



CHALMERS

Chalmers Publication Library

Long-Term Clock Synchronization in Wireless Sensor Networks with Arbitrary Delay Distributions

This document has been downloaded from Chalmers Publication Library (CPL). It is the author's version of a work that was accepted for publication in:

Proc. IEEE Global Communications Conference (GlobeCom) (ISSN: 1930-529X)

Citation for the published paper:

Wanlu, S. ; Ström, E. ; Brännström, F. (2012) "Long-Term Clock Synchronization in Wireless Sensor Networks with Arbitrary Delay Distributions". Proc. IEEE Global Communications Conference (GlobeCom) pp. 359-364.

<http://dx.doi.org/10.1109/GLOCOM.2012.6503139>

Downloaded from: <http://publications.lib.chalmers.se/publication/163323>

Notice: Changes introduced as a result of publishing processes such as copy-editing and formatting may not be reflected in this document. For a definitive version of this work, please refer to the published source. Please note that access to the published version might require a subscription.

Chalmers Publication Library (CPL) offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all types of publications: articles, dissertations, licentiate theses, masters theses, conference papers, reports etc. Since 2006 it is the official tool for Chalmers official publication statistics. To ensure that Chalmers research results are disseminated as widely as possible, an Open Access Policy has been adopted. The CPL service is administrated and maintained by Chalmers Library.

(article starts on next page)

Long-Term Clock Synchronization in Wireless Sensor Networks with Arbitrary Delay Distributions

Wanlu Sun, Erik G. Ström, Fredrik Brännström, and Debarati Sen

Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden

{wanlu, erik.strom, fredrik.brannstrom, debarati}@chalmers.se

Abstract—Clock synchronization is a crucial issue in the operation of wireless sensor networks. Although the existing synchronization algorithms under linear clock model assumptions perform well for short periods, they will become problematic for applications with long-term requirements. In this paper, we consider a more realistic and flexible relationship model for two clocks and exploit a Taylor expansion to approximate the relationship. Based on this model and a two-way time message exchange procedure, an estimation algorithm is proposed to recover the relationship and then achieve the synchronization. Finally, simulation results demonstrate that the proposed algorithm improves the accuracy of synchronization as compared to existing algorithms in many scenarios, and is also robust to different distributions of random delays.

Index Terms—Wireless sensor network, clock synchronization, arbitrary delay distributions.

I. INTRODUCTION

Recently, wireless sensor networks (WSNs) have emerged as an interesting and important research area. As WSNs consist of several small scale devices and all the devices run their own clocks, clock synchronization becomes critical for efficient operation of WSNs.

Existing studies on clock synchronization mainly focus on the protocol design. However, the clock synchronization problem is inherently related to parameter estimation. The performance of synchronization can be improved by adopting a statistical signal processing framework.

In this paper we consider a simple network comprised by two nodes, A and B , with imperfect clocks. The nodes exchange a number of time stamps over a channel with random delays, and the data collected at Node B is used to estimate the clock values of Node A . Note that even though this paper considers the synchronization between a pair of nodes, extension to network-wide synchronization can be directly achieved by the combination of building a hierarchical structure and synchronizing two nodes with adjacent levels.

Currently, various schemes have been proposed for clock synchronization under the assumption that the imperfect clock is a linear function of the reference time (see, e.g., [1]–[5] and references therein). In this model, the frequency of a clock is assumed to be constant. However, this assumption as well as the associated algorithms are useful only for short-term applications (object tracking and surveillance [6]). They will become problematic for some applications having stringent and long-term clock synchronization requirements

(duty cycling and synchronized sampling [6]) as they may spend lot of energy on re-synchronization.

In fact, the frequency of a clock should include different orders of frequency drifts [7]. Correspondingly, the clock value is a not a linear function of time. Several papers have taken into account this non-linear property in clock synchronization [6], [8], [9]. In [6], the authors synchronize an imperfect clock to a perfect clock. That is, the value given by the perfect clock equals to the reference time, and the model of the imperfect clock is a quadratic function of the reference time. In general, however, we are required to do the synchronization between two imperfect clocks. In this case, the relationship between the values provided by the two clocks is no longer strictly a quadratic model.

To summarize, the main contributions of this paper are

1. the introduction of a framework for modeling the relationship between two imperfect clocks that is more general than what is typically considered in the existing literature;
2. the derivation of a novel algorithm for estimating Node A 's clock value from Node B 's clock value which
 - (a) is not dependent on accurate modeling of the random delay distributions;
 - (b) outperforms previously proposed synchronization algorithms in many scenarios;
 - (c) is possible to compute with reasonable complexity.

Notation: Uppercase boldface letters denote matrices and lowercase boldface letters designate vectors. The superscript $(\cdot)^T$ stands for the transposition; $\det(\cdot)$ is the determinant of a matrix; $\text{span}(\cdot)$ is defined as the linear span of a set of vectors; $\mathbb{E}[\cdot]$ and $\mathbb{V}[\cdot]$ indicate the expectation and variance, respectively. For any $m \times n$ matrix \mathbf{T} , $[\mathbf{T}]_{i,j}$ denotes the (i, j) -th element of \mathbf{T} . Furthermore, $\mathbf{T}(i, j_1 : j_N)$ denotes the row vector $([\mathbf{T}]_{i,j_1}, [\mathbf{T}]_{i,j_2}, \dots, [\mathbf{T}]_{i,j_N})$, and $\mathbf{T}(i_1 : i_N, j)$ denotes the column vector $([\mathbf{T}]_{i_1,j}, [\mathbf{T}]_{i_2,j}, \dots, [\mathbf{T}]_{i_N,j})^T$.

