53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

Distributed Time Synchronization under Bounded Noise in Wireless Sensor Networks

Jianping He[1,2], Xiaoming Duan[1], Peng Cheng[1], Ling Shi[3] and Lin Cai[2]

Abstract— **It is important and challenging to achieve accurate time synchronization in wireless sensor networks. Various noises, e.g., communication delay, clock fluctuation and measurement errors, are inevitable and hard to be estimated accurately, which is the main challenge for achieving an accurate time synchronization. In this paper, we focus on how to achieve accurate time synchronization by considering a practical noise model, bounded noise, which may not satisfy any known distributions. The principle that a bounded monotonic sequence must possess a limit and the concept of maximum consensus are exploited to design a novel time synchronization algorithm for the network to achieve accurate synchronization under bounded noise. The proposed algorithm is fully distributed, with high synchronization accuracy and fast convergence speed, and is able to compensate both clock skew and offset simultaneously. Moreover, we prove that the algorithm converges with probability one, which means that time synchronization is achieved completely, and the probability of the complete synchronization converges exponentially.**

I. Introduction

Time synchronization is a fundamental requirement for various applications in wireless sensor networks (WSNs), e.g., data fusion, sensor scheduling and node cooperation [1]. It has been extensively studied in WSNs, and many protocols have been proposed for time synchronization in various scenarios [2]–[12]. For most of those protocols, time synchronization can be achieved completely when the noise is ignored, however, when taking noises, e.g., communication delay and measurement errors, into consideration, a highly accurate synchronization will no longer be guaranteed by these protocols, e.g., the accuracy of the maximum time synchronization (MTS) protocol proposed in [11] decreases with the variance of the random communication delay. Therefore, an accurate time synchronization in WSNs under noises is still a challenging problem.

Recently, the concept of consensus is developed to design consensus-based time synchronization protocols to achieve global accuracy time synchronization for WSNs [6]–[12]. These consensus-based protocols can be classified into two categories, i.e., average, maximum or minimum-consensus based time synchronization protocols. For example, the protocols in [6]–[9] are average consensus-based, since the basic idea of them is that each node takes an average of its own clock parameter and its neighboring ones to drive the network to achieve a consensus reference clock. He. et al. [11], [12] utilized the maximum and minimum consensus to design time synchronization protocol, which is able to achieve a much faster convergence speed. These consensus-based protocols are fully distributed, so they are robust against different uncertainties in WSNs, e.g., packet losses, node failures, the addition of new nodes and etc., and are promising for real applications in the networks.

Unfortunately, the noises, including communication delay, measurement error and the fluctuation of clock speed, are ignored in the design of above consensus-based protocols. Taking the noises into consideration, complete time synchronization may not be achieved by these consensus-based protocols [12]. Hence, it is of great interests to design new consensus-based or distributed time synchronization protocol to handle the noises, and then improve the robustness and the accuracy of time synchronization. There are some works which investigated time synchronization under different noise models for WSNs [11], [13], [14]. For example, the authors in [13], [14] proposed new average consensus-based protocols to improve the robustness of the typical average time synchronization (ATS) algorithm proposed in [8], including the drift of clock skew and the fluctuation of clock in the stable state of the algorithm. However, these algorithms may not be able to improve the synchronization accuracy, and they are still average consensus-based algorithms, which have slow convergence speed as ATS. In this paper, we develop a distributed time synchronization protocol by adapting the concept of maximum consensus which provides higher accuracy and faster converging speed than ATS under bounded noise in WSNs. The main contributions of this work are summarized as follows:

1) This paper investigates how to achieve accurate time synchronization under bounded noise which is a practical and general model in WSNs. By exploiting the principle that a bounded monotonic sequence must posses a limit and the idea of maximum consensus, we propose a novel distributed algorithm, including the relative skew estimation, clock skew and offset compensation. The algorithm achieves high synchronization accuracy and fast convergence speed.
2) We prove that the proposed algorithm guarantees that

1: The State Key Lab. of Industrial Control Technology, Zhejiang University, China {jphe, pcheng}@iipc.zju.edu.cn, xiaomingduan.zju@gmail.com
2: The Department of Electrical & Computer Engineering at the University of Victoria, BC, Canada cai@ece.uvic.ca
3: The Department of Electric and Computer Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong eesling@ust.hk
This work was partially supported by NSFC under grant 61473251, ZJSF under grant LY14F030016, CPSF under grant 2014M551739, and State Key Lab. of Synthetical Automation for Process Industries. The work by L. Shi was supported by an HKUST internal grant FSGRF14EG48. This work by L. Cai was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

the time synchronization can be achieved with probability one. We also prove that the probability of the complete synchronization converges to one exponentially. Extensive simulations demonstrate that the proposed algorithm has a higher accuracy and faster convergence speed than that of ATS.

