Robust Clock Synchronization in Wireless Sensor Networks Through Noise Density Estimation

Jang-Sub Kim, Jaehan Lee, Erchin Serpedin, and Khalid Qaraqe, *Senior Member, IEEE*

***Abstract*—Assuming that the network delays are normally distributed and the network nodes are subject to clock phase offset errors, the maximum likelihood estimator (MLE) and the Kalman filter (KF) have been recently proposed with the goal of maximizing the clock synchronization accuracy in wireless sensor networks (WSNs). However, because the network delays may assume any distribution and the performance of MLE and KF is quite sensitive to the distribution of network delays, designing clock synchronization algorithms that are robust to arbitrary network delay distributions appears as an important problem. Adopting a Bayesian framework, this paper proposes a novel clock synchronization algorithm, called the Iterative Gaussian mixture Kalman particle filter (IGMKPF), which combines the Gaussian mixture Kalman particle filter (GMKPF) with an iterative noise density estimation procedure to achieve robust performance in the presence of unknown network delay distributions. The Kullback-Leibler divergence is used as a measure to assess the departure of estimated observation noise density from its true expression. The posterior Cramér–Rao bound (PCRB) and the mean-square error (MSE) of IGMKPF are evaluated via computer simulations. It is shown that IGMKPF exhibits improved performance and robustness relative to MLE. The prior information plays an important role in IGMKPF. A MLE-based method for obtaining reliable prior information for clock phase offsets is presented and shown to ensure the convergence of IGMKPF.**

***Index Terms*—Clock synchronization, Cramér–Rao bound, filter, Kalman, particle filter, wireless sensor networks (WSNs).**

I. Introduction

CLOCK synchronization in wireless sensor networks (WSNs) has been a topic of extensive research over the last few years [1]–[4]. WSN applications that implement time-based collaboration of one or more nodes require its sensor nodes to be time synchronized. Clock synchronization is important for many other reasons. When an event occurs in WSNs, it is often necessary to know where and when it occurred. Time synchronization is also required for many system and application tasks such as sleep/wake-up scheduling, localization, data fusion, tracking and velocity estimation.

The key parameters used for evaluating the clock offset between two nodes are phase offset, skew and drift. Due to the unpredictability and imperfect measurability of message delays in a networked environment, physical clock synchronization is always imperfect. Obtaining more accurate and energy-efficient synchronization protocols for WSNs that achieve the best performance limits represents a fundamental design problem in the deployment of large-scale sensor networks. This is due in part to the fact that the energy and bandwidth constraints in WSNs are very strict. Therefore, using additional overhead (message exchanges) for more accurate synchronization is not a reliable solution. To overcome this challenge, one approach is to reduce the amount of energy spent on RF signal transmissions by transmitting fewer messages and by using high performance statistical signal processing techniques to improve the accuracy of clock synchronization [5], [6]. This fact is corroborated by Pottie and Kaiser's findings who showed in [8] that the RF energy required to transmit 1 kbit over 100 m (i.e., 3 Joules) is equivalent to the energy required to execute 3 millions of instructions. Thus, the computational power in a sensor node can be traded for reduced RF transmission energy.

Clock synchronization between any two nodes is generally accomplished by message exchanges. Due to the presence of nondeterministic and possible unbounded message delays, messages can be delayed arbitrarily, which makes the clock synchronization very difficult [9]. The most commonly proposed network delay distributions are the Gaussian and exponential probability density functions (pdfs) [10]–[12]. In general, it is difficult if not impossible to assess which distribution model may be fit to capture the network delay distributions in a given WSN. This is due to the fact that various factors might impact differently the distribution of network delays [18], [19]. Therefore, one important problem is to design clock offset estimation schemes which are robust to the distribution of unknown network delays.

The Gaussian [11] and exponential pdfs [10] were recently proposed to model the network delays in WSNs, and the corresponding maximum likelihood estimators (MLE) were derived in [12]. The MLEs for clock offset estimation in the presence of symmetric Gaussian and exponential network delay distributions will be referred to as MLEg and MLEe, respectively. Notice that the symmetry condition herein refers to the fact that the network delays assume the same distribution in the uplink and downlink. Reference [12] also proposed a lower complexity estimator although its performance is degraded with respect to the MLE. On the other hand, [13] proposed an alternative low complexity estimator that achieves the same performance as the MLE. In [12], it is shown that MLEg and MLEe are quite sensitive to the network delay distributions. Also, the Cramér–Rao lower bound (CRLB) is shown to be proportional to the variance of the network delay noise, and inverse proportional to the number of observations [12]. Thus, it appears that to improve

Manuscript received May 09, 2010; revised October 28, 2010 and March 01, 2011; accepted March 29, 2011. Date of publication April 11, 2011; date of current version June 15, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ut-Va Koc. This work was supported by Qtel and QNRF-NPRP.

The authors are with the ECEN Department, Texas A&M University, College Station, TX 77843 USA (e-mail: serpedin@ece.tamu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2011.2141660

the performance, MLEg and MLEe require a larger number of observations. However, since WSNs are power-limited systems, such a solution might not be appropriate. Therefore, there is a need for alternative methods to improve the accuracy of clock offset estimation.

Because of the uncertainties in modeling the network delay distributions and local clock oscillators, as well as the possible time-variations in the clock values, herein paper we will adopt a Bayesian approach for estimating the clock phase offset between two nodes. No skew or drift is assumed between the clocks. The signaling mechanism between the two nodes is the standard two-way message exchange mechanism encountered in standard protocols such as NTP [14], PBS [15], [16], and TPSN [17]. Because the synchronization framework corresponding to the two way message exchange mechanism can be mathematically described in terms of a linear Gauss-Markov state-space model, the problem of clock phase offset estimation is reduced to the estimation of the state of a linear Gauss-Markov state-space model, where the observation noise representing the distribution of network delays may assume an arbitrary distribution. A particle filter (PF) could be applied for estimating the state. A PF-based method provides an approximate Bayesian solution to the discrete-time recursive problem by updating an approximate description of the posterior filtering density. Since the PF cannot calculate the likelihood function, due to the unknown measurement noise density, the proposal distribution plays an important role in the performance of PF. Notice also that there are a lot of particle filtering techniques that could be applied for increasing the estimation accuracy. However, in general, particle filtering techniques assume inefficient proposal distributions and the observation noise density is in general assumed to be *a priori* known and modeled in terms of its first two moments: mean and variance. Therefore, in general there exists a bias in both the MLE and general PFs regardless of whether the network delay model is Gaussian or non-Gaussian, and this is due to the finite number of observations. Thus, the MLE and CRLB cannot serve as an efficient estimator and tight lower bound, respectively. In addition, the PF may not be an efficient estimator due to the finite number of observations and unknown observation noise density. The recursive posterior Cramér–Rao bound (PCRB) has been shown to be the information-theoretic mean-square error (MSE) bound for an unbiased sequential Bayesian estimator [30]. However, the expectation integrals for the Fisher information components, which arise out of the recursive PCRB formulation, are intractable in general and must be approximated numerically. Therefore, herein paper we will able to plot simulated values for PCRB.

To cope with the limitations of MLE and PF, in this paper we propose a novel clock phase offset estimator, called the iterative Gaussian mixture Kalman particle filter (IGMKPF), and analyze the PCRB as a lower bound on its MSE-performance. The general features of IGMKPF are next described. First of all, IGMKPF is capable of tracking the posterior and observation noise densities in order to reduce the bias which might stem from the observation noise estimation and the effects induced by the finite number of observations. Therefore, IGMKPF presents robustness to network delays of arbitrary distribution and enhanced MSE-performance. In general, if the estimator is able to track the original noise density, and not the mean and variance of noise (moments), then the bias caused by expectation and finite number of observations might be reduced, which further leads to improved MSE-performance. Second, the proposal distribution is chosen to reduce the error accumulation due to the iterative mechanism and the adopted initializations are ensuring the convergence of IGMKPF. Third, the estimator deals with non-Gaussian noise efficiently by adopting Gaussian mixture models (GMMs) to capture general densities. The proposed IGMKPF estimation approach combines the Gaussian mixture Kalman particle filter (GMKPF) proposed in [22] with a network delay density estimator by means of an iterative scheme. IGMKPF tracks the real probability density (prior distribution, posterior distribution, etc.) by using random and deterministic sampling methods. Therefore, the more accurate the estimation of network delays distribution is, the better the performance of IGMKPF is due to the improved proposal distribution.

Thus far, in the synchronization literature for WSNs, Kalman filtering and general adaptive signal processing techniques have been proposed (see, e.g., [23]–[27]) to improve the MSE-performance of protocols such as RBS [11] or TPSN [17] under the assumption that the network delays are Gaussian. To deal with non-Gaussian noise, [22] proposed recently GMKPF, which was shown to perform well in a number of distributions: asymmetric exponential, asymmetric gamma, asymmetric Weibull, and mixture of them. However, one critical condition assumed by GMKPF is the *a-priori* knowledge of network delay distribution. As seen in [22], in GMKPF, the importance sampling (IS) based measurement update step is combined with the time-update and proposal density generation steps, which exploit a Kalman filter (KF)-based Gaussian sum filter. Since GMKPF utilizes new observations and uses the expectation-maximization (EM) algorithm to estimate the parameters of the Gaussian mixture models, GMKPF exhibits better estimation performance relative to MLEg and MLEe in general asymmetric network delay models whose distributions are assumed known. However, the performance of GMKPF is limited in the presence of unknown observation noise density. In cases when the initial observation noise density is far away from the real observation noise density, the performance of GMKPF might be poor due to its slow convergence. Therefore, this paper could be seen as a extension of our previous results in [22] to handle the critical situation of unknown observation noise distribution.

