Probabilistic Graphical Models: An Application in Synchronization and Localization

DISSERTATION

zur Erlegung des akadimischen Grades

Doktor-Ingenieur (Dr.-Ing.)
im Fach Informatik

eingereicht an der
Mathematisch-Naturwissenschaftlichen Fakultät
der Humboldt-Universität zu Berlin

von
M.Sc. Meysam Goodarzi

Präsidentin der der Humboldt-Universität zu Berlin:
Prof. Dr. Julia von Blumenthal

Dekanin der Mathematisch-Naturwissenschaftlichen Fakultät:
Prof. Dr. Caren Tischendorf

Gutachter/innen:

1. Dr.-Ing Eckhard Grass
2. Dr.-Ing Milos Krstic
3. Prof. Guiseppe Caire

Tag der mündlichen Prüfung: 17. April 2023

Abstract

Mobile User (MU) localization in ultra dense networks often requires, on one hand, the Access Points (APs) to be synchronized among each other, and, on the other hand, the MU-AP synchronization. In this work, we firstly address the former, which eventually provides a basis for the latter, i.e., for the joint MU synchronization and localization (sync&loc). In particular, firstly, this work focuses on tackling the time synchronization problem in 5G networks by adopting a hybrid Bayesian approach for clock offset and skew estimation. Specifically, we investigate and demonstrate the substantial benefit of Belief Propagation (BP) running on Factor Graphs (FGs) in achieving precise network-wide synchronization. Moreover, we take advantage of Bayesian Recursive Filtering (BRF) to mitigate the time-stamping error in pairwise synchronization. Finally, we reveal the merit of hybrid synchronization by dividing a large-scale network into common and local synchronization domains, thereby being able to apply the most suitable synchronization algorithm (BP- or BRF-based) on each domain.

Secondly, we propose a Deep Neural Network (DNN)-assisted Particle Filter-based (DePF) approach to address the MU joint sync&loc problem. In particular, DePF deploys an asymmetric time-stamp exchange mechanism between the MUs and the APs, which provides information about the MUs' clock offset, skew, and AP-MU distance. In addition, to estimate the Angle of Arrival (AoA) of the received synchronization packet, DePF draws on the Multiple Signal Classification (MUSIC) algorithm that is fed by the Channel Impulse Response (CIR) experienced by the sync packets. The CIR is also leveraged on to determine the link condition, i.e. Line-of-Sight (LoS) or Non-LoS (NLoS). Finally DePF capitalizes on particle Gaussian mixtures which allow for a hybrid particle-based and parametric BRF fusion of the aforementioned pieces of information and jointly estimate the position and clock parameters of the MUs.

Kurzfassung

Die Lokalisierung von mobilen Nutzern (MU) in sehr dichten Netzen erfordert häufig die Synchronisierung der Access Points (APs) untereinander. Erstens konzentriert sich diese Arbeit auf die Lösung des Problems der Zeitsynchronisation in 5G-Netzwerken, indem ein hybrider Bayesischer Ansatz für die Schätzung des Taktversatzes und des Versatzes verwendet wird. Wir untersuchen und demonstrieren den beträchtlichen Nutzen der Belief Propagation (BP), die auf factor graphs läuft, um eine präzise netzwerkweite Synchronisation zu erreichen. Darüber hinaus nutzen wir die Vorteile der Bayesischen Rekursiven Filterung (BRF), um den Zeitstempel-Fehler bei der paarweisen Synchronisierung zu verringern. Schließlich zeigen wir die Vorzüge der hybriden Synchronisation auf, indem wir ein großes Netzwerk in gemeinsame und lokale Synchronisationsdomänen unterteilen und so den am besten geeigneten Synchronisationsalgorithmus (BP- oder BRF-basiert) auf jede Domäne anwenden können.

Zweitens schlagen wir einen Deep Neural Network (DNN)-gestützten Particle Filter-basierten (DePF)-Ansatz vor, um das gemeinsame MU-Sync&loc-Problem zu lösen. Insbesondere setzt DePF einen asymmetrischen Zeitstempel-Austauschmechanismus zwischen den MUs und den APs ein, der Informationen über den Taktversatz, die Zeitverschiebung der MUs, und die AP-MU Abstand liefert. Zur Schätzung des Ankunftswinkels des empfangenen Synchronisierungspakets nutzt DePF den multiple signal classification Algorithmus, der durch die Channel Impulse Response (CIR) der Synchronisierungspakete gespeist wird. Die CIR wird auch genutzt, um den Verbindungszustand zu bestimmen, d. h. Line-of-Sight (LoS) oder Non-LoS (NLoS). Schließlich nutzt DePF particle Gaussian mixtures, die eine hybride partikelbasierte und parametrische BRF-Fusion der vorgenannten Informationen ermöglichen und die Position und die Taktparameter der MUs gemeinsam schätzen.

Preface and Declaration

I declare that I have completed the thesis independently using only the aids and tools specified. I have not applied for a doctor's degree in the doctoral subject elsewhere and do not hold a corresponding doctor's degree. I have taken due note of the Faculty of Mathematics and Natural Sciences PhD Regulations, published in the Official Gazette of Humboldt-Universität zu Berlin no. 42/2018 on 11/07/2018. Furthermore, I declare that the written and electronic version of the dissertation written by me in the University Library agrees with the accepted dissertation.

For all materials, i.e., the chapters of this manuscript and the papers, the initial idea was from my side. Moreover, I carried out the literature review on the topics relevant to the ideas, formulated the problems, suggested the solutions, implemented the simulation platforms, performed the analyses, and prepared the manuscript. Valuable feedback, guides on the direction of research, and cross-validation of the obtained results were provided by my supervisor. My co-authors contributed by providing feedback on the manuscript and the insightful discussions we had on the technical aspects of the papers. To the best of my knowledge and belief, this work was prepared without aid from any other sources except where indicated. Any reference to the materials previously published by any other person has been duly acknowledged. This work contains no material which has been submitted or accepted for the award of any other degree in any institution.

Chapters 3 and 4 are based on the following published research papers

- Journal publications:
  - M. Goodarzi, V. Sark, N. Maletic, J. Gutiérrez, G. Caire, and E. Grass, "DNN-assisted particle-based joint synchronization and localization," *Transactions on Communications*, 2022.
  - M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez, and E. Grass, "Synchronization in 5G networks: a hybrid Bayesian approach toward clock offset/skew estimation and its impact on localization," *EURASIP Journal on Wireless Communications and Networking*, 2021(1):1–22, 2021.
- Conference publications:
  - M. Goodarzi, N. Maletic, J. Gutiérrez, and E. Grass, "Bayesian Joint Synchronization and Localization Based on Asymmetric Time-stamp Exchange," In *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–7. IEEE, 2020.

- M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez, and E. Grass, "A Hybrid Bayesian Approach towards Clock Offset and Ekew Estimation in 5G Networks," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE, 2020, pp. 1–7.
- M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez, and E. Grass, "Synchronization in 5G: a Bayesian Approach.," in *European Conference on Networks and Communications (EuCNC)*. IEEE, 2020, pp. 194–199.

And the following are the further contributions of the author not directly used in this thesis.

- T. Cogalan, D. Camps-Mur, Daniel, J. Gutiérrez,S. Videv, V. Sark,J. Prados-Garzon, J. Ordonez-Lucena, H. Khalili, F. Cañellas, A. Fernández-Fernández, M. Goodarzi, A. Yesilkaya, R. Bian, S. Raju, M. Ghoraishi, H. Haas, O. Adamuz-Hinojosa, A. Garcia, C. Colman-Meixner, A. Mourad, E. Aumayr, "5G-CLARITY: 5G-Advanced Private Networks Integrating 5GNR, WiFi, and LiFi," In *IEEE Communications Magazine*, IEEE, 2022.
- N. Maletic, L. Lopacinski, M. Goodarzi, M. H. Eissa, J. Gutiérrez, and E. Grass, "A Study of LOS MIMO for Short-range Sub-THz Wireless Links," In *Osnabruck conference*, IEEE, 2021.
- M. Goodarzi, A. Krishnamoorthy, R. Schober, and M. Breiling, "Resource Allocation for Outdoor-to-indoor Amplify-and-forward SUDAS with Independent Relay Processing," In *SCC 2019; 12th International ITG Conference on Systems, Communications and Coding*, pages 1-6. VDE, 2019.
- M. Goodarzi, N. Maletic, J. Gutiérrez, V. Sark, and E. Grass, "Next-cell Prediction Based on Cell Sequence History and Intra-cell Trajectory," In *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, pages 257–263. IEEE, 2019.

Berlin, May 22, 2023

Meysam Goodarzi

Contents

Glossary

Abbreviations

5G	Fifth Generation
AP	Access Point
AoA	Angle of Arrival
BMCA	Best Master Clock Algorithm
BN	Bayesian Network
DBN	Dynamic Bayesian Network
BRF	Bayesian Recursive Filtering
L-BRF	Linearized Bayesian Recursive Filter
BS	Base Station
BP	Belief Propagation
CIR	Channel Impulse Response
COS	Commercial Off-the-Shelf
CRB	Cramer-Rao-Bound
CSI	Channel State Information
DAG	Directed Acyclic Graph
DePF	DNN-assisten Particle-based Bayesian Filter
EKF	Extended Kalman Filter
ESPRIT	Estimation of Signal Parameters via Rational Invariance Techniques
FDMA	Frequency Division Multiple Access
FG	Factor Graph
FTM	Fine Time Measurement
GPS	Global Positioning System
KKT	Karush-Kuhn-Tucker
KL	Kullback-Leibler
LoS	Line-of-Sight
NLoS	Non-Line-of-Sight
ML	Machine Learning
MIMO	Multi-Input Multi-Output
MMSE	Mimimum Mean Square Error
mm-Wave	Millimeter Wave
MN	Master Node
MU	Mobile User
MUSIC	Multiple Signal Classification
OFDMA	Orthogonal Frequency Division Multiple Access
PF	Particle Filter
PGM	Probabilistic Graphical Model

PGMF	Particle Gaussian Mixture Filter
PDF	Probability Density Function
PTP	Precision Time Protocol
QoS	Quality of Service
RMSE	Root Mean Square Error
STD	Standard Deviation
sync&loc	Synchronization and Localization
ToA	Time of Arrival
WSN	Wireless Sensor Network

Operators

$(\cdot)^*$	Complex conjugate
$(\cdot)^{-1}$	Inverse of a matrix
$(\cdot)^T$	Transpose of a matrix
$(\cdot)^H$	Hermitian transpose of a matrix
$\mathrm{Tr}(\cdot)$	Trace of a matrix
$\det(\cdot)$	Determinant of a matrix
$\mathbb{E}\{\cdot\}$	Statistical expectation
$\|\|\cdot\|\|_2$	Rank of a matrix
$\mathrm{diag}(x_1, ..., x_k)$	Diagonal matrix with diagonal elements $\{x_1, ..., x_k\}$
$\odot$	Hadamard product of matrices
$\bullet$	Inner scalar product of vectors
$[x]^+$	Returns $\begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$

Symbols

$\mathbf{I}_n$	Identity Matrix with size of $n \times n$
$\mathbf{0}_n$	All-zero Matrix with size of $n \times n$
$\sim$	stands for “is distributed as"
$\mathbf{A}$	A matrix
$\mathbf{a}$	A vector
$\mathbf{a}[n]$	n-th element of vector $\mathbf{a}$
$\mathcal{N}(\mathbf{x}\|\boldsymbol{\mu}, \boldsymbol{\Sigma})$	random vector $\mathbf{x}$ distributed as Gaussian with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{U}(a, b)$	a continuous uniform probability distribution in the interval between a and b with the probability level of $\frac{1}{\|a-b\|}$
$\mathrm{ne}(X_i)$	Set of neighboring nodes of node X_i

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Why Synchronization and Localization Are Important?

The fifth Generation (5G) of mobile networks is expected to deliver a wide range of location-based services [1]–[3] such as user tracking [4] and location-assisted beamforming [5]. To pave the way for those services, a myriad of precise positioning techniques have been introduced in the literature, the majority of which rely on the cooperation between the Access Points (APs) serving the Mobile Users (MUs) [6]. Such APs are expected to be deployed in a high spatial density manner to meet the increasing traffic demand, which results in MUs being in Line-of-Sight (LoS) with a number of APs for most of the time. Moreover, these APs are likely to be equipped with antenna arrays and Fine Time Measurement (FTM) capability introduced in several standards, e.g., IEEE 802.11 [7]. The former facilitates the Angle of Arrival (AoA) estimation, while the latter allows for the inter-AP and AP-MU time-stamp exchange, by means of which inter-AP and AP-MU synchronization and distance measurements are enabled. In particular, to estimate the MU's location, these techniques capitalize on the time measurements carried out between the agents, i.e., MUs and APs, requiring them to have a common time base [8]. Therefore, for the cooperative approaches to function, the APs need to be accurately synchronized among each other as well as with MUs [9], [10]. In summary, MU localization depends, on one hand, on the inter-AP synchronization, and, on the other hand, on the MU-AP synchronization. The latter can be performed jointly with the MU localization itself. Therefore, in this section, we focus on the inter-AP synchronization and MU joint synchronization and localization (sync&loc) problems and present their corresponding backgrounds necessary for this dissertation.

1.2 Background

1.2.1 Synchronization

Time synchronization is primarily defined as the time alignment of multiple devices, often based on a common time base. A wide variety of networks, from Wireless Sensor Networks (WSN) [11] and wireless communication networks [12], [13] to internet of things [14] can deliver most of their services only when they rely on a common basic time. Traditionally, such an alignment is achieved through a Global Positioning System (GPS) at every Base Station (BS), which not only incurs a high financial cost, but also demands maintenance [12]. In addition, the deployment of GPS in indoor environments is limited as its signals can barely penetrate the buildings [15]. Furthermore, in the case where satellite systems are unavailable, the BSs will be immediately out of service. To prevent such network shutdowns, we typically resort to an alternative solution based on the construction of time synchronization networks. Such networks deliver timing information via the transport networks, usually using synchronization protocols and algorithms.

The network synchronization algorithms are traditionally developed to satisfy the maximum end-to-end allowable time error. Nevertheless, for a number of applications such as localization, the absolute time error is of minor importance, rather it is only necessary to achieve a certain precision at the local level. Given the aforementioned points, to satisfy the time requirements of the absolute time error at the global level and the relative time error in the local areas, the synchronization network architecture of 5G is suggested by [12] to comprise common synchronization areas and various synchronization clusters as shown in Figure 1.1. The green and purple highlighted sections denote the common area and clusters, respectively. If the networks are equipped with multiple synchronization algorithms (or a combination thereof), each domain can, based on its topology and capabilities, leverage the most suitable algorithm. In this manner, the requirement of the relative time error is easier to satisfy in the clusters. Such an approach brings a great deal of flexibility for the operators as well. For instance, if one cluster area has an ultrahigh time accuracy requirement such as providing positioning service, then a better synchronization technology can be employed only for that specific area, without any need to update all technologies for large-scale networks.

Generally, state-of-the-art synchronization algorithms rely on time-stamp exchange among the nodes to obtain the statistics required to estimate the clock

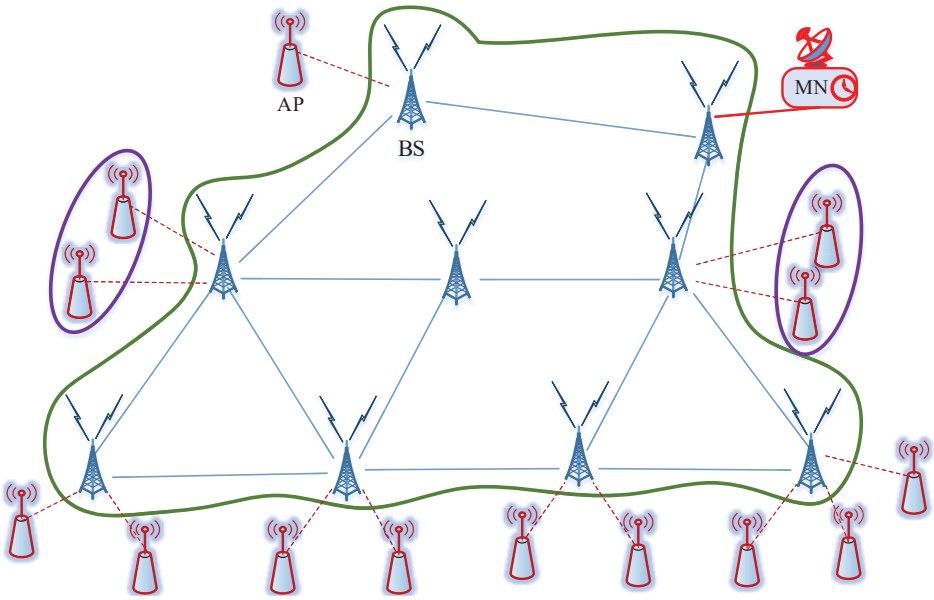


Figure 1.1: Division of the synchronization network to a common area (surrounded by a green line) and several clusters (denoted by the purple ellipses).

parameters. Such a time-stamp exchange mechanism is already defined in several standards, i.e., IEEE 802.12 under the name fine time measurement [16], and can be implemented by means of existing protocols, e.g., Precision Time Protocol (PTP). Perhaps PTP, also denoted as IEEE 1588 [17], is the most well-known synchronization protocol employed in a wide variety of applications. PTP uses hardware time-stamping and pairwise communication between nodes to obtain the pairwise time-stamps. The pairwise statistics derived from the time stamps, can then be either directly employed in a structural way to perform network synchronization, e.g., by pairwise synchronization in a layer-by-layer manner [18]–[20], or can be utilized in a broader context to jointly estimate the clock parameters of all nodes in a network [21]–[24]. The former is eminently suitable for tree structure networks, while the latter can, in addition to tree networks, address the synchronization problem in mesh networks. For example, in Figure 1.1, the common area, highlighted in green, is an example of a mesh network, while the clusters, highlighted in purple, are examples of tree networks. We elaborate on each type of algorithm in Chapter 3 and reveal the merit of each when employed in their corresponding synchronization areas, i.e., common areas or clusters.

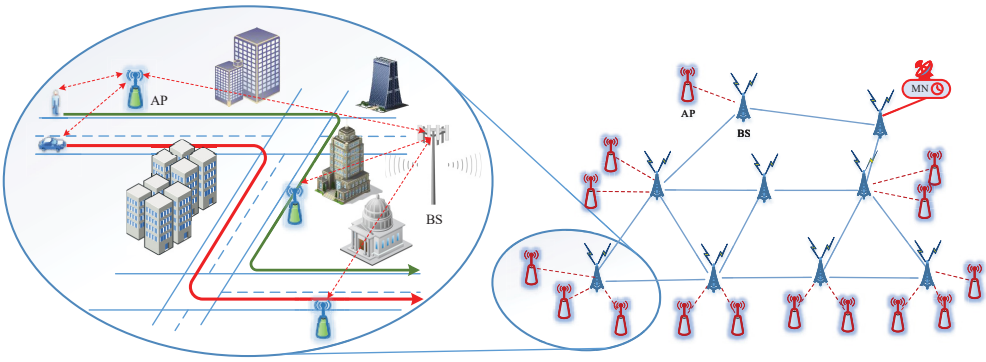


Figure 1.2: Mobile user joint synchronization and localization scenario.

1.2.2 Joint Mobile User Synchronization and Localization

In many systems, MU location is obtained from the time measurements between the agents, e.g., in wireless communication, the agents are MU, APs, and BSs. Typically, the distance between the agents can be obtained from the time of arrival of a packet containing its own departure time-stamp. The MU location can then be geometrically estimated by means of the obtained distances. Nevertheless, such schemes presuppose a common time base among the agents, meaning that they are required to be synchronized among each other with a high precision. In this dissertation, the inter-agent synchronization problem is split into two phases. In the first phase, as explained in Section 1.2.1, the BSs and the APs are synchronized (shown in Figure 1.1), while in the second, the MU is synchronized with its serving APs. Furthermore, the second phase is combined with the MU localization representing the joint sync&loc problem. Figure 1.2 depicts a zoomed-in snapshot of the network depicted in Figure 1.1 visualizing the scenario considered for joint MU sync&loc. Specifically, MUs are moving in an urban area where they can establish links to the available APs in their vicinity, which are in turn backhauled by a BS.

The MU joint sync&loc problem has been the topic of many research works, e.g., [25]–[28], majority of which rely on Time of Arrival (ToA) or time-stamp exchange among the agents to estimate the clock and position parameters in a centralized or distributed manner. Nevertheless, they do not exploit additional available information from agents which are not collected explicitly for the purpose of synchronization or localization. An example of such data is the Channel Impulse Response (CIR) available at APs. In particular, APs are expected to be equipped with multiple antennas to ensure high signal-to-noise-ratio and provide services such as beamforming [29]. Therefore, the CIR estimation capability in such multi-antenna systems is anticipated. Furthermore, the CIR is essential to a number

of tasks, e.g., equalization [30] and resource allocation [31], [32], rendering its estimation vital for reliable functionality of future networks.

Some information about the location of MU is intrinsic to the AP-MU CIR. There have been attempts in [33]–[35] to perform localization by means of CIR and Machine Learning (ML) algorithms. Although a CIR contains invaluable information about the environment and the scatterers wherein, it cannot alone serve for high-precision localization, i.e., for achieving an accuracy of below one meter. Nevertheless, the CIR can be used to, for example, determine the link condition, i.e., Non-Line-of-Sight (NLoS) or LoS [36]. This is considered as one of the most crucial decisions in wireless communication networks. The reason is that measurements taken under NLoS conditions are highly erroneous and can lead to a poor estimation of the location parameters, if not mitigated or dropped. Before the advent of ML, such a decision used to be made based on simple methods relying only on one aspect of the communication link, e.g., signal strength or ToA [37]. However, the remarkable ability of ML algorithms, in particular DNNs, in extracting task-based features from the data (CIRs in the case of this dissertation) has stimulated a number of research works in NLoS identification, albeit in indoor environments. In this dissertation, we draw on the same idea and employ CIR and DNNs for NLoS identification in outdoor environments.

Another domain where the CIR can be leveraged to facilitate more accurate localization is AoA estimation, for which there exists a wide spectrum of approaches. Algorithms such as Multiple Signal Classification (MUSIC) [38], Estimation of Signal Parameters via Rotational Invariant Techniques (ESPRIT) [39], and their variants [40] rely on the received signal to estimate the AoA in wireless communication systems. They can treat CIRs as received signals and perform AoA estimation, which can eventually be employed to enhance the accuracy of localization [41]. If CIRs are collected under LoS conditions, the outcome of these algorithms can be fused with the time-stamp exchange measurements to obtain a more precise estimation of the location parameters.

All above-mentioned pieces of information, i.e., time-stamps, link condition, and estimated AoA, can be combined to obtain the clock offset, skew, and the MUs' locations. The challenge is, however, the manner in which they are fused. In particular, defining variables that are descriptive of multiple facets of the entire phenomenon and the relation among them is integral to any accurate fusion algorithm. In what follows, we further explain the tools we can employ to model

the synchronization and localization problems and, subsequently, to compute the MUs' clock and position parameters.

1.2.3 Modeling of Synchronization and Localization Problem

From the mathematical point of view, all above-mentioned problems, i.e., synchronization and MU localization, can be formulated as *parameter estimation* problems. Specifically, synchronization is defined as the clock parameter (offset and skew) estimation, while localization is considered as the location parameter estimation, i.e., (x_i, y_i), where x_i and y_i denote the location of the i-th MU on the x and y axes[1]. In other words, obtaining the value of those parameters is equivalent to performing synchronization and localization.

Parameter estimation is defined as *to experimentally determine the values of the parameters of a particular system* [42]. Therefore, the cornerstone of any parameter estimation is the data obtained through measurements and the statistics derived therefrom. In the work presented in this dissertation, the primary source of such data is time-stamp exchange among the nodes as explained in Section 1.2.1. Nevertheless, one of the aspects which distinguishes this work, is that it also leverages other types of data which is not explicitly collected for the purpose of localization or synchronization, i.e., CIRs. In particular, such approach allows for taking advantage of all available information to reach a more accurate and reliable estimation of the parameters of interest.

The data obtained from experiments contain valuable statistics about the clock and position parameters. To derive these statistics, one needs to turn to the tools available in probability theory and statistics. However, knowing the statistics alone does not assist in parameter estimation if the relationships between the parameters are not clear. Generally, such modelings are conducted in the context of graph theory. In brief, what a successful parameter estimation requires, is not only the statistics derived from the data, but also a descriptive model of interaction between the parameters. Both of above-mentioned topics are merged in the framework of Probabilistic Graphical Models (PGMs). The PGMs are capable of modeling a real-world system by defining relevant parameters and their interactions by combining tools stemming from probability and graph theory. The framework is quite general in that many of the commonly proposed statistical models (Kalman filters, hidden Markov models, etc.) can be described as graphical models [43]. Once the models are ready, a wide variety of inference

[1] Throughout this dissertation, we focus on the two-dimensional localization and, therefore, we do not mention the z axis.

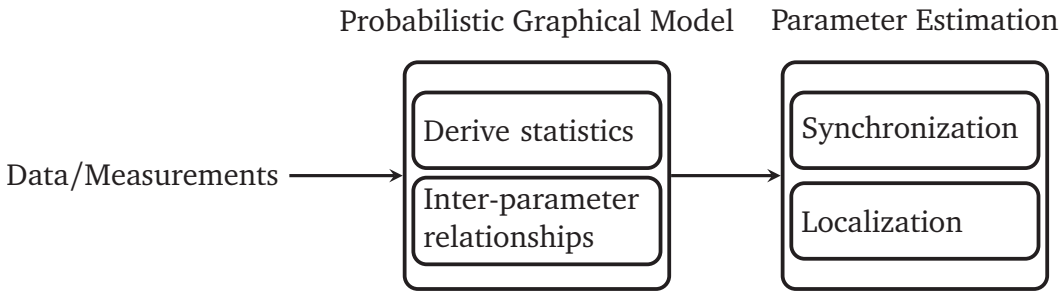


Figure 1.3: Summary of the process leading to synchronization and localization.

algorithms can be run to perform parameter estimation. In brief, PGMs model the random variables and their corresponding interconnections, paving the way to perform parameter estimation with the aid of inference algorithms. Figure 1.3 summarizes the above-mentioned explanations.

Compared to other existing and popular frameworks, i.e., Neural networks (NNs), PGMs allow for combining heuristics (by embedding prior knowledge obtained from basic principles, intuition, etc.) and data to design more intelligent algorithms. Furthermore, in contrast to the NNs, which lack interpretability, PGMs provide detailed intermediate steps of the reasoning process leading to a particular inference [44]. Nevertheless, PGMs and NNs should not be considered as mutually exclusive frameworks, rather two different frameworks, each suitable for certain types of problems and can be complementary to each other, e.g., by deploying NNs in the intermediate steps of reasoning of PGMs, as we do in Chapter 4 of this dissertation. The details of PGM are explained in Chapter 2.

1.3 Dissertation Contributions and Organization

This dissertation aims at developing a precise MU localization technique, thereby facilitating the provision of location-based services. It firstly focuses on the network synchronization, which lays the ground for a precise localization. Subsequently, it tackles the MU joint sync&loc at the edge of wireless communication networks.

Network synchronization in the Chapter 3 of this dissertation relies mainly on the research published in [45]–[47]. In particular, in these publications, a novel DBN- and BRF-based pairwise synchronization algorithm based on time-stamp exchange has been developed. In addition, the principles of the FG- and BP-based network-wide synchronization have been presented. Finally, a hybrid synchronization approach was proposed to address the network synchronization problem and to meet the local and global time error requirements.

In addition to network synchronization, Chapter 4 of this dissertation provides contributions towards joint synchronization and localization. These contributions are based on the materials published in [48], [49]. Specifically, in the aforementioned publications, the principles of CIR-based AoA estimation have been presented. Furthermore, a novel DNN-based NLoS identification scheme has been developed to determine the communication link condition. All aforementioned pieces as well as time-stamp exchange measurements have been combined by two proposed algorithms, namely linearized BRF (L-BRF) and DNN-assisted Particle-based Bayesian (DePF) joint sync&loc algorithm, to estimate the joint probability distribution of MU's clock and position parameters using Particle Gaussian Mixture Filters (PGMFs).

The pillar of all above-mentioned contributions, i.e., hybrid network synchronization and joint MU synch&loc, is PGMs. In particular, DBN, BRF, FG, BP, and PGMF can be brought under the umbrella of PGMs, which are extensively deployed across multiple applications to represent the conception of real-world phenomena by means of interactive random variables in the shape of a graph [50], [51]. Therefore, as a primary step, Chapter 2 of this dissertation begins with elaborating on PGMs and the elements whereof integral to the algorithms developed in this dissertation.

The contribution of this thesis is summarized as follows:

- The principles of DBN, BRF, FG, and BP are presented and discussed in the context of PGMs.
- The fundamental ideas behind BP-based network-wide and BRF-based pairwise synchronization are described.
- A hybrid statistical synchronization algorithm is developed by combining two Bayesian approaches, i.e., BP- and BRF-based.
- The performance of the hybrid approach is analysed in terms of clock offset and skew estimation error.
- A DNN-based NLoS identification algorithm is developed, and the principles of AoA estimation are presented.
- A BRF-based joint sync&loc algorithm is proposed to perform MU joint synchronization and localization.
- A DNN-assisted PF-based joint sync&loc algorithm is proposed to obtain an estimation of the clock parameters and position of an MU in a hybrid parametric and particle-based manner.

- The performance of the proposed approaches are analysed and compared with the state-of-the-art methods with the aid of detailed simulations in a challenging real-world scenario.

The rest of this dissertation is structured as follows: In Chapter 2, the preliminaries are extensively explained. In particular, the details of PGMs are explained and discussed. Chapter 3 deals with the estimation methods for network-wide, pairwise, and hybrid synchronization. Furthermore, the detailed analysis of the synchronization algorithms developed in this work is presented and compared with the state-of-the-art schemes by means of computer simulations. Chapter 4 illustrates the principles of the joint sync&loc algorithms. Moreover, an extensive analysis is provided and a comparison with the state-of-the-art is provided. Finally, Chapter 5 concludes this work and points to the future work. Figure 1.4 depicts an in-detail representation of this dissertation as well as the structural relationships among the chapters.

