

Robust Wireless Localization in Harsh Mixed Line-of-Sight/Non-Line-of-Sight Environments

Vom Fachbereich 18
Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung der Würde eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von
Feng YIN, M.Sc.
geboren am 11.12.1984 in Liaoning, China

Referent:	Prof. Dr.-Ing. Abdelhak M. Zoubir
Korreferent:	Prof. Dr. Fredrik Gustafsson
Tag der Einreichung:	11. 06. 2014
Tag der mündlichen Prüfung:	08. 08. 2014

To my family

Acknowledgments

This doctoral dissertation summarizes the best part of my life in the past three years and four months with the signal processing group at the Technische Universität Darmstadt (TUD). This dissertation would not have been possible without the guidance, encouragement, and help from my advisor, colleagues, collaborators, family, and friends over the past years.

First and foremost, I would like to express my deepest gratitude to my advisor Prof. Abdelhak M. Zoubir for offering me a precious opportunity to work in a vibrant group full of brilliant fellows. To me, you are not only a great researcher and leader, but also like a merciful father. From you, I have gained not only the knowledge but also self-confidence. Thank you so much for taking me to the world of statistical signal processing. The first DSP lecture and the first excursion to ATLAS Elektronik in 2008 were so fascinating and unforgettable. It has been a great pleasure to be a doctoral student under your supervision. I am also greatly indebted to Prof. Fredrik Gustafsson from the Linköping University, Sweden for the collaborations in the past three years. I strongly agree with all your students that you are a never-ending source of smart ideas and inspiration. I am also impressed by your amazing insight and magic of folding a half-baked idea into a bigger picture. Many thanks for guiding me into the fascinating field of Bayesian inference for localization. Your IEEE signal processing magazine paper (2005) is the first paper that I have ever read in my PhD career—indeed an excellent start. I can barely imagine at that time that I will work with you and Prof. Gunnarsson in Linköping some years later. Many thanks also go to Prof. Rolf Jokoby, Prof. Marius Pesavento, and Prof. Ulrich Konigorski for being a member of my PhD oral examination committee.

During my Master study, I worked in parallel as a student research associate at the Fraunhofer Institute SIT, Darmstadt. I am very grateful to my mentor Dr. Huajian LIU, who taught me patiently how to combine theory with practice and how to code more efficiently and elegantly. It has been always a pleasure to work with you, even late in the evening. Many thanks also go to Dr. Christian Debes, supervisor of my Master thesis, for his generous help on everything. I was very lucky to have met you by accident in the corridor, talking about possible thesis topics. I owe a huge gratitude to my collaborator Dr. Carsten Fritsche for his guidance, immense technical expertise, and fruitful discussions. You always have a full basket of fresh ideas and strong enthusiasm for research. I have learnt a lot from you. Many thanks to Dr. Gebremichael Teame and Dr. Roy Howard from the Curtin University, Australia,

with whom I have shared an office. Many thanks also go to the members of the signal processing group, both current and past, including Adrian Šošić, Sara Al-Sayed, Mouhammad Alhumaidi, Mark Balthasar, Toufik Boukaba, Nevine Demitri, Michael Fauss, Gökhan Gül, Jürgen Hahn, Lala Khadidja Hamaidi, Sahar Khawatmi, Michael Lang, Dr. Stefan Leier, Michael Leigsnering, Abdelmalek Mennad, Dr. Michael Muma, Tim Schäck, Wassim Suleiman, Christian Weiß, Dr. Zhihua Lu, Dr. Christian Debes, Dr. Fiky Suratman, Dr. Philipp Heidenreich, Dr. Waqas Sharif, Dr. Ahmed Moustafa, Dr. Raquel Fandos, Dr. Yacine Chakhchoukh, Dr. Ulrich Hammes, Dr. Ramon Brcic for their kind help and fruitful discussions all along my PhD study. Special thanks go to Dr. Michael Muma, Dr. Stefan Leier, Michael Leigsnering, Jürgen Hahn, Christian Weiß, Adrian Šošić, Sara Al-sayed, Nevine Demitri, Gökhan Gül, Lala Khadidja Hamaidi, Michael Lang, Tim Schäck for their valuable feedback on my disseration and presentation slides. I feel very proud to have you guys around. Without the hard work of our secretary, Renate Koschella, and system administrator, Hauke Fath, my PhD study would not have gone so smoothly. You are the best! I am also very grateful to Dr. Gulam RAZUL Sirajudeen and Dr. Chong Meng Samson See for their support when I was visiting the Temasek Lab at the Nanyang Technological University, Singapore. Many thanks also go to Dr. Lei LEI, Dr. Guohua WANG, and Dr. Zhihua LU for their warm reception. During my PhD study, I am very lucky to have quite a few brilliant students who helped me develop and test new ideas in the Master thesis; they are Ang LI, Di JIN, Yi ZHANG.

Lastly and most importantly, I would like to thank my wife Yiyao LI, my parents Fengguang YIN and Lin XU, as well as my parents-in-law Dayong LI and Huichun HE for their encouragement and endless love. I also owe a lot to my Chinese friends in Germany; they are Hua ZHONG, Gan ZHOU, Angran YANG, Dr. Zhihua LU, Xiyue FANG, Dr. Huajian LIU, Di JIN, Xin HUANG, Ang LI, Yi ZhANG, Zhiliang CHEN, Zheng LI, Weibin ZHANG, Xiao LUO, Jing NING, Tai FEI for their company.

Darmstadt, 12 August, 2014

Kurzfassung

Diese Dissertation befasst sich mit dem Problem der Lokalisierung von Knoten in verschiedenen drahtlosen Infrastrukturen, wie zum Beispiel Mobilfunknetzen und drahtlosen Sensornetzen. Um so realistisch wie möglich zu sein, werden gemischte Lokalisierungsbedingungen mit und ohne direkter Sichtverbindung (LOS/NLOS) vorgestellt. Sowohl herkömmliche nicht-kooperative, als auch neuartige kooperative Lokalisierungsmethoden wurden gründlich untersucht. Aufgrund der zufälligen Natur der Messungen, bilden probabilistische Methoden im Vergleich zu traditionellen geometrischen Methoden die fortgeschritteneren Ansätze. Die Quintessenz der probabilistischen Methoden besteht darin, die unbekannt Positionen der Zielknoten in einem Schätzprozess zu bestimmen. Gegeben sind hierbei verrauschte positionsbezogene Messwerte, ein probabilistisches Messmodell, sowie einige bekannte Referenzpositionen.

Im Gegensatz zur Mehrheit des existierenden Methoden werden strenge, jedoch praktisch relevante Beschränkungen behandelt: Das gewünschte Lokalisierungssystem beinhaltet weder eine Offline-Kalibrierung, noch ist es möglich die Existenz einer direkten Sichtverbindung zu erkennen. Dadurch ist die Messfehlerstatistik unbekannt, wodurch die Folgerung von Rückschlüssen eine extreme Herausforderung darstellt. Zwei neue Klassen von Lokisierungsalgorithmen zur gemeinsamen Schätzung von Positionen und Messfehlerstatistik werden vorgeschlagen. In dieser Dissertation werden alle unbekannt Parameter als deterministisch betrachtet und es wird jeweils nach dem Maximum-Likelihood (ML) Schätzer gesucht.

Algorithmen der ersten Klasse setzen keine Kenntnis der Messfehlerstatistik voraus und wenden ein nichtparametrisches Modell an. Die Idee besteht in der alternierenden Anwendung einer Schätzung der Wahrscheinlichkeitsdichtefunktion einerseits, wobei eine Approximation der unbekannt Messfehlerstatistik über eine adaptive Kerndichteschätzung erfolgt. Andererseits wird eine Parameterschätzung der Position ausgeführt, welche auf eine Approximation der Log-Likelihood Funktion beruht. Der Rechenaufwand für Algorithmen dieser Klasse wächst quadratisch mit der Anzahl der Messwerte, wodurch sich die Anwendbarkeit im Wesentlichen auf die nicht-kooperative Lokalisierung in Mobilfunknetzen beschränkt. Eine zweite Klasse von Algorithmen zielt daher auf eine Reduzierung des Rechenaufwandes ab, wofür eine Approximation der Messfehlerstatistik mittels einer Kombination von Gaussischen Dichtefunktionen verwendet wird. Iterative Algorithmen, welche zwischen Aktualisierungen von Positionen und anderen Parametern alternieren, wurden mit Hilfe von Expectation-Maximization (EM), Expectation-Conditional Maximization (ECM) und Joint Max-

imum A Posteriori-ML (JMAP-ML) Prinzipien entwickelt. Wie sich herausstellte, wächst der Rechenaufwand von Algorithmen dieser zweiten Klasse nunmehr linear mit der Anzahl der zur Verfügung stehenden Messwerte, wodurch eine Erweiterung des Anwendungsbereiches auf kooperative Lokalisierung für drahtlose Sensornetzen möglich wird.

Abgesehen von dem Algorithmenentwurf selbst wurden zur umfassenden Evaluierung derselben systematische Analysen im Hinblick auf die Cramer-Rao-Schranken, den Rechenaufwand sowie den für die Kommunikation anfallenden Leistungsverbrauch durchgeführt. Anhand der Simulations- und Versuchsergebnisse konnte gezeigt werden, dass die vorgeschlagenen Algorithmen für hinreichend große Datensätze die fundamentalen Schranken der Lokalisierungsgenauigkeit erreichen. Sofern der Einfluss etwaiger Modellfehlanspassungen vernachlässigt werden kann, sind die vorgeschlagenen Verfahren den konkurrierenden weit überlegen.

Abstract

This PhD thesis considers the problem of locating some target nodes in different wireless infrastructures such as wireless cellular radio networks and wireless sensor networks. To be as realistic as possible, mixed line-of-sight and non-line-of-sight (LOS/NLOS) localization environment is introduced. Both the conventional non-cooperative localization and the new emerging cooperative localization have been studied thoroughly. Owing to the random nature of the measurements, probabilistic methods are more advanced as compared to the old-fashioned geometric methods. The gist behind the probabilistic methods is to infer the unknown positions of the target nodes in an estimation process, given a set of noisy position related measurements, a probabilistic measurement model, and a few known reference positions.

In contrast to the majority of the existing methods, harsh but practical constraints are taken into account: neither offline calibration nor non-line-of-sight state identification is equipped in the desired localization system. This leads to incomplete knowledge about the measurement error statistics making the inference task extremely challenging. Two new classes of localization algorithms have been proposed to jointly estimate the positions and measurement error statistics. All unknown parameters are assumed to be deterministic, and maximum likelihood estimator is sought after throughout this thesis.

The first class of algorithms assumes no knowledge about the measurement error distribution and adopts a nonparametric modeling. The idea is to alternate between a pdf estimation step, which approximates the exact measurement error pdf via adaptive kernel density estimation, and a parameter estimation step, which resolves a position estimate numerically from an approximated log-likelihood function. The computational complexity of this class of algorithms scales quadratically in the number of measurements. Hence, the first class of algorithms is applicable primarily for non-cooperative localization in wireless cellular radio networks. In order to reduce the computational complexity, a second class of algorithms resorts to approximate the measurement error distribution parametrically as a linear combination of Gaussian distributions. Iterative algorithms that alternate between updating the position(s) and other parameters have been developed with the aid of expectation-maximization (EM), expectation conditional maximization (ECM) and joint maximum *a posteriori*-maximum likelihood (JMAP-ML) criteria. As a consequence, the computational complexity turns out to scale linearly in the number of measurements. Hence, the second class of algorithms is also applicable for cooperative localization in wireless sensor networks.

Apart from the algorithm design, systematical analyses in terms of Cramér-Rao lower bound, computational complexity, and communication energy consumption have also been conducted for comprehensive algorithm evaluations. Simulation and experimental results have demonstrated that the proposed algorithms all tend to achieve the fundamental limits of the localization accuracy for large data records and outperform their competitors by far when model mismatch problems can be ignored.

Contents

1	Introduction	1
1.1	Wireless Localization and Applications	1
1.2	Localization Problems Addressed	2
1.3	Localization Measurements and Models	4
1.3.1	Measurement Categories	4
1.3.2	Error Sources	5
1.3.3	Mode-Dependent Modeling	6
1.4	Harsh Mixed LOS/NLOS Environments	8
1.4.1	Simplified Modeling and Optimality	8
1.4.2	Key Assumptions	10
1.5	A Taxonomy of Existing Algorithms	11
1.5.1	Non-cooperative Localization	11
1.5.2	Cooperative Localization	13
1.6	Thesis Outline and Contributions	14
2	Background	19
2.1	Overview of Classical Parameter Estimation	19
2.2	ML Estimation with Incomplete Data	22
2.2.1	EM Algorithm	23
2.2.2	ECM Algorithm	25
2.2.3	JMAP-ML Algorithm	27
2.2.4	Connections	27
2.3	Appendix	28
2.3.1	Gradient and Laplace Operators	28
3	Localization in Cellular Radio Networks: Nonparametric Modeling	31
3.1	Signal Model and Problem Statement	31
3.2	Joint ML Estimation using KDE	33
3.2.1	Competing Algorithm	34
3.2.2	Proposed RIN Algorithm	36
3.3	Cramér-Rao Lower Bound Computation	39
3.4	Theoretical Performance Metrics	40
3.4.1	Bias, RMSE, and Efficiency	40
3.4.2	Geometric Dilution of Precision	40
3.5	Simulations	41
3.5.1	Simulation Setup	41
3.5.2	Simulation Results	43

3.6	Conclusions	51
3.7	Appendix	51
3.7.1	Adaptive Kernel Density Estimation	51
3.7.2	Derivations of (3.23)	53
3.7.3	Expressions of $p_V(v)$ and $\nabla_v p_V(v)$	54
4	Localization in Cellular Radio Networks: Parametric Modeling	57
4.1	Signal Model and Problem Statement	58
4.2	Joint ML Estimation	59
4.2.1	EM Algorithm	60
4.2.2	JMAP-ML Algorithm	61
4.2.3	Implementation Details	63
4.3	Performance Evaluation	67
4.3.1	Convergence Properties	67
4.3.2	Computational Complexity	72
4.4	An Alternative View of the EM Algorithm	73
4.5	Cramér-Rao Lower Bound Computation	79
4.6	Simulations	80
4.6.1	Simulation Setup	81
4.6.2	Simulation Results	81
4.7	Conclusions	98
4.8	Appendix	99
4.8.1	Derivations of (4.11)	99
4.8.2	Optimality of (4.58), (4.59), and (4.60)	100
4.8.3	Derivations of $\mathcal{O}(CNK)$ Complexity	102
4.8.4	Expression of $\nabla_{\theta} \ln p(\mathbf{r}; \theta)$	106
4.8.5	An Initialization Example	107
5	Cooperative Localization in WSNs: Parametric Modeling	109
5.1	Signal Model and Problem Statement	110
5.2	Competing Algorithms Revisited	111
5.2.1	LS Estimation Based Algorithms	112
5.2.2	NBP Based Algorithms	113
5.3	Joint ML Estimation	115
5.3.1	ML Estimation	116
5.3.2	Centralized ECM Algorithms	116
5.3.3	Distributed ECM Algorithms	122
5.3.4	Dynamic Network Localization	126
5.4	Performance Evaluation	127
5.4.1	Computational Complexity	128

5.4.2	Energy for Communication	129
5.5	Cramér-Rao Lower Bound Computation	131
5.6	Simulations	133
5.6.1	Simulation Setup	133
5.6.2	Simulation Results	134
5.7	Conclusions	140
5.8	Appendix	142
5.8.1	Derivations of (5.19)	142
6	Conclusions and Ongoing Work	145
6.1	Conclusions	145
6.2	Ongoing Work	147
	List of Acronyms	149
	List of Symbols	153
	Bibliography	161
	Publications	171
	Curriculum vitae	173

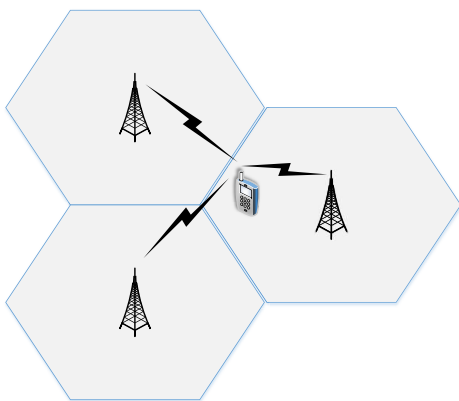
Chapter 1

Introduction

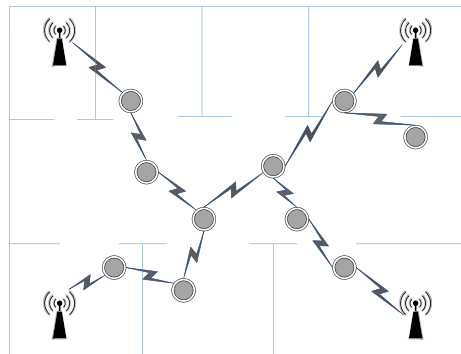
1.1 Wireless Localization and Applications

Wireless localization refers to the problem of finding the positions of some target nodes in different wireless infrastructures such as cellular radio networks and wireless sensor networks (WSNs) [1]. A target node can be a mobile station (MS) in cellular radio networks or an agent in WSNs. Fig. 1.1(a) and Fig. 1.1(b) provide two illustrating examples. Wireless localization systems serve as replacement or complement of the conventional global positioning system (GPS) in harsh indoor environments, urban areas and underwater environments, where the GPS signals can be either largely impaired or unavailable [2].

Over the past two decades, wireless localization has received considerable attention due to the expanding location-based services, such as wireless emergency service Enhanced-911 (E-911), location-sensitive billing, fraud detection, asset tracking, intelligent transportation, mobile yellow pages [3], soldier and first responder locating [4,5], and animal tracking [6], to enumerate a few. High-accuracy wireless localization will continue to play a key role for public safety and drive many more location-based services (especially on smart phones) in the forthcoming years [7].



(a) Localization of a mobile station in a cellular radio network for E-911 service in an outdoor rural environment.



(b) Localization of several sensor nodes (marked by circles) in a wireless sensor network for environment surveillance in an indoor environment.

Figure 1.1. Two illustrating examples of wireless localization.

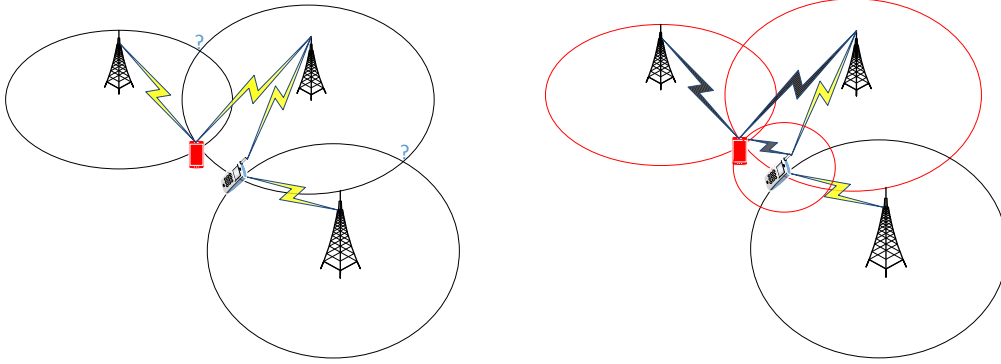
1.2 Localization Problems Addressed

Throughout this thesis, the unknown target node positions are inferred in an estimation process, given a batch of noisy position-related measurements, a few known reference nodes and a probabilistic signal model. A reference node can be a base station (BS) in cellular radio networks or an anchor in WSNs. In [8], estimation problems are broadly categorized into localization of stationary targets, tracking of moving targets, self-navigation, calibration, and simultaneous localization and mapping (SLAM).

In this thesis, localization of stationary targets is mainly studied. Two fundamental problems are distinguished as follows:

- **Non-cooperative localization.** Non-cooperation is a conventional paradigm for localization in that the target nodes communicate only with a sufficient number of reference nodes. With the position-related measurements, either each target node determines its own position (mobile-based) or a fusion center determines the target node position and sends the information back (network-based) [1]. In order to resolve localization ambiguities, each target node must be able to communicate with at least three reference nodes in a two-dimensional (2-D) space or four reference nodes in a three-dimensional (3-D) space. Fig. 1.2(a) shows an example for the 2-D case.
- **Cooperative localization.** Cooperation is a new emerging paradigm for localization in that the target nodes additionally exploit the measurements collected between themselves. Similarly, the target nodes self-localize themselves (in a distributed manner) or a fusion center determines their positions (in a centralized manner) and transmit the information back to each individual via multi-hops. With the cooperations among the target nodes, the communication range of wireless devices as well as the number of anchors to be deployed can be tremendously reduced, which in turn economizes the overall cost for building a localization system. In addition, cooperations can help resolve localization ambiguities and as reward bring more robust and accurate position estimates. Fig. 1.2(b) shows the benefit of using cooperations among nodes.

It is noteworthy that in the special case where there is no motion model available or the state uncertainty is sufficiently large in the system dynamics, both target tracking and self-navigation can be done through conducting a stationary target localization algorithm repeatedly at different time instances—a snapshot-based method.



(a) Non-cooperative localization (2-D) using a cellular radio network. The two mobile stations do not cooperate and can only reach two base stations, giving rise to localization ambiguities (the locations with question-mark).

(b) Cooperative localization (2-D) using the same cellular radio network. The two mobile stations now cooperate and as a result the localization ambiguity can be resolved completely.

Figure 1.2. Non-cooperative localization versus cooperative localization.

These two fundamental localization problems can be cast into a general parameter estimation problem with the measurement model in form of

$$\mathbf{r} = \mathbf{h}(\boldsymbol{\theta}_p, \boldsymbol{\theta}_a) + \mathbf{v} \quad (1.1)$$

where the parameters are defined as:

- Column vector \mathbf{r} includes a set of position-related measurements. For non-cooperative localization, \mathbf{r} contains the measurements obtained between the target node and several reference nodes. For cooperative localization, \mathbf{r} contains additionally the measurements obtained between target nodes. The dimension of \mathbf{r} is assumed to be M .
- Nonlinear function $\mathbf{h}(\boldsymbol{\theta}_p, \boldsymbol{\theta}_a)$ represents the ideal measurement model, which depends on the unknown positions $\boldsymbol{\theta}_p$ and some auxiliary parameters $\boldsymbol{\theta}_a$. For non-cooperative localization, $\boldsymbol{\theta}_p$ contains one unknown target node position. For cooperative localization, $\boldsymbol{\theta}_p$ contains a set of unknown target node positions to be determined concurrently.
- Column vector \mathbf{v} contains a set of measurement error terms, v_1, v_2, \dots, v_M , that follow a certain probability density function (pdf).

The primary goal is to provide an accurate estimate of the unknown position(s), $\boldsymbol{\theta}_p$, in a short response time. The diversity of the existing localization algorithms stems from:

- The manner of localization, for instance, mobile-based versus network-based, centralized versus distributed.
- The use of different measurement models, for instance time-of-arrival (TOA) and received-signal-strength (RSS) measurement models.
- The nature of the unknown positions, deterministic or random.
- The use of different estimation criteria, for instance, maximum likelihood (ML) criterion or minimum-mean-square-error (MMSE) criterion.
- The amount of knowledge (known or partially known or unknown) about the measurement error statistics.
- The need for offline calibration of θ_a and/or the measurement error statistics prior to the localization process.
- The consideration of different constraints, e.g., road constraint, geometric constraint, and communication constraints (bandwidth and energy).
- The number of measurements collected at each node.
- The approximations used to trade-off desired properties of the designed algorithm, e.g., linearization of $\mathbf{h}(\theta_p, \theta_a)$ for lower complexity and better convergence.

In Section 1.5, a survey of the existing algorithms will be given for both non-cooperative localization and cooperative localization.

1.3 Localization Measurements and Models

1.3.1 Measurement Categories

In this thesis, a two-step procedure for localization is adopted primarily due to the lower complexity as compared to direct localization (see e.g., [9], [10]). In the first step, position-related measurements are extracted from the received signals. In the second step, the obtained measurements are processed (either centralized or distributed) to give an estimate of the unknown positions. The most commonly used measurement categories are classified into signal waveform, time-of-arrival, time-difference-of-arrival (TDOA), round-trip time-of-arrival (RTOA), received signal strength, and angle-of-arrival (AOA) [11]. In the sequel, the acquisition of TOA, TDOA, RTOA and RSS

measurements is briefly reviewed with the aid of a toy example in which N base stations (BSs) attempt to locate a single MS in a cellular radio network.

- **TOA, TDOA, RTOA:** For an MS-BS wireless link (say the i th), a complex channel impulse response, $h_i(t)$, is first computed and then converted to power delay profile (PDP) $|h_i(t)|^2$ [12]. From the PDP, a TOA measurement for this channel, τ_i , is obtained by estimating the first arrived path, if the transmitter is time-synchronized with the receiver. The existing methods for TOA estimation include correlation based methods, deconvolution methods, maximum likelihood estimation based methods, and subspace based methods [13, Chapter 7]. In practice, TOA is usually converted into distance in light of $d_i = \tau_i \cdot c$, where c_0 is the propagation speed of a radio wave. Alternatively, a TDOA measurement (say between BS i and BS j) can be computed by taking time differences $\tau_i - \tau_j$. The advantage of this method is that only the BSs need to be synchronized in time. Collecting RTOA measurements requires no time synchronization among nodes. The acquisition of a RTOA measurement can be obtained, for instance at the i th BS, by $\tau_i^{\text{rec}} - \tau_i^{\text{send}} - \tau_{i,\Delta} = 2d_i/c$ where τ_i^{rec} is the time instance that this BS received the waveform that it sent at τ_i^{send} and bounced by the MS with a time delay $\tau_{i,\Delta}$ [14]. The time delay $\tau_{i,\Delta}$ is usually predetermined and known to both the MS and the i th BS.

- **RSS:** A received signal strength measurement can be obtained by integrating the PDP with respect to time. More precisely, the RSS [dBm] measured at the i th BS is

$$P_i = 10 \log \left(\frac{\int |h_i(t)|^2 dt}{1 \text{ mW}} \right). \quad (1.2)$$

In contrast to the timing measurements mentioned above, received signal strength is only empirically related to the actual distance, for instance according to the classical Okumura-Hata model [3]. Although RSS measurement provides rather coarse distance information, the acquisition of it is easy to conduct in almost any existing wireless infrastructure and requires no time-synchronization among nodes.

1.3.2 Error Sources

Given precise timing measurements, localization can be easily performed in a simple geometric approach called trilateration. However, not all the circles (for TOA or RTOA measurements) or hyperbolas (for TDOA measurements) intersect at a single point in

practice due to the measurement error. Fig. 1.3 depicts one example. Error sources stem from the measurement equipment *per se* and propagation environments for both narrow-band and wide-band systems [15]. The error sources are:

- **Equipment deficiency:** Equipment deficiency includes thermal noise in the electronic circuits of the hardware, quantized equipment readings, and incorrect operations of an equipment.
- **Multi-path propagation:** Multi-path propagation causes the phenomenon that the transmit signal reaches the receiver via numerous paths with different attenuations and time delays. The overlap of multiple replicas of the transmit signal incurs ambiguities when detecting the first arrival path of a signal, even if a line-of-sight (LOS) path exists [15]. One way to alleviate the ambiguities is to use ultra-wide-band (UWB) transmit signals, whose time-resolution is high [16]. The constructive and destructive interference of these replicas also incurs large fluctuations of the received signal strength over a distance in the order of the carrier wavelength. This phenomena is also called small-scale fading [17].
- **Shadowing:** Shadowing effect is due to the energy absorption at large obstacles between the transmitter and receiver. This effect causes difficulty in determining the TOA and introduces an approximately log-normal distributed error term in the received signal strength [17].
- **Non-line-of-sight (NLOS) propagation:** NLOS propagation describes either the scenario where the LOS path between a transmitter and a receiver is completely obstructed, which is known as LOS blockage, or the scenario where the LOS path is only partially obstructed and the signal can still penetrate obstacles like walls and windows, which is known as LOS excess delay [18]. In both scenarios, NLOS propagation tends to incur a positive bias (for TOA) and increase uncertainties in the estimate (for both TOA and RSS).

It is noteworthy that for simplicity the influence of algorithm inefficiency (e.g., for TOA estimation) and multiple access interference (MAI) among nodes are neglected. In the sequel, the errors introduced by the hardware are also ignored, since they are usually negligible in comparison with the environmental errors [15].

1.3.3 Mode-Dependent Modeling

Essentially, different error sources influence TOA and RSS estimates concurrently. To find an adequate modeling, various measurement campaigns have been conducted in

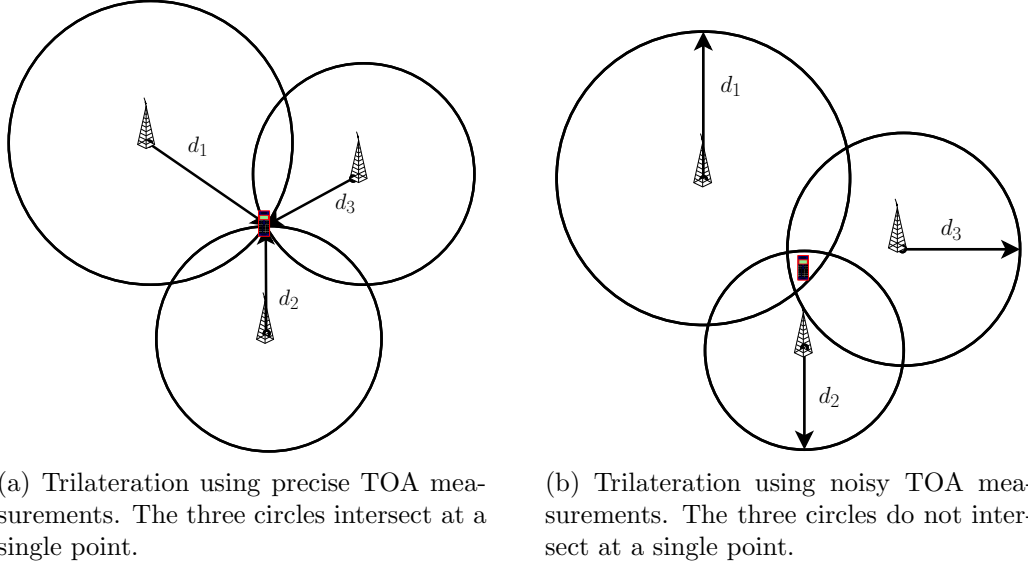


Figure 1.3. Trilateration using TOA measurements with and without errors.

different scenarios, see for instance [14, 19–23] for the TOA modeling and [19, 20, 22, 24] for the RSS modeling. It is not surprising that the campaign results vary with scenarios. To approximate these results as close as possible and meanwhile maintain mathematical tractability, two simple mode-dependent modelings are given as follows:

- **Mode-dependent TOA modeling:** A time-of-arrival measurement obtained at a receiver node can be modeled as

$$\tau_{\text{TOA}} = \begin{cases} d/c + v_{\text{L}}, & \text{LOS condition} \\ d/c + v_{\text{NL}}, & \text{NLOS condition} \end{cases} \quad (1.3)$$

where v_{L} is the measurement error under the LOS condition and v_{NL} is the measurement error under the NLOS condition. In the literature, for instance [3, 22, 25–31]), v_{L} is favorable to be represented by a Gaussian distribution with mean μ_{L} (around zero) and variance σ_{L}^2 . While depending on the localization scenario, v_{NL} may follow a shifted Gaussian distribution (e.g., in [3, 22, 27–32]), an exponential distribution (e.g., in [25, 26, 32]), a Rayleigh distribution (e.g., in [27, 32–34]), or a Weibull distribution [35].

- **Mode-dependent RSS modeling:** A received-signal-strength measurement obtained at a receiver node can be modeled as

$$P_{\text{RSS}} = \begin{cases} P_{\text{T}} - (A_{\text{L}} + 10B_{\text{L}} \log(\frac{d}{d_0})) + v_{\text{L}}, & \text{LOS condition} \\ P_{\text{T}} - (A_{\text{NL}} + 10B_{\text{NL}} \log(\frac{d}{d_0})) + v_{\text{NL}}, & \text{NLOS condition} \end{cases} \quad (1.4)$$

where P_{T} (dBm) is the transmit power, $A_{(\text{N})\text{L}}$ and $B_{(\text{N})\text{L}}$ (in dB scale) denote respectively the path loss value at a reference distance d_0 and path loss exponent

value for (N)LOS scenario. The measurement error $v_L \sim \mathcal{N}(0, \sigma_L^2)$ under the LOS condition and $v_{NL} \sim \mathcal{N}(0, \sigma_{NL}^2)$ under the NLOS condition. The standard deviation σ_{NL} (dB) is usually much larger than σ_L (dB), reflecting the fact that the shadowing effect is more prominent under the NLOS condition. It is assumed here that the multi-path (small scale) fading effect has been effectively eliminated by time-averaging [36].

Lastly, it is noteworthy that the NLOS effect is the main, yet not the only, reason for the above mode (LOS or NLOS)-dependent modelings.

1.4 Harsh Mixed LOS/NLOS Environments

Throughout this thesis, both the non-cooperative localization and cooperative localization are considered in harsh mixed line-of-sight and non-line-of-sight (LOS/NLOS) environments, where

- LOS measurements coexist with NLOS measurements in the given batch of observed data;
- Offline calibration is either not performed or only coarsely performed;
- NLOS identification is not performed.

1.4.1 Simplified Modeling and Optimality

From the two mode-dependent modelings given in Section 1.3.3, it is easy to conclude that

$$v = \begin{cases} v_L \sim p_V^{(L)}(v; \boldsymbol{\beta}_L), & \text{LOS condition} \\ v_{NL} \sim p_V^{(NL)}(v; \boldsymbol{\beta}_{NL}), & \text{NLOS condition} \end{cases}. \quad (1.5)$$

In order to be robust against the NLOS measurements (or outliers), the idea proposed originally in [3] is followed. That is, the measurement error terms observed for different wireless channels are independently and identically distributed (iid) and follow a two-mode mixture distribution in the form of

$$p_V(v) = \alpha_L p_V^{(L)}(v; \boldsymbol{\beta}_L) + \alpha_{NL} p_V^{(NL)}(v; \boldsymbol{\beta}_{NL}) \quad (1.6)$$

where v follows an ‘‘LOS’’ distribution $p(v|\text{LOS}) = p_V^{(\text{L})}(v; \boldsymbol{\beta}_\text{L})$ with prior probability $\Pr\{\text{LOS}\} = \alpha_\text{L}$ while an ‘‘NLOS’’ distribution $p(v|\text{NLOS}) = p_V^{(\text{NL})}(v; \boldsymbol{\beta}_\text{NL})$ with prior probability $\Pr\{\text{NLOS}\} = \alpha_\text{NL} = 1 - \alpha_\text{L}$. In the literature, α_NL is also known as the NLOS contamination ratio. This is the best model that can be chosen, given a batch of measurement error terms but without knowing which channel state has generated the corresponding measurement. However, it is noted that this simplified model may shuffle any temporal or spatial correlated patterns in the observed data and lead to information loss. In the sequel, $p_V(v)$ is called the measurement error distribution despite that it may deviate from the actual one.

For lack of offline calibration of the environmental parameters, incomplete knowledge about the measurement error statistics is assumed throughout this thesis. Two cases are distinguished as follows:

1. $p_V(v)$ is completely unknown due to uncertainties about the distributive profiles of both $p_V^{(\text{L})}(v; \boldsymbol{\beta}_\text{L})$ and $p_V^{(\text{NL})}(v; \boldsymbol{\beta}_\text{NL})$.
2. $p_V(v)$ is partially known with both $p_V^{(\text{L})}(v; \boldsymbol{\beta}_\text{L})$ and $p_V^{(\text{NL})}(v; \boldsymbol{\beta}_\text{NL})$ belonging to designated families of distributions.

In the following chapters, the main aim is to jointly estimate the unknown positions and rebuild $p_V(v)$. We adopt two distinct approaches to the modeling of $p_V(v)$, namely a nonparametric approach and a parametric approach, in our joint estimation problems. In the nonparametric model, $p_V(v)$ is approximated by a kernel density estimate [37] as

$$p_V(v) \approx \hat{p}_V(v) = \frac{1}{M} \sum_{m=1}^M \frac{1}{w\lambda_m} \mathcal{K} \left(\frac{(v - v_m)}{w\lambda_m} \right) \quad (1.7)$$

where M is the total number of measurements, $\mathcal{K}(v)$ can be any favorable classes of kernel densities, like standard Gaussian kernels, and the meanings of the other parameters w and λ_m will be explained in more detail in Chapter 3. In the parametric model, $p_V(v)$ is approximately represented as a C -mode mixture model, namely,

$$p_V(v) \approx \hat{p}_V(v) = \sum_{l=1}^C \alpha_l p_V^{(l)}(v; \boldsymbol{\beta}_l) \quad (1.8)$$

where $p_V^{(l)}(v; \boldsymbol{\beta}_l)$ is favorable to be Gaussian, as any distribution can be approximated as closely as desired, for instance in \mathcal{L}_1 norm, by a Gaussian mixture [38]. In contrast to the nonparametric approach where the parameters w , λ_m are set adaptively and nonparametricly (according to rule-of-thumb or some optimality criteria) before

performing localization, the mixture model parameters $\boldsymbol{\theta}_e = [\alpha_1, \dots, \alpha_C, \beta_1, \dots, \beta_C]$ are determined jointly with the unknown positions.

As it is well known, model mismatch will degrade the ML estimation performance. Specific to the addressed localization problems, various possible model mismatch problems that may occur are explained with the aid of Fig. 1.4 as follows:

- **Problem I:** The mode-dependent model in (1.5) is insufficient to represent the underlying measurement error.
- **Problem II:** The iid assumption is invalid when approximating (1.5) by a two-mode mixture distribution in (1.6). An example for this case is that a sequence of measurements obtained from the same wireless link but at different time instances are more likely to be generated according to a sequential pattern with a constant state (either LOS or NLOS) or according to a specific Markov chain model [39].
- **Problem III:** Even if (1.6) precisely characterizes the underlying measurement error, i.e., the above two model mismatch problems do not appear, both the non-parametric and parametric representations of (1.6) lead to approximation error when $p_V(v)$ is unknown. However, this can be avoided when $p_V(v)$ is partially known and the parametric model is used.

As a conclusion, we note that it is extremely difficult to find an optimal model analytically. Despite the sub-optimality of (1.6) in many practical problems, imagining it as the actual model allows for quite good robustness against outliers and meanwhile facilitates the design of new localization algorithms. These will be seen in the subsequent chapters.

1.4.2 Key Assumptions

To facilitate the algorithm design and performance evaluations in the subsequent chapters, the following assumptions are made throughout the thesis.

- A0 : The target(s) to be located remain stationary during the localization process.
- A1 : The measurement error terms in \mathbf{v} are assumed to be iid although it might not be true in reality.

A2 : Wireless transmission of data packages is lossless.

A3 : No quantization of the measurements.

In addition, there are other assumptions made specific to the considered problem in each chapter.

1.5 A Taxonomy of Existing Algorithms

In this section, we survey the existing localization algorithms with emphasis on those considering NLOS mitigation (cf. Section 1.5.1 for non-cooperative localization and cf. Section 1.5.2 for cooperative localization). For detailed descriptions of the existing algorithms, interested readers are referred to [13, 40].

1.5.1 Non-cooperative Localization

Non-cooperative localization is primarily considered in wireless cellular radio networks. Numerous existing algorithms (e.g., [41–49]) assume pure LOS environments and use simple Gaussian model of the measurement error. In indoor environments and dense urban areas, non-line-of-sight (NLOS) effect significantly degrades the estimation performance of these algorithm. Therefore, advanced algorithms that are robust to the NLOS measurements (outliers) are constantly sought after. The existing NLOS mitigation algorithms can be broadly categorized into the identify and discard based algorithms, the programming based algorithms, and the robust estimation based algorithms.

The essence of the identify and discard based algorithms (e.g., [50–52]) is to identify and discard those NLOS-corrupted distance measurements. The remaining distance measurements, classified as LOS measurements, are then used by conventional algorithms (e.g., least-squares (LS) estimation based algorithms) to compute an accurate position estimate. The key idea of programming based algorithms is to formulate the position estimation problem as a constraint optimization problem, which can be solved with the aid of some mathematical programming techniques (e.g., quadratic programming (QP) [53], linear programming (LP) [54], and semi-definite programming (SDP) [55]). In order to combat the NLOS effect, robust estimation based algorithms resort to replace the least-squares residual formulation by robust statistics based on [56], [57]. In [58, 59], robust least-median-squares (LMS) based algorithms were proposed. In [60],

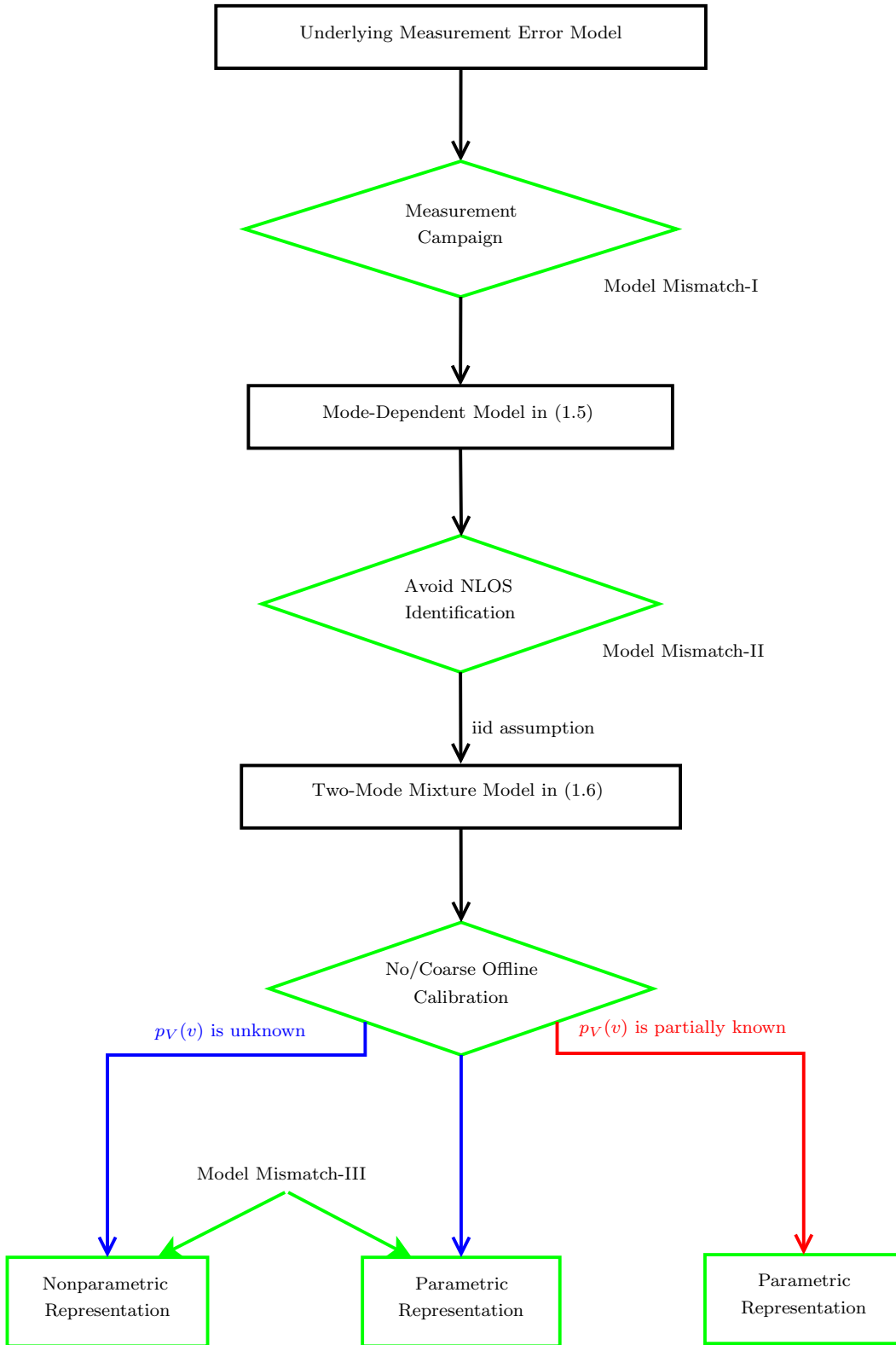


Figure 1.4. Constraints imposed and approximations made to arrive at a nonparametric or a parametric model of the actual measurement error.

a robust bootstrapping M-estimation algorithm that combines Huber's M-estimation and bootstrap techniques [61–63] was proposed. For further improvement, an adaptive algorithm that tunes Huber's score function was introduced in [32, 64].

In the harsh mixed LOS/NLOS environments assumed in Section 1.4, robust position estimation becomes extraordinarily challenging. To the best of our knowledge, those algorithms from the first and second categories are not able to work with incomplete knowledge about the measurement error statistics. Robust estimation based algorithms are favorable to be employed under harsh situations, while the classical algorithms are merely robust for up to 50% of outliers. In order to achieve higher robustness against the NLOS effect, novel robust estimation algorithms have been proposed in [27, 64–67]. The common gist of these approaches is to approximate the maximum likelihood estimator of the unknown parameters jointly in an iterative algorithm that alternates between a position estimation step and a pdf estimation step.

1.5.2 Cooperative Localization

Cooperative localization is primarily considered in wireless sensor networks. In the past decade, a plethora of cooperative localization algorithms has been proposed based on different position-related measurement categories as those listed in Section 1.3.1. Herein, the class of concurrent algorithms are the main focus, as they can avoid localization error propagation as compared to the class of sequential algorithms. The existing concurrent algorithms can be further categorized into non-Bayesian algorithms and Bayesian algorithms. In the non-Bayesian algorithms, the unknown (true) positions are assumed to be deterministic. Classical non-Bayesian algorithms (both centralized and distributed) include: (1) least-squares estimation based algorithms [19, 68], and [14, Algorithm 1]; (2) multidimensional scaling (MDS) based algorithms [69–71]; (3) programming based algorithms [72, 73]; (4) iterative parallel projection method (IPPM) based algorithms [74–76]; (5) expectation-maximization (EM) based algorithms [77–80]. Whereas in the Bayesian algorithms, unknown (true) positions are assumed to be random variables with certain prior distributions. Representative Bayesian algorithms include the nonparametric belief propagation (NBP) algorithm [81], sum-product-algorithm over wireless networks (SPAWN) algorithm [14], and some new variations [82–85] built upon them. They all perform message passing by taking advantages of the belief propagation algorithm [86] or the sum-product algorithm [87] in different graphical models. Restricted by the ad-hoc nature of WSNs, distributed cooperative localization (or self-localization) algorithms are highly demanded. This is owing to their advantageous features of being scalable, independent of a fusion center, and less

sensitive to sensor failure as compared to the centralized solutions [88]. In the distributed non-Bayesian algorithms, wireless sensors exchange their position estimates mutually; while in the conventional, distributed Bayesian algorithms, they exchange local belief messages (distributions represented by a set of particles) about their own true positions and consume much more energy for wireless communication. At the sacrifice of localization accuracy, however, some recent work demonstrated that the communication overhead can be significantly reduced by using transmit- and receive censoring [85] and parametric representation of the local belief messages [84, 89].

Among the listed algorithms, the least-squares estimation based algorithms, the MDS based algorithms, and the EM based algorithms are independent of offline calibration and NLOS identification. However, the first two classes of algorithms are extremely sensitive to the outliers induced by the NLOS propagation. In [77], an EM algorithm was proposed for outlier compensation but not in the context of NLOS mitigation. We proposed several centralized algorithms in [78] (for RSS model in (1.4)) and [79] (for TOA model in (1.3) that extend [77] for NLOS mitigation without using offline calibration and NLOS identification. The most recent work in [80] further developed a series of centralized- and distributed ECM algorithms for TOA based cooperative localization in WSNs.

1.6 Thesis Outline and Contributions

The focus of this thesis is to investigate NLOS mitigation in harsh mixed LOS/NLOS environments from a statistical signal processing perspective. To keep a good consistency, only the TOA based localization algorithms will be introduced in this thesis. This section introduces the organization of this thesis and highlights the main findings of each chapter.

In Chapter 2, the background of the maximum likelihood estimation is briefly introduced. Then, the expectation-maximization (EM), expectation-conditional maximization (ECM), and joint maximum *a posteriori*-maximum likelihood (JMAP-ML) algorithms are introduced, that can be adopted to tackle the difficulties in the incomplete-data situations. These serve as the basics of the following chapters.

In Chapter 3, TOA based non-cooperative localization is considered in harsh mixed LOS/NLOS environments with unknown measurement error distribution $p_V(v)$. A nonparametric approach to the modeling of $p_V(v)$ is employed. This work finds applications primarily in cellular radio networks because of the relatively high computational

complexity. The original contributions, which lead to one conference contribution [27] and one journal contribution [65], are summarized as follows:

- A robust iterative nonparametric (RIN) algorithm has been developed, which alternates between a nonparametric pdf estimation step and a position estimation step. Starting with a carefully selected initial position estimate, an estimate of the true measurement error distribution is first constructed via adaptive kernel density estimation (AKDE) [37]. An approximated log-likelihood function is then formulated, from which a refined position estimate is resolved via a quasi-Newton (QN) method. These two steps are repeated as necessary.
- The best achievable localization accuracy has been presented in terms of Cramér-Rao lower bound (CRLB), which serves as a benchmark for evaluating different localization algorithms.

In Chapter 4, the localization problem in Chapter 3 is re-considered. To reduce the computational complexity, a parametric approach to the modeling of the measurement error distribution is adopted instead. This work finds applications in both cellular radio networks and wireless sensor networks due to the relatively low computational complexity. The original contributions, which lead to one journal publication [67], are summarized as follows:

- Two iterative algorithms have been developed based on the well-known ECM criterion and JMAP-ML criterion to approximate the ideal maximum likelihood estimator of the unknown parameters, including position and mixture model parameters.
- Convergence analysis and complexity analysis of the proposed algorithms have been shown with concrete examples.

Although not introduced in this thesis, an EM based non-cooperative localization algorithm has been developed based on the mode-dependent RSS model in (1.3), which leads to one conference contribution [66].

In Chapter 5, TOA based cooperative localization is studied in harsh mixed LOS/NLOS environments. The parametric approach to the modeling of the measurement error distribution is again adopted. The original contributions, which lead to one conference contribution [79] and one journal contribution [80], are summarized as follows:

- A series of centralized- and distributed ECM algorithms have been developed to approximate the ML estimator of the unknown parameters.
- The proposed algorithms have been evaluated in terms of computational complexity and communication overhead.
- The best achievable localization accuracy has been presented in terms of CRLB (with possibly any distribution), which generalizes the results in [19] (valid merely for the Gaussian model).

Although not introduced in this thesis, an EM based cooperative localization algorithm has been developed based on the mode-dependent RSS model in (1.3), which leads to one conference contribution [78].

Finally, Chapter 6 concludes this thesis and shortly summarizes some ongoing work.

For a better view of the main findings of each chapter and their connections, Figure 1.5 is depicted below.

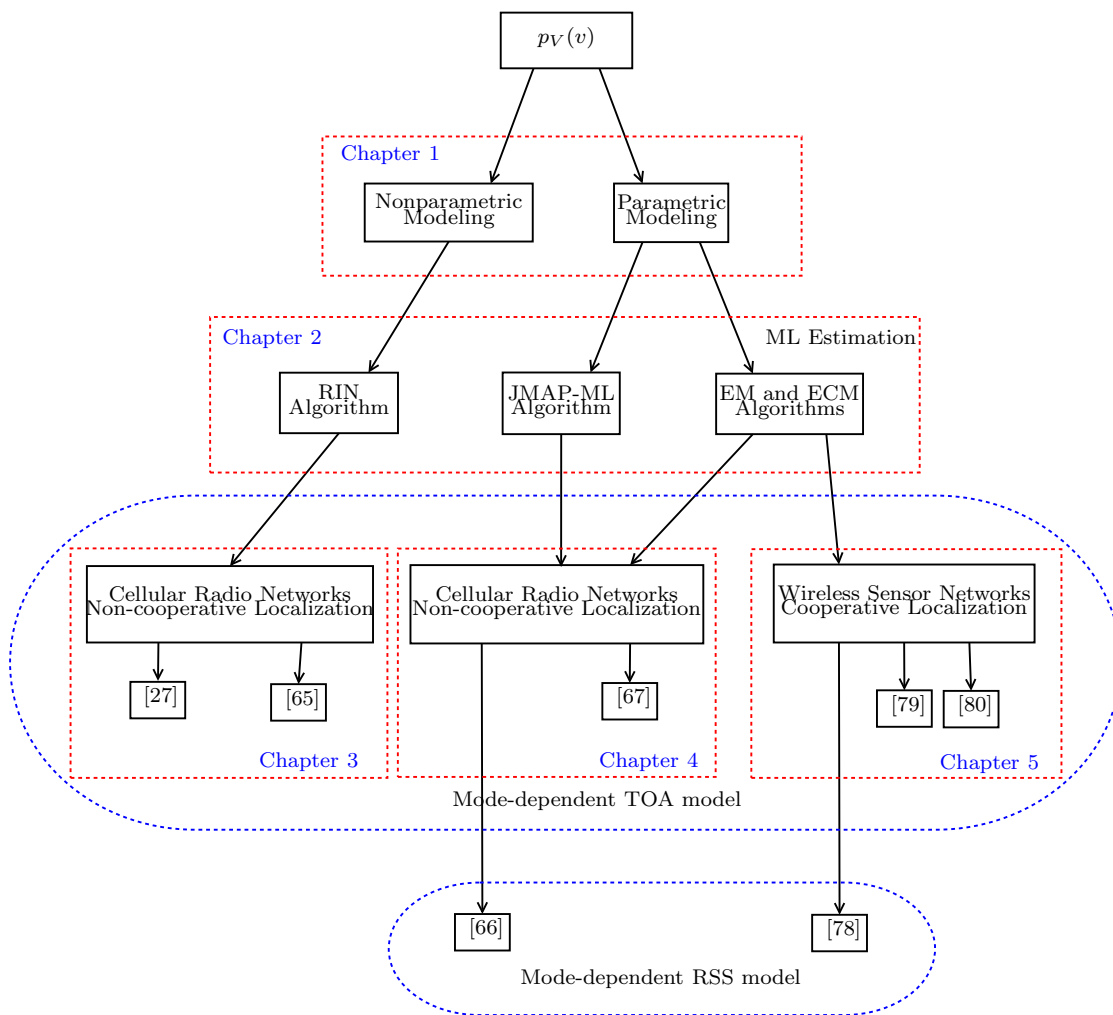


Figure 1.5. Main findings of each chapter and their connections.

Chapter 2

Background

This chapter serves as the cornerstone of this thesis. Section 2.1 briefly revisits several well-known optimality criteria for developing classical estimators of unknown deterministic parameters. Therein, the important statements and results are taken from [8, 90–92] without proof. Section 2.2 briefly revisits several alternative maximum likelihood estimation techniques that are easier to use in incomplete-data situations. Therein, the important statements and results are taken from [93–97] without proof.

2.1 Overview of Classical Parameter Estimation

Statistical parameter estimation plays an important role in many electronic signal processing systems, as it can extract useful information from a batch of noise corrupted measurements [90]. In this chapter, the class of estimators of unknown deterministic parameters is mainly considered. In the literature, they are commonly referred to as the “classical” estimators. The overview of different optimality criteria starts from the most natural choice, i.e., the minimum-mean-square-error (MMSE) criterion. The MMSE estimator, as its name suggests, minimizes the mean-square-error (MSE), which is defined by

$$\text{MSE}(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ \|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\|^2 \right\} \quad (2.1)$$

where the parameter is a real vector $\boldsymbol{\theta} \in \Theta$ on \mathbb{R}^d , and $\hat{\boldsymbol{\theta}}$ is an estimator of $\boldsymbol{\theta}$. Essentially, $\hat{\boldsymbol{\theta}}$ is a function of observations, namely, $\hat{\boldsymbol{\theta}} = \mathbf{f}(\mathbf{r}) = \mathbf{f}(r_1, r_2, \dots, r_N)$, but for brevity the dependency of any estimator on the observations is ignored in the sequel. The expectation is taken with respect to the probability density function (pdf) $p(\mathbf{r}; \boldsymbol{\theta})$. The MSE expression in (2.1) can be rewritten as

$$\begin{aligned} \text{MSE}(\hat{\boldsymbol{\theta}}) &= \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ \sum_{i=1}^d (\hat{\theta}_i - \theta_i)^2 \right\} \\ &= \text{tr} \left\{ \text{Cov}(\hat{\boldsymbol{\theta}}) \right\} + \|\text{Bias}(\hat{\boldsymbol{\theta}})\|^2 \end{aligned} \quad (2.2)$$

where $\hat{\theta}_i$ is the i th entry of $\hat{\boldsymbol{\theta}}$, $\text{Cov}(\hat{\boldsymbol{\theta}})$ is the covariance matrix of $\hat{\boldsymbol{\theta}}$ defined by

$$\text{Cov}(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ (\hat{\boldsymbol{\theta}} - \mathbb{E}\{\hat{\boldsymbol{\theta}}\})(\hat{\boldsymbol{\theta}} - \mathbb{E}\{\hat{\boldsymbol{\theta}}\})^T \right\}, \quad (2.3)$$

and $\text{Bias}(\hat{\boldsymbol{\theta}})$ is the bias of $\hat{\boldsymbol{\theta}}$ defined by

$$\text{Bias}(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \{\hat{\boldsymbol{\theta}}\} - \boldsymbol{\theta}. \quad (2.4)$$

Since $\text{Bias}(\hat{\boldsymbol{\theta}})$ is a function of the unknown (true) parameter $\boldsymbol{\theta}$, the MMSE estimator is in general unrealizable [90].

Alternatively, the minimum variance unbiased (MVU) estimator can be adopted. The MVU estimator must satisfy for all $\boldsymbol{\theta} \in \Theta$ that (i) $\hat{\boldsymbol{\theta}}$ is unbiased and (ii) $\text{var}(\hat{\theta}_i) \leq \text{var}(\hat{\theta}'_i)$, $i = 1, 2, \dots, d$, holds for any other unbiased estimator $\hat{\boldsymbol{\theta}}'$. In general, given an unbiased estimator with computed variances, $\text{var}(\hat{\theta}_i)$, $i = 1, 2, \dots, d$, it is still difficult to determine whether it is an MVU estimator or not. In the special case where the variance of each entry of $\hat{\boldsymbol{\theta}}$ equals the corresponding Cramér-Rao lower bound (CRLB), we can immediately tell that $\hat{\boldsymbol{\theta}}$ is an MVU estimator. This idea leads to a powerful approach for finding an MVU estimator in linear measurement model. Before this approach is given in details, the CRLB theorem for vector parameter case [90] is first reviewed in the following.

Theorem 2.1. *When the regularity conditions [8, C.2] are all fulfilled, the covariance matrix of any unbiased estimator $\hat{\boldsymbol{\theta}}$ satisfies*

$$\text{Cov}(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T \right\} \succeq \text{CRLB}(\boldsymbol{\theta}) = \mathcal{F}^{-1}(\boldsymbol{\theta}) \quad (2.5)$$

where $\mathcal{F}(\boldsymbol{\theta})$ denotes the Fisher's information matrix (FIM) and is defined by

$$\mathcal{F}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ -\Delta_{\boldsymbol{\theta}}^{\theta} \ln p(\mathbf{r}; \boldsymbol{\theta}) \right\}. \quad (2.6)$$

The definitions of the gradient operator $\nabla_{\boldsymbol{\theta}}$ and Laplace operator $\Delta_{\boldsymbol{\theta}}^{\theta} = \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\theta}}^T$ are given in Appendix 2.3.1.

Theorem 2.2. *An MVU estimator $\hat{\boldsymbol{\theta}} = \mathbf{f}(\mathbf{r})$ may be found that attains the bound in that $\text{Cov}(\hat{\boldsymbol{\theta}}) = \text{CRLB}(\boldsymbol{\theta})$ if and only if*

$$\frac{\partial \ln p(\mathbf{r}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \mathcal{F}(\boldsymbol{\theta})(\mathbf{f}(\mathbf{r}) - \boldsymbol{\theta}) \quad (2.7)$$

for some d -dimensional function $\mathbf{f}(\mathbf{r}) : \mathbb{R}^N \mapsto \mathbb{R}^d$.

Unfortunately for nonlinear measurement models shown in Chapter 1, which is the sole focus of this thesis, an MVU estimator is generally hard to derive even if it does exist. Alternatively, the estimator based on the maximum likelihood (ML) criterion, termed

the ML estimator, is easier to derive even for complicated estimation problems [90]. Most importantly, the ML estimator is asymptotically efficient and becomes the MVU estimator as the number of measurements, N , goes to infinity.

