# Some Signal Processing Techniques for Wireless Cooperative Localization and Tracking

Hadi Noureddine Al Moussawi

**N° d'ordre** : 2012telb0237

**Sous le sceau de l'Université européenne de Bretagne**

# Télécom Bretagne

**En habilitation conjointe avec l'Université de Bretagne Occidentale**

Ecole Doctorale – **sicma**

---

# Some Signal Processing Techniques for Wireless Cooperative Localization and Tracking

---

## Thèse de Doctorat

Mention : **STIC Sciences et Technologies Information Communication**

Présentée par : **Hadi NOUREDDINE AL MOUSSAWI**

Préparée dans les laboratoires de Mitsubishi Electric R&D Centre Europe
En collaboration avec le département Signal et Communications de Telecom Bretagne
Laboratoire : **LabSTICC (UMR 6285)**

Directeur de thèse : **Ramesh PYNDIAH**

Soutenue le 16 novembre 2012

**Jury :**

M. Emanuel Radoï, Professeur, Université de Bretagne Occidentale / président
M. Bernard Henry Fleury, Professeur, Université d'Aalborg / rapporteur
M. Bernard Uguen, Professeur, Université de Rennes 1 / rapporteur
M. Benoît Denis, Ingénieur de recherche, CEA-Leti Grenoble / examinateur
M. Ramesh Pyndiah, Professeur, Telecom Bretagne / directeur de thèse
M. Damien Castelain, Expert technique, Mitsubishi Electric R&D Centre Europe / encadrant
M. Nicolas Gresset, Ingénieur de recherche, Mitsubishi Electric R&D Centre Europe / invitée

# Abstract

Advancements in information technology and communication systems have enabled the development of a wide variety of location based applications such as vehicle navigation and tracking, fleet management, sensor networks applications, home automation, telematics, security and location based services. Furthermore, the location information has improved communication systems performance. The global navigation satellite systems (GNSS) (e.g., the global positioning system (GPS)) are among the fundamental localization solutions. In harsh environments (e.g., urban canyons and indoor areas), the satellites signals are attenuated or completely obstructed, and these solutions do not provide a good accuracy or even become unavailable. In order to offer accurate and ubiquitous localization solutions, wireless communication systems have been considered, where several location dependent parameters of the transmitted signals can be measured and exploited (e.g., the time-of-arrival (ToA), the received signal strength (RSS) and the angle-of-arrival (AoA)).

In this thesis, we explore the topic of wireless localization from a statistical signal processing perspective, and we focus on two axes.

The first axis is cooperative localization applied to ad-hoc networks, where the nodes perform pair-wise ranging measurements (i.e., ToA or RSS) between each other in order to simultaneously estimate their positions. The limited number of measurements can result in ambiguities leading to high location estimation errors. Thus, it is important to know the conditions that guarantee an absence of ambiguity. For this purpose, we start by studying the unique solvability conditions based on the two approaches of graph rigidity and semi-definite programming, and we derive the identifiability conditions. Then, we consider the location estimation solutions, where we focus on probabilistic estimation and its application in Markov random fields using the nonparametric belief propagation (NBP) algorithm. This algorithm is based on message exchanges between the nodes and enables the computation of the marginal probability distribution function of the nodes positions in a distributed fashion. More specifically, we develop a new variant of the NBP that improves the positions estimation accuracy and that can be used for mitigating the ambiguities by exploiting the connectivity information.

The second axis is mobile terminals tracking based on RSS measurements. These measurements are affected by a shadowing phenomenon caused by obstacles in the propagation environment. We start by studying the improvement brought by the knowledge of the sha-

dowing maps to the position estimation accuracy. The classical solution for obtaining these maps is fingerprinting, which consists of making measurements at selected positions during an offline phase and saving them in a database. This operation can be costly in time and effort, and should be repeated periodically or whenever modifications occur in the deployment area. To overcome these difficulties, we develop the following two solutions : In the first solution, we jointly track the position and the shadowing, and we derive an auto-regressive model describing the shadowing stochastic process. We show that applying an auto-regressive order higher than one can be relevant in some applications. In the second solution, unlabeled traces, which are sequences of measurements made by moving terminals at unknown positions, are used to estimate the shadowing maps. In addition to the RSS, the unlabeled traces can include any other kind of positioning measurements. This solution allows a continuous refinement and update of the maps over time. Both developed solutions apply Bayesian filtering and estimation techniques, and are implemented by means of particle filters.

The proposed algorithms are investigated via Monte Carlo simulations in different deployment and application scenarios, where several advantageous points are demonstrated. In addition, several theoretical and practical results related to the topics of cooperative localization and tracking are derived.

**Keywords :** Cooperative localization, position tracking, graph rigidity, semidefinite programming, identifiability, Fisher information, Markov random field, nonparametric belief propagation, Monte Carlo methods, Bayesian filtering, particle filters, Rao-Blackwellization.

# Acknowledgements

The thesis work was an enjoyable and unforgettable experience of my life. It is a pleasure to thank many people for making this four years experience enriching and rewarding.

First of all, I would like to thank Prof. Emanuel Radoï for honoring me by serving as chairman of my oral committee. Furthermore, I am very grateful to my reviewers Prof. Bernard H. Fleury and Prof. Bernard Uguen for their careful review of my work, and to my examiner Dr. Benoît Denis for his insightful and encouraging comments.

I would like to thank my supervisors Eng. Damien Castelain and Prof. Ramesh Pyndiah for providing me with the opportunity to pursue this experience. I am very grateful to have had their trust in my skills, and for their support and advices.

I would like to express my deep gratitude to my colleague Dr. Nicolas Gresset, who has been actively interested in my work, for his guidance and his proposals. I am thankful for his motivation and his enthusiasm.

This thesis work was done in collaboration between the company Mitsubishi Electric R&D Centre Europe and the institute Telecom Bretagne. This work was generously financed by the French foundation ANRT and Mitsubishi Electric. I would like therefore to thank the administrative staff at Mitsubishi Electric and Telecom Bretagne for their help.

I would like also to thank Dr. David Mottier and Dr. Loïc Brunel for having welcomed me inside their Wireless Communication Systems team at Mitsubishi Electric and for their support.

Many thanks go also to my colleagues at Mitsubishi Electric, who have made my thesis years the most pleasant possible. Special thanks go to my colleague Dr. Jonathan Letessier for being always there to help me. I would like also to thank my colleague Dr. Julien Guillet for his enthusiasm and for many pleasant and inspiring discussions, and my colleague Dr. Mourad Khanfouci for his advices and scientific comments.

During this thesis, I have had the opportunity to participate to the European FP7 project WHERE2. This participation allowed me to collaborate with high-skilled researchers and has very much enriched my experience. I would like to express my warm thanks to the project coordinator Dr. Ronald Raulefs and to all the project partners.

I would like to express my sincere feelings of love and gratitude to my relatives and my friends for being my second family during my stay in France.

Finally, I dedicate my thesis to my family members. I am most grateful for their love, patience, encouragement and support throughout my life.

# Contents

# List of Figures

# List of Tables

# Acronyms

**AGPS** Assisted GPS

**AoA** Angle-of-Arrival

**AR** Auto-regressive

**BP** Belief propagation

**BS** Base station

**CR** Cognitive radio

**CRB** Cramér-Rao bound

**dwMDS** distibuted weighted MDS

**EKF** Extended Kalman filter

**ESA** European Space Agency

**EU** European Union

**FCC** (U.S.) Federal Communications Comission

**FIM** Fisher information matrix

**GBAS** Ground-based augmentation system

**GNSS** Global Navigation Satellite System

**GPS** Global Positioning System

**GRF** Gaussian random field

**ISDP** Iterative SDP

**INS** Inertial navigation sensor

**KF** Kalman filter

**KL** Kullback-Leibler

**LBS** Location-based service

**LoS** Line-of-sight

**MAP** Maximum a posteriori

**MDS** Multidimensional scaling

**ML** Maximum likelihood

**MMSE** Minimum mean square error

**MPR** Multipoint relay

**MRF** Markov random field

**MS** Mobile station

**MSE** Mean square error

**NBP** Nonparametric BP

**NLoS** Non-line-of-sight

**PSAP** Public safety answering point

**PU** Primary user

**RB** Rao-Blackwellization

**RFID** Radio frequency identification

**RMSE** Root MSE

**RSS** Received signal strength

**SAR** Search and Rescue

**SBAS** Satellite-based augmentation system

**SDP** Semidefinite programming

**SIS** Sampling importance sampling

**SIR** Sampling importance resampling

**SMACOF** Stress majorization

**SNR** Signal-to-noise ratio

**SPEB** square position error bound

**SSDP** Simple SDP

**TDoA** Time-difference-of-arrival

**ToA** Time-of-arrival

**TTFF** Time-to-first-fix

**UWB** Ultra-wide-band

**WHERE2** Wireless Hybrid Enhanced Mobile Radio Estimators - Phase 2

**WLAN** Wireless local area network

**WLS** Weighted least-squares

**WPAN** Wireless personal area network

**WSN** Wireless sensor network

# Notations

The following lists of notations are neither exhaustive nor exclusive, but may be helpful.

| Mathematical notations | |
|---|---|
| $\mathbb{R}$ | Set of real numbers |
| $\mathbb{R}^d$ | Set of real vectors of dimension $d$ |
| $\mathbb{R}^{n\times m}$ | Set of real matrices of dimension $n \times m$ |
| $\lvert S\rvert$ | Cardinality of set $S$ |
| $S\backslash i$ | Set $S$ without element $i$ |
| $\nabla_\theta$ | Gradient operator w.r.t. $\theta$ |
| max, min | Maximum and minimum |
| arg max | Argument that maximizes |
| arg min | Argument that minimizes |
| $p(.)$ | pdf of the random variable in parenthesis |
| $E\{.\}$ | Expected value of the random variable in brackets |
| $\mathcal{U}$ | Uniform distribution |
| $\mathcal{N}(\mathbf{m}, \mathbf{C})$ | Multivariate Gaussian distribution of mean vector $\mathbf{m}$ and covariance matrix $\mathbf{C}$ |
| $\mathcal{N}(.; \mathbf{m}, \mathbf{C})$ | pdf of $\mathcal{N}(\mathbf{m}, \mathbf{C})$ |
| $\sim$ | Distributed according to |
| $\propto$ | Proportional to |
| $\otimes$ | Kronecker product |
| $\delta$ | Dirac delta function |

| Vectors and matrices | |
|---|---|
| $\mathbf{A}^T$ | Transpose of matrix $\mathbf{A}$ |
| $\mathbf{A}^{-1}$ | Inverse of matrix $\mathbf{A}$ |
| trace$\mathbf{A}$ | Trace of matrix $\mathbf{A}$ |
| rank$\mathbf{A}$ | Rank of matrix $\mathbf{A}$ |
| $[\mathbf{A}]_{k:l}$ | Square submatrix of $\mathbf{A}$ with row and column entries going from $k$ to $l$ |
| $\mathbf{A} \succeq \mathbf{B}$ | Matrix $\mathbf{A} - \mathbf{B}$ is positive semidefinite |
| $\mathbf{I}_d$ | Identity matrix of rank $d$ |
| $(\mathbf{u}; \mathbf{v})$ | Column vector resulting from the concatenation of two column vectors $\mathbf{u}$ and $\mathbf{v}$ |
| $\|\mathbf{v}\|$ | Euclidean norm of vector $\mathbf{v}$ |
| $\langle \mathbf{u}, \mathbf{v} \rangle$ | Inner product of vectors $\mathbf{u}$ and $\mathbf{v}$ |
| $\mathbf{e}_i$ | Column vector with 1 at the $i^{th}$ entry and 0s elsewhere |
| $\mathbf{e}_{i,j}$ | Subtraction of vectors $\mathbf{e}_i$ and $\mathbf{e}_j$ : $\mathbf{e}_i - \mathbf{e}_j$ |

| Functions and sets | |
|---|---|
| $\mathcal{L}$ | Network of static nodes |
| $G = (V, E)$ | Undirected graph |
| $V$ | Vertex set |
| $E$ | Edge set |
| $(G, \mathbf{p})$ | Framework of graph $G$ and mapping $\mathbf{p}$ |
| $\mathbf{J}(.)$ | FIM of the parameter in parenthesis |
| $m_{i,j}^{(l)}(.)$ | Message function from node $i$ to node $j$ at iteration $l$ |
| $b_i^{(l)}(.)$ | Belief function of node $i$ at iteration $l$ |
| $\Psi_{i,j}(.,.)$ | Pair-wise potential function corresponding to nodes $i$ and $j$ |
| $\Phi_i(.)$ | Evidence on node $i$ position |
| $s(\theta)$ | Cost function of the WLS estimator |
| $\eta(i)$ | Set of direct neighbors of node $i$ |
| $\eta_k(i)$ | Set of $k$-step neighbors of node $i$ |
| $\hat{\theta}$ | Estimator of $\theta$ |
| $\rho(.)$ | Shadowing correlation function |
| $\{\psi_k\}$ | Set of basis functions defined on a space area |

| Parameters and random variables | |
|---|---|
| $N$ | Total number of nodes in network $\mathcal{L}$ |
| $n$ | Number of target nodes in network $\mathcal{L}$ |
| $m$ | Number of anchor nodes in network $\mathcal{L}$ |
| $d$ | Dimension of the network deployment space |
| $\mathbf{x}_i$ | Coordinates vector of node $i$ |
| $\mathbf{a}_k$ | Coordinates vector of anchor node $k$ |
| $(i, j)$ | Edge connecting nodes $i$ and $j$ |
| $d_{i,j}$ | True distance between nodes $i$ and $j$ |
| $\tilde{d}_{i,j}$ | Estimated distance between nodes $i$ and $j$ |
| $\theta$ | Vector of the target nodes coordinates |
| $y_{u,v}$ | Scalar ranging measurement between nodes $u$ and $v$ |
| $\mathbf{y}_{u,v}$ | Pair-wise vector measurement between nodes $u$ and $v$ |
| $\mathbf{y}$ | Vector of all pair-wise measurements |
| $P_0$ | Average received power at a distance of one meter |
| $n_p$ | Path loss exponent |
| $X_{i,j}$ | Shadowing component in the link $(i, j)$ |
| $\epsilon_i$ | Shadowing in RSS measurement with BS $i$ |
| $\sigma^2_{sh}$ | Shadowing variance |
| $e$ | Observation noise |
| $\sigma^2_e$ | Noise variance |
| $\hat{\epsilon}$ | Root mean square error |
| $M$ | Number of centroids in the NBP exchanged messages |
| $N_{BS}$ | Number of base stations in a deployment area |
| $\alpha_i$ | Vector of shadowing coefficients associated to base station $i$ |
| $L_{map}$ | Length of shadowing coefficients vector |
| $\Lambda$ | Shadowing atlas vector |
| $L_{atlas}$ | Length of shadowing atlas vector |
| $\beta$ | Shadowing cross-correlation value |
| $\mathbf{\Omega}$ | Shadowing vector at given position corresponding to all the base stations |
| $T$ | Time step |
| $\mathbf{s}_k$ | Hidden state vector at time $kT$ |
| $\mathbf{c}_k$ | Kinematic vector at time $kT$ |

CHAPTER 1

# Introduction

Advancements in information technology and communication systems have enabled the development of a wide variety of location based applications such as vehicle navigation and tracking, fleet management, sensor networks applications, home automation, telematics, security and location based services. Furthermore, the location information has improved communication systems performance.

The global navigation satellite systems (GNSSs) (e.g., the global positioning system (GPS)) are among the fundamental localization solutions. In harsh environments (e.g., urban canyons and indoor areas), these solutions may become unavailable or highly inaccurate due to signal blockage and multipath propagation.

In order to offer accurate and ubiquitous localization solutions, wireless communication systems have been considered, where several location dependent parameters of the transmitted signals can be exploited. These parameters are obtained from 'pair-wise' measurements (e.g., the time-of-arrival (ToA), the received signal strength (RSS) and the angle-of-arrival (AoA)). A measurement is called 'pair-wise' since it is made between two wireless communication devices and contains information about their relative positions or their relative displacement. The measurements depend on the propagation channel which introduces random fluctuations. Additionally, the unknown locations can be treated either as deterministic parameters or as realizations of random variables. For these reasons, wireless localization can be considered as a 'statistical signal processing' or 'estimation theory' problem.

Several challenges have emerged in the different phases of a wireless localization solution, such as the process of obtaining the measurements, the modeling of the noise and the dependence of the measurements on locations, the selection of the most relevant measurements when many of them are available, and the data processing algorithm or the estimator that outputs the estimated locations. The above mentioned issues impact the accuracy, in addition to the kind, the number and the quality of the measurements.

In this thesis, we address the wireless localization estimators from a statistical signal processing perspective, and we focus on two axes. The first axis is cooperative static localization applied to ad-hoc networks, where static (i.e., non-moving) nodes perform pair-wise ranging measurements (i.e., ToA or RSS) between each other in order to simultaneously estimate their positions. The second axis is non-cooperative dynamic localization based on RSS measurements, which consists in tracking over time the position of a single moving terminal.

Probabilistic models of the measurements are assumed to be available, but the process of obtaining the measurements is not fully investigated.

## 1.1 Outline of the thesis

**N.B. : The titles of the sections and subsections including the main contributions of the thesis are underlined.**

The outline of the thesis is presented in the following. The main contributions are listed, and any publications reporting them are also provided.

Chapter 2 is a general overview of wireless localization solutions and applications. It presents the evolution of the wireless localization systems, and briefly describes several interesting location based applications and services. The fundamental localization techniques and the ranging measurements are also described. At the end of this chapter, we define a classification of the wireless localization algorithms according to two criteria : Whether they are cooperative (i.e., exploit the pair-wise measurements between nodes of unknown positions), and whether they perform a tracking of mobile nodes positions (i.e., based on a displacement model or accounts for the temporal correlation of the observations).

Cooperative static localization is treated in Chapters 3 and 4, and non-cooperative dynamic localization based on RSS measurements is treated in Chapter 5.

Chapter 3 addresses the unique solvability and identifiability conditions. They allow the detection of ambiguities that are due to the lack of measurements. It is important to detect these ambiguities since they can result in high location estimation errors. In studying the unique solvability, the ranging measurements are assumed to be noiseless, and two approaches are used : Graph rigidity theory and semidefinite programming. We provide a survey of the graph rigidity theory results in the context of cooperative localization. For the semidefinite programming, we develop an algorithm that improves the detection of the uniquely solvable nodes, compared with the state of the art algorithm. The identifiability theory concerns the possibility of drawing inference about unknown parameters from probability distribution functions. It can be seen as the counterpart of the unique solvability when the measurements are noisy. In this chapter we derive correspondences between the rigidity and the identifiability. They enable the application of tools provided by the graph rigidity theory in checking identifiability properties. The main contributions of this chapter are the following :

- **Identifiability in static cooperative localization :** The identifiability conditions in static cooperative localization are based on correspondences established between the rigidity and the identifiability.
- **Rigidity and Fisher information matrix :** A relationship is derived between the network rigidity property and the FIM. It establishes the following result : For a category of networks called generic networks, the non-singularity of the FIM is a graph property that can be checked from the network connectivity.
- **Detection of uniquely solvable nodes using semidefinite programming :** Development of an algorithm for improving the detection of the nodes that have a unique

solution verifying a set of distance constraints ([2]).

Chapter 4 considers the cooperative localization algorithms with a main focus on probabilistic estimators and their application in graphical models. It starts by reviewing state of the art algorithms and classifying them according to several criteria, such as the kind of information used, the way the computation is processed and the need for anchor nodes. Then the weighted least-squares (WLS) estimator is described. The deterministic stability conditions of this estimator are derived, where the deterministic stability is related to the uniqueness of the global optimum. The derivation of these conditions is based on the unique solvability and identifiability conditions derived in Chapter 3. The main algorithm treated in this chapter is the nonparametric belief propagation (NBP). This algorithm is based on exchanging messages between the nodes, in a distributed manner, allowing them to compute probability distribution functions, called beliefs, of their locations. We develop a new variant of this algorithm which improves the accuracy and reduces the amount of exchanged data when the connectivity information is used to mitigate the ambiguities. The main contributions of this chapter are the following :

- **Deterministic stability :** Derivation of the deterministic stability conditions of the WLS estimator.
- **Cooperative localization algorithm based on NBP :** Development of a distributed cooperative localization algorithm based on NBP. This algorithm consists of two phases : A first phase that computes the beliefs, and a second phase that reduces the errors due to beliefs approximations and eliminates some ambiguities by using the connectivity information ([3]).

In Chapter 5, non-cooperative Bayesian tracking based on RSS measurements is studied. The RSS observations are random due to the presence of the shadowing. We show that the knowledge of the shadowing maps greatly improves the tracking accuracy. The classical solution for obtaining these maps is the fingerprinting which requires huge calibration efforts. To overcome these efforts, two solutions are developed : The first one jointly tracks the position and the shadowing, and the second one tracks the position and estimates the shadowing maps, jointly. The main contributions of this chapter are the following :

- **Joint shadowing and position tracking :** This solution takes the shadowing as a part of the hidden state vector to be estimated. As the mobile terminal moves, the spatial correlation of the shadowing is transformed into a temporal correlation. An auto-regressive model is developed for representing the temporal shadowing process evolution. The tracking is performed using a Rao-Blackwellized particle filter which reduces the required number of particles ([4]).
- **Joint position tracking and shadowing maps estimation :** This solution performs a Bayesian tracking of the position and shadowing maps estimation. Unlabeled traces and positioning measurements including the RSS are used in this solution ([5]).

Finally, in Chapter 6, the contributions of the thesis are summarized with concluding remarks, and future research directions are discussed.

CHAPTER 2

# Wireless Localization Systems and Applications

## 2.1 Introduction

Location or position finding has always been of major importance in many fields of human activities such as navigation, transportation and security. A breakthrough occurred with the introduction of the Global Positioning System (GPS) which became available to public users in the last decade of the $20^{th}$ century. The GPS provides an accuracy of the order of several meters when the GPS receiver is in the line-of-sight (LoS) of a sufficient number of GPS satellites (i.e., four satellites). In dense urban areas, the accuracy can be highly degraded, and this service becomes completely unavailable in most indoor environments.

The progress in wireless communication systems and their widespread availability allowed the development of several wireless localization solutions that can complement or replace the GPS in situations where the latter is not reliable or not operational, such as indoor and other harsh environments, and paved the way for the emergence of a variety of location-aware applications and services.

The objective of this chapter is to provide an overview of the basic wireless localization solutions and services.

We start by presenting a short history of the evolution of the wireless localization systems in Section 2.2. Then, in Section 2.3, we discuss the needs for localization and the basic location-based applications and services. In Section 2.4, we present the fundamental geometrical localization techniques which are performed in two phases : In the first phase location-dependent measurements are made where each measurement defines a geometrical locus or set ; in the second phase the location is computed as the intersection of the different sets. The most attractive measurement methods are the ranging measurements which allow estimating the distance between two devices. These measurements include timing and power measurements, and are discussed in Section 2.5 with the sources of noise and statistical modelings. A localization technique that has gained much attraction due to its accuracy is the fingerprinting technique in which measurements are compared to entries of a database where

an entry is a fingerprint of measurements at a known selected location. The database entries are obtained either manually or by using calibration tools. This technique is described in Section 2.6. A classification of the localization algorithms based on the measurement methods or information used is provided in Section 2.7. Finally, in Section 2.8, concluding remarks are provided.

## 2.2 Evolution of wireless localization systems

The need for location information goes back to a long time ago. At first, when human beings started to explore new territories, the issue was to locate either themselves or their destination, and be able to come back home. These tasks were solved using terrestrial marks. Then, with the development of maritime transportation, needs for other solutions arose because of the absence of marks at sea. The first sea navigations followed the shore where terrestrial marks are available. Then, with the development of astronomical tools, navigators were able to compute their latitude at night by observing the locations of navigational stars. But unfortunately, the astronomical observation was not able to solve the longitude. The compass discovery allowed the development of dead reckoning technique, which is the process of estimating the present location from the past location, speed and direction of displacement, but this method lacks precision due to errors accumulation and was far from the final answer for sea navigation. The longitude problem was solved with the invention of the marine chronometer in the late eighteenth century, where the longitude was computed by relating the locations of the stars to the time.

### 2.2.1 Era of wireless communications

The development of radio transmission systems at the beginning of the $20^{th}$ century paved the way for a new era regarding location systems. With the improvement of the accuracy of local time generators (oscillators and atomic clocks), it becomes possible to use the radio signals as new localization marks. Since the middle of the $20^{th}$ century several terrestrial location systems have been developed and commercially deployed, including Decca, Loran and Omega systems [6]. These systems rely on an infrastructure of synchronized stations, and the localization method is based on measuring the time difference between pairs of signals from several stations. A given constant time difference can be represented by a hyperbolic line of positions, and the intersection of two lines is the location of the receiver. The hyperbolic method is described in Section 2.4.

The Decca system uses low-frequencies (30-300kHz) and is able to deliver accuracies within 50 meters ($m$) with a range of around $400km$. The Loran system also uses low frequencies (90-110kHz) with a range of up to $1900km$ and absolute error between 185 and $463m$. In fact, there is a trade off between range and accuracy. Microwave systems can offer a higher accuracy (in the order of few meters) but with a very limited range, while low frequency systems offer a much reduced accuracy ($50m$) but with a higher range. The Omega system offers a worldwide coverage and operates at very-low-frequency (10.2-13.6kHz) but the accuracy is of the order

1-3$km$. This system was designed principally for maritime and aeronautical users.

### 2.2.2 Global navigation satellite systems

In a global navigation satellite system (GNSS), signals transmitted by satellites of known locations are used as localization marks [7, 6]. The use of satellites for positioning resulted in real improvements in terms of availability, coverage and accuracy.

The first satellite navigation system was launched in 1958 under the name of TRANSIT project. This system became operational for the U.S. Navy in 1964. The obtained accuracy was in the range of 200-500$m$. Then in 1973, the NAVSTAR-GPS project was launched by the U.S. Direction of Defense (DoD) to overcome the limitations of the previous system. The first four GPS satellites were launched in 1978 and the $24^{th}$ was launched in 1994. Initially, the highest quality of service was reserved for military use, and the signal available for civilian use was intentionally degraded (selective availability). The selective availability was turned off in year 2000 improving the precision of civilian GPS from 100$m$ to 20$m$.

The GPS system consists of three segments :

- **Space segment** consisting of a constellation of orbiting satellites at an altitude of approximately 20 000$km$.
- **Control segment** consisting of several ground stations and antennas for monitoring the system and keeping it operational, synchronizing the atomic clocks on board satellites and controlling the orbital configuration.
- **User segment** consisting of the user receiver equipment capable of receiving and processing the GPS signals. The receiver determines the transit time of messages sent by synchronized satellites and computes the distance to each satellite. The distances along with the satellites locations are used to compute the receiver location. A possible computation method is trilateration which is explained in Section 2.4. The receiver is not synchronized with the satellites, and thus, it needs at least four satellites for solving both location and time.

Over the last decades, several improvements to the GPS have been implemented, and the different segments of the system are continuously improved to increase its performance and its accuracy. Several satellites were launched to update the constellation and replace out of order satellites, and more launches are scheduled in the upcoming years. The evolution in the field of integrated circuits and electronics allowed the construction of GPS receiver chips of small size and low power consumption, and which are nowadays implemented in different kinds of digital equipments such as mobile phones, digital cameras and computers.

Other GNSSs have been equally developed or are under development :

- The GLONASS system was launched by the Soviet Union in 1978 and became fully operational in 1995 with a 24 satellites constellation. Because of the short life time of the satellites and the limited financial budgets, the constellation declined, and only seven satellites were operational in 2002. Then a renewal program was launched and in 2011 the constellation was fully restored.
- The Compass system is a Chinese GNSS which became operational in China in 2011

and the global system should be finished by 2020.

– The Galileo system is currently built by the European Union (EU) and European Space Agency (ESA) and aims at providing a high precision positioning system upon which European nations can rely. The initial service of this system is expected around 2014 and completion by 2019. The new major upgrade service of Galileo compared to GPS and GLONASS is the Search and Rescue (SAR) function [8]. To implement this function, the satellites should be equipped with transponders for transferring the distress signals from the user's transmitter to the Rescue Coordination Center, which will then initiate the rescue operation. At the same time, the system will provide a signal to the users, informing them that their situation has been detected and that help is under way.

### 2.2.3 GNSS augmentation

This technique aims at improving the accuracy, integrity and availability of GNSS services. It is called augmentation since it integrates external information in the positioning process. There are three kinds of external information :

– Information for compensating the errors that are due to clock drift, ephemeris or ionospheric delay.
– Information consisting of additional ranging measurements for improving the coverage in the case of reduced visibility of the GNSS satellites.
– Information for improving the startup performance and reducing the time-to-first-fix (TTFF). The TTFF is the time required for a GPS receiver to acquire satellite signals and navigation data.

External information can be delivered either by a satellite-based augmentation system (SBAS) in which messages are broadcast by additional satellites [9], or by a ground-based augmentation system (GBAS) in which messages are broadcast by terrestrial wireless networks [10].

An example of a GBAS is the assisted GPS (AGPS). This system uses data acquired by a wireless network such as the cellular network for reducing the TTFF and improving the startup. The AGPS is currently used in many GPS-capable cellular phones or mobile stations (MS) (see Figure 2.1) . There are two categories of AGPS [11] :

– MS-assisted in which unprocessed GPS data is sent to the network server.
– MS-based in which the almanac and GPS ephemeris are sent to the handset.

The GNSS localization service may become unavailable or highly inaccurate in harsh environments where there are signal blockage or multipath propagation, such as urban canyons and indoor environments.

### 2.2.4 Localization in terrestrial broadcast networks

This technique allows one to take benefit from the already existing infrastructures of FM and TV broadcast systems.

The team at Rosum Corporation developed a series of techniques to take advantage of digital broadcast television signals to localize mobile users and devices in urban and indoor

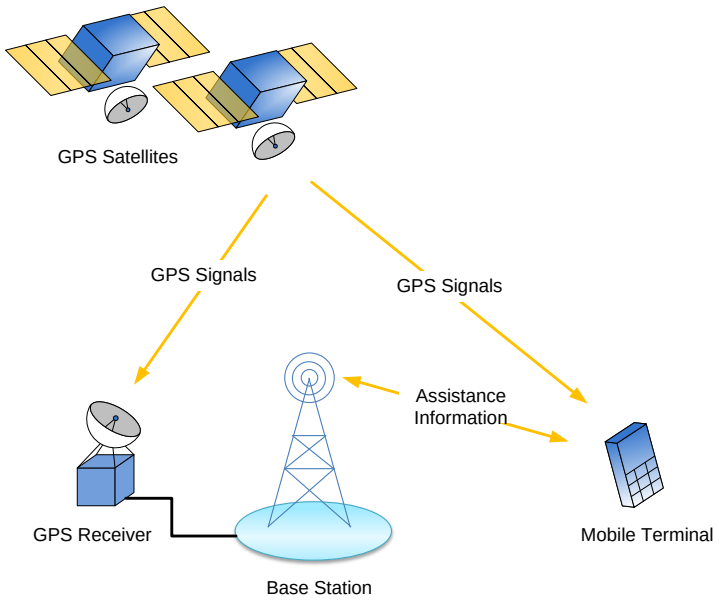



***Figure 2.1*** — Assisted GPS.

areas [12]. This solution has several advantages over the GPS :

- The TV signals have higher power levels than satellite signals (about 40 decibels ($dB$) indoor power advantage).
- The TV signals are spread over a wide range of frequency bands allowing higher temporal precision and additional strength against multipath.
- Moreover, the locations of TV towers are well known, and do not cause any Doppler frequency shift as in the case of moving GPS satellites.

The main disadvantages of this solution are the lack of precision of the implemented clocks which are not intended to be used for localization and the insufficiency in the number of detected broadcast towers especially in rural areas. But since GPS provides a good accuracy in rural areas where GNSS satellites are in LoS, this solution can be used as a complementary to GPS. The system proposed by Rosum is based on the use of synchronization codes included in the broadcast signals for estimating the time-of-arrival which are converted into pseudo-ranges [12]. Rosum is manufacturing TV+GPS hybrid positioning modules for enabling a reliable localization [13]. An indoor accuracy of 30-50$m$ is claimed.

Location solutions using the FM signals have also been reported in [14].

### 2.2.5 Localization in cellular networks

Location finding in cellular networks has been an active research area in response to an order issued by the American Federal Communications Commission (FCC) in 1996. The FCC order requires all cellular carriers to deliver accurate location information of emergency 911 callers to public safety answering points (PSAP). Some additional details concerning the FCC requirements are discussed in Section 2.3. This order has been issued to face the fact that most 911 calls originate from mobile phones, and the wireless callers do not receive the same

quality of assistance compared with fixed-network callers. This is due to the unknown location of the wireless callers. One solution to localize the mobile stations is the implementation of a specialized equipment such as a GPS receiver. But this solution requires a replacement of all the MSs in the market, and additionally, at that time, the GPS receivers were costly in price, size and power consumption, without forgetting the TTFF delay. The alternative solution was to use the cellular network, where the signals traveling between the MS and a set of fixed base stations (BSs) of known locations can be used. The required modifications are placed on the network side without affecting the MSs.

Several challenges are facing this solution, including low signal-to-noise ratio (SNR), interference, non-line-of-sight (NLoS) and multipath propagations [15].

Several methods have been developed for exploiting the location-dependent parameters measured from the transmitted signals [16, 17]. Examples of these methods are the Cell-ID method which assigns the MS to the serving BS and the time-difference-of-arrival (TDoA) based methods. The Cell-ID method is illustrated by Figure 2.2 and its precision depends on the cell size. The TDoA technique is discussed in Section 2.4.

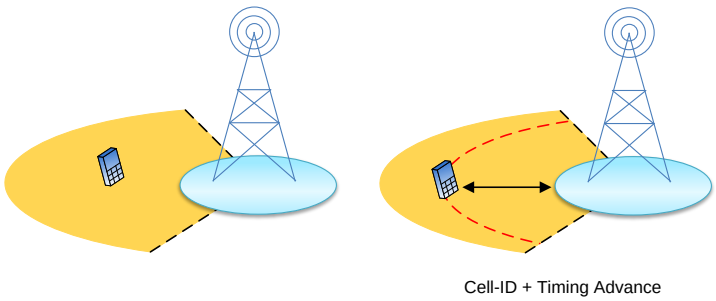



***Figure 2.2*** — Cell-ID localization with sectorized cells.

### 2.2.6 Localization in wireless local area networks

Wireless local area networks (WLANs) are generally used to provide a wireless internet connection to mobile terminals through access points. The Wi-Fi is the main standard for WLANs. The Wi-Fi access points are mainly deployed in indoor areas (e.g., homes, malls, stores, airports, stations) and some outdoor areas (e.g., parks). The connectivity range of an access point depends on several factors such as the transmit power and the obstacles in the surrounding area, and it typically varies from a few tens of meters in indoors to about $100m$ in outdoors.

The massive deployment of WLANs offers a realistic solution for indoor localization [18]. Most of the developed WLAN localization solutions rely on received signal strength (RSS) measurements, and perform either a trilateration or a fingerprinting in which observations from several access points are compared against constructed models in order to find the best matching. Fingerprinting is discussed in more details in Section 2.6.

Several solutions using Wi-Fi access points have been developed and are currently commercialized. Examples of these solutions are Skyhook [19] and Navizon [20] systems. These

two solutions are based on collecting information on Wi-Fi access points and cellular BSs locations all over the world and maintaining them in databases. A client location is computed by collecting a raw data and sending it to a location server which returns a location estimate. Google and Apple are both maintaining such databases for providing location-based services on their platforms.

### 2.2.7 Localization in wireless personal area networks

Wireless personal area networks (WPANs) are networks of wireless devices communicating over short ranges. The devices are characterized by their low complexity and low power consumption. WPANs are based on the working group IEEE 802.15 which consists of several task groups such as WPAN/Bluetooth, High Rate WPAN and Low Rate WPAN.

Low Rate WPANs are based on the standard 802.15.4 which specifies the physical and data link layers for low power consumption and low complexity devices. The amendment 802.15.4a specifies additional physical layers aiming at making the data rate scalable, increasing the connectivity range and improving the ranging and localization accuracy while at the same time reducing the power consumption and other costs. Among the technologies selected is the ultra-wide-band (UWB) which allows to estimate the time for signal propagation with a high precision thanks to its high time resolution.

The localization solutions in WPANs rely on received signal strength, connectivity and proximity information, or time-of-arrival measured using UWB signals. The order of accuracy of UWB localization in indoor can be less than one meter.

The standard 802.15.4 is attractive for wireless sensor networks since these networks are generally deployed with a large number of nodes (hundreds or even thousands) and it is interesting to have low cost nodes with long battery lifetime. In sensor networks localization, a subset of the nodes are reference or anchor nodes and obtain their locations either manually or by other means (e.g., GPS), and the remaining nodes are localized using inter-node measurements. If a node is capable of making measurements with a sufficient number of anchor nodes, then it can be localized solely from these measurements. But due to energy conserving constraints, the nodes can lack the energy necessary for long-range communication, and in this case, multi-hop or cooperative localization techniques are used where all the locations are computed simultaneously and measurements between pairs of nodes of unknown locations are used [21].

### 2.2.8 Localization of RFID tags

RFID is an identification technology enabling objects and people tracking. An RFID system consists of readers, tags and a middleware. Tags are small and cheap devices that can be carried by persons or fixed on objects. Readers are able to read the information stored in the tags from a distance varying from few centimeters to several meters, depending on the kind of tags (passive or active) and the used frequency. RFID technology has been used in localization where the delivered accuracy depends on the communication range and density

of readers [22].

### 2.2.9 Pervasive localization systems

Wireless localization is currently an active research field. Many solutions have been developed for the different kinds of environments and with different levels of precision and complexity, and new solutions are still emerging [23].

With the proliferation of location-aware applications and services, ubiquitous and accurate localization is becoming a key enabling technology. A high quality of localization accuracy and availability can be achieved either by using hybrid localization solutions which combine different kinds of measurements obtained using one or several radio access technologies [24], or by switching the mobile devices seamlessly from one technology to another.

## 2.3 Applications

Advancements in information technology and communication systems enabled the development of a wide variety of applications and services that are either based on the availability of location information or take benefit of it. These applications cover several fields of human activities such as personal (e.g., navigation, finding places of interest, children location tracking), professional (e.g., fleet management, tracking components in manufacturing places) and security and safety (e.g., tracking workers in dangerous areas and firefighters, emergency calls, driving assistance and active safety). Additionally, location information has been shown to be beneficial for wireless communication systems performance optimization. Several fundamental applications are presented below.

### 2.3.1 Navigation

Navigation is the original localization application. It is the process of ascertaining the position and planning a route from one place to another. The GNSS is currently the main navigation solution for pedestrians and vehicles in outdoor environment.

### 2.3.2 Tracking

Tracking applications are based on following the positions of moving bodies or vehicles over time. Examples of these applications are fleet management such as emergency vehicles and taxi companies, real-time traffic information by tracking the positions of mobile phones inside vehicles, cargo tracking, and asset tracking at manufacturing sites and hospitals.

### 2.3.3 Emergency calls

In recent years, the number of 911 calls using mobile phones has significantly increased, and it is estimated to constitute about 70% of the total number of 911 calls [25]. In order

to allow a rapid response to the calls and improve the relibaility of wireless 911 services, the U.S. FCC has adopted rules requiring wireless service providers to provide the PSAP with the location of the caller. Phase I rules required to provide the location of the cell site or base station transmitting the call. Phase II required to deliver the location accurate to within 50 to $300m$.

For the 112 emergency calls in Europe, a recommendation of the European Commission requires the operators to provide the best information available as to the location of the caller, to the extent technically feasible.

### 2.3.4 Location-based services

Location-based services (LBSs) are applications that exploit the location of a mobile device by adding value to this information [26]. Most of the LBSs are developed by third parties and require the availability of data services. The data services can be maintained by third parties and delivered using the communication networks. Thus, LBSs are a source of revenue for application developers, data providers and operators.

Some examples of LBSs are :
- Social networking : friend-finder, social events in a city, etc.
- Yellow pages : requesting the nearest service such as ATM or restaurant.
- Location-based mobile advertising and location-aware information.
- Game : where the location is a part of the game play.
- Location sensitive billing.

Privacy is an important issue in LBSs [27]. A user may not accept that service providers have an unauthorized access to personal location information. Thus, preventing any unauthorized access to personal location records and forcing saving or forwarding of the location information to be anonymous are prerequisites to protect privacy.

### 2.3.5 Wireless sensor networks

A wireless sensor network (WSN) is a set of sensor nodes which are spatially distributed to measure local quantities and communicate with each other to relay the gathered data to a sink node. Based on this data, an inference or a decision can be made. In many applications, sensors' locations must be known for their data to be meaningful. Among the wide variety of WSN applications [21], we have :
- Environmental monitoring : The sensors collect local environmental information such as temperature, humidity, pressure, light or radiations, soil properties, etc. Having only this raw data does not yield much information about the variations of these factors over the deployment area. In contrast, the knowledge of the locations allows to construct a map of the distribution of these factors and enables a deeper analysis.
- Civil structures monitoring : Monitoring deformations and cracks in structures such as buildings and bridges.
- Agriculture : Soil monitoring and environmental conditions for irrigation and fertilizing

control.
- Animal tracking : Studying behaviors and interactions between species.
- Logistics : Monitoring and controlling goods or components within manufacturing floors and warehouses.

### 2.3.6 Communications enhancement

The location information can be used for enhancing communication systems at the different layers. Some examples are :

- Channel estimation : If the channel parameters maps are known and saved in a database, then the knowledge of the mobile device position allows reading the corresponding values directly in the database and improving the channel estimation process. Examples of maps are shadowing, delay spread and power delay profile maps. There are several methods for constructing the maps such as manual calibration and ray tracing tools.
- Interference coordination : In [28], the known power maps and the location information are used to determine whether the user equipment is in a cell-edge and suffers from a strong inter-cell interference, and to apply a resource scheduling accordingly.
- Synchronization : The location information can be used to compute the distance between an MS and a BS. By measuring the time-of-arrival of the signal transmitted by the BS, a common reference time can be obtained. In the case where there are several synchronized (i.e., have a common reference time) BSs, the signals from these BSs are no more considered as interfering with each other and can be jointly used for improving the synchronization accuracy by employing a joint scheme [28].
- Cognitive radio : A main function of a cognitive radio (CR) is spectrum sensing, which is performed for detecting the unused spectrum and sharing it without harmful interference with the primary users (PUs) and other CRs. In [29], CRs cooperate with each other to construct their shadowing maps and to track the transmitted power of the PUs. In the proposed solution, the path loss is assumed to be known, and a CR estimates its shadowing map based on power measurements made by other CRs of known locations.
- Geographical routing : It is mainly proposed for wireless networks, and relies on the knowledge of the locations of the network nodes. Several geographical routing protocols have been developed and their performances depend on network density [30].
- Cooperative communication : Position information can be useful in relay selection techniques [31].
- Beam-forming : Adaptive or self-adjusted antenna array beam-formers have great potential for improving SINR and hence achieving higher cell capacities. The position information can be useful for reducing the convergence time of weighting vector determination [32].

## 2.4 Fundamental localization methods

In this section, we present some fundamental localization methods. By definition, a target node is a wireless communication device of unknown location and an anchor node is such a device of known location, or in other words, of known coordinates in a 2 or 3-dimensional Euclidean space. The localization methods considered here compute the location of a single target node in two phases :

- In the first phase, signals are exchanged between the target node and several anchor nodes. By measuring or estimating one or several signal metrics, information regarding the location of the target node relative to the anchor nodes can be extracted.
- In the second phase, the metrics are aggregated and processed by a localization or data fusion algorithm that outputs the location.

By assuming error free metrics, each metric defines a constraint on the coordinates of the target node, or in other words a feasibility region, and the location can be computed when there is a sufficient number of metrics or feasibility regions intersecting in one single point. But in real applications, the metrics are affected by several kinds of errors. Therefore, the estimated feasibility regions do not necessarily include the true location and the intersection of these regions can be the empty set. In this case, appropriate estimation techniques are needed for defining the localization algorithm. The errors affecting the metrics are realizations of random variables, and the knowledge of their statistical properties is very important for several reasons : First, if the error affecting a metric has a very high variance, then this metric will not provide a good improvement to the localization accuracy and discarding it allows reducing the computational complexity, and second, the development of efficient probabilistic estimators relies on this knowledge.

In the following, we present several localization methods using different kinds of metrics that we assume error free.

### 2.4.1 Angle-of-arrival based method

This method is based on finding the angle-of-arrival (AoA) of incident signals to the anchor nodes (i.e., uplink signals). In a 2-dimensional space, each AoA defines a straight line along which the target node is located, and a minimum number of two AoA observations is needed for computing the 2-dimensional location as shown in Figure 2.3. The AoA based method is also called triangulation method.

The advantages of this method are that it requires less observations than other methods and that it does not require a synchronization of the anchor nodes and target node clocks. The disadvantages are the hardware complexity since an antenna array is required for measuring the AoA and that small errors in the angle estimation result in a high localization error especially when the target node is far from the anchor nodes.

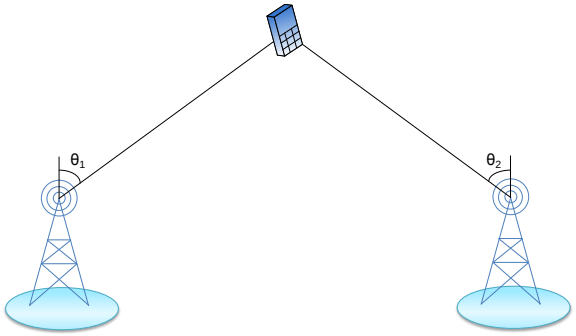



***Figure 2.3*** — Angle-of-arrival based localization in a 2-dimensional space.

### 2.4.2 Time-of-arrival based method

For signals propagating in free space at the speed of light ($c \approx 3 \times 10^8 m/s$), the distance between the target node and the anchor node $i$ is given by $d_i = c(t_i - t_0)$ where $t_0$ is the time instant at which the emitter node begins transmission and $t_i$ is the time-of-arrival (ToA) at the receiver node. If the emitter and receiver are synchronized and $t_0$ is known at the receiver (e.g., can be included in the transmitted data packet), then the distance can be obtained by measuring the ToA.

In a 2-dimensional localization, a ToA observation defines a circle feasibility region, and the intersection of at least three circles gives the target node location, as shown in Figure 2.4. The ToA based method is also called trilateration method.

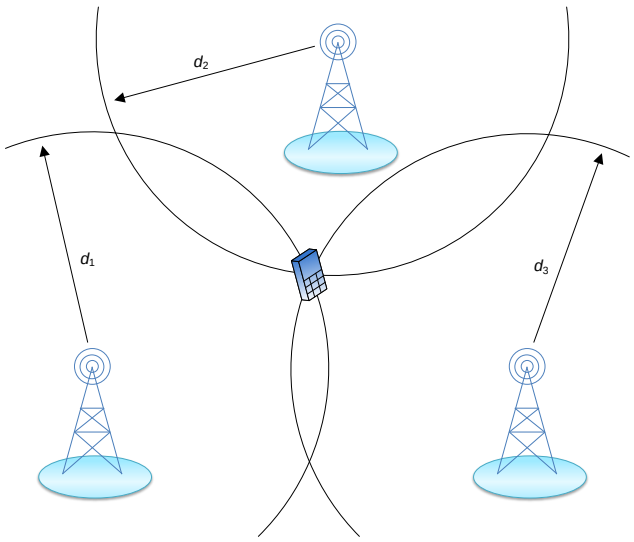



***Figure 2.4*** — Time-of-arrival based localization in a 2-dimensional space.

The need for time synchronization can be overtaken by measuring the round trip delay, which is the time elapsed between the transmission of a signal and the reception of an acknowledgment. This technique is described in Section 2.5.

### 2.4.3 Time-difference-of-arrival based method

The TDoA based method only requires the anchor nodes to be synchronized. In a 2-dimensional space, the pseudo-range measurement between anchor node $i$ and the target node is given by

$$p_i = d_i + b = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2} + b \tag{2.1}$$

where $d_i$ denotes the anchor-target distance, $b$ is an unknown clock bias common to all pseudo-ranges, $[x_t, y_t]^T$ and $[x_i, y_i]^T$ denote the location vectors of the target node and anchor node $i$, respectively, and $^T$ denotes the vector transpose operator.