In Bayesian statistics, the prior information could come from operational or observational data, from previous comparable experiments or from engineering knowledge. Since the prior information is important for convergence and performance, a simple method for generating reliable prior information consists in using the ML-estimate as an initialization. First of all, one can consider the clock offset estimate yielded by MLE [12] as an initialization $x_0 = \hat{x}_{\mathrm{ML}}$. Next, the prior density $p(x_0) = p(\hat{x}_{\mathrm{ML}})$ can be built using a set of samples (particles) that approximate the required density or distribution. Given the observation equation and prior information (see Fig. 4), one can infer the observation noise density. Since the estimated clock offset $(\hat{x}_{\mathrm{ML}})$ is reliable, the prior information will be reliable. To validate the ac-

curacy of IGMKPF, the PCRB is derived and a sequential Monte Carlo simulation approach is used for computing the PCRB in unknown delay models. Computer simulations are conducted to compare MLEg, MLEe, CRLB, IGMKPF, IGMKPF with perfect network delay noise estimation, and PCRB. As a result, when the accuracy of noise distribution estimation is improved, the performance of IGMKPF is significantly better than MLE in the presence of a reduced number of observations and arbitrary network delay distributions. In order to assess the process of estimating the unknown network delay density, we introduce the Kullback-Leibler divergence (KLD) as a measure between the estimated noise density and the true density. The simulation results show that the KLD is roughly proportional to the MSE of IGMKPF. To derive the PCRB, the second-order derivatives of posterior pdf must be evaluated. In case of Gaussian network delays, the second-order derivatives of posterior pdf can be calculated in closed-form expression. However, for exponential network delays and arbitrary delay distributions, the second-order derivatives of posterior pdf cannot be calculated directly. Therefore, we derive simulated PCRBs for Gaussian, exponential, and gamma network delays, and then compare them to CRLB. Computer simulations show that IGMKPF presents improved performance relative to MLE.

In this paper, upon designing the IGMKPF, we first carry out a performance analysis of IGMKPF, MLEg, and MLEe in the two-way message exchange mechanism between two nodes under symmetric Gaussian, exponential, and Gamma network delay distributions. In order to assess the performance of the proposed estimator, we derive PCRBs for Gaussian, exponential, and gamma delay models, respectively. The performance of IGMKPF, IGMKPF with perfect noise estimation, MLE, CRLB, and PCRB is simulated under Gaussian and exponential delays. In addition, the performance of IGMKPF, IGMKPF with perfect noise estimation, MLEe, PCRB is simulated under the gamma delay model. To illustrate the effects of noise estimation, we compare KLD with the MSE of IGMKPF. The computer simulation results corroborate the superior performance of the proposed method relative to MLEg and MLEe, and its robustness to general (Gaussian, exponential, gamma) network delay distributions. Therefore, the proposed IGMKPF method represents a reliable clock offset estimation scheme fit to overcome the uncertainties caused by the unknown network delay distributions.

The rest of this paper is organized as follows. Section II formulates the problem and introduces the state-space clock phase offset estimation model which is used throughout the paper. Section III derives the PCRB for the Gaussian, exponential, and gamma delay models. Section IV provides a description of the IGMKPF approach for estimating the clock offset in WSNs. The results of computer simulations are given in Section V. Finally, concluding remarks are presented in Section VI.

II. Problem Formulation and Objectives

The two-way timing message exchange mechanism is a recently proposed clock synchronization scheme for wireless sensor networks [12], [16], [17]. In this mechanism, the synchronization of two nodes A and B is achieved through a number of N cycles. Each cycle assumes two message transmissions: one from node A to node B, followed by a reverse transmission from node B to node A. At the beginning of the kth cycle, the node A sends its time reading $T_{1,k}$ to node B, which records the arrival time of the message as $T_{2,k}$, according to its own time scale. Similarly, a time message exchange is performed from node B to node A. At time $T_{3,k}$, node B transmits the time information $T_{2,k}$ and $T_{3,k}$ back to node A. Denoting by $T_{4,k}$ the arrival time at node A of the message sent by node B, node A would then have access to the time information $T_{j,k}$, $j = 1, \ldots, 4$ at the end of the kth cycle, which provide sufficient information for estimating the clock phase offset θ_A of node A relative to node B clock.

Similarly to [12], the differences between the kth up and down-link delay observations corresponding to the kth timing message exchange are defined by $U_k := T_{2,k} - T_{1,k} = d + \theta_A + L_k$ and $V_k := T_{4,k} - T_{3,k} = d - \theta_A + M_k$, respectively. The fixed value d denotes the fixed (deterministic) propagation delay component (which in general is neglected $(d \approx 0)$ in small range networks that assume RF transmissions). Parameters L_k and M_k stand for the variable portions of the network delays, and may assume any distribution such as Gaussian, exponential, Gamma, Weibull or mixture of two different distributions.

Given the observation samples $\mathbf{z}_k = [U_k, V_k]^T$, our goal is to find the minimum mean-square error estimate of the unknown clock offset θ_A. For convenience, the notation $x_k := \theta_A$ will be used henceforth. Thus, it turns out that we need to determine the estimator

$$\hat{x_k} = E\{x_k | \mathbf{z}^l\} \tag{1}$$

where $\mathbf{z}^l$ denotes the set of observed samples up to time l, $\mathbf{z}^l = \{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_l\}$. Since the clock offset value is assumed to be a constant, the clock offset can be modeled as following the Gauss-Markov model:

$$x_k = F x_{k-1} + v_{k-1} \tag{2}$$

where F stands for the state transition matrix of the clock offset. The additive process noise component v_k can be modeled as Gaussian with zero mean and covariance $E[v_k v_k^T] = Q = \sigma_v^2$. The vector observation model follows from the observed samples and it assumes the following expression:

$$\mathbf{z}_k = \begin{bmatrix} d + x_k + L_k \\ d - x_k + M_k \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} d + \begin{bmatrix} 1 \\ -1 \end{bmatrix} x_k + \mathbf{n}_k \tag{3}$$

$$= \mathbf{A}d + \mathbf{B}x_k + \mathbf{n}_k \tag{4}$$

where $\mathbf{A} = [1\ 1]^T$, $\mathbf{B} = [1\ -1]^T$, and the observation noise vector $\mathbf{n}_k = [L_k, M_k]^T$ has zero mean and covariance $R = \mathrm{diag}\{\sigma_n^2, \sigma_n^2\}$, and it accounts for the random network delays. One can now observe that (2) and (4) recast our initial clock offset estimation problem into a Gauss-Markov estimation problem with unknown state. Notice that the state-space model depicted by the (2)–(4) subsumes as a special case the original measurement equations. Choosing $F = 1$ and $v_{k-1} = 0$ in (2) leads to a constant phase-offset model. However, the Gauss-Markov state-space equations (2)–(4) represent a more realistic model because they can track time-varying clock phase offsets. This corroborates the experimental results concerning the drifting of clock values due to aging, temperature, air pressure, and other unpredictable factors.

III. PCRB for Sequential Bayesian Estimation

We need a lower bound on the MSE of the estimator, $\hat{x}_k$ for the true state x_k, defined by (2) and (4). Since we are interested in the class of trackers that are unbiased, bounding the MSE can be achieved by the PCRB alone. Assuming that the regularity condition holds for the probability density functions, the posterior Cramer-Rao bound (PCRB or Bayesian CRLB) [30] provides a lower bound on the MSE matrix for random parameters. Let $p(\mathbf{z}, x) = p(\mathbf{z}|x)p(x)$ denote the joint probability density function (pdf) of $\mathbf{z}$ and x, where $p(x)$ and $p(\mathbf{z}|x)$ stand for the *a-priori* pdf of x and the conditional pdf of $\mathbf{z}$ given x, respectively. Letting $\hat{x}(\mathbf{z})$ denote an estimate of x which is a function of the observations $\mathbf{z}$, the MSE matrix is

$$\mathbf{M} = E_{\mathbf{z},x}\left\{\left[\hat{x}(\mathbf{z}) - x\right]\left[\hat{x}(\mathbf{z}) - x\right]^T\right\}. \tag{5}$$

The PCRB $\mathbf{C}$ provides a lower bound on the MSE matrix $\mathbf{M}$, and it is expressed as the inverse of the Bayesian Fischer Information Matrix (BFIM) $\mathbf{J}$

$$\mathbf{M} \geq \mathbf{C} \equiv \mathbf{J}^{-1}. \tag{6}$$

The BFIM for x is defined as

$$\mathbf{J} = E_{\mathbf{z},x}\left\{-\triangle_x^x \ln p(\mathbf{z}, x)\right\} \tag{7}$$

where $\triangle_\phi^\theta$ is the $m \times n$ matrix of second-order partial derivatives with respect to the $m \times 1$ parameter vector ϕ and $n \times 1$ parameter vector θ

$$\triangle_\phi^\theta = \begin{pmatrix} \frac{\partial^2}{\partial\phi_1\theta_1} & \frac{\partial^2}{\partial\phi_1\theta_2} & \cdots & \frac{\partial^2}{\partial\phi_1\theta_n} \\ \frac{\partial^2}{\partial\phi_2\theta_1} & \frac{\partial^2}{\partial\phi_2\theta_2} & \cdots & \frac{\partial^2}{\partial\phi_2\theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial\phi_m\theta_1} & \frac{\partial^2}{\partial\phi_m\theta_2} & \cdots & \frac{\partial^2}{\partial\phi_m\theta_n} \end{pmatrix}.$$

In [30], the BFIM is shown to obey the recursion

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - \left(\mathbf{D}_k^{21}\right)^T \left(\mathbf{J}_k + \mathbf{D}_k^{11}\right)^{-1} \mathbf{D}_k^{12} \tag{8}$$

where

$$\mathbf{D}_k^{11} = E_x\left\{-\triangle_{x_k}^{x_k} \ln p(x_{k+1}|x_k)\right\} \tag{9}$$

$$\mathbf{D}_k^{12} = E_x\left\{-\triangle_{x_k}^{x_{k+1}} \ln p(x_{k+1}|x_k)\right\} \tag{10}$$

$$\mathbf{D}_k^{22} = E_x\left\{-\triangle_{x_{k+1}}^{x_{k+1}} \ln p(x_{k+1}|x_k)\right\} + E_x\left\{-\triangle_{x_{k+1}}^{x_{k+1}} \ln p(\mathbf{z}_{k+1}|x_{k+1})\right\}. \tag{11}$$