The ML estimator of $\boldsymbol{\theta} \in \Theta$ is defined as

$$\hat{\boldsymbol{\theta}}^{\text{ML}} = \arg \max_{\boldsymbol{\theta}} p(\mathbf{r}; \boldsymbol{\theta}), \quad (2.8)$$

i.e., the value that globally maximizes the joint pdf for all observations, $p(\mathbf{r}; \boldsymbol{\theta})$, which is also known as the likelihood function. The idea behind the ML criterion is that given a set of observations \mathbf{r} , some values of $\boldsymbol{\theta}$ are more probable to have generated \mathbf{r} than the others, which can be easily recognized from plotting $p(\mathbf{r}; \boldsymbol{\theta})$ over all $\boldsymbol{\theta} \in \Theta$. Usually, it is more convenient to work with the log-likelihood function, defined as $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r}) = \ln p(\mathbf{r}; \boldsymbol{\theta})$ throughout this thesis, as logarithm is a monotonic operation.

Asymptotic property and invariance property of the ML estimator were given in [90]. Before these results are revisited, the following two definitions [91] are given in the first place for better exposition.

Definition 2.1. *An estimator is said to be efficient when it is unbiased and its covariance matrix equals the CRLB.*

Definition 2.2. *An estimator $\hat{\boldsymbol{\theta}} = \mathbf{f}(r_1, r_2, \dots, r_N)$ is said to be consistent if*

$$\lim_{N \rightarrow \infty} \Pr\{|\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}|^2 \geq \epsilon\} = 0, \quad \forall \epsilon > 0. \quad (2.9)$$

Theorem 2.3. *When $p(\mathbf{r}; \boldsymbol{\theta})$ fulfills the regularity conditions [8, C.5] and the number of parameters, d , to be estimated is much less than the number of measurements, N , the ML estimator of the unknown parameter $\boldsymbol{\theta}$ is in general asymptotically distributed according to*

$$\hat{\boldsymbol{\theta}} \stackrel{a}{\sim} \mathcal{N}(\boldsymbol{\theta}, \mathcal{F}^{-1}(\boldsymbol{\theta})) \quad (2.10)$$

where $\boldsymbol{\theta}$ is the true value of the unknown parameter and $\mathcal{F}^{-1}(\boldsymbol{\theta})$ is evaluated at the true value $\boldsymbol{\theta}$ in this context.

Theorem 2.4. *The ML estimator of the parameter $\boldsymbol{\tau} = \mathbf{g}(\boldsymbol{\theta})$, when $\mathbf{g}(\boldsymbol{\theta}) : \mathbb{R}^d \mapsto \mathbb{R}^p$ is an invertible function, is given by*

$$\hat{\boldsymbol{\tau}} = \mathbf{g}(\hat{\boldsymbol{\theta}}) \quad (2.11)$$

where $\hat{\boldsymbol{\theta}}$ is the ML estimator of $\boldsymbol{\theta}$. Moreover,

$$\hat{\boldsymbol{\tau}} \stackrel{a}{\sim} \mathcal{N}\left(\mathbf{g}(\boldsymbol{\theta}), \left(\frac{\partial \mathbf{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)^T \mathcal{F}^{-1}(\boldsymbol{\theta}) \left(\frac{\partial \mathbf{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\right)\right). \quad (2.12)$$

Some remarks are given as follows:

- When $N \gg d$, the ML estimator is asymptotically efficient according to the **Definition 2.1**.
- When all entries of $\mathcal{F}(\boldsymbol{\theta})$ are proportional to N , the ML estimator is asymptotically consistent according to the **Definition 2.2**.

Owing to the nice properties, almost all practical estimators have been developed in light of the maximum likelihood criterion [90]. However, implementing an ML estimator is not always an easy task, and we may confront various difficulties in the practice. We list some in the following.

- An ML estimate can be found from (2.8) only numerically, for instance, via the grid search, Newton-type methods, or expectation-maximization (EM) algorithm [90], and the point found might be a local maximum or a saddle point instead of the global maximum if $p(\mathbf{r}; \boldsymbol{\theta})$ is multi-modal.
- The maximization problem in (2.8) is cumbersome, meaning that a lot of computational efforts are required to compute the ML estimator.
- The chosen pdf $p(\mathbf{r}; \boldsymbol{\theta})$ is a poor model of the actual distribution that gave rise to the observations [92].
- More serious problems would occur when $p(\mathbf{r}; \boldsymbol{\theta})$ is completely unknown or only known to a certain extent, cf. Chapter 1.

2.2 ML Estimation with Incomplete Data

Finding an ML estimator with conventional implementation might be complicated in incomplete-data situations, where there are missing data, truncated distributions, or grouped observations [96]. In the sequel, we provide several advanced algorithms that can be applied for approximating the ML estimator with less computational hurdles.

The notations introduced in the previous section are reused here, namely, let $\mathbf{r} = [r_1, r_2, \dots, r_N]$ be a set of observations having pdf $p(\mathbf{r}; \boldsymbol{\theta})$ where $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d]$. Additionally, a vector \mathbf{y} is introduced to denote the missing data and a vector \mathbf{z} with $\mathbf{z} = \{\mathbf{r}, \mathbf{y}\}$ is introduced to denote the complete data. The complete-data log-likelihood function is defined by $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{r}, \mathbf{y}) = \ln p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta})$. In contrast, $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ is referred to as the incomplete-data log-likelihood function in the sequel.

2.2.1 EM Algorithm

The expectation-maximization (EM) algorithm is a general-purpose algorithm for ML estimation in incomplete-data situations [96]. Since the appearance of the seminal paper [93], the EM algorithm has found a plethora of applications in the literature. The EM algorithm is an iterative approach, and, as its name suggests, it performs an expectation (E)-step and a maximization (M)-step on each iteration. As compared to the conventional ML estimation that adopts for instance Newton-type numerical methods, the EM algorithm can tackle incomplete-data problems in a computationally profitable manner. For clarity, the key steps for computing an EM estimate on the $(\eta + 1)$ th iteration are summarized in Algorithm 2.1.

Algorithm 2.1 EM Algorithm (General Routine) [93]

Step 1—Initialization:

Set the iteration index $\eta = 0$. Choose an initial guess $\boldsymbol{\theta}^{(0)}$. Choose a convergence tolerance Δ .

Step 2—EM stage:

On the $(\eta + 1)$ th iteration ($\eta \in \mathbb{Z}, \eta \geq 0$), do:

- **E-step:** Perform conditional expectation of the complete-data log-likelihood function in terms of \mathbf{y} given \mathbf{r} and $\boldsymbol{\theta}^{(\eta)}$ and obtain a Q -function

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)})} \{ \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) \}. \quad (2.13)$$

- **M-step:** Find $\boldsymbol{\theta}^{(\eta+1)}$ that globally maximizes $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$.

Step 3—Convergence Check:

If $\mathcal{L}_I(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}) - \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}) \leq \Delta$, then terminate this algorithm and obtain $\hat{\boldsymbol{\theta}}^{\text{EM}} = \boldsymbol{\theta}^{(\eta+1)}$; otherwise set $\eta \leftarrow \eta + 1$ and return to **Step 2**.

Note that the convergence condition in Algorithm 2.1 is due to the nice property proven in [93]; that is, the incomplete-data log-likelihood is monotonically increased after each EM iteration, more precisely,

$$\mathcal{L}_I(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}) \geq \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}), \quad \eta = 0, 1, 2, \dots \quad (2.14)$$

Therefore, when $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ is bounded above, the sequence of incomplete-data log-likelihood values $\{ \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}) \}$ would converge to some value L^* .

The overwhelming drawback of the EM algorithm is that the global maximizer is often difficult to obtain in the M-step. To tackle this problem, a generalized EM (GEM) algorithm was proposed in [93]. The GEM algorithm is revisited in Algorithm 2.2.

Algorithm 2.2 GEM Algorithm (General Routine) [93]

Step 1—Initialization:

Set the iteration index $\eta = 0$. Choose an initial guess $\boldsymbol{\theta}^{(0)}$. Choose a convergence tolerance Δ .

Step 2—GEM Stage:

On the $(\eta + 1)$ th iteration ($\eta \in \mathbb{Z}, \eta \geq 0$), do:

- **E-step:** Perform conditional expectation of the complete-data log-likelihood function in terms of \mathbf{y} given \mathbf{r} and $\boldsymbol{\theta}^{(\eta)}$ and obtain

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)})} \{ \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) \}. \quad (2.15)$$

- **M-step:** Find $\boldsymbol{\theta}^{(\eta+1)}$ such that

$$Q(\boldsymbol{\theta}^{(\eta+1)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}^{(\eta)}; \boldsymbol{\theta}^{(\eta)}). \quad (2.16)$$

Step 3—Convergence Check:

If $\mathcal{L}_I(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}) - \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}) \leq \Delta$, then terminate this algorithm and obtain $\hat{\boldsymbol{\theta}}^{\text{GEM}} = \boldsymbol{\theta}^{(\eta+1)}$; otherwise set $\eta \leftarrow \eta + 1$ and return to **Step 2**.

Simply speaking, the GEM algorithm modifies the M-step so that $\boldsymbol{\theta}^{(\eta+1)}$ is found to increase the Q -function over its value evaluated at the prior estimate $\boldsymbol{\theta}^{(\eta)}$ [96]. Clearly, the EM algorithm is a special case of the GEM algorithm. The property in (2.14) also holds for the GEM algorithm. Under some regularity conditions [94], Wu gave the following two theorems to describe the convergence of the incomplete-data log-likelihood values $\{\mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r})\}$ to some stationary point L^* satisfying that $L^* = \mathcal{L}_I(\boldsymbol{\theta}^*; \mathbf{r})$ and $\partial \mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r}) / \partial \boldsymbol{\theta}|_{\boldsymbol{\theta}=\boldsymbol{\theta}^*} = \mathbf{0}$.

Theorem 2.5. Let $\{\boldsymbol{\theta}^{(\eta)}\}$ be a GEM sequence generated by $\boldsymbol{\theta}^{(\eta+1)} = \mathcal{M}(\boldsymbol{\theta}^{(\eta)})$, and suppose that (i) $\mathcal{M}(\cdot)$ is a closed point-to-set map over the complement of \mathcal{S} with

$$\mathcal{S} = \text{set of stationary points in the interior of } \Theta, \quad (2.17)$$

and (ii)

$$\mathcal{L}_I(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}) \geq \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}), \quad \text{for all } \boldsymbol{\theta}^{(\eta)} \notin \mathcal{S}. \quad (2.18)$$

Then all the limit points of $\{\boldsymbol{\theta}^{(\eta)}\}$ are stationary points, and $\{\mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r})\}$ converges monotonically to $L^* = \mathcal{L}_I(\boldsymbol{\theta}^*; \mathbf{r})$ for some stationary points $\boldsymbol{\theta}^* \in \mathcal{S}$.

Theorem 2.6. Suppose that the Q -function $Q(\boldsymbol{\theta}; \boldsymbol{\phi})$ satisfies the continuous condition:

$$Q(\boldsymbol{\theta}; \boldsymbol{\phi}) \text{ is continuous in both } \boldsymbol{\theta} \text{ and } \boldsymbol{\phi}. \quad (2.19)$$

Then all the limit points of $\{\boldsymbol{\theta}^{(\eta)}\}$ of an EM algorithm are stationary points of $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$, and $\{\mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r})\}$ converges monotonically to $L^* = \mathcal{L}_I(\boldsymbol{\theta}^*; \mathbf{r})$ for some stationary points $\boldsymbol{\theta}^*$.

Remark 2.1. For the EM algorithm, $\mathcal{M}(\boldsymbol{\theta}^{(n)})$ defines the point-to-set map:

$$\mathcal{M}(\boldsymbol{\theta}^{(n)}) = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(n)}). \quad (2.20)$$

In other words, $\mathcal{M}(\boldsymbol{\theta}^{(n)})$ is the set of global maximizer of the Q -function. For the GEM algorithm, $\mathcal{M}(\boldsymbol{\theta}^{(n)})$ is defined to be the set of all $\boldsymbol{\theta}^{(n+1)}$ satisfying (2.16).

Note that the condition in **Theorem 2.6** is easy to verify in almost all applications, while the two conditions in **Theorem 2.5** are, in general, hard to verify [94]. But note that it is uttermost important that the sequence of log-likelihood values $\{\mathcal{L}_I(\boldsymbol{\theta}^{(n)}; \mathbf{r})\}$ of both the EM algorithm and GEM algorithm monotonically increase to some point and $\{\boldsymbol{\theta}^{(n)}\}$ is an EM sequence or a GEM sequence. In [96], McLachlan and Krishnan pointed out that, almost in all applications, the sequence of log-likelihood values $\{\mathcal{L}_I(\boldsymbol{\theta}^{(n)}; \mathbf{r})\}$ would ultimately converge to a stationary point. Only in rare cases, a fixed point of the algorithm will be achieved. In [94], Wu also pointed out that the convergence to either kind of stationary point (e.g., saddle point, local optimum, or global optimum) depends on the choice of initial guess $\boldsymbol{\theta}^{(0)}$.

2.2.2 ECM Algorithm

Both the EM algorithm and GEM algorithm are unattractive when the complete-data maximum likelihood estimation is still cumbersome. The expectation-conditional maximization (ECM) algorithm, which was proposed by Meng and Rubin in [95], can be adopted for such difficult situation. The idea is to replace a complicated maximization step in the EM algorithm with several computationally simpler conditional maximization (CM) steps in the ECM algorithm. The ECM algorithm typically requires more iterations as compared to the EM algorithm, and a single CM step might also involve iterations if closed form solution can not be found. But the total computational time of the ECM algorithm might be faster, because the conditional maximization problems therein handles only lower dimensional searches that can be solved faster, more efficiently, and with higher stability [96].

Similar to the previous subsections, the key steps of the ECM algorithm are summarized in Algorithm 2.3. Some remarks are in order.

- Both the variety and complexity of the CM steps depend on the partition of $\boldsymbol{\theta}$.

Algorithm 2.3 ECM Algorithm (General Routine) [95]

Step 1—Initialization:

Set the iteration index $\eta = 0$. Choose an initial guess $\boldsymbol{\theta}^{(0)}$. Choose a convergence tolerance Δ .

Step 2—ECM Stage:

On the $(\eta + 1)$ th iteration ($\eta \in \mathbb{Z}, \eta \geq 0$), do:

- **E-step:** Performing conditional expectation, exactly as in the previous algorithms, yields $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$.
- **CM-steps:**
 1. Find a proper partition of the unknown parameters, $\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \dots, \boldsymbol{\vartheta}_S^T]^T$, where $\boldsymbol{\vartheta}_s$, $s = 1, 2, \dots, S$ is a sub-vector of $\boldsymbol{\theta}$.
 2. Select a set of S vector functions of $\boldsymbol{\theta}$, namely,

$$g_s(\boldsymbol{\theta}) = [\boldsymbol{\vartheta}_1^T, \dots, \boldsymbol{\vartheta}_{s-1}^T, \boldsymbol{\vartheta}_{s+1}^T, \dots, \boldsymbol{\vartheta}_S^T]^T. \quad (2.21)$$

3. For $s = 1, 2, \dots, S$, solve $\boldsymbol{\theta}^{(\eta+s/S)}$ sequentially via

$$\boldsymbol{\theta}^{(\eta+s/S)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}), \quad (2.22)$$

subject to the constraint $g_s(\boldsymbol{\theta}) = g_s(\boldsymbol{\theta}^{(\eta+(s-1)/S)})$.

4. Obtain $\boldsymbol{\theta}^{(\eta+1)} = \boldsymbol{\theta}^{(\eta+S/S)}$ after the final CM step.

Step 3—Convergence Check:

If $\mathcal{L}_I(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}) - \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}) \leq \Delta$, then terminate this algorithm and obtain $\hat{\boldsymbol{\theta}}^{\text{ECM}} = \boldsymbol{\theta}^{(\eta+1)}$; otherwise reset $\eta \leftarrow \eta + 1$ and return to **Step 2**.

- On the s th CM-step of the η th ECM iteration, the Q -function is maximized in an attempt to update $\boldsymbol{\vartheta}_s^{(\eta+1)}$ with the other subvectors held fixed at their current values.
- The ECM algorithm is a member of the GEM algorithm, because $Q(\boldsymbol{\theta}^{(\eta+1)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}^{(\eta)}; \boldsymbol{\theta}^{(\eta)})$ holds for every η .
- According to [96], almost all the convergence properties established for the EM algorithm in [93, 94] hold for the ECM algorithm as well when:
 1. $g_s(\boldsymbol{\theta})$ is differentiable;
 2. $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta})$ is of full rank at $\boldsymbol{\theta}^{(\eta)} \in \Theta$, for all η ;
 3. The “space filling” condition $\bigcap_{s=1}^S G_s(\boldsymbol{\theta}^{(\eta)}) = \{\mathbf{0}\}$ holds for η , where $G_s(\boldsymbol{\theta})$ is the column space of the matrix $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta})$.

- Simply speaking, the “space filling” condition tells us that the maximization is done over the whole parameter space Θ rather than a subspace of it.

2.2.3 JMAP-ML Algorithm

When a set of missing data is taken into account to facilitate the classical ML estimation problem, the complete-data likelihood relates to the incomplete-data likelihood through

$$p(\mathbf{r}; \boldsymbol{\theta}) = \int p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}) d\mathbf{y}. \quad (2.23)$$

As another means of approximating the ML estimator, the complete-data log-likelihood function is maximized directly with respect to both $\boldsymbol{\theta}$ and \mathbf{y} rather than performing an integration with respect to \mathbf{y} , that is,

$$\arg \max_{\boldsymbol{\theta}, \mathbf{y}} \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) \equiv \arg \max_{\boldsymbol{\theta}} \left\{ \arg \max_{\mathbf{y}} \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) \right\}. \quad (2.24)$$

This criterion is called the joint maximum *a posteriori*-maximum likelihood (JMAP-ML) criterion [97]. This is due to the fact that it incorporates a maximum *a posteriori* (MAP) estimation step in terms of the latent variables in \mathbf{y} and an ML estimation step in terms of the deterministic parameters in $\boldsymbol{\theta}$. In general, an algorithm developed based on the JMAP-ML criterion starts with a carefully selected initial guess $\boldsymbol{\theta}^{(0)}$ and alternates between the above mentioned two steps in an iterative process. The general routine is given in Algorithm 2.4. The JMAP-ML algorithm is less popular than the EM-type algorithm in that it only produces a biased estimator, but this bias estimator may generate better MSE as compared to the unbiased estimators [97].

2.2.4 Connections

In the previous subsections, several salient algorithms are briefly revisited that can be used to facilitate the conventional ML estimation in incomplete-data situations. They all work with complete data, more precisely, with the complete-data likelihood function. It is easy to see that the incomplete-data likelihood function $p(\mathbf{r}; \boldsymbol{\theta})$ is the marginalization of the complete-data likelihood function $p(\mathbf{y}, \mathbf{r}; \boldsymbol{\theta})$ in terms of the latent variables \mathbf{y} . Replacing this integration (with respect to \mathbf{y}) with a conditional expectation given the current parameter estimate (E-step) or a direct maximization with respect to \mathbf{y} given the current parameter (MAP step), an approximation of $p(\mathbf{r}; \boldsymbol{\theta})$ is obtained from which a parameter estimate is much easier to solve as compared to the conventional ML implementation. Figure 2.1 demonstrates the connections between these algorithms.

Algorithm 2.4 JMAP-ML Algorithm (General Routine) [97]

Step 1—Initialization:

Set the iteration index $\eta = 0$. Choose an initial guess $\boldsymbol{\theta}^{(0)}$. Choose a convergence tolerance Δ .

Step 2—Joint MAP-ML stage:

On the $(\eta + 1)$ th iteration ($\eta \in \mathbb{Z}, \eta \geq 0$), do:

- **MAP-step:** Find an MAP estimate of $\mathbf{y}, \mathbf{y}^{(\eta+1)}$, where

$$\mathbf{y}^{(\eta+1)} = \arg \max_{\mathbf{y}} \mathcal{L}_C(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}, \mathbf{y}). \quad (2.25)$$

- **ML-step:** Find a complete-data ML estimate of $\boldsymbol{\theta}, \boldsymbol{\theta}^{(\eta+1)}$, where

$$\boldsymbol{\theta}^{(\eta+1)} = \arg \max_{\boldsymbol{\theta}} \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{r}, \mathbf{y}^{(\eta+1)}). \quad (2.26)$$

Step 3—Convergence Check:

If $\mathcal{L}_C(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}, \mathbf{y}^{(\eta+1)}) - \mathcal{L}_C(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}, \mathbf{y}^{(\eta)}) \leq \Delta$, then terminate this algorithm and obtain $\hat{\boldsymbol{\theta}}^J = \boldsymbol{\theta}^{(\eta+1)}$; otherwise set $\eta \leftarrow \eta + 1$ and return to **Step 2**.

2.3 Appendix

2.3.1 Gradient and Laplace Operators

The gradient of a vector function $\mathbf{g}(\boldsymbol{\theta}) : \mathbb{R}^d \mapsto \mathbb{R}^p$ with $\boldsymbol{\theta} = [\theta_1, \theta_2, \dots, \theta_d]^T$ is defined as

$$\nabla_{\boldsymbol{\theta}} \mathbf{g}(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial g_1(\boldsymbol{\theta})}{\partial \theta_1} & \frac{\partial g_2(\boldsymbol{\theta})}{\partial \theta_1} & \cdots & \frac{\partial g_p(\boldsymbol{\theta})}{\partial \theta_1} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial g_1(\boldsymbol{\theta})}{\partial \theta_d} & \frac{\partial g_2(\boldsymbol{\theta})}{\partial \theta_d} & \cdots & \frac{\partial g_p(\boldsymbol{\theta})}{\partial \theta_d} \end{pmatrix}, \quad (2.27)$$

and the Laplace of a scalar function $g(\boldsymbol{\theta}) : \mathbb{R}^d \mapsto \mathbb{R}$ is defined as

$$\Delta_{\boldsymbol{\theta}}^{\theta} g(\boldsymbol{\theta}) = \begin{pmatrix} \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \theta_1^2} & \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \theta_1 \partial \theta_d} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \theta_d \partial \theta_1} & \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \theta_d \partial \theta_2} & \cdots & \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \theta_d^2} \end{pmatrix}. \quad (2.28)$$

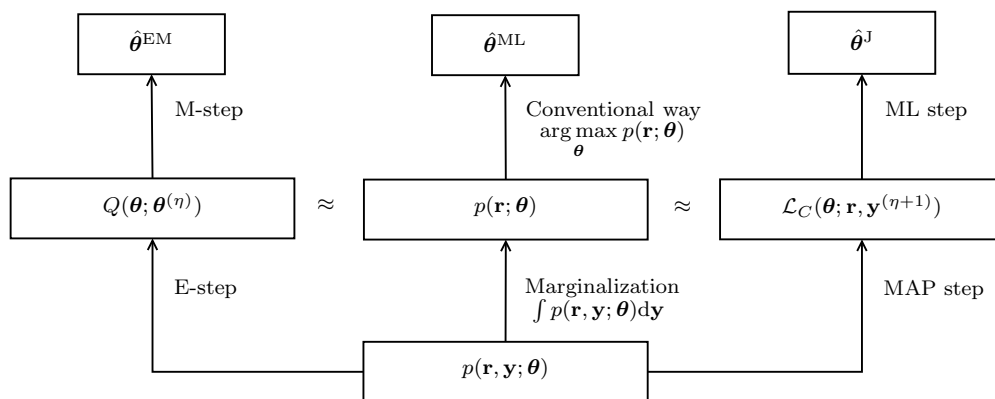


Figure 2.1. Connections between the conventional ML estimation, EM estimation and JMAP-ML estimation.

Chapter 3

Localization in Cellular Radio Networks: Nonparametric Modeling

In the past decade, wireless localization systems are mainly considered in cellular radio networks [1]. Among a myriad of applications, emergency services like E-911 [98] in the USA, are more demanding. This requires a localization system to be able to locate emergency calls accurately and rapidly in different scenarios including, certainly, those harsh ones. In this chapter, time-of-arrival (TOA) based robust localization is considered in harsh mixed line-of-sight (LOS)/non-line-of-sight (NLOS) environments as sketched in Section 1.4. Herein, the probability density function (pdf) of the measurement error (or measurement error distribution) is assumed to be completely unknown. An iterative algorithm, called robust iterative nonparametric (RIN) algorithm, has been developed for NLOS mitigation under the harsh conditions assumed in Chapter 1.

This chapter is organized as follows. Section 3.1 introduces the signal model and states the problem at hand. Section 3.2 revisits an existing robust position estimation algorithm and further introduces the proposed RIN algorithm. Section 3.3 computes the Cramér-Rao lower bound and Section 3.4 introduces several theoretical performance metrics for evaluating an unbiased position estimator. Section 3.5 shows some simulation results. Conclusions are drawn in Section 3.6. Finally, Section 3.7 assembles some important definitions and derivations.

3.1 Signal Model and Problem Statement

Consider the scenario where $N(N \geq 3)$ base stations (BSs) surround a stationary mobile station (MS) of interest in a wireless cellular radio network. Let $\mathbf{p}_i = [x_i, y_i]^T$ denote the *a priori* known position of the i th BS, $i = 1, 2, \dots, N$, and let $\boldsymbol{\theta}_p = [x, y]^T$ denote the unknown MS position. For each BS, a number of $K(K \geq 1)$ distance measurement(s) (time-of-arrival estimate(s) multiplied by c_0) are obtained and subsequently relayed to a fusion center for post-processing [99]. Figure 3.1 illustrates such a scenario and explains the notations therein.

Assuming a precise time synchronization between the BSs and MS, the k th distance

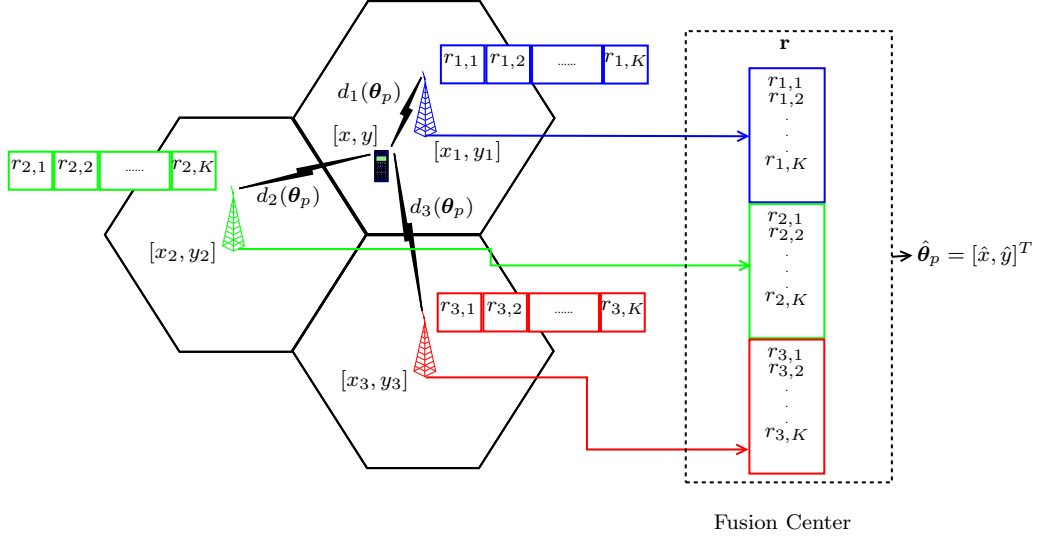


Figure 3.1. An illustrating example of wireless localization in a cellular radio network in a 2-D space. In this example, there are $N = 3$ BSs and one stationary MS. Each BS (say the i th) collects a total number of K distance measurements, $r_{i,k}$, $k = 1, 2, \dots, K$ and transmits them terrestrially to a fusion center, where a localization algorithm will be run to give an estimate of the MS position of interest, $\hat{\theta}_p$, based on \mathbf{r} .

measurement $r_{i,k}$ measured at the i th BS can be expressed by

$$r_{i,k} = \underbrace{\sqrt{(x - x_i)^2 + (y - y_i)^2}}_{d_i(\theta_p)} + v_{i,k}, \quad (3.1)$$

$k = 1, 2, \dots, K$, where $d_i(\theta_p)$ represents the actual Euclidean distance between the MS and the i th BS. Herein, $v_{i,k}$ is assumed to follow $p_V^{(L)}(v; \beta_L)$ under the LOS condition or $p_V^{(NL)}(v; \beta_{NL})$ under the NLOS condition. In the literature, $p_V^{(L)}(v; \beta_L)$ is usually modeled by a zero mean Gaussian distribution, whereas $p_V^{(NL)}(v; \beta_{NL})$ can be modeled by a shifted Gaussian distribution (e.g., in [3, 22, 27–32]) or a Rayleigh distribution (e.g., in [27, 32–34]) or an exponential distribution (e.g., in [25, 26, 32]), depending on the actual scenario for localization. However, it is stressed that the localization algorithm developed in the sequel does not assume any knowledge about the measurement error distribution.

For better readability, the signal model is expressed in a compact vector form as

$$\mathbf{r} = \mathbf{h}(\theta_p) + \mathbf{v} \quad (3.2)$$

where

$$\mathbf{r} = [r_{1,1}, \dots, r_{1,K}, \dots, r_{N,1}, \dots, r_{N,K}]^T, \quad (3.3)$$

$$\mathbf{h}(\boldsymbol{\theta}_p) = [\underbrace{d_1(\boldsymbol{\theta}_p), \dots, d_1(\boldsymbol{\theta}_p)}_{K \text{ repetitions}}, \dots, \underbrace{d_N(\boldsymbol{\theta}_p), \dots, d_N(\boldsymbol{\theta}_p)}_{K \text{ repetitions}}]^T, \quad (3.4)$$

$$\mathbf{v} = [v_{1,1}, \dots, v_{1,K}, \dots, v_{N,1}, \dots, v_{N,K}]^T. \quad (3.5)$$

The column vectors \mathbf{r} , $\mathbf{h}(\boldsymbol{\theta}_p)$ and \mathbf{v} are all of dimension $NK \times 1$. In the sequel, r_m , $h_m(\boldsymbol{\theta}_p)$ and v_m will represent the m th element of \mathbf{r} , $\mathbf{h}(\boldsymbol{\theta}_p)$ and \mathbf{v} , respectively, with $m = 1, 2, \dots, NK$. As explained in Section 1.4.1, we assume that the measurement error terms in \mathbf{v} are iid and follow a two-mode mixture distribution $p_V(v)$ as defined in (1.6). Throughout this chapter, $p_V(v)$ is assumed to be completely unknown, and $\boldsymbol{\theta}_a$ is null and thus discarded in $\mathbf{h}(\boldsymbol{\theta}_p)$.

3.2 Joint ML Estimation using KDE

The estimation performance that relies on maximizing the exact log-likelihood function generally degrades once the true measurement error distribution deviates from the assumed one. Since the measurement error distribution $p_V(v)$ is assumed to be completely unknown, finding a robust position estimator whose performance is close to that of the “ideal” maximum likelihood (ML) estimator (assuming known $p_V(v)$) is extremely challenging.

The idea followed here is to combine the position estimation and pdf estimation in an iterative process. According to the definition in [100], the resulting parameter estimation algorithms fall in the class of semi-parametric algorithm when the pdf $p_V(v)$ is estimated non-parametrically and the vector parameter $\boldsymbol{\theta}_p$ to be estimated is of finite dimension. The semi-parametric algorithm, which was initially proposed for robust multiuser detection in impulsive noise channels in [101], has its merits in dealing with the problem where the noise pdf is unknown. In the context of localization, however, the design of semi-parametric position estimation algorithm is more challenging due to the nonlinear signal model in general. Before proceeding with the new algorithm, however, an existing semi-parametric algorithm for robust position estimation [64] is first revisited. This type of algorithms give not only an estimate of the MS position but also an estimate of the measurement error distribution, which might be of use, for instance, in radio network optimization.

3.2.1 Competing Algorithm

In [64], the nonlinear signal model is first linearized by squaring both sides of (3.1) as follows:

$$r_{i,k}^2 = d_i^2(\boldsymbol{\theta}_p) + v_{i,k}^2 + 2d_i(\boldsymbol{\theta}_p)v_{i,k} = R_i + R - 2x_i x - 2y_i y + \tilde{v}_{i,k} \quad (3.6)$$

where

$$R = x^2 + y^2, \quad (3.7)$$

$$R_i = x_i^2 + y_i^2, \quad (3.8)$$

$$\tilde{v}_{i,k} = v_{i,k}^2 + 2d_i(\boldsymbol{\theta}_p)v_{i,k}. \quad (3.9)$$

Reformulating (3.6) as

$$r_{i,k}^2 - R_i = -2x_i x - 2y_i y + R + \tilde{v}_{i,k}, \quad (3.10)$$

and stacking all the terms in a vector, a linear regression model is thus obtained as

$$\tilde{\mathbf{r}} = \mathbf{S}\tilde{\boldsymbol{\theta}}_p + \tilde{\mathbf{v}} \quad (3.11)$$

with the vector notations defined by

$$\tilde{\mathbf{r}} = \begin{bmatrix} r_{1,1}^2 - R_1 \\ \vdots \\ r_{1,K}^2 - R_1 \\ \vdots \\ r_{N,1}^2 - R_N \\ \vdots \\ r_{N,K}^2 - R_N \end{bmatrix}, \quad \mathbf{S} = \begin{bmatrix} -2x_1, & -2y_1, & 1 \\ \vdots & & \\ -2x_1, & -2y_1, & 1 \\ \vdots & & \\ -2x_N, & -2y_N, & 1 \\ \vdots & & \\ -2x_N, & -2y_N, & 1 \end{bmatrix}, \quad (3.12)$$

$\tilde{\mathbf{v}} = [\tilde{v}_{1,1}, \dots, \tilde{v}_{1,K}, \dots, \tilde{v}_{N,1}, \dots, \tilde{v}_{N,K}]^T$ and $\tilde{\boldsymbol{\theta}}_p = [x, y, R]^T$. Note that $\tilde{\mathbf{r}}$ and $\tilde{\mathbf{v}}$ are both of dimension $NK \times 1$ and \mathbf{S} is of dimension $NK \times 3$. Assuming that the error terms in $\tilde{\mathbf{v}}$ are iid random variables with pdf $p_{\tilde{V}}(\tilde{v})$, although not always true, the log-likelihood function can be expressed by

$$\sum_{m=1}^{NK} \ln p_{\tilde{V}}(\tilde{r}_m - \mathbf{S}_m \tilde{\boldsymbol{\theta}}_p) \quad (3.13)$$

where \mathbf{S}_m denotes the m th row of the matrix \mathbf{S} . Since $p_{\tilde{V}}(\tilde{v})$ is in fact unknown, two conceptually similar iterative algorithms (cf. [64, Table I]) have been proposed to resolve an approximate ML estimator of $\tilde{\boldsymbol{\theta}}_p$ from

$$\sum_{m=1}^{NK} \mathbf{S}_m^T \varphi(\tilde{r}_m - \mathbf{S}_m \tilde{\boldsymbol{\theta}}) = \mathbf{0} \quad (3.14)$$

where the score function is defined by

$$\varphi(\tilde{v}) = -\frac{\hat{p}'_{\tilde{V}}(\tilde{v})}{\hat{p}_{\tilde{V}}(\tilde{v})} \quad (3.15)$$

with $\hat{p}_{\tilde{V}}(\tilde{v})$ denoting the estimated pdf using transformation kernel density estimation (TKDE) and $\hat{p}'_{\tilde{V}}(\tilde{v})$ denoting the first order derivative of $\hat{p}_{\tilde{V}}(\tilde{v})$ with respect to \tilde{v} . Given an extracted residual vector $\hat{\tilde{\mathbf{v}}}$ on one iteration, TKDE is carried out to give an estimated pdf $\hat{p}_{\tilde{V}}(\tilde{v})$ in the following four steps:

1. Transform the extracted residual vector $\hat{\tilde{\mathbf{v}}}$ via a transformation $\mathbf{u} = t(\hat{\tilde{\mathbf{v}}}; \zeta)$, where the parameter ζ steers the shape of this transformation function.
2. Symmetrize the transformed residual vector by $\mathbf{u}_s = [-\mathbf{u}^T, \mathbf{u}^T]^T$.
3. Perform conventional kernel density estimation upon \mathbf{u}_s and consequently obtain an estimated pdf $\hat{p}_U(u)$.
4. Transform $\hat{p}_U(u)$ back to $\hat{p}_{\tilde{V}}(\tilde{v})$.

Note that the transformation parameter ζ has to be determined in an optimization procedure prior to applying the TKDE. Details of the TKDE and selection of a proper transformation parameter can be found in [64, Appendix A and B], respectively.

Although this semi-parametric algorithm has shown considerable improvements in the estimation performance as compared to several other salient competitors [64], some issues still remain unsolved. Firstly, an auxiliary parameter R is introduced, but the constraint condition $R = x^2 + y^2$ is not incorporated in the optimization process, which surely leads to a sub-optimal solution. The technique proposed in [102] may serve as a powerful tool for remedying this drawback. Secondly, after the linearization, the transformed error terms $\tilde{v}_i(k)$, for $i = 1, \dots, N$ and $k = 1, \dots, K$, in $\tilde{\mathbf{v}}$ are no more identically distributed due to the factor $d_i(\boldsymbol{\theta})$ in the expressions of $\tilde{v}_i(k)$. Thirdly, in [64], a transformation of the original residual vector $\hat{\tilde{\mathbf{v}}}$ has to be conducted but there is no well established rule underpinning the selection of a parametric transformation function $t(\hat{\tilde{\mathbf{v}}}; \zeta)$ as well as an appropriate interval $[\zeta_L, \zeta_U]$ for optimizing ζ , which are crucial to the pdf estimation. As illustrated in [64, Fig. 7], a wrongly selected transformation parameter ζ may severely degrade the localization accuracy.

3.2.2 Proposed RIN Algorithm

The proposed algorithm is also a semi-parametric algorithm according to the definition given at the beginning of this section. In order to distinguish between the two semi-parametric algorithms, our algorithm is referred to as robust iterative nonparametric (RIN) algorithm with the term “nonparametric” indicating the fact that the measurement error pdf $p_V(v)$ is estimated non-parametrically. It is noteworthy that the RIN algorithm that we will focus on is essentially an iterative version of the robust nonparametric algorithm proposed in [27].

The main features of the new algorithm as compared to the existing semi-parametric algorithm in [64] are highlighted as follows. Firstly, the proposed algorithm directly uses the nonlinear signal model. As a consequence, evaluation of the constraint condition $R = x^2 + y^2$ is avoided and the measurement error terms in \mathbf{v} are iid. Secondly, TKDE is replaced by nonparametric adaptive kernel density estimation (AKDE) to obtain an estimate of the measurement error distribution. The latter is advantageous since the parameters required for constructing a density estimator are set adaptively and fully automatically as compared to the TKDE. Thirdly, a numerical method is employed to resolve a position estimate from the approximate log-likelihood function, which is derived based on the *a priori* calculated pdf estimate. The key steps of the RIN localization algorithm are summarized in Algorithm 3.1. It is noteworthy that,

- The initial estimate $\boldsymbol{\theta}_p^{(0)}$ is set by the first two entries of the least-squares solution of (3.11), i.e., by $[\tilde{x}_{\text{LS}}, \tilde{y}_{\text{LS}}]^T$ of

$$\tilde{\boldsymbol{\theta}}_p^{\text{LS}} = [\tilde{x}_{\text{LS}}, \tilde{y}_{\text{LS}}, \tilde{R}_{\text{LS}}]^T = (\mathbf{S}^T \mathbf{S})^{-1} \mathbf{S}^T \tilde{\mathbf{r}}. \quad (3.18)$$

- The approximated measurement error distribution $\hat{p}_V^{(\eta)}(v)$ is composed of a sum of Gaussian kernels as follows

$$\hat{p}_V^{(\eta)}(v) = \frac{1}{NK} \sum_{m=1}^{NK} \frac{1}{\sqrt{2\pi}w^{(\eta)}\lambda_m^{(\eta)}} \exp \left[-\frac{(v - \hat{v}_m^{(\eta)})^2}{2(w^{(\eta)}\lambda_m^{(\eta)})^2} \right] \quad (3.19)$$

where $\hat{v}_m^{(\eta)}$ denotes the m th element of the residual vector $\hat{\mathbf{v}}^{(\eta)}$, $w^{(\eta)}$ denotes the window width and $\lambda_m^{(\eta)}$, $m = 1, 2, \dots, NK$, denote the local bandwidth factors calculated on the η th iteration.

- The cost function $g^{(\eta)}(\boldsymbol{\theta}_p) = -\mathcal{L}^{(\eta)}(\boldsymbol{\theta}_p)$ is given explicitly by

$$g^{(\eta)}(\boldsymbol{\theta}_p) = -\sum_{i=1}^N \sum_{k=1}^K \ln \left(\frac{1}{NK} \sum_{m=1}^{NK} \frac{1}{\sqrt{2\pi}w^{(\eta)}\lambda_m^{(\eta)}} \exp \left[-\frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \hat{v}_m^{(\eta)})^2}{2(w^{(\eta)}\lambda_m^{(\eta)})^2} \right] \right). \quad (3.20)$$

Algorithm 3.1 Robust Iterative Nonparametric (RIN) Algorithm

Step1—Initialization:

Define the convergence tolerance Δ and the maximum number of iterations N_{itr} . Set the iteration index $\eta = 0$. Choose an initial guess $\boldsymbol{\theta}_p^{(0)}$.

Step2—Joint Estimation:

On the η th ($\eta \geq 1$) iteration, sequentially do:

1. Determine the residual vector $\hat{\mathbf{v}}^{(\eta)} = \mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p^{(\eta-1)})$.
2. Construct an estimate of the actual pdf $p_V(v)$, $\hat{p}_V^{(\eta)}(v)$, from $\hat{\mathbf{v}}^{(\eta)}$ via the nonparametric AKDE described in Appendix 3.7.1.
3. Approximate the exact log-likelihood function by

$$\mathcal{L}^{(\eta)}(\boldsymbol{\theta}_p) = \sum_{i=1}^N \sum_{k=1}^K \ln \left(\hat{p}_V^{(\eta)}(r_{i,k} - d_i(\boldsymbol{\theta}_p)) \right). \quad (3.16)$$

4. Solve an updated position estimate numerically through

$$\boldsymbol{\theta}_p^{(\eta)} = \arg \min_{\boldsymbol{\theta}_p} -\mathcal{L}^{(\eta)}(\boldsymbol{\theta}_p). \quad (3.17)$$

Step3—Convergence Check:

If $\|\boldsymbol{\theta}_p^{(\eta)} - \boldsymbol{\theta}_p^{(\eta-1)}\| < \Delta$ or the maximum number of iterations N_{itr} is reached, then terminate the algorithm and obtain $\hat{\boldsymbol{\theta}}_p^{\text{RIN}} = \boldsymbol{\theta}_p^{(\eta)}$; otherwise reset $\eta \leftarrow \eta + 1$ and return to the joint estimation stage.

- Due to the use of the kernel density estimation, the evaluations of (3.20) as well as its gradient make the computational complexity of the RIN algorithm to scale as $\mathcal{O}((NK)^2)$ FLOPs per iteration.
- The robust nonparametric algorithm [27] is a special case of the RIN algorithm in that N_{itr} is restricted to one to save some computational resources.

Many numerical methods can be utilized to solve the minimization problem formulated in (3.17), e.g., the Newton-Raphson method [103] and quasi-Newton (QN) method [104]. Alternatively, it is safest to perform a two-dimensional (2-D) grid search with rather fine grid resolution in the vicinity of a good initial guess [90]. But the drawback lies in the high computational complexity. In this work, the Broyden-Fletcher-Goldfarb-Shanno (BFGS)-QN method is used to minimize the nonlinear cost function $g^{(\eta)}(\boldsymbol{\theta}_p)$ on each RIN iteration, since it theoretically guarantees downhill progress towards the local minimum in each Newton step [105, page 141]. The key steps of the BFGS-QN method are listed in Algorithm 3.2 for the η th iteration.

The reasons for choosing the least-squares solution as the initial value in Algorithm 3.1 are due to its simplicity and rather low computational complexity [3]. A more sophisticated strategy is to test several candidate initial values and choose the best one. The initial values can be, for instance, the position estimates obtained from the robust M-estimation algorithm [32,64] and the robust semi-parametric estimation algorithm [64] as well as the grid points in the vicinity of them.

It is conspicuous from Algorithm 3.1 that the parametric position estimation and the nonparametric pdf estimation are tightly combined in an iteration process. Intuitively, the improved position estimate will lead to a refined pdf estimate and vice versa so that at the convergence of this iterative algorithm a good estimation performance can be achieved. However, it remains difficult, even asymptotically, to theoretically analyze the performance difference between the new position estimator and the corresponding ML estimator (cf. (3.33)) in terms of bias and root-mean-square-error (RMSE). While the ML estimator is known to be asymptotically efficient from Section 2.1, this is not easy to show for the new position estimator, due to the difficulties in quantifying how well the true pdf $p_V(v)$ can be approximated for a given number of observations. But it is still believed that the performance of the new position estimator can be very close to that of the ML estimator when the number of distance measurements is sufficiently large.

Algorithm 3.2 BFGS-QN Method (on the η th RIN Iteration)

- 1) Set the sub-iteration index $j = 0$ and obtain a search direction $\mathbf{s}_j = -\mathbf{H}_j \cdot \nabla_{\boldsymbol{\theta}_p} g^{(\eta)}(\boldsymbol{\theta}_p^{(\eta,j)})$, where $\nabla_{\boldsymbol{\theta}_p} g^{(\eta)}(\boldsymbol{\theta}_p^{(\eta,j)})$ is the gradient of the cost function $g^{(\eta)}(\boldsymbol{\theta}_p)$ evaluated at $\boldsymbol{\theta}_p^{(\eta,j)}$. The initial value $\boldsymbol{\theta}_p^{(\eta,0)}$ is set to $\boldsymbol{\theta}^{(\eta-1)}$.
- 2) Find the step size ς_j along the direction \mathbf{s}_j via the cubic line search method introduced in [106, Algorithm 3.5 and 3.6].
- 3) Update the estimate by $\boldsymbol{\theta}_p^{(\eta,j+1)} = \boldsymbol{\theta}_p^{(\eta,j)} + \varsigma_j \mathbf{s}_j$.
- 4) Set $\boldsymbol{\delta}_j = \varsigma_j \mathbf{s}_j$ and $\boldsymbol{\gamma}_j = \nabla_{\boldsymbol{\theta}_p} g^{(\eta)}(\boldsymbol{\theta}_p^{(\eta,j+1)}) - \nabla_{\boldsymbol{\theta}_p} g^{(\eta)}(\boldsymbol{\theta}_p^{(\eta,j)})$.
- 5) Update the approximate Hessian matrix by

$$\mathbf{H}_{j+1} = \mathbf{H}_j + \left(1 + \frac{\boldsymbol{\gamma}_j^T \mathbf{H}_j \boldsymbol{\gamma}_j}{\boldsymbol{\delta}_j^T \boldsymbol{\gamma}_j} \right) \frac{\boldsymbol{\delta}_j \boldsymbol{\delta}_j^T}{\boldsymbol{\delta}_j^T \boldsymbol{\gamma}_j} - \left(\frac{\boldsymbol{\delta}_j \boldsymbol{\gamma}_j^T \mathbf{H}_j + \mathbf{H}_j \boldsymbol{\gamma}_j \boldsymbol{\delta}_j^T}{\boldsymbol{\delta}_j^T \boldsymbol{\gamma}_j} \right). \quad (3.21)$$

The initial approximate Hessian matrix \mathbf{H}_0 is set to an identity matrix \mathbf{I}_2 .

- 6) If $\|\boldsymbol{\theta}_p^{(\eta,j+1)} - \boldsymbol{\theta}_p^{(\eta,j)}\| < \Delta_p$, then stop; otherwise update the iteration index $j \leftarrow j + 1$ and return to step 2.
-

3.3 Cramér-Rao Lower Bound Computation

Let $\hat{\boldsymbol{\theta}}_p = [\hat{x}, \hat{y}]^T$ denote an unbiased estimator of a deterministic vector position parameter $\boldsymbol{\theta}_p = [x, y]^T$ and let $\text{Cov}(\hat{\boldsymbol{\theta}}_p)$ denote the covariance matrix of $\hat{\boldsymbol{\theta}}_p$. In chapter 2, Theorem 2.1 tells us that

$$\text{Cov}(\hat{\boldsymbol{\theta}}_p) = \mathbb{E}_{p(\mathbf{r}; \boldsymbol{\theta}_p)} \left\{ (\hat{\boldsymbol{\theta}}_p - \boldsymbol{\theta}_p)(\hat{\boldsymbol{\theta}}_p - \boldsymbol{\theta}_p)^T \right\} \succeq \mathcal{F}^{-1}(\boldsymbol{\theta}_p). \quad (3.22)$$

It is further shown in Appendix 3.7.2, for the nonlinear signal model in (3.2) with non-Gaussian error terms in \mathbf{v} that are iid according to a known pdf $p_V(v)$, that the FIM is given by

$$\mathcal{F}(\boldsymbol{\theta}_p) = \mathcal{I}_v \cdot \mathcal{H}(\boldsymbol{\theta}_p) \mathcal{H}^T(\boldsymbol{\theta}_p) \quad (3.23)$$

where

$$\mathcal{H}(\boldsymbol{\theta}_p) = \nabla_{\boldsymbol{\theta}_p} \mathbf{h}(\boldsymbol{\theta}_p) = \begin{bmatrix} \underbrace{\frac{x - x_1}{d_1(\boldsymbol{\theta}_p)}, \dots, \frac{x - x_1}{d_1(\boldsymbol{\theta}_p)}}_{K \text{ repetitions}}, \dots, \underbrace{\frac{x - x_N}{d_N(\boldsymbol{\theta}_p)}, \dots, \frac{x - x_N}{d_N(\boldsymbol{\theta}_p)}}_{K \text{ repetitions}} \\ \underbrace{\frac{y - y_1}{d_1(\boldsymbol{\theta}_p)}, \dots, \frac{y - y_1}{d_1(\boldsymbol{\theta}_p)}}_{K \text{ repetitions}}, \dots, \underbrace{\frac{y - y_N}{d_N(\boldsymbol{\theta}_p)}, \dots, \frac{y - y_N}{d_N(\boldsymbol{\theta}_p)}}_{K \text{ repetitions}} \end{bmatrix}, \quad (3.24)$$

and the intrinsic accuracy [107]

$$\mathcal{I}_v = \mathbb{E}_{p_V(v)} \left\{ \frac{[\nabla_v p_V(v)]^2}{p_V^2(v)} \right\} = \int \frac{[\nabla_v p_V(v)]^2}{p_V^2(v)} p_V(v) dv. \quad (3.25)$$

Remark 3.1. For most of the measurement error distributions, it is difficult to derive \mathcal{I}_v in closed form. A notable exception, however, is the Gaussian distribution $\mathcal{N}(v; \mu_v, \sigma_v^2)$ whose $\mathcal{I}_v = \sigma_v^{-2}$. In most cases, the integral in (3.25) has to be approximated using Monte Carlo integration [108], yielding

$$\mathcal{I}_v \approx \frac{1}{N_M} \sum_{n=1}^{N_M} \frac{[\nabla_v p_V(v^{(n)})]^2}{p_V^2(v^{(n)})} \quad (3.26)$$

where $v^{(n)}$, $n = 1, 2, \dots, N_M$ are iid error terms generated from $p_V(v)$. In Appendix 3.7.3, two representative mixture distributions as well as their gradients in analytical form are provided.

Remark 3.2. Only one Monte Carlo integration has to be performed to compute \mathcal{I}_v if the error terms in \mathbf{v} are iid, and this \mathcal{I}_v can be used to compute the CRLB for different BSs-MS geometries.

3.4 Theoretical Performance Metrics

In this section, various different performance metrics are presented that can be used to assess the performance of an unbiased position estimator. To that end, $p_V(v)$ is assumed to be known in the subsequent sections.

3.4.1 Bias, RMSE, and Efficiency

In localization applications, it is desirable to design an unbiased estimator with RMSE as small as possible [49]. Herein, the bias of a position estimator is defined by

$$\text{Bias}(\hat{\boldsymbol{\theta}}_p) = \mathbb{E}_{p(\mathbf{r}; \boldsymbol{\theta}_p)}\{\hat{\boldsymbol{\theta}}_p\} - \boldsymbol{\theta}_p \quad (3.27)$$

and the localization RMSE, frequently interpreted as the localization accuracy of a position estimator in the literature, is related to the obtained CRLB through

$$\begin{aligned} \text{RMSE}(\hat{\boldsymbol{\theta}}_p) &= \sqrt{\mathbb{E}_{p(\mathbf{r}; \boldsymbol{\theta}_p)}\{(\hat{x} - x)^2 + (\hat{y} - y)^2\}} \\ &= \sqrt{\text{tr}\{\text{Cov}(\hat{\boldsymbol{\theta}}_p)\}} \\ &\geq \text{CRLB}_{\text{pos}}(\boldsymbol{\theta}_p) = \sqrt{\text{tr}\{\mathcal{F}^{-1}(\boldsymbol{\theta}_p)\}} \end{aligned} \quad (3.28)$$

where $\text{CRLB}_{\text{pos}}(\boldsymbol{\theta}_p)$ can be interpreted as the best achievable localization RMSE of an unbiased position estimator. Besides, the estimation efficiency of an unbiased position estimator is re-defined as

$$\eta_{\text{eff}}(\hat{\boldsymbol{\theta}}_p) = \frac{\text{CRLB}_{\text{pos}}(\boldsymbol{\theta}_p)}{\text{RMSE}(\hat{\boldsymbol{\theta}}_p)} = \sqrt{\frac{\text{tr}(\mathcal{F}^{-1}(\boldsymbol{\theta}_p))}{\text{tr}(\text{Cov}(\hat{\boldsymbol{\theta}}_p))}}. \quad (3.29)$$

It follows from (3.28) that $0 \leq \eta_{\text{eff}}(\hat{\boldsymbol{\theta}}_p) \leq 1$. If a position estimator is unbiased and simultaneously attains $\eta_{\text{eff}}(\hat{\boldsymbol{\theta}}_p) = 1$, then it is referred to as an efficient position estimator. As it is well known, the ML estimator obtained from (3.33) is asymptotically efficient, i.e., unbiased and $\eta_{\text{eff}}(\hat{\boldsymbol{\theta}}_p) = 1$ as the number of measurements goes to infinity.

3.4.2 Geometric Dilution of Precision

Another important metric is called geometric dilution of precision (GDOP), which is used to describe the influence of BSs-MS geometry on the relationship between the

measurement error and localization accuracy [109]. Since identical variance of the measurement errors is assumed across all BSs, the GDOP is easily calculated by

$$\text{GDOP} = \sqrt{\frac{\text{tr}\{\text{Cov}(\hat{\boldsymbol{\theta}}_p)\}}{\sigma_v^2}} \quad (3.30)$$

where σ_v^2 denotes the variance of the measurement error distribution, $p_V(v)$. For an efficient position estimator, (3.30) becomes

$$\text{GDOP} = \frac{\text{CRLB}_{\text{pos}}}{\sigma_v}. \quad (3.31)$$

It was reported in [40] that GDOP values smaller than three imply well-suited geometry, whereas those larger than six imply a deficient geometry.

3.5 Simulations

In this section, the performance of the proposed RIN algorithm will be evaluated and further compared with several competing algorithms, including the robust semi-parametric algorithm surveyed in Section 3.2.1, in comprehensive simulations. Section 3.5.1 introduces the overall simulation setup and Section 3.5.2 shows the simulation results.

3.5.1 Simulation Setup

In what follows, a stationary MS is to be located using $N = 8$ BSs in a cellular radio network. In order to be as realistic as possible, the BS positions are taken from an operating cellular radio network in a German city center [22, 29]. The geometry of the BSs as well as the approximate location of the city center are shown in Fig. 3.2. The BS antennae are generally deployed on rooftops and the city center can be characterized as urban area with multistory buildings and narrow streets.

Field trials conducted in this city have revealed that the distance measurement error terms collected from different BSs can be well approximated by a Gaussian mixture distribution [29]. In the simulations considered in Section 3.5.2, iid measurement error terms are generated from the following two-mode Gaussian mixture distribution

$$p_V(v) = \alpha_L \mathcal{N}(v; \mu_L, \sigma_L^2) + \alpha_{NL} \mathcal{N}(v; \mu_{NL}, \sigma_{NL}^2) \quad (3.32)$$

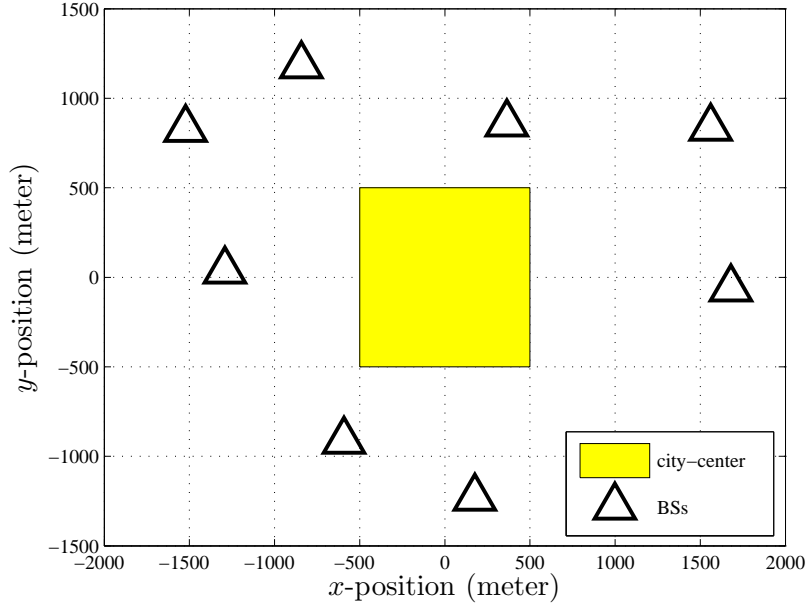


Figure 3.2. 2-D illustration of the geometry of BSs and the city center area in a real-world scenario in Germany.

where $\mu_L = 0$ meter (m), $\sigma_L = 55$ m, $\mu_{NL} = 380$ m and $\sigma_{NL} = 120$ m.

The proposed RIN estimator is compared to the following position estimators:

- Least-squares estimator, cf. (3.18)
- Robust M-estimator [32, 64]
- Robust semi-parametric estimator [64]
- Robust nonparametric estimator [27]
- ML estimator with known $p_V(v)$

The first three algorithms are all developed under the linear regression model in (3.11). In the robust M-estimation algorithm, the clipping point c_H of Huber's score function $\psi(\tilde{v}; c_H)$ (cf. [64, (7)]) is adaptively calculated by $c_H = 0.6/(1.483 \cdot \text{mad}(\hat{\tilde{v}}))$, where $\hat{\tilde{v}}$ denotes the *a priori* extracted residual vector on each iteration and $\text{mad}(\cdot)$ denotes the median absolute deviation. The same transformation function $t(\hat{\tilde{v}}; \zeta)$ and search interval $[\zeta_L = 0.9, \zeta_U = 1]$ as suggested in [64] are chosen. The robust nonparametric algorithm follows Algorithm 3.1, except that the number of iterations is constrained to

one. The above competitors do not assume $p_V(v)$ to be known a priori. By assuming $p_V(v)$ to be known and the measurement errors are iid, the ML estimator is computed through

$$\hat{\boldsymbol{\theta}}_p^{\text{ML}} = \arg \max_{\boldsymbol{\theta}_p} \sum_{i=1}^N \sum_{k=1}^K \ln p_V(r_{i,k} - d_i(\boldsymbol{\theta}_p)). \quad (3.33)$$

Again, the BFGS-QN method is adopted with the initial guess set ideally by the true MS position. The RIN algorithm terminates if the convergence tolerance $\Delta = 0.1$ m or the maximum number of iterations $N_{\text{itr}} = 20$ trials has been reached. All the simulations have been performed under MATLABTMR2010a environment on a PC equipped with Intel[®]CoreTMi5-760 processor (2.80GHz) and 8GB RAM.

3.5.2 Simulation Results

3.5.2.1 GDOP

In the first experiment, GDOP values are computed according to (3.31) (i.e., assuming an efficient position estimator) for various different positions in the city center area as illustrated in Fig. 3.2. In this simulation, the number of distance measurements K obtained at each BS is set to one, and the NLOS contamination ratio α_{NL} is set to 0.5. According to [105, Section 1.4.16], the variance of $p_V(v)$ in (3.32), σ_v , is given by

$$\sigma_v = \sqrt{\alpha_L \sigma_L^2 + \alpha_{\text{NL}} \sigma_{\text{NL}}^2 + \alpha_L \alpha_{\text{NL}} \mu_{\text{NL}}^2}. \quad (3.34)$$

The resulting GDOP values are shown in Fig. 3.3, indicating a well-suited BS geometry for locating an MS in the city center area. Increasing K or decreasing α_{NL} will yield better GDOP values.