One method to compute the location is to solve the system of pseudo-range equations. Another method consists in eliminating the bias by subtracting the pseudo-ranges, yielding the TDoA observation for anchor nodes 1 and $i$ :

$$D_{1,i} = p_i - p_1 = \sqrt{(x_t - x_i)^2 + (y_t - y_i)^2} - \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2} \tag{2.2}$$

The set of locations verifying (2.2) define a branch of a hyperbola. Two hyperbola branches can intersect in one or two points, and in the case of two intersection points a third hyperbola branch is needed to eliminate the ambiguity. Thus, three or four anchor nodes are needed. The TDoA based method is illustrated in Figure 2.5.

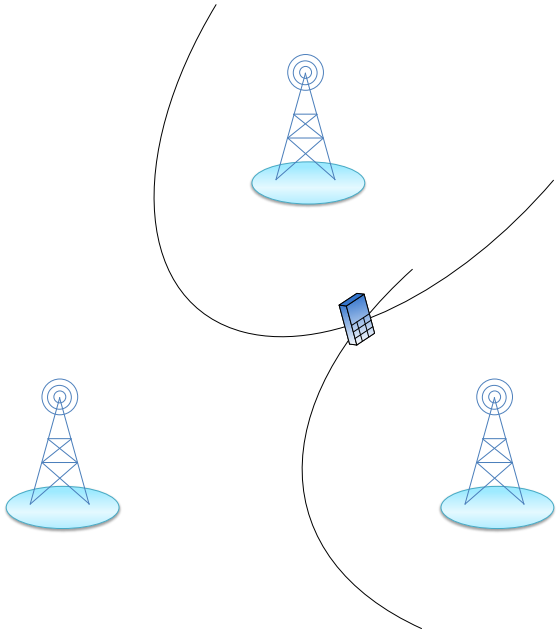


***Figure 2.5*** — Time-difference-of-arrival based localization in a 2-dimensional space.

The advantage of the TDoA based method over the ToA based method is that it only requires the anchor nodes to be synchronized with each other. But in the presence of noisy observation, the ToA based method is more accurate since it involves less unknown parameters (the bias is known), assuming a perfect synchronization.

### 2.4.4 Hybrid methods

In hybrid methods, several kinds of metrics are used, allowing to improve the accuracy and to eliminate the ambiguity when only a limited number metrics are available [24, 33].

For example, in cellular networks, when the MS is close to its serving BS, the SNR of the received signals at other BSs can be very low resulting in a degradation of the localization accuracy. If both the AoA and ToA are estimated, then only one BS is sufficient for computing the location. TDoA/AoA hybrid methods can be used when the target node communicates with two anchor nodes.

### 2.4.5 Limits on localization accuracy

The localization accuracy depends on several factors such as the kind, number and quality of measurements, and the network topology. In addition to these factors, the accuracy also depends on the estimator or data fusion algorithm.

The Cramér-Rao bound (CRB) gives a lower bound on the localization error variance that can be achieved by unbiased estimators [34]. This bound is a function of the above mentioned factors and is derived from the Fisher information matrix (FIM). The contribution of an observation to the FIM is a kind of measurement of the amount of information about the target node location contained in this observation. A solution for reducing the complexity of a location estimator is developed in [24] and performs by only considering the most relevant observations among the available ones, where the selection of the most relevant observations is based on the CRB.

## 2.5 Ranging measurements and sources of errors

A ranging measurement is a metric allowing the estimation of the distance between a transmitter and a receiver. The two main ranging measurements are ToA and received signals strength (RSS).

### 2.5.1 ToA ranging

The distance is computed by multiplying the propagation time by the propagation speed. There are two ToA ranging techniques :

- One-way ranging which is based on one transmission and requires the transmitter and the receiver to be synchronized.
- Two-way ranging which overcomes the need for synchronization by two transmissions.

These two techniques are illustrated in Figure 2.6.

In ToA ranging, a timing error of 1 microsecond corresponds to a ranging error of $300m$ and a timing error of 1 nanoseconds corresponds to a ranging error of 0.3m. Thus, for most indoor applications, the error must not exceed a few nanoseconds. The main sources of errors are :

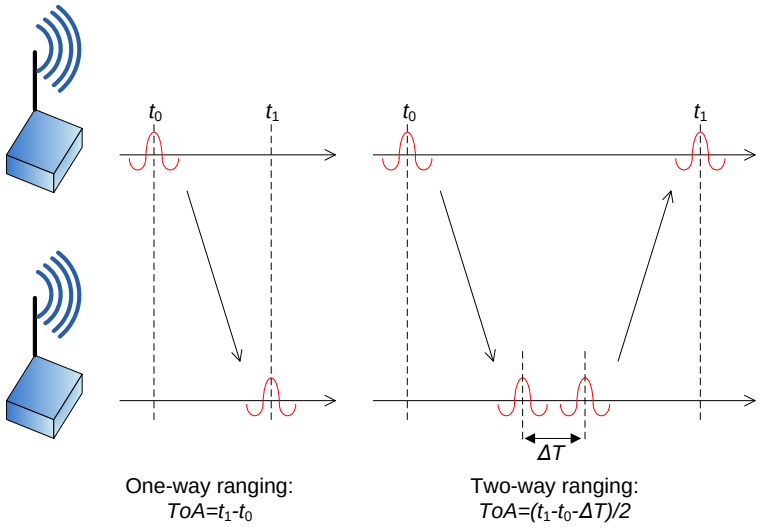



***Figure 2.6*** — One-way and two-way ranging techniques.

- Timing errors : synchronization errors, clock errors and drift, estimation of the reply delay in two-way ranging.
- Additive noise : In LoS and multipath free propagation, the additive noise limits the accuracy of ToA estimation. In [35], it is shown that the variance of unbiased ToA estimators is bounded below by

$$var(\text{ToA}) \geq \frac{1}{8\pi^2 BTF_c^2 \text{SNR}} \tag{2.3}$$

  where $B$ is the signal bandwidth, $T$ is the transmitted symbol duration, $F_c$ is the center frequency and SNR is the signal-to-noise ratio.
- Multipath : In multipath propagation, several versions of the same signal arrive at the receiver via different propagation paths. In the absence of multipaths, a simple receiver computes the ToA by finding the peak of the cross-correlation between the received signal and the known transmitted waveform. In multipath propagation, the LoS path might not be the strongest path and the receiver detects the first arriving path and not the one with the highest peak. A common detection technique consists in measuring the time when the cross-correlation first crosses a given threshold [36].
- Non-line-of-sight : When the direct path is blocked, the receiver can only observe NLoS components, and the estimated distance is positively biased. The mitigation of NLoS impact is still an active research field and several approaches have been developed. Some of these approaches perform by identifying and discarding the NLoS observations. Others perform by solving a weighted least squares problem where higher weights are assigned to the links with high LoS probability. And other methods perform by assuming statistical models of the NLoS bias and applying a probabilistic estimation technique. [37] provides a survey of the different existing approaches.

#### Ranging with UWB

UWB employs short time duration pulses of the order of nanoseconds that are widely spread in the frequency domain. Thus, according to (2.3), it enables an accurate ToA ranging in the presence of additive noise. The short time duration can also solve the multipath components. The theoretical limits and the techniques for estimating the ToA with UWB are discussed in [36].

#### Statistical error model

Short ranging measurements have shown that ranging errors of LoS propagation can be roughly modeled as Gaussian distributed [38]. In the cases of NLoS and obstructed LoS, other models have been proposed to account for errors that can take large values. Examples of these models are the mixture of Gaussian distributions [21] and the mixture of Gaussian and exponential distributions [38].

### 2.5.2 RSS ranging

RSS is a measure of the received power that can be easily obtained without additional hardware complexity. RSS ranging is based on the principle that the average received power is decreasing with the distance separating the transmitting node from the receiving node due to path loss. Thus, a path loss model is needed for estimating the distance from the RSS value. A widely accepted model for characterizing the RSS is given by [39]

$$P_r(d) = P_0 - 10n_p \log_{10} d + \epsilon \tag{2.4}$$

where $P_r(d)$ (in $dB$) is the RSS value at a distance $d$, $P_0$ is the average RSS value (in $dB$) at a distance of $1m$, $n_p$ is the path loss exponent, and $\epsilon$ is a centered Gaussian random variable of variance $\sigma_{sh}^2$ representing the large scale fading or shadowing.

By assuming that the RSS observation follows (2.4) with known $P_0$ and $n_p$, a lower bound of the variance of unbiased distance estimators from the RSS is given by

$$var(\hat{d}) \geq \sigma_{sh}^2 d^2 (\frac{\log 10}{10 n_p})^2. \tag{2.5}$$

This bound corresponds to the CRB and can be computed following the indications provided in Appendix A. This bound is increasing with the square of the distance $d$. When $n_p = 3$ and $\sigma_{sh} = 4dB$, its value is about $9.4m^2$ for d=$10m$ and about $942m^2$ for $d = 100m$ which is very high. This inaccuracy is the main disadvantage of RSS ranging and positioning.

In fact, estimation techniques allow estimating the location directly from several RSS measurements without the need for estimating the distances [40], but the order of accuracy of distance estimation provides information about the relevance of a particular RSS measurement to the location estimation.

The shadowing can be modeled as a combination of two parts : A first part which is random in space and a second part which is random in space and time. The knowledge

of the values of the first part at the different locations can highly improve the localization accuracy. The fingerprinting method consists in constructing the RSS or shadowing maps via measurements campaigns or calibration tools. This method is presented in the next section.

## 2.6 Fingerprinting method

This method performs in two phases :

- In the offline phase, data collection is performed where location dependent parameters of the signals are measured at selected locations, and then processed and saved in a database.
- In the online phase, the measurements vector is compared with the database values where a cost function is used in the comparison, and the target location is computed from the locations associated to the database entries that best match the measurements.

Fingerprinting has been mostly experimented with RSS in both WLAN [41] and cellular [42] networks. Other location dependent parameters have been also studied such as the power delay profile [43]. Calibration tools, such as ray tracing, can be equally used in the offline phase, but their impact on the positioning accuracy has to be considered.

The online phase estimation techniques can be classified as deterministic or probabilistic. An example of a deterministic technique is the $k$-nearest neighbors averaging where the coordinates of the $k$ best matching locations are averaged to give the estimated coordinates [41]. In the probabilistic techniques, the calibration measurements made during the offline phase are used to derive the statistical models of the location-dependent parameters, and probabilistic inference methods are used to estimate the location during the online phase. In [44], a spatial correlation model of the RSS is assumed and the calibration measurements made at selected locations are used to compute the posterior distribution of the RSS at any other space location.

The main difficulties of the fingerprinting method reside in constructing the database and maintaining it up-to-date by repeating the calibration process to track changes in the environment. Few solutions considered the issue of reducing the calibration effort. One of these solutions consists of reducing the number of locations at which measurements are collected [45], at the expense of an accuracy reduction.

## 2.7 Cooperative localization and tracking

The localization methods we have described in the two previous sections consider a target node connected to (or communicating with) several anchor nodes and the location is computed from the pair-wise measurements made with these anchor nodes. In many scenarios, target nodes may not be able to connect to a sufficient number of anchor nodes. Additionally, even when a sufficient number of anchors is available, the achievable accuracy may not meet the intended application requirements. To overcome these problems, other localization methods have been developed by exploiting other kinds of measurements such as intra-node

measurements in the case of moving target nodes and inter-node measurements between pairs of target nodes.

According to the kind of information used, localization algorithms can be classified into one of the following four classes :

1. Non-cooperative static localization : The location is computed solely from the measurements made with anchor nodes. If the target node is moving, the location at a given time instant is computed without using a motion model or the measurements made at the previous time instants.
2. Non-cooperative dynamic localization : The algorithms that fall in this category perform a location tracking over time. Tracking allows improving the accuracy by exploiting the measurements made at multiple time instants. The dependence between these measurements is derived from a motion model specifying the dependence between the target node locations at the different time instants, as for example, a mobile target cannot travel a long distance over a short time period. Intra-node measurements can be used in tracking for enhancing the motion model. These measurements can be delivered by inertial navigation sensors (INS) such as accelerometers and gyroscopes. Additionally, tracking allows reducing the impact of some errors that are spatially correlated such as the NLoS bias in ToA [46] and the shadowing in RSS [4].
3. Cooperative static localization : Cooperative localization allows circumventing the need for high density of anchor nodes and high transmit power for long range connectivity. It is based on using the measurements made between pairs of target nodes in addition to those made with anchor nodes, and simultaneously estimating multiple target locations. Figure 2.7 shows an example of a two-dimensional network where the target nodes cannot compute their location by trilateration, but using all the inter-node ranging measurements makes the computation feasible.

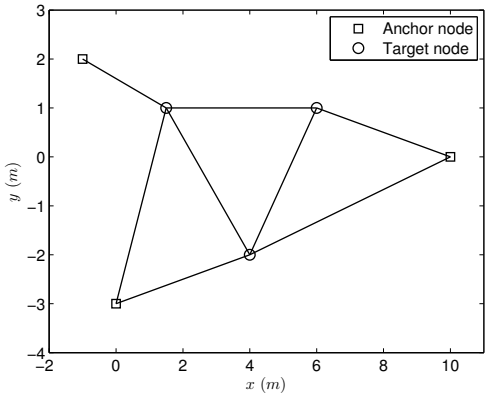



***Figure 2.7*** — A 2-dimensional network of three anchor nodes and three target nodes. Two nodes are connected by an edge if they are performing a pair-wise ranging measurement. Cooperation allows the computation of the target nodes locations.

Furthermore, it is proven in [47] that adding a target node to a network can only improve the location estimation accuracy for the remaining nodes.

Several names have been attributed to cooperative localization in the literature such as self-localization, multihop and ad-hoc localization, and self-calibration, and many algorithms have been developed.

An application example of indoor localization is shown in Figure 2.8. It considers femto BSs placed in several apartments. The locations of the femto BSs need to be known in order to use them as anchor nodes for locating mobile terminals or for communications enhancement. Since the femto BSs are intended to be plug and play devices, it is inappropriate to ask the users to manually localize them, and cooperative localization can be the alternative.

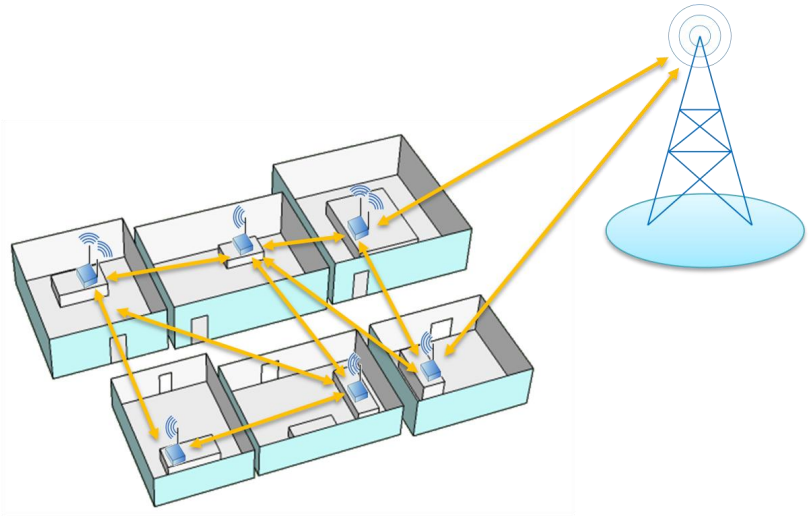


***Figure 2.8*** — Femto BSs self-localization.

4. Cooperative dynamic localization : A tracking operation of the mobile targets is performed while at the same time the cooperation is maintained by exploiting the inter-node measurements. Hence, the algorithms that fall in this category can achieve the best performance. In [48], cooperative dynamic localization is seen as an extension of cooperative static localization where there are cooperations in both space and time.

   In the example of Figure 2.9, the measurements made at mobile stations can be used for estimating the locations of these mobile stations and improving the estimated locations of the femto BSs.

## 2.8 Conclusion

In this chapter, we reviewed several wireless localization systems. Apart from the GNSS and its augmentation, the sensor networks and the RFID, these systems are based on communication and broadcast infrastructures. Originally, these systems were not designed for localization purposes. Many standardization efforts have been made in order to make them suitable for localization and to improve the delivered localization accuracy, such as the introduction of the UWB ranging in the 802.15.4a amendment and the timing reference units in

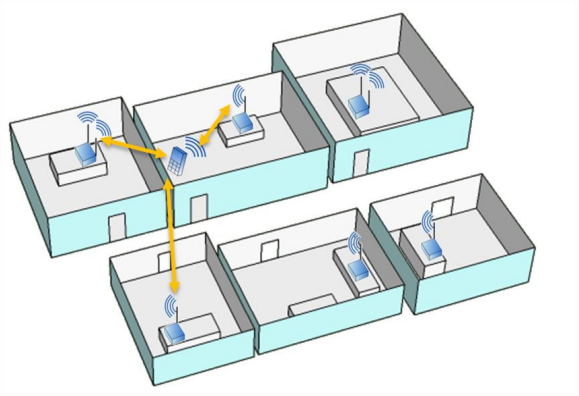


***Figure 2.9*** — User terminal localization by femto BSs.

the cellular systems. Nonetheless, network planning strategies can be developed and applied to offer some targeted quality of localization in specific deployment areas. We described several location based applications covering several fields of activities. We also presented several fundamental localization solutions. These solutions fall in the category of non-cooperative static localization, according to the classification provided at the end of this chapter.

In the following chapters, cooperative static and non-cooperative dynamic localization (i.e., the second and the third categories of the classification previously described) are considered from a statistical signal processing perspective.

CHAPTER 3

# Rigidity, Identifiablity and Localizability in Cooperative Network Localization

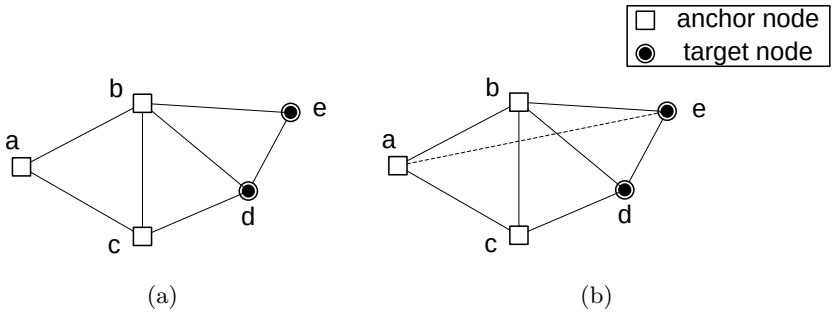
## 3.1 Introduction

Cooperative network localization aims at determining the locations of wireless communication devices (also called nodes), in an Euclidean space, which are consistent with a set of pair-wise measurements made between the nodes. In general, measurements are not available for all the pairs of nodes, i.e., the network graph is not complete, and as a consequence, there may not be enough information to estimate the different locations without ambiguities. Even with the knowledge of the true separating distances between pairs of nodes, ambiguities exist when there are multiple feasible solutions of the target node coordinates verifying the distance constraints. In Figure 3.1(a), for example, the 2-dimensional network consists of three anchor nodes $a$, $b$ and $c$ of known positions and two target nodes $d$ and $e$ of unknown positions. Two nodes are connected by an edge if their true separating distance is known. The target nodes are not uniquely solvable : Node $e$ can be reflected in the line passing through nodes $b$ and $d$, and nodes $d$ and $e$ can be reflected in the line passing through nodes $b$ and $c$. By connecting nodes $a$ and $e$, the ambiguities are eliminated and the nodes positions become uniquely solvable. Graph rigidity theory and semidefinite programming define sufficient unique solvability conditions of the target nodes, assuming perfect ranging measurements.

In probabilistic estimation, where the pair-wise measurements are affected by random fluctuations, the ambiguity can be defined using the identifiability theory which concerns the possibility of drawing inference about unknown parameters from probability distributions.

The limitation of the number of pair-wise measurements can be due to several factors, such as :

– Propagation issues : The decrease of the average power and the presence of noise limit the connectivity between the nodes. The non-line-of-sight (NLoS) propagation introduces a bias in the measured distances degrading the localization accuracy. It is shown




(a) (b)

***Figure 3.1*** — A 2-dimensional network of three anchor nodes and two target nodes. In (a) the target nodes have ambiguities on their position solutions. And in (b) the ambiguities are eliminated by adding an edge between nodes $a$ and $e$.

in [49] that by identifying and eliminating NLoS measurements, the accuracy can be improved .

– Link selection : In densely crowded networks, the computational complexity of the localization algorithms can be reduced by discarding some measurements, without degrading the accuracy. This procedure is also beneficial to distribute the localization algorithm as it allows a reduction of the traffic overhead [50, 51].
– Heterogeneous networks : Two nodes with non-compatible physical interfaces may not be able to perform a pair-wise measurement or communicate with each other.

Before applying a localization algorithm, it is important to know the conditions that guarantee an absence of ambiguities, since they can result in high estimation errors, especially in large networks where the computed spatial configuration can be much different from the true one due to error accumulation. Additionally, the detection of the ambiguities allows us to deal with them either by eliminating some nodes from the set of nodes to be localized and thus improving the robustness of the solution [52, 53], or by mitigating them by making additional measurements [54] or by using additional a priori information [55].

The objective of this chapter is to study, in static networks, the unique solvability conditions based on the knowledge of the true distance values, and the identifiability conditions based on noisy ranging measurements. Two approaches are considered when studying the unique solvability : Graph rigidity theory and semidefinite programming (SDP). Moreover, in studying the identifiability properties, the unknown locations are treated as deterministic parameters and probabilistic models of the measurements are assumed. We start in Section 3.2 by reviewing the graph rigidity theory in the context of cooperative localization. Then, in Section 3.3, the presence of noise in the measurements is considered and we derive a relationship between the rigidity and the Fisher information matrix (FIM), and the conditions of non-singularity of the FIM. The identifiability is also studied in this section, where we derive correspondences between the rigidity and the identifiability. These correspondences allow checking identifiability properties by using graph rigidity tools. SDP is the subject of Section 3.4, where the state of the art method is described, and a new SDP based technique is proposed to improve the detection of the uniquely solvable nodes. In Section 3.5, numerical results about the statistical occurrence of rigid and uniquely solvable networks are provided.

Finally, concluding remarks are drawn in Section 3.6.

## 3.2 Graph rigidity theory

Graph rigidity theory has been studied by mathematicians since the $19^{th}$ century. It is related to the graph realization problem, which consists in computing relative locations of vertices in an Euclidean space verifying a set of inter-vertex distance values. It tries to determine whether a graph has a unique realization in a given dimension. This theory has applications in several disciplines such as chemistry (e.g., structures of molecules) and construction engineering (e.g., bar-and-joint structures) [56, 57, 58, 59, 60, 61].

The results of this theory have been applied to network localization [62, 1], and provide sufficient conditions to guarantee unique position solutions. In this section we present a survey of these results, and we start by providing some notations and preliminary definitions.

### 3.2.1 Definitions

Consider a $d$-dimensional network $\mathcal{L}$ of size $N$ consisting of $m < N$ anchor nodes of known locations labeled 1 through $m$ and $n = N - m$ target nodes of unknown locations labeled $m+1$ through $N$ lying in a $d$-dimensional Euclidean space, where $d = 1, 2$ or 3. The position of node $i$ is denoted by the column vector $\mathbf{x}_i \in \mathbb{R}^d$, where $\mathbb{R}$ is the field of real numbers. The distance between nodes $i$ and $j$ is denoted by $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$, or in other words the Euclidean norm of $\mathbf{x}_i - \mathbf{x}_j$. We assume that $\mathbf{x}_i \neq \mathbf{x}_j$ for all $1 \leq i < j \leq N$ throughout this chapter.

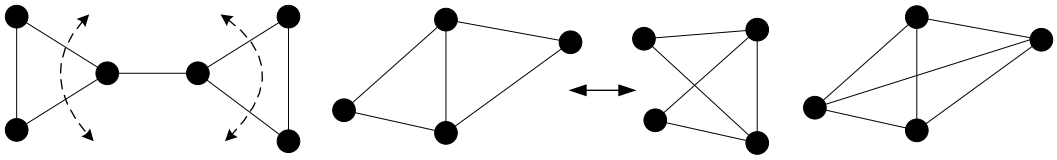
A graph $G = (V, E)$ of vertex set $V = \{1, 2, \cdots, N\}$ and edge set $E = \{(i, j)|\ i, j \in V$ and $i$ and $j$ are neighbors$\}$ is associated to the network $\mathcal{L}$ as follows : Vertex $i$ is associated to network node $i$, and two vertices are connected by an edge if they correspond to anchor nodes or if they are performing a pair-wise ranging measurement, i.e., either their separating distance is known or it can be estimated. The terms node and vertex will be used interchangeably.

A $d$-dimensional framework $(G, \mathbf{p})$ is defined as being a pair consisting of a graph $G = (V, E)$ and a mapping $\mathbf{p}$ from $V$ to $\mathbb{R}^d$ such that $\mathbf{p}(i) \neq \mathbf{p}(j)$ for all $(i, j) \in E$.

For two frameworks $(G, \mathbf{p})$ and $(G, \mathbf{q})$ of the same dimension, the following terminology is introduced in [60] :

- $(G, \mathbf{p})$ and $(G, \mathbf{q})$ are equivalent if $\|\mathbf{p}(i) - \mathbf{p}(j)\| = \|\mathbf{q}(i) - \mathbf{q}(j)\|$ for all $(i, j) \in E$.
- $(G, \mathbf{p})$ and $(G, \mathbf{q})$ are congruent if $\|\mathbf{p}(i) - \mathbf{p}(j)\| = \|\mathbf{q}(i) - \mathbf{q}(j)\|$ for all $i, j \in V$.
- $(G, \mathbf{p})$ is globally rigid if every framework which is equivalent to $(G, \mathbf{p})$ is congruent to $(G, \mathbf{p})$ (e.g., Figure 3.2(c)).
- $(G, \mathbf{p})$ is rigid if there exists an $\epsilon > 0$ such that every framework $(G, \mathbf{q})$ equivalent to $(G, \mathbf{p})$ and satisfying $\|\mathbf{p}(i) - \mathbf{q}(i)\| < \epsilon$ for all $i \in V$ is congruent to $(G, \mathbf{p})$ (e.g., Figure 3.2(b)).
- A framework is flexible if it is not rigid, i.e., susceptible to continuous deformations

(e.g., Figure 3.2(a)).



(a) Flexible (b) Rigid (c) Globally rigid

***Figure 3.2*** — Three 2-dimensional frameworks.

A framework $(G, \mathbf{p})$ is said to be a realization of the network $\mathcal{L}$ if they both have the same dimension and $\|\mathbf{p}(i) - \mathbf{p}(j)\| = d_{i,j}$ for all $(i, j) \in E$, where $G$ is the graph associated to the network.

A framework $(G, \mathbf{p})$ is said to be the fundamental realization of the network $\mathcal{L}$ if it is a realization framework verifying $\mathbf{p}(i) = \mathbf{x}_i$ for all $i \in V$.

The network localization problem is defined as

$$\begin{array}{ll} \text{find} & \mathbf{x}_j,\ j = m+1, \ldots, N, \\ \text{s.t.} & \mathbf{x}_i,\ i = 1, \ldots, m, \\ & \|\mathbf{x}_u - \mathbf{x}_v\| = d_{u,v},\ \forall (u, v) \in E. \end{array} \tag{3.1}$$

The network $\mathcal{L}$ is said to be uniquely solvable if problem (3.1) has a unique solution, i.e., all the nodes are uniquely solvable.

**Theorem 3.1** ([62])**.** *The d-dimensional network $\mathcal{L}$ is uniquely solvable if and only if any realization framework of $\mathcal{L}$ is globally rigid and the network has at least $d + 1$ anchor nodes in general positions.*

Nodes are said to be in general positions if they have affinely independent position vectors, i.e., not collinear for $d = 2$ and not coplanar for $d = 3$.

A graph $G' = (V', E')$ is a subgraph of $G = (V, E)$ if $V' \subseteq V$ and $E' = \{(i, j) \in E | i, j \in V'\}$. The framework $(G', \mathbf{p})$ is then a subframework of $(G, \mathbf{p})$.

A node is said to be uniquely solvable if its solution is the same for every solution of problem (3.1). A direct result of Theorem 3.1 is the following corollary :

**Corollary 3.1.** *A node is uniquely solvable if it belongs to a globally rigid subframework having at least $d + 1$ anchor nodes in general positions, for any realization framework of $\mathcal{L}$.*

In fact, the condition provided by Corollary 3.1 is only sufficient for unique solvability, but no sufficient and necessary conditions are known yet.

Graph rigidity theory provides tools for checking rigidity and global rigidity of frameworks, and thus, answering the question whether the network localization problem is uniquely solvable, and if not, finding a portion of the uniquely solvable nodes. Now, we present the rigidity matrix which is an important tool and which will be used later in deriving the relationship between the rigidity and the FIM.

### 3.2.2 Rigidity matrix

A smooth motion of a framework $(G, \mathbf{p})$ is defined as being a function $\mathbf{P}$ from $\mathbb{R}^+ \times V$ to $\mathbb{R}^d$ verifying the three following points :

1. $\mathbf{P}(t, v)$ is a differentiable function of $t$, where $t$ can be seen as time.
2. $\mathbf{P}(0, v) = \mathbf{p}(v)$ for all $v \in V$.
3. $\|\mathbf{P}(t, v) - \mathbf{P}(t, u)\| = \|\mathbf{p}(v) - \mathbf{p}(u)\|$ for all $(u, v) \in E$ and all $t \geq 0$.

Thus, for $t = 0$, $(G, \mathbf{p})$ is rigid if and only if every smooth motion of $(G, \mathbf{p})$ results in a framework which is congruent to $(G, \mathbf{p})$.

Condition 3, described above, can be rewritten as

$$\|\mathbf{P}(t, v) - \mathbf{P}(t, u)\|^2 = \text{constant} \quad \text{w.r.t. } t \tag{3.2}$$

and by differentiating with respect to $t$, we obtain

$$[\mathbf{P}(t, v) - \mathbf{P}(t, u)]^T \left[\mathbf{P}'(t, v) - \mathbf{P}'(t, u)\right] = 0 \tag{3.3}$$

where $\mathbf{P}'(t, v)$ represents the instantaneous velocity of the function $\mathbf{P}$, and $^T$ denotes the transpose operator. At time $t = 0$, denote $\mathbf{p}'(v) = \mathbf{P}'(0, v)$ :

$$[\mathbf{p}(v) - \mathbf{p}(u)]^T \left[\mathbf{p}'(v) - \mathbf{p}'(u)\right] = 0 \text{ for all } (u, v) \in E \tag{3.4}$$

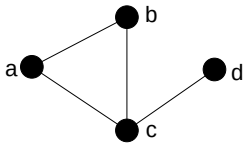
which can be written as

$$\begin{bmatrix} \mathbf{p}(v) - \mathbf{p}(u) \\ \mathbf{p}(u) - \mathbf{p}(v) \end{bmatrix}^T \begin{bmatrix} \mathbf{p}'(v) \\ \mathbf{p}'(u) \end{bmatrix} = 0 \text{ for all } (u, v) \in E. \tag{3.5}$$

An assignment of velocities $\mathbf{p}'(v)$, for all $v \in V$, satisfying the system of equations (3.5) is called an infinitesimal motion of $(G, \mathbf{p})$. This system of equations can be used to determine a basis for all infinitesimal motions.

The rigidity matrix $M(G, \mathbf{p})$ of $(G, \mathbf{p})$ is the matrix of the coefficients of (3.5). Its size is $|E| \times d|V|$, where $|.|$ denotes the cardinality of a set. Each edge in the graph indexes a row in this matrix and each vertex indexes $d$ consecutive columns. For a row indexed by an edge $(u, v) \in E$, the entries corresponding to vertex $u$ are given by $(\mathbf{p}(u) - \mathbf{p}(v))^T$ and those corresponding to vertex $v$ are given by $(\mathbf{p}(v) - \mathbf{p}(u))^T$, and the remaining entries are equal to zero. A framework $(G, \mathbf{p})$ and its rigidity matrix are shown in Figure 3.3, where $\mathbf{p}(i) = [x_i \ , \ y_i]^T \in \mathbb{R}^2$.

The infinitesimal motions of a framework constitute the null space of the rigidity matrix. This null space always has a non-zero rank as it contains the trivial motions, i.e., rotations and translations of the whole framework, and thus, the rigidity matrix cannot have a full column rank. For a $d$-dimensional framework with at least $d$ nodes, that are not all collinear for $d = 3$, there are $d$ independent trivial translations and $d(d-1)/2$ independent trivial rotations. Thus, the total number of independent trivial motions is $d(d+1)/2$.




(a)

$$\begin{bmatrix} x_a - x_b & y_a - y_b & x_b - x_a & y_b - y_a & 0 & 0 & 0 & 0 \\ x_a - x_c & y_a - y_c & 0 & 0 & x_c - x_a & y_c - y_a & 0 & 0 \\ 0 & 0 & x_b - x_c & y_b - y_c & x_c - x_b & y_c - y_b & 0 & 0 \\ 0 & 0 & 0 & 0 & x_c - x_d & y_c - y_d & x_d - x_c & y_d - y_c \end{bmatrix}$$

(b)

***Figure 3.3*** — A 2-dimensional network and its rigidity matrix. The first row corresponds to edge $(a, b)$, the second to edge $(a, c)$, the third to edge $(b, c)$ and the fourth to edge $(c, d)$

For a number of nodes $N \geq 2$, let

$$S(N,d) = \begin{cases} dN - d(d+1)/2 & \text{if } N \geq d+2 \\ N(N-1)/2 & \text{if } N \leq d+1 \end{cases}. \tag{3.6}$$

Notice that for $N \geq d+2$, $S(N, d)$ is equal to the number of the columns of the rigidity matrix minus the number of the trivial motions. The minimum rank of the null space of $M(G, \mathbf{p})$ is equal to the number of trivial motions.

**Theorem 3.2** ([60])**.** *Let $(G, \mathbf{p})$ be a $d$-dimensional framework with $N \geq 2$ nodes. Then $rank M(G, \mathbf{p}) \leq S(N, d)$. Furthermore, if equality holds, then $(G, \mathbf{p})$ is rigid.*

In other words, when equality holds, the only possible motions are the trivial ones and the framework cannot have a continuous deformation.

$(G, \mathbf{p})$ is said to be infinitesimally rigid if $\text{rank} M(G, \mathbf{p}) = S(N, d)$. According to Theorem 3.2, infinitesimal rigidity is a sufficient condition for rigidity. Figure 3.4 shows that infinitesimal rigidity is not equivalent to rigidity, where the framework is rigid and the rank of the rigidity matrix is less than $S(N, d)$. In fact, this framework is particular (i.e., non-generic as we will see) as nodes $a$, $b$ and $c$ are collinear, and non-trivial velocity of node $b$ orthogonal to the line passing through the aligned nodes belongs to the null space of the rigidity matrix, though this deformation is not allowed by the rigid framework. Infinitesimal rigidity is equivalent to rigidity for a category of frameworks called generic frameworks.

### 3.2.3 Generic frameworks

A framework $(G, \mathbf{p})$ is said to be generic if its rigidity matrix has the maximum rank among all possible frameworks $(G, \mathbf{q})$ of the same dimension [63]. For a $d$-dimensional framework of $N$ nodes, the mappings of the nodes can be seen as a single point in $\mathbb{R}^{Nd}$. The generic frameworks form an open dense subset of $\mathbb{R}^{Nd}$, and the non-generic frameworks form an

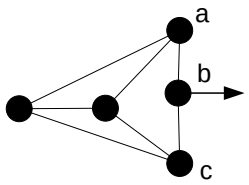



***Figure 3.4*** — A 2-dimensional non-generic rigid framework which is not infinitesimally rigid.

algebraic subset of $\mathbb{R}^{Nd}$. Thus, if we draw randomly $N$ uniformly distributed positions in an open set of $\mathbb{R}^d$, the obtained framework is generic with probability one. The framework of Figure 3.4 is non-generic as there are nodes lying on the same line.

For generic frameworks, rigidity is equivalent to infinitesimal rigidity :

**Theorem 3.3** ([64])**.** *Let $(G, \mathbf{p})$ be a d-dimensional generic framework. Then $(G, \mathbf{p})$ is rigid if and only if $(G, \mathbf{p})$ is infinitesimally rigid.*

For a graph $G$, the rank of $M(G, \mathbf{p})$ will be maximized when $(G, \mathbf{p})$ is generic, and hence, rank$M(G, \mathbf{p})$ is the same for all generic frameworks sharing the same graph in a given dimension. This fact allows us to deduce the following theorem :

**Theorem 3.4** ([65])**.** *If a graph has a single infinitesimally rigid framework, then all its generic frameworks are rigid.*

In other words, given a graph $G$, if $(G, \mathbf{p})$ is a generic and rigid framework in $\mathbb{R}^d$ and $(G, \mathbf{q})$ is another generic framework in $\mathbb{R}^d$, then $(G, \mathbf{q})$ is rigid. Thus, rigidity is a combinatorial property of the graph in all dimensions.

Next, we will present solutions for testing generic rigidity.

### 3.2.4 Generic rigidity testing

We restrict our attention to generic frameworks. As previously stated, given a dimension $d$, the rigidity is not a characteristic of a specific framework, but rather a characteristic of the graph. Combinatorial polynomial time algorithms for testing rigidity are only known for $d = 1$ and 2.

For the case $d = 1$, rigidity is equivalent to graph connectivity. A simple algorithm for testing graph connectivity is depth first search with complexity $O(N)$, $N$ being the number of vertices.

For the case $d = 2$, Laman's theorem characterizes rigidity [66]. Before stating this theorem, we define the terms of independent and redundant edges. By setting $d = 2$ in (3.6), we get $S(N, 2) = 2N - 3$ which means that there are at most $2N - 3$ linearly independent rows in the rigidity matrix of any generic framework. The edges whose corresponding rows are linearly independent are said to be independent. Each independent edge eliminates a degree of freedom, or a continuous deformation, in the framework structure. The other edges are

redundant edges : Their removal does not affect the rank of the rigidity matrix. The presence of $2N-3$ independent edges is a sufficient and necessary condition for rigidity. As rigidity is a graph theoretic property, the independence of edges is also a graph theoretic property.

**Theorem 3.5** (Laman [66]). *The edges of a graph $G=(V,E)$ are independent in two dimensions if and only if no subgraph $G'=(V',E')$ of $G$ has more than $2N'-3$ edges, where $N'=|V'|$.*

**Corollary 3.2** ([56]). *A graph with $2N-3$ edges is rigid in two dimensions if and only if no subgraph $G'$ has more than $2N'-3$ edges.*

Using Laman's condition, in its original form, to characterize rigidity results in a poor algorithm as it involves counting the edges in every subgraph, of which there are an exponential number. There are several combinatorial algorithms for testing rigidity when $d=2$. The best are of $O(N^2)$ complexity [56, 57, 58]. In [56], an $O(N^2)$ algorithm based on bipartite matching is developed. This algorithm is recast in [57] in terms of a simple pebble game that allows the identification of all the rigid subgraphs and the redundant edges.

For the case $d=3$, there is no theoretical characterization of rigidity similar to Laman's theorem.

Moreover, the rigidity matrix rank test can be used in all dimensions : When nodes positions are not known, symbolic calculations can be used or a randomly generated generic framework. The computation of the rank can be done using QR decomposition with complexity $O(|E|N^2)$.

**Identification of rigid subgraphs using the rigidity matrix**

Now, we derive two solutions for identifying the rigid subgraphs using the rigidity matrix for the cases $d=2$ and $d=3$.

For the case $d=2$, it can be shown, using Laman's theorem, that the union of two rigid graphs sharing an edge forms a rigid graph [57]. Thus, if we select an edge in a 2-dimensional framework and fix it by preventing its two endpoints from moving, then all the nodes that become fixed belong to the same rigid subgraph containing this edge. These nodes can be identified using the rigidity matrix by removing the columns of the two fixed nodes and computing a basis of the null space of the obtained matrix. The nodes that cannot move have their entries equal to zero in all the vectors of this basis. For example, consider the framework of Figure 3.3 and let $\mathbf{p}(a)=[-1\ 0]^T$, $\mathbf{p}(b)=[0\ 1]^T$, $\mathbf{p}(c)=[0\ -1]^T$ and $\mathbf{p}(d)=[1\ 0]^T$. By fixing nodes $a$ and $b$ and removing their columns from $M(G,\mathbf{p})$, we obtain the following rank-3 matrix :

$$M_{a,b}(G,\mathbf{p})=\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & -2 & 0 & 0 \\ -1 & -1 & 1 & 1 \end{bmatrix}. \tag{3.7}$$

The null space of $M_{a,b}(G,\mathbf{p})$ is generated by the vector $[0\ 0\ -1\ 1]^T$. Node $c$ has its entries

equal to zero in this vector, where the first two entries correspond to $x_c$ and $y_c$ and the last two entries correspond to $x_d$ and $y_d$, and thus, nodes $a$, $b$ and $c$ form a rigid subgraph.

For the case $d = 3$, a reasoning similar to the case $d = 2$ can be applied according to the following theorem :

**Theorem 3.6** ([58])**.** *(Gluing Theorem) If* $(G_1, \mathbf{p_1})$ *and* $(G_2, \mathbf{p_2})$ *are d-dimensional generically rigid frameworks sharing at least d vertices, then the union of the two frameworks* $(G, \mathbf{p}) = (G_1 \cup G_2, \mathbf{p_1} \cup \mathbf{p_2})$ *is generically rigid.*

Thus for the case $d = 3$, if we fix three vertices connected to each other by three edges, all the vertices that become fixed form a rigid subgraph with them. This method cannot be always applied since the existence of three vertices connected to each other is not always guaranteed (e.g., the complete bipartite graph $K_{4,6}$ is rigid [58]). Notice that the union of multiple rigid subgraphs sharing an edge is not necessarily rigid. By fixing two vertices connected by an edge, we can determine the subgraph of all the vertices that can rotate around the fixed edge, i.e., their velocities are orthogonal to this edge. Denote by $G'$ the union of this subgraph and the two fixed vertices. Then a rigid subgraph containing the two fixed vertices must be a subgraph of $G'$.

### 3.2.5 Generic global rigidity testing

In this section, we will focus again on generic frameworks. It is shown in [61] that generic global rigidity is a graph property in all dimensions. Indeed, the rigidity is a necessary condition for global rigidity.

A graph is $k$-connected if it remains connected after the removal of any $k - 1$ vertices.

For the case $d = 1$, a graph is globally rigid if and only if it is completely connected on two vertices or it is 2-connected [60]. Depth first search can be used to test 2-connectivity in linear time.

For the case $d = 2$, a graph of 3 or less vertices is globally rigid if and only if it is completely connected. And for larger graphs, two conditions are proven to be necessary [56] and together sufficient [67] for global rigidity : 3-connectivity and redundant rigidity.

- 3-connectivity : This condition is necessary to avoid reflections. In Figure 3.1(a), the graph is only 2-connected since the removal of nodes $b$ and $c$ makes it disconnected, and a portion of the framework can be reflected in the line connecting these two nodes. An efficient graph connectivity order test of complexity at most $O(N^{1/2}|E|)$ is described in [68].
- Redundant rigidity : A graph is said to be redundantly rigid if it remains rigid after the removal of any of its edges. Figure 3.5 shows a framework whose associated graph is 3-connected but not redundantly rigid (it becomes flexible when the dashed edge is removed). This framework has another equivalent but non-congruent framework.

**Identification of the globally rigid subgraphs when** $d = 2$ **:** The globally rigid subgraphs are the 3-connected and redundantly rigid subgraphs. A recursive algorithm for finding them is described in Table 3.1 [1].

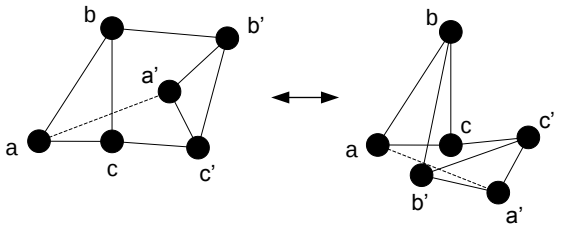



***Figure 3.5*** — A 3-connected and non-redundantly rigid graph. The space dimension is $d = 2$.

```
0 :  Function FindGRSubgraphs (graph G)
1 :    GRSubgraphs = empty
2 :    for each rigid subgraph G' of G
3 :      if G' is not 3-connected
4 :        recurse on each 3-connected subgraph of G'
5 :        append the globally rigid subgraphs to GRSubgraphs
6 :      else if G' is not redundantly rigid
7 :        recurse on each redundantly rigid subgraph of G'
8 :        append the globally rigid subgraphs to GRSubgraphs
9 :      else
11 :       append G' to GRSubgraphs
12 :   return GRSubgraphs
```

***Table 3.1*** — Recursive algorithm for identifying the globally rigid subgraphs in two dimensions [1].

For the case $d = 3$, a graph of 4 or less vertices is globally rigid if and only if it is completely connected. For larger graphs, the conditions of 4-connectivity and redundant rigidity are necessary but not sufficient.

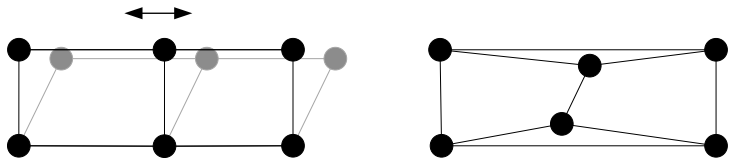
A sufficient [59] and necessary [61] condition for generic global rigidity does exist and applies for all dimensions $d$. To check this condition, the positions of all the nodes need to be known. When the positions are not known, a randomly generated generic framework can be used.

### 3.2.6 Non-generic frameworks

In the previous sections, we described algorithms for checking generic rigidity and generic global rigidity. Generic frameworks are only a subset of the frameworks for a given dimension. In [69], it is shown that it is NP-hard to decide whether a $d$-dimensional framework is globally rigid even for $d = 1$.

The generic graph rigidity property does not imply the rigidity of non-generic frameworks. Figure 3.6(a) shows a non-generic and flexible framework and Figure 3.6(b) shows a generic rigid framework. Both frameworks have the same graph.

When the positions are not known, one may not be able to assert whether the nodes are in generic positions. If the positions are randomly generated according to the uniform distribution in an open set of $\mathbb{R}^d$, then the obtained network is generic with probability one.



(a) (b)

***Figure 3.6*** — Two 2-dimensional frameworks sharing the same graph. (a) Non-generic and flexible. (b) Generic and rigid.

### 3.2.7 Networks with known vertical elevations

In some localization applications, the vertical elevations or the altitude of the nodes might be known, e.g., access points or femtocells placed on a desk or fixed to the ceiling, equipments carried by pedestrians. In this case, the rigidity and global rigidity conditions are not the same as those when the elevations are not known.

Let $(G, \mathbf{p})$ and $(G, \mathbf{q})$ be two 3-dimensional frameworks such that $\mathbf{p}(i)$ and $\mathbf{q}(i)$ have the same elevations for all $i \in V$, and let $(G, \bar{\mathbf{p}})$ and $(G, \bar{\mathbf{q}})$ be their orthogonal projections on the horizontal plane. It can be easily shown that $(G, \bar{\mathbf{p}})$ and $(G, \bar{\mathbf{q}})$ are equivalent (respectively congruent) if and only if $(G, \mathbf{p})$ and $(G, \mathbf{q})$ are equivalent (respectively congruent). Thus, the rigidity and global rigidity conditions when the elevations are known are the same as those of planar networks.

### 3.2.8 Unique solvability

The $d$-dimensional network $\mathcal{L}$ is said to be generic if its fundamental realization framework is generic.

Assume here that the network $\mathcal{L}$ is generic. According to Theorem 3.1, the network $\mathcal{L}$ is uniquely solvable if and only if the associated graph is globally rigid and there are at least $d+1$ anchor nodes (these anchor nodes are in general positions since the network is generic [1]). When the graph is not globally rigid, a node is uniquely solvable if it belongs to a globally rigid subgraph containing the vertices of at least $d+1$ anchor nodes. This condition is only sufficient and allows identifying only a subset of the uniquely solvable nodes. In Figure 3.7(a), node $a$ is uniquely solvable while it does not belong to a globally rigid subgraph. In such a case, the unique solvability depends on the distance values as illustrated by Figure 3.7(b) where the graph is the same as that of Figure 3.7(a) but node $a$ is not uniquely solvable.

