ROBUST CLOCK SYNCHRONIZATION METHODS FOR WIRELESS SENSOR NETWORKS

A Dissertation

by

JAE HAN LEE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2010

Major Subject: Electrical Engineering

ROBUST CLOCK SYNCHRONIZATION METHODS FOR WIRELESS SENSOR NETWORKS

A Dissertation

by

JAE HAN LEE

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	Erchin Serpedin
	Khalid Qaraqe
Committee Members,	Deepa Kundur
	Aydin Karsilayan
	Eun Jung Kim
Head of Department,	Costas Georghiades

August 2010

Major Subject: Electrical Engineering

ABSTRACT

Robust Clock Synchronization Methods

for Wireless Sensor Networks. (August 2010)

Jae Han Lee,

B.S., Seoul National University, Seoul, Korea;

M.S., Seoul National University, Seoul, Korea

Co–Chairs of Advisory Committee: Dr. Erchin Serpedin
Dr. Khalid Qaraqe

Wireless sensor networks (WSNs) have received huge attention during the recent years due to their applications in a large number of areas such as environmental monitoring, health and traffic monitoring, surveillance and tracking, and monitoring and control of factories and home appliances. Also, the rapid developments in the micro electro-mechanical systems (MEMS) technology and circuit design lead to a faster spread and adoption of WSNs. Wireless sensor networks consist of a number of nodes featured in general with energy-limited sensors capable of collecting, processing and transmitting information across short distances. Clock synchronization plays an important role in designing, implementing, and operating wireless sensor networks, and it is essential in ensuring a meaningful information processing order for the data collected by the nodes. Because the timing message exchanges between different nodes are affected by unknown possibly time-varying network delay distributions, the estimation of clock offset parameters represents a challenge. This dissertation presents several robust estimation approaches of the clock offset parameters necessary for time synchronization of WSNs via the two-way message exchange mechanism. In this dissertation the main emphasis will be put on building clock phase offset estimators

robust with respect to the unknown network delay distributions.

Under the assumption that the delay characteristics of the uplink and the downlink are asymmetric, the clock offset estimation method using the bootstrap bias correction approach is derived. Also, the clock offset estimator using the robust M-estimation technique is presented assuming that one underlying delay distribution is mixed with another delay distribution.

Next, although computationally complex, several novel, efficient, and robust estimators of clock offset based on the particle filtering technique are proposed to cope with the Gaussian or non-Gaussian delay characteristics of the underlying networks. One is the Gaussian mixture Kalman particle filter (GMKPF) method. Another is the composite particle filter (CPF) approach viewed as a composition between the Gaussian sum particle filter and the KF. Additionally, the CPF using bootstrap sampling is also presented. Finally, the iterative Gaussian mixture Kalman particle filter (IGMKPF) scheme, combining the GMKPF with a procedure for noise density estimation via an iterative mechanism, is proposed.

To Najung Kim

ACKNOWLEDGMENTS

I have no words to thank my father, parents-in-law, my wife and two sons enough for their everlasting love and prayers that helped me finish this course successfully. I would also like to deeply thank my co-chair, Dr. Erchin Serpedin, who not only gave me a lot of sincere advice for my research and life, but also supported me in every possible aspect.

In addition, I want to express my gratitude for my co-chair, Dr. Khalid Qaraqe, and my committee members, Dr. Deepa Kundur, Dr. Aydin Karsilayan, Dr. Eun Jung Kim and the Department Head Dr. Costas Georghiades for their academic guidance. I must also acknowledge my colleagues Yi Zhou, Sangwoo Park, Kwadwo Agyepong, Sabit Ekin and Sawin Saibua, with whom I had a wonderful time at Texas A&M University in College Station.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

A. Clock Synchronization in Wireless Sensor Networks

The recent technological advances in micro-electro-mechanical systems (MEMS) have accelerated the development of tiny, low-cost, low-power, and multifunctional sensors with the ability to perform tasks such as sensing, data processing, and communication [1]-[4]. In general, a wireless sensor network (WSN) consists of a large number of sensors, which are densely distributed over a wide geographical area to collect data and monitor a certain physical phenomenon. The positions of wireless sensor nodes do not need to be engineered or predetermined beforehand, a fact which enables random deployment in inaccessible terrains or disaster relief operations. Hence, wireless sensor network protocols and algorithms with self-organizing capabilities are needed. Another peculiar characteristic of wireless sensor networks is the collaborative effort of sensor nodes to carry out tasks such as detection, measurement, and data fusion. Specifically, sensor nodes take advantage of their own processing abilities to locally perform simple computations and send only the necessary and partially processed data to the destination node without sending the raw data to the sink node. In other words, a single meaningful result value is created from the data collected from each sensor [5].

The applications of wireless sensor networks include areas as diverse as medical, industrial, civil, military, environmental, scientific, and home appliances [6]-[12]. For instance, doctors can identify predefined symptoms by monitoring the physiological data of patients remotely via WSNs. In terms of military applications, WSNs can be

---

´The journal model is *IEEE Transactions on Automatic Control*.

used to detect nuclear, biological, and chemical attacks and the presence of hazardous materials, prevent enemy attacks by means of alerts when enemy aircrafts are spotted, and monitor friendly forces, equipment and ammunition. Besides, WSNs are useful in monitoring forest fires, observing ecological and biological habitats, and detecting floods and earthquakes. As examples of civilian applications, WSNs can be used to determine spot availability in a public parking lot, track active badge at the workplace, observe security in public places such as banks and shopping malls, and monitor highway traffic. Moreover, WSNs can satisfy the needs for scientific applications such as space and interplanetary exploration, high energy physics, and deep undersea exploration [13].

Since the sensors in a WSN operate independently, their own local clocks may not be synchronized to the same time value, and thus this can cause difficulties when trying to integrate and interpret information collected at different nodes. For example, if a moving car is detected at two different times along a certain road, the detection times need to be compared appropriately before we decide which direction the car is going in. Moreover, we should be able to transform the two time values into a common frame of reference before estimating the speed of the vehicle. It is also important to estimate precisely the time differences across nodes in node localization applications. As an example, many localization algorithms utilize ranging technologies to estimate internode distances; in these technologies, synchronization is necessary for time-of-flight measurements which are then transformed into distances by multiplying with the medium propagation speed for the type of signal used such as radio frequency or ultrasonic. There are additional examples where cooperative sensing requires the nodes involved to agree on a common time frame such as configuring a beam-forming array and setting a TDMA (Time Division Multiple Access) radio scheduling mechanism [14]. For all these situations, a common notion of time

is absolutely necessary in WSNs. Hence, there is currently a tremendous interest towards developing energy efficient clock synchronization protocols for WSNs.

The clock synchronization problem has been studied thoroughly in the areas of Internet and local-area networks (LANs) for the last several decades. Many existing synchronization algorithms depend on the clock information from GPS (Global Positioning System). However, GPS-based clock acquisition mechanisms exhibit some weaknesses: GPS is not available ubiquitously (for example, underwater, indoors, under foliage) and requires a relatively high-power receiver, which is almost impossible on tiny and cheap sensor nodes. In addition, GPS is very sensitive to narrowband interferences. This is the motivation for developing software-based approaches to achieve in-network time synchronization. Among many protocols that have been developed for maintaining synchronization in computer networks, NTP (Network Time Protocol) [15] is outstanding due to its ubiquitous deployment, scalability, robustness related to failures, and self-configuration in large multihop networks. Futhermore, the combination of NTP and GPS has shown that it is capable of achieving high accuracy on the order of a few microseconds [16]. However, NTP is not appropriate for a wireless sensor network environment because WSNs pose numerous challenges of their own; to name a few, limited energy and bandwidth, limited hardware, latency, and unstable network conditions caused by mobility of sensors, dynamic topology, and multi-hopping. Therefore, clock synchronization protocols different from the conventional protocols are needed to deal with the challenges specific to WSNs. Subsequently, a large number of synchronization protocols have been devised in recent years.

One of the representative examples of clock synchronization protocols in WSNs is Reference Broadcast Synchronization (RBS) [17], which is a pioneering work based on the post-facto receiver-receiver synchronization scheme. In RBS, a broadcast message

is only used to synchronize a set of receivers with one another, in contrast with traditional protocols that synchronize the sender of a message with its receiver. A parent node broadcasts a reference message to the neighboring nodes which receive the broadcasted message and record the time according to their own local clocks. The nodes exchange their own collected observations and use a linear regression technique to estimate the relative clock offset and skew.

Timing-Synch Protocol for Sensor Networks (TPSN) [18] is based on the traditional approach of sender-receiver synchronization. TPSN relies on the two-way message exchange scheme to acquire the synchronization between two nodes. The first step of the TPSN protocol, which is referred to as level discovery phase, is to create a hierarchical spanning tree in the network. Each sensor node is assigned a level in this hierarchical structure. A node belonging to level i can communicate with at least one node belonging to level $i-1$. Only one node is assigned level 0, and it is referred to as the root node. Once the hierarchical structure is completed, the root node initiates the second stage of the protocol, called the synchronization phase. In this second phase, a node with level i synchronize to a node with level $i-1$. Finally, every node is synchronized to the root node with level 0 by means of a two-way message exchange scheme by adjusting the clock offset, and network-wide time synchronization is achieved.

Flooding Time Synchronization Protocol (FTSP) [19] aims at attaining a network wide synchronization of the local clocks of participating nodes by means of multi-hop synchronization. FTSP synchronizes the time of a sender to possibly multiple receivers taking advantage of a single radio message time-stamped at both the sender and the receiver sides. MAC layer time-stamping can eliminate many of the errors, as shown in TPSN [18]. However, accurate clock synchronization at discrete points in time is only a partial solution and thus compensation for the clock drift of the nodes

is necessary in order to obtain high precision in-between synchronization points and to keep the communication overhead low. Linear regression technique is used in this protocol to compensate for clock drift as in RBS [17].

Time Diffusion Synchronization Protocol (TDP) [20] is a network-wide time synchronization protocol and enables all the sensors in the network to have a local time that is within a small bounded time deviation from the network-wide equilibrium time. TDP architecture comprises many algorithms and procedures, which are used to autonomously synchronize the nodes, remove the false tickers (clocks deviating from those of the neighbors), and balance the load required for time synchronization among the sensor nodes.

Recently, Pairwise Broadcast Synchronization (PBS) protocol was proposed in [21], where both sender-receiver synchronization and receiver-only synchronization shemes are efficiently combined to obtain network-wide synchronization with a significantly reduced number of timing messages [21].

B. General Clock Model

In general, computer clocks are based on crystal oscillators which present a local time to each network node. The time in a computer clock is called software clock in that it is just a counter increased by crystal oscillators. Whenever an interrupt takes place, the corresponding interrupt handler has to increase the software clock by one. Since the frequency making time increase is not perfect, most hardware oscillators are not so accurate. Even a frequency deviation of only 0.001% would lead to a clock error of about one second per day. Considering the physical clock synchronization in a distributed system referenced to UTC (Universal Time Controller), the computer clock indicates time $C(t)$ at real time t, which may or may not be t. In case of a

perfect clock, the derivative $dC(t)/dt$ should be 1. This term is referred to as clock skew. Due to environmental conditions, such as temperature and humidity, the clock skew can actually vary over time, but it is assumed that it stays bounded and close to 1, so that

$$1 - \beta \leq \frac{dC(t)}{dt} \leq 1 + \beta \quad , \tag{1.1}$$

where β represents the maximum skew rate. A typical value of the maximum skew for today's hardware is 10^{-6}. It is noted that the clocks of two nodes are synchronized to one common time at an arbitrary time, but they do not remain synchronized in the future because of clock skew. Even if there is no skew, the clocks of different nodes may not be equal. Time differences from the lack of a common time origin are called clock phase offsets. Fig. 1.1 shows the behavior of fast, slow, and perfect clocks with respect to UTC [13], [14]. The time of a clock in a sensor node P is the

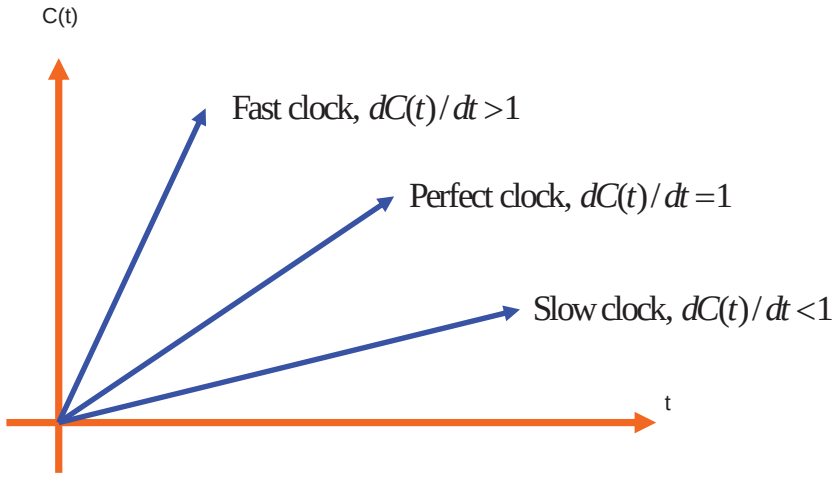

Fig. 1.1. Behavior of fast, slow, and perfect clocks

function $C_P(t)$, where $C_P(t) = t$ for a perfect clock and the clock frequency is the rate

at which a clock progresses. Clock offset is defined to be the difference between the times reported by the clocks at two sensor nodes; in other words, the offset of clock C_P relative to C_Q at time t is given by $C_P(t) - C_Q(t)$. Clock skew is defined as the difference in the frequencies of two clocks $C'_P(t) - C'_Q(t)$, i.e., the rate of variation or derivative of clock offset. Another clock terminology is clock drift, which is the second derivative of the clock offset with respect to time. The mathematical expression of the drift of clock C_P relative to C_Q at time t is $C''_P(t) - C''_Q(t)$ [13].

In order for sensor nodes to be synchronized to one common time, they need to access and use for a certain amount of time a communication channel where the message delays between nodes can be reliably estimated. However, the obstacle of accurate network time synchronization is the non-determinism of the delay estimation process. Random events, which give rise to asymmetric round-trip-message delivery delays, make the estimation of latencies in channel more difficult, which finally causes synchronization errors. The latency in channel can be decomposed into several distinct components.

C. Decomposition of Packet Delay

An outline of the various delay components is presented from a system perspective in [18], and will be briefly described below.

1. Send time: The delay in the packet traversal from the message assembly at the application layer all the way down to MAC layer.

2. Access time: The waiting time of a packet for accessing the channel after reaching the MAC layer. This is the most critical factor contributing to packet delay, and highly variable according to the specific MAC protocol. This is nondeterministic and varies up to hundreds of milliseconds.

3. Transmission time: The time taken by a packet which is transmitted bit by bit at the physical layer over the wireless link. This delay is mainly deterministic in nature and can be estimated given the packet size and the radio speed.

4. Propagation time: The actual time taken by the packet to traverse the wireless link from the sender to the receiver.

5. Reception time: The time spent in receiving the bits and passing them to the MAC layer.

6. Receive time: The time taken in constructing the bits into a packet and then passing this packet to the application layer.

These are the sources of delay and most of the existing clock synchronization protocols focus on removing them.

D. Contributions of This Research

In [22], it was proved that the maximum likelihood estimate (MLE) of the clock offset does exist and corresponds to the estimate proposed previously in [23], which was inferred based on experimental data, for an unknown fixed delay and propagation delay that obeys a symmetric exponential delay model. Also, [24] derived the MLE of the clock offset under a Gaussian delay model. This dissertation substantially provides an extension of these two researches on clock synchronization problem as follows.

Chapter II focuses on developing clock offset estimators based on the bootstrap bias correction approach, which is used to estimate and correct the bias of the MLE in the exponential delay model, and more general asymmetric exponential, gamma, and Weibull random delay models assuming the two-way timing message exchange

mechanism. The developed estimators show better performance than the exponential MLE [22] in terms of mean-square error (MSE).

Chapter III describes the clock offset estimators based on the robust M-estimation method [25]. The underlying delay distribution may be mixed with or contaminated by other delay distributions, and thus obtaining long-term reliable and energy efficient clock offset estimators becomes more difficult. For such general mixed delay models, the proposed estimators are shown to exhibit better performance than MLE.

Chapter IV discusses a novel clock offset estimation method based on the Gaussian Mixture Kalman Particle Filter (GMKPF), which has the merits of being robust and yielding very accurate clock offset estimates in the presence of arbitrary network delay distributions and general two-way timing message exchange mechanism. GMKPF combines the importance sampling (IS) based measurement update step with a KF (Kalman Filter) based Gaussian sum filter for the time-update and proposal density generation. Since GMKPF employs new observations and exploits the Expectation-Maximization (EM) algorithm to fit a Gaussian Mixture Model (GMM) to the posterior and observation distributions, GMKPF is expected to exhibit better estimation performance in general non-Gaussian/non-exponential delays when compared to MLE, which was originally derived only for Gaussian or exponential delays.

Chapter V deals with another novel clock offset estimation method, called the composite particle filter (CPF), which is shown to be robust to the unknown distribution of network delays. The CPF approximates the filtering and predictive distributions by using weighted Gaussian mixtures and is basically implemented via banks of Kalman filters (KFs) instead of Gaussian Particle filters (GPFs) [26]. Therefore, CPF appears as a variation of the Gaussian sum particle filter (GSPF) [27], fit for estimation of linear models perturbed by non-Gaussian random noise components. CPF is also combined with the bootstrap sampling technique, and shown to exhibit

improved performance.

Chapter VI proposes a novel iterative clock offset estimation method, referred to as the Iterative Gaussian Mixture Kalman Particle Filter (IGMKPF), where the Gaussian Mixture Kalman Particle Filter (GMKPF) is combined with a network delay density estimation approach to yield better performance than GMKPF. The proposed method has the merits of being robust and yielding very accurate clock offset estimates in the presence of unknown network delay distributions. At each iteration, by taking into account the results (posterior density, observation noise density, and so on) from the previous iteration, a new set of parameter values is obtained by adding small changes to the previous parameters in such a way that the new parameters are likely to lead to improved performance. By estimating the posterior density and observation noise density using an iterative technique, the estimator reduces the bias caused by taking expectation. To validate the accuracy of IGMKPF, its MSE performance is evaluated via computer Monte Carlo simulations and the posterior Cramer-Rao bound (PCRB) is derived too. A sequential Monte Carlo method is also proposed for computing the PCRB in unknown delay models. This chapter provides also a comparison of Gaussian MLE, exponential MLE, the Cramer-Rao Lower Bound (CRLB), IGMKPF, IGMKPF with perfect network delay noise estimation, and PCRB.

Lastly, Chapter VII summarizes the research results of this dissertation and also presents some directions for possible future research work. It is remarkable that the results from this dissertation can be applied and extended to improve the performance of a variety of clock synchronization protocols.

CHAPTER II

CLOCK OFFSET ESTIMATION USING BOOTSTRAP BIAS CORRECTION

Under the assumption that there is no clock skew between sensor nodes, the MLE of clock offset was proved by [22] for clock synchronization protocols that assume exponential random delays and a two-way message exchange mechanism as is the case with TPSN [18], NTP [15], and PBS [21] protocols. The proposed MLE is asymptotically unbiased. However, the MLE estimator is biased in the presence of a finite number of samples and much more biased in asymmetric random delay models, where the upstream delay characteristics are different from the downstream delay characteristics, and thus its performance is deteriorated. This chapter deals with clock offset estimators based on the bootstrap bias correction approach, which estimates and corrects the bias of the MLE in the exponential delay model, and hence it results in better performance in mean-square error (MSE).

A. Problem Modeling

The two-way timing message exchange scheme between two nodes is shown in Fig. 2.1. *Node A* sends the synchronization message to *Node B* with its current timestamp $T_{1,k}$ (although it is not required, timestamping in the MAC layer increases the accuracy, as suggested by [18]). *Node B* records its current time $T_{2,k}$ at the reception of this message, and then sends at time $T_{3,k}$ a synchronization message to *Node A* containing $T_{2,k}$ and $T_{3,k}$. *Node A* timestamps the reception time of the message sent by *Node B* as $T_{4,k}$ (see Fig. 2.1). Hence, *Node A* has access to the set of timestamps $\{T_{1,k}, T_{2,k}, T_{3,k}, T_{4,k}\}$, $k = 1, \ldots, N$, at the end of each of the N rounds of message exchanges. Note that $T_{1,1}$ is considered to be the reference time, and hence every reading $\{T_{1,k}, T_{2,k}, T_{3,k}, T_{4,k}\}$ is actually the difference between the recorded time and

$T_{1,1}$.

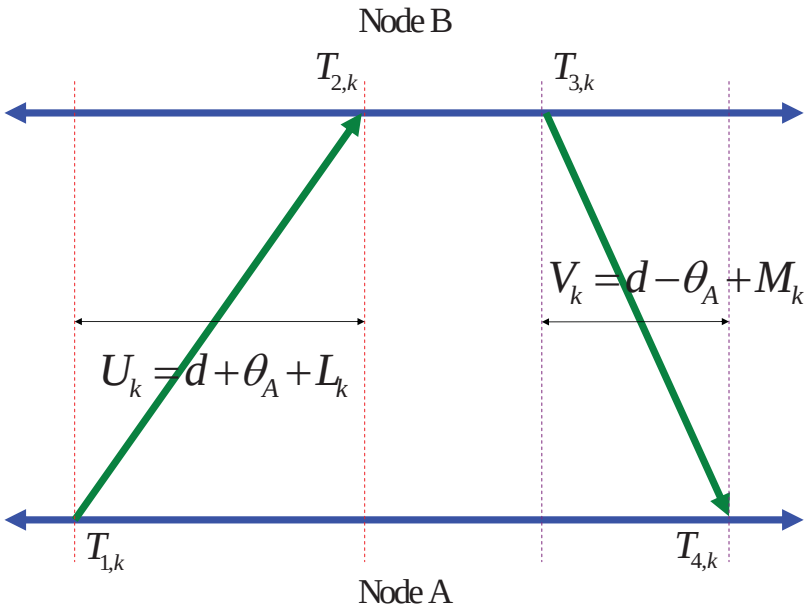

Fig. 2.1. Two-way timing message exchange model with clock offset (θ_A: clock offset, d: propagation delay, L_k, M_k: random delays)

Modeling of network delays in WSN seems to be a challenging task [28]. Several probability distribution function (pdf) models for random queuing delays have been proposed so far, the most widely used being exponential, gamma, Weibull, and log-normal distributions [29], [30]. Amongst them, the exponential distribution is suited well for several applications [31]. Also, a single-server M/M/1 queue can fittingly represent the cumulative link delay for point-to-point Hypothetical Reference Connection (HRC), where the random delays are independently modeled as exponential random variables [32]. Moreover, [23] experimentally demonstrated the superiority of the MnLD (Minimum Link Delay) algorithm among the various algorithms proposed by [32]. Reference [22] has shown mathematically that MnLD represents the MLE in the context of exponential delays, thus confirming the fact that the exponential delay assumption fits well the experimental observations. The reason for adopting

the Gaussian model is due to the Central Limit Theorem (CLT), which asserts that the pdf of the sum of a large number of independent or in general, slightly correlated and identically distributed (iid) random variables (RVs) approaches a Gaussian RV. This model is proper if the delays are thought to be the addition of numerous independent random processes. The Gaussian distribution for the clock offset errors was also reported by a few authors based on laboratory tests [17].

