \mathcal{N}_i), \quad (12)$$

where $G_i(\cdot)$ is a linear function. $\hat{\alpha}_i(t)$ is the virtual clock skew compensation quantity based on the information available at node i and node $j \in \mathcal{N}_i$ and by multiplying $\hat{\alpha}_i(t)$ with physical clock reading $\tau_i(t)$, it aims to reduce synchronization error of local clock skews.

If $\alpha_i(t) = \alpha_i, \forall i \in \mathcal{V}$, that is, all α_i 's are constant quantities, then $\bar{\alpha}(t) = \bar{\alpha}$. In this case the virtual clock skew and clock reading for node i should asymptotically track $\bar{\alpha}$ and $\bar{\tau}(t)$,

$$\lim_{t \rightarrow \infty} \hat{\alpha}_i(t)\alpha_i = \bar{\alpha}, \forall i \in \mathcal{V}, \quad \lim_{t \rightarrow \infty} (\hat{\tau}_i(t) - \bar{\tau}(t)) = 0, \forall i \in \mathcal{V}. \quad (13)$$

Define the indicator variable that measures the accuracy of skew compensation: $\varepsilon_i(t) = \hat{\alpha}_i(t)\alpha_i(t) - \bar{\alpha}(t)$. If $\alpha_i(t)$ is a slowly time-varying variable, the objective is to bound the synchronization error for virtual clock skew of node i as follows:

$$\lim_{t \rightarrow \infty} \varepsilon_i(t) = \lim_{t \rightarrow \infty} |\hat{\alpha}_i(t)\alpha_i(t) - \bar{\alpha}(t)| \leq \rho_3, \quad (14)$$

where $\rho_3 < \rho_1$ can be regarded as the synchronization accuracy of clock skew compensation and ρ_3 needs to be made as small as possible.

3. Proportional integral estimator-based clock synchronization protocol

The proposed distributed protocol includes three parts which are similar to the ones proposed in [23]: relative clock skew estimation, clock skew compensation and offset compensation. The proposed protocol uses the same components as [23] but deploys different approaches compared with [23]. The main contribution is to present a new synchronization protocol under time-varying clock skews.

3.1. Relative clock skew estimation

Relative clock skew estimation algorithm aims to estimate the relative clock skew of each node i with respect to its neighbour node $j \in \mathcal{N}_i$. The estimation value of relative clock skew will be used to develop a clock skew compensation protocol.

Definition 3.1. The definition of relative clock skew at time t is as follows:

$$\alpha_{ij}(t) = \frac{\alpha_j(t)}{\alpha_i(t)}, \quad i, j \in \mathcal{V}. \quad (15)$$

The estimation of relative clock skew will be discussed in two cases.

3.1.1. Relative clock skew estimation in delay-free case

Some notations are listed as follows:

1. $t_j(k)$ indicates global time at which node j 's clock reading $\tau_j(t_j(k))$ just reaches kT .
2. $\tau_i(t_j(k)) (k \in \mathbb{Z}, \forall j \in \mathcal{N}_i)$ indicates node i 's local clock reading when node j announces that its local clock reading just reaches kT .

In delay-free case, if we take unavoidable measurement, quantization errors and small drift of clock skews into

consideration, a low-pass filter-based algorithm introduced by [23] is proposed in Algorithm 1.

In Algorithm 1, $\rho \in (0, 1)$ is a tuning parameter. The choice of ρ can be treated as a tradeoff between the precision of new measurement and prior estimation based on the old measurement. t^+ and t^- represent, respectively, the right-hand limit and left-hand limit of t . According to [23], for the case of constant clock skew α_i , applying Algorithm 1 yields the following convergent result

$$\lim_{t \rightarrow \infty} \hat{\alpha}_{ij}(t) = \alpha_{ij}. \quad (16)$$

Algorithm 1. Relative clock skew estimation in delay-free case.

Input: $\tau_i(t_j(k)), \tau_i(t_j(k-1))$ for $i \in \mathcal{V}$.

Output: $\hat{\alpha}_{ij}(t^+)$ for $i, j \in \mathcal{V}$.

- 1: Initialize $\hat{\alpha}_{ij}(0^-) = 1$.
- 2: Set a common broadcast T to each node.
- 3: **while** 1 **do**
- 4: Update $\hat{\alpha}_{ij}(t^+) \leftarrow \rho \hat{\alpha}_{ij}(t^-) + (1-\rho) \frac{T}{\tau_i(t_j(k)) - \tau_i(t_j(k-1))}$ at $t = t_j(k), k \in \mathbb{Z}$.
- 5: Update $\hat{\alpha}_{ij}((t+1)^-) \leftarrow \hat{\alpha}_{ij}(t^+)$.
- 6: **end while**

3.1.2. Relative clock skew estimation in random delay case

Some notations are listed as follows:

1. $t_j(k)$ is the real broadcasting time at which node j 's clock reading $\tau_j(t_j(k))$ just reaches kT . At $t_j(k)$, node j broadcasts its hardware clock reading $\tau_j(t_j(k))$ to node i .
2. $t_j(k)$ indicates the real receiving time for node i . At $t_j(k)$, node i receives packets from node j and immediately records its hardware clock reading $\tau_i(t_j(k))$.
3. $d_k = t_j(k) - t_j'(k), k \in \mathbb{Z}$, is the communication delay from node j to node i . For different d_k 's, they are mutually independent of each other.

In random delay case, if we take unavoidable measurement, quantization errors and small drift of clock skews into consideration, a low-pass filter-based algorithm introduced by [25] is presented in Algorithm 2.

Algorithm 2. Relative clock skew estimation in random delay case.

Input: $\tau_i(t_j(k)), \tau_i(t_j(k-1))$ for $i \in \mathcal{V}$.

Output: $\hat{\alpha}_{ij}(t^+)$ for $i, j \in \mathcal{V}$.

- 1: Initialize $\hat{\alpha}_{ij}(0) = 1$.
- 2: Set a common broadcast T to each node.
- 3: **while** 1 **do**
- 4: $\beta_{ij}(t) \leftarrow \frac{T}{\tau_i(t_j(k)) - \tau_i(t_j(k-1))}$ at $t = t_j(k)$.
- 5: $\hat{\alpha}_{ij}(t^+) \leftarrow \frac{\beta_{ij}(t) + (k-1)\hat{\alpha}_{ij}(t^+)}{k}$, $k \in \mathbb{Z}$ at $t = t_j(k)$.
- 6: **end while**

According to [25], for the case of constant clock skew α_i , applying Algorithm 2 yields the following mean square convergent

result

$$E\{\hat{\alpha}_{ij}(k)\} = \alpha_{ij} \quad (17)$$

and

$$\lim_{k \rightarrow \infty} \text{Var}\{\hat{\alpha}_{ij}(k)\} = 0. \quad (18)$$

3.2. Clock skew compensation

We apply the relative clock skew estimation [Algorithms 1 and 2](#) which could guarantee the convergence result under constant input of clock skew α_i . Under slowly time-varying input of clock skew $\alpha_i(t)$, $\hat{\alpha}_{ij}(t)$ is also a slowly time-varying variable. Considering the slow varying properties, we will study the convergence property of skew compensation using the same algorithm.

After $\hat{\alpha}_{ij}(t^+)$ is acquired, every node i uses a distributed updating protocol to achieve clock skew compensation, which bounds the error $\varepsilon_i(t)$ under slowly changing input $\alpha_i(t)$. At time instant $t = t_j(k)$, $k \in \mathbb{Z}$, each node i updates its clock skew compensation quantity $\hat{\alpha}_i(t)$. The virtual clock skew is compensated by multiplying $\hat{\alpha}_i(t)$ with its physical clock reading $\tau_i(t)$ as $\hat{\alpha}_i(t)\tau_i(t)$.

The proportional integral estimator-based protocol is proposed as follows:

Initialization: $\hat{\alpha}_i(0) = 1$, $\omega_i(0) = 0$, $\forall i \in \mathcal{V}$.

Main Loop:

$$\begin{cases} \hat{\alpha}_i(t^+) = (1 - \epsilon\gamma)\hat{\alpha}_i(t^-) + \epsilon K_I \sum_{j \in \mathcal{N}_i} (\omega_j(t^-) - \omega_j(t^-)\hat{\alpha}_{ij}(t^+)) + \epsilon\gamma \\ \quad - \epsilon K_P \sum_{j \in \mathcal{N}_i} (\hat{\alpha}_i(t^-) - \hat{\alpha}_j(t^-)\hat{\alpha}_{ij}(t^-)), \\ \omega_i(t^+) = \omega_i(t^-) - \epsilon K_I \sum_{j \in \mathcal{N}_i} (\hat{\alpha}_i(t^-) - \hat{\alpha}_j(t^-)\hat{\alpha}_{ij}(t^+)), \end{cases} \quad (19)$$

where ω_i is an internal estimator state which acts as an integrator. Specifically, $\omega_i(t^+)$ accumulates the difference of $(\hat{\alpha}_i(t^-) - \hat{\alpha}_j(t^-)\hat{\alpha}_{ij}(t^-))$ with its neighbours' $j \in \mathcal{N}_i$. $\hat{\alpha}_i$ is called estimator state which tries to estimate the virtual clock skew compensation quantity such that $\hat{\alpha}_i(t^+)\alpha_i(t^+)$ asymptotically approaches $\frac{\sum_{i=1}^n \hat{\alpha}_i(t^+)\alpha_i(t^+)}{n}$ as close as possible when t goes to infinity. $K_p, K_I > 0$ are called estimator gains; $\gamma > 0$ is the information rate as it depicts the proportion of how much new information is introduced; $\epsilon > 0$ is the step size. Each node i not only communicates its physical clock reading $\tau_i(t^-)$ but also communicates its clock compensation quantity $\hat{\alpha}_i(t^-)$ and internal estimator state $\omega_i(t^-)$ with its neighbours $j \in \mathcal{N}_i$.