3.5.2.2 Bias, RMSE and CRLB

In the second experiment, the different position estimators are evaluated in terms of bias and RMSE. To that end, the bias and RMSE of different position estimators are evaluated in a large-scale Monte Carlo simulation with 2500 independent trials. Two examples are investigated in the sequel.

The first example assumes that the MS is located at $[x = 0.25 \text{ km}, y = 0.5 \text{ km}]^T$ and the number of distance measurements at each BS is $K = 20$ samples. The bias and RMSE of the aforementioned position estimators are evaluated as a function of the

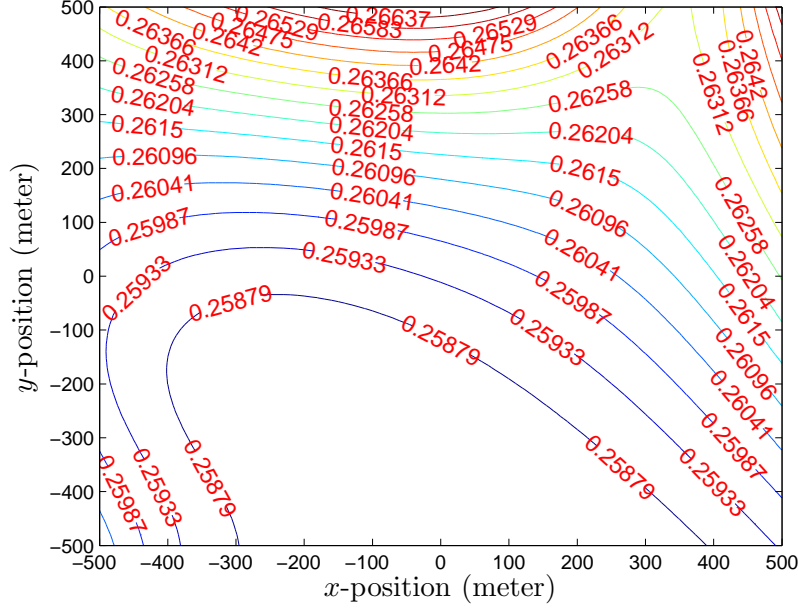


Figure 3.3. GDOP values measured for an efficient position estimator in the city center area with $K = 1$ and $\alpha_{\text{NL}} = 0.5$.

NLOS contamination ratio $0 \leq \alpha_{\text{NL}} \leq 1$. The results for the bias and RMSE are shown in Fig. 3.4 and Fig. 3.5. The performance lower bound CRLB_{pos} is calculated according to (3.28) and shown here to serve as a benchmark for comparing different RMSE curves. It is noteworthy that for the cases $\alpha_{\text{NL}} = 0$ and $\alpha_{\text{NL}} = 1$, the FIM has the following analytical form:

$$\mathcal{F}(\boldsymbol{\theta}_p) = \begin{cases} \sigma_{\text{L}}^{-2} \cdot \mathcal{H}(\boldsymbol{\theta}_p) \mathcal{H}^T(\boldsymbol{\theta}_p), & \text{for } \alpha_{\text{NL}} = 0 \\ \sigma_{\text{NL}}^{-2} \cdot \mathcal{H}(\boldsymbol{\theta}_p) \mathcal{H}^T(\boldsymbol{\theta}_p), & \text{for } \alpha_{\text{NL}} = 1 \end{cases} \quad (3.35)$$

The results mainly reveal the following two aspects. Firstly, the selected position estimators all perform similarly (close to the ML estimator) when the NLOS contamination ratio α_{NL} is close to zero. The reasons are as follows:

1. For the first two robust estimators developed under the linear regression model, the measurement error $v_{i,k}$ is most probably generated from the LOS mixture component $\mathcal{N}(v; \mu_{\text{L}}, \sigma_{\text{L}}^2)$ resulting in $\tilde{v}_{i,k} \approx 2d_i(\boldsymbol{\theta}_p)v_{i,k}$ in (3.9) for the assumed simulation scenario. As a result, $\tilde{v}_{i,k}$, $i = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$, can be regarded as approximately jointly Gaussian distributed. It was shown in [32, 64] that they perform nearly optimally under the Gaussian model.
2. Due to the good quality of the initial guess (i.e., the least-squares estimate), the measurement error distribution estimated on the first iteration of Algorithm 3.1 is

already close to true one, so extra iterations can only ameliorate it slightly. This explains why the robust nonparametric estimator is close to the RIN estimator.

Secondly, the RIN estimator is closest to the ML estimator and outperforms all the other robust competitors by far when α_{NL} is large. The reasons are as follows:

1. When α_{NL} is large, the pdf of $\tilde{v}_{i,k}$ starts to deviate from a Gaussian model, leading to deteriorated performance of the ones developed under the linear regression model.
2. The robust M-estimator breaks down at best for a contamination ratio equal to 0.5 [14]. Therefore, it is not surprising to see the drastic performance degradation from $\alpha_{\text{NL}} > 0.3$ in Fig. 3.5.
3. The robust semi-parametric estimator [64] performs even worse than the least-squares estimator or the robust M-estimator for some α_{NL} (e.g., $\alpha_{\text{NL}} = 0.9$ in Fig. 3.5). The reason may lie in the fact that the transformation function $t(\hat{\mathbf{v}}; \zeta)$ and the associated search interval $[\zeta_L = 0.9, \zeta_U = 1]$ needed in the TKDE are inappropriate for the assumed simulation scenario.
4. Having conquered the drawbacks of the semi-parametric algorithm (cf. Section 3.2.1), improved performance has been harvested in terms of bias and RMSE for both the robust nonparametric algorithm [27] and RIN algorithm.
5. Since the quality of the initial guess degenerates as α_{NL} increases, larger discrepancy has been observed between the exact measurement error distribution $p_V(v)$ and its estimate calculated on the first iteration of Algorithm 3.1. Therefore, the robust nonparametric algorithm becomes worse as α_{NL} increases. Introducing extra iterations as is done in the RIN algorithm successfully reduces the discrepancy and brings improved localization accuracy.

In the second example, the overall performance of the position estimators in the city center area is examined. A total number of 2000 sets of different positions are generated uniformly from the city center area. The localization RMSE and CRLB_{pos} are both calculated for each set and finally averaged. The results are shown in Fig. 3.6, from which we observe that the performance of the RIN estimator and ML estimator stays almost unaltered while the performance of the others becomes worse. It is noteworthy that the RMSE curve of the ML estimator coincides well with the corresponding CRLB_{pos} curve in Fig. 3.5 and Fig. 3.6.

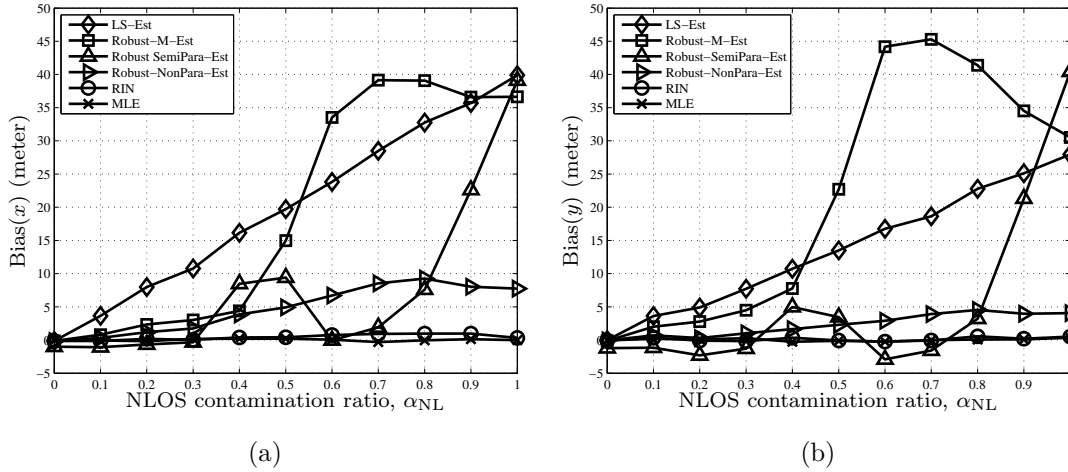


Figure 3.4. Bias (assuming a fixed position $[x = 0.25 \text{ km}, y = 0.25 \text{ km}]$) of different position estimators versus the NLOS contamination ratio, α_{NL} . (a) describes the bias in x -position; and (b) describes the bias in y -position.

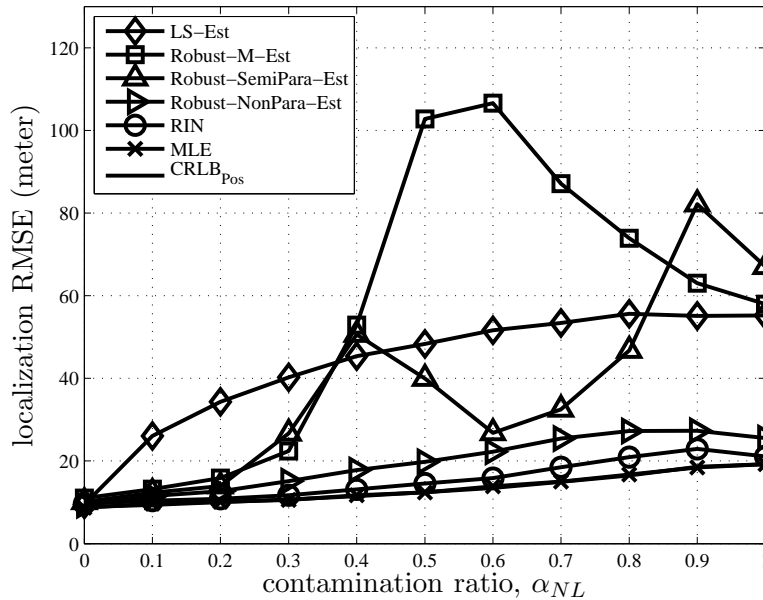


Figure 3.5. Localization RMSE (assuming a fixed position $[x = 0.25 \text{ km}, y = 0.25 \text{ km}]$) of different position estimators versus the NLOS contamination ratio, α_{NL} .

3.5.2.3 Estimation Efficiency

Next, estimation efficiency of the RIN estimator is shown as a function of the number of measurements. The MS locates at $[x = 0.25 \text{ km}, y = 0.25 \text{ km}]$, K varies from 5 to 75 at an increment of 5 samples, and $\alpha_{\text{NL}} = 0.5$. For each K , 1000 independent Monte Carlo trials are carried out to first calculate the RMSE, then the estimation efficiency is evaluated according to (3.29). Fig. 3.7 shows the resulting estimation efficiency versus the number of measurements at each BS. Gathering these results, the performance of the RIN estimator are summarized as follows:

- It largely deviates from the ML estimator when NK is small (e.g., $NK \leq 50$ samples), but still outperforms other robust estimators by far.
- It performs closer to the ML estimator as NK increases. When $NK = 100$ samples are available, it attains an efficiency of $\eta_{eff} \approx 0.8$. When we further increase NK to 300 samples, the efficiency will be $\eta_{eff} \approx 0.9$. Although not shown here, the efficiency will increase very slowly thereafter and attain $\eta_{eff} \approx 1$ at $NK = 1000$ samples.

In addition, we also found that the existing robust semi-parametric estimator in [64] does not fulfill the asymptotic efficiency property.

3.5.2.4 PDF Estimation via AKDE

In the previous experiments, the RIN algorithm have demonstrated improved performance as compared to the robust nonparametric algorithm. The improvement mainly stems from the extra enhancement in the pdf estimation achieved in the RIN algorithm. In order to experimentally confirm this statement, Fig. 3.8 shows the pdf estimates obtained on different iterations of Algorithm 3.1 versus the actual measurement error distribution for a particular Monte Carlo trial in the first example of this section. Moreover, Fig. 3.9 shows the histogram of both the actual measurement error terms and the error residuals (cf. Step 2 of Algorithm 3.1) extracted on the last iteration of this trial.

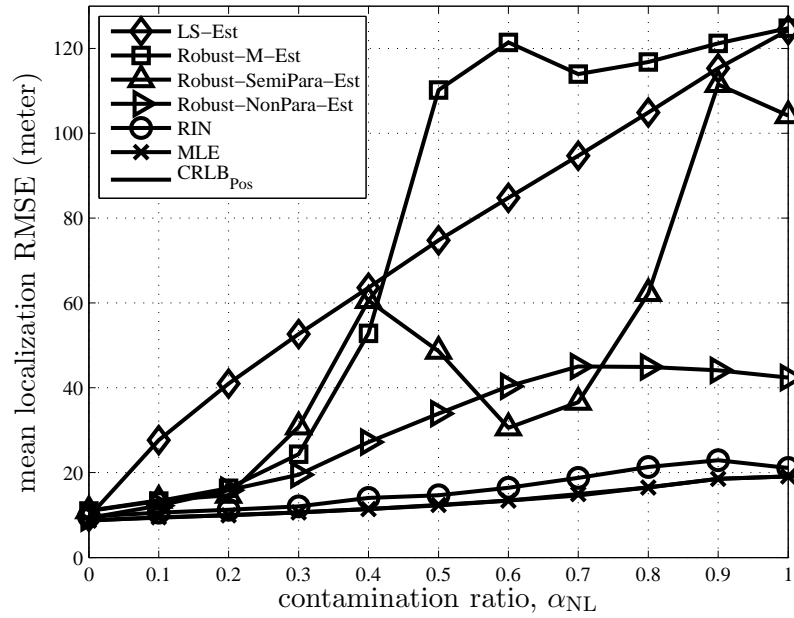


Figure 3.6. Mean localization RMSE (over 2000 sets of positions uniformly generated from the city center area) of different position estimators versus NLOS contamination ratio, α_{NL} .

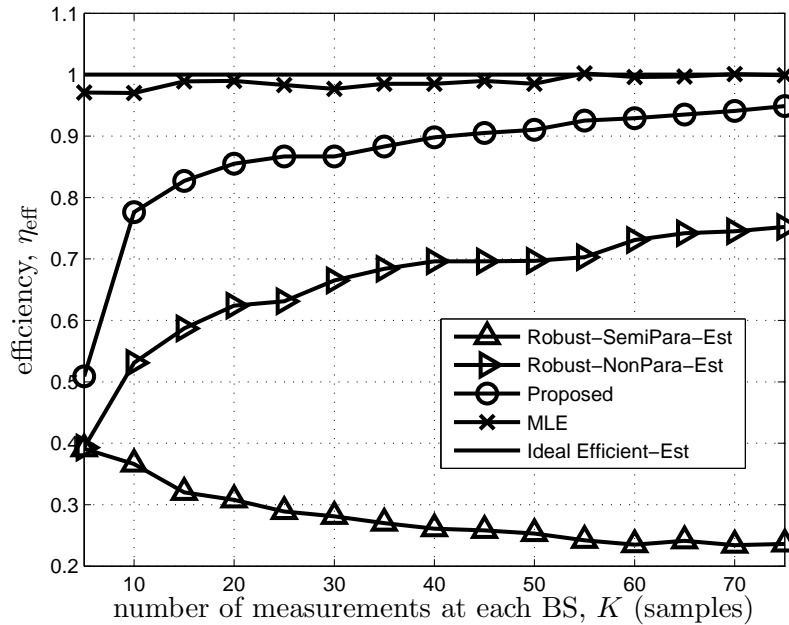


Figure 3.7. Estimation efficiency, η_{eff} , of different position estimators versus the number of measurements collected at each BS, K .

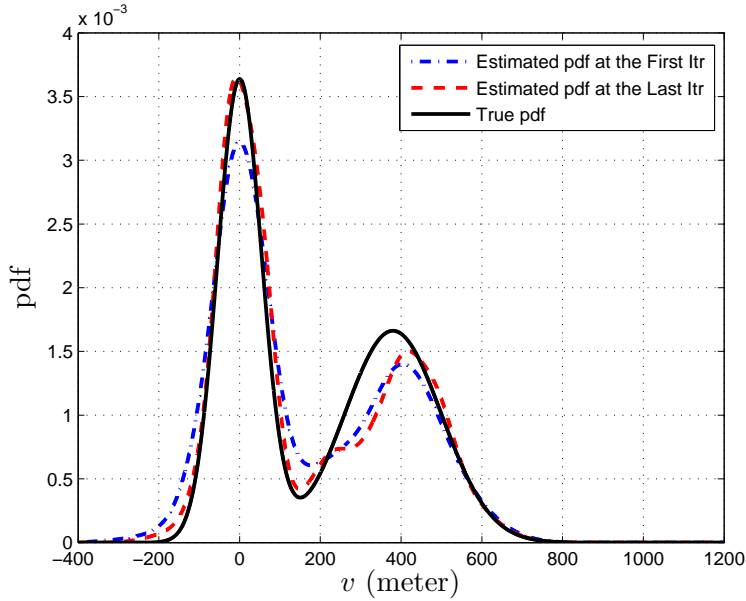


Figure 3.8. A comparison among the actual measurement error distribution $p_V(v)$ (black solid line), the estimated distribution (red dashed line) on the last iteration of Algorithm 3.1, and the estimated distribution (blue dash-dot line) on the first iteration of the Algorithm 3.1 (as same as the one constructed in the robust nonparametric algorithm [27]) in a single Monte Carlo trial with $[x = 0.25 \text{ km}, y = 0.25 \text{ km}]$, $K = 20$ samples, and $\alpha_{NL} = 0.5$.

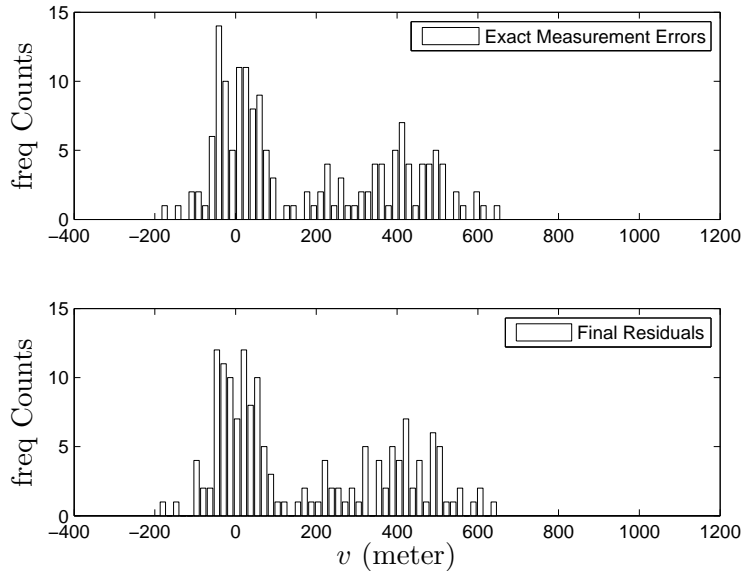


Figure 3.9. **Top sub-figure:** Histogram of the actual measurement error terms in one particular Monte Carlo trial. **Bottom sub-figure:** Histogram of the residuals extracted on the last iteration of Algorithm 3.1 in the same trial.

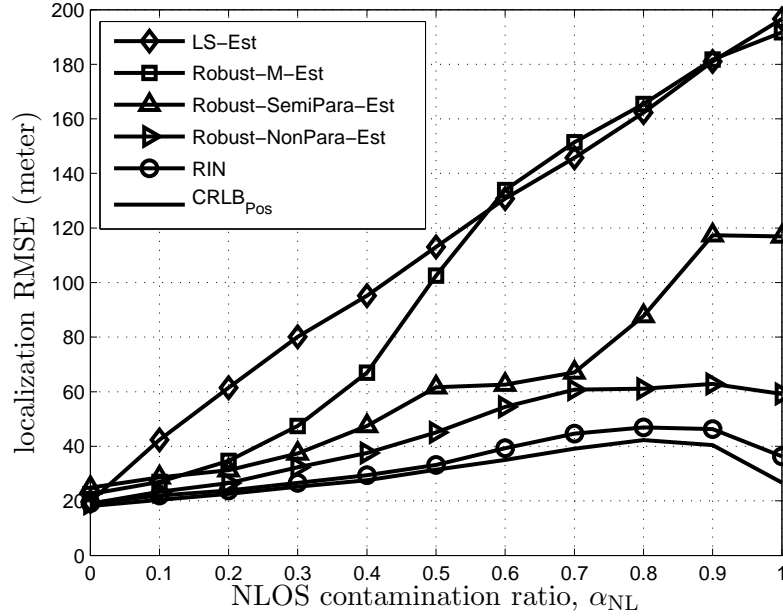


Figure 3.10. Localization RMSE of different position estimators versus the NLOS contamination ratio, α_{NL} , according to new settings. Herein, the measurement error distribution is modeled by a Gaussian-Rayleigh mixture with $\mu_L = 0$ m, $\sigma_L = 150$ m and $\gamma_{NL} = 500$ m.

3.5.2.5 One Additional example

In the above simulations, the Gaussian mixture measurement error model is the main focus, although the proposed algorithm is able to work for any other model. For rigor, the localization performance of the RIN algorithm will be shown in another example where we assume a simulated network and a two-mode Gaussian-Rayleigh measurement error model.

Consider a cellular radio network with $N = 10$ BSs and one MS. The BSs are located in a 2-D plane with fixed positions $[x_1 = 2.5, y_1 = 5]$, $[x_2 = 1, y_2 = 3.5]$, $[x_3 = 4.5, y_3 = 1.75]$, $[x_4 = 1.5, y_4 = 4]$, $[x_5 = 3, y_5 = 4.5]$, $[x_6 = 1.75, y_6 = 1]$, $[x_7 = 4, y_7 = 0.75]$, $[x_8 = 5, y_8 = 1.25]$, $[x_9 = 0.5, y_9 = 2]$, $[x_{10} = 3, y_{10} = 0.25]$ all in km. The true MS locates at $[x = 2.5 \text{ km}, y = 2 \text{ km}]$. At each BS, a number of $K = 30$ distance measurements are obtained. The measurement error distribution is modeled by a Gaussian-Rayleigh mixture, i.e.,

$$p_V(v) = \alpha_L \mathcal{N}(v; \mu_L, \sigma_L^2) + \alpha_{NL} \mathcal{R}(v; \gamma_{NL}) \quad (3.36)$$

where

$$\mathcal{R}(v; \gamma_{\text{NL}}) = \begin{cases} \frac{v}{\gamma_{\text{NL}}} \exp\left[\frac{-v^2}{2\gamma_{\text{NL}}^2}\right], & v \geq 0 \\ 0, & v < 0 \end{cases}, \quad (3.37)$$

with the true parameters set by $\mu_{\text{L}} = 0$ m, $\sigma_{\text{L}} = 150$ m and $\gamma_{\text{NL}} = 500$ m. The localization RMSE is shown as a function of the contamination ratio in Fig. 3.10. Similar conclusions can be drawn accordingly.

3.6 Conclusions

We developed a robust iterative nonparametric (RIN) algorithm for position estimation in harsh mixed line-of-sight/non-line-of-sight environments with completely unknown measurement error distribution. The idea of the RIN algorithm is to jointly estimate the position and measurement error distribution. We presented the best achievable localization accuracy in terms of the Cramér-Rao lower bound in the assumed situation. Simulations were performed in various different scenarios. The results have shown that the proposed position estimator can achieve significantly improved performance as compared to some salient competitors. Especially when the number of distance measurements is large, the proposed RIN position estimator is very close to the actual maximum likelihood estimator, and the localization RMSE approaches the best achievable performance. However, the improvement comes at the expense of higher computation as compared to other competitors.

3.7 Appendix

3.7.1 Adaptive Kernel Density Estimation

Kernel density estimation (KDE) is a nonparametric approach for estimating a pdf based on a given set of observations [37]. The gist of the KDE is to approximate a pdf (usually without ground truth) by a linear combination of kernel densities with carefully tuned bandwidths. Amongst a large number of variations of this kind, the class of adaptive kernel density estimation methods is adopted, which gives overall good performance in estimating a long-tailed and/or multi-modal pdf [37]. Assuming that we have a set of NK iid observations $\mathbf{v} = \{v_1, v_2, \dots, v_{NK}\}$ generated from a continuous

Algorithm 3.3 Adaptive Kernel Density Estimator

1) Find a pilot density estimator

$$\hat{p}_0(v) = \frac{1}{NK} \sum_{m=1}^{NK} \frac{1}{w_0} \mathcal{K}_G \left(\frac{v - v_m}{w_0} \right) \quad (3.38)$$

where $\mathcal{K}_G(\cdot)$ denotes the standard Gaussian kernel, $w_0 = 0.79 \cdot \text{iqr}\{\mathbf{v}\}(NK)^{-1/5}$ denotes an initial global bandwidth and $\text{iqr}\{\mathbf{v}\}$ denotes the interquartile range of $\mathbf{v} = \{v_1, v_2, \dots, v_{NK}\}$.

2) Define local bandwidths λ_m , $m = 1, 2, \dots, NK$, by

$$\lambda_m = \left(\hat{p}_0(v_m) / \left[\prod_{m=1}^{NK} \hat{p}_0(v_m) \right]^{\frac{1}{NK}} \right)^{-\beta} \quad (3.39)$$

where the sensitivity parameter β is set to 0.5 as suggested in [110].

3) An adaptive kernel density estimator $\hat{p}_V(v)$ is finally constructed by

$$\hat{p}_V(v) = \frac{1}{NK} \sum_{m=1}^{NK} \frac{1}{w\lambda_m} \mathcal{K}_G \left(\frac{v - v_m}{w\lambda_m} \right). \quad (3.40)$$

univariate distribution $p_V(v)$, the steps for constructing an adaptive kernel density estimator $\hat{p}_V(v)$ are demonstrated in Algorithm 3.3.

As we desire a reliable window width w that can be selected adaptively and fully automatically, the least-squares cross-validation (LSCV) technique, see for instance [111] and [112], is utilized. The principle of the LSCV is to find a window width w in the sense of minimizing the mean-integrated-square-error (MISE),

$$\mathbb{E} \left\{ \int_{-\infty}^{\infty} (\hat{p}_V(v) - p_V(v))^2 dv \right\}, \quad (3.41)$$

which is widely used to measure the discrepancy between $\hat{p}_V(v)$ and $p_V(v)$.

It follows from [37] that minimizing the MISE is equivalent to minimizing the expectation of a score function $M_0(w)$ defined by

$$M_0(w) = \int_{-\infty}^{\infty} \hat{p}_V^2(v) dv - \frac{2}{NK} \sum_{m=1}^{NK} \hat{p}_V^{-m}(v_m). \quad (3.42)$$

Due to the difficulty in calculating the expectation, $M_0(w)$ is minimized instead to give a good window width. In (3.42), $\hat{p}_V^{-m}(v)$ is also a density estimator, which is

constructed from all iid observations except v_m , more precisely,

$$\hat{p}_V^{-m}(v) = \frac{1}{M-1} \sum_{m'=1, m' \neq m}^{NK} \frac{1}{w\lambda_{m'}} \mathcal{K}_G\left(\frac{v-v_{m'}}{w\lambda_{m'}}\right). \quad (3.43)$$

Since $\mathcal{K}_G(\cdot)$ is the standard Gaussian kernel, it was shown in [37] that

$$\int_{-\infty}^{\infty} \hat{p}_V^2(v) dv = \frac{1}{(NK)^2} \sum_{m=1}^{NK} \sum_{m'=1}^{NK} \frac{1}{\sqrt{2\pi(w^2\lambda_m^2 + w^2\lambda_{m'}^2)}} \exp\left[-\frac{(v_m - v_{m'})^2}{2(w^2\lambda_m^2 + w^2\lambda_{m'}^2)}\right]. \quad (3.44)$$

3.7.2 Derivations of (3.23)

When the regularity conditions are all satisfied, the FIM can be re-expressed by

$$\mathcal{F}(\boldsymbol{\theta}_p) = \mathbb{E}_{p(\mathbf{r}; \boldsymbol{\theta}_p)} \left\{ \frac{\nabla_{\boldsymbol{\theta}_p} p(\mathbf{r}; \boldsymbol{\theta}_p) \nabla_{\boldsymbol{\theta}_p}^T p(\mathbf{r}; \boldsymbol{\theta}_p)}{p(\mathbf{r}; \boldsymbol{\theta}_p)^2} \right\}. \quad (3.45)$$

Applying the chain rule on $\nabla_{\boldsymbol{\theta}_p} \ln p(\mathbf{r}; \boldsymbol{\theta}_p)$ [91, 113] yields

$$\begin{aligned} \nabla_{\boldsymbol{\theta}_p} \ln p(\mathbf{r}; \boldsymbol{\theta}_p) &= \nabla_{\boldsymbol{\theta}_p} \ln p_{\mathbf{v}}(\mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p)) \\ &= \frac{-\nabla_{\boldsymbol{\theta}_p} \mathbf{h}(\boldsymbol{\theta}_p) \cdot [\nabla_{\mathbf{v}} p_{\mathbf{v}}(\mathbf{v}) |_{\mathbf{v}=\mathbf{r}-\mathbf{h}(\boldsymbol{\theta}_p)}]}{p_{\mathbf{v}}(\mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p))}. \end{aligned} \quad (3.46)$$

Inserting (3.46) into (3.45) yields

$$\begin{aligned} \mathcal{F}(\boldsymbol{\theta}_p) &= \nabla_{\boldsymbol{\theta}_p} \mathbf{h}(\boldsymbol{\theta}_p) \mathbb{E}_{p(\mathbf{r}; \boldsymbol{\theta}_p)} \left\{ \frac{\nabla_{\mathbf{v}} p_{\mathbf{v}}(\mathbf{v}) \nabla_{\mathbf{v}}^T p_{\mathbf{v}}(\mathbf{v}) |_{\mathbf{v}=\mathbf{r}-\mathbf{h}(\boldsymbol{\theta}_p)}}{p_{\mathbf{v}}^2(\mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p))} \right\} \nabla_{\boldsymbol{\theta}_p}^T \mathbf{h}(\boldsymbol{\theta}_p) \\ &= \nabla_{\boldsymbol{\theta}_p} \mathbf{h}(\boldsymbol{\theta}_p) \mathbb{E}_{p_{\mathbf{v}}(\mathbf{v})} \left\{ \frac{\nabla_{\mathbf{v}} p_{\mathbf{v}}(\mathbf{v}) \nabla_{\mathbf{v}}^T p_{\mathbf{v}}(\mathbf{v})}{p_{\mathbf{v}}^2(\mathbf{v})} \right\} \nabla_{\boldsymbol{\theta}_p}^T \mathbf{h}(\boldsymbol{\theta}_p) \\ &= \boldsymbol{\mathcal{H}}(\boldsymbol{\theta}_p) \boldsymbol{\mathcal{I}}_{\mathbf{v}} \boldsymbol{\mathcal{H}}^T(\boldsymbol{\theta}_p) \end{aligned} \quad (3.47)$$

where $\boldsymbol{\mathcal{H}}(\boldsymbol{\theta}_p)$ is defined in (3.24), and

$$\boldsymbol{\mathcal{I}}_{\mathbf{v}} = \mathbb{E}_{p_{\mathbf{v}}(\mathbf{v})} \left\{ \frac{\nabla_{\mathbf{v}} p_{\mathbf{v}}(\mathbf{v}) \nabla_{\mathbf{v}}^T p_{\mathbf{v}}(\mathbf{v})}{p_{\mathbf{v}}^2(\mathbf{v})} \right\}. \quad (3.48)$$

Due to the statistical properties of the vector elements v_m , $m = 1, 2, \dots, NK$, the (m, m') th entry of the matrix $\boldsymbol{\mathcal{I}}_{\mathbf{v}}$ can be expressed by

$$[\boldsymbol{\mathcal{I}}_{\mathbf{v}}]_{m, m'} = \mathbb{E}_{p_{\mathbf{v}}(\mathbf{v})} \left\{ \frac{\nabla_{v_m} p_V(v_m) \nabla_{v_{m'}} p_V(v_{m'})}{p_V(v_m) p_V(v_{m'})} \right\}. \quad (3.49)$$

If $m = m'$, the m th main diagonal element is simplified to

$$[\mathcal{I}_{\mathbf{v}}]_{m,m} = \mathcal{I}_{v_m} = \mathbb{E}_{p_V(v_m)} \left\{ \frac{\nabla_{v_m}^2 p_V(v_m)}{p_V^2(v_m)} \right\}. \quad (3.50)$$

Since v_m , $m = 1, 2, \dots, NK$ are identically distributed, it can be concluded that

$$\mathcal{I}_{\mathbf{v}} = \mathcal{I}_{v_1} = \mathcal{I}_{v_2} = \dots = \mathcal{I}_{v_{NK}}. \quad (3.51)$$

if $m \neq m'$, $[\mathcal{I}_{\mathbf{v}}]_{m,m'}$ is equal to

$$\mathbb{E}_{p_V(v_m)} \left\{ \frac{\nabla_{v_m} p_V(v_m)}{p_V(v_m)} \right\} \mathbb{E}_{p_V(v_{m'})} \left\{ \frac{\nabla_{v_{m'}} p_V(v_{m'})}{p_V(v_{m'})} \right\}, \quad (3.52)$$

since v_m and $v_{m'}$ are mutually independent. In order to prove the diagonal property of $\mathcal{I}_{\mathbf{v}}$, we recall that for the computation of the CRLB, the following regularity condition

$$\mathbb{E}_{p(\mathbf{r}; \boldsymbol{\theta}_p)} \left\{ \nabla_{\boldsymbol{\theta}_p} \ln p(\mathbf{r}; \boldsymbol{\theta}_p) \right\} = \nabla_{\boldsymbol{\theta}_p} \mathbf{h}(\boldsymbol{\theta}_p) \mathbb{E}_{p_{\mathbf{v}}(\mathbf{v})} \left\{ \frac{\nabla_{\mathbf{v}} p_{\mathbf{v}}(\mathbf{v})}{p_{\mathbf{v}}(\mathbf{v})} \right\} = \mathbf{0} \quad (3.53)$$

must hold for all $\boldsymbol{\theta}_p$ in its parameter space, which is equivalent to saying that

$$\mathbb{E}_{p_V(v)} \left\{ \frac{\nabla_v p_V(v)}{p_V(v)} \right\} = 0, \quad (3.54)$$

when the elements in \mathbf{v} are iid. As a result, $[\mathcal{I}_{\mathbf{v}}]_{m,m'} = 0$ for $m \neq m'$ and $[\mathcal{I}_{\mathbf{v}}]_{m,m'} = \mathcal{I}_v$ for $m = m'$. Hence, $\mathcal{I}_{\mathbf{v}} = \mathcal{I}_v \cdot \mathbf{I}_{NK}$ is proven. Inserting this result into (3.47) gives (3.23).

3.7.3 Expressions of $p_V(v)$ and $\nabla_v p_V(v)$

To capture the characteristics of the measurement error in mixed LOS/NLOS environments, the two-mode Gaussian-Gaussian mixture distribution and two-mode Gaussian-Rayleigh mixture distribution are favorable to use. Under the LOS condition, $p_V^{(L)}(v; \boldsymbol{\beta}_L) = \mathcal{N}(v; \mu_L, \sigma_L^2)$ is assumed. When $p_V^{(NL)}(v; \boldsymbol{\beta}_{NL})$ is modeled by a shifted Gaussian distribution, i.e.,

$$\mathcal{N}(v; \mu_{NL}, \sigma_{NL}^2) = \frac{1}{\sqrt{2\pi}\sigma_{NL}} \exp \left[\frac{-(v - \mu_{NL})^2}{2\sigma_{NL}^2} \right], \quad (3.55)$$

the Gaussian-Gaussian mixture distribution and its gradient are given, respectively, by

$$p_V(v) = \frac{\alpha_L}{\sqrt{2\pi}\sigma_L} \exp \left[\frac{-(v - \mu_L)^2}{2\sigma_L^2} \right] + \frac{\alpha_{NL}}{\sqrt{2\pi}\sigma_{NL}} \exp \left[\frac{-(v - \mu_{NL})^2}{2\sigma_{NL}^2} \right], \quad (3.56)$$

$$\nabla_v p_V(v) = \frac{-\alpha_L(v - \mu_L)}{\sqrt{2\pi}\sigma_L^3} \exp \left[\frac{-(v - \mu_L)^2}{2\sigma_L^2} \right] - \frac{\alpha_{NL}(v - \mu_{NL})}{\sqrt{2\pi}\sigma_{NL}^3} \exp \left[\frac{-(v - \mu_{NL})^2}{2\sigma_{NL}^2} \right]. \quad (3.57)$$

When $p_V^{(\text{NL})}(v; \boldsymbol{\beta}_{\text{NL}})$ is modeled by a two-mode Rayleigh distribution as given in (3.37), the Gaussian-Rayleigh mixture distribution and its gradient are given, respectively, by

$$p_V(v) = \begin{cases} \frac{\alpha_L}{\sqrt{2\pi}\sigma_L} \exp\left[-\frac{(v-\mu_L)^2}{2\sigma_L^2}\right] + \frac{\alpha_{\text{NL}}v}{\gamma_{\text{NL}}^2} \exp\left[\frac{-v^2}{2\gamma_{\text{NL}}^2}\right], & v \geq 0 \\ \frac{\alpha_L}{\sqrt{2\pi}\sigma_L} \exp\left[-\frac{(v-\mu_L)^2}{2\sigma_L^2}\right], & v < 0 \end{cases}, \quad (3.58)$$

$$\nabla_v p_V(v) = \begin{cases} \frac{-\alpha_L(v-\mu_L)}{\sqrt{2\pi}\sigma_L^3} \exp\left[-\frac{(v-\mu_L)^2}{2\sigma_L^2}\right] + \frac{\alpha_{\text{NL}}}{\gamma_{\text{NL}}^2} \exp\left[\frac{-v^2}{2\gamma_{\text{NL}}^2}\right] \left(1 - \frac{v^2}{\gamma_{\text{NL}}^2}\right), & v \geq 0 \\ \frac{-\alpha_L(v-\mu_L)}{\sqrt{2\pi}\sigma_L^3} \exp\left[-\frac{(v-\mu_L)^2}{2\sigma_L^2}\right], & v < 0 \end{cases}. \quad (3.59)$$

Chapter 4

Localization in Cellular Radio Networks: Parametric Modeling

In Chapter 3, the measurement error distribution is assumed to be completely unknown. The proposed robust iterative nonparametric (RIN) algorithm jointly estimates the unknown position and measurement error distribution. Optimal localization performance was demonstrated for large data records when the measurement error terms are truly iid. As a side product, an estimate of the measurement error distribution is also found non-parametrically via the adaptive kernel density estimation (AKDE). However, the use of the AKDE in the RIN algorithm leads to high computational complexity— $\mathcal{O}((NK)^2)$ floating-point operations (FLOPs) per iteration. In order to reduce the complexity to a reasonable level, a parametric model is adopted in this chapter to represent the measurement error distribution. Prior knowledge about the measurement error distribution can be either unknown or partially known. The use of such parametric model allows for better mathematical tractability and consequently leads to an expectation-conditional maximization (ECM) algorithm and a joint maximum *a posteriori*-maximum likelihood (JMAP-ML) algorithm. Similar to the RIN algorithm, the new proposed algorithms are iterative, which alternate between a position estimation step and a measurement error distribution estimation step. But the key difference lies in that an estimate of the measurement error distribution is found parametrically in the new algorithms. As a consequence, the proposed ECM- and JMAP-ML algorithms considerably reduce the computational complexity to $\mathcal{O}(CNK)$ FLOPs per iteration.

The organization of this chapter is as follows. Section 4.1 introduces the signal model and states the problem at hand. Section 4.2 first provides a general routine of implementing an ECM algorithm and a JMAP-ML algorithm, where a C -mode Gaussian mixture is adopted to approximate the unknown measurement error distribution. Then, the implementation details of the two algorithms are elaborated. Section 4.3 investigates the developed algorithms in terms of the convergence properties and computational complexity. Section 4.4 generalizes the parametric model to be a C -mode mixture of exponential family of distributions and provides an alternative way of reformulating an expectation maximization (EM) algorithm. In Section 4.5, the Cramér-Rao lower bound (CRLB) is computed, and the best achievable localization accuracy is further presented. Section 4.6 evaluates the proposed algorithms in various simulations. Two model mismatch problems are studied therein. Section 4.7 concludes this chapter and Section 4.8 assembles some useful derivations.

4.1 Signal Model and Problem Statement

Consider a scenario where N base stations (BSs) surround a stationary mobile station (MS) of interest in a cellular radio network. Let $\mathbf{p}_i = [x_i, y_i]^T$, $i = 1, 2, \dots, N$, be the *a priori* known positions of the i th BS and let $\boldsymbol{\theta}_p = [x, y]^T$ be the unknown position of the MS. Each BS collects K ($K \geq 1$) distance measurement(s) (time-of-arrival estimate(s) multiplied by c_0) $r_{i,k}$, $k = 1, 2, \dots, K$, and relays them to a fusion center for post-processing [99]. Figure 3.1 has already illustrated such a scenario and explained the notations therein. Assuming a precise time synchronization between the BSs and MS, the k th distance measurement $r_{i,k}$ measured at the i th BS can be expressed by

$$r_{i,k} = \underbrace{\sqrt{(x - x_i)^2 + (y - y_i)^2}}_{d_i(\boldsymbol{\theta}_p)} + v_{i,k} \quad (4.1)$$

where $d_i(\boldsymbol{\theta}_p) = \|\mathbf{p}_i - \boldsymbol{\theta}_p\|$ represents the actual Euclidean distance between the MS and the i th BS, and $v_{i,k}$ is the corresponding measurement error in $r_{i,k}$. As opposed to Chapter 3, a C -mode Gaussian mixture distribution is adopted in the parametric approach to the modeling of $p_V(v)$, i.e.,

$$p_V(v) \approx \hat{p}_V(v) = \sum_{l=1}^C \alpha_l \mathcal{N}(v; \mu_l, \sigma_l^2), \quad (4.2)$$

but the Gaussian mixture model parameters α_l , μ_l and σ_l^2 are assumed to be unknown and determined jointly with the unknown positions. The reason for adopting a linear superposition of Gaussians in (4.2) has been explained in Chapter 1.

For better readability, the signal model is expressed in a compact vector form as

$$\mathbf{r} = \mathbf{h}(\boldsymbol{\theta}_p) + \mathbf{v} \quad (4.3)$$

where

$$\mathbf{r} = [r_{1,1}, \dots, r_{1,K}, \dots, r_{N,1}, \dots, r_{N,K}]^T, \quad (4.4)$$

$$\mathbf{h}(\boldsymbol{\theta}_p) = \underbrace{[d_1(\boldsymbol{\theta}_p), \dots, d_1(\boldsymbol{\theta}_p)]}_{K \text{ repetitions}}, \dots, \underbrace{[d_N(\boldsymbol{\theta}_p), \dots, d_N(\boldsymbol{\theta}_p)]}_{K \text{ repetitions}}^T, \quad (4.5)$$

$$\mathbf{v}(\boldsymbol{\theta}_e) = [v_{1,1}, \dots, v_{1,K}, \dots, v_{N,1}, \dots, v_{N,K}]^T. \quad (4.6)$$

Column vectors \mathbf{r} , $\mathbf{h}(\boldsymbol{\theta}_p)$ and \mathbf{v} are all of dimension $NK \times 1$. Note that $\boldsymbol{\theta}_a$ is null and discarded in $\mathbf{h}(\boldsymbol{\theta}_p)$ throughout this chapter. The task is to jointly estimate the BS position $\boldsymbol{\theta}_p$ and the mixture model parameters $\boldsymbol{\theta}_e = [\alpha_1, \dots, \alpha_C, \mu_1, \sigma_1^2, \dots, \mu_C, \sigma_C^2]^T$. We aim at the maximum likelihood (ML) estimator of $\boldsymbol{\theta} = [\boldsymbol{\theta}_e^T, \boldsymbol{\theta}_p^T]^T$.

4.2 Joint ML Estimation

Based on the signal model in (4.3) and the parametric measurement error model in (4.2), the log-likelihood function of $\boldsymbol{\theta}$ is given by

$$\begin{aligned}\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r}) &= \ln(p(\mathbf{r}; \boldsymbol{\theta})) = \ln\left(\prod_{i=1}^N \prod_{k=1}^K p(r_{i,k}; \boldsymbol{\theta})\right) \\ &= \sum_{i=1}^N \sum_{k=1}^K \ln\left(\sum_{l=1}^C \alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)\right),\end{aligned}\quad (4.7)$$

and the ML estimator, $\hat{\boldsymbol{\theta}}^{\text{ML}}$, is obtained through solving

$$\begin{aligned}\arg \max_{\boldsymbol{\theta}} \quad & \mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r}) \\ \text{subject to} \quad & \alpha_l \geq 0, \quad l = 1, 2, \dots, C, \\ & \sum_{l=1}^C \alpha_l = 1.\end{aligned}\quad (4.8)$$

The cost function in (4.8) is cumbersome due to the ‘‘logarithm of summation’’. In order to approximate the ML estimator with low computational complexity, a complete-data set $\mathbf{z} = \{\mathbf{y}, \mathbf{r}\}$ is introduced with $\mathbf{y} = [y_{1,1}, \dots, y_{1,K}, \dots, y_{N,1}, \dots, y_{N,K}]^T$ being a vector of NK random variables (also called latent variables) whose value tell us which mixture component has generated the corresponding measurement error term. The complete-data log-likelihood function is easily expressed by

$$\begin{aligned}\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) &= \ln(p(\mathbf{y}, \mathbf{r}; \boldsymbol{\theta})) = \ln\left(\prod_{i=1}^N \prod_{k=1}^K p(y_{i,k}, r_{i,k}; \boldsymbol{\theta})\right) \\ &= \sum_{i=1}^N \sum_{k=1}^K \ln(\alpha_{y_{i,k}} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_{y_{i,k}}, \sigma_{y_{i,k}}^2))\end{aligned}\quad (4.9)$$

where the second equality relies on the assumption that $r_{i,k}$'s are independent and as a consequence $y_{i,k}$'s are also independent. To avoid ambiguity, the original log-likelihood function $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ in (4.7) is referred to as the incomplete-data log-likelihood function. It is clear that the newly introduced complete-data log-likelihood function $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ has a more tractable form, based on which an EM algorithm and a JMAP-ML algorithm are first developed. As will be seen, conventional way of optimizing $\boldsymbol{\theta}$ is still complicated, thus the conditional maximization [95] is introduced to meet this challenge.

4.2.1 EM Algorithm

As it was introduced in Section 2.2.1, the idea behind the EM criterion is to estimate the unknown parameters iteratively in two steps—an expectation (E)-step and a maximization (M)-step. In the first step, i.e., the E-step, statistical expectation of the complete-data log-likelihood is taken with respect to the conditional probability of the latent variables. In the second step, i.e., the M-step, the conditional expectation obtained above is maximized with respect to the parameters of interest. The two steps iterate until a predetermined convergence condition is met. Given the *a priori* parameter estimate $\boldsymbol{\theta}^{(\eta)}$, we show in the sequel the work-flow of the proposed EM algorithm on the $(\eta + 1)$ th iteration.

The first step (E-step): Let us first define the conditional expectation of the complete-data log-likelihood as follows:

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) = \sum_{\mathbf{y}} \ln(p(\mathbf{y}, \mathbf{r}; \boldsymbol{\theta})) \Pr\{\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)}\}. \quad (4.10)$$

Follow the derivations shown in Appendix 4.8.1,

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) = \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^C \ln(\alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) P_{i,k,l}^{(\eta)} \quad (4.11)$$

where $P_{i,k,l}^{(\eta)}$ is a short-hand notation of the conditional probability $\Pr\{y_{i,k} = l | r_{i,k}; \boldsymbol{\theta}^{(\eta)}\}$, which can be computed by means of Bayes' rule as follows:

$$P_{i,k,l}^{(\eta)} = \Pr\{y_{i,k} = l | r_{i,k}; \boldsymbol{\theta}^{(\eta)}\} = \frac{\alpha_l^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_l^{(\eta)}, \sigma_l^{2,(\eta)})}{p(r_{i,k}; \boldsymbol{\theta}^{(\eta)})} \quad (4.12)$$

with

$$p(r_{i,k}; \boldsymbol{\theta}^{(\eta)}) = \sum_{l=1}^C \alpha_l^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_l^{(\eta)}, \sigma_l^{2,(\eta)}). \quad (4.13)$$

The second step (M-Step): We maximize $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$ derived in (4.11) with respect to the vector parameter $\boldsymbol{\theta}$ and obtain on the $(\eta + 1)$ th iteration

$$\boldsymbol{\theta}^{(\eta+1)} = \arg \max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}). \quad (4.14)$$

4.2.2 JMAP-ML Algorithm

As another means of approximating the ML estimator, we next adopt the JMAP-ML criterion, where the complete-data log-likelihood function is maximized directly with respect to both $\boldsymbol{\theta}$ and \mathbf{y} , that is,

$$\arg \max_{\boldsymbol{\theta}, \mathbf{y}} \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) \equiv \arg \max_{\boldsymbol{\theta}} \left\{ \arg \max_{\mathbf{y}} \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) \right\}. \quad (4.15)$$

As it was introduced in Section 2.2.3, the JMAP-ML criterion incorporates a maximum *a posteriori* (MAP) estimation step (in terms of the latent variables in \mathbf{y}) and an ML estimation step (in terms of the deterministic parameters in $\boldsymbol{\theta}$). The work-flow of the proposed JMAP-ML algorithm on the $(\eta + 1)$ th iteration is as follows.

The first step (MAP estimation of \mathbf{y}): Let us first re-write the log-likelihood function of the complete-data as follows:

$$\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) = \ln(p(\mathbf{y}, \mathbf{r}; \boldsymbol{\theta})) = \ln(\Pr\{\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}\}) + \ln(p(\mathbf{r}; \boldsymbol{\theta})) \quad (4.16)$$

where the term $\ln(p(\mathbf{r}; \boldsymbol{\theta}))$ is independent of \mathbf{y} . Replacing $\boldsymbol{\theta}$ with $\boldsymbol{\theta}^{(\eta)}$ in (4.16) and solving for the MAP estimate of \mathbf{y} yields

$$\mathbf{y}^{(\eta+1)} = \arg \max_{\mathbf{y}} \ln(\Pr\{\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)}\}), \quad (4.17)$$

which can be converted into NK simpler ones

$$y_{i,k}^{(\eta+1)} = \arg \max_{y_{i,k}} \ln(\Pr\{y_{i,k}|r_{i,k}; \boldsymbol{\theta}^{(\eta)}\}). \quad (4.18)$$

Since $y_{i,k}$ is discrete-valued, the global optimal solution to (4.18) must be one of $\{y_{i,k} = 1, 2, \dots, C\}$ that maximizes $\ln(\Pr\{y_{i,k}|r_{i,k}; \boldsymbol{\theta}^{(\eta)}\})$. Since logarithm is a monotonic operation, we need only to compare

$$\Phi_{i,k,l}^{(\eta)} = \alpha_l^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_l^{(\eta)}, \sigma_l^{2,(\eta)}), \quad l = 1, 2, \dots, C. \quad (4.19)$$

The second step (ML estimation of $\boldsymbol{\theta}$): Substituting the obtained MAP estimate $\mathbf{y}^{(\eta+1)}$ into the complete-data log-likelihood $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ yields

$$\begin{aligned} \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}^{(\eta+1)}, \mathbf{r}) &= \sum_{i=1}^N \sum_{k=1}^K \ln \left(\alpha_{y_{i,k}^{(\eta+1)}} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_{y_{i,k}^{(\eta+1)}}^{(\eta+1)}, \sigma_{y_{i,k}^{(\eta+1)}}^2) \right) \\ &= \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^C \ln(\alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) \delta(l - y_{i,k}^{(\eta+1)}) \end{aligned} \quad (4.20)$$

where

$$\delta(l - y_{i,k}^{(\eta+1)}) = \begin{cases} 1, & \text{if } l = y_{i,k}^{(\eta+1)} \\ 0, & \text{otherwise} \end{cases} \quad (4.21)$$

is a Kronecker's delta function. On the $(\eta+1)$ th iteration, we maximize $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}^{(\eta+1)}, \mathbf{r})$ with respect to $\boldsymbol{\theta}$ and obtain

$$\boldsymbol{\theta}^{(\eta+1)} = \arg \max_{\boldsymbol{\theta}} \mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}^{(\eta+1)}, \mathbf{r}). \quad (4.22)$$

Remark 4.1. *The cost functions (4.11) in the EM algorithm and (4.20) in the JMAP-ML algorithm can be unified as*

$$\Lambda^{(\eta)}(\boldsymbol{\theta}) = \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^C \ln(\alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) w_{i,k,l}^{(\eta)} \quad (4.23)$$

where merely the weighting factor $w_{i,k,l}^{(\eta)}$ is distinguished by

$$w_{i,k,l}^{(\eta)} = \begin{cases} P_{i,k,l}^{(\eta)}, & \text{EM algorithm} \\ \delta(l - y_{i,k}^{(\eta+1)}), & \text{JMAP-ML algorithm} \end{cases}. \quad (4.24)$$

It is interesting to see that $w_{i,k,l}^{(\eta)}$ corresponds to a “soft fusion” of information in the EM algorithm whereas a “hard fusion” of information in the JMAP-ML algorithm.

Remark 4.2. *When the measurement error terms are truly iid and no model mismatch is assumed in (4.2), the EM algorithm is able to reproduce the ML estimator that globally maximizes the incomplete-data log-likelihood function in (4.7) [93]. However, the JMAP-ML algorithm can merely produce a biased and inconsistent estimator [97].*

Although $w_{i,k,l}^{(\eta)}$ differs in the two algorithms, it is *a priori* determined and contains no optimization variable, meaning that we could follow the same strategy to optimize (4.11) and (4.20). Consequently, the corresponding results differ only by the weighting factors. However, we also noticed that directly optimizing $\Lambda^{(\eta)}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ can be complicated when the dimension of $\boldsymbol{\theta}$ is large. In order to tackle this problem, we apply the conditional maximization (CM) as introduced in Chapter 2 for solving (4.11) and (4.20). The idea is as follows. First, we choose a proper partition of the unknown parameters, i.e.,

$$\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \boldsymbol{\vartheta}_2^T, \dots, \boldsymbol{\vartheta}_S^T]^T \quad (4.25)$$

where $\boldsymbol{\vartheta}_s$, $s = 1, 2, \dots, S$ is a sub-vector of $\boldsymbol{\theta}$. Furthermore, we let

$$\mathcal{G} = \{g_s(\boldsymbol{\theta}) : s = 1, 2, \dots, S\} \quad (4.26)$$

be a set of S preselected vector functions of $\boldsymbol{\theta}$ defined by

$$g_s(\boldsymbol{\theta}) = [\boldsymbol{\vartheta}_1^T, \dots, \boldsymbol{\vartheta}_{s-1}^T, \boldsymbol{\vartheta}_{s+1}^T, \dots, \boldsymbol{\vartheta}_S^T]^T, \quad (4.27)$$

meaning that $g_s(\boldsymbol{\theta})$ is a vector that contains all the sub-vectors of $\boldsymbol{\theta}$ except $\boldsymbol{\vartheta}_s$. With the above partition of $\boldsymbol{\theta}$, we can convert the complicated maximization problem in (4.23) into S easier ones. More precisely, the s th CM step of the $(\eta + 1)$ th iteration solves $\boldsymbol{\theta}^{(\eta+s/S)}$ from

$$\begin{aligned} \arg \max_{\boldsymbol{\theta}} \quad & \Lambda^{(\eta)}(\boldsymbol{\theta}) \\ \text{subject to} \quad & g_s(\boldsymbol{\theta}) = g_s(\boldsymbol{\theta}^{(\eta+(s-1)/S)}). \end{aligned} \quad (4.28)$$

According to Section 2.2.2, the resulting EM-type algorithm is referred to as the ECM algorithm.