The remainder of the paper is organized as follows. In Section II, the problem of time synchronization under bounded noise is formulated. Section III presents the detailed distributed time synchronization algorithm. Simulation is presented in Section IV to evaluate the performance of proposed algorithm. Finally, Section V concludes the paper and discusses some future works.

II. System Models and Problem Setup

A WSN is modelled as a connected, undirected graph $G = (V; E)$, where V is the set of sensor nodes, with $|V| = n$ ($n \geq 2$), and E is the set of communication links (edges) between them. The neighbor set of sensor node i is denoted by N_i, where $j \in N_i$ if and only if (iff) the link $(j, i) \in E$.

A. Clock Model

Each sensor has a hardware clock, calculated by counting pulses of its hardware oscillator running at a particular frequency. For a relatively longer period of time (minutes to hours), by referring to [4], [5], the hardware clock can be approximated with good accuracy by an oscillator of fixed frequency. Thus, the local hardware clock of sensor node i, denoted by $H_i(t)$, can be approximated as

$$H_i(t) = \alpha_i t + \beta_i, i \in V, \quad (1)$$

where t is the real time, α_i is the hardware clock skew which determines the clock speed and β_i is the hardware clock offset. In practice, $\alpha_i \neq \alpha_j$ for $\forall i \neq j$ as the qualities of sensors oscillators are usually different which leads to that sensor's oscillators run at slightly different frequencies [6]. One also has $\beta_i \neq \beta_j$ for $\forall i \neq j$ as the start-up times of sensor nodes are different [12]. Hence, different nodes usually have different hardware clocks, due to different clock skew and offset. Since the value of hardware clock cannot be adjusted manually [7], a software clock is provided to represent the synchronized time, which is given by

$$S_i(t) = \tilde{\alpha}_i H_i(t) + \tilde{\beta}_i = x_i t + y_i, i \in V,$$

where $x_i = \tilde{\alpha}_i \alpha_i$ and $y_i = \tilde{\alpha}_i \beta_i + \tilde{\beta}_i$ are the software clock skew and offset, respectively.

B. Bounded Noise Model

Let $H_i^+(t)$ be the hardware clock information sent from node i at time t. Each $H_i^+(t)$ is assumed to satisfy

$$H_i^+(t) = H_i(t) + \theta_i(t) = \alpha_i t + \beta_i + \theta_i(t), i \in V, \quad (2)$$

where $\theta_i(t) \in [a, b]$ is defined as the bounded noise due to the communication delay [13], measurement error [23], and clock fluctuation [24]–[27], etc. Assume for each node i that the noises $\theta_i(t_1)$ and $\theta_i(t_2)$ are independent with each other for $t_1 \neq t_2$. Since the values of both a and b can be obtained from experiments, it is assumed in this paper that a and b are known to each sensor node. The above bounded model is a general and practical model for the noises, and a typical example is the random bounded communication delay considered in [13]. Since each noise $\theta_i(t)$ may be of any value in $[a, b]$, we give the following assumption.

Assumption 2.1: Given a constant $\delta > 0$, assume that there exists $0 < \epsilon \leq 1$ such that $\Pr\{\theta_i(t) \in [c-\delta, c+\delta] | \forall c \in [a, b]\} \geq \epsilon$ for $i \in V$.

Remark 2.2: In many existing works, it is usually assumed that the noise is a random variable following certain distributions, e.g., Gaussian and exponential distribution [17]–[19], or with constant mean and variance [11], which can be seen as special cases of this assumption. For these special cases, it has been proved in the previous papers that it can achieve a high accuracy and even complete clock synchronization in expectation. However, the noise in (2) may have different probability distribution at different time t and may not have a certain constant mean and variance, which may render the time synchronization unreachable with existing algorithms in these papers.

C. Problem Setup

The goal of clock synchronization is to find the parameters $(\tilde{\alpha}_i, \tilde{\beta}_i)$ for each node i such that all nodes have the same software clock skews and offsets, i.e., $|x_i - x_j| = 0$ and $|y_i - y_j| = 0$ for $\forall i, j \in V$, which means that $|S_i(t) - S_j(t)| = 0, \forall i, j \in V$. Consensus-based time synchronization algorithms have been developed to realize this goal [7]–[9], where time synchronization can be achieved completely using the algorithm when the noises, e.g., communication delay and the fluctuation of hardware clock, are ignored. Unfortunately, as pointed out in [15], [16] that the noises are the fundamental limits which affects the synchronization accuracy and even renders the synchronization impossible, as the noises cannot be estimated accurately.

