

JOINT SYNCHRONIZATION OF CLOCK PHASE OFFSET, SKEW AND DRIFT
IN REFERENCE BROADCAST SYNCHRONIZATION (RBS) PROTOCOL

A Thesis

by

ILKAY SARI

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

MASTER OF SCIENCE

August 2006

Major Subject: Electrical Engineering

JOINT SYNCHRONIZATION OF CLOCK PHASE OFFSET, SKEW AND DRIFT
IN REFERENCE BROADCAST SYNCHRONIZATION (RBS) PROTOCOL

A Thesis

by

ILKAY SARI

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

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Erchin Serpedin
Committee Members,	Jean-Francois Chamberland
	Narasimha Reddy
	Natarajan Gautam
Head of Department,	Costas N. Georghiades

August 2006

Major Subject: Electrical Engineering

ABSTRACT

Joint Synchronization of Clock Phase Offset, Skew and Drift
in Reference Broadcast Synchronization (RBS) Protocol. (August 2006)

Ilkay Sari, B.S., Bilkent University, Ankara, Turkey

Chair of Advisory Committee: Dr. Erchin Serpedin

Time-synchronization in wireless ad-hoc sensor networks is a crucial piece of infrastructure. Thus, it is a fundamental design problem to have a good clock synchronization amongst the nodes of wireless ad-hoc sensor networks. Motivated by this fact, in this thesis, the joint maximum likelihood (JML) estimator for relative clock phase offset and skew under the exponential noise model for the reference broadcast synchronization protocol is formulated and found via a direct algorithm. The Gibbs Sampler is also proposed for joint estimation of relative clock phase offset and skew, and shown to provide superior performance compared to the JML-estimator. Lower and upper bounds for the mean-square errors (MSE) of the JML-estimator and the Gibbs Sampler are introduced in terms of the MSE of the uniform minimum variance unbiased estimator and the conventional best linear unbiased estimator, respectively. The suitability of the Gibbs Sampler for estimating additional unknown parameters is shown by applying it to the problem in which synchronization of clock drift is also needed.

To my Family

ACKNOWLEDGMENTS

First and foremost, I express my sincere gratitude to my advisor Dr. Erchin Serpedin, for his guidance and support throughout the research and his invaluable comments on the thesis. I am also thankful to my committee members, Dr. Jean-Francois Chamberland, Dr. Narasimha Reddy and Dr. Natarajan Gautam for their valuable comments and suggestions. Among my colleagues, special thanks go to Eddie, Yi and Quasim for discussions we had both on research and life. To my friend, Feridun, I thank him for all his efforts to keep me enthusiastic about the research whenever I felt discouraged. To my family, I offer sincere thanks for their continuous support and strong faith in me.

TABLE OF CONTENTS

CHAPTER		Page
I	INTRODUCTION	1
	A. Wireless Sensor Networks	1
	B. Importance of Time-Synchronization in WSN	2
	C. Contributions	3
	D. Outline of the Thesis	4
II	LITERATURE SURVEY AND MOTIVATIONS	6
	A. Definition of Clock	6
	B. Time-Synchronization Protocols	8
	C. Message (Time-Stamp) Exchange and Message Delay	10
	D. Data Filtering Algorithms in Time-Synchronization Protocols	13
	E. Recent Works on Filtering Methods	15
	F. Reference Broadcast Synchronization	17
	G. Characterization of Receive Time in RBS	19
	H. Summary and Problem Statement	21
III	MONTE-CARLO STATISTICAL METHODS	25
	A. Monte-Carlo Integration	25
	B. Indirect Sampling Techniques	26
	C. Gibbs Sampling	26
	D. Final Remarks	27
IV	JOINT SYNCHRONIZATION OF CLOCK PHASE OFF- SET AND SKEW	29
	A. Likelihood Function and Joint Maximum Likelihood Estimation	29
	B. Application of Gibbs Sampling	34
	C. Performance Bounds	35
	1. Lower Bounds	36
	2. Upper Bounds	37
	D. Simulations	38
V	EFFECT OF UNKNOWN NOISE PARAMETERS	45

CHAPTER	Page
A. Joint Maximum Likelihood Estimator	45
B. Application of Gibbs Sampling	46
C. Discussions and Simulations	47
VI JOINT SYNCHRONIZATION OF CLOCK PHASE OFF- SET, SKEW AND DRIFT	49
A. Application of Gibbs Sampling	49
B. Simulations	50
VII SUMMARY	53
A. Conclusions	53
B. Future Research Work	54
REFERENCES	55
APPENDIX A	61
APPENDIX B	63
VITA	64

LIST OF FIGURES

FIGURE		Page
1	Transmission of time-stamps in a sender-to-receiver type of time-synchronization protocol	12
2	Clustering for RBS	18
3	Transmission of time-stamps in RBS	19
4	Characterization of the receive time in RBS	20
5	System model	23
6	\mathbf{S} and the solution s_2 for $N = 4$	31
7	\mathbf{S} and the solution s_2 for $N = 4$ and f_3 inactive	32
8	\mathbf{S} and the solution for $N = 3$	33
9	Gibbs sampling and estimates by BLUE, JML and Gibbs Sampler	40
10	MSE performances of the offset estimators	40
11	MSE performances of the skew estimators	41
12	Bias on offset estimators	41
13	Convergence of the Gibbs Sampler for offset and skew estimates	42
14	Histograms for the offset estimators	43
15	Histograms for the skew estimators	44
16	MSE performances of the offset estimators (with unknown λ_x)	48
17	MSE performances of the drift estimators (with unknown drift)	50
18	MSE performances of the skew estimators (with unknown drift)	51

FIGURE	Page
19	MSE performances of the offset estimators (with unknown drift) . . . 52

CHAPTER I

INTRODUCTION

A. Wireless Sensor Networks

Recently, design of wireless sensor networks (WSN), which can briefly be described as networks of hundreds or thousands of transceivers spread over the area of interest collaboratively conveying the information that they gather from their own neighborhood to a fusion center (sink), has caught a huge attention from research communities with diverse concentrations. One reason for their popularity is that setting up WSN can serve for many different purposes, which are difficult to handle with traditional networks. They can be utilized in battle fields, environmental monitoring, traffic controlling, material testing, health applications, creating smart environments and many other applications. The other reason which makes WSN more and more popular is the strong belief that with the advances in wireless communication techniques and fabrication technologies of micro-electro-mechanical systems (MEMS), deployment of WSN will be cost-wise feasible in the near future.

For successful deployment of WSN, many design factors should be taken into account such as tolerance to node failures, scalability, time-varying network topology, hardware-constraints, production cost and power consumption [1]. Since energy in WSN is non-renewable, the efficient use of energy lie on top of all other design challenges [2]. Consequently, various techniques to reduce the power consumption in WSN have been suggested like application of power-save protocols and relaying. Local processing of the data, before transmitting it to the sink, also appears to be promising for the energy efficiency in WSN, since the energy required to transmit 1

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

bit over 100 meters (3 Joules) is approximately equivalent to the energy required to execute 3 millions of instructions [3].

B. Importance of Time-Synchronization in WSN

Time-synchronization is a crucial element of WSN and can be defined, with general terms, as the task to assure that all nodes in a network have a common reference timescale. The synchronized time in WSN can be necessary for various reasons from well-functioning of the network to cooperative operation of separate nodes.

For a network to successfully operate, it is not always necessary to have a reference timescale. However, if the phenomenon to be detected is time-varying, time-synchronization is a must for the sink to successfully interpret the raw data coming from individual nodes [4]. For example, if the aim of WSN is to track an object, it is impossible to achieve this without having a synchronized time.

The role of time-synchronization in assuring well-functioning of the network might be observed in other distributed systems as well. However, there are some other cases in which the need for time-synchronization is unique to WSN. For example, there are some special techniques like distributive communication protocols and TDMA-like sleep/wake-up MAC protocols which are suggested to comply with the design requirements of WSN. To implement these techniques successfully, the nodes need to have a common timescale [4].

In this section, we pointed out some of the important scenarios where time-synchronization is required for WSN. Elson, in [4], provides a good overview with many other examples and an extensive list of references.

C. Contributions

Although in the previous section we discussed the importance of the time synchronization in WSN, the target in this thesis is more specific. As the title and the abstract partially suggest, we aim to achieve the joint synchronization of the phase offset, skew and drift of the sensor clocks which are using the reference broadcast synchronization (RBS) as the time-synchronization protocol. Estimating the relative clock skew and drift together with the relative phase offset is much more important in WSN, since it guarantees long-term reliability and increases synchronization accuracy.

In a computer network, (since the energy is not limited) the nodes can run the time-synchronization protocol to estimate the relative phase offset whenever it is necessary. In WSN, on the other hand, prediction appears to be a valuable tool for long-term reliability, and hence for energy efficiency (instead of running the time-synchronization protocol every time a relative phase offset estimation is needed). However, to be able to utilize the prediction technique, estimating the relative skew and drift is necessary [5].

Additionally, in some sensor network applications like underwater acoustic networks, a fixed value model for the relative phase offset is not sufficient and such a model decreases the accuracy of the time-synchronization protocol significantly [6]. The only way to increase the accuracy of the time-synchronization protocol is to compensate for the relative skew and drift.

The importance of the estimation of the relative skew together with the estimation of the relative phase offset is also emphasized in [4]. However, in that work, Elson considers a noise model with the normal distribution which seems to be unrealistic for couple of reasons. This is the basic motivation of the thesis and will be further discussed in the following chapter. With this motivation, we formulated the joint

maximum likelihood (JML) estimator for the relative phase offset and skew under the exponential noise model for the reference broadcast synchronization protocol and found the solution via a direct algorithm. We also proposed the Gibbs Sampler to jointly estimate the relative phase offset and skew, and showed that the Gibbs Sampler provides superior performance compared to the JML-estimator. We also introduced lower and upper bounds for the mean-square errors (MSE) of the JML-estimator and the Gibbs Sampler in terms of the MSE of the uniform minimum variance unbiased (UMVU) estimator and the conventional best linear unbiased estimator (BLUE), respectively. Lastly, we used the Gibbs Sampler for the joint estimation of the relative phase offset, skew and drift.

D. Outline of the Thesis

After this brief discussion on *Wireless Sensor Networks* and *Importance of Time-Synchronization in WSN*, in what follows, we will explore the problem of joint synchronization of the clock phase offset, skew and drift in more detail.

In Chapter II, we present the related work, which also serves as the motivation of the problem. In that chapter, we prefer a constructive style, i.e., we build the complete problem step-by-step.

In Chapter III, we explore some tools from statistics such as Monte-Carlo integration, Markov Chain Monte-Carlo sampling techniques (MCMC) and Gibbs sampling, which are utilized in the solution of the problem.

In Chapter IV, we look at the joint synchronization of the clock phase offset and skew only. We benefit from the tools presented in Chapter III to solve this problem. This chapter also includes derivations of performance bounds and simulations regarding the performances of different estimators.

Chapter V considers the effect of unknown noise parameters on the results of the previous chapter.

Chapter VI proposes the Gibbs Sampler for the joint synchronization of the clock phase offset, skew and drift. This chapter shows how flexible the Gibbs Sampler is for estimating additional unknown parameters.

Finally, in Chapter VII, we present a summary of the thesis and conclude the thesis by suggesting possible future works.

CHAPTER II

LITERATURE SURVEY AND MOTIVATIONS

As pointed out before, time-synchronization is the task of assuring that every node in the network has a common reference clock. However, this is a general description and from this statement it is not obvious at all what the difficulties are in achieving this goal. In this chapter, we will elaborate on this issue more and list the motivations which lead to the construction of the problem.

In Section A, we give the definition of a clock in an individual node. This section is followed by a summary on the time-synchronization protocols for distributed systems and more specifically for WSN (Section B). In Section C, we discuss message exchange methods used in different time-synchronization protocols and the delays affecting the performance of these methods. Section D explores basic data-filtering techniques used in time-synchronization protocols. Recent works, which analyze the data-filtering techniques from a statistical point of view, are presented in Section E. This section is followed by a brief summary on the reference broadcast synchronization (RBS) protocol for WSN (Section F) . Section G investigates the statistical characteristics of the receive time uncertainty (delay) in RBS. The last section of this chapter, Section H, is dedicated to the detailed statement of the problem.

A. Definition of Clock

Every individual node in a network has its own clock. The clock consists of hardware and software parts. The hardware part includes an oscillator and a counter. The counter is incremented in accordance with the zero-crossings or the edges of the periodic output signal of the oscillator. When the counter hits a certain threshold value, an interrupt is created. The interrupt triggers the software part to increment

the data, which is kept in the memory representing the clock, by one clock-tick. The frequency of the oscillator and the threshold value determine the resolution of the clock. The software and the hardware parts of the clock are configured depending on the expected frequency of the oscillator such that the clock function of the i^{th} node will be $C_i(t) = t$, with t standing for the ideal or reference time. However, due to nondeterminism in the oscillator's frequency, the clock function is given by

$$C_i(t) = \beta_{i,0} + \beta_{i,1}t + \beta_{i,2}t^2, \quad (2.1)$$

where the parameters $\beta_{i,0}$, $\beta_{i,1}$ and $\beta_{i,2}$ are called clock phase offset, clock skew and clock drift, respectively [7]-[9].

Having the definition of the clock function as in (2.1), the reasons which make the time-synchronization in WSN a problem, can be explored in more details. First of all, the clock parameters are different for different type of oscillators, for example cesium based or crystal-quartz based oscillators have completely different characteristics. Another important point is that for every individual oscillator of the same kind, the parameters can have different values, i.e., they are different realizations of a random variable. For example, the data-sheet of a typical cheap crystal-quartz oscillator used in WSN can contain the range information about the effect of frequency deviations as 40ppm, which means clocks of different nodes can loose as much as 40ms in an ideal second. In other words, after a mass-production cycle, every single oscillator can have a different skew parameter from 0 to 40ppm, independent of each other.

If the whole problem was as explained above, there would be no need to use any complex protocol to solve the time-synchronization problem. Calibrating each node's clock with a reference clock before setting up the network would solve the problem. However, there is another issue. Although the clock phase offset, skew and drift are labeled as parameters, they are indeed time-dependent random processes [10].

There are two terms used in clock terminology regarding time-dependent randomness present in clock parameters, which are *short-term* and *long-term stabilities* [10] and [11]. For crystal-quartz based oscillators which are currently used in sensor networks, all these parameters are almost constant in short term. As an example, the skew is actually governed by $\beta_{i,1}(t) = \beta_{i,1} + e(t)$ for short term. The additive noise term's ($e(t)$) spectral density is a power-law type that is $S_e(f) = 1/f^\alpha$ with $\alpha \geq 1$ [10]. Additionally, the total power in the noise process is too small to be effective in short time-spans [12]. These two factors are valid for all three clock parameters and these are the fundamental reasons which lead to the assumption that the parameters of a clock are unknown constants for time-periods of interest in time-synchronization problems.

As far as the long-term stability is concerned, the parameters are subject to change due to environment or other external effects to the oscillator such as change in temperature, ionizing radiation or power supply voltage [11].

In summary, the discussion presented in this section has revolved around two main points. First of all, the most frequently used model for the clock in the time-synchronization literature is the one given in (2.1). The second concern of this section was to show that it is indeed this model which leads to the periodic application of a protocol to achieve time-synchronization in sensor networks. Since clock parameters are subject to change in long term, they have to be estimated from time to time.