Next, it is assumed that there is no clock skew in the two-way timing message exchange model. The k_{th} up and down link delay observations corresponding to the k_{th} timing message exchange are given by $U_k = T_{2,k} - T_{1,k} = d + \theta_A + L_k$ and $V_k = T_{4,k} - T_{3,k} = d - \theta_A + Y_k$, respectively. The fixed value θ_A, d, L_k and M_k denote the clock offset between the two nodes, the propagation delay, and variable portions of delays, respectively. In general, the delay distribution in the upstream, F_L is not equal to that in the downstream, F_M, because the *node A* $\rightarrow$ *node B* and *node B* $\rightarrow$ *node A* transmission paths through the network typically have different traffic characteristics, and thus the network delays in each path are potentially different. Hence, it is important to estimate the clock offset more accurately in asymmetric delay models.

In the symmetric exponential delay model, the MLE of θ_A exists and is given by

$$\hat{\theta}_A = \frac{\min\limits_{1\le i\le N} U_k - \min\limits_{1\le i\le N} V_k}{2} \quad , \tag{2.1}$$

where N denotes the number of observations of delay measurements [22].

B. The Bootstrap Estimation Method

If the MLE is used in the asymmetric exponential delay model, there will be a bias in the clock offset. Therefore, it is necessary to achieve a more accurate estimate of the

clock offset by using an alternate approach. In this chapter, bootstrap techniques are applied to estimating the clock offset in the asymmetric delay models. Bias-corrected estimators through non-parametric and parametric bootstrap will be analyzed. The procedure for bootstrap bias correction is as follows [33], [34].

1. The Nonparametric Bootstrap

Step 0. Conduct the experiment to obtain the random sample $X = \{X_1, X_2, \ldots, X_n\}$ and calculate the estimate $\hat{\theta}$ from the sample X.

Step 1. Construct the empirical distribution $\hat{F}$, which puts equal mass $1/n$ at each observation $X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n$

Step 2. From $\hat{F}$, draw a sample $X^* = \{X_1^*, X_2^*, \ldots, X_n^*\}$, called the bootstrap resample.

Step 3. Approximate the distribution of $\hat{\theta}$ by the distribution of $\hat{\theta}^*$ derived from the bootstrap resample X^*.

2. The Parametric Bootstrap

Suppose that one has some partial information about F. For example, F is known to be the exponential distribution but with unknown mean μ. This suggests that we should draw a resample of size n from the exponential distribution with mean $\hat{\mu}$ where $\hat{\mu}$ is estimated from X rather than from a non-parametric estimate $\hat{F}$ of F. We use the exponential distribution in the suggested bias correction approach through parametric bootstrapping. The parametric bootstrap principle is almost the same as the above non-parametric bootstrap principle except some steps.

3. The Bootstrap Estimate of Bias

Let us suppose that an unknown probability distribution F assumes the data $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ by random sampling, $F \to x$. We want to estimate a real-valued parameter $\theta = t(F)$. For now we will assume the estimator to be any statistic $\hat{\theta} = s(\mathbf{x})$. The *bias* of $\hat{\theta} = s(\mathbf{x})$ as an estimate of θ is defined to be the difference between the expectation of $\hat{\theta}$ and the value of the parameter θ,

$$\text{bias}_F = \text{bias}_F(\hat{\theta}, \theta) = E_F[s(\mathbf{x})] - t(F) \quad . \tag{2.2}$$

A large bias is usually an undesirable aspect of an estimator's performance. We can use the bootstrap to assess the bias of any estimator $\hat{\theta} = s(\mathbf{x})$. The *bootstrap estimate of bias* is defined to be the estimate $\text{bias}_{\hat{F}}$ obtained by substituting $\hat{F}$ for F,

$$\text{bias}_{\hat{F}} = E_F[s(\mathbf{x}^*)] - t(\hat{F}) \quad . \tag{2.3}$$

For most statistics that arise in practice, the ideal bootstrap estimate $\text{bias}_{\hat{F}}$ must be approximated by Monte Carlo simulations. We generate independent bootstrap samples $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \ldots, \mathbf{x}^{*B}$, evaluate the boostrap replications $\hat{\theta}^*(b) = s(\mathbf{x}^{*b})$, and approximate the bootstrap expectation $E_{\hat{F}}[s(\mathbf{x}^*)]$ by the average

$$\hat{\theta}^*(\cdot) = \frac{\sum_{b=1}^{B} \hat{\theta}^*(b)}{B} = \frac{\sum_{b=1}^{B} s(\mathbf{x}^{*b})}{B} \quad . \tag{2.4}$$

The bootstrap estimate of bias based on the B replications $\hat{\text{bias}}_B$ is (2.3) with $\hat{\theta}^*(\cdot)$ substituted for $E_{\hat{F}}[s(\mathbf{x}^*)]$,

$$\hat{\text{bias}}_B = \hat{\theta}^*(\cdot) - t(\hat{F}) \quad . \tag{2.5}$$

4. Bias Correction

The usual reason why we want to estimate the bias of $\hat{\theta}$ is to correct $\hat{\theta}$ so that it becomes less biased. If $\widehat{\text{bias}}$ is an estimate of $\text{bias}_F(\hat{\theta}, \theta)$, then the obvious *bias-corrected estimator* is

$$\bar{\theta} = \hat{\theta} - \widehat{\text{bias}} \quad . \tag{2.6}$$

Taking $\widehat{\text{bias}}$ equal to $\widehat{\text{bias}}_B = \hat{\theta}^*(\cdot) - \hat{\theta}$ gives

$$\bar{\theta} = 2\hat{\theta} - \hat{\theta}^*(\cdot) \quad . \tag{2.7}$$

C. Performance Analysis

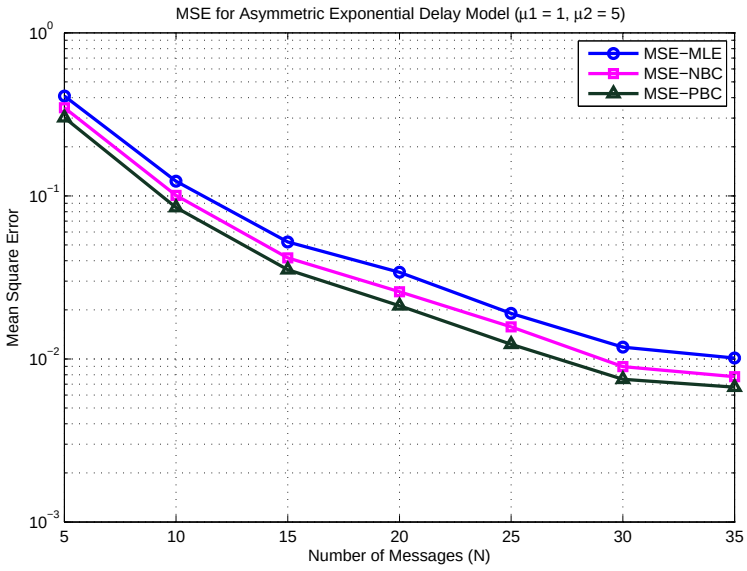

Fig. 2.2. MSEs of clock offset estimators for asymmetric exponential delays ($\mu_1 = 1, \mu_2 = 5$)

This section provides the simulation results of the performances of the MLE and the estimators based on bootstrap bias correction approach for asymmetric exponential, gamma, and Weibull delays, respectively. Figs. 2.2 - 2.5 show the MSEs in

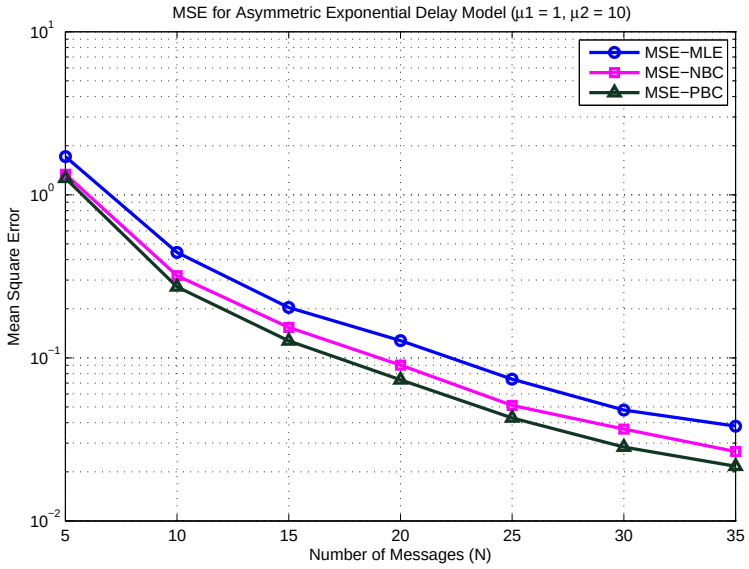

Fig. 2.3. MSEs of clock offset estimators for asymmetric exponential delays $(\mu_1 = 1, \mu_2 = 10)$

environments where the distributions of the network delays are exponential, gamma, and Weibull. In Figs. 2.2 and 2.3, μ_1 and μ_2 denote the exponential delay parameters for the uplink and the downlink delay distributions, respectively. For the gamma delay model, the shape parameters (α_1 and α_2) were chosen to be two, and the scale parameters (β_1 and β_2) taken to be one and two for the uplink delay and the downlink delay, respectively. In the Weibull case, the shape parameters (β_1 and β_2) were set to be two, and the scale parameters (α_1 and α_2) chosen to be two and six for the uplink delay and the downlink delay, respectively. In Figs. 2.2 - 2.5, MSE-MLE, MSE-NBC, and MSE-PBC denote the mean square error of estimate (2.1), which is the MLE in the symmetric exponential delay model, that of the bias-corrected estimator through nonparametric bootstrapping, and that of the bias-corrected estimator obtained through parametric bootstrapping, respectively. It is clear that the performance of the bias-corrected estimators is improved for several asymmetric delay

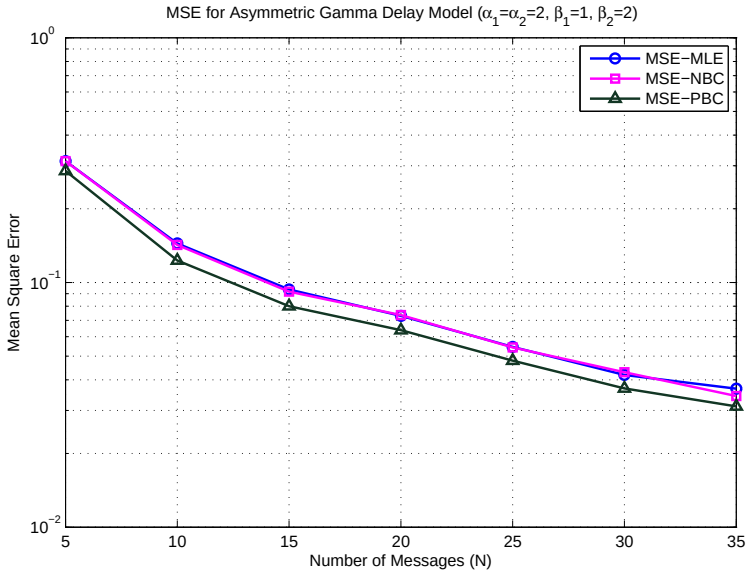

Fig. 2.4. MSEs of clock offset estimators for asymmetric gamma delays $(\alpha_1 = \alpha_2 = 2, \beta_1 = 1, \beta_2 = 2)$

models and the bias corrected estimator via the parametric bootstrapping method has the best performance for the asymmetric exponential, gamma, and Weibull delay distributions. In Figs. 2.2 and 2.3, we also notice that as the asymmetry of the downlink and the uplink increases, the MSE values are affected and increased by the variance of observations.

D. Summary

In this chapter, clock offset estimators based on the bootstrap bias correction method were proposed for synchronization protocols involving a two-way message exchange mechanism. It is also illustrated that bootstrap bias-corrected clock offset estimators present smaller MSEs than MLE for several general exponential family delay distributions. Moreover, the bootstrap bias-corrected estimator with parametric bootstrapping produces the best performance in terms of MSE.

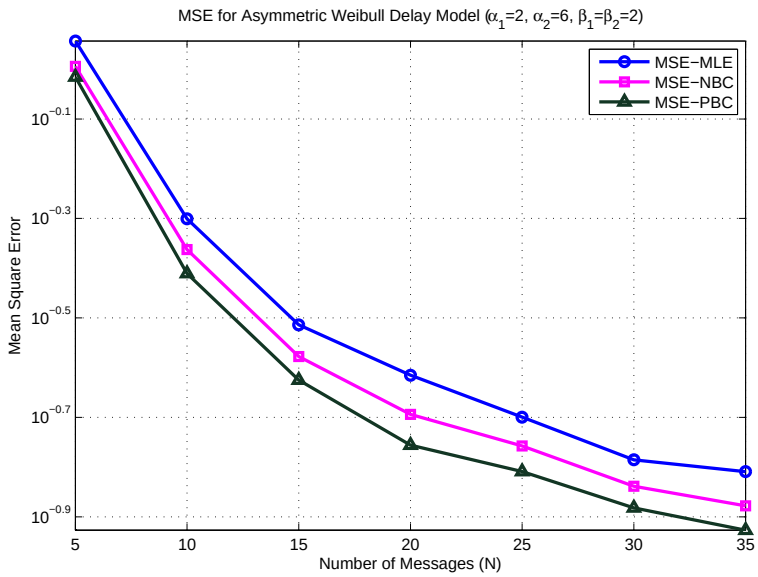

Fig. 2.5. MSEs of clock offset estimators for asymmetric Weibull delays $(\alpha_1 = 2, \alpha_2 = 6, \beta_1 = \beta_2 = 2)$

CHAPTER III

CLOCK OFFSET ESTIMATION USING ROBUST M-ESTIMATION

In practice, the distribution of network delays might be affected by numerous factors and might change over time. This chapter deals with developing clock phase offset estimation techniques that are robust to situations where the underlying network delay distribution is contaminated by or mixed with other random delay distributions. In these cases, the performance of the MLE is deteriorated considerably. This chapter focuses on clock offset estimators based on the robust M-estimation method and shows that the proposed estimators present excellent performance in the mean square error (MSE) sense when the underlying distribution is mixed with other distribution.

A. Problem Modeling

The problem set-up is the same as the one introduced in Chapter II. It is assumed that there is no clock skew in the two-way timing message exchange model. The k_{th} up and down link delay observations corresponding to the k_{th} timing message exchange are given 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 variables θ_A, d, L_k and M_k denote the clock offset between the two nodes, the propagation delay, and variable portions of delays, respectively.

In the exponential delay model and the Gaussian delay model, the MLE of θ_A exists and is given by

$$\hat{\theta}_A = \frac{\min\limits_{1\le i\le N} U_k - \min\limits_{1\le i\le N} V_k}{2} \quad , \tag{3.1}$$

$$\hat{\theta}_A = \frac{\sum\limits_{i=1}^{N}(U_k - V_k)}{2N} \quad , \tag{3.2}$$

respectively, where N denotes the number of observations of delay measurements [22], [24]. However, these MLEs are not suited for the case that the underlying distribution is mixed with or contaminated by other distributions. Therefore, it is important to estimate the clock offset more accurately in mixed delay models.

B. The M-Estimation Method

The procedures of applying robust M-estimation [25], [35] to the clock offset model are as follows.

The M-estimator T_n is defined as the solution of the minimization problem:

$$\arg\min_{T_n} \sum_{k=1}^{n} \rho\left(X_k, \theta\right) \quad ,$$

where $\rho(\cdot,\cdot)$ is a properly chosen function. The class of M-estimators covers also the *maximal likelihood estimator* (MLE) of parameter θ in the parametric model $P = \{P_\theta, \theta \in \Theta\}$. If $f(x,\theta)$ is the density function of P_θ, then the MLE is a solution of the minimization:

$$\arg\min_{T_n} \sum_{k=1}^{n} \left(-\log f\left(X_k, \theta\right)\right), \quad \theta \in \Theta \quad .$$

If ρ is differentiable with regard to θ with a continuous derivative $\psi(\cdot,\theta) = \frac{\partial}{\partial\theta}\rho(\cdot,\theta)$, then T_n is a root (or one of the roots) of the equation:

$$\sum_{k=1}^{n} \psi\left(X_k, \theta\right) = 0, \quad \theta \in \Theta \quad .$$

Hence

$$\frac{1}{n}\sum_{k=1}^{n} \psi(X_k, T_n) = \mathbb{E}_{P_n}\left[\psi\left(X_k, T_n\right)\right] = 0, \quad T_n \in \Theta \quad .$$

An important special case is the model with the shift parameter θ, where $X_1, \ldots, X_n$ are independent observations with the same distribution function $F(x-\theta)$, $\theta \in \mathbb{R}$.

The distribution function F is generally unknown. The M-estimator T_n is defined as the solution of the minimization

$$\arg\min_{T_n} \sum_{k=1}^{n} \rho\left(X_k - \theta\right) \quad ,$$

and if $\rho(\cdot)$ is differentiable with absolutely continuous derivative $\psi(\cdot)$, then T_n satisfies the equation

$$\sum_{k=1}^{n} \psi\left(X_k - \theta\right) = 0 \quad .$$

If we look for an M-estimator of the location parameter of a distribution not very far from the normal distribution, but possibly containing an ε ratio of nonnormal data, more precisely, belonging to the family of distributions

$$\mathcal{F} = \{F : F = (1-\varepsilon)\Phi + \varepsilon H\} \quad ,$$

where H runs over the set of symmetric distribution functions, we should use the function ψ, proposed and motivated by P.J. Huber (1964) [25]. This function is linear in the bounded interval $[-k, k]$, and constant outside this interval.

$$\psi_H(x) = \begin{cases} x & \text{for } |x| \le k \\ k\text{sign}(x) & \text{for } |x| > k \end{cases} \quad , \tag{3.3}$$

where $k > 0$ is a fixed constant, connected with ε through the following identity:

$$2\Phi(k) - 1 + \frac{2\Phi'(k)}{k} = \frac{1}{1-\varepsilon} \quad . \tag{3.4}$$

The corresponding M-estimator is often called the *Huber estimator* in the literature. It is a robust estimator of the center of symmetry, insensitive to the extreme and outlying observations. As Huber proved in 1964, an estimator, generated by the function $\psi_H(x)$, is minimaximally robust for a contaminated normal distribution, while the value k depends on the contamination ratio. An interesting and natural

question is whether there exists a distribution F such that the Huber M-estimator is the maximal likelihood estimator of θ for $F(x-\theta)$, i.e., such that ψ is the likelihood function for F. Such a distribution really exists, and its density is normal in the interval $[-k, k]$, and exponential outside.

On one hand, if L_k and M_K are i.i.d. exponentially distributed random variables with mean α, $(U_k - V_k)/2$ becomes a zero mean *Laplace* distributed RV with variance $\alpha^2/2$. In this case, the ψ function is taken as

$$\psi^*(x) = \begin{cases} \text{sign}(x) & \text{for } |x| \leq k \\ 0 & \text{for } |x| > k \end{cases} \quad . \tag{3.5}$$

In the clock offset model,

$$\frac{U_k - V_k}{2} = \theta_A + \frac{L_k - M_k}{2} \quad .$$

This equation can be rewritten as

$$\frac{U_k - V_k}{2} - \theta_A = \frac{L_k - M_k}{2} \quad , \tag{3.6}$$

where $U_k = T_{2,k} - T_{1,k}$, $V_k = T_{4,k} - T_{3,k}$. θ_A denotes the clock offset between two nodes, and X_k and Y_k denote the variable portions of delays.

Therefore, the M-estimator of the location parameter can be applied to this model using the relationship

$$\sum_{k=1}^{n} \psi\left(\frac{U_k - V_k}{2} - \theta_A\right) = 0 \quad . \tag{3.7}$$

C. Simulation Results

We compared the performances of Huber M-estimators which use $\psi_H(\cdot)$ and $\psi^*(\cdot)$, respectively, under various delay models; i.e., zero-mean Gaussian, exponential, gamma,

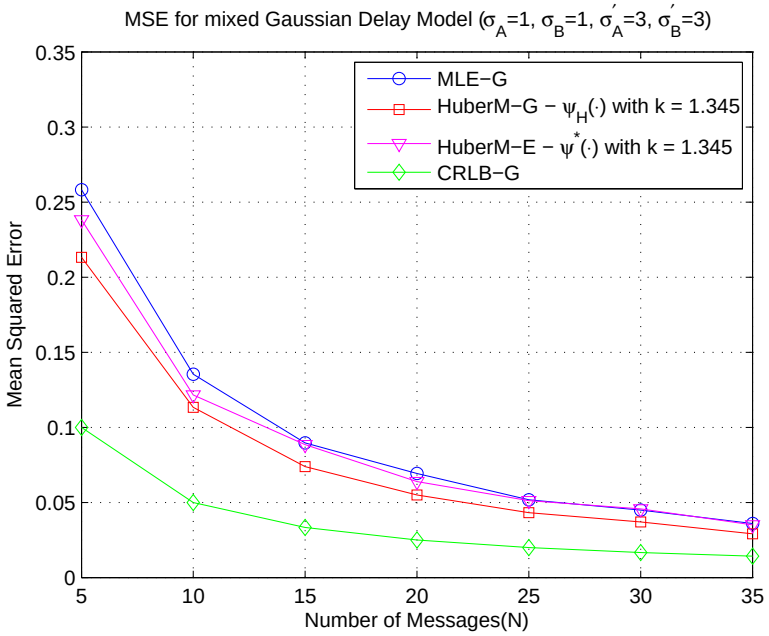

Fig. 3.1. MSEs of clock offset estimators for the Gaussian delay model with contaminated Gaussian delays

Weibull, and lognormal delay distributions, respectively, assuming that each distribution is mixed with or contaminated by another zero-mean Gaussian distribution. In this simulation, the rate of contamination or mixing is 20% and the cutoff value of Huber estimation, k, is equal to 1.345. In Figs. 3.1 - 3.5, MLE-E and MLE-exponential denote the MLE in the exponential delay model, and MLE-G and CRLB-G denote the MLE and the Cramer-Rao lower bound in the Gaussian delay model, respectively, when no contamination is assumed. In Fig. 3.1, σ_A^2 and σ_B^2 denote the delay variances for the uplink and the downlink delay distributions, respectively, and ${\sigma'_A}^2$ and ${\sigma'_B}^2$ represent the delay variances for the uplink and the downlink Gaussian distributions, which are mixed with the underlying Gaussian distributions. In Figs. 3.2 - 3.5, σ_A^2 and σ_B^2 indicate the delay variances for the uplink and the downlink Gaussian distributions, which contaminate the underlying distributions.