This protocol is inspired by the continuous form of proportional–integral dynamic average consensus estimator [\[33,35,36\]](#) which can allow each node to approximately track the average of the slowly time-varying inputs and bound the estimation error related to the rate of slow change of inputs under a fixed network. We indirectly use the discrete form of proportional integral dynamic average consensus estimator as each node i cannot access its clock skew $\alpha_i(t_j(k))$ while it can only use relative clock skew estimation $\hat{\alpha}_{ij}(t^+)$ calculated in [Algorithms 1 and 2](#) respectively.

As $t_j(k)$ is based on node j 's measurement of kT , each node i receives $\hat{\alpha}_{ij}(t_j(k))$ and performs (19) at different time instants before synchronization has finished. However, here we only consider the synchronous form of (19) as in the implementation part, we show that under pseudo-synchronous implementation scheme, the protocol has exactly the same performance as synchronous one does. By replacing t^+ and t^- with a common time notation $t(k+1)$ and $t(k)$ from a perspective of global clock, the following

equations hold:

$$\begin{aligned} \hat{\alpha}_i(k+1) &\triangleq \hat{\alpha}(t_j(k)^+), \quad \hat{\omega}_i(k+1) \triangleq \hat{\omega}(t_j(k)^+), \\ \hat{\alpha}_i(k) &\triangleq \hat{\alpha}(t_j(k)^-), \quad \hat{\omega}_i(k) \triangleq \hat{\omega}(t_j(k)^-), \quad \hat{\alpha}_{ij}(k) \triangleq \hat{\alpha}_{ij}(t^-). \end{aligned} \quad (20)$$

The synchronous form of (19) is presented as follows:

Synchronous form:

$$\begin{cases} \hat{\alpha}_i(k+1) = \hat{\alpha}_i(k) + \epsilon K_I \sum_{j \in \mathcal{N}_i} (\omega_j(k) - \omega_j(k)\hat{\alpha}_{ij}(k)) + \epsilon\gamma(1 - \hat{\alpha}_i(k)) \\ \quad - \epsilon K_P \sum_{j \in \mathcal{N}_i} (\hat{\alpha}_i(k) - \hat{\alpha}_j(k)\hat{\alpha}_{ij}(k)), \\ \omega_i(k+1) = \omega_i(k) - \epsilon K_I \sum_{j \in \mathcal{N}_i} (\hat{\alpha}_i(k) - \hat{\alpha}_j(k)\hat{\alpha}_{ij}(k)). \end{cases} \quad (21)$$

The indicator that measures the accuracy of node i 's local clock skew is as follows:

$$\varepsilon_i(k) = \hat{\alpha}_i(k)\alpha_i(k) - \frac{\sum_{i=1}^n \alpha_i(k)}{n} = \bar{\alpha}_i(k) - \frac{\sum_{i=1}^n \alpha_i(k)}{n}. \quad (22)$$

If we let

$$\begin{aligned} \mathbf{e}^\alpha(k) &= [\varepsilon_1(k), \dots, \varepsilon_n(k)]^T, \quad \bar{\boldsymbol{\alpha}}(k) = [\bar{\alpha}_1(k), \dots, \bar{\alpha}_n(k)]^T, \\ \hat{\boldsymbol{\alpha}}(k) &= [\hat{\alpha}_1(k), \dots, \hat{\alpha}_n(k)]^T, \quad \boldsymbol{\omega}(k) = [\omega_1(k), \dots, \omega_n(k)]^T, \end{aligned} \quad (23)$$

then (21) can be written collectively as

$$\begin{cases} \hat{\boldsymbol{\alpha}}(k+1) = \hat{\boldsymbol{\alpha}}(k) + \epsilon K_I (D_I - \Lambda(k))\boldsymbol{\omega}(k) + \epsilon\gamma(1 - \hat{\boldsymbol{\alpha}}(k)) \\ \quad - \epsilon K_P (D_P - \Lambda(k))\hat{\boldsymbol{\alpha}}(k), \\ \boldsymbol{\omega}(k+1) = \boldsymbol{\omega}(k) - \epsilon K_I (D_I - \Lambda(k))\hat{\boldsymbol{\alpha}}(k). \end{cases} \quad (24)$$

where $\Lambda(k)$ is defined as follows:

$$\Lambda(k)_{ij} = \begin{cases} \hat{\alpha}_{ij}(k) & i \neq j, j \in \mathcal{N}_i, \\ 0 & \text{otherwise.} \end{cases} \quad (25)$$

And the error dynamics can be written collectively as

$$\mathbf{e}^\alpha(k) = \bar{\boldsymbol{\alpha}}(k) - \frac{\mathbf{1}\mathbf{1}^T}{n}\boldsymbol{\alpha}(k). \quad (26)$$

Our main result is presented as [Theorem 3.1](#). It shows that the updating rule will lead to the boundedness of $\|\lim_{k \rightarrow \infty} \varepsilon_i(k)\|$.

Theorem 3.1. Consider the communication topology of WSNs of n sensors being a connected undirected weighted graph \mathcal{G} with its Laplacian matrix L where λ_2 and λ_{\max} correspond to the second smallest eigenvalue and the largest eigenvalue of L respectively. Each node i implements discrete-time synchronous form of (21) with proportional and integral gains $K_p = \kappa K'_p$, $K_I = \kappa K'_I$, $\gamma = \kappa\gamma'$ where $K'_p, K'_I, \gamma' > 0$. If Assumptions 2.1, 2.2 hold, $\varepsilon_i(k)$ converges exponentially to a ball at the origin of radius $\frac{\epsilon\gamma\rho_2(1-\rho_1+\rho_2)(1-\rho_1)}{\epsilon^2\gamma'^2(1-\rho_1)^2 - 2\epsilon\gamma\rho_2(1-\rho_1) + \rho_2^2}$

+ $\frac{\gamma'\rho_2(1-\rho_1+\rho_2)\|\mathbf{V}^{-1}\|}{(2-2\rho_1)K'_I\lambda_2(L)}$ as $k \rightarrow \infty$ under the following control parameter constraints:

1. $\frac{\rho_2}{1-\rho_1} < \epsilon\gamma$,
2. $\frac{\rho_2}{\beta_1(1-\rho_1)} < \epsilon < \frac{2-\rho_2-2\rho_1}{\beta_2(1-\rho_1)}$,

where $\beta_1, \beta_2, \|\mathbf{V}^{-1}\|$ are global information of the network defined in the proof.

Proof. The proof is given in [Appendix](#). \square

Theoretically, in the proof of [Theorem 3.1](#), the proposed protocol (24) has been transferred into a discrete-time-invariant linear system. Once the conditions of control parameters are satisfied, the internal stability theorem would guarantee the asymptotic exponential convergence of (24) as all eigenvalues of system matrix are assigned into the unit circle. As the system input (physical clock skew) is a slow time-varying variable, finite-time convergence is unrealistic.

For constant clock skew α_i , the following corollary is established.

Corollary 3.1. Consider the communication topology of WSNs represented by a weighted connected and undirected graph \mathcal{G} . Each sensor implements the synchronous protocol (21) with proportional and integral gains $K_p = \kappa K'_p > 0$, $K_i = \kappa K'_i > 0$, $\gamma = \kappa \gamma' > 0$. If each local clock skew α_i is constant, the virtual clock skew $\bar{\alpha}_i(k)$ converges to the average consensus value of $\frac{\sum_{i=1}^n \alpha_i(0)}{n}$ and $\mathbf{e}^a(k)$ converges exponentially to zero as $k \rightarrow \infty$ under the following control parameter constraints:

1. $\frac{\rho_2}{1 - \rho_1} < \epsilon \gamma$,
2. $\frac{\rho_2}{\beta_1(1 - \rho_1)} < \epsilon < \frac{2 - \rho_2 - 2\rho_1}{\rho_2(1 - \rho_1)}$.

The proof of [Corollary 3.1](#) follows directly from [Theorem 3.1](#).

For random delay case, the following corollary is established.