The recursion is initialized with the value

$$\mathbf{J}_0 = E_x\left\{-\triangle_{x_0}^{x_0} \ln p(x_0)\right\}. \tag{12}$$

In general, the expectations in (9)–(11) admit no closed-form analytical solution and must be approximated numerically. If the equations are nonlinear, then the expectation integrals can be evaluated via Monte Carlo integration. If either the process model or observation model is linear, then some of the terms in (9) and (10) become simply products of matrices. As a first step in applying the Monte Carlo integration, we need to define the following matrix functions:

$$\mathbf{\Lambda}^{11}(x_k, x_{k+1}) = -\triangle_{x_k}^{x_k} \ln p(x_{k+1}|x_k) \tag{13}$$

$$\mathbf{\Lambda}^{12}(x_k, x_{k+1}) = -\triangle_{x_k}^{x_{k+1}} \ln p(x_{k+1}|x_k) \tag{14}$$

$$\mathbf{\Lambda}^{22,a}(x_k, x_{k+1}) = -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(x_{k+1}|x_k) \tag{15}$$

$$\mathbf{\Lambda}^{22,b}(x_{k+1}, \mathbf{z}_{k+1}) = -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(\mathbf{z}_{k+1}|x_{k+1}). \tag{16}$$

We can rewrite the (9)–(11) as follows:

$$D_k^{11} = \int \Lambda^{11}(x_k, x_{k+1}) p(x_{k+1}|\mathbf{z}_{k+1}) dx_{k+1} \tag{17}$$

$$D_k^{12} = \int \Lambda^{12}(x_k, x_{k+1}) p(x_{k+1}|\mathbf{z}_{k+1}) dx_{k+1} \tag{18}$$

$$D_k^{22} = \int \left(\Lambda^{22,a}(x_k, x_{k+1}) p(x_{k+1}|\mathbf{z}_{k+1}) + \Lambda^{22,b}(x_{k+1}, \mathbf{z}_{k+1})\right) p(x_{k+1}|\mathbf{z}_{k+1}) dx_{k+1}. \tag{19}$$

Once we have a sample representation of the posterior density, these expectation integrals can be calculated through sample mean approximations. We can obtain the sample-based representation of the posterior pdf $p(x_{k+1}|\mathbf{z}_{k+1})$ by exploiting the work done in particle filtering [31]. The *a posteriori* samples at k denoted by $X_k^{(n)}$ with weight $w_k^{(n)}$ are passed through the process model. The output samples of the process model are then fed into the measurement model to assign new weights to the samples. The weighted samples are then used in the sequential importance resampling (SIR) step to produce the *a posteriori* samples at time $k+1$. Therefore, using the process model density and likelihood density, we can generate weighted samples on a stochastic grid to represent the posterior density and estimate the Fisher component matrices with the empirical averages

$$D_k^{11} \simeq \frac{1}{N} \sum_{n=1}^{N} \Lambda^{11}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) \tag{20}$$

$$D_k^{12} \simeq \frac{1}{N} \sum_{n=1}^{N} \Lambda^{12}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) \tag{21}$$

$$D_k^{22} \simeq \frac{1}{N} \sum_{n=1}^{N} \left(\Lambda^{22,a}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) + \Lambda^{22,b}\left(X_k^{(n)}, X_{k+1}^{(n)}\right)\right) \tag{22}$$

where $X_{k+1}^{(n)}$, $n = 1, \ldots, N$, are the *a posteriori* samples representing the density $p(x_{k+1}|\mathbf{z}_{k+1})$ and N stands for the number of samples. We will refer to the algorithm for computing PCRB via sequential Monte Carlo integration as PCRB-IGMKPF (see, e.g., [32] for additional details). The detailed derivation of PCRB-IGMKPF is described next.

PCRB-IGMKPF

1) Initialize the samples $X_0^{(n)}$, $n = 1, \ldots, N$, from $p(x_0)$, and then compute J_0 from (12).
2) Predict $X_{k+1}^{(n)}$ by sampling for $n = 1, \ldots, N$ from $p(x_{k+1}|x_k = x_k^{true})$.
3) From $\mathbf{n}_k^{true}$, the observation noise density is approximated by $p_g(\mathbf{n}_k) = \sum_{j=1}^{J} \gamma_k^{(J)} \mathrm{N}(\mathbf{n}_k; \mu_{\mathbf{n}_k}^{(j)}, \mathbf{R}_k^{(j)})$
   - The posterior density is approximated by
   $$p_g(x_{k-1}|\mathbf{z}_{k-1}) = \sum_{g=1}^{G} \alpha_{k-1}^{(g)} \mathrm{N}\left(x_{k-1}; \mu_{k-1}^{(g)}, P_{k-1}^{(g)}\right).$$
   - The process noise density is approximated by
   $$p_g(v_{k-1}) = \sum_{i=1}^{I} \beta_{k-1}^{(i)} \mathrm{N}\left(v_{k-1}; \mu_{v_{k-1}}^{(i)}, Q_{k-1}^{(i)}\right).$$
   - The observation noise density is approximated by
   $$p_g(\mathbf{n}_k) = \sum_{j=1}^{J} \gamma_k^{(J)} \mathrm{N}\left(\mathbf{n}_k; \boldsymbol{\mu}_{\mathbf{n}_k}{}^{(j)}, \mathbf{R}_k^{(j)}\right).$$
4) Prepredition step
   - Calculate the prepredictive state density $\tilde{p}_g(x_k|\mathbf{z}_{k-1})$ using KF.
   - Calculate the preposterior state density $\tilde{p}_g(x_k|\mathbf{z}_k)$ using KF.
5) Prediction step
   - Calculate the predictive state density $\hat{p}_g(x_k|\mathbf{z}_{k-1})$ using GMM.
   - Calculate the posterior state density $\hat{p}_g(x_k|\mathbf{z}_k)$ using GMM.
6) Observation update step
   - Draw N samples $\{\chi_k^{(l)}; l = 1, \cdots, N\}$ from the importance density function $q(x_k|\mathbf{z}_k) = \hat{p}_g(x_k|\mathbf{z}_k)$.
   - Calculate their corresponding importance weights
   $$\tilde{w}_k^{(l)} = \frac{p\left(\mathbf{z}_k|\chi_k^{(l)}\right)\hat{p}_g\left(\chi_k^{(l)}|\mathbf{z}_{k-1}\right)}{\hat{p}_g\left(\chi_k^{(l)}|\mathbf{z}_k\right)}$$
   - Normalize the weights $w_k^{(l)} = \frac{\tilde{w}_k^{(l)}}{\sum_{l=1}^{N}} \tilde{w}_k^{(l)}$.
   - Approximate the state posterior distribution $p_g(x_k|\mathbf{z}_k)$ using the EM-algorithm.
7) Infer the conditional mean and covariance:
   - $\bar{x}_k = \sum_{l=1}^{N} w_k^{(l)} \chi_k^{(l)}$ and $\bar{P}_k = \sum_{l=1}^{N} w_k^{(l)} (\chi_k^{(l)} - \bar{x}_k)(\chi_k^{(l)} - \bar{x}_k)^T$.
   - Or equivalently, upon fitting the posterior GMM, calculate the variables in (40).

A. PCRB for the Gaussian Network Delay Model

The relationship (2) determines the conditional pdf $p(x_{k+1}|x_k)$

$$p(x_{k+1}|x_k) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{1}{2\sigma_v^2}\left[(x_{k+1}-x_k)^2\right]} \tag{23}$$

and the (4) decides the conditional pdf:

$$p(\mathbf{z}_{k+1}|x_{k+1}) = \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{1}{2\sigma_n^2}\left[(\mathbf{z}_{k+1}-\mathbf{A}d-\mathbf{B}x_{k+1})^2\right]}. \tag{24}$$

Accordingly, we can obtain the following equations:

$$\boldsymbol{\Lambda}_k^{11}(x_k, x_{k+1}) = -\triangle_{x_k}^{x_k} \ln p(x_{k+1}|x_k) = \frac{1}{\sigma_v^2} \tag{25}$$

$$\boldsymbol{\Lambda}_k^{12}(x_k, x_{k+1}) = -\triangle_{x_k}^{x_{k+1}} \ln p(x_{k+1}|x_k) = -\frac{1}{\sigma_v^2} \tag{26}$$

$$\boldsymbol{\Lambda}_k^{22,a}(x_k, x_{k+1}) = -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(x_{k+1}|x_k) = \frac{1}{\sigma_v^2} \tag{27}$$

$$\begin{aligned}\boldsymbol{\Lambda}_{k+1}^{22,b}(x_{k+1}, \mathbf{z}_{k+1}) &= -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(\mathbf{z}_{k+1}|x_{k+1}) \\ &= \frac{\mathbf{B}^T\mathbf{B}}{\sigma_n^2} = \frac{2}{\sigma_n^2}.\end{aligned} \tag{28}$$

Therefore, we can estimate the Fisher component matrices with the empirical averages

$$\mathbf{D}_k^{11} \simeq \frac{1}{N}\sum_{n=1}^{N} \Lambda^{11}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) = \frac{1}{\sigma_v^2} \tag{29}$$

$$\mathbf{D}_k^{12} \simeq \frac{1}{N}\sum_{n=1}^{N} \Lambda^{12}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) = -\frac{1}{\sigma_v^2} \tag{30}$$

$$\begin{aligned}\mathbf{D}_k^{22} &\simeq \frac{1}{N}\sum_{n=1}^{N}\left(\Lambda^{22,a}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) + \Lambda^{22,b}\left(X_k^{(n)}, X_{k+1}^{(n)}\right)\right) \\ &= \frac{1}{\sigma_v^2} + \frac{2}{\sigma_n^2}.\end{aligned} \tag{31}$$

From (8), it turns out that the evaluation of BFIM for Gaussian case can be done via the recursion

$$\mathbf{J}_{k+1} = \frac{1}{\sigma_v^2} + \frac{2}{\sigma_n^2} - \left(-\frac{1}{\sigma_v^2}\right)^T \left(\mathbf{J}_k + \frac{1}{\sigma_v^2}\right)^{-1} \left(-\frac{1}{\sigma_v^2}\right). \tag{32}$$

From (6), PCRB is the inverse of BFIM ($\mathbf{J}$). From the above equations, we note that PCRB is a function of σ_v^2 and σ_n^2. In [12], the CRLB is expressed as $\mathrm{var}(\hat{x}_{\mathrm{ML}}) \geq \frac{\sigma_n^2}{2N}$. Thus, the lower bound of MLE is limited by σ_n^2. In other words, to improve the MSE performance, the number of observations has to be increased. However, WSNs are power limited systems, and a large number of message exchanges is not desirable. Therefore, alternative algorithms with improved MSE-performance are desirable.