II. SYSTEM MODEL

A. Clock Value Relationship Model

Assume Node B needs to synchronize to Node A . The clock values of the two nodes are

$$T_A = g_A(t) \quad \text{and} \quad T_B = g_B(t), \quad (1)$$

respectively, where t is the reference time. Here, g_A and g_B are continuous, differentiable, and strictly increasing functions, since we assume clocks can neither stop nor run backwards. No other constraints are required on the functions g_A and g_B . For example, they can be linear functions, quadratic functions

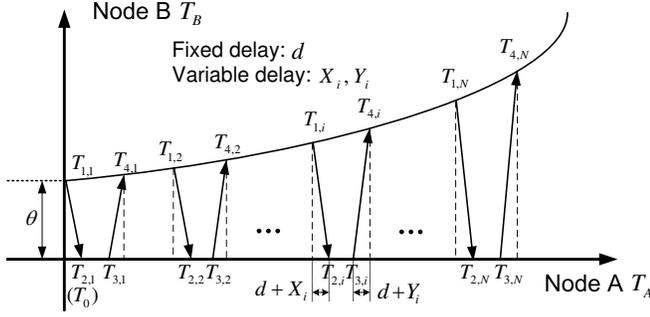


Figure 1. Two-way timing message exchange between nodes B and A .

or even polynomials with higher orders, which is more consistent to real clock models [7]. The important thing is that under any assumption of the clock model, there is always a continuous and differentiable function $g(\cdot)$ representing the relationship between T_B and T_A ,

$$T_B = g_B(g_A^{-1}(T_A)) = g(T_A). \quad (2)$$

We can expand the function $g(T_A)$ around the point $T_A = T_0$ through a Taylor expansion

$$\begin{aligned} T_B &= g(T_A) \\ &= g(T_0) + g'(T_0)(T_A - T_0) + \frac{g''(T_0)}{2}(T_A - T_0)^2 \\ &\quad + \sum_{k=3}^{\infty} \frac{g^{(k)}(T_0)}{k!}(T_A - T_0)^k. \end{aligned} \quad (3)$$

According to the real clock model, the coefficients of the higher-order terms in the Taylor expansion in (3) are very small [7]. Notice that, if we just keep the zeroth-order and first-order terms in (3), the model will reduce to the linear models in [1]–[5]. In this paper, to improve the precision of long-term synchronization, we retain the first three terms in (3) and approximate the clock value relationship as

$$\begin{aligned} T_B &\approx g(T_0) + g'(T_0)(T_A - T_0) + \frac{g''(T_0)}{2}(T_A - T_0)^2 \\ &= \theta + (T_A - T_0)f + (T_A - T_0)^2 D, \end{aligned} \quad (4)$$

where the parameters $\theta \triangleq g(T_0)$, $f \triangleq g'(T_0)$, and $D \triangleq g''(T_0)/2$ denote the coefficients of the zeroth-, first- and second-order terms, respectively. Based on the estimation of these three parameters $\hat{\theta}$, \hat{f} and \hat{D} , Node B can estimate Node A 's clock value as

$$\hat{T}_A \triangleq \frac{\sqrt{\hat{f}^2 - 4\hat{D}(\hat{\theta} - T_B)} - \hat{f}}{2\hat{D}} + T_0, \quad (5)$$

and the synchronization error can be defined as

$$E \triangleq \hat{T}_A - T_A. \quad (6)$$

We note that the statistical properties of E depend on the reference time t . Hence, we can think of E as a non-stationary random process (although the t -dependency is not explicit in the notation).

B. Time Stamp Exchange Model

Fig. 1 shows a mechanism of a two-way timing message exchange [10] between Node B and Node A , where timing messages are assumed to be exchanged N times. In the i -th round of exchange, Node B records its current clock value as the time stamp $T_{1,i}$ and sends a pulse to Node A at the same time. Node A records its clock value $T_{2,i}$ at the reception of that pulse. Then Node A sends at $T_{3,i}$ another message containing $T_{2,i}$ and $T_{3,i}$ to Node B . Finally, Node B records its clock value as $T_{4,i}$ when receiving the message. Note that $T_{1,i}$ and $T_{4,i}$ are the time stamps provided by Node B 's clock, while $T_{2,i}$ and $T_{3,i}$ are the time stamps recorded by Node A 's clock. Therefore, after N rounds of message exchanges, Node B has access to a set of time stamps $\{T_{1,i}, T_{2,i}, T_{3,i}, T_{4,i}\}_{i=1}^N$. Let T_0 be the clock value of Node A when it receives the first pulse, i.e., $T_0 = T_{2,1}$.

If there is no delay in the transmission between Node B and Node A , Node B will immediately know the relative clock value difference with respect to Node A . However, in a real wireless network, various delays have effects on the time stamp exchange procedure, which will complicate clock synchronization. These delays can be grouped into two parts: fixed delays and random delays. According to different applications and environments, there are different probability density function models for the random delays, including Gaussian, exponential, Gamma, Weibull and log-normal [11]. Consequently, synchronization algorithms are required to be robust to the various distributions of network delays.