Sufficient and necessary conditions for generic unique solvability are not yet known.

1. If the $d+1$ nodes are collinear when $d = 2$ or coplanar when $d = 3$, then the network ceases to be generic.

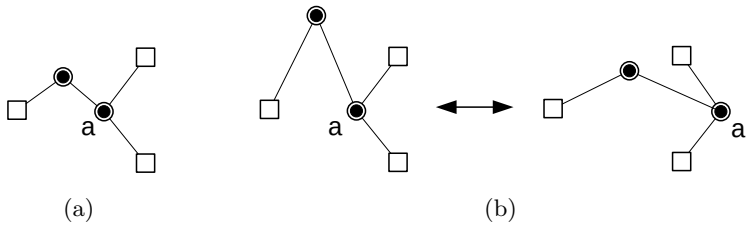



***Figure 3.7*** — Two 2-dimensional networks. In (a), node $a$ is uniquely solvable while it does not belong to a globally rigid subgraph. In (b), the distance values are changed and the node is no more uniquely solvable.

### 3.2.9 Distributed construction of rigid and globally rigid networks

In the case of distributed cooperative localization, the computational tasks are distributed among the nodes and there might not be a central node at which the connectivity information or the measurements are collected. In such situations, it is desirable to have a distributed algorithm for finding the uniquely solvable nodes or rigid subgraphs.

For rigidity testing, the distributed application of the pebble game algorithm requires an important overhead of message exchanges and some coordination mechanism for defining the order of edges appearance. For global rigidity testing, distributed testing of 3-connectivity and redundant rigidity seems to be very cumbersome.

Fortunately, there are categories of networks for which the determination of rigid and globally rigid subgraphs and uniquely solvable nodes can be easily carried out in a distributed sequential manner.

A network constructed by sequentially adding to a rigid network new nodes connected to at least $d$ nodes of it is rigid. For $d = 2$, the networks obtained by this way are called bilateration networks. We mention that not all rigid networks are bilateration ones : The framework of Figure 3.6(b) is rigid but the associated network is not a bilateration one.

Similarly, a network constructed by inserting a node connected to $d + 1$ nodes in general positions of a globally rigid network is globally rigid. For $d = 2$, these networks are called trilateration networks [53], and as for the rigidity, trilateration networks are only a subset of the globally rigid networks.

In [70], it is shown that another category of networks, called wheel networks, are globally rigid and a distributed algorithm is developed for identifying them. Trilateration graphs are a special case of wheel graphs.

Finally, we mention the following necessary condition for unique solvability [1] : If a node is uniquely solvable, then their exist $d+1$ disjoint paths connecting this node to $d+1$ anchor nodes, where two paths are called disjoint if they are not sharing any common node. For example, when $d = 2$, if a node has only two disjoint paths to two anchor nodes in a network containing three anchor nodes, then the associated graph is 2-connected and not 3-connected. This condition can be used to identify non-uniquely solvable nodes.

## 3.3 Rigidity , FIM and identifiability

In this section, the unknown target nodes locations in the network $\mathcal{L}$, which has been defined in the previous section, are considered as deterministic parameters, and the ranging measurements are considered as random due to the presence of noise.

Before providing the novel contributions of this section, which are the correspondence between the Fisher information matrix and the rigidity and the correspondences between the rigidity and the identifiability, we start by presenting the FIM and its computation.

Let $\theta = [\mathbf{x}_{m+1}^T, \cdots, \mathbf{x}_N^T]^T$ be the vector parameter of size $nd$ consisting of the coordinates of the $n = N - m$ target nodes. $d$ is the space dimension.

The scalar ranging measurement $y_{u,v}$ between nodes $u$ and $v$ is assumed to be the realization of a random variable $\mathrm{y}_{u,v}$ of known distribution parametrized by the distance $d_{u,v}$ :

$$y_{u,v} \sim p_{\mathrm{y}_{u,v}|d_{u,v}} \tag{3.8}$$

where $d_{u,v}$ is the only unknown parameter in this distribution, for all $(u, v) \in E$, $u < v$ and $m < v$. We also assume that the different measurements are independent. Let $\mathbf{y}$ denote the vector of all the measurements. This vector is a realization of a random vector $Y$. The likelihood of $\theta$ can be written as

$$p(\mathbf{y}|\theta) = \prod_{\substack{(u,v)\in E \\ u<v,\ m<v}} p(y_{u,v}|d_{u,v}) \tag{3.9}$$

where the subscripts of the distribution functions are dropped for notational brevity.

A cooperative localization algorithm is an estimator $\hat{\theta}(\mathbf{y})$ of $\theta$, and the probabilistic estimators depend on the distribution functions (3.8).

Now, we will present the FIM and see how it defines a lower bound for unbiased estimators. This issue is well known in estimation theory. Then, we will derive the computation of the FIM for likelihood function (3.9). And next, we will establish correspondences between the rigidity and the FIM rank by proofing some propositions, and which will be used to derive other correspondences between the rigidity and the identifiability.

The FIM is a kind of measurement of the amount of information contained in the likelihood function about the unknown parameter. Under the regularity condition, i.e., $\log p(\mathbf{y}|\theta)$ is continuously differentiable w.r.t. $\theta$, the FIM is given by [71] :

$$\mathbf{J}(\theta) = E_{Y|\theta}\left\{\nabla_\theta \log\, p(\mathbf{y}|\theta)\,(\nabla_\theta \log\, p(\mathbf{y}|\theta))^T\right\} \tag{3.10}$$

where $E_{Y|\theta}\{.\}$ indicates the conditional expected value, and $\nabla_\theta$ is the gradient operator w.r.t. to $\theta$.

Provided that $\mathbf{J}(\theta)$ is non-singular, any unbiased estimator $\hat{\theta}$ satisfies

$$\mathrm{Cov}(\hat{\theta}) \succeq \mathbf{J}(\theta)^{-1} \tag{3.11}$$

where $\mathrm{Cov}(\hat{\theta}) = E_{Y|\theta}\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^T\}$ is the covariance matrix of $\hat{\theta}$ and $\mathbf{A} \succeq \mathbf{B}$ is equivalent to $\mathbf{A} - \mathbf{B}$ is positive semidefinite. Equation (3.11) defines the Cramér-Rao lower bound (CRB).

According to the definition of $\theta$, the location estimate of node $i+m$ corresponds to the entries of $\hat{\theta}$ going from $(i-1)d+1$ to $id$ :

$$\hat{\mathbf{x}}_{i+m} = \hat{\theta}_{(i-1)d+1:id} \tag{3.12}$$

where $m$ is the number of anchor nodes. The mean square error (MSE) of an unbiased estimate $\hat{\mathbf{x}}_{i+m}$ is lower bounded by the squared position error bound (SPEB) as follows :

$$E\left\{\|\hat{\mathbf{x}}_{i+m} - \mathbf{x}_{i+m}\|^2\right\} \geq \text{trace}\left\{[\mathbf{J}(\theta)^{-1}]_{(i-1)d+1:id}\right\} \tag{3.13}$$

where $[\mathbf{J}(\theta)^{-1}]_{(i-1)d+1:id}$ is the square submatrix of $\mathbf{J}(\theta)^{-1}$ with row and column entries going from $(i-1)d+1$ to $id$. The CRB in cooperative localization based on ranging measurements has been well studied in the literature [47, 72, 73], and has been considered as a benchmark for evaluating the performance of location estimators. But the conditions of obtaining a non-singular FIM have not been well studied. The results developed in this section aim at providing these conditions.

### 3.3.1 Computation of the FIM

Since the pair-wise measurements are independent, $\mathbf{J}(\theta)$ is obtained by summing the FIMs corresponding to the different observations and is derived in Appendix A :

$$\begin{aligned}\mathbf{J}(\theta) &= \sum_{\substack{(u,v)\in E \\ u<v,\ m<v}} \mathbf{J}_{u,v}(\theta) \\ &= \sum_{\substack{(u,v)\in E \\ u<v,\ m<v}} E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} \nabla_\theta d_{u,v} \left(\nabla_\theta d_{u,v}\right)^T .\end{aligned} \tag{3.14}$$

In the following sections, we assume that $E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} \neq 0$ for all $d_{u,v} > 0$. This condition is verified by ToA measurements and RSS measurements in additive Gaussian and additive mixture of Gaussian noises, as shown in Appendix A

### 3.3.2 Correspondence between the rigidity and the FIM

The novel results derived now describe the correspondence between the rigidity and the FIM. Let $(G, \mathbf{p})$ be the fundamental realization framework of the network $\mathcal{L}$, and let $M^{\ddagger}(G, \mathbf{p})$ be the matrix obtained form the rigidity matrix $M(G, \mathbf{p})$ by removing all the columns corresponding to the anchor nodes. The columns of $M^{\ddagger}(G, \mathbf{p})$, indexed by the target nodes, appear in the same order as in the vector parameter $\theta$.

**Proposition 3.1.** *The ranks of FIM* $\mathbf{J}(\theta)$ *and* $M^{\ddagger}(G, \mathbf{p})$ *are equal :*

$$rank\mathbf{J}(\theta) = rankM^{\ddagger}(G, \mathbf{p}). \tag{3.15}$$

*Proof* We have the following equality

$$\nabla_\theta d_{u,v} = \frac{1}{d_{u,v}} l_{u,v}^T \tag{3.16}$$

where $l_{u,v}$ is the row vector of $M^\ddagger(G,\mathbf{p})$ corresponding to edge $(u,v)$. This equality allows us to write (3.14) as

$$\mathbf{J}(\theta) = \sum E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} \frac{1}{d_{u,v}^2} l_{u,v}^T l_{u,v}. \tag{3.17}$$

Let $\mathbf{c}$ be a vector of the null space of $M^\ddagger(G,\mathbf{p})$, then it verifies

$$l_{u,v}\mathbf{c} = 0 \qquad \forall (u,v) \in E, u < v \text{ and } v > m, \tag{3.18}$$

and using (3.17), we can deduce that $\mathbf{J}(\theta)\mathbf{c} = \mathbf{0}$ and $\mathbf{c}$ belongs to the null space of $\mathbf{J}(\theta)$. On the other hand, if $\mathbf{c}$ is a vector in the null space of $\mathbf{J}(\theta)$, then $\mathbf{c}^T\mathbf{J}(\theta)\mathbf{c} = 0$ which can be expressed as

$$\sum E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} \frac{1}{\sigma_{u,v}^2 d_{u,v}^2} \mathbf{c}^T l_{u,v}^T l_{u,v}\mathbf{c} = 0, \tag{3.19}$$

and since $E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} > 0$ and $\mathbf{c}^T l_{u,v}^T l_{u,v}\mathbf{c} \geq 0$, therefore $\mathbf{c}^T l_{u,v}^T l_{u,v}\mathbf{c} = 0$ and $\mathbf{c}$ belongs to the null space of $M^\ddagger(G,\mathbf{p})$.

Thus, $M^\ddagger(G,\mathbf{p})$ and $\mathbf{J}(\theta)$ have the same null space and their ranks are equal. ■

Removing the columns corresponding to nodes in $M(G,\mathbf{p})$ is equivalent to fixing these nodes. The null space of the obtained matrix represents the infinitesimal motions of the framework that do not move the fixed nodes.

**Lemma 3.1.** *If $(G,\mathbf{p})$ is generic, then for any generic framework $(G,\mathbf{q})$ the following equality holds :*

$$rank M^\ddagger(G,\mathbf{p}) = rank M^\ddagger(G,\mathbf{q}). \tag{3.20}$$

*Proof* By fixing one anchor node, all the trivial translations are eliminated. For $d = 2$, there is one trivial rotation, and by fixing a second anchor node, all the trivial motions are removed. For $d = 3$, there are 3 independent trivial rotations, and by fixing a second anchor node, there remains only one rotation around the axis formed by the two fixed points, and fixing a third anchor node eliminates all the trivial motions. Thus, all the trivial motions are eliminated by fixing $d$ anchor nodes.

Since the subframework of $(G,\mathbf{p})$ corresponding to the anchor nodes is fully connected and infinitesimally rigid, none of the non-trivial infinitesimal motions is eliminated by fixing the anchor nodes.

By fixing the anchor nodes in $(G,\mathbf{p})$ and $(G,\mathbf{q})$, the same trivial motions are eliminated and $M^\ddagger(G,\mathbf{p})$ and $M^\ddagger(G,\mathbf{q})$ have the same null space rank. ■

**Lemma 3.2.** *If $M^\ddagger(G,\mathbf{p})$ is full column rank, then the framework $(G,\mathbf{p})$ is rigid.*

*Proof* The subframework of $(G, \mathbf{p})$ corresponding to the anchor nodes is rigid since it is fully connected. Thus, fixing the anchor nodes do not eliminate any non-trivial infinitesimal motion of the target nodes. ■ In other words, the non-singularity of the FIM is a sufficient condition for the rigidity of the fundamental realization framework.

**Lemma 3.3.** *If $M^{\ddagger}(G, \mathbf{p})$ is full column rank and the subframework of $(G, \mathbf{p})$ corresponding to the anchor nodes is generic, then the framework $(G, \mathbf{p})$ is infinitesimally rigid.*

*Proof* The subframework of $(G, \mathbf{p})$ corresponding to the anchor nodes is infinitesimally rigid since it is generic and fully connected. Thus, fixing the anchor nodes do not eliminate any non-trivial infinitesimal motion. ■

Now, assume that the $d$-dimensional network $\mathcal{L}$ has at least $d$ anchor nodes and the subframework of $(G, \mathbf{p})$ corresponding to the anchor nodes is generic.

**Theorem 3.7.** *The framework $(G, \mathbf{p})$ is infinitesimally rigid if and only if FIM $\mathbf{J}(\theta)$ is non-singular.*

*Proof* The sufficient condition follows from Proposition 3.1 and Lemma 3.3. The necessary condition follows from the fact that there are $d$ anchor nodes in general positions and all the trivial motions are eliminated when these nodes are fixed, and thus the null space of the $M^{\ddagger}(G, \mathbf{p})$ contains only the non-trivial infinitesimal motions. If the realization framework is infinitesimally rigid, this null space is empty and the FIM has a full rank. ■ When the FIM is singular, the subset of nodes belonging to an infinitesimally rigid subframework containing $d$ anchor nodes have a non-singular equivalent FIM.

Since rigidity is equivalent to infinitesimal rigidity for generic networks, we can deduce the following corollary :

**Corollary 3.3.** *A $d$-dimensional generic network with at least $d$ anchor nodes has a rigid graph if and only if the corresponding FIM is non-singular.*

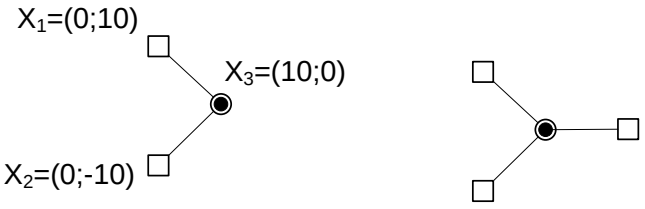
Thus, for generic networks, the singularity of the FIM is a graph property.

When the FIM is non-singular and the network graph is not globally rigid, estimators without ambiguities or consistent estimators may not exist, as we will show in the next section. In this case, the SPEBs (defined by (3.13)) of the nodes locations can be computed even though they are unachievable for some nodes due to position estimation biases caused by ambiguities, and the MSEs can be much higher than the SPEBs. The following example illustrates this issue.

The network in Figure 3.8(a), consisting of two anchor nodes and one target node, is rigid and has a non-singular FIM. The target node has a flipping ambiguity as it has two solutions verifying the distance constraints, and thus, an unbiased estimator does not exist. The distance measurements are affected by additive Gaussian errors of the same variance $\sigma^2$. In Figure 3.9(a), the SPEB is plotted for different variance values together with the MSE of two estimators : One that takes the maximum of the likelihood function (3.9) with positive abscissa and one that selects uniformly randomly one point of the two maxima of the

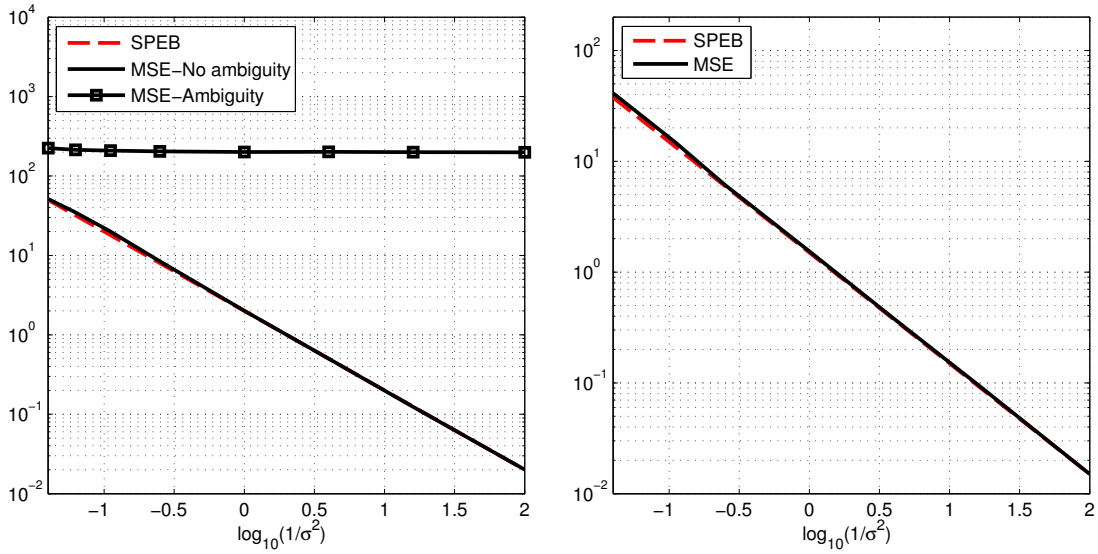
likelihood function. For the latter, we can notice that the MSE is much far from the SPEB even at low error variances. In Figure 3.8(b), a third anchor node is added and the network is globally rigid, and we can notice in Figure 3.9(b) that the MSE of the maximum likelihood (ML) estimator converges asymptotically to the CRB. The ML estimator proceeds by finding the position that maximizes the likelihood function :

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(\mathbf{y}|\theta). \tag{3.21}$$




(a) (b)

***Figure 3.8*** — (a) Rigid network. (b) Globally rigid network.




(a) (b)

***Figure 3.9*** — (a) SPEB and MSE corresponding to the network of Figure 3.8(a). (b) SPEB and MSE of the ML estimate corresponding to the network of Figure 3.8(b).

### 3.3.3 Identifiability theory

A theory of identification is developed in [74] for a general stochastic model whose probability distribution is determined by a finite number of parameters. It concerns the possibility of drawing inferences from the observations. In this section, we derive correspondences between the concepts of identifiability and rigidity.

Let $\mathbf{y}$ be the observation vector of size $k$. We state the following definitions provided in [74] :

- Two parameter points $\theta^0$ and $\theta^1$ are said to be observationally equivalent if $p(\mathbf{y}|\theta^0) = p(\mathbf{y}|\theta^1)$ for all $\mathbf{y} \in \mathbb{R}^k$.
- A parameter point $\theta^0$ is said to be globally identifiable if there is no other $\theta \in \mathbb{R}^{nd}$ which is observationally equivalent to it.
- A parameter point $\theta^0$ is said to be locally identifiable if there exists an open neighborhood of $\theta^0$ containing no other $\theta$ which is observationally equivalent to it.

**Local identifiability**

We derive the proofs of the following propositions that describe relationships between the local identifiability of the vector parameter and the rigidity of the fundamental realization framework.

**Proposition 3.2.** *For a d-dimensional network with at least d anchor nodes in general positions, the local identifiability of the target nodes positions implies the rigidity of the fundamental realization framework.*

*Proof* Let $\theta^0$ denote the vector of the target nodes positions. Assume that the fundamental realization framework is not rigid, then, due to continuous deformations, any neighborhood of $\theta^0$ contains a point $\theta^1$ which is observationally equivalent to it and $\theta^0$ is not locally identifiable. ■

The entries of FIM $\mathbf{J}(\theta)$ are continuous functions of $\theta$. A point $\theta^0$ is said to be a regular point of $\mathbf{J}(\theta)$ if there exists an open neighborhood of $\theta^0$ in which $\mathbf{J}(\theta)$ has a constant rank. The set of points $\theta$, such that the vectors $\mathbf{x}_i \in \mathbb{R}^d$, $i = 1, \cdots, N$ are generic, form an open dense set $S \subset \mathbb{R}^{nd}$ as stated in 3.2.3. Thus, any point $\theta \in S$ has an open neighborhood included in $S$. According to Lemma 3.1, the rank of $\mathbf{J}(\theta)$ is the same for all $\theta \in S$, and thus, all the points in $S$ are regular.

An important theorem in the identifiability theory is the following :

**Theorem 3.8** ([74])**.** *Let $\theta^0$ be a regular point of $\mathbf{J}(\theta)$. Then $\theta^0$ is locally identifiable if and only if $\mathbf{J}(\theta^0)$ is non-singular.*

**Proposition 3.3.** *For a d-dimensional network with at least d anchor nodes forming a generic subframework, if the vector of the target nodes positions is a regular point of the FIM, then the local identifiability of the target nodes positions is equivalent to the infinitesimal rigidity of the fundamental realization framework.*

*Proof* It follows from Theorem 3.8 and Theorem 3.7 ■

**Proposition 3.4.** *For a d-dimensional generic network with at least d anchor nodes, the local identifiability is equivalent to the graph rigidity.*

*Proof* It follows from Theorem 3.8 and Corollary 3.3 ■

**Global identifiability**

Indeed, the local identifiability is a necessary condition for global identifiability, but not sufficient, and so is the rigidity. The global rigidity is also a necessary condition of the global identifiability :

**Proposition 3.5.** *For a d-dimensional network with at least $d+1$ anchor nodes in general positions, the global identifiability of the target nodes positions implies the global rigidity of any realization framework.*

*Proof* Let $\theta^0$ denote the vector of the target nodes positions and let $(G, \mathbf{p})$ be the fundamental realization framework. Assume that $(G, \mathbf{p})$ is not globally rigid, then there exists a framework $(G, \mathbf{q})$ equivalent to $(G, \mathbf{p})$ and verifying $\mathbf{p}(i) = \mathbf{q}(i)$ for $i \leq m$ and $\mathbf{p}(i) \neq \mathbf{q}(i)$ for some $i > m$, $m$ being the number of anchor nodes. Let $\theta^1$ denote the vector of the mappings by $\mathbf{q}$ of the vertices corresponding to the target nodes. Then $\theta^0$ and $\theta^1$ are observationally equivalent and $\theta^0$ is not globally identifiable. ■

Now, we will show that when the distance values are globally identifiable from the ranging measurements, the global rigidity becomes a sufficient condition of the global identifiability. For this purpose, assume that the distance values $d_{u,v}$ are globally identifiable from the ranging measurements $y_{u,v}$ for all $(u, v) \in E$, $u < v$ and $m < v$, i.e., a distance value $d^0_{u,v}$ is not observationally equivalent to any other distance value $d^1_{u,v}$. The measurement models described in Appendix A verify this assumption.

**Theorem 3.9.** *For a d-dimensional network with at least $d+1$ anchor nodes in general positions, the global identifiability of the target nodes positions is equivalent to the global rigidity of any realization framework.*

*Proof* The proof of the necessary condition follows from Proposition 3.5. For the proof of the sufficient condition, let $\theta^0$ denote the vector of the target nodes positions, and assume that the network is globally rigid but $\theta^0$ is not globally identifiable. Let $\theta^1 \neq \theta^0$ such that $p(\mathbf{y}|\theta^0) = p(\mathbf{y}|\theta^1)$ for all $\mathbf{y} \in \mathbb{R}^k$. $\theta^1$ exists since $\theta^0$ is not globally identifiable. Let $D^0 = \{d^0_{u,v}\}$ and $D^1 = \{d^1_{u,v}\}$ be the assignments of distances corresponding to $\theta^0$ and $\theta^1$, respectively. We have $D^0 \neq D^1$ since the network is globally rigid and there are $d+1$ anchor nodes in general positions.

We can write

$$\prod_{\substack{(u,v)\in E \\ u<v,\ m<v}} p(y_{u,v}|d^0_{u,v}) = \prod_{\substack{(u,v)\in E \\ u<v,\ m<v}} p(y_{u,v}|d^1_{u,v}) \tag{3.22}$$

for all $\mathbf{y} \in \mathbb{R}^k$.

For a fixed couple $(i, j)$ such that $d^0_{i,j} \neq d^1_{i,j}$ and for a fixed set of values $\{y_{u,v}|(u, v) \neq (i, j)\}$, let

$$c_0 = \prod_{(u,v)\neq(i,j)} p(y_{u,v}|d^0_{u,v})$$

and

$$c_1 = \prod_{(u,v)\neq(i,j)} p(y_{u,v}|d^1_{u,v}).$$

Then we can write

$$c_0 p(y_{i,j}|d^0_{i,j}) = c_1 p(y_{i,j}|d^1_{i,j}) \qquad \forall y_{i,j} \in \mathbb{R} \tag{3.23}$$

and since $\int p(y_{i,j}|d^0_{i,j})\mathrm{d}y_{i,j} = \int p(y_{i,j}|d^1_{i,j})\mathrm{d}y_{i,j} = 1$, we obtain

$$p(y_{i,j}|d^0_{i,j}) = p(y_{i,j}|d^1_{i,j}) \qquad \forall y_{i,j} \in \mathbb{R} \tag{3.24}$$

resulting in a contradiction since $d^0_{i,j}$ is globally identifiable. ■ Thus, in order to check the global identifiability, it suffices to check the global rigidity. For non-generic networks, it is NP-hard to decide whether a realization framework is globally rigid. But for generic networks, global identifiability is a graph property that depends on the network connectivity and can be checked using the algorithm described in section 3.2.5.

**Remark** At high measurement errors, a portion of the network can be reflected or flipped, even when the network is globally rigid. In this case, the location estimation error can be very high. For example, consider the globally rigid network of Figure 3.10 consisting of three anchor nodes and one target node. Since the three anchor nodes are close to the dashed line (nearly collinear), the target node can be flipped at high error variance. In Figure 3.12(a), the SPEB and the MSE of the ML estimator are plotted, and in Figure 3.12(b) the flip probability (i.e., the estimated position is to the left of the dashed line) is plotted. We can notice that at low error variance, the flip probability tends to zero and the MSE of the ML estimator tends asymptotically to the SPEB, since the ML estimator is consistent [34]. The kind of flips illustrated by this example is not taken into account by the identifiability theory.

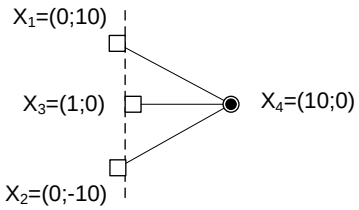



***Figure 3.10*** — A globally rigid network. The anchors are nearly collinear.

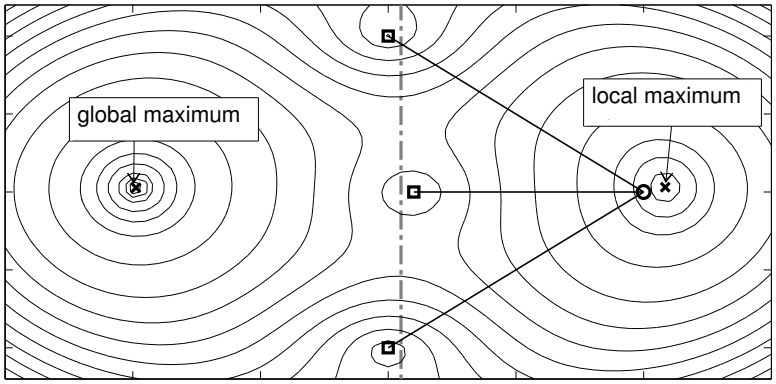



***Figure 3.11*** — Contour plot of the likelihood function for randomly generated measurements.

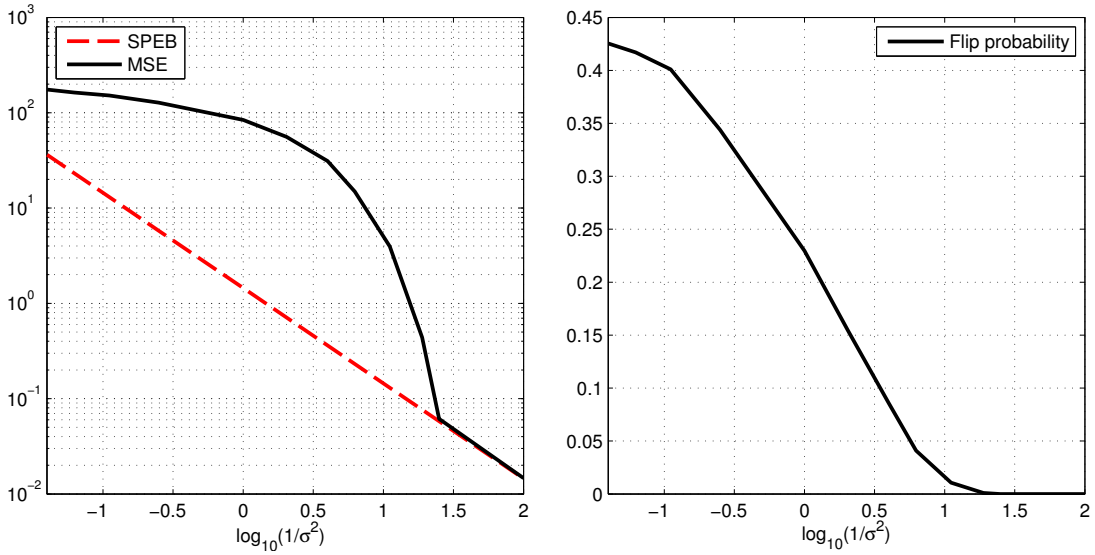



***Figure 3.12*** — (a) SPEB and MSE(in meters) of the ML estimate corresponding to the network of Figure 3.10. (b) Flip probability.

## 3.4 Localizability via semidefinite programming

An alternative approach for studying unique solvability is semidefinite programming. There has been a lot of theoretical work in using SDP for distance geometry [75]. In [76], it is shown that the SDP model can be used to identify only the portion of uniquely solvable nodes verifying properties that will be described later. In [2], we developed an iterative SDP-based solution that improves the identification of the uniquely solvable nodes.

The SDP method presented here is based on the relaxation model introduced in [77] with exact distance information. The tools provided by the SDP techniques are not restricted to generic networks as in the case of rigidity theory, and apply as well to non-generic networks which can occur in real application (e.g., a deployment where there are several nodes lying on the same line).

### 3.4.1 Definitions

A $d$-dimensional network $\mathcal{L}$ of size $N$ and its associated graph $G = (V, E)$ are defined as in Section 3.2.1. We use the following notations. $\mathbf{I}_d$ is the identity matrix of rank $d$. $\mathbf{0}$ is the column vector of zeros. $\mathbf{e}_i$ is the column vector with 1 at the $i^{th}$ entry and zeros elsewhere. $\mathbf{e}_{i,j} = \mathbf{e}_i - \mathbf{e}_j$. The dimensions of the vectors will be clarified by the context. $\langle \mathbf{v}, \mathbf{u} \rangle$ is the inner product of vectors $\mathbf{v}$ and $\mathbf{u}$. $(\mathbf{u}; \mathbf{v}) = [\mathbf{u}^T \ , \ \mathbf{v}^T]^T$. $\mathcal{L}_A = \{(k, j) \in E | k \leq m \text{ and } m < j\}$ and $\mathcal{L}_T = \{(i, j) \in E | m < i < j\}$ where $m$ is the number of anchor nodes.

For the anchor nodes, we let $\mathbf{a}_k = \mathbf{x}_k$, $k = 1, \cdots, m$.

The network localization problem (3.1) can be stated as follows :

$$\begin{array}{lll} \text{find} & \mathbf{x}_j,\ j = m+1,\ldots,N, & \\ \text{s.t.} & \|\mathbf{a}_k - \mathbf{x}_j\|^2 = d_{k,j}^2 & \forall (k,j) \in \mathcal{L}_A \\ & \|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{i,j}^2 & \forall (i,j) \in \mathcal{L}_T. \end{array} \tag{3.25}$$

Let $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_{m+1},\ldots,\bar{\mathbf{x}}_N] \in \mathbb{R}^{h\times n}$, $h \geq d$ and $n = N - m$. We admit that $\bar{\mathbf{X}}$ is an $h$-dimensional solution of the network localization problem if it verifies the following equations :

$$\begin{array}{ll} \|(\mathbf{a}_k;\mathbf{0}) - \bar{\mathbf{x}}_j\|^2 = d_{k,j}^2 & \forall (k,j) \in \mathcal{L}_A \\ \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|^2 = d_{i,j}^2 & \forall (i,j) \in \mathcal{L}_T, \end{array} \tag{3.26}$$

where $(\mathbf{a}_k;\mathbf{0}) = [\mathbf{a}_k^T\ \mathbf{0}^T]^T$ is an $h$-dimensional vector.

Now, we define the term of unique localizability and restate the term of unique solvability :

- A target node $\mathbf{x}_{j+m}$ is uniquely solvable if for every pair of $d$-dimensional solutions $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}'$, the equation $\left(\bar{\mathbf{X}} - \bar{\mathbf{X}}'\right)\mathbf{e}_j = \mathbf{0}$ is verified, or in other words, the node has a unique solution verifying the constraints in (3.25).
- A target node $\mathbf{x}_{j+m}$ is uniquely localizable if for every $h > d$ and for every pair of $h$-dimensional solutions $\bar{\mathbf{X}}$ and $\bar{\mathbf{X}}'$, the equation $\left(\bar{\mathbf{X}} - \bar{\mathbf{X}}'\right)\mathbf{e}_j = \mathbf{0}$ is verified.
- The network is uniquely solvable if all its target nodes are uniquely solvable.
- The network is uniquely localizable if all its target nodes are uniquely localizable.

Unique localizability implies unique solvability since the network has at least one $d$-dimensional solution $\bar{\mathbf{X}}$ and $[(\bar{\mathbf{x}}_{m+1};\mathbf{0}),\ldots,(\bar{\mathbf{x}}_N;\mathbf{0})]$ is an $h$-dimensional solution.

### 3.4.2 SDP method and localizability test

Problem (3.25) can be written in matrix form as follows [76] :

$$\begin{array}{lll} \text{find} & \mathbf{X} \in \mathbb{R}^{d\times n}, \mathbf{Y} \in \mathbb{R}^{n\times n}, & \\ \text{s.t.} & \mathbf{e}_{i,j}^T \mathbf{Y} \mathbf{e}_{i,j} = d_{i+m,j+m}^2 & \forall (i+m,j+m) \in \mathcal{L}_T \\ & (\mathbf{a}_k;-\mathbf{e}_j)^T \mathbf{Z} (\mathbf{a}_k;-\mathbf{e}_j) = d_{k,j+m}^2 & \forall (k,j+m) \in \mathcal{L}_A \\ & \mathbf{Y} = \mathbf{X}^T\mathbf{X}, & \end{array} \tag{3.27}$$

where $\mathbf{X} = [\mathbf{x}_{m+1},\ldots,\mathbf{x}_N]$ and $\mathbf{Z}$ is defined as follows :

$$\mathbf{Z} = \begin{bmatrix} \mathbf{I}_d & \mathbf{X} \\ \mathbf{X}^T & \mathbf{Y} \end{bmatrix}. \tag{3.28}$$

Problem (3.27) is a nonconvex optimization problem. It can be transformed into a semidefinite program (and thus convex) by relaxing the equality constraint $\mathbf{Y} = \mathbf{X}^T\mathbf{X}$ into a semidefinite condition $\mathbf{Y} - \mathbf{X}^T\mathbf{X} \succeq 0$, which is equivalent to $\mathbf{Z} \succeq 0$ [77], i.e., $\mathbf{Z}$ is positive semidefinite. Then, we arrive at the following SDP problem [76] :

$$\begin{array}{lll} \text{minimize}_{\mathbf{Z}\in\mathbb{R}^{(n+d)\times(n+d)}} & 0, & \\ \text{s.t.} & \mathbf{Z}_{1:d,1:d} = \mathbf{I}_d & \\ & (\mathbf{0};\mathbf{e}_{i,j})^T \mathbf{Z} (\mathbf{0};\mathbf{e}_{i,j}) = d_{i+m,j+m}^2 & \forall (i+m,j+m) \in \mathcal{L}_T \\ & (\mathbf{a}_k;-\mathbf{e}_j)^T \mathbf{Z} (\mathbf{a}_k;-\mathbf{e}_j) = d_{k,j+m}^2 & \forall (k,j+m) \in \mathcal{L}_A \\ & \mathbf{Z} \succeq 0. & \end{array} \tag{3.29}$$

Problem (3.29) is equivalent to finding a matrix $\mathbf{Z}$ that satisfies the constraints (also called constraint satisfaction problem).

#### Localizability Test

Let $\bar{\mathbf{Z}}$ be a solution of problem (3.29), then $d \leq \text{rank}\bar{\mathbf{Z}} \leq d+n$. A max-rank solution is a solution that has the highest rank among all feasible ones. Such a solution can be computed by means of an interior-point algorithm. The complexity of this algorithm is discussed in [75] and is typically bounded by $O(N^3 + N^2K + K^3)$ where $K$ is the number of constraints bounded by $O(N^2)$.

**Theorem 3.10** ([76])**.** *Let $\bar{\mathbf{Z}}$ be a max-rank solution of problem (3.29), then the following statements are equivalent :*

- *The network is uniquely localizable.*
- *The rank of $\bar{\mathbf{Z}}$ is equal to d.*
- *$\bar{\mathbf{Z}}$, represented as (3.28), satisfies $\bar{\mathbf{Y}} = \bar{\mathbf{X}}^T\bar{\mathbf{X}}$.*

Thus, by finding a max-rank solution of problem (3.29), we can answer the question whether the network is uniquely localizable. We can also test the unique localizability of the different nodes according to the following important properties [76] :

- Node $\mathbf{x}_{j+m}$ is uniquely localizable if and only if $\bar{\mathbf{Y}}_{j,j} - \|\bar{\mathbf{x}}_{j+m}\|^2 = 0$, where $\bar{\mathbf{Y}}_{j,j}$ is the $j^{th}$ diagonal entry of $\bar{\mathbf{Y}}$ and $\bar{\mathbf{x}}_{j+m}$ is the $j^{th}$ column of $\bar{\mathbf{X}}$.
- Node $\mathbf{x}_{j+m}$ is not uniquely localizable if and only if $\bar{\mathbf{Y}}_{j,j} - \|\bar{\mathbf{x}}_{j+m}\|^2 > 0$.

We shall call this test the simple SDP-based test (SSDP) in the sequel. The uniquely localizable nodes constitute only a subset of the uniquely solvable nodes, and the SSDP test cannot detect all these latter nodes (e.g., Figure 3.14(a)).

### 3.4.3 Improving the unique solvability test

Now, we develop a solution for improving the identification of the uniquely solvable nodes. We shall call it the iterative SDP-based algorithm (ISDP). Our focus will be on 2-dimensional networks. The algorithm can be extended to the 3-dimensional case straightforwardly.

We begin by showing how to reduce the rank of the SDP solution for a network of 3 nodes by a simple modification of the objective function of the SDP problem (3.29).

#### Rank Reduction via Scalar Product

We consider the network of Figure 3.13 consisting of two anchor nodes of position vectors $\mathbf{a}_1$ and $\mathbf{a}_2$, and one target node of position vector $\mathbf{x}_3$.

By solving the SDP problem (3.29) corresponding to this network, we obtain the following matrix $\bar{\mathbf{Z}}$ :

$$\bar{\mathbf{Z}} = \begin{bmatrix} \mathbf{I}_2 & \bar{\mathbf{x}}_3 \\ \bar{\mathbf{x}}_3^T & \bar{\mathbf{Y}} \end{bmatrix}. \tag{3.30}$$

The rank of $\bar{\mathbf{Z}}$ is equal to 3 since the target node is not uniquely localizable (it can have a 3-dimensional solution by a rotation around the axis formed by the anchor nodes), and $\bar{\mathbf{Y}} > \|\bar{\mathbf{x}}_3\|^2$. This result can be seen as if the SDP solver finds a 3-dimensional vector $\left(\bar{\mathbf{x}}_3; \pm\sqrt{\bar{\mathbf{Y}} - \|\bar{\mathbf{x}}_3\|^2}\right)$ and the computed location $\bar{\mathbf{x}}_3$ is the orthogonal projection of this vector on the plane of the network.

Let $\mathbf{a}_{1,2} = \mathbf{a}_2 - \mathbf{a}_1$. The constraint of problem (3.29)

$$(\mathbf{a}_k; -1)^T \mathbf{Z} (\mathbf{a}_k; -1) = d_{k,3}^2 \quad k = 1, 2 \tag{3.31}$$

is verified by $\bar{\mathbf{Z}}$, which implies that the scalar product $\langle \mathbf{a}_{1,2}, \bar{\mathbf{x}}_3 \rangle = \langle \mathbf{a}_{1,2}, \mathbf{x}_3 \rangle = $ constant, or in other words, the computed location lies on the line perpendicular to the segment joining $\mathbf{a}_1$ and $\mathbf{a}_2$ and passing through the true position $\mathbf{x}_3$.

Let $\mathbf{a}_{1,2}^{\perp}$ be a vector perpendicular to the vector $\mathbf{a}_{1,2}$. The equality $\bar{\mathbf{Y}} = \|\bar{\mathbf{x}}_3\|^2$ is verified when the inner product $\left\langle \mathbf{a}_{1,2}^{\perp}, \bar{\mathbf{x}}_3 - \mathbf{a}_1 \right\rangle$ is maximized or minimized. Thus, we can obtain a solution in the plane of the network and reduce the rank of $\bar{\mathbf{Z}}$ to 2 by modifying the objective function of (3.29) to one of the following two linear functions that keep the problem a semidefinite program [2] :

$$\text{minimize} \quad \left\langle \mathbf{a}_{1,2}^{\perp}, \mathbf{x}_3 \right\rangle \tag{3.32}$$

or

$$\text{minimize} \quad -\left\langle \mathbf{a}_{1,2}^{\perp}, \mathbf{x}_3 \right\rangle. \tag{3.33}$$

Node 3 is not uniquely solvable. The two solutions obtained for each of the objective functions (3.32) and (3.33) are plotted in Figure 3.13. These solutions, denoted by $\bar{\mathbf{x}}_3$ and $\bar{\mathbf{x}}_3'$, verify $\bar{\mathbf{Y}} - \|\bar{\mathbf{x}}_3\|^2 = 0$ and $\bar{\mathbf{Y}}' - \|\bar{\mathbf{x}}_3'\|^2 = 0$.

### Iterative SDP-Based Algorithm

In the first step of the ISDP algorithm, the SDP problem (3.29) is solved and the uniquely localizable target nodes identified by the SSDP test are promoted to anchor nodes. Then, for a target node $j$ connected to two anchor nodes $k$ and $l$, the following steps are processed :

– The two SDP problems with the following two objective functions are solved :

$$\text{minimize} \quad \pm\left\langle \mathbf{a}_{k,l}^{\perp}, \mathbf{X}\mathbf{e}_j \right\rangle. \tag{3.34}$$

The obtained solutions are denoted by $\bar{\mathbf{Z}}$ and $\bar{\mathbf{Z}}'$.

– If $\bar{\mathbf{Y}}_{j,j} - \|\bar{\mathbf{x}}_{j+m}\|^2 = 0$ and $\bar{\mathbf{Y}}'_{j,j} - \left\|\bar{\mathbf{x}}'_{j+m}\right\|^2 > 0$ (or vice versa), then node $j+m$ and all the nodes $i+m$ verifying $\bar{\mathbf{Y}}_{i,i} - \|\bar{\mathbf{x}}_{i+m}\|^2 = 0$ are uniquely solvable. These nodes are promoted to anchor nodes.

2. A general semidefinite program can be defined as follows :

$$\begin{array}{cc} \text{minimize}_{\mathbf{X}\in\mathbb{R}^q} & \text{trace}\mathbf{C}^T\mathbf{X} \\ \text{s.t.} & \text{trace}\mathbf{A}_i^T\mathbf{X} = b_i \quad i = 1, \cdots, L \\ & \mathbf{X} \succeq 0. \end{array}$$

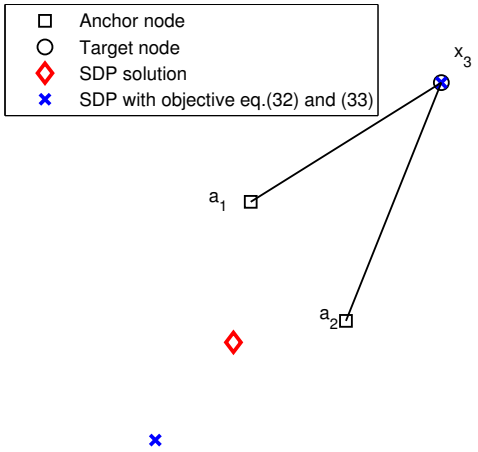



***Figure 3.13*** — A network of one target node connected to two anchor nodes.

– Otherwise the node is skipped but it can be revisited later.

The algorithm performs iteratively by testing all the targets connected to two anchors until no more targets can be promoted. The different steps are summarized in Table 3.2.

| | |
|---|---|
| 0 : | Function ISDP |
| 1 : | **apply** the SSDP algorithm |
| 2 : | **promote** uniquely localizable targets to anchors |
| 3 : | **find** the set $T$ of targets connected to two anchors |
| 4 : | **for** each target $\in T$ |
| 5 : | **solve** the 2 SDP problems with objective functions (3.34) |
| 6 : | **if** the selected target is uniquely solvable |
| 7 : | **promote** the identified targets to anchors |
| 8 : | **go to** 3 |

***Table 3.2*** — Iterative SDP-based algorithm (ISDP)

This test can be extended to the 3-dimensional case by selecting a node connected to three anchor nodes in general position at each iteration.

### 3.4.4 Examples

Here, we provide two examples to show the efficiency of the ISDP in situations where the global rigidity test and the SSDP fail in identifying uniquely solvable nodes. The considered networks are 2-dimensional. We also provide two examples that show that the ISDP does not identify all the uniquely solvable nodes. In fact, as mentioned previously, sufficient and necessary conditions for unique nodes solvability are not yet known.

In Figure 3.14(a), the two target nodes 4 and 5 are non-uniquely localizable although the network is globally rigid : Node 5 is not aligned with nodes 2 and 3 and the network can have

a 3-dimensional solution. The red diamonds represent the positions obtained by solving the SDP problem (3.29). By applying the ISDP, we can show that they are uniquely solvable. The blue crosses represent the positions computed by the ISDP algorithm.

In Figure 3.14(b), node 4 is non-uniquely localizable and does not belong to a globally rigid subgraph, while it is uniquely solvable and can be identified and correctly localized by the ISDP algorithm. Nodes 5 and 6 are not uniquely solvable.

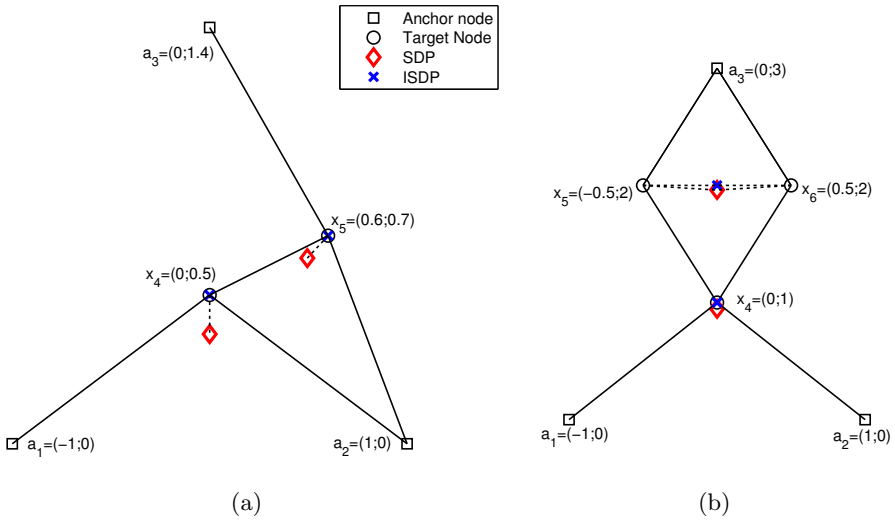



***Figure 3.14*** — (a) A globally rigid network which is not uniquely localizable. (b) Node 4 is uniquely solvable but does not belong to a globally rigid subgraph and is not uniquely localizable.