Remark: In order to make it easier for the readers to read this dissertation, upon moving to a new chapter, we expand the previously defined acronyms when used for the first time in that specific chapter.

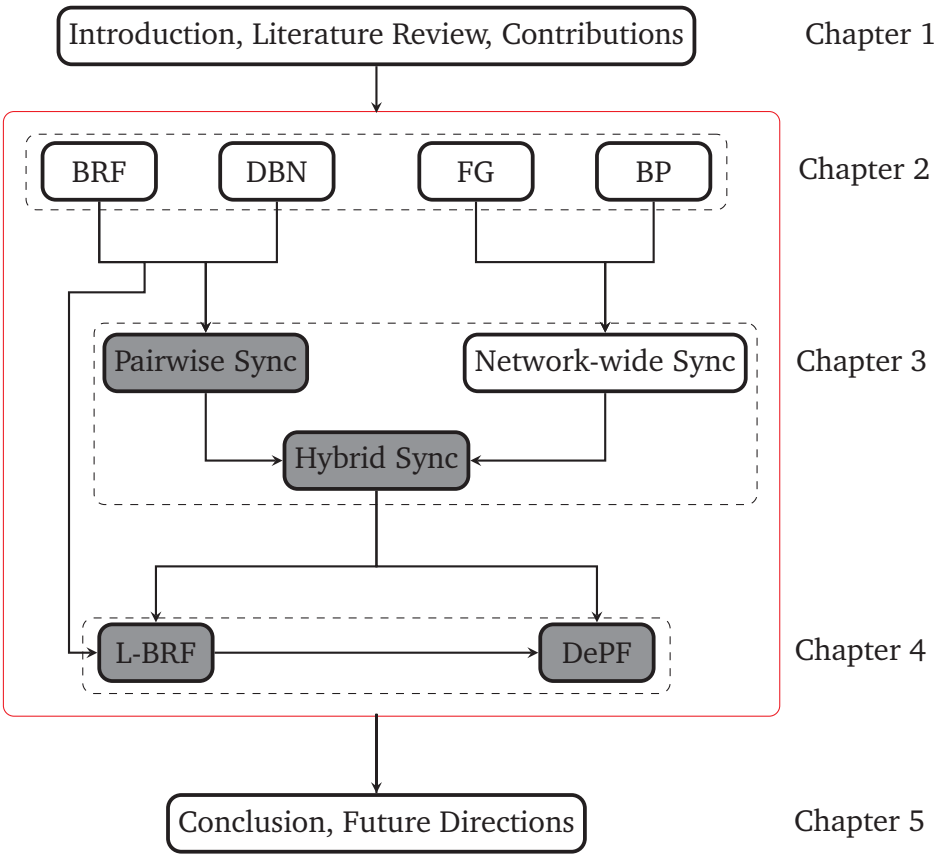


Figure 1.4: Flow of the dissertation and relations between the chapters. Gray colored rectangles denote the parts where this dissertation delivers its main contributions.

Chapter 2

Review of Probabilistic Graphical Models

In the previous chapter, we briefly discussed that Probabilistic Graphical Models (PGMs) can accomplish a successful parameter estimation by merging the statistics derived from the data/measurements with a descriptive model of interactions between the parameters. It was also mentioned that when compared to other existing frameworks such as Neural Networks (NNs), PGMs permit the embedding of priors as well as interpretability. In this chapter, we provide more details on several PGMs that will be employed in the next chapters to perform synchronization and localization. Throughout the whole chapter, wherever it is possible, we elaborate on the relation between these two topics and the tools described in this chapter.

2.1 Introduction

PGMs are widely employed across different fields to represent our understanding of various phenomena in the world with the aid of interactive random variables shown in the form of a graph. The probabilistic feature stems from the fact that these models allow for dealing with a large amount of uncertainty due to different reasons, e.g., having only partial knowledge, noisy observations, etc. Moreover, they employ graphical representations from computer science to offer the possibility of representing complicated models involving a large number of variables. Such a combination of ideas from probability theory and computer science provides, on the one hand, an intuitive and compact data structure for capturing high-dimensional Probability Distribution Functions (PDFs), and, on the other hand, with a set of methods and algorithms for efficient reasoning [52].

Given that each of the random variables $X_1, \cdots, X_n$ reflects the impact of a facet of a real-world phenomenon, the first goal is to capture the uncertainty about the possible states of the phenomenon in question in terms of the joint probability distribution of the random variables, i.e., $p(X_1, \cdots, X_n)$. It bears emphasizing that even in simple cases, e.g. binary X and $n = 10$, the number of possible states is substantially large (2^{10}), rendering the computation of joint distribution extremely costly. What PGMs do instead, is to search for and exploit any structures in the data to alleviate such burden.

Once the interaction between the random variables is graphically formed and the probabilistic model is constructed, any question of interest can be answered by running inference on the joint distribution. This can typically require a great deal of computational effort and, therefore, is a crucial step if a successful modeling is to be achieved. Figure 2.1 summarizes the above-mentioned steps.

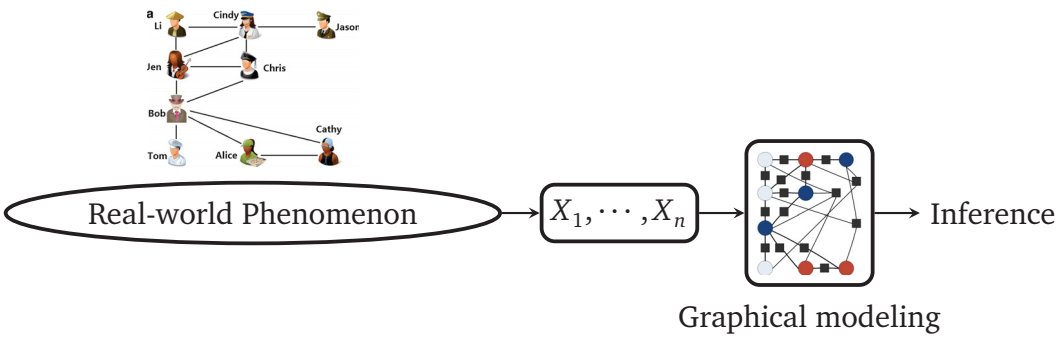


Figure 2.1: Steps taken in PGMs to model a phenomenon.

PGMs are generally classified into many categories such as Bayesian Network (BN), clique tree, Factor Graph (FG), Markov random fields, etc. All the aforementioned models have their own strengths and weaknesses. Nonetheless, they are different representations of the same information and can be converted between them. Which model to employ is highly dependent on the type of the problem one deals with. For the purpose of this work, we focus on two powerful and widely used ones, namely, BN and FG. In the sections to come, we will elaborate on the upsides and downsides of each as well as the inference algorithms that can be run on them.

2.2 Bayesian Networks

BNs, also called belief networks or causal networks, comprise a set of nodes connected to each other via directed edges. The former represents a set of

random variables while the latter indicates the influence of its corresponding nodes on each other, or alternatively, their conditional dependencies. Figure 2.2 depicts a simple example of a BN.

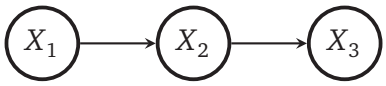


Figure 2.2: An example of a BN.

We know from the chain rule of probability theory that any joint distribution can be written in the form of a product of a set of conditional distributions [53]. For instance, using the chain rule we can write

$$p(X_1, X_2, X_3) = p(X_3|X_2, X_1)p(X_2|X_1)p(X_1). \tag{2.1}$$

whose corresponding BN is given in Figure 2.3, which is an example of fully connected BNs, where each random variable is connected to all lower-numbered variables. Therefore, a BN is considered a directed acyclic graph, where following the edge direction never leads to a loop.

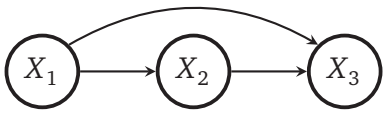


Figure 2.3: An example of a fully connected BN.

Given this, the joint probability distribution modelled by a belief network can be given by [54],

$$p(X_1, \cdots, X_N) = \prod_{i=1}^{N} p(X_i|\text{pa}(X_i)) \tag{2.2}$$

where $\text{pa}(X_i)$ represents the parental variables of X_i. Consequently, the joint distribution corresponding to the PGM in Figure 2.2 is given by

$$p(X_1, X_2, X_3) = p(X_1)p(X_2|X_1)p(X_3|X_2). \tag{2.3}$$

Note that the Markov condition is implicit in Equation (2.2). That is, every node of a BN is conditionally independent of its non-descendants given its parents.

There are, in general, many types of BNs, each featuring a special modeling capability. One specific form, incorporating the temporal aspect of a phenomenon, is called Dynamic Bayesian Network (DBN). Such a model is particularly useful when the same type of inference needs to be repeated on a structurally fixed

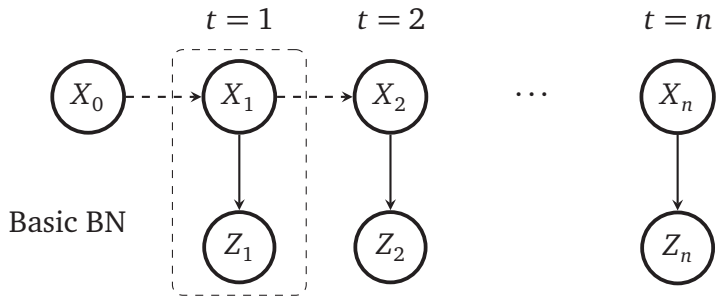


Figure 2.4: An example of a DBN with time indices $1, \cdots, n$.

BN, albeit with different variable values [55]. In the sequel, we describe the properties of DBNs in detail.

2.2.1 Dynamic Bayesian Networks

DBNs are a singly connected variant of BNs specifically suitable for modeling of time series [56]. There is in principle no difference between the variables and edges of DBNs and those of BNs. Nevertheless, the variables carry a time index accounting for their temporal dimension, what also enables DBNs to model a dynamic phenomenon. All variables with the same time index are considered as a *state* of DBN. The states of DBN satisfy the Markovian property, that is, the state of the system at time t depends only on its immediate past, i.e., its state at time $t-1$ [57].

Figure 2.4 depicts an example of a DBN which is constructed by beginning from an initial state X_0 and then repeating a simple BN block comprising two variable nodes and one directed edge. The dash arrow designates the temporal evolution of the DBN while the solid one connects the nodes in the basic BN. As mentioned before, there is no difference between these two edges and we only differentiate them to properly describe the temporal behavior of the DBN.

The joint PDF corresponding to the DBN in Figure 2.4 can be written as

$$p(X_1, \cdots, X_n, X_{n+1}, Z_1, \cdots, Z_n) = p(X_0) \prod_{t=1}^{n} p(X_t|X_{t-1}) p(Z_t|X_t), \tag{2.4}$$

where $p(X_t|X_{t-1})$ accounts for the temporal dependencies between the states while $p(Z_t|X_t)$ indicates the dependencies between X and Z at time t. It turns out that the DBN in Figure 2.4 is capable of modeling of a wide range of phenomena in practice. For example, in the context of synchronization between two nodes, the clock parameters of the nodes can be modeled by X while the time measurements can be represented by Z, which are periodically collected. Similar modeling can

be done for the localization problem. In the Section 2.4.1, we deal with the inference on DBNs and in particular the methods applicable to the model shown in Figure 2.4.

2.3 Factor Graphs

FGs are bipartite graphs used to represent the factorization of a PDF. An FG consists of a number of nodes, each representing a variable, and several factor nodes, each being a function of its neighboring variables. The variable nodes are normally represented by circles while factors are denoted by rectangles. Furthermore, the edges between all nodes are undirected. Figure 2.5 depicts an example of an FG with three variable nodes X_1, X_2 and X_3 connected to each other via the factor nodes denoted by rectangles, each representing a function $f(\cdot)$.

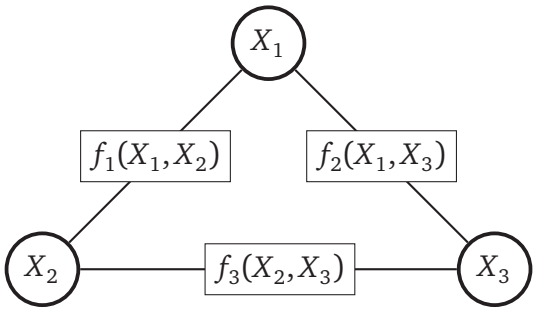


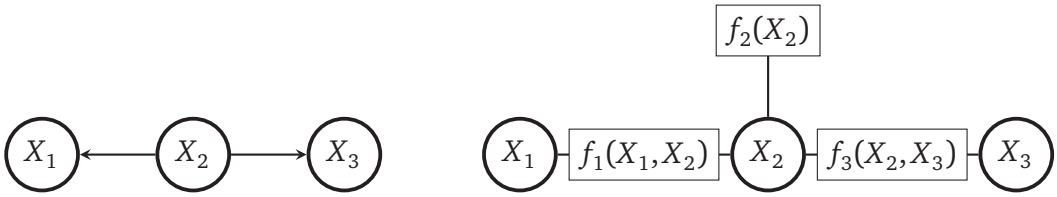
Figure 2.5: An example of FGs.

In general, the joint probability distribution corresponding to a factor graph is given by

$$p(X_1, \cdots, X_n) = \prod_{j=1}^{m} f_j(S_j), \tag{2.5}$$

where $f_j(\cdot)$ denotes the j-th factor, whose argument is the j-th subset of the variables represented by S_j. We know from Equation (2.2) that the BNs also represent their corresponding joint distributions by means of factorization on the basis of conditional dependencies. Given that, BNs and FGs can be transformed into each other, whereby each factor is equal to a specific conditional probability or a multiplication of several. Figure 2.6b depicts the factor graph equivalent to the BN in Figure 2.6a where

$$f_1(X_1,X_2) = p(X_1)p(X_2|X_1), \qquad f_2(X_2) = p(X_2), \qquad f_3(X_2,X_3) = p(X_3|X_2).$$



(a) An example of a BN. (b) The FG equivalent to the BN.

Figure 2.6: BN and its equivalent FG.

It is worth mentioning that both BN and FG are different representations of the same information, i.e., the joint probability distribution of a set of random variables. In fact, if we had two exact inference algorithms for an equivalent BN and FG, the results of inference would be equal. Nevertheless, depending on the context and the type of problem we are dealing with, one might be preferred over another to facilitate inference. An example where FG eases the inference is the *sum-product* algorithm, also known as *Belief Propagation* (BP). In the next section, we delve into the details of this algorithm.

2.4 Inference

Assuming that all relevant variables $X_1, \cdots, X_n$ for the phenomenon in question are identified and the probabilistic model $p(X_1, \cdots, X_n)$ is constructed, *inference*, or alternatively *reasoning*, is then performed by introducing evidence that sets the variables in known states and, consequently, allows for computing any probability of interest conditioned on this evidence [58]. Among all the possible inferences, *marginal inference* is of crucial importance as a substantial number of real-world inferences can be brought under the umbrella of this term. The term *marginal* is defined as the distribution of a subset of variables calculated from the joint distribution. For example,

$$p(X_1, X_2) = \int p(X_1, \cdots, X_n) dX_3 \cdots dX_n, \tag{2.6}$$

where $p(X_1, X_2)$ is a marginal of distribution $p(X_1, \cdots, X_n)$ and the process in Equation (2.6) is referred to as *marginalization*. Given this, marginal inference is defined as the computation of the distribution of a subset of variables given another subset.

2.4.1 Inference in Dynamic Bayesian Networks

To further clarify the concept of inference in PGMs, let us calculate two PDFs of interest typical for networks with the structure similar to that of Figure 2.4. In particular, we are interested in the probability of X_{n+1} given all the previous variables $Z_n \forall n = 1, \cdots, n$. Furthermore, we seek to calculate the probability of X_n given $Z_n \forall n = 1, \cdots, n$. In the context of filtering, the former is referred to as *prediction* while the latter is called *estimation*.

$$p(X_{n+1}|\mathbf{z}_n) \propto \int p(X_1, \cdots, X_n, X_{n+1}, Z_1, \cdots, Z_n) d\mathbf{x}_n, \tag{2.7}$$

where $\mathbf{x}_n = [X_1, \cdots, X_n]$ and $\mathbf{z}_n = [Z_1, \cdots, Z_n]$. Given Equation (2.4), the integrand of Equation (2.7) can be rewritten as

$$p(X_{n+1}|\mathbf{z}_n) \propto \int \left[p(X_0) \prod_{t=1}^{n} p(X_t|X_{t-1}) p(Z_t|X_t) \right] p(X_{n+1}|X_n) d\mathbf{x}. \tag{2.8}$$

We can rewrite the above equation as

$$p(X_{n+1}|\mathbf{z}_n) \propto \int p(Z_n|X_n) p(X_{n+1}|X_n) dX_n \int \left[p(X_0) \prod_{t=1}^{n-1} p(X_t|X_{t-1}) p(Z_t|X_t) \right] p(X_n|X_{n-1}) d\mathbf{x}_{n-1}. \tag{2.9}$$

It is straightforward to see that

$$p(X_n|\mathbf{z}_{n-1}) \propto \int \left[p(X_0) \prod_{t=1}^{n-1} p(X_t|X_{t-1}) p(Z_t|X_t) \right] p(X_n|X_{n-1}) d\mathbf{x}_{n-1}$$

is obtained by decrementing the indices in Equation (2.8). Finally,

$$p(X_{n+1}|\mathbf{z}_n) \propto \int p(X_n|\mathbf{z}_{n-1}) p(Z_n|X_n) p(X_{n+1}|X_n) dX_n. \tag{2.10}$$

Reconducting the steps in Equations (2.7)-(2.10) for $p(X_n|\mathbf{z}_n)$ results in

$$p(X_n|\mathbf{z}_n) \propto p(X_n|\mathbf{z}_{n-1}) p(Z_n|X_n). \tag{2.11}$$

It is worth mentioning that both Equations (2.10) and (2.11) are recursive relations [59]. Such a property is extremely vital for applications where X is only

indirectly observable through Z, e.g., parameter estimation in synchronization and localization. In particular, it allows for efficiently updating the previous estimate of X upon reception of the new observation Z without recomputing all previous distributions.

In the context of statistical signal processing and estimation, Equation (2.11) is referred to as *Bayesian Recursive Filtering* (BRF)[59]. The term $p(X_n|\mathbf{z}_{n-1})$ denotes the *prediction* step while $p(Z_n|X_n)$ is known as *measurement update* or *correction* step. These steps are integral to performing synchronization and joint sync&loc in Chapters 3 and 4.

2.4.2 Inference in Factor Graphs

We mentioned before that inference is defined as the calculation of a probability of interest given specific evidence. Such calculation is costly in many applications where the number of variables is high. Nevertheless, graphical representations can, depending on the type of the problem, lighten the computational burden by revealing conditional dependencies or enabling distributed computation. A famous example is BP running over FGs, where the factorization enables efficient inference [60]. In what follows, we describe the details of BP when running over singly connected and loopy FGs.

2.4.2.1 Belief Propagation

Belief propagation is an efficient inference algorithm for tree-structured graphical models [54]. It relies mainly on one of the central concepts fundamental to every efficient inference, namely *message passing*, where information from the graph is summarized in local messages and passed over. Imagine we would like to calculate the marginal $p(X_1)$ in the FG of Figure 2.6b. We can write

$$p(X_1) = \int p(X_1,X_2,X_3)dX_2dX_3 = \int f_1(X_1,X_2)f_2(X_1)f_3(X_2,X_3)dX_2dX_3. \tag{2.12}$$

Since $f_1(X_1,X_2)$ is not dependent on X_3, the above integral can be rewritten as

$$p(X_1) = \underbrace{\int f_1(X_1,X_2)\underbrace{f_2(X_2)\underbrace{\left(\int f_3(X_2,X_3)dX_3\right)}_{\lambda_{f_3\to X_2}}}_{\lambda_{X_2\to f_1}}dX_2}_{\lambda_{f_1\to X_1}}, \tag{2.13}$$

where λ denotes the messages between the nodes. What was conducted in Equation (2.13) is called *variable elimination* since we sequentially take the integral over the variables, or, alternatively, we eliminate them one by one from the distribution. The elimination of a variable can be viewed as passing a message from a node to its neighboring node. Such a mathematical procedure can be generalized to more complex networks, e.g., multi-node FGs, resulting in their corresponding inter-node messages.

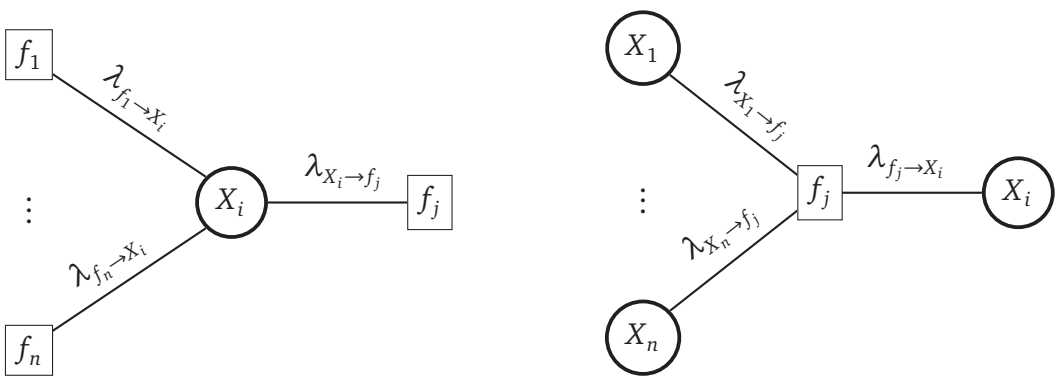
The messages in FGs are divided into two categories, variable-to-factor and factor-to-variable messages. The message from variable node i to factor node j is calculated as

$$\lambda_{X_i \to f_j} = \prod_{f_k \in \{\mathrm{ne}(X_i) \setminus f_j\}} \lambda_{f_k \to X_i} \tag{2.14}$$

where $\mathrm{ne}(X_i)$ denotes the set of neighboring factor nodes of X_i. The message from the factor j to the variable node i is calculated as

$$\lambda_{f_j \to X_i} = \int_{\mathrm{ne}(f_j) \setminus X_i} f_j(\mathrm{ne}(f_j)) \left[\prod_{X_k \in \{\mathrm{ne}(f_j) \setminus X_i\}} \lambda_{X_k \to f_j} \right], \tag{2.15}$$

where $\mathrm{ne}(f_j)$ denotes the set of neighboring variable nodes of f_j. Figures 2.7a and 2.7b depict the principles described in Equations (2.14) and (2.15), respectively.



(a) Node to factor message. (b) Factor to node message.

Figure 2.7: Principles of message passing in FGs.

2.4.2.2 Loopy Belief Propagation

As explained in the Section 2.4.2.1, BP is a technique to draw the exact inference of marginals $p(X_i)$ in singly-connected FGs. The algorithm is purely local, i.e., the message update at each node is performed without any awareness of the graph's

global structure, relying only on the local neighborhood. Given this, even if the graph is loopy (multiply-connected), the algorithm can still be deployed in the hope that it converges, albeit with a good approximation [61], [62]. Although the convergence is generally not guaranteed, when it converges, the results can be surprisingly accurate. There are a number of applications such as [21], [23], [63], [64] where loopy BP is guaranteed to converge under certain conditions. We will deal with these applications further in Chapter 3.

2.5 Summary

In this chapter, we discussed the principles of PGMs and their capabilities in modeling real-world phenomena by reviewing some aspects of graph and probability theory. In particular, we elaborated on how DBNs and FGs can be helpful when aiming to estimate a particular random variable representing an aspect of a phenomenon. Finally, we presented two inference algorithms that can be run on DBNs and FGs for estimating any distribution of interest (see Table 2.1). In the following chapters, we leverage the preliminaries presented in this chapter to perform synchronization and localization in wireless communication networks.

Table 2.1: Summary of the algorithms presented in this chapter.

Inference Alg.	Corresponding PGM	Strengths	Procedure
Bayesian Recursive Filtering (BRF)	Dynamic Bayesian Networks (DBNs)	Time-series modeling	Recursive prediction, correction, and estimation
Belief Propagation (BP)	Factor Graphs (FGs)	Factorized PDF modeling	Message passing

Chapter 3

Statistical Network Synchronization

3.1 Introduction

Network synchronization is integral to a large variety of services such as distributed beamforming [65], tracking [4], mobility prediction [66], and localization [8], [9], [26]. To pave the way for these services, considerable effort has been made to design fast, continuous, and precise synchronization algorithms across different networks, from Wireless Sensor Networks (WSNs) to wireless communication networks [13]. Generally, state-of-the-art synchronization algorithms adopt two main macroscopic approaches: a) designing a *network-wide synchronization* algorithm from scratch [21]–[24], and b) employing the existing *pairwise synchronization* protocols in a structural manner, e.g. layer-by-layer pairwise synchronization [18]–[20].

Network-wide synchronization in WSNs has been addressed in [21], [23], [24] by employing the Belief Propagation (BP) algorithm. Typically, BP runs on a Factor Graph (FG) corresponding to the network and calculates the marginals at each node by iteratively exchanging beliefs between neighboring nodes [67]. The algorithm is advantageous in the sense that it is fully distributed and estimates the clock offset and skew with high accuracy. However, the amount of time required to compute the pairwise conditional Probability Distribution Functions (PDFs) needed for FGs, and then conducting the iterative message passing, can be considered as a potential drawback rendering its practical applicability limited.

Pairwise synchronization is mostly conducted by exchanging time-stamps between the nodes using the Precision Time Protocol (PTP) [68]. To perform network synchronization in a layer-by-layer manner, the PTP is then combined with the Best Master Clock Algorithm (BMCA), whose purpose is to determine the Master Node (MN) in the network. While this combination operates sufficiently robust in tree-structured networks with medium time-sensitivity (sub-μs range),

BMCA's poor performance in networks with mesh topology on the one hand, and uncertainty in time-stamping on the other hand, render the algorithm futile in highly time-sensitive (sub-hundred *nanosecond* (ns) range) loopy networks. Despite the attempts in [18] and [45] to address the time-stamping uncertainty (or error) by virtue of Kalman filtering, this approach is not optimal in the Bayesian sense since not all information available from time-stamps is utilized. Instead, the Bayesian Recursive Filtering (BRF) utilized in [11] can be employed on a Dynmaic Bayesian Network (DBN) to capture all available information in time-stamps, thereby optimally rectifying the time-stamping error.

Although all the aforementioned techniques have made invaluable contributions, none of them alone can be expected to meet the global and local time precision aimed for by 5G for accurate localization [69]. Instead, a combination of these algorithms is more likely to deliver a superior performance owing to diverse network topologies (mesh, tree, or a combination thereof) [45]. In particular, to successfully achieve precise network synchronization, each domain can, based on its topology and capabilities, leverage the most suitable algorithm, whereby satisfying its own requirement of the relative time error while keeping the absolute time error low. This is particularly of interest in applications where ultra-high time accuracy is required in a specific synchronization domain, e.g., for positioning services.

In summary, what is covered in this chapter is as follows:

- We provide a discussion on the time-stamp exchange mechanism on the theoretical and practical level. The mechanism helps collecting the necessary measurements to estimate the nodes' clock parameters.
- We present the principles of a network-wide synchronization algorithm based on the BP running on an FG. Such an approach achieves end-to-end synchronization by passing messages among the neighboring nodes.
- We develop a pairwise synchronization algorithm based on the BRF performing on a DBN, where the focus is primarily on the fast synchronization of two neighboring nodes.
- We adopt a hybrid approach to accurately estimate the clock offset and skew, whose performance is then studied and evaluated by comparing it with a non-hybrid algorithm, i.e., BP.

In what follows, we firstly introduce the system model, i.e., the clock model as well as time-stamp exchange mechanism. Subsequently, we deal with the

estimation methods for network-wide, pairwise, and hybrid synchronization based on the principles explained in Chapter 2. Finally, we present an in-depth analysis of the algorithms proposed in this chapter with the aid of simulation results.

3.2 System Model

In this section, we firstly present the clock model for each node in the communication network. Then, we explain the components constructing the clock offset in detail. Finally, the time-stamp exchange mechanism is comprehensively described.

3.2.1 Clock Model

The clock behavior for each node i is modeled as [70]

$$c_i(t) = \gamma_i t + \theta_i, \tag{3.1}$$

where $c_i(t)$ shows the local time at each node, t represents the reference time, γ_i denotes the clock skew, and θ_i is the clock offset. We consider the parameter γ_i as random and varying over time. However, it is common to assume that it remains unchanged during one synchronization period [18], [23]. Moreover, θ_i consists of several components, all thoroughly discussed in the following subsection. In the light of above-mentioned points, time synchronization can be deemed equivalent to estimating γ_i and θ_i (or transformations thereof) for each node. Corrections are then applied such that, ideally, all clocks show the same time as the reference time t.

3.2.2 Clock Offset Decomposition

We decompose the clock offset θ_i as shown in Figure 3.1, thereby elaborating on its constituent components. The parameter t_i/t_j is the time it takes for a packet to leave the transmitter after being time-stamped (the term "time-stamp" is referred to hardware time-stamping hereafter), $(\frac{d_{ji}}{v_c})/(\frac{d_{ij}}{v_c})$ denotes the propagation delay, and r_i/r_j represents the time that a packet needs to reach the time-stamping

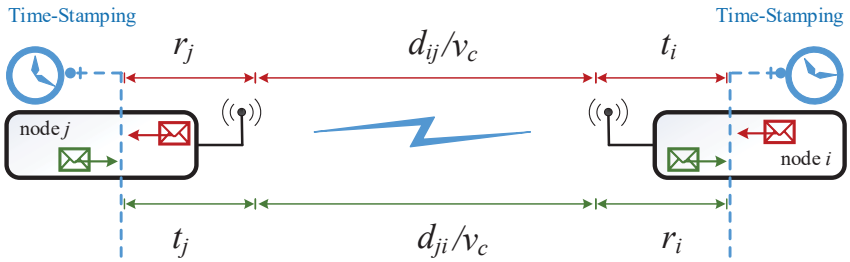


Figure 3.1: Decomposition of delay into its constituent components.