For the exponential delay model, the mean values (μ_A and μ_B) were chosen to be

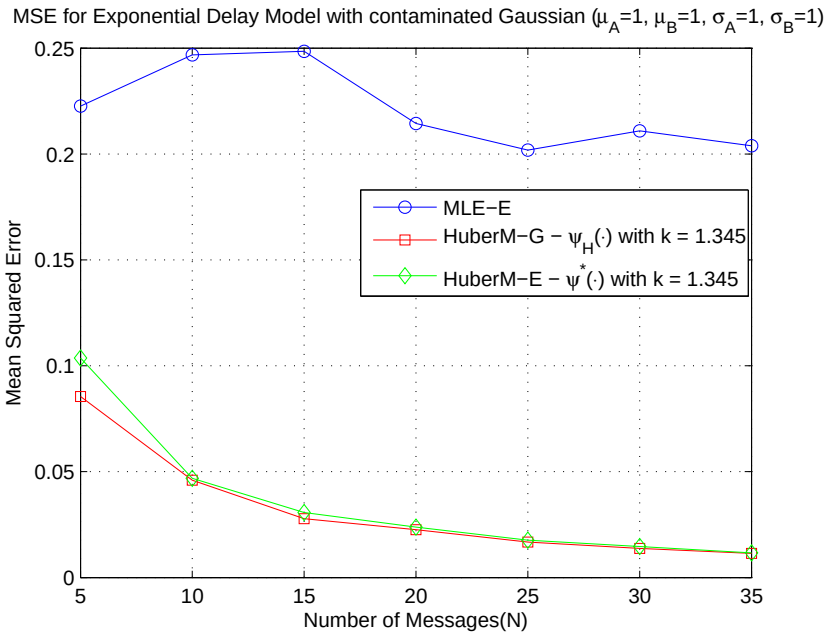

Fig. 3.2. MSEs of clock offset estimators for the exponential delay model with contaminated Gaussian delays

one. In the gamma delay model, the shape parameter (α) and the scale parameter (β) were taken to be two and one, respectively. For the Weibull case, the shape parameter (k) and the scale parameter (λ) were selected to be two. In the lognormal case, the mean and the variance were chosen to be one. The gamma, Weibull, and lognormal families are considered as representatives to appropriately cover a wide range of typical network delay distributions. Figs. 3.1 - 3.5 consistently show that the clock estimators based on the robust M-estimation method present excellent MSE performances in the presence of exponential distributions as well as in Gaussian distributions, which are mixed with another Gaussian delay distribution. In other words, the proposed estimators are capable of producing good performances in unknown contaminated delay distributions.

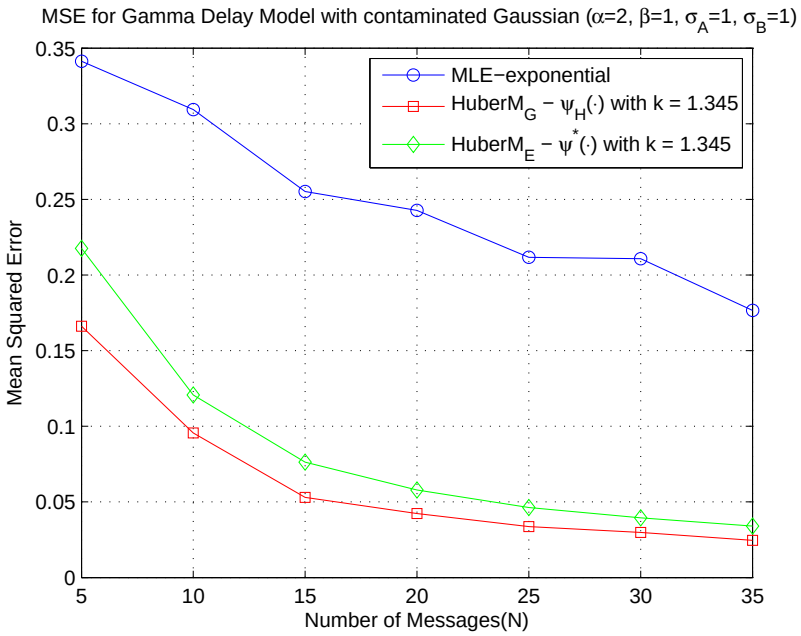

Fig. 3.3. MSEs of clock offset estimators for the gamma delay model with contaminated Gaussian delays

D. Summary

Clock offset estimators based on the robust M-estimation method are described in a clock synchronization protocol involving a two-way message exchange model. It is clearly illustrated that the robust M-estimation based clock offset estimators present good MSE performance under several delay distributions that assume contaminated distributions.

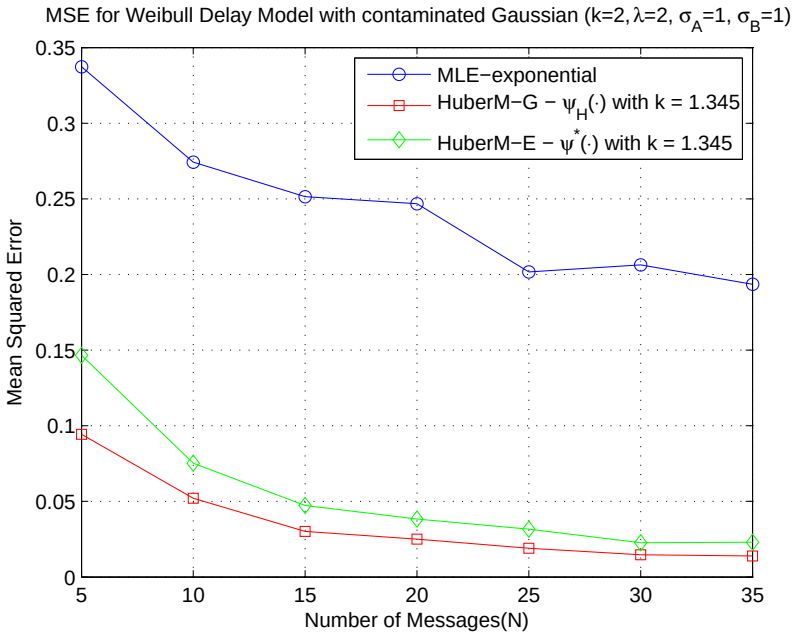

Fig. 3.4. MSEs of clock offset estimators for the Weibull delay model with contaminated Gaussian delays

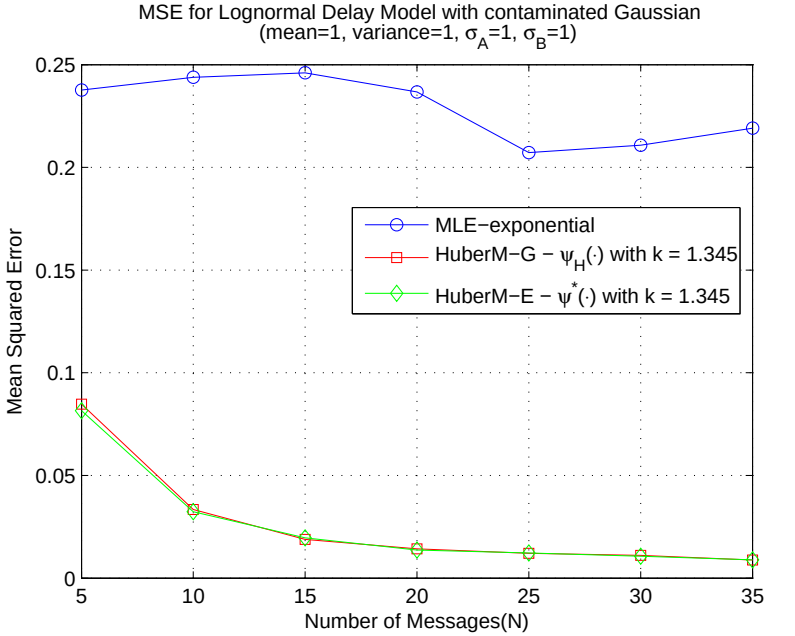

Fig. 3.5. MSEs of clock offset estimators for the lognormal delay model with contaminated Gaussian delays

CHAPTER IV

A ROBUST ESTIMATOR FOR CLOCK PHASE OFFSET IN THE PRESENCE OF NON-GAUSSIAN RANDOM DELAYS

Now we turn our attention to the clock offset estimation methods based on the particle filtering (PF) technique in order to obtain more accurate estimators of clock offset. This chapter proposes a novel clock offset estimation method, called the Gaussian Mixture Kalman Particle Filter (GMKPF), which is robust and yields very accurate clock offset estimates in the presence of arbitrary network delay distributions. The clock offset estimation framework assumes the two-way message exchange mechanism between two nodes, a set-up encountered in NTP [15] and TPSN protocol [18]. GMKPF combines the importance sampling (IS) based measurement update step with a KF (Kalman Filter) based Gaussian sum filter for the time-update and proposal density generation. Since GMKPF employs new observations and exploits the Expectation-Maximization (EM) algorithm to obtain the Gaussian Mixture Model (GMM), GMKPF is expected to exhibit better estimation performance when compared to symmetric Gaussian MLE (SGML) and symmetric exponential MLE (SEML) in general non-Gaussian/non-exponential delay models. Thus far, in the synchronization literature for WSNs, it appears that only very few preliminary and straightforward applications of standard Kalman filtering or general adaptive signal processing techniques were reported (see [36], [37] and [38]) to improve the mean square error (MSE) performance of protocols such as RBS [17] or TPSN [18].

In this chapter, upon designing the GMKPF, a thorough performance analysis of GMKPF, GML, and EML in the presence of the two-way message exchange mechanism between two nodes and symmetric/asymmetric Gaussian, exponential, Gamma, Weibull network delay distributions is first carried out. The performance of GMKPF,

SGML, and SEML is also simulated under the mixing of two different distributions: Gaussian and exponential, Gaussian and Gamma, Gaussian and Weibull, exponential and Gamma, exponential and Weibull, and Gamma and Weibull delay distributions, respectively. The computer simulation results corroborate the superior performance of the proposed method relative to SGML and SEML, and its robustness to general network delay distributions. Therefore, the proposed GMKPF method represents a high-performance and very reliable clock offset estimation scheme fit to overcome the uncertainties caused by the network delay distributions.

A. Problem Formulation

Similarly to [24], the differences between the kth up and down-link delay observations corresponding to the kth timing message exchange are given 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 might assume any distribution such as Gaussian, exponential, Gamma, Weibull or mixtures of two different distributions.

Given the observation samples $\mathbf{z}_k = [U_k, V_k]^T$, our goal is to find the minimum MSE 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 are looking to determine the estimator:

$$\hat{x_k} = E\{x_k|\mathbf{z}^l\} \ , \tag{4.1}$$

where $\mathbf{z}^l$ denotes the set of observed samples up to time l, $\mathbf{z}^l = \{\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_l\}$. Since

the clock offset value is assumed constant, the clock offset can be modeled as obeying a Gauss-Markov dynamic channel model of the form:

$$x_k = Fx_{k-1} + v_{k-1} \ , \tag{4.2}$$

where F is the state transition matrix for clock offset. The additive noise component v_k can be modeled as Gaussian with zero mean and covariance $Ev_k v_k^T = Q$. The vector observation model follows from the observed samples and it assumes the expression:

$$\mathbf{z}_k = \begin{bmatrix} d + x_k + L_k \\ d - x_k + M_k \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} d + \begin{bmatrix} 1 \\ -1 \end{bmatrix} x_k + \mathbf{n}_k \ , \tag{4.3}$$

where the observation noise vector $\mathbf{n}_k = [L_k, M_k]^T$ accounts for the random delays. One can now observe that (4.2) and (4.3) recast our initial clock offset estimation problem into a Gauss-Markov estimation problem with unknown states.

B. The GMKPF Method

Particle filtering is a sequential Monte Carlo sampling method built within the Bayesian paradigm. From a Bayesian perspective, at time k, the posterior distribution $p(x_k|\mathbf{z}_{0:k})$ is the main entity of interest. However, due to the non-Gaussianity of the model (4.3), the analytical expression of $p(x_k|\mathbf{z}_{0:k})$ cannot be obtained in closed-form expression, excepting for some special cases like Gaussian or exponential pdfs. Alternatively, particle filtering can be applied to approximate $p(x_k|\mathbf{z}_{0:k})$ by stochastic samples generated using a sequential importance sampling strategy.

Since the particle filtering with the prior importance function employs no information from observations in proposing new samples, its use is often ineffective and leads to poor filtering performance. Herein, we implement a slightly changed version of the Gaussian Mixture Sigma Point Particle Filter (GMSPPF) proposed in [39], and

which will be referred to as a composite approach. This composite approach comes out from the utilization of another filtering technique producing a filtering probability density function used as importance function (IF) for the particle filtering.

The GMSPPF is a family of methodologies that use hybrid sequential Monte Carlo simulation and a Gaussian sum filter to efficiently estimate posterior distributions of unknown states in a non-linear dynamic system. However, in our state space modeling, because of the linear model, we do modify this method further. Following [39], we will next describe briefly the general framework assumed by the GMKPF method, obtained by replacing the SPKF with a KF. We next outline the main features of the proposed approach. First, we remark that any probability density $p(x)$ can be approximated as closely as desired by a Gaussian mixture model (GMM) of the following form [40],

$$p(x) \approx p_g(x) = \sum_{g=1}^{G} \alpha^{(g)} \mathrm{N}(x; \mu^{(g)}, P^{(g)}) \ , \tag{4.4}$$

where G stands for the number of mixing components, $\alpha^{(g)}$ denote the mixing weights and $\mathrm{N}(x; \mu, P)$ is a normal distribution with mean μ and covariance P. Thus, the predicted and updated Gaussian components, i.e., the means and covariances of the involved probability densities (posterior, importance, and so on) are calculated using the Kalman filter (KF) instead of the Sigma Point Kalman Filter (SPKF) [39], [41]. Since the state and observation equations are linear, the KF was employed instead of the SPKF. Therefore, the resulting approach is called the Gaussian mixture Kalman particle filter (GMKPF). In order to avoid the particle depletion problem in cases where the observation (measurement) likelihood is very peaked, the GMKPF represents the posterior density by a GMM which is recovered from the re-sampled equally weighted particle set using the Expectation-Maximization (EM) algorithm.

In general for the particle filtering approach, the posterior density $p(x_{0:k}|\mathbf{z}_{1:k})$, where $x_{0:k} = \{x_0, \cdots, x_k\}$ and $\mathbf{z}_{1:k} = \{\mathbf{z}_1, \cdots, \mathbf{z}_k\}$, constitutes the complete solution to the sequential estimation problem. Our objective is to generate samples from the distribution $p(x_{0:k}|\mathbf{z}_{1:k})$. For this purpose, we have collected N sets of samples $x_{0:k}^{(i)} = \{x_0^{(i)}, \cdots, x_k^{(i)}\}$ with weights $w_k^{(i)}, i = 1, ..., N$. The particles $\{x_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ approximate $p(x_{0:k}|\mathbf{z}_{1:k})$. Finally, the conditional mean state and the corresponding error covariance can be calculated:

$$\bar{x}_k = \sum_{i=1}^{N} w_k^{(i)} x_k^{(i)}, \qquad \Phi = \sum_{i=1}^{N} w_k^{(i)} [\bar{x}_k - x_k^{(i)}][\bar{x}_k - x_k^{(i)}]^T. \tag{4.5}$$

At the end of each recursion, the particles are resampled to ensure they occur with the same probability as the weights.

The GMKPF combines the importance sampling (IS) based measurement update step with a KF based Gaussian sum filter for the time-update and proposal density generation. In the time update stage, GMKPF approximates the prior, proposal and posterior density function as GMMs using banks of parallel KFs. The updated mean and covariance of each mixand follow from the KF updates. In the measurement update stage, the GMKPF uses a finite GMM representation of the posterior filtering density

$$p_g(x_k|\mathbf{z}_k) = \sum_{g=1}^{G} \alpha_k^{(g)} N(x_k; \mu_k^{(g)}, P_k^{(g)}) \,, \tag{4.6}$$

where G is the number of GMMs, $\alpha_l^{(g)}$ are the mixing weights and $N(x_k; \mu_k^{(g)}, P_k^{(g)})$ is a normal distribution determined from the gth KF with predicted mean $\mu_k^{(g)} = \bar{x}_k$ and positive definite covariance $P_k^{(g)}$. This is recovered from the weighted posterior particle set of the IS based measurement update stage, by means of an Expectation-Maximization (EM) [42] step. The EM algorithm can be used to obtain Gaussian Mixture approximations from these particles and weights. Through this mechanism, the

EM-based posterior GMM further mitigates the "sample depletion" problem through its inherent "kernel smoothing" nature. The EM algorithm provides an iterative method to estimate $\bar{\boldsymbol{\theta}}$ via

$$\bar{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} \; p(\mathbf{z}|\boldsymbol{\theta}) \; , \tag{4.7}$$

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 works as follows: given a $\boldsymbol{\theta}^{(j)}$, 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 reader is directed to reference [42] for more detailed explanations of the EM algorithm for GMM. Finally, the conditional mean state estimate and the corresponding error covariance can be calculated as follows:

$$\bar{x}_k \;\; = \;\; \sum_{g=1}^{G} \alpha_k^{(g)} \mu_l^{(g)}, \quad \bar{P}_k = \sum_{g=1}^{G} \alpha_k^{(g)} [P_k^{(g)} + (\mu_k^{(g)} - \bar{x}_k)(\mu_k^{(g)} - \bar{x}_k)^T]. \tag{4.8}$$

Below we provide a pseudo-code for a GMKPF algorithm.

GMKPF Algorithm Pseudo-Code:

1. At time $k-1$, initialize the densities
   - The posterior density is approximated by
     $p_g(x_{k-1}|\mathbf{z}_{k-1}) = \sum_{g=1}^{G} \alpha_{k-1}^{(g)} \mathrm{N}(x_{k-1}; \mu_{k-1}^{(g)}, P_{k-1}^{(g)})$.
   - The process noise density is approximated by
     $p_g(v_{k-1}) = \sum_{i=1}^{I} \beta_{k-1}^{(i)} \mathrm{N}(v_{k-1}; \mu_{v_{k-1}}^{(i)}, Q_{k-1}^{(i)})$.
   - The observation noise density is approximated by
     $p_g(\mathbf{n}_k) = \sum_{j=1}^{J} \gamma_k^{(J)} \mathrm{N}(\mathbf{n}_k; \mu_{\mathbf{n_k}}{}^{(j)}, \mathbf{R}_k^{(j)})$.

2. Pre-prediction step

   - Calculate the pre-predictive state density $\tilde{p}_g(x_k|\mathbf{z}_{k-1})$ using KF.
   - Calculate the pre-posterior 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: $\tilde{w}_k^{(l)} = \frac{p(\mathbf{z}_k|\chi_k^{(l)})\hat{p}_g(\chi_k^{(l)}|\mathbf{z}_{k-1})}{\hat{p}_g(\chi_k^{(l)}|\mathbf{z}_k)}$
   - Normalize the weights: $w_k^{(l)} = \tilde{w}_k^{(l)} / \sum_{l=1}^{N} \tilde{w}_k^{(l)}$
   - 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 (4.8).

For computational complexity comparisons, it is difficult to assess exactly the number of floating point operations (flops) involved in GMKPF due to its iterative and complex operations involved. Therefore, we will only quantify the most computationally demanding steps in all algorithms using the Big O notation. Let L denote the

dimension of the state vector x ($L = 1$). Assume also that N represents the number of particles, and G is the number of GMM components. Thus, $x \in R^{1\times 1}$, $F \in R^{1\times 1}$, $P \in R^{1\times 1}$. The Kalman Filter is approximately $O(L^3)$ due to the fact that the matrix times matrix multiplication is the most time consuming step ($P_{k+1|k} = FP_k + Q$), whereas the particle filter is approximately $O(NL^2)$ because of the matrix times vector multiplication operation ($\bar{x}^{(i)}_{k+1|k} = F\bar{x}^{(i)}_k, i = 1, 2, \cdots, N$) and sampling step. The Gaussian Mixture Kalman Particle Filter requires approximately $O(GL^3)$ flops due to the KF step and $O(GL^2N)$ flops due to the EM step. If $N \gg L$, GMKPF is approximately $O(GL^2N)$. Notice also that SGML is approximately $O(L)$. This indicates that GMKPF is approximately 300 times slower than GML in an application with $L = 1$, $N = 100$, and $G = 3$.

C. Simulation Results

In this section, computer simulation results are presented to assess the performance of GMKPF, SGML [24], and SEML [24]-approaches for estimating the clock offset in wireless sensor networks. We consider a total of 10 delay models: asymmetric Gaussian, exponential, Gamma, Weibull, and mixtures of Gaussian and exponential, Gaussian and Gamma, Gaussian and Weibull, exponential and Gamma, exponential and Weibull, and Gamma and Weibull. The reason for this study is to illustrate that the proposed method is robust, exhibits superior performance and can be applied to deal with any delay distribution. The stationary process v_k exhibits a given constant variance $Q = 10^{-4}$. The number of particles and GMMs are 100 and 3, respectively. One aspect about using GMKPF is that it requires proper initialization. Depending on the problem, the initial guesses may need to be close to the correct value for convergence. However, this initialization problem is easily solved. ML-estimator for

symmetric Gaussian/exponential random delays provide good estimates. This initial value ($\hat{x}_0 = \hat{x}_{SGML}$) is near the true value. The GMKPF is used to start the tracking algorithm fairly close to the true values. The convergence of GMKPF is achieved after a number of iterations equal to the number of measurements.