Therefore, the objective of this paper is to design a distributed synchronization algorithm, including skew and offset compensation, to find $(\tilde{\alpha}_i(k), \tilde{\beta}_i(k))$ for each node i, such that

$$\Pr\{\lim_{k \to \infty} |x_i(k) - x_j(k)| = 0\} = 1; \quad (3)$$

$$\Pr\{\lim_{k \to \infty} |y_i(k) - y_j(k)| = 0\} = 1, \quad (4)$$

for $i, j \in V$, where k is the iteration, $x_i(k) = \tilde{\alpha}_i(k)\alpha_i$ and $y_i(k) = \tilde{\alpha}_i(k)\beta_i + \tilde{\beta}_i(k)$. Equations (3) and (4) guarantee that a highly accurate time synchronization can be achieved under the bounded noise, and even a complete synchronization is achieved when $t \to \infty$.

III. Distributed Time Synchronization Algorithm Design

In this section, we propose a new distributed time synchronization algorithm for realizing the goals in both (3) and (4). In the algorithm, the principle that a bounded monotonic sequence must possess a limit is utilized to design the estimation method to counteract the impact of bounded noise, and the maximum consensus is used as the update rule of each iteration, which guarantees a fast convergence speed of the algorithm. This algorithm includes three parts, relative skew estimation, skew compensation and offset compensation, and the details will be given in the following subsections, respectively.

A. Relative Skew Estimation

Since the real time t is unavailable to each node, α_i and β_i cannot be computed [8]. However, if we can obtain the local hardware clock readings, we can obtain a relative clock between any two nodes i and j as

$$H_i(t) = \frac{\alpha_i}{\alpha_j} H_j(t) + \left(\beta_i - \frac{\alpha_i}{\alpha_j}\beta_j\right) = \alpha_{ji} H_j(t) + \beta_{ji}, \quad (5)$$

where $\alpha_{ji} = \frac{\alpha_i}{\alpha_j}$ is the **relative skew** and $\beta_{ji} = \beta_i - \alpha_{ji}\beta_j$ is the **relative offset** [8], [11], [20], [21]. After obtaining the relative clock, each node i thus can synchronize its clock with the neighbor j's clock. Then the problem becomes how to obtain an accurate estimation of the relative skew under the noise.

The estimation of the relative skew is crucial in time synchronization, since both the clock skew and offset compensation depend on it, and the accuracy of its estimation will directly affect the synchronization accuracy [22]. Define $e_{ij}(k)$ as a one-step estimation of the relative skew, and its estimation is designed as

$$e_{ij}(k) = \frac{H_j^+(t_k) - H_j^+(t_{k-1}) - (b-a)}{H_i(t_k) - H_i(t_{k-1})}, \; i, j \in V, \quad (6)$$

where $H_j^+(t_k)$ and $H_j^+(t_{k-1})$ are the hardware clock information received from node j at iterations k and $k-1$, respectively, and $H_i(t_k)$ and $H_i(t_{k-1})$ are the two times of the time-sampling of node i. Let $\hat{\alpha}_{ij}(1) = e_{ij}(1)$ be the initial estimation of relative skew. Then, we have the following equation to estimate each relative skew α_{ij} at each iteration,

$$\hat{\alpha}_{ij}(k+1) = \max\{e_{ij}(k+1), \hat{\alpha}_{ij}(k)\}, i \in V, j \in N_i. \quad (7)$$

Note that $\theta_j(t_k) \in [a, b]$, we have $f_{ij}(t_k, t_{k-1}) \leq 0$, which means that $e_{ij}(k) \leq \frac{\alpha_j}{\alpha_i}$ for each iteration k. Meanwhile, from (7), we obtain that $\hat{\alpha}_{ij}(k)$ is an increasing (non-decreasing) function of iteration k with an upper bound α_{ij}, i.e., $\hat{\alpha}_{ij}(k) \leq \alpha_{ij}$. Hence, according to the principle that a bounded monotonic sequence must possess a limit, $\hat{\alpha}_{ij}(k)$ will converge to a constant. Then, we obtain a lemma as follows.

Lemma 3.1: Given any small constant $\sigma > 0$, using (7) to estimate the relative skew, we have

$$\Pr\{\lim_{k\to\infty} \alpha_{ij} - \hat{\alpha}_{ij}(k) \leq \sigma | i, j \in V\} = 1. \quad (8)$$

Proof: Since the time interval between two iterations is usually positive, there exists $\tau > 0$ such that $H_i(t_k) - H_i(t_{k-1}) \geq \tau$.