B. Time-Synchronization Protocols

To solve the problem of time-synchronization, many different synchronization methods have been suggested starting with the advent of computer networks. The network time protocol (NTP), as an example, is the standard time-synchronization scheme of

the Internet [13] and [14]. Although the NTP works quite well in computer networks, it is not suitable for WSN. The NTP benefits from practically-unlimited energy resources, use of the Global Positioning System (GPS) and existence of a nearly-static topology in computer networks. However, these benefits are not available in WSN. Because of this reason, many other time-synchronization protocols are designed to accommodate the special needs of WSN, such as the timing-sync protocol for sensor networks (TPSN) [15], the flooding time synchronization protocol (FTSP) [16], the time-diffusion synchronization protocol (TDP) [17], the reference broadcast synchronization protocol (RBS) [18] and many others [8]. Ideally, a time-synchronization protocol for WSN would try to work optimally in terms of all requirements of time-synchronization, which are energy efficiency, scalability, precision, robustness, lifetime, cost and size. However, the complexity of WSN makes it very difficult to optimize the protocol with respect to all requirements at the same time. Instead, in general, some tradeoffs are made. Thus, all the protocols mentioned above put different weights on different requirements. Roughly speaking, this is what makes them different in many ways.

There are many possible categorizations of time-synchronization protocols such as

- ***master-slave*** where first the network is arranged in a tree-like manner, after this arrangement only the connected nodes in the tree synchronize with each other as in TPSN *vs.* ***peer-to-peer*** where any pair of nodes in the network can synchronize with each other as in RBS,
- ***clock correcting*** where the clock function kept in memory is modified after each run of the protocol *vs.* ***untethered clocks*** where every node keeps its own clock as it is, and keeps a time-translation table relating its clock to other

nodes' as in RBS; thus, instead of updating its clock constantly, each node translates the time information in the data packets coming from other nodes to its own clock by using the time-translation table,

- *sender-to-receiver* where one of two nodes, which are synchronizing with each other at the moment, sends a time-stamp while the other one receives it as in TPSN *vs.* *receiver-to-receiver* where a third node transmits synchronization-signals and two synchronizing nodes both receive these signals and record the time of receptions (time-stamps) as in RBS.

The reader is referred to [8] and [9] for additional categorizations of time-synchronization protocols.

C. Message (Time-Stamp) Exchange and Message Delay

There is one common feature of all network time-synchronization protocols. They rely on some sort of message (time-stamp) exchange mechanism between nodes.

Time-stamps are the type of messages used in time-synchronization protocols, they simply have local clock information at the time of recording. Message exchange in time-synchronization protocols is a rather simple process, one node tells the other one its time in a controlled manner just like saying “it’s ten past five according to my clock”. There are two main methods to exchange time-stamps. The first method is called sender-to-receiver and the second one is called receiver-to-receiver.

The main source of difficulty in achieving time-synchronization of two nodes arises from the uncertainty (nondeterminism) of the network delays that are experienced by messages (time-stamps) while traveling from one node to the other. When a node creates a time-stamp and sends it to the other node, the message experiences a variable amount of delay, which prevents the receiver node to directly compare the

incoming time-stamp to its own local clock reading.

The message (packet or network) delay is generally decomposed as *send time*, *access time*, *propagation time* and *receive time*. Since more or less all these time-uncertainties are common to different message exchange ways, we will discuss them using a sender-to-receiver type of message exchange scenario as illustrated in Figure 1.

We will assume that local clock drifts are zero and skews are equal to one in both nodes. Therefore, the only concern is the relative phase offset ($\beta_0 = \beta_{y,0} - \beta_{x,0}$), which is apparently 2 since $C_Y(0) - C_X(0)$ is equal to 2 from Figure 1. Node X wants to know what the relative phase offset (β_0) is. Thus, first, X sends a time-stamp $T_X[1]$ to Y . Then node Y checks its own clock right after it receives the time-stamp from X and records its own clock reading which becomes the time-stamp $T_Y[1]$. If there was no delay experienced by messages, $\beta_0 = T_Y[1] - T_X[1]$ would be what X needed. However, in a real-life, the network delay does exist and is composed of

- ***Send Time*** which is the time elapsed while the message being prepared by the operating system of the sender,
- ***Access Time*** which is the time spent in medium access layer (MAC), while the packet waits for the channel to be available for transmission,
- ***Propagation Time*** which is the time taken for messages to propagate from one node to the other,
- ***Receive Time*** which is the time needed to transport the received time-stamp from the physical layer to the application layer of the receiver [7].

Since all these time uncertainties are certainly positive valued, the most widely used

network delay model is as depicted in the following equations:

$$T_{nd,XY}[i] = d_{XY} + e_{XY}[i], \quad (2.2)$$

$$T_{nd,YX}[i] = d_{YX} + e_{YX}[i], \quad (2.3)$$

where $T_{nd,XY}$ and $T_{nd,YX}$ stand for the aggregate network delays, d_{XY} and d_{YX} are positive unknown constants, and e_{YX} and e_{YX} are additive and positive noise terms for the links $X \rightarrow Y$ and $Y \rightarrow X$, respectively, with $i = 1, 2, \dots, N/2$ [19],[20] and [21].

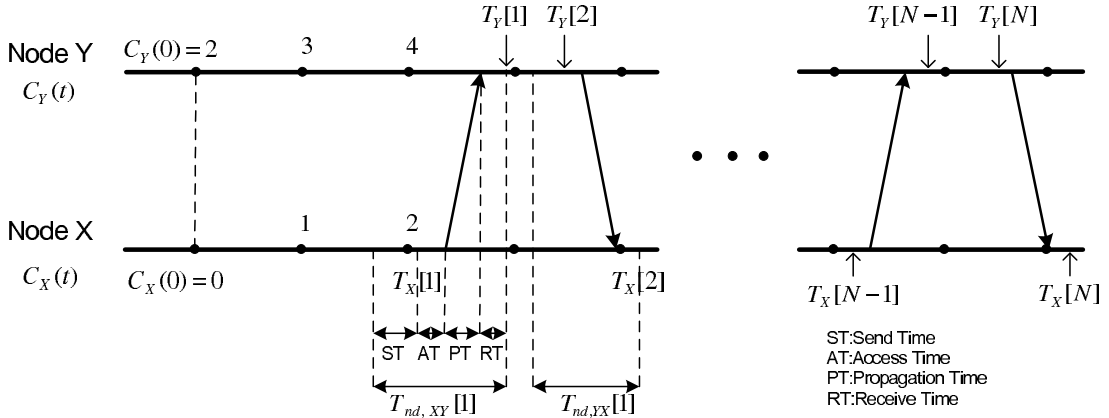


Fig. 1. Transmission of time-stamps in a sender-to-receiver type of time-synchronization protocol.

In Figure 1, X initiates the time-synchronization process by sending a time-stamp ($T_X[1]$) and Y records the time-stamp ($T_Y[1]$) upon receiving $T_X[1]$. From the model in (2.2), with these 2 time-stamps, only $T_Y[1] - T_X[1] = d_{XY} + e_{XY}[1] + \beta_0$ can be formed. At least to be able to eliminate constant parts of the network delays (d_{XY} and d_{YX} , which are mostly considered as being equal to each other), the cycle is completed with the same steps from Y to X (with time-stamps $T_X[2]$ and $T_Y[2]$) as illustrated in Figure 1. Thus, with the completion of the first cycle, X will have 4 time

stamps as $\{T_X[1], T_Y[1], T_X[2], T_Y[2]\}$. This process will be repeated $N/2$ times to be able to average out the random effects of noises (e_{XY} and e_{YX}) from the estimation of the parameters. Therefore, at the end, X will have the data set

$$D = \{T_X[2i - 1], T_Y[2i - 1], T_X[2i], T_Y[2i] : i = 1, 2, \dots, N/2\}. \quad (2.4)$$

In this section, we explored components of the network delay (time uncertainty) with the help of a sender-to-receiver type of message exchange method. The second type of message exchange method, i.e., receiver-to-receiver, will be discussed in Section G.

D. Data Filtering Algorithms in Time-Synchronization Protocols

Once the message exchange process is complete, the next step is to find a good estimator for β_0 , i.e, the relative clock phase offset¹. Methods which are used for this purpose are named as *data-filtering algorithms* and the first step which is common to all is to form an estimation set (E) consisting of the estimates of the relative phase offset and the average network delay for every cycle, by using the data set D from (2.4) as,

$$E = \{(\hat{\beta}_0[i], \hat{d}[i]) : i = 1, 2, \dots, N/2\}, \quad (2.5)$$

¹We still continue to use the example from Figure 1. Hence, β_0 is still the only unknown parameter to be estimated and notations regarding the time-stamps in the following equations still refer to that example.

where $\hat{\beta}_0[i]$ and $\hat{d}[i]$ stand for the relative phase offset and the average network delay estimates in the i^{th} cycle, respectively, and they are equal to

$$\begin{aligned}\hat{\beta}_0[i] &= \frac{(T_Y[2i-1] - T_X[2i-1])}{2} - \frac{(T_X[2i] - T_Y[2i])}{2} \\ &= \frac{d_{XY} + \beta_0 + e_{XY}[i]}{2} - \frac{d_{YX} - \beta_0 + e_{YX}[i]}{2} \\ &= \beta_0 + \frac{d_{XY} - d_{YX}}{2} + \frac{e_{XY}[i] - e_{YX}[i]}{2},\end{aligned}\quad (2.6)$$

and

$$\begin{aligned}\hat{d}[i] &= \frac{(T_Y[2i-1] - T_X[2i-1])}{2} + \frac{(T_X[2i] - T_Y[2i])}{2} \\ &= \frac{d_{YX} + d_{XY}}{2} + \frac{e_{YX}[i] + e_{XY}[i]}{2},\end{aligned}\quad (2.7)$$

respectively [19].

Data-filtering algorithms differ in the way they form the final estimate of the relative phase offset. Some of the most frequently referred ones are the median round delay algorithm (MdRD), the minimum round delay algorithm (MnRD) which is also used in the NTP, and the minimum link delay algorithm (MnLD) [19]. The ways these algorithms form their relative phase offset estimators ($\hat{\beta}_0$) are as follows;

- **MdRD**: $\hat{\beta}_0 = \hat{\beta}_0[k]$ is used as the final relative phase offset estimator, where k is the index of the median of the set $\{\hat{d}[i] : i = 1, 2, \dots, N/2\}$, i.e., $k = \#med_{<i>}(\hat{d}[i])$,
- **MnRD**: $\hat{\beta}_0 = \hat{\beta}_0[k]$ is used as the final relative phase offset estimator, where k is the index of the minimum of the set $\{\hat{d}[i] : i = 1, 2, \dots, N/2\}$, i.e., $k = \#min_{<i>}(\hat{d}[i]) = \#\hat{d}(1)$,
- **MnLD**: First, $U[i] = T_Y[2i-1] - T_X[2i-1]$ and $V[i] = T_X[2i] - T_Y[2i]$ are

formed, then

$$\begin{aligned}\hat{\beta}_0 &= \frac{\min_{\langle i \rangle}(U[i]) - \min_{\langle i \rangle}(V[i])}{2} = \frac{U(1) - V(1)}{2} \\ &= \beta_0 + \frac{d_{XY} - d_{YX}}{2} + \frac{e_{XY}(1) - e_{YX}(1)}{2}\end{aligned}\quad (2.8)$$

is used as the final relative phase offset estimator².

The reason behind the use of the MnRD algorithm in the NTP can be explained with the dynamics of computer networks. Since, in the Internet, it is very hard to characterize the probabilistic properties of the random part of the network delay (noise terms from the model given in (2.2) and (2.3)), the sample with the minimum error terms are selected for estimation [13]. The MnRD algorithm is also shown to be superior, through experiments using real-life data in [13] and references therein. However, Paxson, in [22] and [23], observed that the MnLD algorithm produced more accurate results for the relative phase offset estimation compared to the MnRD algorithm. Since the main topic of Paxson's works is not time-synchronization, he argues this observation informally. He argues that formation of $\hat{\beta}_0[i]$ as $(U[i] - V[i])/2$ causes information loss. Since the noise terms in $U[i]$ and $V[i]$ are definitely positive-valued, using $U(1)$ and $V(1)$ to form the relative phase offset estimator further reduces the effect of the noise.

E. Recent Works on Filtering Methods

Paxson's arguments on the superiority of the MnLD algorithm are reasonable. However, they are not complete and satisfactory enough, because it is possible to find a probability density function for the noises such that the variance of the random

²In theory of order statistics, parentheses (.) are used to index the set after ascending ordering [24]. For regular (or time-domain) indexing, square brackets [.] are used.

variable created by $(e_{XY}(1) - e_{YX}(1))/2$ from (2.8) (MnLD) will be higher than that of $(e_{XY}[k] - e_{YX}[k])/2$ from (2.6) (MnRD). This will be the case, as an example, when $\min(\cdot)$ of a random process is a rare event and the density is locally flat around $\min(\cdot)$. Thus, more detailed and rigorous arguments explaining the superiority of the MnLD algorithm over the MnRD algorithm is needed.

The first comprehensive study on statistical analysis of different ad-hoc data-filtering algorithms is presented by Abdel-Ghaffar in [19]. In his work, Abdel-Ghaffar assumes that links $X \rightarrow Y$ and $Y \rightarrow X$ are symmetric that is $d_{XY} = d_{YX} = d$ and $f_{e_{XY}} = f_{e_{YX}}$ with f standing for the probability density function. Additionally, Abdel-Ghaffar assumes that d is known and the noise distributions are independent and identically (*i.i.d*) distributed exponentials. Under these assumptions, Abdel-Ghaffar shows that the MnLD algorithm has a better mean square error (MSE) performance compared to the other ad-hoc algorithms both through mathematical analysis and simulations. However, he concludes that a unique maximum likelihood estimator (MLE) for this problem set-up does not exist.

Jeske, in [20], uses the same problem set-up as Abdel-Ghaffar. He, on the other hand, assumes that d is unknown. Under these assumptions, he shows that the MnLD algorithm is actually the MLE. Jeske also notes that if the noises are not symmetric, that is if exponential distributions have different mean-parameters, the MnLD algorithm is still the MLE but not unbiased anymore. In addition to this contribution, Jeske, in [21], successfully applies *bootstrapping* to reduce the bias created due to the model with asymmetric noise distributions.

F. Reference Broadcast Synchronization

All of the works mentioned in the previous section use a sender-to-receiver type of set-up as illustrated in Figure 1. Although those discussions will definitely prove to be very useful, the main problem considered in this thesis assumes a receiver-to-receiver type of protocol, namely RBS. Thus, it will be beneficial at this point to discuss some aspects of RBS.

RBS is a recently proposed receiver-to-receiver time-synchronization protocol for wireless sensor networks [18]. RBS exploits the broadcast nature of WSN. The first step, in RBS, is partitioning of the sensor nodes into clusters as in Figure 2. Every cluster has a server node called transmitter as shown in Figure 2. Clustering and choice of transmitters is dynamically handled to provide an energy efficient implementation. Roughly speaking, in each cluster the transmitter broadcasts N synchronization signals and the receivers record time-stamps using their local clocks. In other words, an individual transmitter of a cluster does not produce time-stamps, but it triggers all the receivers inside its broadcast domain to produce time-stamps by broadcasting synchronization signals. Then, any pair of receivers can exchange the time-stamps to estimate the relative clock parameters. However, the pair-wise synchronization can create a considerable amount of overhead and can lead to more power consumption. Hence, for a more efficient implementation, the receivers pass the data, consisting of N time-stamps, back to the transmitter where the relative clock parameters between all pairs of receivers are calculated, and this information is broadcasted back to the receivers as a table. This process describes the time-synchronization procedure in the broadcast domain of a single transmitter, i.e., inside a cluster only. Time-synchronization between different clusters in RBS is achieved simply by exchanging the time-translation tables.

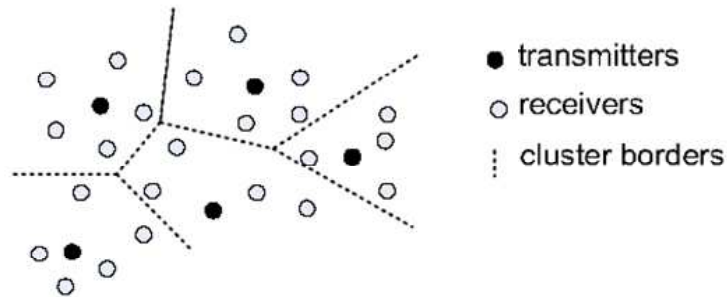


Fig. 2. Clustering for RBS.