Corollary 3.2. Consider the communication topology of WSNs represented by a weighted connected and undirected graph \mathcal{G} . Each sensor implements the synchronous protocol (21) with proportional and integral gains $K_p = \kappa K'_p > 0$, $K_i = \kappa K'_i > 0$, $\gamma = \kappa \gamma' > 0$. If we take [Assumptions 2.1 and 2.2](#) and random communication delay into consideration, $\mathbf{e}_i(k)$ converges exponentially to a ball at the origin of radius in mean square sense:

1. $\rho_3 = \frac{\epsilon \gamma \rho_2 (1 - \rho_1 + \rho_2) (1 - \rho_1)}{\epsilon^2 \gamma^2 (1 - \rho_1)^2 - 2 \epsilon \gamma \rho_2 (1 - \rho_1) + \rho_2^2} + \frac{\gamma' \rho_2 (1 - \rho_1 + \rho_2) \|V^{-1}\|}{(2 - 2\rho_1) K_i \lambda_2(L_1)}$,
2. $\lim_{k \rightarrow \infty} E\{|\mathbf{e}_i(k)|\} \leq \rho_3$,
3. $\lim_{k \rightarrow \infty} \text{Var}\{|\mathbf{e}_i(k)|\} = 0$,

under the following control parameter constraints:

1. $\frac{\rho_2}{1 - \rho_1} < \epsilon \gamma$,
2. $\frac{\rho_2}{\beta_1(1 - \rho_1)} < \epsilon < \frac{2 - \rho_2 - 2\rho_1}{\rho_2(1 - \rho_1)}$.

The proof of [Corollary 3.2](#) follows directly from [Theorem 3.1](#), (17) and (18).

3.3. Offset compensation

At the end of clock skew compensation procedure, all virtual clock skews in WSNs have achieved average consensus value, i.e., they will run at the same speed $\bar{\alpha}$ under constant input α_i 's or their synchronization errors will be bounded by a relative small steady error $\rho_3 = \frac{\epsilon \gamma \rho_2 \rho(k)}{(\rho(k) - 1 + \epsilon \gamma)^2} + \frac{\gamma \rho_2 \rho(k) \|V^{-1}\|}{(\rho(k) - 1 - \epsilon \hat{\alpha}_{i+n-1}) K_i \lambda_2(L_1)}$ under time-

varying input $\alpha_i(k)$'s. Hence, it is necessary to compensate for possible offset errors. We present the offset compensation protocol as follows.

Initialization: $\hat{\tau}_i(0) = \hat{\alpha}_i(0) \tau_i(0) \quad \forall i \in \mathcal{V}$.

Main loop:

$$\begin{cases} \hat{\tau}_i(t^+) = \frac{\gamma_i \hat{\tau}_i(t^-) + \gamma_j \hat{\tau}_j(t^-)}{\gamma_i + \gamma_j}, \\ \gamma_i = \gamma_i + 1, \quad i \in \mathcal{V}, i \in \mathcal{N}_j, \\ \hat{\tau}_i((t+1)^-) = \tau_i(t^+) + \hat{\alpha}_i(t^+) (\tau_i(t+1)^-) - \tau_i(t^+), \end{cases} \quad (27)$$

where γ_i is the confidence parameter. $\hat{\alpha}_i(t^+)$ is acquired from (19).

4. Implementation

4.1. Implementation of clock skew compensation protocol

As shown in our proof of skew compensation, the synchronous form (21) is applied. However, it is unrealistic to guarantee the simultaneous actions of transmitting, receiving and updating at the same time before local clock skews are synchronized. As a result, synchronous implementation is impractical. To tackle this, we present a pseudo-synchronous implementation. This idea is triggered by the fact that although each local clock has different local clock readings, the difference between either their clock skews or offsets is bounded. Different from asynchronous implementation where node i updates its information immediately once receives only one packet from one of its neighbours $j \in \mathcal{N}_i$, in pseudo-synchronous implementation, node i does not update its states until it receives messages sent by all its neighbours. On the other hand, pseudo means the transmission and updating instants are determined by its local information, including its relative clock with its neighbours while synchronous implementation requires each local sensor share a common global clock, that is, a piece of global information. We first specify the transmission and updating time instants.

The transmission time instants of $i \in \mathcal{V}$ are defined by

$$t_{tr}^i(k) = t, \quad \text{where } \tau_i(t) = kT. \quad (28)$$

T is a positive parameter set as a default value known by all sensor nodes. Namely, $t_{tr}^i(k)$ indicates that the i th clock reading just reaches hT .

The receiving time instants of $j \in \mathcal{N}_i$ are defined by

$$t_{re}^j(k) = t_{tr}^i(k), \quad \forall j \in \mathcal{N}_i. \quad (29)$$

The updating time instants of node i are defined by

$$t_{up}^i(k) = \max\{t_{tr}^j(k) \mid j \in \mathcal{N}_i \cup \{i\}\}. \quad (30)$$

namely the i th clock updates its state right after all its neighbour nodes finish their transmission actions, included its own transmission. Notice that, from the above definitions, $t_{up}^i(k) \geq t_{tr}^i(k)$. Moreover, $t_{tr}^i(k)$ and $t_{up}^i(k)$ can be determined by node i relying only on its local information. Another important concept is defined as

$$t(k) := \min_i t_{tr}^i(k), \quad \forall i \in \mathcal{V}. \quad (31)$$

The pseudo-synchronous implementation of clock skew compensation is presented in [Algorithm 3](#).

Algorithm 3. Pseudo-synchronous implementation of skew compensation.

- 1: Initialize $\hat{\alpha}_i(0) = 1, \omega_i(0) = 0, \forall i \in \mathcal{V}$. Set a common broadcast period T to each node.
- 2: At time instants $t_{tr}^i(k)$ s node i broadcasts $\omega_i(t_{tr}^i(k))$ and $\hat{\alpha}_i(t_{tr}^i(k))$ to its neighbours $o \in \mathcal{N}_i$.
- 3: Node i receives a packet containing $\omega_j(t_{tr}^j(k))$ and $\hat{\alpha}_j(t_{tr}^j(k))$ from its neighbours $j \in \mathcal{N}_i$ at time instant $t_{tr}^j(k)$ and computes $\hat{\alpha}_{ij}(t_{tr}^j(k))$ according to Algorithm 1 and Algorithm 2.
- 4: At time instant $t_{up}^i(k)$, node i updates both $\omega_i(t_{up}^i(k))$ and $\hat{\alpha}_i(t_{up}^i(k))$ using its stored information as follows:

$$\begin{aligned} \hat{\alpha}_i(t_{up}^i(k)) &\Leftarrow \hat{\alpha}_i(t_{up}^i(k)) + \epsilon K_I \sum_{j \in \mathcal{N}_i} (\omega_i(t_{tr}^j(k)) - \omega_j(t_{tr}^j(k))) \hat{\alpha}_{ij}(t_{tr}^j(k)) \\ &\quad - \epsilon K_P \sum_{j \in \mathcal{N}_i} (\alpha_i(t_{tr}^j(k)) - \alpha_j(t_{tr}^j(k))) \hat{\alpha}_{ij}(t_{tr}^j(k)) + \epsilon \gamma (1 - \hat{\alpha}_i(t_{up}^i(k))), \quad (32) \\ \omega_i(t_{up}^i(k)) &\Leftarrow \omega_i(t_{up}^i(k)) - \epsilon K_I \sum_{j \in \mathcal{N}_i} (\hat{\alpha}_i(t_{tr}^j(k)) - \hat{\alpha}_j(t_{tr}^j(k))) \hat{\alpha}_{ij}(t_{tr}^j(k)). \end{aligned}$$

- 5: During $t \in (t_{up}^i(k), t_{up}^i(k+1)]$, the states of node $i \in \mathcal{V}$ remain constant

$$\hat{\alpha}_i(t) = \hat{\alpha}_i(t_{up}^i(k)), \omega_i(t) = \omega_i(t_{up}^i(k)). \quad (33)$$

- 6: When time instant is at $t_{tr}^j(k+1)$, go to step 3.

The performance of pseudo-synchronous implementation with clock skew compensation is illustrated in [Theorem 4.1](#). Before that, the following assumption needs to be guaranteed.

Assumption 4.1. To guarantee the performance of pseudo-synchronous implementation, $t_{up}^i(k)$, $t(k+1)$ should satisfy the following inequality

$$t_{up}^i(k) \leq t(k+1), \quad \forall i \in \mathcal{V}. \quad (34)$$

Remark 4.1. Another interpretation of (34) is

$$t_1 \leq t_2, \quad \text{where } t_1: \max_i \tau_i(t_1) = kT, \quad t_2: \min_i \tau_i(t_2) = (k+1)T, \quad \forall i \in \mathcal{V}. \quad (35)$$

According to [Assumption 2.1](#), physical clock skews are restricted by a certain bound. As long as there exists a slight difference between α_{min} and α_{max} , this assumption is rather rational.

Theorem 4.1. The states of the pseudo-synchronous protocol (32) evolve according to the linear discrete-time equation (21), which is exactly equal to the synchronous protocol.