B. PCRB for Exponential and Gamma Network Delay Models

Equation (2) determines the conditional pdf $p(x_{k+1}|x_k)$

$$p(x_{k+1}|x_k) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{1}{2\sigma_v^2}\left[(x_{k+1}-x_k)^2\right]} \tag{33}$$

while (4) decides the conditional pdf

$$p(\mathbf{z}_{k+1}|x_{k+1}) = \lambda e^{-\lambda[(\mathbf{z}_{k+1}-\mathbf{A}d-\mathbf{B}x_{k+1})]}. \tag{34}$$

Equation (34) cannot be used to evaluate the entries of Fisher matrix because of the second-order derivatives. Using the fact

that a gamma-distributed random variable $z \sim \Gamma(\alpha = 1, \beta = \lambda)$ assumes an exponential distribution with rate parameter λ, we will exploit this alternative route to evaluate the Fisher matrix entries. Equation (4) leads to the conditional pdf

$$p(\mathbf{z}_{k+1}|x_{k+1}) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \left[(\mathbf{z}_{k+1} - \mathbf{A}d - \mathbf{B}x_{k+1})\right]^{\alpha-1} \times e^{-\beta[(\mathbf{z}_{k+1} - \mathbf{A}d - \mathbf{B}x_{k+1})]} \quad (35)$$

where if α is a positive integer, then $\Gamma(\alpha) = (\alpha-1)!$. From the above equation, it follows that

$$\begin{aligned} &\mathbf{\Lambda}_{k+1}^{22,b}(x_{k+1}, \mathbf{z}_{k+1}) \\ &\quad = -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(\mathbf{z}_{k+1}|x_{k+1}) \\ &\quad = \frac{(\alpha-1)\mathbf{B}^T\mathbf{B}}{\left[(\mathbf{z}_{k+1} - \mathbf{A}d - \mathbf{B}x_{k+1})\right]^T \left[(\mathbf{z}_{k+1} - \mathbf{A}d - \mathbf{B}x_{k+1})\right]}. \end{aligned}$$

Therefore, we estimate the Fisher matrix entries with the empirical averages

$$\mathbf{D}_k^{11} \simeq \frac{1}{N}\sum_{n=1}^{N} \Lambda^{11}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) = \frac{1}{\sigma_v^2}$$

$$\mathbf{D}_k^{12} \simeq \frac{1}{N}\sum_{n=1}^{N} \Lambda^{12}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) = -\frac{1}{\sigma_v^2}$$

$$\begin{aligned} \mathbf{D}_k^{22} &\simeq \frac{1}{N}\sum_{n=1}^{N} \left(\Lambda^{22,a}\left(X_k^{(n)}, X_{k+1}^{(n)}\right) + \Lambda^{22,b}\left(X_k^{(n)}, X_{k+1}^{(n)}\right)\right) \\ &= \frac{1}{\sigma_v^2} + \frac{1}{N}\sum_{n=1}^{N} \Lambda^{22,b}\left(X_{k+1}^{(n)}, \mathbf{z}_{k+1}\right). \end{aligned}$$

C. Comparison Between PCRB and CRLB

Figs. 1 and 2 show CRLB and PCRB when the random delay model is Gaussian with zero mean and variance $\sigma_n^2 = 1$ for various initializations of the Fisher information matrix J_0 and different power levels for the process noise (σ_v^2). Figs. 1 and 2 show that for small power levels of the process noise $\sigma_v^2 = 10^{-4}, 10^{-5}, 10^{-6}$, depending on the initialization J_0, PCRB might achieve similar or smaller values than CRLB [12] depending on how J_0 is selected. Fig. 3 shows that for negligible process noise $(\sigma_v^2 = 1e-6)$ relative to observation noise $(\sigma_n^2 = 1, 0.7, 0.5)$, PCRB exhibits lower values than CRLB. These plots illustrate the fact that PCRB depends on the initialization (J_0), observation noise power (σ_n^2), and process noise power (σ_v^2). It is natural that process modeling errors exist and their effect on PCRB becomes noticeable at certain power levels (σ_v^2). As the process noise variance decreases to zero, the modeling error is removed and the state-space model becomes identical with the one adopted by the ML approach. The simulation results corroborate the fact that CRLB exhibits almost the same performance levels as PCRB when the process noise variance is sufficiently small (less than 10^{-4}) and the Fisher Information matrix used for initialization is $J_0 = 1$. The simulation results also illustrated an interesting fact, when the process noise variance is sufficiently small (less than 10^{-4}) and the initial value of Fisher information matrix is $J_0 = \frac{1}{(\sigma_v^2)}$, PCRLB is significantly lower than CRLB.

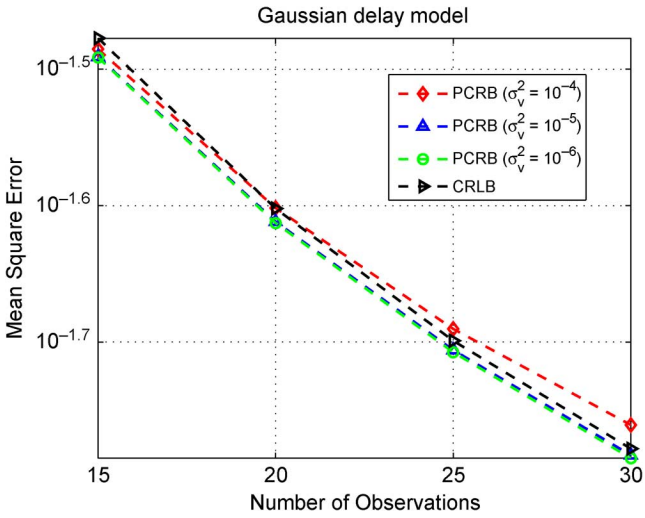


Fig. 1. PCRB and CRLB for symmetric Gaussian random delays ($\sigma_n^2 = 1$ and $J_0 = 1$).

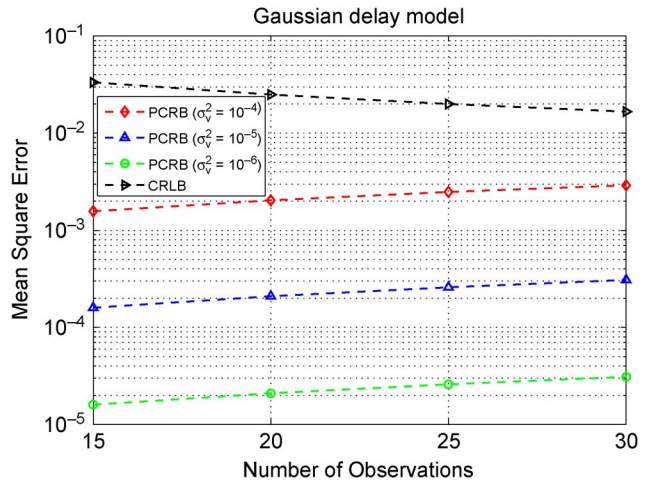


Fig. 2. PCRB and CRLB for symmetric Gaussian random delays ($\sigma_n^2 = 1$ and $J_0 = \frac{1}{\sigma_v^2}$).

Notice also that even in the presence of a reduced number of observations, IGMKPF make use of a sufficient number of samples (particles) for generating the pdf and statistics of observation noise. Because the state representing the phase offset can be estimated using the MLE $(\hat{x}_{\mathrm{ML}})$, the density $p(\hat{x}_{\mathrm{ML}})$ can be also estimated using sequential Monte Carlo simulation. Therefore, given the prior information $(\hat{x}_{\mathrm{ML}}, p(\hat{x}_{\mathrm{ML}}))$ and observations (z_k), the observation noise's statistics can be estimated. Since the estimated clock offset value yielded by ML is close to the real value and the measurement model is linear, the estimated observation noise's statistics are close to the real statistics. Therefore, IGMKPF is expected to be a robust estimator and exhibit improved performance relative to MLE in the presence of unknown network delay distributions and reduced number of observations. Therefore, PCRB is expected to represent a tighter bound than CRLB in the presence of reduced number of observations and unknown observation noise distributions.

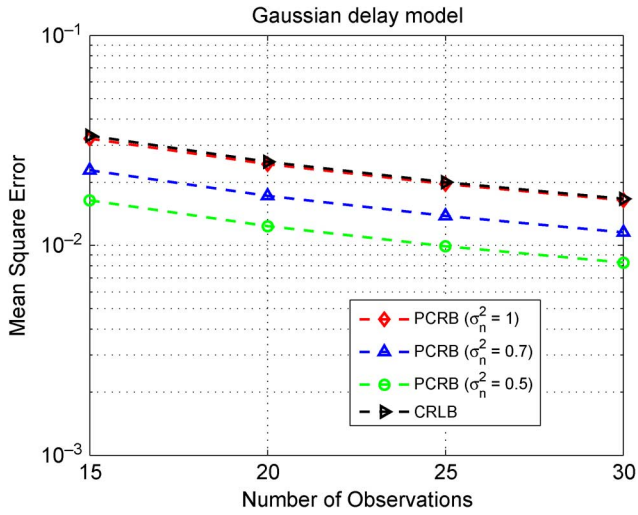


Fig. 3. PCRB and CRLB for symmetric Gaussian random delays ($\sigma_v^2 = 10^{-6}$ and $J_0 = 1$).