Based on (4) and Fig. 1, the clock value difference between the two nodes can be calculated at $T_{1,i}$ and $T_{4,i}$ as

$$\begin{aligned} T_{1,i} &= \theta + (T_{2,i} - d - X_i - T_0)f + (T_{2,i} - d - X_i - T_0)^2 D \\ T_{4,i} &= \theta + (T_{3,i} + d + Y_i - T_0)f + (T_{3,i} + d + Y_i - T_0)^2 D \end{aligned} \quad (7)$$

where d denotes the fixed delay, $\{X_i\}_{i=1}^N$ and $\{Y_i\}_{i=1}^N$ represent the random delays in the transmissions from Node B to Node A and from Node A to Node B , respectively. Here $\{X_i\}_{i=1}^N$ and $\{Y_i\}_{i=1}^N$ are modeled as independent and identically distributed random variables with mean μ and variance σ^2 .

III. PROPOSED CLOCK SYNCHRONIZATION ALGORITHM

Based on (7), maximum likelihood estimation (MLE) can be formulated through maximizing the likelihood function. Unfortunately, with the above delay distributions, no closed-form expression is available for $\{\hat{\theta}, \hat{f}, \hat{D}\}$ using MLE. Also, the likelihood function is not convex or log-convex, so we cannot find effective numerical methods to guarantee the globally optimal $\{\hat{\theta}, \hat{f}, \hat{D}\}$. In this case, three-dimensional grid search scheme could be the choice for MLE, which is computationally expensive and not feasible in practice. Furthermore, MLE is designed for one specific delay distribution but not the general case, i.e., the robustness of MLE is not ensured under various delay distributions. Therefore, it is motivated to formulate an alternative estimator that is robust to arbitrary

delay distributions and to offer another trade-off between performance and complexity.

By adding the two equations in (7), we get

$$2\theta + (p_i + q_i - \tilde{X}_i + \tilde{Y}_i)f + \left((p_i - \tilde{X}_i)^2 + (q_i + \tilde{Y}_i)^2 \right) D = b_i, \quad (8)$$

where $\tilde{X}_i = X_i - \mu$, $\tilde{Y}_i = Y_i - \mu$, $p_i = T_{2,i} - d - \mu - T_0$, $q_i = T_{3,i} + d + \mu - T_0$, $b_i = T_{1,i} + T_{4,i}$, and $i = 1, 2, \dots, N$.

Rewriting (8) in a matrix form and separating the noise part \mathbf{U} from the observation \mathbf{A} gives

$$(\mathbf{A} - \mathbf{U})\boldsymbol{\alpha} = \mathbf{b}, \quad (9)$$

where

$$\mathbf{A} \triangleq \begin{pmatrix} 2 & p_1 + q_1 & p_1^2 + q_1^2 \\ 2 & p_2 + q_2 & p_2^2 + q_2^2 \\ & \dots & \\ 2 & p_N + q_N & p_N^2 + q_N^2 \end{pmatrix},$$

$$\mathbf{U} \triangleq \begin{pmatrix} 0 & \tilde{X}_1 - \tilde{Y}_1 & 2p_1\tilde{X}_1 - 2q_1\tilde{Y}_1 - \tilde{X}_1^2 - \tilde{Y}_1^2 \\ 0 & \tilde{X}_2 - \tilde{Y}_2 & 2p_2\tilde{X}_2 - 2q_2\tilde{Y}_2 - \tilde{X}_2^2 - \tilde{Y}_2^2 \\ & \dots & \\ 0 & \tilde{X}_N - \tilde{Y}_N & 2p_N\tilde{X}_N - 2q_N\tilde{Y}_N - \tilde{X}_N^2 - \tilde{Y}_N^2 \end{pmatrix},$$

$$\boldsymbol{\alpha} \triangleq (\theta, f, D)^\top,$$

$$\mathbf{b} \triangleq (b_1, b_2, \dots, b_N)^\top.$$

After observing (9), least squares (LS) is a natural way to do the estimation. However, because the ‘‘noise \mathbf{U} ’’ (variable delays) is subtracted from the observation matrix \mathbf{A} but not the vector \mathbf{b} , the LS algorithm will result in degraded performance [12]. Further, The authors of [12] introduce total least squares (TLS) to take the errors in \mathbf{A} into account by perturbing both \mathbf{A} and \mathbf{b} . TLS estimation can be expressed as the optimization problem

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{3 \times 1}, \Delta \mathbf{A} \in \mathbb{R}^{N \times 3}} \sum_{i=1}^N \|\Delta \mathbf{A}(i, 1:3)\|^2, \quad (10)$$

s.t. $(\mathbf{A} - \Delta \mathbf{A})\boldsymbol{\alpha} = \mathbf{b}$,

where $\Delta \mathbf{A}$ is the disturbance matrix to satisfy the equality constraint in (10). Similar to the basic idea of TLS, we also want to perturb the observation matrix \mathbf{A} to accord with the reality, and then minimize the amount of perturbation. However, in our problem, there are three major differences with the traditional TLS problem and they have effects on the algorithm design for enhanced performance: 1) the first column of \mathbf{U} is zero, which implies that the first column of disturbance matrix is required to be zero as well; 2) the second and third columns of \mathbf{U} are correlated, which implies that the decorrelating procedure should be considered when calculating the amount of perturbation; 3) matrix \mathbf{U} is row-wise independent and the covariance matrixes differ from row to row, which implies that the decorrelation should be completed independently for each row. From aforementioned reasons, we define the optimization problem as