In Figure 3.15(a), node 6 is uniquely solvable while it is not uniquely localizable and does not belong to a globally rigid subgraph. The ISDP also fails in identifying this node as there is a 3-dimensional solution for which node 6 lies in the plane of the network. The two blue crosses represent the two solutions for node 6 obtained by solving the SDP problems with objective equations (3.32) and (3.33) These two solutions, denoted by $\bar{\mathbf{x}}_6$ and $\bar{\mathbf{x}}'_6$, verify $\bar{\mathbf{Y}}_{3,3} - \|\bar{\mathbf{x}}_6\|^2 = 0$ and $\bar{\mathbf{Y}}'_{3,3} - \|\bar{\mathbf{x}}'_6\|^2 = 0$.

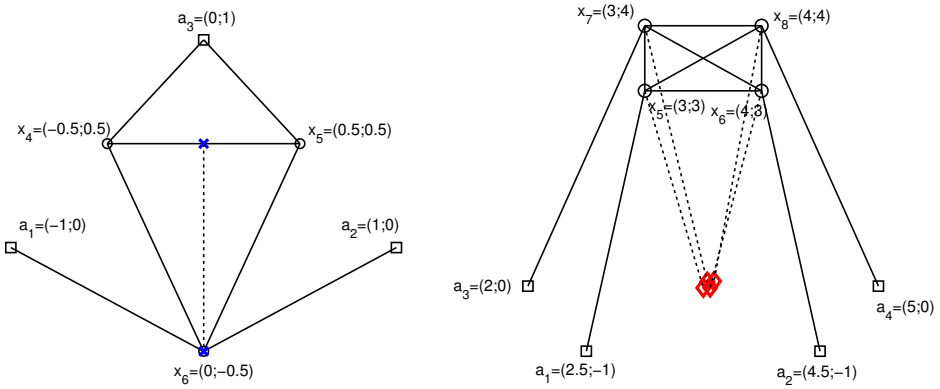
In Figure 3.15(b), the target nodes are uniquely solvable and the network is globally rigid. None of the target node is uniquely localizable and the ISDP does not apply in this case since there is no target node connected to two anchors.

### 3.4.5 Correspondence between rigidity and localizability

Universal rigidity of frameworks is a rigidity theoretic counterpart of unique localizability of networks. A $d$-dimensional framework $(G, \mathbf{p})$ is said to be universally rigid if for any $h$-dimensional framework $(G, \mathbf{q})$ we have :

$$\begin{aligned} &\|\mathbf{p}(i) - \mathbf{p}(j)\| = \|\mathbf{q}(i) - \mathbf{q}(j)\| \quad \forall\ (i,j) \in E \\ \Rightarrow \quad &\|\mathbf{p}(i) - \mathbf{p}(j)\| = \|\mathbf{q}(i) - \mathbf{q}(j)\| \quad \forall\ i,j \in V. \end{aligned} \tag{3.35}$$

In other words, $(G, \mathbf{p})$ is the unique realization, up to congruence, of a network, in any Euclidean space. It is shown in [78] that for a $d$-dimensional network having at least $d+1$




(a) (b)

***Figure 3.15*** — (a) Node 6 is uniquely solvable but does not belong to a globally rigid subgraph and is not uniquely localizable. (b) A globally rigid network but not uniquely localizable.

anchor nodes in general positions, unique localizability is equivalent to universal rigidity of any realization framework. Thus, unique localizability implies global rigidity and universally rigid frameworks are a subset of the globally rigid ones.

SDP provides an efficient method for checking unique localizability. Moreover, given the connectivity graph and the distance information of a uniquely localizable network, then the unique realization can also be found using SDP. For globally rigid frameworks that are not universally rigid, there does not exist an efficient algorithm for finding a realization.

Finally, we mention that universal rigidity is not a generic property for all frameworks : The network of Figure 3.14(a) is not uniquely localizable, but it becomes so by setting $\mathbf{x}_5 = (0.3; 0.7)$.

## 3.5 Numerical results

Simulations are performed on 2-dimensional networks in order to study the statistical occurrences of uniquely localizable and solvable nodes and rigid and globally rigid networks under different deployment scenarios.

The following simple connectivity model based on received signal power is considered. Two nodes $i$ and $j$ are connected if the power $P_{i,j}$ (in decibel ($dB$)) at node $i$ transmitted by node $j$ (or vice versa) is above a threshold value $P_{th}$. A widely accepted model for the received signal power is

$$P_{i,j} = P_0 - 10n_p \log_{10}(d_{i,j}) + X_{i,j}, \tag{3.36}$$

where $P_0$ is the average received power at a distance of 1 meter, $n_p$ is the path loss exponent and $X_{i,j}$ is a centered Gaussian random variable of variance $\sigma^2_{sh}$ representing the shadowing loss.

The probability that two nodes are connected is a function of their separating distance.

By assuming that $P_{i,j} = P_{j,i}$ for every pair of nodes $(i, j)$ (i.e., by setting $P_0$ to be the same for all transmitting nodes and assuming a reciprocal channel with $X_{i,j} = X_{j,i}$), this probability is equal to

$$\begin{aligned} p_c(d) &= \text{probability}(P_0 - 10n_p\log_{10}(d) + X > P_{th}) \\ &= \text{probability}(X > 10n_p\log_{10}(d/R)) \\ &= Q\left(\frac{10n_p\log_{10}(d/R)}{\sigma_{sh}}\right), \end{aligned} \tag{3.37}$$

where $Q$ is the $Q$-function, $R = 10^{\frac{P_0 - P_{th}}{10n_p}} d_0$ and $p_c(R) = Q(0) = 1/2$.

We take $n_p = 3$ and $\sigma_{sh} = 8dB$, and assume that the shadowing losses are independent for any pair of different links (i.e., the two random variables $X_{i,j}$ and $X_{k,l}$ are independent if $(i, j) \neq (k, l)$).

In Figure 3.16, the average number of neighbors of a node is plotted against the total number of nodes $N$. The nodes are uniformly drawn in a square region of size $L \times L$ ($L$ in meters). It is evident that the average number of neighbors should be increasing with the nodes density and the range $R$. This figure can be useful to give an idea on the average number of neighbors since selected values of $N$ and $R$ are used in the following simulations.

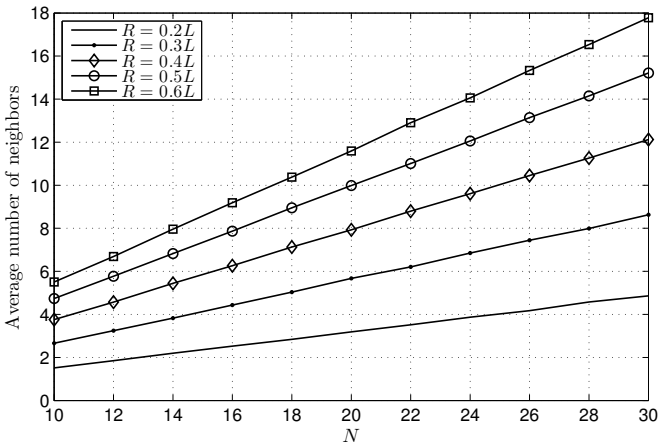



***Figure 3.16*** — Variation of the average number of neighbors of a node with the total number of nodes $N$ for several values of $R$.

### 3.5.1 SSDP vs. ISDP vs. global rigidity

Here, we set $m = 3$ anchors and $R = 0.35L$, and the nodes are uniformly drawn. In Figure 3.17, we plot the probability distribution of the number of uniquely localizable nodes identified using the SSDP and the probability distribution of the number of uniquely solvable nodes identified using the ISDP and global rigidity test described in Table 3.1.

We can remark that there are more uniquely solvable nodes than uniquely localizable ones : The uniquely localizable nodes are a subset of the uniquely solvable ones.

The ISDP identifies more uniquely solvable nodes than the SSDP but less than the global rigidity test. The combination of the two methods yields the best identification performance.

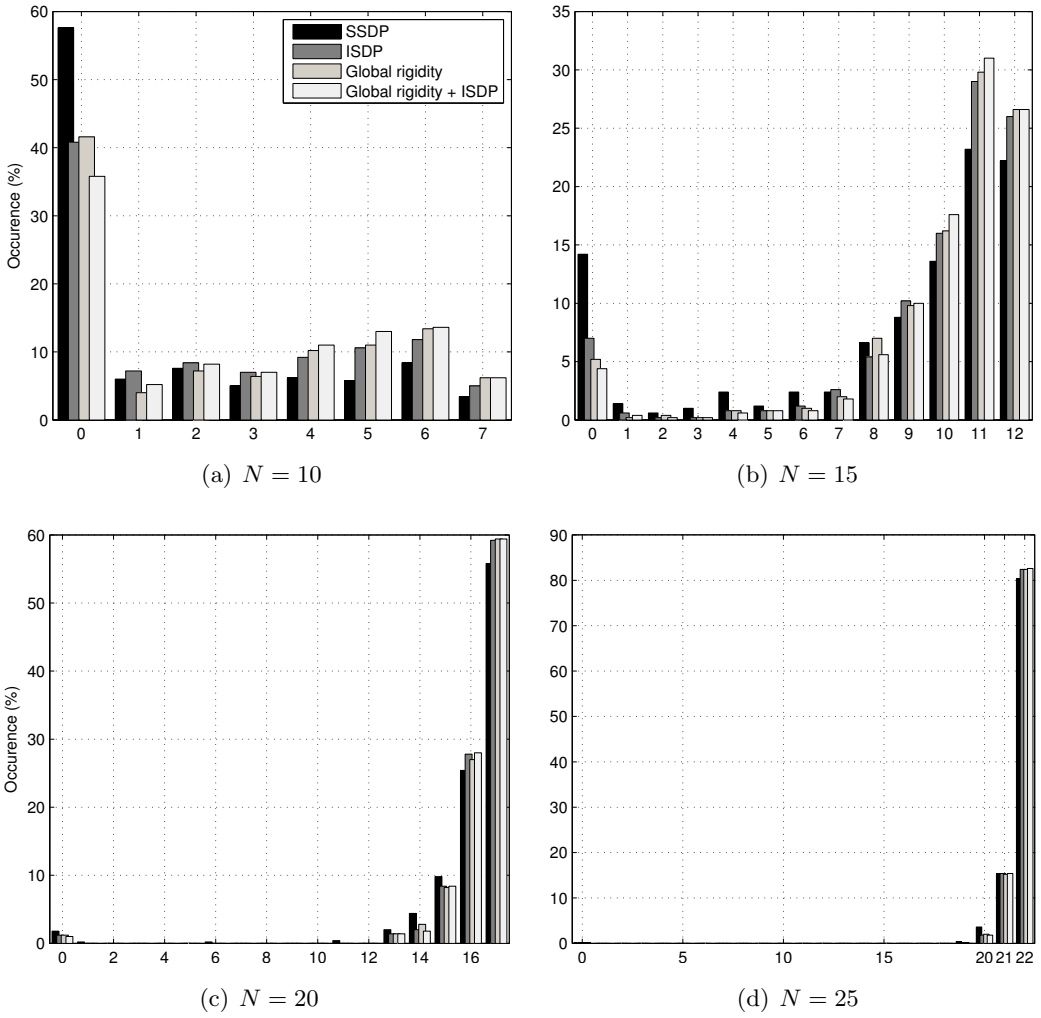



***Figure 3.17*** — Probability distribution of the number of uniquely solvable and localizable target nodes for different network sizes $N$. The number of anchor nodes is 3.

### 3.5.2 SDP accuracy

The accuracy of the locations computed by the SDP for non-uniquely localizable nodes is investigated in terms of mean location error $E\{\|\hat{\mathbf{x}} - \mathbf{x}\|\}$.

The simulation scenario is similar to that of the previous section except that the considered networks are connected (i.e., there is no isolated nodes and a path does exist between every two nodes). We set $L = 10$ meters and $R = 3.5$ meters. The nodes that are correctly localized by the ISDP but not by the SDP are called 'uncommon nodes', these nodes are uniquely solvable but not uniquely localizable. And the nodes that are not correctly localized by both methods are called 'common nodes'.

Figure 3.18 depicts the mean location error against the network size $N$. We can see that the mean location error of uncommon nodes is smaller than that of common nodes. The uncommon nodes are uniquely solvable and are, in general, connected to more nodes than the common nodes. As their error is small, we can conclude that the SDP method provides

a good starting point for descent optimization solutions.

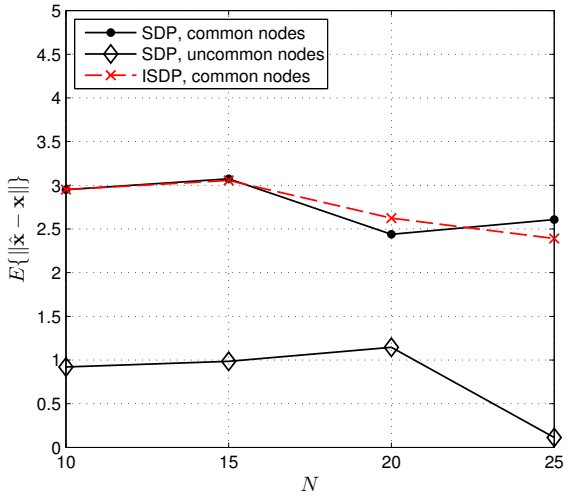
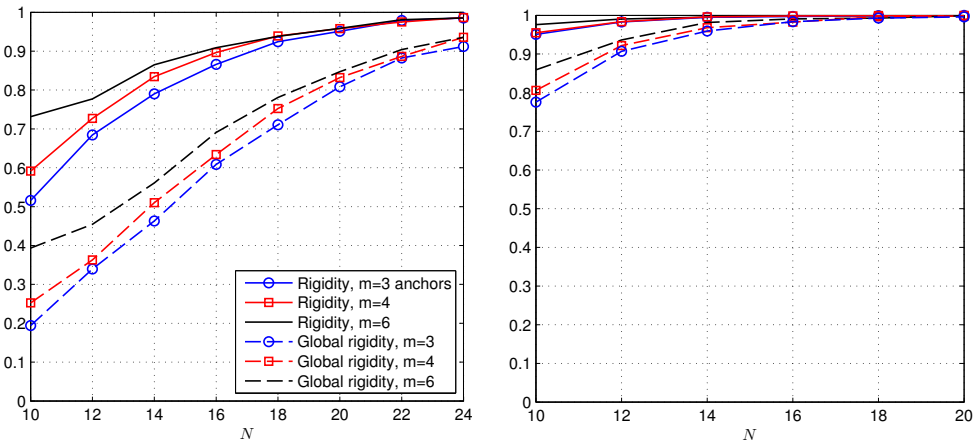



***Figure 3.18*** — Mean location error (in meters) vs. network sizes $N$.

### 3.5.3 Effect of number and placement of anchor nodes and range $R$

In Figure 3.19, we plot the probability of network rigidity and global rigidity for different numbers of anchor nodes $m$ and ranges $R$ where there are $N$ nodes with positions uniformly drawn. We can notice that these probabilities increases with $m$ and $R$, and they are above 0.9 for $R = 0.6L$ and $N \geq 12$.



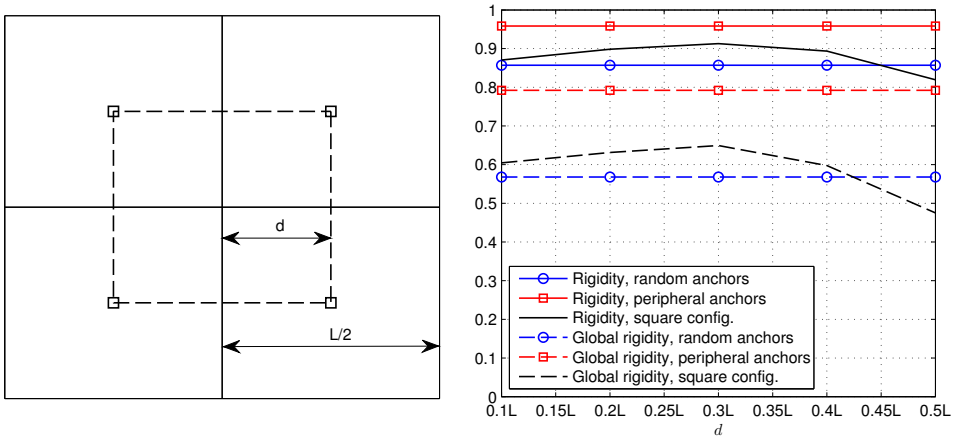

(a) $R = 0.4L$ (b) $R = 0.6L$

***Figure 3.19*** — Probability of network rigidity and global rigidity vs. network size $N$ for different values of $m$.

To asses the effect of the anchor nodes placement, we consider the network configuration shown in Figure 3.20(a), where the target nodes are uniformly placed inside the square of size $L \times L$ and the anchor nodes are the corners of a square of edge length $2d$. We set $N = 15$

and $R = 0.4L$. Figure 3.20(b) shows the network rigidity and global rigidity probabilities for different values of $d$. Two other configuration are considered : one with randomly placed anchor nodes and one in which the $N$ nodes of the network are randomly generated then the peripheral ones (i.e., the nearest to the corners of the $L \times L$ deployment area) are chosen as anchors.

We see that the probabilities of rigidity and global rigidity increase with $d$ then begin to decrease at around $d = 0.3L$. In fact, at high values of $d$, the anchor nodes become far from the target nodes and thus less connected to them. This justifies the decreasing behavior of the plots. We can also notice that the probabilities are the highest when the anchors nodes are the peripheral ones, but this configuration is not feasible since the nodes positions are not known a priori in order to set the peripheral nodes as anchors.




(a) (b)

***Figure 3.20*** — (a) Typical network deployment with 4 anchors placed on the corners of a square. (b) Probability of network rigidity and global rigidity vs. the value of $d$ and for other anchors configurations.

### 3.5.4 Effect of link shadowing correlation

The independent shadowing model is a simplification model that does not accurately represent the radio channel for multi-hop networks : Two nearly overlapping links would have independent shadowing. The shadowing is determined by the environmental obstruction and the shadowing losses on different links are not independent. The correlations have impact on the connectivity and the rigidity of networks.

We consider the statistical propagation model presented in [79] which accounts for the shadowing correlation in multi-hop networks. According to this model, the link's shadowing $X_{i,j}$ for the pair of nodes $(i, j)$ is a weighted integral of a spatial loss field $p(\mathbf{x})$ :

$$X_{i,j} = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^{1/2}} \int_{\mathbf{x}_i}^{\mathbf{x}_j} p(\mathbf{x}) d\mathbf{x}. \tag{3.38}$$

The considered spatial loss field $p(\mathbf{x})$ is an isotropic Gaussian random field of zero mean and

exponentially decaying correlation function :

$$E\left\{p(\mathbf{x}_i)p(\mathbf{x}_j)\right\} = \frac{1}{\sigma_X^2}\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\delta}\right). \tag{3.39}$$

A simulation is performed in order to see the effect of correlations on rigidity. We set $m = 3$ anchors, $L = 10m$ and $R = 4m$, and for shadowing $\sigma_X = \sigma_{sh} = 8dB$ and $\delta$ takes one of the two values $0.2m$ and $0.6m$. In Figure 3.21, we plot the probability of rigidity against the total number of nodes $N$. The nodes positions are uniformly selected among a finite set of positions for which the shadowing covariance matrix for the different pairs of links have been computed. We can notice that the probability that the network is rigid diminishes when the shadowing is correlated. This result indicates that independent link shadowing model over-estimates the probability of rigidity.

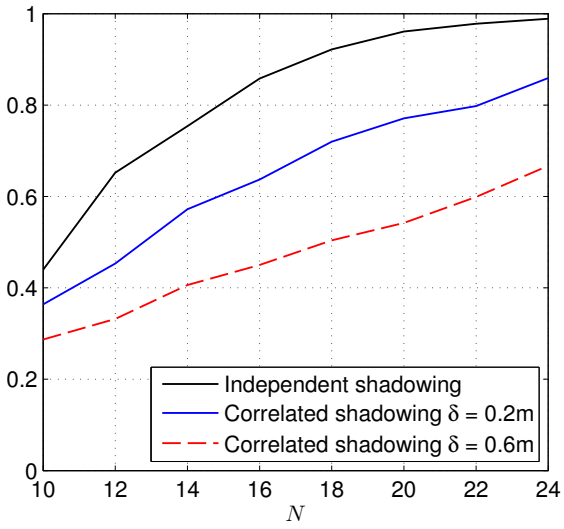



***Figure 3.21*** — Probability of network rigidity vs. network sizes $N$.

## 3.6 Conclusion

In this chapter, the unique solvability in cooperative localization was addressed by considering the two approaches of graph rigidity theory and semidefinite programming. They define sufficient unique solvability conditions. We provided a survey of the results of the graph rigidity theory. This theory allows checking unique solvability in generic networks from the connectivity perspective and regardless of the distance values. For the SDP approach, we described the state of the art algorithm and developed a new algorithm that improves the detection of the uniquely solvable nodes. SDP based methods require the true distance values to be known. These values might not be available due to the presence of noise in the measurements. Nevertheless, the SDP based methods are useful in studying the statistical occurrences of uniquely solvable nodes.

The following correspondence between the FIM and the rigidity matrix was derived : The rank of the FIM is equal to the rank of the rigidity matrix of the fundamental realization

framework after discarding the columns corresponding to the anchor nodes. Based on this correspondence, we showed that for a $d$-dimensional generic network having at least $d$ anchor nodes, the non-singularity of the FIM is equivalent to the graph rigidity. This result is important for several points : Firstly, the non-singularity of the FIM can be checked form the network connectivity. In the case of 2-dimensional networks, algorithms with complexity lower than matrix rank computation can be applied (e.g., the pebble game algorithm which also determines the rigid subgraphs, and consequently, the portions of the network that have a non-singular FIM). Secondly, the non-singularity of the FIM is a necessary but not sufficient condition for the global rigidity, and does not guarantee the absence of ambiguity. We also proved that for generic networks, local identifiability is equivalent to graph rigidity.

It is known from estimation theory that consistent estimators can exist only when the parameter is globally identifiable. In cooperative localization based on ranging measurements, consistent estimators can exist only when there are $d + 1$ anchor nodes in general positions and realization frameworks are globally rigid. The consistency of the maximum likelihood estimator in presence of additive Gaussian noises is studied in the next chapter. For generic networks, we showed that global identifiability is a graph property that only depends on the number of anchor nodes and the network connectivity, and thus, it can be checked without the need to know the positions of the target nodes.

Occurrence probabilities of rigid and globally rigid networks and uniquely solvable nodes were computed via Monte Carlo simulations. A simple connectivity model based on received power under log-normal shadowing was considered. Other site specific connectivity models (e.g., partition losses or ray tracing, consideration of medium access) can be used in order to establish the required number of anchor nodes and their placement and other related parameters (e.g., transmission power) that guarantee a high probability of unique solvability in a deployment area.

The next chapter is devoted to the task of computing the positions of the target nodes from the set of pair-wise ranging measurements.

CHAPTER 4

# Cooperative Localization Algorithms

## 4.1 Introduction

After having studied the identifiability conditions under ranging measurements in the previous chapter, we devote this chapter to the task of estimating the positions.

The number of anchor nodes in a wireless network is typically small and a target node may be several hops away from the anchor nodes. Thus, nodes need to cooperate with each other in order to converge to a consistent assignment of coordinates.

In this chapter, we are interested in probabilistic estimation, where the positions are considered as random variables and the localization problem is formulated as an inference in probabilistic graphical models. This formalism enables capturing the dependencies among the random variables and applying the belief propagation (BP) algorithm, where each node computes a probability density function of its coordinates, in a distributed manner, based on its local a priori information, and on the measurements and probability densities provided at each iteration by neighboring nodes. This algorithm produces both estimates of positions and metrics of uncertainties. Additionally, some flip ambiguities can be eliminated using the BP algorithm by assuming a probabilistic connectivity model and by exchanging information between non-neighboring nodes.

The BP is implemented using its particle-based version which numerically approximates the messages that cannot be computed in a closed-form. This implementation is known under the name of nonparametric belief propagation (NBP). The main contribution of this chapter is the development of a new variant of the NBP method that performs two phases :

- In the first phase, the classical NBP is implemented by exchanging messages between direct neighbors in order to compute the belief of each node regarding its position.
- In the second phase, each node constructs a small set of space points by exploiting its computed belief. Then, the analytical BP is applied to estimate the positions which are assumed to be discrete random variables taking on values in the constructed sets.

This solution improves the localization accuracy and allows reducing the amount of exchanged data.

In addition to the probabilistic estimation, we study the weighted least-squares (WLS)

estimators, and derive their deterministic stability conditions based on the unique solvability results provided in the previous chapter.

This chapter is organized as follows. We start with an overview of different existing cooperative localization algorithms in Section 4.2. In Section 4.3, we present the WLS and probabilistic estimators, and we derive the stability conditions of the WLS and consistency-conditions of the maximum likelihood (ML) estimator. The implementation of probabilistic estimators in undirected graphical models is reviewed in Section 4.4. This lays the foundation for the NBP method which is the subject of Section 4.5. Other message passing algorithms are reviewed in Section 4.6. Then, the two-phases NBP solution is developed in Section 4.7, and its performances are validated in Section 4.8 via Monte-Carlo simulations. Concluding remarks are drawn in Section 4.9.

## 4.2 Overview of cooperative algorithms

Cooperative localization has been an active research topic in the last ten years, and many algorithms have been developed. These algorithms can be classified according to several criteria such as the kind of information used, the way the computation is processed, the need of a special infrastructure or anchor nodes, etc. Four main classes are described in this section.

### 4.2.1 Centralized vs. distributed

In centralized algorithms, measurements are sent to a central processor where the overall processing is done. In distributed algorithms, all the nodes or a portion of them are involved in the computation process.

The distributed algorithms do not require the existence of a central processor with a high computational capacity to which all measurements are forwarded. This fact makes them more attractive for large networks. They are also more suitable for tracking the mobile nodes over time as they do not require forwarding the measurements to a central processor.

On the other hand, centralized algorithms can be more accurate and allow to exploit the correlations in the measurements (e.g., correlation of the shadowing in RSS measurements [80] or sensor data measurements [81]). They are also unavoidable when the nodes do not have sufficient computational capacity.

Examples of centralized algorithms are the maximum likelihood estimator [82, 47], semi-definite programming [77], WLS [83] and multidimensional scaling (MDS) [84].

The distributed algorithms fall into one of the following four categories :

#### Successive refinement

These algorithms implement a distributed optimization solution in order to find the optimum of a global cost function. They perform iteratively, where at each iteration, one or several nodes update their estimates by minimizing local cost functions and send them to their

neighbors, until a convergence criterion is met. We refer to the neighborhood of a node as the group of nodes in its vicinity, with which measurements can be obtained. Examples of these algorithms are the distributed gradient descent [38], distributed weighted MDS (dwMDS) [85], distributed extended Kalman filter (distribute EKF) [86] and simulated annealing [87]. An initialization phase that provides starting search positions is required for these algorithms. The selection of these positions is critical and impacts the final convergence solution as will be shown in Section 4.3.

#### Message passing

These algorithms perform inference in probabilistic graphical models (Markov random fields or factor graphs), where every node computes its posterior marginal distribution from its local a priori information and messages received from neighboring nodes. Probabilistic models for the measurements need to be assumed in order to define the potential functions between every pair of connected nodes in the probabilistic graph. Examples of these algorithms are nonparametric belief propagation [55, 88, 3], variational message passing [89] and expectation propagation [90]. These algorithms will constitute the core of this chapter, where their theoretical framework and implementation issues will be discussed in the following sections.

#### Multilateration

When a target node does not have a sufficient number of anchor neighbors to compute directly its position, it estimates its distances to several nearby anchor nodes based on the number of hops and range measurements along their shortest connecting path. These distance estimates are then used by the node to compute its position via a multilateration solution. Several methods for estimating the distances and the positions have been developed [91, 92, 93]. A quantitative comparison of their accuracy is made in [94, 95]. Multilateration algorithms have several advantages over other distributed algorithms : a) low communication cost as they are not iterative ; b) privacy/security as the nodes do not communicate their positions ; and c) nodes can independently choose when to perform self-localization. However, these algorithms are inaccurate when the number of anchor nodes is small or when the network has a non-isotropic topology and the target nodes are not inside the convex hull of the anchor nodes.

#### Incremental algorithms

Incremental algorithms perform iteratively, where additional nodes are localized at each iteration. In [96, 52, 53], the computed positions are promoted to anchors in the subsequent iterations. A flipped position estimate, which corresponds to an erroneous geometrical realization, can degrade the remaining position estimates and propagate in an avalanche fashion. In order to avoid this situation, a test is applied in [52, 53] for identifying the nodes with high flip probabilities and remove them from the localization procedure. While this approach

can guarantee a high robustness of the solution, it can discard an important portion of the nodes.

In [97, 98, 99], every node builds a local map of the immediate vicinity, and then maps are merged together to form a global map. These algorithms can be seen as hybrid algorithms which combine centralized and distributed features.

The incremental algorithms are simple to implement and their complexity is linear with the number of nodes, but errors can propagate with iterations (due to hard decisions). Unlike multilateration algorithms, incremental algorithms are suitable for networks with non-convex and non-isotropic topologies.

### 4.2.2 Range based vs. range free

Range free algorithms allow to estimate the positions from mere connectivity. They are attractive when the nodes are not able to perform accurate ranging measurements.

The multilateration distributed algorithms have been applied as range free by associating approximate distance values to the links, which are computed by dividing the distance between two anchor nodes by the number of hops in their shortest connecting path [91].

MDS [84] and similarity based algorithms, such as the Laplacian eigenmap [100] which reduces the sum of the distances between the connected nodes are also range free.

In [101], two connected nodes are considered to be within a certain range from each other, and the cooperative localization problem is modeled as a constraint satisfaction problem with convex semidefinite constraints. This method is not very accurate when the anchor nodes are not placed on the outer boundary of the network since the feasibility areas of the target nodes become very large.

### 4.2.3 Anchor based vs. anchor free

The outputs of anchor free algorithms are relative positions, or in other words, an embedding of the nodes in a space of given dimension. On the other hand, when a sufficient number of anchor nodes are available, the computed positions are absolute and can be related to a geographical map.

The MDS-based algorithms [84, 98] provide relative positions. The incremental algorithms can be also applied as anchor free. In [102], an anchor free algorithm is proposed, which performs in two phases : first a range free algorithm is applied to compute a fold free coordinate assignment, then a successive refinement solution is applied using the measured values.

Relative maps may be all that is obtainable in situations where anchor nodes are not available (e.g., costly or impossible manual calibration, unavailability of GPS).

Once the absolute positions of some nodes become available, one can compute the absolute positions of the remaining nodes by applying a translation, a rotation or a mirroring of the network [82].

### 4.2.4 Probabilistic vs. non-probabilistic

Probabilistic algorithms exploit the available probabilistic models on the measurements (i.e., likelihood functions) and positions (i.e., a priori information) by computing distributions conditioning on the observations

These algorithms can be more accurate than their non-probabilistic counterparts (e.g., WLS) as they can exploit the a priori information and can mitigate the NLoS effect whenever a probabilistic model of NLoS propagation is available [38]. They are also more resilient to the presence of outliers, where an outlier is an observation that lies outside the overall pattern of a distribution.

Cooperative algorithms can be compared based on several criteria such as accuracy, complexity (e.g., the number of computational operation), latency (e.g., the time needed to deliver the position estimates or the number of iterations in iterative algorithms), etc. A trade off does exist between the different criteria and the importance of each of them depends on the considered application.

## 4.3 WLS and probabilistic estimation

In this section, the WLS estimators and probabilistic estimators are formulated, and the deterministic stability conditions of the WLS and consistency conditions of the ML are derived.

The WLS estimators have been widely considered in non-cooperative and cooperative localization since their formulation is simple and does not require specific assumptions on the noise model. A WLS estimator will be used as a benchmark against which the probabilistic message passing algorithms will be compared.

The formulation of the probabilistic estimators will lay the foundations for the development of message passing algorithms.

We begin by stating the system model and providing notations that will be used throughout this chapter.

### 4.3.1 Definitions

We consider a network $\mathcal{L}$ of size $N$ consisting of $m < N$ anchor nodes of known positions labeled 1 through $m$ and $n = N - m$ target nodes labeled $m + 1$ through $N$ and whose positions are yet to be determined. A graph $G = (V, E)$ is associated to the network where $V$ is the vertex set representing the different nodes and $E$ is the edge set representing the neighborhood relationship. The network lies in a $d$-dimensional space where node $i$ is located at $\mathbf{x}_i \in \mathbb{R}^d$. The nodes are assumed to be static.

Let $\theta = [\mathbf{x}_{m+1}^T, \cdots, \mathbf{x}_N^T]^T$ be the vector of the unknown positions. The pair-wise measurement between nodes $i$ and $j$ is denoted by $\mathbf{y}_{i,j}$ and it can be either a scalar value (e.g., RSS, ToA) or multiple values (e.g., channel impulse response). When the distance is estimated

from $\mathbf{y}_{i,j}$, we denote it by $\tilde{d}_{i,j} = f(\mathbf{y}_{i,j})$. The true distance value is $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$. Let $\mathbf{y}$ denote the vector of all measurements.

### 4.3.2 Weighted least-squares

A WLS estimator seeks to minimize a cost function of the form :

$$\hat{\theta}_{WLS} = \underset{\theta \in \mathbb{R}^{dn}}{\arg\min} \sum_{\substack{(i,j)\in E \\ i<j,\ m<j}} w_{i,j}(\tilde{d}_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 + \sum_{i>m} r_i \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2, \tag{4.1}$$

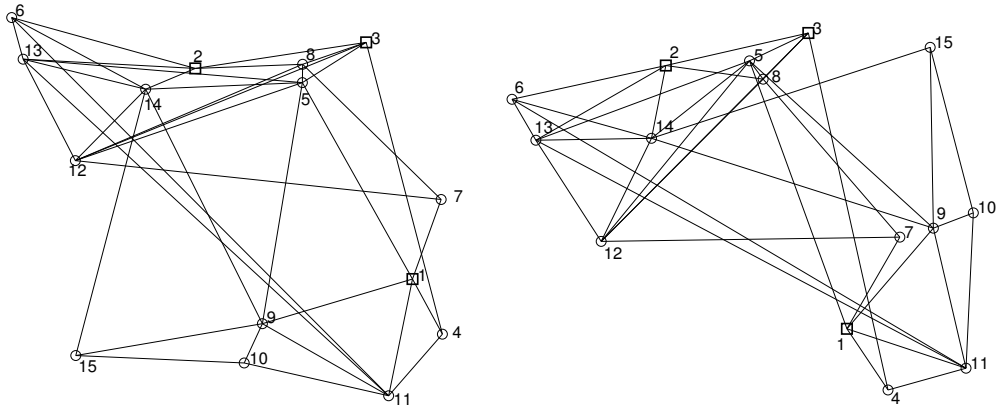
where $w_{i,j}$ is a positive weight reflecting the accuracy of $\tilde{d}_{i,j}$, and the parameters $\bar{\mathbf{x}}_i$ and $r_i$ encode the a priori knowledge that $\mathbf{x}_i$ is believed to lie around $\bar{\mathbf{x}}_i$ with accuracy $r_i$. In the absence of information for target node $i$, $r_i$ is set equal to zero. We will denote the cost function of (4.1) by $s(\theta)$.

For the selection of weights, several strategies have been adopted. In [77], the weights are set equal to the inverse of estimated variances of the measurements (i.e., $w_{i,j} = 1/\hat{\sigma}^2_{i,j}$ where $\hat{\sigma}^2_{i,j} \propto \tilde{d}^2_{i,j}$), while in [85], the weights are exponentially decreasing with the measured distances. Weighing have been used in [83] to mitigate the effect of bias in the presence of unknown LoS/NLoS conditions where ToA measurements are involved.

Problem (4.1) is a non-convex optimization problem. It can be seen as a mass spring optimization problem, where we imagine each edge in the graph as a spring between two masses. If the distance between two nodes is different from the observed value, the spring incurs a force that pushes them apart or pulls them together. The target nodes will move due to the forces until an equilibrium is reached where the sum of the forces on each node is equal to zero. The final potential energy at equilibrium depends on the initial positions of the different nodes and corresponds either to a global minimum or to a local one. Local minima are frequent in problems of this kind and can result in wrong configurations of the network and high errors in the estimated positions. For example, in Figure 4.1(a), where there are three anchor nodes (of indices 1, 2 and 3) and the target nodes are uniquely solvable, the distance measurements are taken equal to the true distance values. The network of Figure 4.1(b) shows the estimated positions corresponding to a local minimum reached by starting at random initial positions. When the positions are not uniquely solvable, there might be multiple configurations of the network that have the same global minimal energy.

A minimum of $s(\theta)$ can be computed using an iterative numerical solution starting at initial positions, such as the gradient descent, Levenberg-Marquardt algorithm and the stress majorization (SMACOF) [85]. It can be noticed that the gradient of the cost function (4.1) with respect to $\mathbf{x}_i$ (i.e., $\nabla_{\mathbf{x}_i} s$) depends only on the coordinates of the neighboring nodes. Thus, this partial gradient can be computed locally at a given iteration, provided that the coordinates of the neighboring nodes at the previous iteration are known. This has allowed the development of distributed iterative optimization solutions (that we called successive refinement in the previous section). The dwMDS [85] is such a solution applying a distributed version of the SMACOF algorithm that guarantees a non-increase of the cost function $s(\theta)$ at each iteration.

Initial positions need to be chosen carefully in order to reduce the probability of falling into a local minimum. Examples of algorithms for finding initial positions are the SDP, the MDS and the distributed incremental and mulitlateration algorithms described in the previous section.




(a) (b)

***Figure 4.1*** — (a) Uniquely solvable 2-dimensional network. (b) Estimated positions corresponding to a local minimum of the WLS cost function.

## WLS deterministic stability

Here, we examine the deterministic stability of the WLS problem (4.1) based on the definitions and results provided in [103]. For a nonlinear estimation problem expressed as a minimization problem of some cost function, the deterministic stability implies the uniqueness of the global minimum provided the observation noise is small enough.

We start by providing some definitions and a theorem applicable to a general nonlinear least-squares (NLS) estimation problem, as they are stated in [103], and then we show that the WLS problem (4.1) is a special case of this problem.

Let $y$ be an observation vector of the form

$$y = g(x) + e \tag{4.2}$$

where $y \in \mathcal{Y} \subset \mathbb{R}^k$, $x \in \mathcal{X} \subset \mathbb{R}^l$ and $e$ is a noise random vector. The term 'deterministic' is used in the expression 'deterministic stability' since no assumption is made on the probability distribution of $e$.

The NLS problem is defined as follows :

$$\hat{x}_{NLS} = \arg\min_{x \in \mathcal{X}} \|y - g(x)\|^2. \tag{4.3}$$

Let $d_{x_0}(x, e)$ be the following cost function :

$$d_{x_0}(x, e) = \|y - g(x)\|^2 = \|g(x_0) - g(x) + e\|^2. \tag{4.4}$$

This function is another formulation of the NLS that explicitly depends on $x$, on the noise $e$, and on the true value of the parameter $x_0$. Based on this formulation, we state the definition of the deterministic stability along with other definitions :

- Unique solvability : The unknown $x_0 \in \mathcal{X}$ is the unique solution of $d_{x_0}(x, \mathbf{0}) = 0$ if

$$\{x \in \mathcal{X} : d_{x_0}(x, \mathbf{0}) = \mathbf{0}\} = \{x_0\}.$$

  In other words, $x_0$ is the only parameter that matches the data in the noiseless case.
- Strict global minimizer : A vector $x' \in \mathcal{X}$ is a strict global minimizer of $d_{x_0}(x, e)$ with respect to $x$ for fixed $x_0$ and $e$ if

$$d_{x_0}(x', e) < d_{x_0}(x, e) \quad \forall x \in \mathcal{X}, \quad x \neq x', \tag{4.5}$$

  i.e., $x' = \arg\min_{x \in \mathcal{X}} d_{x_0}(x, e)$.
  In the next definition, $B_\delta(t)$ denotes the standard open ball of radius $\delta$ and center $t$ in a metric space $\mathcal{T}$ (i.e., $B_\delta(t) = \{t' \in \mathcal{T} : \|t - t'\| < \delta\}$), and $\mathcal{X}^o$ denotes the set of interior points of $\mathcal{X}$ (i.e., $\mathcal{X}^o = \{x \in \mathcal{X} : \exists \delta > 0 : B_\delta(x) \subset \mathcal{X}\}$).
- Deterministic stability : A NLS problem is stable at $x_0 \in \mathcal{X}^o$ if $\exists \epsilon > 0$ and a continuously differentiable function $\phi : B_\epsilon(\mathbf{0}) \longmapsto \mathcal{X}$ such that $\phi(e)$ is a strict global minimizer of $d_{x_0}(x, e)$ with respect to $x$ for all $e \in B_\epsilon(\mathbf{0})$.

According to this definition, deterministic stability at $x_0$ implies that $x_0$ is the unique solution of $d_{x_0}(x, \mathbf{0}) = 0$, since $\phi$ is a strict global minimizer and

$$\phi(\mathbf{0}) = \arg\min_{x \in \mathcal{X}} d_{x_0}(x, \mathbf{0}) = x_0.$$

A main result of [103] on the deterministic stability is the following theorem :

**Theorem 4.1.** *Let $g : \mathcal{X} \longmapsto \mathcal{Y}$ be a continuous map, where $\mathcal{X} \subset \mathbb{R}^l$, $\mathcal{Y} \subset \mathbb{R}^k$ and $l \leq k$. Suppose, furthermore, that $g$ is twice continuously differentiable in an open neighborhood of $x_0 \in \mathcal{X}^o$, and $x_0$ is the unique solution of $d_{x_0}(x, \mathbf{0}) = 0$. Then the following are equivalent :*

1. *Gradient matrix of $g$, defined by $[\mathbf{G}(x_0)]_{i,j} = (\partial g_i(x))/(\partial x_j)|_{x=x_0}$ with $1 \leq i \leq k$ and $1 \leq j \leq l$, is full column rank.*
2. *NLS problem is stable in the deterministic sense at $x_0$.*

Now, we return to our WLS problem (4.1) and re-examine it with respect to this theorem. For simplification, we assume that $r_i = 0$ for all $i > m$.

Let $(G, \mathbf{p})$ be the fundamental realization framework of $\mathcal{L}$ as defined in Section 3.2.1. We assume that $\mathbf{p}(i) \neq \mathbf{p}(j)$ for all $(i, j) \in E$.

We define $x$ as

$$x = \theta = [\mathbf{p}^T(m+1), \cdots, \mathbf{p}^T(N)]^T, \tag{4.6}$$

and we can write $s(\theta)$ as

$$\begin{aligned} s(\theta) &= \sum_{\substack{(i,j) \in E \\ i<j,\ m<j}} (y_{i,j} - \sqrt{w_{i,j}} \|\mathbf{p}(i) - \mathbf{p}(j)\|)^2 \\ &= \|y - g(x)\|^2 \end{aligned} \tag{4.7}$$

where $y_{i,j} = \sqrt{w_{i,j}}\tilde{d}_{i,j}$, and $y$ is the vector of all $y_{i,j}$ values and $g(x)$ is defined accordingly. Thus, the WLS problem (4.1) is a special case of the NLS problem (4.3), and we can apply Theorem 4.1 to it.

A parameter $x_0$ is the unique solution of $d_{x_0}(x, \mathbf{0})$ if the network $\mathcal{L}$ is uniquely solvable. According to Theorem 3.1, this property is verified if the framework $(G, \mathbf{p})$ is globally rigid and there are at least $d+1$ anchor nodes in general positions when the network is deployed in a $d$-dimensional space.

Notice that the gradient matrix $\mathbf{G}(x_0)$ has the same rank as the matrix $M^{\ddagger}(G, \mathbf{p})$ obtained from the rigidity matrix by removing the columns corresponding to the anchor nodes as defined in Section 3.3.2. This property follows from the equality

$$\mathbf{G}(x_0) = DM^{\ddagger}(G, \mathbf{p})$$

where $D$ is a full rank diagonal matrix.

If the network $\mathcal{L}$ is generic and uniquely solvable, then, according to Corollary 3.3, $\mathbf{G}(x_0)$ is full column rank and the WLS problem is stable in the deterministic sense at $x_0$.

The theory we presented in this section allows to make claims on the behavior of the WLS estimator with respect to small noise perturbations. A result that can be concluded from the continuity of the map $\phi$ is that the error in the estimated positions is bounded.

### 4.3.3 Probabilistic estimators

The observation $\mathbf{y}$ is a realization of a random variable $\mathbf{Y}$ of joint probability distribution function $p_{Y|\theta}(\mathbf{y}|\theta)$, which will be written as $p(\mathbf{y}|\theta)$ for notational brevity. For a given $\mathbf{y}$ and variable $\theta$, $p(\mathbf{y}|\theta)$ is called the likelihood function of $\theta$.

In order to apply a probabilistic estimator, probabilistic models for the observations need to be assumed. The probabilistic estimators can be classified as non-Bayesian or Bayesian, depending on whether $\theta$ is considered as an unknown deterministic parameter or a realization of a random variable $\Theta$ of known a priori distribution $p_{\Theta}(\theta)$, which will be denoted as $p(\theta)$.

#### Non-Bayesian context

$\theta$ is treated as an unknown deterministic parameter. A common non-Bayesian estimator is the maximum likelihood (ML). It operates by maximizing the likelihood function :

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(\mathbf{y}|\theta). \tag{4.8}$$

It is known from estimation theory [34] that the ML estimator is consistent, i.e., converges asymptotically to the CRB at low error variances, provided that the problem is globally identifiable as we showed in the previous chapter.

Assume that the distance observations $\{\tilde{d}_{i,j}\}$, obtained according to $\tilde{d}_{i,j} = f(\mathbf{y}_{i,j})$, form a sufficient statistic for the estimation of $\theta$ (e.g., under ToA measurements). Then, the ML

estimator becomes :

$$\hat{\theta}_{ML} = \arg\max_{\theta} p(\{\tilde{d}_{i,j}\}|\theta). \tag{4.9}$$

When the distance values are affected by independent additive Gaussian noises, i.e., $\tilde{d}_{i,j} = d_{i,j}+e_{i,j}$ and $e_{i,j}$ is an independent Gaussian distributed variable with zero mean and variance $\sigma^2_{i,j}$, then the optimization problem (4.9) reduces to the WLS estimator (4.1) with $w_{i,j} = 1/\sigma^2_{i,j}$ and $r_i = 0$. This result is obtained by maximizing the log-likelihood function :

$$\hat{\theta}_{ML} = \arg\max_{\theta} \log p(\{\tilde{d}_{i,j}\}|\theta). \tag{4.10}$$

**Consistency of the ML estimator**

Here, we assume that the pair-wise measurements are scalar ranging measurements (i.e., RSS or timing measurements) affected by independent additive Gaussian noises. We also assume that the means of the noises are equal to zero. Under these assumption, the ML estimator is equivalent to the NLS problem defined by (4.3).

Furthermore, we assume that $k$ independent realizations of the observation vector $\mathbf{Y}$ can be obtained. We denote by $\mathbf{y}^i$ the $i^{th}$ realization. We also denote by $\hat{\theta}_k$ the ML estimate that accounts for all the realizations :

$$\hat{\theta}_k = \arg\max_{\theta} \prod_{l=1}^{k} p(\mathbf{y}^l|\theta). \tag{4.11}$$

The consistency of the ML estimator means that the true parameter value $\theta_0$ can be found with arbitrary precision if we allow the number of realizations $k$ to go to infinity. In mathematical terms, this means that $\hat{\theta}_k$ converges in probability to $\theta_0$ as $k$ goes to infinity : For every $\epsilon > 0$, Probability$(\|\hat{\theta}_k - \theta_0\| < \epsilon) \to 1$ as $k \to \infty$. This statement will be written as $\hat{\theta}_k \xrightarrow{P} \theta_0$.