4.2.3 Implementation Details

In the previous subsection, we gave a general routine for implementing an ECM algorithm and a JMAP-ML algorithm for our joint estimation problem. As a complement, this subsection elaborates on the implementation of the CM steps. We start with a toy example, where C is set to two in the parametric model in (4.2), namely,

$$p_V(v) = \sum_{l=1}^2 \alpha_l \mathcal{N}(v; \mu_l, \sigma_l^2). \quad (4.29)$$

Here, we assume σ_l^2 , $l = 1, 2$, to be strictly larger than zero. In this example, the vector parameter to be estimated is $\boldsymbol{\theta} = [\alpha_1, \alpha_2, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2, x, y]^T$ and the cost function in (4.23) is written as

$$\Lambda^{(\eta)}(\boldsymbol{\theta}) = \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^2 \ln(\alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) w_{i,k,l}^{(\eta)}. \quad (4.30)$$

In the sequel, the partition of $\boldsymbol{\theta}$ is chosen to be $\boldsymbol{\vartheta}_1 = [\alpha_1, \alpha_2]^T$, $\boldsymbol{\vartheta}_2 = \mu_1$, $\boldsymbol{\vartheta}_3 = \sigma_1^2$, $\boldsymbol{\vartheta}_4 = \mu_2$, $\boldsymbol{\vartheta}_5 = \sigma_2^2$ and $\boldsymbol{\vartheta}_6 = \boldsymbol{\theta}_p$. Hence, we have $S = 6$ in this example. We note that different partitions of $\boldsymbol{\theta}$ lead to different conditional maximization steps. Next, we re-formulate the cost function $\Lambda^{(\eta)}(\boldsymbol{\theta})$ in (4.30) as

$$\Lambda^{(\eta)}(\boldsymbol{\theta}) = \Lambda_0^{(\eta)}(\alpha_1, \alpha_2) + \sum_{l=1}^2 \Lambda_l^{(\eta)}(\mu_l, \sigma_l^2, \boldsymbol{\theta}_p) \quad (4.31)$$

where

$$\Lambda_0^{(\eta)}(\alpha_1, \alpha_2) = \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^2 \ln(\alpha_l) w_{i,k,l}^{(\eta)} \quad (4.32)$$

and for $l = 1, 2$,

$$\begin{aligned} \Lambda_l^{(\eta)}(\mu_l, \sigma_l^2, \boldsymbol{\theta}_p) &= \sum_{i=1}^N \sum_{k=1}^K \ln(\mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) w_{i,k,l}^{(\eta)} \\ &= \sum_{i=1}^N \sum_{k=1}^K \left(-\frac{\ln(2\pi\sigma_l^2)}{2} - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l)^2}{2\sigma_l^2} \right) w_{i,k,l}^{(\eta)}. \end{aligned} \quad (4.33)$$

In the first CM step of the $(\eta + 1)$ th iteration, we solve $\boldsymbol{\theta}^{(\eta+1/S)}$ from

$$\begin{aligned} \arg \max_{\boldsymbol{\theta}} \quad & \Lambda^{(\eta)}(\boldsymbol{\theta}) \\ \text{subject to} \quad & \mu_1 = \mu_1^{(\eta)}, \sigma_1^2 = \sigma_1^{2,(\eta)}, \mu_2 = \mu_2^{(\eta)}, \sigma_2^2 = \sigma_2^{2,(\eta)}, \boldsymbol{\theta}_p = \boldsymbol{\theta}_p^{(\eta)}. \end{aligned} \quad (4.34)$$

This is equivalent to updating $\alpha_l^{(\eta+1)}$, $l = 1, 2$, from

$$\frac{\partial}{\partial \alpha_l} \left[\Lambda_0^{(\eta)}(\alpha_1, \alpha_2) + \lambda(\alpha_1 + \alpha_2 - 1) \right] = 0 \quad (4.35)$$

where λ is the Lagrange multiplier. After some simple manipulations, we have

$$\alpha_l^{(\eta+1)} = -\frac{1}{\lambda} \sum_{i=1}^N \sum_{k=1}^K w_{i,k,l}^{(\eta)}. \quad (4.36)$$

The Lagrange multiplier λ in (4.36) is calculated to be equal to $-NK$ because

$$\underbrace{\sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^2 w_{i,k,l}^{(\eta)}}_{=NK} = -\lambda \underbrace{\sum_{l=1}^2 \alpha_l^{(\eta+1)}}_{=1}. \quad (4.37)$$

Consequently, we obtain after the first CM step

$$\boldsymbol{\theta}^{(\eta+1/S)} = [\alpha_1^{(\eta+1)}, \alpha_2^{(\eta+1)}, \mu_1^{(\eta)}, \sigma_1^{2,(\eta)}, \mu_2^{(\eta)}, \sigma_2^{2,(\eta)}, (\boldsymbol{\theta}_p^{(\eta)})^T]^T. \quad (4.38)$$

In the second CM step of the $(\eta + 1)$ th iteration, we solve $\boldsymbol{\theta}^{(\eta+2/S)}$ from

$$\begin{aligned} \arg \max_{\boldsymbol{\theta}} \quad & \Lambda^{(\eta)}(\boldsymbol{\theta}) \\ \text{subject to} \quad & \alpha_1 = \alpha_1^{(\eta+1)}, \alpha_2 = \alpha_2^{(\eta+1)}, \sigma_1^2 = \sigma_1^{2,(\eta)}, \mu_2 = \mu_2^{(\eta)}, \sigma_2^2 = \sigma_2^{2,(\eta)}, \boldsymbol{\theta}_p = \boldsymbol{\theta}_p^{(\eta)}. \end{aligned} \quad (4.39)$$

This is equivalent to updating $\mu_1^{(\eta+1)}$ from

$$\frac{\partial}{\partial \mu_1} \sum_{i=1}^N \sum_{k=1}^K \left(-\frac{\ln(2\pi\sigma_1^{2,(\eta)})}{2} - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_1)^2}{2\sigma_1^{2,(\eta)}} \right) w_{i,k,1}^{(\eta)} = 0, \quad (4.40)$$

which is obtained by taking the derivative of $\Lambda_1^{(\eta)}(\mu_1, \sigma_1^{2,(\eta)}, \boldsymbol{\theta}_p^{(\eta)})$ with respect to μ_1 and setting it to zero. Equation (4.40) can be reduced to be

$$\sum_{i=1}^N \sum_{k=1}^K \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_1)}{\sigma_1^{2,(\eta)}} w_{i,k,1}^{(\eta)} = 0. \quad (4.41)$$

Assuming $\sigma_1^{2,(\eta)}$ is strictly larger than zero and solving for $\mu_1^{(\eta+1)}$ yields

$$\mu_1^{(\eta+1)} = \frac{\sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)})) w_{i,k,1}^{(\eta)}}{\sum_{i=1}^N \sum_{k=1}^K w_{i,k,1}^{(\eta)}}. \quad (4.42)$$

Hence, after the second CM step, we have

$$\boldsymbol{\theta}^{(\eta+2/S)} = [\alpha_1^{(\eta+1)}, \alpha_2^{(\eta+1)}, \mu_1^{(\eta+1)}, \sigma_1^{2,(\eta)}, \mu_2^{(\eta)}, \sigma_2^{2,(\eta)}, (\boldsymbol{\theta}_p^{(\eta)})^T]^T. \quad (4.43)$$

In the third CM step of the $(\eta + 1)$ th iteration, we solve $\boldsymbol{\theta}^{(\eta+3/S)}$ from

$$\begin{aligned} \arg \max_{\boldsymbol{\theta}} \quad & \Lambda^{(\eta)}(\boldsymbol{\theta}) & (4.44) \\ \text{subject to} \quad & \alpha_1 = \alpha_1^{(\eta+1)}, \alpha_2 = \alpha_2^{(\eta+1)}, \mu_1 = \mu_1^{(\eta+1)}, \mu_2 = \mu_2^{(\eta)}, \sigma_2^2 = \sigma_2^{2,(\eta)}, \boldsymbol{\theta}_p = \boldsymbol{\theta}_p^{(\eta)}. & (4.45) \end{aligned}$$

This is equivalent to updating $\sigma_1^{2,(\eta+1)}$ from

$$\frac{\partial}{\partial \sigma_1^2} \sum_{i=1}^N \sum_{k=1}^K \left(-\frac{\ln(2\pi\sigma_1^2)}{2} - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_1^{(\eta+1)})^2}{2\sigma_1^2} \right) w_{i,k,1}^{(\eta)} = 0, \quad (4.46)$$

which is obtained by taking the derivative of $\Lambda_1^{(\eta)}(\mu_1^{(\eta+1)}, \sigma_1^2, \boldsymbol{\theta}_p^{(\eta)})$ with respect to σ_1^2 and setting it equal to zero. Equation (4.46) can be reduced to be

$$\sum_{i=1}^N \sum_{k=1}^K \frac{w_{i,k,1}^{(\eta)}}{\sigma_1^2} - \sum_{i=1}^N \sum_{k=1}^K \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_1^{(\eta+1)})^2 w_{i,k,1}^{(\eta)}}{\sigma_1^4} = 0, \quad (4.47)$$

from which solving for $\sigma_1^{2,(\eta+1)}$ yields

$$\sigma_1^{2,(\eta+1)} = \frac{\sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_1^{(\eta+1)})^2 w_{i,k,1}^{(\eta)}}{\sum_{i=1}^N \sum_{k=1}^K w_{i,k,1}^{(\eta)}}. \quad (4.48)$$

Hence, after the third CM step, we obtain

$$\boldsymbol{\theta}^{(\eta+3/S)} = [\alpha_1^{(\eta+1)}, \alpha_2^{(\eta+1)}, \mu_1^{(\eta+1)}, \sigma_1^{2,(\eta+1)}, \mu_2^{(\eta)}, \sigma_2^{2,(\eta)}, (\boldsymbol{\theta}_p^{(\eta)})^T]^T. \quad (4.49)$$

Similar to the second step, an updated estimate $\mu_2^{(\eta+1)}$ is found to be

$$\mu_2^{(\eta+1)} = \frac{\sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)})) w_{i,k,2}^{(\eta)}}{\sum_{i=1}^N \sum_{k=1}^K w_{i,k,2}^{(\eta)}} \quad (4.50)$$

in the fourth CM step, and

$$\boldsymbol{\theta}^{(\eta+4/S)} = [\alpha_1^{(\eta+1)}, \alpha_2^{(\eta+1)}, \mu_1^{(\eta+1)}, \sigma_1^{2,(\eta+1)}, \mu_2^{(\eta+1)}, \sigma_2^{2,(\eta)}, (\boldsymbol{\theta}_p^{(\eta)})^T]^T. \quad (4.51)$$

Similar to the third step, an updated estimate $\sigma_2^{2,(\eta+1)}$ is found to be

$$\sigma_2^{2,(\eta+1)} = \frac{\sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_2^{(\eta+1)})^2 w_{i,k,2}^{(\eta)}}{\sum_{i=1}^N \sum_{k=1}^K w_{i,k,2}^{(\eta)}} \quad (4.52)$$

in the fifth CM step, and

$$\boldsymbol{\theta}^{(\eta+5/S)} = [\alpha_1^{(\eta+1)}, \alpha_2^{(\eta+1)}, \mu_1^{(\eta+1)}, \sigma_1^{2,(\eta+1)}, \mu_2^{(\eta+1)}, \sigma_2^{2,(\eta+1)}, (\boldsymbol{\theta}_p^{(\eta)})^T]^T. \quad (4.53)$$

In the sixth CM step (i.e., the last CM step) of the $(\eta + 1)$ th iteration, we solve $\boldsymbol{\theta}^{(\eta+6/S)} = \boldsymbol{\theta}^{(\eta+1)}$ from

$$\begin{aligned} & \arg \max_{\boldsymbol{\theta}} \quad \Lambda^{(\eta)}(\boldsymbol{\theta}) \\ & \text{subject to} \quad \alpha_1 = \alpha_1^{(\eta+1)}, \alpha_2 = \alpha_2^{(\eta+1)}, \mu_1 = \mu_1^{(\eta+1)}, \sigma_1^2 = \sigma_1^{2,(\eta+1)}, \mu_2 = \mu_2^{(\eta+1)}, \sigma_2^2 = \sigma_2^{2,(\eta+1)}. \end{aligned} \quad (4.54)$$

It is easy to see that the last CM step attempts to update the position estimate, $\boldsymbol{\theta}_p^{(\eta+1)}$, through solving

$$\boldsymbol{\theta}_p^{(\eta+1)} = \arg \max_{\boldsymbol{\theta}_p} \sum_{l=1}^2 \Lambda_l^{(\eta)}(\mu_l^{(\eta+1)}, \sigma_l^{2,(\eta+1)}, \boldsymbol{\theta}_p), \quad (4.55)$$

which can be reduced, after some tedious manipulations, to be

$$\boldsymbol{\theta}_p^{(\eta+1)} = \arg \min_{\boldsymbol{\theta}_p} \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^2 \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)})^2}{\sigma_l^{2,(\eta+1)}} w_{i,k,l}^{(\eta)}. \quad (4.56)$$

Unfortunately, a closed form solution does not exist because $d_i(\boldsymbol{\theta}_p)$ is nonlinear in terms of both x and y . Hence, we resort to numerical methods. In order to keep this position update step computationally fast and meanwhile maintain a good numerical result, we adopt here the BFGS quasi-Newton (QN) method as introduced in Chapter 3 as it

guarantees downhill progress towards the local minimum in each Newton step [105]. Finally, we obtain on the $(\eta + 1)$ th iteration

$$\boldsymbol{\theta}^{(\eta+1)} = [\alpha_1^{(\eta+1)}, \alpha_2^{(\eta+1)}, \mu_1^{(\eta+1)}, \sigma_1^{2,(\eta+1)}, \mu_2^{(\eta+1)}, \sigma_2^{2,(\eta+1)}, (\boldsymbol{\theta}_p^{(\eta+1)})^T]^T. \quad (4.57)$$

The methodology demonstrated above can be easily applied to the general case where $C > 2$. For simplicity, we summarize the key results in Algorithm 4.1. Therein, one of the convergence conditions is selected according to the convergence properties given in Section 4.3.1.

Before closing this subsection, we briefly comment on the difference between the non-parametric approach to the modeling of $p_V(v)$ in the RIN algorithm and the parametric approach to the modeling of $p_V(v)$ in the ECM- and JMAP-ML algorithms. In the former approach, the pdf estimate is constructed by a linear superposition of NK Gaussian kernel densities with identical weighting factor (prior probability) $1/NK$ and each residual extracted from the corresponding distance measurement contributes to one such kernel. However, in the latter approach, several different residuals extracted from the corresponding distance measurements are more likely to be generated from a same Gaussian kernel, which can be easily seen from the probabilistic assignment in both (4.12) for the EM algorithm and (4.21) for the JMAP-ML algorithm. Hence, fixing a small C (relative to NK) in the parametric approach is analogous to appending an extra Kernel pruning/merging step after the kernel density estimation. Besides, we have more freedom to choose C (as compared to the fixed number NK in the AKDE), depending on the estimation accuracy we desire and the computational cost we are able to afford. Lastly but most importantly, both the ECM algorithm and JMAP-ML algorithm are guaranteed to converge when a C -mode Gaussian mixture is adopted to approximate the unknown $p_V(v)$, while the RIN algorithm is not ensured to converge.

4.3 Performance Evaluation

In this section, the proposed algorithms will be studied in terms of the convergence properties in Section 4.3.1 and computational complexity in Section 4.3.2.

4.3.1 Convergence Properties

As it was shown in Algorithm 4.1, both the ECM algorithm and JMAP-ML algorithm are iterative in nature. The following theorems show their convergence properties.

Algorithm 4.1 ECM- and JMAP-ML Algorithms (Assuming a C -mode Gaussian Mixture in the Parametric Model)

Step 1—Initialization:

Choose a convergence tolerance Δ and the maximum number of iterations N_{itr} . Set the iteration index $\eta = 0$. Choose an initial guess $\boldsymbol{\theta}^{(0)}$.

Step 2—Joint Estimation:

On the $(\eta + 1)$ th iteration ($\eta \in \mathbb{Z}, \eta \geq 0$),

- Compute $w_{i,k,l}^{(\eta)}$ according to (4.24) for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$, and $l = 1, 2, \dots, C$.
- Choose a partition of $\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \boldsymbol{\vartheta}_2^T, \dots, \boldsymbol{\vartheta}_{2C+2}^T]^T$ with $\boldsymbol{\vartheta}_1 = [\alpha_1, \alpha_2, \dots, \alpha_C]^T$, $\boldsymbol{\vartheta}_{2l} = \mu_l$, $\boldsymbol{\vartheta}_{2l+1} = \sigma_l^2$, for $l = 1, 2, \dots, C$, and $\boldsymbol{\vartheta}_{2C+2} = \boldsymbol{\theta}_p$.
- Solve $\boldsymbol{\theta}_e^{(\eta+1)}$ in the first $2C + 1$ CM steps and obtain closed form expressions:

$$\alpha_l^{(\eta+1)} = \frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K w_{i,k,l}^{(\eta)}, \quad (4.58)$$

$$\mu_l^{(\eta+1)} = \frac{\sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)})) w_{i,k,l}^{(\eta)}}{NK \alpha_l^{(\eta+1)}}, \quad (4.59)$$

$$\sigma_l^{2,(\eta+1)} = \frac{\sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}))^2 w_{i,k,l}^{(\eta)}}{NK \alpha_l^{(\eta+1)}} - (\mu_l^{(\eta+1)})^2, \quad (4.60)$$

for $l = 1, 2, \dots, C$.

- Solve $\boldsymbol{\theta}_p^{(\eta+1)}$ in the $(2C + 2)$ th CM step and obtain a numerical solution

$$\boldsymbol{\theta}_p^{(\eta+1)} = \arg \min_{\boldsymbol{\theta}_p} \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^C \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)})^2}{\sigma_l^{2,(\eta+1)}} w_{i,k,l}^{(\eta)}. \quad (4.61)$$

- As a side-product of $\boldsymbol{\theta}^{(\eta+1)}$, we obtain on the $(\eta + 1)$ th iteration an estimate of $p_V(v)$ as:

$$\hat{p}_V(v) = \sum_{l=1}^C \alpha_l^{(\eta+1)} \mathcal{N}(v; \mu_l^{(\eta+1)}, \sigma_l^{2,(\eta+1)}). \quad (4.62)$$

Step 3—Convergence Check:

If the increment of the log-likelihood value is less than Δ or N_{itr} has been reached, then terminate this algorithm; otherwise set $\eta \leftarrow \eta + 1$ and return to **Step 2**.

Theorem 4.1. *The proposed ECM algorithm in Algorithm 4.1 is a generalized EM (GEM) algorithm and ensures that the sequence of incomplete-data log-likelihood values $\{\mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r})\}$, when bounded above, converges monotonically over iterations to some point L^* .*

Proof. As is verified in Appendix 4.8.2, $\alpha_l^{(\eta+1)}$, $\mu_l^{(\eta+1)}$ and $\sigma_l^{2,(\eta+1)}$, $l = 1, 2, \dots, C$, are global maximizers found in each conditional maximization step. Therefore,

$$Q(\boldsymbol{\theta}^{(\eta+s/S)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}), \quad s = 1, 2, \dots, 2C + 1, \quad (4.63)$$

for any $\boldsymbol{\theta} \in \Theta_s(\boldsymbol{\theta}^{(\eta+(s-1)/S)}) = \{\boldsymbol{\theta} \in \Theta : g_s(\boldsymbol{\theta}) = g_s(\boldsymbol{\theta}^{(\eta+(s-1)/S)})\}$. In other words, $\boldsymbol{\theta}^{(\eta+s/S)}$ is the global maximizer in the given subspace of Θ . Therefore, it can be concluded that

$$\begin{aligned} Q(\boldsymbol{\theta}_e^{(\eta+1)}, \boldsymbol{\theta}_p^{(\eta)}; \boldsymbol{\theta}^{(\eta)}) &= Q(\boldsymbol{\theta}^{(\eta+2C+1/S)}; \boldsymbol{\theta}^{(\eta)}) \\ &\geq Q(\boldsymbol{\theta}^{(\eta+2C/S)}; \boldsymbol{\theta}^{(\eta)}) \\ &\geq Q(\boldsymbol{\theta}^{(\eta+2C-1/S)}; \boldsymbol{\theta}^{(\eta)}) \\ &\vdots \\ &\geq Q(\boldsymbol{\theta}^{(\eta)}; \boldsymbol{\theta}^{(\eta)}). \end{aligned} \quad (4.64)$$

In the $(2C+2)$ th conditional maximization step, the position estimate $\boldsymbol{\theta}_p^{(\eta+1)}$ is updated via the BFGS-QN method with the initial guess, $\boldsymbol{\theta}_p^{(\eta)}$. As mentioned beforehand, it guarantees downhill progress towards the local minimum in each Newton step, thus it can be concluded that the new position estimate $\boldsymbol{\theta}_p^{(\eta+1)}$ will not decrease $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$ on the $(\eta + 1)$ th iteration. Therefore,

$$Q(\boldsymbol{\theta}^{(\eta+1)}; \boldsymbol{\theta}^{(\eta)}) = Q(\boldsymbol{\theta}_e^{(\eta+1)}, \boldsymbol{\theta}_p^{(\eta+1)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}_e^{(\eta+1)}, \boldsymbol{\theta}_p^{(\eta)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}^{(\eta)}; \boldsymbol{\theta}^{(\eta)}). \quad (4.65)$$

Hence, the proposed ECM algorithm is essentially a GEM algorithm according to Section 2.2.1. When $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ is bounded above, which holds under the assumption that $\sigma_l^2 > 0$, $l = 1, 2, \dots, C$, the proposed ECM algorithm converges monotonically over iterations to some value L^* of the incomplete-data log-likelihood function $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$. \square

Theorem 4.2. *When the position update found by the BFGS-QN method in the $(2C + 2)$ th CM step is a global maximizer, L^* is ensured to be a stationary point for the proposed ECM algorithm.*

Proof. For L^* to be a stationary point, however, we need to further prove, according to Section 2.2.2, that (1) $g_s(\boldsymbol{\theta})$ is differentiable; (2) the corresponding gradient (or

Jacobian matrix) $\nabla_{\theta} g_s(\boldsymbol{\theta})$ is of full rank at $\boldsymbol{\theta}^{(\eta)} \in \Theta$, for all η ; and (3) the “space filling” condition holds as

$$\bigcap_{s=1}^S G_s(\boldsymbol{\theta}^{(\eta)}) = \{\mathbf{0}\}, \quad \text{for all } \eta, \quad (4.66)$$

where $G_s(\boldsymbol{\theta})$ is the column space of the matrix $\nabla_{\theta} g_s(\boldsymbol{\theta})$. In Algorithm 4.1, we choose the partition of $\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \boldsymbol{\vartheta}_2^T, \dots, \boldsymbol{\vartheta}_S^T]^T$, where $S = 2C + 2$, $\boldsymbol{\vartheta}_1 = [\alpha_1, \alpha_2, \dots, \alpha_C]^T$, $\boldsymbol{\vartheta}_{2l} = \mu_l$, $\boldsymbol{\vartheta}_{2l+1} = \sigma_l^2$, $l = 1, 2, \dots, C$, and $\boldsymbol{\vartheta}_{2C+2} = \boldsymbol{\theta}_p = [x, y]^T$. The dimension of $\boldsymbol{\theta}$ is $\dim(\boldsymbol{\theta}) = 3C + 2$.

It is easy to show that

$$\nabla_{\theta} g_s(\boldsymbol{\theta}^{(\eta)}) = \begin{cases} [\mathbf{e}_{C+1}, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})}], & s = 1 \\ [\mathbf{e}_1, \dots, \mathbf{e}_{C+s-2}, \mathbf{e}_{C+s}, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})}], & s = 2, \dots, 2C + 1 \\ [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{3C}], & s = S = 2C + 2 \end{cases} \quad (4.67)$$

are differentiable and irrespective of $\boldsymbol{\theta}^{(\eta)}$ because

$$\mathbf{e}_j = \left[\underbrace{0, \dots, 0}_{j-1 \text{ copies}}, 1, \underbrace{0, \dots, 0}_{\dim(\boldsymbol{\theta})-j \text{ copies}} \right]^T, \quad \forall j \in \{1, 2, \dots, \dim(\boldsymbol{\theta})\}. \quad (4.68)$$

It is clear that $\nabla_{\theta} g_1(\boldsymbol{\theta}^{(\eta)})$ is of dimension $\dim(\boldsymbol{\theta}) \times (\dim(\boldsymbol{\theta}) - C)$; and $\nabla_{\theta} g_s(\boldsymbol{\theta}^{(\eta)})$, for $s = 2, \dots, 2C + 1$, are all of dimension $\dim(\boldsymbol{\theta}) \times (\dim(\boldsymbol{\theta}) - 1)$; and $\nabla_{\theta} g_S(\boldsymbol{\theta}^{(\eta)})$ is of dimension $\dim(\boldsymbol{\theta}) \times (\dim(\boldsymbol{\theta}) - 2)$. Column vectors \mathbf{e}_j , $j = 1, 2, \dots, \dim(\boldsymbol{\theta})$, are all of dimension $\dim(\boldsymbol{\theta}) \times 1$. Moreover, \mathbf{e}_j and $\mathbf{e}_{j'}$ are mutually orthogonal if $j \neq j'$. For any $s = 1, 2, \dots, S$, $\nabla_{\theta} g_s(\boldsymbol{\theta}^{(\eta)})$ has a full column rank. So far, the first two conditions have been proven. In the sequel, the superscript η is omitted for brevity.

The proof of the third condition starts with the definition of the column space, that is, $G_s(\boldsymbol{\theta})$ is a linear combination of the columns of the matrix $\nabla_{\theta} g_s(\boldsymbol{\theta})$, i.e.,

$$G_s(\boldsymbol{\theta}) = \begin{cases} \sum_{j \in \{C+1, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j, & s = 1 \\ \sum_{j \in \{1, \dots, \dim(\boldsymbol{\theta})\} \setminus \{C+s-1\}} c_j \mathbf{e}_j, & s = 2, \dots, 2C + 1 \\ \sum_{j \in \{1, \dots, 3C\}} c_j \mathbf{e}_j, & s = S = 2C + 2 \end{cases} \quad (4.69)$$

where c_j is a real scalar coefficient of \mathbf{e}_j . Since $G_s(\boldsymbol{\theta})$ is a subspace of $\mathbb{R}^{\dim(\boldsymbol{\theta})}$, (4.66) is reformulated as

$$\bigcap_{s=1}^S G_s(\boldsymbol{\theta}) = \bigcap_{s=0}^S G_s(\boldsymbol{\theta}) = G_S(\boldsymbol{\theta}) \cap G_{S-1}(\boldsymbol{\theta}) \cap \dots \cap G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta}) \quad (4.70)$$

where $G_0(\boldsymbol{\theta})$ is the whole space of $\mathbb{R}^{\dim(\boldsymbol{\theta})}$, spanned by $\sum_{j \in \{1, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j$. The right-hand-side of the second equation in (4.70) can be performed sequentially in the order $s = 1, 2, \dots, S$ as

$$\tilde{G}_s(\boldsymbol{\theta}) = G_s(\boldsymbol{\theta}) \cap \tilde{G}_{s-1}(\boldsymbol{\theta}) \quad (4.71)$$

where

$$\tilde{G}_{s-1}(\boldsymbol{\theta}) = G_{s-1}(\boldsymbol{\theta}) \cap G_{s-2}(\boldsymbol{\theta}) \cap \dots \cap G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta}) \quad (4.72)$$

for $s > 1$ and $\tilde{G}_{s-1}(\boldsymbol{\theta}) = G_0(\boldsymbol{\theta})$ for $s = 1$. Starting from $s = 1$, we obtain, owing to the dimension formula [114], that

$$\dim(\tilde{G}_1(\boldsymbol{\theta})) = \dim(G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta})) = \dim(G_1(\boldsymbol{\theta})) + \dim(G_0(\boldsymbol{\theta})) - \dim(G_1(\boldsymbol{\theta}) + G_0(\boldsymbol{\theta})). \quad (4.73)$$

Since $G_1(\boldsymbol{\theta})$ and $G_0(\boldsymbol{\theta})$ are both spanned by a set of orthogonal basis vectors, we have $\dim(G_1(\boldsymbol{\theta})) = \dim(\boldsymbol{\theta}) - C = 2C + 2$ and $G_0(\boldsymbol{\theta}) = \dim(\boldsymbol{\theta})$. The dimension of the sum of the column spaces, $\dim(G_1(\boldsymbol{\theta}) + G_0(\boldsymbol{\theta}))$, is equal to the rank of the matrix $[\mathbf{e}_{C+1}, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})} | \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})}]$, which is $\dim(\boldsymbol{\theta})$. As a consequence, we have $\dim(\tilde{G}_1(\boldsymbol{\theta})) = 2C + 2$. The basis vectors that span $\tilde{G}_1(\boldsymbol{\theta})$ are just the column vectors that $G_1(\boldsymbol{\theta})$ and $G_0(\boldsymbol{\theta})$ have in common, namely,

$$\tilde{G}_1(\boldsymbol{\theta}) = G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta}) = \sum_{j \in \{C+1, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j. \quad (4.74)$$

Similarly for $s = 2, 3, \dots, 2C + 1$, we have

$$\begin{aligned} \dim(\tilde{G}_s(\boldsymbol{\theta})) &= \dim(G_s(\boldsymbol{\theta}) \cap \tilde{G}_{s-1}(\boldsymbol{\theta})) \\ &= (\dim(\boldsymbol{\theta}) - 1) + (\dim(\boldsymbol{\theta}) - C - (s - 2)) - \dim(\boldsymbol{\theta}) \\ &= \dim(\boldsymbol{\theta}) - C - (s - 1) \end{aligned} \quad (4.75)$$

and

$$\tilde{G}_s(\boldsymbol{\theta}) = G_s(\boldsymbol{\theta}) \cap \tilde{G}_{s-1}(\boldsymbol{\theta}) = \sum_{j \in \{C+s, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j. \quad (4.76)$$

Note that (4.75) is due to the fact that the dimension of $G_s(\boldsymbol{\theta})$, $s = 2, 3, \dots, 2C + 1$, is always equal to $\dim(\boldsymbol{\theta}) - 1$ and the dimension of the sum of column spaces $\dim(G_s(\boldsymbol{\theta}) + \tilde{G}_{s-1}(\boldsymbol{\theta}))$ is equal to the rank of the matrix

$$[\mathbf{e}_1, \dots, \mathbf{e}_{C+s-2}, \mathbf{e}_{C+s}, \mathbf{e}_{\dim(\boldsymbol{\theta})} | \mathbf{e}_{C+s-1}, \mathbf{e}_{C+s}, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})}], \quad (4.77)$$

which is always equal to $\dim(\boldsymbol{\theta})$. Hence, ultimately we have

$$\tilde{G}_{2C+1}(\boldsymbol{\theta}) = \sum_{j \in \{3C+1, 3C+2\}} c_j \mathbf{e}_j. \quad (4.78)$$

From the definition in (4.69), we know that $G_{2C+2}(\boldsymbol{\theta})$ and $\tilde{G}_{2C+1}(\boldsymbol{\theta})$ are orthogonal, hence

$$\bigcap_{s=1}^S G_s(\boldsymbol{\theta}^{(\eta)}) = G_S(\boldsymbol{\theta}) \bigcap \tilde{G}_{S-1}(\boldsymbol{\theta}) = \{\mathbf{0}\}, \quad (4.79)$$

which completes the proof of the “space filling” condition. \square

Theorem 4.3. *The proposed JMAP-ML algorithm in Algorithm 4.1 ensures that the sequence of complete-data log-likelihood values $\{\mathcal{L}_C(\boldsymbol{\theta}^{(\eta)}; \mathbf{y}, \mathbf{r})\}$, when bounded above, converges monotonically over iterations to some point L^* .*

Proof. In the first step of the JMAP-ML algorithm, we maximize $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ with respect to \mathbf{y} , given the prior parameter estimate $\boldsymbol{\theta}^{(\eta)}$. Since $\mathbf{y}^{(\eta+1)}$ is the global optimal solution, it is guaranteed that

$$\mathcal{L}_C(\boldsymbol{\theta}^{(\eta)}; \mathbf{y}^{(\eta+1)}, \mathbf{r}) \geq \mathcal{L}_C(\boldsymbol{\theta}^{(\eta)}; \mathbf{y}^{(\eta)}, \mathbf{r}) \quad (4.80)$$

holds for any $\mathbf{y}^{(\eta)}$ in its parameter space. In the second step, we maximize $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}^{(\eta+1)}, \mathbf{r})$ with respect to $\boldsymbol{\theta}$. Following a similar procedure for proving $Q(\boldsymbol{\theta}^{(\eta+1)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}^{(\eta)}; \boldsymbol{\theta}^{(\eta)})$ in Theorem 4.1, we can easily arrive at

$$\mathcal{L}_C(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{y}^{(\eta+1)}, \mathbf{r}) \geq \mathcal{L}_C(\boldsymbol{\theta}^{(\eta)}; \mathbf{y}^{(\eta+1)}, \mathbf{r}), \quad (4.81)$$

meaning that the value of $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ increases monotonically over iterations. When $\mathcal{L}(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ is bounded above, its convergence to some point \mathcal{L}^* of $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ is ensured. \square

4.3.2 Computational Complexity

Next, we evaluate the computational complexity of the proposed algorithms in terms of floating-point operations (FLOPs). We focus our attention on the joint estimation step (i.e., the second step) on a single iteration of Algorithm 4.1. It is shown in Appendix 4.8.3 that the complexity of the ECM algorithm scales as $\mathcal{O}(CNK)$ FLOPs per iteration. The computation of a JMAP-ML estimate is very similar to that of an ECM estimate. Hence, it is easy to verify that the computational complexity of the JMAP-ML algorithm is also of order $\mathcal{O}(CNK)$ FLOPs per iteration. In contrast, the computational complexity of the RIN algorithm scales as $\mathcal{O}((NK)^2)$ FLOPs per iteration due to the use of the nonparametric kernel density estimation to approximate the measurement error distribution.

4.4 An Alternative View of the EM Algorithm

In the previous sections, a Gaussian mixture distribution is adopted to represent the unknown measurement error. As is known from [38], any distribution can be approximated as closely as desired, for instance in \mathcal{L}_1 norm, by the Gaussian mixtures. Besides, the use of the Gaussian mixtures allows for closed form expression of $\boldsymbol{\theta}_e^{(\eta+1)}$ as was shown in Algorithm 4.1 and facilitates systematical analyses on the convergence properties and computational complexity as was performed in Section 4.3. In this section, we generalize the parametric model to be

$$p_V(v) \approx \hat{p}_V(v) = \sum_{l=1}^C \alpha_l \mathcal{K}_E^{(l)}(v; \boldsymbol{\beta}_l) \quad (4.82)$$

where $\mathcal{K}_E^{(l)}(v; \boldsymbol{\beta}_l)$ can be any distribution that belongs to an exponential family. Many representative distributions are member of the exponential family such as Gaussian, Rayleigh, exponential and Weibull distributions that are adequate to represent the NLOS measurement error characteristics in different localization environments. Note that the Gaussian mixture distribution is a member of (4.82). In the sequel, an alternative view of developing the expectation and maximization algorithm based on (4.82) is provided.

We start with reformulating the complete-data likelihood function as

$$p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}) = b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \exp [\boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) - a(\boldsymbol{\theta}_e)] \quad (4.83)$$

where $b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})$ is a scalar that may depend on \mathbf{y} and/or $\tilde{\mathbf{r}}(\boldsymbol{\theta}_p) \triangleq \mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p)$, $a(\boldsymbol{\theta}_e)$ is a scalar that may depend on $\boldsymbol{\theta}_e$, $\boldsymbol{\psi}(\boldsymbol{\theta}_e)$ is a vector that depends on $\boldsymbol{\theta}_e$ and $\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})$ is a vector that depends on \mathbf{y} and $\tilde{\mathbf{r}}(\boldsymbol{\theta}_p)$. For the special case that $\boldsymbol{\theta}_p$ is completely known, $\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})$ is the sufficient statistics for $\boldsymbol{\theta}_e$. In the expectation step, taking conditional expectation of the complete-data log-likelihood $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ in terms of \mathbf{y} given \mathbf{r} and $\boldsymbol{\theta}^{(\eta)}$ yields $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$. In the maximization step, a two-step procedure is adopted. In the first step, the mixture model parameters are updated, given the prior position estimate $\boldsymbol{\theta}_p^{(\eta)}$, through

$$\boldsymbol{\theta}_e^{(\eta+1)} = \Xi(\mathbf{T}(\boldsymbol{\theta}_p^{(\eta)})) \quad (4.84)$$

where

$$\mathbf{T}(\boldsymbol{\theta}_p^{(\eta)}) = \mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)})} [\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p^{(\eta)}), \mathbf{y})] \quad (4.85)$$

is the conditional expectation of the approximated sufficient statistic for $\boldsymbol{\theta}_e$ given $\boldsymbol{\theta}_p^{(\eta)}$ on the $(\eta+1)$ th EM iteration and $\Xi(\cdot)$ maps the obtained $\mathbf{T}(\boldsymbol{\theta}_p^{(\eta)})$ to the complete-data

maximum likelihood estimate of $\boldsymbol{\theta}_e$, namely $\boldsymbol{\theta}_e^{(\eta+1)}$. We assume that the mapping $\Xi(\cdot)$ can lead to closed form $\boldsymbol{\theta}_e^{(\eta+1)}$. In the second step, we insert $\boldsymbol{\theta}_e^{(\eta+1)}$ into $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$ and resolve $\boldsymbol{\theta}_p^{(\eta+1)}$ using the BFGS-QN method with the initial guess set by $\boldsymbol{\theta}_p^{(\eta)}$. These two steps are repeated as necessary. For clarity, the key steps are summarized in Algorithm 4.2. Two examples are given below for better understanding of Algorithm 4.2.

Algorithm 4.2 An Alternative View of the EM Algorithm

Step 1—Initialization:

Choose a convergence tolerance Δ and the maximum number of iterations N_{itr} . Set the iteration index $\eta = 0$. Choose an initial guess $\boldsymbol{\theta}^{(0)}$.

Step 2—Joint Estimation:

On the $(\eta + 1)$ th EM iteration ($\eta \in \mathbb{Z}, \eta \geq 0$),

- Reformulate the complete-data likelihood function as (4.83).
- Perform conditional expectation of the complete-data log-likelihood function $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ and obtain $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$.
- Perform conditional expectation of the approximated sufficient statistic for $\boldsymbol{\theta}_e$ according to (4.85) and map it to the complete-data ML estimator $\boldsymbol{\theta}_e^{(\eta+1)}$.
- Replace $\boldsymbol{\theta}_e$ with $\boldsymbol{\theta}_e^{(\eta+1)}$ in $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$ and solve for $\boldsymbol{\theta}_p^{(\eta+1)}$ numerically via the BFGS-QN method.

Step 3—Convergence Check:

If $\mathcal{L}_I(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}) - \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}) < \Delta$ or N_{itr} is reached, then terminate this algorithm and obtain $\hat{\boldsymbol{\theta}}^{\text{EM}} = \boldsymbol{\theta}^{(\eta+1)}$; otherwise set $\eta \leftarrow \eta + 1$ and return to **Step 2**.

Example-I: C-mode Gaussian mixture

In the first example, the C-mode Gaussian mixture is re-considered in the parametric model. The complete-data likelihood function can be expressed as

$$p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}) = b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \exp[\boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) - a(\boldsymbol{\theta}_e)] \quad (4.86)$$

where

$$\tilde{\mathbf{r}}(\boldsymbol{\theta}_p) = \mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p), \quad b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = \left(\frac{1}{2\pi}\right)^{\frac{NK}{2}}, \quad a(\boldsymbol{\theta}_e) = 0, \quad (4.87)$$

$$\boldsymbol{\psi}(\boldsymbol{\theta}_e) = [\boldsymbol{\psi}_1^T(\boldsymbol{\theta}_e), \boldsymbol{\psi}_2^T(\boldsymbol{\theta}_e), \dots, \boldsymbol{\psi}_C^T(\boldsymbol{\theta}_e)]^T, \quad (4.88)$$

$$\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = [\mathbf{t}_1^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}), \mathbf{t}_2^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}), \dots, \mathbf{t}_C^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})]^T, \quad (4.89)$$

and for $l = 1, 2, \dots, C$,

$$\boldsymbol{\psi}_l(\boldsymbol{\theta}_e) = \left[\ln\left(\frac{\alpha_l}{\sigma_l}\right) - \frac{\mu_l^2}{2\sigma_l^2}, \frac{\mu_l}{\sigma_l^2}, \frac{-1}{2\sigma_l^2} \right]^T, \quad (4.90)$$

$$\mathbf{t}_l(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = \begin{bmatrix} \sum_{i=1}^N \sum_{k=1}^K \delta(y_{i,k} - l) \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p)) \delta(y_{i,k} - l) \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p))^2 \delta(y_{i,k} - l) \end{bmatrix}. \quad (4.91)$$

This is due to the fact that

$$\begin{aligned} p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}) &= \prod_{i=1}^N \prod_{k=1}^K p(r_{i,k}, y_{i,k}; \boldsymbol{\theta}) \\ &= \prod_{i=1}^N \prod_{k=1}^K \left(\frac{\alpha_{y_{i,k}}}{\sqrt{2\pi}\sigma_{y_{i,k}}} \exp \left[\frac{-(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_{y_{i,k}})^2}{2\sigma_{y_{i,k}}^2} \right] \right) \\ &= \left(\frac{1}{2\pi} \right)^{\frac{NK}{2}} \exp \left[\sum_{i=1}^N \sum_{k=1}^K \ln \left(\frac{\alpha_{y_{i,k}}}{\sigma_{y_{i,k}}} \right) - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_{y_{i,k}})^2}{2\sigma_{y_{i,k}}^2} \right] \end{aligned} \quad (4.92)$$

where the summation inside the exponential function can be expressed in a compact form through the following steps:

$$\begin{aligned} &\sum_{i=1}^N \sum_{k=1}^K \ln \left(\frac{\alpha_{y_{i,k}}}{\sigma_{y_{i,k}}} \right) - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_{y_{i,k}})^2}{2\sigma_{y_{i,k}}^2} \\ &= \sum_{l=1}^C \sum_{i=1}^N \sum_{k=1}^K \left(\ln \left(\frac{\alpha_l}{\sigma_l} \right) - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l)^2}{2\sigma_l^2} \right) \delta(y_{i,k} - l) \\ &= \sum_{l=1}^C \sum_{i=1}^N \sum_{k=1}^K \left\{ \left(\ln \left(\frac{\alpha_l}{\sigma_l} \right) - \frac{\mu_l^2}{2\sigma_l^2} \right) + \frac{\mu_l(r_{i,k} - d_i(\boldsymbol{\theta}_p))}{\sigma_l^2} - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p))^2}{2\sigma_l^2} \right\} \delta(y_{i,k} - l) \\ &= \sum_{l=1}^C \boldsymbol{\psi}_l^T(\boldsymbol{\theta}_e) \mathbf{t}_l(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \\ &= \boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}). \end{aligned} \quad (4.93)$$

The Q -function in this example is computed as

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(n)}) = -\frac{NK}{2} \ln(2\pi) + \boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(n)})} \{ \mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \}. \quad (4.94)$$

For ease of the subsequent derivations, the elements in the vectorized signal model (4.3) are re-labeled such that r_m and y_m are the m th element of \mathbf{r} and \mathbf{y} , respectively. The computation of $\mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(n)})} \{ \mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \}$ is decomposed into three parts. We start

with

$$\begin{aligned}
& \mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \left\{ \sum_{i=1}^N \sum_{k=1}^K \delta(y_{i,k} - l) \right\} \\
&= \sum_{\mathbf{y}} \left[\sum_{m=1}^{NK} \delta(y_m - l) \times \prod_{j=1}^{NK} \Pr\{y_j|r_j; \boldsymbol{\theta}^{(\eta)}\} \right] \\
&= \sum_{m=1}^{NK} \left[\sum_{y_1=1}^C \cdots \sum_{y_{NK}=1}^C \left(\delta(y_m - l) \times \prod_{j=1}^{NK} \Pr\{y_j|r_j; \boldsymbol{\theta}^{(\eta)}\} \right) \right] \\
&= \sum_{m=1}^{NK} \Pr\{y_m = l|r_m; \boldsymbol{\theta}^{(\eta)}\} \times \left[\sum_{y_1=1}^C \cdots \sum_{y_{m-1}=1}^C \sum_{y_{m+1}=1}^C \cdots \sum_{y_{NK}=1}^C \prod_{\substack{j=1, \\ j \neq m}}^{NK} \Pr\{y_j|r_j; \boldsymbol{\theta}^{(\eta)}\} \right] \\
&= \sum_{m=1}^{NK} \Pr\{y_m = l|r_m; \boldsymbol{\theta}^{(\eta)}\} \times \left[\prod_{\substack{j=1, \\ j \neq m}}^{NK} \underbrace{\left(\sum_{y_j=1}^C \Pr\{y_j|r_j; \boldsymbol{\theta}^{(\eta)}\} \right)}_1 \right] \\
&= \sum_{i=1}^N \sum_{k=1}^K P_{i,k,l}^{(\eta)}. \tag{4.95}
\end{aligned}$$

As it was introduced beforehand $P_{i,k,l}^{(\eta)}$ is a short-hand notation of $\Pr\{y_{i,k} = l|r_{i,k}; \boldsymbol{\theta}^{(\eta)}\}$, which can be computed by means of Bayes' rule as follows:

$$P_{i,k,l}^{(\eta)} = \Pr\{y_{i,k} = l|r_{i,k}; \boldsymbol{\theta}^{(\eta)}\} = \frac{\alpha_l^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_l^{(\eta)}, \sigma_l^{2,(\eta)})}{\sum_{l=1}^C \alpha_l^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_l^{(\eta)}, \sigma_l^{2,(\eta)})}. \tag{4.96}$$

Similarly, we obtain

$$\mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \left\{ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p)) \delta(y_{i,k} - l) \right\} = \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p)) P_{i,k,l}^{(\eta)} \tag{4.97}$$

and

$$\mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \left\{ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p))^2 \delta(y_{i,k} - l) \right\} = \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p))^2 P_{i,k,l}^{(\eta)}. \tag{4.98}$$

Having the explicit expressions of $\mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \{\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})\}$, it is easy to verify that

$$\begin{aligned}
Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) &= -\frac{NK}{2} \ln(2\pi) + \boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \{\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})\} \\
&= \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^C \ln(\alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) P_{i,k,l}^{(\eta)}. \tag{4.99}
\end{aligned}$$

So far, another way of deriving $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$ under the Gaussian mixture distribution has been provided, which leads to the same result as that obtained in the ECM algorithm implemented in Section 4.2.3.

Next, $\mathbf{T}(\boldsymbol{\theta}_p^{(\eta)})$ is mapped to $\boldsymbol{\theta}_e^{(\eta+1)}$ in the first step of the maximization stage. In this example,

$$\mathbf{T}(\boldsymbol{\theta}_p^{(\eta)}) = [\mathbf{T}_1(\boldsymbol{\theta}_p^{(\eta)}), \mathbf{T}_2(\boldsymbol{\theta}_p^{(\eta)}), \dots, \mathbf{T}_C(\boldsymbol{\theta}_p^{(\eta)})]^T \quad (4.100)$$

where

$$\mathbf{T}_l(\boldsymbol{\theta}_p^{(\eta)}) = \begin{bmatrix} T_{l,1}(\boldsymbol{\theta}_p^{(\eta)}) \\ T_{l,2}(\boldsymbol{\theta}_p^{(\eta)}) \\ T_{l,3}(\boldsymbol{\theta}_p^{(\eta)}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N \sum_{k=1}^K P_{i,k,l}^{(\eta)} \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)})) P_{i,k,l}^{(\eta)} \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}))^2 P_{i,k,l}^{(\eta)} \end{bmatrix}, \quad l = 1, 2, \dots, C. \quad (4.101)$$

It is known from [115] that the mapping $\Xi(\cdot)$ for this case is

$$\alpha_l^{(\eta+1)} = \frac{T_{l,1}(\boldsymbol{\theta}_p^{(\eta)})}{\sum_{l=1}^C T_{l,1}(\boldsymbol{\theta}_p^{(\eta)})} = \frac{T_{l,1}(\boldsymbol{\theta}_p^{(\eta)})}{NK}, \quad \mu_l^{(\eta+1)} = \frac{T_{l,2}(\boldsymbol{\theta}_p^{(\eta)})}{T_{l,1}(\boldsymbol{\theta}_p^{(\eta)})}, \quad \sigma_l^{2,(\eta+1)} = \frac{T_{l,3}(\boldsymbol{\theta}_p^{(\eta)})}{T_{l,1}(\boldsymbol{\theta}_p^{(\eta)})} - \mu_l. \quad (4.102)$$

After replacing the unknown μ_l with its latest estimate $\mu_l^{(\eta+1)}$ in the expression of $\sigma_l^{2,(\eta+1)}$, the results in (4.102) coincide with those given in (4.58), (4.59) and (4.60).

In the second step of the maximization stage, $\boldsymbol{\theta}_e$ is replaced with $\boldsymbol{\theta}_e^{(\eta+1)}$ in (4.99) before solving for $\boldsymbol{\theta}_p^{(\eta+1)}$ numerically from

$$\arg \min_{\boldsymbol{\theta}_p} \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^C \frac{\left(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)} \right)^2 P_{i,k,l}^{(\eta)}}{\sigma_l^{2,(\eta+1)}}. \quad (4.103)$$

Example-II: two-mode Gaussian-exponential mixture

In the second example, a two-mode Gaussian-exponential mixture is considered. Herein, the exponential distribution is written as

$$\mathcal{E}(v; \lambda_{\mathcal{E}}) = \begin{cases} \frac{1}{\lambda_{\mathcal{E}}} \exp\left(\frac{-v}{\lambda_{\mathcal{E}}}\right), & v \geq 0 \\ 0, & v < 0 \end{cases}. \quad (4.104)$$

Following a similar methodology as is shown in the first example, we start with a reformulation of the complete-data likelihood function as

$$p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}) = b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \exp[\boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) - a(\boldsymbol{\theta}_e)] \quad (4.105)$$

where

$$\tilde{\mathbf{r}}(\boldsymbol{\theta}_p) = \mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p), \quad b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = 1, \quad a(\boldsymbol{\theta}_e) = 0, \quad (4.106)$$

and

$$\boldsymbol{\psi}(\boldsymbol{\theta}_e) = [\boldsymbol{\psi}_1^T(\boldsymbol{\theta}_e), \boldsymbol{\psi}_2^T(\boldsymbol{\theta}_e)]^T, \quad (4.107)$$

with

$$\boldsymbol{\psi}_1(\boldsymbol{\theta}_e) = \left[\ln \left(\frac{\alpha_1}{\sqrt{2\pi}\sigma_1} \right) - \frac{\mu_1^2}{2\sigma_1^2}, \frac{\mu_1}{\sigma_1^2}, \frac{-1}{2\sigma_1^2} \right]^T, \quad (4.108)$$

$$\boldsymbol{\psi}_2(\boldsymbol{\theta}_e) = \left[\ln \left(\frac{\alpha_2}{\lambda_{\mathcal{E}}} \right), -\frac{1}{\lambda_{\mathcal{E}}} \right]^T, \quad (4.109)$$

and

$$\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = [\mathbf{t}_1^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}), \mathbf{t}_2^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})]^T, \quad (4.110)$$

with

$$\mathbf{t}_1(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = \begin{bmatrix} \sum_{i=1}^N \sum_{k=1}^K \delta(y_{i,k} - 1) \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p)) \delta(y_{i,k} - 1) \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p))^2 \delta(y_{i,k} - 1) \end{bmatrix}, \quad (4.111)$$

$$\mathbf{t}_2(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = \begin{bmatrix} \sum_{i=1}^N \sum_{k=1}^K \delta(y_{i,k} - 2) \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p)) \delta(y_{i,k} - 2) \end{bmatrix}. \quad (4.112)$$

Performing conditional expectation of the complete-data log-likelihood function in the next step yields

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) &= \boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)})} [\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})] \\ &= \sum_{i=1}^N \sum_{k=1}^K \ln(\alpha_1 \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_1, \sigma_1^2)) P_{i,k,1}^{(\eta)} + \ln(\alpha_2 \mathcal{E}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \lambda_{\mathcal{E}})) P_{i,k,2}^{(\eta)} \end{aligned} \quad (4.113)$$

where

$$P_{i,k,1}^{(\eta)} = \frac{\alpha_1^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_1^{(\eta)}, \sigma_1^{2,(\eta)})}{\alpha_1^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_1^{(\eta)}, \sigma_1^{2,(\eta)}) + \alpha_2^{(\eta)} \mathcal{E}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \lambda_{\mathcal{E}}^{(\eta)})} \quad (4.114)$$

and

$$P_{i,k,2}^{(\eta)} = \frac{\alpha_2^{(\eta)} \mathcal{E}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \lambda_{\mathcal{E}}^{(\eta)})}{\alpha_1^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_1^{(\eta)}, \sigma_1^{2,(\eta)}) + \alpha_2^{(\eta)} \mathcal{E}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \lambda_{\mathcal{E}}^{(\eta)})}. \quad (4.115)$$

In this example, we have

$$\mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \{ \mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p^{(\eta+1)}), \mathbf{y}) \} = \mathbf{T}(\boldsymbol{\theta}_p^{(\eta)}) = [\mathbf{T}_1(\boldsymbol{\theta}_p^{(\eta)}), \mathbf{T}_2(\boldsymbol{\theta}_p^{(\eta)})]^T \quad (4.116)$$

where

$$\mathbf{T}_1(\boldsymbol{\theta}_p^{(\eta)}) = \begin{bmatrix} T_{1,1}(\boldsymbol{\theta}_p^{(\eta)}) \\ T_{1,2}(\boldsymbol{\theta}_p^{(\eta)}) \\ T_{1,3}(\boldsymbol{\theta}_p^{(\eta)}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N \sum_{k=1}^K P_{i,k,1}^{(\eta)} \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)})) P_{i,k,1}^{(\eta)} \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}))^2 P_{i,k,1}^{(\eta)} \end{bmatrix} \quad (4.117)$$

and

$$\mathbf{T}_2(\boldsymbol{\theta}_p^{(\eta)}) = \begin{bmatrix} T_{2,1}(\boldsymbol{\theta}_p^{(\eta)}) \\ T_{2,2}(\boldsymbol{\theta}_p^{(\eta)}) \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^N \sum_{k=1}^K P_{i,k,2}^{(\eta)} \\ \sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)})) P_{i,k,2}^{(\eta)} \end{bmatrix}. \quad (4.118)$$

It is known from [116] that the mapping $\Xi(\cdot)$ used for this example is

$$\begin{aligned} \alpha_1^{(\eta+1)} &= \frac{T_{1,1}(\boldsymbol{\theta}_p^{(\eta)})}{T_{1,1}(\boldsymbol{\theta}_p^{(\eta)}) + T_{2,1}(\boldsymbol{\theta}_p^{(\eta)})}, & \mu_1^{(\eta+1)} &= \frac{T_{1,2}(\boldsymbol{\theta}_p^{(\eta)})}{T_{1,1}(\boldsymbol{\theta}_p^{(\eta)})}, & \sigma_1^{2,(\eta+1)} &= \frac{T_{1,3}(\boldsymbol{\theta}_p^{(\eta)})}{T_{1,1}(\boldsymbol{\theta}_p^{(\eta)})} - \mu_1, \\ \alpha_2^{(\eta+1)} &= \frac{T_{2,1}(\boldsymbol{\theta}_p^{(\eta)})}{T_{1,1}(\boldsymbol{\theta}_p^{(\eta)}) + T_{2,1}(\boldsymbol{\theta}_p^{(\eta)})}, & \lambda_{\mathcal{E}}^{(\eta+1)} &= \frac{T_{2,2}(\boldsymbol{\theta}_p^{(\eta)})}{T_{2,1}(\boldsymbol{\theta}_p^{(\eta)})}. \end{aligned} \quad (4.119)$$

Again, the unknown μ_1 needs to be replaced with its latest estimate $\mu_1^{(\eta+1)}$ in the expression of $\sigma_1^{2,(\eta+1)}$.

In the second step of the maximization step, $\boldsymbol{\theta}_e$ is replaced with $\boldsymbol{\theta}_e^{(\eta+1)}$ in (4.113) and solve for $\boldsymbol{\theta}_p^{(\eta+1)}$ from

$$\arg \min_{\boldsymbol{\theta}_p} \sum_{i=1}^N \sum_{k=1}^K \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_1^{(\eta+1)})^2 P_{i,k,1}^{(\eta)}}{2\sigma_1^{2,(\eta+1)}} + \sum_{\substack{i=1 \\ r_{i,k} \geq d_i(\boldsymbol{\theta}_p)}}^N \sum_{k=1}^K \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p)) P_{i,k,2}^{(\eta)}}{\lambda_{\mathcal{E}}^{(\eta+1)}}. \quad (4.120)$$

Note that the conditional summations in (4.120) stems from the definition of $\mathcal{E}(v; \lambda_{\mathcal{E}})$, where the pdf is equal to zero when $v < 0$. It is easy to verify that the results obtained in this example will coincide with that found by the regular ECM algorithm when the partition of $\boldsymbol{\theta}$ is chosen to be $\boldsymbol{\vartheta}_1 = [\alpha_1, \alpha_2]^T$, $\boldsymbol{\vartheta}_2 = \mu_1$, $\boldsymbol{\vartheta}_3 = \sigma_1^2$, $\boldsymbol{\vartheta}_4 = \lambda_{\mathcal{E}}$, and $\boldsymbol{\vartheta}_5 = \boldsymbol{\theta}_p$.

4.5 Cramér-Rao Lower Bound Computation

In this section, a numerical method is proposed to compute the Cramér-Rao lower bound (CRLB) for our joint estimation problem. The vector parameter to be estimated

is $\boldsymbol{\theta}$. Let $\hat{\boldsymbol{\theta}}$ be an unbiased estimator of $\boldsymbol{\theta}$ and let $\text{Cov}(\hat{\boldsymbol{\theta}})$ denote the covariance matrix of $\hat{\boldsymbol{\theta}}$. According to Theorem 2.1 given in Chapter 2, we have

$$\text{Cov}(\hat{\boldsymbol{\theta}}) = \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ (\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^T \right\} \succeq \mathcal{F}^{-1}(\boldsymbol{\theta}) \quad (4.121)$$

where $\mathcal{F}(\boldsymbol{\theta})$ denotes the Fisher's information matrix (FIM). Often, it is more convenient to express the FIM as follows:

$$\begin{aligned} \mathcal{F}(\boldsymbol{\theta}) &= \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{r}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}^T \ln p(\mathbf{r}; \boldsymbol{\theta}) \right\} \\ &= \int \left\{ \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{r}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}^T \ln p(\mathbf{r}; \boldsymbol{\theta}) \right\} p(\mathbf{r}; \boldsymbol{\theta}) d\mathbf{r}. \end{aligned} \quad (4.122)$$

Based on (4.2), the integration in (4.122) is hard to evaluate analytically. In order to give a numerical solution, we perform Monte Carlo integration [108] as follows:

$$\mathcal{F}(\boldsymbol{\theta}) \approx \frac{1}{N_M} \sum_{n=1}^{N_M} \nabla_{\boldsymbol{\theta}} \ln p(\mathbf{r}^{(n)}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}^T \ln p(\mathbf{r}^{(n)}; \boldsymbol{\theta}) \quad (4.123)$$

where $\mathbf{r}^{(n)}$, $n = 1, 2, \dots, N_M$ are sample vectors generated independently from $p(\mathbf{r}; \boldsymbol{\theta})$. The expression of $\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{r}; \boldsymbol{\theta})$ is given in Appendix 4.8.4.

Serving as a metric of localization accuracy in practice, the localization root-mean-square-error (RMSE)

$$\text{RMSE}(\hat{\boldsymbol{\theta}}_p) = \sqrt{\mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \{ (\hat{x} - x)^2 + (\hat{y} - y)^2 \}} \quad (4.124)$$

is chosen to evaluate different position estimators. This metric relates to the obtained CRLB according to

$$\text{RMSE}(\hat{\boldsymbol{\theta}}_p) \geq \sqrt{[\mathcal{F}^{-1}(\boldsymbol{\theta})]_{q-1,q-1} + [\mathcal{F}^{-1}(\boldsymbol{\theta})]_{q,q}} = \text{CRLB}_{\text{pos}}(\boldsymbol{\theta}_p) \quad (4.125)$$

where $q = \dim(\boldsymbol{\theta})$ is the dimension of $\boldsymbol{\theta}$, and CRLB_{pos} interprets the best achievable localization accuracy of any unbiased position estimator.

4.6 Simulations

In this section, the performance of the proposed ECM- and JMAP-ML algorithms will be evaluated and further compared with several competing algorithms, including the RIN algorithm introduced in Chapter 3, in comprehensive simulations. Section 4.6.1 introduces the overall simulation setup, and Section 4.6.2 shows some simulation results.

4.6.1 Simulation Setup

In the following simulations, a stationary MS is to be located in cellular radio networks with different BSs-MS geometries. To be realistic as much as possible, the BS positions are taken from an operating network in a German city [29]. The measurement error is assumed to be mode-dependent in mixed LOS/NLOS localization environments, more precisely,

$$v \sim \begin{cases} p_V^{(\text{L})}(v; \boldsymbol{\beta}_{\text{L}}), & \text{LOS condition} \\ p_V^{(\text{NL})}(v; \boldsymbol{\beta}_{\text{NL}}), & \text{NLOS condition} \end{cases} \quad (4.126)$$

We perform localization in three different scenarios:

- **Scenario I:** The measurement error terms, $v_{i,k}$, $i = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$, are iid and follow a two-mode Gaussian mixture distribution. Prior knowledge about the family of each mixture component is assumed.
- **Scenario II:** The measurement error terms $v_{i,k}$, $i = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$ are iid and follow a two-mode Gaussian-exponential mixture distribution. However, no prior knowledge about the measurement error distribution is assumed.
- **Scenario III:** The iid assumption is violated due to a temporal pattern in the observed data. Specifically, we assume that the channel state of a BS-MS pair stays unaltered (either in LOS or in NLOS) during the measurement interrogation (quasi-stationary for at least K samples). Using the parametric approach to the modeling of $p_V(v)$ in (4.2) is sub-optimal.

4.6.2 Simulation Results

Simulation results will be given in Section 4.6.2.1 regarding the first localization scenario, Section 4.6.2.2 regarding the second localization scenario, and Section 4.6.2.3 regarding the third localization scenario.

4.6.2.1 Results for Scenario I

For ease of comparisons, we reuse the BSs geometry illustrated in Fig. 1 of Chapter 3. The measurement error distribution that generates the simulation data is assumed to be

$$p_V(v) = \alpha_{\text{L}} \mathcal{N}(v; \mu_{\text{L}}, \sigma_{\text{L}}^2) + \alpha_{\text{NL}} \mathcal{N}(v; \mu_{\text{NL}}, \sigma_{\text{NL}}^2) \quad (4.127)$$

where $\mu_L = 0$ m, $\sigma_L = 55$ m, $\mu_{NL} = 380$ m and $\sigma_{NL} = 120$ m. These parameters are set according to the measurement campaign results published in [29] except that μ_L is re-set from 51 m to 0 m so as to comply with the common assumption that the measurement error under the LOS condition is generally unbiased. As we assume the prior knowledge about the families of $p_V^{(L)}(v; \boldsymbol{\beta}_L)$ and $p_V^{(NL)}(v; \boldsymbol{\beta}_{NL})$, the measurement error distribution $p_V(v)$ can be precisely represented by

$$p_V(v) = \alpha_1 \mathcal{N}(v; \mu_1, \sigma_1^2) + \alpha_2 \mathcal{N}(v; \mu_2, \sigma_2^2) \quad (4.128)$$

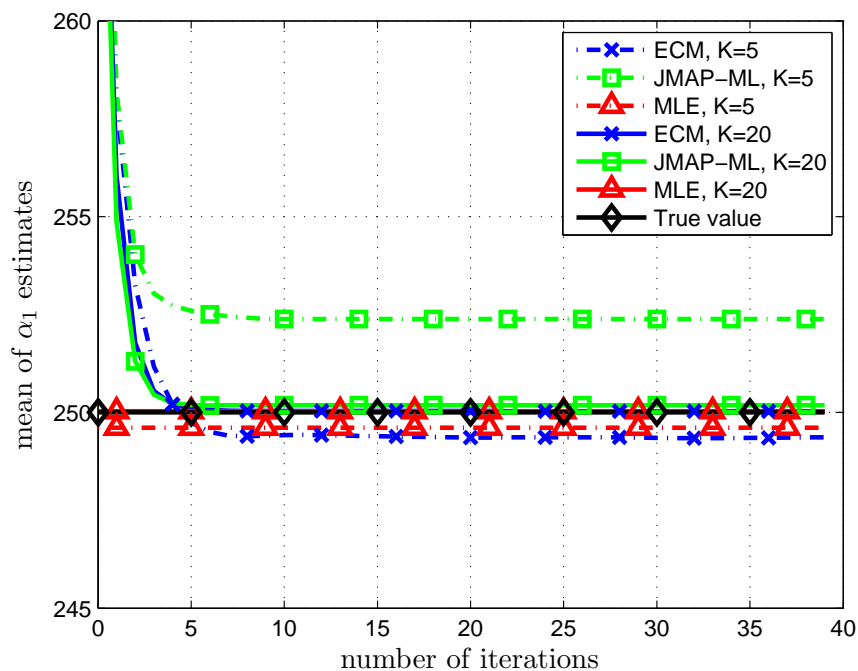
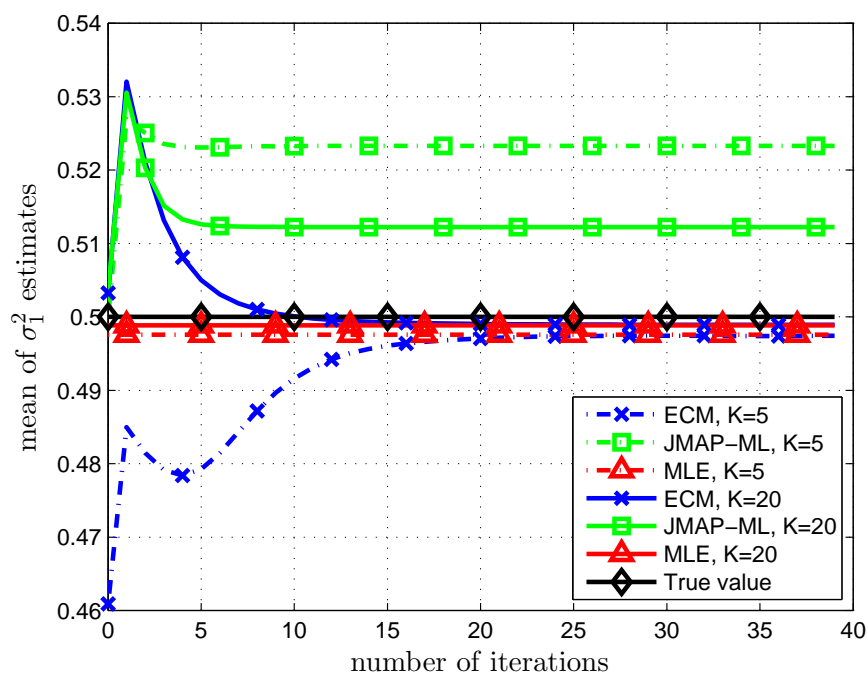
in the parametric model, where $\{\alpha_1, \alpha_2, \mu_1, \sigma_1^2, \mu_2, \sigma_2^2\}$ is the set of unknown deterministic parameters with the true values $\{\alpha_L, \alpha_{NL}, \mu_L, \sigma_L^2, \mu_{NL}, \sigma_{NL}^2\}$.