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{3 \times 1}, \Delta \mathbf{A} \in \mathbb{R}^{N \times 3}} \sum_{i=1}^N \|\tilde{\boldsymbol{\Omega}}_i^{-\frac{1}{2}} \Delta \mathbf{a}_i^\top\|^2, \quad (11)$$

s.t. $(\mathbf{A} - \Delta \mathbf{A})\boldsymbol{\alpha} = \mathbf{b}$,

$$\Delta \mathbf{A}(1:N, 1) = \mathbf{0},$$

where $\Delta \mathbf{A}$ is the disturbance matrix to satisfy the equality constraints in (11), $\Delta \mathbf{a}_i \triangleq \Delta \mathbf{A}(i, 2:3)$, and $\tilde{\boldsymbol{\Omega}}_i \triangleq \mathbb{E}[(\mathbf{u}_i - \mathbb{E}[\mathbf{u}_i])^\top (\mathbf{u}_i - \mathbb{E}[\mathbf{u}_i])]$ is the covariance matrix of $\mathbf{u}_i \triangleq \mathbf{U}(i, 2:3)$. We assume $\tilde{\boldsymbol{\Omega}}_i$ is a positive semidefinite matrix, and, therefore, there exists another positive semidefinite matrix $\tilde{\boldsymbol{\Omega}}_i^{\frac{1}{2}}$ which satisfies $\tilde{\boldsymbol{\Omega}}_i^{\frac{1}{2}} \cdot \tilde{\boldsymbol{\Omega}}_i^{\frac{1}{2}} = \tilde{\boldsymbol{\Omega}}_i$. So $\tilde{\boldsymbol{\Omega}}_i^{-\frac{1}{2}} \Delta \mathbf{a}_i^\top$ can be interpreted as a decorrelating process. The calculation of $\tilde{\boldsymbol{\Omega}}_i$ depends on the distribution of variable delays, which is expressed as

$$\tilde{\boldsymbol{\Omega}}_i = \begin{pmatrix} 2\sigma^2 & 2\sigma^2(p_i + q_i) \\ 2\sigma^2(p_i + q_i) & 4\sigma^2(p_i^2 + q_i^2) + \boldsymbol{\Xi}_i \end{pmatrix}, \quad (12)$$

where $\boldsymbol{\Xi}_i = \mathbb{V}[\tilde{X}_i^2] + \mathbb{V}[\tilde{Y}_i^2] - 4p_i\mathbb{E}[\tilde{X}_i^3] + 4q_i\mathbb{E}[\tilde{Y}_i^3]$. Because the time stamp is given when the data stream goes into the physical layer, the values of random delays are very small and change slightly, i.e., the values of $\mathbb{V}[\tilde{X}_i^2]$, $\mathbb{V}[\tilde{Y}_i^2]$, $\mathbb{E}[\tilde{X}_i^3]$, and $\mathbb{E}[\tilde{Y}_i^3]$ are quite small. In order to achieve the robustness to different delay distributions in the synchronization scheme, we consider an approximation of $\tilde{\boldsymbol{\Omega}}_i$ as

$$\boldsymbol{\Omega}_i \triangleq 2\sigma^2 \begin{pmatrix} 1 & p_i + q_i \\ p_i + q_i & 2(p_i^2 + q_i^2) \end{pmatrix}. \quad (13)$$

Owing to the non-convexity, it is very difficult to solve the original optimization problem in (11) directly. Hence, the optimization procedure is divided into two steps. In **Step1**, for a fixed $\boldsymbol{\alpha}$, we find the optimal value of the objective problem (11), which is a function of $\boldsymbol{\alpha}$. In **Step2**, we take $\boldsymbol{\alpha}$ into account and then optimize (11). With this two-step strategy, (11) can be transformed into an unconstrained problem firstly and then be solved easily.

Step1: consider a component $i \in \{1, 2, \dots, N\}$ of the optimization problem (11) as,

$$h_i(\boldsymbol{\alpha}) = \min_{\Delta \mathbf{a}_i \in \mathbb{R}^{1 \times 2}} \|\boldsymbol{\Omega}_i^{-\frac{1}{2}} \Delta \mathbf{a}_i^\top\|^2, \quad (14)$$

s.t. $(\mathbf{a}_i - \Delta \mathbf{a}_i)\phi = b_i - 2\theta$,

where $\mathbf{a}_i \triangleq \mathbf{A}(i, 2:3)$ and $\phi \triangleq (f, D)^\top$.

Define two linear manifolds

$$\Upsilon_i \triangleq \{\boldsymbol{\xi} \in \mathbb{R}^{2 \times 1} : \boldsymbol{\xi}^\top \phi = b_i - 2\theta\}, \quad i = 1, 2, \dots, N, \quad (15)$$

$$\Upsilon_0 \triangleq \{\boldsymbol{\xi} \in \mathbb{R}^{2 \times 1} : \boldsymbol{\xi}^\top \phi = 0\}. \quad (16)$$

Consequently, the problem (14) is transformed into,

$$h_i(\boldsymbol{\alpha}) = \min_{\boldsymbol{\eta} \in \Upsilon_i} \|\boldsymbol{\Omega}_i^{-\frac{1}{2}} (\mathbf{a}_i^\top - \boldsymbol{\eta})\|^2. \quad (17)$$