Proof. In order to analyze the pseudo-synchronous protocol (32), we need to fix the sampling time instants. As $t(k)$ represents the first time instant in which the local clock reading of a node reaches the value kT , according to the definitions given in (28), (30) and

considering [Assumption 4.1](#), the following inequalities hold

$$\begin{aligned} t_{up}^i(k) &\leq t(k+1) \leq t_{up}^i(k+1), \quad t_{up}^i(k-1) \leq t(k) \leq t_{up}^i(k), \\ t_{up}^i(k-1) &\leq t(k) \leq t_{tr}^i(k) \leq t_{up}^i(k), \\ t_{up}^i(k-1) &\leq t(k) \leq t_{tr}^j(k) \leq t_{up}^i(k), \quad \forall j \in \mathcal{N}_i \cup \{i\}. \end{aligned} \quad (36)$$

According to (33) we can deduce that

$$\begin{aligned} \hat{\alpha}_i(t_{up}^i(k)) &= \hat{\alpha}_i(t(k)), \quad \omega_i(t_{up}^i(k)) = \omega_i(t(k)), \\ \hat{\alpha}_j(t_{tr}^j(k)) &= \hat{\alpha}_j(t(k)), \quad \omega_j(t_{up}^j(k)) = \omega_j(t(k)). \end{aligned} \quad (37)$$

Substituting (37) into (32), it follows that $\forall i \in \mathcal{V}$

$$\begin{cases} \hat{\alpha}_i(t(k+1)) = \epsilon K_I \sum_{j \in \mathcal{N}_i} (\omega_i(t(k)) - \omega_j(t(k))) \hat{\alpha}_{ij}(t(k)) + \epsilon \gamma (1 - \hat{\alpha}_i(t(k))) \\ \quad - \epsilon K_P \sum_{j \in \mathcal{N}_i} (\alpha_i(t(k)) - \alpha_j(t(k))) \hat{\alpha}_{ij}(t(k)) + \hat{\alpha}_i(t(k)), \\ \omega_i(t(k+1)) = \omega_i(t(k)) - \epsilon K_I \sum_{j \in \mathcal{N}_i} (\hat{\alpha}_i(t(k)) - \hat{\alpha}_j(t(k))) \hat{\alpha}_{ij}(t(k)). \end{cases} \quad (38)$$

As $t(k)$ is the common time reference, each local node renews its states exactly at the same reference time. This completes the proof of [Theorem 4.1](#). \square

Remark 4.2. The proposed proportional integral estimator-based protocol applies a pseudo-synchronous implementation. With

pseudo-synchronous implementation, each pair of neighbouring nodes in WSNs communicate with each other according to their own clocks and require no simultaneous actions of transmitting, receiving and updating at a common global time instant. As a result, EBP with pseudo-synchronous implementation applies to both full-duplex and half-duplex communication radio technology in wireless communications.

5. Simulation results

This section provides two numerical examples. In Example 1, comparison is done between our proposed protocol and two other existing state of the art algorithms: the second-order linear consensus algorithm (SLCA) [31] and the maximum time synchronization (MTS) [25] protocol. Example 2 illustrates the suitability of large WSNs.

5.1. Example 1: Comparison between EBP and two other consensus-based protocols

Consider a network topology in Fig. 1 composed of 15 labelled nodes.

For EBP, the control parameters are selected as follows: $\hat{\alpha}_i(0) = 1, \hat{\omega}_i(0) = 0, \epsilon = 0.2, \gamma = 0.09, K_i = 0.75, K_p = 1.65$. As crystal oscillators normally have a drift ranged from 30 to 100 ppm, where an oscillator with 100 ppm drifts apart $100 \mu\text{s}$ in one second. Therefore the initial clock skew α_i 's are randomly selected from $[0.9999, 1.0001]$ and clock offset β_i from $[0, 0.0002]$ s. As each local clock skew experiences small drift, during one sampling period the local clock skew is added by a random noise range from $[-0.00000915, 0.00000915]$, corresponding to $[-0.03, 0.03]$ ticks. It can be seen from Fig. 2 that it takes nearly 250 iterations to reduce the maximum difference of skew below 0.1 ticks ($1 \text{ tick} = 1/32768 \text{ Hz} = 30.5 \mu\text{s}$). Fig. 3 shows the convergent properties of offset compensation.

The comparison between EBP and other two consensus-based algorithms (SLCA, MTS) are made from the following aspects:

1. *Computational complexity*: We assess the number of floating point operations required for a global synchronization of a wireless sensor network.
2. *Synchronization accuracy*: We compare synchronization accuracy for each algorithm under time-varying clock skew.
3. *Convergent speed*: We plot convergent curves of each algorithm in one figure to do explicit comparison of convergent speed.

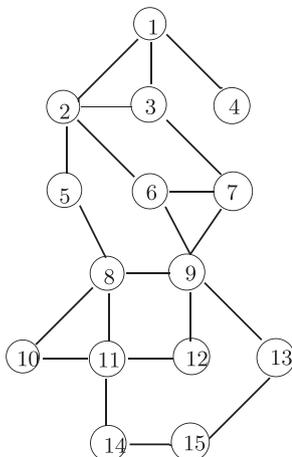


Fig. 1. Network topology composed of 15 labelled nodes.

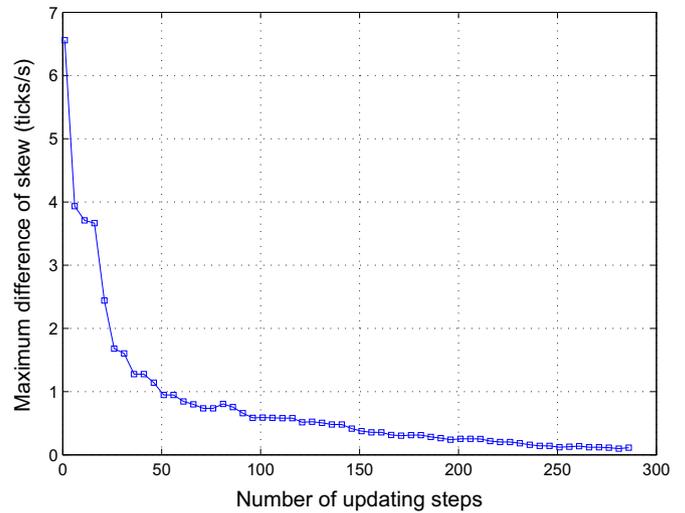


Fig. 2. Convergent performance of skew compensation for EBP.

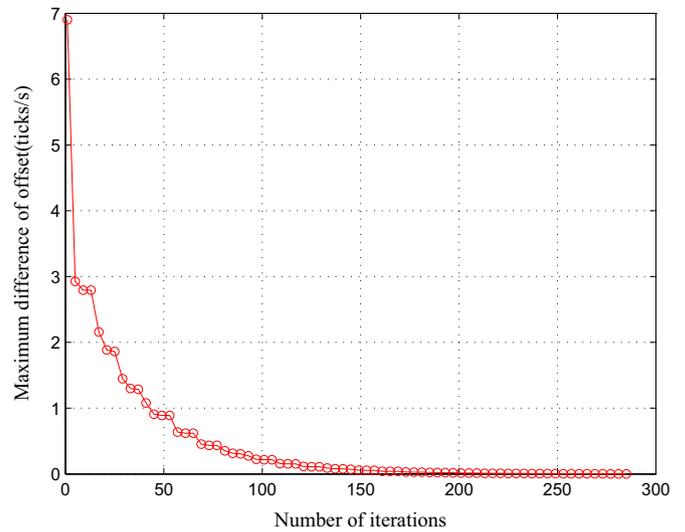


Fig. 3. Convergent performance of offset compensation for EBP.

Table 1

Computational complexity between EBP and other two algorithms with $\rho_2 = 0.03$ ticks/s.

Algorithm	EBP	MTS	SLCA
Number of iterations required for convergence	130	20	200
Number of operations in one iteration	18	12	13
Total number of operations	2340	240	2600

5.1.1. Computational complexity

Comparison in computational complexity is done between the proposed algorithm and other two consensus-based algorithms. In particular, the comparison standard is based on how many number of floating point operations are required for each algorithm after one-round synchronization with $\rho_2 = 0.03$ ticks/s.

Table 1 shows that the proposed protocol requires more floating point operations than the standard fast-convergent-oriented MTS algorithm but requires fewer operations than SLCA.

5.1.2. Synchronization accuracy

We have compared the synchronization accuracy among the proposed protocol, MTS and SLCA under time-varying clock skews. Table 2 contains statistics of the maximum synchronization error

Table 2
Synchronization accuracy between EBP and other two algorithms with $\rho_2 = 0.03$ ticks/s after convergence.

Algorithm	EBP	MTS	SLCA
Maximum synchronization error (ticks/s)	0.0364	0.2055	0.0578
Average synchronization error (ticks/s)	0.0511	0.1751	0.0752

Table 3
Synchronization accuracy between EBP and other two algorithms with $\rho_2 = 0.03$ ticks/s after convergence.

Algorithm	EBP	MTS	SLCA
Number of iterations required for convergence	130	20	200

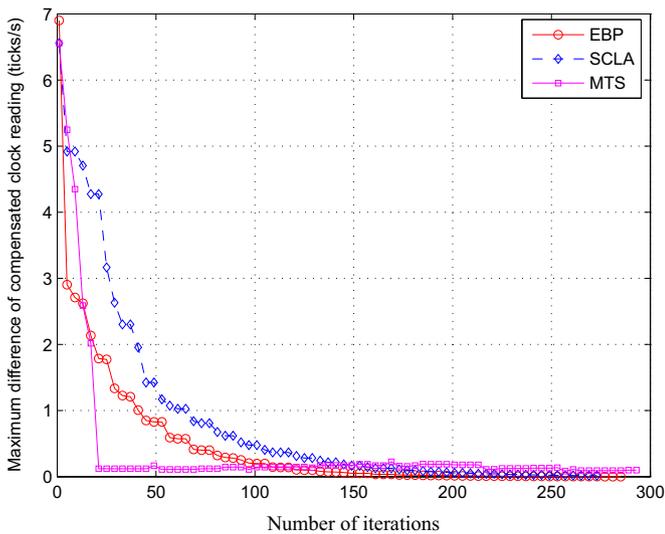


Fig. 4. Comparison in convergent speed among EBP, MTS, SLCA.

and average synchronization error for each algorithm.