One advantage of RBS is that by using N synchronization signals all receivers inside a cluster can be synchronized. The other one is that by the help of RBS, two (even three) of the main error sources of clock synchronization are eliminated, which are uncertainties at *send time* and *access time* and partially *propagation time*. Closer examination of Figure 3 will make this point clearer. Since the transmitter does not produce time-stamps (only receivers produce time-stamps upon receiving a synchronization signal), there is neither send nor access time uncertainty in final time-stamps. From the time that a synchronization signal is broadcasted to the time that the time-stamps are recorded in receivers, the message experiences propagation and receive delays only. Furthermore, the difference between propagation times for different receivers in a cluster is negligible compared to the uncertainty at the receive time, which becomes the only error source. Since the total propagation time inside a cluster is negligible compared to the receive delay, the difference between propagation times of two different receivers will be even smaller [18].

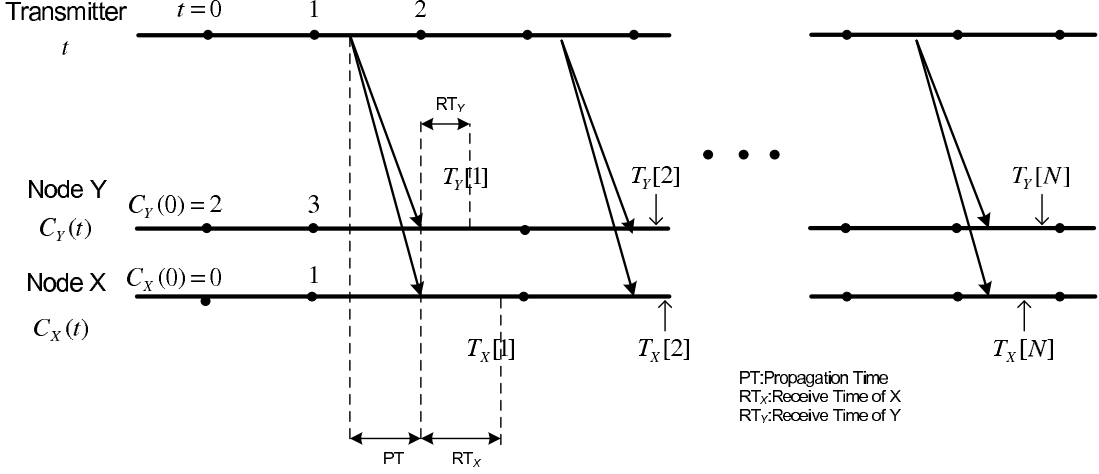


Fig. 3. Transmission of time-stamps in RBS.

G. Characterization of Receive Time in RBS

In [18], it is experimentally argued that uncertainty at the receive time can be modeled with a normal distribution. In the experiment, there are 5 receiver nodes and the transmitter sends 160 broadcast signals in the course of 3 minutes. The receivers are connected to an external logic analyzer which creates time-stamps. The receiver nodes are programmed such that as soon as receiving a signal they produce an interrupt for the logic analyzer. This experiment is also illustrated in Figure 4.

The logic analyzer creates 160 time-stamps for each receiver. Then, the differences between time-stamps for each possible pair of receivers is calculated and the resulting 1600 relative receive time error terms are plotted as a histogram (Figure 2 in [18]), which can be approximated with a normal distribution quite well.

Starting from this observation that the receive time is normal, the authors use the mean estimator ($\hat{\beta}_0 = 1/N \sum_{\langle i \rangle} (T_Y[i] - T_X[i])$) to time-synchronize the receivers,

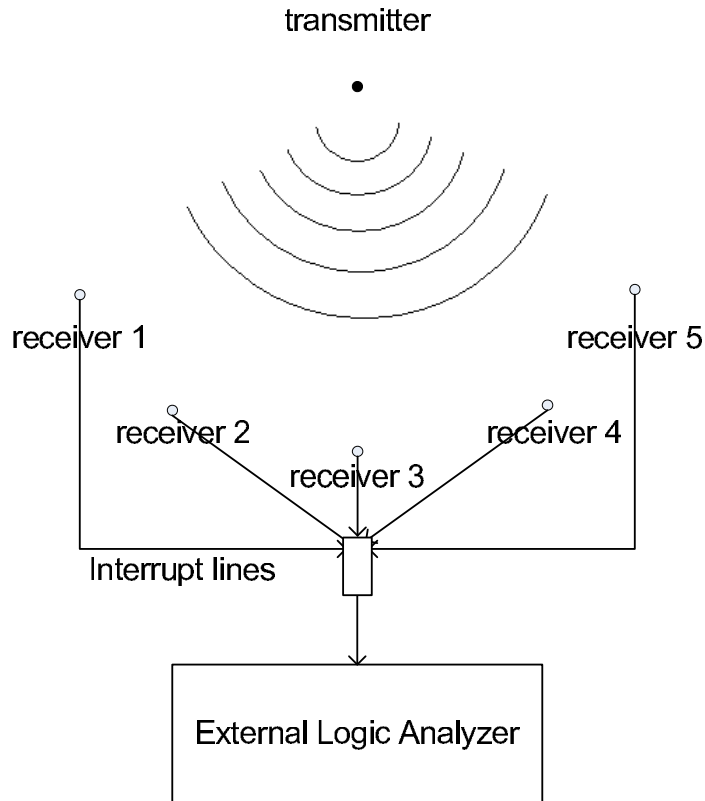


Fig. 4. Characterization of the receive time in RBS from [18].

since it is optimal in additive Gaussian noise.

However, several objections can be raised over the competency of this experiment in characterizing the receive time uncertainty in RBS. First of all, the mean estimator is optimal only if the receive time uncertainties for individual receivers are normally distributed. However, the experiment characterizes only the difference between receive time uncertainties, not the individual receive time uncertainties. Fundamentally, individual receive time uncertainties are positive-valued; hence, they cannot have a normal distribution, although their difference might look like one.

Secondly, the experimental set-up is idealistic. The receivers just wait for synchronization signals, and they do not have any other duty. They even do not process the packets after reception. They just inform the logic analyzer about the reception so that the logic analyzer will record time-stamps. It is doubtful that such an experimental set-up will successfully characterize the receive time uncertainty.

Modeling of network delays in WSN, using a closed-form expression, seems to be a challenging task [25]. On the other hand, the exponential, Gamma, log-normal and Weibull distributions have been shown to be feasible for different cases [26]. Amongst them, the exponential distribution fits quite well several applications [27]. One of these applications is point-to-point wireless links with a single server [19], [21] and [28]. It was also the exponential network delay assumption enabling the mathematical proof of Paxson's observation on the superiority of the MnLD algorithm as mentioned in Section E [20]. These arguments advocate the use of exponential distribution for the receive time uncertainty.

H. Summary and Problem Statement

To summarize the analysis and the survey which have been presented up to this point, it can be said that for time-synchronization in sender-to-receiver type of protocols, rigorous statistical analysis led to substantial performance gains [19]-[23]. It is also noted that RBS, because of the broadcast nature of WSN, is a convenient time-synchronization protocol for WSN and it provides considerable noise power reduction by eliminating the send and access time uncertainties. However, the network delay assumption which was found through an idealized experiment in [18] seems to be unrealistic. In real-life, there will be communication signals going around, not just synchronization signals, and nodes will have other jobs, not just the time-stamping.

Therefore, the link from the transmitter to an individual receiver is expected to behave as a regular point-to-point network link (M/M/1), where the receive time uncertainty has exponential distribution [19] and [28]. (In some texts, this type of uncertainty is also named as the processing delay.) It is also emphasized that in addition to relative phase offset, estimation of relative skew and relative drift is much more important in WSN compared to traditional computer networks because of resulting energy-save. Considering all these points, the problem can be expressed as how to achieve the joint synchronization of clock phase offset, skew and drift in RBS under the exponential receive time uncertainty model. Mathematically speaking, the problem is to estimate the relative phase offset, skew and drift between the clocks of nodes X and Y , which are given by

$$\beta_0 = \beta_{y,0} - \beta_{x,0} , \quad (2.9)$$

$$\beta_1 = \beta_{y,1} - \beta_{x,1} , \quad (2.10)$$

$$\beta_2 = \beta_{y,2} - \beta_{x,2} , \quad (2.11)$$

using N time-stamps from each node, of which the i^{th} ones are given by

$$T_X[i] = T_1 + \beta_{x,0} + \beta_{x,1}\tau[i] + \beta_{x,2}\tau^2[i] + v_{x,\lambda_x}[i], \quad (2.12)$$

$$T_Y[i] = T_1 + \beta_{y,0} + \beta_{y,1}\tau[i] + \beta_{y,2}\tau^2[i] + v_{y,\lambda_y}[i], \quad (2.13)$$

where T_1 stands for the time on the transmitter when it sends the first synchronization signal, $\beta_{x,0}$, $\beta_{x,1}$ and $\beta_{x,2}$ stand for the phase offset, skew and drift between the clocks of X and the transmitter, $\tau[i]$ stands for the difference between T_1 and the time of i^{th} synchronization signal (with respect to the transmitter's clock) and $v_{x,\lambda_x}[i]$ stands for the additive *i.i.d* exponential noise (with mean $1/\lambda_x$), with $i = 1, \dots, N$. (Similar descriptions also apply for the terms with subscript Y , see Figure 5.)

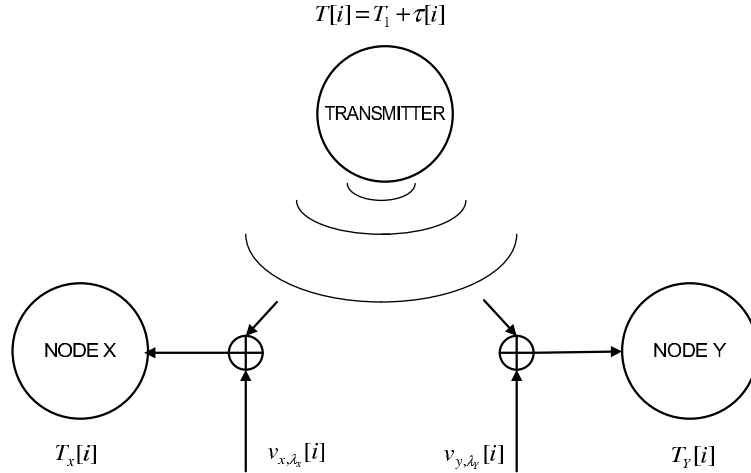


Fig. 5. System model.

The problem formulation given above includes some implicit assumptions. We will conclude this chapter by addressing them.

First of all, the model assumes that constant parts of the network receive delays are known and both are equal to zero ($d_{XY} = d_{YX} = d = 0$). This is a reasonable assumption as explained before. Since the receiver nodes are in the broadcast domain of the transmitter, the propagation time, which constitutes the constant part of the network delay, is negligible compared to receive time uncertainty. Even if d is unknown, it would not have any effect on the final result due to (2.9). If, on the other hand, d_{XY} and d_{YX} are not equal and their difference is a non-negligible unknown constant, the problem of estimating relative phase offset becomes unidentifiable. The difference $d_{YX} - d_{XY}$ adds up to β_0 and this produces a new relative offset $\beta'_0 = \beta_0 + d_{YX} - d_{XY}$.

The other assumption is that during the broadcast of N synchronization signals, the noise is stationary, i.e., λ_x and λ_y stay the same. Extensive studies on the Internet show that delay parameters are approximately stationary for 30 to 60 minutes [29].

This time-duration is much longer compared to the time typically taken to broadcast N synchronization signals.

CHAPTER III

MONTE-CARLO STATISTICAL METHODS

To solve the problem stated in the previous chapter, we will first consider the MLE as in [20], and it will turn out that although a unique MLE exists, it is not a simple task to express it in a closed-form. In such cases, one must resort to more convenient tools. For example, Jeske shows that when the assumption of symmetry on the network delays is dropped, a better solution is achieved by *bootstrapping* [21]. These reasons and successful application of it to many challenging problems of communication theory as in [30], suggest the use of Markov Chain Monte-Carlo (MCMC) techniques. For this reason, in this chapter, we will give a brief description of MCMC techniques and in particular the Gibbs Sampler. The tools discussed in this chapter will be applied to the problem in the following chapters.

A. Monte-Carlo Integration

We will start with the description of a generic inference problem. Assuming that the data vector \mathbf{z} is observable, it is of interest to estimate the parameters $\Phi = [\phi_1, \phi_2, \dots, \phi_M]^T$. For any kind of statistical inference, the joint posterior distribution of the parameters $p(\Phi|\mathbf{z}) \propto p(\mathbf{z}|\Phi)p(\Phi)$ can be used. For inference problems, most of the time, the evaluation of mathematical expressions like $\gamma = \int f(\Phi)p(\Phi|\mathbf{z})d\Phi$, $p(\phi_1|\mathbf{z}) = \int_{\langle \phi_2, \dots, \phi_M \rangle} p(\Phi|\mathbf{z})d\phi_2 \dots d\phi_M$ or $\gamma = \max_{\phi_1} p(\phi_1|\mathbf{z})$ produces reasonable estimators [31]. When it is hard to carry out this type of mathematical derivations on the posterior distribution, Monte-Carlo methods are used, i.e., to draw as many samples as possible from the posterior distribution so that the inference, which is made using these samples, will be arbitrarily close to the exact solution. For example, the previous integration can be approximated by $\hat{\gamma} = (1/N) \sum_{i=\langle 1, N \rangle} f(\Phi^i)$ where Φ^1, \dots, Φ^N

are *i.i.d* samples from $p(\Phi|\mathbf{z})$. Furthermore, this structure allows evaluating the accuracy of the approximation for any fixed N with $\text{var}(\hat{\gamma}) = (1/N)\text{var}(f(\Phi))$ [32]. This also means that once a way to sample from $p(\Phi|\mathbf{z})$ is found, the accuracy of the approximate estimation method is controllable.

B. Indirect Sampling Techniques

Most of the time, $p(\Phi|\mathbf{z})$ is not a distribution readily available to draw samples from. For this kind of situations, there are some sampling techniques which enable sampling from the target distribution indirectly. Some of the most frequently used ones are *Importance Sampling*, *Rejection Sampling* and *Weighted Bootstrap* [31], [32] and [33]. These techniques are non-iterative and fast. Most of them need an auxiliary distribution to draw from, which is similar to the target distribution in some sense. However, these non-iterative methods are not good and efficient enough for high dimensional parameter spaces since it is very hard to find a good auxiliary distribution for high dimensional parameter spaces. For such cases, Markov Chain Monte-Carlo (MCMC) type iterative sampling techniques are used. That resumes to setting up a Markov chain whose stationary distribution is the target joint posterior. The most well known method amongst these techniques is called *Metropolis-Hastings Algorithm*, in which the rejection sampling is used iteratively and adaptively [34] and [35].

C. Gibbs Sampling

One of the most convenient forms of the MCMC techniques is called Gibbs sampling, which was first suggested by Geman and Geman in [36]. In Gibbs sampling, the samples are iteratively drawn from one-dimensional conditionals $p(\phi_i|\mathbf{z}, \overline{\Phi}_i)$, where $\overline{\Phi}_i$ is a vector of dimensions $(M - 1) \times 1$ with entries $\{\phi_j\}_{j \neq i}$ [37]. Under mild

conditions, these one dimensional conditional distributions uniquely determine the joint posterior distribution [38].