In fact, each $\boldsymbol{\eta} \in \Upsilon_i$ equals $\boldsymbol{\eta} = \mathbf{s}_i + \mathbf{s}$, with $\mathbf{s}_i \in \Upsilon_i$ and $\mathbf{s} \in \Upsilon_0$. In this way,

$$h_i(\boldsymbol{\alpha}) = \min_{\mathbf{s} \in \Upsilon_0} \|\boldsymbol{\Omega}_i^{-\frac{1}{2}} ((\mathbf{a}_i^\top - \mathbf{s}_i) - \mathbf{s})\|^2 \quad (18)$$

$$= \|\boldsymbol{\Omega}_i^{-\frac{1}{2}} (\mathbf{a}_i^\top - \mathbf{s}_i)\|^2 - \min_{\mathbf{s}' \in \Upsilon_i} \|\boldsymbol{\Omega}_i^{-\frac{1}{2}} ((\mathbf{a}_i^\top - \mathbf{s}_i) - \mathbf{s}')\|^2 \quad (19)$$

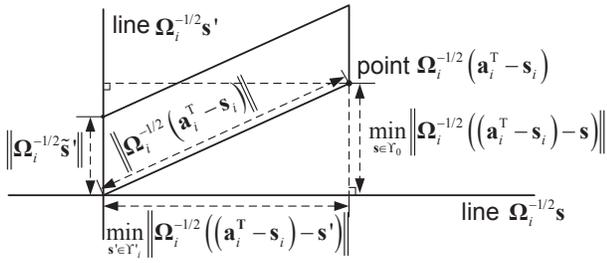


Figure 2. Geometrical illustration for (19) and (20).

$$= \|\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i)\|^2 - \frac{(\det([\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i), \Omega_i^{-\frac{1}{2}}\tilde{\mathbf{s}}'])^2}{\|\Omega_i^{-\frac{1}{2}}\tilde{\mathbf{s}}'\|^2}} \quad (20)$$

$$= \|\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i)\|^2 - \frac{(\det([\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i), \Omega_i^{-\frac{1}{2}}\phi])^2}{\|\Omega_i^{-\frac{1}{2}}\phi\|^2}}, \quad (21)$$

where $\Upsilon'_i = \text{span}(\Omega_i\phi)$, and $\tilde{\mathbf{s}}'$ could be any vector which belongs to Υ'_i . To pass from (18) to (19), we exploit the fact that the value of “ $\min_{\mathbf{s} \in \Upsilon_0} \|\Omega_i^{-\frac{1}{2}}((\mathbf{a}_i^\top - \mathbf{s}_i) - \mathbf{s})\|$ ” is the distance from “the point $\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i)$ ” to “the line $\Omega_i^{-\frac{1}{2}}\mathbf{s}$ ”, where $\mathbf{s} \in \Upsilon_0$; and from (19) to (20), we consider two facts: 1) the value of “ $\min_{\mathbf{s}' \in \Upsilon'_i} \|\Omega_i^{-\frac{1}{2}}((\mathbf{a}_i^\top - \mathbf{s}_i) - \mathbf{s}')\|$ ” is the distance from “the point $\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i)$ ” to “the line $\Omega_i^{-\frac{1}{2}}\mathbf{s}'$ ”, where $\mathbf{s}' \in \Upsilon'_i$; 2) the geometrical meaning of the determinant of a 2×2 matrix is that a area of the parallelogram is the absolute value of the determinant of the matrix $[\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i), \Omega_i^{-\frac{1}{2}}\tilde{\mathbf{s}}']$, formed by the column vectors $\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i)$ and $\Omega_i^{-\frac{1}{2}}\tilde{\mathbf{s}}'$ that defines the parallelogram’s sides [13]. Fig. 2 illustrates the derivations from (18) to (19) and from (19) to (20).

Moreover, we note that

$$(\Omega_i^{-\frac{1}{2}}(\mathbf{a}_i^\top - \mathbf{s}_i))^\top (\Omega_i^{-\frac{1}{2}}\phi) = \mathbf{a}_i\phi - b_i + 2\theta. \quad (22)$$

Therefore, equation (21) can be further simplified as

$$h_i(\boldsymbol{\alpha}) = \frac{(b_i - 2\theta - \mathbf{a}_i\phi)^2}{\phi^\top \Omega_i \phi} \quad (23)$$

$$= \frac{(2\theta + (p_i + q_i)f + (p_i^2 + q_i^2)D - b_i)^2}{2\sigma^2(f^2 + 2(p_i^2 + q_i^2)D^2 + 2(p_i + q_i)Df)}.$$

So far, we have finished **Step1**, and the original optimization problem (11) is equivalent to the following form,

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{3 \times 1}} \sum_{i=1}^N \frac{(2\theta + (p_i + q_i)f + a_{i2}D - b_i)^2}{f^2 + 2(p_i^2 + q_i^2)D^2 + 2(p_i + q_i)Df}, \quad (24)$$

which is an unconstrained optimization problem, and the optimal value of $\boldsymbol{\alpha}$ is independent of σ^2 .

Step2: after a change of variables, (24) becomes

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{3 \times 1}} \underbrace{\sum_{i=1}^N \frac{(\delta + (p_i + q_i) + (p_i^2 + q_i^2)\varepsilon - b_i\rho)^2}{1 + 2(p_i^2 + q_i^2)\varepsilon^2 + 2(p_i + q_i)\varepsilon}}_{\triangleq w(\boldsymbol{\beta})}, \quad (25)$$

where $\boldsymbol{\beta} \triangleq (\delta, \rho, \varepsilon)^\top$, $\delta \triangleq \frac{2\theta}{f}$, $\rho \triangleq \frac{1}{f}$, and $\varepsilon \triangleq \frac{D}{f}$. Now $w(\boldsymbol{\beta})$ is not a convex function over $\{\delta, \rho, \varepsilon\}$, and thus iterative algorithms may approach a local minimum. Nevertheless, for a fixed ε , $w(\boldsymbol{\beta})$ is convex over $\{\delta, \rho\}$. Differentiating the function $w(\boldsymbol{\beta})$ with respect to $\{\delta, \rho\}$, respectively, gives