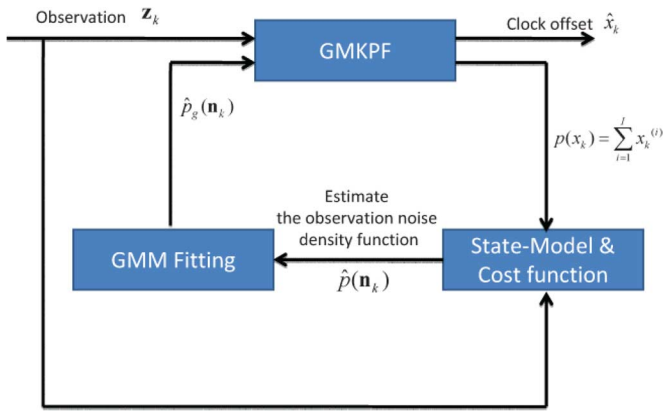


Fig. 4. Block diagram representation of the IGMKPF estimator.

Computer simulations also illustrate the fact that PCRB might increase as the number of observations increases. In the state-space iterative estimation model, prediction errors and random sampling errors [31] might occur due to the inefficient proposal distribution, which might cause a constant error accumulation. Therefore, in the state tracking process, these errors might accumulate from one iteration to the next iteration. These accumulative errors increase as the number of observations increase. When the performance improvement due to the increase in the number of observations is larger than the performance degradation due to the cumulative errors, the MSE decreases as the number of observations increases, and in the limit the cumulative errors are neglected. On the other hand, when the performance improvement due to the increase in the number of observations is less than the performance degradation induced by the cumulative errors, the MSE increases as the number of observations increases. The cumulative errors play a more important role in the performance as the number of observations increases. For a large number of observations, tracking of the error accumulation is important in the performance of the Bayesian iterative estimation approach for phase offset that we will propose later. In order to reduce the accumulative error, a more accurate proposal distribution is needed. Since it is impossible to obtain an efficient proposal distribution, there will always exist random sampling errors.

Notice that particle filters (PFs) track the unknown state using knowledge of the observation and process prior densities. However, the particle filter is not optimal due to the random sampling errors, inefficient proposal distribution (particle filter samples from the proposal distribution) etc. Additionally, note that the convergence of particle filter has not been proved to be uniform in time t [32]. For a given fixed t, there is convergence as $N \to \infty$ but nothing is said about that limit at $t \to \infty$. In practice, this could mean that at a later time t, a large value of N may be required, though that value could depend also on other factors such as the nature of the state-space model. GMKPF gives better performance than general PF by using improved prediction and proposal distributions. In case that the observation noise density is perfectly known and process noise variance is lower than 10^{-4}, GMKPF has the same performance as MLE. However, in the case that the observation noise density is unknown and can be estimated accurately, and the process noise variance is negligible relative to the observation noise power, we will show in the next section that IGMKPF yields better performance than MLE and GMKPF.

IV. IGMKPF

The proposed IGMKPF estimator combines the GMKPF with the observation noise density estimator. The observation noise density estimator consists of the state model and a cost function in the form of an innovation equation expressed as the difference between the observation and estimated state posterior pdfs: $p(\mathbf{z})-p(\hat{\mathbf{z}})$. The innovation equation is produced by considering the estimate yielded by a standard Kalman filter, as well as a GMKPF estimator. In order to analyze the MSE-performance of the proposed IGMKPF technique, computer simulations will be conducted, and the PCRB and the KLD between the true and estimated network delay noise will be evaluated as well.

Fig. 4 provides a perspective on the proposed IGMKPF estimator. In IGMKPF, the first processing stage is represented by the GMKPF which is used to estimate the state posterior density using the observation density, prior density, and process density. The second processing stage consists in estimating the observation noise density using the innovation attained, by considering the estimate of the standard Kalman filter. The estimated observation noise density, which is used as an input to the first processing stage, is approximated by a GMM fitting function. The iterative process between the two processing stages is repeated until all the observations are processed.

Because GMKPF represents the main component of the proposed iterative estimator, we will next review its key features [22]. Particle filtering is a sequential Monte Carlo sampling scheme established within the Bayesian paradigm. In terms of a Bayesian perspective, the main entity of interest is the posterior distribution $p(x_k|\mathbf{z}_{0:k})$ at time k. However, it is impossible to find the analytical expression of $p(x_k|\mathbf{z}_{0:k})$ in closed form except for some special cases such as Gaussian or exponential pdfs because of the non-Gaussianity of the model (4). Alternatively, particle filtering might be used to approximate $p(x_k|\mathbf{z}_{0:k})$

by stochastic samples generated using a sequential importance sampling technique. Since the particle filtering with the prior importance function does not utilize any information from observations in proposing new samples, its use and filtering performance may be ineffective or poor. Herein, the GMKPF is implemented as an adaptation of the Gaussian mixture sigma point particle filter (GMSPPF) [6], which comes out from the utilization of another filtering technique producing a filtering probability density function used as importance function (IF) for the particle filtering.

GMSPPF uses hybrid sequential Monte Carlo simulation and a Gaussian sum filter in order to efficiently estimate posterior distributions of unknown states in nonlinear dynamic systems. However, since our state space model is linear, GMKPF is obtained from GMSPPF by substituting a KF for the sigma point Kalman filter (SPKF). Accordingly, the predicted and updated Gaussian components in the GMMs, i.e., the means and covariances of the involved probability densities (posterior, importance, and so on) are calculated using the KF instead of the SPKF [6], [34]. In order to properly cope with the particle depletion problem in case that the shape of the observation (measurement) likelihood is very peaked, GMKPF expresses the posterior density by a GMM obtained from the resampled equally weighted particle set using the EM algorithm.

As aforementioned, GMKPF combines the measurement update step based on the importance sampling (IS) with a KF-based Gaussian sum filter for the time update and proposal density generation step, and approximates the prior, proposal and posterior density functions as GMMs using banks of parallel KFs in the time update stage. The updated mean and covariance of each mixand or component are calculated from the KF updates. From the Gaussian sum filtering (GSF) approach in [7], the output of a bank of (G' and G'') parallel KFs are used to calculate GMM approximations of $p(x_k|\mathbf{z}_{k-1})$ and $p(x_k|\mathbf{z}_k)$. The predictive state density is now approximated by the GMM

$$p_g(x_k|\mathbf{z}_{k-1}) = \sum_{g'=1}^{G'} \alpha_k^{(g')} N\left(x_k; \tilde{\mu}_k^{(g')}, \tilde{P}_k^{(g')}\right) \tag{36}$$

and the posterior state density is approximated by the GMM

$$p_g(x_k|\mathbf{z}_k) = \sum_{g''=1}^{G''} \alpha_k^{(g'')} N\left(x_k; \mu_k^{(g'')}, P_k^{(g'')}\right). \tag{37}$$

This posterior state density will only be used as the proposal distribution in the IS-based measurement update step.

In the measurement update stage, we draw N samples (particles) $\{\chi_k^{(n)}; n = 1, \ldots, N\}$ from the proposal distribution $p_g(x_k|\mathbf{z}_k)$ (37) and calculate their corresponding importance weights. Using a weighted EM algorithm to fit a G-component GMM, GMKPF represents the posterior filtering density as follows:

$$p_g(x_k|\mathbf{z}_k) = \sum_{g=1}^{G} \alpha_k^{(g)} N\left(x_k; \mu_k^{(g)}, P_k^{(g)}\right) \tag{38}$$

where G, $\alpha_l^{(g)}$, and $N(x_k; \mu_k^{(g)}, P_k^{(g)})$ denote the number of GMMs, the mixing weights, and a normal distribution obtained from the gth KF with predicted mean $\mu_k^{(g)} = \bar{x}_k$ and positive definite covariance $P_k^{(g)}$, respectively. As a result, Gaussian Mixture approximations can be obtained from these particles and weights. Using this mechanism, the EM-based posterior GMM further mitigates the "sample depletion" problem through its inherent "kernel smoothing" nature [22].

The EM algorithm presents an iterative method to estimate $\bar{\theta}$ through

$$\bar{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \quad p(\mathbf{z}|\boldsymbol{\theta}) \tag{39}$$

with the Gaussian mixture specified by the parameter set $\boldsymbol{\theta} = \{\alpha_l^{(1)}, \cdots, \alpha_l^{(G)}, \mu_l^{(1)}, \cdots, \mu_l^{(G)}, P_l^{(1)}, \cdots, P_l^{(G)}\}$. Specifically, the EM algorithm is a two-step iterative algorithm which operates as follows. Given $\boldsymbol{\theta}^{(j)}$, it finds the next value $\boldsymbol{\theta}^{(j+1)}$ via

- E-step : $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)}) = E[\log p(\mathbf{z}|\boldsymbol{\theta})|\boldsymbol{\theta}^{(j)}]$.
- M-step : $\boldsymbol{\theta}^{(j+1)} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(j)})$.

The [35] gives more detailed explanations of the EM algorithm for modeling GMMs. Finally, the conditional mean state estimate and the corresponding error covariance can be expressed as

$$\begin{aligned} \bar{x}_k &= \sum_{g=1}^{G} \alpha_k^{(g)} \mu_l^{(g)} \\ \bar{P}_k &= \sum_{g=1}^{G} \alpha_k^{(g)} \left[P_k^{(g)} + \left(\mu_k^{(g)} - \bar{x}_k\right)\left(\mu_k^{(g)} - \bar{x}_k\right)^T\right]. \end{aligned} \tag{40}$$

The pseudocode of the GMKPF algorithm is described next.

GMKPF Algorithm

1) At time $k-1$, initialize the densities:
   - The posterior density is approximated by
   $$p_g(x_{k-1}|\mathbf{z}_{k-1}) = \sum_{g=1}^{G} \alpha_{k-1}^{(g)} \mathrm{N}\left(x_{k-1}; \mu_{k-1}^{(g)}, P_{k-1}^{(g)}\right).$$
   - The process noise density is approximated by
   $$p_g(v_{k-1}) = \sum_{i=1}^{I} \beta_{k-1}^{(i)} \mathrm{N}\left(v_{k-1}; \mu_{v_{k-1}}^{(i)}, Q_{k-1}^{(i)}\right).$$
   - The observation noise density is approximated by
   $$p_g(\mathbf{n}_k) = \sum_{j=1}^{J} \gamma_k^{(J)} \mathrm{N}\left(\mathbf{n}_k; \boldsymbol{\mu}_{\mathbf{n}_k}^{(j)}, \mathbf{R}_k^{(j)}\right).$$
2) Prepredition step:
   - Calculate the prepredictive state density $\tilde{p}_g(x_k|\mathbf{z}_{k-1})$ using KF.
   - Calculate the preposterior state density $\tilde{p}_g(x_k|\mathbf{z}_k)$ using KF.
3) Prediction step:

- Calculate the predictive state density $\hat{p}_g(x_k|\mathbf{z}_{k-1})$ using GMM.
- Calculate the posterior state density $\hat{p}_g(x_k|\mathbf{z}_k)$ using GMM.