From (1) and (2), it follows that

$$\begin{aligned} e_{ij}(k) =& \frac{H_j(t_k) - H_j(t_{k-1})}{H_i(t_k) - H_i(t_{k-1})} + \frac{\theta_j(t_k) - \theta_j(t_{k-1}) - (b-a)}{H_i(t_k) - H_i(t_{k-1})} \\ =& \frac{\alpha_j}{\alpha_i} + f_{ij}(t_k, t_{k-1}), \; i, j \in V, \end{aligned} \quad (9)$$

where

$$f_{ij}(t_k, t_{k-1}) = \frac{\theta_j(t_k) - \theta_j(t_{k-1}) - (b-a)}{H_i(t_k) - H_i(t_{k-1})}.$$

According to Assumption 2.1, there exists δ and ϵ such that $\Pr\{|\theta_j(t_k) - b| \leq \delta\} \geq \epsilon$ and $\Pr\{|\theta_j(t_{k-1}) - a| \leq \delta\} \geq \epsilon$. Then, we have

$$\begin{aligned} &\Pr\{f_{ij}(t_k, t_{k-1}) \geq -\frac{2\delta}{\tau}\} \\ \geq& \Pr\{|\theta_j(t_k) - b| \leq \delta \cap |\theta_j(t_{k-1}) - a| \leq \delta\} \geq \epsilon^2 \end{aligned}$$

which means that

$$\Pr\{\alpha_{ij} - e_{ij}(k) \leq \frac{2\delta}{\tau}\} \geq \epsilon^2, \quad (10)$$

for $i, j \in V$.

From (7) and (10), we have

$$\begin{aligned} &\Pr\{\alpha_{ij} - \hat{\alpha}_{ij}(k) \leq \frac{2\delta}{\tau}\} \\ =& \Pr\{\alpha_{ij} - \max\{e_{ij}(1), e_{ij}(2), ..., e_{ij}(k)\} \leq \frac{2\delta}{\tau}\} \\ =& 1 - \Pr\{\alpha_{ij} - e_{ij}(l) > \frac{2\delta}{\tau} | l = 1, 2, ..., k\} \\ \geq& 1 - (1 - \epsilon^2)^k \end{aligned} \quad (11)$$

for each iteration k. Therefore, let $\sigma = \frac{2\delta}{\tau}$ and $k \to \infty$, we have equation (8) hold. ■

From Assumption 2.1, we can find a positive ϵ for $\forall \delta > 0$, which guarantees that (8) hold. And, note that $\sigma = \frac{2\delta}{\tau}$, which means that for any estimation accuracy $\sigma > 0$, it can be achieved with probability one guaranteed by Lemma 3.1. Hence, (7) can guarantee any accuracy requirement to be reached with probability one, i.e., the estimation of the relative skew converges to the real relative skew almost surely. Meanwhile, from (11), it follows that the probability of an accurate estimation of the relative skew converges to one exponentially fast, which means that the estimation algorithm (7) has an exponential speed in probability. Moreover, from Lemma 3.1, note that for any a small positive δ, we have $\Pr\{|\theta_j(t_k) - b| \leq \delta\} \geq \epsilon > 0$ and $\Pr\{|\theta_j(t_{k-1}) - a| \leq \delta\} \geq \epsilon > 0$ which can guarantee the estimation accuracy as $\sigma = \frac{2\delta}{\tau}$, especially when $\delta \to 0$ a complete accurate estimation is achieved, i.e.,

$$\Pr\{\lim_{k\to\infty} \alpha_{ij} - \hat{\alpha}_{ij}(k) = 0 | i, j \in V\} = 1.$$

Thus, in the reminder of this paper, we assume that $\Pr\{\theta_i(t) = c\} \geq \epsilon$ for all $i \in V$, where $c = a$ or $c = b$, and $0 < \epsilon < 1$ is a small positive constant.

Remark 3.2: If the noise bounds a and b are unknown to each node i, then we can give a relative large bound B such that each $\theta_i(t) \in [-B, B]$, and using $2B$ to substitute

$b-a$ in (6). It also has $e_{ij}(k) \leq \alpha_{ij}$ which guarantees that $\hat{\alpha}_{ij}(k) \leq \alpha_{ij}$. Meanwhile, the pair $(H_j^+(t_{k-1}), H_i(t_{k-1}))$ can be substituted with $(H_j^+(t_0), H_i(t_0))$ in (6), and then we have $\lim_{t_k\to\infty} f_{ij}(t_k, t_0) = 0$, which also guarantees that the relative skew estimation of (7) converges to α_{ij}.

B. Skew Compensation

By referring to our early work [11], based on the maximum consensus approach, we design the following skew compensation iteration algorithm. When node i receives information $H_j^+(t_k)$ and $\tilde{\alpha}_j(k)$ from neighbor node j, it updates its clock skew as

$$\tilde{\alpha}_i(t_k^+) = \max\{\tilde{\alpha}_i(t_k), \hat{\alpha}_{ij}(t_k)\tilde{\alpha}_j(t_k)\},$$

where t_k^+ is the time just after updating at time t_k (at iteration k), with the initial condition $\tilde{\alpha}_i(t_0) = 1$. By multiplying with α_i on both sides of the above equation, we have

$$\begin{aligned} x_i(t_k^+) &= \max\{x_i(t_k), x_j(t_k) - \alpha_i[\alpha_{ij} - \hat{\alpha}_{ij}(t_k)]\} \\ &= \max\{x_i(t_k), x_j(t_k) - g_{ij}(t_k)\}, \end{aligned} \tag{12}$$

where $g_{ij}(t_k) = \alpha_i[\alpha_{ij} - \hat{\alpha}_{ij}(t_k)]$.