$$\frac{\partial w(\boldsymbol{\beta})}{\partial \delta} = \sum_{i=1}^N \frac{2(\delta + (p_i + q_i) + (p_i^2 + q_i^2)\varepsilon - b_i\rho)}{1 + 2(p_i^2 + q_i^2)\varepsilon^2 + 2(p_i + q_i)\varepsilon}, \quad (26)$$

$$\frac{\partial w(\boldsymbol{\beta})}{\partial \rho} = \sum_{i=1}^N \frac{-2b_i(\delta + (p_i + q_i) + (p_i^2 + q_i^2)\varepsilon - b_i\rho)}{1 + 2(p_i^2 + q_i^2)\varepsilon^2 + 2(p_i + q_i)\varepsilon}. \quad (27)$$

By setting (26) and (27) to zero, the estimates of $\{\delta, \rho\}$ (here denoted by $\{\hat{\delta}, \hat{\rho}\}$) for a fixed ε can be obtained in closed forms.

Furthermore, to acquire the estimate of ε (here denoted by $\hat{\varepsilon}$), a one-dimensional grid search method is utilized. The grid search method offers some protections against local minimum, but is in general not very efficient. However, in our situation, because D is usually bounded in a small range and the value of f is usually close to 1, the grid search over the single parameter ε is fairly efficient. The search range can be decided by the prior knowledge about the oscillators. Finally, note that the optimal values of $\{\hat{\theta}, \hat{f}, \hat{D}\}$ is equivalent to that of $\{\hat{\delta}, \hat{\rho}, \hat{\varepsilon}\}$, since they are related by an invertible one-to-one transformation. In other words, through one-dimensional grid search method, three unknown parameters in the relationship model (4) can be estimated, and then the synchronization process can be achieved by calculating \hat{T}_A in (5).

To sum up, in the proposed scheme, based on a practical and very general clock model (1) in the long-term synchronization process, the consideration of the special properties of the relationship model (7) brings a more reasonable optimization problem (11). In addition, approximating the covariance matrix by Ω_i in (13), and then solving this problem with two-step implementation guarantee the precision, robustness, as well as low complexity. The complete procedure of the proposed clock synchronization scheme is explained in Algorithm 1.

Algorithm 1 Proposed Clock Synchronization Algorithm

- 1: Through the two-way timing message exchange mechanism in Fig. 1, Node B collects time stamps $\{T_{1,i}, T_{2,i}, T_{3,i}, T_{4,i}\}_{i=1}^N$.
- 2: With $\{T_{1,i}, T_{2,i}, T_{3,i}, T_{4,i}\}_{i=1}^N$ and the prior information of d and μ , calculate p_i and q_i , where $i \in \{1, 2, \dots, N\}$.
- 3: Select a range and step size for ε , perform a one-dimensional grid search method over ε , and solve the optimization problem (25) to obtain $\{\hat{\delta}, \hat{\rho}, \hat{\varepsilon}\}$. In each step of the grid search, $\{\delta, \rho\}$ are found by setting (26) and (27) to zero.
- 4: Calculate $\hat{\theta} = \frac{\hat{\delta}}{2\hat{\rho}}$, $\hat{f} = \frac{1}{\hat{\rho}}$, and $\hat{D} = \frac{\hat{\varepsilon}}{\hat{\rho}}$.
- 5: Based on (5), Node B estimates Node A 's clock value as \hat{T}_A .

IV. SIMULATION RESULTS

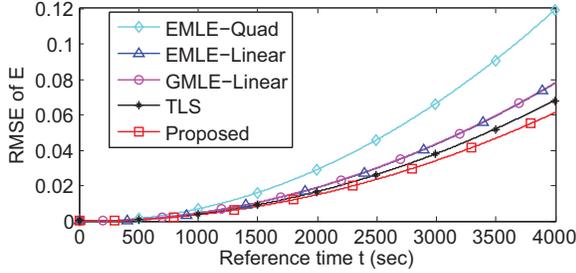
In this section, simulation results are depicted to compare the performance of the proposed clock synchronization algorithm with the following schemes: 1) TLS estimation in (10); 2) MLE for Gaussian delays with the assumption of linear

Table I
CLOCK MODEL SCENARIOS

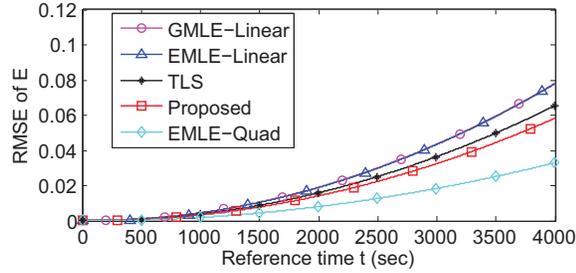
Node A \ Node B	Linear clock model	Quadratic clock model
Linear clock model	①	②
Quadratic clock model	③	④

Table II
CLOCK MODEL PARAMETERS

	Linear clock model	Quadratic clock model
$T_A = g_A(t)$	$1.000042t + 0.1$	$4.9 \cdot 10^{-9}t^2 + 1.000042t + 0.1$
$T_B = g_B(t)$	$1.000051t + 0.2$	$9.8 \cdot 10^{-9}t^2 + 1.000051t + 0.2$