Specifically, the general algorithm for Gibbs sampling with initial values $\Phi^{(0)} = [\phi_1^{(0)}, \dots, \phi_M^{(0)}]$ is to iterate the following steps:

- Draw $\phi_1^{(k+1)}$ from $p(\phi_1 | \mathbf{z}, \phi_2^{(k)}, \dots, \phi_M^{(k)})$
- Draw $\phi_2^{(k+1)}$ from $p(\phi_2 | \mathbf{z}, \phi_1^{(k+1)}, \phi_3^{(k)}, \dots, \phi_M^{(k)})$
- \vdots
- Draw $\phi_M^{(k+1)}$ from $p(\phi_M | \mathbf{z}, \phi_1^{(k+1)}, \dots, \phi_{M-1}^{(k+1)})$.

After a threshold value t , the set $\{\Phi^{(t)}, \Phi^{(t+1)}, \dots\}$ behaves like samples from the joint posterior distribution of the parameters. Then this set can be used to form an estimator such as the sample median, the sample mean and so on [37].

One important point is that the joint posterior distribution should be proper. That is it should be positive-valued and integrable. Otherwise, the Gibbs Sampler always converges to some local points, but not necessarily to a meaningful one [39].

D. Final Remarks

MCMC techniques are generally discussed in the Bayesian framework. However, we do not have any prior information regarding the clock parameters; thus, we will perform frequentist estimation, i.e., we will only utilize the likelihood function. When MCMC tools are used for frequentist estimation only, the priors are chosen as uninformative and flat. When the prior distribution is chosen as flat, the posterior distribution becomes equivalent to the likelihood function. However, if the likelihood function in the parameter space is not proper, it cannot be used as the posterior distribution directly. The prior distribution should be chosen in a way that the resulting

posterior distribution is proper while the prior itself is uninformative (frequentist). In the literature, there are many different prior distributions suggested to serve this purpose such as conjugate priors. Conjugate priors are beneficial because the posterior distribution assumes the same form as the prior distribution [40]. For the problem considered in the thesis, the likelihood function does have a bounded integral over the domain, i.e., it is proper. Therefore, the prior distribution can safely be taken as flat.

Another important concept in MCMC techniques is the convergence diagnostics [33]. The issue is to be able to determine if the chain has forgotten its initial point and converged to its stationary distribution. Methods to test the convergence generally use statistical similarity tests performed on a couple of parallel chains. The other method is to plot the estimator acquired from the chain, as the chain moves, and to observe if it converges to the true value or not. The latter is generally preferred in communication applications [30].

CHAPTER IV

JOINT SYNCHRONIZATION OF CLOCK PHASE OFFSET AND SKEW

In this chapter, we consider the synchronization of clock phase offset and skew with known network delay characteristics. In the first section of the chapter, we take a closer look at the likelihood function and suggest two algorithms to find the solution which maximizes the likelihood function. In the second section, the Gibbs Sampler is used as an estimator, we also present the motivations that lead us to use this method. In Section C, the performance bounds are derived. Finally, the chapter is concluded with simulation results.

A. Likelihood Function and Joint Maximum Likelihood Estimation

Although important, the joint estimation of relative skew and phase offset might not be easy. Though in a different context, in [41], Sadler shows that under uniform noise, there are infinite solutions for the MLE. Furthermore, the support of the likelihood function is not convex which leaves out the possibility of taking the mean of all equally likely solutions. Turning back to the problem of this thesis, we will consider the case described in (2.9)-(2.13). We will assume that the parameter (β_2) of equation (2.11), is known and equal to zero. We will also assume that λ_x and λ_y are known. As long as the two parameter sets $\{\beta_{x,0}, \beta_{x,1}, \lambda_x\}$ and $\{\beta_{y,0}, \beta_{y,1}, \lambda_y\}$ do not have a direct relationship and the noise sources in different nodes (v_{x,λ_x} and v_{y,λ_y}) are independent (both of which are realistic assumptions), the joint maximum likelihood estimator (JML) for relative offset and skew (β_0 and β_1) can be found, without loss of any information, by first estimating the parameters $(\beta_{x,0}, \beta_{x,1})$ and $(\beta_{y,0}, \beta_{y,1})$ separately and then plugging these estimates back into (2.9) and (2.10). Thus, from now on we will concentrate on the estimation of $\beta_{x,0}$ and $\beta_{x,1}$ only. (For a brief discussion on the

possible performance loss due to application of the transformation $(T_{\mathbf{Y}} - T_{\mathbf{X}})$ before estimation, as it is done in [18], please see the Appendix A.)

First of all, for simplicity, we will assume that T_1 is zero and $\tau[i] = i - 1$. Under these assumptions the likelihood function becomes

$$\begin{aligned} L(\beta_{x,0}, \beta_{x,1}) &= \prod_{i=1}^N \lambda_x e^{-\lambda_x (T_X[i] - (\beta_{x,0} + (i-1)\beta_{x,1}))} I(T_X[i] \geq \beta_{x,0} + (i-1)\beta_{x,1}) \\ &= \lambda_x^N e^{-\lambda_x N (\bar{T}_{\mathbf{X}} - (\beta_{x,0} + \frac{N-1}{2}\beta_{x,1}))} \prod_{i=1}^N I(T_X[i] \geq \beta_{x,0} + (i-1)\beta_{x,1}) \end{aligned} \quad (4.1)$$

$$= \lambda_x^N e^{-\lambda_x N (\bar{T}_{\mathbf{X}} - f)} \prod_{i=1}^N I(T_X[i] \geq f_i), \quad (4.2)$$

where $f(\beta_{x,0}, \beta_{x,1}) = \beta_{x,0} + ((N-1)/2)\beta_{x,1}$ and $f_i(\beta_{x,0}, \beta_{x,1}) = \beta_{x,0} + (i-1)\beta_{x,1}$, $\bar{T}_{\mathbf{X}}$ stands for the sample mean of observations $T_X[i]$ ($i = 1, \dots, N$), and $I(\cdot)$ denotes the indicator function. Note that in (4.2), the multiplication of indicator functions defines a convex region (\mathbf{S}) on the parameter space $(\beta_{x,0}, \beta_{x,1})$, with $\mathbf{S} = \{(\beta_{x,0}, \beta_{x,1}) : \bigcap_{i=1}^N f_i(\beta_{x,0}, \beta_{x,1}) \leq T_X[i]\}$, which is the domain of the likelihood function. \mathbf{S} has k vertices $\{s_j\}_{j=1}^k$ and $k+1$ edges ($1 \leq k \leq N-1$). Specifically, the shape of this region and the value of k will strongly depend on the ordering of $T_X[1], \dots, T_X[N]$. To maximize the likelihood function over the convex set (\mathbf{S}) , we have to maximize the objective function, $f(\beta_{x,0}, \beta_{x,1}) = \beta_{x,0} + ((N-1)/2)\beta_{x,1}$, which is also convex. In that case, the maximum is achieved on the boundary of (\mathbf{S}) [42]. Because of the inequality $0 < ((N-1)/2) < N-1$, the support of the solution is guaranteed to be a closed-convex region on the boundary of \mathbf{S} .

In general, if $N = 2m$, the solution will be one of the vertices s_j . If $N = 2m - 1$ the solution will assume possibly a segment of the line $f_m : \beta_{x,0} + (m-1)\beta_{x,1}$ (or one of the vertices s_j , depending on the ordering of the observations). For $N = 2m - 1$, the slope of the objective function is aligned with the line f_m . If f_m is active, i.e.,

it is one of the lines constituting the borders of \mathbf{S} , the line segment, which is the intersection of f_m and \mathbf{S} , will be the JML-estimator.

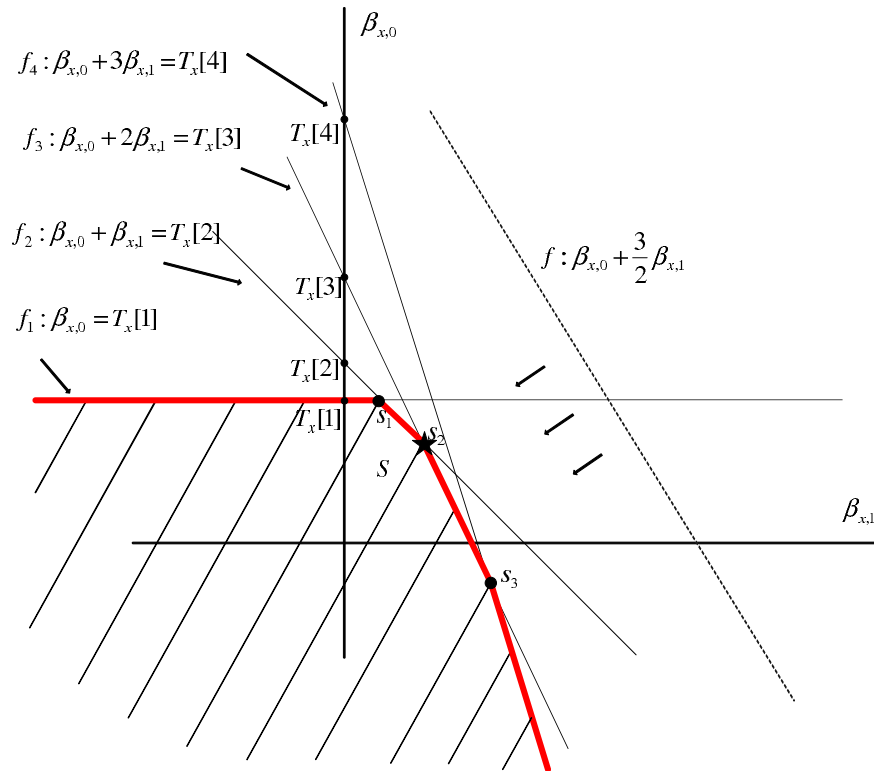


Fig. 6. \mathbf{S} and the solution s_2 for $N = 4$.

Figure 6 illustrates these remarks for $N = 4$. In this illustrative example, since f attains its maximum on s_2 amongst all points on \mathbf{S} , s_2 gives the JML-estimate of $\beta_{x,0}$ and $\beta_{x,1}$. However, there are different possibilities for the same scenario. For example, in Figure 7, N is still equal to 4, but this time f_3 becomes inactive, i.e., f_3 has no effect on the shape of the domain \mathbf{S} , since $T_X[3]$ is too noisy. The domain of likelihood function completely changes compared to the previous example. However, since N is even, there exists no f_i matching the slope of f and the solution is guaranteed to be a unique point.

Lastly, Figure 8 illustrates all possibilities for \mathbf{S} with $N = 3$ (odd). There are two

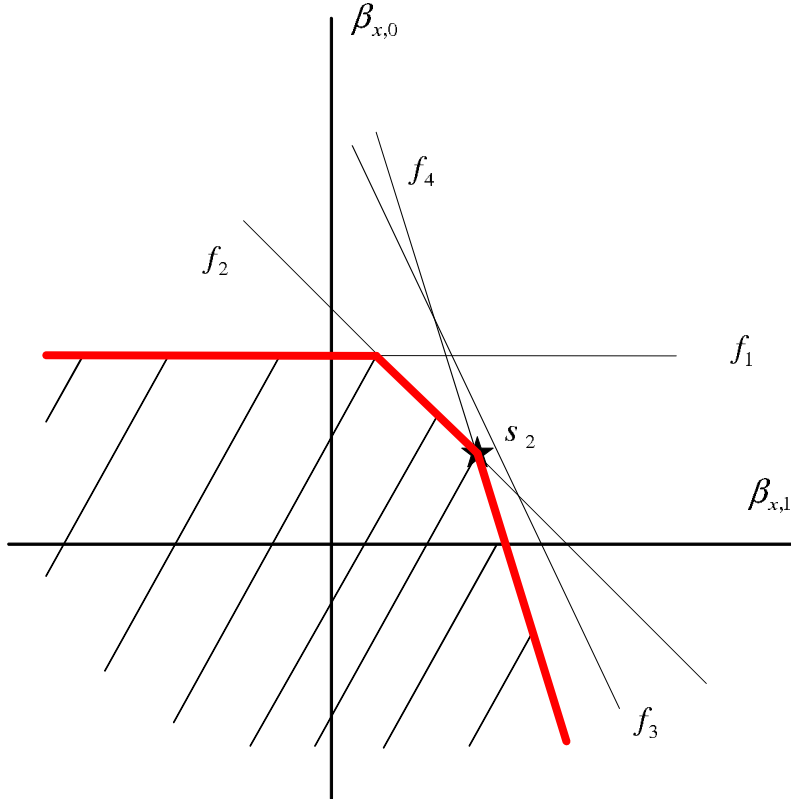


Fig. 7. \mathbf{S} and the solution s_2 for $N = 4$ and f_3 inactive.

different cases which is either f_2 active as in Figure 8.a, or inactive as in Figure 8.b. In Figure 8.a, the solution is a complete line segment that is the part of f_2 inside the domain of the likelihood function. In Figure 8.b, on the other hand, the solution is the point s_1 . As mentioned before, for odd-valued N , the solution depends on the situation whether the line whose slope match that of f is active or inactive. If it is active then the solution is a line segment, otherwise a unique point.

While f_1 and f_N are always active in the formation of (\mathbf{S}) , the remaining $N - 2$ lines could be active or inactive. In other words, there are 2^{N-2} possibilities for the shape of \mathbf{S} . Therefore, it seems difficult to come up with a closed-form expression for the JML-estimator. However, it is possible to set up different algorithms to find the JML-estimate. One way to find the JML-estimate would be to check all possibilities.

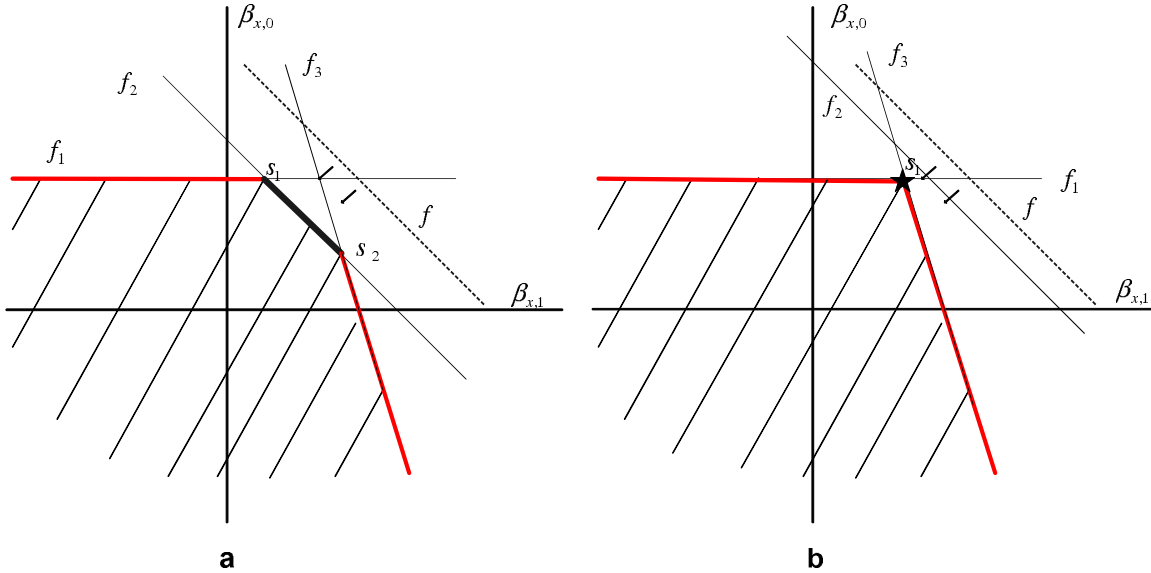


Fig. 8. \mathbf{S} and the solution for $N = 3$: **a)** f_2 active, **b)** f_2 inactive.

When $N = 2m$, for example, we can find all $N(N - 1)/2$ intersection points of the lines $f_i = T_X[i]$, and set active those of which are on \mathbf{S} . Then, amongst these active points we can choose the one that maximizes f .