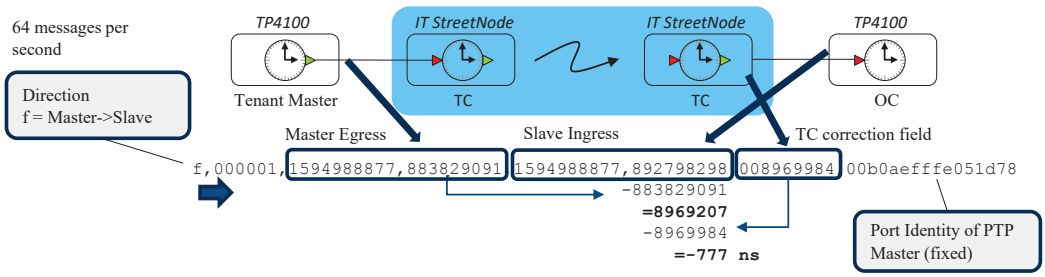


Figure 3.2: Experiment conducted to monitor the clock offset and skew evolution.

point upon arrival at the receiver. Furthermore, d_{ij}/d_{ji} and v_c represent the distance from node i/j to node j/i and the speed of light, respectively. Generally,

$$t_i + \frac{d_{ij}}{v_c} + r_j \neq t_j + \frac{d_{ji}}{v_c} + r_i,$$

meaning that the packets sent from node i to node j do not experience the same delay as the packets sent from node j to node i. In particular $T_{ij} = t_i + r_j$, and $R_{ij} = t_j + r_i$ are random variables due to multiple hardware-related random independent processes and can, therefore, be assumed i.i.d. Gaussian random variables distributed as $\mathcal{N}(T_{ij}|\mu_T, \sigma_T^2)$ and $\mathcal{N}(R_{ij}|\mu_R, \sigma_R^2)$, respectively [21], [23], [24]. Conversely, d_{ij} and d_{ji} are usually assumed to be deterministic and symmetric ($d_{ij} = d_{ji}$)[21]. To provide more insights into the statistical behavior of the clock offset, the experiment depicted in Figure 3.2 was conducted in [71, Section 6]. One Microsemi TP4100 node is configured as master and the other as slave. They are connected to two street nodes equipped with Transparent Clocks (TC) capable of time-stamping a packet upon entering and exiting the node. The clock frequency of the nodes is 125 MHz (8 ns time step). The total delay (offset + time of flight) is calculated as *the difference between the Master and slave clocks minus the TC correction time (the time a packet travels within the two street nodes)*.

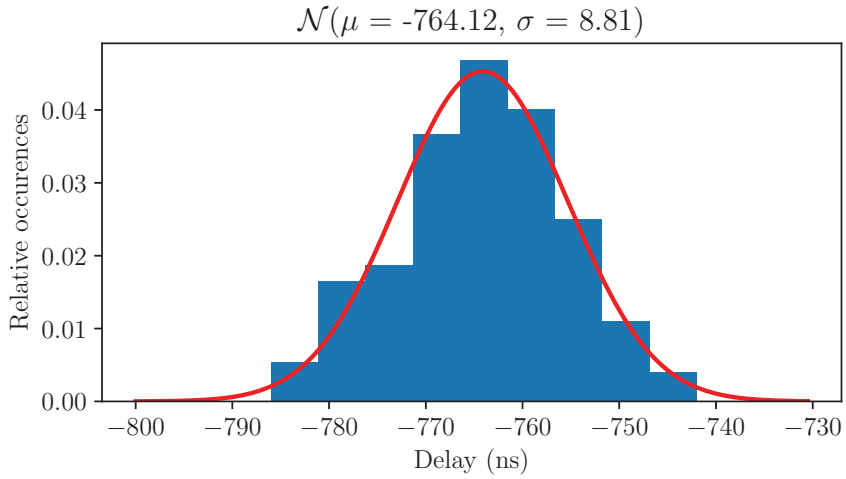


Figure 3.3: Histogram of measured offset and its Gaussian fit for 5000 packet exchange.

Figure 3.3 depicts the histogram of the clock offset and its Gaussian fit for 5000 packet exchanges [1]. In particular, the variance of the offset turns out to be around 9 ns, what is crucial to know if we are to reduce the error in the clock offset/skew estimation. Another experiment which verifies our results is the distance measurement using two-way ranging conducted in [72]. There, we also observe a Gaussian distribution of distance error that corresponds to a Gaussian distribution for the delay. In the following, we describe the time-stamp exchange mechanism.

3.2.3 Time-stamp Exchange Mechanism

We employ the asymmetric time-stamping mechanism introduced in [73], employed in [8], [23], and shown in Figure 3.4. It functions as follows: node j transmits a *sync* message wherein the local time $c_j(t_1^k)$ is incorporated. Node i receives the packet and records the local reception time $c_i(t_2^k)$. After a certain time, the process repeats again with $c_j(t_3^k)$ and $c_i(t_4^k)$. Subsequently, at local time $c_i(t_5^k)$, node i sends back a *sync* message to node j with $c_i(t_2^k)$, $c_i(t_4^k)$ and $c_i(t_5^k)$

[1] The uncertainty in time-stamping is due to the precision of the devices as well as the manner of hardware time-stamping implementation. For the nodes in this experiment, the precision of time-stamping was 8 ns, meaning that the time-stamps were always an integer multiple of 8 ns. Newly designed devices can fulfill the accuracy of below 5 ns.

incorporated. Upon reception, node j records the local time $c_j(t_6^k)$. Given that, the relation between local clocks can be written as:

$$\frac{1}{\gamma_i}(c_i(t_2^k)-\theta_i)=\frac{1}{\gamma_j}(c_j(t_1^k)-\theta_j)+d_{ij}+T_{ij}^{k,0}, \tag{3.2}$$

$$\frac{1}{\gamma_i}(c_i(t_4^k)-\theta_i)=\frac{1}{\gamma_j}(c_j(t_3^k)-\theta_j)+d_{ij}+T_{ij}^{k,1}, \tag{3.3}$$

$$\frac{1}{\gamma_i}(c_i(t_5^k)-\theta_i)=\frac{1}{\gamma_j}(c_j(t_6^k)-\theta_j)-d_{ij}-R_{ij}^k\ , \tag{3.4}$$

where $(t_1^k,\ t_3^k)/t_6^k$ and $t_5^k/(t_2^k,\ t_4^k)$ are the time points where neighboring nodes j and i send/receive the *sync* messages, respectively. Stacking the weighted sum of Equations (3.2), (3.3) and (3.4) for K rounds of time-stamp exchange gives

$$\mathbf{W}_{ji}\boldsymbol{\vartheta}_i+\mathbf{W}_{ij}\boldsymbol{\vartheta}_j=\mathbf{z}_{ij}, \tag{3.5}$$

where $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$ are $K\times 2$ matrices with the k-th row being

$$\left[\frac{1}{2}\left(c_i(t_2^k)+c_i(t_4^k)\right)+c_i(t_3^k),-2\right],$$

and

$$-\left[\frac{1}{2}\left(c_i(t_3^k)+c_i(t_4^k)\right)+c_j(t_4^k),-2\right],$$

respectively. Moreover, we introduce the vector variables $\boldsymbol{\vartheta}_i\triangleq\left[\frac{1}{\gamma_i},\frac{\theta_i}{\gamma_i}\right]^T$, and $\boldsymbol{\vartheta}_j\triangleq\left[\frac{1}{\gamma_j},\frac{\theta_j}{\gamma_j}\right]^T$ with $\frac{1}{\gamma_i}$, $\frac{\theta_i}{\gamma_i}$, $\frac{1}{\gamma_j}$, and $\frac{\theta_j}{\gamma_j}$ being Gaussian distributed [8], [24]. Finally $\mathbf{z}_{ij}\sim\mathcal{N}(\mathbf{z}|\mathbf{0},\sigma_{ij}^2\mathbf{I}_K)$, where $\sigma_{ij}^2=\frac{\sigma_{T_{ij}}^2}{2}+\sigma_{R_{ij}}^2$. In concrete terms, what Equation (3.5) implicitly states is that for given $\boldsymbol{\vartheta}_i$ and $\boldsymbol{\vartheta}_j$, the probability that we measure $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$ is equal to $\mathcal{N}(\mathbf{z}=\mathbf{W}_{ji}\boldsymbol{\vartheta}_i+\mathbf{W}_{ij}\boldsymbol{\vartheta}_j|\mathbf{0},\sigma_{ij}^2\mathbf{I}_N)$. This can be expressed as

$$p(\mathbf{W}_{ji},\mathbf{W}_{ij}|\boldsymbol{\vartheta}_i,\boldsymbol{\vartheta}_j)\sim\mathcal{N}(\mathbf{z}=\mathbf{W}_{ji}\boldsymbol{\vartheta}_i+\mathbf{W}_{ij}\boldsymbol{\vartheta}_j|\mathbf{0},\sigma_{ij}^2\mathbf{I}_N). \tag{3.6}$$

As we will see in the following sections, the structure obtained in (3.6) will help us in approximating the joint distribution of all the clock parameters by a more tractable mathematical expression.

3.3 Clock Offset and Skew Estimation

In this section, first, the principles of BP-based network-wide synchronization are described. Subsequently, we introduce the BRF-based pairwise synchronization.

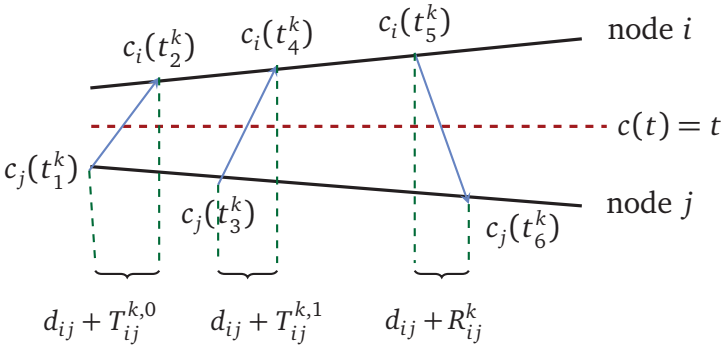


Figure 3.4: Asymmetric time-stamp exchange between node i and node j.

Lastly, we present an approach, where both techniques are employed in a hybrid manner.

3.3.1 Network-wide Offset and Skew Estimation

In network-wide synchronization, the goal is to synchronize each node with a global MN. Alternatively, the problem can be reformulated as an estimation of parameters γ_i and θ_i (or vector parameter $\boldsymbol{\vartheta}_i$), based on the observation matrices $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$. Mathematically, this is translated to the following marginal calculation:

$$p(\boldsymbol{\vartheta}_i|\{\mathbf{W}_{ji},\mathbf{W}_{ij}\}_{i=1:M,j\in\mathrm{ne}(i)}) = \int\cdots\int p(\boldsymbol{\vartheta}_1,\cdots,\boldsymbol{\vartheta}_M|\{\mathbf{W}_{ji},\mathbf{W}_{ij}\}_{i=1:M,j\in\mathrm{ne}(i)})d\boldsymbol{\vartheta}_1\cdots d\boldsymbol{\vartheta}_{i-1}d\boldsymbol{\vartheta}_{i+1}\cdots d\boldsymbol{\vartheta}_M, \quad (3.7)$$

where $\mathrm{ne}(i)$ denotes the set of neighboring nodes of node i and M is the total number of the nodes in the network. Consequently, $\boldsymbol{\vartheta}_i$ can be estimated as

$$\hat{\boldsymbol{\vartheta}}_i = \arg\max_{\boldsymbol{\vartheta}_i} p(\boldsymbol{\vartheta}_i|\{\mathbf{W}_{ji},\mathbf{W}_{ij}\}_{i=1:M,j\in\mathrm{ne}(i)}), \quad (3.8)$$

where $\hat{\boldsymbol{\vartheta}}_i$ denotes the estimated value of the clock parameters. Unfortunately, the computation cost and complexity of the marginal PDF in Equation (3.7) is extremely high. Instead, as a compromise, one can resort to approximating the integrand of Equation (3.7). This is carried out in the following with the aid of *variational methods*.

3.3.1.1 Variational Methods

The basic idea underpinning variational methods is to approximate an intractable complex distribution $p(\mathbf{x})$ by a straightforward tractable distribution $q(\mathbf{x})$. To this end, one can minimize the discrepancy measure Kullback-Leibler (KL) divergence between $p(\mathbf{x})$ and $q(\mathbf{x})$, given by [67]

$$D_{KL}(p\|q) = \int_{-\infty}^{+\infty} p(\mathbf{x}) \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) d\mathbf{x}. \tag{3.9}$$

The minimization is then achieved by deploying the *Bethe method*, which imposes the following structure on $q(\mathbf{x})$ [74]:

$$q(\mathbf{x}) \propto \prod_i q(x_i) \prod_{i,j} q(x_i, x_j), \tag{3.10}$$

where x_j and x_i are neighboring nodes. The structure in Equation (3.10) can be appropriately represented by an FG, the details of which are extensively explained in Section 2.3.

Adopting the above-mentioned approximation, the conditional probability under the integral of Equation (3.7) turns into

$$p(\boldsymbol{\vartheta}_1, \cdots, \boldsymbol{\vartheta}_M | \{\mathbf{W}_{ji}, \mathbf{W}_{ij}\}_{i=1:M, j\in ne(i)}) \propto \prod p(\boldsymbol{\vartheta}_i) \prod p(\mathbf{W}_{ij}, \mathbf{W}_{ji} | \boldsymbol{\vartheta}_i, \boldsymbol{\vartheta}_j), \tag{3.11}$$

where $p(\boldsymbol{\vartheta}_i)$ indicates the Gaussian distributed prior knowledge on $\boldsymbol{\vartheta}_i$ and $p(\mathbf{W}_{ji}, \mathbf{W}_{ij} | \boldsymbol{\vartheta}_i, \boldsymbol{\vartheta}_j)$ is the pairwise conditional probability computed from Equation (3.6). The approximation in (3.11) can be well represented by FGs. In the following section, we present the details of this representation and how it can facilitate the clock parameter estimation.

3.3.1.2 Factor Graph and Belief propagation

We construct the graphical model in Figure 3.5, where a number of APs are backhauled by a mesh network, each represented by $\boldsymbol{\vartheta}_i$. Note that, when performing network-wide synchronization, APs are also considered as variable nodes and there is a factor node between each AP and its corresponding backhauling node. This has not been depicted in the figure to avoid unnecessary complexity. The main objective is then to compute the marginal illustrated in Equation (3.7) by means of BP described in Section 2.4.2.1.

As mentioned before, BP is a technique which relies primarily on the exchange of beliefs between neighboring nodes to infer the marginals. This inference is

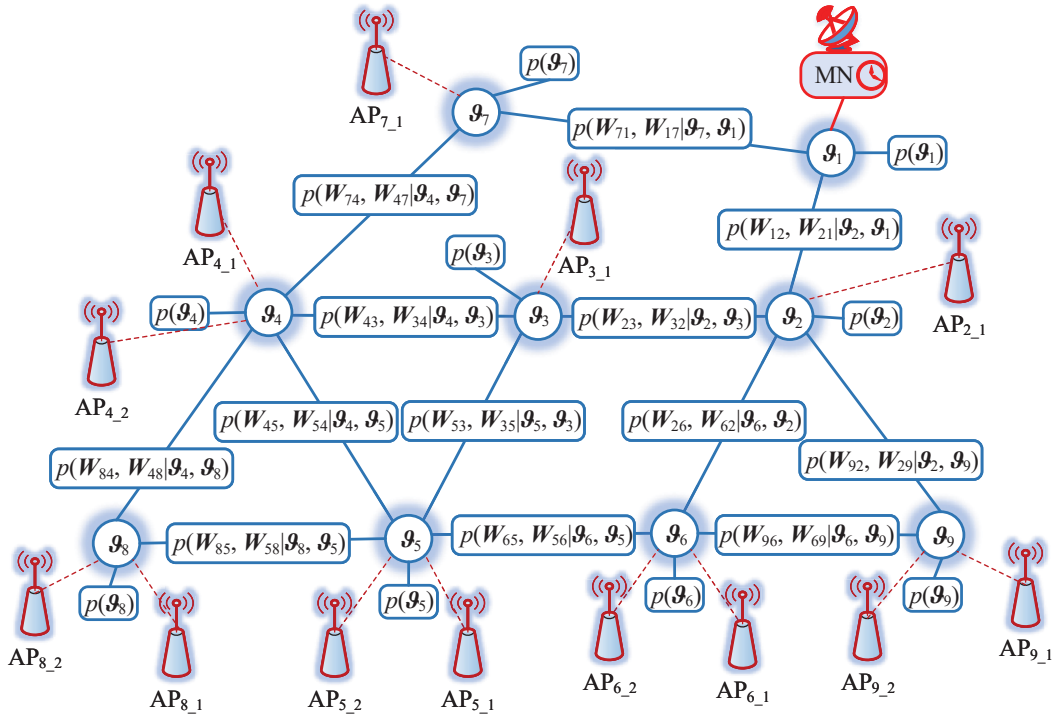


Figure 3.5: The FG corresponding to an exemplifying network. Note that the FG is drawn only for the backhaul network to avoid unnecessary complexity. To draw the FG of the whole network, one can simply consider APs as variable nodes connected to their corresponding backhaul nodes via factor nodes.

Figure 3.6: Message passing principles in BP.

proved to be exact when the graphs are singly connected and approximate if they contain loops [67]. While generally there is no guarantee that the algorithm converges in loopy graphs, [24] and [23] have indicated that, if there exists at least one MN in the network, the convergence of BP is certain. Figure 3.6 illustrates the details of the message passing in BP for the nodes $\boldsymbol{\vartheta}_i$ and $\boldsymbol{\vartheta}_j$. For the sake of simplicity, we denote the factor $p(\mathbf{W}_{ji}, \mathbf{W}_{ij}|\boldsymbol{\vartheta}_i, \boldsymbol{\vartheta}_j)$ with p_{ij}. The message from a factor vertex p_{ij} to a variable vertex $\boldsymbol{\vartheta}_i$ in iteration l is then given by [67]

$$\lambda^{(l)}_{p_{ij}\to\boldsymbol{\vartheta}_i}(\boldsymbol{\vartheta}_i) = \int p(\mathbf{W}_{ji}, \mathbf{W}_{ij}|\boldsymbol{\vartheta}_i, \boldsymbol{\vartheta}_j)\lambda^{(l)}_{\boldsymbol{\vartheta}_j\to p_{ij}}(\boldsymbol{\vartheta}_j)d\boldsymbol{\vartheta}_j, \tag{3.12}$$

where $\lambda^{(l)}_{\boldsymbol{\vartheta}_j \to p_{ij}}(\boldsymbol{\vartheta}_j)$ denotes the message from a variable node $\boldsymbol{\vartheta}_j$ to the variable vertex p_{ij} and is given by

$$\lambda^{(l)}_{\boldsymbol{\vartheta}_j \to p_{ij}}(\boldsymbol{\vartheta}_j) = p(\boldsymbol{\vartheta}_j) \prod_{k \in \{\mathcal{T}(j) \setminus i\}} \lambda^{(l-1)}_{p_{kj} \to \boldsymbol{\vartheta}_j}(\boldsymbol{\vartheta}_j). \tag{3.13}$$

Finally,

$$b^{(l)}(\boldsymbol{\vartheta}_i) \propto p(\boldsymbol{\vartheta}_i) \prod_{k \in \mathcal{T} i} \lambda^{(l)}_{p_{ik} \to \boldsymbol{\vartheta}_i}(\boldsymbol{\vartheta}_i), \tag{3.14}$$

where $b^{(l)}(\boldsymbol{\vartheta}_i)$ denotes the marginal belief of variable node $\boldsymbol{\vartheta}_i$ in the l-th iteration. It is expected that the result of the integral in Equation (3.12) is Gaussian distributed as its arguments are also Gaussian distributed. We note that, in practice, both Equation (3.12) and Equation (3.13) are locally computed at each node and only $\lambda^{(l)}_{p_{ij} \to \boldsymbol{\vartheta}_i}(\boldsymbol{\vartheta}_i)$ is transmitted from node j to node i as shown in Figure 3.7.

Let $\lambda^{(l)}_{j \to i}(\boldsymbol{\vartheta}_i) \sim \mathcal{N}(\boldsymbol{\vartheta}_i | \boldsymbol{\mu}^{(l)}_{j \to i}, \boldsymbol{\Sigma}^{(l)}_{j \to i})$ denote the message sent from j to i. Considering Equations (3.12) and (3.13), the covariance matrix $\boldsymbol{\Sigma}^{(l)}_{j \to i}$ can be calculated by [24], [46], [75]

$$\boldsymbol{\Sigma}^{(l)}_{j \to i} = \left[\mathbf{W}^T_{ji} \left(\boldsymbol{\Omega}^{(l-1)}_{j \to i} \right)^{-1} \mathbf{W}_{ji} \right]^{-1}, \tag{3.15}$$

where

$$\boldsymbol{\Omega}^{(l-1)}_{j \to i} = \sigma^2_{ij} \mathbf{I}_N + \mathbf{W}_{ij} \left[\boldsymbol{\Sigma}^{-1}_j + \sum_{k \in ne(j) \setminus i} \left(\boldsymbol{\Sigma}^{(l-1)}_{k \to j} \right)^{-1} \right]^{-1} \mathbf{W}^T_{ij}, \tag{3.16}$$

and $\boldsymbol{\Sigma}_j$ is the covariance matrix of $p(\boldsymbol{\vartheta}_j)$. Furthermore,

$$\begin{aligned} \boldsymbol{\mu}^{(l)}_{j \to i} = & \\ & - \boldsymbol{\Sigma}^{(l)}_{j \to i} \mathbf{W}^T_{ji} \boldsymbol{\Omega}^{(l-1)}_{j \to i} \mathbf{W}_{ij} \left[\boldsymbol{\Sigma}^{-1}_j + \sum_{k \in ne(j) \setminus i} \left(\boldsymbol{\Sigma}^{(l-1)}_{k \to j} \right)^{-1} \right]^{-1} \\ & \times \left[\boldsymbol{\Sigma}^{-1}_j \boldsymbol{\mu}_j + \sum_{k \in ne(j) \setminus i} \left(\boldsymbol{\Sigma}^{(l-1)}_{k \to j} \right)^{-1} \boldsymbol{\mu}^{(l-1)}_{k \to j} \right], \end{aligned} \tag{3.17}$$

where $\boldsymbol{\mu}_j$ represents the mean vector of $p(\boldsymbol{\vartheta}_j)$. It should be noted that $\boldsymbol{\Sigma}_j$ and $\boldsymbol{\mu}_j$ remain unchanged during the message updating process. The BP algorithm initializes the message from node j to node i as $\lambda^{(0)}_{j \to i}(\boldsymbol{\vartheta}_i) \sim \mathcal{N}(\boldsymbol{\vartheta}_i | \mathbf{0}, +\infty \mathbf{I}_2)$. Node j computes its outgoing message to node i according to Equations (3.15)

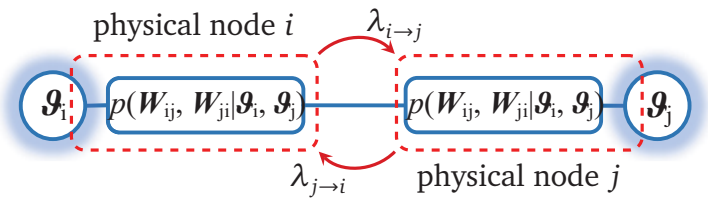


Figure 3.7: $\lambda_{i\to j} = \lambda_{p_{ij}\to\vartheta_j}$ and $\lambda_{j\to i} = \lambda_{p_{ij}\to\vartheta_i}$ illustrate the BP messages exchanged between physical nodes in practice.

and (3.17) in iteration l with its available $\boldsymbol{\Sigma}_{k\to j}^{(l-1)}$ and $\boldsymbol{\mu}_{k\to j}^{(l-1)}$ $(k \in ne(j) \setminus i)$. The belief of node i is then computed as

$$b^{(l)}(\boldsymbol{\vartheta}_i) \sim \mathcal{N}(\boldsymbol{\vartheta}_i | \boldsymbol{\nu}_i^{(l)}, \mathbf{P}_i^{(l)}), \tag{3.18}$$

where

$$\mathbf{P}_i^{(l)} = \left[\boldsymbol{\Sigma}_i^{-1} + \sum_{j\in ne(i)} \left(\boldsymbol{\Sigma}_{j\to i}^{(l-1)} \right)^{-1} \right]^{-1}, \tag{3.19}$$

and

$$\boldsymbol{\nu}_i^{(l)} = \mathbf{P}_i^{(l)} \left[\boldsymbol{\Sigma}_i^{-1} \boldsymbol{\mu}_i + \sum_{j\in ne(i)} \left(\boldsymbol{\Sigma}_{j\to i}^{(l-1)} \right)^{-1} \boldsymbol{\mu}_{j\to i}^{(l-1)} \right]. \tag{3.20}$$

Finally, the clock skew and offset estimation can be computed by

$$\hat{\gamma}_i^{(l)} = \frac{1}{\boldsymbol{\nu}_i^{(l)}(1)}, \; \hat{\theta}_i^{(l)} = \frac{\boldsymbol{\nu}_i^{(l)}(2)}{\boldsymbol{\nu}_i^{(l)}(1)}. \tag{3.21}$$

The calculation in (3.21) stems from the transformation $\boldsymbol{\vartheta}_i = \left[\frac{1}{\gamma_i}, \frac{\theta_i}{\gamma_i} \right]^T$. In what follows, we describe the procedure of pairwise synchronization.

3.3.2 Pairwise Offset and Skew Estimation

In pairwise synchronization, one node is assumed to be the MN. In particular, in Figure 3.4, instead of a global reference $c(t) = t$, we take node j as MN. We can then introduce the transformations

$$\frac{1}{\tilde{\gamma}_i} = \frac{\gamma_j}{\gamma_i}, \tag{3.22}$$

$$\tilde{\theta}_i = \theta_i - \tilde{\gamma}_i \theta_j, \tag{3.23}$$

$$\tilde{d}_{ij} + \tilde{T}^k_{ij} = \gamma_j (d_{ij} + T^k_{ij}), \tag{3.24}$$

$$\tilde{d}_{ij} - \tilde{R}^k_{ij} = \gamma_j (d_{ij} - R^k_{ij}). \tag{3.25}$$

For the sake of simplicity, as done in [4], we assume $\tilde{d}_{ij} = d_{ij}$, $\tilde{R}^k_{ij} = R^k_{ij}$, and $\tilde{T}^k_{ij} = T^k_{ij}$. This is valid owing to $\gamma_j \approx 1$ and the value of $d_{ij} + T^k_{ij}$ and $d_{ij} - R^k_{ij}$ being low. Finally, Equations (3.2), (3.3) and (3.4) turn into

$$\frac{1}{\tilde{\gamma}_i}(c_i(t^k_2) - \tilde{\theta}_i) = c_j(t^k_1) + d_{ij} + T^{k,0}_{ij}, \tag{3.26}$$

$$\frac{1}{\tilde{\gamma}_i}(c_i(t^k_4) - \tilde{\theta}_i) = c_j(t^k_3) + d_{ij} + T^{k,1}_{ij}, \tag{3.27}$$

$$\frac{1}{\tilde{\gamma}_i}(c_i(t^k_5) - \tilde{\theta}_i) = c_j(t^k_6) - d_{ij} - R^k_{ij}. \tag{3.28}$$

By the end of the k-th round of time-stamp exchange, each node is expected to have collected the time-stamps $\mathbf{c}^{1:k}_{ij} = \left[\mathbf{c}^1_{ij}, \cdots, \mathbf{c}^k_{ij}\right]^T$, where

$$\mathbf{c}^k_{ij} = \left[c_j(t^k_1), c_i(t^k_2), c_j(t^k_3), c_i(t^k_4), c_i(t^k_5), c_j(t^k_6)\right].$$

Let $\tilde{\boldsymbol{\vartheta}}^k_i$ be the state of the vector variable $\tilde{\boldsymbol{\vartheta}}_i \triangleq \left[\frac{1}{\tilde{\gamma}_i}, \frac{\tilde{\theta}_i}{\tilde{\gamma}_i}\right]^T$ after the k-th round of time-stamp exchange (visualized in Figure 3.8). Similar to Equation (3.7) the PDF corresponding to the k-th state can be written as

$$p(\tilde{\boldsymbol{\vartheta}}^k_i | \mathbf{c}^{1:k}_{ij}) = \int p(\tilde{\boldsymbol{\vartheta}}^0_i, \cdots, \tilde{\boldsymbol{\vartheta}}^k_i | \mathbf{c}^{1:k}_{ij})\, d\tilde{\boldsymbol{\vartheta}}^0_i \cdots d\tilde{\boldsymbol{\vartheta}}^{k-1}_i. \tag{3.29}$$

Following the steps explained in Section 2.4.1, Equation (3.29) can be simplified to

$$p(\tilde{\boldsymbol{\vartheta}}^k_i | \mathbf{c}^{1:k}_{ij}) \propto p(\tilde{\boldsymbol{\vartheta}}^k_i | \mathbf{c}^{1:k-1}_{ij}) p(\mathbf{c}^k_{ij} | \tilde{\boldsymbol{\vartheta}}^k_i) \sim \mathcal{N}(\tilde{\boldsymbol{\vartheta}}^k_i | \boldsymbol{\mu}^k_i, \boldsymbol{\Sigma}^k_i). \tag{3.30}$$

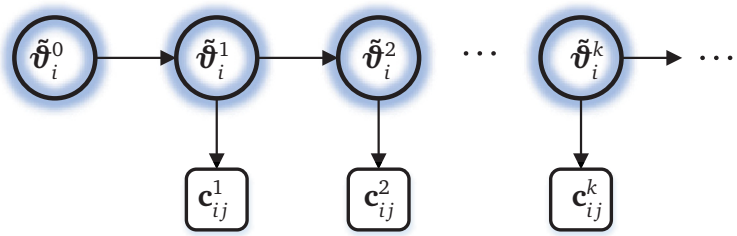


Figure 3.8: Bayesian representation of offset and skew estimation.