Drawn from Table 2, the maximum synchronization error and the average synchronization error for EBP in this static network are 0.0364 and 0.0511 ticks/s respectively, compared with 0.1751 ticks/s and 0.2055 ticks/s of MTS and 0.0578 and 0.0752 ticks/s of

SLCA. This comparison indicates that EBP has the best performance in synchronization accuracy than other two algorithms when the clock skew is time-varying.

5.1.3. Convergent speed

Comparison in convergent speed is done between the proposed algorithm and other two state of the art consensus-based algorithm. Specifically, the comparison standard is based on how many iterations are required for each algorithm after one-round synchronization with $\rho_2 = 0.03$ ticks/s. Table 3 shows the comparison result.

We have now plotted the convergent curves of each algorithm in one figure to do explicit comparison of convergent speed. As shown in Fig. 4, EBP demonstrates slower convergent rate than MTS but performs better than SLCA.

In conclusion, we classify consensus-based algorithms into two categories: one is synchronous protocol with pseudo-synchronous implementation, e.g. EBP, SLCA; the other is asynchronous protocol with asynchronous implementation, e.g. ATS, MTS. Asynchronous protocol has its advantage of being easier to implement as it requires fewer floating point operations. Moreover, the convergent rate is faster than synchronous protocol. On the other hand, synchronous protocol can realize higher synchronization accuracy especially when physical clock parameters are time-varying. Hence we could choose the appropriate synchronization protocol according to the actual demand in practice.

5.2. Example 2: Suitability of large WSNs

Consider a 10×10 grid WSN that composed of 100 sensor nodes. The initial physical clock skews were assigned randomly from a normal distribution with mean 1 and standard deviation of 100 ppm. Each node broadcasts one sync packet per round in random order with a one-hop range of its neighbours. Control parameters are chosen as $\epsilon = 0.3$, $\gamma = 0.09$, $K_i = 0.1$, $K_p = 0.01$. Each sensor node is implemented with EBP. The geographic distribution of clock errors in the network is shown in Fig. 5 before synchronization. Fig. 6 shows the convergent results. After 32 rounds of iteration corresponding to 9.6 s, it can be seen that the maximum error of virtual clock reading is reduced under 0.1 clock ticks/s, well below the clock resolution.

If the communication range of each inner node increases from 4 to 8 nodes, it takes only 20 rounds of iteration (corresponding to 6.0 s) to constrain the maximum error of virtual clock reading

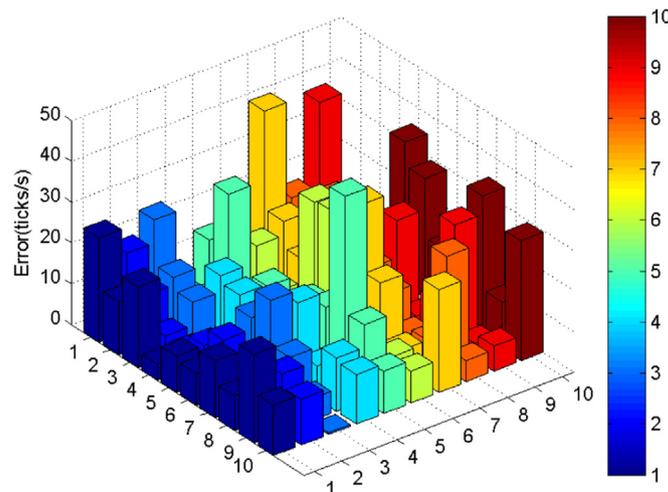


Fig. 5. Geographic distribution of error ticks before synchronization.

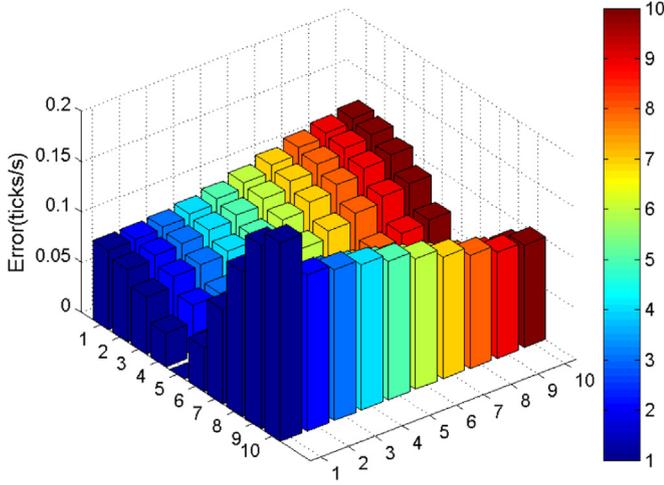


Fig. 6. Geographic distribution of error ticks after 32 rounds of synchronization.

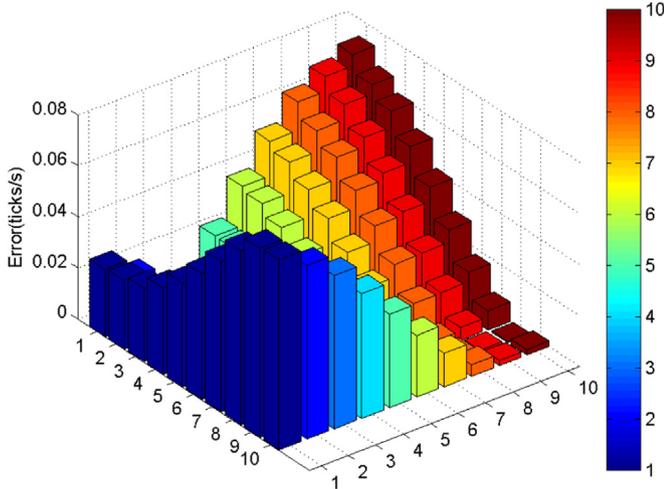


Fig. 7. Geographic distribution of error ticks after 20 rounds of synchronization.

under 0.1 clock ticks/s, see Fig. 7. Notice that the increase of transmission range and the number of sync packets contribute to the performances of EBP, e.g. precision of synchronization, the convergent property. Furthermore, if the skew compensation is viewed as a preliminary step, the offset compensation demonstrates better performance in convergent rate as each compensated local skew is bounded by a relative small range, no matter what degree the physical skew drifts to.

6. Conclusion

This paper presents a proportional integral estimator-based protocol to handle clock synchronization under time-varying parameters over wireless sensor networks. Under the proposed distributed protocol, a network of sensors can achieve clock synchronization in a distributed manner. Furthermore, the proposed protocol achieves higher synchronization accuracy against time-varying clock skew. We investigate the stability property of EBP and analyse the convergence property under both delay-free and random delay cases. Finally, a realistic pseudo-synchronous implementation is proposed. Future work includes optimization of control parameters in order to acquire better convergent property of EBP.

Appendix

Proof of Theorem 3.1. Define

$$\alpha_i(k)\omega_i(k) = \bar{\omega}_i(k), \quad \bar{\omega}(k) = [\bar{\omega}_1(k), \dots, \bar{\omega}_n(k)]^T. \quad (39)$$

Multiplying (21) with $\alpha_i(k)$, (21) becomes the following equivalent form

$$\begin{cases} \rho(k)\bar{\alpha}_i(k+1) = \bar{\alpha}_i(k) + \epsilon K_I \sum_{j \in N_i} (\bar{\omega}_i(k) - \bar{\omega}_j(k)) - \epsilon K_P \sum_{j \in N_i} (\bar{\alpha}_i(k) - \bar{\alpha}_j(k)) \\ \quad + \epsilon \gamma (\alpha_i(k) - \bar{\alpha}_i(k)) + \Delta_i^\alpha(k), \\ \rho(k)\bar{\omega}_i(k+1) = \bar{\omega}_i(k) - \epsilon K_I \sum_{j \in N_i} (\bar{\alpha}_i(k) - \bar{\alpha}_j(k)) + \Delta_i^\beta(k). \end{cases} \quad (40)$$

where we have the following relationship with $\rho(k)$, $\Delta_i^\alpha(k)$ and $\Delta_i^\beta(k)$

$$\begin{aligned} 1 - \frac{\rho_2}{1 - \rho_1} \leq \rho(k) &= 1 - \frac{\Delta \alpha_i(k)}{\alpha_i(k+1)} \leq 1 + \frac{\rho_2}{1 - \rho_1}. \\ \Delta_i^\alpha(k) &= -\epsilon K_P \sum_{j \in N_i} \bar{\alpha}_j(k) \left(1 - \frac{\alpha_i(k)}{\alpha_j(k)} \hat{\alpha}_{ij}(k) \right) \\ &\quad + \epsilon K_I \sum_{j \in N_i} \bar{\omega}_j(k) \left(1 - \frac{\alpha_i(k)}{\alpha_j(k)} \hat{\alpha}_{ij}(k) \right), \\ \Delta_i^\beta(k) &= -\epsilon K_I \sum_{j \in N_i} \bar{\alpha}_j(k) \left(1 - \frac{\alpha_i(k)}{\alpha_j(k)} \hat{\alpha}_{ij}(k) \right). \end{aligned} \quad (41)$$