It is also possible to find computationally less demanding algorithms. As an example, assume that $N = 10$. First, we can find the 9 intersection points of all the other lines with $f_1 = T_X[1]$. Amongst these points, we will choose the one with minimum $\beta_{x,1}$ and record it to a list. Assume that this point is the intersection of the lines $f_5 = T_X[5]$ and $f_1 = T_X[1]$ in a specific scenario, then we can discard the lines $f_6 = T_X[6], \dots, f_{10} = T_X[10]$ and the intersection points defined by these lines beforehand. Then, we need to apply the same step as above to $f_5 = T_X[5]$ and $f_2 = T_X[2], \dots, f_4 = T_X[4]$, and again record the intersection point with minimum $\beta_{x,1}$ amongst these 3 points to the list. At the very end, we will choose the median of the elements present in the list which will give the JML-estimate.

In the first method, we need to find all $N(N - 1)/2$ intersection points first.

However, in the second method, we start with finding $N - 1$ of them, then in each iteration we have the chance to discard some or all of the intersection points without calculating them, which will reduce the computational load.

B. Application of Gibbs Sampling

In this section we will apply the Gibbs Sampler to jointly estimate the parameters $(\beta_{x,0}$ and $\beta_{x,1})$. The Gibbs Sampler can be used to find an approximate JML-estimate which is arbitrarily close to the exact one. However, there are some more important benefits of using the Gibbs Sampler. First of all, it can be shown that the JML-estimator $(\hat{\beta}_{(x,0),JML}, \hat{\beta}_{(x,1),JML})$ is biased for finite N . (As an example consider the case with $N = 2$, $E[\hat{\beta}_{(x,0),JML}] = E[T_X[1]] = E[\beta_{x,0} + v_{x,\lambda_x}[1]] = \beta_{x,0} + 1/\lambda_x$.) For this reason, we need to look for the uniform minimum variance unbiased (UMVU) estimator. To find the UMVU estimator, the Lehmann-Scheffe theorem can be used, which requires to find sufficient statistics first. The Neyman-Fisher factorization theorem provides $\min_i(T_X[i] - \beta_{x,0} + (i - 1)\beta_{x,1})$ as sufficient statistics, which is not independent of the parameters to be estimated. Thus, it seems even harder to find the UMVU estimator. On the other hand, if we use the Gibbs Sampler; at the end, we do not have just a single point estimate but the posterior distribution for the parameters to be estimated as the output. Then, we can either find the JML-estimate or set the corresponding estimator as the mean value of the posterior distribution of the parameter, which will automatically perform the marginalization and will give better results with reduced bias and variance. In this work, we will follow this approach, i.e., using the mean value of the samples of the parameters as the estimator. Another appealing feature of the Gibbs Sampler is its straightforward extendability for estimating additional unknown parameters. For example, it is possible that λ_x is unknown (a set-up that

will be considered in the next chapter) or in addition to the clock phase offset and skew, we could have clock drifts: $\beta_{x,2}$ and $\beta_{y,2}$ (which will be considered in Chapter VI).

As mentioned before, it is important to make sure that the joint posterior distribution used for inference is proper. However, for the problem of this thesis, the likelihood function itself can be used as posterior distribution, since its integral is always bounded and positive-valued which makes it proper. Then, using (4.1), the procedure becomes

- Draw $\beta_{x,0}^{(k+1)}$ from $\propto e^{\lambda_x N \beta_{x,0}} I(\beta_{x,0} \leq \min_i (T_X[i] - (i-1)\beta_{x,1}^{(k)}))$
- Draw $\beta_{x,1}^{(k+1)}$ from $\propto e^{\lambda_x \frac{N(N-1)}{2} \beta_{x,1}} I(\beta_{x,1} \leq \min_i (\frac{T_X[i] - \beta_{x,0}^{(k+1)}}{i-1}))$.

For $\beta_{x,0}^{(k+1)}$, we will draw a sample from an exponential distribution with parameter $\lambda_x N$, multiply it with -1 and add $\min_i (T_X[i] - (i-1)\beta_{x,1}^{(k)})$ to it. The procedure for $\beta_{x,1}^{(k+1)}$ will be similar. We will draw a sample from an exponential distribution with parameter $\lambda_x N(N-1)/2$, multiply it with -1 and add $\min_i ((T_X[i] - \beta_{x,0}^{(k+1)})/(i-1))$ to it. Final estimators will be formed as

$$\hat{\beta}_{(x,0),GIBBS} = \frac{1}{K+1} \sum_{i=t}^{t+K} \beta_{x,0}^{(i)} ,$$

$$\hat{\beta}_{(x,1),GIBBS} = \frac{1}{K+1} \sum_{i=t}^{t+K} \beta_{x,1}^{(i)} ,$$

where t indicates the number after which the chain converges and K is the number of samples which are taken from the output of the Gibbs Sampler.

C. Performance Bounds

The next step would be to simulate some realistic scenarios and observe the performances of the two estimation methods suggested in the previous sections, the JML-

estimator and the Gibbs Sampler. However, it will be useful to have some benchmarks with whom to compare their performances. For this reason, this section is dedicated to derivation of such bounds. Some theorems from Statistics are used in this section and their explanations can be found in Appendix B.

1. Lower Bounds

The first step is to look for lower bounds for skew and offset estimation. Since the likelihood function does not satisfy the regularity conditions required by the CRLB (Cramer-Rao Lower Bound), calculating CRLBs is dropped out of the list. One possible lower bound can be found by assuming that all the parameters are known but one. Then we can find the UMVU estimators both for the phase offset and skew in closed forms. Since in the original problem of this chapter, both parameters (offset and skew) are unknown at the same time, we cannot perform better than UMVU estimators which will be derived in this section.

- For $\beta_{x,0}$, it is known that

$$\hat{\beta}_{(x,0),UMVU} = \min(T_X[i] - (i-1)\beta_{x,1}) - \frac{1}{N\lambda_x}, \quad (4.3)$$

and the MSE (Mean-Square Error) of the estimator equals $1/(N\lambda_x)^2$ [43].

- For $\beta_{x,1}$, with known $\beta_{x,0}$, the likelihood function can be written as

$$L(\beta_{x,1}) = C e^{\lambda_x \frac{N(N-1)}{2} \beta_{x,1}} \prod_{i=2}^N I\left(\beta_{x,1} \leq \frac{T_X[i] - \beta_{x,0}}{i-1}\right), \quad (4.4)$$

where C is a normalizing constant. By the Factorization Theorem [43], $Z = \min_{i=2,3,\dots,N}((X[i] - \beta_{x,0})/(i-1))$ is a sufficient statistics. The next question is whether it is complete or not. To check the completeness of the sufficient statistics Z for estimating the parameter $\beta_{x,1}$, we first need to find the distri-

bution function F_Z . From the theory of order statistics [24], the distribution of the minimum of a sample set is given by

$$F_Z(z) = 1 - (1 - F_2(z))(1 - F_3(z))\dots(1 - F_N(z)), \quad (4.5)$$

where $F_i(z) = F((T_X[i] - \beta_{x,0})/(i - 1) \leq z) = F(v_{x,\lambda_x}[i] \leq (i - 1)(z - \beta_{x,1})) = (1 - e^{-\lambda_x(i-1)(z-\beta_{x,1})})I(z \geq \beta_{x,1})$. Then, plugging these back into (4.5), the distribution function of the sufficient statistics becomes

$$F_Z(z) = 1 - e^{-\lambda_x(z-\beta_{x,1})(1+2+\dots+N-1)} = 1 - e^{-\lambda_x(z-\beta_{x,1})\frac{N(N-1)}{2}}, \quad (4.6)$$

which is an exponential distribution with the scale parameter $\lambda_x N(N - 1)/2$ and the location parameter $\beta_{x,1}$.

Completeness of sufficient statistics (Z). Assume that $h(\beta_{x,1}) = E[g(Z)] = 0$ for all $\beta_{x,1}$. Since $h(\beta_{x,1}) = \int_{\beta_{x,1}}^{\infty} g(Z)f_Z(z)dz = c \int_{\beta_{x,1}}^{\infty} g(Z)e^{-c(z-\beta_{x,1})}dz$, using Leibniz Rule $h'(\beta_{x,1}) = c(h(\beta_{x,1}) - g(\beta_{x,1})) = -cg(\beta_{x,1})$. Since $h'(\beta_{x,1}) = 0$ then $g(\beta_{x,1}) = 0$. Hence, the sufficient statistics (Z) is complete.

Then, by the Lehmann-Scheffe Theorem [43], the UMVU estimator for clock skew when the clock phase offset and λ_x are known takes the following form:

$$\hat{\beta}_{(x,1),UMVU} = \min_i \left(\frac{T_X[i] - \beta_{x,0}}{i - 1} \right) - \frac{2}{\lambda_x N(N - 1)}. \quad (4.7)$$

The MSE of $\hat{\beta}_{(x,1),UMVU}$ equals the variance of Z which is $4/(\lambda_x N(N - 1))^2$.

2. Upper Bounds

We will also consider the BLUE (Best Linear Unbiased Estimator). There are two reasons why we want to use the BLUE in our simulations. Since we have the JML-estimator, the BLUE will definitely represent an upper bound. Additionally, the

BLUE is suggested as an estimator in the RBS in [18], for joint estimation of clock skew and offset; however, the BLUE is optimal when the noise is normally distributed. It is not optimal with exponentially distributed noise. The simulations will enable us to see this distinction clearly.

Since there is abundant amount of information on the BLUE (mostly due to the Gauss-Markov Theorem [42]), we discuss it briefly. We will use the same notation as [41] (except \mathbf{X} is replaced with \mathbf{A} , to prevent possible confusion). However, since the noise is not zero-mean, after treating the data as if it is embedded in zero-mean noise, we need to subtract $1/\lambda_x$ from the resulting linear estimate of $\beta_{x,0}$ so as to end up with the BLUE, i.e., $[\hat{\beta}_{(x,0),BLUE}, \hat{\beta}_{(x,1),BLUE}]^\top = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top T_{\mathbf{X}} - [1/\lambda_x, 0]^\top$. It is well known that $var([\hat{\beta}_{(x,0),BLUE}, \hat{\beta}_{(x,1),BLUE}]^\top) = diag\{1/\lambda_x^2 (\mathbf{A}^\top \mathbf{A})^{-1}\} \propto [1/N, 1/N^3]^\top$. For a detailed discussion on this estimator, the interested readers are referred to [41] and [44].

D. Simulations

In this section, we present the simulation results regarding the performance comparison of the estimators considered in this chapter, the convergence properties of the Gibbs Sampler and the sample distributions of the estimators.

By setting $\beta_{x,0} = 1.0$, $\beta_{x,1} = 0.001$, $\lambda_x = 1$ and $N = 20$, and using a single realization of the noise process $(\mathbf{v}_{x,\lambda_x})$, the estimates of the estimators (JML, GIBBS and BLUE), are depicted in Figure 9. From the figure, it can be seen that the BLUE can produce an estimate outside the domain of the likelihood function. The same figure also shows the samples of the Gibbs Sampler. Theoretically, the sampling procedure can visit any point inside the domain of the likelihood function. However, the samples of the Gibbs Sampler are concentrated on the part of the domain where

the likelihood function has most of its volume.

The MSE of the Gibbs Sampler and the JML-estimator for phase offset and skew estimations, with $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$ and $\lambda_x = 10^3$ ($\text{var}(v_{x,\lambda_x}) = 10^{-6}$) are presented in Figures 10 and 11, respectively. The figures also include the lower and upper bounds derived in the previous section. The MSE are plotted against the number of synchronization signals from 4 to 36. It is interesting to note that the MSE of the Gibbs Sampler and the JML-estimator behave like the lower bounds; i.e., the decay rates scale with $1/N^2$ and $1/N^4$ for the phase offset and skew estimates, respectively. Additionally, in terms of MSE performances, compared to the JML-estimator, the Gibbs Sampler performs 40% and 25% better for the phase offset and skew estimates, respectively. Compared to the JML-estimator, the Gibbs Sampler reduces the bias in phase offset estimation remarkably, which is illustrated in Figure 12.

The convergence behavior of the Gibbs Sampler for $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$, $N = 20$ and $\lambda_x = 1$, is illustrated in Figure 13. In this figure, the first 100 samples drawn by the Gibbs Sampler of the phase offset and skew parameters are plotted. The true values of the clock parameters are also depicted in the same figure as the straight lines. It is remarkable that the Gibbs Sampler converges within 20 iterations, even though $\lambda_x = 1$ (note that the smaller the value of λ_x gets, the slower the Gibbs Sampler converges).

The Figures 14 and 15, illustrate the probability density functions of the estimators for offset and skew estimates, respectively, using the corresponding histograms. The histograms are based on 1000 Monte-Carlo samples for $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$, $N = 20$ and $\lambda_x = 1$. Since the regularity conditions are violated, the density of the Gibbs Sampler does not resemble a normal distribution. On the contrary, it is more like a Laplace distribution.

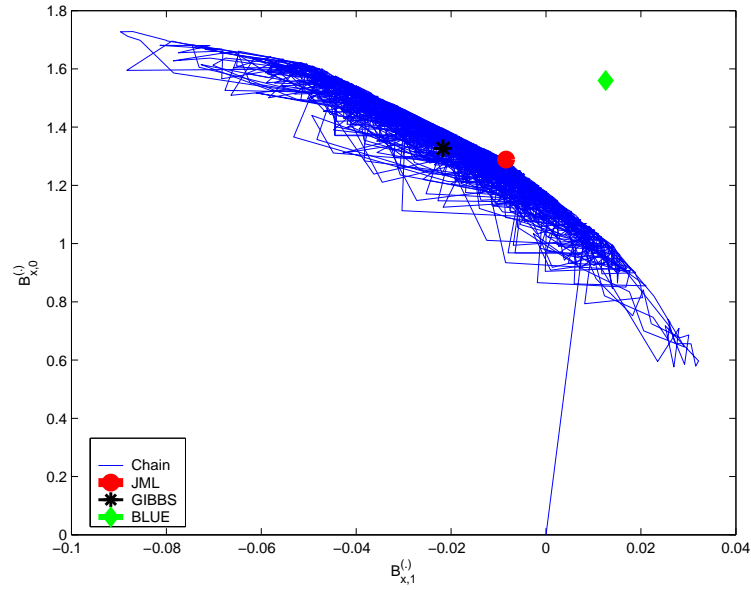


Fig. 9. Gibbs sampling and estimates by BLUE, JML and Gibbs Sampler. The parameters are set as $\beta_{x,0} = 1.0$, $\beta_{x,1} = 0.001$, $\lambda_x = 1$ and $N = 20$.

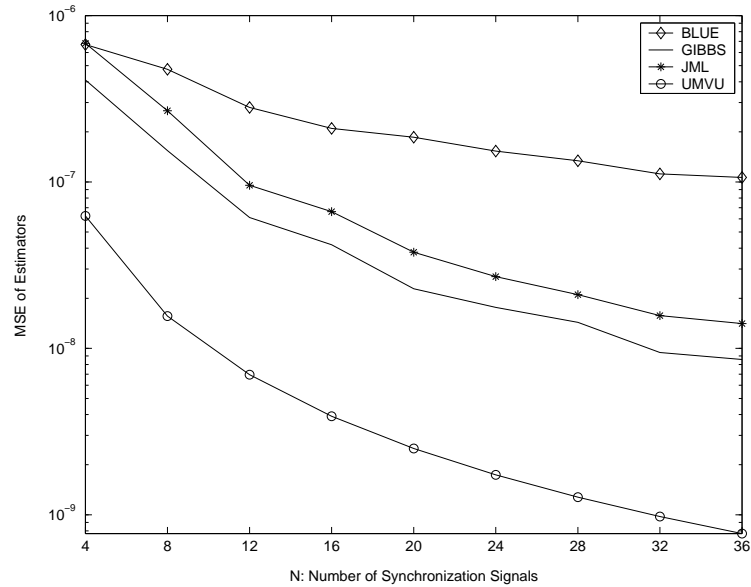


Fig. 10. MSE performances of the offset estimators. The estimators are $\hat{\beta}_{(x,0),BLUE}$, $\hat{\beta}_{(x,0),JML}$, $\hat{\beta}_{(x,0),GIBBS}$, and $\hat{\beta}_{(x,0),UMVU}$. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$ and $\lambda_x = 10^3$.