4) Observation Update step
   - Draw N samples $\{\chi_k^{(l)}; l = 1, \cdots, N\}$ from the importance density function $q(x_k|\mathbf{z}_k) = \hat{p}_g(x_k|\mathbf{z}_k)$.
   - Calculate their corresponding importance weights

$$\tilde{w}_k^{(l)} = \frac{p\left(\mathbf{z}_k|\chi_k^{(l)}\right)\hat{p}_g\left(\chi_k^{(l)}|\mathbf{z}_{k-1}\right)}{\hat{p}_g\left(\chi_k^{(l)}|\mathbf{z}_k\right)}.$$

   - Normalize the weights $w_k^{(l)} = \frac{\tilde{w}_k^{(l)}}{\sum_{l=1}^{N}\tilde{w}_k^{(l)}}$.
   - Approximate the state posterior distribution $p_g(x_k|\mathbf{z}_k)$ using the EM-algorithm.

5) Infer the conditional mean and covariance
   - $\bar{x}_k = \sum_{l=1}^{N} w_k^{(l)}\chi_k^{(l)}$ and $\bar{P}_k = \sum_{l=1}^{N} w_k^{(l)}(\chi_k^{(l)} - \bar{x}_k)(\chi_k^{(l)} - \bar{x}_k)^T$.
   - Or equivalently, upon fitting the posterior GMM, calculate the variables in (40).

The improved performance of GMKPF comes with the price of an increased computational complexity. Evaluating exactly the number of floating point operations (flops) involved in GMKPF is not possible because it is an iterative and quite complex algorithm. However, we will evaluate its computational complexity by quantifying only the most computationally demanding steps in terms of the Big O notation. Let $L(=1)$ denote the dimension of the state vector (x), and represent by N the number of particles (samples in pseudocode) and through G the number of GMM components. The state vector and matrices are $x \in R^{1\times 1}$, $F \in R^{1\times 1}$, $P \in R^{1\times 1}$, where P is the *a posteriori* error covariance matrix. The Kalman Filter requires approximately $O(L^3)$ flops since the matrix times matrix multiplication is the most time consuming step $(P_{k+1|k} = FP_k + Q)$, whereas the particle filter assumes approximately $O(NL^2)$ flops due to the matrix times vector multiplication $(\bar{x}_{k+1|k}^{(i)} = F\bar{x}_k^{(i)}, i = 1, 2, \cdots, N)$ and sampling step. The GMKPF necessitates approximately $O(GL^3)$ flops due to the KF-step and $O(GL^2N)$ flops due to the EM-step (considering only the most computationally demanding steps and neglecting the rest of operations). If $N \gg L$, GMKPF requires approximately $O(GL^2N)$ flops. Notice also that MLEg necessitates approximately $O(L)$ flops because it involves summations and a division. This indicates that GMKPF is approximately 300 times slower than MLEg in an application with $L = 1$, $N = 100$, and $G = 3$.

In IGMKPF, at the first iteration, GMKPF initializes the densities and estimates the state posterior pdf using particles for generation of pdfs (posterior pdf, prior pdf, process noise pdf, observation noise pdf, and so on). Next, the observation noise density generator block estimates the observation noise pdf using both the measurement observations and the posterior density computed by GMKPF. This iteration is repeated up to the number of observations. The pseudocode of IGMKPF algorithm is next described.

IGMKPF Algorithm

1) At time k, initialize the densities and set the initial state $\hat{x}_{k-1} = \hat{x}_{\mathrm{ML}}$.
2) GMKPF step (estimate the state posterior density).
   - Calculate the state posterior density $p_g(x_k|\mathbf{z}_k)$ using GMKPF.
   - If k reaches the end of observations, go to "Infer the conditional mean and covariance step."
3) Estimate the Observation Noise Density (OND) step
   - Calculate the observation noise density $p(\hat{n})$ given $\mathbf{z}_k$ and $p_g(x_k|\mathbf{z}_k)$, and state model [(2) and (4)].
   - The observation noise density using GMM is approximated by

$$p_g(\mathbf{n}_k) = \sum_{j=1}^{J}\gamma_k^{(J)}\mathrm{N}\left(\mathbf{n}_k; \boldsymbol{\mu}_{\mathbf{n}_k}^{(j)}, \mathbf{R}_k^{(j)}\right).$$

4) $k = k + 1$, go to the GMKPF step.
5) Infer the conditional mean and covariance
   - $\bar{x}_k = \sum_{l=1}^{N} w_k^{(l)}\chi_k^{(l)}$ and $\bar{P}_k = \sum_{l=1}^{N} w_k^{(l)}(\chi_k^{(l)} - \bar{x}_k)(\chi_k^{(l)} - \bar{x}_k)^T$.
   - Or equivalently, upon fitting the posterior GMM, calculate the variables in (40).

To validate the accuracy of the observation noise density estimation, we use the KLD [28] as a measure of fitting the estimated noise density to the true density. Fig. 5 shows the relationship between KLD and MSE-performance of IGMKPF. In Fig. 5, the upper graph refers to KLD, while the lower graph depicts the MSE performance of IGMKPF. Upon 10 Monte Carlo simulation runs, KLD and MSE exhibit a quite similar behavior. These plots illustrate the fact that the more accurately the observation noise density is estimated, the better the MSE performance of IGMKPF is.

V. Simulation Results

In this section, computer simulations will be conducted to assess the performance of IGMKPF, PCRB-IGMKPF, MLEg [12], MLEe [12], and CRLB for estimating the clock offset in WSNs that are subject to three types of network delays: symmetric Gaussian, exponential, and Gamma. The process noise assumes the power $\sigma_v^2 = 10^{-6}$. The number of particles and GMMs are 500 and 3, respectively. One aspect about using IGMKPF is that it requires proper initialization. Depending on the problem, the initial guess may need to be close to the correct value to achieve fast convergence. The ML estimators proposed in [12] for symmetric Gaussian and exponential random delays are good examples of initializations. The initial values $\hat{x}_0 = \hat{x}_{MLEg}$, and $\hat{x}_0 = \hat{x}_{\mathrm{MLEe}}$ are near the true values in Gaussian and non-Gaussian (exponential and gamma) delay distributions, respectively. The convergence of IGMKPF is achieved after a number of iterations on the order of the number of measurements. Since GMKPF does not initialize the observation noise density and does not track it, its performance might be limited. However, IGMKPF tracks the observation noise density using

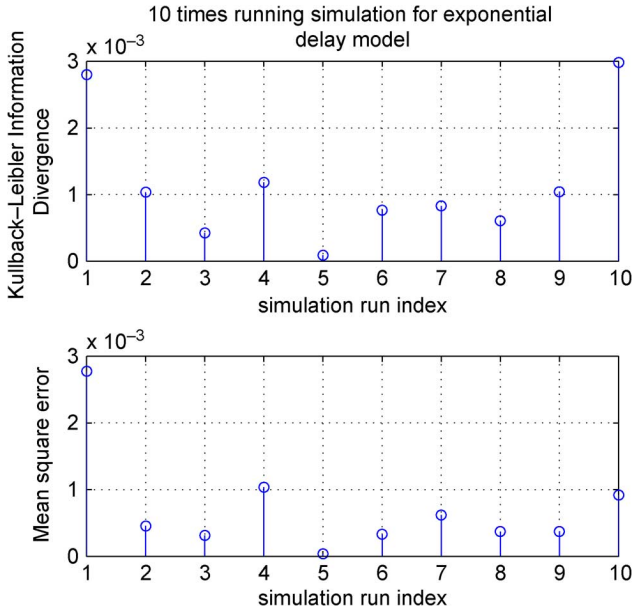


Fig. 5. KLD and MSE for symmetric exponential random delays ($\lambda = 1$).

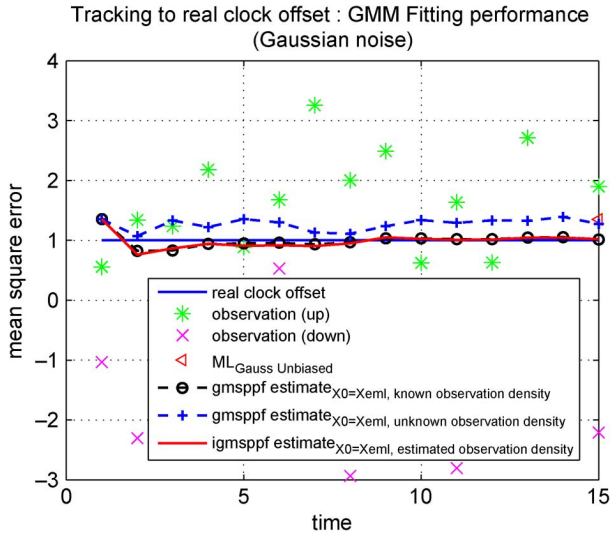


Fig. 6. MSEs of clock offset estimators for Gaussian ($\sigma_n^2 = 1$).