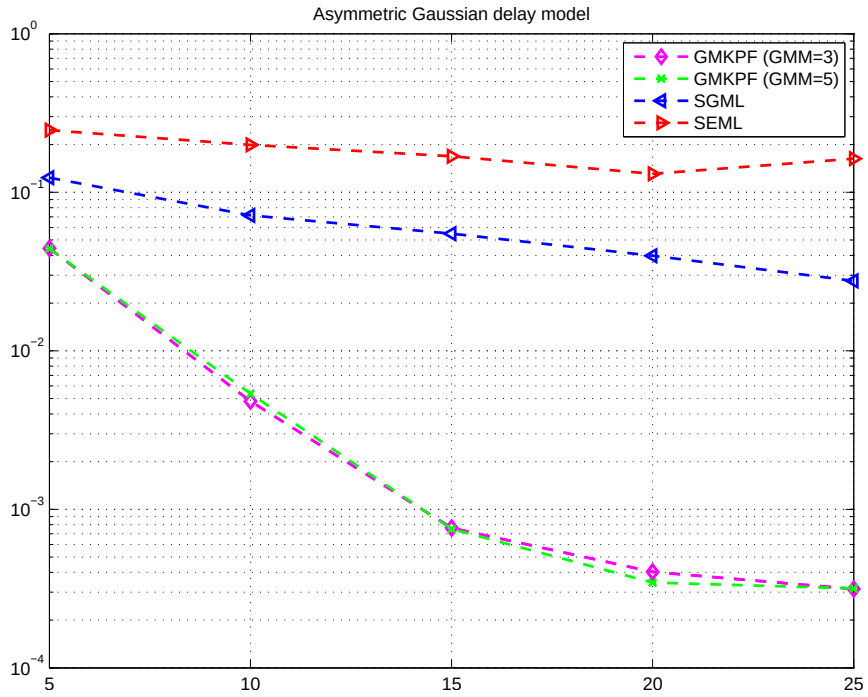

Fig. 4.1. MSEs of clock offset estimators for asymmetric Gaussian random delays ($\sigma_1 = 1, \sigma_2 = 4$)

Figs. 4.1 - 4.4 show the MSE of the estimators assuming that the random delay models are asymmetric Gaussian, exponential, Gamma, Weibull pdfs, respectively. The subscripts 1 and 2 are used to differentiate the parameters of delay distributions corresponding to uplink and downlink, respectively. As an example, the parameters σ_1 and σ_2 in Fig. 4.1 denote the standard deviations of uplink and downlink asymmetric Gaussian network delay densities, respectively. The MSEs are plotted against the number of observations, ranging from 5 to 25. Note that the GMKPF ($G = 3$ and $G = 5$) performs much better (a reduction of MSE with over 100%) when compared

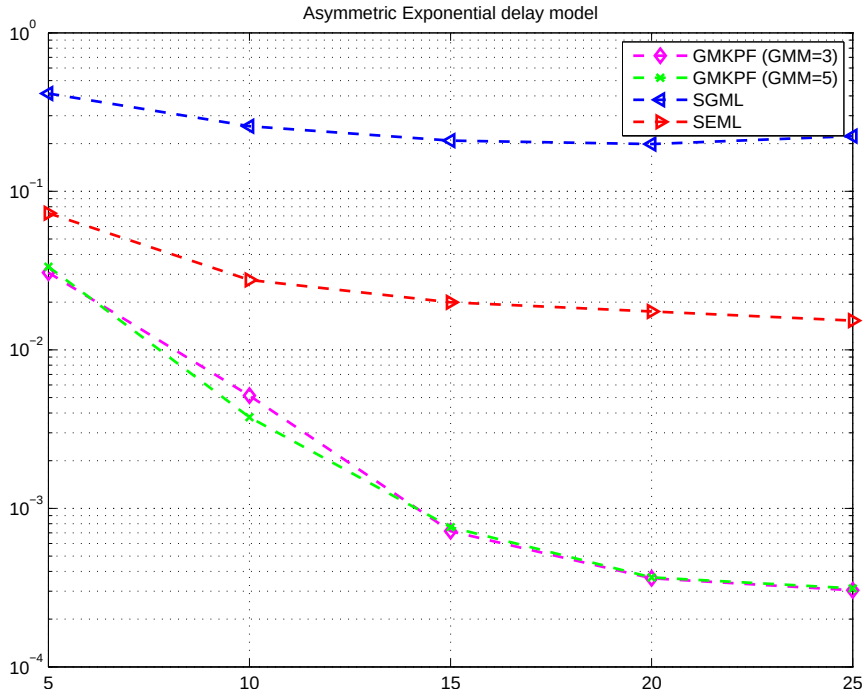

Fig. 4.2. MSEs of clock offset estimators for asymmetric exponential random delays ($\lambda_1 = 1, \lambda_2 = 5$)

to SGML or SEML. Two different GMKPF ($G = 3$ and $G = 5$) were compared. The first GMKPF estimator ($G = 3$) uses a 3-component GMM for the state posterior, a 1-component GMM for the process noise density, and a 3-component GMM for the measurement noise density. The second GMKPF estimator ($G = 5$) uses a 3-, 1-, 5-component GMM for the state posterior, process noise density and measurement noise density, respectively. In this case, a 3 and 5-component GMM is used to approximate the non-Gaussian distribution of measurement noise. GMKPF ($G = 5$) is slightly better than most of results except for Figs. 4.4 and 4.5. The GMKPF-estimator did adaptively fit the posterior probability function (likelihood function) using the EM, so the performance of GMKPF depends on the ability of GMM to capture the distributions of process noise, measurement noise, posterior PDFs, and so on. As shown in the Figs. 4.4 and 4.5, when the GMKPF has more components included the

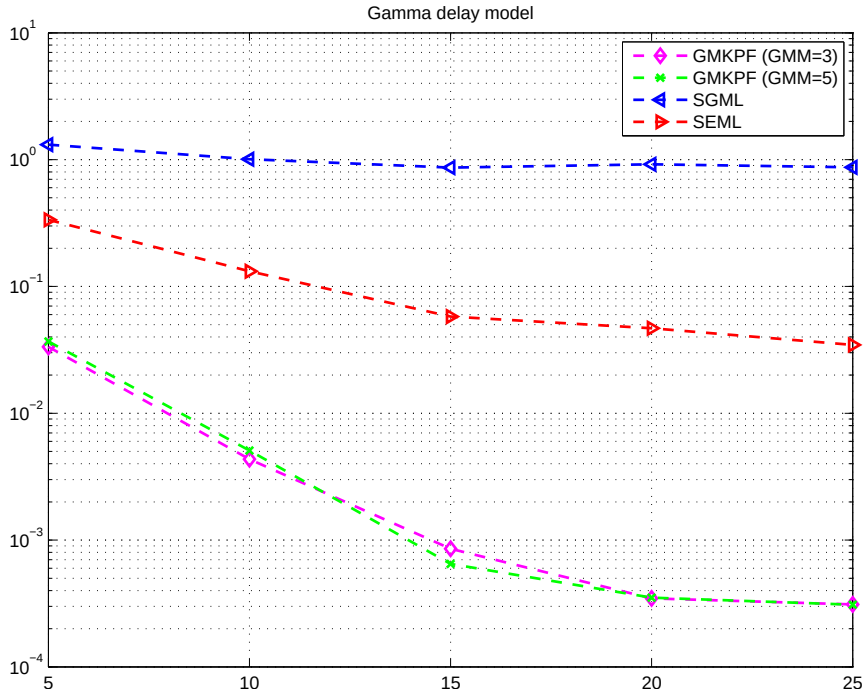

Fig. 4.3. MSEs of clock offset estimators for Gamma random delays ($\alpha_1 = 2, \beta_1 = 1$)

GMM, its performance is not always better than in the case when a reduced number of components are present in GMM. However, in general, the performance of GMKPF is better when a larger number of components are used in GMM.

It is interesting to note that the MSE of SGML exhibits better performance than SEML in the asymmetric Gaussian delay model case and poorer performance in the presence of asymmetric exponential, Gamma, and Weibull delay models. The reason for this is that Gamma and Weibull delay models are closer to the exponential distribution than the Gaussian distribution.

To further quantify the robustness of the estimators, we studied the performance of the GMKPF, GML, and EML under various network delay conditions, where the random delay models are mixtures of two distributions. For example, in Fig. 4.5, we mix equally a Gaussian with an exponential delay model, each having a weight of 50%. This means that if 10 observations are received, 5 observations are Gaussian

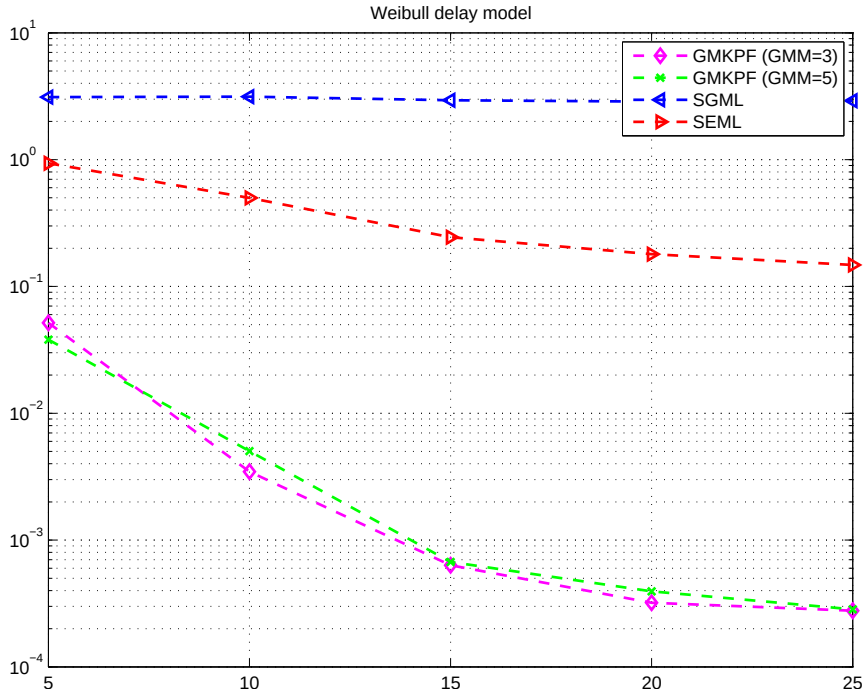

Fig. 4.4. MSEs of clock offset estimators for Weibull random delays ($\alpha_1 = 2, \beta_1 = 2$) and ($\alpha_2 = 6, \beta_2 = 2$)

and the remaining 5 samples assume an exponential distribution. From Figs. 4.5 - 4.10, we observe that GMKPF clearly outperforms the SGML and SEML. In these cases, the SGML presents better performance than SEML if the network delay process is closer to a Gaussian. Otherwise, the SEML exhibits better performance than the SGML, while GMKPF outperforms both the SGML and SEML.

D. Summary

This chapter presented a novel method for estimating the clock offset in wireless sensor networks. The benefits of the proposed synchronization method are in terms of improved performance and applicability to arbitrary random delay models such as asymmetric Gaussian, asymmetric exponential, Gamma, Weibull, as well as to mixtures of these delay models. One negative aspect is the fact that analytical closed

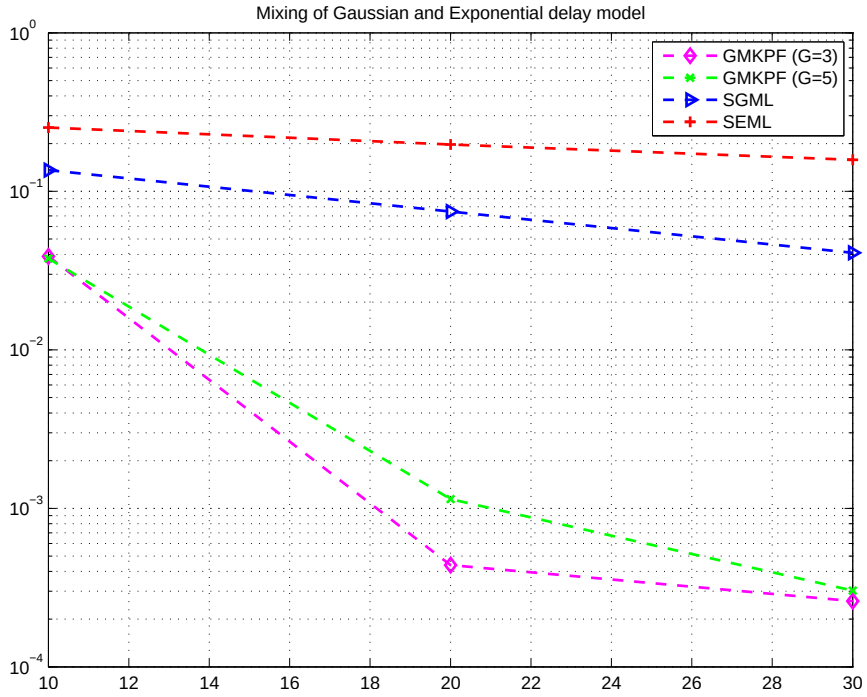

Fig. 4.5. MSEs of clock offset estimators for mixing of a Gaussian $[\sigma_1 = 1, \sigma_2 = 1]$ and an exponential $[\lambda_1 = 1, \lambda_2 = 5]$

form expressions do not necessarily exist and in general it is hard to derive lower bounds in the presence of (unknown) non-Gaussian distributions. The chapter presented a robust estimator based on the GMKPF that is capable of estimating the clock offset in arbitrary delay models, a result which might present applications in numerous wireless sensor networks applications with tight synchronization requirements [43], [44]. Computer simulations also show that the proposed method yields superior performance.

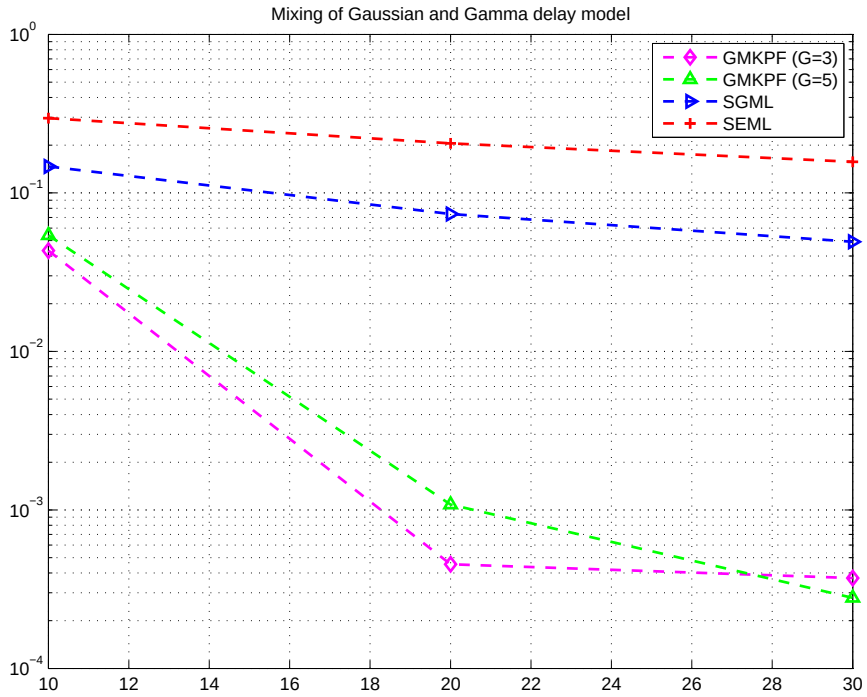

Fig. 4.6. MSEs of clock offset estimators for mixing a Gaussian $[\sigma_1 = 1, \sigma_2 = 1]$ and a Gamma $[\alpha_1 = 2, \beta_1 = 2]$

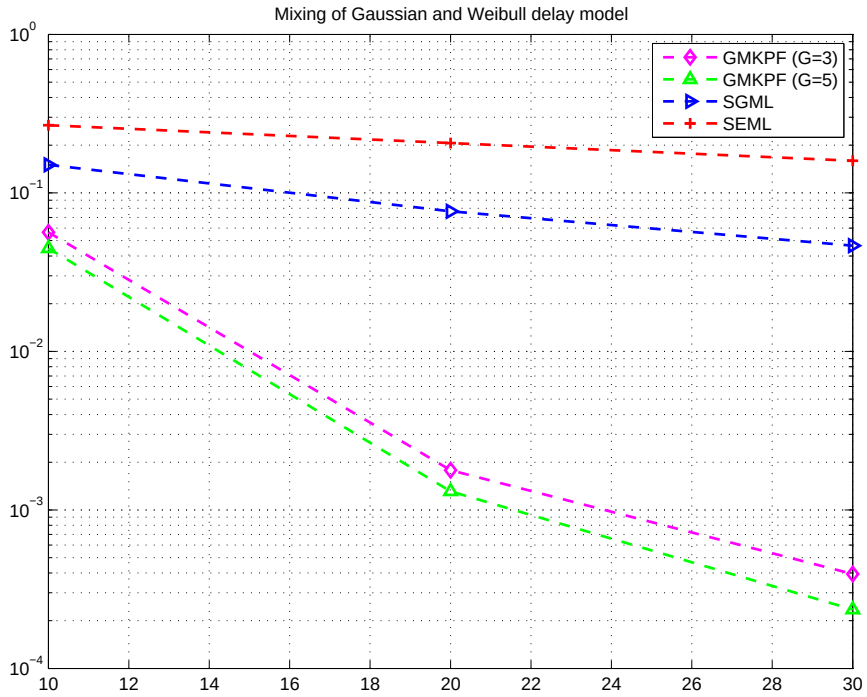

Fig. 4.7. MSEs of clock offset estimators for mixing a Gaussian $[\sigma_1 = 1, \sigma_2 = 1]$ and a Weibull random delay $[\alpha_1 = 2, \beta_1 = 2$ and $\alpha_2 = 6, \beta_2 = 2]$

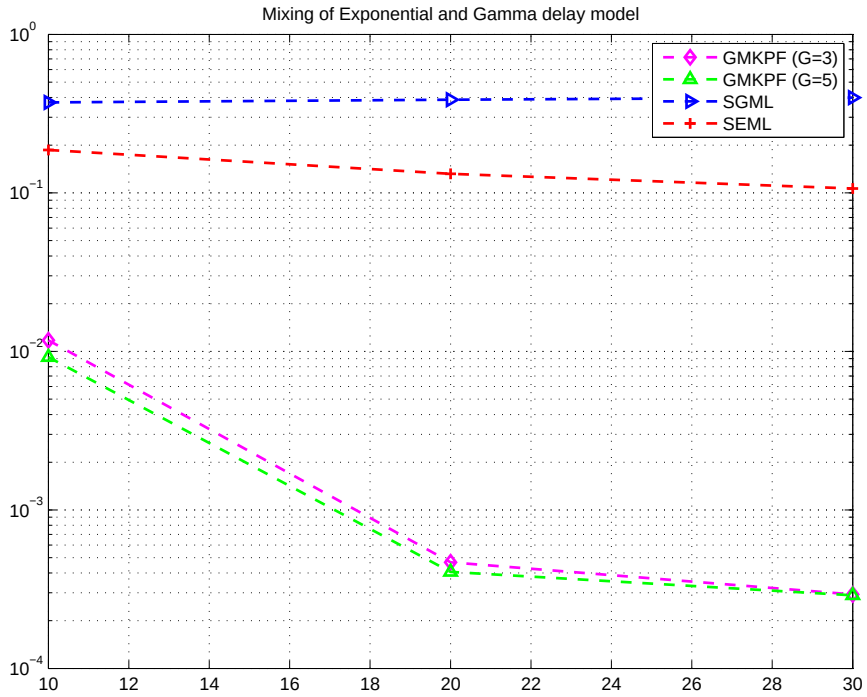

Fig. 4.8. MSEs of clock offset estimators for mixing an exponential $[\lambda_1 = 1, \lambda_2 = 5]$ and a Gamma $[\alpha_1 = 2, \beta_1 = 5]$ and $[\alpha_2 = 2, \beta_2 = 2]$

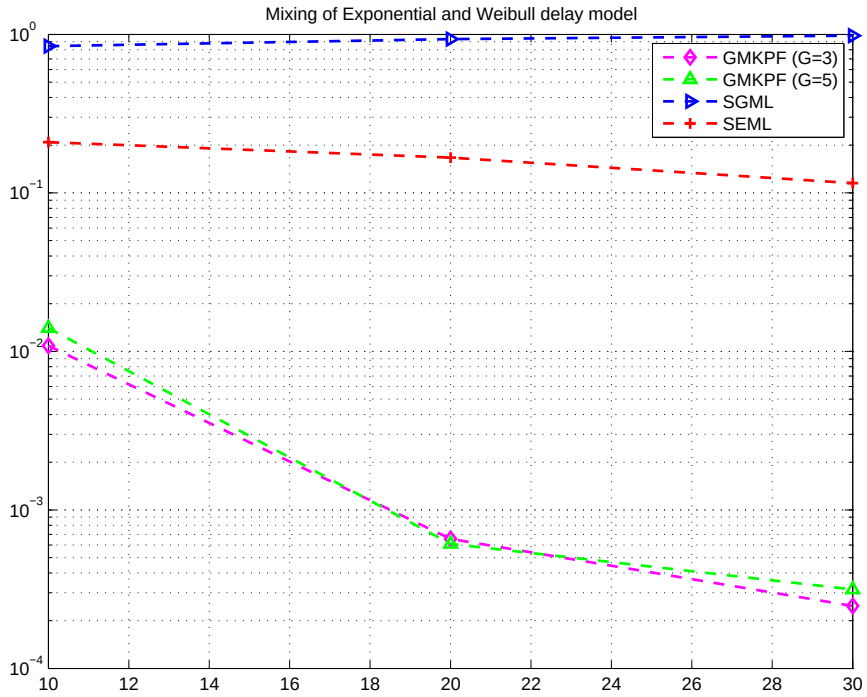

Fig. 4.9. MSEs of clock offset estimators for mixing an exponential $[\lambda_1 = 1, \lambda_2 = 5]$ and a Weibull $[\alpha_1 = 2, \beta_1 = 2$ and $\alpha_2 = 6, \beta_2 = 2]$

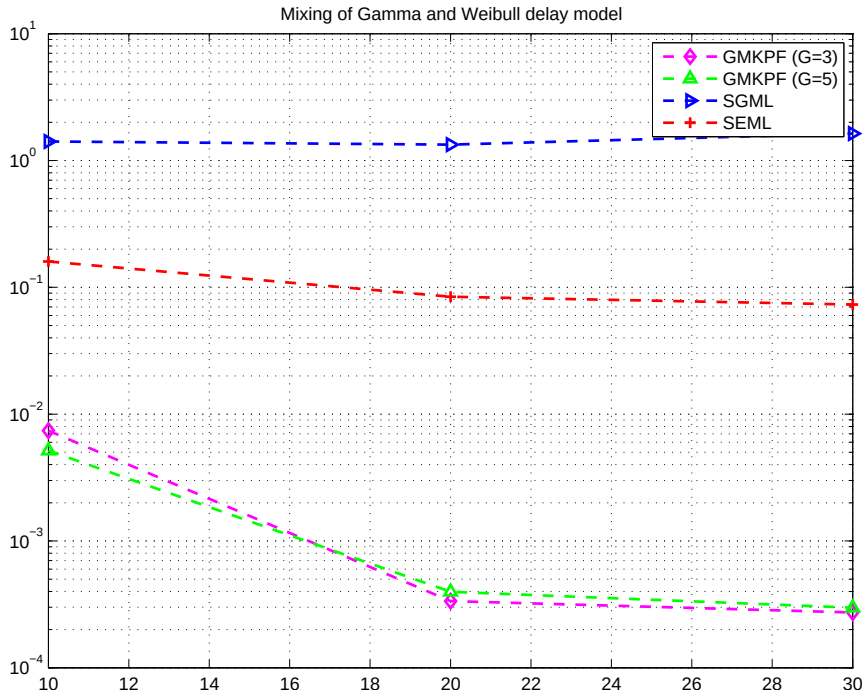

Fig. 4.10. MSEs of clock offset estimators for mixing a Gamma with parameters $[\alpha_1 = 2, \beta_1 = 5]$ and $[\alpha_2 = 2, \beta_2 = 2]$ and a Weibull $[\alpha_1 = 2, \beta_1 = 2$ and $\alpha_2 = 6, \beta_2 = 2]$

CHAPTER V

A ROBUST COMPOSITE PARTICLE FILTER APPROACH FOR CLOCK OFFSET ESTIMATION

This chapter discusses a clock offset estimator based on the composite particle filter (CPF) to cope with the possible asymmetries and non-Gaussianity of the network delay distributions. The CPF approximates the posterior and predictive distributions by weighted Gaussian mixtures and is basically implemented via banks of Kalman filters (KFs). The CPF can be alternatively viewed as a composition between the Gaussian sum particle filter (GSPF) [27] and the KF. In addition, a variant of the CPF approach based on the bootstrap sampling (BS) is shown to exhibit good performance in the presence of reduced number of observations. The idea behind the CPF with BS is to generate sampled observation data from the original observation data by using the BS, and then to estimate the clock offset using the CPF.