Now, we will show that $\hat{\theta}_k$ is consistent when the network is generic and uniquely solvable. First of all, according to our previous results, $\hat{\theta}_k$ is deterministically stable.

It can be shown that the $k$ pair-wise measurements between nodes $i$ and $j$ can be replaced by the sufficient statistics

$$\bar{y}^k_{i,j} = \frac{1}{k}\sum_{l=1}^{k} y^k_{i,j} = g_{i,j}(d_{i,j}) + \bar{e}^k_{i,j}. \tag{4.12}$$

$\bar{e}^k_{i,j}$ is the average of noises in the independent measurements. Let $\bar{e}^k$ be the vector of all noises averages. Thus, according to the weak law of large numbers, it can be shown that $\|\bar{e}^k\| \xrightarrow{P} 0$. Now we state the following theorem.

**Theorem 4.2** ([104]). *If $c_n$ is a sequence of random variables such that $c_n \xrightarrow{P} c$, and if $f$ is a function which is continuous at $c$, then $f(c_n) \xrightarrow{P} f(c)$.*

For large values of $k$, $\bar{e}^k$ will be, with high probability, inside the ball over which the continuous strict global minimizer $\phi$ is defined (see the definition of deterministic stability 4.3.2). Thus, according to theorem 4.2, $\hat{\theta}_k = \phi(\bar{e}^k) \xrightarrow{P} \phi(0) = \theta_0$.

**Bayesian context**

$\theta$ is assumed to be a random variable drawn from a known a priori distribution. Two common Bayesian estimators are the minimum mean square error (MMSE) and the maximum a posteriori (MAP) estimators.

The a posteriori distribution of $\theta$ is

$$p(\theta|\mathbf{y}) \propto p(\theta)p(\mathbf{y}|\theta). \tag{4.13}$$

– The MMSE estimator finds the mean of (4.13) :

$$\hat{\theta}_{MMSE} = \int \theta p(\theta|\mathbf{y}) \mathrm{d}\theta. \tag{4.14}$$

– The joint MAP estimator finds the maximum of (4.13) :

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta|\mathbf{y}). \tag{4.15}$$

Probabilistic inference applies to all kinds of measurements provided that a probabilistic model is available. When the assumed models deviate from the true distributions, the estimation accuracy may degrade.

Assume that the a priori distributions of the different nodes are independent and Gaussian (i.e., $p(\theta) = \prod_{i>m} p_i(\mathbf{x}_i)$ where $p_i$ is a Gaussian distribution function of mean $\bar{\mathbf{x}}_i$ and covariance matrix $\mathbf{C}_i$), and that the distance observations form a sufficient statistic and are affected by independent additive Gaussian noises, as we assumed in the non-Bayesian context. Then the joint MAP estimator (4.15) reduces to the WLS estimator (4.1) with a replacement of $r_i\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2$ by $(\mathbf{x}_i - \bar{\mathbf{x}}_i)^T\mathbf{C}_i^{-1}(\mathbf{x}_i - \bar{\mathbf{x}}_i)$. Thus, the WLS can be seen as a special case of the probabilistic estimators, and it can take advantage of the tools available in probabilistic inference such as the marginalization and its associated message passing techniques, which will be presented in the following sections.

## 4.4 Inference in graphical models

Graphical models provide an efficient and powerful framework for modeling probabilistic relationships. Many algorithms for computing basic statistical quantities have been expressed in terms of recursions operating on these graphs (e.g., Kalman filters, Bayesian smoothing and predication, and belief propagation). The main advantage of this formalism is that some formerly exponential computations complexity become linear when using the graphs.

Graphical models have been applied in many fields including bioinformatics, speech processing, image processing, control theory and error correcting codes [105, 106, 107].

There are two classes of graphical models :

– Bayesian networks which are directed graphical models and are useful for expressing causal relationships between random variables.

– Markov random fields (MRFs) which are undirected graphical models and are better suited for expressing soft constraints between random variables.

MRFs enable specifying the factorization of the conditional independence properties of joint distributions of the form of equation (4.13), and thus they are suitable for performing inference in cooperative localization problems. In this section, we review the definition of the MRFs and some of their basic associated marginalization and inference techniques. All the graphs considered in the sequel are undirected ones.

### 4.4.1 Markov random fields

Given a graph $G = (V, E)$, a set of random variables $X = \{X_i\}_{i \in V}$ indexed by the vertex set $V$ form an MRF w.r.t. $G$ if their joint probability distribution function belongs to a family of distributions $L(G)$. This family will be defined after stating the following definitions :

- **Neighbors :** The set of neighbors of node $i$, denoted by $\eta(i)$, can be defined as

$$\eta(i) = \{u \in V \mid (u, i) \in E\}. \tag{4.16}$$

- **Clique :** A clique is defined as a subset of nodes such that there exists an edge between all pairs of nodes in the subset :

$$C \subset V \text{ is a clique if } \forall (u, v) \in C \times C, (u, v) \in E. \tag{4.17}$$

- **Maximal clique :** It is a clique such that it is not possible to include any other node from the graph in the set without breaking the clique property.
- **Loop :** It is an $n$-tuple $(u_1, \cdots, u_n)$ $(n \geq 3)$ such that $\forall p \in \{1, \cdots, n-1\}$, $(u_p, u_{p+1}) \in E$ and $u_1 = u_n$.
- **Tree :** It is defined as a graph in which there is one, and only one, path between any pair of nodes (i.e., connected and do not have loops).

Let $\mathbf{C}$ be the set of maximal cliques of the graph $G$, then the family of distributions $L(G)$ associated to $G$ can be expressed as

$$L(G) = \left\{ p \mid p(\mathbf{X} = \mathbf{x}) = \frac{1}{Z} \prod_{C \in \mathbf{C}} \Psi_c(\mathbf{x}_C),\ \Psi_c(\mathbf{x}_C) \geq 0 \right\}, \tag{4.18}$$

where $Z$ is a normalization constant, $X_c$ is the set of random variables associated to the set nodes $C$ and $\mathbf{x}_C$ is the realization value. $\Psi_c$ is called a potential function. This function need not to have a direct relation to conditional distributions defined on the graph cliques.

In the cooperative localization problem, under the assumption that the observations $\{\mathbf{y}_{i,j}\}$ and the positions $\{\mathbf{x}_i\}$ are realizations of independent variables, and by considering the anchor nodes positions as random variables of prior distributions equal to the Dirac delta function, the joint a posteriori distribution factorizes as

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z} \prod_{i \in V} \Phi_i(\mathbf{x}_i) \prod_{(i,j) \in E} \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j), \tag{4.19}$$

where $\mathbf{x} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_m^T, \theta^T]^T$ , $\Phi_i(\mathbf{x}_i) = p(\mathbf{x}_i)$ is the a priori probability or evidence on the position of node $i$, and $\Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = p(\mathbf{y}_{i,j}|\mathbf{x}_i, \mathbf{x}_j)$ is a pair-wise potential function. The distribution function (4.19) belongs to the family $L(G)$ where the potential functions of the maximal cliques are products of pair-wise potential functions and evidences.

**Conditional independence**

For a graph $G$, define the family of distribution functions $L'(G)$ verifying the following property : For any two non-neighboring nodes $i$ and $j$, the random variables $X_i$ and $X_j$ are independent conditionally on the knowledge of all other random variables in the graph (i.e., $X_i \perp X_j | X \backslash \{i,j\}$ where $X \backslash \{i,j\}$ denotes the set of all variables without $X_i$ and $X_j$). The Hammersley-Clifford theorem states that the two families $L(G)$ and $L'(G)$ are the same [106].

For any three disjoint sets of nodes $A$, $B$ and $C$ in $G$, the conditional independence property $X_A \perp X_B | X_C$ is satisfied if all possible paths that connect nodes in $A$ to nodes in $B$ pass through one or several nodes in $C$. For example, in the graph $G$ of Figure 4.2, the following equation (4.20) is verified by any joint probability distribution function belonging to $L(G)$ :

$$p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c | \mathbf{x}_d, \mathbf{x}_e, \mathbf{x}_f) = p(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c | \mathbf{x}_d). \tag{4.20}$$

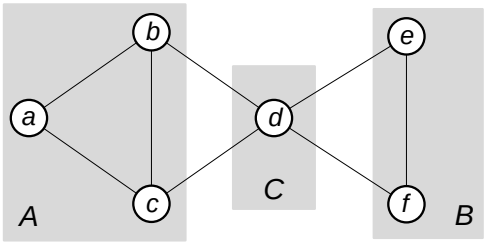



***Figure 4.2*** — Undirected graph of six nodes grouped into three sets.

### 4.4.2 Marginalization

The goal of inference in graphical models is exploiting the graphical structure in order to find efficient algorithms for computing the marginal distribution $p(\mathbf{x}_k|\mathbf{y})$ for a specific node $k$, and finding the most likely sequence of states $\hat{\mathbf{x}}_{MAP}$ that maximizes the joint distribution $p(\mathbf{x}|\mathbf{y})$.

By definition, the marginal distribution is obtained by summing the joint distribution function over all variables except $\mathbf{x}_k$ :

$$p(\mathbf{x}_k|\mathbf{y}) = \int p(\mathbf{x}|\mathbf{y}) \mathrm{d}\mathbf{x}_1 \cdots \mathrm{d}\mathbf{x}_{k-1} \mathrm{d}\mathbf{x}_{k+1} \cdots \mathrm{d}\mathbf{x}_N. \tag{4.21}$$

The computation of marginal distributions has many advantages. One advantage is that it enables estimating the different random variables separately, for example by computing the MMSE estimate according to

$$\begin{aligned} \hat{\mathbf{x}}_{k,MMSE} &= \int \mathbf{x}_k \left( \int p(\mathbf{x}|\mathbf{y}) \mathrm{d}\mathbf{x}_{m+1} \cdots \mathrm{d}\mathbf{x}_{k-1} \mathrm{d}\mathbf{x}_{k+1} \cdots \mathrm{d}\mathbf{x}_N \right) \mathrm{d}\mathbf{x}_k \\ &= \int \mathbf{x}_k p(\mathbf{x}_k|\mathbf{y}) \mathrm{d}\mathbf{x}_k, \end{aligned} \tag{4.22}$$

or by computing the local MAP according to

$$\hat{\mathbf{x}}_{k,local\ MAP} = \arg\max_{\mathbf{x}_k} p(\mathbf{x}_k|\mathbf{y}). \tag{4.23}$$

We mention that the local MAP estimate is in general different from the joint MAP estimate that maximizes the joint posterior distribution $p(\mathbf{x}|\mathbf{y})$ [108]. If the marginals are available at each node, the random variables can be estimated locally. Moreover, the marginals provide a representation of uncertainties and can be used in the detection of potential ambiguities.

### Marginalization in a chain

We consider the Markov chain of Figure 4.3 consisting of $N$ nodes where each node has two neighbors except nodes 1 and $N$ which have only one.

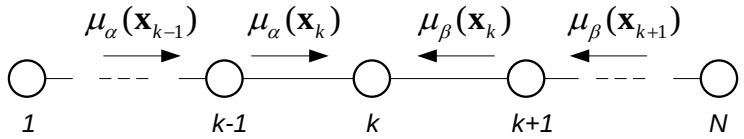



***Figure 4.3*** — A chain graph with the forward and backward messages propagated to node $k$.

The joint distribution of this graph can be expressed as

$$p(\mathbf{x}) = \frac{1}{Z}\Psi_{1,2}(\mathbf{x}_1, \mathbf{x}_2)\ldots\Psi_{N-1,N}(\mathbf{x}_{N-1}, \mathbf{x}_N). \tag{4.24}$$

By assuming that the random variables are discrete, the marginal distribution for node $k$ is

$$p(\mathbf{x}_k) = \sum_{\mathbf{x}_1}\cdots\sum_{\mathbf{x}_{k-1}}\sum_{\mathbf{x}_{k+1}}\cdots\sum_{\mathbf{x}_N} p(\mathbf{x}). \tag{4.25}$$

If we suppose that each node representing a discrete variable has $K$ states, there will be $K^N$ values for $\mathbf{x}$. The naive computation is done by evaluating the joint distribution and then performing the summation explicitly. This would involve storage and computational complexity that scale exponentially with $N$. If we group the potential functions and summations in another way, the desired marginal can be expressed in the form

$$\begin{aligned} p(\mathbf{x}_k) \quad = \quad & \frac{1}{Z}\underbrace{\left[\sum_{\mathbf{x}_{k-1}}\Psi_{k-1,k}(\mathbf{x}_{k-1},\mathbf{x}_k)\ldots\left[\sum_{\mathbf{x}_2}\Psi_{2,3}(\mathbf{x}_2,\mathbf{x}_3)\left[\sum_{\mathbf{x}_1}\Psi_{1,2}(\mathbf{x}_1,\mathbf{x}_2)\right]\right]\right]}_{\mu_\alpha(\mathbf{x}_k)} \times \\ & \underbrace{\left[\sum_{\mathbf{x}_{k+1}}\Psi_{k,k+1}(\mathbf{x}_k,\mathbf{x}_{k+1})\ldots\left[\sum_{\mathbf{x}_N}\Psi_{N-1,N}(\mathbf{x}_{N-1},\mathbf{x}_N)\right]\right]}_{\mu_\beta(\mathbf{x}_k)}. \end{aligned} \tag{4.26}$$

In this way, $p(\mathbf{x}_k)$ factorizes as

$$p(\mathbf{x}_k) = \frac{1}{Z}\mu_\alpha(\mathbf{x}_k)\mu_\beta(\mathbf{x}_k). \tag{4.27}$$

$\mu_\alpha(\mathbf{x}_k)$ can be interpreted as a message passed forwards along the chain from node $k-1$ to node $k$, and $\mu_\beta(\mathbf{x}_k)$ as a message passed backwards from node $k+1$ to node $k$.

$$\begin{aligned}\mu_\alpha(\mathbf{x}_k) &= \sum_{\mathbf{x}_{k-1}} \Psi_{k-1,k}(\mathbf{x}_{k-1}, \mathbf{x}_k)\mu_\alpha(\mathbf{x}_{k-1}).\\ \mu_\beta(\mathbf{x}_k) &= \sum_{\mathbf{x}_{k+1}} \Psi_{k,k+1}(\mathbf{x}_k, \mathbf{x}_{k+1})\mu_\beta(\mathbf{x}_{k+1}).\end{aligned} \tag{4.28}$$

The cost of evaluating $p(\mathbf{x}_k)$ is now $O(NK^2)$ instead of $O(K^N)$.

When the random variables are continuous, equation (4.27) applies and the sums in (4.28) are replaced by integrals.

When the nodes are separate physical entities, the marginal computation can be carried out in a distributed fashion where messages $\mu_\alpha(\mathbf{x}_{k-1})$ and $\mu_\beta(\mathbf{x}_{k+1})$ are received from neighboring nodes $k-1$ and $k+1$, respectively. This message passing scheme enables computing the exact marginals in tree graphs (i.e., graphs without loops) and is called the sum-product algorithm. For the computation of the joint MAP sequence of states in tree graphs, another distributed message passing algorithm exists and is called the max-product algorithm. These two algorithms are described in the next section.

### 4.4.3 Sum-product and max-product algorithms

#### Sum-product algorithm

This algorithm is also called belief propagation. It can be implemented in an iterative and distributed way, in which messages are exchanged in parallel. It can proceed as follows. First, all the messages are initialized to an arbitrary constant value, for example

$$m_{j,i}^{(0)}(\mathbf{x}_i) = 1. \tag{4.29}$$

Then, the message from node $i$ to node $j$ at iteration $l$ is updated as follows :

$$m_{j,i}^{(l)}(\mathbf{x}_i) = \alpha \int \Phi_j(\mathbf{x}_j)\Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \prod_{k\in\eta(j)\backslash i} m_{k,j}^{(l-1)}(\mathbf{x}_j)\mathrm{d}\mathbf{x}_j, \tag{4.30}$$

where $\alpha$ is a multiplicative constant introduced to avoid numerical underflow and thus contribute to the stability of the computations, and $\eta(j)\backslash i$ is the set of neighbors of node $j$ without node $i$. The belief of node $i$ at iteration $l$ is given by

$$b_i^{(l)}(\mathbf{x}_i) = \frac{1}{Z}\Phi_i(\mathbf{x}_i) \prod_{j\in\eta(i)} m_{j,i}^{(l)}(\mathbf{x}_i). \tag{4.31}$$

In tree graphs, the needed number of iterations before convergence is equal to the graph diameter (i.e., longest shortest path) and the beliefs computed at convergence are equal to the true marginals.

**Max-product algorithm**

Let us consider the chain network example of the previous section where we were interested in finding the joint MAP estimate. A naive algorithm is to compute the value of the joint distribution for each of the $K^N$ combinations of states, and then find the maximum. If, instead, the summation in the forward and backward messages of equations (4.28) are replaced by the maximization operator, then the set of states that maximize the different computed beliefs correspond to the joint MAP estimate. This algorithm is called the max-product algorithm. The message from node $i$ to node $j$ corresponds to

$$m_{j,i}^{(l)}(\mathbf{x}_i) = \alpha \max_{\mathbf{x}_j} \left( \Phi_j(\mathbf{x}_j)\Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \prod_{k \in \eta(j)\setminus i} m_{k,j}^{(l-1)}(\mathbf{x}_j) \right). \tag{4.32}$$

For tree graphs, the following statements can be proven [108] :

- The max-product algorithm converges to a unique fixed point in a finite number of iterations equal to the graph diameter.
- At convergence, the belief for any value $\mathbf{x}_k$ of a node $k$ is the maximum of the joint posterior distribution, conditioned on that node having the value $\mathbf{x}_k$ :

$$b_k(\mathbf{x}_k) = \max_{\mathbf{x}} p(\mathbf{x}|\mathbf{x}_k). \tag{4.33}$$

- Define the max-product assignment $\theta^* = [\mathbf{x}_{m+1}^{*T}, \cdots, \mathbf{x}_N^{*T}]^T$ where $\mathbf{x}_k^* = \arg\max_{\mathbf{x}_k} b_k(\mathbf{x}_k)$. Then $\theta^*$ is the joint MAP assignment (i.e., $\theta^* = \hat{\theta}_{MAP}$).

Messages equations (4.30) and (4.32) can be computed either at node $j$ or at node $i$. In the former case, the messages that node $j$ propagates to its neighbors are different, while in the latter case, node $j$ can propagate its belief at iteration $l-1$ and this information is the same for all the neighbors. The message (4.30) can be computed at node $i$ from the belief received from node $j$ according to

$$m_{j,i}^{(l)}(\mathbf{x}_i) = \alpha \int \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \frac{b_j^{(l-1)}(\mathbf{x}_j)}{m_{i,j}^{(l-1)}(\mathbf{x}_j)} \mathrm{d}\mathbf{x}_j. \tag{4.34}$$

Thus, the belief can be broadcast to all the neighbors instead of sending a dedicated message to each neighbor, allowing a reduction in the number of transmissions. But we can notice in (4.34) that the message $m_{i,j}^{(l-1)}(\mathbf{x}_j)$ needs to be known at node $i$ and the reduction in the number of transmissions comes at the expense of a double computation of the messages.

**Graphs with loops - loopy belief propagation**

The belief propagation (BP) rules given above are derived for singly connected networks (i.e., tree graphs in which there is a single path between any two nodes), where the convergence to true marginals is guaranteed.

When the same rules are applied to graphs with loops, they are called 'loopy BP'. In this case, the computed beliefs may differ from the true marginals. A relationship is derived in

[108] between steady state beliefs (i.e., beliefs at BP convergence) and the true marginals for graphs with a single loop when the state space is discrete. It is also shown that the computed beliefs provide good approximations.

## 4.5 Nonparametric belief propagation

The integral (4.30) can be computed analytically when the random variables are discrete and have a finite number of states or when the joint posterior distribution is multivariate Gaussian. When these conditions are not fulfilled, as in the case of the localization problem, this equation cannot be computed in a closed-form and must be replaced by an approximation.

In [109], the nonparametric BP (NBP) algorithm is developed, which is based on stochastic methods for propagating kernel-based approximations of the messages. The heart of the NBP is a Monte-Carlo integration of (4.30).

The NBP is applied in the first phase of the two-phases NBP solution developed in Section 4.7. In the interest of completeness, we provide a recast of the NBP algorithm [109] in this section.

### 4.5.1 Monte-Carlo integration

Let $p_{j,i}$ be the following probability distribution

$$p_{j,i}(\mathbf{x}_j) = \frac{1}{Z}\Phi_j(\mathbf{x}_j) \prod_{k\in\eta(j)\backslash i} m_{k,j}(\mathbf{x}_j) \tag{4.35}$$

where the iteration index has been dropped for simplification. By drawing $M$ independent samples $\{\mathbf{s}_{j,i}^q\}_{q=1}^M$ from $p_{j,i}$, we can approximate it as

$$p_{j,i}(\mathbf{x}_j) \approx \frac{1}{M}\sum_{q=1}^{M} \delta(\mathbf{s}_{j,i}^q, \mathbf{x}_j) \tag{4.36}$$

where $\delta$ is the dirac delta function. The drawn samples will be also called particles. A Monte-Carlo integration of (4.30) is obtained by replacing $p_{j,i}$ by (4.36) yielding the approximate message $\tilde{m}_{j,i}$ :

$$\begin{aligned} \tilde{m}_{j,i}(\mathbf{x}_i) &\propto \int \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \sum_{q=1}^{M} \delta(\mathbf{s}_{j,i}^q, \mathbf{x}_j) \mathrm{d}\mathbf{x}_j \\ &\propto \sum_{q=1}^{M} \Psi_{i,j}(\mathbf{x}_i, \mathbf{s}_{j,i}^q). \end{aligned} \tag{4.37}$$

A classical solution in Monte Carlo integration is to resort to importance sampling when it is difficult to draw samples directly from $p_{j,i}$. Importance sampling is carried out as follows : Let $g_{j,i}(\mathbf{x}_j)$ be an importance probability distribution function verifying $g_{j,i}(\mathbf{x}_j) > 0$ when

$p_{j,i}(\mathbf{x}_j) > 0$. Equation (4.30) can be written as

$$m_{j,i}(\mathbf{x}_i) \propto \int \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \frac{p_{j,i}(\mathbf{x}_j)}{g_{j,i}(\mathbf{x}_j)} g_{j,i}(\mathbf{x}_j) \mathrm{d}\mathbf{x}_j. \tag{4.38}$$

The samples $\{\mathbf{s}_{j,i}^q\}$ can be drawn from the importance function $g_{j,i}$, in which case $\tilde{m}_{j,i}$ becomes the weighted mixture

$$\tilde{m}_{j,i}(\mathbf{x}_i) \propto \sum_{q=1}^{M} \pi_{j,i}^q \Psi_{i,j}(\mathbf{x}_i, \mathbf{s}_{j,i}^q), \tag{4.39}$$

where $\pi_{j,i}^q \propto p_{j,i}(\mathbf{s}_{j,i}^q)/g_{j,i}(\mathbf{s}_{j,i}^q)$ is an importance weight. The estimator (4.39) of $m_{j,i}$ is unbiased, and its variance depends on the considered importance function and tends to zero with $M$.

By choosing the importance function equal to the approximation of the belief

$$\tilde{b}_j(\mathbf{x}_j) = \frac{1}{Z} \Phi_j(\mathbf{x}_j) \prod_{k \in \eta(j)} \tilde{m}_{k,j}(\mathbf{x}_j), \tag{4.40}$$

the drawn samples are the same for all the neighbors of node $j$ and sampling can be done only once, and the attributed weights are $\pi_{j,i}^q \propto 1/\tilde{m}_{i,j}(\mathbf{s}_{j,i}^q)$.

### 4.5.2 Kernel-based message approximation

It is not always straightforward to draw samples directly from the approximate belief (4.40), which is a product of mixtures of the form of (4.39), or to select a suitable importance function that reduces the variance of errors due to approximations. A second stage of the NBP is to further stochastically approximate the message (4.39), where each particle $\mathbf{s}_{j,i}^q$ is propagated to node $i$ by drawing a sample $\mu_{j,i}^q$ for $\alpha \Psi_{i,j}(\mathbf{x}_i, \mathbf{s}_{j,i}^q)$, $\alpha$ being a normalization value, and then placing a Gaussian kernel at every $\mu_{j,i}^q$ with appropriate covariance matrix. The message becomes the following mixture of $M$ Gaussian components :

$$\tilde{m}_{j,i}(\mathbf{x}_i) \propto \sum_{q=1}^{M} \pi_{j,i}^q \mathcal{N}(\mathbf{x}_i; \mu_{j,i}^q, \Sigma_{j,i}) \tag{4.41}$$

where $\mathcal{N}(\mathbf{x}_i; \mu_{j,i}^q, \Sigma_{j,i})$ is distribution function of the Gaussian kernel of centroid $\mu_{j,i}^q$ and bandwidth (or covariance matrix) $\Sigma_{j,i}$. To perform this step we assume that $\int \Psi_{i,j}(\mathbf{x}_i, \mathbf{s}_{j,i}^q) \mathrm{d}\mathbf{x}_i < \infty$.

When there are spatial constraints on the values of $\mathbf{x}_i$, they can be taken into account in this phase by keeping the samples $\mu_{j,i}^q$ that are inside the feasibility set. Thus, NBP can efficiently handle the presence of constraints. This point is very important in localization problems where the nodes might be known to lie inside admissible space regions.

### 4.5.3 Kernel-based belief approximation

This approximation consists of sampling $\tilde{b}_i(\mathbf{x}_i)$ and placing a Gaussian kernel at each drawn sample $\mu_i^q$ :

$$\tilde{b}_i(\mathbf{x}_i) = \sum_{q=1}^{M} w_i^q \mathcal{N}(\mathbf{x}_i; \mu_i^q, \Sigma_i), \tag{4.42}$$

where $w_i^q$ are non-negative weights having the sum equal to one, and the bandwidth $\Sigma_i$ is set to be the same for all components. We mention that using different bandwidths is also possible.

For the moment, if we assume that the local evidence $\Phi_i(\mathbf{x}_i)$ is represented by a weighted Gaussian mixture, then $\tilde{b}_i(\mathbf{x}_i)$, given by (4.40), is a product of $d = |\eta(i)| + 1$ Gaussian mixtures, each containing $M$ components, and will produce another Gaussian mixture with $M^d$ components. To sample directly from this product, we need to find the weights of the $M^d$ components. Then a component is selected and a sample is drawn from it. Thus, direct sampling would require $O(M^d)$ operations. However, several methods for drawing samples from products of Gaussian mixtures are investigated in [110], such as Gibbs sampling and mixture importance sampling with a complexity of at least $O(dM^2)$. When there are constraints on the values of $\mathbf{x}_i$, they are also taken into account in this phase and $\mu_i^q$ should verify these constraints.

In loopy networks, the NBP algorithm can be processed until a predefined number of iterations is reached or until the Kullback-Leibler(KL) divergence [1] between the beliefs at two consecutive iterations becomes less than a fixed threshold. The different steps of the NBP algorithm are summarized in Table 4.1.

### 4.5.4 Application to localization

NBP has been first applied to cooperative localization in [55], and then, several variants have emerged. In [88], samples are drawn by building a box that covers the region where anchors radio ranges overlap. In [3], measurement errors are assumed to lie in known intervals allowing the construction of limited space regions for each node, and then, rejection sampling is used when drawing samples from the beliefs equations. Both variants allow the reduction of the number of samples needed to achieve a good accuracy. Concerning the errors due to the loops, several solutions have been proposed to alleviate them [111, 112, 113].

Kernel-based message approximation performs by sampling the pair-wise potential functions. In the case of a 2-dimensional localization and when observation $\mathbf{y}_{i,j}$ between nodes $i$ and $j$ is a ranging observation and does not carry any information about the direction or the absolute positions (i.e., $p(\mathbf{y}_{i,j}|\mathbf{x}_i, \mathbf{x}_j) \triangleq p(\mathbf{y}_{i,j}|d_{i,j})$), then the sampling operation $\mu_{j,i}^q \sim \alpha\Psi_{i,j}(\mathbf{x}_i, \mathbf{s}_{j,i}^q)$ is equivalent to

$$\mu_{j,i}^q = \mathbf{s}_{j,i}^q + d_{j,i}^q[\ \cos(\theta^q)\ ;\ \sin(\theta^q)\ ] \tag{4.43}$$

where $\theta^q$ is drawn from the uniform distribution $\mathcal{U}(0, 2\pi)$ and $d_{j,i}^q$ is drawn from $\alpha p(\mathbf{y}_{i,j}|d_{i,j})$, $\alpha$ being a normalization value. A similar operation can be performed when the space dimension is three.

1. The Kullback-Leibler(KL) divergence between two distributions $p_X$ and $q_X$ is measure of closeness between them and is given by

$$KL(p_X \| q_X) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})} \mathrm{d}\mathbf{x}.$$

The KL divergence is always non-negative and is equal to zero if and only if the two distributions are equal.

0 : NBP algorithm
1 : **for** each node $i$ in $V$
2 : **for** each node $j$ in $\eta(i)$
3 : **initialize** the received message $m_{j,i}^{(0)}(\mathbf{x}_i) = 1$
4 : **approximate** the belief $b_i^{(0)}$ using a mixture of Gaussian components
5 : **draw** $M$ independent samples $\{\mathbf{s}_i^q\}_{q=1}^M$ from the belief approximation
6 : **for** each node $j$ in $\eta(i)$
7 : **for** $q = 1, \cdots, M$
8 : **sample** $\mu_{i,j}^q \sim \alpha\Psi_{j,i}(\mathbf{x}_j, \mathbf{s}_i^q)$
9 : **set** $\pi_{i,j}^q = 1/M$
10 : **send** the set $\{\pi_{i,j}^q, \mu_{i,j}^q\}$ to node $j$
11 : **iterate** until convergence or for a fixed number of iterations
12 : **for** each node $i$ in $V$
13 : **receive** messages from neighboring nodes $\eta(i)$
14 : **approximate** the belief $\tilde{b}_i$ using a mixture of Gaussian components
15 : **draw** $M$ independent samples $\{\mathbf{s}_i^q\}_{q=1}^M$ from the belief approximation
16 : **for** each node $j$ in $\eta(i)$
17 : **for** $q = 1, \cdots, M$
18 : **sample** $\mu_{i,j}^q \sim \alpha\Psi_{j,i}(\mathbf{x}_j, \mathbf{s}_i^q)$
19 : **set** $\pi_{i,j}^q \propto 1/\tilde{m}_{j,i}(\mathbf{s}_i^q)$
20 : **send** the set $\{\pi_{i,j}^q, \mu_{i,j}^q\}$ to node $j$

***Table 4.1*** — Implementation of the NBP with a parallel scheduling.

The kernel covariance matrix, which is also called kernel bandwidth, is a smoothing parameter. The following thumb rule is adopted in [55] for computing this matrix for a set of weighted samples $\{w_i^q, \mu_i^q\}_{q=1}^M$ :

$$\Sigma_i = \frac{1}{M^{1/3}} \sum_{q=1}^{M} w_i^q (\mu_i^q - \bar{\mu}_i)(\mu_i^q - \bar{\mu}_i)^T \tag{4.44}$$

where $\bar{\mu}_i = \sum w_i^q \mu_i^q$. This equation might result in high eigenvalues of the covariance matrix when the nodes are distant from each other or when the belief has several modes as the samples become far from each other and spread over a wide space region. We propose another adaptive technique for choosing the kernel bandwidth. It proceeds by clustering the samples (e.g, K-means clustering) and computing a covariance matrix for each cluster using (4.44).

To illustrate the effect of kernel bandwidth selection, we consider the message propagated from an anchor node (node $j$) at position $\mathbf{x}_j = [0, 0]^T$ to a target node (node $i$) where the measurement is $y_{i,j} = d_{i,j} + e_{i,j} = 10m$ and $e_{i,j} \sim \mathcal{N}(0, \sigma^2)$ with $\sigma = 0.5m$. Figure 4.4(a) shows the true message. Figure 4.4(b) shows the kernel-based message approximation with $M = 500$ samples and covariance matrix computed according to thumb rule (4.44). Finally, Figure 4.4(c) shows the kernel-based message approximation with $M = 500$ samples and covariance matrix computed according to the adaptive method by taking four clusters. The covariance matrix computed using (4.44) has higher eigenvalues than the one computed using

the adaptive method, and the obtained ring shaped message has higher values over a wider area. But however, the approximation in Figure 4.4(c) has more modes than the one in Figure 4.4(b) since the kernel bandwidth is a smoothing parameter (when the covariance matrix has higher eigenvalues, the kernel distribution function will take higher values over wider space areas).

The average KL divergence between the approximated message and true message is plotted in Figure 4.5 as a function of the error standard deviation $\sigma$, for the two covariance computation techniques and for several values of $M$. The used equation for computing the average KL divergence between two probability distributions $p(\mathbf{x})$ and $q(\mathbf{x})$ is the following :

$$KL_{av}(p\|q) = \int (p(\mathbf{x}) - q(\mathbf{x}))\log\frac{p(\mathbf{x})}{q(\mathbf{x})}\mathrm{d}\mathbf{x}. \tag{4.45}$$

As can be noticed, the KL divergence is smaller for the adaptive technique which results in a better approximation of the messages.

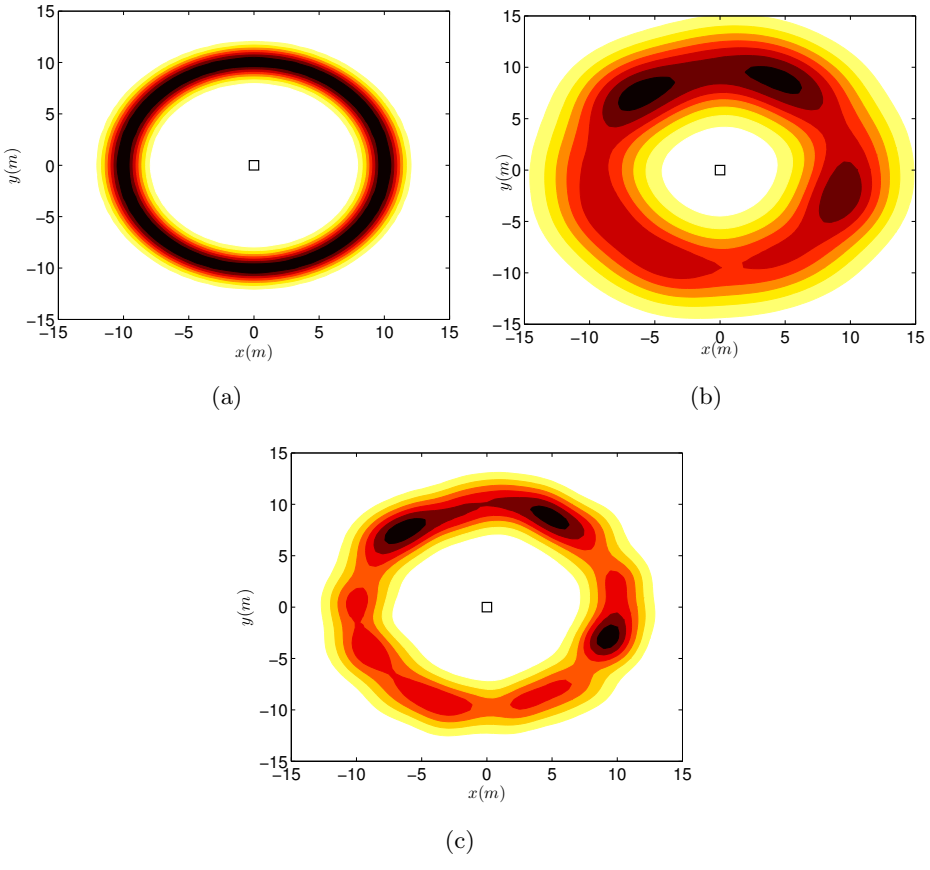



***Figure 4.4*** — (a) True message. (b) Kernel based approximation with kernel bandwidth computed according to (4.44). (c) Kernel based approximation with adaptive kernel bandwidth computation.

An example illustrating the application of the NBP algorithm to cooperative localization is provided in Figure 4.6 where there are two target nodes each connected to two anchor nodes and one target node. At the first iteration, the product of messages propagated from anchor

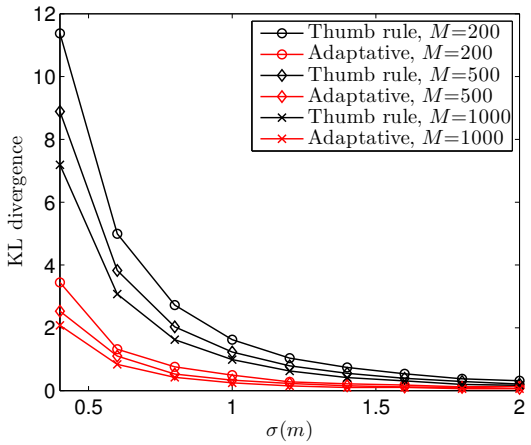



***Figure 4.5*** — Variation with $\sigma$ of the average KL divergence between the true message and the approximated one.

nodes 1 and 2 to target node 5 result in a bimodal belief, and at the second iteration, the message received from target node 6 makes the final belief at convergence unimodal. It can be noticed that the two modes of the approximated belief in Figure 4.6(b) are not equiprobable due to the randomness of the NBP method and the limitation of the number of samples.

Another example illustrating the powerfulness of the NBP algorithm when locations are not globally identifiable is provided in Figure 4.7 where target node 7 has four feasible solutions and the corresponding belief after three NBP iterations has four modes. Thus, the computed beliefs can be exploited to detect the potential ambiguities and to avoid wrong geometric embeddings of the network, making the NBP more attractive over other distributed successive refinement solutions.

### 4.5.5 State estimation

After computing a kernel-based approximation of the belief of node $i$

$$\tilde{b}_i(\mathbf{x}_i) = \sum_{q=1}^{M} w_i^q \mathcal{N}(\mathbf{x}_i; \mu_i^q, \Sigma_i), \tag{4.46}$$

the MMSE estimator of the position of node $i$ becomes

$$\hat{\mathbf{x}}_{i,MMSE} = \sum_{q=1}^{M} w_i^q \mu_i^q. \tag{4.47}$$

The local MAP estimator is the mode of (4.46) with the highest probability. To find all the modes, no direct method exists and iterative numerical algorithms are necessary. The number of modes depends on the separation between the components centroids and their covariance matrices, and is less than the number of components when they have the same covariance matrix. A hill-climbing algorithm starting from every centroid will not miss any mode [114]. Two such algorithms adapted to the Gaussian components case are described in [114] : One is the the gradient-quadratic search which is a combination of the Newton method and the

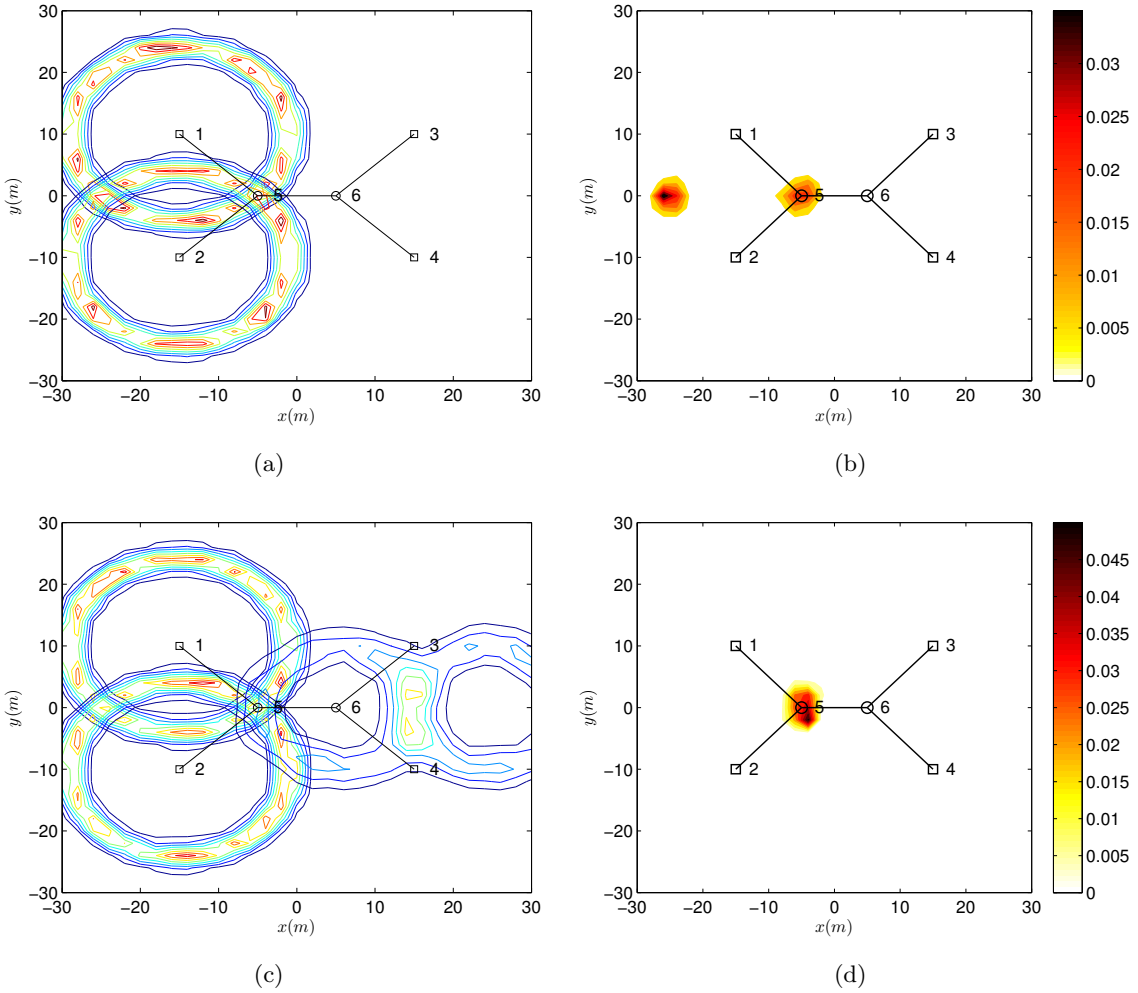
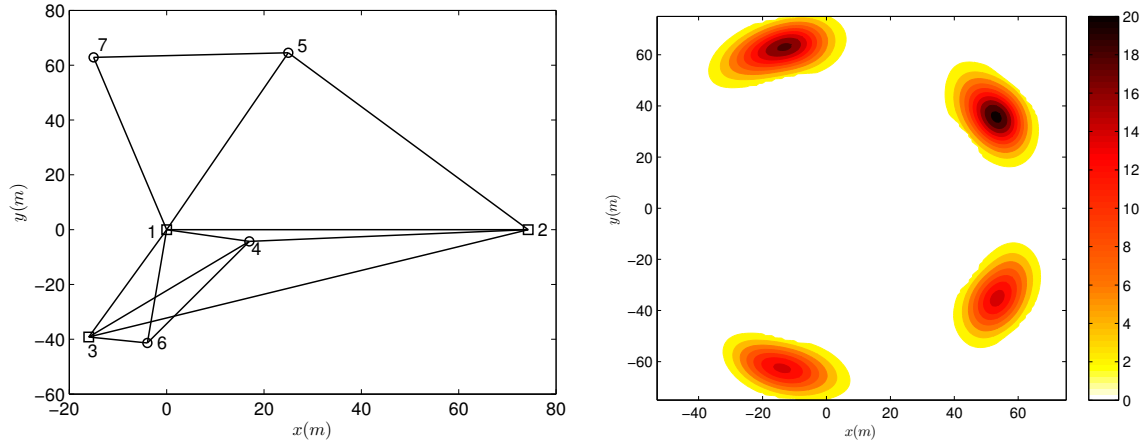



***Figure 4.6*** — (a) Messages propagated from anchor nodes 1 and 2 to target node 5 at the first iteration of the NBP. (b) Bimodal belief of node 5 at the first iteration of the NBP obtained by the product of messages from nodes 1 and 2. (c) Messages propagated from anchor nodes 1 and 2 and target node 6 to target node 5 at the second iteration of the NBP. (d) Unimodal belief of node 5 at the second iteration of the NBP.

gradient ascent, and another is the fixed point method. The fixed point method requires more iterations than the gradient-quadratic search but is simpler to implement and does not require a step size selection.

In the localization problem, the beliefs only have few modes. When the number of components $M$ is large, performing iterative searches that start at every centroid results in a high complexity and many searches will converge to the same mode. Several solutions can be used for reducing this complexity. One solution is to randomly select a subset of starting points from the $M$ centroids. Another solution is to approximate the Gaussian mixture by another mixture containing fewer components. Several methods have been developed for reducing the number of components [115].

Finally we mention that approximating the belief by a Gaussian mixture with fewer components can be beneficial for reducing the amount of data to be exchanged between




(a) (b)

***Figure 4.7*** — (a) Network of 7 nodes where target nodes 5 and 7 are not uniquely solvable. (b) Multimodal belief of node 7 after four iterations of the NBP.

the nodes. In this case, the trade off between the amount of transmitted samples and the complexity of the mixture reduction operation should be taken into account.

## 4.6 Other message passing algorithms

The goal of message passing algorithms is to compute, in a distributed way, approximate distributions $b_i(\mathbf{x}_i)$ of the marginal distributions $p(\mathbf{x}_i|\mathbf{y})$ so that the joint approximate $b(\mathbf{x})$ is close to the joint distribution $p(\mathbf{x}|\mathbf{y})$. The KL divergence has been considered as a measure of closeness and is given by

$$KL(b_X \| p_{X|\mathbf{Y}}) = \int b(\mathbf{x}) \log \frac{b(\mathbf{x})}{p(\mathbf{x}|\mathbf{y})} \mathrm{d}\mathbf{x}. \tag{4.48}$$

To make the computation tractable, $b_X$ is restricted to belong to a certain class $\mathcal{C}$ and we try to find the the distribution that minimizes the KL divergence [116] :

$$\hat{b}_X = \arg\min_{b_X \in \mathcal{C}} KL(b_X \| p_{X|\mathbf{Y}}). \tag{4.49}$$

For the belief propagation algorithm, it is shown in [116] that the beliefs obtained at convergence correspond to a local stationary point of the KL divergence when $\mathcal{C}$ is the set of probability distributions that factorize as

$$b_X(\mathbf{x}) = \frac{\prod_{(i,j)\in E} b_{i,j}(\mathbf{x}_i, \mathbf{x}_j)}{\prod_{i \in V} b_i^{|\eta(i)|-1}(\mathbf{x}_i)}, \tag{4.50}$$

where $|\eta(i)|$ denotes the cardinality of the set of neighbors of node $i$ and the joint probability distribution $b_{i,j}$ verifies the condition $\int b_{i,j}(\mathbf{x}_i, \mathbf{x}_j)\mathrm{d}\mathbf{x}_j = b_i(\mathbf{x}_i)$. The joint posterior distribution of tree graphs factorizes as (4.50), and thus, the beliefs obtained at convergence are equal to the true marginals.

Variational message passing simplifies the dependency structure by choosing distributions in which disjoint groups of variables are independent [106]. In other words, $b_{\mathbf{X}}$ factorizes as

$$b_X(\mathbf{x}) = \prod_{i \in V} b_i(\mathbf{x}_i). \tag{4.51}$$

Another message passing technique is the expectation propagation which minimizes the inclusive KL divergence $KL(p_{\mathbf{X}|\mathbf{Y}}||b_{\mathbf{X}})$ by considering distributions that factorize according to (4.51). Variational message passing and expectation propagation have been applied to cooperative localization in [89] and [90], respectively, where the marginal approximate distributions are additionally restricted to be Gaussian densities, in which case the exchanged messages reduce to the first two moments instead of the large number of samples in the case of NBP algorithms. But this may result in an accuracy reduction since multimodal distributions cannot be represented.

Figure 4.8 illustrates the information flow in message passing algorithms for one target node. The computations of messages and approximate marginals differ from one message passing algorithm to another.