The term $p(\tilde{\boldsymbol{\vartheta}}_i^k|\mathbf{c}_{ij}^{1:k-1})$ is known as *prediction* step while the term $p(\mathbf{c}_{ij}^k|\tilde{\boldsymbol{\vartheta}}_i^k)$ is referred to as *measurement update* or *correction* step [59]. Considering the clock properties discussed in Section 3.2.1, it is typical in wireless networks to assume that $\tilde{\boldsymbol{\vartheta}}_i^k$ is Gaussian distributed [4], [8], [23]. Given this assumption, in the sequel, we show that the relation between the states is linear, implying that the marginal in Equation (3.30) is also Gaussian distributed. In particular, we first describe the details of *prediction* step, where all the parameters are denoted by $(\cdot)_-$. Next, we obtain the likelihood of the measurements whose parameters are represented by $(\cdot)_+$. Lastly, we compute the parameters of the posterior distribution in (3.30).

3.3.2.1 Prediction

Assuming a constant skew in one synchronization period ($= K$ rounds of time-stamp exchange), a reasonable prediction for $\tilde{\boldsymbol{\vartheta}}_i^k$ is given by [18]

$$\tilde{\boldsymbol{\vartheta}}_i^k = \mathbf{A}\tilde{\boldsymbol{\vartheta}}_i^{k-1} + \mathbf{n}_i^{k-1}, \tag{3.31}$$

where $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ c_j(t_1^k) - c_j(t_1^{k-1}) & 1 \end{bmatrix}$, and $\mathbf{n}_i^{k-1}$ denotes the Gaussian noise vector. Given Equation (3.31), the prediction term can be written as

$$p(\tilde{\boldsymbol{\vartheta}}_i^k|\mathbf{c}_{ij}^{1:k-1}) \sim \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|(\boldsymbol{\mu}_i^k)_-, (\boldsymbol{\Sigma}_i^k)_-), \tag{3.32}$$

where $(\boldsymbol{\mu}_i^k)_- = \mathbf{A}\boldsymbol{\mu}_i^{k-1}$ and $(\boldsymbol{\Sigma}_i^k)_- = \mathbf{A}\boldsymbol{\Sigma}_i^{k-1}\mathbf{A}^T + \mathbf{Q}_n$ with $\mathbf{Q}_n$ representing the noise covariance matrix.

3.3.2.2 Correction

To obtain the correction term in Equation (3.30) we conduct the following mathematical manipulations. Subtracting Equation (3.26) from Equation (3.27) leads to

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_4^k)-c_i(t_2^k)) = c_j(t_3^k)-c_j(t_1^k)+T_{ij}^{k,1}-T_{ij}^{k,0}, \tag{3.33}$$

while weighted sum of Equations (3.26), (3.27) and (3.28) gives

$$\frac{1}{\tilde{\gamma}_i}\left(\frac{c_i(t_2^k)+c_i(t_4^k)}{2}+c_i(t_5^k)-2\tilde{\theta}_i\right) = \frac{c_j(t_1^k)+c_j(t_3^k)}{2}+c_j(t_6^k)+\frac{T_{ij}^{k,0}+T_{ij}^{k,1}}{2}-R_{ij}^k, \tag{3.34}$$

where, given the assumptions in Section 3.2.2, $\frac{T_{ij}^{k,0}+T_{ij}^{k,1}}{2}-R_{ij}^k$ and $T_{ij}^k-T_{ij}^{k-1}$ are zero mean and have the variances $\frac{\sigma_{T_{ij}}^2}{2}+\sigma_{R_{ij}}^2$ and $2\sigma_{T_{ij}}^2$, respectively. This is straightforward to observe since they are a linear subtraction of independent random processes. Alternatively, we can write Equations (3.33) and (3.34) in matrix form as

$$\mathbf{B}_{ij}\tilde{\boldsymbol{\vartheta}}_i^k = \mathbf{r}_{ij}+\mathbf{z}_{ij}, \tag{3.35}$$

where $\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{z}|\mathbf{0},\mathbf{R}_{ij})$ with

$$\mathbf{R}_{ij} = \mathrm{diag}\left([2\sigma_{T_{ij}}^2, \frac{\sigma_{T_{ij}}^2}{2}+\sigma_{R_{ij}}^2]\right), \qquad \mathbf{B}_{ij} = \begin{bmatrix} c_i(t_4^k)-c_i(t_2^k) & 0 \\ \frac{c_i(t_2^k)+c_i(t_4^k)}{2}+c_i(t_5^k) & -2 \end{bmatrix},$$

and $\mathbf{r}_{ij} = \left[c_j(t_3^k)-c_j(t_1^k), \frac{c_j(t_1^k)+c_j(t_3^k)}{2}+c_j(t_6^k)\right]^T$. Consequently,

$$p(\mathbf{c}_{ij}^k|\tilde{\boldsymbol{\vartheta}}_i^k) \sim \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|(\boldsymbol{\mu}_i^k)_+,(\boldsymbol{\Sigma}_i^k)_+), \tag{3.36}$$

where $(\boldsymbol{\mu}_i^k)_+ = \mathbf{B}_{ij}^{-1}\mathbf{r}_{ij}$ and $(\boldsymbol{\Sigma}_i^k)_+ = \mathbf{B}_{ij}^{-1}\mathbf{R}_{ij}\mathbf{B}_{ij}^{-T}$.

3.3.2.3 Estimation

Considering Equations (3.32) and (3.36), the estimated distribution in Equation (3.30) is given by

$$p(\tilde{\boldsymbol{\vartheta}}_i^k|\mathbf{c}_{ij}^{1:k}) \sim \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|\boldsymbol{\mu}_i^k,\boldsymbol{\Sigma}_i^k), \tag{3.37}$$

Algorithm 1 Pairwise synchronization based on BRF

1: Initialize $p(\tilde{\boldsymbol{\vartheta}}_i^0)$ to be non-informative
2: **for** $k = 1, 2, \cdots, K$ **do**
3: Calculate the mean vector and covariance matrix of the *prediction* PDF using Equation (3.32)
4: Construct $\mathbf{B}_{ij}$, $\mathbf{R}_{ij}$, and $\mathbf{r}_{ij}$ using the measurements and obtain the mean vector and covariance matrix of *update* PDF using Equation (3.36)
5: Compute the mean vector and covariance matrix of the PDF of $\tilde{\boldsymbol{\vartheta}}_i^k$ using Equation (3.37)
6: Compute the final estimation of offset and skew using Equation (3.40)
7: **end for**

where

$$\boldsymbol{\mu}_i^k = \left[(\boldsymbol{\Sigma}_i^k)_- + (\boldsymbol{\Sigma}_i^k)_+\right]^{-1} \left((\boldsymbol{\Sigma}_i^k)_-(\boldsymbol{\mu}_i^k)_+ + (\boldsymbol{\Sigma}_i^k)_+(\boldsymbol{\mu}_i^k)_-\right), \tag{3.38}$$

$$\boldsymbol{\Sigma}_i^k = \left[(\boldsymbol{\Sigma}_i^k)_-^{-1} + (\boldsymbol{\Sigma}_i^k)_+^{-1}\right]^{-1}. \tag{3.39}$$

The parameters in Equations (3.32), (3.36) and (3.37) are calculated recursively and, in each iteration k, an estimation of the clock skew and offset can be obtained by

$$\tilde{\gamma}_i^k = \frac{1}{\boldsymbol{\mu}_i^k(1)} \text{ and } \tilde{\theta}_i^k = \frac{\boldsymbol{\mu}_i^k(2)}{\boldsymbol{\mu}_i^k(1)}. \tag{3.40}$$

The calculations conducted in (3.40) are the results of the transformation $\tilde{\boldsymbol{\vartheta}}_i = \left[\frac{1}{\tilde{\gamma}_i}, \frac{\tilde{\theta}_i}{\tilde{\gamma}_i}\right]^T$ introduced in Section 3.3.2.

Algorithm 1 summarizes the above-mentioned recursive process. In particular, after initializing the PDFs in step 1, the recursive process begins with computing the prediction PDF using Equation (3.32) in step 3. Next, in step 4, the matrices and vectors corresponding to the measurements are constructed, i.e., $\mathbf{B}_{ij}$, $\mathbf{R}_{ij}$, and $\mathbf{r}_{ij}$. Finally, in step 5 the mean vector and covariance matrix of the clock parameters can be obtained by Equation (3.37). In each iteration, one can obtain the clock offset and skew using Equation (3.40) (step 6).

3.3.3 Hybrid Synchronization

Given Sections 3.3.1 and 3.3.2, to ensure a low end-to-end synchronization error at the global level, BP can be performed over the backhaul network. At the same time, we can employ the BRF algorithm to perform synchronization between the backhaul nodes and the APs at the edge of the network, where fast and frequent synchronization is required to keep the relative time error small. This is,

Algorithm 2 Hybrid synchronization algorithm

1: Determine the suitable algorithm for each cluster (BP-nodes and BRF-nodes)
2: Commence time-stamp exchange and the BRF-based synchronization (algorithm 1) at BRF-nodes
3: Start the time-stamp exchange between adjacent BP-nodes and, subsequently, construct $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$ for each pair
4: **for** $l = 1, 2, \cdots, L$ **do**
5: Calculate the messages using Equations (3.15) and (3.17) for each BP-node and send them to its neighboring nodes.
6: Update the beliefs at each BP-node by means of Equation (3.18)
7: **if** $\hat{\boldsymbol{\vartheta}}_i^{(l)} - \hat{\boldsymbol{\vartheta}}_i^{(l-1)} \leq \epsilon \ \forall i$ **then**
8: Compute the offset and skew estimation using Equation (3.21)
9: Go to step 3
10: **end if**
11: **end for**

in particular, crucial to a number of applications such as localization as will be discussed in Chapter 4.

The steps of the hybrid synchronization are described in algorithm 2. Firstly, step 1 determines the network sections suitable for BP and BRF (they are labeled as BP-nodes and BRF-nodes, respectively). Then, step 2 initiates the time-stamp exchange mechanism (Figure 3.4) and, correspondingly, the BRF algorithm at BRF-nodes. Step 3, triggers the time-stamp exchange among the BP-nodes, thereby collecting the required time-stamps to construct the matrices $\mathbf{W}_{ji}$ and $\mathbf{W}_{ij}$. Step 4 is where the BP iterations commence and continue until it converges, or the maximum number of iterations L is achieved. In step 5, the outgoing messages are computed by each BP-node using Equations (3.15) and (3.17). They are then sent to their corresponding nodes. Step 6 updates each node's belief. Lastly, in steps 7-10, we check for the convergence by comparing the difference between clock offset and skew estimation in iterations (l) and $(l-1)$ with a predefined small value ϵ. If the algorithm has converged, the clock offset and skew estimation are calculated by means of Equation (3.18) and Equation (3.21), respectively. Note that step 2 and steps 3-11 can run simultaneously.

3.3.3.1 Convergence analysis

Convergence of the hybrid synchronization algorithm depends on the behavior of BRF, and BP. In particular, at the edge of the network where we aim to locally synchronize the APs using BRF, the convergence has no meaning. Nevertheless, as a measure to evaluate the estimator's performance, given the set of linear equations presented in Section 3.3.2, we can refer to BRF with Gaussian parameters

as minimum variance unbiased estimator [76]. Thus, the convergence of algorithm 2 depends solely on the convergence of BP, which is of crucial importance for global network synchronization.

While BP converges to the exact marginal on loop-free FGs, its convergence on loopy FGs is highly conditional. In the context of clock synchronization, a detailed convergence analysis of loopy BP has been conducted in [21], [23], [24], [77]. For the set of message passing formulas presented in this paper, we can leverage on [24, Lemma 1] and [24, Lemma 2] to prove that the mean vector $\boldsymbol{\nu}_i^{(l)}$ in Equation (3.20) and the covariance matrix $\mathbf{P}_i^{(l)}$ in Equation (3.19) of the belief $b^{(l)}(\boldsymbol{\vartheta}_i)$ in Equation (3.18) converge to a constant vector/matrix regardless of the network topology [24, Theorem 1], [24, Theorem 2]. The crucial point of this proof is that, regardless of the network topology, the belief parameters (mean vector and covariance matrix) converge as long as there is an informative prior, i.e., there exist at least one MN in the network.

3.4 Simulation Results and Comparison with State-of-the-Art

In this section, we evaluate the performance of the hybrid synchronization algorithm proposed in this work. A detailed analysis of its impact on the achievable performance of the joint synchronization and localization (sync&loc) algorithm at the edge of the network is left to the next chapter.

Figure 3.5 exemplifies a wireless network where the algorithm proposed in this work can be applied. It comprises a number of APs, all backhauled by a

Table 3.1: Simulation parameters.

Parameters	Values
Number of independent simulations	10000
Master node	$\boldsymbol{\vartheta}_1$
Initial random skew interval	$\mathcal{U}(1-10^{-4}, 1+10^{-4})$
Initial random delays interval	$\mathcal{U}(-10^3, 10^3)$ ns
Number of time-stamp exchange (K)	10
Standard deviation of T_{ij}^k and R_{ij}^k	9 ns
Pairwise random propagation delay	$[200, 300]$ ns
Initial PDF of the offset/skew for each node	$\mathcal{N}(0, +\infty)/\mathcal{N}(1, 10^{-4})$
Initial PDF of the offset/skew of MN	$\mathcal{N}(0, 0)/\mathcal{N}(1, 0)$
Process noise covariance matrix ($\mathbf{Q}_n$)	$\mathrm{diag}(10^{-5}, 100)$

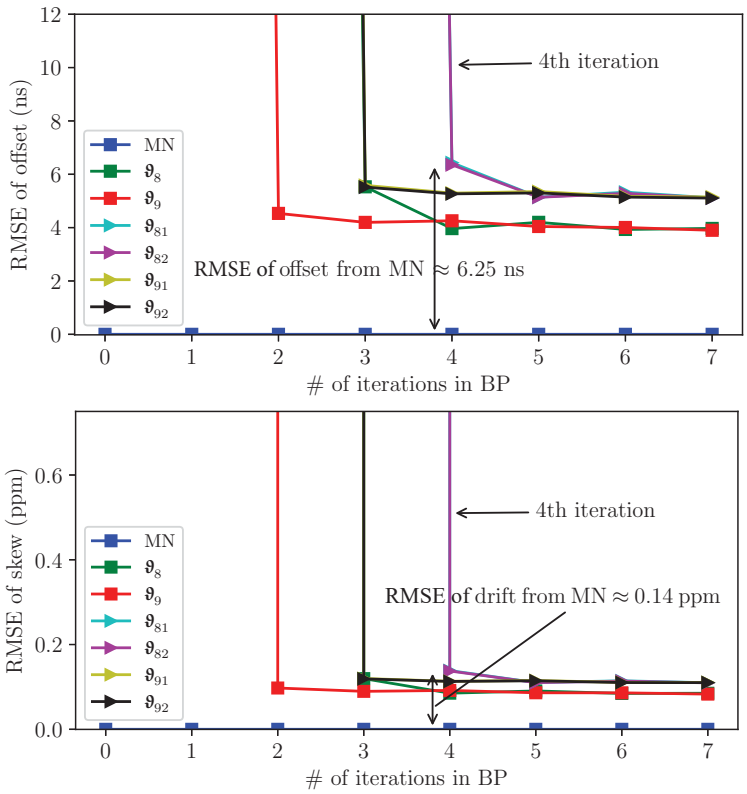


Figure 3.9: BP applied to the whole network.

wireless mesh network and delivering services to MUs. The following scenarios are simulated: a) synchronizing the whole network using only BP (the APs in Figure 3.5 are assumed to be variable nodes connected to the mesh network via factor nodes), b) performing hybrid synchronization as described in algorithm 2, where we synchronize the mesh backhaul network by means of BP and the APs at the edge of the network using BRF, and c) carrying out synchronization across the rounds of time-stamp exchange K. Scenario (a) is considered as the baseline for comparison with the hybrid approach. Furthermore, we compute the Root Mean Square Error (RMSE) of clock offset and skew estimation as a measure to evaluate the performance. For the sake of simplicity and without loss of generality, in (a), (b), and (c), we consider only the nodes $\boldsymbol{\vartheta}_8$ and $\boldsymbol{\vartheta}_9$ and their corresponding APs.

We initialize all the clock offsets from the uniform distribution $\mathscr{U}(-10^3, 10^3)$ ns. The initial skews of all clocks are drawn from the uniform distribution $\mathscr{U}(1-10^{-4}, 1+10^{-4})$, which corresponds to skew values between 0 and 100 *part-per-million* (ppm). The covariance of the clock process noise $\mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i)$ is set to $\mathrm{diag}(10^{-5}, 100)$ to account for the residual errors from the previous iterations as well as the external noises on the clock skew and offset. Further simulation parameters can be found in Table 3.1.

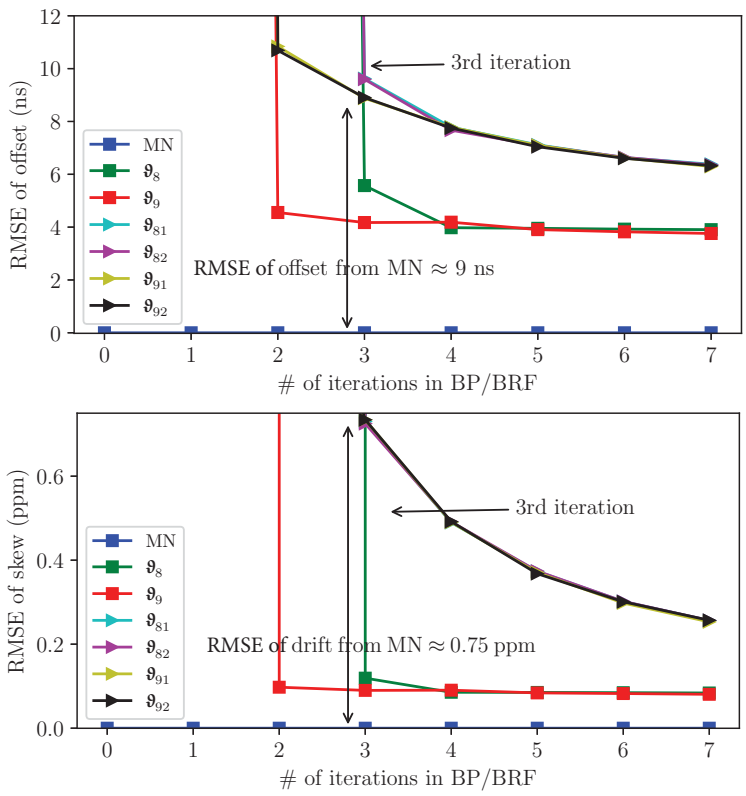


Figure 3.10: BP and BRF applied to the network in a hybrid manner.

Figure 3.9 shows the RMSEs of the clock offset and skew estimation versus the number of message passing iterations for scenario (a). The RMSEs of offset and skew are indicated in ns and ppm, respectively. As can be observed, BP converges after four iterations and achieves an offset and skew RMSE below 6 ns and 0.2 ppm, respectively. As shown in [21], [67], when there exist at least one MN in the network, the convergence is guaranteed. However, the value to which BP converges in loopy networks is deemed to be approximate. Note that, although this simulation setup reveals the potential performance of BP, the nodes, and particularly the APs, must wait at least four message passing iterations in addition to K rounds of time-stamp exchange (required for obtaining the conditional probabilities) to be fully synchronized. This is particularly unfavorable in certain synchronization-based services such as localization, where continuous time alignment is essential for accurate estimation of the MUs' positions. Therefore, it is necessary that the APs synchronize themselves to the backhaul network more frequently to be able to deliver those services at an increased performance as required in 5G networks.

Figure 3.10 depicts the RMSEs of the clock offset and skew estimation versus the number of message passing and BRF iterations for scenario (b). We can

observe a slight deterioration in performance (RMSE increases by $2-3$ ns for the offset and $0.5-0.6$ ppm for the skew) compared to scenario (a). In fact, this is the cost of economizing on the number of BP iterations as well as rounds of time-stamp exchange. To clarify, BP commences only when the nodes have already conducted K rounds of time-stamp exchange (to construct $\mathbf{W}_{ij}$ and $\mathbf{W}_{ji}$). Even then, it takes four iterations, or n if there are n nodes between an AP and the MN, to estimate the clock parameters and correspondingly perform synchronization. Conversely, BRF is faster, as it is directly applied after each round of time-stamp exchange and runs independently (it does not need any information from the other network sections as BP does). Therefore, it can conduct more iterations, thereby continuously fulfilling the requirement of very low relative time error on a local level. Given the above-mentioned properties of BP and BRF, the hybrid approach sacrifices a fraction of global accuracy to rapidly achieve synchronization at a local level, which is crucial to a number of applications such as Mobile User (MU) localization.

Figure 3.11 presents the RMSEs of the clock offset and skew estimation versus the rounds of time-stamp exchange K. As can be observed, the RMSEs of both offset and skew estimation decrease as K grows, indicating that the higher number of time-stamp exchanges leads to a more accurate estimation. The gradient is, however, slightly smaller for the APs owing to the fact that their RMSEs comprise two components, i.e., the synchronization error of the backhaul mesh network and the error arising when synchronizing APs with their corresponding backhaul nodes. Although the former decreases as K grows, the latter remains constant, resulting in a slower decline of RMSEs of clock offset and skew estimation at the APs.

The network in Figure 3.9 is only a random example picked to lucidly convey the fundamental concepts of hybrid synchronization introduced in this work. However, the intuitions obtained from the above simulations are still valid even if we replace the network by any other network with arbitrary size. Nevertheless, while the size of the network, in particular the backhaul network, does not play a role when locally synchronizing adjacent APs, it can prolong the time needed for reaching convergence for BP depending on the number of nodes between node i and the MN.

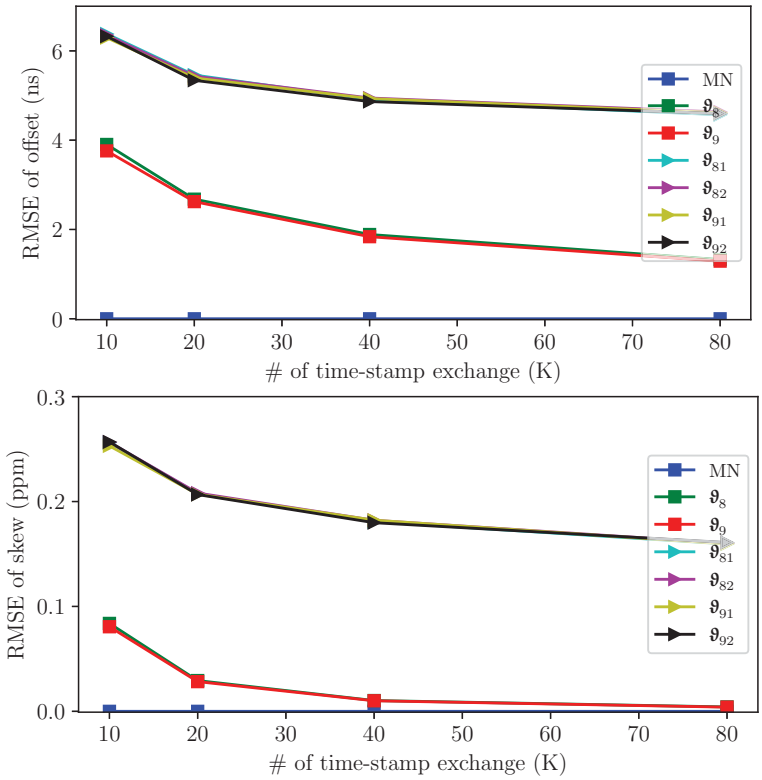


Figure 3.11: Impact of number of time-stamp exchanges K.

3.5 Summary

In this chapter, we presented three synchronization methods (summarized in Table 3.2) based on the PGMs and inference algorithms described in Chapter 2. The first method based on BP running on an FG, representing the statistical relation between the network nodes. The second relies on pairwise statistical relations and is only capable of synchronizing pairs of nodes. The former can achieve a high precision network-wide, while the latter is capable of performing fast pairwise synchronization at the cost of a slight deterioration in precision. We then combined these two methods to construct the hybrid synchronization algorithm. Hybrid synchronization can maintain a low end-to-end time error while providing fast synchronization at the local level. This approach provides a high-accuracy inter-AP synchronization at the edge of the network, thereby paving the way for a precise joint mobile user synchronization and localization, the details of which will be discussed in the upcoming chapter.

Table 3.2: Summary of the synchronization algorithms presented in this chapter.

Sync. Algorithm	Corresponding PGM	Strengths	Weaknesses
Pairwise BRF-based	Dynamic Bayesian Networks (DBNs)	Fast, mitigates the time-stamp uncertainty	Synchronizes only two nodes
Network-wide BP-based	Factor Graphs (FGs)	High accuracy on both local and global levels	Slow due to the message-passing
Hybrid BP-BRF	DBNs and FGs	High accuracy on the global and local levels, fast on the local level	Slow on the global level

Chapter 4

Joint Statistical Synchronization & Localization

4.1 Introduction

State-of-the-art Mobile User (MU) localization techniques rely primarily on the cooperation among the Access Points (APs), requiring them to be precisely synchronized. In addition, for many of the existing techniques to function, the clock parameters of the MUs need to be known (or to be continuously corrected). Therefore, it appears that the aforementioned problems, namely inter-AP synchronization, MU's clock parameter estimation, and MU localization, are closely intertwined and can be tackled jointly.

In Chapter 3, we have addressed the end-to-end synchronization in 5G networks. In particular, we employed Belief Propagation (BP) and Bayesian Recursive Filtering (BRF) not only to achieve high-precision end-to-end synchronization, but also to keep the inter-AP relative clock offset and skew low. In other words, the algorithms therein pave the way for the joint synchronization and localization (sync&loc) of MUs by accurately synchronizing the neighboring APs.

The joint MU synchronization and localization problem has been extensively addressed in the literature. The authors in [64] rely on symmetric AP-MU time-stamp exchange and the BP to jointly estimate MUs' location and clock offset. Furthermore, the authors of [8], [14] adopt a similar approach by means of the asymmetric time-stamp exchange mechanism proposed in [73]. While time-stamp exchange is expected to be supported in 5G networks [7], the high number of message-passings required by BP renders the approach limited in practice. Additionally, [7], [8], [14], [64], [73] provide the estimation of the clock and position parameters at MUs, whereas for the location-based services to be delivered, these parameters need to be computed on the network side. In [6], [9],

the authors leverage Extended Kalman Filtering (EKF) to obtain the estimation of clock parameters and position in ultra-dense networks. In particular, they assume accurate inter-AP synchronization and perform MU joint sync&loc in the presence of uncertainty about Time of Arrival (ToA) and Angle of Arrival (AoA) parameters. The level of uncertainty is then determined based on the derived Cramer Rao Bound (CRB). While EKF can partially mitigate the destructive impact of nonlinearities in the measurements, in addition to the covariance matrix underestimation, it is likely to diverge if a reliable estimate of the initial state is not available [78]. A promising approach, on the one hand, to avoid such shortcomings of EKF and, on the other hand, to boost the accuracy of position estimation, is estimating the (prediction, measurement likelihood, and posterior) distributions by means of Particle Gaussian Mixture Filters (PGMFs) introduced in [79]. Specifically, in this approach, instead of a single Gaussian function, each distribution is approximated with a sum of weighted Gaussian functions, or alternatively, mixtures [80]. Nevertheless, the problem that immediately arises when using PGMFs is dimensionality, rendering the approach computationally expensive for multi-variable estimation. To overcome this drawback, we resort to a hybrid parametric and particle-based approach where we capitalize on the linear relations in the measurements to reduce the dimensionality. In comparison to the standard Particle Filer (PF), such an approach has, as a result of Rao-Blackwell's lemma discussed in [81], a strictly lower estimation variance and leads to more accurate estimates given the same number of particles [80]. Specifically, PGMF's performance stands out when the uncertainty increases.

However, even the PGMF-based localization techniques can suffer from divergence under certain conditions, e.g., improper tuning of the filter's hyperparameters and faulty measurements, resulting mostly from Non-Line-of-Sight (NLoS) links [82]. The former must be addressed when designing the filter, while the latter can be dealt with using NLoS mitigation methods such as those proposed in [83]–[85]. The technique in [83] relies on the multipath components of NLoS links to enhance the positioning accuracy. However, such a method functions well only in the presence of strong multipath components and prior statistics on NLoS-induced errors. The latter are also estimated and utilized along with trajectory tracking in [84] to perform indoor positioning. The authors in [85], however, take another approach and model the measurement noise by a two-mode mixture distribution and approximate the maximum likelihood estimator using expectation maximization. Such approaches add an extra computation overhead that may not be necessary in dense networks where the LoS probability

is around 0.8 and increases with the AP density [86]. Therefore, to mitigate the estimation inaccuracy stemming from the faulty measurements, we deploy NLoS identification techniques to identify the NLoS links and discard them. Such an approach boosts the accuracy and features less complexity compared to the methods proposed in [83]–[85].