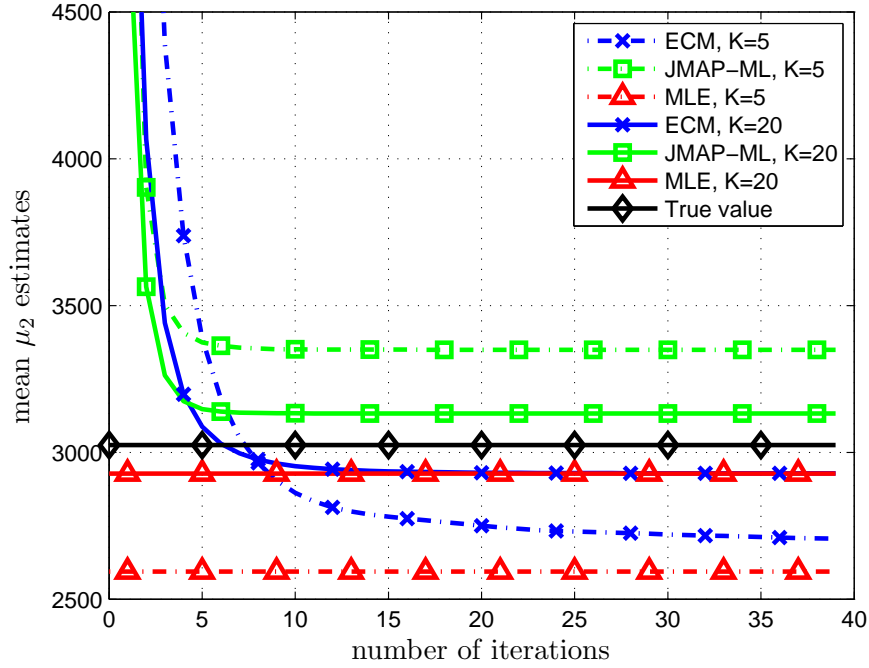
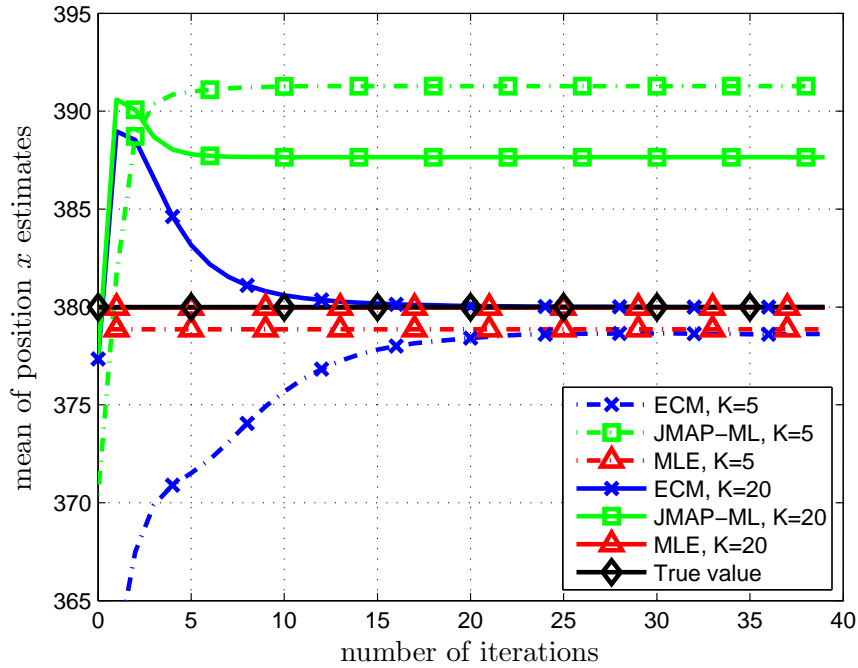
The proposed ECM- and JMAP-ML estimators are compared to:

- The Maximum likelihood estimator, cf. (4.8), which is solved numerically via the MATLABTM function FMINCON, configured with the “interior-point” algorithm.
- The robust iterative nonparametric (RIN) position estimator, cf. Chapter 3.

Note that we assume the same convergence conditions for the ECM- and JMAP-ML algorithms, namely they will be terminated when the convergence tolerance $\Delta = 10^{-4}$ or the maximum number of iterations, $N_{itr} = 40$, trials has been reached. In both algorithms, an initial guess of the MS position is set by the first two entries of the least-squares solution (3.11) derived in Chapter 3, and an initial guess of the Gaussian mixture model parameters is computed according to Algorithm 4.3 given in Appendix 4.8.5. The RIN algorithm follows the same settings as given in Chapter 3.

In the first experiment, we numerically study the parameter estimation bias and some convergence aspects of the proposed algorithms. We assume that the MS is located at $[x = 0.25 \text{ m}, y = 0.25 \text{ km}]$ and the prior probability α_{NL} is set to 0.5. Two different values of K are used, namely, $K = 5$, modeling a small sample size, and $K = 20$, modeling a large sample size.

Figure 4.1. Mean of α_1 estimates versus the number of iterations.Figure 4.2. Mean of σ_1^2 estimates versus the number of iterations.

Figure 4.3. Mean of μ_2 estimates versus the number of iterations.Figure 4.4. Mean of position x estimates versus the number of iterations.

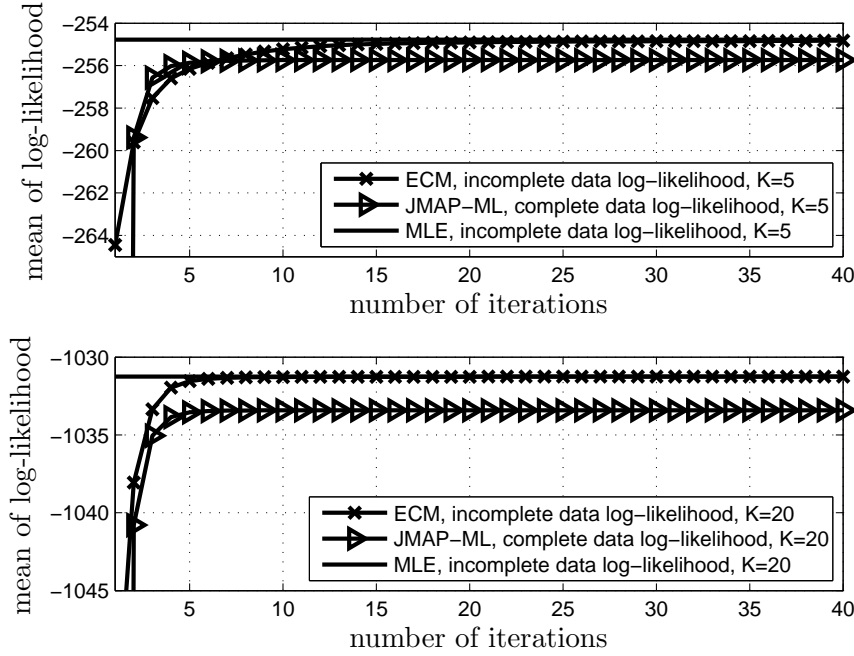


Figure 4.5. Convergence of the mean of the log-likelihood values versus the number of iterations for both the ECM algorithm and JMAP-ML algorithm. Top: a small sample size, $K = 5$. Bottom: a large sample size, $K = 20$.

We perform a Monte Carlo simulation with 2500 independent trials. For each trial, we record the ultimate parameter estimates and log-likelihoods for all iterations of the ECM- and JMAP-ML algorithms. It is necessary to stress that we consider the incomplete-data log-likelihood for the ECM algorithm whereas the complete-data log-likelihood for the JMAP-ML algorithm. We compute the mean of each obtained set of parameter estimates and log-likelihood values, respectively. We show the mean parameter estimates of α_1 , σ_1^2 , μ_2 and x in Fig. 4.1 through Fig. 4.4. For better comparisons, the maximum likelihood estimates as well as the true values of the desired parameters are also depicted in the figures. The mean of each set of log-likelihood values is shown versus the number of iterations in Fig. 4.5.

Next, we summarize some important observations and give the relevant explanations as follows:

- Given the same initial guess, the ECM estimator performs closest to the ML estimator when the number of measurements NK is large. This can be clearly seen from the overlap of the ECM- and MLE curves depicted for the case $K = 20$ in Fig. 4.1 through Fig. 4.4. However, for the case $K = 5$, we can observe an obvious gap between the two curves in Fig. 4.2, Fig. 4.3, and Fig. 4.4. The reason

is that the ECM algorithm only achieves local maximum or saddle point in some Monte Carlo trials.

- The JMAP-ML algorithm generates biased estimates independent of NK . Nevertheless, the JMAP-ML estimator still serves as a good approximation of the ML Estimator.
- Given the same initial guess, the JMAP-ML algorithm outperforms the ECM algorithm in terms of convergence speed.
- Figure 4.5 confirms that the log-likelihood values increase monotonically over iterations.

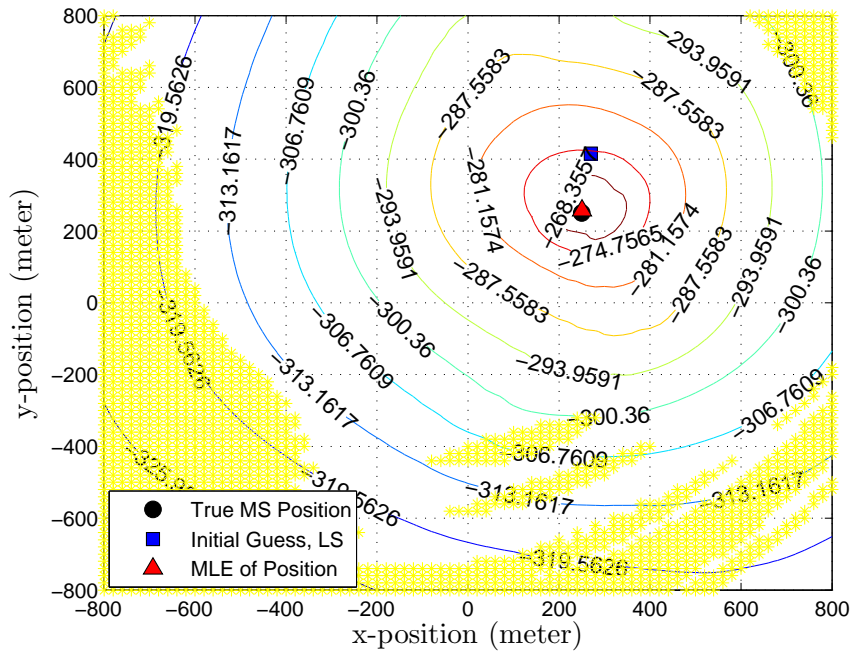


Figure 4.6. Convergence region of the ECM algorithm in one particular Monte Carlo trial with $K = 5$ samples and $\alpha_L = 0.5$. Herein, ‘●(black)’ denotes the true MS position $\theta_p = [x = 0.25 \text{ km}, y = 0.25 \text{ km}]$, ‘■ (blue)’ denotes the least-squares estimate of the actual MS position, which serves as an initial guess of the proposed algorithms and ‘▲ (red)’ denotes the ML estimate of the actual MS position.

As it was mentioned in Chapter 2, the performance of an EM-type algorithm highly depends on the quality of an initial guess. Our strategy is to initialize first the MS position by running a simple localization algorithm and then the remaining Gaussian mixture model parameters according to Algorithm 4.3 given in Appendix 4.8.5. As our initialization strategy is essentially a two-step procedure, a bad initial MS estimate

will ruin the whole procedure. It is thus interesting to investigate how the initial MS position estimate will influence the ultimate performance of the ECM algorithm in the sense of achieving the global maximum. To this end, we define a two-dimensional (2-D) square area with x and y both ranging from -800 meter to 800 meter at a grid size 20 meter. For each grid point $\mathbf{p}_g^{(0)} = [x_g, y_g]$ in the defined area, we apply then Algorithm 4.3 to compute an estimate of the Gaussian mixture parameters and finally obtain $\theta^{(0)}$ for initializing the ECM algorithm. We assign one to that grid point if the ECM algorithm ultimately converges to the global maximum of the log-likelihood function $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ or zero otherwise. Figure 4.6 depicts the convergence region for one particular Monte Carlo trial with $K = 5$ and $\alpha_L = 0.5$. In Fig. 4.6, the region painted in yellow includes all the grid points assigned with zero (an indicator of local maximum or saddle point), whereas the region painted in white includes those assigned with one (an indicator of global maximum). To shed some light on the quality of the initial guess, a contour plot of the log-likelihood value at each grid point $\theta^{(0)}$ is also superimposed in Fig. 4.6. Besides, we also depict the true position as well as the least-squares estimator and the ML estimator of it obtained for this particular trial. As per our observations, the least-squares estimate serves as an adequate initial guess of the MS position of the ECM algorithm for the first scenario.

In the second experiment, we shall investigate the ECM- and JMAP-ML algorithms in terms of localization accuracy and compare their performance mainly with that of the RIN algorithm. Localization RMSE of different position estimators will be demonstrated in two examples.

The first example of this experiment assumes that 1500 pairs of different MS positions are uniformly generated from the city center area as shown in Fig. 1 of Chapter 3. The number of distance measurements collected at each BS is $K = 10$ samples. We evaluate the mean localization RMSE of all candidate position estimators as a function of the NLOS contamination ratio α_{NLOS} . The results are shown along with the best achievable localization accuracy CRLB_{pos} in Fig. 4.7. In the second example of this experiment, we investigate how the number of measurements NK will influence the localization accuracy of different position estimators. We fix the NLOS contamination ratio, α_{NLOS} , to 0.5 but vary K from 5 to 45 with an increment of 4 samples. The mean localization RMSE is evaluated versus NK and the results are shown in Fig. 4.8.

From the illustrations, we summarize the performance of different position estimators as follows:

- The ECM position estimator is closest to the ML estimator in terms of the localization RMSE. The JMAP-ML estimator shows inferior localization RMSE as

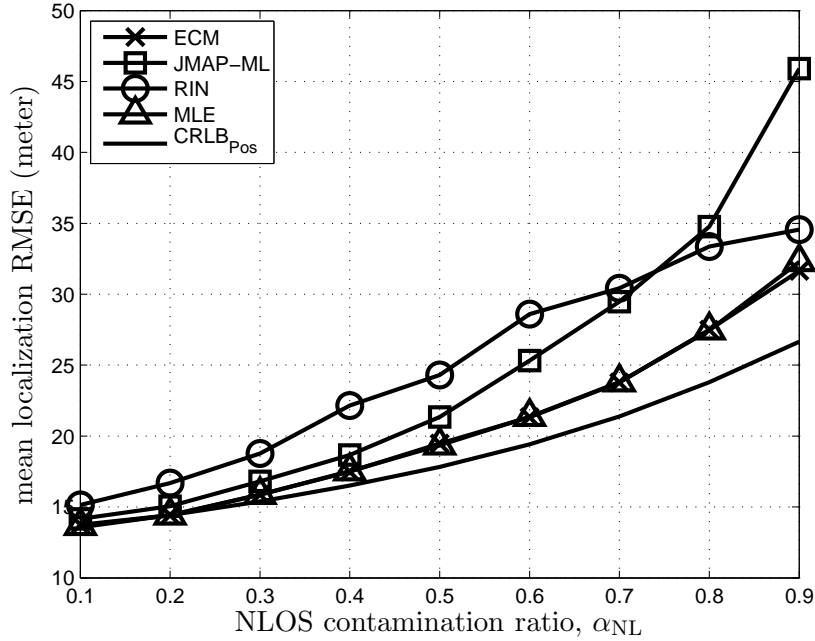


Figure 4.7. Mean localization RMSE (over 1500 sets of uniformly generated MS position) of different position estimators versus the NLOS contamination ratio with $K = 10$ at each BS.

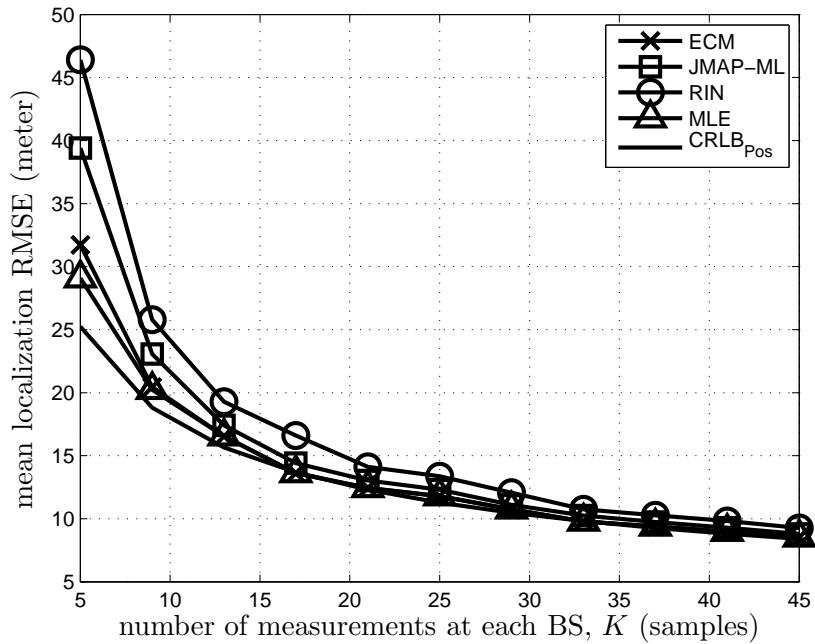


Figure 4.8. Mean localization RMSE (over 1500 sets of uniformly generated MS position) of different position estimators versus the number of measurements K at each BS with $\alpha_L = 0.5$.

compared to that of the ECM estimator. This confirms the statistical properties of the ECM- and JMAP-ML algorithms; that is the ECM algorithm is able to reproduce the ML estimator that globally maximizes the incomplete-data log-likelihood function in (4.7) when the iid measurement error assumption holds and $p_V(v)$ can be precisely represented by our parametric model [93]. However, the JMAP-ML algorithm merely produces a biased and inconsistent estimator according to Chapter 2. But still, it was shown to be a good approximation of the ML estimator in the above examples. The overgrown bias is responsible for the drastic increase of the localization RMSE, beyond $\alpha_{\text{NL}} = 0.6$, in Fig. 4.7.

- In most cases, both the ECM algorithm and JMAP-ML algorithm outperform the RIN algorithm. The key reason is that the proposed algorithms have more information about the measurement error (prior knowledge about the families of $p_V^{(\text{L})}(v; \boldsymbol{\beta}_{\text{L}})$ and $p_V^{(\text{NL})}(v; \boldsymbol{\beta}_{\text{NL}})$) than the RIN algorithm.

4.6.2.2 Results for Scenario II

In the second scenario, we follow the same BS geometry used in the previous subsection. The MS locates at $[x = 0.25 \text{ km}, y = 0.25 \text{ km}]$. The measurement error distribution that generates the simulation data is a two-mode Gaussian-exponential mixture

$$p_V(v) = \alpha_{\text{L}} \mathcal{N}(v; \mu_{\text{L}}, \sigma_{\text{L}}^2) + \alpha_{\text{NL}} \mathcal{E}(v; \lambda_{\text{NL}}) \quad (4.129)$$

where $\mu_{\text{L}} = 0 \text{ m}$, $\sigma_{\text{L}} = 55 \text{ m}$, $\lambda_{\text{NL}} = 80 \text{ m}$. The true value of λ_{NL} is set empirically to 20 m. In contrast to the assumption made for the first scenario, no prior knowledge about the measurement error distribution is assumed here. As a consequence, the parametric modeling of $p_V(v)$ as Gaussian mixtures causes approximation error, namely,

$$p_V(v) \approx \hat{p}_V(v) = \sum_{l=1}^C \alpha_l \mathcal{N}(v; \mu_l, \sigma_l^2). \quad (4.130)$$

In the previous subsection, we have seen that the ECM algorithm outperforms the RIN algorithm in terms of the localization RMSE, since it has some prior knowledge about the measurement error distribution. The exclusive task of this subsection is to evaluate the localization performance of the ECM algorithm and RIN algorithm by assuming unknown measurement error distribution. Two Monte Carlo experiments (each with 1000 independent trials) are conducted. In the first experiment, the localization RMSE is computed by the two algorithms as a function of the NLOS contamination ratio α_{NL} with fixed $K = 20$ samples at each BS. In the second experiment, the localization

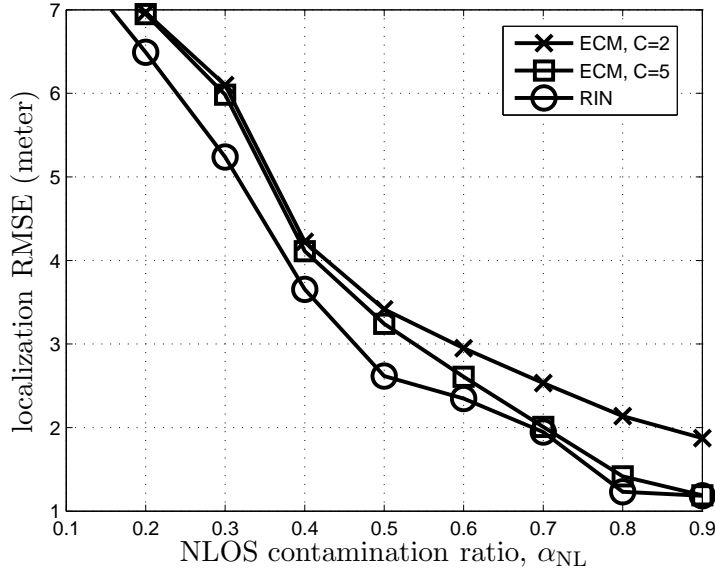


Figure 4.9. Localization RMSE (for a fixed MS position) of the ECM- and RIN algorithms versus the NLOS contamination ratio, α_{NL} .

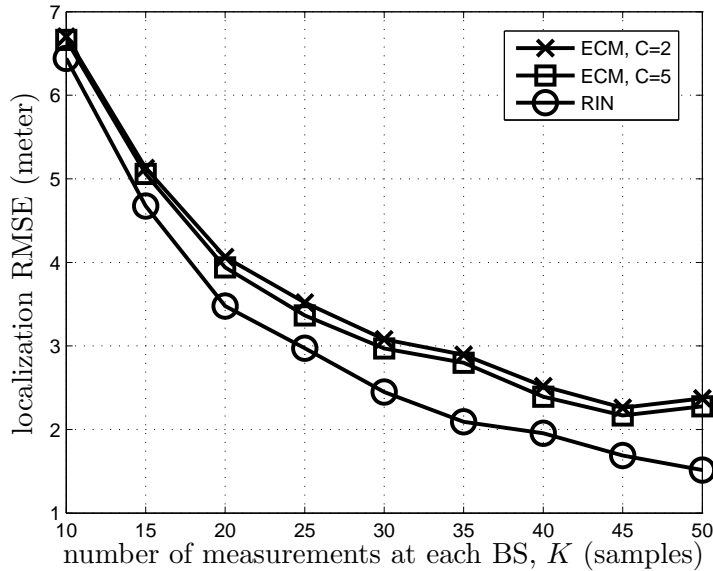


Figure 4.10. Localization RMSE (for a fixed MS position) of the ECM- and RIN algorithms versus the number of measurements K .

RMSE is computed as a function of K with fixed NLOS contamination ratio $\alpha_{NL} = 0.5$. In these two experiments, we test different mode numbers, i.e., $C = 2$ and $C = 5$, in the parametric modeling. The results are shown in Fig. 4.9 and Fig. 4.10, respectively.

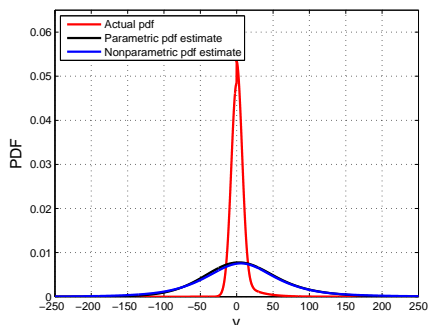
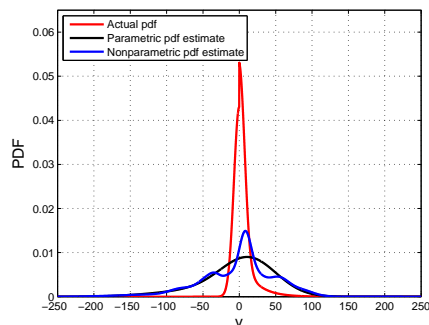
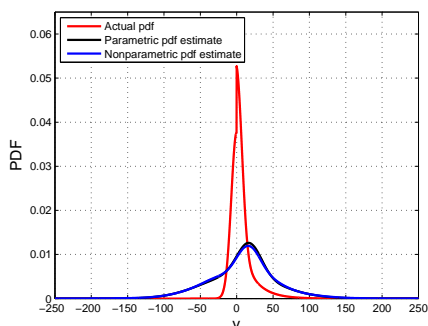
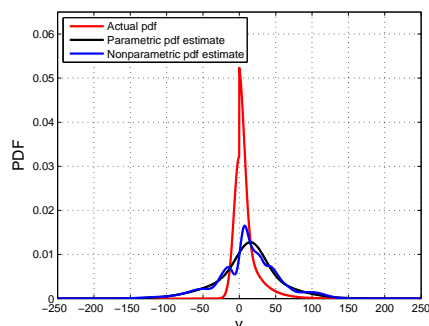
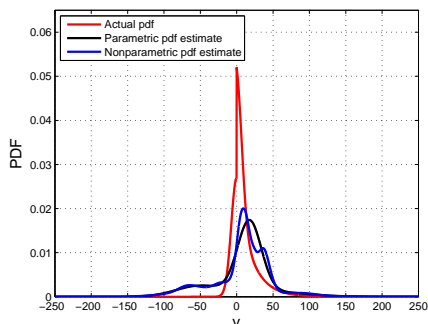
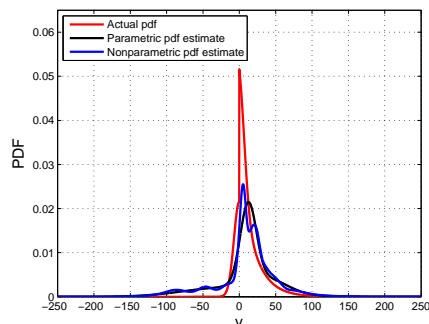
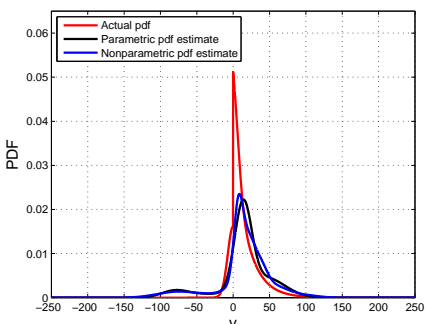
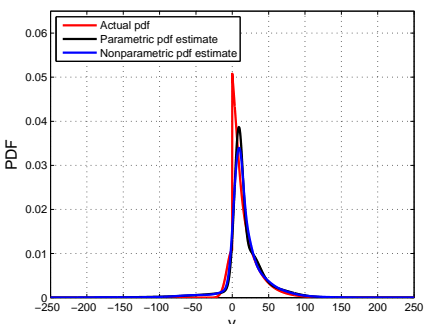
(a) $\alpha_{NL} = 0.1$ (b) $\alpha_{NL} = 0.2$ (c) $\alpha_{NL} = 0.3$ (d) $\alpha_{NL} = 0.4$ (e) $\alpha_{NL} = 0.5$ (f) $\alpha_{NL} = 0.6$ (g) $\alpha_{NL} = 0.7$ (h) $\alpha_{NL} = 0.8$

Figure 4.11. Actual measurement error distribution for different α_{NL} versus its estimates obtained both nonparametrically from the RIN algorithm and parametrically from the ECM algorithm in some Monte Carlo trials.

We observe that:

- The RIN algorithm outperforms the ECM algorithm but only modestly. This is due to the fact that more accurate pdf estimate can be found by the RIN algorithm. We give some examples in Fig. 4.11. The reason for the modest improvement lies in that although $p_V(v)$ is unknown, one component of it is a Gaussian distribution. For an unknown $p_V(v)$ of arbitrary shape, the non-parametric approach using the KDE is more advantageous than the parametric approach using only a small number of modes.
- The ECM algorithm with $C = 5$ performs slightly better than that with $C = 2$. But we note that given a fixed number of measurements, increasing C does not necessary lead to better performance. This is because more parameters need to be estimated as well. As a rule of thumb, we point out that good trade-off can be balanced when $C \approx 5$ modes, since on the one hand the main characteristics of many distributions can be well captured and on the other hand the number of parameters need to be estimated is not too large.

4.6.2.3 Results for Scenario III

In the third scenario, we consider another cellular radio network with a smaller number ($N = 4$) of BSs. Their positions are depicted in Fig. 4.12. Various experiments will be conducted according to the third scenario in which the measurement error terms $v_{i,k}$, for any fixed i and $k = 1, 2, \dots, K$, are generated either all from $p_V^{(L)}(v; \beta_L)$ or all from $p_V^{(NL)}(v; \beta_{NL})$. In other words, the channel state (LOS or NLOS) between the MS and the i th BS is unchanged at least for K samples. Again, we assume that both $p_V^{(L)}(v; \beta_L)$ and $p_V^{(NL)}(v; \beta_{NL})$ belong to the Gaussian family of distributions. Although the iid assumption does not hold any longer, we still assume that the data are generated from

$$p_V(v) = \alpha_L \mathcal{N}(v; \mu_L, \sigma_L^2) + \alpha_{NL} \mathcal{N}(v; \mu_{NL}, \sigma_{NL}^2) \quad (4.131)$$

and run the proposed algorithms.

The proposed ECM- and JMAP-ML estimators are compared with the following competitors:

- The robust iterative nonparametric (RIN) position estimator, cf. Chapter 3.

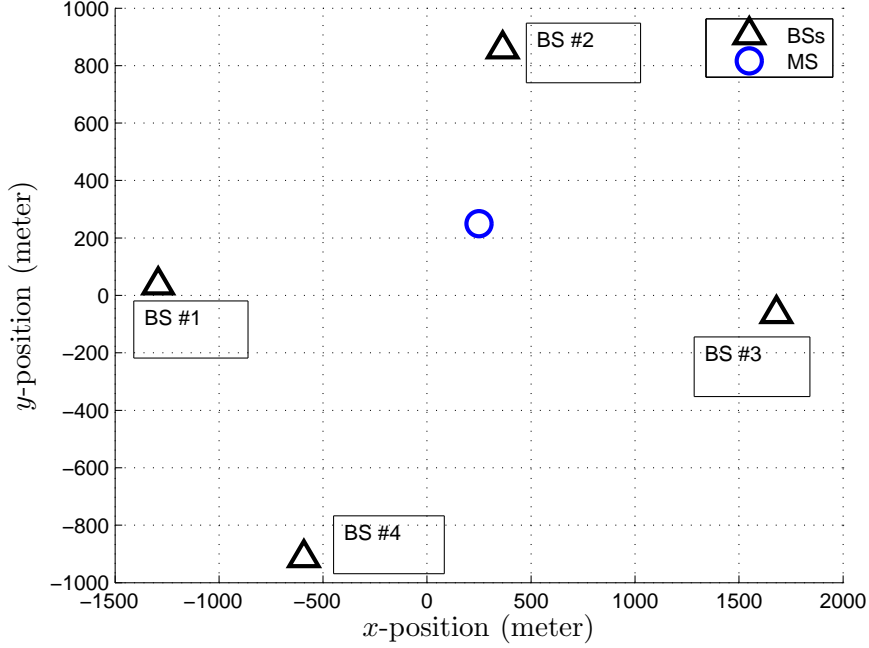


Figure 4.12. 2-D illustration of the BS geometry in an operating cellular radio network in Germany.

- The generic identify and discard (IAD)-ML estimator. The main idea of the IAD based algorithms is to first identify and discard the NLOS measurements and then use the LOS measurements only for localization. For comparison purposes only, we consider two simplified cases. In the first case, we ideally assume both the LOS channels and NLOS channels are precisely categorized in every Monte-Carlo trial. In the second case, we assume one NLOS channel is wrongly recognized as LOS channel for 5% of the total Monte-Carlo trials. Like in [52], the remaining LOS measurements are used in the maximum likelihood estimation of the position.
- The “ideal” maximum likelihood estimator, given the precise knowledge about the channel states, cf. (4.132), which is solved from

$$\begin{aligned}
 & \arg \max_{\boldsymbol{\theta}} \ln(\tilde{p}(\mathbf{r}; \boldsymbol{\theta})) \\
 &= \ln \left(\prod_{i=1}^M \prod_{k=1}^K \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_1, \sigma_1^2) \cdot \prod_{i=M+1}^N \prod_{k=1}^K \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_2, \sigma_2^2) \right) \\
 &= \sum_{i=1}^M \sum_{k=1}^K \ln \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_1, \sigma_1^2) + \sum_{i=M+1}^N \sum_{k=1}^K \ln \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_2, \sigma_2^2).
 \end{aligned} \tag{4.132}$$

The term “ideal” used here indicates that the precise knowledge about the channel

states is rarely available in reality. This “ideal” estimator can be efficiently solved in an iterative process. In (4.132), we assume, without loss of generality, the first M BSs are LOS BSs, while the remaining $N - M$ BSs are NLOS BSs. Here, (N)LOS BS is a short notation of a BS in (N)LOS with the MS.

Moreover, we compare the localization RMSE of all candidate estimators with the best achievable localization accuracy computed by (4.125) with the log-likelihood function defined in (4.132).

The initial guess of the MS position for the above algorithms is equal to the true value contaminated by Δ_{xy} whose elements are generated from a uniform distribution $\mathcal{U}[-50, 50]$ (in meter). We note that for this scenario the least-squares estimate (3.18) performs much worse. The initial guess of the Gaussian mixture model parameters for the JMAP-ML algorithm and ECM algorithm can be computed in a similar way as that derived in Algorithm 4.3 for the first scenario. More details about this initialization strategy can be found in [67].

In the first experiment, we show the parameter estimation bias of the proposed algorithms. Here, we assume that the second BS is an NLOS BS while the rest are LOS BSs. To that end, we show the mean parameter estimates of α_1 , μ_2 , σ_1^2 and x for the case $K = 25$ samples in Fig. 4.13 and Fig. 4.14. The true values of the unknown parameters are also depicted for better illustration. It is clear from the results that both the ECM algorithm and JMAP-ML algorithm generate biased estimates in the assumed scenario.

In the second experiment, localization accuracy of different position estimators is studied in three examples. The first example assumes three different mixed LOS/NLOS settings: (1) {LOS BS#1, LOS BS#2, LOS BS#3, NLOS BS#4}, (2) {NLOS BS#1, LOS BS#2, NLOS BS#3, LOS BS#4}, and (3) {LOS BS#1, NLOS BS#2, NLOS BS#3, NLOS BS#4}. For each setting, we evaluate the localization RMSE of different estimators. The results are shown in Fig. 4.15. In the second example, we study the influence of σ_L (varies from 30 to 70 meter) on the localization RMSE. Herein, we focus on the setting {LOS BS#1, LOS BS#2, LOS BS#3, NLOS BS#4}. The results are shown in Fig. 4.16. In the third example, we investigate how the number of measurements K (varies from 10 to 50 samples) affects the RMSE curves. In this example, we consider the setting {LOS BS#1, NLOS BS#2, LOS BS#3, NLOS BS#4}. The RMSE curves are depicted in Fig. 4.17. In the above examples, the best achievable localization accuracy, CRLB_{pos} , is also computed and depicted along with the localization RMSE curves.

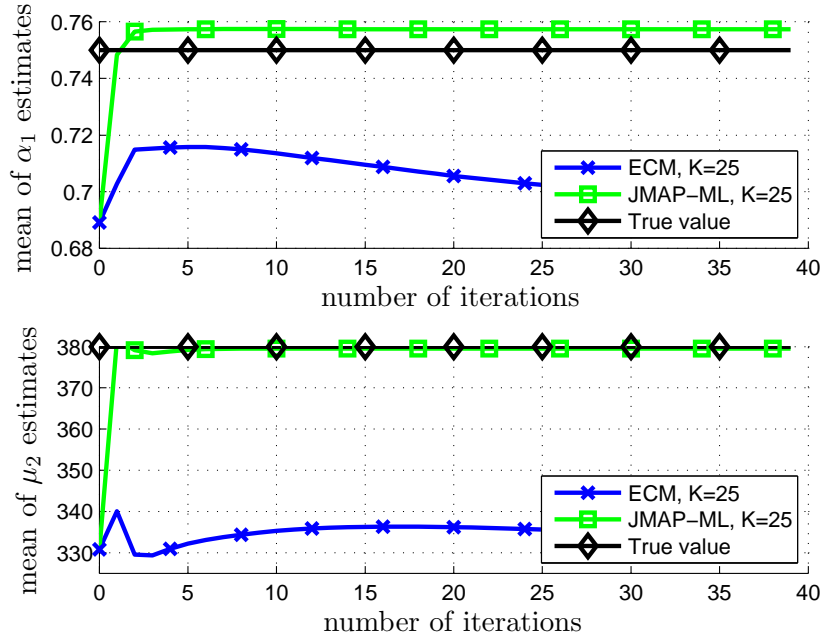


Figure 4.13. **Top sub-figure:** Mean of α_1 estimates versus the number of iterations. **Bottom sub-figure:** Mean of μ_2 estimates versus the number of iterations.

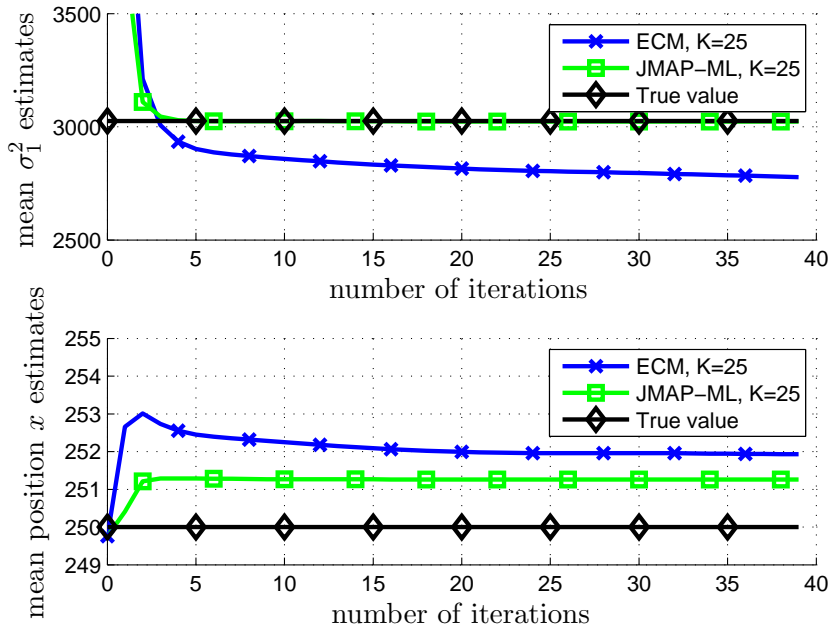


Figure 4.14. **Top sub-figure:** Mean of σ_1^2 estimates versus the number of iterations. **Bottom sub-figure:** Mean of position x estimates versus the number of iterations.

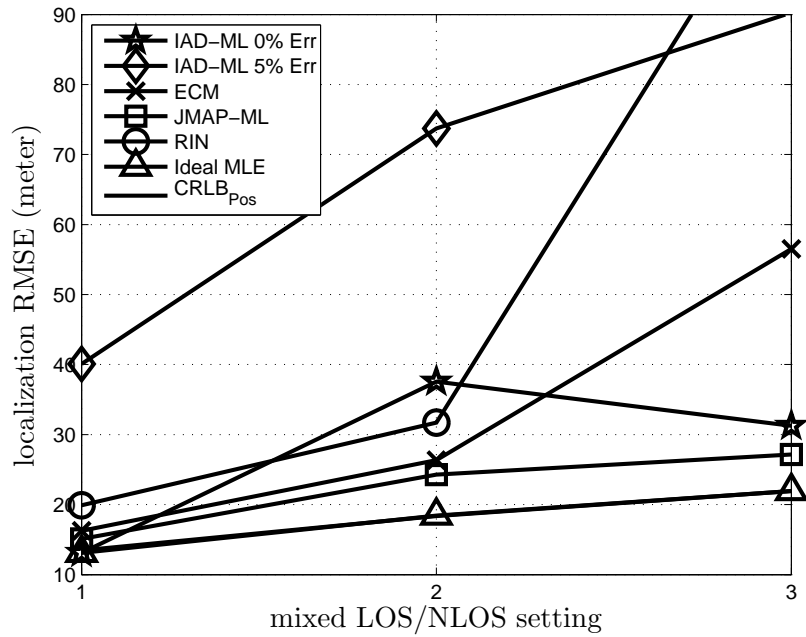


Figure 4.15. Localization RMSE of different position estimators in three different scenarios.

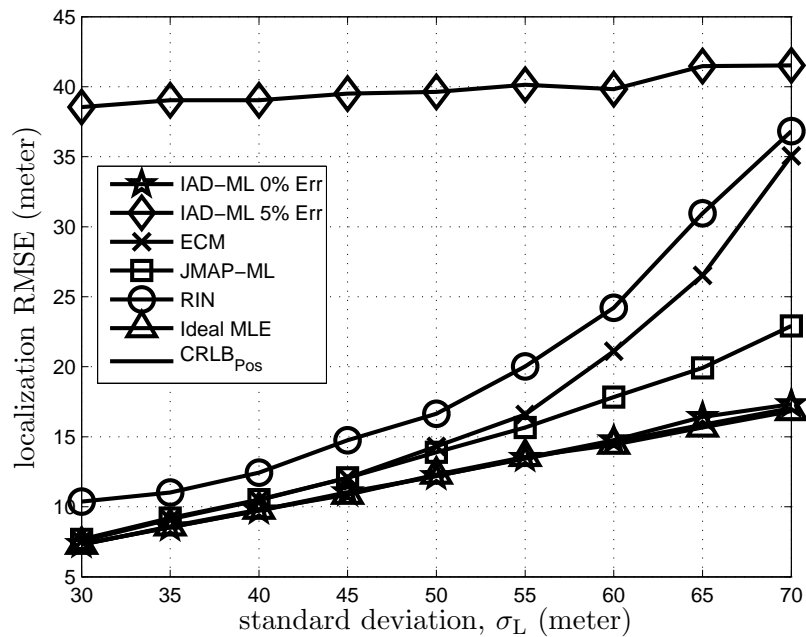


Figure 4.16. Localization RMSE of different position estimators versus the LOS measurement error standard deviation σ_L .

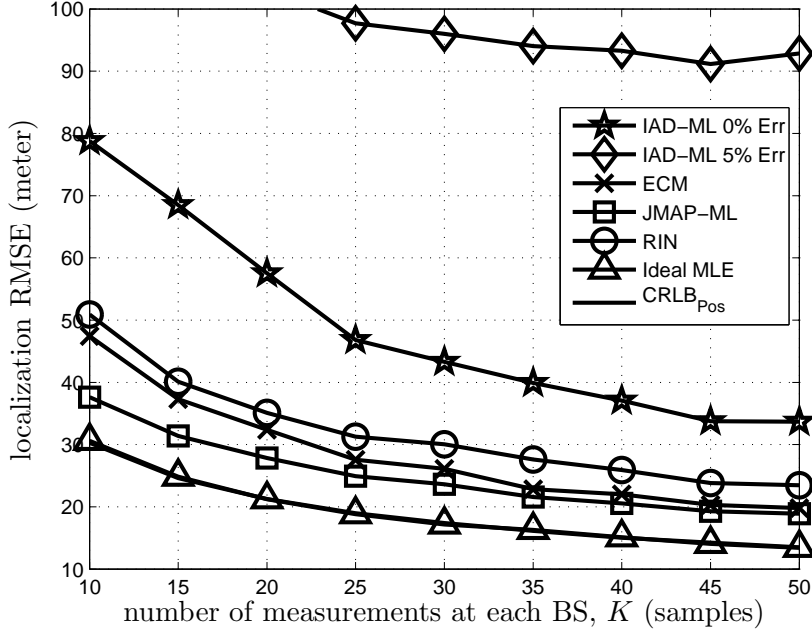


Figure 4.17. Localization RMSE of different position estimators versus the number of measurements K .

From the illustrations, we summarize the performance of the considered estimators as follows:

- The JMAP-ML estimator is closest to the “ideal” ML estimator in terms of the localization RMSE. This is because, after inserting the accurately estimated $y_{i,k}^{(\eta+1)}$, $\alpha_l^{(\eta+1)}$, $\mu_l^{(\eta+1)}$, and $\sigma_l^{2,(\eta+1)}$ into (4.20), the reduced cost function (in terms of the position θ_p only) can well approximate the ideal one with known μ_l and σ_l^2 , $l = 1, 2$ replaced by the true values in (4.132).
- The ECM estimator turns out to be biased for the considered scenario and provides inferior localization RMSE as compared to that of the JMAP-ML estimator. But still, it serves as a good approximation of the “ideal” ML estimator in many cases.
- In some cases, both the ECM algorithm and RIN algorithm break down. The key reason is that the cost function for updating the position deviates too much from the “ideal” one.
- The IAD-ML estimator (with 5% erroneous identification rate) performs the worst due to the outliers (the survival NLOS measurements). The performance will further deteriorate as the erroneous identification rate increases.

- The IAD-ML estimator (with 0% erroneous identification rate) can even achieve the ideal performance in many cases. Possible reasons are the following. Firstly, the number of the remaining LOS BSs is still sufficient (larger or equal to three). Secondly, after discarding the NLOS measurements the geometric dilution of precision (GDOP) is improved. Thirdly, although we assume known distribution of the NLOS measurement error terms, the associated parameters are unknown and the variance is large. Hence, the information about the position hidden in the NLOS measurements is negligible. As it was proven in [52], NLOS measurements can be discarded without losing any information when the noise variance goes to infinity. Fourthly, the proposed algorithms have more parameters to estimate as compared to the IAD-ML algorithm.

4.7 Conclusions

We considered robust target localization using cellular radio networks in mixed LOS/NLOS environments. In contrast to the previous chapter, we adopted Gaussian mixtures to represent the measurement error distribution in a parametric manner. In addition, we took into account some latent information, with which an EM algorithm and a JMAP-ML algorithm were first developed in general form. The two algorithms arrived at a unified maximization problem, from which a parameter estimate can be revolved as an approximation of the ML estimate. To reduce the computational hurdles, conditional maximization was adopted to convert a complicated maximization problem into several easier ones. The resulting computational complexity of the ECM algorithm and JMAP-ML algorithm scales as $\mathcal{O}(CNK)$ FLOPs per iteration, and for a fixed small C , it is much smaller than the $\mathcal{O}((NK)^2)$ FLOPs of the RIN algorithm. Therefore, it is more favorable to be used for emergence services that require very short response time. As opposed to the RIN algorithm, using Gaussian mixtures in the parametric model facilitates the proof of some convergence properties of the new algorithms, which is critical to an iterative algorithm. We then generalized the parametric model to be a mixture of distributions that belong to exponential family. An alternative view of the EM algorithm was provided to serve as a complement of the regular method. Two examples were given to elaborate on this idea. Simulations were conducted in three different scenarios, with one scenario assuming ideal conditions and the other two assuming model mismatch problems. Under the ideal conditions, the simulation results confirmed that the ECM algorithm is able to reproduce the ML estimator, given a good starting point, while the JMAP-ML algorithm can merely produce a biased estimator. In the remaining two scenarios, both the ECM algorithm

and JMAP-ML algorithm are suboptimal. Despite some performance degradation, they still presented good localization accuracy and considerable robustness against the NLOS measurements.

4.8 Appendix

4.8.1 Derivations of (4.11)

The following derivations are similar to those given in [117] for Gaussian mixture learning. For notational brevity, we introduce r_m , y_m and $h_m(\boldsymbol{\theta}_p)$ to denote the m th entry, $m = 1, 2, \dots, NK$, of \mathbf{r} , \mathbf{y} and $\mathbf{h}(\boldsymbol{\theta}_p)$, respectively. In order to evaluate (4.11), it is necessary to derive in the first place the conditional probability of the latent variables \mathbf{y} given \mathbf{r} , that is,

$$\begin{aligned} \Pr\{\mathbf{y}|\mathbf{r};\boldsymbol{\theta}\} &= \Pr\{y_{NK}|r_1, \dots, r_{NK};\boldsymbol{\theta}\} \prod_{m=1}^{NK-1} \Pr\{y_m|y_{m+1}, \dots, y_{NK}, r_1, \dots, r_{NK};\boldsymbol{\theta}\} \\ &= \prod_{m=1}^{NK} \Pr\{y_m|r_m;\boldsymbol{\theta}\} \end{aligned} \quad (4.133)$$

where the second equality follows from the fact that given r_m , y_m is independent of other latent variables and measurements. Therefore,

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) &= \sum_{\mathbf{y}} \ln(p(\mathbf{y}, \mathbf{r}; \boldsymbol{\theta})) \Pr\{\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)}\} \\ &= \sum_{\mathbf{y}} \sum_{m=1}^{NK} \ln(\alpha_{y_m} \mathcal{N}(r_m - h_m(\boldsymbol{\theta}_p); \mu_{y_m}, \sigma_{y_m}^2)) \prod_{m'=1}^{NK} \Pr\{y_{m'}|r_{m'}; \boldsymbol{\theta}^{(\eta)}\} \\ &= \sum_{y_1=1}^C \cdots \sum_{y_{NK}=1}^C \left[\sum_{m=1}^{NK} \ln(\alpha_{y_m} \mathcal{N}(r_m - h_m(\boldsymbol{\theta}_p); \mu_{y_m}, \sigma_{y_m}^2)) \prod_{m'=1}^{NK} \Pr\{y_{m'}|r_{m'}; \boldsymbol{\theta}^{(\eta)}\} \right]. \end{aligned} \quad (4.134)$$

By introducing the Kronecker's Delta function, which is given by

$$\delta(l - r_m) = \begin{cases} 1, & \text{if } l = r_m \\ 0, & \text{otherwise} \end{cases}, \quad (4.135)$$

for $l = 1, \dots, C$, the third equality in (4.134) can be rewritten as

$$\begin{aligned}
Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) &= \sum_{y_1=1}^C \cdots \sum_{y_{NK}=1}^C \left[\sum_{m=1}^{NK} \sum_{l=1}^C \delta(l - y_m) \ln(\alpha_l \mathcal{N}(r_m - h_m(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) \prod_{m'=1}^{NK} \Pr\{y_{m'} | r_{m'}; \boldsymbol{\theta}^{(\eta)}\} \right] \\
&= \sum_{m=1}^{NK} \sum_{l=1}^C \ln(\alpha_l \mathcal{N}(r_m - h_m(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) \underbrace{\left[\sum_{y_1=1}^C \cdots \sum_{y_{NK}=1}^C \delta(l - y_m) \prod_{m'=1}^{NK} \Pr\{y_{m'} | r_{m'}; \boldsymbol{\theta}^{(\eta)}\} \right]}_{(4.136)}.
\end{aligned}$$

The term in the brackets underlined can be further reduced as follows:

$$\begin{aligned}
&\sum_{y_1=1}^C \cdots \sum_{y_{NK}=1}^C \delta(l - y_m) \prod_{m'=1}^{NK} \Pr\{y_{m'} | r_{m'}; \boldsymbol{\theta}^{(\eta)}\} \\
&= \left(\sum_{y_1=1}^C \cdots \sum_{y_{m-1}=1}^C \sum_{y_{m+1}=1}^C \cdots \sum_{y_{NK}=1}^C \prod_{m'=1, m' \neq m}^{NK} \Pr\{y_{m'} | r_{m'}; \boldsymbol{\theta}^{(\eta)}\} \right) \Pr\{y_m = l | r_m; \boldsymbol{\theta}^{(\eta)}\} \\
&= \left[\prod_{m'=1, m' \neq m}^{NK} \underbrace{\left(\sum_{y_{m'}=1}^C \Pr\{y_{m'} | r_{m'}; \boldsymbol{\theta}^{(\eta)}\} \right)}_{=1} \right] \cdot \Pr\{y_m = l | r_m; \boldsymbol{\theta}^{(\eta)}\} \\
&= \Pr\{y_m = l | r_m; \boldsymbol{\theta}^{(\eta)}\}.
\end{aligned} \tag{4.137}$$

Inserting (4.137) into (4.136) yields

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) = \sum_{m=1}^{NK} \sum_{l=1}^C \ln(\alpha_l \mathcal{N}(r_m - h_m(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2)) \Pr\{y_m = l | r_m; \boldsymbol{\theta}^{(\eta)}\}. \tag{4.138}$$

Reverting to the original labeling rule, i.e., $i = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$ in (4.138), finally gives (4.11).

4.8.2 Optimality of (4.58), (4.59), and (4.60)

In order to verify the optimality of (4.58), (4.59), and (4.60), we need to evaluate the second order derivative of the corresponding cost function.

The cost function of α_l , $l = 1, 2, \dots, C$ on the $(\eta + 1)$ th iteration is given by

$$f(\alpha_1, \dots, \alpha_C, \lambda) = \sum_{i=1}^N \sum_{k=1}^K \sum_{l=1}^C \ln(\alpha_l) w_{i,k,l}^{(\eta)} + \lambda \left(\sum_{l=1}^C \alpha_l - 1 \right), \tag{4.139}$$

and (4.58) is the solution of

$$\frac{\partial f(\alpha_1, \dots, \alpha_C, \lambda)}{\partial \alpha_l} = 0, \quad l = 1, 2, \dots, C. \quad (4.140)$$

In order to prove the global optimality of (4.58), the matrix of second order derivatives of $f(\alpha_1, \dots, \alpha_C, \lambda)$ is derived as

$$F(\alpha_1, \dots, \alpha_C) = \begin{pmatrix} -\sum_{i=1}^N \sum_{k=1}^K \frac{w_{i,k,1}^{(\eta)}}{\alpha_1^2} & 0 & 0 & \cdots & 0 \\ 0 & -\sum_{i=1}^N \sum_{k=1}^K \frac{w_{i,k,2}^{(\eta)}}{\alpha_2^2} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -\sum_{i=1}^N \sum_{k=1}^K \frac{w_{i,k,C}^{(\eta)}}{\alpha_C^2} \end{pmatrix}. \quad (4.141)$$

Since both $\alpha_l, l = 1, 2, \dots, C$ and $w_{i,k,l}^{(\eta)}, i = 1, 2, \dots, N, k = 1, 2, \dots, K$, and $l = 1, 2, \dots, C$ are non-negative, the cost function $f(\alpha_1, \dots, \alpha_C, \lambda)$ is concave in terms of $\alpha_1, \dots, \alpha_C$. As a result, (4.58) is the global maximizer.

The cost function of μ_l on the $(\eta + 1)$ th iteration is given by

$$f(\mu_l) = \sum_{i=1}^N \sum_{k=1}^K \left(-\frac{\ln(2\pi\sigma_l^{2,(\eta)})}{2} - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_l)^2}{2\sigma_l^{2,(\eta)}} \right) w_{i,k,l}^{(\eta)} \quad (4.142)$$

and (4.59) is the solution of

$$\frac{\partial f(\mu_l)}{\partial \mu_l} = \sum_{i=1}^N \sum_{k=1}^K \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_l)}{\sigma_l^{2,(\eta)}} w_{i,k,l}^{(\eta)} = 0. \quad (4.143)$$

Similarly, the second order derivative of $f(\mu_l)$ with respect to μ_l is derived as

$$\frac{\partial^2 f(\mu_l)}{\partial \mu_l^2} = -\frac{1}{\sigma_l^{2,(\eta)}} \sum_{i=1}^N \sum_{k=1}^K w_{i,k,l}^{(\eta)} \leq 0, \quad (4.144)$$

which proves the concavity of $f(\mu_l)$ in terms of μ_l .

The cost function of σ_l^2 on the $(\eta + 1)$ th iteration is given by

$$f(\sigma_l^2) = \sum_{i=1}^N \sum_{k=1}^K \left(-\frac{\ln(2\pi\sigma_l^2)}{2} - \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_l^{(\eta+1)})^2}{2\sigma_l^2} \right) w_{i,k,l}^{(\eta)} \quad (4.145)$$

and (4.60) is the solution of

$$\frac{\partial f(\sigma_l^2)}{\partial \sigma_l^2} = 0. \quad (4.146)$$

To prove that (4.60) is the global maximizer of $f(\sigma_l^2)$ is a bit tricky. The methodology provided in [118, Chapter 7] is adopted here for the proof, namely, we introduce a new optimization variable $\lambda_l = 1/\sigma_l^2$, whose support is as same as σ_l^2 . Reformulating the cost function in terms of λ_l yields

$$f(\lambda_l) = \sum_{i=1}^N \sum_{k=1}^K \left(-\frac{1}{2} \ln(2\pi \frac{1}{\lambda_l}) - \frac{1}{2} \lambda_l (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_l^{(\eta+1)})^2 \right) w_{i,k,l}^{(\eta)}. \quad (4.147)$$

Taking the first order derivative of $f(\lambda_l)$ with respect to λ_l gives

$$\frac{\partial f(\lambda_l)}{\partial \lambda_l} = \sum_{i=1}^N \sum_{k=1}^K \left(\frac{1}{2\lambda_l} - \frac{1}{2} (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_l^{(\eta+1)})^2 \right) w_{i,k,l}^{(\eta)}. \quad (4.148)$$

Setting the above obtained first order derivative to zero and solving for λ_l yields

$$\lambda_l^{(\eta+1)} = \frac{\sum_{i=1}^N \sum_{k=1}^K w_{i,k,l}^{(\eta)}}{\sum_{i=1}^N \sum_{k=1}^K (r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_l^{(\eta+1)})^2 w_{i,k,l}^{(\eta)}}. \quad (4.149)$$

Interestingly, the inverse of $\lambda_l^{(\eta+1)}$ coincides with (4.60) solved from (4.146). The second order derivative of $f(\lambda_l)$ with respect to λ_l is derived as

$$\frac{\partial^2 f(\lambda_l)}{\partial \lambda_l^2} = -\frac{1}{2\lambda_l^2} \sum_{i=1}^N \sum_{k=1}^K w_{i,k,l}^{(\eta)} \leq 0, \quad (4.150)$$

which proves that the cost function $f(\lambda_l)$ is concave in terms of λ_l . Hence, $\lambda_l^{(\eta+1)}$ is the global maximizer, and this indirectly proves that $f(\sigma_l^2)$ is globally maximized at (4.60).

4.8.3 Derivations of $\mathcal{O}(CNK)$ Complexity

We define the FLOPs required for some elementary operations as follows:

1. \mathcal{E}_{add} : FLOPs for addition.
2. \mathcal{E}_{sub} : FLOPs for subtraction.
3. \mathcal{E}_{mul} : FLOPs for multiplication.
4. \mathcal{E}_{div} : FLOPs for division.

5. \mathcal{E}_{exp} : FLOPs for exponential.
6. \mathcal{E}_{pow} : FLOPs for raising to real power.
7. \mathcal{E}_{sqr} : FLOPs for square root.

Note that the actual FLOPs required for the above operations may vary with processors. Typical values based on a Pentium®4 processor were given in [119].