Hence the aggregated synchronous form becomes

$$\begin{aligned} \begin{bmatrix} \bar{\alpha}(k+1) \\ \bar{\omega}(k+1) \end{bmatrix} &= \begin{bmatrix} \frac{(1 - \epsilon \gamma)I - \frac{\epsilon}{\rho(k)} L_P}{\rho(k)} & \frac{\epsilon}{\rho(k)} L_I \\ -\frac{\epsilon}{\rho(k)} L_I & \frac{1}{\rho(k)} I \end{bmatrix} \begin{bmatrix} \bar{\alpha}(k) \\ \bar{\omega}(k) \end{bmatrix} + \begin{bmatrix} \frac{\epsilon \gamma}{\rho(k)} \alpha(k) \\ \mathbf{0} \end{bmatrix} \\ &\quad + \Theta(k) \begin{bmatrix} \bar{\alpha}(k) \\ \bar{\omega}(k) \end{bmatrix}, \end{aligned} \quad (42)$$

where $L_p, L_l, \Theta(k), \Xi(k)$ are defined as follows:

$$\begin{aligned} L_p &= K_p L, \quad L_l = K_l L, \\ \Theta(k) &= \begin{bmatrix} -\frac{\epsilon K_p}{\rho(k)} (A_p - \Xi(k)) & \frac{\epsilon K_l}{\rho(k)} (A_l - \Xi(k)) \\ -\frac{\epsilon K_l}{\rho(k)} (A_l - \Xi(k)) & \mathbf{0}^n \end{bmatrix}, \\ \Xi(k)_{ij} &= \begin{cases} \frac{\alpha_i(k)}{\alpha_j(k)} \hat{\alpha}_{ij}(k) & i \neq j, j \in N_i, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (43)$$

Consider the state coordinate change

$$\begin{aligned} \bar{\alpha}(k) &= [v_e S] \chi(k), \quad \chi(k) = \begin{bmatrix} v_e^T \\ S^T \end{bmatrix} \bar{\alpha}(k), \\ \bar{\omega}(k) &= [v_e S] \sigma(k), \quad \sigma(k) = \begin{bmatrix} v_e^T \\ S^T \end{bmatrix} \bar{\omega}(k). \end{aligned} \quad (44)$$

where $v_e = \frac{1}{\sqrt{n}}$ is the unit eigenvector corresponding to zero eigenvalue of Laplacian matrix L, L_p . Define S such that $[v_e S]$ is an orthogonal matrix. Under the new state coordinate system, (42) becomes

$$\begin{aligned} \chi(k+1) &= \begin{bmatrix} \frac{1-\epsilon\gamma}{\rho(k)} & \mathbf{0}^T \\ \mathbf{0} & \frac{(1-\epsilon\gamma)I - \frac{\epsilon}{\rho(k)}S^T L_p S}{\rho(k)} \end{bmatrix} \chi(k) + \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \frac{\epsilon}{\rho(k)}S^T L_p S \end{bmatrix} \sigma(k) \\ &+ \frac{\epsilon\gamma}{\rho(k)} \begin{bmatrix} v_e^T \\ S^T \end{bmatrix} \alpha(k) + \begin{bmatrix} b_1(k) & \mathbf{0}^T \\ \mathbf{0} & B_1(k) \end{bmatrix} \chi(k) - \begin{bmatrix} c_1(k) & \mathbf{0}^T \\ \mathbf{0} & C_1(k) \end{bmatrix} \sigma(k), \\ \sigma(k+1) &= \frac{1}{\rho(k)} \sigma(k) + \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & -\frac{\epsilon}{\rho(k)}S^T L_p S \end{bmatrix} \chi(k) + \begin{bmatrix} d_1(k) & \mathbf{0}^T \\ \mathbf{0} & D_1(k) \end{bmatrix} \chi(k), \end{aligned}$$

where

$$\begin{aligned} b_1(k) &= -\frac{\epsilon K_p}{n\rho(k)} \sum_{j \in \mathcal{N}_1} \bar{\alpha}_j(k) \left(1 - \frac{\alpha_1(k)}{\alpha_j(k)} \hat{\alpha}_{1j}(k) \right), \\ B_1(k) &= -\frac{\epsilon K_l}{\rho(k)} S^T (A_p - \Lambda(k)) S, \\ c_1(k) &= -\frac{\epsilon K_l}{n\rho(k)} \sum_{j \in \mathcal{N}_1} \bar{\omega}_j(k) \left(1 - \frac{\alpha_1(k)}{\alpha_j(k)} \hat{\alpha}_{1j}(k) \right), \\ C_1(k) &= -\frac{\epsilon K_l}{\rho(k)} S^T (A_p - \Lambda(k)) S, \\ d_1(k) &= -\frac{\epsilon K_l}{n\rho(k)} \sum_{j \in \mathcal{N}_1} \bar{\alpha}_j(k) \left(1 - \frac{\alpha_1(k)}{\alpha_j(k)} \hat{\alpha}_{1j}(k) \right), \\ D_1(k) &= -\frac{\epsilon K_l}{\rho(k)} S^T (A_p - \Lambda(k)) S. \end{aligned} \tag{45}$$

The error dynamic becomes

$$e^\alpha(k) = [v_e \ S] \chi(k) - \frac{\mathbf{1}^T}{n} \alpha(k). \tag{46}$$

The first element $\sigma_1(k)$ of $\sigma(k)$ has the following dynamic:

$$\sigma_1(k+1) = \sigma_1(k) + d_1(k) \chi_1(k). \tag{47}$$

As $\sigma_1(k)$ is an uncontrollable and unobservable state, it is dropped.

Define $\sigma^* = [\sigma_2(k), \dots, \sigma_n(k)]^T = S^T \bar{\sigma}(k)$. The remaining dynamics are given as

$$\begin{aligned} \begin{bmatrix} \chi(k+1) \\ \sigma^*(k+1) \end{bmatrix} &= (A_1 + \Delta A_1) \begin{bmatrix} \chi(k) \\ \sigma^*(k) \end{bmatrix} + A_2 \alpha(k), \\ e(k) &= A_3 \begin{bmatrix} \chi(k) \\ \sigma^*(k) \end{bmatrix} + A_4 \alpha(k). \end{aligned} \tag{48}$$

where

$$\begin{aligned} A_1 &= \begin{bmatrix} \frac{1-\epsilon\gamma}{\rho(k)} & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \frac{(1-\epsilon\gamma)I - \frac{\epsilon}{\rho(k)}S^T L_p S}{\rho(k)} & \frac{\epsilon}{\rho(k)}S^T L_p S \\ \mathbf{0} & -\frac{\epsilon}{\rho(k)}S^T L_p S & \frac{1}{\rho(k)}I \end{bmatrix}, \\ A_2 &= \frac{\epsilon\gamma}{\rho(k)} \begin{bmatrix} v_e^T \\ S^T \\ \mathbf{0} \end{bmatrix}, \\ \Delta A_1 &= \begin{bmatrix} b_1(k) & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & B_1(k) & -C_1(k) \\ \mathbf{0} & D_1(k) & \mathbf{0}^{n-1} \end{bmatrix}, \\ A_3 &= [v_e \ S \ \mathbf{0}], \quad A_4 = -\frac{\mathbf{1}^T}{n}. \end{aligned} \tag{49}$$

As $S^T L_p S$ is of full rank, it is also invertible. Define new vectors as follows:

$$\bar{\chi}(k) = \begin{bmatrix} \frac{\epsilon\gamma v_e^T \alpha(k)}{\epsilon\gamma + \rho(k) - 1} \\ \mathbf{0} \end{bmatrix}, \quad \bar{\sigma}^*(k) = -\gamma (S^T L_p S)^{-1} S^T \alpha(k). \tag{50}$$

Then it satisfies

$$\begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix} \begin{bmatrix} \bar{\chi}(k) \\ \bar{\sigma}^*(k) \\ \alpha(k) \end{bmatrix} = \begin{bmatrix} \bar{\chi}(k) \\ \bar{\sigma}^*(k) \\ \frac{\mathbf{1}^T}{n} \frac{1 - \rho(k)}{\epsilon\gamma + \rho(k) - 1} \end{bmatrix}. \tag{51}$$

Define

$$\zeta(k) = \begin{bmatrix} \chi(k) \\ \sigma^*(k) \end{bmatrix} - \begin{bmatrix} \bar{\chi}(k) \\ \bar{\sigma}^*(k) \end{bmatrix}. \tag{52}$$

Hence we can give the dynamics of $\zeta(k)$

$$\begin{aligned} \zeta(k+1) &= A_1 \zeta(k) - \left(\begin{bmatrix} \bar{\chi}(k+1) \\ \bar{\sigma}^*(k+1) \end{bmatrix} - \begin{bmatrix} \bar{\chi}(k) \\ \bar{\sigma}^*(k) \end{bmatrix} \right) + \Delta A_1 \begin{bmatrix} \chi(k) \\ \sigma^*(k) \end{bmatrix}, \\ e(k) &= A_3 \zeta(k) + \frac{\mathbf{1}^T}{n} \frac{1 - \rho(k)}{\epsilon\gamma + \rho(k) - 1}. \end{aligned} \tag{53}$$