There is a wide spectrum of NLoS identification approaches adopted in the literature, e.g., hypothesis testing as in [87], the statistical approach taken in [88], and Machine Learning (ML)-based methods such as that of [89]. However, recently ML algorithms, particularly Deep Neural Network (DNN)-based approaches such as AmpN [36], have drawn substantial attention in classification problems. In particular, DNNs exhibit a remarkable performance due to their ability, on one hand, in implementing almost any classifier function, and, on the other hand, in extracting task-related features from the input data [90], [91]. Other approaches such as Support Vector Machine (SVM), or Bayesian sequential testing require human intervention that may be, given the limited intuition, flawed, and erroneous. Furthermore, DNN units are also expected to be part of the communication devices as they are the cornerstone of many solutions for different communication problems such as slice management and anomaly detection [92]. Therefore, a DNN-based NLoS identifier appears to be a reasonable choice. The input to the DNN can be signals containing class-relevant features such as received signal strength or Channel Impulse Response (CIR). The CIR turns out to be more informative about the communication environment and link condition. Therefore, for the sake of prediction accuracy, we rely on AP-MU CIRs in this work.

In addition to NLoS-identification, the CIR can also be fed into one of the state-of-the-art AoA estimation algorithms to obtain the signal's direction of arrival. AoA estimation has been extensively investigated in the literature. Algorithms such as Multiple Signal Classification (MUSIC) [93], reduced-dimension MUSIC [94], and Estimation of Signal Parameters via Rational Invariance Techniques (ESPRIT) [95] can accurately estimate the AoA. A detailed comparison between them has been conducted in [96] concluding that the difference is negligible, albeit MUSIC slightly outperforms the others and, therefore, it is employed for the purpose of this work.

In this chapter, as a first step, we propose a Linearized BRF (L-BRF)-based approach for the joint sync&loc. It particularly relies on Taylor expansions of the non-linear terms in the measurements (i.e., time-stamp exchange and AoA) to obtain an estimate of the clock and position parameters. Next, to overcome the

drawbacks of L-BRF, we propose a DNN-assisted PF-based (DePF) joint sync&loc algorithm which uses the CIR to estimate the AoA and to determine the link condition, i.e., LoS or NLoS, thereby excluding the faulty measurements to enable a more precise parameter estimation. DePF then estimates the joint probability distribution of MUs' clock and position parameters using the PGMF. The dimension of the PGMF is then reduced by revealing and exploiting the existing linearities in the measurements, thereby tackling the dimensionality problem. To the best of our knowledge, this is the first work employing PGMF in a hybrid particle-based and parametric manner to perform joint sync&loc.

The contribution of this chapter is summarized as follows:

- We present and discuss the principles of asymmetric time-stamp exchange and AoA estimation. The former assists in the estimation of the clock skew, offset, and the AP-MU distance, while the latter aids in the position estimation by providing the direction of an MU relative to the position and orientation of its serving APs.

- We develop a DNN for NLoS identification based on AP-MU CIRs. The outcome of such a DNN helps to identify erroneous measurements, i.e., time-stamps and AoAs, and discard them, thereby preventing large errors in the estimation.

- We propose a joint sync&loc algorithm based on the linearized BRF, which estimates the clock parameters and the position of an MU by drawing on the Taylor expansion of the measurement equations. Such an approach features low complexity, albeit it needs prior knowledge on the initial position of MUs and is prone to divergence.

- We propose a DNN-assisted particle filter-based joint sync&loc algorithm that estimates the clock parameters and the position of an MU in a hybrid parametric and particle-based manner. Such an approach not only boosts the estimation accuracy but also overcomes the dimensionality problem that arises in particle Gaussian mixture filters due to the high number of parameters.

- We analyze the performance of the proposed techniques with the aid of detailed simulations in a challenging real-world scenario. In particular, the MUs' movement profile comprises acceleration, deceleration, and constant speed. Furthermore, the APs are distributed to provide signal coverage for the MUs.

In the sequel, we firstly introduce the system model. Subsequently, we deal with the preliminaries of the Bayesian joint sync&loc algorithms, i.e., NLoS identification and AoA estimation. Later on, we present the principles of the L-BRF and DePF joint sync&loc algorithms, based on the synchronization scheme developed in Chapter 3. Finally, we present an in-depth analysis of the algorithms proposed in this chapter with the aid of simulation results.

4.2 System Model and Preliminaries

We consider a network of multiple APs with known locations, all backhauled by BSs. The APs are assumed to feature Uniform Planar Arrays (UPA), which allows for accurate azimuth and elevation AoA estimations, and to be able to continuously synchronize themselves with the backhauling BSs using the hybrid synchronization algorithm described in [46], [47]. This, in particular, guarantees a low time error among the neighboring APs, enabling a more precise cooperative localization. Further assumption is that, at each sync&loc period T, a set of APs, denoted by $\mathrm{ne}(i)$, are able to periodically exchange time-stamps with the i-th MU using the Fine Time Measurement (FTM) feature embedded in the communication devices and implemented by an existing protocol, e.g., Precision Time Protocol (PTP) [17]. From a packet containing these time-stamps, APs can also estimate the CIRs and AoA. The AP-MU link condition is considered both LoS and NLoS for the sake of generality. Nevertheless, it is known from [86], that for such a scenario the LoS probability is around 0.8, even growing to 0.95 when the AP density is 40 meters. A DNN trained by means of CIRs is employed to distinguish the LoS condition from the NLoS, permitting the localization unit to neglect the measurements conducted under NLoS conditions, thereby enhancing the accuracy of synchronization and localization. In what follows, we firstly present the DNN architecture that allows for a reliable NLoS/LoS identification based on the CIR. Subsequently, we briefly describe the principles of the MUSIC algorithm for AoA estimation.

4.2.1 Channel Impulse Response and NLoS Identification

The ability to estimate the CIR, or the Channel Frequency Responses (CFRs), is highly ubiquitous among the APs. It can be usually performed relatively straightforward with the already existing hardware and software components. Thus, relying on the CIR (or CFR) to develop localization algorithms appears to be a realistic approach. In fact, the AP-MU CIR is a rich source of information about

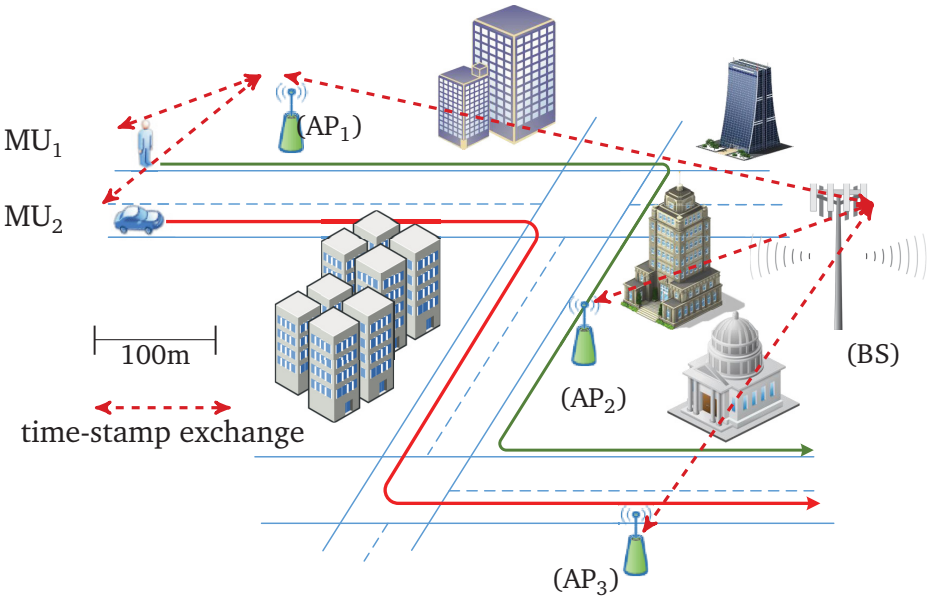


Figure 4.1: An example where MU joint sync&loc can be carried out.

the condition of the communication link, e.g., LoS or NLoS, and the location of the MU. More precisely, the former is crucial to know when estimating the latter since the reliability of the distance, time, and AoA measurements significantly decrease if they are conducted under NLoS conditions.

Furthermore, DNN units are also expected to be a part of communication devices as they are the cornerstone of many solutions for different communication problems such as slice management and anomaly detection [92]. Apart from that, in this specific application, i.e., NLoS identification, DNNs are proven to have a superb performance due to their remarkable ability to extract task-relevant features, which eventually paves the way towards an accurate prediction. On the contrary, other approaches such as SVM or Bayesian sequential test require human intervention which may be flawed and erroneous, given the limited intuition. In particular, humans may ignore/take into account the features that are relevant/irrelevant. Nevertheless, the challenge when using DNNs is to choose the optimal number of layers and neurons.

Figure 4.2 shows the architecture of the DNN deployed for NLoS-identification. The input layer has one channel fed with N samples, i.e., with the magnitude of the CIR [1]. The number of hidden layers and neurons in each hidden layer is set to l_H and n_H, respectively. The rationale for selecting these numbers is that, according to [90], any classifier function can be realized by two hidden layers, i.e.,

[1] We note that the phase of the CIR is extremely noisy and therefore extracting any information from it requires removing the noise. This is out of the scope of this work and, therefore, we simply rely only on the magnitude.

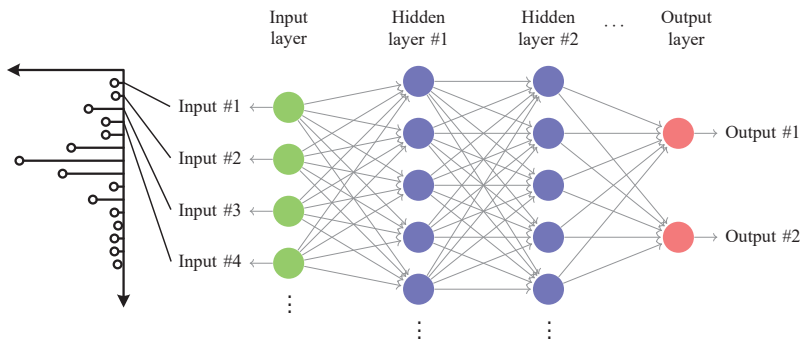


Figure 4.2: DNN architecture for NLoS-identification.

currently there is no theoretical reason to use more than two. However, the lack of evidence does not imply that the DNNs with more hidden layers do not improve the accuracy of classification. It rather suggests that the number of required hidden layers does not follow a well-established logic and is mostly determined by a trial-and-error process. Therefore, for the algorithm proposed in this work, we empirically determine the l_H that delivers the best performance. Furthermore, as a rule of thumb, the number of neurons is suggested to be between the number of inputs and that of the outputs to prevent under/overfitting.

Let the output probability vector of the DNN be $[1-p_{\text{nlos}}, p_{\text{nlos}}]$, where p_{nlos} denotes the probability of the CIR corresponding to an NLoS link. For the NLoS-identifier, we seek to train the DNN such that the output probability vector is as close as possible to $[1,0]/[0,1]$ for the LoS/NLoS CIRs. In other words, from the optimization point of view, we aim to design a loss function whose output is small when the DNN returns the correct vector and it is large otherwise. It turns out that the function that possesses the above-mentioned property is the logarithmic function [97]. Mathematically, the loss function is given by [91]

$$\mathscr{L} = -\frac{1}{M_c}\sum_{i=1}^{M_c} p^i_{\text{nlos}} \log(\hat{p}^i_{\text{nlos}}) + (1-p^i_{\text{nlos}})\log(1-\hat{p}^i_{\text{nlos}}), \tag{4.1}$$

where p^i_{nlos} denotes the true label corresponding to the i-th CIR sample in the data set and is '1' if the CIR corresponds to an NLoS link and '0' otherwise. Furthermore, M_c represents the total number of CIRs in the training set. The formulation in (4.1) is also known in the literature as the binary cross-entropy loss function. The goal of training is then to adjust the weights of the neurons such that (4.1) is minimized. Finally, when the trained DNN is employed in the context of the joint sync&loc algorithm, the decision on the link condition between the i-th MU and the j-th AP is fed into the algorithm using the binary parameter ζ_{ij}, which is set to '1' when $p_{\text{nlos}} > 0.5$ and '0' otherwise. Specifically,

if ζ_{ij} is '1', the communication link is considered NLoS and any measurement corresponding to it, i.e., time-stamp exchange and AoA, is dropped.

In the sequel, we present the principles of the AoA estimation algorithm, which draws on the same CIRs previously employed for NLoS identification.

4.2.2 Angle of Arrival

The CIRs/CFRs fed into the DNN to identify the link condition can be treated as a signal and be passed into the MUSIC algorithm to obtain the AoA. In the sequel, we present the principles of AoA estimation for UPAs based on [98]–[100]. The estimated AoA is given by

$$(\varphi_{ij}, \alpha_{ij}) = \arg\max_{\varphi,\alpha} \frac{1}{\mathbf{a}_n(\varphi,\alpha)^H \mathbf{N}\mathbf{N}^H \mathbf{a}_n(\varphi,\alpha)}, \tag{4.2}$$

where φ_{ij} and α_{ij} are the azimuth and elevation AoA of the signal received from the MU i at AP j, respectively. Parameter $\mathbf{a}_n(\varphi,\alpha)$ is the signal vector rotation on the n-th subcarrier and is given by

$$\mathbf{a}_n(\varphi,\alpha) = \begin{bmatrix} 1 \\ e^{i\frac{2\pi d}{\lambda}\sin(\alpha)(\sin(\varphi)+\cos(\varphi))} \\ e^{i\frac{2\pi d}{\lambda}\sin(\alpha)(\sin(\varphi)+2\cos(\varphi))} \\ \vdots \\ e^{i\frac{2\pi d}{\lambda}\sin(\alpha)((N_{\text{ant}}-1)\sin(\varphi)+(N_{\text{ant}}-2)\cos(\varphi))} \\ e^{i\frac{2\pi d}{\lambda}(N_{\text{ant}}-1)\sin(\alpha)(\sin(\varphi)+\cos(\varphi))} \end{bmatrix}_{N_{\text{ant}}^2\times 1}, \tag{4.3}$$

where N_{ant} denotes the number of AP antennas in one row (or column). Matrix $\mathbf{N}$ is constructed by $N_{\text{ant}}^2 - 1$ far right columns of the eigenvectors obtained when performing the eigen decomposition of the covariance matrix of the received signal. That is

$$\mathbf{R} = \mathbf{V}\mathbf{A}\mathbf{V}^H, \tag{4.4}$$

where matrices $\mathbf{A}$ and $\mathbf{V}$ contain the eigenvalues and eigenvectors, respectively. Furthermore,

$$\mathbf{R} = \frac{1}{N_s}\sum_{n=1}^{N_s} \mathbf{x}_n \mathbf{x}_n^H, \tag{4.5}$$

where the vector $\mathbf{x}_n$ is of dimension $N_{\text{ant}}^2 \times 1$ and represents the n-th element of the complex CIRs/CFRs. The number of time points/subcarriers (the size of FFT) is denoted by N_s. It is worth mentioning that, when constructing $\mathbf{N}$, the eigen

decomposition in (4.4) is assumed to sort the eigenvalues in a decreasing order. Lastly, each AP is assumed to have N_{ant}^2 CIRs at its disposal.

4.3 Clock Parameters and Position Estimation

4.3.1 Probabilistic Formulation of the Problem

Let ξ_i^k be the state of the vector variable $\xi_i \triangleq \begin{bmatrix} \tilde{\boldsymbol{\vartheta}}_i & \mathbf{p}_i \end{bmatrix}^T$ where $\tilde{\boldsymbol{\vartheta}}_i = \begin{bmatrix} \frac{1}{\tilde{\gamma}_i} & \frac{\tilde{\theta}_i}{\tilde{\gamma}_i} \end{bmatrix}$ and $\mathbf{p}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}$ after the k-th round of time-stamp exchange. Parameters x_i and y_i denote the position of node i on the x and y axes, respectively. The aim is then to infer the PDF corresponding to the k-th state, which can be written as

$$p(\xi_i^k|\{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \text{ne}(i)}) = \int p(\xi_i^0, \cdots, \xi_i^k|\{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \text{ne}(i)})\, d\xi_i^0 \cdots d\xi_i^{k-1}. \tag{4.6}$$

Following the steps explained in Section 2.4.1, Equation (4.6) can be simplified to

$$\begin{aligned} p(\xi_i^k|\{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \text{ne}(i)}) \propto \\ p(\xi_i^k|\{\mathbf{c}_{ij}^{1:k-1}, \varphi_{ij}^{1:k-1}, \zeta_{ij}^{1:k-1}\}_{\forall j \in \text{ne}(i)}) p(\{\mathbf{c}_{ij}^{k}, \varphi_{ij}^{k}, \zeta_{ij}^{k}\}_{\forall j \in \text{ne}(i)}|\xi_i^k). \end{aligned} \tag{4.7}$$

Figure 4.3 depicts the temporal evolution of ξ_i as well as its relation to the measurements at each time step. If the Gaussian assumption about ξ_i^0 held and the relation between all the states in Figure 4.3 were linear, we could conclude that the marginal in (4.7) would also be Gaussian distributed. Unfortunately, that is not the case in the joint sync&loc problem as the measurement equations (and consequently the correction steps) are partially non-linear. In concrete terms, the aforementioned problem stems from the non-linear relation between, on one hand, d_{ij} and the time-stamps in (3.2), (3.3), (3.4), and, on the other hand, the x_i/y_i and the measured AoA, φ_{ij}, in (4.2).

There are several approaches to tackle the non-linearity problem and, consequently, estimate the non-Gaussian posterior distribution. In [48], we proposed to undertake the Taylor expansion of the non-linear terms around the prediction point, while [6], [9], [101] have employed EKF to address the non-linearity. In addition to being prone to divergence, which is hard to mitigate analytically, all of these methods require initialization and even then are only able to deliver medium accuracy. In what follows, we first discuss the principles of the L-BRF

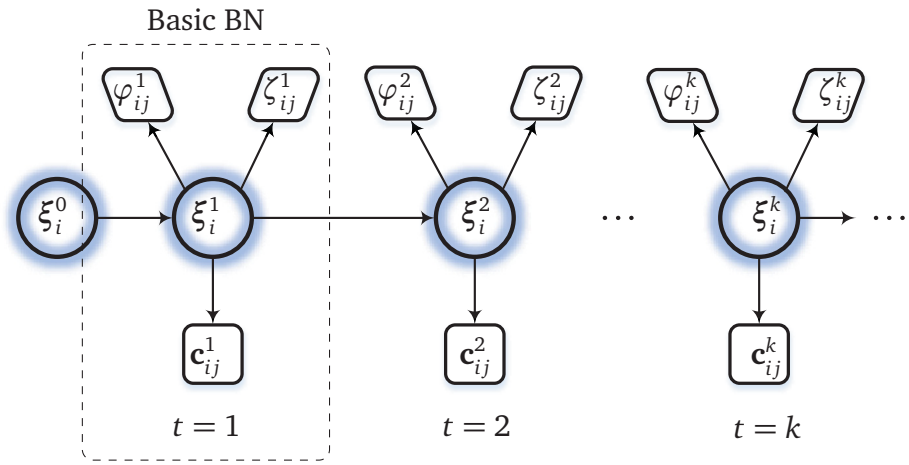


Figure 4.3: Dynamic Bayesian network representing the temporal evolution of the vector variable ξ_i and its relation to the measurements.

approach we proposed in [47], [48]. Subsequently, we present the details of a novel joint sync&loc approach based on PGMFs introduced in [49].

4.3.2 Linear Bayesian Recursive Filtering (L-BRF)

The non-linearity in BRF stems from the relation between the distance measurements and the time-stamps, on the one hand, and the relation between the location parameters and the AoA measurements, on the other hand. In particular, location-related parameters appear when expanding the propagation delay d_{ij} (or d_{ji}) as

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{4.8}$$

Furthermore, each AP is assumed to be equipped with an UPA and is able to perform Angle of Arrival (AoA) estimation in each round of time-stamp exchange. This estimation is given by

$$\arctan \frac{y_i - y_j}{x_i - x_j} = \varphi_{ij} + n_\varphi, \tag{4.9}$$

where φ_{ij} denotes the true AoA and $\mathcal{N}(n_\varphi|0, \sigma_\varphi^2)$ is the zero mean Gaussian noise stemming from the AoA estimation algorithm. In our simulations we rely on the MUSIC algorithm explained in Section 4.2.2, to calculate σ_φ while φ_{ij} is computed when feeding the corresponding CIR to the MUSIC algorithm.

In what follows, we further delve into the steps of parameter estimation of the above-mentioned distribution. Firstly, the details of *prediction* step are described, where all the parameters are denoted by $(\cdot)\_$. Next, we obtain the likelihood of

the measurements whose parameters are represented by $(\cdot)_+$. Lastly, we compute the parameters of the posterior distribution in (4.7).

4.3.2.1 Prediction

Given the dynamics of MUs' clocks and movements, a reasonable prediction for ξ_i^k is given in [8]. That is,

$$\xi_i^k = \mathbf{A}\xi_i^{k-1} + \mathbf{u} + \mathbf{n}_i^{k-1}, \tag{4.10}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ T & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} 0 & T & 0 & 0 \end{bmatrix}^T.$$

Parameter T denotes the duration of one time-stamping round. Furthermore, $\mathbf{n}_i^{k-1}$ denotes the Gaussian noise vector and assumed to have zero mean and covariance matrix $\mathbf{Q}_n = \text{diag}(\sigma_\gamma^2, \sigma_\theta^2, \sigma_x^2, \sigma_y^2)$. In general, the design of $\mathbf{Q}_n$ is a difficult task. In particular, if it is too small, the filter will be overconfident in its prediction model and will diverge from the actual solution. In contrast, if it is too large, then it will be unduly dominated by the noise in the measurements and perform sub-optimally. In this work, we follow the design model discussed in [101], [102]. Given (4.10), the prediction term can be written as

$$p(\xi_i^k | \{\mathbf{c}_{ij}^{1:k-1}, \varphi_{ij}^{1:k-1}, \zeta_{ij}^{1:k-1}\}_{\forall j \in \text{ne}(i)}) \sim \mathcal{N}(\xi_i^k | (\boldsymbol{\mu}_i^k)_-, (\boldsymbol{\Sigma}_i^k)_-), \tag{4.11}$$

where $(\boldsymbol{\mu}_i^k)_- = \mathbf{A}\boldsymbol{\mu}_i^{k-1}$ and $(\boldsymbol{\Sigma}_i^k)_- = \mathbf{A}\boldsymbol{\Sigma}_i^{k-1}\mathbf{A}^T + \mathbf{Q}_n$. Such recursive relationships are straightforward to obtain given Equation (4.10).

4.3.2.2 Correction

To permit (4.7) to have a closed-form solution, the relation between parameters in the measurement equations (3.2), (3.3), (3.4), and (4.9) must be linear. However, this is not the case as the distance function is not linear. Therefore, we draw on Taylor expansion to linearize the non-linear terms, thereby allowing for a

closed-form solution for (4.7). In particular, *we write the Taylor expansion around the point predicted by the prediction step* in (4.10). Thus

$$\frac{d_{ij}}{v_c} \approx a^k_{ij,0} + \mathbf{a}^k_{ij} \bullet (\mathbf{p}_i - (\mathbf{p}^k_i)_-), \tag{4.12}$$

$$\arctan(\frac{y_i - y_j}{x_i - x_j}) \approx b^k_{ij,0} + \mathbf{b}^k_{ij} \bullet (\mathbf{p}_i - (\mathbf{p}^k_i)_-), \tag{4.13}$$

with "•" denoting the inner scalar product of vectors. Furthermore, $a^k_{ij,0}$, $\mathbf{a}^k_{ij}$, $b^k_{ij,0}$, and $\mathbf{b}^k_{ij}$ are calculated by

$$a^k_{ij,0} = \frac{1}{v_c}\left|\left|(\mathbf{p}^k_i)_- - \mathbf{p}_j\right|\right|, \qquad \mathbf{a}^k_{ij} = \frac{1}{v_c^2 a^k_{ij,0}}\left((\mathbf{p}^k_i)_- - \mathbf{p}_j\right), \tag{4.14}$$

$$b^k_{ij,0} = \arctan(\frac{\mathbf{a}^k_{ij}[2]}{\mathbf{a}^k_{ij}[1]}), \qquad \mathbf{b}^k_{ij} = \frac{1}{a^k_{ij,0}}\left[-\mathbf{a}^k_{ij}[2], \quad \mathbf{a}^k_{ij}[1]\right], \tag{4.15}$$

where $||\cdot||$ denotes the Euclidean norm. Given Equations (4.12) and (4.13), for the localization performed by a single AP[2], we can write Equations (3.2), (3.3), (3.4) and (4.9) in matrix form as

$$\mathbf{B}_{ij}\xi^k_i = \mathbf{r}_{ij} + \mathbf{z}_{ij}, \tag{4.16}$$

where $\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{R}_{ij})$ with $\mathbf{R}_{ij} = \mathrm{diag}(\sigma^2_{T_{ij}}, \sigma^2_{T_{ij}}, \sigma^2_{R_{ij}}, \sigma^2_{\varphi})$, and

$$\mathbf{B}_{ij} = \begin{bmatrix} c_i(t^k_2) & -1 & -\mathbf{a}^k_{ij} \\ c_i(t^k_4) & -1 & -\mathbf{a}^k_{ij} \\ c_i(t^k_5) & -1 & \mathbf{a}^k_{ij} \\ 0 & 0 & \mathbf{b}^k_{ij} \end{bmatrix},$$

and $\mathbf{r}_{ij}$ is constructed as

$$\mathbf{r}_{ij} = \left[c_j(t^k_1) - \mathbf{a}^k_{ij} \bullet (\mathbf{p}^k_i)_-, \quad c_j(t^k_3) - \mathbf{a}^k_{ij} \bullet (\mathbf{p}^k_i)_-, \quad c_j(t^k_6) + \mathbf{a}^k_{ij} \bullet (\mathbf{p}^k_i)_-, \quad \varphi^k_{ij} + \mathbf{b}^k_{ij} \bullet (\mathbf{p}^k_i)_-\right]^T. \tag{4.17}$$

[2]We note that the extension to the multiple-AP case is straightforward as the process is recursive and can incorporate new measurements from any additional AP.

Finally, we can write the correction term as

$$p(\{\mathbf{c}_{ij}^k, \varphi_{ij}^k, \zeta_{ij}^k\}_{\forall j \in \mathrm{ne}(i)} | \boldsymbol{\xi}_i^k) \sim \mathcal{N}((\boldsymbol{\mu}_i^k)_+, (\boldsymbol{\Sigma}_i^k)_+), \tag{4.18}$$

where $(\boldsymbol{\mu}_i^k)_+ = (\mathbf{B}_{ij}^T \mathbf{B}_{ij})^{-1} \mathbf{B}_{ij}^T \mathbf{r}_{ij}$, and $(\boldsymbol{\Sigma}_i^k)_+ = (\mathbf{B}_{ij}^T \mathbf{B}_{ij})^{-1} \mathbf{B}_{ij}^T \mathbf{R}_{ij} \mathbf{B}_{ij} (\mathbf{B}_{ij}^T \mathbf{B}_{ij})^{-T}$. Such recursive relationships for the mean vector and the covariance matrix are readily obtained from Equation (4.16).

4.3.2.3 Estimation

Considering (4.11) and (4.18), the estimated distribution in (4.7) is given by

$$p(\boldsymbol{\xi}_i^k | \{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j \in \mathrm{ne}(i)}) \sim \mathcal{N}(\boldsymbol{\mu}_i^k, \boldsymbol{\Sigma}_i^k), \tag{4.19}$$

where

$$\boldsymbol{\mu}_i^k = \left[(\boldsymbol{\Sigma}_i^k)_+ + (\boldsymbol{\Sigma}_i^k)_-\right]^{-1} \left((\boldsymbol{\Sigma}_i^k)_+ (\boldsymbol{\mu}_i^k)_- + (\boldsymbol{\Sigma}_i^k)_- (\boldsymbol{\mu}_i^k)_+\right), \tag{4.20}$$

$$\boldsymbol{\Sigma}_i^k = \left[(\boldsymbol{\Sigma}_i^k)_-^{-1} + (\boldsymbol{\Sigma}_i^k)_+^{-1}\right]^{-1}. \tag{4.21}$$

The parameters in (4.11), (4.18), and (4.19) are calculated recursively and, in each iteration k, an estimation of the clock skew, clock offset, and position can be obtained by

$$\tilde{\gamma}_i^k = \frac{1}{\boldsymbol{\mu}_i^k(1)}, \ \tilde{\theta}_i^k = \frac{\boldsymbol{\mu}_i^k(2)}{\boldsymbol{\mu}_i^k(1)}, \ x_i^k = \boldsymbol{\mu}_i^k(3), \text{ and } y_i^k = \boldsymbol{\mu}_i^k(4). \tag{4.22}$$

Algorithm 3 summarizes this recursive process. In particular, we initialize the prior distribution $\mathrm{p}(\xi_i^0)$ in step 1. It is worth mentioning that the position initialization has a major impact on the performance of the algorithm and can, if inappropriately chosen, lead to its divergence. For the L-BRF, similar to [6], we assume that the initial position of the MUs is available via Global Navigation Satellite System (GNSS). The initialization of clock parameters is, however, straightforward and can be done according to [24], [45], [46], with $\mathcal{N}(\gamma_i | 1, \infty)$ and $\mathcal{N}(\theta_i | 0, \infty)$ for clock skew and offset, respectively. In steps 2-6, the timestamps are exchanged between a mobile user and its serving APs. Then, the CIRs are computed, the AoAs are estimated, and the link conditions for all MU-AP links are determined. In steps 7-11 the posterior distribution is computed recursively using the measurements from the APs with LoS to the MU. Finally, in step 12 an estimation of the clock and position parameters can be obtained. The process repeats itself periodically.

Algorithm 3 BRF-based joint sync&loc.

1: Initialize $\mathrm{p}(\xi_i^0)$ using information about MU position available via, e.g., GNSS.

2: **for all** the APs in $\mathrm{ne}(i)$ **do**
3: Perform the time-stamp exchange mechanism described in Section 3.2.3 and Figure 3.4.
4: Estimate the CIR using QuaDRiGa model.
5: Estimate the AoA and the link condition
6: **end for**
7: **for all** LoS links (ζ_{ij}^k=0) **do**
8: Calculate the mean vector and covariance matrix of the *prediction* using (4.11).
9: Construct $\mathbf{B}_{ij}$, $\mathbf{R}_{ij}$, and $\mathbf{r}_{ij}$ using the measurements and obtain the mean vector and covariance matrix of *correction* PDF using (4.18).
10: Update the parameters of the posterior distribution using (4.19).
11: **end for**
12: Estimate the clock and position parameters using (4.22).
13: Go to step 7.