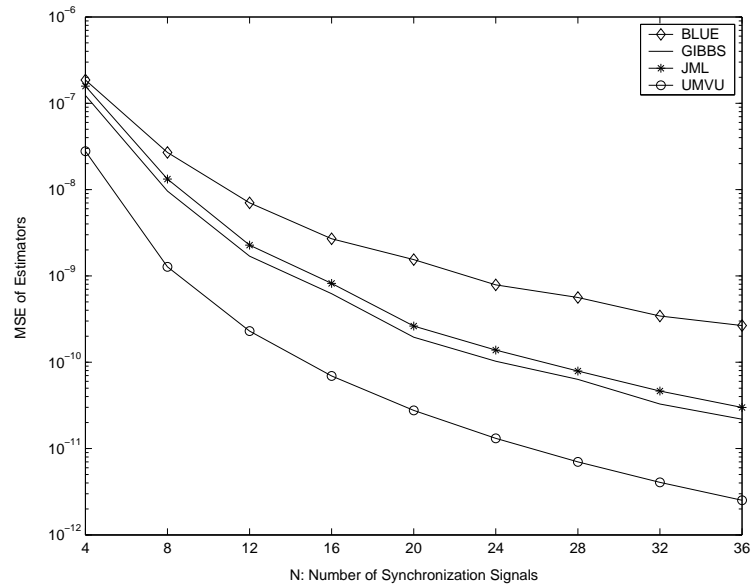


Fig. 11. MSE performances of the skew estimators. The estimators are $\hat{\beta}_{(x,1),BLUE}$, $\hat{\beta}_{(x,1),JML}$, $\hat{\beta}_{(x,1),GIBBS}$ and $\hat{\beta}_{(x,1),UMVU}$. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$ and $\lambda_x = 10^3$.

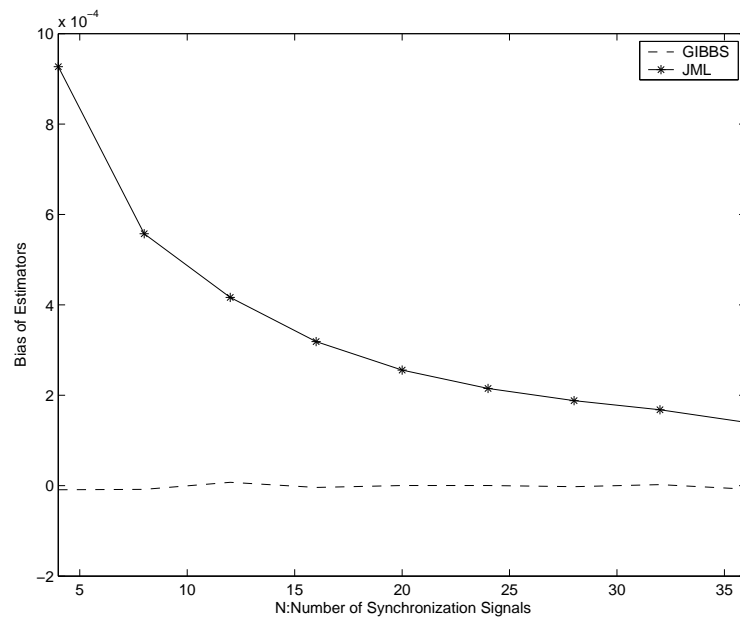


Fig. 12. Bias on offset estimators. The estimators are $\hat{\beta}_{(x,0),JML}$ and $\hat{\beta}_{(x,0),GIBBS}$. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$ and $\lambda_x = 10^3$.

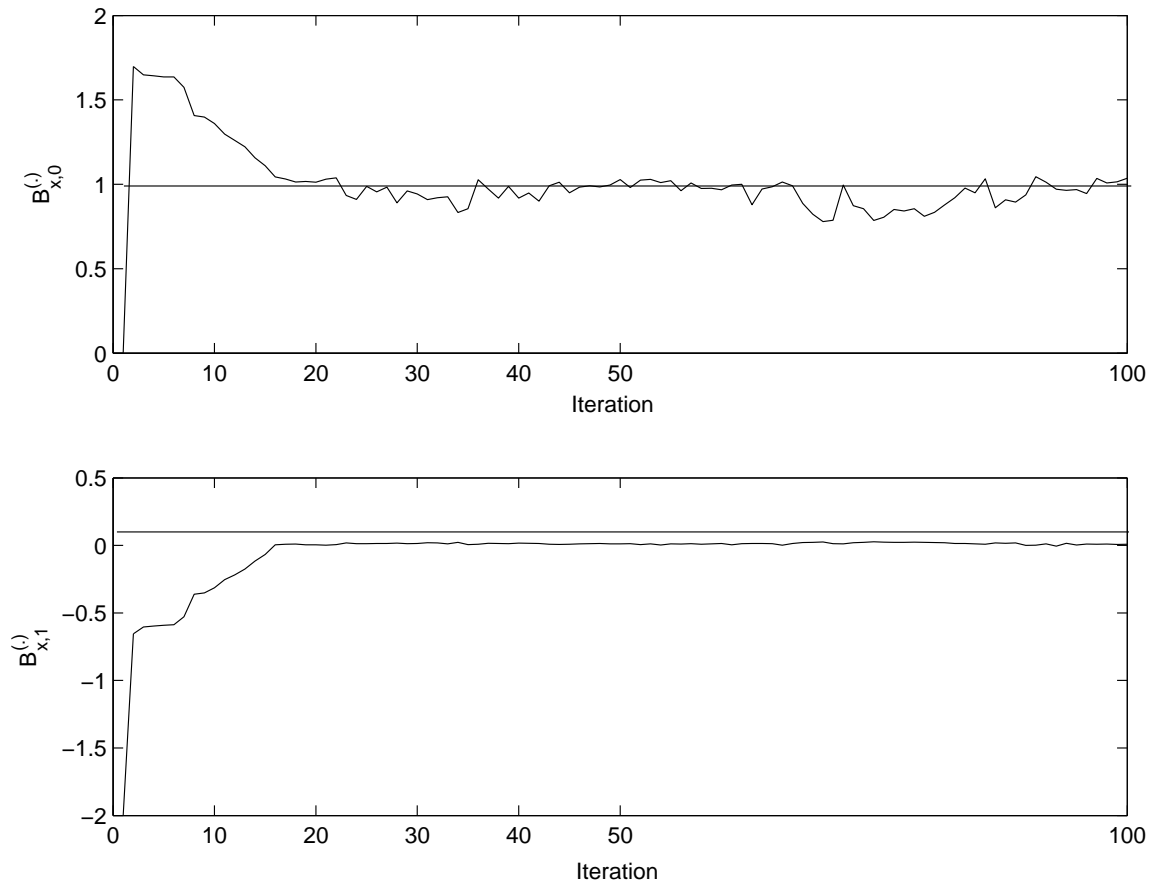


Fig. 13. Convergence of the Gibbs Sampler for offset and skew estimates. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$, $N = 20$ and $\lambda_x = 1$.

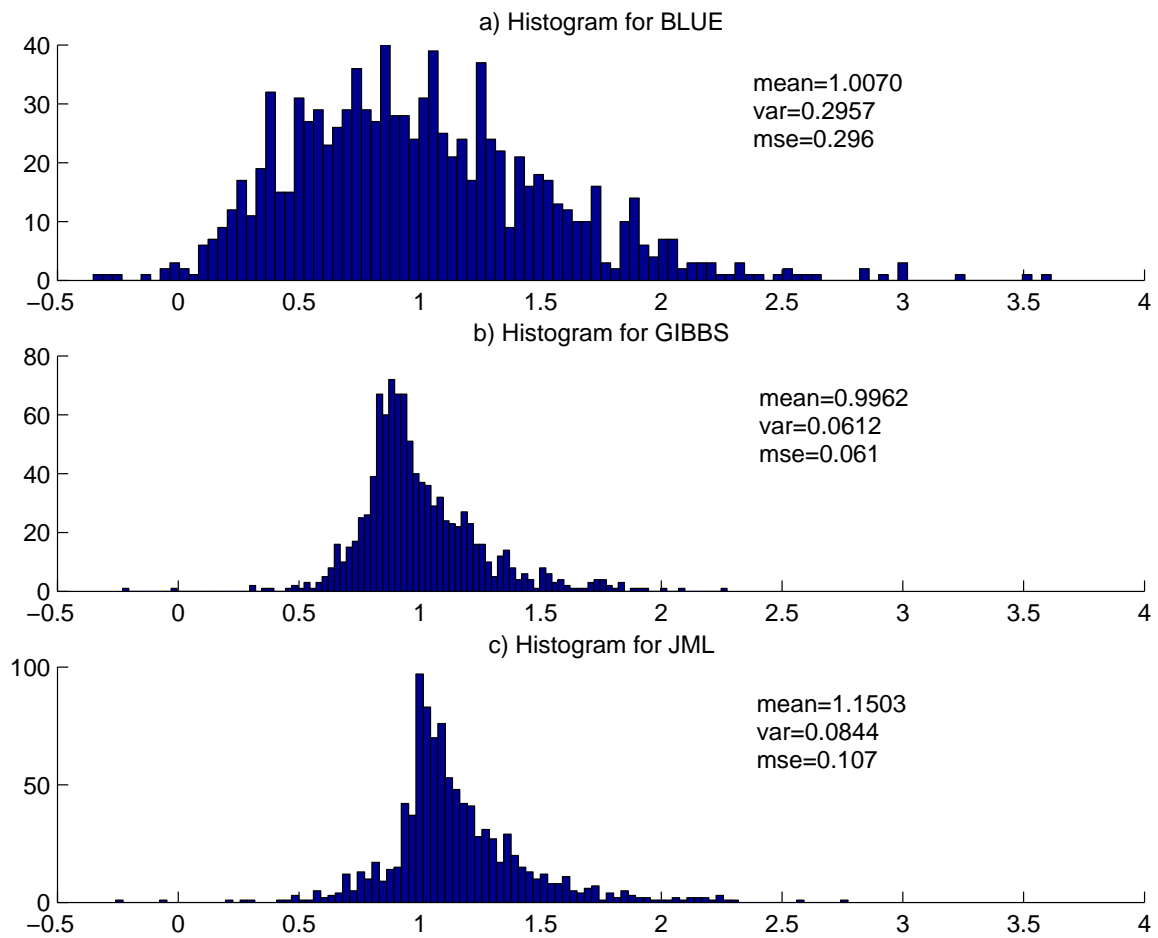


Fig. 14. Histograms for the offset estimators. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$, $N = 20$ and $\lambda_x = 1$. The estimators are **a)** BLUE, **b)** GIBBS and **c)** JML.

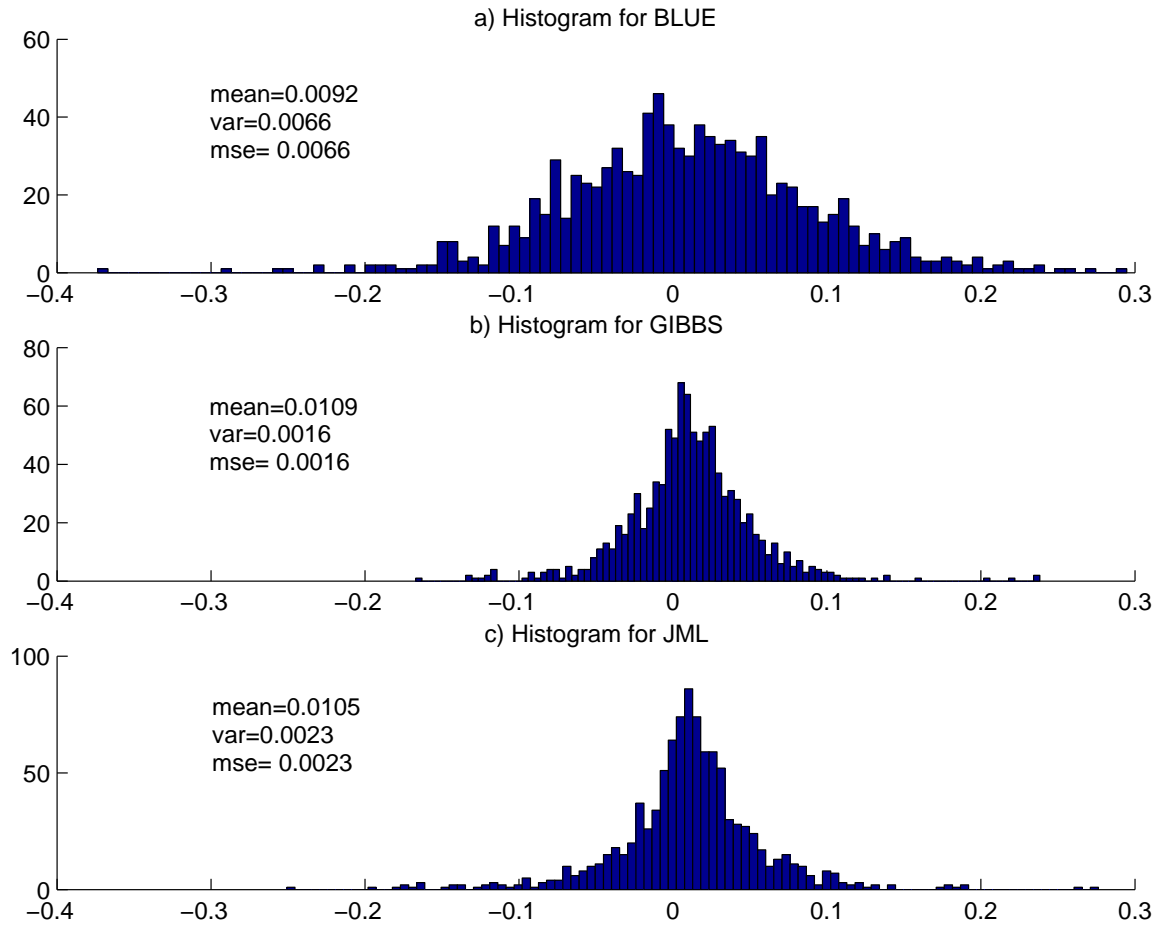


Fig. 15. Histograms for the skew estimators. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$, $N = 20$ and $\lambda_x = 1$. The estimators are a) BLUE, b) GIBBS and c) JML.

CHAPTER V

EFFECT OF UNKNOWN NOISE PARAMETERS

In the previous chapter, we assumed that the parameters of the exponential noise terms λ_x and λ_y are known to the transmitter. As a matter of fact, it is more likely that the transmitter will not know the values of λ_x and λ_y . However, we made this assumption, not because it is more reasonable, but because it is more convenient. The assumption of unknown noise parameters would introduce a third dimension to the parameter space. This would reduce the readability of most of the derivations and figures, at no return. As it will be clear, in this section, unknown noise parameters has no effect on the maximum likelihood estimation and negligible effect on the Gibbs Sampler.

In short, this chapter is dedicated to analyzing the effect of unknown noise parameters on the performance of the estimators. In the first section of this chapter, the JML-estimator is derived. In the second section, the Gibbs Sampler is applied, and the last section presents the simulation results.