The dynamics of $\zeta(k)$ can be divided into two parts $\zeta_1(k)$ and $\zeta^*(k)$ where $\zeta^*(k) = [\zeta_2(k), \dots, \zeta_{2n-1}(k)]^T$. The dynamics of $\zeta_1(k)$ are as follows:

$$\begin{aligned} \zeta_1(k+1) &= \frac{(1-\epsilon\gamma)}{\rho(k)} \zeta_1(k) - \frac{\epsilon\gamma}{\epsilon\gamma + \rho(k) - 1} v_e^T (\alpha(k+1) - \alpha(k)) \\ &+ b_1(k) \chi_1(k). \end{aligned} \tag{54}$$

Based on (16), $\lim_{k \rightarrow \infty} |b_1(k)|$ converges to 0 exponentially. To guarantee the convergence of $\zeta_1(k)$, $\frac{(1-\epsilon\gamma)}{\rho(k)}$ must be less than one. As $1 - \frac{\rho_2}{1-\rho_1} < \rho(k) < 1 + \frac{\rho_2}{1-\rho_1}, \frac{(1-\epsilon\gamma)}{\rho(k)} < 1$ indicates the following condition of control parameter constraint

$$\frac{\rho_2}{1-\rho_1} < \epsilon\gamma. \tag{55}$$

As $v_e^T |\alpha(k+1) - \alpha(k)|$ is bounded by ρ_2 ,

$$\begin{aligned} \lim_{k \rightarrow \infty} |\zeta_1(k)| &= \lim_{k \rightarrow \infty} \left| \left(\frac{1-\epsilon\gamma}{\rho(k)} \right)^k \zeta_1(0) - \frac{\epsilon\gamma}{\epsilon\gamma + \rho(k) - 1} \frac{\rho_2}{1 - \frac{1-\epsilon\gamma}{\rho(k)}} \left(1 - \left(\frac{1-\epsilon\gamma}{\rho(k)} \right)^k \right) \right. \\ &\left. + b_1(k) \chi_1(k) \right| = \frac{\epsilon\gamma \rho(k) \rho_2}{(\rho(k) - 1 + \epsilon\gamma)^2}, \end{aligned} \tag{56}$$

which indicates that $\zeta_1(k)$ converges exponentially to a ball of radius $\frac{\epsilon\gamma \rho(k) \rho_2}{(\rho(k) - 1 + \epsilon\gamma)^2}$ as $k \rightarrow \infty$.

The dynamics of $\zeta^*(k)$ are given by

$$\begin{aligned} \zeta^*(k+1) &= \left(\frac{1}{\rho(k)} I + \frac{\epsilon}{\rho(k)} A_5 \right) \zeta^*(k) \\ &+ A_6(k) \begin{bmatrix} \hat{\chi}(k) \\ \sigma^*(k) \end{bmatrix} + \left(\begin{bmatrix} \mathbf{0} \\ \bar{\sigma}^*(k+1) \end{bmatrix} - \begin{bmatrix} \mathbf{0} \\ \bar{\sigma}^*(k) \end{bmatrix} \right), \end{aligned} \tag{57}$$

where

$$\begin{aligned} A_5 &= \begin{bmatrix} -\gamma I - S^T L_p S & S^T L_p S \\ -S^T L_p S & \mathbf{0}^{n-1} \end{bmatrix}, \quad A_6(k) = \begin{bmatrix} B_1(k) & -C_1(k) \\ D_1(k) & \mathbf{0}^{n-1} \end{bmatrix}, \\ \hat{\chi}(k) &= [\chi_2(k), \dots, \chi_n(k)]^T. \end{aligned} \tag{58}$$

As L_l corresponds to a connected undirected graph \mathcal{G} , the

following well-known property [20] is established:

$$\min_{x \neq 0, t^T S = 0} \frac{\|S^T L_t S\|}{\|S^T\|^2} = \lambda_2(L_I). \quad (59)$$

Hence $|\bar{\sigma}_*(k+1) - \bar{\sigma}_*(k)|$ is bounded as follows:

$$\begin{aligned} |\bar{\sigma}_1^*(k+1) - \bar{\sigma}_1^*(k)| &\leq \frac{\gamma \rho_2}{K_I \lambda_2(L_I)}, \\ |\bar{\sigma}_*(k+1) - \bar{\sigma}_*(k)| &\leq \frac{\gamma \rho_1}{K_I \lambda_2(L_I)}. \end{aligned} \quad (60)$$

In order to give a bound on $\zeta^*(k)$, we need to prove that system (57) is input-to-state stable, hence the spectrum of A_5 needs to be analyzed. Define the eigenvalue set of A_5 as $\{\hat{\lambda}_1, \dots, \hat{\lambda}_{2n-2}\}$. Based on Lemma 2.1, the spectrum of A_5 satisfies the bound

$$-\kappa\beta_2 \leq \text{Re}\{\text{spectrum}(A_5)\} \leq -\kappa\beta_1, \quad (61)$$

where

$$\begin{aligned} \beta_1 &= \min_{\lambda} \frac{1}{2} \text{Re} \left[\gamma + K_p \lambda - \sqrt{(\gamma + K_p \lambda)^2 - 4(K_I \lambda_2(L))^2} \right], \\ \beta_2 &= \frac{1}{2} \text{Re} \left[\gamma + \sqrt{(\gamma + K_p \lambda_{\max}(L))^2 - 4(K_I \lambda_2(L))^2} + K_p \lambda_{\max}(L) \right], \\ \lambda &\in \{\lambda_2(L), \lambda_{\max}(L)\}. \end{aligned} \quad (62)$$

Eq. (61) indicates that A_5 is not only Hurwitz stable, but also has a stability margin bounded by β_1, β_2 . To guarantee the spectrum of $\left(\frac{1}{\rho(k)}I + \frac{\epsilon}{\rho(k)}A_5\right)$ lying in the unit disk, the following condition needs to be satisfied

$$|1 + \epsilon \hat{\lambda}_i| < \rho(k). \quad (63)$$

As $\rho(k)$ is bounded by $1 - \frac{\rho_2}{1-\rho_1} < \rho(k) < 1 + \frac{\rho_2}{1-\rho_1}$, a more conservative condition is presented as

$$|1 + \epsilon \hat{\lambda}_i| \leq 1 - \frac{\rho_2}{1-\rho_1}. \quad (64)$$

By mathematical manipulations to (64), the step-size ϵ should satisfy the following condition

$$\frac{\rho_2}{\beta_1(1-\rho_1)} \leq \epsilon \leq \frac{2-\rho_2-2\rho_1}{\beta_2(1-\rho_1)}. \quad (65)$$

As long as (65) is satisfied, there exists a ϵ such that $\left(\frac{1}{\rho(k)}I + \frac{\epsilon}{\rho(k)}A_5\right)$ lies in the unit disk. By Lemma 2.2, system (57) is input-to-state stable.

Let V become the similar transformation of $I + \epsilon A_5$ and suppose that $I + \epsilon A_5 = V \Lambda V^{-1}$ such that Λ is diagonal. By similarity transformation V

$$\eta(k) = V^{-1} \zeta_*(k), \quad (66)$$

system (57) is transformed into

$$\begin{aligned} \eta(k+1) &= \begin{bmatrix} \Lambda_1 & \mathbf{0}^{n-1} \\ \mathbf{0}^{n-1} & \Lambda_2 \end{bmatrix} \eta(k) \\ &+ V^{-1} \begin{bmatrix} B_1(k) & -C_1(k) \\ D_1(k) & \mathbf{0}^{n-1} \end{bmatrix} \begin{bmatrix} \hat{\chi}(k) \\ \sigma^*(k) \end{bmatrix} \\ &+ V^{-1} \begin{bmatrix} \mathbf{0} \\ \bar{\sigma}_*(k+1) - \bar{\sigma}_*(k) \end{bmatrix}, \end{aligned} \quad (67)$$

where $\Lambda_1 = \text{diag}\left\{\frac{1+\epsilon \hat{\lambda}_1}{\rho(k)}, \dots, \frac{1+\epsilon \hat{\lambda}_{n-1}}{\rho(k)}\right\}$ and $\Lambda_2 = \text{diag}\left\{\frac{1+\epsilon \hat{\lambda}_n}{\rho(k)}, \dots, \frac{1+\epsilon \hat{\lambda}_{2n-2}}{\rho(k)}\right\}$ are diagonal matrices with its diagonal elements being the

eigenvalues of $\frac{I + \epsilon A_5}{\rho(k)}$. If we let $\eta^1(k) = [\eta_1(k), \dots, \eta_{n-1}(k)]$ and $\eta^2(k) = [\eta_n(k), \dots, \eta_{2n-2}(k)]$,

$$\begin{aligned} \begin{bmatrix} \eta^1(k+1) \\ \eta^2(k+1) \end{bmatrix} &= \begin{bmatrix} \Lambda_1 & \mathbf{0}^{n-1} \\ \mathbf{0}^{n-1} & \Lambda_2 \end{bmatrix} \begin{bmatrix} \eta^1(k) \\ \eta^2(k) \end{bmatrix} \\ &+ V^{-1} \begin{bmatrix} \mathbf{0} \\ \bar{\sigma}_*(k+1) - \bar{\sigma}_*(k) \end{bmatrix} \\ &+ V^{-1} \begin{bmatrix} B_1(k) & -C_1(k) \\ D_1(k) & \mathbf{0}^{n-1} \end{bmatrix} \begin{bmatrix} \hat{\chi}(k) \\ \sigma^*(k) \end{bmatrix}. \end{aligned} \quad (68)$$