(a) Gaussian distribution delays

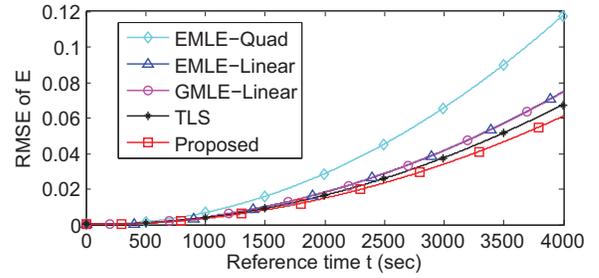


(b) Exponential distribution delays

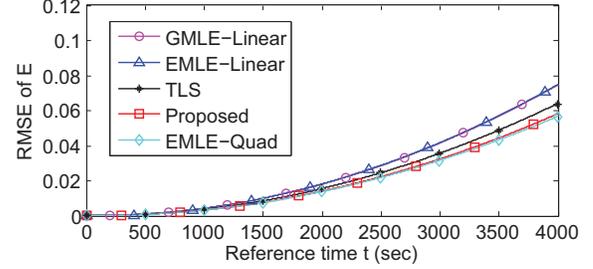
Figure 3. RMSE of the synchronization error E versus reference time t in model ④, with $N = 20$.

clock model (GMLE-Linear) [2]; 3) MLE for exponential delays with the assumption of linear clock model (EMLE-Linear) [5]; 4) MLE for exponential delays with the assumption of quadratic clock model (EMLE-Quad) [6]. Also, we consider four scenarios of clock models in Table I. Reasonable parameters of the clock models are given by Table II [14]. Moreover, in order to evaluate the robustness of different schemes against different distributed random delays, both Gaussian and exponential distributions of random delays are taken into account. As explained in Section III, since random delays are very small and change slightly, we set the mean as $\mu = 0.00002s$, and set the standard deviation as $\sigma = 0.00002s$. We choose the fixed delay as $d = 0.0001s$. Besides, the interval between the two-way timing messages, in terms of the reference time t , is assumed to be around $5s$. In addition, in the proposed algorithm, we set the step size to 5×10^{-11} and the range $[-10^{-8}, 10^{-8}]$ for the grid search to find $\hat{\epsilon}$. Finally, in all simulations, we consider the root mean square error (RMSE) of the synchronization error E in (6) as the performance metric, i.e., $\mathbb{E}[E^2]^{1/2}$.

Fig. 3 describes the RMSE of the synchronization error versus time for five algorithms for Gaussian and exponential



(a) Gaussian distribution delays



(b) Exponential distribution delays

Figure 4. RMSE of the synchronization error E versus reference time t in model ②, with $N = 20$.

distributed delays, under the assumption of model ④. In this case, there is some loss of precision during the approximation of the clock relationship in (4). In Fig. 3, GMLE-Linear and EMLE-Linear exhibit close performance regardless of the type of random delays. However, both of them suffer from the mismatch of clock models and consequently have this performance degradation. Compared to them, TLS has improved accuracy due to its consideration of the quadratic clock relationship. The TLS performance, however, is limited by the reasons mentioned in Section III. Besides, it is revealed from Fig. 3(b) that EMLE-Quad shows the best performance in the environment with exponential distributed delays. However, when the random delays have Gaussian distribution, EMLE-Quad deteriorates severely in Fig. 3(a), which illustrates the sensitivity of this algorithm to different delay distributions. Furthermore, the proposed algorithm presents an improvement compared to other methods in Fig. 3(a), and maintains this synchronization precision in Fig. 3(b), which demonstrate its good accuracy and robustness to different delay environments.

When model ③ is assumed, the approximated relationship (4) is exactly the same as the real one given by (2). Under this assumption, all estimators have similar performance as in Fig. 3. Hence, results for model ③ are not shown due to the space limitation. This performance similarity demonstrates that the simplification of the clock value relationship in (4) is reasonable.

Fig. 4 shows the corresponding results for model ②. Here, the approximation in (4) results in some accuracy loss as in model ④. By comparing Fig. 3 and Fig. 4, we find that GMLE-Linear, EMLE-Linear, TLS, and the proposed algorithm have similar performance in these two figures. On the other hand, in Fig. 4(b), the advantage of EMLE-Quad becomes not as large as in Fig. 3(b) although the delays

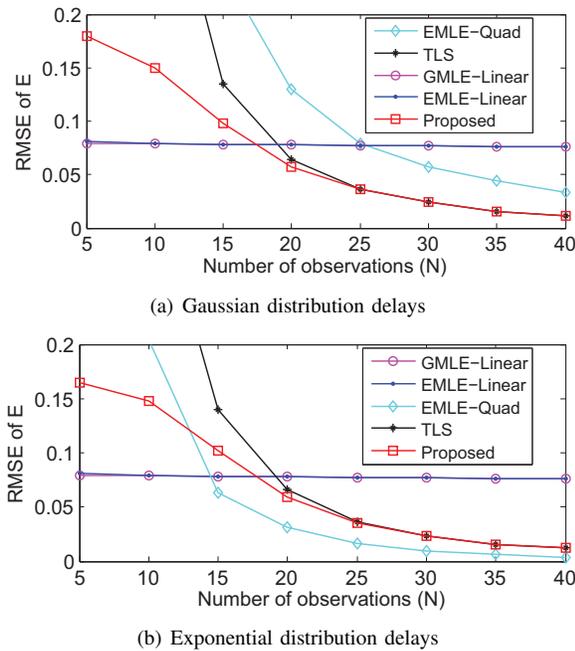


Figure 5. RMSE of synchronization error E versus number of observations N at $t = 4000$ s in model ④.

come from exponential distribution here. This phenomenon indicates that EMLE-Quad is not robust to various clock model scenarios. Furthermore, the sensitivity of EMLE-Quad to different delay distributions can also be seen from Fig. 4.