In [27], the inference of general state-space models characterized by nonlinear process and observation equations is addressed via the concept of Gaussian sum particle filter (GSPF) which approximates the filtering and predictive distributions by weighted Gaussian mixtures, i.e., banks of Gaussian particle filters (GPFs). With non-Gaussian noise approximated by Gaussian mixtures, the non-Gaussian noise models are approximated by banks of Gaussian noise models. However, in wireless sensor networks, the process and observation equations are linear functions. Therefore, we extend the use of a new filter, the composite particle filter (CPF), to encompass linear and additive non-Gaussian noise models. For a linear state-space model with additive non-Gaussian noise, CPF approximates the posterior distributions as Gaussian mixtures using banks of parallel Kalman filters (KFs). The main contribution of this chapter is a novel clock offset estimation method, called the composite

particle filter (CPF), which is shown to be robust to the unknown distribution of network delays.

A. Problem Modeling and Objectives

The problem model is the same as the one described in Chapter IV. Following [24, **?**], the time differences corresponding to the kth up and down link delay observations of the kth timing message exchange can be expressed respectively as $U_k = T_{2,k} - T_{1,k} = d + \theta_A + N_k$ and $V_k = T_{4,k} - T_{3,k} = d - \theta_A + M_k$, respectively. The fixed value θ_A denotes the clock offset between the two nodes, d stands for the (deterministic) propagation delay, and N_k and M_k model the variable portions of delay, and might assume any distribution such as Gaussian, exponential, Gamma, Weibull or a mixture of two distributions. For notational convenience, we adopt the notation $x_k = \theta_A$ to denote the unknown clock phase offset. Given the observation samples $\mathbf{z}_k = [U_k, V_k]^T$, our goal is to find the minimum mean-square estimator of the unknown clock offset θ_A, which is given by

$$\hat{x_k} = E\{x_k|\mathbf{Z}^l\} \ , \tag{5.1}$$

where $\mathbf{Z}^l$ denotes the set of observed samples up to time l, $\mathbf{Z}^l = \{\mathbf{z}_0, \mathbf{z}_1, ..., \mathbf{z}_l\}$. Since the clock offset value is constant, the clock offset is assumed to obey a Gauss-Markov dynamic channel model of the form:

$$x_{k+1} = Fx_k + v_k \ , \tag{5.2}$$

where F represents the state transition scalar value (if x_k is vector, F is matrix) for the clock offset. The noise vector v_k is modeled as a Gaussian random variable with zero mean and covariance $Ev_kv_k^T = Q$. Notice also that the vector observation model

follows from the observed data and takes the expression:

$$\mathbf{z}_k = \begin{bmatrix} U_k \\ V_k \end{bmatrix} = \begin{bmatrix} d + x_k + N_k \\ d - x_k + M_k \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} d + \begin{bmatrix} 1 \\ -1 \end{bmatrix} x_k + \mathbf{n}_k \ , \tag{5.3}$$

where the observation noise vector $\mathbf{n}_k = [N_k, M_k]^T$ may assume any probability density function (pdf). Hence, it turns out that our initial problem is now casted as the estimation problem of a Gauss-Markov model with unknown state (see (5.2) and (5.3)).

B. Composite Particle Filtering and Bootstrap Sampling Approach

The maximum likelihood clock offset estimator was reported in [24] for the two-way timing message exchange protocol such as TPSN and NTP under the assumption of Gaussian or exponential delay models. Herein we will derive a CPF for clock phase offset estimation assuming a general unknown distribution of network delays, and then compare the CPF and existing maximum likelihood clock offset estimators that were derived for Gaussian (GML) and exponential (EML) delay models. Under the Bayesian framework, an emerging powerful technique for obtaining the posterior, predictive, and filtering probability density functions is referred to as the particle filtering (PF) (see e.g., [45], [46]). The PF technique allows for a complete representation of the state posterior distribution, which approximates $p(x_k|\mathbf{z}_{0:k})$, by stochastic samples generated using a sequential importance sampling strategy. The most common employed PF strategy is to sample from the transition prior distribution due to its simplicity. Since the prior importance sampling distribution employs no information from observations in proposing new samples, its use is often ineffective and leads to poor filtering performance. To overcome these challenges, we will derive an extension of GSPF [27] applicable for linear non-Gaussian models.

In the GSPF, the filtering and predictive distributions are recursively represented as finite Gaussian mixtures (GMs) using Gaussian Particle Filtering (GPF) [26]. One set of methods approximates the mixture components of the predictive and filtering distributions as Gaussian. The approximation can be implemented by the GSPF, resulting in a parallel bank of GPFs. Since Gaussian mixture models are increasingly used for modeling non-Gaussian densities [**?**, **?**], herein we plan extending the use of the GSPF to linear non-Gaussian models. The resulting new approach will be referred to as CPF. Notice that in the measurement and time-update equations of CPF, the updated mean and covariance of each mixand follow from the KF. The CPF is implemented by means of G parallel KFs, and the weights are adjusted according to the given update equations. Notice also that the CPF approach comes out of the utilization of another filtering technique (KF) producing a filtering probability density function used as importance function (IF) for the particle filtering.

We next describe briefly the general framework of CPF methods. Assume at time $k-1$, the posterior distribution $p(x_{k-1}|\mathbf{z}_{0:k-1})$ is approximated as closely as desired by a Gaussian mixture model (GMM) of the following form [40]

$$p(x_{k-1}|\mathbf{z}_{0:k-1}) \approx \sum_{g=1}^{G} w_{(k-1)g} \mathrm{N}(x_{k-1}; \mu_{(k-1)g}, P_{(k-1)g}), \tag{5.4}$$

where G stands for the number of mixing components, $w_{(\cdot)g}$ denote the mixing weights, and $\mathrm{N}(x; \mu, P)$ is the normal distribution of RV x with mean μ and covariance P. The transition prior is modeled as

$$p(x_k|x_{k-1}) = \sum_{l=1}^{L} \alpha_{(k)l} \mathrm{N}(x_k; Fx_{k-1} + \bar{\mu}_{(k-1)l}, \bar{P}_{(k-1)l}). \tag{5.5}$$

where $\alpha_{(k)l}$ denote the mixing weights, L is the number of Gaussian mixing components for modeling the transition prior. The time update stage that the previous

observations and state are used to predict the current state is encompassed by the predictive distribution which can be approximated as follows

$$\begin{aligned} p(x_k|\mathbf{z}_{1:k-1}) &= \int p(x_k|x_{k-1})p(x_{k-1}|\mathbf{z}_{1:k-1})dx_{k-1} \\ &= \int p(x_k|x_{k-1}) \sum_{g=1}^{G} w_{(k-1)g}\mathrm{N}(x_{k-1};\mu_{(k-1)g},P_{(k-1)g})dx_{k-1} \\ &= \int \sum_{l=1}^{L} \alpha_{(k)l}\mathrm{N}(x_k;Fx_{k-1}+\bar{\mu}_{(k-1)l},\bar{P}_{(k-1)l}) \cdot \\ &\cdot \sum_{g=1}^{G} w_{(k-1)g}\mathrm{N}(x_{k-1};\mu_{(k-1)g},P_{(k-1)g})dx_{k-1}. \end{aligned} \tag{5.6}$$

The predicted and updated Gaussian component means and covariances are calculated using the KF, where the integral in (5.6) is approximated by a Gaussian. Then, the predictive distribution can be approximated as

$$p(x_k|\mathbf{z}_{1:k-1}) \approx \sum_{j=1}^{GL} \tilde{w}_{kg}\mathrm{N}(x_k;\tilde{\mu}_{(k)j},\tilde{P}_{(k)j}), \tag{5.7}$$

where the parameters of the mixture are obtained according to KF:

$$\tilde{\mu}_{kj} = \bar{\mu}_{(k-1)l} + F\mu_{(k-1)g}, \tag{5.8}$$

$$\tilde{P}_{kj} = FP_{(k-1)g}F^T + \bar{P}_{(k-1)l}, \tag{5.9}$$

$$\tilde{w}_{kj} = \alpha_l w_{(k-1)g}, \tag{5.10}$$

for appropriate $j = 1,\cdots,GK$, $g = 1,\cdots,G$ and $l = 1,\cdots,L$ and $j = g + (l-1)L$.

In the measurement update stage, the operation of acting on new observations to improve on previously "predictive" states $p(x_k|\mathbf{z}_{1:k-1})$ can be approximated as

$$\begin{aligned} p(x_k|\mathbf{z}_{1:k}) &= \frac{p(\mathbf{z}_k|x_k)p(x_k|\mathbf{z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{z}_{k-1})} \\ &\approx C_n \sum_{j=1}^{GL} \tilde{w}_{kj}p(\mathbf{z}_k|x_k)\mathrm{N}(x_k;\tilde{\mu}_{(k)j},\tilde{P}_{(k)j}), \end{aligned} \tag{5.11}$$

where C_n is the normalizing constant. Therefore, the updated filtering distribution is approximated as

$$p(x_k|\mathbf{z}_{1:k}) \approx \sum_{j=1}^{GL} w_{kj}\mathrm{N}(x_k; \mu_{(k)j}, P_{(k)j}). \tag{5.12}$$

The reader is directed to [27] for more detailed explanations of the GSPF algorithm. Finally, notice that the conditional mean state estimate and the corresponding error covariance can be calculated respectively as:

$$\bar{x}_k = \sum_{j=1}^{GK} w_k^{(j)}\mu_k^{(j)}, \quad \bar{P}_k = \sum_{j=1}^{GK} w_k^{(j)}[P_k^{(j)} + (\mu_k^{(j)} - \bar{x}_k)(\mu_k^{(j)} - \bar{x}_k)^T]. \tag{5.13}$$

Next, we introduce another clock estimation scheme obtained through the integration of the CPF technique with the bootstrap sampling (BS) approach. The reader is directed to references [33, 34] for more detailed explanations about bootstrap sampling. In order to provide a consistent amount of observation data in the presence of errors during timing message transmissions, new sampled observation data from the original observation data are generated via the BS. Then, the clock offset is estimated based on the CPF. Notice that even in the presence of corrupted or lost data packets, BS can create additional samples to the original sample set, by drawing at random with replacement from $\mathbf{Z}$, and without being necessary for additional retransmissions. Each of the bootstrap samples is considered as new data. Based on the additional sampled observation data, we can then approximate the clock offset x by using the CPF. The major steps of the CPF approach with bootstrap sampling are summarized by the following pseudo-code:

Algorithm: CPF with BS

1. Conduct the experiment to obtain the random sample $\mathbf{Z} = \{\mathbf{Z}_1, \cdots, \mathbf{Z}_n\}$ and calculate the estimate $\hat{\theta}$ from the sample $\mathbf{Z}$.

2. Construct the empirical distribution $\hat{H}$, which puts equal mass $1/n$ at each observation $\mathbf{Z}_1 = \mathbf{z}_1, \cdots, \mathbf{Z}_n = \mathbf{z}_n$.

3. From $\hat{H}$, draw a sample $\mathbf{Z}^* = \{\mathbf{Z}_1^*, \cdots, \mathbf{Z}_n^*\}$, operation called bootstrap resampling.

4. From the bootstrap resample $\mathbf{Z}^*$, estimate the clock offset $\hat{x}$ by CPF.

The calculation of the computational cost of CPF is very complex, compared to GML and EML. In general, the computational cost of CPF is a function of the number of particles and the number of measurements. However, GML and EML are a function of the number of measurements, and do not use the particle filtering method. Hence, it is difficult explicitly to compare the complexities of GML/EML and CPF. However, we will use the big O notation to express the computational complexities of GML/EML and CPF in terms of flops by evaluating only the most computationally demanding steps. Letting L, N, and G denote the dimension of the state vector (x), the number of particles, and the number of GMM, respectively, the GML is approximately $O(L)$, and the CPF is approximately $O(GNL^3)$ which is the maximum complexity and occurs in the posterior pdf step. This shows that the CPF is approximately 300 times slower than GML in an application with $L = 1$, $N = 100$, and $G = 3$.

C. Simulation Results

In this section, extensive computer simulation results are presented to illustrate the performance of the CPF, CPF with BS, GML [24], and EML [24] approaches for estimating the clock offset in wireless sensor networks, assuming a variety of random network delay models such as asymmetric Gaussian, exponential, Gamma, and

Weibull as well as a mixture of Gamma and Weibull, respectively. These computer simulations and numerous other simulations not shown herein due to space limitations corroborate the conclusion that the proposed method can be widely and flexibly applied for any delay distribution. The stationary process v_k assumes the constant variance $Q = 1e-4$, while the number of particles and GMMs are set to 100 and 3, respectively. The bootstrap samples are twice the number of measurements.

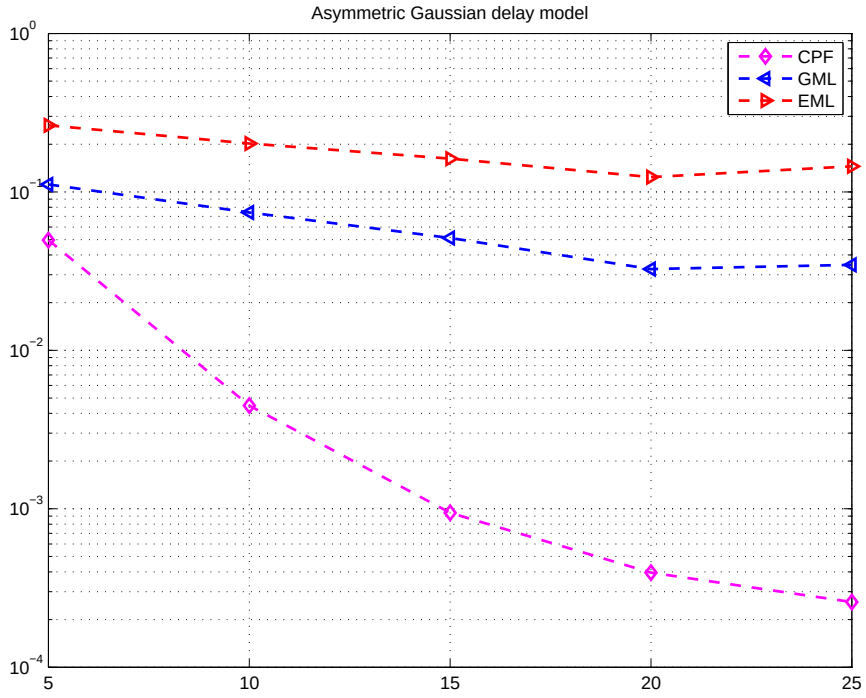

Fig. 5.1. MSEs of clock offset estimators for asymmetric Gaussian random delays $[\sigma_1 = 1, \sigma_2 = 4]$

Figs. 5.1-5.4 show the MSE (Mean Square Error) of the estimators when the network delay distributions are asymmetric Gaussian, exponential, Gamma, and Weibull pdfs, respectively. The subscript attached to the distribution parameters are used to differentiate the parameters of the uplink distribution with respect to those of the downlink distribution. For example, for an asymmetric Gaussian delay model, (σ_1^2) and (σ_2^2) denote the uplink and downlink variances, respectively. The MSE curves are

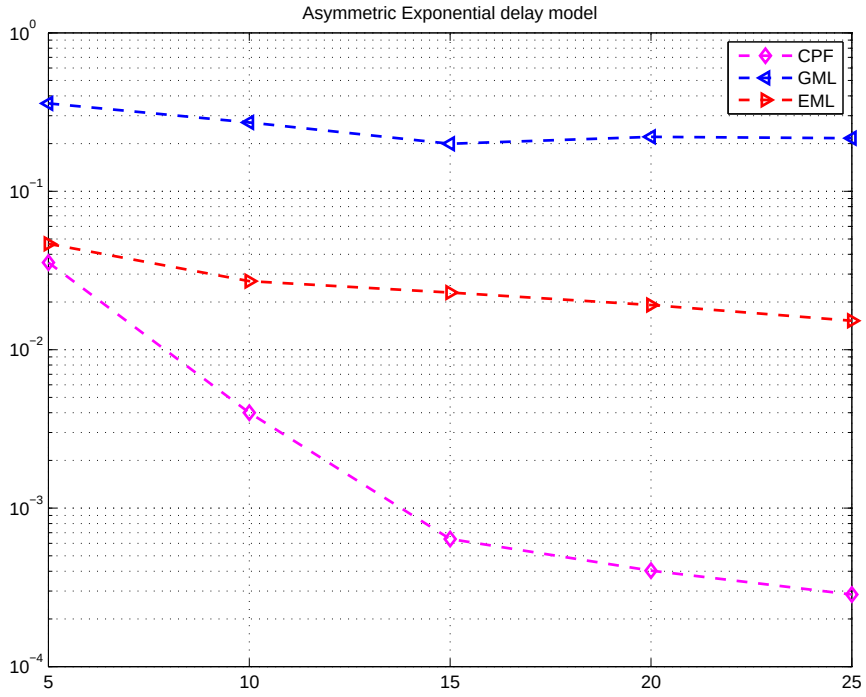

Fig. 5.2. MSEs of clock offset estimators for asymmetric exponential random delays $[\lambda_1 = 1, \lambda_2 = 5]$

plotted against the number of observations ranging from 5 to 25. Note that the CPF performs much better with over 100% MSE-reduction when compared to the GML or EML. Notice also that the CPF with BS exhibits the best performance in the presence of reduced number of observation data. Notice also the MSE of GML achieves better performance than EML in asymmetric Gaussian delay models, while EML assumes superior performance relative to GML in asymmetric exponential, Gamma, and Weibull delay models. The reason for this is that Gamma and Weibull delay models are closer to the exponential distribution than the Gaussian distribution.

To further quantify the robustness of the estimators, we studied the performance of the CPF with BS, CPF, GML, and EML under more general random delay models obtained by mixing two arbitrary distributions. For example, in Fig. 5.5, we mix uniformly the Gamma with the Weibull delay model, each distribution accounting

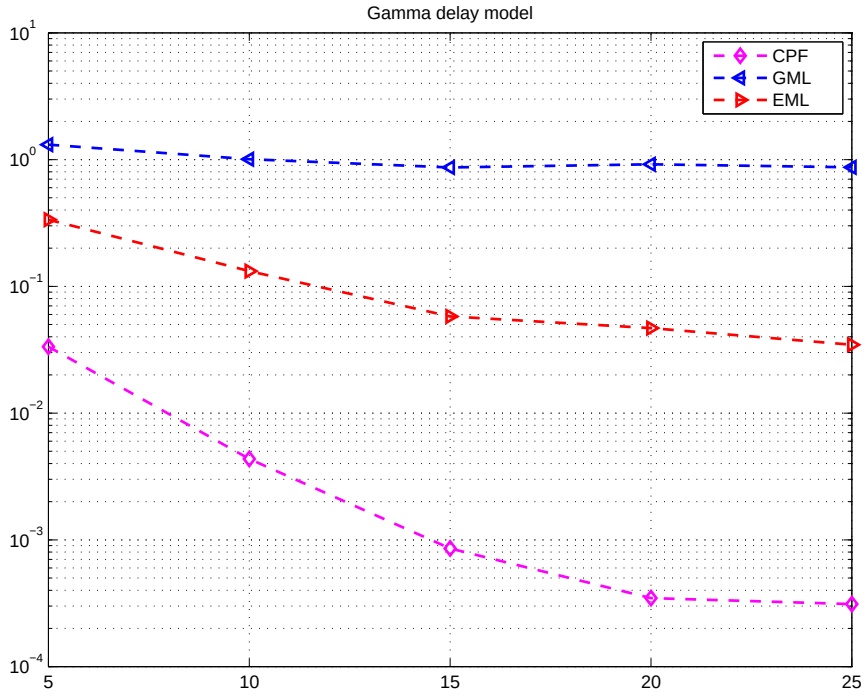

Fig. 5.3. MSEs of clock offset estimators for Gamma random delays $[\alpha_1 = 2, \beta_1 = 1]$

for 50% of samples. This means that if 10 observations are observed, 5 observations are Gamma and the remaining ones are Weibull distributed. From Fig. 5.5, we observe that CPF clearly outperforms the GML and EML. Notice that additional simulations not shown herein and performed assuming different mixture models for the network delays such as mixtures of Gaussian and exponential, Gaussian and Gamma, exponential and Gamma, and exponential and Weibull corroborate the same conclusion, namely the fact that CPF outperforms both the GML and EML no matter what distribution model is assumed for the network delays.