According to (16),

$$\lim_{k \rightarrow \infty} |A_6(k)| = \begin{bmatrix} B_1(k) & -C_1(k) \\ D_1(k) & \mathbf{0}^{n-1} \end{bmatrix} = \begin{bmatrix} \mathbf{0}^{n-1} & \mathbf{0}^{n-1} \\ \mathbf{0}^{n-1} & \mathbf{0}^{n-1} \end{bmatrix}. \quad (69)$$

Hence it follows that $\eta^1(k)$ exponentially converges to 0 as $k \rightarrow \infty$. As $|\bar{\sigma}_1^*(k+1) - \bar{\sigma}_1^*(k)|$ is bounded by $\frac{\gamma \rho_2}{K_I \lambda_2(L_I)}$,

$$\begin{aligned} \lim_{k \rightarrow \infty} |\eta_i^2(k)| &= \lim_{k \rightarrow \infty} \left| \left(\frac{1 + \epsilon \hat{\lambda}_{i+n-1}}{\rho(k)} \right)^k \eta_i^2(0) \right. \\ &\left. + \frac{\gamma \rho_2 \|V^{-1}\|}{K_I \lambda_2(L_I)} \frac{1 - \left(\frac{1 + \epsilon \hat{\lambda}_{i+n-1}}{\rho(k)} \right)^k}{1 - \left(\frac{1 + \epsilon \hat{\lambda}_{i+n-1}}{\rho(k)} \right)} \right| \\ &= \frac{\gamma \rho_2 \rho(k) \|V^{-1}\|}{(\rho(k) - 1 - \epsilon \hat{\lambda}_{i+n-1}) K_I \lambda_2(L_I)}. \end{aligned} \quad (70)$$

Adding the bounds (56) and (70) together, we conclude that $\epsilon_i(k)$ converges exponentially to a ball of radius

$$\rho_3 = \frac{\epsilon \gamma \rho_2 \rho(k)}{(\rho(k) - 1 + \epsilon \gamma)^2} + \frac{\gamma \rho_2 \rho(k) \|V^{-1}\|}{(\rho(k) - 1 - \epsilon \hat{\lambda}_{i+n-1}) K_I \lambda_2(L_I)}. \quad (71)$$

This completes the proof of Theorem 3.1. \square

References

- [1] Mukherjee B, Yick J, Ghosal D. Wireless sensor network survey. *Comput Netw* 2008;52(12):2292–330.
- [2] Krishnamurthy SV, Yuan W, Tripathi SK. Synchronization of multiple levels of data fusion in wireless sensor networks. In: IEEE global telecommunications conference, 2003. GLOBECOM'03, vol. 1. IEEE; 2003. p. 221–5.
- [3] Buy U, Sundararaman B, Kshemkalyani AD. Clock synchronization for wireless sensor networks: a survey. *Ad hoc Netw* 2005;3(3):281–323.
- [4] Faizulkhakov YR. Time synchronization methods for wireless sensor networks: a survey. *Program Comput Softw* 2007;33(4):214–26.
- [5] Graham SR, Freris NM, Kumar PR. Fundamental limits on synchronizing clocks over networks. *IEEE Trans Autom Control* 2011;56(6):1352–64.
- [6] Kumar R, Ganeriwala S, Srivastava MB. Timing-sync protocol for sensor networks. In: Proceedings of the 1st international conference on embedded networked sensor systems. ACM; 2003. p. 138–49.
- [7] Simon G, Maroti M, Kusy B, Ledeczi A. The flooding time synchronization protocol. In: Proceedings of the 2nd international conference on embedded networked sensor systems. ACM; 2004. p. 39–49.
- [8] Girod L, Elson J, Estrin D. Fine-grained network time synchronization using reference broadcasts. *ACM SIGOPS Oper Syst Rev* 2002;36(SI):147–63.
- [9] Sari I, Serpedin E, Noh KL, Chaudhari Q, Suter B. On the joint synchronization of clock offset and skew in rbs-protocol. *IEEE Trans Commun* 2008;56(5):700–3.
- [10] Noh KL, Serpedin E. Pairwise broadcast clock synchronization for wireless sensor networks. In: 2007 IEEE international symposium on a world of wireless, mobile and multimedia networks. IEEE; 2007. p. 1–6.
- [11] Ahmad A, Zennaro D, Serpedin E, Vangelista L. A factor graph approach to

- clock offset estimation in wireless sensor networks. *IEEE Trans Inf Theory* 2012;58(7):4244–60.
- [12] Giridhar A, Kumar PR. Distributed clock synchronization over wireless networks: algorithms and analysis. In: Proceedings of the 45th IEEE conference on decision and control. IEEE; 2006. p. 4915–20.
- [13] Borkar VS, Freris NM, Kumar PR. A model-based approach to clock synchronization. In: Proceedings of the 48th IEEE conference on decision and control (or Chinese control conference). IEEE; 2009. p. 5744–9.
- [14] Freris NM, Zouzas A. Fast distributed smoothing of relative measurements. In: Proceedings of the 51st IEEE conference on decision and control (CDC). IEEE; 2012. p. 1411–6.
- [15] Tewari G, Werner-Allen G, Patel A. Firefly-inspired sensor network synchronicity with realistic radio effects. In: Proceedings of the 3rd international conference on embedded networked sensor systems. ACM; 2005. p. 142–53.
- [16] Hespanha JP, Barooah P, Swami A. On the effect of asymmetric communication on distributed time synchronization. In: Proceedings of the 46th IEEE conference on decision and control. IEEE; 2007. p. 5465–71.
- [17] Simeone O, Spagnolini U. Distributed time synchronization in wireless sensor networks with coupled discrete-time oscillators. *EURASIP J Wirel Commun Netw* 2007;2007(1):1–13.
- [18] Du J, Wu YC. Distributed clock skew and offset estimation in wireless sensor networks: asynchronous algorithm and convergence analysis. *IEEE Trans Wirel Commun* 2013;12(11):5908–17.
- [19] Luo B, Wu YC. Distributed clock parameters tracking in wireless sensor network. *IEEE Trans Wirel Commun* 2013;12(12):6464–75.
- [20] Olfati-Saber R, Murray RM. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans Autom Control* 2004;49(9):1520–33.
- [21] Fax JA, Olfati-Saber R, Murray RM. Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 2007;95(1):215–33.
- [22] Prabhakar B, Boyd S, Ghosh A, Shah D. Randomized gossip algorithms. *IEEE/ACM Trans Netw* 2006;14(SI):2508–30.
- [23] Schenato L, Fiorentin F. Average timesynch: a consensus-based protocol for clock synchronization in wireless sensor networks. *Automatica* 2011;47(9):1878–86.
- [24] O'Keefe SG, Maggs MK, Thiel DV. Consensus clock synchronization for wireless sensor networks. *IEEE Sens J* 2012;12(6):2269–77.
- [25] Cheng P, He J, Shi L. Time synchronization in wsns: a maximum-value-based consensus approach. *IEEE Trans Autom Control* 2014;59(3):660–75.
- [26] Chen J, He J, Cheng P. Secure time synchronization in wireless sensor networks: a maximum consensus-based approach. *IEEE Trans Parallel Distrib Syst* 2014;25(4):1055–65.
- [27] Shen X, Choi B, Liang H, Zhuang W. Dcs: distributed asynchronous clock synchronization in delay tolerant networks. *IEEE Trans Parallel Distrib Syst* 2012;23(3):491–504.
- [28] Sommer P, Wattenhofer R. Gradient clock synchronization in wireless sensor networks. In: Proceedings of the 2009 international conference on information processing in sensor networks. IEEE Computer Society; 2009. p. 37–48.
- [29] Schenato L, Carli R, Chiuso A, Zampieri S. A pi consensus controller for networked clocks synchronization. *IFAC Proc* 2008;41(2):10289–94.
- [30] Schenato L, Carli R, Chiuso A, Zampieri S. Optimal synchronization for networks of noisy double integrators. *IEEE Trans Autom Control* 2011;56(5):1146–52.
- [31] Carli R, Zampieri S. Network clock synchronization based on the second-order linear consensus algorithm. *IEEE Trans Autom Control* 2014;59(2):409–22.
- [32] Carli R, Yildirim KS, Schenato L. Adaptive control-based clock synchronization in wireless sensor networks. In: Proceedings of IEEE 2015 European control conference. IEEE, 2015. p. 2806–11.
- [33] Yang P, Lynch KM, Schwartz BI, Freeman RA. Decentralized environmental modeling by mobile sensor networks. *IEEE Trans Robotics* 2008;24(3):710–24.
- [34] Khalil HK, Grizzle JW. *Nonlinear systems*, vol. 3. New Jersey: Prentice Hall; 1996.
- [35] Gordon CJ, Yang P, Freeman RA, Lynch KM. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica* 2010;46(2):390–6.
- [36] Yang P, Freeman RA, Lynch KM. Stability and convergence properties of dynamic average consensus estimators. In: Proceedings of the 45th IEEE conference on decision and control, 2006.