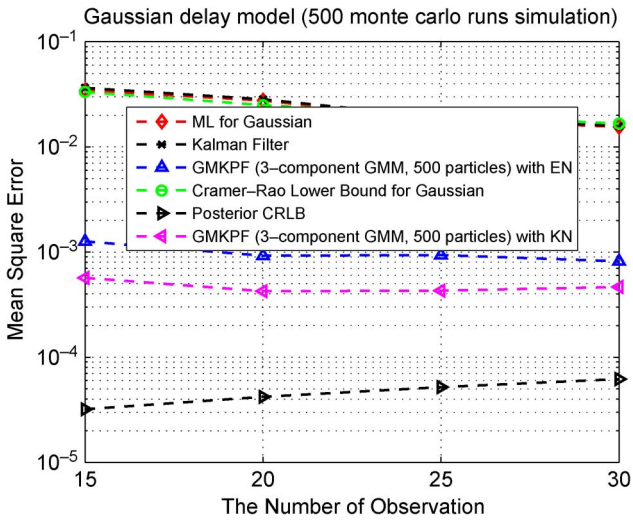


Fig. 7. MSEs of clock offset estimators for Gaussian ($\sigma_n^2 = 1$).

the observation and estimated posterior pdfs. Therefore, the performance of IGMKPF is expected to be better than GMKPF and MLE.

Fig. 6 shows the MSE performances of MLEe, MLEg, GMKPF, and IGMKPF. It shows the convergence of GMKPF and IGMKPF in accordance with the accuracy of fitting the observation noise density. In case of unknown observation density, GMKPF exhibits similar performance as ML. However, in the case of known observation density, GMKPF converges to the real clock offset. Notice also that IGMKPF converges to the real clock offset even in the presence of unknown observation noise distribution. This indicates that if IGMKPF can accurately estimate the observation density, its MSE performance is much better than that of MLE. Left-triangles and right-triangles denote MLEe and MLEg, respectively. The dashed line indicates GMKPF with the initial condition $x_0 = X_{\mathrm{MLEe}}$ and known observation noise density. The dashed-with-circles line represents GMKPF with initial condition $x_0 = X_{\mathrm{true}} + 1$ and unknown observation noise density. Dashed-with-addition-sign line denotes GMKPF with initial condition $x_0 = X_{\mathrm{MLEe}}$ and unknown observation density. Solid line denotes IGMKPF with initial condition $x_0 = X_{\mathrm{MLEe}}$ and estimated observation density. In the symmetric Gaussian model, GMKPF shows the same performance as MLE because it does not exploit the observation noise density and the fixed parameter is $J_0 = 1$. When the observation noise density is estimated, IGMKPF shows better performance than MLE and GMKPF. If the observation noise density is not tracked, a bias will be induced in the GMKPF estimate and it exhibits performance close to MLEg. However, tracking the observation noise density helps to reduce the bias, which will help IGMKPF estimate to be closer to the real clock offset.

Figs. 7–9 show the MSE of the estimators under the assumption that the random delay models are symmetric Gaussian, exponential, and Gamma, respectively. Notations KN and EN denote the setups with known observation noise density and estimated observation noise density, respectively. The MSEs are plotted against the number of observations ranging from 15 to 30. To use the GMM fitting function in Matlab (such as kmean), the number of observations starts at 15. Fig. 7 is for the symmetric Gaussian delay model. Assuming the state-space model depicted in Section II is linear-Gaussian, a closed-form solution for Kalman filter can be considered. Theoretically, the Kalman filter exhibits the same performance as MLE for the linear Gaussian delay model. As indicated in Fig. 7, the Kalman filter presents almost same performance as MLE. Note that IGMKPF ($G = 3$) performs much better (over 100% reduction in MSE) relative to MLEg in the presence of a Gaussian delay model. It is remarkable that the performance of MLE is proportional to the number of observations, whereas that of IGMKPF is proportional to the number of particles, the number of GMMs, and the accuracy of noise density estimation, but it might not strictly depend on the number of observations. The initial conditions (initial state value ($x_0 = \hat{x}_{\mathrm{ML}}$), initial prior density [$p(x_0) = p(\hat{x}_{\mathrm{ML}})$], measurement noise density, etc.)

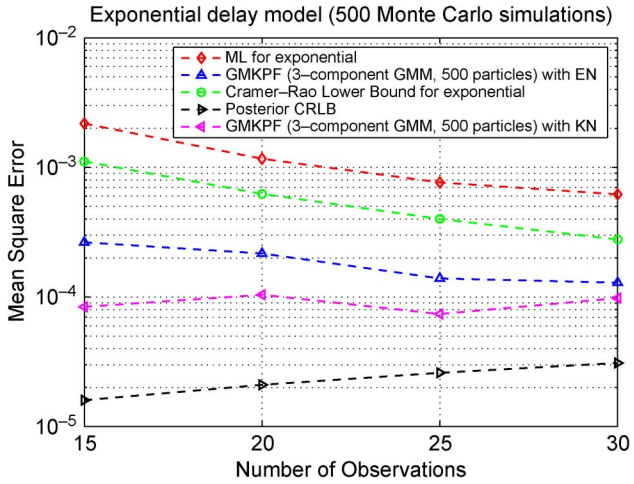


Fig. 8. MSEs of clock offset estimators for exponential ($\lambda = 1$).

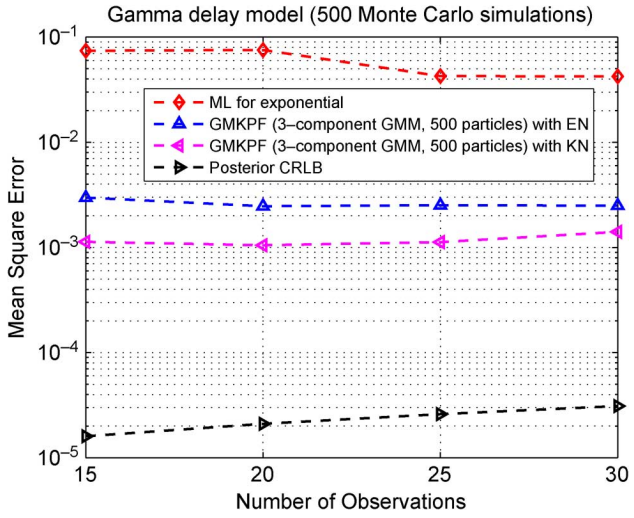


Fig. 9. MSEs of clock offset estimators for gamma ($\alpha_1 = 2$, $\beta_1 = 2$).

play a major role in the performance of IGMKPF. Since we can get the initial conditions from MLE, which ensure the convergence of IGMKPF even in the presence of reduced number of observations and unknown network delay distributions, the performance of IGMKPF is expected to be much better than that of MLE. This is a desirable feature for WSNs in order to keep the number of timing exchanges low so that energy is conserved.

IGMKPF uses a 3-component GMM ($G = 3$) to predict the state posterior, a 1-component GMM to capture the process noise density, and a 3-component GMM for estimating the measurement noise density. The GMKPF-estimator predicts the posterior probability function (likelihood function) adaptively using the EM-algorithm, and thus its performance depends on the ability of GMM to capture distributions such as process noise, posterior pdfs, and so on. However, GMKPF is neglecting the observation noise density in estimating the posterior pdf because of lack of any adaptation mechanism to predict the observation noise density. In order to overcome this challenge, IGMKPF, which is configured in terms of two estimators: GMKPF and observation noise density estimator, is iterated until the required accuracy of observation noise density estimation is obtained. Therefore, IGMKPF yields more accurate estimates of the state as illustrated in Figs. 7 and 8, for exponential ($\lambda = 1$) and gamma ($\alpha = 2$ and $\beta = 2$) network delay models, respectively. In Fig. 9, IGMKPF exhibits also much better performance than MLE in the presence of reduced number of observations. When IGMKPF is properly initialized, its performance goes close to PCRB. Notice also that PCRB is much lower than CRLB.

Next we compare IGMKPF with other well-known clock estimation schemes MLEg and MLEe in TPSN [17], RBS [11], and FTSP [36], with respect to the number of required timing messages (which practically indicates the amount of energy consumption) to achieve a specific MSE-performance. Let $N_{\mathrm{TPSN-MLEg}}$, $N_{\mathrm{RBS-MLEg}}$, and $N_{\mathrm{FTSP-MLEg}}$ denote the numbers of required timing messages for synchronization in TPSN, RBS, and FTSP, respectively, assuming the MLE scheme and a Gaussian delay model. In TPSN [17], denoting by J the overall number of sensor nodes, and assuming that $2K$ timing messages are required in every pairwise synchronization, $N_{\mathrm{TPSN-MLEg}}$ is equal to the number of pairwise synchronization times the number of required timing messages per pairwise synchronization. Thus, $N_{\mathrm{TPSN-MLEg}} = 2K(J-1)$. In RBS [11], $N_{\mathrm{RBS-MLEg}} = \frac{K+J(J-1)}{2}$, since the number of unique pairs in the network is $\frac{J(J-1)}{2}$. In FTSP [36], the number of required timing messages is $N_{\mathrm{FTSP-MLEg}} = JK$. When $\mathrm{MSE} = 10^{-3}$ and the number of nodes is 100, $N_{\mathrm{TPSN-MLEg}} = 2K(100-1) = 2 \times 500 \times 99 = 99000$ for MLEg [12], $N_{\mathrm{TPSN-IGMKPF}} = 2K(100-1) = 2 \times 15 \times 99 = 2970$ for IGMKPF according to Fig. 7. Consequently, the benefit of IGMKPF over MLEg is huge in terms of energy consumption in the network. Similar statements hold for $N_{\mathrm{RBS-MLEg}}$ and $N_{\mathrm{FTSP-MLEg}}$.

VI. Conclusion

This paper provided a novel Bayesian sequential Monte Carlo method, IGMKPF, for estimating the clock offset in WSNs. The benefits of the proposed synchronization method are improved performance compared to MLE, and applicability to arbitrary random delay models such as symmetric Gaussian, symmetric exponential, and Gamma delay models. In general, in case of (unknown) non-Gaussian distributions, analytical closed-form expressions for MSE-performance do not necessarily exist and it is also hard to derive lower bounds. The paper derived the PCRB and IGMKPF, an estimator that is robust to the lack of knowledge concerning the network delay distribution and to the presence of reduced number of observations. An important element in improving the estimator performance is the prediction of unknown observation noise density which led to an improved IGMKPF estimator. The paper presented a robust estimator based on IGMKPF which is capable of estimating the clock offset in arbitrary delay models and in the presence of time-varying clock offsets, and that can be used in a number of WSN applications with tight synchronization requirements [18], [19]. Since IGMKPF estimates iteratively the unknown delay probability density function, IGMKPF exhibits better performance than MLE or standard particle filtering algorithms. Most particle filters are not realistic due

to many assumptions (including *a-priori* knowledge of observation noise distribution). PCRB represents a lower bound for the MSE performance of IGMKPF. Because of the difficulty in evaluating PCRB in closed-form expression, the plotted PCRB graphs represent simulated bounds (obtained via IGMKPF) of the actual PCRB. In the presence of optimal settings (efficient proposal, optimal number of particles, etc.), the simulated PCRB might represent a tighter bound. The performance gap between IGMKPF and PCRB is due to the impossibility of estimating precisely the observation noise density with a reduced number of observations. There are also the sampling errors, the error accumulation due to the iterative nature of algorithm and inefficient proposal distribution that contribute to this. The proposed clock synchronization algorithm considers only the correction of clock phase-offset errors. Extension of this work to the situation when clock skew and drift errors are present represents an interesting open research problem.