Figs. 5.6 - 5.10 depict the MSEs versus the number of observation data in the case when 2 message exchange errors occur with uniform distribution for the scenarios assumed by Figs. 5.1 - 5.5. From the Figs. 5.6 - 5.9, we observe that CPF with BS clearly outperforms CPF, GML, and EML. From Fig. 5.10, we observe that unlike

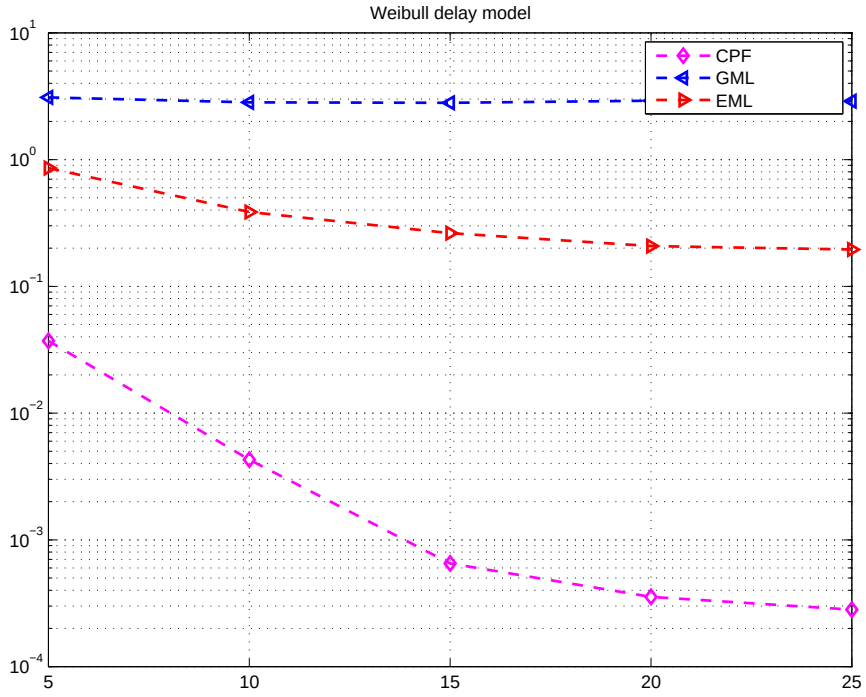

Fig. 5.4. MSEs of clock offset estimators for Weibull random delays [$\alpha_1 = 2, \beta_1 = 2$ and $\alpha_2 = 6, \beta_2 = 2$]

the Figs. 5.6 - 5.9, CPF with BS exhibits the best performance in the presence of a reduced number of observations. These simulation results corroborate the general conclusion that the CPF with BS, and CPF are reliable methods in the presence of a reduced number of samples.

D. Summary

In this chapter, we provided novel methods such as CPF and BS for estimating the clock offset in wireless sensor networks. The benefits are in terms of improved performance and applicability to any-random delay models such as asymmetric Gaussian, exponential, Gamma, Weibull, as well as mixtures of these delay models. In addition, the proposed CPF approaches are robust to the presence of a small number of observations, message exchange errors and unknown network delay distributions. Possible

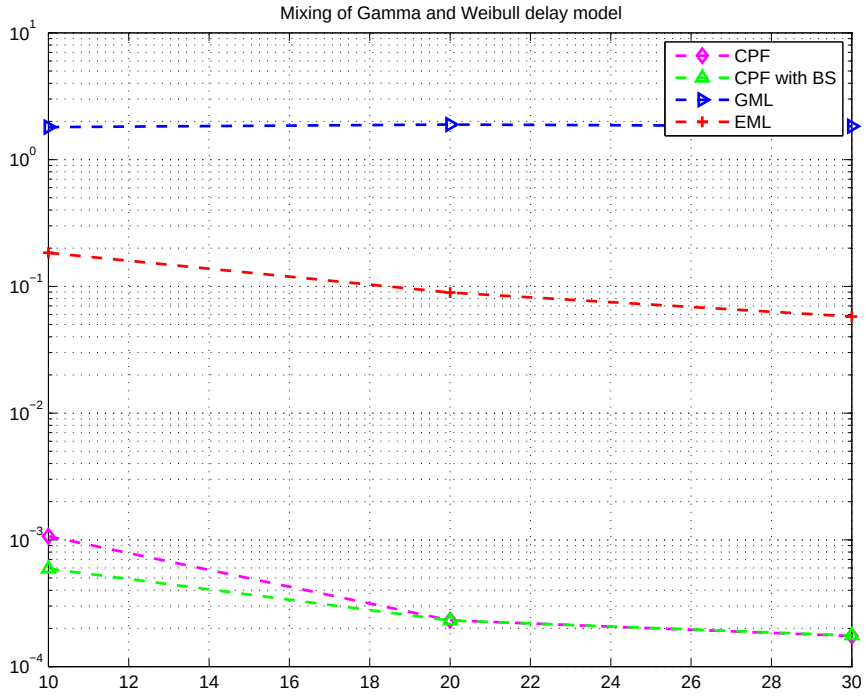

Fig. 5.5. MSEs of clock offset estimators for mixing of Gamma $[\alpha_1 = 2, \beta_1 = 5$ and $\alpha_2 = 2, \beta_2 = 2]$ and Weibull $[\alpha_1 = 2, \beta_1 = 2$ and $\alpha_2 = 6, \beta_2 = 2]$

disadvantages of the proposed composite particle filtering based approaches are the facts that analytical closed form expressions do not seem to exist for the clock estimators and the computation of the lower bound performance bounds appears difficult due to the non-Gaussian nature of involved distributions. In addition, the CPF with BS and CPF achieve excellent performance compared to GML and EML in environments which manifest in message exchange errors and time-varying network delay distributions.

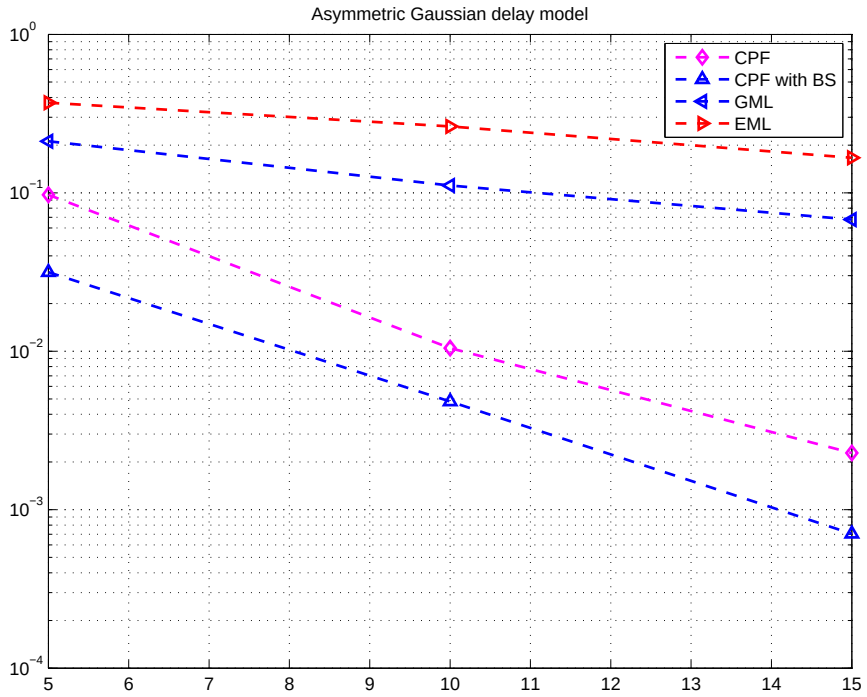

Fig. 5.6. MSEs of clock offset estimators for asymmetric Gaussian random delay $[\sigma_1 = 1, \sigma_2 = 4]$ and two message exchange errors

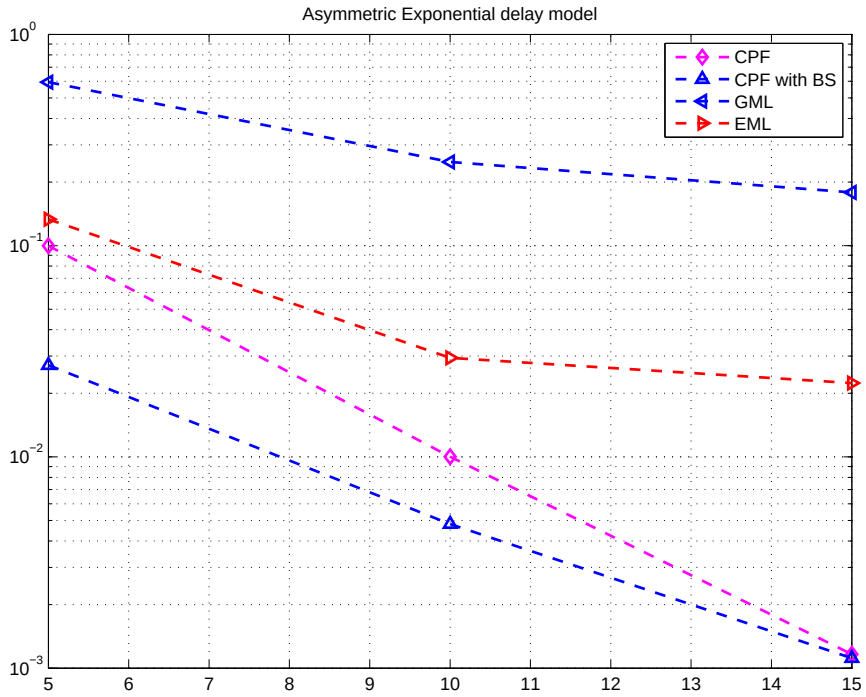

Fig. 5.7. MSEs of clock offset estimators for asymmetric exponential random delay $[\lambda_1 = 1, \lambda_2 = 5]$ and two exchange message errors occur

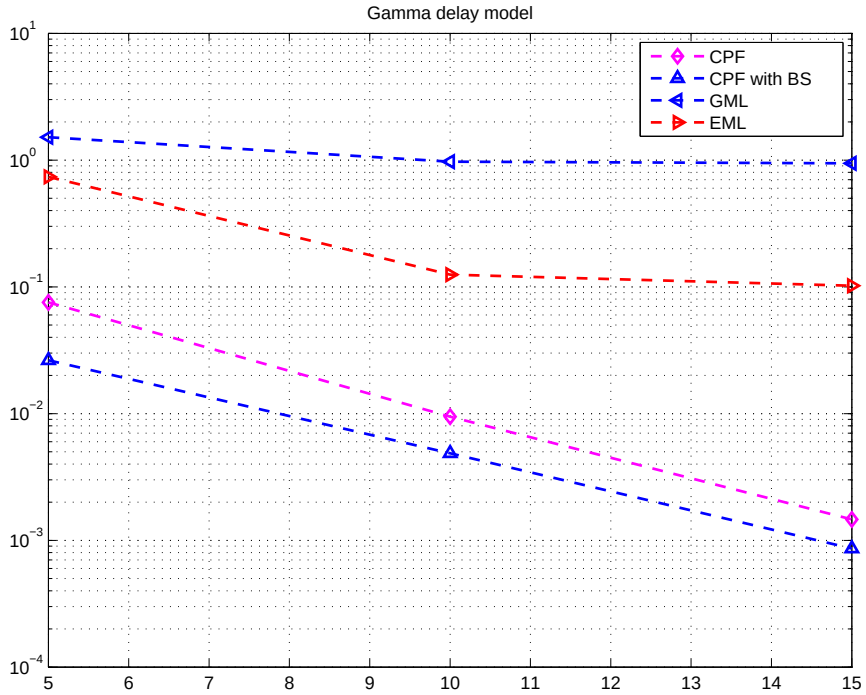

Fig. 5.8. MSEs of clock offset estimators for Gamma random delay $[\alpha_1 = 2, \beta_1 = 1]$ and two exchange message errors occur

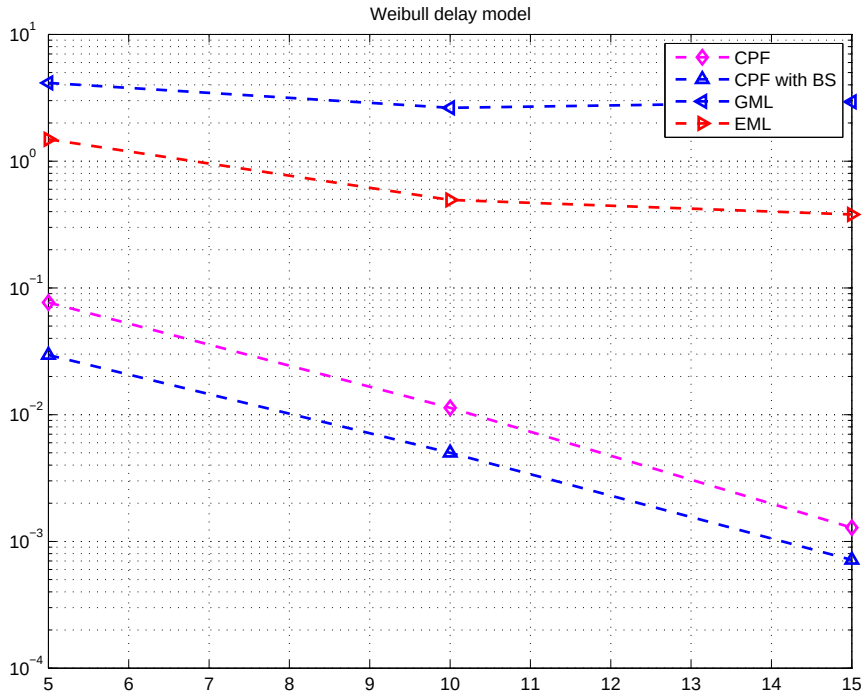

Fig. 5.9. MSEs of clock offset estimators for Weibull random delay [$\alpha_1 = 2, \beta_1 = 2$ and $\alpha_2 = 6, \beta_2 = 2$] and two exchange message errors occur

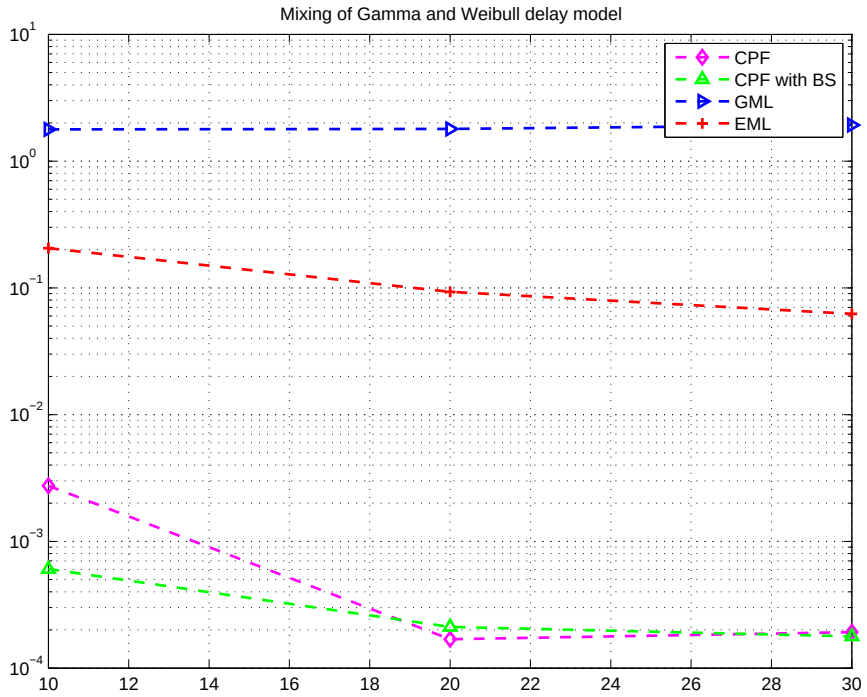

Fig. 5.10. MSEs of clock offset estimators for mixing of Gamma $[\alpha_1 = 2, \beta_1 = 5$ and $\alpha_2 = 2, \beta_2 = 2]$ and Weibull $[\alpha_1 = 2, \beta_1 = 2$ and $\alpha_2 = 6, \beta_2 = 2]$ and two exchange message errors occur

CHAPTER VI

ROBUST CLOCK SYNCHRONIZATION VIA NOISE DENSITY ESTIMATION

As a more advanced step, this chapter presents an extension of the previous results in Chapter IV to handle the situation of unknown observation noise distribution. In this chapter, a novel clock synchronization algorithm based on a Bayesian framework, called Iterative Gaussian Mixture Kalman Particle Filter (IGMKPF), is proposed. The algorithm combines the Gaussian Mixture Kalman Particle Filter (GMKPF) with a noise density estimation procedure through an iterative mechanism to achieve good and robust performance in the presence of unknown network delay distributions.

As mentioned in the previous chapter, a particle filter (PF) can be applied to the estimation of the state (clock offset). The PF-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 can not calculate the likelihood function, due to unknown measurement noise density, the proposal distribution plays an important role in the performance of PF. There are a lot of particle filtering techniques that could be applied for increasing the accuracy. Notice that general particle filtering techniques have no optimal proposal distribution and the observation noise density is in general modeled by its first two moments: mean and variance. Therefore, in general there exists a bias in both the MLE and general PF 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 Cramer-Rao Lower Bound (CRLB) can not serve as an efficient estimator and tight lower bound, respectively. In addition, the PF and its Posterior Cramer-Rao Bound (PCRB) may not be an optimal estimator or lower bound due to the finite number of observations and unknown observation noise density. The recursive PCRB has been shown to be

the information-theoretic mean-square error (MSE) bound for an unbiased sequential Bayesian estimator. 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. The bound is shown to have the same form as when the measurements for clock offset are with variance equal to the deterministic CRLB for a single data snapshot.

To cope with the limitations of MLE and PF, in this chapter we propose a novel clock offset estimator, the IGMKPF, and analyze the PCRB as a lower bound on its MSE-performance. The general design 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. 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 samples of observations might be reduced. Thus, if the observation noise density is fully captured and not partially modeled through the information provided by its mean and variance, then the estimator might reduce its bias and lead to improved performance. Second, the proposal distribution should be close to the optimal to reduce the error accumulation due to the iterative mechanism. Third, the estimator deals with non-Gaussian noise efficiently (e.g., by adopting Gaussian Mixture Models (GMMs) to capture various densities). Lastly, the optimal setting should be inferred from the analytical lower bound (e.g., PCRB).

Thus far, in the synchronization literature for WSNs, Kalman filtering and general adaptive signal processing techniques have been proposed (see e.g., [36], [37], [38] and [47]) to improve the MSE-performance of protocols such as RBS [17] or TPSN [18] under the assumption that the network delays are Gaussian. To deal with non-

Gaussian noise, [48] proposed recently the Gaussian Mixture Kalman Particle Filter (GMKPF) which was shown to perform well in a number of distributions: asymmetric exponential, asymmetric gamma, asymmetric Weibull, and mixture of them. As seen in [48], in the GMKPF, the importance sampling (IS) based measurement update step is combined with the time-update and proposal density generation steps that use a KF (Kalman Filter) 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 quite general asymmetric network delay models whose distributions are assumed to be 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 chapter could be seen as an extension of our previous results in [48] to handle the situation of unknown observation noise distribution.

The proposed estimation approach IGMKPF combines the GMKPF with a network delay density estimator by means of an iterative scheme. The proposed method has the merits of being robust and yielding very accurate clock offset estimates in the presence of unknown network delay distributions. During each iteration, by taking into account the results (posterior density, observation noise density, etc.) from the previous iteration, a new set of parameter values is obtained by adding small changes to the previous parameters in such a way that the new parameters are likely to lead to improved performance. By estimating the posterior density and observation noise density iteratively, the estimator reduces the bias caused by taking the expectation. 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 is, the better the performance of IGMKPF is due to the improved proposal distribution. To validate the accuracy of IGMKPF, the posterior Cramer-Rao bound is derived and a sequential Monte Carlo method is presented 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 CRLB in the presence of a reduced number of observations. In order to assess the process of fitting the unknown network delay density, we introduce the Kullback-Leibler Divergence (KLD) as measure between the estimated noise density and the true density. The simulation results show that the KLD is roughly proportional to the MSE. 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, the second-order derivatives of posterior pdf cannot be calculated directly. Therefore, to bypass this issue, we will utilize a gamma network delay model instead of exponential model. This is due to the fact that a gamma distributed random variable (RV) $z \sim \Gamma(\alpha = 1, \beta = \lambda)$, also assumes an exponential distribution with the rate parameter λ. We derive the PCRB for Gaussian, exponential, and gamma network delays, and then compare the CRLB with PCRB. Computer simulations shown that IGMKPF and PCRB can improve the performance of MLE and CRLB.

In this chapter, 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 proposed estimator, we derive

the PCRB for Gaussian, exponential, and gamma network delay noise models. The performance of IGMKPF, IGMKPF with perfect noise estimation, MLE, CRLB, and PCRB is simulated under the Gaussian and exponential network delay models. In addition, the performance of IGMKPF, IGMKPF with perfect noise estimation, MLEe, PCRB is simulated under the gamma network delay model. To illustrate the effects of noise estimation, we compare the KLD and 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.

A. Problem Formulation

Similarly to the mechanism in the previous chapters, 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:

$$\hat{x_k} = E\{x_k|\mathbf{z}^l\} \ , \tag{6.1}$$

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

$$x_k = Fx_{k-1} + v_{k-1} \ , \tag{6.2}$$

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:

$$\begin{aligned} \mathbf{z}_k &= \begin{bmatrix} d + x_k + L_k \\ d - x_k + M_k \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} d + \begin{bmatrix} 1 \\ -1 \end{bmatrix} x_k + \mathbf{n}_k \\ &= \mathbf{A}d + \mathbf{B}x_k + \mathbf{n}_k \ , \end{aligned} \tag{6.3}$$

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 $R = \text{diag}\{\sigma_n^2, \sigma_n^2\}$, and it accounts for the random network delays. One can now observe that (6.2) and (6.3) recast our initial clock offset estimation problem into a Gauss-Markov estimation problem with unknown state.