Since both the ECM algorithm and JMAP-ML algorithm are iterative in nature, the computational complexity is evaluated in terms of total FLOPs consumed for the joint estimation (i.e., the second stage of Algorithm 4.1) on one iteration, say the $(\eta + 1)$ th.

4.8.3.1 ECM Algorithm

The joint estimation step first evaluates $w_{i,k,l}^{(\eta)} = P_{i,k,l}^{(\eta)}$, for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$ and $l = 1, 2, \dots, C$, given the prior parameter estimate $\boldsymbol{\theta}^{(\eta)}$. This requires us to compute

$$\text{Eval(1)} : d_i(\boldsymbol{\theta}_p^{(\eta)}) = \sqrt{(x_i - x^{(\eta)})^2 + (y_i - y^{(\eta)})^2}, \quad (4.151)$$

for $i = 1, 2, \dots, N$;

$$\text{Eval(2)} : r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}), \quad (4.152)$$

for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$;

$$\text{Eval(3)} : \Phi_{i,k,l}^{(\eta)} = \alpha_l^{(\eta)} (2\pi\sigma_l^{2,(\eta)})^{-1/2} \cdot \exp \left[\frac{-(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}) - \mu_l^{(\eta)})^2}{2\sigma_l^{2,(\eta)}} \right], \quad (4.153)$$

for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$, $l = 1, 2, \dots, C$; and

$$\text{Eval(4)} : P_{i,k,l}^{(\eta)} = \frac{\Phi_{i,k,l}^{(\eta)}}{\sum_{l=1}^C \Phi_{i,k,l}^{(\eta)}}, \quad (4.154)$$

for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$, $l = 1, 2, \dots, C$.

It is clear that Eval(1) requires $N(2\mathcal{E}_{sub} + 2\mathcal{E}_{pow} + 1\mathcal{E}_{sqr} + 1\mathcal{E}_{add})$ FLOPs, Eval(2) $NK\mathcal{E}_{sub}$ FLOPs, Eval(3) $C((NK+3)\mathcal{E}_{mul} + (NK+1)\mathcal{E}_{pow} + NK\mathcal{E}_{sub} + NK\mathcal{E}_{div} + NK\mathcal{E}_{exp})$ FLOPs, and Eval(4) $NK((C-1)\mathcal{E}_{add} + C\mathcal{E}_{div})$ FLOPs.

Having $P_{i,k,l}^{(\eta)}$'s, we then compute

$$\text{Eval(5)} : \alpha_l^{(\eta+1)}, \quad l = 1, 2, \dots, C, \quad (4.155)$$

$$\text{Eval(6)} : \mu_l^{(\eta+1)}, \quad l = 1, 2, \dots, C, \quad (4.156)$$

$$\text{Eval(7)} : \sigma_l^{2,(\eta+1)}, \quad l = 1, 2, \dots, C, \quad (4.157)$$

according to (4.58), (4.59) and (4.60), respectively. It is easy to verify that Eval(5) requires $(C - 1)[(NK - 1)\mathcal{E}_{add} + 1\mathcal{E}_{div}] + (C - 1)\mathcal{E}_{sub}$ FLOPs. The calculation of Eval(5) is performed with the aid of $\sum_{l=1}^C \alpha_l^{(\eta+1)} = 1$. Eval(6) requires $C(NK\mathcal{E}_{mul} + (NK - 1)\mathcal{E}_{add} + \mathcal{E}_{div})$ FLOPs and Eval(7) requires $C((NK + 1)\mathcal{E}_{pow} + NK\mathcal{E}_{mul} + (NK - 1)\mathcal{E}_{add} + \mathcal{E}_{div} + \mathcal{E}_{sub})$ FLOPs. Let us define $\text{FLOP}(\boldsymbol{\theta}_e)$ to be the total number of FLOPs consumed for computing an estimate of $\boldsymbol{\theta}_e$ on one ECM iteration. It is straightforward that $\text{FLOP}(\boldsymbol{\theta}_e)$ is equal to the total FLOPs consumed in Eval(1) through Eval(7) with the final result

$$\text{FLOP}(\boldsymbol{\theta}_e) \approx 4CNK\mathcal{E}_{add} + CNK\mathcal{E}_{sub} + 3CNK\mathcal{E}_{mul} + 2CNK\mathcal{E}_{div} + 2CNK\mathcal{E}_{pow} + CNK\mathcal{E}_{exp}. \quad (4.158)$$

The numerical evaluation of $\boldsymbol{\theta}_p^{(\eta+1)}$ is performed in an attempt to minimize $f(\boldsymbol{\theta}_p)$, cf. (4.159), via the BFGS-QN method which involves another iterative procedure. Analogously, the FLOPs required for one iteration of this local search is counted. This requires repetitive evaluation of the cost function $f(\boldsymbol{\theta}_p)$ and its gradient at a certain point $\boldsymbol{\theta}_p$, namely,

$$\text{Eval(8)} : f(\boldsymbol{\theta}_p) = \sum_{l=1}^C \frac{1}{\sigma_l^{2,(\eta+1)}} \sum_{i=1}^N \sum_{k=1}^K \left(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)} \right)^2 P_{i,k,l}^{(\eta)}. \quad (4.159)$$

$$\text{Eval(9)} : \nabla_x f(\boldsymbol{\theta}_p) = \sum_{l=1}^C \frac{2}{\sigma_l^{2,(\eta+1)}} \sum_{i=1}^N \frac{x_i - x}{d_i(\boldsymbol{\theta}_p)} \sum_{k=1}^K \left(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)} \right) P_{i,k,l}^{(\eta)} \quad (4.160)$$

$$\text{Eval(10)} : \nabla_y f(\boldsymbol{\theta}_p) = \sum_{l=1}^C \frac{2}{\sigma_1^{2,(\eta+1)}} \sum_{i=1}^N \frac{y_i - y}{d_i(\boldsymbol{\theta}_p)} \sum_{k=1}^K \left(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)} \right) P_{i,k,l}^{(\eta)} \quad (4.161)$$

Some parts of (4.159), (4.160) and (4.161) can be calculated *a priori*. They are,

$$\text{Eval(11)} : d_i(\boldsymbol{\theta}_p), \quad i = 1, 2, \dots, N. \quad (4.162)$$

$$\text{Eval(12)} : \frac{x_i - x}{d_i(\boldsymbol{\theta}_p)}, \quad i = 1, 2, \dots, N. \quad (4.163)$$

$$\text{Eval(13)} : \frac{y_i - y}{d_i(\boldsymbol{\theta}_p)}, \quad i = 1, 2, \dots, N. \quad (4.164)$$

$$\text{Eval(14)} : r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)}, \quad i = 1, 2, \dots, N, k = 1, 2, \dots, K, l = 1, 2, \dots, C \quad (4.165)$$

$$\text{Eval(15)} : \sum_{k=1}^K \left(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l^{(\eta+1)} \right) P_{i,k,l}^{(\eta)}, \quad i = 1, 2, \dots, N, l = 1, 2, \dots, C. \quad (4.166)$$

It is easy to verify that Eval(11) requires $N(2\mathcal{E}_{sub} + 2\mathcal{E}_{pow} + 1\mathcal{E}_{sqrt} + 1\mathcal{E}_{add})$ FLOPs; both Eval(12) and Eval(13) require $N(1\mathcal{E}_{sub} + 1\mathcal{E}_{div})$ FLOPs; Eval(14) requires $(C+1)NK\mathcal{E}_{sub}$

FLOPs; and Eval(15) requires $CN(K\mathcal{E}_{mul} + (K - 1)\mathcal{E}_{add})$ FLOPs. Hence, Eval(8) requires $C(NK\mathcal{E}_{pow} + NK\mathcal{E}_{mul} + (NK - 1)\mathcal{E}_{add} + 1\mathcal{E}_{div})$ FLOPs; both Eval(9) and Eval(10) require $C(N\mathcal{E}_{mul} + (N - 1)\mathcal{E}_{add} + 1\mathcal{E}_{div})$ FLOPs. Let us define $\text{FLOP}(f(\boldsymbol{\theta}_p))$ and $\text{FLOP}(\nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p))$ to denote respectively the total number of FLOPs required to evaluate the cost function as well as its gradient at a certain point $\boldsymbol{\theta}_p$. $\text{FLOP}(f(\boldsymbol{\theta}_p))$ is the total FLOPs consumed in Eval(11) through Eval(14) and Eval(8). The final result is

$$\text{FLOP}(f(\boldsymbol{\theta}_p)) \approx CNK\mathcal{E}_{add} + CNK\mathcal{E}_{sub} + CNK\mathcal{E}_{mul} + CNK\mathcal{E}_{pow}. \quad (4.167)$$

Similarly, $\text{FLOP}(\nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p))$ is the total FLOPs consumed in Eval(9) through Eval(15), and the final result is

$$\text{FLOP}(\nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p)) \approx CNK\mathcal{E}_{add} + CNK\mathcal{E}_{sub} + CNK\mathcal{E}_{mul}. \quad (4.168)$$

Having the above knowledge, we are then able to proceed with the complexity analysis of the BFGS-QN method on one iteration, cf. Algorithm 3.2. First, a search direction is computed according to Eval(16) : $\mathbf{s}_j = -\mathbf{H}_j \cdot \nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p^{(\eta+1,j)})$, which requires $4\mathcal{E}_{mul} + 2\mathcal{E}_{add} + \text{FLOP}(\nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p))$ FLOPs. Note that j is the iteration index and $\boldsymbol{\theta}_p^{(\eta+1,0)} = \boldsymbol{\theta}_p^{(\eta)}$ when $j = 0$. The selection of a step size τ_j is done via cubic line search (Eval(17)) in the second step, which requires approximately $N_{itr}^{\text{CLS}}(\text{FLOP}(f(\boldsymbol{\theta}_p)) + \text{FLOP}(\nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p)))$ FLOPs, where N_{itr}^{CLS} is the number of trials used to determine a suitable step size. In the third step, we update the estimate by $\boldsymbol{\theta}_p^{(\eta+1,j+1)} = \boldsymbol{\theta}_p^{(\eta+1,j)} + \tau_j\mathbf{s}_j$, which (Eval(18)) requires $2\mathcal{E}_{mul} + 2\mathcal{E}_{add}$ FLOPs. In the fourth step, the evaluation (Eval(19)) of $\boldsymbol{\gamma}_j = \nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p^{(\eta+1,j+1)}) - \nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p^{(\eta+1,j)})$ requires $\text{FLOP}(\nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p)) + 2\mathcal{E}_{sub}$ FLOPs. The fifth step approximates the Hessian matrix, which (Eval(20)) requires in total $36\mathcal{E}_{mul} + 19\mathcal{E}_{add} + 10\mathcal{E}_{div}$ FLOPs. Let us define $\text{FLOP}(\boldsymbol{\theta}_p)$ to be the total number of FLOPs required to compute an estimate of $\boldsymbol{\theta}_p$ for one ECM iteration. It is equal to N_{itr}^{QN} times of the total FLOPs consumed in Eval(16) through Eval(20). The final result is

$$\text{FLOP}(\boldsymbol{\theta}_p) \approx N_{itr}^{\text{QN}} [(N_{itr}^{\text{CLS}} + 2) \cdot \text{FLOP}(\nabla_{\boldsymbol{\theta}_p}f(\boldsymbol{\theta}_p)) + N_{itr}^{\text{CLS}} \cdot \text{FLOP}(f(\boldsymbol{\theta}_p))] \quad (4.169)$$

where N_{itr}^{QN} is the total number of QN iterations.

Gathering the above results, it is concluded that the computational complexity of one single ECM iteration scales as $\mathcal{O}(CNK)$ FLOPs under the conditions:

1. The maximum iteration numbers of N_{itr}^{QN} and N_{itr}^{CLS} are constants.
2. Only the factors C , N , and K can go to infinity.

4.8.3.2 JMAP-ML Algorithm

The difference between the ECM algorithm and JMAP-ML algorithm lies in the calculation of the weighting factor $w_{i,k,l}$. The joint estimation step on the $(\eta + 1)$ th iteration of the JMAP-ML algorithm starts with the MAP estimation of the latent variables $y_{i,k}^{(\eta+1)}$, for $i = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$. Since $y_{i,k}^{(\eta+1)}$ is discrete valued and belongs to $\{1, 2, \dots, C\}$, the global optimal solution can be easily found by comparing the values of $\Phi_{i,k,l}^{(\eta)}$, $l = 1, 2, \dots, C$. This requires us to compute

$$\Phi_{i,k,l}^{(\eta)} = \alpha_l^{(\eta)} \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p^{(\eta)}); \mu_l^{(\eta)}, \sigma_l^{2,(\eta)}) \quad (4.170)$$

for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$, and $l = 1, 2, \dots, C$. As it was shown for the ECM algorithm, this step requires in total $C((NK + 3)\mathcal{E}_{mul} + (NK + 1)\mathcal{E}_{pow} + NK\mathcal{E}_{sub} + NK\mathcal{E}_{div} + NK\mathcal{E}_{exp})$ FLOPs. For a fixed i and k , in total $C - 1$ comparisons of two real numbers need to be performed to determine the global optimal $y_{i,k}^{(\eta+1)}$. Hence, the complexity of determining all $y_{i,k}^{(\eta+1)}$, for $i = 1, 2, \dots, N$, $k = 1, 2, \dots, K$, scales as $\mathcal{O}(CNK)$ FLOPs. Having $y_{i,k}^{(\eta+1)}$, $w_{i,k,l}^{(\eta)}$ is then calculated to be $\delta(l - y_{i,k}^{(\eta+1)})$. The remaining computations in the JMAP-ML algorithm are similar to those described in the ECM algorithm except that $w_{i,k,l}^{(\eta)}$ is non-zero only for one mixture component. Consequently, we can conclude that the computational complexity of the JMAP-ML algorithm scales as $\mathcal{O}(CNK)$ as well.

4.8.4 Expression of $\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{r}; \boldsymbol{\theta})$

The log-likelihood function of the incomplete data, assuming a C -mode Gaussian mixture in the parametric approach to the modeling of $p_V(v)$, is given by

$$\ln p(\mathbf{r}; \boldsymbol{\theta}) = \sum_{i=1}^N \sum_{k=1}^K \ln p(r_{i,k}; \boldsymbol{\theta}) = \sum_{i=1}^N \sum_{k=1}^K \ln \left(\sum_{l=1}^C \alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2) \right). \quad (4.171)$$

The gradient of $\ln p(\mathbf{r}; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ is defined as

$$\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{r}; \boldsymbol{\theta}) = \left[\frac{\partial}{\partial \alpha_1}, \frac{\partial}{\partial \alpha_2}, \dots, \frac{\partial}{\partial \alpha_C}, \frac{\partial}{\partial \mu_1}, \frac{\partial}{\partial \sigma_1^2}, \dots, \frac{\partial}{\partial \mu_C}, \frac{\partial}{\partial \sigma_C^2}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right]^T \ln p(\mathbf{r}; \boldsymbol{\theta}) \quad (4.172)$$

where for $l = 1, 2, \dots, C$,

$$\frac{\partial \ln p(\mathbf{r}; \boldsymbol{\theta})}{\partial \alpha_l} = \sum_{i=1}^N \sum_{k=1}^K \frac{1}{p(r_{i,k}; \boldsymbol{\theta})} \cdot \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2), \quad (4.173)$$

$$\frac{\partial \ln p(\mathbf{r}; \boldsymbol{\theta})}{\partial \mu_l} = \sum_{i=1}^N \sum_{k=1}^K \frac{\alpha_l}{p(r_{i,k}; \boldsymbol{\theta})} \cdot \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2) \cdot \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l)}{\sigma_l^2}, \quad (4.174)$$

$$\frac{\partial \ln p(\mathbf{r}; \boldsymbol{\theta})}{\partial \sigma_l^2} = \sum_{i=1}^N \sum_{k=1}^K \frac{\alpha_l}{p(r_{i,k}; \boldsymbol{\theta})} \cdot \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2) \cdot \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l)^2 - \sigma_l^2}{\sigma_l^4}, \quad (4.175)$$

and

$$\frac{\partial \ln p(\mathbf{r}; \boldsymbol{\theta})}{\partial x} = \sum_{i=1}^N \sum_{k=1}^K \frac{-1}{p(r_{i,k}; \boldsymbol{\theta})} \sum_{l=1}^C \alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2) \cdot \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l)}{\sigma_l^2} \cdot \frac{x - x_i}{d_i(\boldsymbol{\theta}_p)}, \quad (4.176)$$

$$\frac{\partial \ln p(\mathbf{r}; \boldsymbol{\theta})}{\partial y} = \sum_{i=1}^N \sum_{k=1}^K \frac{-1}{p(r_{i,k}; \boldsymbol{\theta})} \sum_{l=1}^C \alpha_l \mathcal{N}(r_{i,k} - d_i(\boldsymbol{\theta}_p); \mu_l, \sigma_l^2) \cdot \frac{(r_{i,k} - d_i(\boldsymbol{\theta}_p) - \mu_l)}{\sigma_l^2} \cdot \frac{y - y_i}{d_i(\boldsymbol{\theta}_p)}. \quad (4.177)$$

4.8.5 An Initialization Example

Algorithm 4.3 is developed based on [105, Sec.1.4.16], that is, the exact mean μ_v and variance σ_v^2 of a two-mode mixture distribution $p_V(v)$ can be found, respectively, by

$$\mu_v = \alpha_1 \mu_1 + \alpha_2 \mu_2, \quad \sigma_v^2 = \sum_{l=1}^2 \alpha_l \sigma_l^2 + \alpha_l (\mu_l - \mu_v)^2. \quad (4.178)$$

Algorithm 4.3 Initialization of Two-Mode Gaussian Mixture Model Parameters

1. Extract residuals $\hat{v}_{i,k} = r_{i,k} - d_i(\hat{\boldsymbol{\theta}}_p)$, $i = 1, 2, \dots, N$ and $k = 1, 2, \dots, K$, given a position estimate $\hat{\boldsymbol{\theta}}_p$.
2. Approximate the actual mean μ_v and variance σ_v^2 of $p_V(v)$, respectively, by

$$\hat{\mu}_v = \frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K \hat{v}_{i,k}, \quad \hat{\sigma}_v^2 = \frac{1}{NK} \sum_{i=1}^N \sum_{k=1}^K (\hat{v}_{i,k} - \hat{\mu}_v)^2. \quad (4.179)$$

3. Let $\varepsilon^{(0)}$ vary from 0.1 to 0.9 at an increment 0.05. For each $\varepsilon^{(0)}$, compute

$$\begin{aligned} \alpha_1^{(0)} &= 1 - \varepsilon^{(0)}, \quad \alpha_2^{(0)} = \varepsilon^{(0)}, \quad \mu_1^{(0)} = 0, \quad \mu_2^{(0)} = |\hat{\mu}_v / \varepsilon^{(0)}|, \\ \sigma_1^{2,(0)} &= \sigma_2^{2,(0)} = \left| \hat{\sigma}_v^2 - \varepsilon^{(0)} (1 - \varepsilon^{(0)}) (\mu_2^{(0)})^2 \right|. \end{aligned}$$

4. Amongst all the candidate initial guesses, choose the one maximizing $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$.
-

Chapter 5

Cooperative Localization in WSNs: Parametric Modeling

It was shown in Chapter 4 that the computational complexity of locating a target node in cellular radio networks can be largely reduced by adopting a parametric approach to the modeling of the measurement error distribution. We adopt the same idea for cooperative localization using wireless sensor networks in this chapter. Specifically, we study TOA based cooperative localization algorithms that apply the expectation-conditional maximization (ECM) criterion upon a parametric C -mode Gaussian mixture modeling of the measurement error. Analogous algorithms that use received-signal-strength measurements and/or joint maximum *a posteriori*-maximum likelihood (JMAP-ML) criterion can be developed in a similar fashion. Both centralized and distributed algorithms will be considered, but more emphases will be put on the distributed algorithms. This is due to the drawbacks of the centralized algorithms, namely the need for a fusion center and heavy energy consumption for communicating data to a fusion center in large-scale wireless sensor networks [6]. On the contrary, each agent determines its own position using only local information in distributed algorithms. Distributed algorithms are scalable in general, independent of a fusion center, less sensitive to sensor failure as compared to the corresponding centralized solutions, making them highly demanded for large-scale sensor networks [88].

The organization of this chapter is as follows. Section 5.1 introduces the signal model and states the problem at hand. In Section 5.2, three representative classes of cooperative localization algorithms are revisited and will be used as competitors of the proposed algorithms in the subsequent sections. In Section 5.3, numerical difficulties in solving the maximum likelihood (ML) estimator is shown in the first place. Then, a series of ECM algorithms are developed to approximate the ML estimator with less computational efforts. The proposed ECM algorithms are employed in a snapshot-based solution for localization of dynamic sensor networks at the end of this section. Section 5.4 systematically studies the computational complexity and communication overhead of the proposed ECM algorithms. Section 5.5 computes the Cramér-Rao lower bound (CRLB) numerically and gives a measure of the localization accuracy. Section 5.6 performs various simulations based on both synthetic and real settings. Section 5.7 concludes this chapter, and Section 5.8 provides some useful derivations.

5.1 Signal Model and Problem Statement

Throughout this chapter, cooperative localization is considered in a connected network comprising a total number of N wireless sensors in a two-dimensional (2-D) space. Without loss of generality, let $\mathcal{N}_a = \{1, 2, \dots, N_u\}$ be the set of indices of the agents, whose positions $\{\mathbf{p}_i = [x_i, y_i]^T, i \in \mathcal{N}_a\}$ are unknown and let $\mathcal{N}_b = \{N_u + 1, N_u + 2, \dots, N\}$ be the set of indices of the anchors with known positions.

In order to localize the agents, a two-stage procedure is adopted. In the first stage, every sensor broadcasts its sensor ID and listen for its neighboring sensors' broadcasts. Then, each agent obtains a set of distance measurements relative to its neighboring sensors, which can be done, for instance, by estimating the TOA of a received signal. In the second stage, the unknown agent positions will be estimated concurrently in a localization algorithm based on the obtained distance measurements.

The development of new cooperative localization algorithms relies on a statistical measurement model as follows:

$$r_{i,j} = d(\mathbf{p}_i, \mathbf{p}_j) + v_{i,j} \quad (5.1)$$

where $r_{i,j}$ is a distance measurement obtained at sensor i in cooperation with sensor j , $d(\mathbf{p}_i, \mathbf{p}_j) = \|\mathbf{p}_i - \mathbf{p}_j\|$ denotes the true Euclidean distance between the two sensors, and $v_{i,j}$ is an additive measurement error term. In the subsequent sections, we will occasionally use d_{ij} to denote $d(\mathbf{p}_i, \mathbf{p}_j)$ for brevity. The measurement error terms observed for different sensor pairs are assumed to be iid according to $p_V(v)$, which is approximated parametrically by a C -mode Gaussian mixture as follows:

$$p_V(v) \approx \hat{p}_V(v) = \sum_{l=1}^C \alpha_l \mathcal{N}(v; \mu_l, \sigma_l^2) \quad (5.2)$$

where C is the total number of mixture components, α_l is the mixing coefficient of the l th Gaussian component $\mathcal{N}(v; \mu_l, \sigma_l^2)$. The mixture model parameters α_l , μ_l and σ_l^2 , $l = 1, 2, \dots, C$, are assumed to be unknown.

In order to reduce the communication overhead and computational complexity, we assume, throughout this chapter, that $r_{i,j} = r_{j,i}$ and only one of them (choosing $r_{i,j}$ with $j > i$) will be routed to a fusion center. This assumption approximately holds when the sensors are equipped with an omni-directional antenna and conduct the interrogation in a pair of reciprocal channels. Alternatively, sensor i could transmit $r_{i,j}$ ($j > i$) back to sensor j such that $r_{j,i}$ is set to be equal to $r_{i,j}$ without performing an additional trial of TOA estimation on sensor j . Besides, distance measurements

obtained between anchors are ignored when estimating the agent positions. But they are useful for the initialization (as will be seen in Section 5.3). In this way, only c.a. half of the measurements will be used in the inference tasks. Before proceeding further, we introduce the following notations:

- $\boldsymbol{\theta} = [\boldsymbol{\theta}_e^T, \boldsymbol{\theta}_a^T, \boldsymbol{\theta}_p^T]^T$ is a vector of all unknown parameters to be determined, where $\boldsymbol{\theta}_p = [\mathbf{p}_1^T, \dots, \mathbf{p}_{N_u}^T]^T$, $\boldsymbol{\theta}_e = [\alpha_1, \dots, \alpha_C, \mu_1, \sigma_1^2, \dots, \mu_C, \sigma_C^2]^T$ and $\boldsymbol{\theta}_a$ is null in this case.
- $\mathcal{H}(i) = \{j : j \in \{\mathcal{N}_a \cup \mathcal{N}_b\} \text{ and } d(\mathbf{p}_i, \mathbf{p}_j) < R_c\}$ is a set of all neighboring sensors of agent i , $i \in \mathcal{N}_a$. Here, R_c is the maximal communication range of a sensor and an ideal model is adopted for determining the neighborhood of each agent. In practice, instead, a neighboring sensor can be determined by comparing the received signal strength with a certain threshold, beyond which data packages cannot be demodulated.
- $\Gamma = \{(i, j) : i \in \mathcal{N}_a, j \in \mathcal{H}(i), \text{ and } j > i\}$ is a set of all sensor pairs that contribute distance measurements.
- \mathbf{r} is a column vector containing all distance measurements $r_{i,j}$, $\forall (i, j) \in \Gamma$. The data structure of \mathbf{r} follows that of Γ .

An illustrating example that better explains the above notations and data transmissions to a fusion center is depicted in Fig. 5.1.

A vectorized measurement model is then given by

$$\mathbf{r} = \mathbf{h}(\boldsymbol{\theta}_p) + \mathbf{v} \quad (5.3)$$

with $r_{i,j}$'s stacked in \mathbf{r} , $d(\mathbf{p}_i, \mathbf{p}_j)$'s in $\mathbf{h}(\boldsymbol{\theta}_p)$, and $v_{i,j}$'s in \mathbf{v} . The aim of this work is to jointly estimate the unknown parameters, including the agent positions $\boldsymbol{\theta}_p$ and Gaussian mixture model parameters $\boldsymbol{\theta}_e$, given the probabilistic measurement model in (5.3), a set of noisy distance measurements \mathbf{r} , and a few known anchor positions.

5.2 Competing Algorithms Revisited

In this section, we briefly review two classes of cooperative localization algorithms that have been frequently considered for wireless sensor network problems in the past. They are:

1. Least-squares (LS) estimation based algorithms
2. Nonparametric belief propagation (NBP) based algorithms

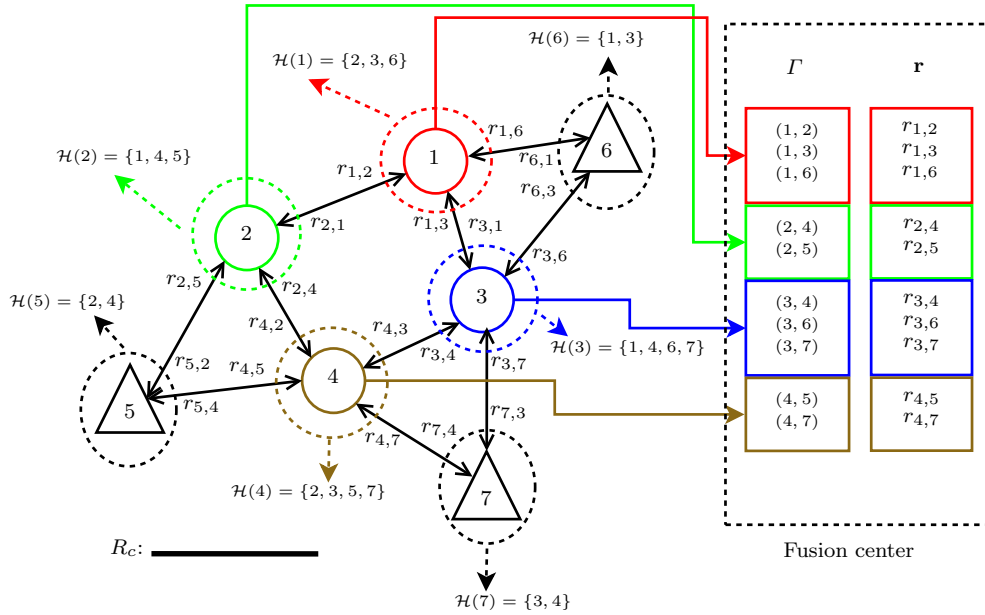


Figure 5.1. An illustrating example of a connected wireless sensor network in a 2-D space. In this example, there are $N = 7$ sensors in total, among which $N_u = 4$ agents, marked by \circ 's, and the rest are anchors, marked by \triangle 's. All measurements that will be routed to a fusion center are stacked in \mathbf{r} which has the same data structure as the set of all feasible sensor pairs, Γ .

5.2.1 LS Estimation Based Algorithms

To the best of our knowledge, the first centralized LS estimation based algorithm for cooperative localization was given in [19]. The resulting LS estimator is the solution of

$$\hat{\boldsymbol{\theta}}_p^{\text{LS}} = \arg \min_{\boldsymbol{\theta}_p} \sum_{(i,j) \in \Gamma} (r_{i,j} - d(\mathbf{p}_i, \mathbf{p}_j))^2, \quad (5.4)$$

which is found by a conjugate gradient algorithm in [19]. To avoid the use of a fusion center, a distributed LS algorithm was proposed in [14, Algorithm 1]. The advantages of these two LS estimators in common lie in the independence of the prior knowledge about the measurement error distribution and NLOS identification. An LS estimator becomes the ML estimator when Gaussian measurement error distribution is assumed. In mixed LOS/NLOS localization environments, the existence of the NLOS measurements (or outliers) ruins the adequacy of the Gaussian measurement error model and brings largely degraded performance as a consequence. To compensate for the NLOS (or outlier) effects, a robust algorithm was developed in [120], where the least-squares function was replaced by a robust Huber function in the minimization problem. As trade-off, this algorithm needs to tune several parameters of the Huber function. Enhanced performance can be observed when only a small fraction of outliers exist in the

observed data.

5.2.2 NBP Based Algorithms

The second class of localization algorithms, which are Bayesian in nature, take advantages of the celebrated belief propagation (BP) algorithm, also known as sum-product-algorithm (SPA) in the literature. Simply speaking, BP is a powerful algorithm for computing the marginal posterior pdf of each agent position (i.e., local belief message) in graph models. To the best of our knowledge, the first work suitable for real applications is the nonparametric BP (NBP) algorithm [81], which introduced particle-based approximation of the local belief messages and internal messages (will be defined later). The main advantages of the NBP based self-localization algorithm [81] are:

- It allows for distributed implementation.
- It is not restricted to Gaussian measurement models.
- It produces not only an estimate of sensor positions but also localization uncertainty.

In what follows, we briefly summarize the basics of the classical NBP algorithm, which serves as the cornerstone of several new variations. In contrast to the non-Bayesian algorithms, the unknown positions are assumed to be random with certain prior probabilities. According to the signal model in Section 5.1, agent i obtains a noisy distance measurement (e.g., TOA measurement) from a neighboring sensor j (can be either an agent or an anchor),

$$r_{i,j} = d(\mathbf{p}_i, \mathbf{p}_j) + v_{i,j} \quad (5.5)$$

where

$$v_{i,j} \sim p_V(r_{i,j} - d(\mathbf{p}_i, \mathbf{p}_j)). \quad (5.6)$$

Let further $p(\mathbf{p}_i)$ be the prior distribution of sensor i 's true position \mathbf{p}_i . The prior distribution is assumed to be uniform in the deployment area for an agent if no extra information about the position is available. The prior distribution is a Dirac delta function for an anchor. The joint distribution is given by

$$\begin{aligned} p(\mathbf{p}_1, \dots, \mathbf{p}_N, \mathbf{r}) &= p(\mathbf{r}|\mathbf{p}_1, \dots, \mathbf{p}_N)p(\mathbf{p}_1, \dots, \mathbf{p}_N) \\ &= \prod_{(i,j) \in \Gamma} p(r_{i,j}|\mathbf{p}_i, \mathbf{p}_j) \prod_{i=1}^N p(\mathbf{p}_i). \end{aligned} \quad (5.7)$$

This result is due to the following simplifying assumptions [121]:

1. The joint prior distribution of the positions of all nodes is equal to the multiplications of each local prior, namely,

$$p(\mathbf{p}_1, \dots, \mathbf{p}_N) = \prod_{i=1}^N p(\mathbf{p}_i). \quad (5.8)$$

2. Given the positions of all nodes, the measurements are mutually independent, namely,

$$p(\mathbf{r}|\mathbf{p}_1, \dots, \mathbf{p}_N) = \prod_{(i,j) \in \Gamma} p(r_{i,j}|\mathbf{p}_1, \dots, \mathbf{p}_N). \quad (5.9)$$

3. Given the positions of nodes i and j , the measurement $r_{i,j}$ is independent of the positions of any other nodes, namely,

$$p(r_{i,j}|\mathbf{p}_1, \dots, \mathbf{p}_N) = p(r_{i,j}|\mathbf{p}_i, \mathbf{p}_j). \quad (5.10)$$

The joint posterior pdf can be easily expressed by means of Bayes' rule as

$$\begin{aligned} p(\mathbf{p}_1, \dots, \mathbf{p}_N|\mathbf{r}) &\propto p(\mathbf{r}|\mathbf{p}_1, \dots, \mathbf{p}_N)p(\mathbf{p}_1, \dots, \mathbf{p}_N) \\ &= \prod_i \psi_i(\mathbf{p}_i) \prod_{(i,j) \in \Gamma} \psi_{ij}(\mathbf{p}_i, \mathbf{p}_j) \end{aligned} \quad (5.11)$$

where

$$\psi_i(\mathbf{p}_i) = p(\mathbf{p}_i), \quad (5.12)$$

$$\psi_{ij}(\mathbf{p}_i, \mathbf{p}_j) = p(r_{i,j}|\mathbf{p}_i, \mathbf{p}_j) = p_V(r_{i,j} - \|\mathbf{p}_i - \mathbf{p}_j\|). \quad (5.13)$$

The task is to compute/approximate the marginal posterior pdf $p(\mathbf{p}_i|\mathbf{r})$ for $\forall i \in \mathcal{N}_a$. For the case that cycles exist in the factor graph [87], which is commonly seen in localization problems, $p(\mathbf{p}_i|\mathbf{r})$ has to be approximated recursively. More precisely, on the η th iteration,

$$p(\mathbf{p}_i|\mathbf{r}) \approx B_i^{(\eta)}(\mathbf{p}_i) \propto \psi_i(\mathbf{p}_i) \prod_{j \in \mathcal{H}(i)} m_{ji}^{(\eta)}(\mathbf{p}_i) \quad (5.14)$$

where the internal message sent from sensor j to sensor i is computed according to

$$m_{ji}^{(\eta)}(\mathbf{p}_i) \propto \int_{\mathbf{p}_j} \psi_{ij}(\mathbf{p}_i, \mathbf{p}_j) \frac{B_j^{(\eta-1)}(\mathbf{p}_j)}{m_{ij}^{(\eta-1)}(\mathbf{p}_j)} d\mathbf{p}_j. \quad (5.15)$$

An iterative particle-based procedure can be adopted to compute the local belief messages. However, the convergence stability is not guaranteed. On each iteration (say the η th), both the local belief messages and internal messages (distributions in essence) are represented by a set of weighted particles, namely $\{w_i^{(\eta,\iota)}, \mathbf{x}_i^{(\eta,\iota)}\}$, $\iota = 1, 2, \dots, R_{bel}$ for $B_i^{(\eta)}(\mathbf{p}_i)$ and $\{w_{ij}^{(\eta,\iota')}, \mathbf{x}_{ij}^{(\eta,\iota')}\}$, $\iota' = 1, 2, \dots, R_{int}$ for $m_{ij}^{(\eta)}(\mathbf{p}_j)$. More details about the

generation of these particles can be found in [121]. In practice $R_{int} \gg R_{bel}$, since $\{w_{ij}^{(\eta, \ell)}, \mathbf{x}_{ij}^{(\eta, \ell)}\}$ are generated locally at sensor j and no extra data packets need to be transmitted from sensor i to sensor j . The communication overhead solely relies on the size of R_{bel} , but the size of R_{int} has great impact on the computational complexity of the NBP algorithm. For good localization performance, both R_{int} and R_{bel} should be set large (e.g., 500-1000 particles). The sum-product-algorithm over wireless networks (SPAWN) algorithm is a generalized version of the above NBP algorithm in that $r_{i,j} \neq r_{j,i}$ and both of them are used in the reference task. In fact, we only need to modify (5.15) to

$$m_{ji}^{(\eta)}(\mathbf{p}_i) \propto \int_{\mathbf{p}_j} \psi_{ij}(\mathbf{p}_i, \mathbf{p}_j) B_j^{(\eta-1)}(\mathbf{p}_j) m_{ij}^{(\eta-1)}(\mathbf{p}_j) d\mathbf{p}_j. \quad (5.16)$$

Many new variations have been built recently upon the NBP and SPAWN algorithms. Most of the efforts have been made to reduce the computational complexity and communication overhead. For instance [83, 84, 89] propose to represent both the local belief messages and internal messages parametrically. Alternatively, [85] adopted transmit-and receive censoring to reduce the communication overhead as well as the computational complexity. The idea is to avoid unnecessary broadcast if a sensor is not confident about the computed belief on its own position in the transmit censoring and to pick up a minority of the most informative internal messages sent to it in the receive censoring. As compared to the listed parametric algorithms, both the computational complexity and communication overhead can be only modestly reduced.

In contrast to the first class of algorithms, the class of NBP based algorithms need precise knowledge about the measurement error distribution prior to performing network localization.

5.3 Joint ML Estimation

In Section 5.3.1, we briefly comment on the difficulties with the ML implementation. To approximate the ML estimator with less hurdles, we develop a series of expectation-conditional maximization (ECM) algorithms (cf. Section 5.3.2 for centralized implementations and Section 5.3.3 for distributed implementations). Finally, Section 5.3.4 integrates the proposed algorithms into a snapshot-based solution for locating dynamic sensor networks.

5.3.1 ML Estimation

We start with the conventional implementation (centralized in nature) of the maximum likelihood estimation. Similar to [19], the log-likelihood function of the considered joint estimation problem is expressed as follows:

$$\begin{aligned}\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r}) &= \sum_{i=1}^{N_u} \sum_{\substack{j \in \mathcal{H}(i), \\ j > i}} \ln p(r_{i,j}; \boldsymbol{\theta}) = \sum_{(i,j) \in \Gamma} \ln p(r_{i,j}; \boldsymbol{\theta}) \\ &= \sum_{(i,j) \in \Gamma} \ln \left(\sum_{l=1}^C \alpha_l \mathcal{N}(r_{i,j} - d(\mathbf{p}_i, \mathbf{p}_j); \mu_l, \sigma_l^2) \right).\end{aligned}\quad (5.17)$$

The centralized ML estimator is obtained through solving:

$$\begin{aligned}\underset{\boldsymbol{\theta}}{\text{maximize}} \quad & \mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r}) \\ \text{subject to} \quad & 0 \leq \alpha_1, \dots, \alpha_C \leq 1, \sum_{l=1}^C \alpha_l = 1, \\ & \sigma_l^2 > 0, l = 1, 2, \dots, C.\end{aligned}\quad (5.18)$$

This optimization problem is cumbersome for two reasons. On the one hand, the cost function contains “the logarithm of the sum”, which hinders the analytical evaluation of the parameters. On the other hand, when the number of agents, N_u , is large, the existing numerical methods, e.g., Newton-type methods, would become less stable [95].

5.3.2 Centralized ECM Algorithms

Instead of solving (5.18) directly, we approximate the ML estimator using the ECM criterion. As it was introduced in Chapter 2, the idea is to replace the complicated M-step of the conventional EM algorithm with a set of computationally simpler conditional maximization (CM) steps in an attempt to split a difficult maximization problem into many easier ones. We first focus on centralized ECM algorithms for cooperative localization, which lays a foundation for developing different distributed algorithms. A centralized ECM algorithm works with a complete-data set $\{\mathbf{y}, \mathbf{r}\}$, in which \mathbf{y} is a column vector enclosing $|\Gamma|$ latent variables $y_{i,j}$ indicating that which mixture component has given rise to $r_{i,j}$. The work-flow of a centralized ECM algorithm on the $(\eta + 1)$ th iteration is as follows:

Expectation: We take conditional expectation of the complete-data log-likelihood $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r}) = \ln(p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}))$ in terms of \mathbf{y} , given \mathbf{r} and $\boldsymbol{\theta}^{(\eta)}$. Following the derivations

in Appendix 5.8.1, we finally arrive at

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) = \underbrace{\sum_{l=1}^C \sum_{\forall (i,j) \in \Gamma} \ln(\alpha_l) P_{i,j,l}^{(\eta)}}_{Q_0^{(\eta)}(\alpha_1, \alpha_2, \dots, \alpha_C)} + \underbrace{\sum_{l=1}^C \sum_{\forall (i,j) \in \Gamma} - \left[\frac{(r_{i,j} - d_{i,j} - \mu_l)^2}{2\sigma_l^2} + \frac{1}{2} \ln(\sigma_l^2) \right] P_{i,j,l}^{(\eta)}}_{Q_l^{(\eta)}(\mu_l, \sigma_l^2, \boldsymbol{\theta}_p)} \quad (5.19)$$

with $P_{i,j,l}^{(\eta)}$ being a short-hand notation of the conditional probability $\Pr\{y_{i,j} = l | r_{i,j}; \boldsymbol{\theta}^{(\eta)}\}$. In light of Bayes' rule, we immediately have

$$P_{i,j,l}^{(\eta)} \propto \alpha_l^{(\eta)} \mathcal{N}(r_{i,j} - d(\mathbf{p}_i^{(\eta)}, \mathbf{p}_j^{(\eta)}); \mu_l^{(\eta)}, \sigma_l^{2,(\eta)}). \quad (5.20)$$

Note that $\mathbf{p}_j^{(\eta)} = \mathbf{p}_j$ if sensor j is an anchor.

Conditional Maximization: First, we need to find a proper partition of the unknown parameters, i.e.,

$$\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \dots, \boldsymbol{\vartheta}_S^T]^T \quad (5.21)$$

where $\boldsymbol{\vartheta}_s^T$ is a sub-vector of $\boldsymbol{\theta}$. Furthermore, we let

$$\mathcal{G} = \{g_s(\boldsymbol{\theta}) : s = 1, 2, \dots, S\} \quad (5.22)$$

be a set of S preselected vector functions of $\boldsymbol{\theta}$ defined by

$$g_s(\boldsymbol{\theta}) = [\boldsymbol{\vartheta}_1^T, \dots, \boldsymbol{\vartheta}_{s-1}^T, \boldsymbol{\vartheta}_{s+1}^T, \dots, \boldsymbol{\vartheta}_S^T]^T, \quad (5.23)$$

meaning that $g_s(\boldsymbol{\theta})$ is a vector that contains all the sub-vectors of $\boldsymbol{\theta}$ except $\boldsymbol{\vartheta}_s$. With the above partition of $\boldsymbol{\theta}$, the s th CM step of the $(\eta + 1)$ th ECM iteration solves $\boldsymbol{\theta}^{(\eta+s/S)}$ from the following optimization problem:

$$\begin{aligned} & \underset{\boldsymbol{\theta}}{\text{maximize}} && Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) \\ & \text{subject to} && g_s(\boldsymbol{\theta}) = g_s(\boldsymbol{\theta}^{(\eta+(s-1)/S)}). \end{aligned} \quad (5.24)$$

For clarity, a general routine is given in Algorithm 5.1.

Different partitions of $\boldsymbol{\theta}$ lead to different ECM algorithms. To elaborate on this, two examples are shown in the sequel with the same partition of the mixture model parameters $\boldsymbol{\vartheta}_1 = [\alpha_1, \alpha_2, \dots, \alpha_C]^T$, $\boldsymbol{\vartheta}_{2l} = \mu_l$, $\boldsymbol{\vartheta}_{2l+1} = \sigma_l^2$, $l = 1, 2, \dots, C$, but with different partitions of the position parameters as follows:

- **Example 1:** $\boldsymbol{\vartheta}_{2C+1+i} = \mathbf{p}_i$, $i = 1, 2, \dots, N_u$ with $S = N_u + 2C + 1$.
- **Example 2:** $\boldsymbol{\vartheta}_{2C+2i} = x_i$ and $\boldsymbol{\vartheta}_{2C+2i+1} = y_i$, $i = 1, 2, \dots, N_u$ with $S = 2N_u + 2C + 1$.

Algorithm 5.1 Centralized ECM Algorithm (General Routine)

Step1—Initialization: Set a convergence tolerance Δ_c and the maximum number of iterations to N_{itr}^c . Choose an initial guess $\boldsymbol{\theta}^{(\eta=0)}$.

Step2—Expectation-Conditional Maximization:

On the $(\eta + 1)$ th iteration ($\eta \in \mathbb{Z}, \eta \geq 0$), do:

1. Perform conditional expectation and obtain $Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$, cf.(5.19).
2. Find a proper partition of the unknown parameters, $\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \dots, \boldsymbol{\vartheta}_S^T]^T$. See for instance the two examples given above.
3. Find $\boldsymbol{\theta}^{(\eta+s/S)}$ that solves (5.24) sequentially for $s = 1, 2, \dots, S$.

Step3—Convergence Check:

If $\mathcal{L}_I(\boldsymbol{\theta}^{(\eta+1)}; \mathbf{r}) - \mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r}) \leq \Delta_c$ or the maximum number of iterations N_{itr}^c has been reached, then terminate the whole algorithm and obtain $\hat{\boldsymbol{\theta}}^{\text{CECM}} = \boldsymbol{\theta}^{(\eta+1)}$; otherwise reset $\eta \leftarrow \eta + 1$ and return to the ECM stage.

With the above partition of the mixture model parameters, it can be easily shown by following the same methodology as demonstrated in Section 4.2.3 that the global optimal solutions of $\alpha_l^{(\eta+1)}$, $\mu_l^{(\eta+1)}$, $\sigma_l^{2,(\eta+1)}$, $l = 1, 2, \dots, C$, are solved respectively from

$$\frac{\partial}{\partial \alpha_l} \left[Q_0^{(\eta)}(\alpha_1, \dots, \alpha_C) + \lambda \left(\sum_{l=1}^C \alpha_l - 1 \right) \right] = 0, \quad (5.25)$$

$$\frac{\partial}{\partial \mu_l} \left[Q_l^{(\eta)}(\mu_l, \sigma_l^{2,(\eta)}, \boldsymbol{\theta}_p^{(\eta)}) \right] = 0, \quad (5.26)$$

$$\frac{\partial}{\partial \sigma_l^2} \left[Q_l^{(\eta)}(\mu_l^{(\eta+1)}, \sigma_l^2, \boldsymbol{\theta}_p^{(\eta)}) \right] = 0, \quad (5.27)$$

and the closed form solutions are given by

$$\alpha_l^{(\eta+1)} = \frac{1}{|\Gamma|} \sum_{(i,j) \in \Gamma} P_{i,j,l}^{(\eta)}, \quad (5.28)$$

$$\mu_l^{(\eta+1)} = \frac{\sum_{(i,j) \in \Gamma} (r_{i,j} - d(\mathbf{p}_i^{(\eta)}, \mathbf{p}_j^{(\eta)})) P_{i,j,l}^{(\eta)}}{|\Gamma| \alpha_l^{(\eta+1)}}, \quad (5.29)$$

$$\sigma_l^{2,(\eta+1)} = \frac{\sum_{(i,j) \in \Gamma} (r_{i,j} - d(\mathbf{p}_i^{(\eta)}, \mathbf{p}_j^{(\eta)}))^2 P_{i,j,l}^{(\eta)}}{|\Gamma| \alpha_l^{(\eta+1)}} - (\mu_l^{(\eta+1)})^2. \quad (5.30)$$

It is easy to verify, as was done in Appendix 4.8.2, that for $s = 1, 2, \dots, 2C + 1$,

$$Q(\boldsymbol{\theta}^{(\eta+s/S)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) \quad (5.31)$$

for any $\boldsymbol{\theta} \in \Theta_s(\boldsymbol{\theta}^{(\eta+(s-1)/S)}) \triangleq \{\boldsymbol{\theta} \in \Theta : g_s(\boldsymbol{\theta}) = g_s(\boldsymbol{\theta}^{(\eta+(s-1)/S)})\}$. In other words, $\boldsymbol{\theta}^{(\eta+s/S)}$ is the global maximizer in the given subspace of Θ .

The position are updated only numerically. In the first example, it can be shown that the positions are updated by

$$\mathbf{p}_i^{(\eta+1)} = \arg \min_{\mathbf{p}_i} \sum_{j \in \mathcal{H}(i)} \sum_{l=1}^C \frac{(r_{i,j} - d(\mathbf{p}_i, \mathbf{p}_j^{(\tilde{\eta})}) - \mu_l^{(\eta+1)})^2 P_{i,j,l}^{(\eta)}}{\sigma_l^{2,(\eta+1)}}, \quad (5.32)$$

for $i = 1, 2, \dots, N_u$, as a consequence of the reciprocal assumption $r_{i,j} = r_{j,i}$. In (5.32), $\mathbf{p}_j^{(\tilde{\eta})} = \mathbf{p}_j^{(\eta+1)}$ if sensor j is an agent with its position updated prior to sensor i , or $\mathbf{p}_j^{(\tilde{\eta})} = \mathbf{p}_j^{(\eta)}$ otherwise; or $\mathbf{p}_j^{(\tilde{\eta})} = \mathbf{p}_j$ if sensor j is an anchor. We adopt the two-dimensional (2-D) BFGS-QN method to solve $\mathbf{p}_i^{(\eta+1)}$, $i = 1, 2, \dots, N_u$. It is ensured that

$$Q(\boldsymbol{\theta}^{(\eta+(2C+1+i)/S)}; \boldsymbol{\theta}^{(\eta)}) \geq Q(\boldsymbol{\theta}^{(\eta+(2C+i)/S)}; \boldsymbol{\theta}^{(\eta)}), \quad (5.33)$$

because the BFGS-QN method guarantees downhill progress towards the local minimum in each Newton step. We stress that the positions need not to be updated in the order of the sensor indicies.

In contrast to the $2N_u$ -dimensional BFGS-QN search used in the EM algorithm [65], which is in fact also an ECM algorithm with $\boldsymbol{\vartheta}_{2C+2} = \boldsymbol{\theta}_p = [\mathbf{p}_1^T, \dots, \mathbf{p}_{N_u}^T]^T$ and $S = 2C+2$ in the partition of $\boldsymbol{\theta}$, the ECM algorithm derived for the first example requires N_u trials 2-D BFGS-QN search. We note that [65] is more suitable to use for small or moderate N_u , because intuitively it should converge much faster. A good example has been given in Chapter 4, where only one ($N_u = 1$) mobile station is to be located in a non-cooperative framework. The first ECM algorithm is, however, favorable for a large N_u , since it is numerically more stable [95].

In the second example, the position update for each agent can be found by applying one-dimensional (1-D) grid search (GS) respectively to

$$\arg \min_{x_i} \sum_{j \in \mathcal{H}(i)} \sum_{l=1}^C \frac{(r_{i,j} - d([x_i, y_i^{(\eta)}]^T, \mathbf{p}_j^{(\tilde{\eta})}) - \mu_l^{(\eta+1)})^2 P_{i,j,l}^{(\eta)}}{\sigma_l^{2,(\eta+1)}} \quad (5.34)$$

and

$$\arg \min_{y_i} \sum_{j \in \mathcal{H}(i)} \sum_{l=1}^C \frac{(r_{i,j} - d([x_i^{(\eta+1)}, y_i]^T, \mathbf{p}_j^{(\tilde{\eta})}) - \mu_l^{(\eta+1)})^2 P_{i,j,l}^{(\eta)}}{\sigma_l^{2,(\eta+1)}}. \quad (5.35)$$

By carefully choosing the search interval as well as grid points, the global maximizer of (5.24) is reachable in every CM step. As a trade-off, in total $2N_u$ trials 1-D grid searches are required by each ECM iteration, which will consume more computational resources than the 2-D BFGS-QN based ECM algorithm.

Theorem 5.1. *The proposed centralized ECM algorithms are in fact generalized EM (GEM) algorithms and ensure that the sequence of incomplete-data log-likelihood values $\{\mathcal{L}_I(\boldsymbol{\theta}^{(\eta)}; \mathbf{r})\}$, when bounded above, converges monotonically over iterations to some point L^* .*

Proof. Similar to the proof given in Theorem 4.1, we have here

$$\begin{aligned} Q(\boldsymbol{\theta}^{(\eta+1)}; \boldsymbol{\theta}^{(\eta)}) &\geq Q(\boldsymbol{\theta}^{(\eta+(S-1)/S)}; \boldsymbol{\theta}^{(\eta)}) \\ &\geq Q(\boldsymbol{\theta}^{(\eta+(S-2)/S)}; \boldsymbol{\theta}^{(\eta)}) \\ &\vdots \\ &\geq Q(\boldsymbol{\theta}^{(\eta)}; \boldsymbol{\theta}^{(\eta)}). \end{aligned} \quad (5.36)$$

Hence, the proposed ECM algorithms are essentially GEM algorithms. When $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ is bounded above, which holds under the assumption that $\sigma_l^2 > 0$, $l = 1, 2, \dots, C$, the proposed centralized ECM algorithms converge monotonically over iterations to some value L^* of the incomplete-data log-likelihood function $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$. \square

Theorem 5.2. *When the position updates found by the local grid search (5.34) and (5.35) are global maximizers, L^* is ensured to be a stationary point.*

Proof. For L^* to be a stationary point, however, we need to prove additionally, according to Section 2.2.2, that: (1) $g_s(\boldsymbol{\theta})$ is differentiable; (2) the corresponding gradient $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta})$ is of full rank at $\boldsymbol{\theta}^{(\eta)} \in \Theta$, for all η ; and (3) the “space filling” condition holds as

$$\bigcap_{s=1}^S G_s(\boldsymbol{\theta}^{(\eta)}) = \{\mathbf{0}\}, \quad \text{for all } \eta, \quad (5.37)$$

where $G_s(\boldsymbol{\theta})$ is the column space of the matrix $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta})$.

In the 1D GS based centralized ECM algorithm, we have $\dim(\boldsymbol{\theta}) = 3C + 2N_u$ and $S = 2N_u + 2C + 1$. It is easy to show that

$$\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta}^{(\eta)}) = \begin{cases} [\mathbf{e}_{C+1}, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})}], & s = 1 \\ [\mathbf{e}_1, \dots, \mathbf{e}_{C+s-2}, \mathbf{e}_{C+s}, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})}], & s = 2, \dots, S-1 \\ [\mathbf{e}_1, \dots, \mathbf{e}_{3C+2N_u-1}], & s = S \end{cases} \quad (5.38)$$

are all differentiable and irrespective of $\boldsymbol{\theta}^{(\eta)}$ as

$$\mathbf{e}_j = [\underbrace{0, \dots, 0}_{j-1 \text{ copies}}, 1, \underbrace{0, \dots, 0}_{\dim(\boldsymbol{\theta})-j \text{ copies}}]^T, \quad \forall j \in \{1, 2, \dots, \dim(\boldsymbol{\theta})\}. \quad (5.39)$$

It is clear that $\nabla_{\boldsymbol{\theta}} g_1(\boldsymbol{\theta}^{(\eta)})$ is of dimension $\dim(\boldsymbol{\theta}) \times (\dim(\boldsymbol{\theta}) - C)$; and $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta}^{(\eta)})$, $s = 2, \dots, S$, are all of dimension $\dim(\boldsymbol{\theta}) \times (\dim(\boldsymbol{\theta}) - 1)$. The vector basis \mathbf{e}_j is of dimension $\dim(\boldsymbol{\theta}) \times 1$. Since \mathbf{e}_j and $\mathbf{e}_{j'}$ are mutually orthogonal for $j \neq j'$, $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta}^{(\eta)})$ has a full column rank. So far, the first two conditions have been proven. In the sequel, we omit the iteration index η for brevity.

The proof of the third condition starts with the definition of the column space, that is, $G_s(\boldsymbol{\theta})$ is a linear combination of the columns of the matrix $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta})$, i.e.,

$$G_s(\boldsymbol{\theta}) = \begin{cases} \sum_{j=\{C+1, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j & s = 1 \\ \sum_{j=\{1, \dots, S\} \setminus \{C+s-1\}} c_j \mathbf{e}_j & s = 2, \dots, S \end{cases} \quad (5.40)$$

where c_j is a real scalar coefficient. Since $G_s(\boldsymbol{\theta})$ is a subspace of $\mathbb{R}^{\dim(\boldsymbol{\theta})}$, (5.37) can be reformulated as

$$\bigcap_{s=1}^S G_s(\boldsymbol{\theta}) = \bigcap_{s=0}^S G_s(\boldsymbol{\theta}) = G_S(\boldsymbol{\theta}) \cap G_{S-1}(\boldsymbol{\theta}) \cap \dots \cap G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta}) \quad (5.41)$$

where $G_0(\boldsymbol{\theta})$ is the whole space of $\mathbb{R}^{\dim(\boldsymbol{\theta})}$, spanned by $\sum_{j=\{1, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j$. The right-hand-side of the second equation in (5.41) can be performed sequentially in the order $s = 1, 2, \dots, S$, more precisely,

$$\tilde{G}_s(\boldsymbol{\theta}) = G_s(\boldsymbol{\theta}) \cap \tilde{G}_{s-1}(\boldsymbol{\theta}) \quad (5.42)$$

where

$$\tilde{G}_{s-1}(\boldsymbol{\theta}) = G_{s-1}(\boldsymbol{\theta}) \cap G_{s-2}(\boldsymbol{\theta}) \cap \dots \cap G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta}) \quad (5.43)$$

for $s > 1$ while $\tilde{G}_{s-1}(\boldsymbol{\theta}) = G_0(\boldsymbol{\theta})$ for $s = 1$. Starting from $s = 1$, we obtain, owing to the dimension formula [114], that

$$\dim(\tilde{G}_1(\boldsymbol{\theta})) = \dim(G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta})) = \dim(G_1(\boldsymbol{\theta})) + \dim(G_0(\boldsymbol{\theta})) - \dim(G_1(\boldsymbol{\theta}) + G_0(\boldsymbol{\theta})). \quad (5.44)$$

Since $G_1(\boldsymbol{\theta})$ and $G_0(\boldsymbol{\theta})$ are both spanned by orthogonal basis vectors, we have $\dim(G_1(\boldsymbol{\theta})) = \dim(\boldsymbol{\theta}) - C$ and $G_0(\boldsymbol{\theta}) = \dim(\boldsymbol{\theta})$. The dimension of the sum of column spaces $\dim(G_1(\boldsymbol{\theta}) + G_0(\boldsymbol{\theta}))$ is equal to the rank of the matrix $[\mathbf{e}_{C+1}, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})} | \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_{\dim(\boldsymbol{\theta})}]$, which turns out to be $\dim(\boldsymbol{\theta})$. As a consequence, we obtain $\dim(\tilde{G}_1(\boldsymbol{\theta})) = \dim(\boldsymbol{\theta}) - C$. It is easy to arrive at the conclusion that the basis vectors that span $\tilde{G}_1(\boldsymbol{\theta})$ are just the column vectors that $G_1(\boldsymbol{\theta})$ and $G_0(\boldsymbol{\theta})$ have in common, namely,

$$\tilde{G}_1(\boldsymbol{\theta}) = G_1(\boldsymbol{\theta}) \cap G_0(\boldsymbol{\theta}) = \sum_{j \in \{C+1, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j. \quad (5.45)$$

Similarly for $s > 1$,

$$\dim(\tilde{G}_s(\boldsymbol{\theta})) = \dim(G_s(\boldsymbol{\theta}) \cap \tilde{G}_{s-1}(\boldsymbol{\theta})) \quad (5.46)$$

$$= (\dim(\boldsymbol{\theta}) - 1) + (\dim(\boldsymbol{\theta}) - C - (s - 2)) - \dim(\boldsymbol{\theta}) \quad (5.47)$$

$$= \dim(\boldsymbol{\theta}) - C - (s - 1). \quad (5.48)$$

and

$$\tilde{G}_s(\boldsymbol{\theta}) = G_s(\boldsymbol{\theta}) \cap \tilde{G}_{s-1}(\boldsymbol{\theta}) = \sum_{j \in \{C+s, \dots, \dim(\boldsymbol{\theta})\}} c_j \mathbf{e}_j. \quad (5.49)$$

Note that the result in (5.48) is due to the fact that the dimension of $G_s(\boldsymbol{\theta})$ is always $\dim(\boldsymbol{\theta}) - 1$ and the dimension of the sum of column spaces $\dim(G_s(\boldsymbol{\theta}) + \tilde{G}_{s-1}(\boldsymbol{\theta}))$ is always $\dim(\boldsymbol{\theta})$. Finally, after the S th intersection, we have

$$\tilde{G}_S(\boldsymbol{\theta}) = G_S(\boldsymbol{\theta}) \cap \tilde{G}_{S-1}(\boldsymbol{\theta}) = \{\mathbf{0}\}. \quad (5.50)$$

□

Theorem 5.3. *When the position updates found by the local BFGS-QN search (5.32) are global maximizers, L^* is ensured to be a stationary point.*

Proof. The proof is very similar to the one provided in **Theorem 5.2**. □

Remark 5.1. *The proposed centralized ECM algorithms may converge to different stationary points if $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ is multi-modal. This is due to the different partitions of the position parameters, which lead to different mappings $\boldsymbol{\theta}^{(n+1)} = \mathcal{M}(\boldsymbol{\theta}^{(n)})$ in the conditional maximization. For the special case that $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ is unimodal, they will reach the same global maximum.*

5.3.3 Distributed ECM Algorithms

The aim of this section is to approximate the centralized ECM algorithms in a distributed manner. Essentially, the above centralized ECM algorithm can be understood as an iterative process consisting of the following two steps on each iteration: (1) updating the mixture model parameters with the position parameters held fixed; and (2) updating the position parameters with the mixture model parameters held fixed. In the sequel, we aim to decentralize these two steps.