Theorem 3.3: Consider the skew update equation given by (12) with the initial condition $\tilde{\alpha}_i(t_0) = 1$. Then,

$$\Pr\{\lim_{k\to\infty} |x_i(t_k) - x_j(t_k)| = 0 | i, j \in V\} = 1. \tag{13}$$

Proof: The proof is omitted here due to space limitation. ∎

Theorem 3.3 guarantees that skew compensation converges with probability one, which means that the clock skews of all nodes may be able to be synchronized completely. Moreover, from (11) and the definition of $g_{ij}(t_k)$, we have

$$\Pr\{g_{ij}(t_k) = 0\} = \Pr\{\alpha_{ij} - \hat{\alpha}_{ij}(t_k) = 0\} \geq 1 - (1-\epsilon^2)^k.$$

Hence, it follows that

$$\begin{aligned} &\Pr\{g_{ij}(t_k) = 0 | \{i,j\} \in E\} \\ \geq & [1 - (1-\epsilon^2)^k]^{n^2} \geq 1 - n^2(1-\epsilon^2)^k. \end{aligned} \tag{14}$$

Meanwhile, if each $g_{ij}(t_k) = 0$, equation (12) is maximum consensus, and then it has $x_i(t_{k+n}) = x_j(t_{k+n}) = \max_{l=1}^n x_l(t_k)$ holds for $i, j \in V$, i.e., clock skew compensation is accomplished. Hence, the clock skew compensation under (12) converges exponentially in probability.

C. Offset Compensation

Similar to the skew compensation, the maximum consensus is also adapted to design the clock offset compensation algorithm. Specifically, when node i receives information $H_j^+(t_k)$ and $\tilde{\beta}_j(k)$ from neighbor node j, it updates its clock offset as

$$\begin{aligned} \tilde{\beta}_i(t_k^+) = \max\{&\tilde{\beta}_i(t_k), \tilde{\alpha}_j(t_k)(H_j^+(t_k) - b) \\ &+ \tilde{\beta}_j(k) - \tilde{\alpha}_i(t_k)H_i(t_k)\}, \end{aligned} \tag{15}$$

with initial condition $\tilde{\beta}_i(t_1) = 0$. The above equation satisfies

$$\begin{aligned} & \tilde{\beta}_i(t_k^+) + \tilde{\alpha}_i(t_k)\beta_i \\ = & \max\{y_i(t_k), \tilde{\alpha}_j(t_k)H_j(t_k) + \tilde{\beta}_j(k) \\ & \quad + \tilde{\alpha}_j(t_k)(\theta_j(t_k) - b) - x_i(t_k)t_k\} \\ = & \max\{y_i(t_k), (x_j(t_k) - x_i(t_k))t_k + y_j(t_k) \\ & \quad + \tilde{\alpha}_j(t_k)(\theta_j(t_k) - b)\}. \end{aligned} \tag{16}$$

Thus, we have

$$\begin{aligned} & y_i(t_k^+) - (\tilde{\alpha}_i(t_k^+) - \tilde{\alpha}_i(t_k))\beta_i \\ = & \max\{y_i(t_k), (x_j(t_k) - x_i(t_k))t_k + y_j(t_k) \\ & \quad + \tilde{\alpha}_j(t_k)(\theta_j(t_k) - b)\}. \end{aligned} \tag{17}$$

Theorem 3.4: Consider the offset update equation given by (17) with the initial condition $\tilde{\beta}_i(t_1) = 0$. Then,

$$\Pr\{\lim_{k\to\infty} |y_i(t_k) - y_j(t_k)| = 0 | i, j \in V\} = 1. \tag{18}$$

Proof: From Theorem 3.3, we have $\lim_{k\to\infty} \tilde{\alpha}_i(t_k) = \frac{c_i}{a_i}$, which means that

$$\begin{aligned} & \lim_{k\to\infty} |\tilde{\alpha}_i(t_k^+) - \tilde{\alpha}_i(t_k)| \\ \leq & \lim_{k\to\infty} |\tilde{\alpha}_i(t_k^+) - \frac{c_i}{a_i}| + \lim_{k\to\infty} |\tilde{\alpha}_i(t_k) - \frac{c_i}{a_i}| \leq 0 \end{aligned} \tag{19}$$

Since the clock skew compensation under (12) converges exponentially in probability, by Theorem 3.3, one can infer

$$\Pr\{\lim_{k\to\infty} (x_j(t_k) - x_i(t_k))t_k = 0 | i, j \in V\} = 1. \tag{20}$$