B. Posterior Cramer-Rao Bound for Sequential Bayesian Estimation

We need a lower bound on the covariance of the estimator, $\hat{x}_k$ for the true state x_k, defined by (6.2) and (6.3). Since we are interested in the class of trackers that are unbiased, bounding the MSE can be achieved by the PCRB alone. Assuming

that regularity condition holds for the probability density functions, the Posterior Cramer-Rao Bound (PCRB or Bayesian CRLB) [49] provides a lower bound on the mean-square error 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

$$\mathbf{M} = E_{\mathbf{z},x}\{[\hat{x}(\mathbf{z}) - x][\hat{x}(\mathbf{z}) - x]^T\}. \tag{6.4}$$

The PCRB 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) J:

$$\mathbf{M} \geq \mathbf{C} \equiv \mathbf{J}^{-1} \ . \tag{6.5}$$

The BFIM for x is defined as

$$\mathbf{J} = E_{\mathbf{z},x}\{-\triangle_x^x \ln p(\mathbf{z}, x)\} \quad , \tag{6.6}$$

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:

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

In [49], the BFIM is shown to follow the recursion:

$$\mathbf{J}_{k+1} = \mathbf{D}_k^{22} - (\mathbf{D}_k^{21})^T(\mathbf{J}_k + \mathbf{D}_k^{11})^{-1}\mathbf{D}_k^{12} \quad , \tag{6.7}$$

where

$$\mathbf{D}_k^{11} = E_x\{-\triangle_{x_k}^{x_k} \ln p(x_{k+1}|x_k)\} \tag{6.8}$$

$$\mathbf{D}_k^{12} = E_x\{-\triangle_{x_k}^{x_{k+1}} \ln p(x_{k+1}|x_k)\} \tag{6.9}$$

$$\begin{aligned}\mathbf{D}_k^{22} &= E_x\{-\triangle_{x_{k+1}}^{x_{k+1}} \ln p(x_{k+1}|x_k)\} \\ &+ E_x\{-\triangle_{x_{k+1}}^{x_{k+1}} \ln p(\mathbf{z}_{k+1}|x_{k+1})\} \quad .\end{aligned} \tag{6.10}$$

The recursion is initialized with

$$\mathbf{J}_0 = E_x\{-\triangle_{x_0}^{x_0} \ln p(x_0)\} \quad . \tag{6.11}$$

In general, the expectations in (6.8) - (6.10) have no closed-form analytical solution and must be approximated. 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 (6.8) - (6.9) become simply products of matrices. As a first step in applying the Monte Carlo integration, we need to define the following matrix functions:

$$\begin{aligned}\mathbf{\Lambda}^{11}(x_k, x_{k+1}) &= -\triangle_{x_k}^{x_k} \ln p(x_{k+1}|x_k) \\ \mathbf{\Lambda}^{12}(x_k, x_{k+1}) &= -\triangle_{x_k}^{x_{k+1}} \ln p(x_{k+1}|x_k) \\ \mathbf{\Lambda}^{22,a}(x_k, x_{k+1}) &= -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(x_{k+1}|x_k) \\ \mathbf{\Lambda}^{22,b}(x_{k+1}, \mathbf{z}_{k+1}) &= -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(\mathbf{z}_{k+1}|x_{k+1}) \quad .\end{aligned}$$

We can rewrite the equations (6.8) - (6.10) as follows:

$$\begin{aligned} D_k^{11} &= \int \Lambda^{11}(x_k, x_{k+1}) p(x_{k+1}|\mathbf{z}_{k+1}) dx_{k+1} \\ D_k^{12} &= \int \Lambda^{12}(x_k, x_{k+1}) p(x_{k+1}|\mathbf{z}_{k+1}) dx_{k+1} \\ D_k^{22} &= \int (\Lambda^{22,a}(x_k, x_{k+1}) p(x_{k+1}|\mathbf{z}_{k+1}) \\ &+ \Lambda^{22,b}(x_{k+1}, \mathbf{z}_{k+1})) p(x_{k+1}|\mathbf{z}_{k+1}) dx_{k+1} \quad . \end{aligned}$$

Once we have a sample representation of the posterior density, these expectation integrals can be calculated through sample mean approximation. 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 [50]. The a posterior 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 posterior 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:

$$\mathbf{D}_k^{11} \simeq \frac{1}{N} \sum_{n=1}^{N} \Lambda^{11}(X_k^{(n)}, X_{k+1}^{(n)}) \tag{6.12}$$

$$\mathbf{D}_k^{12} \simeq \frac{1}{N} \sum_{n=1}^{N} \Lambda^{12}(X_k^{(n)}, X_{k+1}^{(n)}) \tag{6.13}$$

$$\begin{aligned} \mathbf{D}_k^{22} &\simeq \frac{1}{N} \sum_{n=1}^{N} (\Lambda^{22,a}(X_k^{(n)}, X_{k+1}^{(n)}) \\ &+ \Lambda^{22,b}(X_k^{(n)}, X_{k+1}^{(n)})) \quad , \end{aligned} \tag{6.14}$$

where $X_{k+1}^{(n)}$, $n = 1, ..., N$, 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., [51] for additional details). The detailed derivation of PCRB-IGMKPF is described next.

PCRB-IGMKPF

1. Initialize the samples $X_0^{(n)}, n = 1, ..., N$, from $p(x_0)$, and then compute J_0 from (6.11).

2. Predict by sampling for $n = 1, ..., N$ from $p(x_{k+1}|x_k = x_k^{true})$ to choose $X_{k+1}^{\prime(n)}$.

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

     $p_g(x_{k-1}|\mathbf{z}_{k-1}) =$
     $\sum_{g=1}^{G} \alpha_{k-1}^{(g)} \mathrm{N}(x_{k-1}; \mu_{k-1}^{(g)}, P_{k-1}^{(g)})$

   - The process noise density is approximated by

     $p_g(v_{k-1}) = \sum_{i=1}^{I} \beta_{k-1}^{(i)} \mathrm{N}(v_{k-1}; \mu_{v_{k-1}}^{(i)}, Q_{k-1}^{(i)})$

   - The observation noise density is approximated by

     $p_g(\mathbf{n}_k) = \sum_{j=1}^{J} \gamma_k^{(J)} \mathrm{N}(\mathbf{n}_k; \mu_{\mathbf{n_k}}{}^{(j)}, \mathbf{R}_k^{(j)})$.

4. Pre-prediction step

   - Calculate the pre-predictive state density $\tilde{p}_g(x_k|\mathbf{z}_{k-1})$ using KF.

   - Calculate the pre-posterior 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: $\tilde{w}_k^{(l)} = \frac{p(\mathbf{z}_k|\chi_k^{(l)})\hat{p}_g(\chi_k^{(l)}|\mathbf{z}_{k-1})}{\hat{p}_g(\chi_k^{(l)}|\mathbf{z}_k)}$
- Normalize the weights: $w_k^{(l)} = \tilde{w}_k^{(l)} / \sum_{l=1}^{N} \tilde{w}_k^{(l)}$.
- 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 (4.8).

1. PCRB for the Gaussian Network Delay Model

The relationship in (6.2) determines the conditional pdf $p(x_{k+1}|x_k)$

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

and the relationship in (6.3) decides the conditional pdf $p(x_{k+1}|\mathbf{z}_{k+1})$

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

Accordingly, we can obtain the following equations:

$$\begin{aligned}
\mathbf{\Lambda}_k^{11}(x_k, x_{k+1}) &= -\triangle_{x_k}^{x_k} \ln p(x_{k+1}|x_k) = \frac{1}{\sigma_v^2} \\
\mathbf{\Lambda}_k^{12}(x_k, x_{k+1}) &= -\triangle_{x_k}^{x_{k+1}} \ln p(x_{k+1}|x_k) = -\frac{1}{\sigma_v^2} \\
\mathbf{\Lambda}_k^{22,a}(x_k, x_{k+1}) &= -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(x_{k+1}|x_k) = \frac{1}{\sigma_v^2} \\
\mathbf{\Lambda}_{k+1}^{22,b}(x_{k+1}, \mathbf{z}_{k+1}) &= -\triangle_{x_{k+1}}^{x_{k+1}} \ln p(\mathbf{z}_{k+1}|x_{k+1}) = \frac{\mathbf{B}^T\mathbf{B}}{\sigma_n^2} \\
&= \frac{2}{\sigma_n^2} \quad .
\end{aligned}$$

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

$$\begin{aligned}
\mathbf{D}_k^{11} &\simeq \frac{1}{N}\sum_{n=1}^{N} \Lambda^{11}(X_k^{(n)}, X_{k+1}^{(n)}) = \frac{1}{\sigma_v^2} \\
\mathbf{D}_k^{12} &\simeq \frac{1}{N}\sum_{n=1}^{N} \Lambda^{12}(X_k^{(n)}, X_{k+1}^{(n)}) = -\frac{1}{\sigma_v^2} \\
\mathbf{D}_k^{22} &\simeq \frac{1}{N}\sum_{n=1}^{N} (\Lambda^{22,a}(X_k^{(n)}, X_{k+1}^{(n)}) + \Lambda^{22,b}(X_k^{(n)}, X_{k+1}^{(n)})) \\
&= \frac{1}{\sigma_v^2} + \frac{2}{\sigma_n^2}.
\end{aligned}$$

From (6.7), it turns out that the evaluation of BFIM for Gaussian case can be done using the recursion:

$$\mathbf{J}_{k+1} = \frac{1}{\sigma_v^2} + \frac{2}{\sigma_n^2} - (-\frac{1}{\sigma_v^2})^T(\mathbf{J}_k + \frac{1}{\sigma_v^2})^{-1}(-\frac{1}{\sigma_v^2}) \quad . \tag{6.15}$$

Therefore, from (6.5), PCRB is inverse of BFIM (J). From the above equations, we note that PCRB is a function of σ_v^2 and σ_n^2. In [24], the CRLB is expressed as $\text{var}(\hat{x}) \geq \sigma_n^2/2N$. 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.

2. PCRB for Exponential and Gamma Network Delay Models

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

$$p(x_{k+1}|x_k) = \frac{1}{\sqrt{2\pi\sigma_v^2}} e^{-\frac{1}{2\sigma_v^2}[(x_{k+1}-x_k)^2]} \quad , \tag{6.16}$$

while (6.3) decides the conditional pdf $p(x_{k+1}|\mathbf{z}_{k+1})$:

$$p(\mathbf{z}_{k+1}|x_{k+1}) = \lambda e^{-\lambda[(\mathbf{z}_{k+1}-\mathbf{A}d-\mathbf{B}x_{k+1})]} \quad . \tag{6.17}$$

(6.17) 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. (6.3) leads to the conditional pdf $p(x_{k+1}|\mathbf{z}_{k+1})$:

$$\begin{aligned} p(\mathbf{z}_{k+1}|x_{k+1}) &= \frac{\beta^\alpha}{\Gamma(\alpha)}[(\mathbf{z}_{k+1} - \mathbf{A}d - \mathbf{B}x_{k+1})]^{\alpha-1} \\ &\times \quad e^{-\beta[(\mathbf{z}_{k+1}-\mathbf{A}d-\mathbf{B}x_{k+1})]} \quad , \end{aligned} \tag{6.18}$$

where if α is a positive integer, then $\Gamma(\alpha) = (\alpha - 1)!$.

From the above equation, it follows that

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

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

$$\begin{aligned}
\mathbf{D}_k^{11} &\simeq \frac{1}{N}\sum_{n=1}^{N}\Lambda^{11}(X_k^{(n)}, X_{k+1}^{(n)}) = \frac{1}{\sigma_v^2} \\
\mathbf{D}_k^{12} &\simeq \frac{1}{N}\sum_{n=1}^{N}\Lambda^{12}(X_k^{(n)}, X_{k+1}^{(n)}) = -\frac{1}{\sigma_v^2} \\
\mathbf{D}_k^{22} &\simeq \frac{1}{N}\sum_{n=1}^{N}(\Lambda^{22,a}(X_k^{(n)}, X_{k+1}^{(n)}) + \Lambda^{22,b}(X_k^{(n)}, X_{k+1}^{(n)})) \\
&= \frac{1}{\sigma_v^2} + \frac{1}{N}\sum_{n=1}^{N}\Lambda^{22,b}(X_{k+1}^{(n)}, \mathbf{z}_{k+1}) \ .
\end{aligned}$$

3. Comparison between PCRB and CRLB

Figs. 6.1-6.2 shows 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. 6.1-6.2 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 better performance levels than CRLB [24] depending on how J_0 is selected. Fig. 6.3 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 variance of process noise decreases, the modeling error is almost removed and then the model becomes identical with the one adopted by the ML approach. The simulation results corroborated the fact that CRLB exhibits almost the same performance levels as PCRLB provided that 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 $J_0 = 1/(\sigma_v^2)$, PCRLB is significantly lower than CRLB.

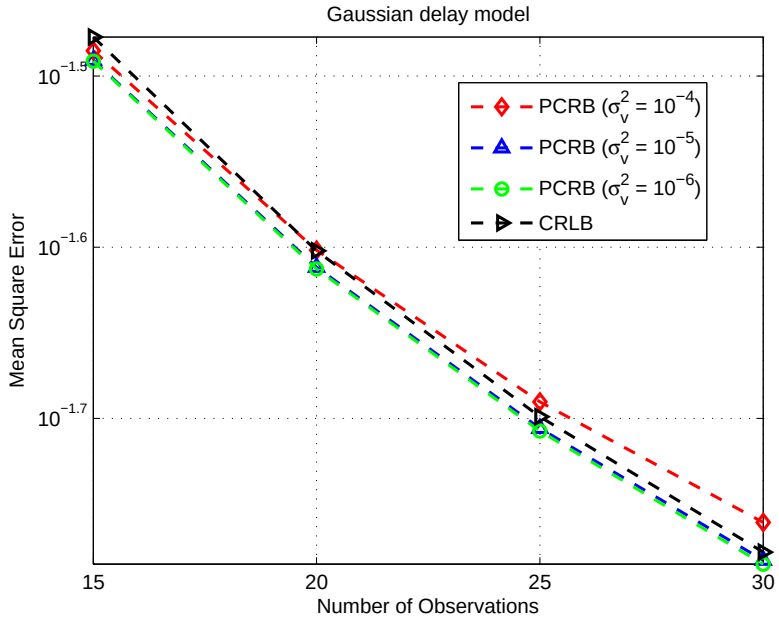

Fig. 6.1. PCRB and CRLB for symmetric Gaussian random delays ($\sigma_n^2 = 1$ and $J_0 = 1$)

Computer simulations also illustrate the fact that PCRLB might increase as the number of observations increases. In the state-space iterative estimation model, prediction errors and random sampling errors [50] might occur due to the non-optimal 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 improve-

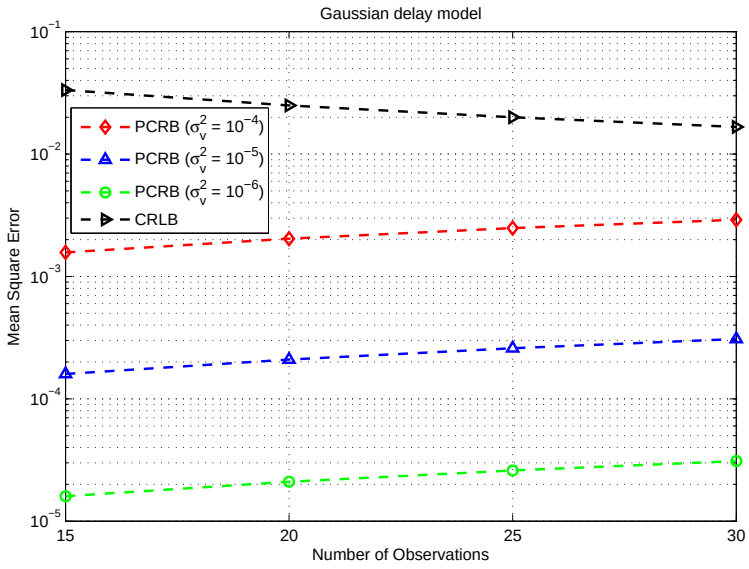

Fig. 6.2. PCRB and CRLB for symmetric Gaussian random delays ($\sigma_n^2 = 1$ and $J_0 = 1/\sigma_v^2$)

ment 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 error accumulation is important in the performance of the Bayesian iterative estimation approach 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 optimal proposal distribution, there will always exist random sampling errors.

Notice that particle filters (PF) track the unknown state using the observation, process, and prior density. However, the particle filter is not optimal due to the random sampling errors, non-optimal 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 [51]. For a given fixed t, there is

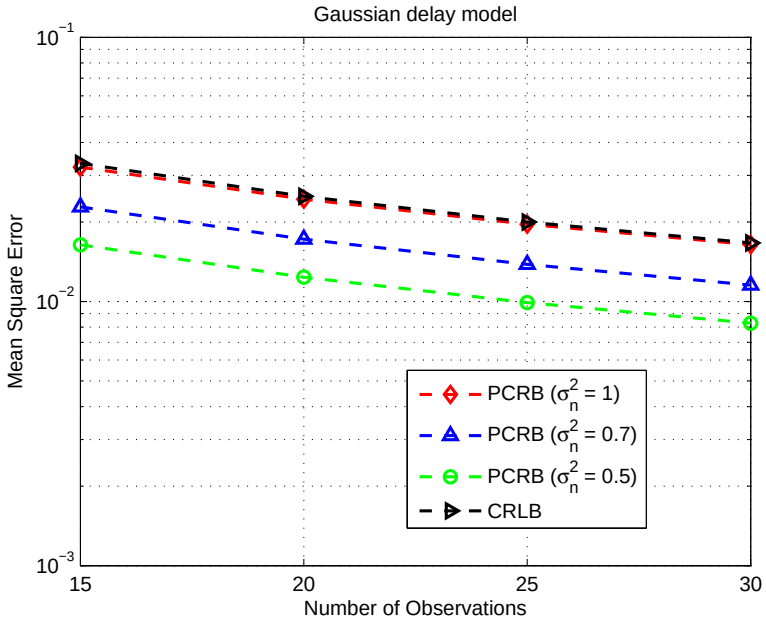

Fig. 6.3. PCRB and CRLB for symmetric Gaussian random delays ($\sigma_v^2 = 10^{-6}$ and $J_0 = 1$)

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 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 distribution. However, the remaining parameter (the observation noise density is fixed) is not changed. In case that the observation noise density is fixed and process noise variance is lower than 10^{-4}, GMKPF has the same performance as MLE or CRLB. However, in the case that the observation noise density can be estimated more accurately and the process noise variance is negligible relative to the observation noise power, we will show in the next sections that IGMKPF yields much better performance than MLE and CRLB.

C. An Iterative Gaussian Mixture Kalman Particle Filtering Approach

In wireless or mobile communication systems, such as CDMA, GSM, and 3G, iterative techniques (e.g., linear parallel interference cancelation, successive interference cancelation) were applied to resolve the multiuser detection problem. It is known that the iterated linear interference cancelation technique employed for the detection of linear multiple access channels is equivalent to the well known iterative methods for finding the solution of a system of linear equations [52]. In general, Successive Interference Cancelation (SIC) decodes the signals from the strongest interferers and subtracts the re-encoded signal, effectively increasing the signal-to-noise ratio. This is iteratively repeated until the first interferer is obtained. Likewise, the proposed Iterative Gaussian Mixture Kalman Particle Filtering (IGMKPF) estimator combines the Gaussian mixture Kalman particle filter (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 using the prior, process, observation, state posterior, and noise density, which are propagated over time. In order to analyze the MSE-performance of the proposed IGMKPF technique, computer simulations will be conducted, and the posterior Cramer-Rao Bound and the Kullback-Leibler Divergence (KLD) between the true and estimated network delay noise will be evaluated as well.

Fig. 6.4 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

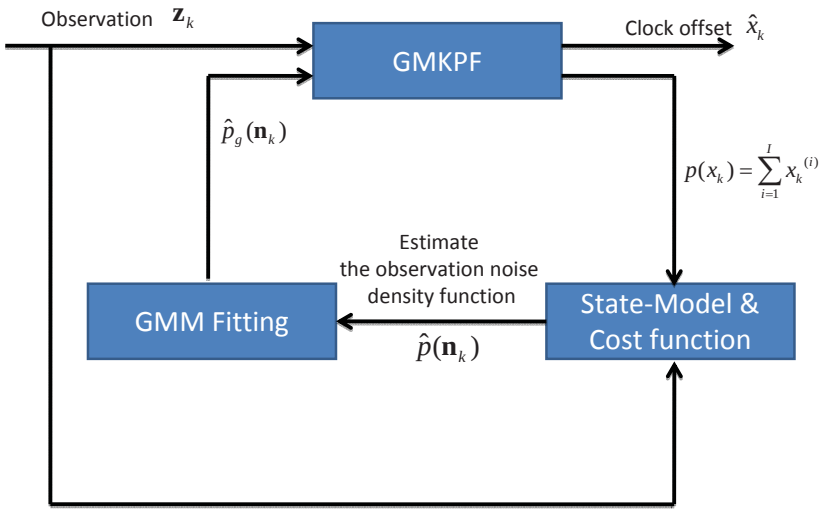

Fig. 6.4. Block diagram representation of the IGMKPF estimator

using the innovation, 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 up to the end of observations.

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 pseudo-code of IGMKPF algorithm is next described. The detailed explanation of GMKPF is already described in Chapter IV.

IGMKPF algorithm

1. At time k, initialize the densities and set the initial state $\hat{x}_{k-1} = x_{ML}$.

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. OND step (estimate the observation noise density)

   - Calculate the observation noise density $p(\hat{n})$ given $\mathbf{z}_k$ and $p_g(x_k|\mathbf{z}_k)$, and state model ((6.2) and (6.3))
   - The observation noise density using GMM is approximated by: $p_g(\mathbf{n}_k) = \sum_{j=1}^{J} \gamma_k^{(J)} \mathrm{N}(\mathbf{n}_k; \mu_{\mathbf{n_k}}{}^{(j)}, \mathbf{R}_k^{(j)})$

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 (4.8).

To validate the accuracy of the observation noise density estimation, we use the Kullback-Leibler Divergence (KLD) [53] as a measure of fitting the estimated noise density to the true one. Fig. 6.5 shows the relationship between KLD and MSE-performance of IGMKPF. In Fig. 6.5, the upper graph refers to KLD, while the lower graph depicts the MSE performance of IGMKPF. Upon 10 Monte Carlo simulation runs, the KLD and the MSE performance exhibit a very similar behavior.