4.3.3 Particle Gaussian Mixture Filter

The idea underpinning PGMFs is to approximate a posterior PDF by the sum of weighted *Gaussian density functions* (gdfs) [79]. Leveraging on this idea, we can write the posterior distribution in (4.7) as

$$p(\xi_i^k|\{\mathbf{c}_{ij}^{1:k}, \varphi_{ij}^{1:k}, \zeta_{ij}^{1:k}\}_{\forall j\in \mathrm{ne}(i)}) = \sum_{f=1}^{F} w_f^k \mathcal{N}(\xi_i^k|\boldsymbol{\mu}_f^k, \boldsymbol{\Sigma}_f^k) \tag{4.23}$$

where

$$\sum_{f=1}^{F} w_f^k = 1, \quad w_f^k \geq 0 \ \forall f.$$

Moreover, $\boldsymbol{\mu}_f^k = \begin{bmatrix} \boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_f^k & \boldsymbol{\mu}(\mathbf{p}_i)_f^k \end{bmatrix}$ and

$$\boldsymbol{\Sigma}_f^k = \begin{bmatrix} \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_f^k & \mathbf{0}_2 \\ \mathbf{0}_2 & \boldsymbol{\Sigma}(\mathbf{p}_i)_f^k \end{bmatrix}$$

denote the mean vector and covariance matrix of the f-th gdf in the k-th round of estimation, respectively. Parameter F represents the total number of gdfs. Furthermore, $\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_f^k/\boldsymbol{\mu}(\mathbf{p}_i)_f^k$ and $\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_f^k/\boldsymbol{\Sigma}(\mathbf{p}_i)_f^k$ represent the mean vector and covariance matrix respectively corresponding to the vector variable $\tilde{\boldsymbol{\vartheta}}_i/\mathbf{p}_i$.

As mentioned before, one of the limitations of PGMFs is the dimensionality problem. To deal with this problem, we can capitalize on the linear relationships

in the measurements, if there exist any. Seeking to further simplify (4.23), we reformulate (3.2), (3.3), and (3.4) as follows. Subtracting (3.2) from (3.3) leads to

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_4^k)-c_i(t_2^k)) = c_j(t_3^k)-c_j(t_1^k)+T_{ij}^{k,1}-T_{ij}^{k,0}, \tag{4.24}$$

while summing up (3.3) and (3.4)

$$\frac{1}{\tilde{\gamma}_i}(c_i(t_4^k)+c_i(t_5^k)-2\tilde{\theta}_i) = c_j(t_3^k)+c_j(t_6^k)+T_{ij}^{k,1}-R_{ij}^k. \tag{4.25}$$

It is straightforward to observe that $\tilde{\boldsymbol{\vartheta}}_i^k$, on one hand, is linearly dependent on the time-stamps, and, on the other hand, does not depend on $\mathbf{p}_i$. This suggests that, although the $p(\mathbf{c}_{ij}^k|\xi_i)$ cannot be considered Gaussian distributed in general, it can be indeed considered Gaussian across the $\tilde{\boldsymbol{\vartheta}}_i$ axis as both T_{ij} and R_{ij} are Gaussian distributed and $\tilde{\boldsymbol{\vartheta}}_i^k$ has linear relationship with the measurements (time-stamps). In fact, we exploit the linear substructures in the model to keep the state dimensions low. Consequently, the gdfs can be employed only across the $\mathbf{p}_i$ axis transforming the structure of (4.23) into multiplication of a single gdf across $\tilde{\boldsymbol{\vartheta}}_i$ and sum of multiple weighted gdfs across $\mathbf{p}_i$ (visualized in Figure 4.4). Such a structure not only lays the ground for the hybrid parametric and particle-based implementation of BRF-based joint sync&loc estimation, but also dramatically reduces the computational burden. Given above, (4.23) can be simplified as

$$p(\{\mathbf{c}_{ij}^{1:k},\varphi_{ij}^{1:k},\zeta_{ij}^{1:k}\}_{\forall j\in \mathrm{ne}(i)}|\xi_i^k) = \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^k,\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^k)\sum_{f=1}^{F} w_f^k\mathcal{N}(\mathbf{p}_i^k|\boldsymbol{\mu}(\mathbf{p}_i)_f^k,\boldsymbol{\Sigma}(\mathbf{p}_i)_f^k). \tag{4.26}$$

We note that when $\boldsymbol{\Sigma}(\mathbf{p}_i)_f^k$ approaches 0, the term $\mathcal{N}(\mathbf{p}_i^k|\boldsymbol{\mu}(\mathbf{p}_i)_f^k,\boldsymbol{\Sigma}(\mathbf{p}_i)_f^k)$ tends towards $\delta(\mathbf{p}_i^k-\boldsymbol{\mu}(\mathbf{p}_i)_f^k)$, where $\delta(\cdot)$ denote the Dirac impulse function. Such a function forms the basis of the classical particle filter. In what follows, we further delve into the steps of parameter estimation of the above-mentioned distribution. Firstly, the details of *prediction* step are described, where all the parameters are denoted by $(\cdot)_-$. Next, we obtain the likelihood of the measurements whose parameters are represented by $(\cdot)_+$. Lastly, we compute the parameters of the posterior distribution in (4.26) and perform the resampling.

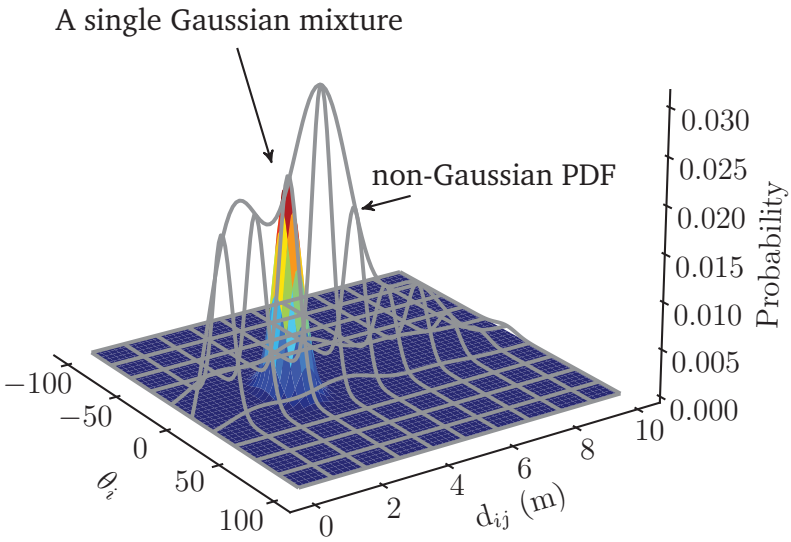


Figure 4.4: An example distribution of the ξ_i for a given time-stamp measurement. Parameters θ_i and d_{ij} represent the clock offset and the distance, respectively. Note that ξ_i has been reduced to only two parameters for the purpose of visualization.

4.3.3.1 Prediction

Given the linear dynamics of MUs' clocks and movements, a reasonable prediction for ξ_i^k is given by

$$p(\xi_i^k|\{\mathbf{c}_{ij}^{1:k-1}, \varphi_{ij}^{1:k-1}, \zeta_{ij}^{1:k-1}\}_{\forall j\in \mathrm{ne}(i)}) = \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k|\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_-^k, \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_-^k)\sum_{f=1}^{F} w_{f-}^k \mathcal{N}(\mathbf{p}_i^k|\boldsymbol{\mu}(\mathbf{p}_i)_{f-}^k, \boldsymbol{\Sigma}(\mathbf{p}_i)_{f-}^k) \quad (4.27)$$

where

$$w_{f-}^k = \frac{1}{F}\mathbf{1}_F, \qquad \boldsymbol{\mu}(\mathbf{p}_i)_{f-}^k = \boldsymbol{\mu}(\mathbf{p}_i)_f^{k-1} + \mathbf{n}_f,$$

with $\mathbf{n}_f$ being the noise vector derived from the distribution $\mathcal{N}(\mathbf{n}|\mathbf{0}, \mathbf{Q}_n(\mathbf{p}_i))$, for $\mathbf{Q}_n(\mathbf{p}_i) = \mathrm{diag}(\sigma_x^2, \sigma_y^2)$. In practice, we initialize $\boldsymbol{\Sigma}(\mathbf{p}_i)_{f-}^k = \mathrm{diag}(F^{-0.4}, F^{-0.4})$ which is proved to be the optimal choice in [78]. Furthermore, according to [46],

$$\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_-^k = \mathbf{F}\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)^{k-1} + \mathbf{u}, \qquad \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_-^k = \mathbf{F}\boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)^{k-1}\mathbf{F}^T + \mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i) \quad (4.28)$$

with

$$\mathbf{F} = \begin{bmatrix} 1 & 0 \\ T & 1 \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} 0 & T \end{bmatrix}^T, \qquad \mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i) = \mathrm{diag}(\sigma_\gamma^2, \sigma_\theta^2).$$

The matrices $\mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i)$ and $\mathbf{Q}_n(\mathbf{p}_i)$ denote the covariance of the zero-mean Gaussian noises on each gdf across the $\tilde{\boldsymbol{\vartheta}}_i$ and $\mathbf{p}_i$ axes, respectively. In general, as mentioned before, designing $\mathbf{Q}_n(\cdot)$ is a challenging task. In particular, if it is too small, the filter will be overconfident in its prediction model and will diverge from the actual solution. In contrast, if it is too large, it will be unduly dominated by the noise in the measurements and perform sub-optimally [102]. Similar to [4], [18], [23]. We set σ_γ^2 and σ_θ^2, such that the external noises as well as the residues from the previous iteration are accounted for. Furthermore, to determine the value of σ_x^2 and σ_y^2, we follow the design model discussed in [101], [102]. That is, opting for a noise variance that is large enough to allow the gdfs to assign a reasonable probability to the locations where the MU might be. In the urban scenario, for example, in most areas, the maximum permitted speed is 50 km/h (≈14 m/s), resulting in $\sigma_x = \sigma_y = 14T$.

4.3.3.2 Measurement Likelihood and Weight Update

The same structure as (4.26) is imposed on the likelihood of the measurements. That is,

$$p(\{\mathbf{c}_{ij}^k, \varphi_{ij}^k, \zeta_{ij}^k\}_{\forall j \in \mathrm{ne}(i)} | \xi_i^k) = \mathcal{N}(\tilde{\boldsymbol{\vartheta}}_i^k | \boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_+^k, \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_+^k) \sum_{f=1}^{F} w_{f+}^k \mathcal{N}(\mathbf{p}_i^k | \boldsymbol{\mu}(\mathbf{p}_i)_{f+}^k, \boldsymbol{\Sigma}(\mathbf{p}_i)_{f+}^k) \quad (4.29)$$

To obtain the parameters of the above likelihood, we firstly transform (4.24) and (4.25) into the matrix form. That is,

$$\mathbf{B}_{ij}^k \tilde{\boldsymbol{\vartheta}}_i^k = \mathbf{r}_{ij}^k + \mathbf{z}_{ij}, \quad (4.30)$$

where $\mathbf{z}_{ij} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{R}_{ij}^k)$ with $\mathbf{R}_{ij}^k = \mathrm{diag}(2\sigma_{T_{ij}}^2, \sigma_{T_{ij}}^2 + \sigma_{R_{ij}}^2)$, and

$$\mathbf{B}_{ij}^k = \begin{bmatrix} c_i(t_4^k) - c_i(t_2^k) & 0 \\ c_i(t_4^k) + c_i(t_5^k) & -2 \end{bmatrix}, \mathbf{r}_{ij}^k = \begin{bmatrix} c_j(t_3^k) - c_j(t_1^k) \\ c_j(t_3^k) + c_j(t_6^k) \end{bmatrix}.$$

The mean and covariance matrix of the gdfs across the $\tilde{\boldsymbol{\vartheta}}_i$ axis can be written as

$$\boldsymbol{\mu}(\tilde{\boldsymbol{\vartheta}}_i)_+^k = \mathbf{A}_{ij}^k \mathbf{r}_{ij}^k, \qquad \boldsymbol{\Sigma}(\tilde{\boldsymbol{\vartheta}}_i)_+^k = \mathbf{A}_{ij}^k \mathbf{R}_{ij}^k (\mathbf{A}_{ij}^k)^T, \quad (4.31)$$

where $\mathbf{A}_{ij}^k = ((\mathbf{B}_{ij}^k)^T \mathbf{B}_{ij}^k)^{-1} (\mathbf{B}_{ij}^k)^T$.

To obtain the location parameters corresponding to each gdf, we can assume that the measurement equations are linear around the points predicted by the prediction step. That is, to approximate them with their first-order Taylor expansions, the details of which are thoroughly explained in [47], [48] and Section 4.3.2. The measurement equations we rely on to estimate the parameters of the likelihoods are (3.4) and

$$\arctan\left(\frac{y_i - y_j}{x_i - x_j}\right) = \varphi_{ij}^k, \tag{4.32}$$

where φ_{ij}^k is calculated as explained in Section 4.2.2. Carrying out the necessary mathematical manipulation, we can write the same relation as (4.30) for each gdf. That is,

$$\mathbf{B}_{ij,f}^k \mathbf{p}_i^k = \mathbf{r}_{ij,f}^k + \mathbf{z}_{ij,f}, \tag{4.33}$$

where $\mathbf{z}_{ij,f} \sim \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{R}_{ij,f})$ with $\mathbf{R}_{ij,f} = \mathrm{diag}(\sigma_{R_{ij}}^2, \sigma_\varphi^2)$. Furthermore,

$$\mathbf{B}_{ij,f} = \begin{bmatrix} \mathbf{a}_{ij,f}^k \\ \mathbf{b}_{ij,f}^k \end{bmatrix}$$

with the vectors $\mathbf{a}_{ij,f}^k$ and $\mathbf{b}_{ij,f}^k$ calculated as

$$a_{ij,f}^k = \frac{1}{v_c}\left|\left|\mu(\mathbf{p}_i)_{f-}^k - \mathbf{p}_j\right|\right|, \qquad \mathbf{a}_{ij,f}^k = \frac{1}{v_c^2 a_{ij,f}^k}\left(\mu(\mathbf{p}_i)_{f-}^k - \mathbf{p}_j\right), \tag{4.34}$$

$$b_{ij,f}^k = \arctan\left(\frac{\mathbf{a}_{ij,f}^k[2]}{\mathbf{a}_{ij,f}^k[1]}\right), \qquad \mathbf{b}_{ij,f}^k = \frac{1}{a_{ij,f}^k}\left[-\mathbf{a}_{ij,f}^k[2], \quad \mathbf{a}_{ij,f}^k[1]\right]. \tag{4.35}$$

Finally, $\mathbf{r}_{ij,f}$ is constructed as

$$\mathbf{r}_{ij,f} = \begin{bmatrix} c_j(t_6^k) - a_{ij,f}^k + \mu(\mathbf{p}_i)_{f-}^k \bullet \mathbf{a}_{ij,f}^k - \left[c_i(t_5^k) \quad -1\right] \bullet \mu(\xi_i i)_+^k \\ \varphi_{ij}^k - b_{ij,f}^k + \mu(\mathbf{p}_i)_{f-}^k \bullet \mathbf{b}_{ij,f}^k \end{bmatrix}. \tag{4.36}$$

We note that (4.34) and (4.35) are computed by means of Taylor expansions of Equations (3.4) and (4.32) around the predicted point $\mu(\mathbf{p}_i)_{f-}^k$ with the known $\mu(\tilde{\boldsymbol{\vartheta}}_i)_+^k$ obtained by (4.31). Given (4.33) (and similar to (4.31)) we can write

$$\mu(\mathbf{p}_i)_{f+}^k = \mathbf{A}_{ij,f}^k \mathbf{r}_{ij,f}^k, \qquad \Sigma(\mathbf{p}_i)_{f+}^k = \mathbf{A}_{ij,f}^k \mathbf{R}_{ij,f}^k (\mathbf{A}_{ij,f}^k)^T, \tag{4.37}$$

where $\mathbf{A}^k_{ij,f} = ((\mathbf{B}^k_{ij,f})^T \mathbf{B}^k_{ij,f})^{-1}(\mathbf{B}^k_{ij,f})^T$. Furthermore, it is straightforward to see that

$$w^k_{f+} = \mathcal{N}(\mathbf{p}^k_i = \mu(\mathbf{p}_i)^k_{f+} | \mu(\mathbf{p}_i)^k_{f+}, \Sigma(\mathbf{p}_i)^k_{f+}). \tag{4.38}$$

In other words, the weights are equal to the likelihood of the mean of each gdf. Having obtained the prediction and correction distribution, in the following we estimate the posterior distribution.

4.3.3.3 Posterior Estimation

Having taken the necessary steps, we can now compute (4.26) as an approximation for the posterior distribution in (4.7). Multiplying (4.27) and (4.29), the parameters of (4.26) can be given by

$$\mu(\tilde{\boldsymbol{\vartheta}}_i)^k = \left[\Sigma(\tilde{\boldsymbol{\vartheta}}_i)^k_- + \Sigma(\tilde{\boldsymbol{\vartheta}}_i)^k_+\right]^{-1}\left(\Sigma(\tilde{\boldsymbol{\vartheta}}_i)^k_+ \mu(\tilde{\boldsymbol{\vartheta}}_i)^k_- + \Sigma(\tilde{\boldsymbol{\vartheta}}_i)^k_- \mu(\tilde{\boldsymbol{\vartheta}}_i)^k_+\right), \tag{4.39}$$

$$\Sigma(\tilde{\boldsymbol{\vartheta}}_i)^k = \left[\left(\Sigma(\tilde{\boldsymbol{\vartheta}}_i)^k_-\right)^{-1} + \left(\Sigma(\tilde{\boldsymbol{\vartheta}}_i)^k_+\right)^{-1}\right]^{-1}. \tag{4.40}$$

The final estimation of the clock skew and offset can then be given by

$$\tilde{\gamma}^k_i = \frac{1}{\mu(\tilde{\boldsymbol{\vartheta}}_i)^k[1]}, \qquad \tilde{\theta}^k_i = \frac{\mu(\tilde{\boldsymbol{\vartheta}}_i)^k[2]}{\mu(\tilde{\boldsymbol{\vartheta}}_i)^k[1]}. \tag{4.41}$$

Furthermore, each gdf can be updated across $\mathbf{p}_i$ axis by

$$\mu(\mathbf{p}_i)^k_f = \left[\Sigma(\mathbf{p}_i)^k_{f-} + \Sigma(\mathbf{p}_i)^k_{f+}\right]^{-1}\left(\Sigma(\mathbf{p}_i)^k_{f+}\mu(\mathbf{p}_i)^k_{f-} + \Sigma(\mathbf{p}_i)^k_{f-}\mu(\mathbf{p}_i)^k_{f+}\right), \tag{4.42}$$

$$\Sigma(\mathbf{p}_i)^k_f = \left[\left(\Sigma(\mathbf{p}_i)^k_{f-}\right)^{-1} + \left(\Sigma(\mathbf{p}_i)^k_{f+}\right)^{-1}\right]^{-1}. \tag{4.43}$$

Next, the weights can be updated as

$$w^k_f = \frac{w^k_{f-} w^k_{f+}}{\sum_{f=1}^F w^k_{f-} w^k_{f+}}. \tag{4.44}$$

Given (4.42), (4.43), (4.44), the final position estimation can be given by

$$\mathbf{p}^k_i = \sum_{f=1}^F w^k_f \mu(\mathbf{p}_i)^k_f, \tag{4.45}$$

which is equivalent to the expectation of the approximated posterior distribution.

Algorithm 4 DePF joint sync&loc.

1: Initialize $p(\xi_i^0)$.
2: **for all** the APs in $ne(i)$ **do**
3: Perform the time-stamp exchange mechanism described in Section 3.2.3 and Figure 3.4.
4: Estimate the CIR.
5: Estimate the AoA and the link condition
6: **end for**
7: **for all** LoS links (ζ_{ij}^k=0) **do**
8: Calculate the mean vector and covariance matrix of the *prediction* using (4.27).
9: Construct $\mathbf{B}_{ij}^k$, $\mathbf{B}_{ij,f}^k$, $\mathbf{R}_{ij}^k$, $\mathbf{R}_{ij,f}^k$, $\mathbf{r}_{ij}^k$ and $\mathbf{r}_{ij,f}^k$ and update the parameters of (4.29) using (4.37) and (4.38).
10: Update the parameters of the posterior distribution using (4.39), (4.40), (4.42), and (4.43).
11: **end for**
12: Estimate the clock and position parameters using (4.41) and (4.45).
13: **if** $N_{\text{eff}} < \frac{2}{3}F$ **then**
14: Perform resampling.
15: **end if**
16: Go to step 7.

4.3.3.4 Resampling and Tuning

Resampling is one of the most crucial steps when using PGMFs. Without the resampling step, the filter would suffer from sample depletion. That is, after a while, all gdfs but a few will have negligible weight. Consequently, the posterior will be approximated with only a few gdfs, leading to its underestimation. To overcome this shortcoming, in each iteration, we replace the minor-weight gdfs with new ones whose means are sampled from the approximated posterior. The sample depletion can be monitored throughout the filtering process by calculating the number of effective gdfs as

$$N_{\text{eff}} = \frac{1}{\sum_{f=1}^{F}(w_f^k)^2}. \tag{4.46}$$

As can be seen, N_{eff} attains its maximum when all the weights are equal to $\frac{1}{F}$ and falls to its minimum when all but a single weight is equal to zero. Following the suggestion in [80], we perform resampling when the $N_{\text{eff}} < \frac{2}{3}F$, striking a balance between the number of effective particles and the frequency of resampling.

Algorithm 4 summarizes the steps required to perform DePF joint sync&loc. In particular, we initialize the gdfs in the area of interest in step 1. In steps 2-6,

Table 4.1: Complexity comparison of L-BRF and PGMF.

	L-BRF	PGMF
Prediction	$2l^3 + l^2$	$2l^3 + l^2$
Likelihood/Correction	$2l^3 + l^2$	$2Fn^3 + Fn^2 + 2l^3 + l^2$
Estimation	$7l^2$	$7Fn^2 + nF + 7l^2$
Total	$O(l^3)$	$O(Fn^3 + l^3)$

the time-stamps are exchanged between a mobile user and its serving APs. Then, the CIRs are computed, the AoAs are estimated, and the link conditions for all MU-AP links are determined. In steps 7-11 the posterior distribution is computed recursively using the measurements from the APs with LoS to the MU. Finally, in steps 13-15, the resampling is performed if the number of effective gdfs has dropped below the threshold. The process repeats itself periodically.

4.3.4 Complexity of the Algorithm

The computational complexity of different types of BRF and PF filters including L-BRF and PGMF has been extensively discussed in [80], [103]. Parameters l, n, and F denote the number of linear state variables, non-linear state variables, and gdfs (or mixtures), respectively. For the sake of simplicity, we only consider the number of multiplications to evaluate the complexity. Table 4.1 shows the complexity for each step of L-BRF and PGMF. For the prediction step, it can be seen from (4.27) and (4.28) that two square matrix multiplications and a matrix-vector multiplication are needed. We note that the computation cost of generating random variables is $O(1)$. The same holds for the likelihood computation given (4.31) and (4.37). In the PGMF, the L-BRF is repeated F times for each gdf. For the estimation step, the L-BRF needs 4 matrix inversions and 3 matrix-vector multiplications. The same number of multiplications is necessary for each gdf of the PGMF in addition to the multiplications between the weights and the particles to obtain the final estimation. Finally, as we need to perform a cumulative sum to perform resampling, its complexity is considered $O(F)$. It is apparent that PGMF adds an overhead, however, it turns out that, according to [80], [103], PGMF is more efficient, especially when the uncertainty of the measurements increases, which is the case when the time-stamp accuracy is low or a measurement is conducted under NLoS condition.

4.4 Simulation Results and Comparison with State-of-the-Art

In this section, we evaluate the performance of the techniques employed in this work. In particular, firstly we evaluate the performance of a DNN-based NLoS identifier. Next, we present the result of AoA estimation. Lastly, the performance of the joint sync&loc algorithms developed in this work is thoroughly analyzed.

4.4.1 DNN-based NLoS Identification

To perform NLoS identification, the DNN in Figure 4.2 needs to be trained first. The training data is obtained using the QuaDRiGa channel model [104]. Specifically, the MU-AP CIRs throughout the MU's movement profile can be implemented under the "3GPP_38.901_UMi" scenario. We collect 5000 CIR realizations for each scenario, i.e., LoS and NLoS, 80% of which is used for the training purpose while the remaining 20% is treated as the test set. To prepare the CIRs to be fed into the DNN, we firstly input them into a 64-point FFT to obtain the CFRs. Subsequently, we take the magnitude of the CFRs and normalize each to its maximum component. The normalized magnitudes of the CFRs are then fed into a DNN with 2 hidden layers, each comprising 64 (size of FFT) neurons with Rectified Linear Unit (ReLU) activation function. The loss function in (4.1) is then optimized using an Adam optimizer to obtain the weights of each neuron. Furthermore, the probability that a CFR corresponds to a LoS and NLoS link condition is indicated by the DNN's two output neurons with softmax activation function. Note that we could alternatively use the CIRs without transforming them into CFRs. The number of neurons in that case would be the size of CIRs. The same holds when estimating the AoA using the MUSIC algorithm. There, the number of snapshots N_s can be equal to the number of time points or the size of FFT. Furthermore, we only draw on the magnitude as the phase is noisy and requires noise reduction in order to be used as input, which is out of the scope of this dissertation.

Figure 4.5 depicts the accuracy of the NLoS-identifier based on SVM, a classical ML algorithm, and DNN, the method proposed in this work. As can be seen, the DNN-based method delivers higher accuracy, outperforming the classical method. Specifically, DNNs are more powerful regarding estimating the classifier function, and, therefore, they deliver superior performance. The performance remains high even if we employ the DNN in an environment other than that of the training

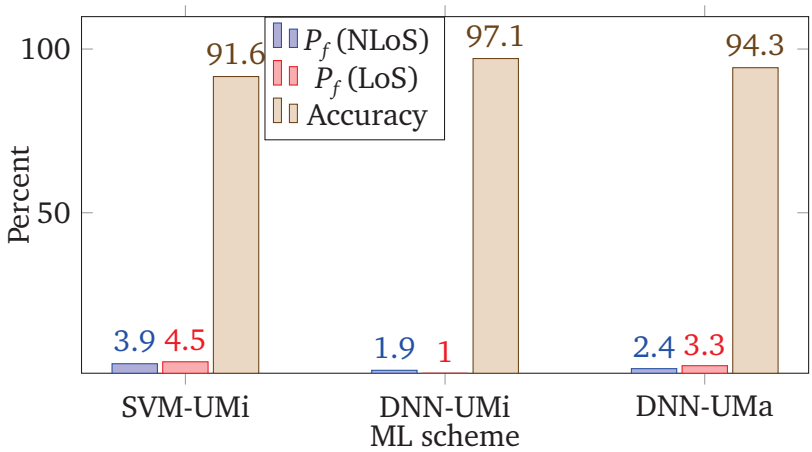


Figure 4.5: Comparison of two ML schemes when performing NLoS-identification. P_f(LoS)/P_f(NLoS) denotes the probability that the true condition of the links detected as LoS/NLoS is NLoS/LoS.

data, i.e., Urban Macro (UMa) cells instead of UMi. If the environment is too dissimilar, the performance will drastically deteriorate. In our simulations, we observed a poor accuracy of 61% for the rural-urban scenario, which is highly different from the UMi or UMa.

As mentioned before, the extremely high accuracy provided by the DNN is crucial as determining the link condition is among the most important decisions to be taken. In particular, false detection of NLoS links as LoS P_f(LoS), not only can result in a poor estimation of the MU position and clock parameters, but also may lead to divergence of the filter. This occurs since the AoA estimation as well as the time-based distance measurement (which in the case of this work is carried out through time-stamp exchange) are highly inaccurate for NLoS links.

4.4.2 AoA estimation

To evaluate the performance of the MUSIC algorithm, we arrange a specific simulation setup where an MU moves with the velocity of 2 m/s along the x axis from the point $[x = 0, y = 0, z = 1.5]$ until $[x = 70, y = 0, z = 1.5]$. An AP with a $N_{\text{ant}} \times N_{\text{ant}}$ UPA and tilted 20° is located at $[x = 35, y = -5, z = 10]$, equally distant from the two edges of the trajectory. Figure 4.6 depicts such a setup where the MU's trajectory and AP's coverage area (for 23 dBm power allocated to each antenna element) are observable. Furthermore, the elements are assumed to be patch antennas with 90° and 180° beam opening in the elevation and azimuth plane, respectively. Such a setup covers all angles that an MU might most likely have with respect to an AP, i.e., from 6° to 171°. Furthermore, it represents the

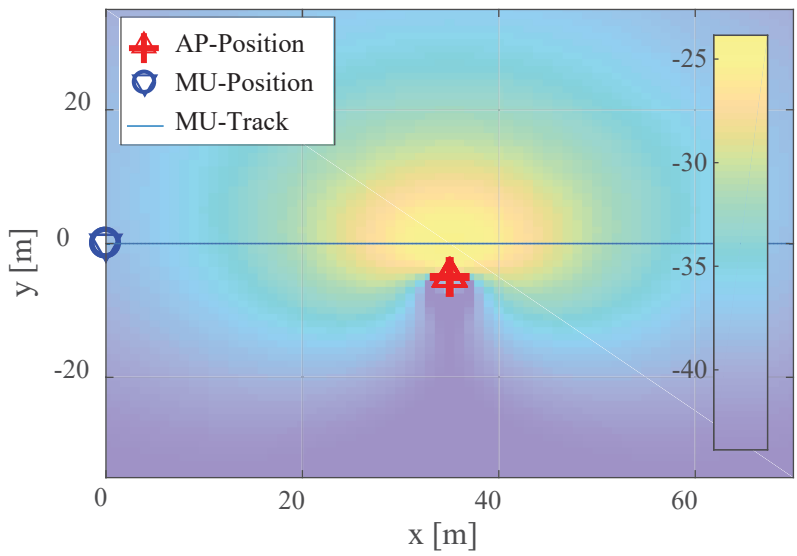


Figure 4.6: Simulation setup for calculating the AoA.

basic movement of the MUs in an urban scenario, e.g., the movement profile of the users shown in Figure 4.1 can be seen as the combination of that depicted in Figure 4.6. Lastly, at each time step, the AoA is estimated using the MUSIC algorithm fed with the corresponding CIR generated by QuaDRiGa. The algorithm estimates the azimuth and elevation AoA using the binary exhaustive search up to the 0.5 degree accuracy level, where the number of search bins are 40 and 20, respectively.