When (19) and (20) hold, we can simplify (17) as

$$y_i(t_k^+) = \max\{y_i(t_k), y_j(t_k) + \tilde{\alpha}_j(t_k)(\theta_j(t_k) - b)\}.$$

Since the limitation of each $\tilde{\alpha}_j(t_k)$ is $\frac{c_j}{\alpha_j}$ and $\Pr\{\theta_j(t_k) = b\} \geq \epsilon$, we have

$$\Pr\{\lim_{k\to\infty} |y_i(t_k) - y_j(t_k)| = 0\} = 1$$

for $\forall i, j \in V$. ∎

Theorem 3.4 guarantees that offset compensation converges with probability one, which means that the clock offset of all nodes may be able to be synchronized completely.

Since both of the skew and offset compensation converge with probability one, i.e., the goals (3) and (4) have been achieved using the above time synchronization algorithm, which guarantees that a highly accurate time synchronization can be achieved under the bounded noise, and a complete synchronization is realized when $t \to \infty$. Meanwhile, a larger ϵ will make the probability converges to one faster, which can be followed from (11) and (14). Moveover, the convergence of both the skew and the offset compensation are not affected by the value of the bounded noise.

IV. Performance Evaluation

In this part, we compare our algorithm with a typical ATS algorithm [8]. Since the maximum consensus is used in our algorithm and a accurate synchronization can be achieved under bounded noise, we name it as Noise-resilient Maximum-consensus-based Time Synchronization (NMTS).

Consider the network with 50 nodes which are randomly deployed in a $100\text{m} \times 100\text{m}$ area, and the maximum communication range of each node is 20m. For the simulation examples, we set the initial condition $\hat{\alpha}(0) = 1$ and $\hat{\beta}(0) = 0$ for both NMTS and ATS. As used in ATS, we set a common broadcast period to be one second. Note that the typical error for a quartz crystal oscillator is between 10 ppm and 100 ppm, [6], which corresponds to a 10 to 100 microsecond (μs) during the broadcast interval. Thus, each skew α_i in simulation is randomly selected from the set $[0.9999, 1.0001]$, and the offset $b_i$is randomly selected from the set $[0, 0.0002]$. The parameters used in ATS is set as $\rho_o = \rho_v = 0.5$ and $\rho_\eta = 0.2$, which are the same as in [8]. Since the value of the bounded noise cannot affect the convergence of our all algorithms, we set $a = 0$ and $b = 0.0001$, and $\Pr\{\theta_i(t_k) = a\} = \Pr\{\theta_i(t_k) = b\} = 0.04$ for each iteration k. We also define two functions as follows:

$$d_s(t) = \max_{i,j\in\mathcal{V}} (x_i(t) - x_j(t)),$$
$$d_o(t) = \max_{i,j\in\mathcal{V}} (y_i(t) - y_j(t)),$$

where $d_s(t)$ and $d_o(t)$ denote the maximum difference of the software skew and of the software offset between any two nodes, respectively. Clearly, the time synchronization is reached completely iff $d_s(t) = 0$ and $d_o(t) = 0$.

First, taking the bounded noise into consideration, we compare our relative skew estimation method used in NMTS and that in ATS, where in NMTS the method is given by (7) while in ATS it uses the weighted average of the current one-step estimation and the last-time estimation as the current relative skew estimation. Fig. 1 shows the estimated results of relative skew α_{12}. It is observed that using (7), α_{12} can be estimated accurately as $\hat{\alpha}_{12}(k)$ (the blue line) will converge to the ideal value α_{12} (the red line), while using the method in ATS, the average estimate error is about 0.0001.

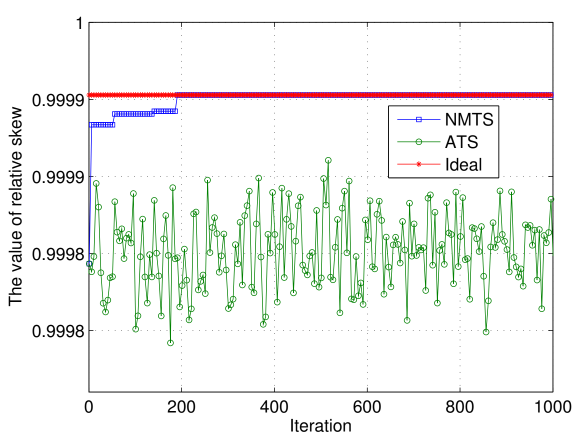
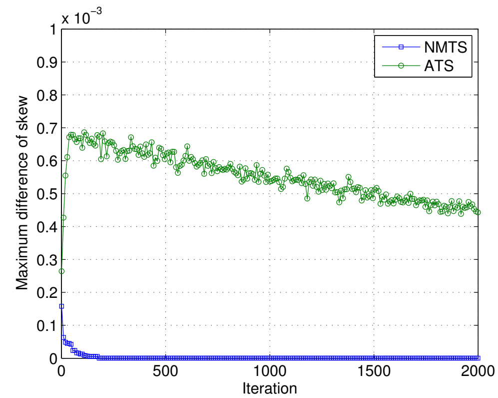
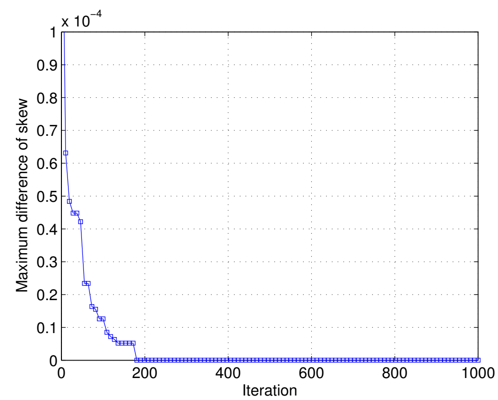