A. Joint Maximum Likelihood Estimator

First of all, the likelihood function, which is given in equation (4.1), is maximized over λ_x when $\tilde{\lambda}_x(\beta_{x,0}, \beta_{x,1}) = (\bar{T}_{\mathbf{X}} - (\beta_{x,0} + (N-1)/2\beta_{x,1}))^{-1}$ for all fixed values of $(\beta_{x,0}, \beta_{x,1})$. Therefore, by plugging $\tilde{\lambda}_x$ back into (4.1), the reduced likelihood function for $(\beta_{x,0}, \beta_{x,1})$ can be found as

$$L_R(\beta_{x,0}, \beta_{x,1}) = C(\bar{T}_{\mathbf{X}} - (\beta_{x,0} + \frac{N-1}{2}\beta_{x,1}))^{-N} \prod_{i=1}^N I(T_X[i] \geq \beta_{x,0} + (i-1)\beta_{x,1}) \quad (5.1)$$

Since the inequality $\bar{T}_{\mathbf{X}} > \beta_{x,0} + (N - 1)\beta_{x,1}/2$ always holds, finding the maximum of (5.1) is equivalent to finding the maximum of (4.1). In other words, to find the JML-estimator when λ_x is unknown, the same objective function $f = \beta_{x,0} + (N - 1)\beta_{x,1}/2$, should be maximized over the same region \mathbf{S} .

In summary whether λ_x is known or not, the JML-estimator is the same for both situations.

B. Application of Gibbs Sampling

To obtain the Gibbs Sampler for the case with unknown noise parameter, we will again use the likelihood function given in (4.1). However, this time, we will treat λ_x as a third parameter to be estimated. Then the procedure for applying the Gibbs Sampler becomes

- Draw $\lambda_x^{(k+1)}$ from $\propto \lambda_x^N e^{-\lambda_x N(\bar{T}_{\mathbf{X}} - (\beta_{x,0}^{(k)} + \frac{N-1}{2}\beta_{x,1}^{(k)})}$
- Draw $\beta_{x,0}^{(k+1)}$ from $\propto e^{\lambda_x^{(k+1)} N \beta_{x,0}} I(\beta_{x,0} \leq \min_i (T_X[i] - (i - 1)\beta_{x,1}^{(k)}))$
- Draw $\beta_{x,1}^{(k+1)}$ from $\propto e^{\lambda_x^{(k+1)} \frac{N(N-1)}{2} \beta_{x,1}} I(\beta_{x,1} \leq \min_i (\frac{T_X[i] - \beta_{x,0}^{(k+1)}}{i-1}))$.

To implement these iterative steps, for $\lambda_x^{(k+1)}$, we will draw a sample from a Gamma distribution with parameters $(N + 1)$ and $(\bar{T}_{\mathbf{X}} - (\beta_{x,0}^{(k)} + \beta_{x,1}^{(k)}(N - 1)/2))^{-1}$. (Note that both of these parameters are guaranteed to be positive-valued, so they do not violate the requirements of the Gamma distribution.) For $\beta_{x,0}^{(k+1)}$, we will draw a sample from an exponential distribution with parameter $\lambda_x^{(k+1)} N$, multiply it with -1 and add $\min_i (T_X[i] - (i - 1)\beta_{x,1}^{(k)})$ to it. The procedure for $\beta_{x,1}^{(k+1)}$ will be similar. We will draw a sample from an exponential distribution with parameter $\lambda_x^{(k+1)} N(N - 1)/2$, multiply it with -1 and add $\min_i ((T_X[i] - \beta_{x,0}^{(k+1)})/(i - 1))$ to it.

C. Discussions and Simulations

In the previous chapter, the value of λ_x was assumed known to the transmitter and it was used in the application of Gibbs sampling. When this parameter is unknown, the transmitter should also estimate it in the Gibbs Sampler, while the JML-estimator does not need to know the value of λ_x . Estimating an additional parameter will cause performance degradation in the Gibbs Sampler. The question is whether the performance loss will be significant or negligible.

The simulation result is shown in Figure 16, where the MSE of the Gibbs Samplers and the JML-estimator for offset estimation are depicted, with $\beta_{x,0} = 1$, $\beta_{x,1} = 0.01$ and $\lambda_x = 10^3$. The figure shows the MSE result of the Gibbs Sampler from the previous section (Figure 10), which is labeled as GIBBS-old, and the Gibbs Sampler when λ_x is unknown, which is labeled as GIBBS-new.

Figure 16 shows that when the total number of synchronization signals broadcasted by the transmitter is smaller than 12, the performance loss in the Gibbs Sampler is noticeable. However, when the total number of synchronization signals is 12 or more, the performance loss is vanishingly small. It is also remarkable that the Gibbs Sampler is superior compared to the JML-estimator even if λ_x is unknown.

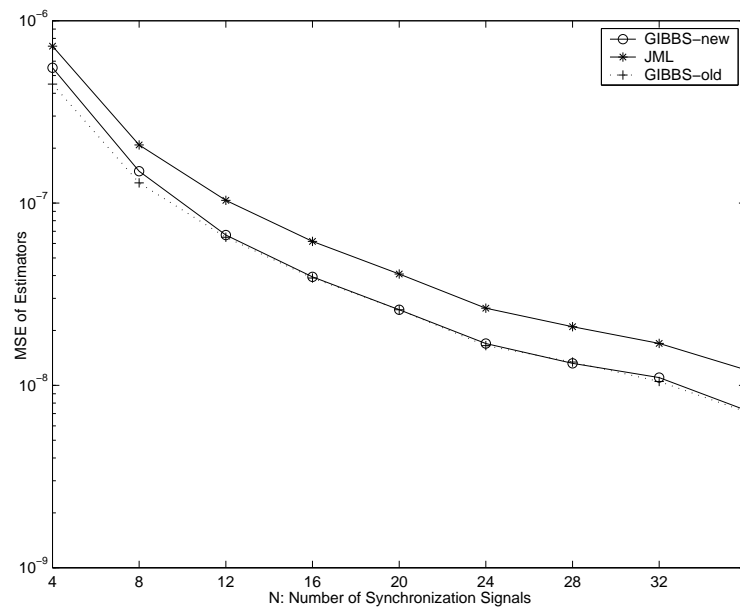


Fig. 16. MSE performances of the offset estimators (with unknown λ_x). The estimators are $\hat{\beta}_{(x,0),JML}$ and $\hat{\beta}_{(x,0),GIBBS}$. The parameters are set as $\lambda_x = 10^3$, $\beta_{x,0} = 1$ and $\beta_{x,1} = 0.01$.

CHAPTER VI

JOINT SYNCHRONIZATION OF CLOCK PHASE OFFSET, SKEW AND DRIFT

In this section, we finally consider the complete problem of the thesis that is the joint synchronization of clock phase offset, skew and drift. In Chapter IV, we considered the joint maximum likelihood approach and Gibbs sampling for estimation of clock skew and offset assuming that the relative drift is zero. We saw that finding the JML-estimator is not easy at all. It will probably be even more difficult when we introduce one more parameter (drift) to the problem. For this reason, in this chapter we want to apply only the Gibbs sampling procedure to the problem. In the first section of this chapter, the Gibbs Sampler will be adopted for the problem. The second section presents the related simulation results.

A. Application of Gibbs Sampling

To obtain the Gibbs Sampler, we first need a new likelihood function, since (4.1) does not include $\beta_{x,2}$. By using the model described by (2.12), we can express the likelihood function for $(\beta_{x,0}, \beta_{x,1}, \beta_{x,2})$ as

$$L = C e^{\lambda_x N ((\beta_{x,0} + \frac{N-1}{2} \beta_{x,1} + \frac{(N-1)(2N-1)}{6} \beta_{x,2}))} \prod_{i=1}^N I(T_X[i] \geq \beta_{x,0} + (i-1)\beta_{x,1} + (i-1)^2 \beta_{x,2}). \quad (6.1)$$

Using the likelihood function from (6.1), the procedure for Gibbs sampling for estimating $(\beta_{x,0}, \beta_{x,1}, \beta_{x,2})$ becomes

- Draw $\beta_{x,0}^{(k+1)}$ from $\propto e^{\lambda_x N \beta_{x,0}} I(\beta_{x,0} \leq \min_i (T_X[i] - (i-1)\beta_{x,1}^{(k)} - (i-1)^2 \beta_{x,2}^{(k)}))$
- Draw $\beta_{x,1}^{(k+1)}$ from $\propto e^{\lambda_x \frac{N(N-1)}{2} \beta_{x,1}} I(\beta_{x,1} \leq \min_i (\frac{T_X[i] - \beta_{x,0}^{(k+1)} - (i-1)^2 \beta_{x,2}^{(k)}}{i-1}))$
- Draw $\beta_{x,2}^{(k+1)}$ from $\propto e^{\lambda_x \frac{N(N-1)(2N-1)}{6} \beta_{x,2}} I(\beta_{x,2} \leq \min_i (\frac{T_X[i] - \beta_{x,0}^{(k+1)} - (i-1)\beta_{x,1}^{(k+1)}}{(i-1)^2}))$.

For all three steps, we can use simple exponential distributions as it was done in the previous chapters. At the end, we can use the mean of samples from parameter space as the estimator for a specific parameter.

B. Simulations

In this section, we will check whether the Gibbs Sampler still preserves its superiority over the conventional BLUE or not. The BLUE will still have the same form, which is $[\hat{\beta}_{(x,0),BLUE}, \hat{\beta}_{(x,2),BLUE}, \hat{\beta}_{(x,2),BLUE}]^\top = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top T_{\mathbf{X}} - [1/\lambda_x, 0, 0]^\top$.

Simulation results are shown in Figures 17, 18 and 19, where the parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.1$, $\beta_{x,2} = 0.01$ and $\lambda_x = 10$. These figures illustrate the MSE of drift, skew and offset estimators, respectively. For all three parameters, the Gibbs Sampler has still much better performance compared to the BLUE.

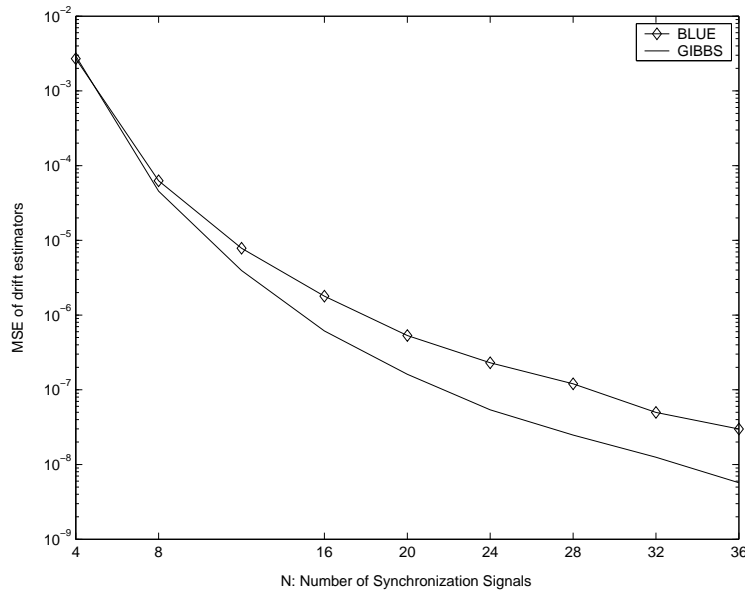


Fig. 17. MSE performances of the drift estimators (with unknown drift). The estimators are $\hat{\beta}_{(x,2),BLUE}$, and $\hat{\beta}_{(x,2),GIBBS}$. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.1$ and $\beta_{x,2} = 0.01$.

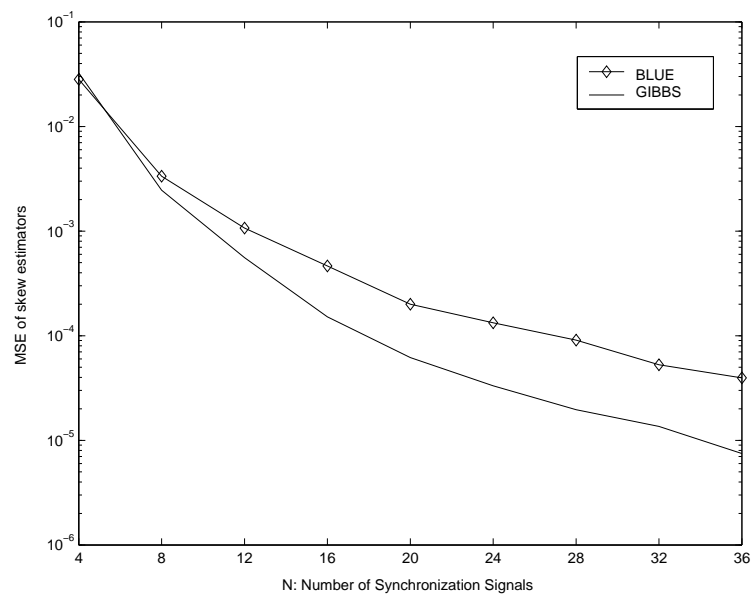


Fig. 18. MSE performances of the skew estimators (with unknown drift). The estimators are $\hat{\beta}_{(x,1),BLUE}$, and $\hat{\beta}_{(x,1),GIBBS}$. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.1$ and $\beta_{x,2} = 0.01$.

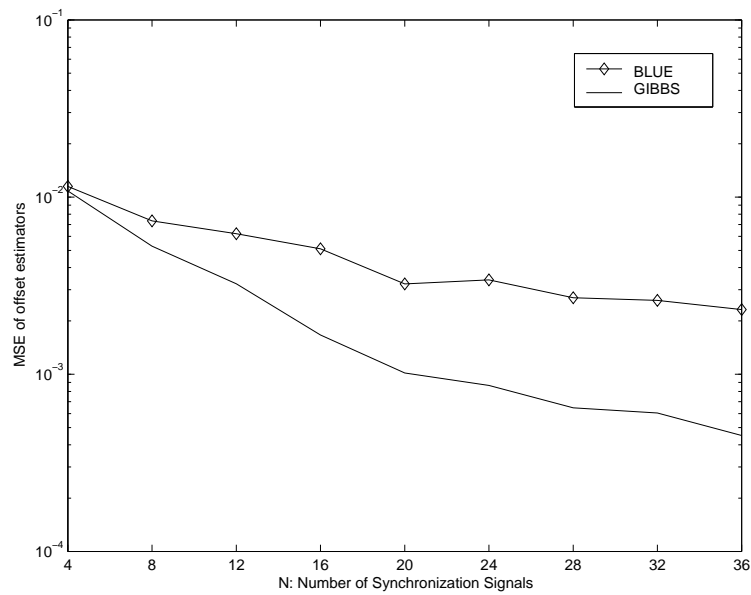


Fig. 19. MSE performances of the offset estimators (with unknown drift). The estimators are $\hat{\beta}_{(x,0),BLUE}$, and $\hat{\beta}_{(x,0),GIBBS}$. The parameters are set as $\beta_{x,0} = 1$, $\beta_{x,1} = 0.1$ and $\beta_{x,2} = 0.01$.

CHAPTER VII

SUMMARY

A. Conclusions

The contributions of this work can be summed up as follows:

- Under the exponential noise model, we have shown that the maximum of the likelihood function in (4.1) is achieved on the boundary of the domain \mathbf{S} . We have also discussed two different methods to find this point, i.e., the JML-estimator.
- The setting was convenient to apply the Gibbs sampling procedure which showed better performance compared to the JML-estimator. Lower and upper-bounds for the performance of the JML-estimator and the Gibbs Sampler were also presented in terms of the MSE-performances of the UMVU estimator and the BLUE, respectively.
- We have also shown by simulations that the MSE performances of the JML-estimator and the Gibbs Sampler are proportional to $1/N^2$ and $1/N^4$ for the phase offset and skew estimations, respectively. Since the MSE performances of the BLUE for the phase offset and skew are proportional to $1/N$ and $1/N^3$, the Gibbs Sampler provides significant improvement over the BLUE.
- We also showed that unknown noise parameter does not change the performance of the JML-estimator, while it degrades the performance of the Gibbs Sampler slightly, only when the number of synchronization signals is smaller than 12.
- Finally, we extended the Gibbs Sampler for estimating the relative drift and showed that the Gibbs Sampler is adequate for high dimensional parameter

estimations, and is still superior compared to the conventional BLUE.