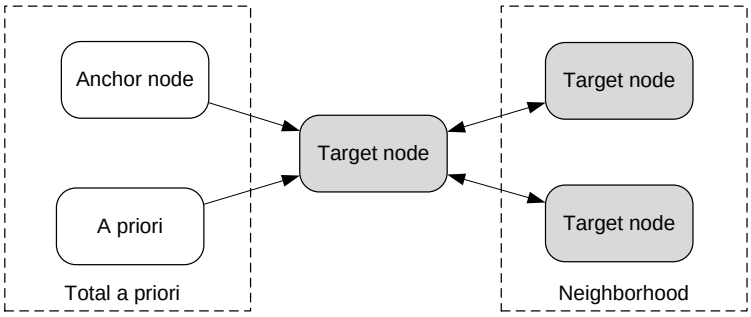



***Figure 4.8*** — Representation of messages flow for one target node.

## 4.7 Two-phases NBP and flip ambiguity mitigation

The two-phases NBP (TP-NBP), which is a main contribution of this chapter, is developed in this section. One application of this algorithm is the mitigation of the flip ambiguity. Before developing the TP-NBP, we start by describing the flip ambiguity and several existing approaches to deal with it.

### 4.7.1 Dealing with flip ambiguity

Flip ambiguity in cooperative network localization, which corresponds to erroneous geometrical realizations, is a fundamental problem that can result in high location estimation errors. It is due either to the lack of measurements necessary to achieve a global identifiability or to the topology of the network and noises in measurements. Global identifiability conditions have been addressed in the previous chapter.

Several approaches can be applied to mitigate the flip ambiguity.

One approach is to only localize the nodes with low flip probability as in [52, 53]. In the incremental solution of [53], the computed positions are promoted to anchors in the subsequent iterations. A flip on an anchor node can degrade the remaining position estimates and propagate in an avalanche fashion. In order to avoid this situation, a test is applied for identifying the nodes with high flip probability and removing them from the localization process. While this approach can guarantee a high robustness of the solution, it can leave an important portion of the nodes unlocalized.

A second approach is to preform additional measurements either with a mobile terminal (e.g., Figure 4.9), or by increasing the transmit power in order to acquire measurements from the far nodes [54]. The feasibility of this solution depends on the deployment scenario and the limitations of the maximum power.

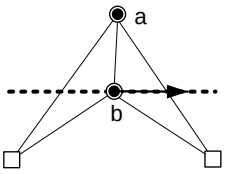



***Figure 4.9*** — The ambiguity on target node $a$ is eliminated after performing measurements with the mobile target node $b$ which is moving on the dashed line.

In some cases, the presence of a priori information may resolve the flip ambiguity, as for example map constraints (e.g., the location of a femto base station or an access point is constrained to be inside the owner's apartment). The NBP method can handle the map constraints by drawing the particles inside the admissible regions.

Another solution is to exploit the connectivity information between the different nodes. The fact that two nodes are not neighbors gives the additional information that they are more likely to be far from each other. Here, our focus will be on exploiting the connectivity information using probabilistic inference in MRFs. This approach has been first considered in [55] using the NBP technique. In this case, a probabilistic connectivity model needs to be assumed. Now, we describe such a model based on average received power.

### Connectivity model based on received power

Two nodes $i$ and $j$ are connected (i.e., can communicate with each other and perform a pair-wise measurement) if the power $P_{i,j}$ (in decibel ($dB$)) at node $i$ transmitted by node $j$ (or vice versa) is above a threshold value $P_{th}$. Such a connectivity model is used in the simulation and analysis of multi-hop networks [79].

We assume the following widely accepted model for the received power :

$$P_{i,j} = P_0 - 10n_p \log_{10}(d_{i,j}) + X_{i,j}, \tag{4.52}$$

where $P_0$ is the average received power at a distance of $1m$, $d_{i,j}$ is the true distance between nodes $i$ and $j$, $n_p$ is the path loss exponent and $X_{i,j}$ is a centered Gaussian random variable of variance $\sigma^2_{sh}$ representing the shadowing loss.

The probability that two nodes are connected is a function of their separating distance. By assuming that $P_{i,j} = P_{j,i}$, this probability is equal to

$$\begin{aligned} p_o(\mathbf{x}_i, \mathbf{x}_j) &= \text{probability}(P_0 - 10n_p\log_{10}(d_{i,j}) + X_{i,j} \geq P_{th}) \\ &= \text{probability}(X_{i,j} \geq f(\mathbf{x}_i, \mathbf{x}_j)) \\ &= Q\left(\frac{10n_p\log_{10}(d_{i,j}/R)}{\sigma_{sh}}\right), \end{aligned} \tag{4.53}$$

where $Q$ is the $Q$-function, $R = 10^{\frac{P_0 - P_{th}}{10n_p}} d_0$ and $p_o = 1/2$ for $d_{i,j} = R$.

The power value is available for two neighboring (i.e., connected) nodes and we can write

$$\begin{aligned} p(P_{i,j} = p_{i,j}, \text{nodes } i \text{ and } j \text{ are neighbors}|\mathbf{x}_i, \mathbf{x}_j) &= p(P_{i,j} = p_{i,j}, P_{i,j} \geq P_{th}|\mathbf{x}_i, \mathbf{x}_j) \\ &= p(P_{i,j} = p_{i,j}|\mathbf{x}_i, \mathbf{x}_j), \end{aligned} \tag{4.54}$$

where $p_{i,j}$ is a particular value taken by $P_{i,j}$. On the other hand,

$$\begin{aligned} p(\text{nodes } i \text{ and } j \text{ are not neighbors}|\mathbf{x}_i, \mathbf{x}_j) &= p(P_{i,j} < P_{th}|\mathbf{x}_i, \mathbf{x}_j) \\ &= 1 - p_o(\mathbf{x}_i, \mathbf{x}_j). \end{aligned} \tag{4.55}$$

Let $E^c = \{(i,j) \in V \times V | i \neq j \text{ and } (i,j) \notin E\}$. The joint distribution of the power measurements and the connectivity information that can be deduced from (4.54) and (4.55) is

$$p(\{P_{i,j} = p_{i,j}\}_{(i,j)\in E}, \{P_{i,j} < P_{th}\}_{(i,j)\in E^c} | \{\mathbf{x}_i\}_{i\in V}). \tag{4.56}$$

In general, the shadowing affecting the different links are correlated. The joint distribution of the shadowing values is multivariate Gaussian with a non-diagonal covariance matrix, and (4.56) is not suitable for the application of the NBP. A multi-link shadowing correlation model is proposed in [79] but is very complex to apply to localization. Instead, if we assume that the shadowing losses on the different links are independent, then the joint a posteriori distribution becomes

$$p\left(\mathbf{x}|\mathbf{y}\right) \propto \prod_{i\in V} \Phi_i(\mathbf{x}_i) \prod_{(i,j)\in E\cup E^c} \Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) \tag{4.57}$$

where $\Phi_i(\mathbf{x}_i)$ is the a priori distribution function of $\mathbf{x}_i$, $\Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = p(\mathbf{y}_{i,j}, P_{i,j} = p_{i,j}|\mathbf{x}_i, \mathbf{x}_j)$ if $(i,j) \in E$ and $\Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = 1 - p_o(\mathbf{x}_i, \mathbf{x}_j)$ otherwise.

In [55, 88, 3], the exponential function (4.58) is assumed as an approximation of (4.53) and has been applied to ambiguity mitigation in variational inference techniques [117].

$$p_o(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\text{log2 } d_{i,j}^2/R^2). \tag{4.58}$$

We mention that site specific models that take into account the layout of the deployment area (e.g., Keenan-Motley model) can be considered and easily handled by the NBP method.

The probabilistic undirected graph associated to (4.57) is fully connected, and the application of message passing algorithms to it is very complex, since it requires an exchange of

messages between all the pairs of nodes. To reduce this complexity, message exchange is limited to the $k$-step neighbors, where two nodes are $k$-step neighbors if their shortest connecting path has a length of $k$ edges.

To reach all the $k$-step neighbors, a message needs to be broadcast several times. For example, to reach a 2-step neighbor, a messages needs to be broadcast twice : A first broadcast by the emitting node and a second broadcast by a direct neighbor. Indeed, to reach all the 2-step neighbors, a message need not to be broadcast by all the direct neighbors as several 2-step neighbors can be connected to the same direct neighbor. Multipoint relaying techniques can be used to reduce the number of redundant retransmissions while diffusing the broadcast messages [118].

### 4.7.2 Two-phases NBP solution

The TP-NBP solution performs in two phases : In the first phase, the NBP is applied by exchanging messages only between direct neighbors and without considering the connectivity information. Then in the second phase, a new algorithm based on estimation in discrete state space is applied. This algorithm is composed of the following steps :

1. For each node, we identify the belief's modes and construct a small set of points consisting of these modes and few points around each mode. This step is performed locally. In the case of a 2-dimensional localization, the set $S_i$ associated to node $i$ can be constructed as follows : For each mode $\mathbf{m}_i^l$, we select $h$ points on the ellipse

$$([x,y]^T - \mathbf{m}_i^l)^T (\Sigma_i^l)^{-1} ([x,y]^T - \mathbf{m}_i^l) = a \tag{4.59}$$

   where $[x,y]^T$ is a coordinates vector, $a$ is a real positive value and $\Sigma_i^l$ is the covariance of the cluster of samples associated to $\mathbf{m}_i^l$ (i.e., nearest kernel centroids to $\mathbf{m}_i^l$). In the simulations, we set $a = 1.5$ and $h = 8$ points. The number of elements of $S_i$ is $h\times$number of modes. This number can be reduced by rejecting the modes with probability smaller than a threshold value.

2. At this point, each node has a small set of points. We apply the discrete analytical version of the BP to find again the beliefs, where the unknown positions take on values in the constructed sets.
   - We can use the sum-product rule, and in this case the message from node $j$ to node $i$ at iteration $n$ is

$$m_{j,i}^{(n)}(\mathbf{s}_i^q) = \sum_{l=1}^{|S_j|} \Psi_{i,j}(\mathbf{s}_i^q, \mathbf{s}_j^l) \prod_{k \in \eta(j) \cup \eta_k(j) \setminus i} m_{k,j}^{(n-1)}(\mathbf{s}_j^l), \tag{4.60}$$

     where $\mathbf{s}_i^q \in S_i$, $q = 1, \cdots, |S_i|$, $\eta(j)$ is the set of direct neighbors of $j$, and $\eta_k(j)$ is the set of indirect neighbors up to the order $k$. We compute $\Psi_{i,j}(\mathbf{s}_i, \mathbf{s}_j) = p(\mathbf{y}_{i,j}, P_{i,j} = p_{i,j} | \mathbf{s}_i, \mathbf{s}_j)$ for direct neighbors, and $\Psi_{i,j}(\mathbf{s}_i, \mathbf{s}_j) = 1 - p_o(\mathbf{s}_i, \mathbf{s}_j)$ for indirect ones.
   - If the max-product is used instead, the message is

$$m_{j,i}^{(n)}(\mathbf{s}_i^q) = \max_{l=1}^{|S_j|} \Psi_{i,j}(\mathbf{s}_i^q, \mathbf{s}_j^l) \prod_{k \in \eta(j) \cup \eta_k(j) \setminus i} m_{k,j}^{(n-1)}(\mathbf{s}_j^l). \tag{4.61}$$

3. The belief at node $i$ is computed with

$$\hat{b}^{(n)}(\mathbf{s}_i^q) = \prod_{k \in \eta(i) \cup \eta_k(i)} m_{k,i}^{(n)}(\mathbf{s}_i^q) \tag{4.62}$$

4. The estimated position is taken as the point with the maximum belief :

$$\hat{\mathbf{x}}_i = \arg \max_{\mathbf{s}_i^q \in S_i} \hat{b}^{(n)}(\mathbf{s}_i^q). \tag{4.63}$$

With this algorithm, the $k$-step neighbors are implicated in the message exchange process, but the amount of data contained in the message is smaller than that of the first phase NBP. Furthermore, this algorithm improves the localization accuracy, even when message exchanges of the second phase occur between the direct neighbors only. This claim will be validated in the next section via Monte Carlo simulations. Additionally, during the second phase, measurements that have not been used during the first phase NBP can be used, and other previously described ambiguity mitigation techniques can be applied.

## 4.8 Numerical results

In this section, we validate advantages of the TP-NBP via two Monte Carlo simulation examples. We also derive a new result : We show, via a Monte Carlo simulation, that the shadowing correlation information can be useful for mitigating the flip ambiguity.

### 4.8.1 Example 1 : Distance measurements in additive Gaussian noise

We consider networks consisting of 4 anchor nodes and 16 target nodes deployed in a $20m \times 15m$ area. The anchor nodes are located as shown in Figure 4.10. The target nodes locations are uniformly drawn inside squares of size $2m \times 2m$, one node per square. The centers of the squares are also shown in Figure 4.10. We divide the target nodes into two categories : The inner nodes that are inside the convex hull of the 4 anchor nodes and the peripheral nodes that are outside this convex hull. The peripheral nodes have higher flip probabilities than the inner ones since they have less neighbors, on average.

For the connectivity, we consider the probabilistic model of (4.53) with $n_p = 3$, $\sigma_{sh} = 8dB$ and range $R$ varying from 4 to $10m$, and consider only rigid networks that cannot have a continuous deformation (i.e., a randomly generated network is retained if it is rigid).

The assumed observations are distance measurements affected by additive Gaussian errors of the same variance $\sigma^2$. We further assume that nodes are mutually neighbors (if $i$ is a neighbor of $j$, then $j$ is a neighbor of $i$) and neighboring nodes share the same measurement (if there are two different measurements, they can be combined in a single one).

The centralized benchmark solution considered is the WLS, where the starting point of the steepest descent optimization is obtained by applying the SDP algorithm for noisy measurements described in [77]. This solution will be also called 'SDP+descent'. In order to have a fair comparison between the NBP and the centralized solution, the potential function of two neighboring nodes $i$ and $j$ is set equal to the likelihood function $\Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = p(\tilde{d}_{i,j}|\mathbf{x}_i, \mathbf{x}_j)$.

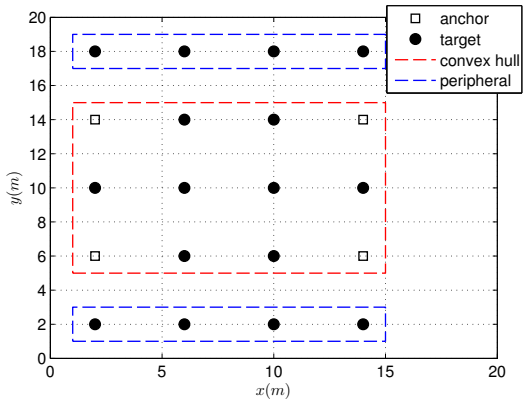
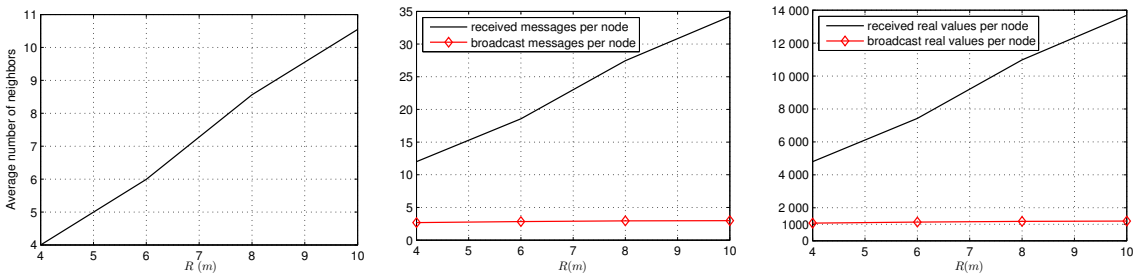



***Figure 4.10*** — Anchor nodes locations and centers of squares in which the target nodes locations are uniformly drawn.

### Communication cost

We assume a perfect medium access (no connection drop, packet loss, interference, etc.). The number of messages broadcast by a node depends on the number of iterations, and the number of messages received by a node depends on the number of neighbors and the number of iterations. Figure 4.11(a) shows the average number of neighbors of a node as a function of $R$.

For the classical NBP, we assume that a node broadcasts a message if it has already computed its belief either from local a priori information or after receiving messages from neighbors. Resampling is performed and all the samples have the same weight, and thus, the exchanged data consists of the samples, and each sample consists of two real values, as the space dimension is two. Figure 4.11(b) shows the variation with $R$ of the average total number of messages broadcast and received per target node when the number of iterations is four, and Figure 4.11(c) shows the corresponding amount of real data values sent and collected by a target node for $M = 200$ particles.

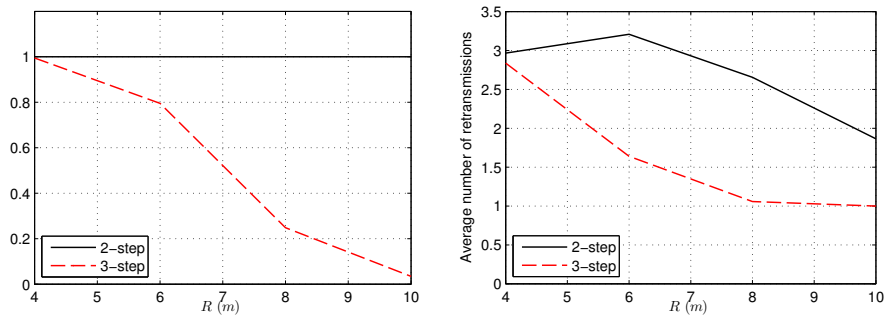



(a) (b) (c)

***Figure 4.11*** — (a) Average number of neighbors of a node. (b) Average number of transmitted and received messages. (c) Average number of transmitted and received real data values. The number of samples is $M = 200$ and the number of iterations is four.

When messages are exchanged between $k$-step neighbors, multipoint relays (MPRs) are considered in order to minimize the number of retransmitted messages. We choose the MPRs

using the heuristic algorithm described in [118]. Figure 4.12(a) shows the probability that a node has 2-step and 3-step neighbors. The average numbers of retransmissions (or number of MPRs) of a message to reach the 2-step and 3-step neighbors are plotted in Figure 4.12(b). We can notice that for $R > 6m$, the number of MPRs is decreasing since the number of direct neighbors is increasing.




(a) (b)

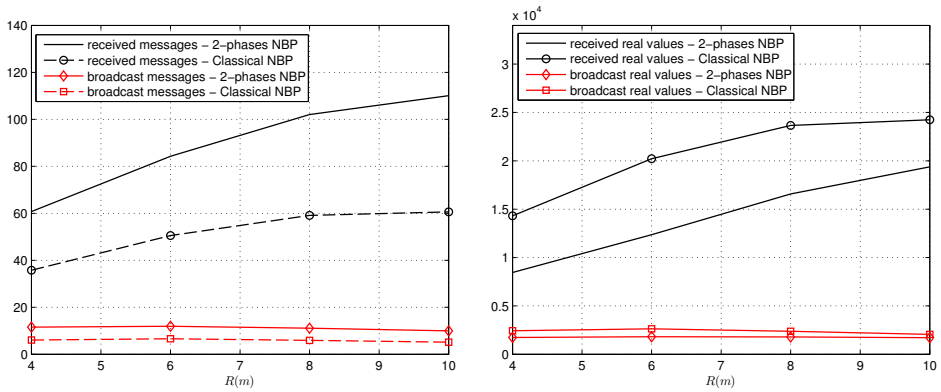
***Figure 4.12*** — (a) Probability that a node has a $k$-step neighbor. (b) Average number of retransmissions per node to reach all $k$-step neighbors.

The message sent by a node to its $k$-step neighbors can be either transmitted by dedicated transmissions in which case every node broadcasts its message then MPRs broadcast the received messages and so on, or it can be concatenated to the message sent to direct neighbors in which case the messages from $k$-step neighbors are deferred $k-1$ iterations.

In Figure 4.13(a) we plot the average total number of messages broadcast and received per node for the classical NBP and the two-phases solution with message exchanges up to 2-step neighbors. The amount of exchanged data per node is plotted in Figure 4.13(b). The number of iterations is fixed to four. We can notice that the number of messages is increased in the two-phases solution as each phase runs four iterations. Nevertheless, this number can be reduced by considering less iterations in the second phase as the beliefs are already computed and the messages from the 2-step neighbors are used to solve the ambiguity. On the other hand, the overall amount of exchanged data is decreased in the two-phases solution as the exchanges occur between the direct neighbors in the first phase with $M = 200$ samples and fewer samples are exchanged in the second phase.

### Localization accuracy

Figures 4.14 to 4.17 show the variation of the root mean square error (RMSE) $\bar{\epsilon} = \sqrt{E\{\|\hat{\mathbf{x}}_i - \mathbf{x}_i\|^2\}}$ with the range $R$ for different error variances. $\bar{\epsilon}$ is obtained by averaging 100 noise and network realizations. Several methods are compared. NBP-MMSE and NBP-MAP solutions consider the mean of the belief samples and the most probable mode of the Gaussian mixture, respectively, after applying the classical NBP with exchanges between direct neighbors only. NBP $k$-step corresponds to the application of the TP-NBP solution, where in the second phase the discrete version of the BP is applied using the sum-product




(a) (b)

***Figure 4.13*** — (a) Average number of transmitted and received messages. (b) Average number of transmitted and received real data values. Messages are exchanged up to the 2-step neighbors

rule (4.60) and messages are exchanged up to the $k$-step neighbors. 1-step neighbors are the direct neighbors.

Several points can be stressed out from these plots :

- NBP-MMSE is more accurate than NBP-MAP. Furthermore, NBP-MMSE is less complex to implement since the position estimate is obtained by a simple averaging of the samples instead of the modes exhaustive search required by NBP-MAP.
- NBP 1-step is more accurate than NBP-MMSE, even though the messages are only exchanged between the direct neighbors at the second phase. A possible justification of this result is the limitation of the number of samples used in approximating the messages, which introduces a randomness in the first phase NBP, while the second phase BP is exact. We can also notice that NBP 1-step is as accurate as the centralized WLS for this deployment and measurements scenario.
- The RMSE is decreasing with $R$ since the nodes are making more measurements. The nodes inside the convex hull have smaller RMSE and less ambiguities than the peripheral nodes.
- NBP 2-step and NBP 3-step bring more accuracy improvement to the peripheral nodes. No improvement is observed at high values of $R$ since the flip probability tends to zero, except for the peripheral nodes when $\sigma = 1.5$ due to the lack in measurements accuracy (Figure 4.17). Thus, by only allowing the nodes with a small number of direct neighbors to request messages from 2 and 3-step neighbors, the complexity and data exchange overhead can be reduced without significant accuracy losses.

The sum-product (4.60) and max-product (4.61) rules are compared in Figure 4.18, where we can notice that the sum-product performs slightly better than the max-product for this scenario.

The median and the 90% errors variations corresponding to the message exchange with the 2-step neighbors are plotted in Figures 4.19 and 4.20, respectively.

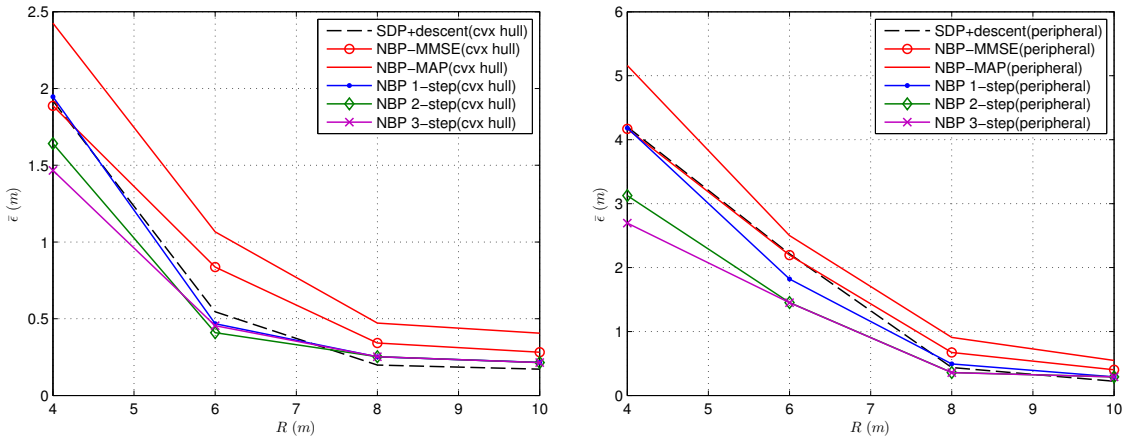



***Figure 4.14*** — Variation of root mean square error with $R$. $\sigma = 0.25$.

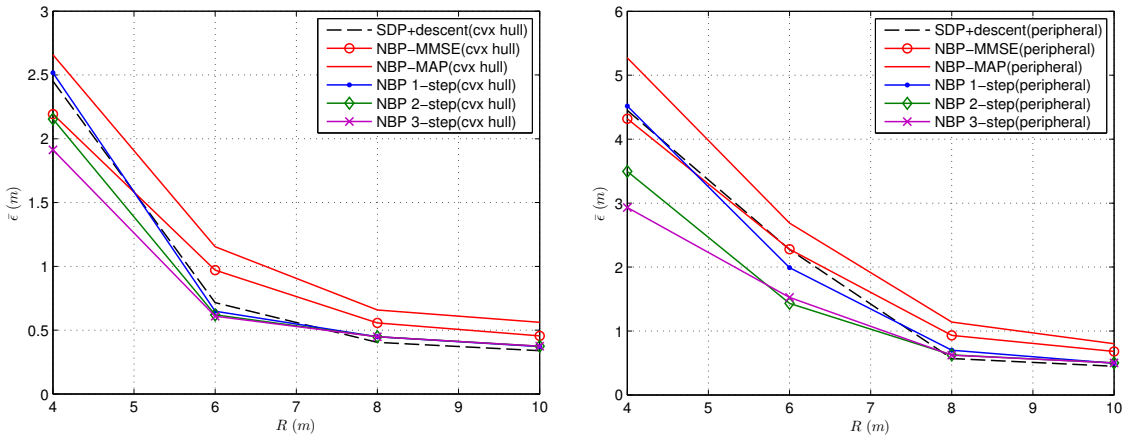



***Figure 4.15*** — Variation of root mean square error with $R$. $\sigma = 0.5$.

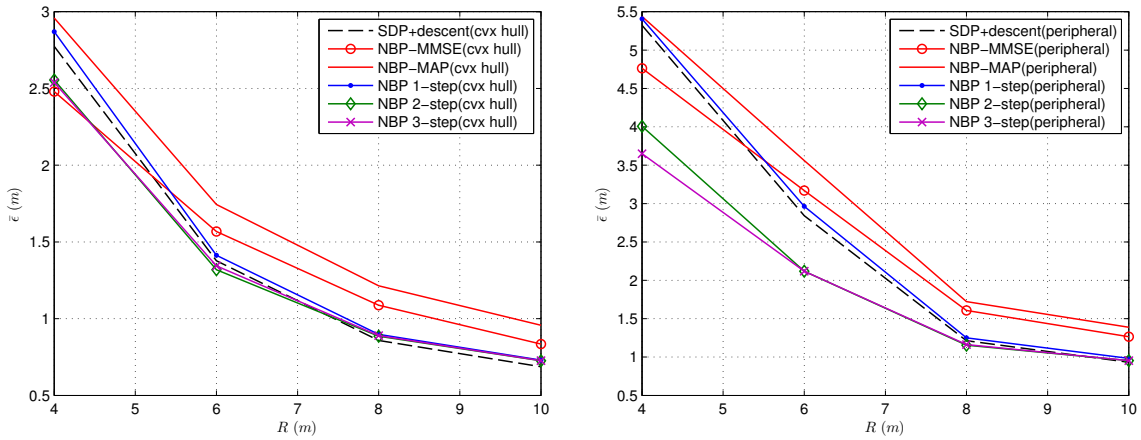



***Figure 4.16*** — Variation of root mean square error with $R$. $\sigma = 1.0$.

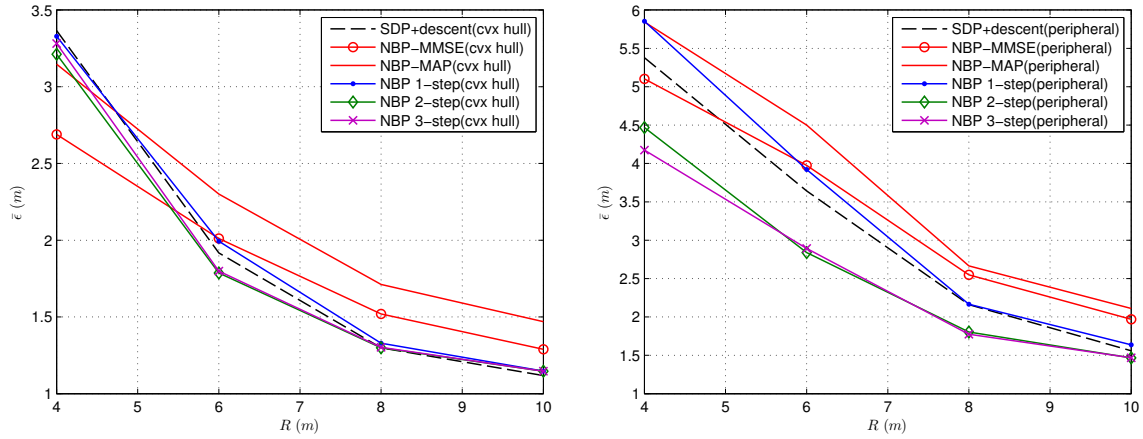



***Figure 4.17*** — Variation of root mean square error with $R$. $\sigma = 1.5$.

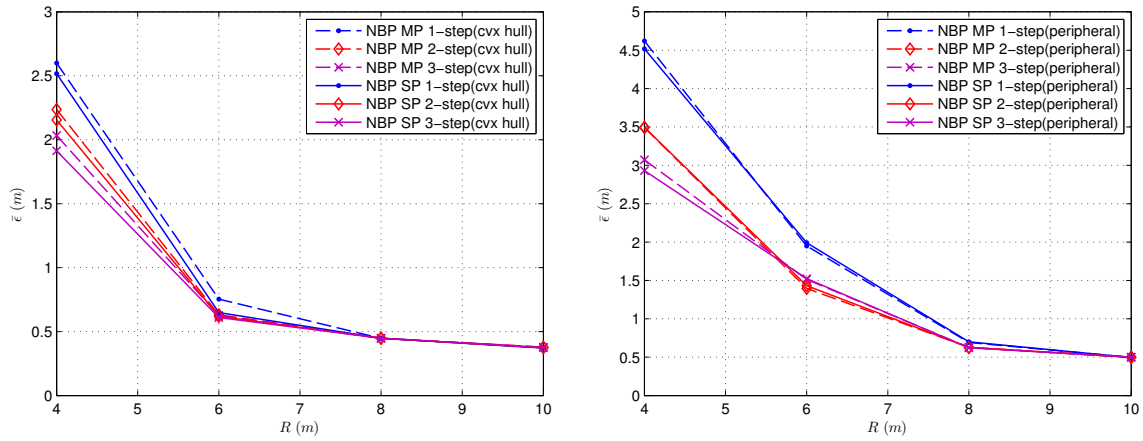



***Figure 4.18*** — Variation of root mean square error with $R$ for the sum-product(SP) and max-product(MP) rules. $\sigma = 0.5$.

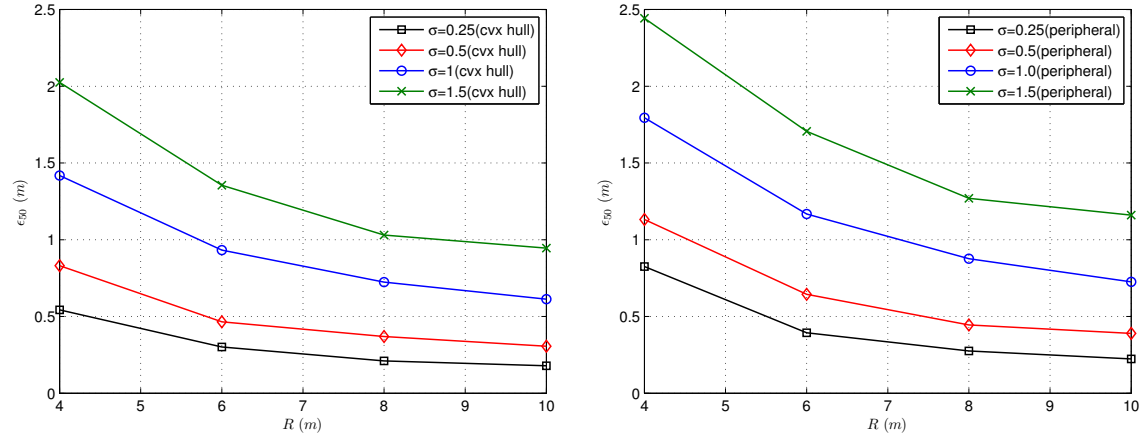



***Figure 4.19*** — Variation of median error $\epsilon_{50}$ with $R$.

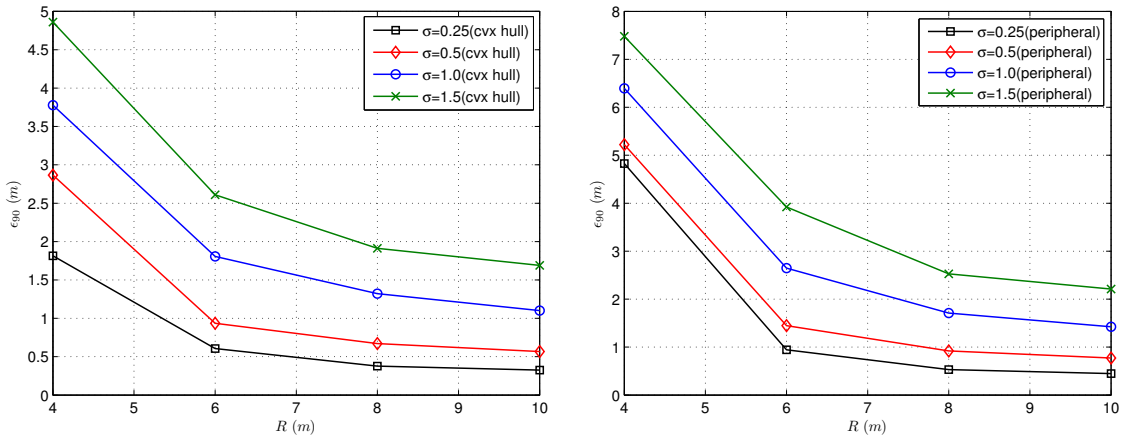



***Figure 4.20*** — Variation of worst case error at 90% error $\epsilon_{90}$ with $R$.

### 4.8.2 Example 2 : Distance measurements in additive noise with outliers

Outliers correspond to observations that are highly deviated from the overall pattern of a distribution. In the case of ranging measurements, an outlier can be caused by an NLoS propagation. This examples studies the performance of the TP-NBP in the presence of outlying observations. We consider networks of size 25 nodes consisting of 4 anchor nodes and 21 target nodes deployed in a $100m \times 100m$ area. The deployment area is partitioned into a grid of $5 \times 5$ squares, each of size $20m \times 20m$. The locations of the anchor nodes are the centers of four squares, as shown in Figure 4.21, and target nodes locations are uniformly drawn inside the remaining squares, one node per square.

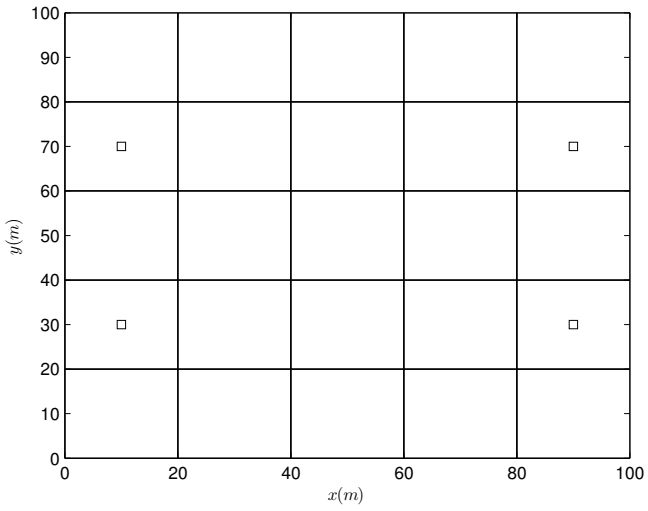



***Figure 4.21*** — Deployment area and the positions of the 4 anchor nodes (example 2).

The probability that two nodes $i$ and $j$ are connected and perform a ranging measurement between each other is assumed to be function of their separating distance $d_{i,j}$ and is given by (4.53).

We take the communication range $R$ varying from 30 to $100m$, and we only consider

connected networks (i.e., there is a path between each pair of the nodes) and not the rigid ones as in the previous example.

The measured distance between two neighboring nodes $i$ and $j$ is affected by an additive error :

$$\tilde{d}_{i,j} = d_{i,j} + e_{i,j}. \tag{4.64}$$

We model the presence of the outlier by a Gaussian mixture. Thus the error $e_{i,j}$ is drawn from the following Gaussian mixture :

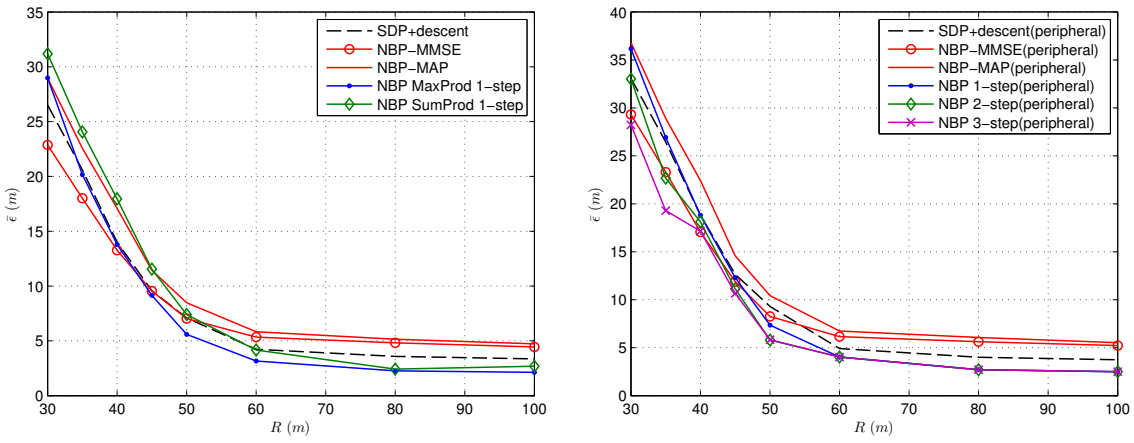
$$e_{i,j} \sim w_1 \mathcal{N}(0, \sigma_1^2) + w_2 \mathcal{N}(e_0, \sigma_2^2), \tag{4.65}$$

where we set $w_1 = 0.75$, $w_2 = 0.25$, $\sigma_1 = \sigma_2 = 0.5m$ and $e_0 = 10m$.

Finally, for NBP-based methods, we use 200 samples, and 5 iterations.

### Localization accuracy

Simulation results are shown in Figure 4.22 where we can notice that the max-product performs better than the sum-product for this scenario. We can also notice that for $R > 45m$, the RMSE of the TP-NBP becomes less than that of the WLS. The fact the NBP is a probabilistic estimator makes it more resilient to the outliers than non-probabilistic estimators such as WLS. Furthermore, we can notice that the TP-NBP is much more accurate than the NBP-MMSE for this scenario, and the plots show an accuracy improvement up to $2m$.




(a) (b)

***Figure 4.22*** — Variation of root mean square error with $R$. (a) All nodes. (b) Peripheral nodes.

## 4.8.3 Impact of the number of iterations

The latency is monotonically increasing with the number of iterations. To asses the effect of the number of iterations on the accuracy, we consider the network of Figure 4.23. The noise in the distance measurements is Gaussian distributed with variance $\sigma = 0.5m$. Figure

4.24 shows the variation of the RMSE with the number of iterations. For the two-phases solution, the same number of iterations is performed during the first and second phase. Node 6 is three hops far from the anchor nodes and starts receiving messages after three iterations, and node 7 is two hops far from the anchor nodes and starts receiving messages after two iterations. At the first iteration of the NBP, node 8 receives messages from two anchor nodes and has a flip ambiguity, but after two iterations of the two-phases NBP (four iteration in total) and three iterations of the classical NBP, the ambiguity is eliminated. It can be noticed that the estimation accuracy does not improve beyond four iterations. We can deduce that the number of iterations needed by the NBP methods is small, and is in general smaller than the network graph diameter (i.e., longest shortest path).

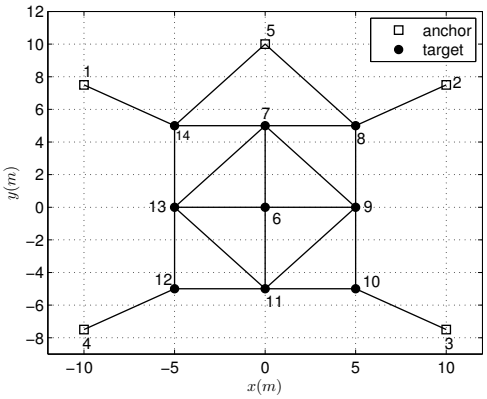



***Figure 4.23*** — Network of 5 anchors nodes and 9 target nodes.

### 4.8.4 Impact of correlated shadowing on flip ambiguity mitigation

In the previous examples, the shadowings on the different links were assumed to be independent. Here, we provide an example showing that the information about the shadowing correlation is beneficial for the position estimation. The example focuses on flip ambiguity mitigation.

We consider the network of Figure 4.25 consisting of one target node located at $(-8.5; 5)$ and four anchor nodes. Anchor nodes 1, 2 and 3 are collinear. Anchor node 2 is located at $(-3.5+d; 0)$. The target node performs power and distance measurements with anchor nodes 1, 2 and 3 and only a power measurement with anchor node 4.

Using only the distance measurements to estimate the target node location results in a flip ambiguity, since the anchor nodes are collinear. Two solutions are feasible. Denote them by $\mathbf{x}_5 = (-8.5; 5)$ and $\mathbf{x}'_5 = (-8.5; -5)$. In this example, the power measurements are used to select one of these two solutions by applying the ML estimator.

The power measurements are given by (4.52) where we assume that the shadowing losses on the different links are correlated according to the model provided in [79]. This model computes the shadowing as the integral of a spatial loss field $p(\mathbf{x})$ :

$$X_{i,j} = \frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^{1/2}} \int_{\mathbf{x}_i}^{\mathbf{x}_j} p(\mathbf{x}) d\mathbf{x}. \tag{4.66}$$

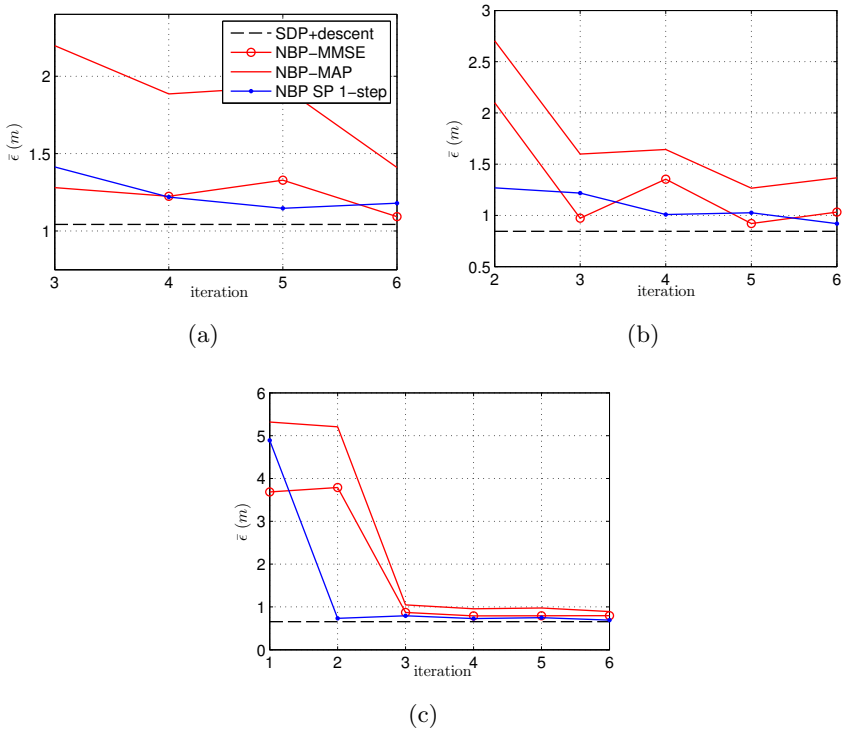



***Figure 4.24*** — Variation of the root mean square error with the iteration number of (a) node 6, (b) node 7 and (c) node 8.

The spatial loss field $p(\mathbf{x})$ is a Gaussian random field of zero mean and exponentially decaying isotropic correlation function :

$$E\left\{p(\mathbf{x}_i)p(\mathbf{x}_j)\right\} = \frac{\sigma_X^2}{\delta}\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|}{\delta}\right). \tag{4.67}$$

We set $n_p = 3$, $\sigma_X = 8dB$, and let $\delta$ take one of the two values 0.4 and $1m$.

We assume that all the pairs of nodes are performing power measurements. Define the shadowing random vector $V = [X_{1,4}\ X_{2,4}\ X_{3,4}\ X_{1,5}\ X_{2,5}\ X_{3,5}\ X_{4,5}]^T$ when the target node is located at $\mathbf{x}_5$, and let $\mathbf{C}$ be its covariance matrix. Similarly, let $V'$ and $\mathbf{C}'$ be the shadowing vector and its rigidity matrix, respectively, for the same links when the target node is located at $\mathbf{x}'_5$.

The shadowing values can be obtained by subtracting the path loss from the power observations. Let $\mathbf{v}$ and $\mathbf{v}'$ be the values computed for $V$ and $V'$, respectively. The ML estimator is processed as follows :

$$\hat{\mathbf{x}}_{ML} = \begin{cases} \mathbf{x}_5 & \text{if } \mathbf{v}^T\mathbf{C}^{-1}\mathbf{v} \leq \mathbf{v}'^T\mathbf{C}'^{-1}\mathbf{v}' \\ \mathbf{x}'_5 & \text{otherwise} \end{cases}. \tag{4.68}$$

Figure 4.26 shows the variation with $d$ (i.e., distance between anchor nodes 1 and 2) of the flip probability. We can notice that the flip probability is decreasing as $d$ tends to $3.5m$. The justification of this result is that when the target node is in the flipped position, the link

connecting nodes 2 and 4 and the link connecting the target node to node 2 overlap with the link connecting the target node to node 4 at $d = 3.5m$, making the shadowing losses more correlated.

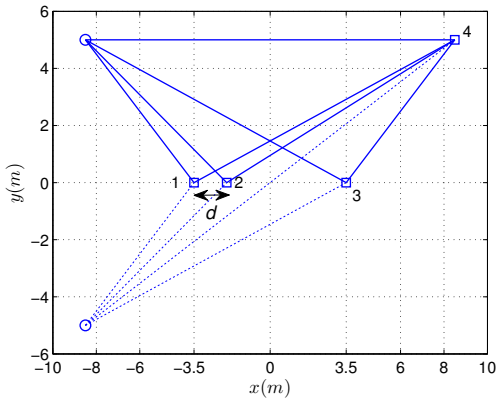



***Figure 4.25*** — Network of one target node and four anchor nodes.

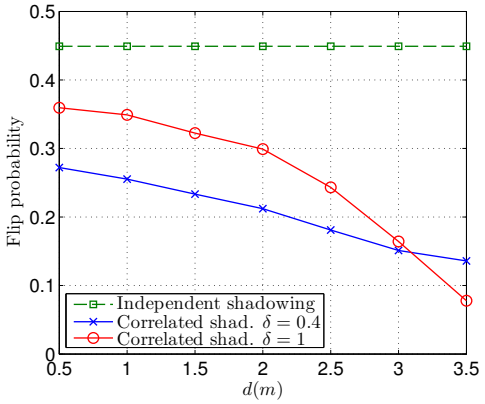



***Figure 4.26*** — Variation with $d$ of the flip probability.

We can deduce from this example that the shadowing correlation information is useful for solving the flips. More accurate site specific propagation models can be employed (e.g., ray tracing). The application of the ML to ambiguity mitigation requires a centralized processing in the case of multiple cooperating target nodes. A two-phases NBP based solution can be applied, where independent shadowing is assumed in the first phase, and the finite sets of potential solutions are sent to a central node for the second phase processing.