Fig. 1. The compare of relative skew estimation.

Then, we compare the performance of skew compensation for the two algorithms. As shown in Fig. 2, NMTS has a faster convergence speed and a higher synchronization accuracy than those of ATS. A more clear result about NMTS is shown in Fig. 2(b), in which we can see that the skew compensation converges after iteration 190 completely, which means that by NMTS all nodes' clock skews can be synchronized completely even with the noise. The results of NMTS validate the theoretical results in Theorem 3.3.



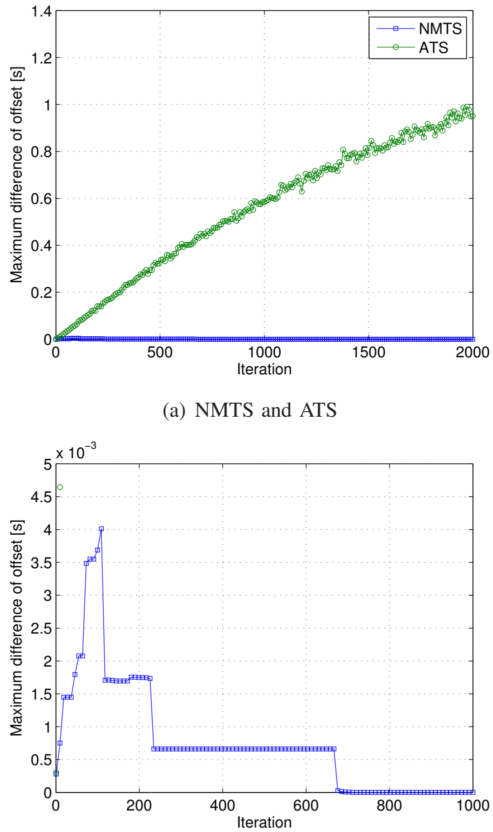
(a) NMTS and ATS



(b) NMTS

Fig. 2. The compare of skew compensation.

Finally, we compare the performance of offset compensation for NMTS and ATS. The result is given in Fig. 3. It is clear that NMTS has a much better accuracy in offset compensation. For ATS, the maximum difference of the software clock offset becomes worse as the iteration increases, which means that the synchronization is not fully achieved. The main reason is that ATS cannot synchronize the node's software clock skews completely under bounded noise. The difference between nodes' software clocks increases with time, and the offset cannot be compensated fast enough. Thus, Garone ea al. [13] have proposed a novel robust ATS algorithm to prevent the error from setting larger under random bounded communication delay. For NMTS, it is clear to see from Fig. 3(b) that although the maximum difference



(a) NMTS and ATS

(b) NMTS

Fig. 3. The compare of offset compensation.

between nodes' software clocks may also increase with time as the software clock skew have not yet been compensated sufficient in the early time, the difference can be reduced and converge to 0, which corresponds to Theorem 3.4.

V. Conclusions

This paper investigates distributed time synchronization under bounded noise for WSNs. By taking the advantage of the unique features of bounded monotonic sequence and the concept of maximum consensus, we propose a novel distributed time synchronization algorithm, including the relative skew estimation and software clock skew and offset compensations, to achieve accurate time synchronization. It's proved that our proposed algorithm can achieve complete time synchronization with probability one under bounded noise. Extensive simulations demonstrate that the proposed algorithm has a much faster convergence speed and a higher synchronization accuracy than that of typical average consensus-based time synchronization algorithm.

References

[1] B. Sundararaman, U. Buy and A. D. Kshemkalyani. Clock synchronization for wireless sensor networks: a survey. *Ad Hoc Networks*, 281–323, 2005.

[2] S. Ganeriwal, R. Kumar, and M. B. Srivastava. Timing-sync protocol for sensor networks. In *Proceedings of SenSys*, 2003.