In summary, we tried to show that by applying powerful statistical signal processing tools, we can obtain significant performance gains in RBS protocol.

B. Future Research Work

We believe that the following problems might be worth to investigate.

- Exponential distribution has only one free parameter. In some situations, we might need more general distributions to capture the network delays. Gamma distribution might be a better candidate since it has two free parameters. We believe that the form of the likelihood function in that case will be much more complex for mathematical analysis. The Gibbs Sampler will most probably help a lot towards the solution of this problem as well.
- Although we have assumed that we do not have any prior information regarding the clock parameters, if we can find a good probabilistic model for the long-term stability properties of the clock parameters, this model can be used to establish a prior distribution for the clock parameters. Then the Gibbs Sampler can be adopted for Bayesian estimation.
- We have also assumed that the transmission of synchronization signals is periodic. However, by relaxing this assumption, the optimal scheme for transmission times of synchronization signals can be obtained.
- This work considered only local time-synchronization. Extension to global synchronization, using the distributive particle-filtering technique, which is a dynamic MCMC technique, might be possible.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, Vol. 38, No. 4, pp. 393-422, March 2002.
- [2] A. Ephremides, "Energy concerns in wireless networks," *IEEE Wireless Communications*, Vol. 9, No. 4, pp. 48-59, August 2002.
- [3] G. J. Pottie, W. J. Kaiser, "Wireless integrated network sensors," *Communications of the ACM*, Vol. 43, No. 5, pp. 51-58, May 2000.
- [4] J. E. Elson, "Time synchronization in wireless sensor networks," Ph.D. dissertation, University of California Los Angeles, Los Angeles, California, April 2003.
- [5] S. Ganeriwal, D. Ganesan, H. Sim, V. Tsiatsis, M. B. Srivastava, "Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks," presented at the 3rd ACM Conference on Embedded Networked Sensor Systems, San Diego, CA, November 2005.
- [6] A. Syed, J. Heidemann, "Time synchronization for high latency acoustic networks," USC/Information Sciences Institute Technical Report ISI-TR-2005-602, April 2005. Available online at <http://www.isi.edu/johnh/PAPERS/Heidemann06a.html>.
- [7] F. Sivrikaya, B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, Vol. 18, No. 4, pp. 45-50, July-August 2004.
- [8] B. Sundararaman, U. Buy, A. D. Kshemkalyani, "Clock synchronization in wireless sensor networks: a survey," *Ad-Hoc Networks*, Vol. 3, No. 3, pp. 281-323, May 2005.

- [9] K. Romer, P. Blum, L. Meier, “Time synchronization and calibration in wireless sensor networks,” in *Handbook of Sensor Networks: Algorithms and Architectures*, I. Stojmenovic, Ed. New York: John Wiley & Sons, 2005.
- [10] S. Bregni, *Synchronization of Digital Communications Networks*, New York: John Wiley & Sons, 1998.
- [11] J. R. Vig, “Introduction to quartz frequency standards,” Army Research Laboratory Electronics and Power Sources Directorate Technical Report SLCET-TR-92-1, October 1992. Available online at <http://www.ieee-uffc.org/freqcontrol/quartz/vig/vigtoc.htm>.
- [12] D. A. Howe, D. W. Alan and J. A. Barnes, “Properties of signal sources and measurement methods,” presented at the 35th Annual Symposium on Frequency Control, Philadelphia, PA, May 1981.
- [13] D. L. Mills, “Internet Time Synchronization: the Network Time Protocol,” *IEEE Transactions on Communications*, Vol. 39, No. 10, pp. 1482-1493, October 1991.
- [14] D. L. Mills, “A brief history of NTP time: confessions of an Internet timekeeper,” *ACM Computer Communications Review*, Vol. 33, No. 2, pp. 9-22, April 2003
- [15] S. Ganeriwal, R. Kumar, M. B. Srivastava, “Timing sync protocol for sensor networks,” presented at the 1st ACM Conference on Embedded Networked Sensor Systems, Los Angeles, CA, November 2003.
- [16] M. Maroti, B. Kusy, G. Simon, A. Ledeczi, “The flooding time synchronization protocol,” presented at the 2nd ACM Conference on Embedded Networked Sensor Systems, Baltimore, MD, November 2004.

- [17] W. Su, I. F. Akyildiz, “Time-diffusion synchronization protocol for wireless sensor networks,” *IEEE/ACM Transactions on Networking*, Vol. 13, No. 2, pp. 384-397, April 2005.
- [18] J. Elson, L. Girod, D. Estrin, “Fine-grained network time synchronization using reference broadcasts,” presented at the 5th Symposium on Operating Systems Design and Implementation, Boston, MA, December 2002.
- [19] H. S. Abdel-Ghaffar, “Analysis of synchronization algorithms with time-out control over networks with exponentially symmetric delays,” *IEEE Transactions on Communications*, Vol. 50, No. 10, pp. 1652-1661, October 2002.
- [20] D. R. Jeske, “On the maximum likelihood estimation of clock offset,” *IEEE Transactions on Communications*, Vol. 53, No. 1, pp. 53-54, January 2005.
- [21] D. R. Jeske, A. Sampath, “Estimation of clock offset using bootstrap bias-correction techniques,” *Technometrics*, Vol. 45, No. 3, pp. 256-261, August 2003.
- [22] V. Paxson, “On calibrating measurements of packet transit times,” Lawrence Berkeley National Laboratory Technical Report LBNL-41535, March 1998. Available online at <ftp://ftp.ee.lbl.gov/papers/vp-clocks-sigmetrics98.ps.gz>
- [23] V. Paxson, “On calibrating measurements of packet transit times,” presented at the 7th ACM Sigmetrics Conference, Madison, WI, June 1998.
- [24] H. A. David, H. N. Nagaraja, *Order Statistics*, 3rd Edition, New York: John Wiley & Sons, 2003.
- [25] S. Narasimhan, S. S. Kunniyur, “Effect of network parameters on delay in wireless ad-hoc networks,” University of Pennsylvania Technical Report, June 2004.

- [26] C. Bovy, H. Mertodimedjo, G. Hooghiemstra, H. Uijterwaal, P. Mieghem, “Analysis of end-to-end delay measurements in Internet,” presented at the Passive and Active Measurements Workshop, Fort Collins, CO, March 2002.
- [27] S. Moon, P. Skelley, D. Towsley, “Estimation and removal of clock skew from network delay measurements,” presented at the IEEE INFOCOM Conference on Computer Communications, New York, NY, March 1999.
- [28] A. Leon-Garcia, *Probability and Random Processes for Electrical Engineering*, 2nd Edition, Massachusetts: Addison-Wesley, 1994
- [29] Y. Zhang, N. Duffield, V. Paxson, S. Shenker, “On the constancy of Internet path properties,” presented at the ACM SIGCOMM Internet Measurement Workshop, San Francisco, November 2001.
- [30] X. Wang, R. Chen, “Adaptive Bayesian multiuser detection for synchronous CDMA with Gaussian and impulsive noise,” *IEEE Transactions on Signal Processing*, Vol. 48, No. 7, pp. 2013-2028, July 2000.
- [31] A. Doucet, X. Wang, “Monte Carlo methods for signal processing,” *IEEE Signal Processing Magazine*, Vol. 22, No. 6, pp. 152-170, November 2005.
- [32] D. J. C. MacKay, *Information Theory, Inference & Learning Algorithms*, Cambridge: Cambridge University Press, 2002.
- [33] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, 2nd Edition, New York: Springer, 2004.
- [34] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, “Equations of state calculations by fast computing machines,” *Journal of Chemical Physics*, Vol. 21, No. 6, pp. 1087-1092, June 1953.

- [35] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, Vol. 57, No. 1, pp. 97-109, January 1970.
- [36] S. Geman, D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 6, No. 6, pp. 721-741, November 1984.
- [37] A. E. Gelfand, A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *Journal of the American Statistical Association*, Vol. 85, No. 410, pp. 398-409, June 1990.
- [38] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 36, No. 2, pp. 192-236, May 1974.
- [39] J. P. Hobert, G. Casella, "Functional compatibility, Markov chains, and Gibbs sampling with improper posteriors," *Journal of Computational and Graphical Statistics*, Vol. 7, No. 4, pp. 42-60, December 1998.
- [40] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Rubin, *Bayesian Data Analysis*, 2nd Edition, London: Chapman & Hall, 2004.
- [41] B. M. Sadler, "Local and broadcast clock synchronization in a sensor network," *IEEE Signal Processing Letters*, Vol. 13, No. 1, pp. 9-12, January 2006.
- [42] D. G. Luenberger, *Optimization by Vector Space Methods*, New York: John Wiley & Sons, 1969.
- [43] E. L. Lehmann, G. Casella, *Theory of Point Estimation*, 2nd Edition, New York: Springer, 1998.

- [44] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 1: Estimation Theory*, New Jersey: Prentice-Hall, 1998.
- [45] N. L. Johnson, S. Kotz, N. Balakrishnan, *Continuous Univariate Distributions, Volume 2*, 2nd Edition, New York: John Wiley & Sons, 1995.

APPENDIX A

PERFORMANCE LOSS DUE TO PROCESSING OF DATA BEFORE
ESTIMATION

In this appendix, we will discuss what we might lose if we apply the transformation $T_Y - T_X$ before estimating the relative clock parameters, under exponential noise model of (2.12) and (2.13). We will look at the case where only the relative offset (β_0) is unknown. When the noise is normal, the transformation does not effect the optimality of the estimator. On the other hand, it degrades the performance severely in exponential noise. From [20] (although it was shown for sender-to-sender systems, the same basic idea applies here as well), the MLE is given by

$$\hat{\beta}_0 = T_Y(1) - T_X(1), \quad (\text{A.1})$$

where the brackets around the index (\cdot) indicate the m^{th} sample after ordering the data (order statistics, i.e., $T_X(1) < T_X(2) < \dots < T_X(N)$) [24]. For the order statistics of an *i.i.d.* exponential sample set, it is well known that, $E[T_X(i)] = \mu_i/\lambda_x + \beta_{x,0}$ and $Var(T_X(i)) = \sigma_i^2/\lambda_x^2$, where $\mu_i = \sum_{j=N-i+1}^N 1/j$ and $\sigma_i^2 = \sum_{j=N-i+1}^N 1/j^2$ [21] and [24]. Therefore, $mse(\hat{\beta}_0)$ of equation (A.1) is $2(\lambda_x^2 + \lambda_y^2)/N^2\lambda_x^2\lambda_y^2$.

When we, on the other hand, apply the transformation first, then the transmitter will have

$$T_Z[i] = T_Y[i] - T_X[i] = \beta_0 + w_{\lambda_x, \lambda_y}[i], \quad (\text{A.2})$$

where $w_{\lambda_x, \lambda_y}[i]$ stands for a skewed (asymmetric) Laplace distribution that is $f_w(w) = (\lambda_y\lambda_x/(\lambda_y + \lambda_x))(e^{-\lambda_y|w|}u(w) + e^{-\lambda_x|w|}u(-w))$. Then the likelihood function for β_0 becomes

$$L(\beta_0) = f(T_Z|\lambda_x, \lambda_y, \beta_0) = \left(\frac{\lambda_x\lambda_y}{\lambda_x + \lambda_y}\right)^N \prod_{i=1}^N e^{-\lambda_i|T_Z[i]-\beta_0|}, \quad (\text{A.3})$$

where $\lambda_i = \lambda_y$ if $\beta_0 < T_Z[i]$ and $\lambda_i = \lambda_x$ otherwise. We can find the point which maximizes $L(\beta_0)$ as follows: First order the N observations. Since $L(\beta_0)$ is a combination of decaying-exponentials, it is enough to consider the points, $T_Z(i)$, only. Then the question is

$$L(T_Z(k)) \underset{T_Z(k+1)}{\overset{T_Z(k)}{\gtrless}} L(T_Z(k+1)).$$

That is we need to find k such that $L(T_Z(k))$ is maximized. By following these steps,

$$\begin{aligned} \frac{L(T_Z(k))}{L(T_Z(k+1))} &= \frac{e^{-(N-k)\lambda_y}}{e^{-k\lambda_x}} = e^{k\lambda_x - (N-k)\lambda_y} \\ k\lambda_x - (N-k)\lambda_y \underset{T_Z(k+1)}{\overset{T_Z(k)}{\gtrless}} 0 &\Rightarrow k(\lambda_y + \lambda_x) \underset{T_Z(k+1)}{\overset{T_Z(k)}{\gtrless}} N\lambda_y \\ \Rightarrow k\left(1 + \frac{\lambda_x}{\lambda_y}\right) \underset{T_Z(k+1)}{\overset{T_Z(k)}{\gtrless}} N &\Rightarrow k \underset{T_Z(k+1)}{\overset{T_Z(k)}{\gtrless}} \frac{N}{1 + \lambda_x/\lambda_y} \\ k &= \lceil \frac{N}{1 + \frac{\lambda_x}{\lambda_y}} \rceil, \end{aligned} \tag{A.4}$$

the maximum for the maximum likelihood function is found as,

$$\hat{\beta}_0 = T_Z\left(\lceil \frac{N}{1 + \frac{\lambda_x}{\lambda_y}} \rceil\right). \tag{A.5}$$

Furthermore, $mse(\hat{\beta}_0)$ of equation (A.5) can be shown to be $\sim C/N$, using some results from [19] and [45]. Recall that it was $\sim C/N^2$ without applying the transformation first.

Thus, it is clear that when the transformation is applied to the data coming from nodes X and Y , before estimating the relative clock parameters, under exponential noise, the performance is degraded severely.

APPENDIX B

DEFINITIONS AND THEOREMS FROM STATISTICAL ANALYSIS

The very first step in statistical analysis is to summarize the data. There are many different ways of doing it, such as moment summaries, percentiles and sufficient statistics.

Definition B.1. (Sufficient Statistics) A statistics $T(\mathbf{z})$ is sufficient for ϕ ($\mathbf{z} \sim F(\cdot|\phi)$) if $p(\mathbf{z}|T(\mathbf{z}))$ does not depend on ϕ .

There are different techniques to find appropriate sufficient statistics for a parameter and one of them is to use the following theorem.

Factorization Theorem. $T(\mathbf{z})$ is sufficient for ϕ iff $f(\mathbf{z}|\phi) = g(T(\mathbf{z})|\phi)h(\mathbf{z})$, for all \mathbf{z} and ϕ , with $f(\mathbf{z}|\phi)$ being the joint pdf of \mathbf{z} .

Definition B.2. (Completeness) A statistics $T(\mathbf{z})$ is complete if $E[g(T(\mathbf{z}))] = 0$ for all ϕ , implies $P(g(T(\mathbf{z}) = 0)) = 1$.

If a sufficient statistics is also complete, it seems to be the most successful one, in summarizing data, as the next theorem suggests.

Lehmann-Scheffe Theorem. If $T(\mathbf{z})$ is a complete sufficient statistics for ϕ , and $h(\mathbf{z})$ is an unbiased estimator for $f(\phi)$, then $E[h(\mathbf{z})|T(\mathbf{z})]$ is the UMVU estimator for $f(\phi)$.

VITA

Ilkay Sari was born in Canakkale, Turkey. He graduated with B.Sc. degree in electrical and electronics engineering from Bilkent University, Ankara, Turkey, in June 2004. He attended Texas A&M University to pursue a M.Sc. degree in the field of telecommunications and signal processing. His permanent address is: Zambak Sk. 29/2 34785 Umraniye-Istanbul Turkey. His e-mail address is: ilkay@ee.tamu.edu

The typist for this thesis was Ilkay Sari.