REFERENCES

[1] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: A survey," *IEEE Netw.*, pp. 45–50, Jul./Aug. 2004.

[2] B. Sadler, "Critical issues in energy-constrained sensor networks: Synchronization, scheduling, and acquisition," in *Proc. ICASSP 2005*, 2005, vol. 5, pp. 785–788.

[3] B. Sadler and A. Swami, "Synchronization in sensor networks: An overview," in *Proc. MILCOM 2006*, Wash. DC, 2006, pp. 1–6.

[4] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: A survey," *Ad Hoc Netw.*, vol. 3, no. 3, pp. 281–323, May 2005.

[5] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[6] R. van der Merwe and E. Wan, "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models," in *Proc. Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Apr. 2003.

[7] O. L. Alspach and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Trans. Autom. Control*, vol. 17, no. 4, pp. 439–448, 1972.

[8] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000, 0001-0782.

[9] I. Akyildiz *et al.*, "Wireless sensor networks: A survey," *Comput. Netw.* , vol. 38, no. 4, pp. 393–422, Mar. 2002.

[10] H. S. Abdel-Ghaffar, "Analysis of synchronization algorithm with time-out control over networks with exponentially symmetric delays," *IEEE Trans. Commun.*, vol. 50, no. 10, pp. 1652–1661, Oct. 2002.

[11] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," in *Proc. 5th Symp. Operat. Syst. Design Implement. (OSDI 2002)*, Dec. 2002.

[12] K.-L. 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. Commun.*, vol. 55, no. 4, Apr. 2007.

[13] M. Leng and Y.-C. Wu, "On clock synchronization Algorithms for wireless sensor networks under unknown delay," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 182–190, Jan. 2010.

[14] D. Mills, "Internet time synchronization: The network time protocol (NTP)," *IEEE Trans. Commun.*, vol. 39, pp. 1482–1493, 1991.

[15] K. Noh, E. Serpedin, and K. Qaraqe, "A new approach for time synchronization in wireless sensor networks: Pairwise Broadcast Synchronization (PBS)," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, Sep. 2008.

[16] I. Skog and P. Handel, "Synchronization by two-way message exchanges: Cramer-Rao bounds, approximate maximum likelihood, and offshore submarine positioning," *IEEE Trans. Signal Process.*, vol. 58, no. 4, pp. 2351–2362, Apr. 2010.

[17] S. Ganeriwal, R. Kumar, M. B., and Srivastava , "Timing-sync protocol for sensor networks," in *Proc. 1st Int. Conf. Embedded Network Sens. Syst.* , 2003, pp. 138–149, ACM.

[18] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2006, vol. 1, pp. 228–235.

[19] A. Syed and J. Heidemann, "Time synchronization for high latency acoustic networks," USC/Inf. Sci. Inst., Tech. Rep. ISI-TR-2005-602, Apr. 2005.

[20] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd ed. New York: Springer, 1998.

[21] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1968, vol. 1.

[22] J. Kim, J. Lee, E. Serpedin, and K. Qaraqe, "A robust estimation scheme for clock phase offsets in wireless sensor networks in the presence of non-Gaussian random delays," *Signal Process.*, pp. 1155–1161, 2009.

[23] F. Kirsch and M. Vossiek, "Distributed Kalman filter for precise and robust clock synchronization in wireless networks," in *Proc. 4th Int. Conf. Radio and Wireless Symp.*, San Diego, CA, 2009, pp. 455–458.

[24] B. R. Hamilton, X. Ma, Q. Xhao, and J. Xu, "ACES: Adaptive clock estimation and synchronization using Kalman filtering," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, San Francisco, CA, 2008, pp. 152–162.

[25] S. Ganeriwal, D. Ganesan, H. Shim, V. Tsiatsis, and M. B. Srivastava, "Estimating clock uncertainty for efficient duty-cycling in sensor networks," in *ACM Sensys Conf.*, San Diego, CA, Nov. 2005, pp. 130–141.

[26] Q. Gao, K. J. Blow, and D. J. Holding, "Simple algorithm for improving time synchronization in wireless sensor networks," *Electron. Lett.*, vol. 40, p. 889, 2004.

[27] D. Tulone, "Resource-efficient time synchronization for wireless sensor networks," in *Proc. DIALM-POMC Workshop Found. Mobile Comput.*, Philadelphia, PA, Oct. 2004, p. 52-50.

[28] S. Kullback, *Information Theory and Statistics*. Mineola, NY: Dover, 1968.

[29] J. Hershey and P. A. Olsen, "Approximating the Kullback-Leibler divergence between Gaussian mixture models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP 2007)*, 2007, vol. 4, pp. 317–320.

[30] P. Tichavsky, C. Muravchik, and A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering," *IEEE Trans. Signal Process.*, vol. 46, no. 5, May 1998.

[31] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag, 2001.

[32] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *Int. J. Comput. Vision*, vol. 29, no. 1, pp. 5–28, 1998.

[33] H. Elders-Boll, H.-D. Schotten, and A. Busboom, "Efficient implementation of linear multiuser detectors for asynchronous CDMA systems by linear interference cancellation," *Eur. Trans. Telecommun.*, vol. 9, no. 5, pp. 427–438, May 1998.

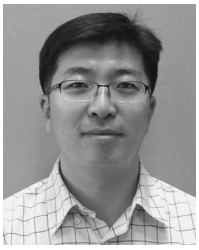
[34] S. Julier and J. K. Uhlmann, "A general method for approximating nonlinear transformations of probability distributions," Robot. Res. Group, Dep. Eng. Sci., Univ. Oxford, Oxford, U.K., Tech. Rep., 1996.

[35] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, Aug. 2005.

[36] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," in *Proc. 2nd Int. Conf. Embedded Netw. Sens. Syst. 2004*, Nov. 2004, pp. 39–49, ACM Press.

Jang-Sub Kim was born June 15, 1974, in Yeongdeok, Korea. He received the M.S. and Ph.D. degrees from the School of Electrical and Computer Engineering, Sungkyunkwan University, Seoul, Korea, in 1999 and 2005, respectively.

He is currently a Visiting Researcher with the Department of Electrical and Computer Engineering, Texas A&M University, College Station. His research interests include signal processing and its applications in wireless communications and networking, synchronization for wireless sensor networks and wireless communications, and sequential Monte Carlo methods.



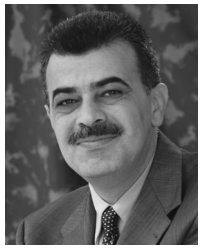
Jaehan Lee received the B.S. degree and the M.S. degree in electrical engineering from Seoul National University, Korea, in 1997 and 1999, respectively. He received the Ph.D. degree in electrical engineering from Texas A&M University, College Station, in 2010.

Since then, he has been working as a principal engineer for Samsung SDS in Seoul, Korea. His research interests include clock synchronization in WSNs, Internet of Things (IoT), Web of Things (WoT), and M2M communications.

Erchin Serpedin received the specialization degree in signal processing and transmission of information from Ecole Superieure D'Electricite (SUPELEC), Paris, France, in 1992, the M.Sc. degree from the Georgia Institute of Technology, Atlanta, in 1992, and the Ph.D. degree in electrical engineering from the University of Virginia, Charlottesville, in January 1999.

In July 1999, he joined Texas A&M University, College Station, where he currently holds the position of Professor. His research interests lie in the areas of signal processing, telecommunications, bioinformatics, and genomics. He is the author of 70 journal papers and 100 conference papers, and coauthor of the research monograph *Synchronization in Wireless Sensor Networks* (Cambridge, U.K.: Cambridge University Press, August 2009).

Dr. Serpedin received the NSF Career Award in 2001, the Outstanding Faculty Award in 2004, and several other awards. He served as an Associate Editor for the IEEE COMMUNICATIONS LETTERS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE SIGNAL PROCESSING LETTERS, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and EURASIP *Journal on Advances in Signal Processing*. He served as a Technical Co-Chair of the Communications Theory Symposium at Globecom 2006 Conference, and for the VTC Fall 2006: Wireless Access Track Symposium. He is currently serving as Associate Editor for the journals *Signal Processing- Elsevier*, the IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION THEORY, EURASIP *Journal on Advances in Signal Processing*, and EURASIP *Journal on Bioinformatics and Systems Biology*.



Khalid Qaraqe (M'97–SM'00) was born in Bethlehem. He received the B.S. degree from the University of Technology, Baghdad, in 1986, with honors, the M.S. degree from the University of Jordan, Jordan, in 1989, and the Ph.D. degree from Texas A&M University, College Station, in 1997, all in electrical engineering.

From 1989 to 2004, he held a variety of positions in many companies. He has more than 12 years of experience in the telecommunication industry. He has worked for Qualcomm, Enad Design Systems, Cadence Design Systems/Tality Corporation, STC, SBC, and Ericsson. He has also worked on numerous GSM, CDMA, WCDMA projects and has experience in product development, design, deployment, testing, and integration. He joined the Department of Electrical Engineering, Texas A&M University at Qatar, in July 2004, where he is now an Associate Professor. His research interests include communication theory and its application to design and performance analysis of cellular systems and indoor communication systems. Particular interests are in the development of 3G UMTS, cognitive radio systems, broadband wireless communications, and diversity techniques.