[3] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. The flooding time synchronization protocol. In *Proceedings of SenSys*, 2004.

[4] M. Sichitiu and C. Veerarittiphan. Simple, accurate time synchronization for wireless sensor networks. In *Proceedings of IEEE WCNC*, 2003.

[5] F. Sivrikaya and B. Yener. Time Synchronization in sensor networks: A survey. *IEEE Network Magazine*, 2004.

[6] B. Choi, H. Liang, X. Shen, and W. Zhuang. DCS: Distributed asynchronous clock synchronization in delay tolerant networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(3): 491–504, 2012.

[7] S. Philipp and W. Roger. Gradient clock synchronization in wireless sensor networks. In *Proceedings of ACM/IEEE IPSN*, 2009.

[8] L. Schenato, and F. Fiorentin. Average timesynch: a consensus-based protocol for time synchronization in wireless sensor networks. *Automatica*, 47(9): 1878-1886, 2011.

[9] R. Carli, E. Elia, and S. Zampieri. A PI controller based on asymmetric gossip communications for clocks synchronization in wireless sensors networks. In *Proceedings of IEEE CDC-ECC*, 2011.

[10] R. Carli, and S. Zampieri. Network clock synchronization based on the second order linear consensus algorithm. *IEEE Transactions on Automatic Control*, 59(2): 409–422, 2014.

[11] J. He, P. Cheng, L. Shi, J. Chen and Y. Sun. Time synchronization in WSNs: A maximum-value-based consensus approach. *IEEE Transactions on Automatic Control*, 59(3): 660-675, 2014.

[12] J. He, H. Li, J. Chen, and P. Cheng. Study of consensus-based time synchronization in wireless sensor network. *ISA Transactions*, 53(2): 347-357, 2014.

[13] E. Garone, A. Gasparri, and F. Lamonaca. Clock synchronization for wireless sensor network with communication delay. In *Proceedings of IEEE ACC*, 2013.

[14] C. Liao, and P. Barooah. DiSync: Accurate distributed clock synchronization in mobile Ad-hoc networks from noise difference measurements. In *Proceedings of IEEE ACC*, 2013.

[15] N. M. Freris, H. Kowshik, and P. R. Kumar. Fundamentals of large sensor networks: connectivity, capacity, clocks, and computation. *Proceedings of the IEEE* , 98(11): 1828–1846, 2010.

[16] N. M. Freris, S. R. Graham, and P. R. Kumar. Fundamental limits on synchronizing clocks over networks. *IEEE Transactions on Automatic Control*, 56(2): 1352–1364, 2011.

[17] L. Mei and Y-C. Wu. Distributed clock synchronization for wireless sensor networks using belief propagation. *IEEE Transactions on Signal Processing*, 5404–5414, 2011.

[18] L. Mei and Y-C. Wu. Low-complexity maximum-likelihood estimator for clock synchronization of wireless sensor nodes under exponential delays. *IEEE Transactions on Signal Processing*, 4860–4870, 2011.

[19] Q. M. Chaudhari, E. Serpedin, and K. Qaraqe. On maximum likelihood estimation of clock offset and skew in networks with exponential delays. *IEEE Transactions on Signal Processing*, 1685–1696, 2008.

[20] J. He, P. Cheng, L. Shi, and J. Chen. SATS: Secure average-consensus-based time synchronization in wireless sensor networks. *IEEE Transactions on Signal Processing*, 61(24): 6387–6400, 2013.

[21] J. He, P. Cheng, J. Chen, and X. Cao. Secured time synchronization in wireless sensor networks: A maximum consensus based approach. *IEEE Transactions on Parallel and Distributed Systems*, 25(4): 1055–1065, 2014.

[22] C. Liao, and P. Barooah. Distributed clock skew and offset estimation from relative measurements in mobile networks with Markovian switching topology. *Automatica*, 49(10): 3015–3022, 2013.

[23] A. Garulli, and A. Giannitrapani. A set-membership approach to consensus problems with bounded measurement errors. In *Proceedings of IEEE CDC*, 2008.

[24] Y. Yang, L. Cai, Y. Liu, and J. Pan. Environment-aware clock skew estimation and synchronization for wireless sensor networks. In *Proceedings of IEEE INFOCOM*, 2012.

[25] X. Cao, J. Chen, Y. Zhang, and Y. Sun. Development of an integrated wireless sensor network micro-environmental monitoring system. *ISA transactions*, 247-255, 2008.

[26] M. Zhou, J. He, P. Cheng, and J. Chen. Discrete average consensus with bounded noise. In *Proceedings of IEEE CDC*, 2013.

[27] Z. Yang, X. Shi, and J. Chen. Optimal coordination of mobile sensors for target tracking under additive and multiplicative noises. *IEEE Transactions on Industrial Electronics*, 61(7): 3459–3468, 2014.