Figure 4.7 depicts the Root Mean Square Error (RMSE) of AoA estimation for several UPA sizes. As can be observed, the RMSE of azimuth AoA estimation remains under 1.5° for almost all investigated UPAs, which enables a precise localization of the MUs. Nevertheless, in our simulations, we observed that for smaller UPAs the RMSE increases drastically due to the large errors at the edges of the trajectory. Although such cases occur rarely, they can potentially lead to filter divergence. Moreover, the same behavior is observed for the elevation AoA estimation. Generally, as can be seen in the figure, the RMSE is slightly higher for the elevation AoA since the MU is always in the $[10° - 50°]$ angle sight of the AP. We know that UPAs' estimation performance deteriorates as we move towards the edges. In practice, due to the density of the APs, the MUs are expected to be in the azimuth angle range of $[20° - 150°]$, and elevation angle range of $[20° - 50°]$, i.e., AP density of less than 60 meters.

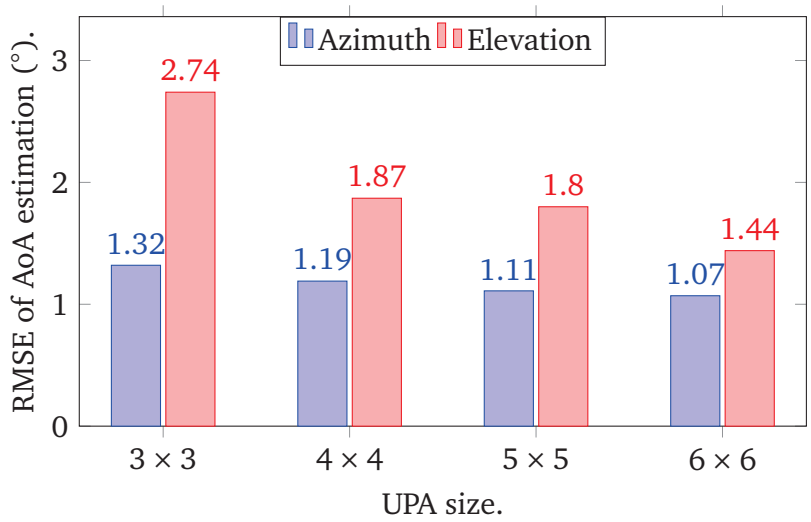


Figure 4.7: AoA estimation accuracy.

4.4.3 Joint Synchronization and Localization

To verify our proposed approach, we perform analysis for the scenario shown in Figure 4.1, which is regarded in [6], [9] as challenging. A car commences its journey by accelerating to reach the velocity of 14 m/s (= 50 km/h). It continues moving with constant velocity and decelerates upon approaching the intersection until it completely stops (e.g., due to the red light). The same repeats between two intersections. At the second intersection, it begins moving and takes a turn and continues to accelerate to 14 m/s limit until it exits the map. All turns as well as acceleration coefficients are chosen randomly. During its journey, at each joint sync&loc round k, the MU exchanges time-stamps with a fixed number of APs (N_{AP}) in ne(i), the link to each of which is LoS/NLoS with the probability of 0.8/0.2. The APs are grouped into ne(i) based on the distance criteria, that is, ne(i) includes the N_{AP} closest APs to the i-th MU. A further assumption is that at each joint sync&loc period T, $N_{\text{ant}} \times N_{\text{ant}}$ CIRs are available at each AP connected to the MU. In our simulations, the CIRs are obtained using the QuaDRiGa channel model. More explicitly, at each round k, knowing the true MU-AP distance and the link condition, i.e., LoS or NLoS, the CIRs are generated using the "3GPP_38.901_UMi" scenario of the QuaDRiGa channel model. Moreover, the RMSEs obtained by [6], [9] serve as the baseline to our approach. In particular, these works rely primarily on the ToA and AoA estimations fused with the aid of an EKF to compute the clock and position parameters. The second scheme with which we compare our proposed algorithm is the L-BRF proposed in [47], [48] and explained in Section 4.3.2. The aforementioned approaches are the most

Table 4.2: Simulation parameters

General parameters	Values
# of independent simulations	1000
Initial random delays ($\tilde{\theta}_i$)	$\mathscr{U}(-10^3, 10^3)$ ns
Initial random skew (γ_i)	$[0-100]$ ppm
Max. MU velocity	14 (m/s)
AP density	50 m
Distance traversed by the MU	600 m
QuaDRiGa parameters	
Scenario	3GPP_38.901_UMi
Center Frequency / FFT size (N_s)	3.8 GHz / 64
# of MU/AP antenna (N_{ant})	1 / 3×3
Filter Parameters	
Period of joint sync&loc (T)	100 ms
Process noise covariance matrix ($\mathbf{Q}_n$)	$\text{diag}(10^{-5}, 10, 1.5, 1.5)$
# of Gaussian mixtures (gdfs)	500
DNN parameters	
l_H, n_H	2, 64
Optimizer	Adam
# of epochs	10
Batch size	16
Activation function of hidden layers	ReLU
Activation function of output layer	Softmax

relevant as they use the same inputs as our proposed method does. We begin our analysis with L-BRF and then compare DePF and L-BRF with [6], [9].

We initialize all the clock offsets from the uniform distribution $\mathscr{U}(-10^3, 10^3)$ ns. The initial skews of all clocks are drawn from the uniform distribution $\mathscr{U}(1-10^{-4}, 1+10^{-4})$, which corresponds to skew values between 0 and 100 *part-per-million* (ppm). The covariance of the clock process noise $\mathbf{Q}_n(\tilde{\boldsymbol{\vartheta}}_i)$ is set to $\text{diag}(10^{-5}, 100)$ to account for the residual errors from the previous iterations as well as the external noises on the clock skew and offset. The covariance of position process noise $\mathbf{Q}_n(\mathbf{p}_i)$ amounts to $\text{diag}((14T)^2, (14T)^2)$ to account for every possible movement of the MU. All additional simulation parameters can be found in Table 4.2.

4.4.3.1 Analysis of L-BRF Joint sync&loc

Figure 4.8 shows the CDF of the clock offset and position estimation error for different time-stamping error mean μ_T (or alternatively μ_R) and $\sigma_T = 1ns$. Note that we use the truncated distribution in the simulations as r_i/r_j and t_i/t_j cannot

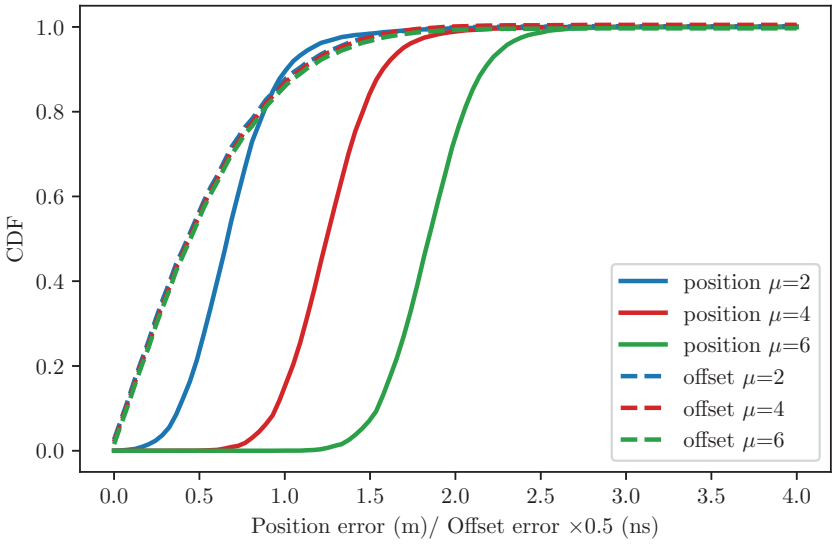


Figure 4.8: Performance of joint sync&loc algorithm ($\sigma_T = 1$ns).

be negative. As can be observed, the error of the position estimation increases as the mean delay of time-stamping grows. However, the clock offset estimation error does not show any increase. The reason is disguised in Equations (4.24) and (4.25), which are obtained by subtracting Equation (3.3) from Equation (3.2) and adding Equation (3.3) to Equation (3.4). As can be seen, the mean of the time-stamping error is cancelled out in the terms $T_{ij}^{k,1} - T_{ij}^{k,0}$ and $T_{ij}^{k,1} - R_{ij}^{k}$. This is not the case for the position estimation as the position parameters (embedded in d_{ij}) will always depend on R_{ij}^{k} no matter how the time-stamp equations are simplified. In practice, however, we can calibrate out the mean of time-stamping error, thereby keeping the position estimation error low. Thus, the main source of error in offset and position estimation becomes the uncertainty in time-stamping. In the following, we analyse the impact of time-stamp uncertainty on the position and clock offset estimation error.

Figure 4.9 shows the CDF of the clock offset and position estimation error for different time-stamping error standard deviation σ_T (or alternatively σ_R) and $\mu_T = \mu_R = 0$. As can be observed, the error for both position and offset estimation increases as the uncertainty of time-stamping grows. In particular, high uncertainty in time-stamping leads to a larger error when estimating the clock skew and offset. Since the offset error translates to an error in position estimation, we observe a degradation in the position estimation as well. According to the figure, sub-meter and sub-ns estimation accuracies require the time-stamping to be less than 2 ns. Typically, new commercial off-the-shelf devices are capable of performing time-stamping with the uncertainty less than 5ns.

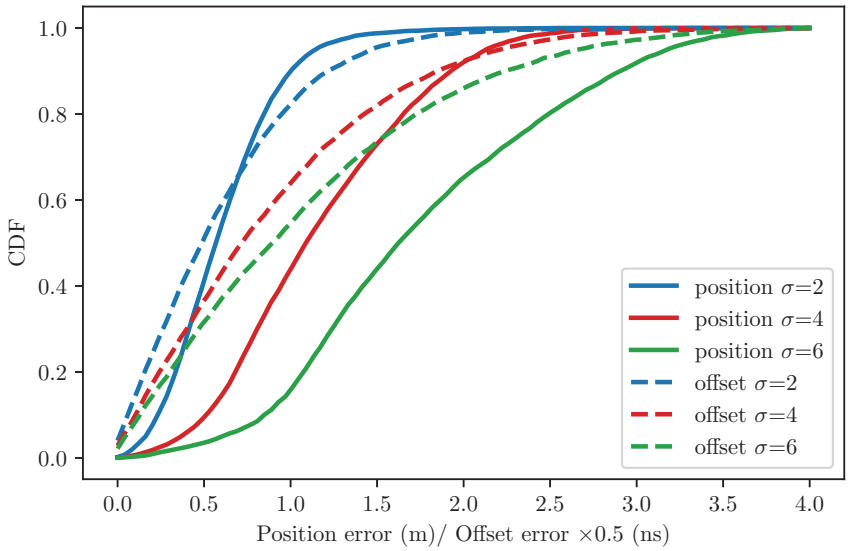


Figure 4.9: Performance of joint sync&loc algorithm for different time-stamp uncertainties ($\mu_T = 0ns$).

Figure 4.10 presents the CDF of position and clock offset estimation versus the number of employed APs for $\sigma_T = 2$ for $\mu_T = \mu_R = 0$ ns. It can be noticed that the position and clock offset estimation error drop with the increase of the number of APs. In particular, given that the APs are synchronized, the growth in the number of APs serving the MUs increases the number of collected time-stamps providing more information about the clock offset and AP-MU distance estimation. More APs not only mitigates the impact of time-stamping uncertainty, but also helps in calibrating out μ_T. Nevertheless, the gain obtained from adding the third AP is clearly less significant than that from the second one. The reason is that the third AP is far from the MU, which results in a less informative MU-AP time-stamp exchange and, therefore, does not significantly improve the accuracy of estimation. We can observe that the localization with two APs ensures sub-meter accuracy.

Considering all Figures, we can remark that while the uncertainty in time-stamping, i.e., σ_T and σ_R, can be mitigated using BRF (especially for the 1-AP case), the delay in time-stamping, i.e., μ_T and μ_R, can only be mitigated by either employing multiple APs, or calibrating out, or improving the hardware deployed for time-stamping. In particular, according to Figures 4.9 and 4.10, for sub-meter accuracy localization with a single AP, the time-stamping mechanism should be designed such that σ_T is kept below 2ns.

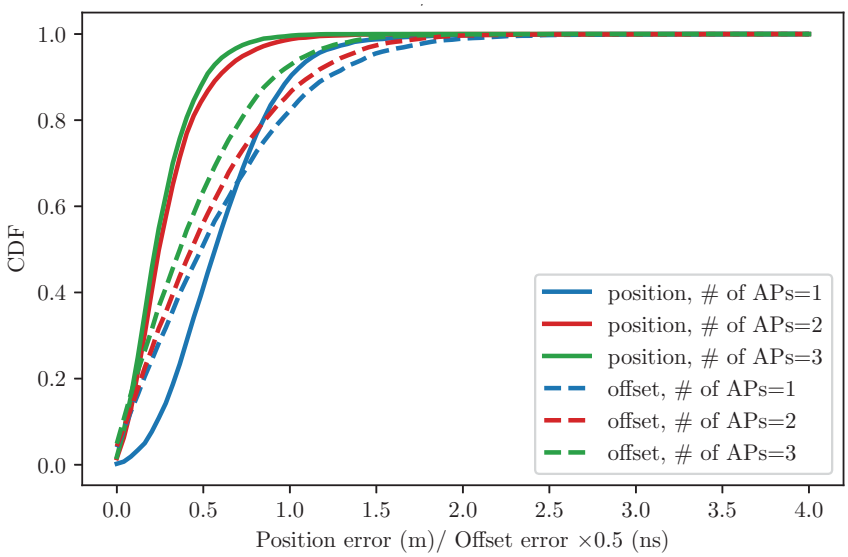


Figure 4.10: Performance of L-BRF joint sync&loc algorithm. $\sigma_T = 2$ns, $\mu_T = 0$ns.

4.4.3.2 Analysis of DePF Joint sync&loc

Figure 4.11 shows the RMSE of the clock offset estimation error for three joint snyc&loc algorithms. The DePF algorithm is compared with two linear Bayesian methods, i.e., EKF [9] and L-BRF [48], in multiple scenarios. In particular, we compute the RMSEs in three scenarios, with the number of LoS APs ranging from 1 to 3. In another additional scenario, we consider the MU being connected to three APs, where each MU-AP link condition is set to LoS with the probability of 0.8. As can be seen, for all scenarios, the L-BRF and DePF deliver an identical performance, what is expected as they rely on the same approach to estimate the clock parameters. On the other hand, the performance of the EKF falls behind as it does not explicitly utilize the synchronization signals to estimate the clock offset. Moreover, the synchronization algorithm scheme utilized to synchronize the APs, i.e., hybrid BP-BRF network synchronization, leads to a more precise inter-AP synchronization and consequently lower MU clock offset estimation error.

Figure 4.12 depicts the RMSE of position estimation error for three joint snyc&loc algorithms. The DePF algorithm is compared with two linear Bayesian methods, i.e., EKF and L-BRF, in the same scenarios as in 4.11. As can be seen, for almost all scenarios, the DePF algorithm delivers superior performance. In particular, since the DePF uses a higher number of gdfs, rather than only one, to represent the posterior distribution, it can estimate the position more accurately. Furthermore, DePF stands out when dealing with NLoS links. This is straightforward to notice as the RMSE of position estimation is lower for DePF in

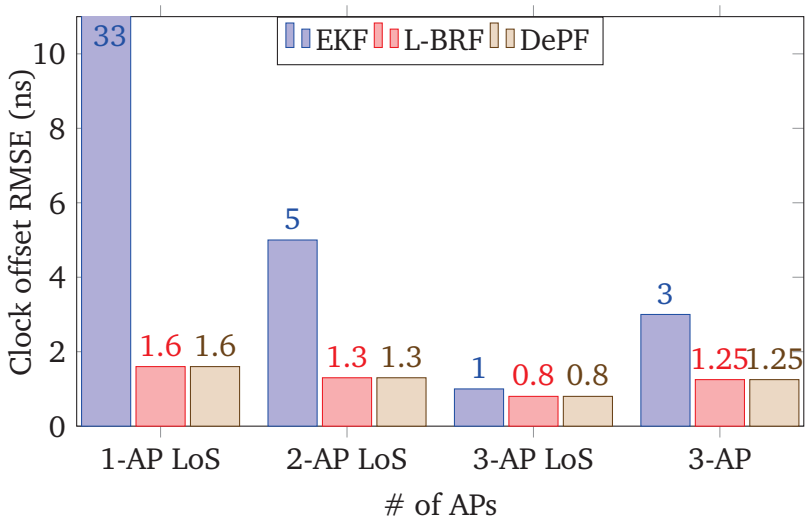


Figure 4.11: Performance comparison of three joint synchronization and localization algorithms in terms of clock offset estimation.

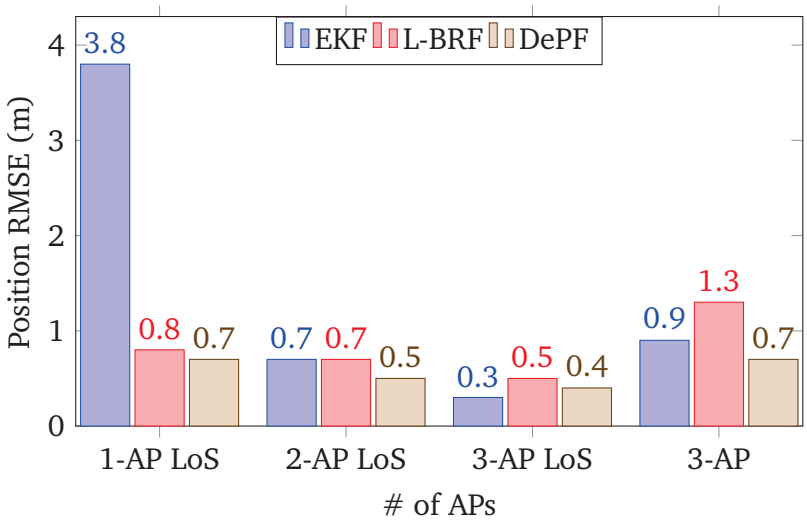


Figure 4.12: Performance comparison of three joint synchronization and localization algorithms in terms of position estimation.

the 3-AP scenario where the L-BRF employs the same NLoS identifier as DePF. Additionally, unlike EKF and L-BRF, DePF does not need any initialization, which is of crucial importance in practice as initialization would require the APs to request position estimation from the MUs, what may not be always possible. Overall, considering 2-AP LoS, 3-AP LoS, and 3-AP scenarios, the linear Bayesian approach and DePF give similar performance when a reliable initialization and MU-AP links with know LoS condition are available. Nevertheless, such requirements are hard to guarantee in practice, rendering EKF and L-BRF algorithms futile in real-world scenarios.

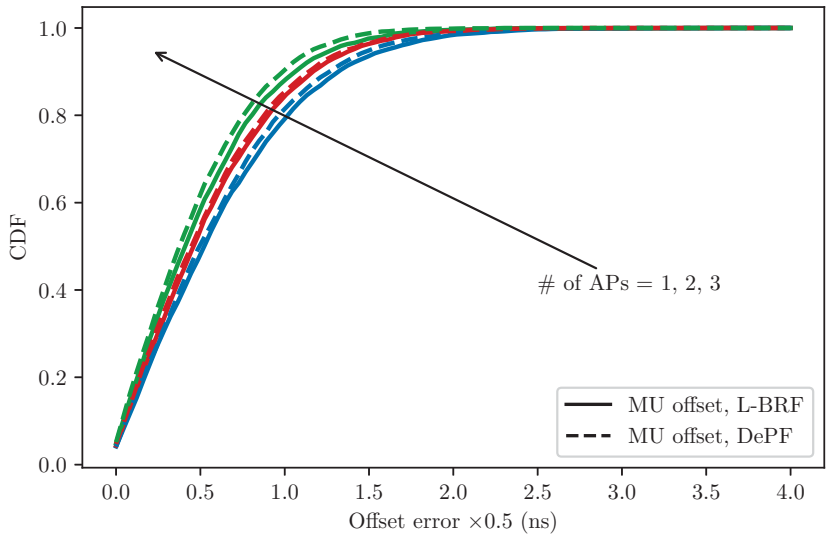


Figure 4.13: Performance comparison of L-BRF and DePF when estimating the MUs' clock offset.

Hereafter, all simulations have been carried out assuming that there is always at least one LoS MU-AP link. Figure 4.13 presents the CDF of the clock offset estimation error when the MU is connected to multiple number of APs. It can be seen that the estimation accuracy increases as both L-BRF and DePF utilize more measurements to estimate the clock offset and skew. In fact, since the APs are synchronized with a high precision, collecting time-stamps from each additional AP does provide additional information about the statistics of MU's clock parameters and, therefore, increases the accuracy of the estimation. Furthermore, the performances of both schemes are identical as they deploy the same approach to estimate the clock parameters. That is, both model the clock parameters with a single Gaussian function.

Figure 4.14 presents the CDF of the position estimation error when the MU is connected to multiple number of APs. As can be seen, the position estimation error is less than 1 meter in 90% of the cases for the DePF algorithm. We observe that DePF significantly outperforms the L-BRF. In particular, unlike the L-BRF which approximates the posterior with a single Gaussian distribution, in DePF, the approximation is based on multiple gdfs. Consequently, the approximated posterior is closer to the true one, resulting in a more precise position estimation. Another subtle observation is that although the position estimation error decays with the growth in the number of APs, increasing the number of APs from 2 to 3 only slightly improves the performance. In fact, the third AP is far away from the MU, leading to a poorer (AoA and time-stamp) measurement accuracy com-

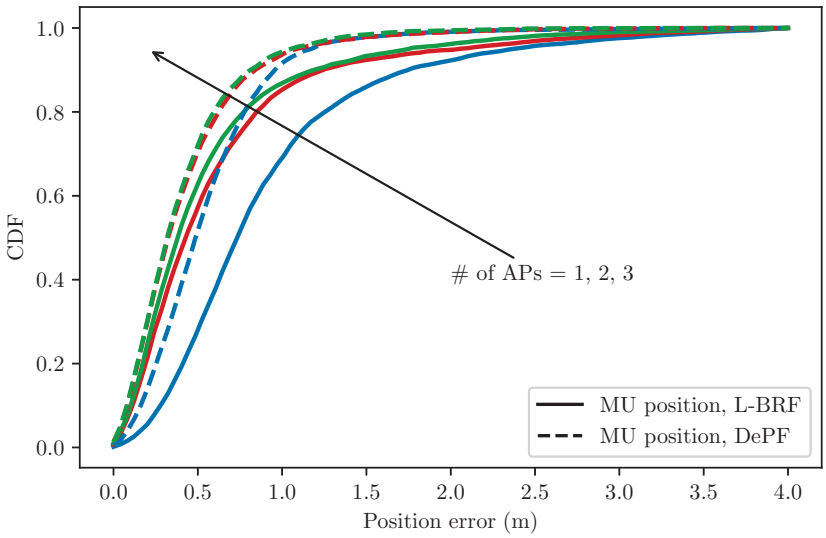


Figure 4.14: Performance comparison of L-BRF and DePF when estimating the MUs' position.

pared to that of the first two APs. Hence, it does not provide substantial further information about the posterior distribution of the MU's location. Furthermore, the performance difference between DePF and L-BRF is more pronounced for 1-AP indicating that DePF is more successful when we have a limited number of measurements.

Figure 4.15 indicates the CDF of position estimation for multiple number of gdfs. It can be noticed that the position estimation improves with the increase of the number of gdfs. This is expected as in PGMFs, the posterior distribution is approximated by multiple gdfs. Consequently, the more gdfs we employ, the more accuracy we achieve, albeit with higher computation time, which linearly increases with respect to the number of gdfs F (Table 4.1). A trade-off can be struck between the computational complexity and the accuracy by determining the number of gdfs. In particular, we can see from the figure that the improvement in accuracy is negligible when increasing F from 200 to 500.

Figure 4.16 shows the CDF of clock offset estimation error for a single AP for different time-stamp uncertainties, i.e., $\sigma_T = 2, 4, 6$. As can be seen, the clock offset estimation accuracy drops as the σ_T grows. It remains, however, less than 6 ns in 90% of the cases. Such degradation can cause an additional error in position estimation as both parameters are intertwined given (3.4). Nevertheless, the uncertainties of the time-stamping of the state-of-the-art devices are expected to be below 5 ns. Moreover, the destructive impact of the uncertainty can be also mitigated by employing more APs as discussed previously and shown in

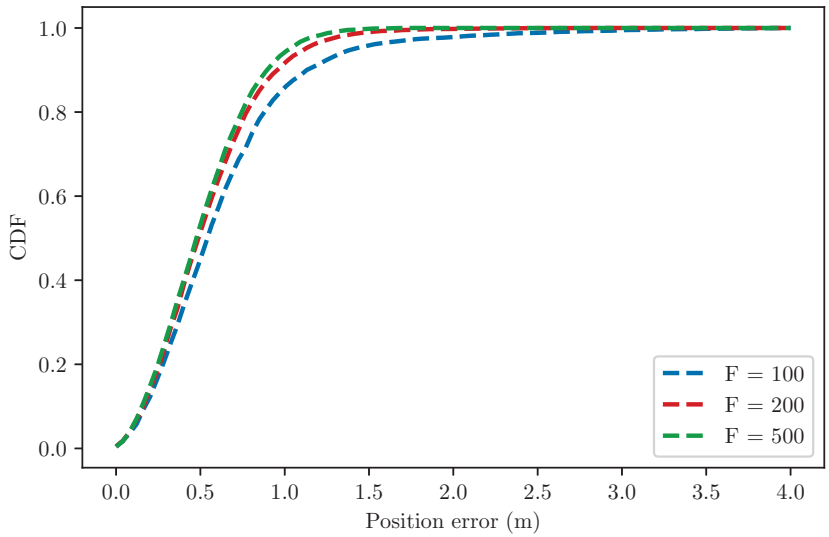


Figure 4.15: Performance of joint sync&loc algorithm for different number of gdfs.

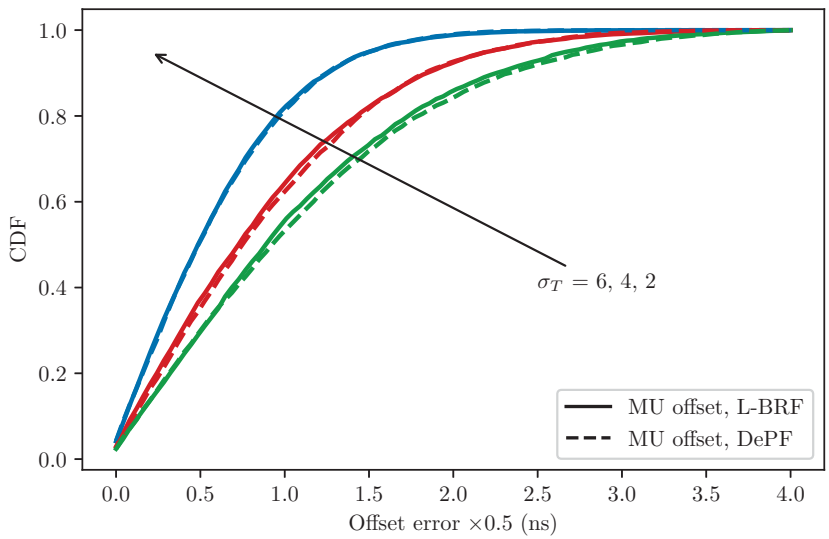


Figure 4.16: Clock offset estimation performance of joint sync&loc algorithm with different number of APs involved.

Figure 4.13. Again, the same performance L-BRF and DePF is due to the fact that they use the same estimation approach for the clock parameter estimation.

Figure 4.17 shows the CDF of position estimation conducted by a single AP for different time-stamp accuracy. It can be noticed that the position estimation accuracy deteriorates with the growth in the time-stamp uncertainty. Specifically, the growth in uncertainty results in more erroneous distance measurements and offset estimation that, consequently, worsens the position estimation accuracy. Nevertheless, it can be readily seen that DePF is more successful in mitigating

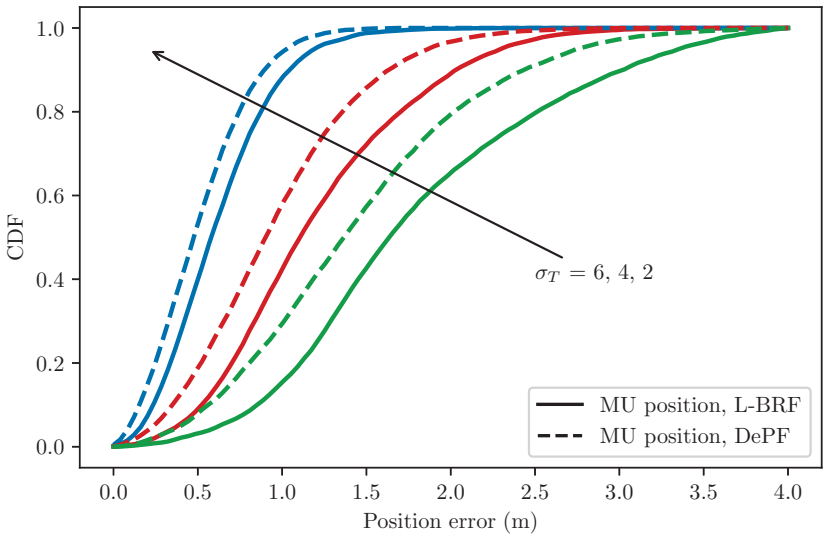


Figure 4.17: Position estimation performance of joint sync&loc algorithm with different time-stamp accuracy.

the destructive effect of the time-stamp uncertainty on the position estimation. Moreover, as seen in Figure 4.13 and 4.14, for both DePF and L-BRF, employing more APs can alleviate the negative impact of large time-stamping uncertainty on both offset and position estimation.