## 4.9 Conclusion

In this chapter, we started with an overview of cooperative localization algorithms for static networks. Then we presented the WLS estimator and derived its deterministic stability conditions. The deterministic stability implies the uniqueness of the global optimum, provided the observation error is small enough. Based on this result, we derived the conditions of

consistency if the ML estimator when the ranging measurements are affected by additive Gaussian noises.

Afterwards, we focused on probabilistic estimation and its graphical formalism using Markov random fields. This formalism allows the application of distributed marginalization techniques such as the NBP algorithm. After restating the classical NBP and its application to localization, we developed a new variant of this algorithm, called the TP-NBP, which performs in two phases. The second phase can exploit additional observations with a reduced communication and computational complexity. We showed, via Monte Carlo simulations, that the TP-NBP improves localization accuracy.

The main complexity of the NBP is drawing particles from the beliefs which are products of the incoming messages. It also requires a large amount of data exchange. However, the needed number of iterations is less than that needed by other message passing or successive refinement algorithms.

By finding the modes of the beliefs, potential flip ambiguities can be detected. We used the connectivity information to mitigate the ambiguities in the second phase of the TP-NBP. We also showed by a simulation example that the shadowing correlation information is useful for improving the ambiguity mitigation.

In the next chapter, Bayesian filtering will be considered for tracking a single moving node using RSS measurements. This solution will be implemented using particle filters, which perform a sequential Monte Carlo integration for approximating the marginal probability distribution of the position over time.

CHAPTER 5

# Position Tracking Based on RSS Measurements

## 5.1 Introduction

In the previous chapter, we considered Bayesian probabilistic solutions for cooperative localization, where the nodes were assumed to be static. In this chapter, we consider that mobile terminals (or target nodes) are moving and we aim at tracking their positions over time. As in the previous chapter, Bayesian probabilistic solutions are developed but no direct cooperation between the mobile stations is taken into account.

Our focus is on received signal strength (RSS) measured between a mobile station and several base stations (or anchor nodes). The obstacles in the propagation path cause attenuations in the form of slow fading or shadowing. The shadowing is usually assumed to follow a spatially correlated log-normal distribution.

Two Bayesian tracking solutions are developed in this chapter for efficiently exploiting the measurements and improving the tracking accuracy in presence of random shadowing :

- In the first solution, the shadowing is jointly tracked with the position. For this purpose we define an auto-regressive modeling of the temporal evolution of the shadowing with the displacement of the mobile station.
- In the second solution, shadowing maps are constructed and updated during the on-line tracking phase. This solution allows a reduction of the calibration effort of the fingerprinting method, and the measurements made by several mobile stations moving in the deployment area can be used in the estimation of the maps.

These solutions are studied via Monte Carlo simulations in different deployment scenarios.

The shadowing correlation model used in this chapter differs from the multi-link model presented in [79] and used in sections 3.5.4 and 4.8.4, and is rather a simpler single-link model.

Bayesian tracking consists in sequentially computing the posterior probability distribution of a hidden state vector conditioning on the measurements made over time. As for the computation of the marginal distributions in the previous chapter, the sought probability distributions are untraceable analytically and are approximated using sequential Monte Carlo methods also called particle filters.

This chapter is organized as follows. We start by providing the assumed RSS modeling in Section 5.2. In this section we also describe the shadowing maps modeling and develop the auto-regressive modeling of the shadowing process. In Section 5.3, we study the localization accuracy under perfect knowledge of the shadowing via Monte Carlo simulations. Then, in Section 5.4, Bayesian tracking is introduced and the joint position and shadowing tracking solution is developed, and applied to vehicle tracking in a macro-cellular system. Afterwards, the maps estimation solution is developed and applied to indoor and train tracking in Section 5.5. Finally, concluding remarks are drawn in Section 5.6

## 5.2 RSS measurements and shadowing modeling

In wireless systems, RSS measurements are performed for channel access, power control and handover, etc. Thus, a localization solution does not require any extra effort for obtaining these measurements.

In a propagation environment, there are obstacles causing several propagation effects such as scattering, reflection and diffraction. In such multipath environments, the received power is difficult to predict and can be modeled as a combination of path loss, large scale fading or shadowing and small scale fading [39]. The small scale fading may be filtered out by averaging the signal over a time window or over the used bandwidth or by using multiple antennas [119]. The shadowing can be divided into two parts : A time varying part caused by moving obstacles (e.g., vehicles and persons) and a time invariant one caused by static obstacles (e.g., buildings, tress and hills in outdoor and walls and furniture in indoor). In the following, we call shadowing the time invariant part.

In this chapter, we consider a network deployment consisting of $N_{BS}$ base stations (BS) of known positions [1]. The RSS measurement in decibel ($dB$) made between a mobile station at position $\mathbf{x}$ and the $i^{th}$ base station can be written as

$$y_i = P_i(\mathbf{x}) + \epsilon_i(\mathbf{x}) + e_i \tag{5.1}$$

where $P_i$ is a deterministic function that accounts for the path loss, the radiated power and antenna gains, $\epsilon_i$ is the shadowing which is assumed invariant with time but depends on the position, and $e_i$ is an error gathering time variant shadowing due to moving obstacles and RSS estimation errors (e.g., remaining small scale fading, non-linearity effects at the receiver, quantization noise and receiver orientation).

Path loss models are derived using a combination of analytical and empirical methods. Many such models for different deployment environments are described in [39]. We do not investigate here the path loss model that we assume to be known.

Measurements have shown that the shadowing is random and log-normally distributed (i.e., Gaussian in $dB$). Thus, $\epsilon_i(\mathbf{x})$ is a realization of a zero-mean Gaussian random variable

1. The term 'base station' is used here instead of 'anchor node' and can represent other kinds of devices such as access points or femto BSs.

with variance $\sigma_{sh}^2$. Measurements have also shown that the shadowing is spatially correlated [120], i.e., $E\{\epsilon_i(\mathbf{x}_p)\epsilon_i(\mathbf{x}_q)\} \neq E\{\epsilon_i(\mathbf{x}_p)\}E\{\epsilon_i(\mathbf{x}_q)\}$ where $\mathbf{x}_p$ and $\mathbf{x}_q$ are two arbitrary positions.

The error $e_i$ is assumed to be mainly due to the shadowing of moving obstacles and the remaining small scale fading. Simulations performed in [119] showed that the distribution of the remaining small scale fading is close to Gaussian. For a moving terminal, the variance of this error depends on the used filter which must be sufficiently narrow in bandwidth to remove the multipath fluctuations yet sufficiently wide to track the shadowing. We model $e_i$ by a zero-mean Gaussian random variable of variance $\sigma_e^2$.

Next, we develop two shadowing models : The first one describes the shadowing maps and relates the shadowing value to the spatial position, and the second one describes the temporal evolution of the shadowing for a moving mobile station. These two models will be used in the positioning and tracking solutions developed throughout this chapter.

### 5.2.1 Shadowing maps modeling

The mobile station is assumed to reside in a geographical area $\mathcal{A} \subset \mathbb{R}^2$. The shadowing of a base station over the area $\mathcal{A}$ is a realization of a Gaussian random field (GRF) that we call a shadowing map. A random field is a generalization of a random process to dimensions higher than one and can be represented using basis expansion (Chapter 3 in [71]). The shadowing map of the $i^{th}$ base station, denoted by $\epsilon_i$, can be represented as follows :

$$\epsilon_i(\mathbf{x}) = \sum_{k=1}^{\infty} \alpha_{i,k}\psi_k(\mathbf{x}) \tag{5.2}$$

where $\{\psi_k\}_{k=1}^{\infty}$ is a complete basis defined on $\mathcal{A}$ and $\{\alpha_{i,k}\}_{k=1}^{\infty}$ are Gaussian distributed coefficients. The basis functions are selected and ordered such that the variance of these coefficients is decreasing with $k$ (where higher values of $k$ correspond to basis functions of higher frequencies), and thus, the map can be approximated using a finite number of coefficients :

$$\epsilon_i(\mathbf{x}) \approx \sum_{k=1}^{L_{map}} \alpha_{i,k}\psi_k(\mathbf{x}). \tag{5.3}$$

The vector $\alpha_i = [\alpha_{i,1}, \cdots, \alpha_{i,L_{map}}]^T$ is multivariate Gaussian distributed with mean vector $\mathbf{m}_i$ and covariance matrix $\mathbf{C}_i$. When there is no information about the shadowing, $\mathbf{m}_i$ and $\mathbf{C}_i$ can be computed from the used basis functions and an assumed spatial correlation model. Several basis functions can be used such as the sine and cosine sets, some orthonormal polynomials and orthonormal rectangular functions. Here, we consider the following approximation : The vector $\alpha_i$ consists of the shadowing values at a grid of positions of $\mathcal{A}$. To obtain the shadowing value at an arbitrary position of $\mathcal{A}$, we apply a piecewise constant interpolation (also called nearest-neighbor interpolation) by assigning the same value at the nearest grid point. This interpolation is equivalent to using a basis of rectangular functions.

The correlation of the shadowing at two positions $\mathbf{x}_p$ and $\mathbf{x}_q$ is

$$E\{\epsilon_i(\mathbf{x}_p)\epsilon_i(\mathbf{x}_q)\} = \sigma_{sh}^2\rho(\|\mathbf{x}_p - \mathbf{x}_q\|) \tag{5.4}$$

where $\rho$ is an isotropic correlation function, i.e., depends only on the distance between the two positions. By definition $\rho(0) = 1$. This correlation function is assumed to be known for all the base stations. The exponential correlation function (5.5) is proposed in [120] and fitted to measured data :

$$\rho(\|\mathbf{x}_p - \mathbf{x}_q\|) = e^{-\gamma\|\mathbf{x}_p - \mathbf{x}_q\|} \tag{5.5}$$

where $\gamma = \log(2)/d_{corr}$ and $d_{corr}$ is the correlation distance at which $\rho(d_{corr}) = 1/2$. $d_{corr}$ varies from one environment to another (few hundreds of meters in suburban environments, less than one hundred meters in urban environments and few meters in indoor), and can be estimated by a calibration process.

Figure 5.1 shows two correlation functions corresponding to two values of $d_{corr}$ and a randomly generated shadowing map for each of them. These maps are generated using a two-dimensional finite impulse response filter.

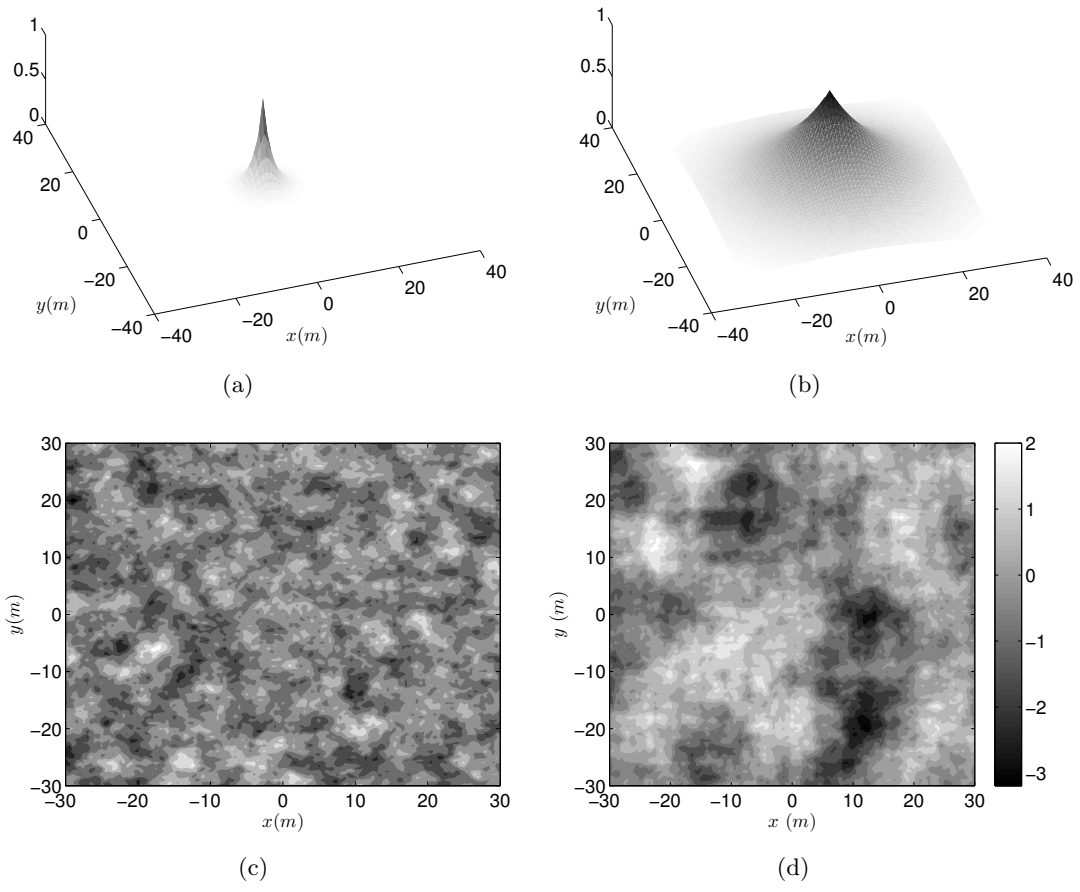



***Figure 5.1*** — (a) Exponential correlation function with $d_{corr} = 1.5m$. (b) Exponential correlation function with $d_{corr} = 10m$. (c) Randomly generated shadowing map with $d_{corr} = 1.5m$ and $\sigma_{sh} = 1dB$. (d) Randomly generated shadowing map with $d_{corr} = 10m$ and $\sigma_{sh} = 1dB$.

Measurements have also shown that shadowings for different base stations are cross-correlated [121] (i.e., $E\{\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})\} \neq E\{\epsilon_i(\mathbf{x})\}E\{\epsilon_j(\mathbf{x})\}$). For the $N_{BS}$ base stations covering

the area $\mathcal{A}$, we define the shadowing atlas as being the collection of the shadowing maps obtained by a concatenation of the vectors :

$$\Lambda = [\alpha_1^T, \cdots, \alpha_{N_{BS}}^T]^T. \tag{5.6}$$

We denote by $L_{atlas} = N_{BS} \times L_{map}$ the length of the atlas vector. We also denote by $\mathbf{M}$ and $\Sigma$ the mean and the covariance of $\Lambda$, respectively. $\Sigma$ is a large $L_{atlas} \times L_{atlas}$ sparse matrix thanks to the low shadowing spatial correlation between distant positions. The RSS measurements made at known positions can be used to update $\mathbf{M}$ and $\Sigma$ as we will see in Section 5.5.1.

For a moving mobile station, the spatial correlation is transformed into a temporal one. Thus, the shadowing evolves with time according to a Gaussian process. Now, we develop an auto-regressive (AR) model for describing this process.

### 5.2.2 Auto-regressive shadowing model

We consider a moving mobile station and we denote by $\mathbf{x}_k = [x_k, y_k]^T$ its position vector at time $kT$, where $k \in \mathbb{N}$ and $T$ is a time step. We start by describing the AR model in the case of a single base station, and then make the generalization for multiple base stations.

#### Shadowing AR model for a single base station

Here, we focus on the case where a single base station lies in the system. For notational brevity, the subscript of the shadowing $\epsilon$ is dropped.

The correlation of the shadowing at times $lT$ and $mT$ is

$$E\{\epsilon(\mathbf{x}_l)\epsilon(\mathbf{x}_m)\} = \sigma_{sh}^2 \rho(\|\mathbf{x}_l - \mathbf{x}_m\|) \tag{5.7}$$

Knowing the positions $\mathbf{x}_{0:k} = [\mathbf{x}_0^T, \cdots, \mathbf{x}_k^T]^T$ and prior to any observation, $[\epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_k)]^T$ is a zero mean Gaussian distributed vector of covariance matrix $\mathbf{R}_k$, with $\mathbf{R}_k(l+1, m+1) = \sigma_{sh}^2 \rho(\|\mathbf{x}_l - \mathbf{x}_m\|)$ and $0 \leq l, m \leq k$. We write $\mathbf{R}_k$ as follows :

$$\mathbf{R}_k = \begin{bmatrix} \mathbf{R}_{k-1} & \mathbf{r}_k \\ \mathbf{r}_k^T & \sigma_{sh}^2 \end{bmatrix} \tag{5.8}$$

where the vector $\mathbf{r}_k = E\{[\epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_{k-1})]^T \epsilon(\mathbf{x}_k)\}$.

Furthermore, $p(\epsilon(\mathbf{x}_k)|\mathbf{x}_{0:k}, \epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_{k-1}))$ being a Gaussian distribution, the process $\epsilon(\mathbf{x}_k)$ can be represented by an order-$k$ AR model $AR(k)$

$$\epsilon(\mathbf{x}_k) = \mathbf{a}_k^T [\epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_{k-1})]^T + \theta_k \tag{5.9}$$

where

$$\mathbf{a}_k = \mathbf{R}_{k-1}^{-1} \mathbf{r}_k \tag{5.10}$$

and $\theta_k$ is a zero mean Gaussian variable of variance $\sigma_{\theta_k}^2$ :

$$\sigma_{\theta_k}^2 = \sigma_{sh}^2 - \mathbf{r}_k^T \mathbf{R}_{k-1}^{-1} \mathbf{r}_k. \tag{5.11}$$

The equations of $\mathbf{a}_k$ and $\sigma_{\theta_k}$ can be derived following the indications provided in Appendix B.

In order to take all the previous states into account, the order of the AR model increases with time. The matrix inversion $\mathbf{R}_k^{-1}$ can be computed recursively without matrix inversion from $\mathbf{R}_{k-1}^{-1}$ and $\mathbf{r}_k$ :

$$\begin{aligned}\mathbf{R}_k^{-1} &= \begin{bmatrix} \mathbf{I}_k & -\mathbf{R}_{k-1}^{-1}\mathbf{r}_k \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{k-1}^{-1} & \mathbf{0} \\ \mathbf{0} & 1/\sigma_{\theta_k}^2 \end{bmatrix} \\ &\quad \times \begin{bmatrix} \mathbf{I}_k & \mathbf{0} \\ -\mathbf{r}_k^T\mathbf{R}_{k-1}^{-1} & 1 \end{bmatrix} \end{aligned} \tag{5.12}$$

where $\mathbf{I}_k$ is the identity matrix of rank $k$.

As a remark, the order of the AR process can be limited to the $p < (k+1)$ previous states, $\mathbf{R}_k^{-1}$ being replaced by the inverse of the $p \times p$ lower right submatrix of $\mathbf{R}_k$ or by the Schur's complement of the $(k-p+1) \times (k-p+1)$ upper left submatrix of $\mathbf{R}_k^{-1}$. Thus, if a sliding window approach of depth $p$ is considered, the Schur's complement is of low complexity as it does not need any matrix inversion.

**Shadowing AR model for multiple base stations**

We denote by $\Omega_k = [\epsilon_1(\mathbf{x}_k), \cdots, \epsilon_{N_{BS}}(\mathbf{x}_k)]^T$ the shadowing vector of the $N_{BS}$ base stations at time $k$. We consider a constant cross-correlation (i.e., $E\{\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})\} = \beta^2 = constant$ for all $\mathbf{x} \in \mathcal{A}$), and we model the shadowing of the $i$-th base station by a weighted sum of two i.i.d. GRFs $GF_i$ and $GF_c$ :

$$\epsilon_i(\mathbf{x}_k) = \alpha GF_i(\mathbf{x}_k) + \beta GF_c(\mathbf{x}_k) \tag{5.13}$$

where $\alpha^2 + \beta^2 = \sigma_{sh}^2$, $GF_c$ is common for all base stations, and the GRFs verify the following equations :

$$\begin{aligned} E\{GF_i(\mathbf{x}_l)GF_j(\mathbf{x}_m)\} &= 0. \\ E\{GF_i(\mathbf{x}_l)GF_c(\mathbf{x}_m)\} &= 0. \\ E\{GF_i(\mathbf{x}_l)GF_i(\mathbf{x}_m)\} &= \rho(\|\mathbf{x}_l - \mathbf{x}_m\|). \\ E\{GF_c(\mathbf{x}_l)GF_c(\mathbf{x}_m)\} &= \rho(\|\mathbf{x}_l - \mathbf{x}_m\|). \end{aligned}$$

Thus, the following equation is verified :

$$E\{\epsilon_i(\mathbf{x}_l)\epsilon_j(\mathbf{x}_m)\} = \beta^2\rho(\|\mathbf{x}_l - \mathbf{x}_m\|).$$

Prior to any observation, $\Omega_{0:k} = [\Omega_0^T, \cdots, \Omega_k^T]^T$ is a zero mean Gaussian distributed vector of covariance matrix $\mathbf{P}_k$ equal to

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{P}_{k-1} & \mathbf{T}_k \\ \mathbf{T}_k^T & \sigma_{sh}^2\mathbf{Z} \end{bmatrix} \tag{5.14}$$

where $\mathbf{T}_k = E\{\Omega_{0:k-1}\Omega_k^T\}$ and $\mathbf{Z}$ is the $N_{BS} \times N_{BS}$ matrix defined according to

$$E\{\Omega_k\Omega_k^T\} = \sigma_{sh}^2 \begin{bmatrix} 1 & \beta^2/\sigma_{sh}^2 & \cdots & \beta^2/\sigma_{sh}^2 \\ \beta^2/\sigma_{sh}^2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \beta^2/\sigma_{sh}^2 \\ \beta^2/\sigma_{sh}^2 & \cdots & \beta^2/\sigma_{sh}^2 & 1 \end{bmatrix} = \sigma_{sh}^2\mathbf{Z}.$$

The Gaussian process $\Omega_k$ can be represented by the $AR(k)$ model

$$\Omega_k = \mathbf{A}_k^T\Omega_{0:k-1} + \Theta_k \tag{5.15}$$

where

$$\mathbf{A}_k = \mathbf{P}_{k-1}^{-1}\mathbf{T}_k \tag{5.16}$$

and $\Theta_k$ is a zero mean Gaussian distributed vector of covariance matrix $\mathbf{Q}_k$. Thus, the Gaussian distribution $p(\Omega_k|\mathbf{x}_{0:k}, \Omega_{0:k-1})$ has a mean $\mathbf{A}_k^T\Omega_{0:k-1}$ and a covariance matrix $\mathbf{Q}_k$.

From (5.8) and (5.14), $\mathbf{P}_k = \mathbf{R}_k \otimes \mathbf{Z}$ where $\otimes$ denotes the Kronecker product [2], which enables computing $\mathbf{P}_k^{-1} = \mathbf{R}_k^{-1} \otimes \mathbf{Z}^{-1}$ [3]. Thus, the matrix $\mathbf{A}_k$ defined in (5.16) can be computed with low complexity [4]

$$\begin{aligned} \mathbf{A}_k &= \left(\mathbf{R}_{k-1}^{-1} \otimes \mathbf{Z}^{-1}\right)(\mathbf{r}_k \otimes \mathbf{Z}) \\ &= \mathbf{a}_k \otimes \mathbf{I}_{N_{BS}}. \end{aligned} \tag{5.17}$$

Similarly, the covariance matrix $\mathbf{Q}_k$ of $\Theta_k$ is given by

$$\mathbf{Q}_k = \sigma_{sh}^2\mathbf{Z} - \mathbf{T}_k^T\mathbf{P}_{k-1}^{-1}\mathbf{P}_k = \sigma_{\theta_k}^2\mathbf{Z}. \tag{5.18}$$

Thus, the transition probabilities of the shadowing process can be computed with low complexity. When the cross-correlation is not constant (i.e., $E\{\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})\} = \beta^2(\mathbf{x})$ ), equations (5.17) and (5.18) are no more applicable, but the inverse of $\mathbf{P}_k$ can be computed recursively as for equation (5.12).

## 5.3 Localization accuracy under known shadowing

In this section, we assess the effect of the knowledge of shadowing maps on static localization accuracy (no motion is considered in this section). For this purpose, we consider the network of Figure 5.2 where four base stations are located at $[-10, -10]^T$, $[10, -10]^T$,

2. If $\mathbf{A}$ is a $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ matrix, then the Kronecker product $\mathbf{A} \otimes \mathbf{B}$ is the $mp \times nq$ matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{1,1}\mathbf{B} & \cdots & a_{1,n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m,1}\mathbf{B} & \cdots & a_{m,n}\mathbf{B} \end{bmatrix}.$$

3. If $\mathbf{A}$ and $\mathbf{B}$ are invertible, then $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$.

4. If $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{D}$ are matrices of such sizes that one can form the matrix products $\mathbf{AC}$ and $\mathbf{BD}$, then $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$.

$[-10, 10]^T$ and $[10, 10]^T$ and one mobile station is located at $\mathbf{x} = [0, 0]^T$, and we adopt the simple log-distance path loss model :

$$y_i = P_0 - 10n_p \log_{10}(d_i) + \epsilon_i(\mathbf{x}) + e_i \tag{5.19}$$

where $d_i = \|\mathbf{x} - \mathbf{x}^{BSi}\|$ is the distance in meters between the $i^{th}$ base station at position $\mathbf{x}^{BSi}$ and the mobile station at position $\mathbf{x}$ and $n_p$ is the path loss exponent that we take equal to 3. We assume that $\epsilon_i(\mathbf{x})$ is known at all positions and for all BSs.

By assuming independent $e_i$ in the RSS measurements, the Fisher information matrix corresponding to the estimation of the unknown coordinates vector $\mathbf{x} = [x, y]^T$ is given by

$$\mathbf{J}(\mathbf{x}) = \frac{1}{\sigma_e^2} \sum_{i=1}^{N_{BS}} \mathbf{v}_i \mathbf{v}_i^T \tag{5.20}$$

where

$$\mathbf{v}_i = \begin{bmatrix} -\frac{10n}{\log(10) d_i^2}(x - x^{BSi}) + \frac{\partial \epsilon_i(\mathbf{x})}{\partial x} \\ -\frac{10n}{\log(10) d_i^2}(y - y^{BSi}) + \frac{\partial \epsilon_i(\mathbf{x})}{\partial y} \end{bmatrix}. \tag{5.21}$$

The regularity condition requires $\epsilon_i(\mathbf{x})$ to be continuously differentiable, and it can be verified by selecting continuously differentiable basis functions $\{\psi_k\}$.

The mean square error (MSE) is lower bounded by the squared position error bound (SPEB), for a given realization of shadowing maps :

$$E\{\|\hat{\mathbf{x}} - \mathbf{x}\|^2\} \geq \text{trace } \mathbf{J}^{-1}(\mathbf{x}) \tag{5.22}$$

where $\hat{\mathbf{x}}$ is an unbiased estimator of $\mathbf{x}$ and the averaging is made over measurements error realizations.

As a shadowing map is a realization of a Gaussian random field, we compute the average SPEB :

$$\text{average SPEB} = E\{\text{trace } \mathbf{J}^{-1}(\mathbf{x})\} \tag{5.23}$$

where the averaging is made over shadowing maps realizations. The partial derivatives in vector $\mathbf{v}_i$ can be computed numerically, i.e., $\frac{\partial \epsilon_i(\mathbf{x})}{\partial x} \approx \frac{\epsilon_i(x+\delta, y) - \epsilon_i(x-\delta, y)}{2\delta}$. In fact, this approximation depends on the value of $\delta$ as can be deduced by computing its variance :

$$\begin{aligned} E\left\{ \left( \frac{\epsilon_i(x+\delta, y) - \epsilon_i(x-\delta, y)}{2\delta} \right)^2 \right\} &= \frac{\sigma_{sh}^2 (1 - \rho(2\delta))}{2\delta^2} \\ &\approx \frac{\log(2) \sigma_{sh}^2}{\delta d_{corr}}. \end{aligned} \tag{5.24}$$

and

$$\lim_{\delta \to 0} E\left\{ \left( \frac{\epsilon_i(x+\delta, y) - \epsilon_i(x-\delta, y)}{2\delta} \right)^2 \right\} = \infty. \tag{5.25}$$

Thus, the average SPEB depends on the selected value for $\delta$. We arbitrarily set $\delta = 0.5m$.

We recall that the mobile station position is $\mathbf{x} = [0, 0]^T$. Figure 5.3(a) shows the average SPEB as a function of $d_{corr}$ for various sets of values of $\sigma_e$ and $\sigma_{sh}$, where no cross-correlation

is assumed (i.e., $\beta = 0$) and the error variance $\sigma_e^2 = E\{e_i^2\}$ is the same for all the measurements. We can notice in this figure that the average SPEB for $\sigma_e^2 = 4dB^2$ and $\sigma_{sh}^2 = 16dB^2$ tends to the SPEB for $\sigma_e^2 = 4dB^2$ and $\sigma_{sh}^2 = 0dB^2$ with $d_{corr}$ since the shadowing becomes constant and the variance given by equation (5.24) tends to zero.

Let $\mathbf{f}(\mathbf{x})$ be the following vector function :

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} P_0 - 10n_p \log_{10}(d_1) + \epsilon_1(\mathbf{x}) \\ \vdots \\ P_0 - 10n_p \log_{10}(d_{N_{BS}}) + \epsilon_{N_{BS}}(\mathbf{x}) \end{bmatrix}. \tag{5.26}$$

The RSS measurements vector can be written as $\mathbf{y} = \mathbf{f}(\mathbf{x}) + \mathbf{e}$ where $\mathbf{e} = [e_1, \cdots, e_{N_{BS}}]^T$ is the error vector. The maximum likelihood (ML) estimator corresponds to

$$\hat{\mathbf{x}}_{ML} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{f}(\mathbf{x})\|^2. \tag{5.27}$$

Figure 5.3(b) shows the MSE of the ML estimator where we can notice that the error is much higher than the average SPEB and is decreasing with $d_{corr}$. This high error can be justified as follows. In the scenario considered here, the mobile station is located at $[0, 0]^T$. Define the scalar function $g(\mathbf{x})$ :

$$g(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{0})\|. \tag{5.28}$$

Figure 5.4 shows the positions (black points) corresponding to the 200 smallest values of $g$ where the area is discretized with a separation step of $0.5m$ between two nearest grid points and random shadowing maps are generated for each value of $d_{corr}$. We can notice that for $d_{corr} = 1m$, the black points are spread in the area, and as a consequence, small measurement errors can result in high localization errors, and for $d_{corr} = 10m$, the black points are near each other making the MSE smaller.

The large difference between the SPEB and the MSE can be explained by the fact that the derivatives in the Fisher matrix $\mathbf{J}(\mathbf{x})$ are computed using shadowing values at nearby points and the SPEB can only capture local information. Other lower bounds can be considered in this case, such as the Barankin bound [122], but these bounds are restricted to unbiased estimators and can be higher than the MSE of biased estimators.

The cumulative distribution function (CDF) plot of the localization error for $d_{corr} = 2m$ and $\sigma_e = 1dB$ is shown in Figure 5.5 where we can notice that the shadowing extends the distribution of the shadowing to higher values which results in higher MSE.

Figure 5.6 shows the MSE when the mobile station is known to lie inside the square of size $6m \times 6m$. The computed MSE in the presence of shadowing (i.e., $\sigma_{sh}^2 = 16dB^2$) is smaller than the MSE in the absence of shadowing (i.e., $\sigma_{sh}^2 = 0dB^2$) for this scenario, and this result is consistent with the result of Figure 5.5 : This a priori information eliminates the points that can result in high localization errors. At high $d_{corr}$ values, the shadowing becomes nearly constant, and the MSE increases and tends to the MSE of the case of shadowing absence.

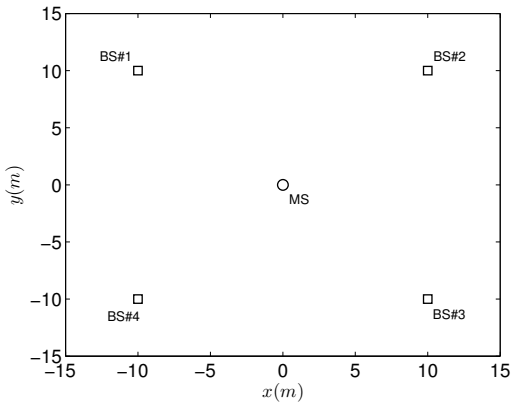
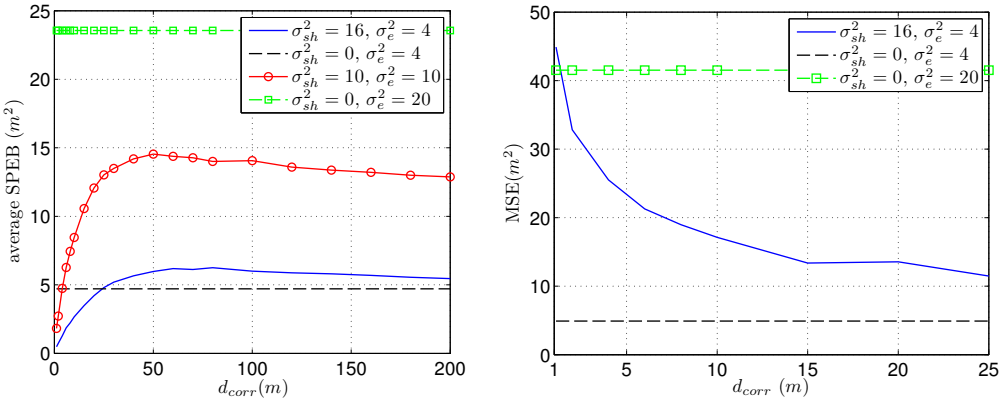



***Figure 5.2*** — Network of four base stations and one mobile station.




(a) (b)

***Figure 5.3*** — (a) Variation with $d_{corr}$ of the average SPEB for $\delta = 0.5m$. (b) Variation with $d_{corr}$ of the MSE of the ML position estimate.

## 5.4 Position tracking

In this section, we treat the problem of tracking over time the position of a moving mobile station. The position tracking relies on the measurements obtained from the radio signals or the outputs of an inertial navigation system (INS) and a motion model. The motion model describes the characteristics of the motion (e.g., a pedestrian in indoor cannot pass through walls, a vehicle on a road has a limited maximum speed), and can account for the kinematic rules. This model also describes the dependence between the positions at different time instants allowing to improve the estimation accuracy by using several measurements made over time.

More specifically, we apply Bayesian tracking which efficiently exploits the incoming measurements by recursively updating the posterior probability distribution. We consider two scenarios : In the first one, the shadowing maps are known, and in the second one, the shadowing maps are not known and the shadowing is considered as a part of the hidden state vector where a joint position and shadowing tracking solution is developed.

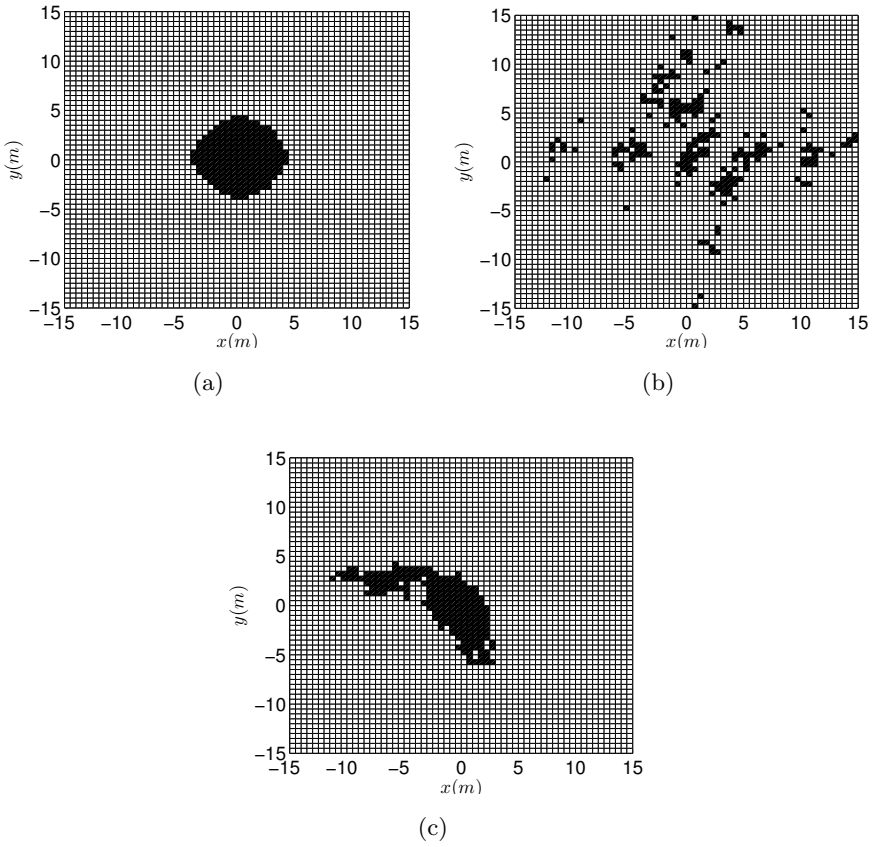



***Figure 5.4*** — Grid positions resulting in the 200 smallest values of $g(\mathbf{x})$ (5.28) : (a) No shadowing ; (b) $d_{corr} = 1m$ ; (c) $d_{corr} = 10m$.

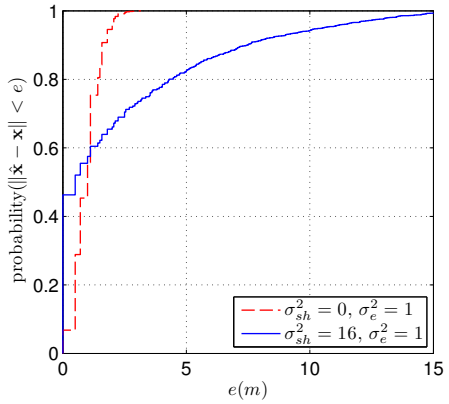
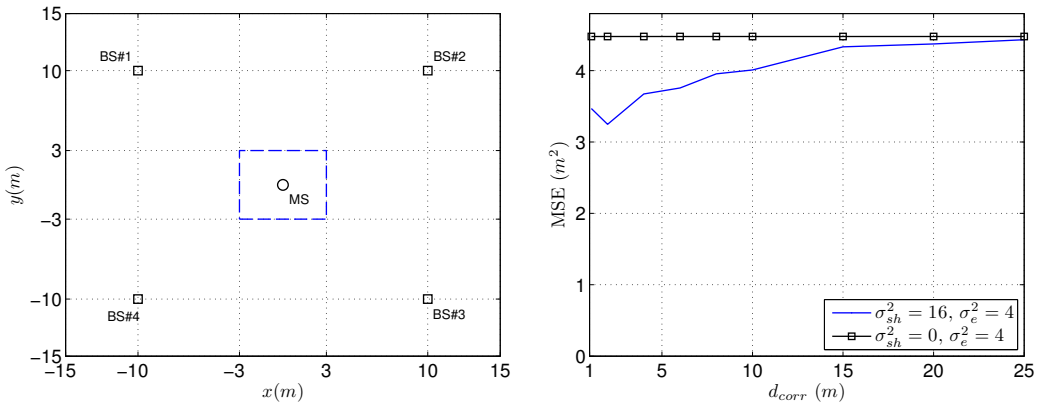



***Figure 5.5*** — CDF of the error for $d_{corr} = 2m$ and $\sigma_e = 1dB$.




(a) (b)

***Figure 5.6*** — (a) Constraint on the position of the mobile station. (b) Variation of the MSE of the ML estimate with $d_{corr}$.

### 5.4.1 Tracking under known shadowing

Here, we assume that the shadowing maps of the $N_{BS}$ base stations in the deployment area $\mathcal{A}$ are known. We start by describing the transition model of the hidden state vector and the observation model. They enable the computation of the a priori and likelihood probabilities in the Bayesian tracking processing, respectively.

#### Transition model

We define the kinematic vector $\mathbf{c}_k$ at time $kT$ comprising the mobile station position $\mathbf{x}_k$ and possibly other kinematic parameters (e.g., velocity). This vector is issued from a known Markov process of transition probability $p(\mathbf{c}_k|\mathbf{c}_{k-1})$ and initial distribution $p(\mathbf{c}_0)$. The hidden state vector $\mathbf{s}_k$ is defined by

$$\mathbf{s}_k = \mathbf{c}_k. \tag{5.29}$$

The transition model is described by the distributions $p(\mathbf{s}_k|\mathbf{s}_{0:k-1}) = p(\mathbf{s}_k|\mathbf{s}_{k-1})$ and $p(\mathbf{s}_0)$. A simple linear transition model in which the state consists only of the position $\mathbf{x}_k$ can be written as

$$\mathbf{c}_k = \mathbf{c}_{k-1} + l_k[\cos\theta_k, \sin\theta_k]^T \tag{5.30}$$

where $l_k$ is a random non-negative real value of known distribution (e.g. uniform distribution or truncated Gaussian distribution) and $\theta_k$ is a random direction of known distribution (e.g., uniform distribution or von Mises distribution).

Another linear model that accounts for the position and the velocity can be written as

$$\mathbf{c}_k = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{c}_{k-1} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ T & 0 \\ 0 & T \end{bmatrix} (\mathbf{a}_{k-1} + \mathbf{q}_k) \tag{5.31}$$

where $\mathbf{c}_k = [\mathbf{x}_k^T, \dot{\mathbf{x}}_k^T]^T$ and $\dot{\mathbf{x}}_k$ is the derivative of $\mathbf{x}_k$ with respect to the time, $\mathbf{a}_{k-1} = [a_{x,k-1}, a_{y,k-1}]^T$ is the acceleration vector provided by an accelerometer and $\mathbf{q}_k$ accounts for the acceleration estimation error. If the acceleration is not available, $\mathbf{a}_{k-1}$ is simply discarded and the variance of $\mathbf{q}_k$ is adjusted accordingly.

### Observation model

At time $kT$, the mobile station makes RSS measurements $\mathbf{y}_k$ with a subset of the $N_{BS}$ base stations. The measurements vector is equal to

$$\mathbf{y}_k = \mathbf{f}_k(\mathbf{x}_k) + \mathbf{e}_k \tag{5.32}$$

where $\mathbf{f}_k$ is a known deterministic function and $\mathbf{e}_k$ is a Gaussian distributed error that we assume white with respect to the time domain. This assumption makes the observations at different time instants independent given the states, i.e., $p(\mathbf{y}_i, \mathbf{y}_j|\mathbf{s}_i, \mathbf{s}_j) = p(\mathbf{y}_i|\mathbf{s}_i)p(\mathbf{y}_j|\mathbf{s}_j)$.

### Bayesian tracking

The aim of Bayesian tracking is to compute recursively over time the posterior distribution $p(\mathbf{s}_k|\mathbf{y}_{1:k})$, in order to apply a Bayesian estimator [123] such as the MMSE estimator :

$$\hat{\mathbf{s}}_k = \int \mathbf{s}_k p(\mathbf{s}_k|\mathbf{y}_{1:k}) \mathrm{d}\mathbf{s}_k. \tag{5.33}$$

$p(\mathbf{s}_k|\mathbf{y}_{1:k})$ is computed according to Bayes' rule :

$$\begin{aligned} p(\mathbf{s}_k|\mathbf{y}_{1:k}) &= \frac{p(\mathbf{y}_k|\mathbf{s}_k, \mathbf{y}_{1:k-1})p(\mathbf{s}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \\ &\propto p(\mathbf{y}_k|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{y}_{1:k-1}). \end{aligned} \tag{5.34}$$

The predictive distribution $p(\mathbf{s}_k|\mathbf{y}_{1:k-1})$ is obtained according to the following marginalization :

$$\begin{aligned} p(\mathbf{s}_k|\mathbf{y}_{1:k-1}) &= \int p(\mathbf{s}_k, \mathbf{s}_{k-1}|\mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{s}_{k-1} \\ &= \int p(\mathbf{s}_k|\mathbf{s}_{k-1}) p(\mathbf{s}_{k-1}|\mathbf{y}_{1:k-1}) \mathrm{d}\mathbf{s}_{k-1}. \end{aligned} \tag{5.35}$$

This recursive computation is called Bayesian filtering. Equations (5.34) and (5.35) are analytically untraceable since the observation equation (5.32) is non-linear with respect to $\mathbf{s}_k$. A solution based on particle filters [123, 124], which uses sequential Monte Carlo methods for approximating numerically the posterior densities, is presented in the following.

### Implementation using particle filters

The distribution $p(\mathbf{s}_k|\mathbf{y}_{1:k})$ can be obtained by a marginalization of $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$. If we are able to draw $N$ i.i.d. samples $\{\mathbf{s}_{0:k}^i\}_{i=1}^N$ from $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$, we can approximate it by

$$p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) \approx \frac{1}{N} \sum_i \delta(\mathbf{s}_{0:k}^i, \mathbf{s}_{0:k}) \tag{5.36}$$

where $\delta$ is the Kronecker delta function, and one obtains the following approximation of the MMSE estimate $\hat{\mathbf{s}}_k$ computed according to (5.33) :

$$\begin{aligned}\bar{\mathbf{s}}_k &= \int \mathbf{s}_k \frac{1}{N}\sum_i \delta(\mathbf{s}^i_{0:k}, \mathbf{s}_{0:k})\mathrm{d}\mathbf{s}_{0:k} \\ &= \frac{1}{N}\sum_i \mathbf{s}^i_k. \end{aligned} \tag{5.37}$$

This quantity converges to $\hat{\mathbf{s}}_k$ as $N \to \infty$. Sometimes it is difficult to sample the posterior ditribution. An alternative is to use importance sampling where samples are drawn from another distribution $\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$ verifying $\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) > 0$ when $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) > 0$.

We can write

$$\hat{\mathbf{s}}_k = \int \mathbf{s}_k \frac{p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})}{\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})}\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})\mathrm{d}\mathbf{s}_{0:k}. \tag{5.38}$$

This latter can be approximated by

$$\bar{\mathbf{s}}_k = \frac{1}{N}\sum_i w^{*i}_k \mathbf{s}^i_k \tag{5.39}$$

where $\{\mathbf{s}^i_{0:k}\}_{i=1}^N$ are i.i.d. samples drawn from $\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$, and the importance weight $w^{*i}_k$ is equal to

$$w^{*i}_k = \frac{p(\mathbf{s}^i_{0:k}|\mathbf{y}_{1:k})}{\pi(\mathbf{s}^i_{0:k}|\mathbf{y}_{1:k})} = \frac{p(\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k})}{p(\mathbf{y}_{1:k})\pi(\mathbf{s}^i_{0:k}|\mathbf{y}_{1:k})} = \frac{w^i_k}{p(\mathbf{y}_{1:k})}. \tag{5.40}$$

The value $p(\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k})$ can be computed according to

$$p(\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k}) = p(\mathbf{s}^i_0)\prod_{l=1}^{k} p(\mathbf{y}_l|\mathbf{s}^i_l)p(\mathbf{s}^i_l|\mathbf{s}^i_{l-1}) \tag{5.41}$$

and the value $p(\mathbf{y}_{1:k})$ can be approximated by

$$\begin{aligned} p(\mathbf{y}_{1:k}) &= \int p(\mathbf{s}_{0:k}, \mathbf{y}_{1:k})\mathrm{d}\mathbf{s}_{0:k} \\ &= \int \frac{p(\mathbf{s}_{0:k}, \mathbf{y}_{1:k})}{\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})}\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})\mathrm{d}\mathbf{s}_{0:k} \\ &\approx \frac{1}{N}\sum_i \frac{p(\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k})}{\pi(\mathbf{s}^i_{0:k}|\mathbf{y}_{1:k})} \\ &= \frac{1}{N}\sum_i w^i_k, \end{aligned} \tag{5.42}$$

and thus, we can replace $\frac{1}{N}w^{*i}_k$ by the normalized weight $\tilde{w}^i_k = \frac{w^i_k}{\sum w^i_k}$

**Sequential importance sampling** In the position tracking applications, measurements arrive sequentially in time and we are interested in making real time estimations. In this case, it is impractical to draw samples of the whole history of states $\mathbf{s}_{0:k}$ at each time step $kT$ and reject the past simulated trajectories $\{\mathbf{s}^i_{0:k-1}\}$. For this reason, we restrict ourselves to importance functions of the form

$$\pi(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) = \pi(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1})\pi(\mathbf{s}_k|\mathbf{s}_{0:k-1}, \mathbf{y}_{1:k}), \tag{5.43}$$

and at time $kT$, the trajectory $\{\mathbf{s}^i_{0:k-1}\}$ is augmented by the sample $\mathbf{s}^i_k$ drawn from $\pi(\mathbf{s}_k|\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k})$. Furthermore, using importance functions of this form enables a recursive evaluation in time of the importance weights $\{w^i_k\}$ as we will show now.