The first step of the proposed centralized ECM algorithms makes no distinction with the conventional EM algorithm for Gaussian mixture learning if the position estimates

were replaced with their true values. After simple manipulations, the results in (5.28)—(5.30) can be re-expressed as:

$$\alpha_l^{(\eta+1)} = \frac{N_u}{|F|} \frac{1}{N_u} \sum_{i=1}^{N_u} w_{i,l} \propto \frac{1}{N_u} \sum_{i=1}^{N_u} w_{i,l}, \quad (5.51)$$

$$\mu_l^{(\eta+1)} = \frac{\frac{1}{N_u} \sum_{i=1}^{N_u} a_{i,l}}{\frac{1}{N_u} \sum_{i=1}^{N_u} w_{i,l}}, \quad (5.52)$$

$$\sigma_l^{2,(\eta+1)} = \frac{\frac{1}{N_u} \sum_{i=1}^{N_u} b_{i,l}}{\frac{1}{N_u} \sum_{i=1}^{N_u} w_{i,l}} - (\mu_l^{(\eta+1)})^2, \quad (5.53)$$

where

$$w_{i,l} = \sum_{\substack{j \in \mathcal{H}(i), \\ j > i}} P_{i,j,l}^{(\eta)}, \quad (5.54)$$

$$a_{i,l} = \sum_{\substack{j \in \mathcal{H}(i), \\ j > i}} P_{i,j,l}^{(\eta)} (r_{i,j} - d(\mathbf{p}_i^{(\eta)}, \mathbf{p}_j^{(\eta)})), \quad (5.55)$$

$$b_{i,l} = \sum_{\substack{j \in \mathcal{H}(i), \\ j > i}} P_{i,j,l}^{(\eta)} (r_{i,j} - d(\mathbf{p}_i^{(\eta)}, \mathbf{p}_j^{(\eta)}))^2. \quad (5.56)$$

It is clear from (5.51)—(5.53) that averaging is the main operation in common for updating the mixture model parameters. This allows us to use the class of average consensus algorithms [122], which does not rely on any prescribed route, like the Hamiltonian path adopted in [123, 124] or the tree structure adopted in [82] and thus is more robust against sensor malfunction and link failure. For simplicity we assume in the sequel that the sensors are time synchronized, and the synchronous average consensus algorithm [125] is modified in Algorithm 5.2 to work for the considered problem. On the one hand, this modified algorithm is easier to implement based on the existing protocol [126]. On the other hand, the synchronous nature of this algorithm facilitates the subsequent analyses on the computational complexity and energy consumption. However, we note that Algorithm 5.2 can be easily recast in an asynchronous manner—well known as the “pair-wise gossip” algorithm in the literature, see for instance [127]. Besides, it is also convenient to replace Algorithm 5.2 with other consensus based methods [128, 129] or diffusion based methods [130, 131]. A comparison of them either in quality or in quantity is, however, beyond the scope of this thesis. Lastly, we note that in the corresponding distributed algorithms $P_{i,j,l}^{(\eta)}$ in (5.54)—(5.56)

has to be replaced with

$$\tilde{P}_{i,j,l}^{(\eta)} \propto \alpha_{i,l}^{(\eta)} \mathcal{N}(r_{i,j} - d(\mathbf{p}_i^{(\eta)}, \mathbf{p}_j^{(\eta)}); \mu_{i,l}^{(\eta)}, \sigma_{i,l}^{2,(\eta)}), \quad (5.57)$$

which is computed locally at each agent i .

Algorithm 5.2 Synchronous Average Consensus Algorithm (for Agent i at the $(\eta+1)$ th ECM Iteration)

1. Repeat 1.a. and 1.b. for N_{itr}^g iterations:

1.a. Choose a neighboring agent j uniformly at random from $\mathcal{H}(i)$. Send $\{w_{i,l}^{(c)}, a_{i,l}^{(c)}, b_{i,l}^{(c)}\}$ to and receive $\{w_{j,l}^{(c)}, a_{j,l}^{(c)}, b_{j,l}^{(c)}\}$ from agent j .

1.b. Agents i and j update

$$\chi_{i,l}^{(n)} = \chi_{j,l}^{(n)} = \frac{\chi_{i,l}^{(c)} + \chi_{j,l}^{(c)}}{2}, \quad (5.58)$$

and re-set $\chi_{i,l}^{(c)} = \chi_{i,l}^{(n)}$, where $\chi \in \{w, a, b\}$.

Note that $\{w_{j,l}^{(c)}, a_{j,l}^{(c)}, b_{j,l}^{(c)}\}$ is initialized by $\{w_{j,l}, a_{j,l}, b_{j,l}\}$ computed by (5.54)—(5.56) with $P_{i,j,l}^{(\eta)}$ replaced by $\tilde{P}_{i,j,l}^{(\eta)}$.

2. Update the mixture model parameters by

$$\alpha_{i,l}^{(\eta+1)} \propto w_{i,l}^{(c)}, \quad \mu_{i,l}^{(\eta+1)} = \frac{a_{i,l}^{(c)}}{w_{i,l}^{(c)}}, \quad \sigma_{i,l}^{2,(\eta+1)} = \frac{b_{i,l}^{(c)}}{w_{i,l}^{(c)}} - (\mu_{i,l}^{(\eta+1)})^2. \quad (5.59)$$

Note that $\alpha_{i,l}^{(\eta+1)}$, $l = 1, 2, \dots, C$ need to be scaled such that they sum up to one.

Remark 5.2. *To alleviate the computational complexity as well as the energy consumption, N_{itr}^g is assumed to be a small and fixed number, irrespective of N_u .*

The position update procedures in the centralized ECM algorithms are readily in decentralized form because the evaluation of (5.32) or (5.34)—(5.35) only requires that each agent solves for its own position, given the local measurements, the updated mixture model parameters, and the positions of the neighboring sensors. We need only to replace the global estimates $\mu_l^{(\eta+1)}$ and $\sigma_l^{2,(\eta+1)}$ in (5.32) or (5.34)—(5.35) with the local estimates $\mu_{i,l}^{(\eta+1)}$ and $\sigma_{i,l}^{2,(\eta+1)}$, respectively, for agent i . A synchronous position update scheme, in which all agents update their positions simultaneously, can be easily obtained by substituting $\mathbf{p}_j^{(\tilde{\eta})}$ with $\mathbf{p}_j^{(\eta)}$. More precisely, the positions are updated by

$$\mathbf{p}_i^{(\eta+1)} = \arg \min_{\mathbf{p}_i} \sum_{j \in \mathcal{H}(i)} \sum_{l=1}^C \frac{(r_{i,j} - d(\mathbf{p}_i, \mathbf{p}_j^{(\eta)}) - \mu_{i,l}^{(\eta+1)})^2 \tilde{P}_{i,j,l}^{(\eta)}}{\sigma_{i,l}^{2,(\eta+1)}}, \quad (5.60)$$

in the 2-D BFGS-QN based distributed ECM algorithm, and updated by

$$\arg \min_{x_i} \sum_{j \in \mathcal{H}(i)} \sum_{l=1}^C \frac{(r_{i,j} - d([x_i, y_i^{(\eta)}]^T, \mathbf{p}_j^{(\eta)}) - \mu_{i,l}^{(\eta+1)})^2 \tilde{P}_{i,j,l}^{(\eta)}}{\sigma_{i,l}^{2,(\eta+1)}} \quad (5.61)$$

and

$$\arg \min_{y_i} \sum_{j \in \mathcal{H}(i)} \sum_{l=1}^C \frac{(r_{i,j} - d([x_i^{(\eta+1)}, y_i]^T, \mathbf{p}_j^{(\eta)}) - \mu_{i,l}^{(\eta+1)})^2 \tilde{P}_{i,j,l}^{(\eta)}}{\sigma_{i,l}^{2,(\eta+1)}}. \quad (5.62)$$

in the 1D GS based distributed ECM algorithm. Algorithm 5.3 summarizes the key steps of two distributed ECM algorithms with synchronous position updates. In the sequel, we use “2-D BFGS-QN D(C)-ECM” to denote the 2-D BFGS-QN based distributed (centralized) ECM algorithm and use “1-D GS D(C)-ECM” to denote the 1-D GS based distributed (centralized) ECM algorithm.

The position estimate $\mathbf{p}_i^{(\eta+1)}$ to be solved in the 2-D BFGS-QN D-ECM algorithms can be imagined as a weighted least-squares (WLS) solution of a conventional infrastructure based localization problem, in which a total number of $|\mathcal{H}(i)|$ virtual anchors with positions $\mathbf{p}_j^{(\eta)}$, $j \in \mathcal{H}(i)$ surround agent i and each virtual anchor (say the j th) collects a number of C distance measurements $r_{i,j} - \mu_{i,l}^{(\eta+1)}$ corrupted by errors with zero mean and variances $\sigma_{i,l}^{2,(\eta+1)} / \tilde{P}_{i,j,l}^{(\eta)}$, $l = 1, 2, \dots, C$. For this kind of problem, many existing linearization strategies, for instance [13, Chapter 2], can be used to further reduce the computational cost of the above proposed distributed ECM algorithms.

Before closing this subsection, we provide two practical solutions for initializing the distributed ECM algorithms. In the first solution, an initial guess of the mixture model parameters is determined by conventional Gaussian mixture learning carried out between a few anchors. An initial guess of the unknown agent positions can be determined in a simple method. For instance, a distributed least-squares algorithm [14, Algorithm 1] or a parametric SPAWN algorithm [83] with a starting point randomly selected in the deployment area. The latter has the potential to generate better initial guess with comparable computational complexity and energy consumption if the mixture model parameters are well initialized beforehand. In the second solution, we could employ a single moving platform (for instance a robot in the MUSAS system [5]) equipped with inertial sensors and ultra-wide-band (UWB) transceiver(s) and possibly also an imaging system. While patrolling with known trajectories inside the deployed area, this moving platform constantly communicates with all sensors within the communication range. Using the measurements received from the anchors, a coarse estimate of the mixture model parameters can be determined and broadcast back to all sensors. Meanwhile, recalibration of the inertial sensors can be conducted to reduce error propagation. Similarly, using the measurements received from the agents at different

Algorithm 5.3 Two Distributed ECM Algorithms for Self-Localization (for Agent i on the $(\eta + 1)$ th ECM Iteration)

Step1—Initialization: Choose initial guesses $\boldsymbol{\theta}_e^{(\eta=0)}$, $\mathbf{p}_i^{(0)}$, and $\mathbf{p}_j^{(0)}$, $\forall j \in \mathcal{H}(i)$. Set the maximum number of iterations N_{itr}^d .

Step2—Expectation-Conditional Maximization:

On the $(\eta + 1)$ th iteration ($\eta \in \mathbb{Z}, \eta \geq 0$), agent i does:

1. Compute $\tilde{P}_{i,j,l}^{(\eta)}$, $\forall l = 1, 2, \dots, C$ and $\forall j \in \mathcal{H}(i)$ according to (5.57), and $\chi_{i,l}^{(c)}$, $\forall l = 1, 2, \dots, C$ and $\forall \chi \in \{w, a, b\}$ according to (5.54)—(5.56) with $P_{i,j,l}^{(\eta)}$ replaced by $\tilde{P}_{i,j,l}^{(\eta)}$ therein.
2. Run Algorithm 5.2 and obtain $\alpha_{i,l}^{(\eta+1)}$, $\mu_{i,l}^{(\eta+1)}$, and $\sigma_{i,l}^{2,(\eta+1)}$, $l = 1, 2, \dots, C$.
3. Solve $\mathbf{p}_i^{(\eta+1)} = [x_i^{(\eta+1)}, y_i^{(\eta+1)}]^T$ using
 - either **2-D BFGS-QN method** according to (5.60);
 - or **1-D GS method** according to (5.61) and (5.62).
4. Broadcast $\mathbf{p}_i^{(\eta+1)}$ to its neighbors.

Step3—Convergence Check:

If N_{itr}^d has been reached, then terminate the whole algorithm and obtain $\hat{\boldsymbol{\theta}}^{\text{DECM}} = \boldsymbol{\theta}^{(\eta+1)}$; otherwise reset $\eta \leftarrow \eta + 1$ and return to the ECM stage.

time instances, the moving platform can determine a coarse estimate of the unknown positions through, for instance, the simple trilateration technique.

5.3.4 Dynamic Network Localization

Till now, the proposed ECM algorithms are considered for cooperative localization of stationary sensor networks. They can also localize dynamic sensor networks in a snapshot-based solution, more precisely, at every sampling time instance t , $t = 1, 2, \dots, T$, the proposed ECM algorithms are employed to compute an estimate of the current agent positions and measurement error statistics. Figure 5.2 illustrates the localization process. At the sampling time instance t , the underlying set of unknown parameters is $\boldsymbol{\theta}_t = [\boldsymbol{\theta}_{t,e}^T, \boldsymbol{\theta}_{t,p}^T]^T$ and a specific ECM algorithm gives an estimate after L iterations ($\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_t^{\text{DECM}}$ and $L = N_{itr}^d$ for a distributed algorithm or $\hat{\boldsymbol{\theta}}_t = \hat{\boldsymbol{\theta}}_t^{\text{CECM}}$ and $L = N_{itr}^c$ for a centralized algorithm). On each ECM iteration, we alternatively update the mixture model parameters $\boldsymbol{\theta}_{t,e}^{(\eta)}$ and agent positions $\boldsymbol{\theta}_{t,p}^{(\eta)}$. The starting point of the whole process, $\boldsymbol{\theta}_1^{(0)}$, can be selected according to the practical solutions given at the

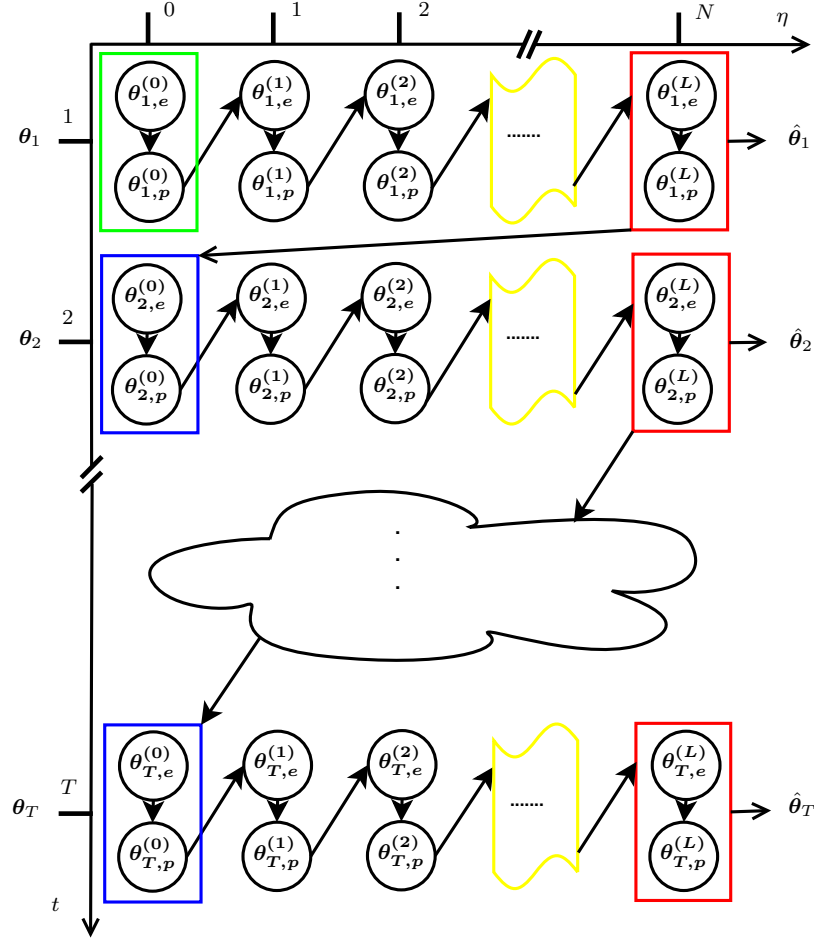


Figure 5.2. Localization of dynamic sensor networks using the ECM algorithm in a snapshot-based solution.

end of Section 5.3.3. Afterwards, $\theta_t^{(0)}$ is initialized by $\hat{\theta}_{t-1}$ for every $t = 2, \dots, T$, namely the ECM estimate obtained at the previous time instance.

5.4 Performance Evaluation

Computational complexity, communication overhead (also known as energy consumption for communication), and localization accuracy are key aspects of designing a cooperative localization algorithm. This section covers the first two and narrows down the focus to distributed algorithms.

5.4.1 Computational Complexity

For simplicity we start the complexity analysis with a single agent (say the i th) on a single iteration of the 2-D BFGS-QN D-ECM algorithm, cf. Algorithm 5.3. We focus on the second stage, namely the stage of expectation-conditional maximization. In the first step, the computation of $\tilde{P}_{i,j,l}^{(\eta)}$, $\forall j \in \mathcal{H}(i)$ according to (5.57) and $\chi_{i,l}^{(c)}$, $\forall \chi \in \{w, a, b\}$ according to (5.54)—(5.56) for all $l = 1, 2, \dots, C$ is of complexity $\mathcal{O}(C|\mathcal{H}(i)|)$. In the second step, local estimates of the mixture model parameters are computed in the average consensus algorithm, cf. Algorithm 5.2. The computational complexity scales as $\mathcal{O}(N_{itr}^g C)$. In the last step, position update $p_i^{(\eta+1)}$ is found numerically via a Newton-type method (iterative in nature). Similar to [67], it can be shown that the evaluation of a new position estimate scales as $\mathcal{O}(N_{itr}^{nu} C |\mathcal{H}(i)|)$, where N_{itr}^{nu} is defined to be the total number of Newton iterations. Therefore, the complexity for the i th agent to run a single iteration scales as $\mathcal{O}(C|\mathcal{H}(i)|(N_{itr}^{nu} + N_{itr}^g/|\mathcal{H}(i)|))$, which can be approximated by $\mathcal{O}(C|\mathcal{H}(i)|N_{itr}^{nu})$ when N_{itr}^g is small. The complexity of the centralized ECM algorithm is easy to obtain as $\mathcal{O}(N_u N_{itr}^{nu} C |\mathcal{H}_{ave}|)$, where $|\mathcal{H}_{ave}| = 1/N_u \sum_{i=1}^{N_u} |\mathcal{H}(i)|$ is the average number of neighboring sensors over all agents. Assuming that the sensors are uniformly distributed in a 2-D space with $|\mathcal{H}(i)| \approx |\mathcal{H}_{ave}|$, $\forall i = 1, 2, \dots, N_u$, the computational power required for the centralized ECM algorithm is nearly evenly distributed to each individual agent. This also holds for the 1D GS based ECM algorithms, as it is easy to verify that the computational complexity of the distributed algorithm scales as $\mathcal{O}(C|\mathcal{H}(i)|R_{grid})$ per iteration and the corresponding centralized algorithm $\mathcal{O}(N_u R_{grid} C |\mathcal{H}_{ave}|)$ per iteration, where R_{grid} is the number of grid points used in the 1-D search of x_i or y_i .

We compare the complexity of the two distributed ECM algorithms with that of six different distributed algorithms. The results are given in Table 5.1. It is noteworthy that R_{int} denotes the number of particles used to represent the internal messages (distributions) for both the classical and parametric SPAWN algorithms. In practice, R_{int} usually spans from 500 to 2000 particles. From the results, the following facts are observed. Firstly, the distributed LS algorithm, the distributed weighted MDS algorithm and the IPPM based algorithm require the lowest complexity, as they require no iterations within an iteration as compared to the distributed ECM algorithms and no particle representation of the messages as compared to the classical SPAWN algorithm. Secondly, the complexity of the 2-D BFGS-QN D-ECM algorithm is expected to be less than that of the SPAWN algorithms. This is because $N_{itr}^{nu} C$ should be, in general, smaller than R_{int} and negligible as compared to R_{int}^2 . The complexity of the 1-D GS D-ECM algorithm should be comparable with that of the parametric SPAWN algorithms but still much less than that of the conventional SPAWN algorithm. This

is because $R_{grid} \approx R_{int} \ll R_{int}^2$.

Table 5.1. Computational complexity for each agent (say the i th) on one iteration of different algorithms

Name	Complexity (FLOPs)
Distributed least-squares [14]	$\mathcal{O}(\mathcal{H}(i))$
Distributed weighted MDS [71]	$\mathcal{O}(\mathcal{H}(i))$
Distributed IPPM [75]	$\mathcal{O}(\mathcal{H}(i))$
Parametric SPAWN [83] [84]	$\mathcal{O}(R_{int} \mathcal{H}(i))$
Classical SPAWN [14]	$\mathcal{O}(R_{int}^2 \mathcal{H}(i))$
2-D BFGS-QN D-ECM	$\mathcal{O}(N_{itr}^{nu}C \mathcal{H}(i))$
2-D BFGS-QN C-ECM	$\mathcal{O}(N_u N_{itr}^{nu}C \mathcal{H}_{ave})$
1-D GS D-ECM	$\mathcal{O}(R_{grid}C \mathcal{H}(i))$
1-D GS C-ECM	$\mathcal{O}(N_u R_{grid}C \mathcal{H}_{ave})$

5.4.2 Energy for Communication

Generally speaking, a centralized, cooperative localization algorithm spends most of the energy on routing the collected measurements wirelessly via multi-hop to a fusion center; while a distributed self-localization algorithm spends most of the energy on both the local computation and wireless transmission of refined point estimates (for non-Bayesian algorithms) or particles (for Bayesian algorithms) among one-hop neighboring sensor pairs. Usually, the latter aspect is solely focused. This is due to the fact that the energy consumed for transmitting one bit far outweighs that for executing a single instruction on board at each agent [132]. In the sequel, we follow the methodology proposed in [124, 133] to quantitatively analyze the total energy consumed for wireless communication. Some assumptions are made as follows:

- (A1) A multi-hop communication model is considered, and the threshold of one-hop distance is $R_c = \mathcal{O}(N_u^{-1/2})$.
- (A2) A real value is represented in double-precision floating-point format (64-bit precision).
- (A3) All sensors are uniformly distributed over a 2-D unit square.

The total energy consumed for communication by any cooperative localization algorithm can be written as

$$E(N_u) = b(N_u) \times h(N_u) \times e(N_u) \quad (5.63)$$

where $b(N_u)$ is the total number of transmitted bits, $h(N_u)$ is the average number of hops required for transmitting one bit to the destination, and $e(N_u)$ is the average amount of energy required for transmitting one bit over one hop.

In the centralized ECM algorithms, the number of bits to be transmitted to the fusion center is calculated as

$$b_{cen}(N_u) = 2^6 |\Gamma| \approx 2^6 \sum_{i=1}^{N_u} \frac{|\mathcal{H}(i)|}{2} = 2^5 N_u |\mathcal{H}_{ave}| \text{ bits.} \quad (5.64)$$

The approximation in (5.64) is due to the assumptions that only one of $r_{i,j}$ and $r_{j,i}$ is used and the number of anchors is relatively small. The average number of hops from an agent to the fusion center is $h_{cen}(N_u) = \mathcal{O}(N_u^{1/2})$. Therefore, the total energy consumption for the centralized ECM algorithms is

$$E_{cen}(N_u) = \mathcal{O}(2^5 |\mathcal{H}_{ave}| N_u^{3/2}) \times e(N_u). \quad (5.65)$$

In the distributed ECM algorithms, the total number of bits exchanged by all neighboring sensor pairs is given by

$$b_{dis}(N_u) = 2^6 N_{itr} N_u (N_{itr}^g 6C + 2) \text{ bits} \approx 2^6 \cdot 6 N_{itr} N_u N_{itr}^g C \text{ bits.} \quad (5.66)$$

Note that the energy used for the position updates is negligible as compared to that for updating the mixture model parameters via average consensus, which leads to the above approximation. Since all the data are communicated between one-hop neighboring sensors, we have $h_{dis}(N_u) = \mathcal{O}(1)$. Therefore, the energy consumed by the distributed ECM algorithms is given by

$$E_{dis}(N_u) = \mathcal{O}(2^6 \cdot 6 N_{itr} N_u N_{itr}^g C) \times e(N_u). \quad (5.67)$$

It is obvious that the energy consumption of the distributed ECM algorithms depends on three factors, i.e., the number of mixture components C , the number of consensus iterations N_{itr}^g and the number of ECM iterations $N_{itr} = N_{itr}^d$. Usually, we set $C < 5$ to capture the main characteristics of the measurement error, $N_{itr}^g \approx 5 - 10$, and $N_{itr}^d \approx 50 - 100$.

For comparison purposes, we listed the energy consumption for different distributed algorithms in Table 5.2. For clarity, we ignore the constant factors and $e(N_u)$ in the results. It is noteworthy that R_{bel} in the classical SPAWN algorithm [14] denotes the number of particles required to represent each local belief message. In general, R_{bel} spans from 500 to 1000 samples, depending on the desired localization accuracy.

Table 5.2. Total energy consumed by a distributed algorithm on one iteration.

Name	Total Energy Consumption
Distributed least-squares [14]	$\mathcal{O}(N_u)$
Distributed weighted MDS [71]	$\mathcal{O}(N_u)$
Parametric SPAWN [83]	$\mathcal{O}(C_{bel}N_u)$
Parametric SPAWN [84]	$\mathcal{O}(C_{bel}N_u)$
Classical SPAWN [14]	$\mathcal{O}(R_{bel}N_u)$
Distributed ECMs	$\mathcal{O}(CN_{itr}^g N_u)$

To reduce the energy consumption, some new fashioned SPAWN algorithms, like [83] and [84], resort to fit each local belief message with different parametric models and to exchange merely the corresponding model parameters between sensors. Here, C_{bel} denotes the total number of parameters to be determined for the parametric model, which is usually less than 10. Gathering the previous analysis and the results shown in Table 5.2, it is easy to conclude that (1) the distributed LS algorithm and distributed weighted MDS algorithm consume the least energy; (2) the energy consumption of the distributed ECM algorithms lies in between that of the classical SPAWN algorithm and that of the parametric SPAWN variations; (3) the centralized ECM algorithms cost the largest amount of energy for large-scale wireless sensor networks.

5.5 Cramér-Rao Lower Bound Computation

Fisher's information matrix (FIM) of position parameters is hard to evaluate in closed form for the considered joint estimation problem. Hence, we resort to a numerical approximation of it. Then, we relate a metric of localization accuracy with this approximated FIM.

We start by expressing the FIM of $\boldsymbol{\theta}$ as

$$\mathcal{F}(\boldsymbol{\theta}) = \mathbb{E}_{p(\mathbf{r};\boldsymbol{\theta})} \left\{ -\Delta_{\boldsymbol{\theta}}^{\boldsymbol{\theta}} \ln p(\mathbf{r}; \boldsymbol{\theta}) \right\} \quad (5.68)$$

where the expectation is taken with respect to $p(\mathbf{r}; \boldsymbol{\theta})$. Often, it is more convenient to express FIM as

$$\mathcal{F}(\boldsymbol{\theta}) = \int (\nabla_{\boldsymbol{\theta}} \ln p(\mathbf{r}; \boldsymbol{\theta}) \nabla_{\boldsymbol{\theta}}^T \ln p(\mathbf{r}; \boldsymbol{\theta})) p(\mathbf{r}; \boldsymbol{\theta}) d\mathbf{r}. \quad (5.69)$$

Due to the difficulty in evaluating the integration in (5.69) analytically, we could apply the Monte Carlo integration [108] on (5.69) directly but at the cost of high computational complexity, like what was done in [67]. As we are considering a localization

problem, position estimation performance is of utmost interest. To reduce the computational complexity, we assume that the actual measurement error distribution $p_V(v)$ is known and focus attention on the unknown positions $\boldsymbol{\theta}_p$.

Combining the procedures given in Chapter 3 and [19], the FIM of $\boldsymbol{\theta}_p$ given $\boldsymbol{\theta}_e$ can be easily obtained as

$$\mathcal{F}(\boldsymbol{\theta}_p) = \begin{pmatrix} \mathcal{F}_{xx} & \mathcal{F}_{xy} \\ \mathcal{F}_{xy}^T & \mathcal{F}_{yy} \end{pmatrix} \quad (5.70)$$

where \mathcal{F}_{xx} , \mathcal{F}_{xy} and \mathcal{F}_{yy} are all square matrices of dimension $N_u \times N_u$ with

$$[\mathcal{F}_{mn}]_{i,i'} = \begin{cases} \mathcal{I}_v \cdot \sum_{\forall j \in \mathcal{H}(i)} \frac{(m_i - m_j)(n_i - n_j)}{\|\mathbf{p}_i - \mathbf{p}_j\|^2}, & i = i' \\ -\mathcal{I}_v \cdot \delta_{i,i'} \cdot \frac{(m_i - m_{i'})(n_i - n_{i'})}{\|\mathbf{p}_i - \mathbf{p}_{i'}\|^2}, & i \neq i' \end{cases}, \quad (5.71)$$

for $\mathbf{m}, \mathbf{n} \in \{\mathbf{x}, \mathbf{y}\}$. Here, $\delta_{i,i'}$ is Kronecker's delta defined by

$$\delta_{i,i'} = \begin{cases} 1, & \text{if } i' \in \mathcal{H}(i) \\ 0, & \text{if } i' \notin \mathcal{H}(i) \end{cases}, \quad (5.72)$$

and

$$\mathcal{I}_v = \int \frac{[\nabla_v p_V(v)]^2}{p_V^2(v)} p_V(v) dv. \quad (5.73)$$

For most of the distributions, \mathcal{I}_v is approximated using the Monte Carlo integration, i.e.,

$$\mathcal{I}_v \approx \frac{1}{N_M} \sum_{n=1}^{N_M} \frac{[\nabla_v p_V(v^{(n)})]^2}{p_V^2(v^{(n)})} \quad (5.74)$$

where $v^{(n)}$, $n = 1, 2, \dots, N_M$ are iid samples generated from $p_V(v)$. For the special case where $p_V(v) = \mathcal{N}(v; \mu_v, \sigma_v^2)$, $\mathcal{I}_v = \sigma_v^{-2}$ can be obtained in closed form and the result in (5.71) coincides with that in [19]. Finally, the Cramér-Rao lower bound of $\boldsymbol{\theta}_p$ is given by $\text{CRLB}(\boldsymbol{\theta}_p) = \mathcal{F}^{-1}(\boldsymbol{\theta}_p)$.

In the simulations, the localization accuracy is evaluated in terms of the overall localization root mean square error (RMSE), which is defined by

$$\overline{\text{RMSE}}(\hat{\boldsymbol{\theta}}_p) = \sqrt{\frac{1}{N_u} \sum_{i=1}^{N_u} \mathbb{E} \{ (\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 \}}. \quad (5.75)$$

Assuming that $[\hat{x}_i, \hat{y}_i]$ is any unbiased estimator of the true position $[x_i, y_i]$, $\overline{\text{RMSE}}(\hat{\boldsymbol{\theta}}_p)$ is lower bound by

$$\overline{\text{CRLB}}_{\text{pos}}(\boldsymbol{\theta}_p) = \sqrt{\frac{1}{N_u} \text{tr} \{ \text{CRLB}(\boldsymbol{\theta}_p) \}}. \quad (5.76)$$

5.6 Simulations

In this section, the performance of the proposed centralized- and distributed ECM algorithms will be evaluated and further compared with several competing cooperative localization algorithms in comprehensive simulations. Section 5.6.1 introduces the simulation setup and Section 5.6.2 shows some simulation results.

5.6.1 Simulation Setup

We consider the following three different localization scenarios:

1. Localization of stationary networks under a simulated setup. Partial knowledge about the measurement error distribution is assumed, which leads to precise representation of $p_V(v)$ in the parametric modeling of the measurement error distribution.
2. Localization of dynamic networks under a simulated setup. Again, partial knowledge about the measurement error distribution is assumed.
3. Localization of stationary networks under a real setup. No knowledge about the measurement error distribution is assumed, which leads to approximated representation of $p_V(v)$ in the parametric modeling of the measurement error distribution.

Throughout the simulations, the proposed ECM algorithms assume $N_{itr}^c = N_{itr}^d = 75$ trials, $N_{itr}^g = 10$ trials, $R_{grid} = 150$ grid points, and $\Delta_c = 0.01$. For comparison purposes, we choose several competing algorithms, including the distributed LS algorithm [14, Algorithm 1], the classical SPAWN algorithm [14], as well as two parametric variations [83, 84]. Essentially, these algorithms were developed under different signal models. The distributed LS algorithm makes no assumption on the measurement error statistics. The SPAWN algorithms all assume known mixture model parameters and underlying channel state (LOS or NLOS state). Details about the implementation of these competitors are as follows. A distributed LS estimate is found by following the routine given in [14]. The classical SPAWN algorithm sets $R_{int} = R_{bel} = 500$ samples. The first parametric SPAWN algorithm [83] uses a three-mode Gaussian mixture to represent the belief messages and uses a ten-mode Gaussian mixture to represent the internal messages. The second parametric SPAWN algorithm modifies [84] by adopting

a circular distribution to represent the internal messages while a three-mode Gaussian mixture to model the belief messages. All SPAWN algorithms terminate after $N_{itr} = 15$ iterations.

5.6.2 Simulation Results

We show the simulation results of localizing a stationary sensor network and a dynamic sensor network respectively in Section 5.6.2.1 and Section 5.6.2.2. Moreover, we show the experimental results of localizing a stationary ad-hoc sensor network in Section 5.6.2.3.

5.6.2.1 Results for Scenario I

In this section, we consider a specific stationary sensor network in a 90-meter (m) by 90-m area, where 96 agents and 4 anchors are uniformly placed into 10 rows and 10 columns, like in [6, Fig. 6]. It is assumed that the NLOS effect overwhelms any other error sources in the given localization environment and the actual distance measurement error can be well modeled by a two-mode Gaussian mixture distribution with parameters $\alpha_L = 0.3$, $\mu_L = 0$ m, $\sigma_L = 2$ m, $\alpha_{NL} = 0.7$, $\mu_{NL} = 10$ m, $\sigma_{NL} = 3$ m. This implies that the NLOS effect introduces both a positive bias and a larger measurement uncertainty. Herein, no model mismatch is assumed between the actual measurement error model and (5.2) when choosing $C = 2$ in the parametric modeling of $p_V(v)$. We aim to experimentally evaluate the proposed ECM algorithms in terms of the overall localization RMSE, cf. (5.75), and localization outage probability, cf. ([14, eq.(47)]).

We perform a Monte Carlo experiment with 500 independent trials. The initial guess of the ECM algorithms is set by $\alpha_1^{(0)} = 0.4$, $\alpha_2^{(0)} = 0.6$, $\mu_1^{(0)} = 0$, $\mu_2^{(0)} = 12$, $\sigma_1^{2,(0)} = 3$, $\sigma_2^{2,(0)} = 6$. An initial guess of the agent positions, $\boldsymbol{\theta}_p^{(0)}$, is set by contaminating the true values with an error term Δ_{xy} whose elements are generated independently from $\mathcal{U}[-5, 5)$ (in meter).

The evaluation starts with the overall localization RMSE versus the communication range, R_c , that varies from 20 m to 40 m. The results are shown along with the localization CRLB bound in Fig. 5.3. It is observed that almost all RMSE curves monotonically decrease as R_c grows.¹ This can be explained from a Fisher's information

¹The fluctuation of the parametric SPAWN algorithm [83] is mainly due to the insufficient number of Monte Carlo trials (only 10 trials) conducted.

theoretic point of view. As R_c increases, more sensor pairs conduct measurements, leading to more information about the relative positions. The RMSE curves of the distributed ECM algorithms are close to those of the centralized counterparts. This is because the local estimates of the mixture model parameters calculated through the average consensus algorithm are close to the corresponding global estimate, which ensures accurate position updates thereafter. It is also observed that the distributed ECM algorithms can generate lower RMSE than that of the centralized ECM algorithm and even the performance bound.

One reason is that the distributed ECM estimators are merely approximations of the centralized counterparts, and they are not necessarily unbiased estimators. Besides, the centralized ECM estimators are in fact also biased due to the convergence to different local optimum, insufficient number of iterations. An additional reason for the distributed ECM algorithms lies in the discretization of the solution space (for positions) by a finite number of grids. Nevertheless, all RMSE curves of the ECM algorithms are close to the performance bound. To further demonstrate the estimation performance of the ECM algorithms, we show the localization outage probability as a function of the allowable error $e_{th} = 0 : 0.5 : 5$ m. The results are depicted in Fig. 5.4. In the above two experiments, the ECM algorithms are also demonstrated to outperform the parametric SPAWN algorithm [83] by far. However, we note that their performance should be comparable with that of the classical SPAWN algorithm with known knowledge about the environment.

Next, we aim to experimentally substantiate our statement that the centralized ECM algorithms can monotonically increase the incomplete data log-likelihood over iterations. To this end, we record the incomplete data log-likelihood versus the number of iterations for each Monte Carlo trial. In Fig. 5.5, we show one particular trial as well as the mean performance averaged over all Monte Carlo trials. The convergence speed of the 2-D BFGS-QN C-ECM is slow. However, 1-D GS C-ECM usually converges much faster but to a smaller incomplete data log-likelihood value due to the discretization of the solution space and a small R_{grid} . This problem can be alleviated by increasing R_{grid} for instance from 150 to 1000 points as is shown in Fig. 5.5. It is also noteworthy that in many Monte Carlo trials, although the 1-D GS C-ECM algorithm tends to stuck at a smaller incomplete data log-likelihood value, the overall RMSE value can be also smaller than that of the 2-D BFGS algorithm.

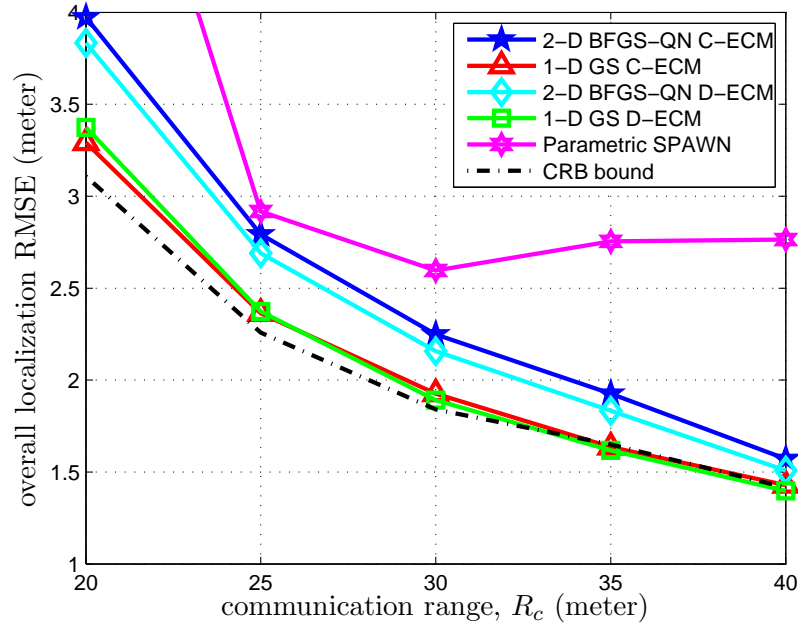


Figure 5.3. Overall localization RMSE of different cooperative localization algorithms as a function of the communication range, R_c .

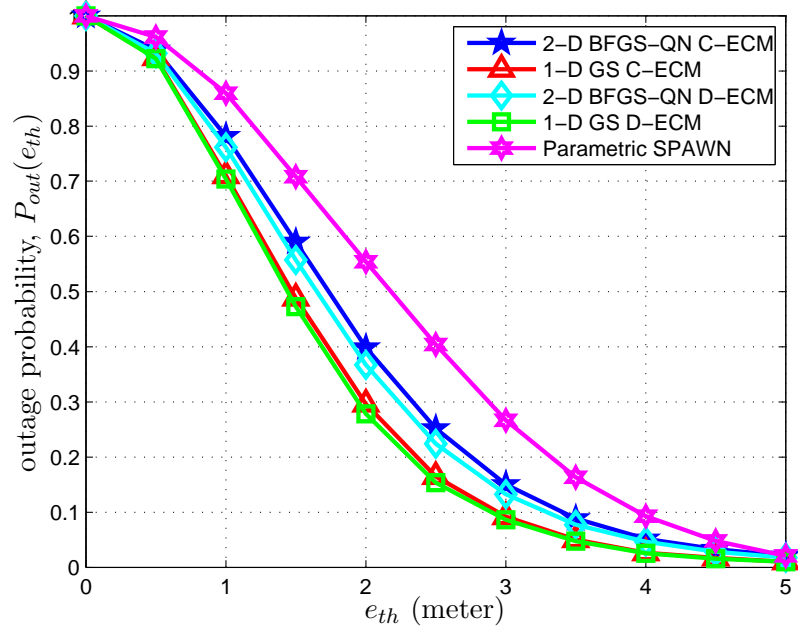


Figure 5.4. Outage probability of different cooperative localization algorithms for the case $R_c = 30$ m.

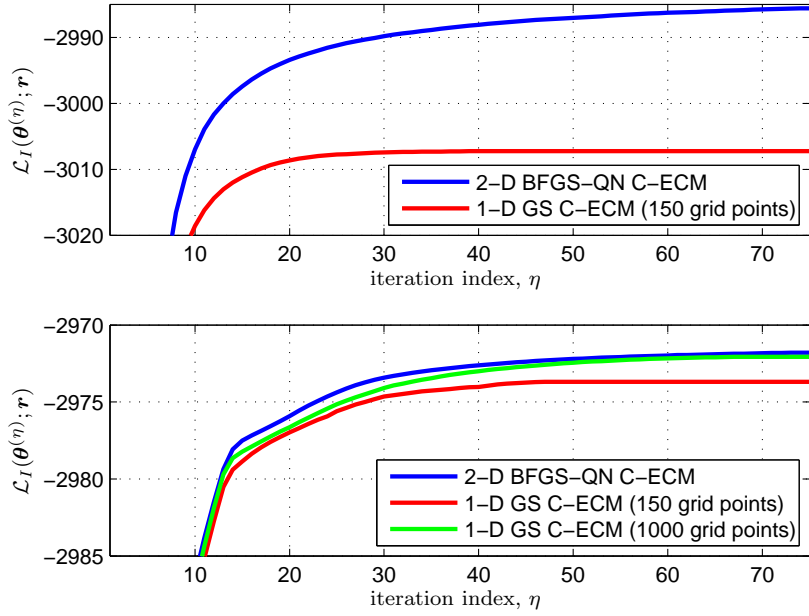


Figure 5.5. **Top sub-figure:** Mean incomplete data log-likelihoods over all Monte Carlo trials as a function of the iteration index. **Bottom sub-figure:** Monotonic increment of the incomplete data log-likelihood over iterations in one particular Monte Carlo trial. Two different R_{grid} values are tested.

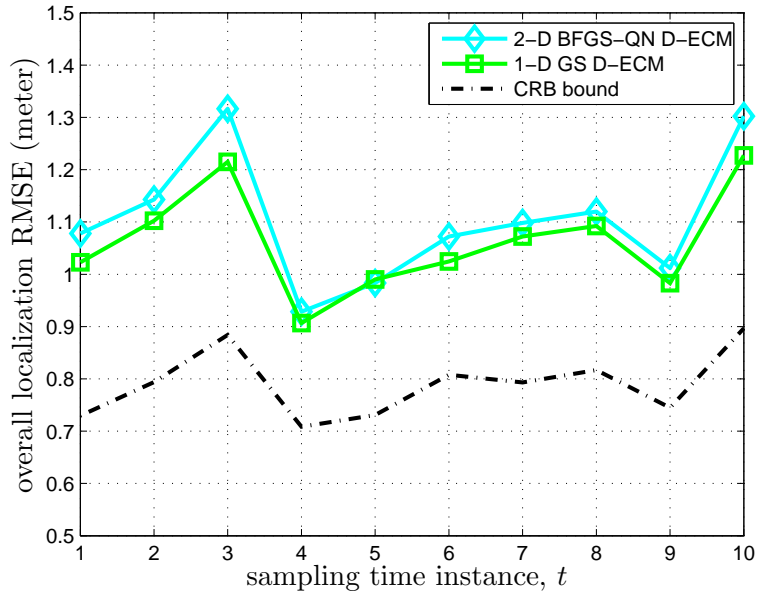


Figure 5.6. Overall localization RMSE of the two distributed ECM algorithms as a function of the sampling time instance t in a dynamic network.

5.6.2.2 Results for Scenario II

In this Section, evaluation of the distributed ECM algorithms is re-considered for localization of a dynamic network using the snapshot based solution introduced in Section 5.3.4. Still, no model mismatch is taken into account. The initial state ($k = 1$) of the network follows the first map data published online along with [14]. This network comprises 100 agents and 13 anchors. Every agent i moves according to $m_{k,i} = m_{k,i} + \mathcal{N}(\Delta_{k,m}; 0, 1)$, $m \in \{x, y\}$ when $k = 2, \dots, 10$. Besides, α_L and μ_{NL} are assumed to be time varying due to the change of obstructions between sensors. The parameters are given in Table 5.3. The rest of the actual mixture model parameters are assumed to be constants and set as $\mu_L = 0$ m, $\sigma_L = 1$ m, $\sigma_{NL} = 2$ m.

Table 5.3. Time varying α_L and μ_{NL} for dynamic network localization.

	$t = 1$	2	3	4	5	6	7	8	9	10
$\alpha_{t,L}$	0.7	0.65	0.6	0.8	0.7	0.72	0.6	0.65	0.7	0.5
$\mu_{t,NL}$	7.5	6	5	6.5	8	5	7	6	9	7

Herein, the overall localization RMSE of the two distributed ECM algorithms is evaluated at each time step $t = 1, 2, \dots, 10$. The initial guess of the snapshot based solution is set as follows. At $t = 1$, we let $\alpha_{1,1}^{(0)} = 0.6, \alpha_{1,2}^{(0)} = 0.4, \mu_{1,1}^{(0)} = 0, \mu_{1,2}^{(0)} = 6, \sigma_{1,1}^{2,(0)} = 2, \sigma_{1,2}^{2,(0)} = 5$ and select an initial guess of the positions in the same way as described in the previous section. An ECM algorithm is run independently at each time step with the current starting point initialized by the final ECM estimate obtained at the previous time instance. The results are averaged over 100 independent Monte Carlo trials and shown in Fig. 5.6 together with $\overline{\text{CRLB}}_{\text{pos}}(\boldsymbol{\theta}_{t,p}), t = 1, 2, \dots, 10$, for comparison. Still, we observe that the overall RMSE values at different sampling time instances are close to the performance bounds. Localization error propagation is not observed in the given example because the previous ECM estimate adequately initialized a current ECM stage. Although comparisons are not shown in the figure, we note that the proposed distributed ECM algorithms should outperform the LS estimation based algorithms by far, since they are suboptimal in non-Gaussian measurement errors. The proposed algorithms should be comparable in localization performance with the SPAWN algorithms. But in contrast to the SPAWN algorithms, the proposed algorithms do not require repetitive offline calibrations and NLOS identifications, which make them more suitable to use in harsh environments.

Table 5.4. Overall localization RMSE of different distributed algorithms.

Algorithms	RMSE for $R_c = \infty$ m
Distributed LS [14]	1.134 m
Classical SPAWN [14]	1.146 m
Parametric SPAWN [83]	1.163 m
Parametric SPAWN [84]	1.260 m
2-D BFGS-QN D-ECM ($C = 2$)	1.389 m
1-D GS D-ECM ($C = 2$)	1.333 m
2-D BFGS-QN D-ECM ($C = 3$)	1.398 m
1-D GS D-ECM ($C = 3$)	1.340 m

5.6.2.3 Results for Scenario III

In this section, we adopt the real sensor network and TOA measurements described in [19]. The network consists of 44 sensors in total, among which 40 agents and 4 anchors. The distance measurement error after bias remedy ($\mu = 10.9 \text{ ns} \times c \approx 3.26 \text{ m}$) was justified via Kolmogorov-Smirnov (KS) test to well fit a Gaussian distribution with zero mean and standard deviation $\sigma = 6.1 \text{ ns} \times c \approx 1.83 \text{ m}$. Herein, the proposed ECM algorithms will be tested under model mismatch.

To test the two distributed ECM algorithms, we use the original distance measurements without bias remedy and assume a C -mode Gaussian mixture to approximate the underlying measurement error pdf. But their competitors all use the manipulated data and assume a Gaussian error model.² The ECM algorithms assume $C = 2$ or $C = 3$. For the case $C = 2$, we let $\alpha_1^{(0)} = 0.5, \mu_1^{(0)} = 2, \sigma_1^{2,(0)} = 1, \alpha_2^{(0)} = 0.5, \mu_2^{(0)} = 4, \sigma_2^{2,(0)} = 4$; while for the case $C = 3$, we let $\alpha_1^{(0)} = 0.1, \mu_1^{(0)} = 0, \sigma_1^{2,(0)} = 3, \alpha_2^{(0)} = 0.8, \mu_2^{(0)} = 3.5, \sigma_2^{2,(0)} = 2, \alpha_3^{(0)} = 0.1, \mu_3^{(0)} = 7, \sigma_3^{2,(0)} = 3$. The overall localization RMSE is evaluated for different distributed algorithms in a fully connected network, and the results are shown in Table 5.4. Besides, the estimated agent positions obtained from the two distributed ECM algorithms are depicted for the case $C = 3$ in Fig. 5.7.

It is obvious that the ECM algorithms, no matter $C = 2$ or $C = 3$, are modestly inferior to their competitors. The reasons are twofold. On the one hand, the measurement error well fits a Gaussian distribution, which is alternatively verified by comparing the single Gaussian approximation with a kernel density estimate in Fig. 5.8.³ It is also shown in Fig. 5.8 that a Gaussian mixture approximation with either $C = 2$ or $C = 3$

²In practice, bias remedy can hardly be done as in [19] for lack of actual agent positions and/or offline calibration.

³Herein, we assume the error residuals are iid and perform the diffusion based kernel density estimation [134].

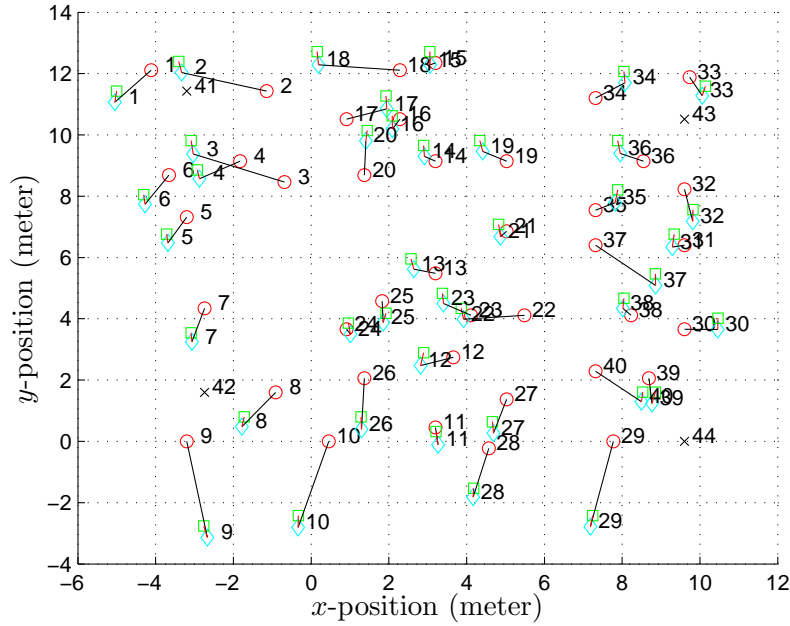


Figure 5.7. Estimated agent positions versus the true ones in 2-D plane. Herein, black \times 's denote the anchor positions; red \circ 's denote the true agent positions; cyan \diamond 's denote the agent positions estimated from the 2-D BFGS-QN D-ECM algorithm; green \square 's denote the agent positions estimated from the 1-D GS D-ECM algorithm; black $-$'s represent the localization errors between the true positions and the estimated positions; red $-$'s represent the distances between the position estimates obtained from the two distributed ECM algorithms.

leads to more severe model mismatch. Note that the two approximated distributions nearly coincide with each other in Fig. 5.8. On the other hand, the distributed ECM algorithms need to estimate an extra set of mixture model parameters as compared to their competitors.

5.7 Conclusions

In this chapter, a series of expectation-conditional maximization (ECM) algorithms have been proposed for cooperative localization in non-Gaussian measurement error approximated by a Gaussian mixture. The centralized ECM algorithms have been proven to be able to increase the incomplete data log-likelihood monotonically towards a stationary value. Distributed ECM algorithms have also been developed to resolve the scalability problem in large-scale sensor networks. Systematical analyses have shown that the proposed distributed algorithms perform similarly to the class of parametric

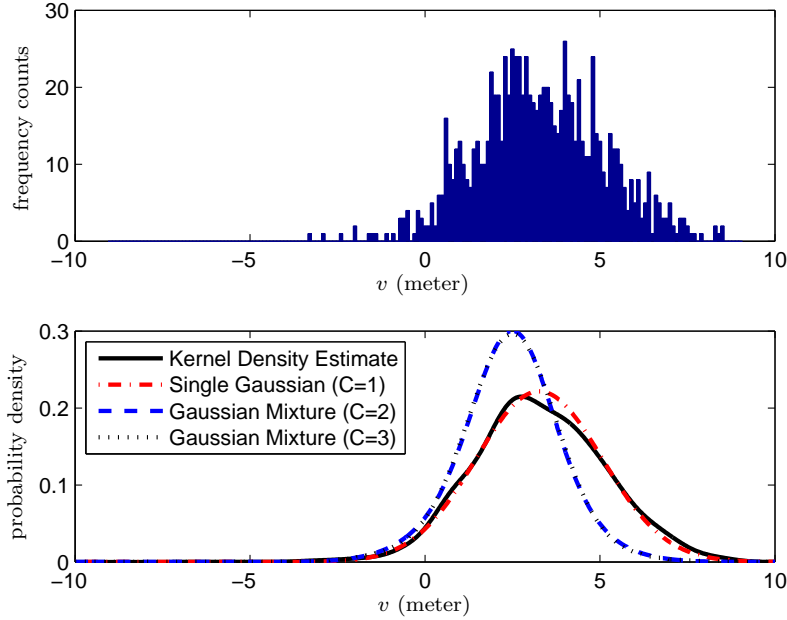


Figure 5.8. **Top sub-figure:** Histogram of the actual measurement error residuals $v_{i,j}$, $\forall(i,j) \in \Gamma$ for $R_c = \infty$ meter. **Bottom sub-figure:** Different distributive profiles (including single Gaussian obtained from [19], two-mode Gaussian mixture and three-mode Gaussian mixture obtained respectively from the 2-D BFGS-QN D-ECM algorithm) versus a kernel density estimate serving as the underlying distribution for lack of ground truth.

SPAWN algorithms in terms of both the computational complexity and energy consumption for data communication. Simulations with both synthetic and real data have demonstrated that (1) the proposed ECM algorithms tend to attain the performance limits when the number of measurements is much larger than the number of unknown parameters and no model mismatch problem occurs; (2) the proposed ECM algorithms can work properly, despite of the sub-optimality, under model mismatch problems.

The proposed ECM algorithms are non-Bayesian in nature. They are more robust against outliers in the observations as compared to the conventional LS algorithms, and they require no precise measurement error statistics and channel states in comparison with the Bayesian algorithms, e.g., the SPAWN algorithms. Nevertheless, many research challenges need to be met in the future work, including (1) reduction of computational complexity and energy consumption; (2) integration of different signal metrics, e.g., TOA and RSS; (3) realistic assumption on lossy wireless transmission and quantized messages (point estimates for non-Bayesian methods and particles representing the posterior distribution).