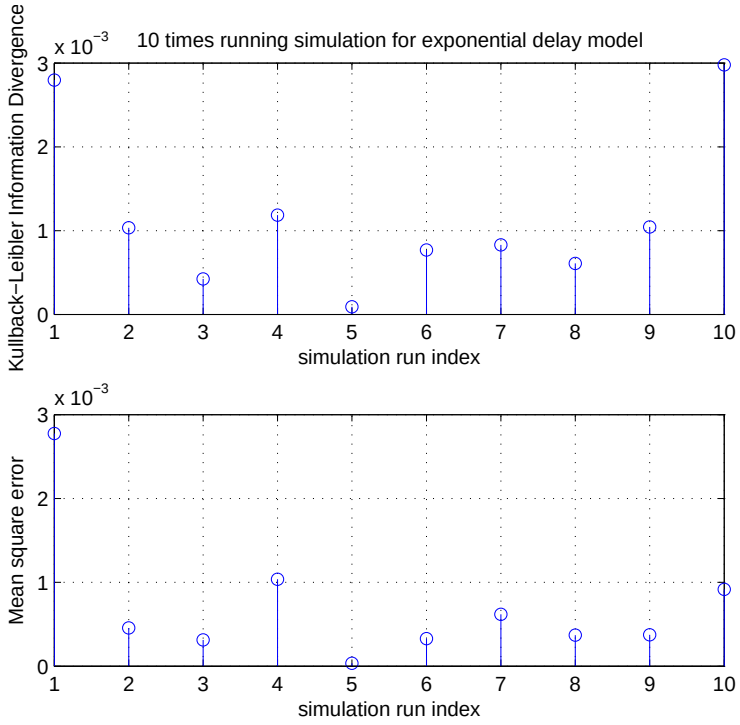

Fig. 6.5. KLD and MSE for symmetric exponential random delays ($\lambda = 1$)

These plots illustrate the fact that the more accurately the observation noise density is estimated, the better the MSE performance of IGMKPF is.

D. Simulation Results

In this section, computer simulations will be conducted to assess the performance of IGMKPF, PCRB-IGMKPF, MLEg [24], MLEe [24], 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 [24] for symmetric Gaussian and exponential random delays are good examples of initializations. The initial values $\hat{x}_0 = \hat{x}_{MLEg}$, and $\hat{x}_0 = \hat{x}_{MLEe}$ 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 the observation and estimated posterior pdfs. Therefore, the performance of IGMKPF is expected to be better than CRLB.

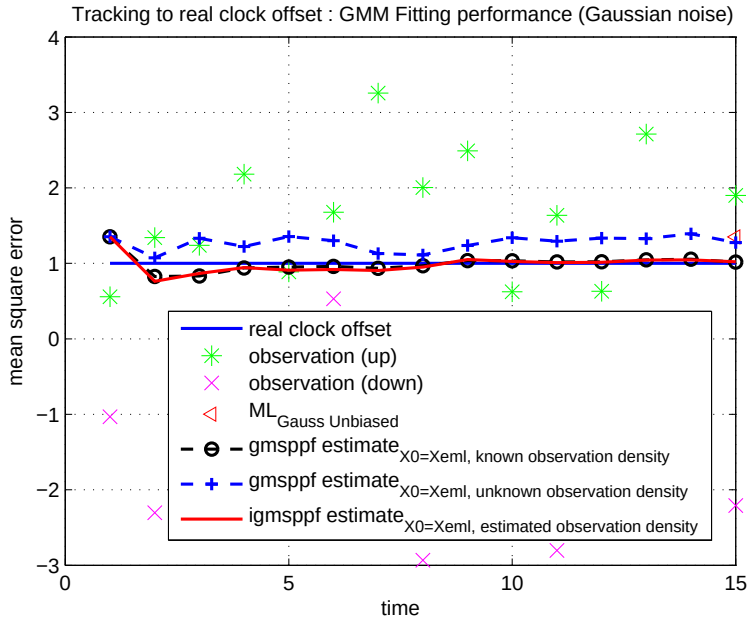

Fig. 6.6. MSEs of clock offset estimators for Gaussian ($\sigma_n^2 = 1$)

Fig. 6.6 shows the MSE performances of MLEe, MLEg, GMKPF, and IGMKPF. Left-triangles and right-triangles denote MLEe and MLEg, respectively. The dashed line indicates GMKPF with the initial condition $x_0 = X_{MLEe}$ and known observation noise density. The dashed-with-circles line represents GMKPF with initial condition

$x_0 = X_{true}+1$ and unknown observation noise density. Dashed-with-addition-sign line denotes GMKPF with initial condition $x_0 = X_{MLEe}$ and unknown observation density. Solid line represents IGMKPF with initial condition $x_0 = X_{MLEe}$ 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 $J_0 = 1$. However, 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 help to reduce the bias, which will help IGMKPF estimate to approach closer the real clock offset.

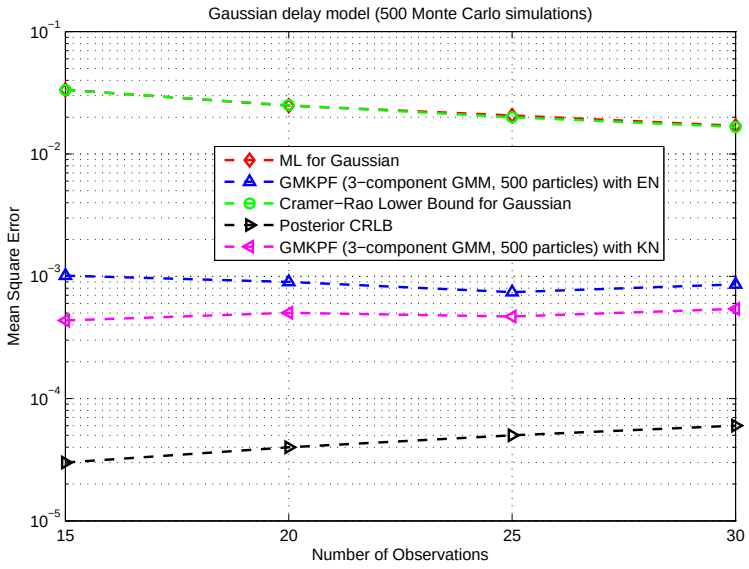

Fig. 6.7. MSEs of clock offset estimators for Gaussian ($\sigma_n^2 = 1$)

Figs. 6.7 - 6.9 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 set-ups with known observation noise density and

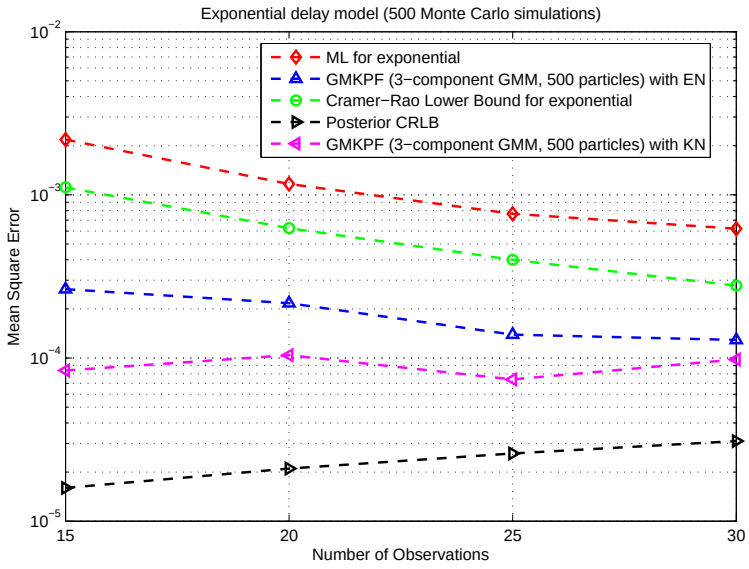

Fig. 6.8. MSEs of clock offset estimators for exponential ($\lambda = 1$)

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. Note that IGMKPF ($G = 3$) performs much better (over 100% reduction in MSE) than CRLB and 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 does not depend on the number of observations [51]. 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,

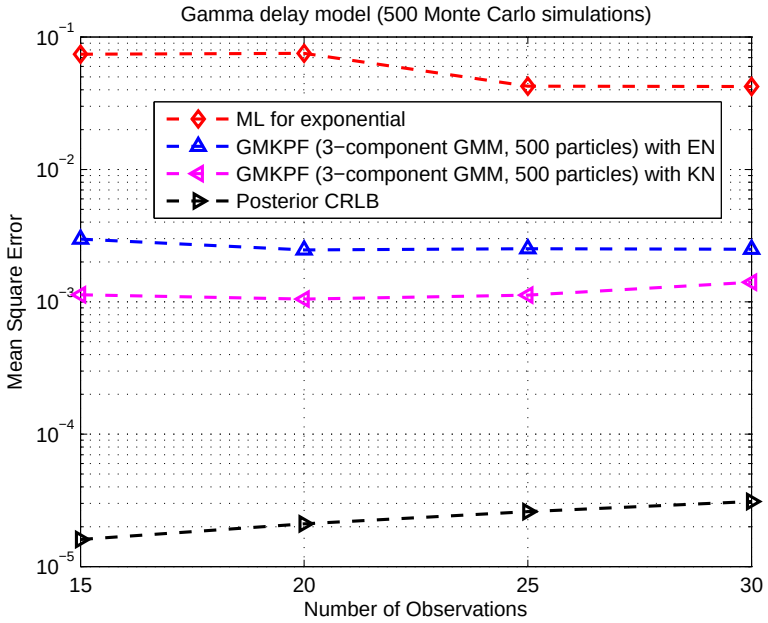

Fig. 6.9. MSEs of clock offset estimators for Gamma ($\alpha_1 = 2, \beta_1 = 2$)

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. 6.7 and 6.8, for exponential ($\lambda = 1$) and gamma ($\alpha = 2$ and $\beta = 2$) network delay models, respectively. In Fig. 6.9, 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 [18], RBS [17], and FTSP [19], with respect to the number of required timing messages (which practically indicates the amount of energy consumption) to achieve a specific MSE-performance. Let $N_{TPSN-MLEg}$, $N_{RBS-MLEg}$, and $N_{FTSP-MLEg}$ 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 [18], denoting by J the overall number of sensor nodes, and assuming that $2K$ timing messages are required in every pairwise synchronization, $N_{TPSN-MLEg}$ is equal to the number of pairwise synchronization times the number of required timing messages per pairwise synchronization. Thus, $N_{TPSN-MLEg} = 2K(J-1)$. In RBS [17], $N_{RBS-MLEg} = K + J(J-1)/2$, since the number of unique pairs in the network is $J(J-1)/2$. In FTSP [19], the number of required timing messages is $N_{FTSP-MLEg} = JK$. When MSE $= 10^{-3}$ and the number of nodes is 100, $N_{TPSN-MLEg} = 2K(100-1) = 2 \times 500 \times 99 = 99000$ for MLEg [24], $N_{TPSN-IGMKPF} = 2K(100-1) = 2 \times 15 \times 99 = 2970$ for IGMKPF from Fig. 6.7. Consequently, the benefit of IGMKPF over MLEg is huge in terms of energy consumption in the network. Similar statements hold for $N_{RBS-MLEg}$ and $N_{FTSP-MLEg}$.

E. Summary

This chapter proposed a novel method for estimating the clock offset in wireless sensor networks. The benefits of the proposed synchronization method are improved performance compared to CRLB and 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 hard

to derive lower bounds. However, this chapter derived the posterior Cramer-Rao bound (PCRB) and IGMKPF. An important element in improving the performance of clock estimator is the prediction of unknown observation noise density which led to an improved estimator. IGMKPF presents better performance than maximum likelihood estimator (MLE). The chapter presented a robust estimator based on the IGMKPF which is capable of estimating the clock offset in arbitrary delay models and in the presence of time-varying clock offset, and that can be used in a number of applications of wireless sensor networks with tight synchronization requirements [43], [44]. Computer simulations corroborate that the proposed method yields a superior performance.

CHAPTER VII

CONCLUSIONS AND FUTURE WORK

In WSNs, clock synchronization is significantly important for a lot of tasks such as tracking, localization, data fusion, and efficient duty cycling. However, the requirements of limited energy consumption and increased clock accuracy makes clock synchronization problem in WSNs very difficult. This dissertation focused on finding robust estimation techniques of the clock offset for synchronization in WSNs under a two-way message exchange mechanism using the sender-receiver synchronization scheme in order to cope with symmetric or asymmetric, and non-Gaussian or non-exponential random delay distributions.

The clock offset estimator based on bootstrap bias correction has been derived for asymmetric exponential random delay models. Next, the clock offset estimator using the robust M-estimation technique is also proposed in order to cope with the circumstances that one underlying delay distribution may be contaminated by or mixed with another delay distribution. Subsequently, more accurate and robust estimators of clock offset based on particle filtering techniques are obtained and compared with the maximum likelihood estimator to show the outstanding performances in the presence of non-Gaussian or even unknown network delay environments. One of the estimators is the Gaussian mixture Kalman particle filter, which is a better and more flexible alternative to Gaussian and exponential maximum likelihood estimators of clock offset in arbitrary random delay models. The clock estimator based on the composite particle filter is also proposed, which approximates the filtering and predictive distributions by exploiting weighted Gaussian mixtures and is basically implemented through banks of Kalman filters replacing Gaussian particle filters. In addition, the composite particle filter combined with the bootstrap sampling is shown

to represent a more accurate clock offset estimator. Lastly, another novel clock offset estimator based on the iterative Gaussian mixture Kalman particle filter combining the Gaussian mixture Kalman particle filter with a noise density estimation scheme via an iterative scheme is shown to outperform GMKPF and applicable to general unknown network delay distributions.

There are several future research directions that can be explored. It is worth examining the possibility that the statistical methodologies adopted in this dissertation can be extended for clock synchronization models that include clock skew and drift in the set of unknown parameters. Furthermore, it is of interest to investigate whether the proposed techniques can be applied to other clock synchronization protocols that use receiver-receiver synchronization or receiver-only synchronization schemes such as RBS (Reference Broadcast Synchronization) or PBS (Pairwise Broadcast Synchronization).

REFERENCES

[1] J. Hill, M. Horton, R. Kling, and L. Krishnamurthy, "The platforms enabling wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 41-46, June 2004.

[2] Intel Mote, *http://techresearch.intel.com/articles/Exploratory/1503.htm*, Retrieved on Nov. 10, 2009.

[3] Smart Dust, *http://robotics.eecs.berkeley.edu/ pister/SmartDust/*, Retrieved on Nov. 10, 2009.

[4] TinyOS, *http://webs.cs.berkeley.edu/tos/*, Retrieved on Nov. 10, 2009.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393-422, Mar 2002.

[6] R. Szewczyk, E. Osterweil, J. Polastre, M. Hamilton, A. Mainwaring, and D. Estrin, "Habitat monitoring with sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 34-40, June 2004.

[7] P. Johnson and D.D. Andrews, "Remote continuous physiological monitoring in the home," *Journal of Telemedicine Telecare*, vol. 2, pp. 107-113, Jan 1996.

[8] I. F. Akyildiz, O. Akan, C. Chen, J. Fang, and W. Su, "InterPlaNetary Internet: state-of-the-art and research challenges," *Computer Networks*, vol. 43, no. 2, pp. 75-112, Oct. 2003.

[9] S. Burleigh, V. Cerf, R. Durst, K. Fall, A. Hooke, K. Scott, and H. Weiss, "The InterPlaNetary Internet: a communications infrastructure for Mars exploration,"

in 53rd International Astronautical Congress, The World Space Congress, October 2002.

[10] Q. Li, M. DeRosa, and D. Rus, "Distributed algorithms for guiding navigation across a sensor network," *in Proc. of Ninth Annual International Conference on Mobile Computing and Networking*, San Diego, CA, pp. 313-326, September 2003.

[11] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Transactions on Information Systems*, vol. 10, no. 1, pp. 91-102, Jan. 1992.

[12] J. Werb and C. Lanzl, "Designing a positioning system for finding things and people indoors," *IEEE Spectrum*, vol. 35, no. 9, pp. 71-78, Sep. 1998.

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

[14] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*, San Francisco, CA, Morgan Kaufmann, 2004.

[15] D. Mills, "Internet time synchronization: The network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482-1493, Oct 1991.

[16] N. Bulusu and S. Jha, *Wireless Sensor Networks: A Systems Perspective*, Norwood, MA, Artech House, 2005.

[17] J. Elson, L. Girod, and D. Estrin, "Fine-grained network time synchronization using reference broadcasts," *in Proc. of the 5th Symposium on Operating System Design and Implementation*, Boston, MA, vol. 36, pp. 147-163, Dec 2002.

[18] S. Ganeriwal, R. Kumar, and M.B. Srivastava, "Timing synch protocol for sensor networks," *in Proc. of 1st International Conference on Embedded Network Sensor Systems*, Los Angeles, CA, pp. 138-149, Nov 2003.

[19] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "The flooding time synchronization protocol," *in Proc. of the 2nd International Conference on Embedded Networked Sensor Systems*, ACM Press, pp. 39-49, Baltimore, MD, Nov 2004.

[20] W. Su and I. F. Akyildiz, "Time-diffusion synchronization protocol for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, pp. 384-397, Apr 2005.

[21] K.-L. Noh, E. Serpedin, and K. A. Qaraqe, "A new approach for time synchronization in wireless sensor networks: pairwise broadcast synchronization," *IEEE Transactions on Wireless Communications*, vol. 7, no. 9, pp. 3318-3322, Sept. 2008.

[22] D. R. Jeske, "On maximum likelihood estimation of clock offset," *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 53-54, Jan 2005.

[23] V. Paxson, "On calibrating measurements of packet transit times," *in Proc. of 7th ACM Sigmetrics Conference*, Madison, WI, vol. 26, pp. 11-21, Jun 1998.

[24] K.-L. Noh, Q. Chaudhari, E. Serpedin, and B. Suter, "Novel clock phase offset and skew estimation using two-way timing message exchanges for wireless sensor networks," *IEEE Transactions on Communications*, vol. 55, no. 4, Apr 2007.

[25] P.J. Huber, *Robust Statistics*, Hoboken, NJ, Wiley, 2004.

[26] J. H. Kotecha and P.M. Djuric, "Gaussian particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2592-2601, Oct. 2003.

[27] J. H. Kotecha and P.M. Djuric, "Gaussian sum particle filtering," *IEEE Transactions on Signal Processing*, vol. 51, no. 10, pp. 2602-2612, Oct. 2003.

[28] S. Narasimhan and S. S. Kunniyur, "Effect of Network Parameters on Delay in Wireless Ad-hoc Netwoks," *University of Pennsylvania Technical Report*, Jun 2004.

[29] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 3rd ed., New York, McGraw-Hill, 1991.

[30] C. Bovy, H. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, and P. Mieghem, "Analysis of end-to-end delay measurements in Internet," presented at *the Passive and Active Measurements Workshop*, Fort Collins, CO, Mar 2002.

[31] S. Moon, P. Skelley and D. Towsley, "Estimation and removal of clock skew from network delay measurements," *in Proc. of the IEEE INFOCOM Conference on Computer Communications*, New York, pp. 227-234, Mar 1999.

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

[33] B. Efron and R.J. Tibshirani, *An Introduction to the Bootstrap*, Boca Raton, FL, Chapman & Hall, 1993.

[34] A.M. Zoubir and D.R. Iskander, *Bootstrap Techniques for Signal Processing*, Cambridge, U.K., Cambridge University Press, 2004.

[35] J. Jureckova and J. Picek, *Robust Statistical Methods with R*, Boca Raton, FL., Chapman & Hall, 2006.

[36] 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'05 Conference*, pp. 130-141, Nov. 2005, San Diego, CA.

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

[38] D. Tulone, "Resource-efficient time synchronization for wireless sensor networks," *in Proc. DIALM-POMC Workshop on Foundations of Mobile Computing*, S. Basagni and C. A. Phillips, editors, pp. 52-50, Philadelphia, PA, Oct. 2004.

[39] R. van der Merwe and E. Wan, "Gaussian mixture sigma-point particle filters for sequential probabilistic inference in dynamic state-space models", *in Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Hong Kong, pp. 701-704, April 2003.

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

[41] S. J. Julier and J. K. Uhlmann, "A General Method for Approximating Nonlinear Transformations of Probability Distributions," *Technical Report., Robotics Report Group, Department of Engineering Science*, University of Oxford, 1996.

[42] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Transactions On Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1344-1348, Aug. 2005.

[43] J. Heidemann, W. Ye, J. Wills, A. Syed, and Y. Li, "Research challenges and applications for underwater sensor networking," *in Proc. of the IEEE Wireless*

Communications and Networking Conference, vol. 1, Las Vegas, NV, pp.228-235, April 2006.

[44] A. Syed and J. Heidemann, "Time Synchronization for High Latency Acoustic Networks," *Technical Report ISI-TR-2005-602*, USC/Information Sciences Institute, April 2005.

[45] M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking", *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, Feb. 2002.

[46] P. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19-38, 2003.

[47] F. Kirsch and M. Vossiek, "Distributed Kalman filter for precise and robust clock synchronization in wireless networks," *in Proc. of the 4th International Conference on Radio and Wireless Symposium*, San Diego, CA, pp. 455-458, 2009.

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

[49] P. Tichavsky, C. Muravchik, A. Nehorai, "Posterior Cramer-Rao bounds for discrete-time nonlinear filtering", *IEEE Transactions on Signal Processing*, vol. 46, no. 5, pp. 1386-1396, May 1998.

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

[51] M. Isard, A. Blake, "CONDENSATION-conditional density propagation for visual tracking", *International Journal of Computer Vision*, vol. 29, no. 1, pp.5-28, 1998.

[52] H. Elders-Boll, H.-D. Schotten, and Axel Busboom, "Efficient implementation of linear multiuser detectors for asynchronous CDMA systems by linear interference cancellation", *European Transactions on Telecommunications*, vol. 9, no. 5, pp.427-438, May 1998.

[53] S. Kullback, *Information Theory and Statistics*, Mineola, New York, Dover Publications Inc., 1968.

VITA

Jae Han Lee received a B.S. degree and an M.S. degree in electrical engineering from Seoul National University, Seoul, Korea in February 1997 and February 1999, respectively. He worked at the Digital TV Lab at LG Electronics Inc., Seoul, Korea, from 1999 to 2005. In 2010, he received a Ph.D. degree in electrical and computer engineering at Texas A&M University, College Station.

His research interests are in the areas of detection and estimation theory, clock synchronization in wireless sensor networks, and statistical signal processing.

His address is 214 Zachry Engineering Center, TAMU 3128, College Station, Texas 77843-3128.

The typist for this dissertation was Jae Han Lee.