To summarize the discussion on L-BRF and DePF, from the Figures 4.11, 4.13, and 4.16 we can conclude that both L-BRF and DePF deliver the same performance in terms offset estimation accuracy. The estimation accuracy is highly dependent on the time-stamp uncertainty and the number of APs serving the MU. Furthermore, from Figures 4.12, 4.14, and 4.17, we can observe that generally DePF outperforms L-BRF in terms of position estimation accuracy. The gap between their performances is more pronounced in the presence of limiting factors such as low number of serving APs and inaccurate measurements.

4.5 Summary

We presented two joint sync&loc techniques for MUs in wireless communication networks (summarized in Table 4.3). These algorithms are primarily based on the hybrid network synchronization developed in Chapter 3. The L-BRF was adopted by linearizing the measurement equations along with the BRF filter. Moreover, we introduced the DePF algorithm, which relies on PGMFs functioning in a hybrid parametric and particle-based manner, to obtain the posterior distribution of the clock and position parameters, thereby estimating the

clock offset and position of MUs. An extensive simulation campaign was carried out and comprehensive results were presented. It was shown that, firstly, an accurate NLoS identification is essential if a successful and robust joint sync&loc algorithm is to be developed. Furthermore, we indicated that DePF is generally more resilient to deal with real-word limiting factors such as lack of MU's initial position, erroneous measurements, and uncertainty in time-stamping.

Joint sync Algorithm	Corresponding PGM	Strengths	Weaknesses
Linearized BRF (L-BRF)	Dynamic Bayesian Networks (DBNs)	Low complexity	Low accuracy with low number of access points, requires initialization
DNN-assisted Particle-based (DePF)	DBNs	High accuracy even with low number of access points	High complexity

Table 4.3: Summary of the joint sync&loc algorithms presented in this chapter.

Chapter 5

Conclusions and Future Works

We explained that a wide variety of services in communication networks are only deliverable when Mobile Users' (MUs) locations are known. To localize the MUs, the Access Points (APs) rely mostly on AP-MU time measurements to cooperatively estimate the position. For such methods to function, APs are required to be synchronized among each other as well as with the MUs. In this dissertation, after introducing the preliminary statistical tools, i.e., Probabilistic Graphical Models (PGMs), in Chapter 2, we firstly addressed the inter-AP synchronization in the framework of network synchronization in Chapter 3. Subsequently, in Chapter 4, we tackled the MU joint synchronization and localization (sync&loc) problem. In what follows, we summarize the contributions of these two chapters and point to the future possible research directions.

5.1 Conclusions

In this section, we highlight the main contributions made by this dissertation. Specifically, we briefly mention the main results of Chapters 3 and 4, which dealt with network synchronization and joint synchronization and localization, respectively.

Synchronization: In Chapter 3, we presented two Bayesian approaches toward clock offset and skew estimation in communication networks. In particular, Belief Propagation (BP) was employed to perform high-precision network-wide synchronization, albeit at the cost of a high number of time-stamp exchanges and message passing iterations. Additionally, Bayesian Recursive Filtering (BRF) was leveraged to carry out pairwise synchronization, delivering a superb performance at the edge of the network. Based on these two algorithms, a hybrid Bayesian approach was proposed to not only fulfill a low relative time error at a local level

but also to maintain a high synchronization accuracy at a global level. Simulation results show that the proposed hybrid approach achieves faster and more frequent synchronization at the cost of only a slight deterioration in performance.

MU joint sync&loc: In Chapter 4, we presented two schemes for joint sync&loc of MUs in communication networks, a) Linearized BRF (L-BRF), and b) DNN-assisted Particle-based filtering (DePF). In particular, we leveraged an asymmetric time-stamp exchange mechanism, traditionally utilized for time synchronization, to estimate the clock offset and skew while simultaneously obtaining information about AP-MU distance. Further on, with the aid of L-BRF and DePF, we combined the aforementioned information with angle of arrival estimation and the link condition, i.e., line-of-sight or non-line-of-sight, to localize the MUs. Simulation results indicate that while the performance of the proposed algorithm is promising and can outperform the state-of-the-art algorithms, the position and clock offset estimation errors are highly dependent on the delay in hardware time-stamping as well as its accuracy. The negative impact of this dependency can be mitigated by improving the hardware time-stamping mechanism and deploying more APs for performing MU joint sync&loc. In addition, it can be noticed that generally DeBRF outperforms L-BRF in terms of position estimation accuracy. The gap between their performances is more noticeable when there are limiting factors such as low number of serving APs and inaccurate measurements.

5.2 Future Works

In the following, we provide a number of possible future directions. In particular, the research problems arising from the contributions of this dissertation are mentioned and briefly explained.

Synchronization: Given the promising simulation results, a possible future direction can be the implementation of the hybrid synchronization algorithm presented in this work using commercial-of-the-shelf hardware. Pursuing such a direction can reveal, on the one hand, the limitations caused by the hardware components, and, on the other hand, the challenges we face when implementing message passing in practice. The former would determine how accurately the time-stamping can be implemented and, therefore, reveals the performance boundary in both network synchronization and MU joint sync&loc. The latter would indicate how feasible BP is at the network level.

MU Joint sync&loc: A particularly interesting research problem is to cross-validate the obtained simulation results. Specifically, the impact of the time-stamp exchange on the accuracy of joint sync&loc can be investigated. Later on, such scheme can be combined with the previous implementation of hybrid synchronization to build up an end-to-end demonstration. Furthermore, it can reveal the challenges for implementing the end-to-end model such as time-stamp exchange, collection of time-stamps in a single central unit responsible for performing the joint sync&loc algorithm, and the time efficiency of the algorithm.

Another compelling track to follow is to design the NLoS-identification and AoA estimation based only on Received Signal Strength (RSS). This may be favorable to some of the currently in operation devices without the CIR estimation capability. In particular, RSS is readily available in all the communication devices and is periodically computed when exchanging control signals. Therefore, an NLoS-identification scheme based on RSS can be more ubiquitously employed.

Finally, although the evaluation scenarios were mostly outdoor, the joint sync&loc algorithms presented in this dissertation can be applied to the indoor environment as well. Therefore, the application of L-BRF and DePF in an indoor environment, both in simulation and practice, is another possible trajectory that can be followed. The challenge in indoor environments is that typically there is only one AP with LoS to the MU due to the distribution of the APs (one per room or hall). In such an environment, highly precise NLoS identification is of crucial importance.

Acknowledgement

First and foremost, I would like to offer my sincerest gratitude to my supervisor, Dr. Eckhard Grass, who supported me throughout my thesis with his patience and knowledge while allowing me to work on my own way. Without him, this thesis would not have been completed or written. One simply could not wish for a better and friendlier supervisor

I would like to greatly thank the members of my doctoral committee, Professor Jens-Peter Redlich, Professor Giuseppe Caire, and Professor Milos Krstic, Professor Verena Hafner, and Professor Thorben Wübbenhorst for their time, their feedback, and their interest in my work.

Besides my supervisor and the doctoral committee, I would like to thank my colleagues Mr. Nebojsa Maletic and Mr. Jesus Gutierrez for their enlightening explanations, keen scientific insight, and valuable time they spent in helping me. The publications underpinning this dissertation would have not been successful without their valuable help.

Many thanks to my friends, Farshad Riyahi, Christopher Fritz, Margherita Lazzeri, and Ali Mansuri Torshizi for standing by my side during the tough times, for their support and empathy, and for our insightful philosophical discussions on the Ph.D. path.

Finally, special thanks to my family. Words cannot express how grateful I am to my mother and father for their support and love in my whole life. Beside them, I am deeply thankful to my brother, Mohsen Goodarzi, who kind-heatedly supported me remotely throughout these years.

References

[1] A. Kaloxylos, A. Gavras und R. De Peppe, *Empowering Vertical Industries through 5G Networks - Current Status and Future Trends*, Aug. 2020. DOI: 10.5281/zenodo.3698113. Adresse: https://doi.org/10.5281/zenodo.3698113.

[2] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson und H. Wymeersch, "Location-aware communications for 5G networks: How location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Processing Magazine*, Jg. 31, Nr. 6, S. 102–112, 2014.

[3] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia und D. Moore, "Environmental wireless sensor networks," *Proceedings of the IEEE*, Jg. 98, Nr. 11, S. 1903–1917, 2010.

[4] Y.-C. Wu, Q. Chaudhari und E. Serpedin, "Clock synchronization of wireless sensor networks," *IEEE Signal Processing Magazine*, Jg. 28, Nr. 1, S. 124–138, 2010.

[5] N. Maletic, V. Sark, J. Gutiérrez und E. Grass, "Device localization using mmWave ranging with sub-6-assisted angle of arrival estimation," in *2018 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, IEEE, 2018, S. 1–6.

[6] J. Werner, M. Costa, A. Hakkarainen, K. Leppanen und M. Valkama, "Joint user node positioning and clock offset estimation in 5G ultra-dense networks," in *2015 IEEE Global Communications Conference (GLOBECOM)*, IEEE, 2015, S. 1–7.

[7] "IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, S. 1–3534, 2016.

[8] B. Etzlinger, F. Meyer, F. Hlawatsch, A. Springer und H. Wymeersch, “Cooperative simultaneous localization and synchronization in mobile agent networks,” *IEEE Transactions on Signal Processing*, Jg. 65, Nr. 14, S. 3587–3602, 2017.

[9] M. Koivisto, M. Costa, J. Werner u. a., “Joint device positioning and clock synchronization in 5G ultra-dense networks,” *IEEE Transactions on Wireless Communications*, Jg. 16, Nr. 5, S. 2866–2881, 2017.

[10] N. Maletic, V. Sark, M. Ehrig, J. Gutiérrez und E. Grass, “Experimental Evaluation of Round-Trip ToF-based Localization in the 60 GHz Band,” in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, S. 1–6.

[11] I.-K. Rhee, J. Lee, J. Kim, E. Serpedin und Y.-C. Wu, “Clock synchronization in wireless sensor networks: An overview,” *Sensors*, Jg. 9, Nr. 1, S. 56–85, 2009.

[12] H. Li, L. Han, R. Duan und G. M. Garner, “Analysis of the synchronization requirements of 5G and corresponding solutions,” *IEEE Communications Standards Magazine*, Jg. 1, Nr. 1, S. 52–58, 2017.

[13] M. Lévesque und D. Tipper, “A survey of clock synchronization over packet-switched networks,” *IEEE Communications Surveys & Tutorials*, Jg. 18, Nr. 4, S. 2926–2947, 2016.

[14] F. Meyer, B. Etzlinger, Z. Liu, F. Hlawatsch und M. Z. Win, “A scalable algorithm for network localization and synchronization,” *IEEE Internet of Things Journal*, Jg. 5, Nr. 6, S. 4714–4727, 2018.

[15] G. Dedes und A. G. Dempster, “Indoor GPS positioning-challenges and opportunities,” in *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.*, Citeseer, Bd. 1, 2005, S. 412–415.

[16] “IEEE Standard for Information technology -Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)*, S. 1–3534, 2016.

[17] J. Eidson und K. Lee, “IEEE 1588 standard for a precision clock synchronization protocol for networked measurement and control systems,” in *Sensors for Industry Conference, 2002. 2nd ISA/IEEE*, Ieee, 2002, S. 98–105.

[18] G. Giorgi und C. Narduzzi, “Performance analysis of Kalman-filter-based clock synchronization in IEEE 1588 networks,” *IEEE Transactions on Instrumentation and Measurement*, Jg. 60, Nr. 8, S. 2902–2909, 2011.

[19] M. Leng und Y.-C. Wu, “Low-complexity maximum-likelihood estimator for clock synchronization of wireless sensor nodes under exponential delays,” *IEEE Transactions on Signal Processing*, Jg. 59, Nr. 10, S. 4860–4870, 2011.

[20] B. Lv, Y. Huang, T. Li u. a., “Simulation and performance analysis of the IEEE 1588 PTP with kalman filtering in multi-hop wireless sensor networks,” *Journal of networks*, Jg. 9, Nr. 12, S. 3445, 2014.

[21] M. Leng und Y.-C. Wu, “Distributed clock synchronization for wireless sensor networks using belief propagation,” *IEEE Transactions on Signal Processing*, Jg. 59, Nr. 11, S. 5404–5414, 2011.

[22] K. J. Zou, K. W. Yang, M. Wang u. a., “Network synchronization for dense small cell networks,” *IEEE Wireless Communications*, Jg. 22, Nr. 2, S. 108–117, 2015.

[23] B. Etzlinger, H. Wymeersch und A. Springer, “Cooperative synchronization in wireless networks,” *IEEE Transactions on Signal Processing*, Jg. 62, Nr. 11, S. 2837–2849, 2014.

[24] J. Du und Y.-C. Wu, “Distributed clock skew and offset estimation in wireless sensor networks: Asynchronous algorithm and convergence analysis,” *IEEE Transactions on Wireless Communications*, Jg. 12, Nr. 11, S. 5908–5917, 2013.

[25] S. Zhu und Z. Ding, “Joint synchronization and localization using TOAs: A linearization based WLS solution,” *IEEE Journal On Selected areas in communications*, Jg. 28, Nr. 7, S. 1017–1025, 2010.

[26] J. Zheng und Y.-C. Wu, “Joint time synchronization and localization of an unknown node in wireless sensor networks,” *IEEE Transactions on Signal Processing*, Jg. 58, Nr. 3, S. 1309–1320, 2009.

[27] A. Ahmad, E. Serpedin, H. Nounou und M. Nounou, “Joint node localization and time-varying clock synchronization in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, Jg. 12, Nr. 10, S. 5322–5333, 2013.

[28] R. M. Vaghefi und R. M. Buehrer, “Cooperative joint synchronization and localization in wireless sensor networks,” *IEEE Transactions on Signal Processing*, Jg. 63, Nr. 14, S. 3615–3627, 2015.

[29] A. F. Molisch, V. V. Ratnam, S. Han u. a., “Hybrid beamforming for massive MIMO: A survey,” *IEEE Communications magazine*, Jg. 55, Nr. 9, S. 134–141, 2017.

[30] J. Mietzner, R. Schober, L. Lampe, W. H. Gerstacker und P. A. Hoeher, “Multiple-antenna techniques for wireless communications-a comprehensive literature survey,” *IEEE communications surveys & tutorials*, Jg. 11, Nr. 2, S. 87–105, 2009.

[31] M. Goodarzi, A. Krishnamoorthy, R. Schober und M. Breiling, “Resource allocation for outdoor-to-indoor amplify-and-forward sudas with independent relay processing,” in *SCC 2019; 12th International ITG Conference on Systems, Communications and Coding*, VDE, 2019, S. 1–6.

[32] A. Krishnamoorthy, R. Schober und M. Breiling, “Resource allocation for outdoor-to-indoor compress-and-forward SUDAS with independent relay processing,” in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, IEEE, 2017, S. 1–7.

[33] A. Decurninge, L. G. Ordóñez, P. Ferrand u. a., “CSI-based outdoor localization for massive MIMO: Experiments with a learning approach,” in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, IEEE, 2018, S. 1–6.

[34] X. Wang, L. Gao, S. Mao und S. Pandey, “CSI-based fingerprinting for indoor localization: A deep learning approach,” *IEEE Transactions on Vehicular Technology*, Jg. 66, Nr. 1, S. 763–776, 2016.

[35] Y.-J. Lin, P.-H. Tseng, Y.-C. Chan, J. He und G.-S. Wu, “A super-resolution-assisted fingerprinting method based on channel impulse response measurement for indoor positioning,” *IEEE Transactions on Mobile Computing*, Jg. 18, Nr. 12, S. 2740–2753, 2018.

[36] F. Xiao, Z. Guo, H. Zhu, X. Xie und R. Wang, “AmpN: Real-time LOS/NLOS identification with WiFi,” in *2017 IEEE International Conference on Communications (ICC)*, IEEE, 2017, S. 1–7.

[37] K. Yu und Y. J. Guo, “Statistical NLOS identification based on AOA, TOA, and signal strength,” *IEEE Transactions on Vehicular Technology*, Jg. 58, Nr. 1, S. 274–286, 2008.

[38] P. Gupta und S. Kar, “MUSIC and improved MUSIC algorithm to estimate direction of arrival,” in *2015 International Conference on Communications and Signal Processing (ICCSP)*, IEEE, 2015, S. 0757–0761.

[39] B. Ottersten, M. Viberg und T. Kailath, “Performance analysis of the total least squares ESPRIT algorithm,” *IEEE transactions on signal processing*, Jg. 39, Nr. 5, S. 1122–1135, 1991.

[40] S. Barua, S. C. Lam, P. Ghosa, S. Xing und K. Sandrasegaran, “A survey of direction of arrival estimation techniques and implementation of channel estimation based on SCME,” in *2015 12th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, IEEE, 2015, S. 1–5.

[41] M. C. Vanderveen, A.-J. Van der Veen und A. Paulraj, “Estimation of multipath parameters in wireless communications,” *IEEE Transactions on Signal Processing*, Jg. 46, Nr. 3, S. 682–690, 1998.

[42] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Prentice-Hall, Inc., 1993.

[43] D. Koller, N. Friedman, L. Getoor und B. Taskar, “Graphical models in a nutshell,” *Introduction to statistical relational learning*, Jg. 43, 2007.

[44] A. B. Dieng, *Deep Probabilistic Graphical Modeling*. Columbia University, 2020.

[45] M. Goodarzi, D. Cvetkovski, N. Maletic, J. Gutiérrez und E. Grass, “Synchronization in 5G: a bayesian approach,” in *2020 European Conference on Networks and Communications (EuCNC)*, IEEE, 2020, S. 194–199.

[46] ——, “A hybrid bayesian approach towards clock offset and skew estimation in 5G networks,” in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, IEEE, 2020, S. 1–7.

[47] ——, “Synchronization in 5G networks: a hybrid Bayesian approach toward clock offset/skew estimation and its impact on localization,” *EURASIP Journal on Wireless Communications and Networking*, Jg. 2021, Nr. 1, S. 1–22, 2021.

[48] M. Goodarzi, N. Maletic, J. Gutiérrez und E. Grass, “Bayesian Joint Synchronization and Localization Based on Asymmetric Time-stamp Exchange,” in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*, IEEE, 2020, S. 1–7.

[49] M. Goodarzi, V. Sark, N. Maletic, J. Gutierrez, G. Caire und E. Grass, "DNN-assisted Particle-based Bayesian Joint Synchronization and Localization," *IEEE Transactions on Communications*, S. 1–1, 2022. DOI: 10.1109/TCOMM.2022.3180069.

[50] P. Larrañaga und S. Moral, "Probabilistic graphical models in artificial intelligence," *Applied soft computing*, Jg. 11, Nr. 2, S. 1511–1528, 2011.

[51] L. E. Sucar, "Probabilistic graphical models," *Advances in Computer Vision and Pattern Recognition. London: Springer London. doi*, Jg. 10, Nr. 978, S. 1, 2015.

[52] D. Koller und N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[53] R. Durrett, *Probability: theory and examples*. Cambridge university press, 2019, Bd. 49.

[54] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[55] V. Mihajlovic und M. Petkovic, "Dynamic bayesian networks: A state of the art," *University of Twente Document Repository*, 2001.

[56] V. I. Pavlovic, *Dynamic bayesian networks for information fusion with applications to human-computer interfaces*. University of Illinois at Urbana-Champaign, 1999, Bd. 44.

[57] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[58] G. Casella und R. L. Berger, *Statistical inference*. Cengage Learning, 2021.

[59] A. L. Barker, D. E. Brown und W. N. Martin, "Bayesian estimation and the Kalman filter," *Computers & Mathematics with Applications*, Jg. 30, Nr. 10, S. 55–77, 1995.

[60] J. S. Yedidia, W. T. Freeman und Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on information theory*, Jg. 51, Nr. 7, S. 2282–2312, 2005.

[61] K. Murphy, Y. Weiss und M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," *arXiv preprint arXiv:1301.6725*, 2013.

[62] A. T. Ihler, J. W. Fisher III, A. S. Willsky und D. M. Chickering, "Loopy belief propagation: convergence and effects of message errors.," *Journal of Machine Learning Research*, Jg. 6, Nr. 5, 2005.

[63] R. J. McEliece, D. J. C. MacKay und J.-F. Cheng, “Turbo decoding as an instance of Pearl’s" belief propagation" algorithm,” *IEEE Journal on selected areas in communications*, Jg. 16, Nr. 2, S. 140–152, 1998.

[64] W. Yuan, N. Wu, B. Etzlinger, H. Wang und J. Kuang, “Cooperative joint localization and clock synchronization based on Gaussian message passing in asynchronous wireless networks,” *IEEE Transactions on Vehicular Technology*, Jg. 65, Nr. 9, S. 7258–7273, 2016.

[65] S. Jagannathan, H. Aghajan und A. Goldsmith, “The effect of time synchronization errors on the performance of cooperative MISO systems,” in *IEEE Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004.*, IEEE, 2004, S. 102–107.

[66] M. Goodarzi, N. Maletic, J. Gutiérrez, V. Sark und E. Grass, “Next-cell Prediction Based on Cell Sequence History and Intra-cell Trajectory,” in *2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, IEEE, 2019, S. 257–263.

[67] D. Barber, *Bayesian Reasoning and Machine Learning*. Cambridge University Press, 2012.

[68] “IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems,” *IEEE Std 1588-2019 (Revision ofIEEE Std 1588-2008)*, S. 1–499, 2020.

[69] S. Ruffini, P. Iovanna, M. Forsman und T. Thyni, “A novel SDN-based architecture to provide synchronization as a service in 5G scenarios,” *IEEE Communications Magazine*, Jg. 55, Nr. 3, S. 210–216, 2017.

[70] B. Sundararaman, U. Buy und A. D. Kshemkalyani, “Clock synchronization for wireless sensor networks: a survey,” *Ad hoc networks*, Jg. 3, Nr. 3, S. 281–323, 2005.

[71] M. Alfageme, J. Agusti, S. Martinez u. a., “5G-PICTURE Deliverable D6.3 Final Demo and Testbed experimentation results,” 2020.

[72] V. Sark, N. Maletic, M. Ehrig, J. Gutiérrez und E. Grass, “Achieving Millimeter Precision Distance Estimation using Two-Way Ranging in the 60 GHz Band,” in *2019 European Conference on Networks and Communications (EuCNC)*, IEEE, 2019, S. 310–314.

[73] S. P. Chepuri, R. T. Rajan, G. Leus und A.-J. van der Veen, “Joint clock synchronization and ranging: Asymmetrical time-stamping and passive listening,” *IEEE Signal Processing Letters*, Jg. 20, Nr. 1, S. 51–54, 2012.

[74] L. Zdeborová und F. Krzakala, “Statistical physics of inference: Thresholds and algorithms,” *Advances in Physics*, Jg. 65, Nr. 5, S. 453–552, 2016.

[75] O. Shental, P. H. Siegel, J. K. Wolf, D. Bickson und D. Dolev, “Gaussian belief propagation solver for systems of linear equations,” in *2008 IEEE International Symposium on Information Theory*, IEEE, 2008, S. 1863–1867.

[76] Y. Pei, S. Biswas, D. S. Fussell und K. Pingali, “An elementary introduction to kalman filtering,” *Communications of the ACM*, Jg. 62, Nr. 11, S. 122–133, 2019.

[77] B. Li, N. Wu und Y.-C. Wu, “Distributed Verification of Belief Precisions Convergence in Gaussian Belief Propagation,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, S. 9115–9119.

[78] A. S. Stordal, H. A. Karlsen, G. Nævdal, H. J. Skaug und B. Vallès, “Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter,” *Computational Geosciences*, Jg. 15, Nr. 2, S. 293–305, 2011.

[79] D. Alspach und H. Sorenson, “Nonlinear Bayesian estimation using Gaussian sum approximations,” *IEEE transactions on automatic control*, Jg. 17, Nr. 4, S. 439–448, 1972.

[80] F. Gustafsson, “Particle filter theory and practice with positioning applications,” *IEEE Aerospace and Electronic Systems Magazine*, Jg. 25, Nr. 7, S. 53–82, 2010.

[81] A. Doucet, N. De Freitas, K. Murphy und S. Russell, “Rao-Blackwellised particle filtering for dynamic Bayesian networks,” *arXiv preprint arXiv:1301.3853*, 2013.

[82] I. Guvenc und C.-C. Chong, “A survey on TOA based wireless localization and NLOS mitigation techniques,” *IEEE Communications Surveys & Tutorials*, Jg. 11, Nr. 3, S. 107–124, 2009.

[83] Y. Qi, H. Kobayashi und H. Suda, “On time-of-arrival positioning in a multipath environment,” *IEEE Transactions on Vehicular Technology*, Jg. 55, Nr. 5, S. 1516–1526, 2006.

[84] S. Li, M. Hedley, I. B. Collings und D. Humphrey, “Joint trajectory and ranging offset estimation for accurate tracking in NLOS environments,” *IEEE Transactions on Aerospace and Electronic Systems*, Jg. 56, Nr. 1, S. 3–14, 2019.

[85] F. Yin, C. Fritsche, F. Gustafsson und A. M. Zoubir, “EM-and JMAP-ML based joint estimation algorithms for robust wireless geolocation in mixed LOS/NLOS environments,” *IEEE Transactions on Signal Processing*, Jg. 62, Nr. 1, S. 168–182, 2013.

[86] B. Mondal, T. A. Thomas, E. Visotsky u. a., “3D channel model in 3GPP,” *IEEE Communications Magazine*, Jg. 53, Nr. 3, S. 16–23, 2015.

[87] K. Yu und E. Dutkiewicz, “NLOS identification and mitigation for mobile tracking,” *IEEE Transactions on Aerospace and electronic systems*, Jg. 49, Nr. 3, S. 1438–1452, 2013.

[88] S. Venkatraman und J. Caffery, “Statistical approach to non-line-of-sight BS identification,” in *The 5th International Symposium on Wireless Personal Multimedia Communications*, IEEE, Bd. 1, 2002, S. 296–300.

[89] S. Marano, W. M. Gifford, H. Wymeersch und M. Z. Win, “NLOS identification and mitigation for localization based on UWB experimental data,” *IEEE Journal on selected areas in communications*, Jg. 28, Nr. 7, S. 1026–1035, 2010.

[90] J. Heaton, *Introduction to neural networks with Java*. Heaton Research, Inc., 2008.

[91] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu und F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, Jg. 234, S. 11–26, 2017.

[92] “AI and ML – Enablers for Beyond 5G Networks,” 2021. Adresse: http://doi.org/10.5281/zenodo.4299895.

[93] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE transactions on antennas and propagation*, Jg. 34, Nr. 3, S. 276–280, 1986.

[94] X. Zhang, L. Xu, L. Xu und D. Xu, “Direction of departure (DOD) and direction of arrival (DOA) estimation in MIMO radar with reduced-dimension MUSIC,” *IEEE communications letters*, Jg. 14, Nr. 12, S. 1161–1163, 2010.

[95] R. Roy und T. Kailath, “ESPRIT-estimation of signal parameters via rotational invariance techniques,” *IEEE Transactions on acoustics, speech, and signal processing*, Jg. 37, Nr. 7, S. 984–995, 1989.

[96] O. A. Oumar, M. F. Siyau und T. P. Sattar, “Comparison between MUSIC and ESPRIT direction of arrival estimation algorithms for wireless communication systems,” in *The First International Conference on Future Generation Communication Technologies*, IEEE, 2012, S. 99–103.

[97] S. Boyd, S. P. Boyd und L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[98] J. Chen, S. Guan, Y. Tong und L. Yan, “Two-dimensional direction of arrival estimation for improved archimedean spiral array with MUSIC algorithm,” *IEEE Access*, Jg. 6, S. 49 740–49 745, 2018.

[99] F.-G. Yan, Z.-K. Chen, M.-J. Sun, Y. Shen und M. Jin, “Two-dimensional direction-of-arrivals estimation based on one-dimensional search using rank deficiency principle,” *International Journal of Antennas and Propagation*, Jg. 2015, 2015.

[100] M. Mohanna, M. L. Rabeh, E. M. Zieur und S. Hekala, “Optimization of MUSIC algorithm for angle of arrival estimation in wireless communications,” *NRIAG journal of Astronomy and Geophysics*, Jg. 2, Nr. 1, S. 116–124, 2013.

[101] R. Khan, S. U. Khan, S. Khan und M. U. A. Khan, “Localization performance evaluation of extended Kalman filter in wireless sensors network,” *Procedia Computer Science*, Jg. 32, S. 117–124, 2014.

[102] R. Labbe, “Kalman and Bayesian filters in Python, 2014,” 2019. Adresse: https://github.%20com/rlabbe/Kalman-and-Bayesian-Filters-in-Python.

[103] R. Karlsson, T. Schon und F. Gustafsson, “Complexity analysis of the marginalized particle filter,” *IEEE Transactions on Signal Processing*, Jg. 53, Nr. 11, S. 4408–4411, 2005.

[104] S. Jaeckel, L. Raschkowski, K. Börner und L. Thiele, “QuaDRiGa: A 3-D multi-cell channel model with time evolution for enabling virtual field trials,” *IEEE Transactions on Antennas and Propagation*, Jg. 62, Nr. 6, S. 3242–3256, 2014.