5.8 Appendix

5.8.1 Derivations of (5.19)

We start with expressing the complete data probability density function as

$$p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}) = b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \exp [\boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbf{s}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) - a(\boldsymbol{\theta}_e)] \quad (5.77)$$

where

$$\tilde{\mathbf{r}}(\boldsymbol{\theta}_p) = \mathbf{r} - \mathbf{h}(\boldsymbol{\theta}_p), \quad b(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = \left(\frac{1}{2\pi}\right)^{\frac{|\Gamma|}{2}}, \quad a(\boldsymbol{\theta}_e) = 0, \quad (5.78)$$

$$\boldsymbol{\psi}(\boldsymbol{\theta}_e) = [\boldsymbol{\psi}_1^T(\boldsymbol{\theta}_e), \boldsymbol{\psi}_2^T(\boldsymbol{\theta}_e), \dots, \boldsymbol{\psi}_C^T(\boldsymbol{\theta}_e)]^T, \quad (5.79)$$

$$\mathbf{s}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = [s_1^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}), s_2^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}), \dots, s_C^T(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})]^T, \quad (5.80)$$

and for $l = 1, 2, \dots, C$,

$$\boldsymbol{\psi}_l(\boldsymbol{\theta}_e) = \left[\ln \left(\frac{\alpha_l}{\sigma_l} \right) - \frac{\mu_l^2}{2\sigma_l^2}, \frac{\mu_l}{\sigma_l^2}, \frac{-1}{2\sigma_l^2} \right]^T, \quad (5.81)$$

$$s_l(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) = \begin{bmatrix} \sum_{\forall (i,j) \in \Gamma} \delta(y_{i,j} - l) \\ \sum_{\forall (i,j) \in \Gamma} (r_{i,j} - d_{i,j}) \delta(y_{i,j} - l) \\ \sum_{\forall (i,j) \in \Gamma} (r_{i,j} - d_{i,j})^2 \delta(y_{i,j} - l) \end{bmatrix}. \quad (5.82)$$

This is due to the fact that the complete data probability density function can be expressed as

$$\begin{aligned} p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta}) &= \prod_{\forall (i,j) \in \Gamma} p(r_{i,j}, y_{i,j}; \boldsymbol{\theta}) \\ &= \prod_{\forall (i,j) \in \Gamma} \left(\frac{\alpha_{y_{i,j}}}{\sqrt{2\pi}\sigma_{y_{i,j}}} \exp \left[\frac{-(r_{i,j} - d_{i,j} - \mu_{y_{i,j}})^2}{2\sigma_{y_{i,j}}^2} \right] \right) \\ &= \left(\frac{1}{2\pi} \right)^{\frac{|\Gamma|}{2}} \exp \left[\sum_{\forall (i,j) \in \Gamma} \ln \left(\frac{\alpha_{y_{i,j}}}{\sigma_{y_{i,j}}} \right) - \frac{(r_{i,j} - d_{i,j} - \mu_{y_{i,j}})^2}{2\sigma_{y_{i,j}}^2} \right]. \end{aligned} \quad (5.83)$$

The summation inside the exponential function can be expressed as

$$\begin{aligned}
& \sum_{\forall(i,j) \in \Gamma} \ln \left(\frac{\alpha_{y_{i,j}}}{\sigma_{y_{i,j}}} \right) - \frac{(r_{i,j} - d_{i,j} - \mu_{y_{i,j}})^2}{2\sigma_{y_{i,j}}^2} \\
&= \sum_{l=1}^C \sum_{\forall(i,j) \in \Gamma} \left(\ln \left(\frac{\alpha_l}{\sigma_l} \right) - \frac{(r_{i,j} - d_{i,j} - \mu_l)^2}{2\sigma_l^2} \right) \delta(y_{i,j} - l) \\
&= \sum_{l=1}^C \sum_{\forall(i,j) \in \Gamma} \left\{ \left(\ln \left(\frac{\alpha_l}{\sigma_l} \right) - \frac{\mu_l^2}{2\sigma_l^2} \right) + \frac{\mu_l(r_{i,j} - d_{i,j})}{\sigma_l^2} - \frac{(r_{i,j} - d_{i,j})^2}{2\sigma_l^2} \right\} \delta(y_{i,j} - l) \\
&= \sum_{l=1}^C \boldsymbol{\psi}_l^T(\boldsymbol{\theta}_e) \mathbf{s}_l(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \\
&= \boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbf{s}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}). \tag{5.84}
\end{aligned}$$

Plugging (5.84) into (5.83) yields (5.77). Taking conditional expectation of the complete data log-likelihood $\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$ in terms of \mathbf{y} given \mathbf{r} and $\boldsymbol{\theta}^{(\eta)}$ and discarding the constant terms, we finally arrive at

$$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) = \boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)})} \{ \mathbf{s}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y}) \}. \tag{5.85}$$

For clarity of the subsequent derivations, the elements in the vectorized signal model (5.3) are re-labeled such that r_m is the m th element of \mathbf{r} and y_m is the latent variable giving rise to r_m . The computation of $\mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)})} [\mathbf{s}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})]$ can be decomposed into three parts. We start with

$$\begin{aligned}
& \mathbb{E}_{p(\mathbf{y}|\mathbf{r}; \boldsymbol{\theta}^{(\eta)})} \left\{ \sum_{\forall(i,j) \in \Gamma} \delta(y_{i,j} - l) \right\} \\
&= \sum_{\mathbf{y}} \left[\sum_{m=1}^{|\Gamma|} \delta(y_m - l) \times \prod_{j=1}^{|\Gamma|} \Pr\{y_j | r_j; \boldsymbol{\theta}^{(\eta)}\} \right] \\
&= \sum_{m=1}^{|\Gamma|} \left[\sum_{y_1=1}^C \cdots \sum_{y_{|\Gamma|=1}}^C \left(\delta(y_m - l) \times \prod_{j=1}^{|\Gamma|} \Pr\{y_j | r_j; \boldsymbol{\theta}^{(\eta)}\} \right) \right] \\
&= \sum_{m=1}^{|\Gamma|} \Pr\{y_m = l | r_m; \boldsymbol{\theta}^{(\eta)}\} \times \left[\sum_{y_1=1}^C \cdots \sum_{y_{m-1}=1}^C \sum_{y_{m+1}=1}^C \cdots \sum_{y_{|\Gamma|=1} \substack{j=1, \\ j \neq m}}^C \prod_{j=1, \\ j \neq m}^{|\Gamma|} \Pr\{y_j | r_j; \boldsymbol{\theta}^{(\eta)}\} \right] \\
&= \sum_{m=1}^{|\Gamma|} \Pr\{y_m = l | r_m; \boldsymbol{\theta}^{(\eta)}\} \times \left[\prod_{\substack{j=1, \\ j \neq m}}^{|\Gamma|} \underbrace{\left(\sum_{y_j=1}^C \Pr\{y_j | r_j; \boldsymbol{\theta}^{(\eta)}\} \right)}_1 \right] \\
&= \sum_{\forall(i,j) \in \Gamma} \Pr\{y_{i,j} = l | r_{i,j}; \boldsymbol{\theta}^{(\eta)}\} = \sum_{\forall(i,j) \in \Gamma} P_{i,j,l}^{(\eta)}. \tag{5.86}
\end{aligned}$$

Note that the first equality in (5.86) is a consequence of the following result:

$$\begin{aligned} \Pr\{\mathbf{y}|\mathbf{r};\boldsymbol{\theta}\} &= \Pr\{y_{|\Gamma|}|r_1, \dots, r_{|\Gamma|};\boldsymbol{\theta}\} \times \prod_{m=1}^{|\Gamma|-1} \Pr\{y_m|y_{m+1}, \dots, y_{|\Gamma|}, r_1, \dots, r_{|\Gamma|};\boldsymbol{\theta}\} \\ &= \prod_{m=1}^{|\Gamma|} \Pr\{y_m|r_m;\boldsymbol{\theta}\}. \end{aligned} \quad (5.87)$$

Similarly, we obtain

$$\mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \left\{ \sum_{\forall(i,j) \in \Gamma} (r_{i,j} - d_{i,j}) \delta(y_{i,j} - l) \right\} = \sum_{\forall(i,j) \in \Gamma} (r_{i,j} - d_{i,j}) P_{i,j,l}^{(\eta)}, \quad (5.88)$$

and

$$\mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} \left\{ \sum_{\forall(i,j) \in \Gamma} (r_{i,j} - d_{i,j})^2 \delta(y_{i,j} - l) \right\} = \sum_{\forall(i,j) \in \Gamma} (r_{i,j} - d_{i,j})^2 P_{i,j,l}^{(\eta)}. \quad (5.89)$$

respectively in the second and third parts of the computation. It is easy to verify, after some tedious manipulations, that

$$\begin{aligned} Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)}) &= \boldsymbol{\psi}^T(\boldsymbol{\theta}_e) \mathbb{E}_{p(\mathbf{y}|\mathbf{r};\boldsymbol{\theta}^{(\eta)})} [\mathbf{s}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})] \\ &= \underbrace{\sum_{l=1}^C \sum_{\forall(i,j) \in \Gamma} \ln(\alpha_l) P_{i,j,l}^{(\eta)}}_{Q_0^{(\eta)}(\alpha_1, \alpha_2, \dots, \alpha_C)} + \underbrace{\sum_{l=1}^C \sum_{\forall(i,j) \in \Gamma} - \left[\frac{(r_{i,j} - d_{i,j} - \mu_l)^2}{2\sigma_l^2} + \frac{1}{2} \ln(\sigma_l^2) \right] P_{i,j,l}^{(\eta)}}_{Q_l^{(\eta)}(\mu_l, \sigma_l^2, \boldsymbol{\theta}_p)}. \end{aligned} \quad (5.90)$$

Chapter 6

Conclusions and Ongoing Work

6.1 Conclusions

This thesis dealt with wireless localization in harsh mixed line-of-sight and non-line-of-sight (LOS/NLOS) environments. Various measurement campaigns have proven the adequacy of a mode-dependent (LOS mode or NLOS mode) measurement error model. To take into account the fact that non-line-of-sight (NLOS) identification is usually infeasible, a simplified two-mode mixture distribution was adopted to represent the actual measurement error. To be realistic as much as possible, we further assume that offline calibration is either infeasible or can be only coarsely conducted. Based on these, various new iterative algorithms have been developed to jointly estimate the positions, measurement error statistics (probability density function (pdf) or mixture model parameters in this thesis). More precisely, new algorithms were introduced for non-cooperative localization in wireless cellular radio networks in Chapter 3 and Chapter 4 and for cooperative localization in wireless sensor networks (WSNs) in Chapter 5.

In Chapter 3, we assumed that the measurement error distribution is completely unknown. An iterative algorithm, termed the RIN algorithm, was introduced to approximate the maximum likelihood position estimator. The RIN algorithm alternates between a pdf estimation step, which approximates the actual measurement error distribution (albeit unknown) non-parametrically via the adaptive kernel density estimation, and a parameter estimation step, which attempts to resolve a position estimate from the approximated log-likelihood function via a quasi-Newton method. Unless the convergence condition is satisfied, the resolved position estimate is then used to refine the pdf estimation on the next iteration. We also presented the best achievable localization accuracy with the aid of Cramér-Rao lower bound (CRLB) analysis. Various simulations were conducted, and the results revealed the following. When the number of the distance measurements is large, the RIN position estimator tends to attain the performance of the maximum likelihood (ML) estimator that ideally assumes known measurement error distribution. When the number of the distance measurements is small, it deviates from the ML estimator but still outperforms several salient robust estimators in terms of the localization accuracy. However, the improvement in the localization accuracy comes at the cost of higher computational complexity— $\mathcal{O}((NK)^2)$ FLOPs per iterations.

In Chapter 4, the measurement error distribution is represented parametrically by finite-mode Gaussian mixtures. We jointly estimate the positions and Gaussian mixture model parameters. In order to approximate the ML estimator in a computational profitable manner, two iterative algorithms, namely an expectation-conditional maximization (ECM) algorithm and a joint maximum *a posteriori* (JMAP)-ML algorithm, were developed based on a complete data set. Analogous to the RIN algorithm, both algorithms alternate between a position estimation step and a parametric pdf estimation step until some convergence condition is met. To give a parameter estimate, the ECM algorithm adopted “soft fusion” of information, while the JMAP-ML algorithm adopted “hard fusion” of information. Both algorithms are simple to implement as was shown in several examples, and they guarantee to converge as per proofs. The ECM algorithm is more favorable to use as it is capable of reproducing the ML estimator, given a good starting point. Although the JMAP-ML algorithm can only generate a biased estimator, it still serves as a reasonable approximation of the ML estimator. Moreover, we presented the best achievable localization accuracy with the aid of CRLB analysis. Various simulations have been conducted to test the proposed algorithms, and the results confirmed that both the ECM algorithm and JMAP-ML algorithm are able to approximate the ML estimator well but with reduced computational complexity— $\mathcal{O}(CNK)$ FLOPs per iteration—as compared to the RIN algorithm. Two model mismatch problems were also studied in the simulations. Despite some performance degradation, the proposed algorithms still presented good localization accuracy and considerable robustness against the NLOS measurements.

In Chapter 5, the parametric modeling of the measurement error distribution using Gaussian mixtures was again adopted, but in the context of cooperative localization in WSNs. The ECM criterion was first used to approximate the ML estimator of the agent positions and Gaussian mixture parameters in a centralized manner. The resulting centralized ECM algorithms lead to easier inference tasks and meanwhile retain several convergence properties with a proof of the “space filling” condition. To meet the scalability requirement of large-scale WSNs, we further developed two distributed ECM algorithms. Both the centralized- and distributed ECM algorithms were analyzed systematically in terms of the computational complexity and communication overhead (or energy consumption equivalently). The computational complexity of the distributed ECM algorithms turned out to be low, and the communication overhead is linear in terms of the number of agents. In addition, localization accuracy of the proposed ECM algorithms is evaluated with both the simulation data and real data. The results pin down that they are superior to several salient competitors for sensor network localization in harsh environments and thus are appropriate for a wide class of wireless sensor network problems.

6.2 Ongoing Work

We present some ongoing work (collaborated with Yi Zhang, Di Jin, Dr. C. Fritsche, Prof. F. Gustafsson and Prof. A.M. Zoubir) in the following.

- **Ongoing Work I:** In Chapter 5, we introduced time-of-arrival (TOA) based centralized- and distributed ECM algorithms for cooperative localization in wireless sensor networks. This requires precise time-synchronization between all sensors. Alternatively, round-trip-time-of-arrival (RTOA) measurements can be used instead with less trouble. But still extra timing devices need to be embedded in every wireless sensor, which should be kept low-cost and light-weight. Instead, received-signal-strength (RSS) measurements are easier to use. In [78], we have proposed a centralized RSS-based EM algorithm. Restricted by the ad-hoc nature of a wireless sensor network, however, RSS based distributed algorithms are more demanding.
- **Ongoing Work II:** In Chapter 5, we briefly reviewed a few Bayesian cooperative localization algorithms that take advantage of the message passing techniques in graph models. The overwhelming advantage as compared to the non-Bayesian algorithms lies in their low sensitivity to an initial guess. The existing algorithms all assume perfect NLOS identification and full knowledge about the measurement error statistics. The former assumption is overoptimistic, although a few existing methods, for instance [21, 135], can provide rather accurate NLOS identification performance. The misclassified NLOS measurements, even if very few, will incur degraded localization results. The latter assumption is problematic when the wireless channel is time varying and repetitive offline calibration is infeasible or the sensor network is deployed in an unacquainted environment. Joint estimation of the unknown measurement error statistics (assumed to be deterministic) and sensor positions (assumed to be stochastic) is very challenging. Similar to the idea used for joint particle filtering (PF) and calibration of unknown noise statistics in [116, 136], we could alternately estimate the positions using message passing techniques and calibrate the measurement error statistics using average consensus techniques. Since the computational complexity and communication overhead are also critical to the algorithm design, robust parametric representation of the local belief messages and internal messages need to be developed under imperfect NLOS identification and inaccurate measurement error statistics in the end.

List of Acronyms

A-GPS	Assisted global positioning system
AKDE	adaptive kernel density estimation
AOA	angle-of-arrival
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BP	belief propagation
BS	base station
C-ECM	centralized expectation-conditional maximization
CM	conditional maximization
CRLB	Cramér-Rao lower bound
D-ECM	distributed expectation-conditional maximization
E-911	Enhanced-911
ECM	expectation-conditional maximization
EM	expectation-maximization
FCC	Federal Communications Commission
FLOPs	floating-point operations
GDOP	geometric dilution of precision
GEM	generalized expectation maximization
GMM	Gaussian mixture model
GPS	global positioning system
GS	grid search
HMM	hidden Markov model
IAD	identify and discard
iid	independent and identically distributed
IPPM	iterative parallel projection method

JMAP-ML	joint maximum <i>a posteriori</i> -maximum likelihood
KDE	kernel density estimation
KS test	Kolmogorov-Smirnov test
LMS	least-median-squares
LOS	line-of-sight
LP	linear programming
LS	least-squares
LSCV	least-squares cross-validation
MAD	median absolute deviation
MAP	maximum <i>a posteriori</i>
MC	Markov chain
MDS	multidimensional scaling
MISE	mean-integrated-square-error
ML	maximum likelihood
MMSE	minimum-mean-square-error
MRF	Markov random fields
MS	mobile station
MSE	mean-square-error
NBP	nonparametric belief propagation
NLOS	non-line-of-sight
pdf	probability density function
PDP	power delay profile
PF	particle filtering
QN	quasi-Newton
QP	quadratic programming

RMSE	root-mean-square-error
RSS	received-signal-strength
RTOA	round-trip-time-of-arrival
SDP	semi-definite programming
SLAM	simultaneous localization and mapping
SPA	sum-product-algorithm
SPAWN	SPA over wireless network
SVD	singular value decomposition
SVM	support vector machine
TDOA	time-difference-of-arrival
TKDE	transformation kernel density estimation
TOA	time-of-arrival
UWB	ultra-wide-band
WLS	weighted-least-squares
WSN	wireless sensor network

List of Symbols

$\arg \max_x f(x)$	the x maximizing $f(x)$
$\arg \min_x f(x)$	the x minimizing $f(x)$
$\mathbf{X} \succeq \mathbf{Y}$	matrix $\mathbf{X} - \mathbf{Y}$ is positive semidefinite
$\mathbb{E}_{p(x)}\{\cdot\}$	expectation taken w.r.t. the pdf $p(x)$
$A \cup B$	union of set A and set B
$A \setminus B$	set-theoretic difference of A and B
\mathbb{Z}	set of integers
\mathbb{R}	set of real numbers
$[\cdot]^T$	transpose of a vector or matrix
$[\cdot]^{-1}$	inverse of a square matrix
$\ \cdot\ $	Euclidean norm of a vector
$ \cdot $	cardinality of a set or absolute value of a complex number
\sim	distributed as
$\stackrel{a}{\sim}$	asymptotically distributed as
\propto	proportional to
\in	element of a set
$\partial/\partial x$	partial derivative taken w.r.t. x
$\nabla_{\boldsymbol{\theta}}$	gradient operator
$\Delta_{\boldsymbol{\theta}}$	Laplace operator
$\ln(\cdot)$	natural logarithm
$\log(\cdot)$	logarithm to the base 10
$\exp[\cdot]$	exponential function
$\sqrt{[\cdot]}$	square root of a scalar or square matrix
$\text{tr}\{\cdot\}$	trace of a matrix
$\mathbf{0}$	vector of all zeros
$\mathbf{1}$	vector of all ones
\mathbf{I}_N	identity matrix of size $N \times N$
$\mathbf{1}_{M \times N}$	$M \times N$ matrix of all ones
$\mathcal{N}(v; \mu, \sigma^2)$	Gaussian distribution with mean μ and variance σ^2
$\mathcal{U}[v; a, b)$	uniform distribution on the interval $[a, b)$ with $a < b$
$\mathcal{R}(v; \gamma)$	Rayleigh distribution with parameter γ
$\mathcal{E}(v; \lambda)$	exponential distribution with parameter λ
$\boldsymbol{\theta}_p$	vector of unknown positions (General expression)

$\hat{\boldsymbol{\theta}}_p$	unbiased estimator of $\boldsymbol{\theta}_p$ (General expression)
$\hat{\boldsymbol{\theta}}_p^{\text{LS}}$	LS estimator/estimate of $\boldsymbol{\theta}_p$ (General expression)
$\hat{\boldsymbol{\theta}}_p^{\text{ML}}$	ML estimator/estimate of $\boldsymbol{\theta}_p$ (General expression)
$\boldsymbol{\theta}_p^{(\eta)}$	position estimate computed on the η th iteration of the proposed algorithms (General expression)
$\boldsymbol{\theta}$	vector of all unknown parameters (General expression)
Θ	parameter space of $\boldsymbol{\theta}$ (General expression)
$\hat{\boldsymbol{\theta}}^{\text{EM}}$	EM estimator/estimate of $\boldsymbol{\theta}$ (General expression)
$\hat{\boldsymbol{\theta}}^{\text{GEM}}$	GEM estimator/estimate of $\boldsymbol{\theta}$ (General expression)
$\hat{\boldsymbol{\theta}}^{\text{ECM}}$	ECM estimator/estimate of $\boldsymbol{\theta}$ (General expression)
$\hat{\boldsymbol{\theta}}^{\text{J}}$	JMAP-ML estimator/estimate of $\boldsymbol{\theta}$ (General expression)
$\hat{\boldsymbol{\theta}}^{\text{ML}}$	ML estimator/estimate of $\boldsymbol{\theta}$ (General expression)
$\hat{\boldsymbol{\theta}}^{\text{LS}}$	LS estimator/estimate of $\boldsymbol{\theta}$ (General expression)
$\boldsymbol{\theta}^{(0)}$	an initial guess of $\boldsymbol{\theta}$ (General expression)
$\boldsymbol{\theta}^{(\eta)}$	parameter estimate computed on the η th iteration of the proposed algorithms (General expression)
$\boldsymbol{\theta}_a$	vector of auxiliary environmental parameters (General expression)
$\boldsymbol{\theta}_e$	vector of mixture model parameters (General expression)
$\boldsymbol{\theta}_e^{(0)}$	initial guess of $\boldsymbol{\theta}_e$ (Chapter 5)
\mathbf{r}	measurement vector (General expression)
$\mathbf{h}(\boldsymbol{\theta}_p, \boldsymbol{\theta}_a)$	vector function of $\boldsymbol{\theta}_p$ and $\boldsymbol{\theta}_a$ (General expression)
\mathbf{v}	vector of measurement error terms that follow a certain pdf $p_V(v)$ (General expression)
\mathbf{y}	vector of missing data that corresponds to \mathbf{r} (General expression)
\mathbf{z}	vector of complete data, i.e., $\mathbf{z} = \{\mathbf{y}, \mathbf{r}\}$ (General expression)
$p(\mathbf{r}; \boldsymbol{\theta}_p)$	likelihood function of $\boldsymbol{\theta}_p$ given \mathbf{r} (General expression)
$p(\mathbf{r}; \boldsymbol{\theta})$	incomplete-data likelihood function of $\boldsymbol{\theta}$ (General expression)
$p(\mathbf{r}, \mathbf{y}; \boldsymbol{\theta})$	complete-data likelihood function of $\boldsymbol{\theta}$ (General expression)
$\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$	incomplete-data log-likelihood function of $\boldsymbol{\theta}$ (General expression)
$\mathcal{L}_C(\boldsymbol{\theta}; \mathbf{y}, \mathbf{r})$	complete-data log-likelihood function of $\boldsymbol{\theta}$ (General expression)
$\mathcal{F}(\boldsymbol{\theta})$	Fisher's information matrix of $\boldsymbol{\theta}$ (General expression)

$\mathcal{F}(\boldsymbol{\theta}_p)$	Fisher's information matrix of $\boldsymbol{\theta}_p$ (General expression)
\mathcal{I}_v	intrinsic accuracy (General expression)
c_0	propagation speed of a radio wave (General expression)
v_L	measurement error term measured under LOS condition (General expression)
v_{NL}	measurement error term measured under NLOS condition (General expression)
$p(v_L)$	pdf of v_L (General expression)
$p(v_{NL})$	pdf of v_{NL} (General expression)
μ_L	mean of $p(v_L)$ (General expression)
σ_L^2	variance of $p(v_L)$ (General expression)
μ_{NL}	mean of $p(v_{NL})$ (General expression)
σ_{NL}^2	variance of $p(v_{NL})$ (General expression)
α_L	prior probability of LOS occurrence (General expression)
α_{NL}	prior probability of NLOS occurrence (General expression)
$p_V^{(L)}(v; \boldsymbol{\beta}_L)$	pdf of v conditioning on LOS propagation (General expression)
$p_V^{(NL)}(v; \boldsymbol{\beta}_{NL})$	pdf of v conditioning on NLOS propagation (General expression)
$\boldsymbol{\beta}_L$	vector of parameters that describes $p_V^{(L)}(v; \boldsymbol{\beta}_L)$ (General expression)
$\boldsymbol{\beta}_{NL}$	vector of parameters that describes $p_V^{(NL)}(v; \boldsymbol{\beta}_{NL})$ (General expression)
$p_V(v)$	underlying measurement error distribution (General expression)
C	total number of mixture components in the parametric model of $p_V(v)$ (General expression)
α_l	the l th mixing coefficient (General expression)
$p_V^{(l)}(v; \boldsymbol{\beta}_l)$	the l th mixture component in the parametric model of $p_V(v)$ (General expression)
$\hat{p}_V(v)$	estimate of $p_V(v)$ built either parametrically or non-parametrically (General expression)
P_T (dBm)	transmit power of a target node (General expression)
A_L (dB)	path loss at a reference distance under LOS condition (General expression)
B_L (dB)	path loss exponent under LOS condition (General expression)

A_{NL} (dB)	path loss at a reference distance under NLOS condition (General expression)
B_{NL} (dB)	path loss exponent under NLOS condition (General expression)
λ	Lagrange multiplier (General expression)
N_{itr}	maximum number of iterations (General expression)
Δ	convergence tolerance (General expression)
N_M	total number of samples generate for use in the Monte Carlo integration (General expression)
L^*	some limiting (stationary) point of $\mathcal{L}_I(\boldsymbol{\theta}; \mathbf{r})$ (General expression)
$Q(\boldsymbol{\theta}; \boldsymbol{\theta}^{(\eta)})$	conditional expectation of the complete-data log-likelihood function given $\boldsymbol{\theta}^{(\eta)}$ (General expression)
$\boldsymbol{\theta}^{(\eta+s/S)}$	ECM estimate or JMAP-ML estimate of $\boldsymbol{\theta}$ computed on the s th CM step of the $(\eta + 1)$ th iteration (General expression)
$\boldsymbol{\vartheta}_s^T$	sub-vector of $\boldsymbol{\theta}$ such that $\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \dots, \boldsymbol{\vartheta}_S^T]^T$ (General expression)
S	total number of sub-vectors in $\boldsymbol{\theta}$ (General expression)
$g_s(\boldsymbol{\theta})$	vector function of $\boldsymbol{\theta}$ (General expression)
$G_s(\boldsymbol{\theta})$	column space of the matrix $\nabla_{\boldsymbol{\theta}} g_s(\boldsymbol{\theta})$ (General expression)
$h_i(t)$	the i th wireless channel impulse response (Chapter 1)
τ_{TOA}	TOA measurement in general (Chapter 1)
τ_i	TOA measurement obtained from the i th wireless channel (Chapter 1)
P_{RSS}	RSS measurement in general (Chapter 1)
N	total number of reference nodes (BSs in Chapter 3 and Chapter 4 or anchors in Chapter 5)
K	total number of measurements observed at each BS (Chapter 3 and Chapter 4)
x	x-coordinate of an MS to be located (Chapter 3 and Chapter 4)
y	y-coordinate of an MS to be located (Chapter 3 and Chapter 4)
x_i	x-coordinate of the i th BS (Chapter 3 and Chapter 4) or the i th sensor (Chapter 5)
y_i	y-coordinate of the i th BS (Chapter 3 and Chapter 4) or the i th sensor (Chapter 5)
\mathbf{p}_i	2-D position of the i th BS (Chapter 3 and Chapter 4) or the i th sensor (Chapter 5)

$\mathbf{p}_i^{(0)}$	initial guess of \mathbf{p}_i (Chapter 3, 4, 5)
$r_{i,k}$	distance measurement (TOA measurement multiplied by c_0) the i th BS obtained at the k th sampling time instance (Chapter 3 and Chapter 4)
$d_i(\boldsymbol{\theta}_p)$	actual Euclidean distance between the MS and the i th BS (Chapter 3 and Chapter 4)
$v_{i,k}$	measurement error in $r_{i,k}$ (Chapter 3 and Chapter 4)
$\tilde{\mathbf{r}}$	vector of measurements re-defined for the linearized signal model in the robust semi-parametric algorithm (Chapter 3)
$\tilde{\mathbf{v}}$	vector of measurement error terms re-defined for the linearized signal model in the robust semi-parametric algorithm (Chapter 3)
$\hat{\tilde{\mathbf{v}}}$	vector of residuals extracted from the linearized signal model in the robust semi-parametric algorithm (Chapter 3)
\mathbf{u}	vector obtained after applying a nonlinear transformation of $\hat{\tilde{\mathbf{v}}}$ in the TKDE (Chapter 3)
$\hat{p}_{\tilde{v}}(\tilde{v})$	estimate of $p_V(v)$ obtained from the TKDE (Chapter 3)
$\hat{p}'_{\tilde{v}}(\tilde{v})$	partial derivative of $\hat{p}_{\tilde{v}}(\tilde{v})$ taken w.r.t. \tilde{v} (Chapter 3)
$\hat{\boldsymbol{\theta}}_p^{\text{LS}}$	LS estimate obtained for the linearized signal model, whose first two entries serve as the initial guess of the RIN algorithm (Chapter 3)
$\hat{p}_V^{(\eta)}(v)$	estimate of $p_V(v)$ obtained via the AKDE on the η th RIN iteration (Chapter 3)
$\mathcal{L}^{(\eta)}(\boldsymbol{\theta}_p)$	approximated log-likelihood function of $\boldsymbol{\theta}_p$ obtained on the η th RIN iteration (Chapter 3)
$\boldsymbol{\theta}_p^{(\eta,j)}$	position estimate on the j th quasi-Newton iteration of the η th RIN algorithm (Chapter 3)
$\hat{\boldsymbol{\theta}}_p^{\text{RIN}}$	RIN position estimator (Chapter 3)
$\hat{\mathbf{v}}$	vector of residuals extracted from the original nonlinear signal model (Chapter 3)
\hat{v}_m	the m th element of vector $\hat{\mathbf{v}}$ (Chapter 3)
$\hat{p}_0(v)$	pilot density estimate of $p_V(v)$ firstly constructed in the AKDE (Chapter 3)
$\mathcal{K}_G(\cdot)$	standard Gaussian kernel (Chapter 3)
w	window width of a kernel density estimate (Chapter 3)
w_0	window width set empirically in the pilot density estimate (Chapter 3)

λ_m	the m th local bandwidth of the adaptive kernel density estimate (Chapter 3)
β	sensitivity parameter selected to give a good kernel density estimate (Chapter 3)
$M_0(w)$	score function from minimizing which a good window width w is solved (Chapter 3)
$\mathcal{H}(\boldsymbol{\theta}_p)$	gradient of $\mathbf{h}(\boldsymbol{\theta}_p)$ taken w.r.t. $\boldsymbol{\theta}_p$ (Chapter 3)
η_{eff}	re-defined estimation efficiency of a position estimator (Chapter 3)
γ_{NL}	Rayleigh distribution parameter under the NLOS condition (Chapter 3)
\mathbf{e}_j	the j th orthogonal basis vector (Chapter 4 and Chapter 5)
c_j	scalar real coefficient of \mathbf{e}_j (Chapter 4 and Chapter 5)
$\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p), \mathbf{y})$	sufficient statistics for $\boldsymbol{\theta}_e$ if $\boldsymbol{\theta}_p$ is known (Chapter 4 and Chapter 5)
$\mathbf{t}(\tilde{\mathbf{r}}(\boldsymbol{\theta}_p^{(\eta)}), \mathbf{y})$	approximated sufficient statistics for $\boldsymbol{\theta}_e$, given $\boldsymbol{\theta}_p^{(\eta)}$ (Chapter 4 and Chapter 5)
$\mathbf{T}(\boldsymbol{\theta}_p^{(\eta)})$	conditional expectation of the approximated sufficient statistic for $\boldsymbol{\theta}_e$, given $\boldsymbol{\theta}_p^{(\eta)}$ on the $(\eta + 1)$ th EM iteration (Chapter 4 and Chapter 5)
$P_{i,k,l}^{(\eta)}$	short-hand notation of $\Pr \{y_{i,k} = l r_{i,k}; \boldsymbol{\theta}^{(\eta)}\}$ (Chapter 4)
$\mathbf{y}^{(\eta)}$	MAP estimate of the latent variables \mathbf{y} obtained on the η th JMAP-ML iteration (Chapter 4)
$w_{i,k,l}^{(\eta)}$	weighting factor which takes $P_{i,k,l}^{(\eta)}$ in the ECM algorithm or $\delta(l - y_{i,k}^{(\eta+1)})$ in the JMAP-ML algorithm (Chapter 4)
$\Lambda^{(\eta)}(\boldsymbol{\theta})$	general cost function from which an updated parameter estimate $\boldsymbol{\theta}^{(\eta+1)}$ is found either by the ECM algorithm or the JMAP-ML algorithm (Chapter 4)
$\mathcal{K}_E^{(l)}(v; \boldsymbol{\beta}_l)$	the l th mixture component that belongs to exponential family (Chapter 4)
N_u	total number of agents in a wireless sensor network (Chapter 5)
\mathcal{N}_a	set of indices of all agents (Chapter 5)
\mathcal{N}_b	set of indices of all anchors (Chapter 5)
$r_{i,j}$	distance measurement obtained at sensor i in cooperation with sensor j (Chapter 5)
$d(\mathbf{p}_i, \mathbf{p}_j)$	Euclidean distance between sensor i and sensor j (Chapter 5)
$v_{i,j}$	measurement error term in $r_{i,j}$ (Chapter 5)

R_c	maximal communication range of a wireless sensor (Chapter 5)
$\mathcal{H}(i)$	set of all neighboring sensors of agent i (Chapter 5)
Γ	set of all sensor pairs that contribute distance measurements (Chapter 5)
$p(\mathbf{p}_i)$	prior probability of \mathbf{p}_i assumed in the NBP algorithm (Chapter 5)
$B_i(\mathbf{p}_i)$	local belief message of \mathbf{p}_i in the NBP algorithm (Chapter 5)
$m_{ji}(\mathbf{p}_i)$	internal message sent from sensor j to sensor i in the NBP algorithm (Chapter 5)
$B_i^{(\eta)}(\mathbf{p}_i)$	estimate of $B_i(\mathbf{p}_i)$ on the η th iteration of the NBP algorithm (Chapter 5)
$m_{ji}^{(\eta)}(\mathbf{p}_i)$	estimate of $m_{ji}(\mathbf{p}_i)$ on the η th iteration of the NBP algorithm (Chapter 5)
$\{w_i^{(\eta,\iota)}, \mathbf{x}_i^{(\eta,\iota)}\}$	weighting factors and particles for representing $B_i^{(\eta)}(\mathbf{p}_i)$ (Chapter 5)
$\{w_{ij}^{(\eta,\iota')}, \mathbf{x}_{ij}^{(\eta,\iota')}\}$	weighting factors and particles for representing $m_{ji}^{(\eta)}(\mathbf{p}_i)$ (Chapter 5)
R_{int}	total number of particles for representing an internal message (Chapter 5)
R_{bel}	total number of particles for representing a belief message (Chapter 5)
C_{bel}	total number of parameters used in a parametric SPAWN algorithm and to be determined accordingly (Chapter 5)
$P_{i,j,l}^{(\eta)}$	short-hand notation of $\Pr\{y_{i,j} = l r_{i,j}; \boldsymbol{\theta}^{(\eta)}\}$ (Chapter 5)
$\tilde{P}_{i,j,l}^{(\eta)}$	local approximate of $P_{i,j,l}^{(\eta)}$ computed at agent i (Chapter 5)
N_{itr}^g	total number of gossip rounds in an average consensus algorithm (Chapter 5)
N_{itr}^{nu}	total number of Newton iterations (Chapter 5)
R_{grid}	total number of grid points used for the local search of $\boldsymbol{\theta}_p$ in the distributed ECM algorithms (Chapter 5)
$\hat{\boldsymbol{\theta}}_t^{\text{CECM}}$	centralized ECM estimator of $\boldsymbol{\theta}$ at the sampling time instance t (Chapter 5)
$\hat{\boldsymbol{\theta}}_t^{\text{DECM}}$	distributed ECM estimator of $\boldsymbol{\theta}$ at the sampling time instance t (Chapter 5)
$\boldsymbol{\theta}_{t,e}$	vector of unknown mixture model parameters at the sampling time instance t (Chapter 5)

$\boldsymbol{\theta}_{t,p}$	vector of unknown agent positions at the sampling time instance t (Chapter 5)
$\boldsymbol{\theta}_t$	vector of all unknown parameters at the sampling time instance t (Chapter 5)
$\boldsymbol{\theta}_t^{(0)}$	initial guess of $\boldsymbol{\theta}_t$ (Chapter 5)
$\boldsymbol{\theta}_{t,e}^{(\eta)}$	ECM estimate of $\boldsymbol{\theta}_{t,e}$ computed on the η th iteration (Chapter 5)
$\boldsymbol{\theta}_{t,p}^{(\eta)}$	ECM estimate of $\boldsymbol{\theta}_{t,p}$ computed on the η th iteration (Chapter 5)
$b(N_u)$	total number of transmitted bits (Chapter 5)
$h(N_u)$	average number of hops required for transmitting one bit to the destination (Chapter 5)
$e(N_u)$	average amount of energy required for transmitting one bit over one hop (Chapter 5)
$\mathcal{E}(N_u)$	total energy consumed for communicating data bits by a cooperative localization algorithm (Chapter 5)

Bibliography

- [1] A. H. Sayed, A. Tarighat, and N. Khajehnouri, “Network-based wireless location: challenges faced in developing techniques for accurate wireless location information,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 24–40, Jul. 2005.
- [2] E. D. Kaplan and C. J. Hegarty, *GPS: Principles and Applications*, Reading, MA: Artech House, 2006.
- [3] F. Gustafsson and F. Gunnarsson, “Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, Jul. 2005.
- [4] J. Rantakokko, J. Rydell, P. Stromback, P. Handel, J. Callmer, D. Tornqvist, F. Gustafsson, M. Jobs, and M. Gruden, “Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization,” *IEEE Wireless Commun. Mag.*, vol. 18, no. 2, pp. 10–18, April 2011.
- [5] M. Bjorkbom, J. Timonen, H. Yigitler, O. Kaltiokallio, J. M. V. Garcia, M. Myrsky, J. Saarinen, M. Korkalainen, C. Cuhac, R. Jantti, R. Virrankoski, J. Vankka, and H. N. Koivo, “Localization services for online common operational picture and situation awareness,” *IEEE Access*, vol. 1, pp. 742–757, 2013.
- [6] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero III, R. L. Moses, and N. S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks,” *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.
- [7] M. Allegra, “Where high-accuracy wireless location is (and isn’t) headed in 2014,” *Directions Magazine Online*, <http://www.directionsmag.com/articles/where-high-accuracy-wireless-location-is-and-isnt-headed-in-2014/377368>, Jan. 2014.
- [8] F. Gustafsson, *Statistical Sensor Fusion*, Studentlitteratur, Lund, Sweden, 2012.
- [9] A. Amar and A. J. Weiss, “Localization of narrowband radio emitters based on Doppler frequency shifts,” *IEEE Trans. Signal Process.*, vol. 56, no. 11, pp. 5500–5508, Nov. 2008.
- [10] A. J. Weiss, “Direct geolocation of wideband emitters based on delay and Doppler,” *IEEE Trans. Signal Process.*, vol. 59, no. 6, pp. 2513–2521, Jun. 2011.
- [11] F. Gustafsson and F. Gunnarsson, “Measurements used in wireless sensor networks localization,” in *Localization algorithms and strategies for wireless sensor networks*, pp. 33–53. Information Science Reference, Hershey, PA, USA, 2009.
- [12] C. Nerguizian, C. L. Despins, S. Affes, and M. Djadel, “Radio-channel characterization of an underground mine at 2.4 GHz,” *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2441–2453, Sep. 2005.
- [13] R. Zekavat and R. M. Buehrer, *Handbook of Position Location*, John Wiley & Sons, Inc., Hoboken, NJ, 2011.

- [14] H. Wymeersch, J. Lien, and M. Z. Win, "Cooperative localization in wireless networks," *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.
- [15] J. Caffery, *Wireless Location in CDMA Cellular Radio Systems*, Kluwer Academic Publisher, Norwell, Massachusetts, USA, 2002.
- [16] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 70–84, July 2005.
- [17] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, Cambridge University Press, 2005.
- [18] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Z. Win, "Ranging with ultrawide bandwidth signals in multipath environments," *IEEE Proc.*, vol. 97, no. 2, pp. 404–426, Feb 2009.
- [19] N. Patwari, A. O. Hero III, M. Perkins, N. S. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.
- [20] K. Yu and Y.-J. Guo, "Statistical NLOS identification based on AOA, TOA, and signal strength," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 274–286, Jan. 2009.
- [21] S. Marano, W. M. Gifford, H. Wymeersch, and M. Z. Win, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, 2010.
- [22] C. Fritsche, *Statistical Data Fusion for Hybrid Localization of Mobile Terminals*, Ph.D. thesis, Technische Universität Darmstadt, Darmstadt, Germany, Mar. 2011.
- [23] Y. Liu, Y.-H. Hu, and Q. Pan, "Distributed, robust acoustic source localization in a wireless sensor network," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4350–4359, Aug. 2012.
- [24] E. Damasso and L. M. Correia, *COST Action 231: Digital Mobile Radio Towards Future Generation Systems : Final Report*, European Commission, Brussels, Belgium, 1999.
- [25] P. C. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *Proc. IEEE Int. Conf. Wireless Commun. Networking (WCNC)*, New Orleans, LA, Sep. 1999, vol. 1, pp. 316–320.
- [26] M. McGuire, K. N. Plataniotis, and A. N. Venetsanopoulos, "Data fusion of power and time measurements for mobile terminal location," *IEEE Trans. Mobile Comput.*, vol. 4, no. 2, pp. 142–153, Mar./Apr. 2005.

- [27] F. Yin and A. M. Zoubir, “Robust positioning in NLOS environments using nonparametric adaptive kernel density estimation,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 3517–3520.
- [28] J.-F. Liao and B.-S. Chen, “Robust mobile location estimator with NLOS mitigation using interacting multiple model algorithm,” *IEEE Trans. Wireless Commun.*, vol. 5, no. 11, pp. 3002–3006, Nov. 2006.
- [29] C. Fritsche and A. Klein, “On the performance of mobile terminal tracking in urban GSM networks using particle filters,” in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Glasgow, Scotland, Aug. 2009, pp. 1953–1957.
- [30] U. Hammes and A. M. Zoubir, “Robust mobile terminal tracking in NLOS environments based on data association,” *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5872–5882, Nov. 2010.
- [31] U. Hammes and A. M. Zoubir, “Robust MT tracking based on M-estimation and interacting multiple model algorithm,” *IEEE Trans. Signal Process.*, vol. 59, no. 7, pp. 3398–3409, Jul. 2011.
- [32] U. Hammes, *Robust Positioning Algorithm for Wireless Networks*, Ph.D. thesis, Technische Universität Darmstadt, Darmstadt, Germany, Dec. 2009.
- [33] J. M. Huerta and J. Vidal, “Mobile tracking using UKF, time measures and LOS-NLOS expert knowledge,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Philadelphia, PA, USA, Mar. 2005, vol. 4, pp. 901–904.
- [34] L Cong and W Zhuang, “Nonline-of-sight error mitigation in mobile location,” *IEEE Trans. Wireless Commun.*, vol. 4, no. 2, pp. 560–573, Mar. 2005.
- [35] V. Savic, H. Wymeersch, and E. G. Larsson, “Simultaneous sensor localization and target tracking in mine tunnels,” in *Proc. Int. Conf. Information Fusion (FUSION)*, July 2013, pp. 1427–1433.
- [36] Y. Qi, *Wireless geolocation in a non-line-of-sight environment*, Ph.D. thesis, Princeton University, Princeton, NJ, USA, Nov. 2003.
- [37] B. W. Silverman, *Density Estimation: for Statistical and Data Analysis*, London: Chapman and Hall, 1986.
- [38] D. Clark and B.-N. Vo, “Convergence analysis of the Gaussian mixture PHD filter,” *IEEE Trans. Signal Process.*, vol. 55, no. 4, pp. 1204–1212, Apr. 2007.
- [39] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.
- [40] I. Guvenc and C. C. Chong, “A survey on TOA based wireless localization and NLOS mitigation techniques,” *Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, third quarter 2009.

- [41] Y. T. Chan and K. C. Ho, "A simple and efficient estimator for hyperbolic location," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 1905–1915, Aug. 1994.
- [42] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Hong Kong, China, Apr. 2003, vol. 6, pp. VI-553–556.
- [43] K. W. Cheung, H. C. So, W. K. Ma, and Y. T. Chan, "Least squares algorithms for time-of-arrival-based mobile location," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 1121–1130, Apr. 2004.
- [44] K. W. Cheung and H. C. So, "A multidimensional scaling framework for mobile location using time-of-arrival measurements," *IEEE Trans. Signal Process.*, vol. 53, no. 2, pp. 460–470, Feb. 2005.
- [45] H. C. So and F. K. W. Chan, "A generalized subspace approach for mobile positioning with time-of-arrival measurements," *IEEE Trans. Signal Process.*, vol. 55, no. 10, pp. 5103–5107, Oct. 2007.
- [46] H. C. So, Y. T. Chan, and F. K. W. Chan, "Closed-form formulae for time-difference-of-arrival estimation," *IEEE Trans. Signal Process.*, vol. 56, no. 6, pp. 2614–2620, Jun. 2008.
- [47] H.-W. Wei, Q. Wan, Z.-X. Chen, and S.-F. Ye, "A novel weighted multidimensional scaling analysis for time-of-arrival-based mobile location," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3018–3022, Jul. 2008.
- [48] H. C. So and L.-X. Lin, "Linear least squares approach for accurate received signal strength based source localization," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 4035–4040, Aug. 2011.
- [49] K. C. Ho, "Bias reduction for an explicit solution of source localization using TDOA," *IEEE Trans. Signal Process.*, vol. 60, no. 5, pp. 2101–2114, May 2012.
- [50] Y. T. Chan, W. Y. Tsui, H. C. So, and P. C. Ching, "Time-of-arrival based localization under NLOS conditions," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 17–24, Jan. 2006.
- [51] J. Riba and A. Urruela, "A non-line-of-sight mitigation technique based on ML-detection," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Quebec, Canada, May 2004, vol. 2, pp. 153–156.
- [52] Y. H. Qi, H. Kobayashi, and H. Suda, "Analysis of wireless geolocation in a non-line-of-sight environment," *IEEE Trans. Wireless Commun.*, vol. 5, no. 3, pp. 672–681, Mar. 2006.
- [53] X. Wang, Z. X. Wang, and B. O’Dea, "A TOA-based location algorithm reducing the errors due to non-line-of-sight (NLOS) propagation," *IEEE Trans. Veh. Technol.*, vol. 52, no. 1, pp. 112–116, Jan. 2003.

- [54] S. Venkatesh and R. M. Buehrer, "A linear programming approach to NLOS error mitigation in sensor networks," in *Proc. IEEE Int. Conf. Information Processing in Sensor Networks (IPSN)*, Nashville, TN, USA, Apr. 2006, pp. 301–308.
- [55] H. Y. Chen, G. Wang, Z. Z. Wang, H. C. So, and H. V. Poor, "Non-line-of-sight node localization based on semi-definite programming in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 108–116, Jan. 2012.
- [56] R. A. Maronna, R. D. Martin, and V. J. Yohai, *Robust Statistics: Theory and Methods*, John Wiley & Sons Ltd., Chichester, England, 2006.
- [57] P. J. Huber and E. M. Ronchetti, *Robust Statistics*, John Wiley & Sons Ltd., Hoboken, NJ, 2009.
- [58] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *Proc. IEEE Int. Symp. Information Processing in Sensor Networks (IPSN)*, Los Angeles, CA, Apr. 2005, pp. 91–98.
- [59] R. Casas, A. Marco, J. J. Guerrero, and J. Falcó, "Robust estimator for non-line-of-sight error mitigation in indoor localization," *EURASIP J. Appl. Signal Process.*, vol. 2006, no. 1, pp. 1–8, Jan. 2006.
- [60] G. L. Sun and W. Guo, "Bootstrapping M-estimators for reducing errors due to non-line-of-sight (NLOS) propagation," *IEEE Commun. Lett.*, vol. 8, no. 8, pp. 509–510, Aug. 2004.
- [61] A. M. Zoubir and D. R. Iskander, *Bootstrap Techniques for Signal Processing*, Cambridge University Press, Cambridge, U.K., 2004.
- [62] A. M. Zoubir and B. Boashash, "The bootstrap and its application in signal processing," *IEEE Signal Process. Mag.*, vol. 15, no. 1, pp. 56–76, Jan. 1998.
- [63] A. M. Zoubir and D. R. Iskander, "Bootstrap methods and applications," *IEEE Signal Process. Mag.*, vol. 24, no. 4, pp. 10–19, Jul. 2007.
- [64] U. Hammes, E. Wolsztynski, and A. M. Zoubir, "Robust tracking and geolocation for wireless networks in NLOS environments," *IEEE J. Sel. Topics. Signal Process.*, vol. 3, no. 5, pp. 889–901, Oct. 2009.
- [65] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "TOA-based robust wireless geolocation and Cramer-Rao lower bound analysis in harsh LOS/NLOS environments," *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2243–2255, May 2013.
- [66] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "Received signal strength-based joint parameter estimation algorithm for robust geolocation in LOS/NLOS environments," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013, pp. 6471–6475.

- [67] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "EM- and JMAP-ML based joint estimation algorithms for robust wireless geolocation in mixed LOS/NLOS environments," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 168–182, Jan. 2014.
- [68] X. Li, "Collaborative localization with received-signal strength in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 56, no. 6, pp. 3807–3817, 2007.
- [69] Y. Shang, W. Rumi, Y. Zhang, and M. Fromherz, "Localization from connectivity in sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 11, pp. 961–974, Nov. 2004.
- [70] X. Ji and H.-Y. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *INFOCOM 2004: Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, Mar. 2004, vol. 4, pp. 2652–2661.
- [71] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed weighted-multidimensional scaling for node localization in sensor networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 39–64, Feb. 2006.
- [72] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Trans. Sen. Netw.*, vol. 2, no. 2, pp. 188–220, May 2006.
- [73] R. M. Vaghefi, M. R. Gholami, R. M. Buehrer, and E. G. Strom, "Cooperative received signal strength-based sensor localization with unknown transmit powers," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1389–1403, Mar. 2013.
- [74] T. Jia and R. M. Buehrer, "Collaborative position location with NLOS mitigation," in *Proc. IEEE Int. Workshop Personal, Indoor and Mobile Radio Communications*, Istanbul, Turkey, Sep. 2010, pp. 267–271.
- [75] T. Jia and R. M. Buehrer, "Collaborative position location for wireless networks using iterative parallel projection method," in *Proc. IEEE Int. Global Telecommunications Conf. (GLOBECOMM)*, Miami, Florida, USA, Dec 2010, pp. 1–6.
- [76] T. Jia and R. M. Buehrer, "A set-theoretic approach to collaborative position location for wireless networks," *IEEE Trans. Mobile Comput.*, vol. 10, no. 9, pp. 1264–1275, Sep. 2011.
- [77] J. N. Ash and R. L. Moses, "Outlier compensation in sensor network self-localization via the EM algorithm," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Philadelphia PA, USA, Mar. 2005, vol. 4, pp. 749–752.
- [78] F. Yin, A. Li, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "RSS-based sensor network localization in contaminated Gaussian measurement noise," in *Proc. IEEE Int. Workshop Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Saint Martin, France, Dec. 2013, pp. 121–124.

- [79] F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, "Robust cooperative sensor network localization via the EM criterion in LOS/NLOS environments," in *Proc. IEEE Int. Workshop Signal Processing Advances in Wireless Communications (SPAWC)*, Darmstadt, Germany, June 2013, pp. 505–509.
- [80] F. Yin, C. Fritsche, D. Jin, F. Gustafsson, and A. M. Zoubir, "Cooperative localization in WSNs using Gaussian mixture modeling: Distributed ECM algorithms," *submitted to IEEE Trans. Signal Process.*, Apr. 2014.
- [81] A. T. Ihler, J. W. Fisher III, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 809–819, Apr. 2005.
- [82] V. Savic and S. Zazo, "Nonparametric belief propagation based on spanning trees for cooperative localization in wireless sensor networks," in *Proc. IEEE Int. Conf. Veh. Technol.*, 2010, pp. 1–5.
- [83] Panagiotis-Agis Oikonomou-Filandras and Kai-Kit Wong, "Hybrid non-parametric belief propagation for localization in wireless networks," in *Proc. Sensor Signal Processing for Defence (SSPD)*, 2011, pp. 1–5.
- [84] J. Lien, U. J. Ferner, W. Srichavengsup, H. Wymeersch, and M. Z. Win, "A comparison of parametric and sample-based message representation in cooperative localization," *International Journal of Navigation and Observation*, vol. 2012, pp. 1–10, 2012.
- [85] K. Das and H. Wymeersch, "Censoring for bayesian cooperative positioning in dense wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 9, pp. 1835–1842, 2012.
- [86] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, San Francisco, CA: Morgan Kaufmann, 2nd edition, 1988.
- [87] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [88] G. Mao, B. Fidan, and B. D. O. Anderson, "Wireless sensor network localization techniques," *Comput. Networks J.*, vol. 51, no. 10, pp. 2529–2553, 2007.
- [89] M. Welling and J. J. Lim, "A distributed message passing algorithm for sensor localization," in *Proc. Int. Conf. Artificial neural networks*, Berlin, Heidelberg, 2007, pp. 767–775, Springer-Verlag.
- [90] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.
- [91] E. L. Lehmann and G. Casella, *Theory of point estimation*, Springer-Verlag, Inc., New York, NY, 1998.
- [92] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science+Business Media, LLC, New York, NY, USA, 2006.

- [93] A. P. Dempster, N. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc. B*, vol. 39, pp. 1–38, 1977.
- [94] C. F. J. Wu, "On the convergence properties of the EM algorithm," *Ann. Statist.*, vol. 11, pp. 95–103, 1983.
- [95] X.-L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.
- [96] G. J. McLachlan and T. Krishnan, *The EM algorithm and extensions*, John Wiley & Sons, Inc., 1997.
- [97] A. Yeredor, "The joint MAP-ML criterion and its relation to ML and to extended least-squares," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3484–3492, Dec. 2000.
- [98] The Federal Communications Commission, "Enhanced 9-1-1 - wireless services," [Online Available:] <http://transition.fcc.gov/pshs/services/911-services/enhanced911/Welcome.html>, (accessed on 27. 04. 2014).
- [99] J. Caffery and G. L. Stuber, "Overview of radiolocation in CDMA cellular systems," *IEEE Commun. Mag.*, vol. 36, no. 4, pp. 38–45, Apr. 1998.
- [100] U. Hammes, E. Wolsztynski, and A. M. Zoubir, "Transformation-based robust semiparametric estimation," *IEEE Signal Process. Lett.*, vol. 15, pp. 845–848, 2008.
- [101] A. M. Zoubir and R. F. Breich, "Multiuser detection in heavy tailed noise," *Digital Signal Process.*, vol. 12, no. 2-3, pp. 262–273, 2002.
- [102] A. Beck, P. Stoica, and J. Li, "Exact and approximate solutions of source localization problems," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1770–1778, May 2008.
- [103] C. T. Kelley, *Solving nonlinear equations with Newton's method*, Fundamentals of Algorithms. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003.
- [104] R. Fletcher, *Practical methods of optimization (2nd ed.)*, John Wiley & Sons Ltd., Chichester, England, 1987.
- [105] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, Inc., New York, NY, 2001.
- [106] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer-Verlag, Inc., New York, NY, 1999.
- [107] G. Hendeby, *Performance and implementation aspects of nonlinear filtering*, Ph.D. thesis, Linköping University, Linköping, Sweden, Feb. 2008.
- [108] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer-Verlag, New York, NY, 1999.

- [109] N. Levanon, “Lowest GDOP in 2-D scenarios,” *Proc. IEE Radar, Sonar and Navigation*, vol. 147, no. 3, pp. 149–155, Jun. 2000.
- [110] I. S. Abramson, “On bandwidth variation in kernel estimates—a square root law,” *Ann. Statist.*, vol. 10, no. 4, pp. 1217–1223, 1982.
- [111] M. Rudemo, “Empirical choice of histograms and kernel density estimators,” *Scand. J. Statist.*, vol. 9, no. 2, pp. 65–78, 1982.
- [112] A. W. Bowman, “An alternative method of cross-validation for the smoothing of density estimates,” *Biometrika*, vol. 71, no. 2, pp. 353–360, 1984.
- [113] H. L. Van Trees, *Optimum Array Processing. Part IV of Detection, Estimation, and Modulation Theory*, John Wiley & Sons, Inc., New York, USA, 2002.
- [114] G. Strang, *Linear Algebra and Its Applications*, Cengage Learning, Boston, Massachusetts, USA, 4th edition, 2005.
- [115] R. Neal and G. E. Hinton, “A view of the EM algorithm that justifies incremental, sparse, and other variants,” in *Learning in Graphical Models*. 1998, pp. 355–368, Kluwer Academic Publishers.
- [116] E. Ozkan, C. Fritsche, and F. Gustafsson, “Online EM algorithm for joint state and mixture measurement noise estimation,” in *Proc. IEEE Int. Conf. Information Fusion*, Singapore, Jul. 2012, pp. 1935–1940.
- [117] J. A. Bilmes, “A gentle tutorial on the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models,” Tech. Rep., University of California, Berkeley, Berkeley, CA, USA, 1998.
- [118] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, NY, USA, 2004.
- [119] T. Minka, “The lightspeed matlab toolbox,” [Online Available:] <http://research.microsoft.com/en-us/um/people/minka/software/lightspeed/>, (accessed on 27. 04. 2014).
- [120] S. Korkmaz and A. J. van der Veen, “Robust localization in sensor networks with iterative majorization techniques,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Taipei, China, Apr. 2009, pp. 2049–2052.
- [121] J. Lien, “A framework for cooperative localization in ultrawideband wireless networks,” M.S. thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, Mass., USA, 2007.
- [122] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, “Gossip algorithms for distributed signal processing,” *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, 2010.
- [123] R. D. Nowak, “Distributed EM algorithms for density estimation and clustering in sensor networks,” *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2245–2253, Aug. 2003.

- [124] M. G. Rabbat and R. D. Nowak, “Decentralized source localization and tracking wireless sensor networks,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Quebec, Canada, May 2004, vol. 3, pp. 921–924.
- [125] W. Kowalczyk and N. Vlassis, “Newscast EM,” in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, Canada, Dec. 2005, pp. 713–720, MIT Press.
- [126] M. Jelasity, W. Kowalczyk, and M. Van Steen, “Newscast computing,” Tech. Rep., Dept. of Computer Science, Vrije Universiteit, Amsterdam, The Netherlands, 2003.
- [127] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Randomized gossip algorithms,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [128] P. A. Forero, A. Cano, and G. B. Giannakis, “Consensus-based distributed expectation-maximization algorithm for density estimation and classification using wireless sensor networks,” in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Las Vegas, Nevada, USA, 2008, pp. 1989–1992.
- [129] S. S. Pereira, S. Barbarossa, and A. Pages-Zamora, “Consensus for distributed EM-based clustering in WSNs,” in *Proc. IEEE Workshop on Sensor Array and Multichannel Signal Processing (SAM)*, Jerusalem, Israel, 2010, pp. 45–48.
- [130] Y. Weng, L.-H. Xie, and W.-D. Xiao, “Diffusion scheme of distributed EM algorithm for Gaussian mixtures over random networks,” in *Proc. IEEE Int. Conf. Control and Automation (ICCA)*, Christchurch, New Zealand, 2009, pp. 1529–1534.
- [131] Z. J. Towfic, J.-S. Chen, and A. H. Sayed, “Collaborative learning of mixture models using diffusion adaptation,” in *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing (MLSP)*, Beijing, China, 2011, pp. 1–6.
- [132] G. J. Pottie and W. J. Kaiser, “Wireless integrated network sensors,” *ACM Commun.*, vol. 43, no. 5, pp. 51–58, May 2000.
- [133] M. G. Rabbat and R. D. Nowak, “Distributed optimization in sensor networks,” in *Proc. Int. Symp. on Information Processing in Sensor Networks (IPSN)*, Berkeley, California, USA, 2004, pp. 20–27.
- [134] Z. I. Botev, J. F. Grotowski, and D. P. Kroese, “Kernel density estimation via diffusion,” *Annals of Statistics*, vol. 38, no. 5, pp. 2916–2957, 2010.
- [135] H. Wymeersch, S. Marano, W. M. Gifford, and M. Z. Win, “A machine learning approach to ranging error mitigation for UWB localization,” *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1719–1728, 2012.
- [136] E. Ozkan, V. Smidl, S. Saha, C. Lundquist, and F. Gustafsson, “Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters,” *Automatica*, vol. 49, no. 6, pp. 1566–1575, 2013.

Publications

Internationally Refereed Journal Articles

- F. Yin, C. Fritsche, D. Jin, F. Gustafsson, and A. M. Zoubir, “Cooperative Localization in WSNs Using Gaussian Mixture Modeling: Distributed ECM Algorithms”, submitted to *IEEE Trans. Signal Process.*, Apr. 2014.
- F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, “EM- and JMAP-ML Based Joint Estimation Algorithms for Robust Geolocation in mixed LOS/NLOS Environment”, *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 168–182, Jan. 2014.
- F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, “TOA Based Robust Wireless Geolocation and Cramér-Rao Lower Bound Analysis in Harsh LOS/NLOS Environments”, *IEEE Trans. Signal Process.*, vol. 61, no. 9, pp. 2243–2255, May 2013.
- F. Yin, C. Debes, and A. M. Zoubir, “Parametric Waveform Design using Discrete Prolate Spheroidal Sequences for Enhanced Detection of Extended Targets”, *IEEE Trans. Signal Process.*, vol. 60, no. 9, pp. 4525–4536, Sep. 2012.

Internationally Refereed Conference Papers

- F. Yin, A. Li, C. Fritsche, F. Gustafsson, and A. M. Zoubir, “RSS-Based Sensor Network Localization in Contaminated Gaussian Measurement Noise”, in *Proc. IEEE Int. Workshop Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Saint Martin, France, pp. 121–124, Dec. 2013. [Nominated for Best Paper Award]
- F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, “Robust cooperative sensor network localization via the EM criterion in LOS/NLOS environments”, in *Proc. IEEE Int. Workshop Signal Processing Advances in Wireless Communications (SPAWC)*, Darmstadt, Germany, pp. 500-504. June 2013.
- F. Yin, C. Fritsche, F. Gustafsson, and A. M. Zoubir, “Received Signal Strength-Based Joint Parameter Estimation Algorithm for Robust Geolocation in LOS/NLOS Environments”, in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, Canada, pp. 6471–6475, May 2013.
- F. Yin and A. M. Zoubir, “Robust Positioning in NLOS Environments using Nonparametric Adaptive Kernel Density Estimation”, in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, Kyoto, Japan, pp. 3517–3520, Mar. 2012.
- F. Yin, C. Debes, and A. M. Zoubir, Parametric waveform design for improved target detection, in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Barcelona, Spain, pp. 2074–2078, Aug. 2011.

Curriculum vitae

Name: Feng YIN
Date of birth: 11. 12. 1984
Place of birth: Liaoning (P. R. China)
Family status: married

Education

10/2008 - 03/2011 Technische Universität Darmstadt, Germany
ETIT M.Sc. with grade *sehr gut* (1.4)
Master Thesis: “Optimal Waveform Design for Enhanced Target Detection”

09/2004 - 07/2008 Shanghai Jiao Tong University, Shanghai, P. R. China
Electrical Engineering (B.Sc.)

09/2001 - 07/2004 North-East Yucai Mid. School, Shenyang, P.R. China
High school degree (Abitur)

Work experience

04/2011 - 09/2014 Research associate at Signal Processing Group, Technische Universität Darmstadt, Germany

04/2013 - 05/2013 Visiting student at Temasek Lab, Nanyang Technological University (NTU), Singapore

03/2009 - 09/2010 Student research assistant at Fraunhofer Institute SIT, Darmstadt, Germany

Erklärung laut §9 der Promotionsordnung

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe. Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 11. 06. 2014