Due to limited space, simulation results are not presented for model ①. In this case, the performance of the methods that consider 3-dimensional estimation, e.g., TLS, EMLE-Quad, and the proposed algorithm, are worse than that of 2-dimensional estimators, e.g., GMLE-Linear and EMLE-Linear. This is no surprise, since the clock value relationship in model ① is perfectly matched to the relationship in the deduction of GMLE-Linear [2] and EMLE-Linear [5]. In reality, however, inexpensive clocks at sensor nodes will not have ideal linear models. The non-linear parts of clock models play an important role and should be considered for long-term synchronization requirements.

In Fig. 5, under the assumption of model ④, we compare the RMSE of the synchronization errors with respect to the number of observations for different algorithms when $t=4000$ s. Firstly, for GMLE-Linear and EMLE-Linear, their performance curves are almost flat with increasing N , which means these two schemes have severe floor effects. This phenomenon is a result from clock model mismatch. Secondly, in the scenario with exponential distributed delays, EMLE-Quad shows better performance than the proposed algorithm when N is larger than 13. This is because MLE has asymptotic properties [2]. Nevertheless, EMLE-Quad may fail when N is small and, as mentioned earlier, EMLE-Quad does not work well if the random delays have a Gaussian distribution, which is again evident in Fig. 5(a). Moreover, when we increase N , the performance of TLS and the proposed algorithm will become closer and closer. This is because in the TLS estimation (9), the negative influence of the correlated matrix

$\Delta \mathbf{A}$ will get smaller with the increased N . However, TLS gives unacceptable accuracy when we have small number of observations (e.g., $N < 15$). Finally, the proposed algorithm has good performance under both of the two delay distribution assumptions. At the same time, it exhibits improved performance when we increase N . Also, it will not fail even if only a small number of observations is available.

V. CONCLUSION

In this paper, we consider clock synchronization for WSNs applications with long-term synchronization requirements. Through a Taylor expansion, the general relationship between two imperfect clocks is approximated as a quadratic function as opposed to the commonly assumed linear relationship. Based on the quadratic model and a two-way time message exchange mechanism, an algorithm is proposed to estimate the coefficients in the relationship model. The proposed algorithm exhibits improved accuracy and robustness to different delay distributions compared to a number of previously proposed algorithms, since the special characteristics of the system model are taken into account. At the same time, acceptable complexity is guaranteed by dividing the estimation procedure into two steps.

REFERENCES

- [1] Y. Wu, Q. M. Chaudhari, and E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 124–138, Jan. 2011.
- [2] K. Noh, Q. M. Chaudhari, E. Serpedin, and B. W. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Trans. Comm.*, vol. 55, no. 4, pp. 766–777, Apr. 2007.
- [3] Q. M. Chaudhari, E. Serpedin, and K. Qaraqe, "On minimum variance unbiased estimation of clock offset in a two-way message exchange mechanism," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2893–2904, Jun. 2010.
- [4] Q. Chaudhari, E. Serpedin, and K. Qaraqe, "On maximum likelihood estimation of clock offset and skew in networks with exponential delays," *IEEE Trans. Signal Processing*, vol. 56, no. 4, pp. 1685–1697, Apr. 2008.
- [5] M. Leng and Y. Wu, "On joint synchronization of clock offset and skew for wireless sensor networks under exponential delay," in *Proc. IEEE ISCAS*, Jun. 2010, pp. 461–464.
- [6] Q. Chaudhari, and E. Serpedin, "Clock estimation for long-term synchronization in wireless sensor networks with exponential delays," *EURASIP Journal on Advances in Signal Processing*, Jan. 2008.
- [7] W. C. Lindsey, F. Ghazvinian, W. C. Haggmann, and K. Dessouky, "Network synchronization," *Proceedings of IEEE*, vol. 73, no. 10, pp. 1445–1467, Oct. 1985.
- [8] B. M. Sadler "Local and broadcast clock synchronization in a sensor node," *IEEE Signal Processing Letters*, vol. 13, no. 1 pp. 9–12, Jan. 2006.
- [9] Y. Quan and G. Liu, "Drifting clock model for network simulation in time synchronization," *IEEE ICICIC*, Jun. 2008, pp. 385–388.
- [10] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync protocol for sensor networks," in *Proc. SenSys 03*, Nov. 2003, pp. 138–149.
- [11] C. Bovy, H. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. Miegheem, "Analysis of end-to-end delay measurements in Internet," in *Proc. Passive and Active Measurements Workshop*, Mar. 2002, pp. 26–33.
- [12] G. H. Golub, and C. F. Van Loan, "An analysis of the total least squares problem," *SIAM J. Numer. Anal.*, vol. 17, no. 6, pp. 883–893, 1980.
- [13] S. Kumaresan, *Linear Algebra: A Geometric Approach*. Prentice Hall of India, 1999, pp. 131–132.
- [14] F. Ferrari, A. Meier, and L. Thiele, "Accurate Clock Models for Simulating wireless sensor networks," *OMNeT++*, Mar. 2010.