At time $kT$, we can write

$$\begin{aligned} p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) &= \frac{p(\mathbf{s}_{0:k}, \mathbf{y}_k|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})} \\ &= \frac{p(\mathbf{y}_k|\mathbf{s}_k)p(\mathbf{s}_k|\mathbf{s}_{k-1})p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}. \end{aligned} \tag{5.44}$$

Assume that the approximation $p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1}) \approx \sum_i \tilde{w}^i_{k-1}\delta(\mathbf{s}^i_{0:k-1}, \mathbf{s}_{0:k-1})$ is available at time $(k-1)T$. Thus, according to (5.44), we can write

$$p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) \approx \frac{\sum_i \tilde{w}^i_{k-1}p(\mathbf{s}_k|\mathbf{s}^i_{k-1})p(\mathbf{y}_k|\mathbf{s}_k)\delta(\mathbf{s}^i_{0:k-1}, \mathbf{s}_{0:k-1})}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}. \tag{5.45}$$

For the $i^{th}$ trajectory, we draw $\mathbf{s}^i_k \sim \pi(\mathbf{s}_k|\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k})$ and we update the weights according to :

$$\begin{aligned} w^i_k &= \frac{p(\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k})}{\pi(\mathbf{s}^i_{0:k}|\mathbf{y}_{1:k})} \\ &= \frac{p(\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k-1})p(\mathbf{s}^i_k|\mathbf{s}^i_{k-1})p(\mathbf{y}_k|\mathbf{s}^i_k)}{\pi(\mathbf{s}^i_{0:k-1}|\mathbf{y}_{1:k-1})\pi(\mathbf{s}^i_k|\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k})} \\ &\propto \tilde{w}^i_{k-1}\frac{p(\mathbf{s}^i_k|\mathbf{s}^i_{k-1})p(\mathbf{y}_k|\mathbf{s}^i_k)}{\pi(\mathbf{s}^i_k|\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k})}. \end{aligned} \tag{5.46}$$

Finally, we compute $\tilde{w}^i_k = w^i_k / \sum w^i_k$.

A simple choice of the importance function is the prior distribution of the Markov transition model $\pi(\mathbf{s}_k|\mathbf{s}_{0:k-1}, \mathbf{y}_{1:k-1}) = p(\mathbf{s}_k|\mathbf{s}_{k-1})$ which results in $w^i_k \propto \tilde{w}^i_{k-1}p(\mathbf{y}_k|\mathbf{s}^i_k)$. By this choice, we don't need to save the history of states and $\{\mathbf{s}^i_k, \tilde{w}^i_k\}$ are obtained from $\{\mathbf{s}^i_{k-1}, \tilde{w}^i_{k-1}\}$.

**Sampling importance resampling (SIR)** The variance of the importance weights increases with time [123], and after several time steps, all the normalized weights tend to zero except one weight which tends to one. This phenomenon is called degeneracy of the particle filter. Degeneracy cannot be avoided by the choice of the importance function, but this choice can reduce its speed by reducing the increase of the variance at each time step. To resolve this problem, resampling has been introduced in [125]. Its basic idea is to eliminate the trajectories with weak weights and repeat the trajectories with strong weights. The effective number of samples can be approximated by $\hat{N}_{eff} = 1/\sum(\tilde{w}^i_k)^2$. It is possible to perform resampling at each time step or when $\hat{N}_{eff}$ goes below a predefined threshold $N_{th} = \gamma N$. Several resampling algorithms are presented in [18]. Sampling importance resampling (SIR) particle filter is summarized in Table 5.1 below and is illustrated by Figure 5.7.

Other variants of the particle filter have been reported in [123]. They differ in the choice of the importance function (e.g., auxiliary particle filter and optimal importance sampling). This choice can affect the required number of particles $N$ for obtaining a given approximation accuracy.

0 : SIR particle filter
1 : **for** $i = 1, \cdots, N$
2 : **sample** $\mathbf{s}_k^i \sim p(\mathbf{s}_k|\mathbf{s}_{k-1}^i)$
3 : **evaluate** the importance weights $w_k^i \propto \tilde{w}_{k-1}^i p(\mathbf{y}_k|\mathbf{s}_k^i)$
4 : **for** $i = 1, \cdots, N$
5 : **normalize** the importance weights $\tilde{w}_k^i = w_k^i / \sum w_k^i$
6 : **evaluate** $\hat{N}_{eff} = 1/\sum(\tilde{w}_k^i)^2$
7 : **if** $\hat{N}_{eff} < N_{th}$
8 : **set** $\tilde{\mathbf{s}}_k^i = \mathbf{s}_k^i$
8 : **for** $i = 1, \cdots, N$
9 : **sample** an index $j(i)$ distributed according to the discrete distribution with $N$ elements satisfying $\Pr\{j(i) = l\} = \tilde{w}_k^l$
11 : **set** $\mathbf{s}_k^i = \tilde{\mathbf{s}}_k^{j(i)}$ and $\tilde{w}_k^i = 1/N$

***Table 5.1*** — Samples generation and weights update using SIR particle filter.

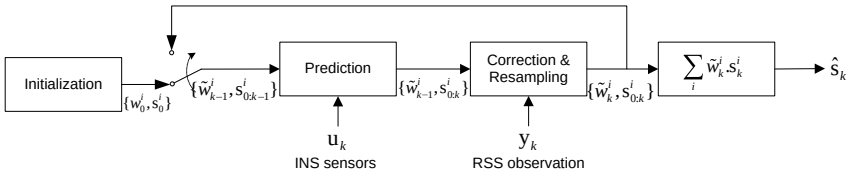


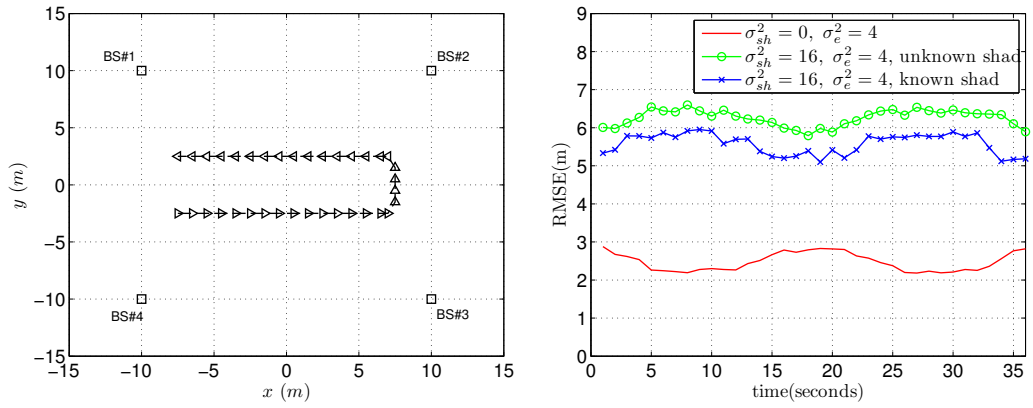

***Figure 5.7*** — Particle filter for estimating the state $\mathbf{s}_k$.

**Numerical results**

We perform Monte Carlo simulations to compute the RMSE in the position tracking solution. We consider the network deployment of Figure 5.8(a) consisting of four base stations, and the mobile station moves along the U-shaped trajectory. The speed is fixed to $1m/s$ and the total trajectory duration is 36 seconds. The path loss model is the log-distance model given by equation (5.19) with $n_p = 3$. The shadowing is exponentially correlated with $d_{corr} = 2m$ and the cross-correlation is equal to zero (i.e., $\beta = 0$). The considered motion model is the one given by (5.30) with $l_k$ drawn from the uniform distribution $\mathcal{U}(0, 2)$ and $\theta_k$ drawn from $\mathcal{U}(0, 2\pi)$. The time step $T$ is set equal to 1 second. The mobile station makes measurements with the four base stations at every time instant and the measurement error is Gaussian distributed with $E\{\mathbf{e}_k\mathbf{e}_k^T\} = \sigma_e^2\mathbf{I}_4$.

Figure 5.8(b) shows the RMSE of the estimated position where no-tracking is performed and positions at different instants are estimated independently using the ML estimator.

In Figure 5.9(a), the position RMSE is plotted vs time where the position is tracked using the SIR particle filter with $N$ particles. When the shadowing is not known, it is treated as a random white variable in space by the particle filter. We can notice that in the absence of shadowing (i.e., $\sigma_{sh}^2 = 0dB^2$ and $\sigma_e^2 = 4dB^2$) and when the shadowing is unknown, the RMSE is reduced compared to the non-tracking ML estimation. We can also notice that increasing of the number of particles from 200 to 1000 does not significantly improve the accuracy.




(a) (b)

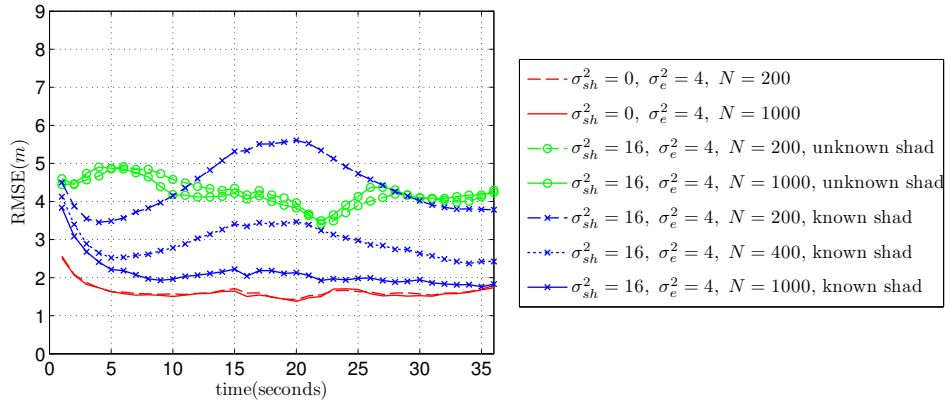
***Figure 5.8*** — (a) Network of four base stations and mobile station trajectory. (b) RMSE of the ML positions estimates.

On the other hand, for the case of known shadowing and $\sigma^2_{sh} = 16dB^2$ and $\sigma^2_e = 4dB^2$, the RMSE for $N = 200$ particles increases compared to the non-tracking RMSE. The justification of this result is that, as we pointed in Section 5.3, positions that are far from the true position might have high likelihoods and positions that are near the true position might have low likelihoods. Thus, particles that are near the true position might have small weights and be eliminated after resampling, resulting in a divergence of the particle filter. Increasing the number of particles to 1000 allows us to remedy this problem but at the expense of an increased complexity.

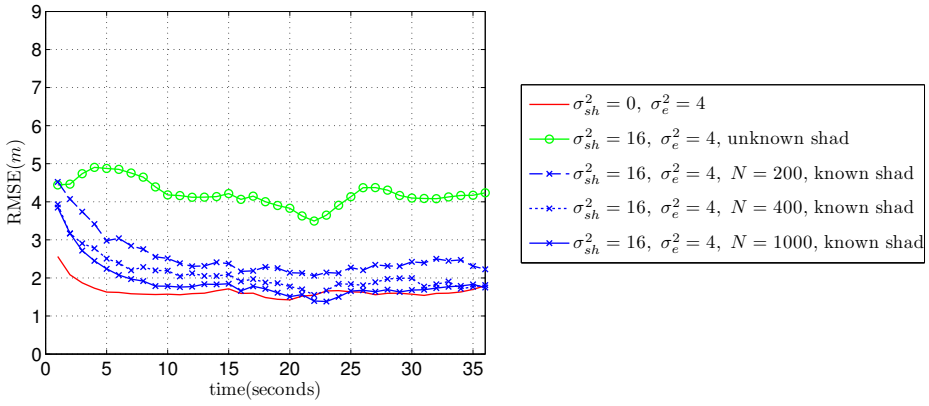
Another solution we consider is the regularization of the particle filter. It consists in replacing 50 particles by 50 grid positions in the deployment area, and perturbing the particles by a Gaussian noise, , that is replace $\mathbf{s}^i_k$ by $\mathbf{s}^i_k + \mathbf{b}^i$ with $\mathbf{b}^i \sim \mathcal{N}(0,\ 0.25\mathbf{I}_2)$, each time a resampling is performed. The resulting RMSE is shown in Figure 5.9(b). By comparing the plots of Figures 5.9(a) and 5.9(b), we can notice a high decrease of the position RMSE for $N = 200$ and $N = 400$ particles when the regularization step is performed. In fact, this step enables the particle filter to catch up the trajectory and thus avoid divergence. The RMSE for $N = 400$ particles becomes close to the RMSE for $N = 1000$ particles allowing a complexity reduction without significant accuracy losses.

By comparing the plots of Figures 5.8(b) 5.9(b), we can observe that the knowledge of the shadowing brings more accuracy improvement in the case of position tracking applications. In position tracking, measurements made at previous time instants are exploited, and this operation is enabled by the motion model.

We also compute the RMSE in the presence of the map constraints of Figure 5.10(a), where the mobile station is known to move inside the U-shaped corridor. These constraints allow a reduction of the number of particles needed to achieve a good accuracy, as shown in Figure 5.10(b) where we set $N = 200$ particles.

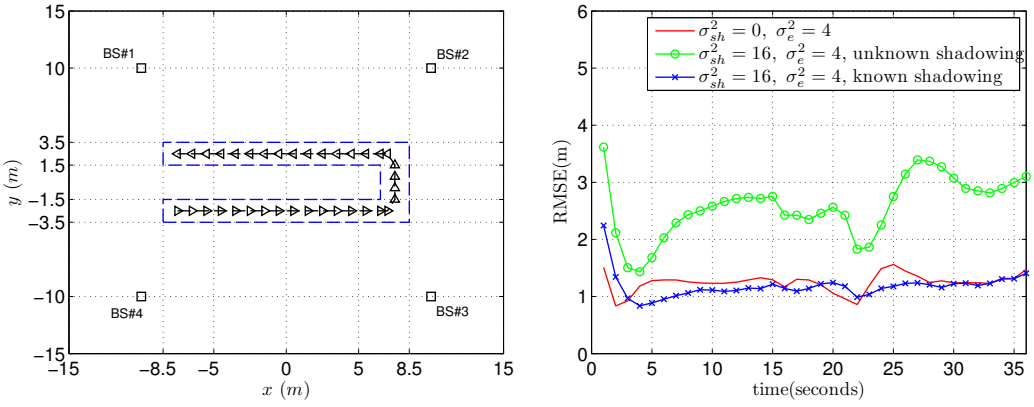



(a)




(b)

***Figure 5.9*** — Variation with time of the RMSE of the SIR particle filters estimates for several values of $N$ : (a) No regularization ; (b) A regularization step is performed after resampling.




(a) (b)

***Figure 5.10*** — (a) Constraints on the mobile station position. (b) Corresponding RMSE with $N = 200$ particles.

### 5.4.2 Joint position and shadowing tracking

Now, we assume that the shadowing is unknown but its probability distribution and spatial correlation function are known. Several algorithms have been proposed in order to improve the position tracking in presence of random shadowing. In [126], a prediction of the shadowing, modeled by a first order auto-regressive Gaussian process, is used along with the RSS measurements for estimating position. This solution is sub-optimal, and can be improved by using a probabilistic approach.

Thus, our solution is based on Bayesian filtering, which efficiently exploits the incoming measurements by recursively updating the posterior probability distribution. The update is performed by taking the shadowing as a part of the state vector, whose stochastic process is no more Markovian. The transition equation of this process is derived from the auto-regressive model defined in Section 5.2.2.

As a remark, an alternative approach was used in a general context in [127], where the shadowing was considered as a measurement noise and the temporal correlation was taken into consideration for evaluating the likelihood function.

The probability density functions, which are analytically untraceable because of the non-linearity of the AR model and the RSS measurements with respect to the position, are estimated using particle filters. More specifically, a Rao-Blackwellized particle filter [123] will be implemented where the part of the state vector consisting of the position and its derivatives is represented by particles and the shadowing part is tracked by means of a Kalman filter. This solution has the advantage of reducing the required number of particles.

Before developing the tracking solution, we start by describing the transition model of the hidden state vector and the observation model.

#### Transition model

The shadowing vector $\Omega_k = [\epsilon_1(\mathbf{x}_k), \cdots, \epsilon_{N_{BS}}(\mathbf{x}_k)]^T$ is taken as a part of the state vector $\mathbf{s}_k$ which is defined by

$$\mathbf{s}_k = [\mathbf{c}_k^T \ , \ \Omega_k^T]^T. \tag{5.47}$$

The state process is no more Markovian because of the shadowing correlation, at it can be decomposed as

$$p(\mathbf{s}_k|\mathbf{s}_{0:k-1}) = p(\mathbf{c}_k|\mathbf{c}_{k-1})p(\Omega_k|\mathbf{x}_{0:k}, \Omega_{0:k-1}). \tag{5.48}$$

The shadowing vector $\Omega_k$ is represented by the AR model, and the mean and covariance of the Gaussian distribution $p(\Omega_k|\mathbf{x}_{0:k}, \Omega_{0:k-1})$ are given by (5.15) and (5.18), respectively.

#### Observation model

At time $kT$, the mobile station makes RSS measurements $\mathbf{y}_k$ with $n_k$ base stations. The measurements vector is equal to

$$\mathbf{y}_k = \mathbf{P}_k(\mathbf{x}_k) + \mathbf{J}_k\Omega_k + \mathbf{e}_k \tag{5.49}$$

where $\mathbf{P}_k$ is a known deterministic vector function, $\mathbf{J}_k$ is an $n_k \times N_{BS}$ matrix where $\mathbf{J}_k(i,j) = 1$ if the $i$-th measurement is made on the $j$-th base station and 0 elsewhere, and $\mathbf{e}_k$ is a Gaussian error considered to be white with respect to the time domain. This assumption and the fact that the shadowing is a part of the state make the observations at different time instants independent given the states, i.e., $p(\mathbf{y}_i, \mathbf{y}_j|\mathbf{s}_i, \mathbf{s}_j) = p(\mathbf{y}_i|\mathbf{s}_i)p(\mathbf{y}_j|\mathbf{s}_j)$.

### Bayesian tracking

A Bayesian filtering consists in determining recursively over time the posterior distribution $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$ in order to apply a Bayesian estimator. This distribution is computed according to

$$p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) = p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1})\frac{p(\mathbf{s}_k|\mathbf{s}_{0:k-1})p(\mathbf{y}_k|\mathbf{s}_k)}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}. \tag{5.50}$$

When an order-$p$ AR process is considered, $p(\mathbf{s}_k|\mathbf{s}_{0:k-1})$ is replaced by $p(\mathbf{s}_k|\mathbf{s}_{k-p:k-1})$.

### Implementation using a Rao-Blackwellized particle filter

The shadowing evolution equation (5.15) and the observation equation (5.49) are non-linear with respect to $\mathbf{x}_k$ making the posterior density (5.50) untraceable analytically. Particle filters enable computing numerically the solutions, with a complexity drawback : The number of particles usually must increase exponentially with the dimension of the state vector. In our case, the dimension of the state vector is high due to the shadowing components.

We can remark that, conditionally on the knowledge of $\mathbf{x}_{0:k}$, equations (5.15) and (5.49) are linear with respect to $\Omega_k$, which can be tracked using a Kalman filter. Thus, we can reduce the number of particles and limit the complexity of the filter by applying a Rao-Blackwellization which consists in computing the posterior of a subset of the state vector analytically, in our case with the Kalman filter.

At time $(k-1)T$, assume that $p(\mathbf{c}_{0:k-1}|\mathbf{y}_{1:k-1})$ is approximated by the set of weighted particles $\left\{\mathbf{c}^i_{0:k-1}, w^i_{k-1}\right\}_{i=1}^N$ and that the Gaussian distribution function $p(\Omega_{0:k-1}|\mathbf{x}^i_{0:k-1}, \mathbf{y}_{1:k-1}) = \mathcal{N}(\Omega_{0:k-1}; \mu^i_{k-1}, \mathbf{\Gamma}^i_{k-1})$ is known, where $\mu^i_{k-1}$ and $\mathbf{\Gamma}^i_{k-1}$ denote the mean vector and the covariance matrix, respectively. Then the approximation of the posterior distribution $p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1})$ can be expressed as

$$\begin{aligned} p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1}) &= p(\Omega_{0:k-1}|\mathbf{x}_{0:k-1}, \mathbf{y}_{1:k-1})p(\mathbf{c}_{0:k-1}|\mathbf{y}_{1:k-1}) \\ &\approx \sum_i w^i_{k-1}\mathcal{N}(\Omega_{0:k-1}; \mu^i_{k-1}, \mathbf{\Gamma}^i_{k-1})\delta(\mathbf{c}^i_{0:k-1}, \mathbf{c}_{0:k-1}). \end{aligned} \tag{5.51}$$

The posterior distribution $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$ at time $kT$ can be obtained in three steps : prediction, correction and resampling.

**Prediction step** The predictive distribution $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k-1})$ can be written as

$$\begin{aligned} p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k-1}) &= p(\mathbf{s}_k|\mathbf{s}_{0:k-1})p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1}) \\ &= p(\Omega_k|\mathbf{x}_{0:k},\Omega_{0:k-1})p(\mathbf{c}_k|\mathbf{c}_{k-1})p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1}) \\ &\approx \sum_i w_{k-1}^i p(\Omega_k|\mathbf{x}_k,\mathbf{x}_{0:k-1}^i,\Omega_{0:k-1})\mathcal{N}(\Omega_{0:k-1};\mu_{k-1}^i,\mathbf{\Gamma}_{k-1}^i)p(\mathbf{c}_k|\mathbf{c}_{k-1}^i)\delta(\mathbf{c}_{0:k-1},\mathbf{c}_{0:k-1}^i). \end{aligned}$$

A Monte Carlo approximation of this distribution is

$$p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k-1}) \approx \sum_i w_{k-1}^i \mathcal{N}(\Omega_{0:k};\mu_{k|k-1}^i,\mathbf{\Gamma}_{k|k-1}^i)\delta(\mathbf{c}_{0:k}^i,\mathbf{c}_{0:k}) \tag{5.52}$$

where $\mathbf{c}_k^i$ is drawn from $p(\mathbf{c}_k|\mathbf{c}_{k-1}^i)$,

$$\mu_{k|k-1}^i = \begin{bmatrix} \mu_{k-1}^i \\ (\mathbf{A}_k^i)^T\mu_{k-1}^i \end{bmatrix} \tag{5.53}$$

and

$$\mathbf{\Gamma}_{k|k-1}^i = \begin{bmatrix} \mathbf{\Gamma}_{k-1}^i & \mathbf{\Gamma}_{k-1}^i\mathbf{A}_k^i \\ (\mathbf{A}_k^i)^T\mathbf{\Gamma}_{k-1}^i & (\mathbf{A}_k^i)^T\mathbf{\Gamma}_{k-1}^i\mathbf{A}_k^i + \mathbf{Q}_k^i \end{bmatrix}. \tag{5.54}$$

$\mathbf{A}_k^i$ and $\mathbf{Q}_k^i$ are obtained from (5.16) and (5.18) for the trajectory $\mathbf{x}_{0:k}^i$, and $\mu_{k|k-1}^i$ and $\mathbf{\Gamma}_{k|k-1}^i$ are derived in Appendix B.

**Correction step** In this step, the observation $\mathbf{y}_k$ is used to compute $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$ by updating $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k-1})$.

The observation equation (5.49) is re-written as follows :

$$\mathbf{y}_k = \mathbf{P}_k(\mathbf{x}_k) + \bar{\mathbf{J}}_k\Omega_{0:k} + \mathbf{e}_k \tag{5.55}$$

where $\bar{\mathbf{J}}_k = [\mathbf{0},\mathbf{J}_k]$ of size $n_k \times (k+1)N_{BS}$.

Before being normalized, the weights are updated by

$$w_k^i \propto w_{k-1}^i p(\mathbf{y}_k|\mathbf{x}_{0:k}^i,\mathbf{y}_{1:k-1}) \tag{5.56}$$

where $p(\mathbf{y}_k|\mathbf{x}_{0:k}^i,\mathbf{y}_{1:k-1})$ is a Gaussian distribution of mean $\mathbf{P}_k(\mathbf{x}_k^i) + \bar{\mathbf{J}}_k\mu_{k|k-1}^i$ and covariance $\bar{\mathbf{J}}_k\mathbf{\Gamma}_{k|k-1}^i\bar{\mathbf{J}}_k^T + E\{\mathbf{e}_k\mathbf{e}_k^T\}$.

The vector $\mu_k^i$ and matrix $\mathbf{\Gamma}_k^i$ are obtained by means of a Kalman filter described as follows :

- Kalman gain :

$$\mathbf{K}_k^i = \mathbf{\Gamma}_{k|k-1}^i\bar{\mathbf{J}}_k^T(\bar{\mathbf{J}}_k\mathbf{\Gamma}_{k|k-1}^i\bar{\mathbf{J}}_k^T + E\{\mathbf{e}_k\mathbf{e}_k^T\})^{-1}. \tag{5.57}$$

- Mean correction :

$$\mu_k^i = \mu_{k|k-1}^i + \mathbf{K}_k^i(\mathbf{y}_k - \mathbf{P}_k(\mathbf{x}_k^i) - \bar{\mathbf{J}}_k\mu_{k|k-1}^i). \tag{5.58}$$

- Covariance correction :

$$\mathbf{\Gamma}_k^i = \left(\mathbf{I}_{(k+1)N_{BS}} - \mathbf{K}_k^i\bar{\mathbf{J}}_k\right)\mathbf{\Gamma}_{k|k-1}^i. \tag{5.59}$$

The equations of this filter are derived in Appendix B.

When an order-$p$ AR model is considered, it is sufficient to store $\mathbf{c}_{k-p+1:k}^i$, the last $p$ elements of $\mu_k^i$ and the $p \times p$ lower right submatrix of $\mathbf{\Gamma}_k^i$.

**Resampling step** In order to avoid the weights degeneracy [123] due to the increase of the variance of the set $\{w_k^i\}_{i=1}^N$, a resampling step is performed. Thus, some trajectories of weak weights are eliminated and others of strong weights are repeated. We execute this step when the effective number of particles $\hat{N}_{eff} = 1/\sum_i (w_k^i)^2$ falls below a threshold. Unfortunately, resampling causes a depletion of the history due to the common past shared by some trajectories [128]. As a remark, when one small region is visited several times during a window time fitting the AR model order, the shadowings observations associated to this region are highly correlated. The performance might be decreased if the number of remaining distinct particles is not sufficient.

### Special case of collinear trajectories

In this section, we consider collinear trajectories for the sake of illustration. In practice, this enables using a map-restricted trajectory consistent with the considered technology. For example, train tracking, or car tracking are examples of such applications. Of course, when a crossroad approaches, all possible trajectories can be taken into account independently in the particle filter, and selected according to the maximal weights after crossing the intersection.

For Gaussian vectors, the precision matrix (i.e., the inverse of the covariance matrix), explicitly contains the information about the conditional independence of their components [129]. Let $\mathbf{v} = [v_1, \cdots, v_n]^T$ be a Gaussian vector of precision matrix $\Xi$, then

$$v_i \perp v_j | \mathbf{v}_{\_i,j} \quad \Leftrightarrow \quad \Xi_{i,j} = 0$$

where $\perp$ denotes the independence, and $\mathbf{v}_{\_i,j}$ is $\mathbf{v}$ without $v_i$ and $v_j$.

Consider a set of $L$ collinear points located on the x-axis and verifying $x_{j+1} > x_j$ as shown in Figure 5.11. Consider also an exponentially decreasing correlation function. In this case,

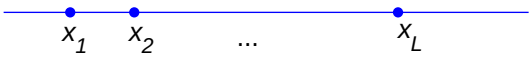



***Figure 5.11*** — A trajectory of collinear points.

it can be shown that the precision matrix $\mathbf{R}_L^{-1}$ of $[\epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_L)]^T$ is tridiagonal :

$$\mathbf{R}_L^{-1} = \begin{bmatrix} a_1 & b_1 & 0 & \cdots & 0 \\ b_1 & a_2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & a_{L-1} & b_{L-1} \\ 0 & \cdots & 0 & b_{L-1} & a_L \end{bmatrix}. \tag{5.60}$$

The proof of this proposition is given in Appendix C. As a result, the Gaussian process $\epsilon(\mathbf{x}_L)$ is an $AR(1)$ process. This can be deduced by looking at the last row of $\mathbf{R}_L^{-1}$ which has only two nonzero entries.

**Numerical results : Vehicle tracking in a Macro-cellular system**

In order to show the improvements brought by the algorithm, Monte Carlo simulations are performed. For this purpose, we consider a collinear trajectory with a varying speed. In order to illustrate the benefit obtained when using higher AR orders, we also consider the case of a U-turn.

The motion of the vehicle is modeled by the linear Markov process (5.31). The kinematic vector is $\mathbf{c}_k = [\mathbf{x}_k^T, \dot{\mathbf{x}}_k^T]^T$, and we assume that $\mathbf{a}_{k-1} = [a_{x,k-1}, a_{y,k-1}]^T$ is the acceleration vector provided by an accelerometer and $\mathbf{q}_k$ is a Gaussian distributed vector that accounts for the acceleration estimation errors. We take the covariance of $\mathbf{q}_k$ arbitrarily equal to $(0.5m/s^2)^2\mathbf{I}_2$. We perform tracking with a map constraint, where the trajectory belongs to a straight road with two lanes.

The model (5.31) is used only for tracking, while the trajectories are generated according to the model developed in [130] which takes into account a dynamic model of the vehicle, the driver's control decisions and the map of lanes. We consider the two trajectories depicted in Figure 5.12. Trajectory 1 is a straight line with an average speed of $57km/h$ and a maximum speed of $72km/h$. Notice that the vehicle is accelerating at the beginning. Trajectory 2 contains a turn and the average and maximum speeds are $34km/h$ and $65km/h$, respectively. The initial positions are assumed to be perfectly known.

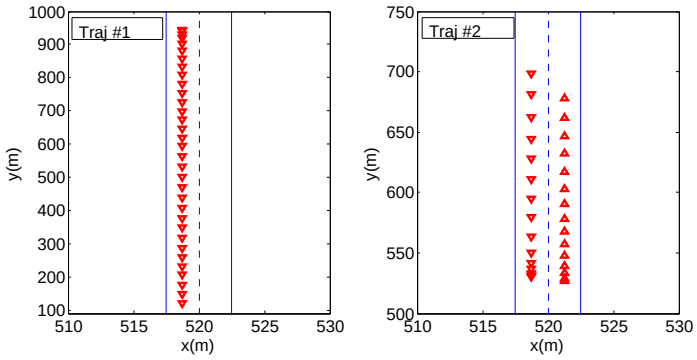



***Figure 5.12*** — The two traveled trajectories.

For the shadowing, the correlation function is the exponentially decreasing one with $d_{corr} = 50m$. The standard deviation of the shadowing is $\sigma_{sh} = 8dB$. The cross-correlation coefficient is equal to 0.5.

The Macro-cell system simulation baseline parameters defined in [131] are used to compute the path loss as a function of the position. The mobile station makes measurements with four sectorized base stations located at $[0, 0]^T$, $[1391, 1032]^T$, $[-199, 1721]^T$ and $[1589, -688]^T$. The measurement error vector is white in time and Gaussian and has a covariance matrix $E\{\mathbf{e}_k\mathbf{e}_k^T\}$, with $E\{e_{j,k}^2\} = 9dB^2$, $E\{e_{i,k}e_{j,k}\} = 0$ if the two error components correspond to different base stations and $E\{e_{i,k}e_{j,k}\} = 4.5dB^2$ if they correspond to different sectors of the same base station.

In Figure 5.13, the RMSE of the position is plotted vs time for trajectory 1 and for a time

step $T = 0.4$ seconds. These results are obtained by means of Monte Carlo simulations for 10 different sets of shadowing maps and for 50 trials per set, where a set represents the shadowing maps attributed to the different base stations. The dead reckoning (DR) curve corresponds to the estimation of the position based on the previous estimation and the acceleration only. In this case the position error is accumulated over time. A great improvement can be observed when the RSS measurements are exploited. Here, the order-1 AR is optimal since the trajectory is collinear, and higher orders do not degrade the accuracy. The joint position and shadowing tracking solution (i.e., $AR(1)$) reduces the position RMSE of up to $10m$ compared to the solution in which the shadowing is treated as white (i.e., zero order), but the remaining shadowing imprecisions do not allow removing its effect, and the position RMSE remains high compared to the known shadowing case.

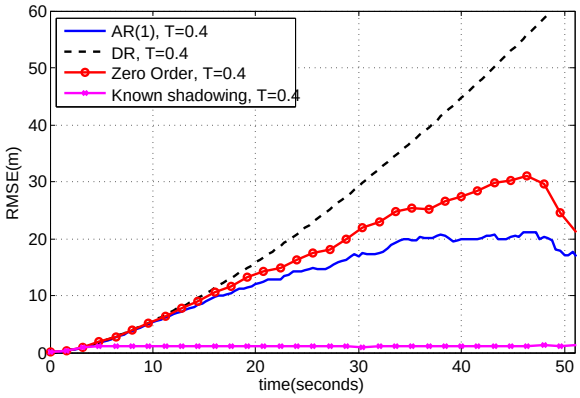



***Figure 5.13*** — RMSE of the position tracking for trajectory 1. The time step is $T = 0.4$ s.

For trajectory 2, the time step is $T = 1$ second and the total duration is 30 seconds. Thus, an order equal to 30 is capable of taking into account all previous states. Figure 5.14 shows the RMSE for this trajectory. After $t = 16$ seconds, when the turning occurs, the position RMSE of $AR(10)$ and $AR(30)$ is lower than for $AR(1)$. Indeed, the adjacency of the two parts of the trajectory improves the estimation that exploits the shadowing correlation. Few seconds after, the RMSE of $AR(10)$ increases. This behavior can be explained by the fact that the AR order is not sufficient to exploit the shadowing correlation in the actual measurements and in the measurements obtained at the beginning of the trajectory. By exploiting previous measurements, the location tracking can be highly improved.

## 5.5 Bayesian Map estimation

The simulation results presented in the previous sections showed that the knowledge of the shadowing can highly improve the localization accuracy. The classical method for constructing the shadowing maps is the fingerprinting, in which, measurements are collected at known positions and saved in a database. The problem with this solution is that it requires substantial time and effort to construct the database. This database should be maintained up-to-date as base stations may be added or displaced or the environment may change due

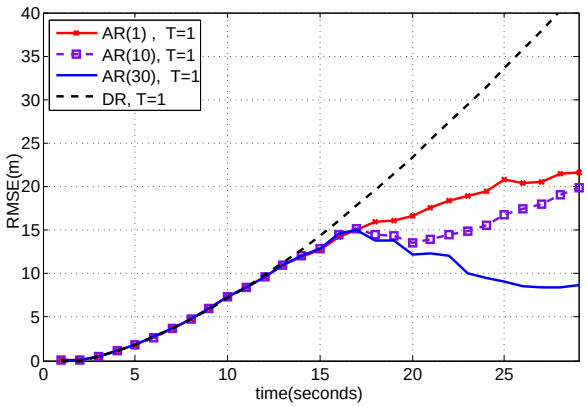



***Figure 5.14*** — RMSE of the position tracking for trajectory 2 and for different AR orders. The time step is $T = 1$ s.

to constructions or destructions of infrastructures.

In this section, we propose a solution for online estimation of the shadowing maps that does not require the positions of the mobile stations to be perfectly known. The mobile stations can have localization capabilities other than the RSS measurements. This solution allows refining and updating the maps in a collaborative way by using measurements made by several mobile stations.

Two solutions based on unsupervised learning have been developed in [45] and [132] for reducing the effort for the map construction.

In [45], traces are made over time, which are unlabeled trajectories or sequences of power measurements at unknown positions. The ML estimator is used to estimate the maps, and all the traces need to be saved. If a new trace becomes available, the map is updated using this new trace and the history of traces.

In [132], the ML estimator is also used but without considering traces and the observations are considered to be independent. This results in an accuracy reduction when measurements are made by a moving terminal, since the successive positions can be dependent of each other.

In the solution we develop here, the shadowing map is updated over time by updating the probability distribution of the coefficients using Bayesian estimation (update of the posterior distribution of the coefficients) and without the need to save all the measurement history. Other kinds of positioning information can be integrated in the update of the maps and positioning procedures, such as ToA, INS and GNSS, which can improve accuracies.

The shadowing maps modeling provided in Section 5.2.1 will be used. We start by describing how to update the shadowing atlas using RSS measurements at known positions, then we describe how to perform this estimation when positions are not known.

### 5.5.1 Atlas Update

Here, we describe how to update the atlas based on RSS measurements made at a known mobile position.

At time $kT$, a mobile station of known position $\mathbf{x}_k$ makes $n_k$ RSS measurements with the $N_{BS}$ base stations. The RSS observation vector of size $n_k$ is

$$\mathbf{y}_k = \mathbf{P}_k(\mathbf{x}_k) + \mathbf{H}_k\mathbf{\Lambda} + \mathbf{e}_k \tag{5.61}$$

where $\mathbf{P}_k$ is a known deterministic vector function, $\mathbf{\Lambda}$ is the shadowing atlas vector of size $L_{atlas}$, $\mathbf{H}_k$ is an $n_k \times L_{atlas}$ matrix that depends on the position $\mathbf{x}_k$, and $\mathbf{e}_k$ is the Gaussian error process.

Denote $\mathbf{M}_{k-1}$ and $\mathbf{\Sigma}_{k-1}$ the mean and covariance of $\mathbf{\Lambda}$ at time $(k-1)T$, respectively, and assume that the process $\mathbf{e}_k$ is white with respect to time. The observation $\mathbf{y}_k$, which is a linear function of $\mathbf{\Lambda}$, is used to update $\mathbf{M}_k$ and $\mathbf{\Sigma}_k$ by means of a linear MMSE estimator as follows :

$$\mathbf{M}_k = \mathbf{M}_{k-1} + \mathbf{Q}_k(\mathbf{y}_k - \mathbf{P}_k(\mathbf{x}_k) - \mathbf{H}_k\mathbf{M}_{k-1}) \tag{5.62}$$

and

$$\mathbf{\Sigma}_k = \mathbf{\Sigma}_{k-1} - \mathbf{Q}_k\mathbf{H}_k\mathbf{\Sigma}_{k-1} \tag{5.63}$$

where $\mathbf{Q}_k$ is the gain factor given by :

$$\mathbf{Q}_k = \mathbf{\Sigma}_{k-1}\mathbf{H}_k^T(E\{\mathbf{e}_k\mathbf{e}_k^T\} + \mathbf{H}_k\mathbf{\Sigma}_{k-1}\mathbf{H}_k^T)^{-1}. \tag{5.64}$$

The equations of the linear MMSE are derived in Appendix B.

### 5.5.2 Joint tracking and atlas update

The mobile station can perform additional localization measurements that we denote by $\mathbf{z}_k$ and that we assume independent given the positions, i.e., $p(\mathbf{z}_k, \mathbf{z}_l|\mathbf{x}_k, \mathbf{x}_l) = p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{z}_l|\mathbf{x}_l)$.

We define the state vector $\mathbf{s}_k$ consisting of the kinematic vector, i.e., $\mathbf{s}_k = \mathbf{c}_k$. The posterior distribution $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k})$ allows us to compute $p(\mathbf{\Lambda}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k})$ as follows :

$$p(\mathbf{\Lambda}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k}) = \int p(\mathbf{\Lambda}|\mathbf{s}_{0:k}, \mathbf{y}_{1:k})p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k})\mathrm{d}\mathbf{s}_{0:k} \tag{5.65}$$

where $p(\mathbf{\Lambda}|\mathbf{s}_{0:k}, \mathbf{y}_{1:k})$ is a Gaussian distribution whose parameters can be computed using the linear MMSE estimator previously described.

The distribution $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k})$ is computed recursively according to

$$p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k}) \propto p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1}, \mathbf{z}_{1:k-1})p(\mathbf{s}_k|\mathbf{s}_{k-1})p(\mathbf{y}_k|\mathbf{s}_{0:k}, \mathbf{y}_{1:k-1})p(\mathbf{z}_k|\mathbf{s}_k). \tag{5.66}$$

By assuming that the error process $\mathbf{e}_k$ is white, the observations at different instants are independent given the atlas :

$$p(\mathbf{y}_i, \mathbf{y}_j|\mathbf{s}_i, \mathbf{s}_j, \mathbf{\Lambda}) = p(\mathbf{y}_i|\mathbf{s}_i, \mathbf{\Lambda})p(\mathbf{y}_j|\mathbf{s}_j, \mathbf{\Lambda}). \tag{5.67}$$

Thus, a possible computation of $p(\mathbf{y}_k|\mathbf{s}_{0:k}, \mathbf{y}_{1:k-1})$ is

$$p(\mathbf{y}_k|\mathbf{s}_{0:k}, \mathbf{y}_{1:k-1}) = \int p(\mathbf{y}_k|\mathbf{\Lambda}, \mathbf{s}_k)p(\mathbf{\Lambda}|\mathbf{s}_{0:k-1}, \mathbf{y}_{1:k-1})\mathrm{d}\mathbf{\Lambda} \tag{5.68}$$

Equations (5.65) and (5.66) are analytically untraceable since the observation equation (5.61) is non-linear with respect to $\mathbf{s}_k$ and $\mathbf{H}_k$ depends on $\mathbf{s}_k$. These densities will be approximated numerically using particle filtering techniques in the following.

### 5.5.3 Implementation using particle filters

Assume that at time $(k-1)T$, $p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1}, \mathbf{z}_{1:k-1})$ is approximated by the set of $N$ weighted trajectories $\left\{\mathbf{s}^i_{0:k-1}, w^i_{k-1}\right\}_{i=1}^N$ :

$$p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1}, \mathbf{z}_{1:k-1}) \approx \sum_i w^i_{k-1}\delta(\mathbf{s}^i_{0:k-1}, \mathbf{s}_{0:k-1}). \tag{5.69}$$

The atlas posterior distribution is computed according to (5.65) :

$$p(\mathbf{\Lambda}|\mathbf{y}_{1:k-1}, \mathbf{z}_{1:k-1}) \approx \sum_i w^i_{k-1} p(\mathbf{\Lambda}|\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k-1}). \tag{5.70}$$

The Gaussian distributions $p(\mathbf{\Lambda}|\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k-1}) = \mathcal{N}(\mathbf{M}^i_{k-1}, \mathbf{\Sigma}^i_{k-1})$ are obtained using the linear MMSE estimator by one of two methods : The first method is to update $p(\mathbf{\Lambda}|\mathbf{s}^i_{0:k-2}, \mathbf{y}_{1:k-2})$ at each time step where a huge amount of memory is needed in order to store the $N$ atlases. The second method is to compute it from scratch at each time step using all the previous observations but with a complexity that increases with time.

We propose a suboptimal solution that approximates the weighted mixture of Gaussian distributions $\left\{\mathcal{N}(\mathbf{M}^i_{k-1}, \mathbf{\Sigma}^i_{k-1}), w^i_{k-1}\right\}_{i=1}^N$ by a single Gaussian one $\mathcal{N}(\hat{\mathbf{M}}_{k-1}, \hat{\mathbf{\Sigma}}_{k-1})$. This operation is repeated every $q$ time steps by substituting $p(\mathbf{\Lambda}|\mathbf{s}^i_{0:k+q-1}, \mathbf{y}_{1:k+q-1})$ by $p(\mathbf{\Lambda}|\mathbf{s}^i_{k:k+q-1}, \mathbf{y}_{k:k+q-1})$ in which the initial distribution $p(\mathbf{\Lambda})$ is substituted by $\mathcal{N}(\hat{\mathbf{M}}_{k-1}, \hat{\mathbf{\Sigma}}_{k-1})$.

The posterior distribution $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k})$ at time $kT$ can be obtained in three steps that are summarized in the following :

#### Prediction Step

A Monte Carlo approximation of the predictive distribution is obtained by drawing $\mathbf{s}^i_k$ from $p(\mathbf{s}_k|\mathbf{s}^i_{k-1})$ :

$$p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k-1}, \mathbf{z}_{1:k-1}) \approx \sum_i w^i_{k-1}\delta(\mathbf{s}^i_{0:k}, \mathbf{s}_{0:k}). \tag{5.71}$$

#### Correction Step

In this step, the observations $\mathbf{y}_k$ and $\mathbf{z}_k$ are used in order to compute $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k})$ by updating the weights according to

$$w^i_k \propto w^i_{k-1} p(\mathbf{y}_k|\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k-1}) p(\mathbf{z}_k|\mathbf{s}^i_k). \tag{5.72}$$

The value $p(\mathbf{y}_k|\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k-1})$ can be estimated with two methods :

- Using (5.68) with the complexity or memory drawbacks mentioned above.
- Using the following equation :

$$\begin{aligned} &p(\mathbf{y}_k|\mathbf{s}^i_{0:k}, \mathbf{y}_{1:k-1}) = \\ &\int p(\mathbf{y}_k|\Omega_k, \mathbf{s}^i_k) p(\Omega_k|\Omega_{0:k-1}, \mathbf{s}^i_{0:k}) p(\Omega_{0:k-1}|\mathbf{s}^i_{0:k-1}\mathbf{y}_{1:k-1}) \mathrm{d}\Omega_{0:k} \end{aligned} \tag{5.73}$$

where $\Omega_k$ is the shadowing vector at time $kT$. The distributions $p(\Omega_k|\Omega_{0:k-1}, \mathbf{s}^i_{0:k})$ and $p(\Omega_{0:k-1}|\mathbf{s}^i_{0:k-1}, \mathbf{y}_{1:k-1})$ can be computed using the Kalman filter described in Section 5.4.2, and the joint distribution $p(\Omega_{0:k}|\mathbf{s}_{0:k})$ is no more described by the assumed correlation function, but rather by the available mean and covariance of the atlas.

**Resampling Step**

In order to avoid the weights degeneracy due to the increase of the variance of the set $\{w^i_k\}^N_{i=1}$, a resampling step is performed.

### 5.5.4 Numerical results : Indoor tracking

We apply the Bayesian map estimation solution to indoor tracking with the deployment and trajectories given in Figure 5.15. The same path loss and shadowing models as for the example given in Section 5.4.1 are assumed.

First, the shadowing maps are estimated using RSS and ToA measurements made with a mobile terminal traveling along the trajectory of Figure 5.15(a). The RMSE of the position tracking is plotted in Figure 5.16(a) where the ToA measurements are made with the 4 BSs with a variance of $1m^2$.

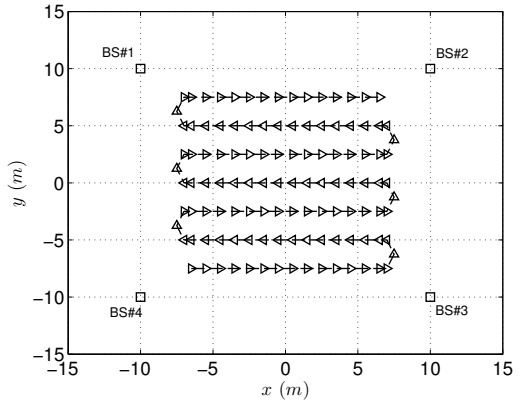
Then, a mobile terminal traveling along the trajectory of Figure 5.15(b) is tracked using RSS measurements and the estimated shadowing maps. The resulting position RMSE is plotted in Figure 5.16(b). We can notice that the location estimation accuracy is improved and the difference between the obtained RMSE and the RMSE in the case of perfect shadowing knowledge is less than one meter.

### 5.5.5 Application to train tracking

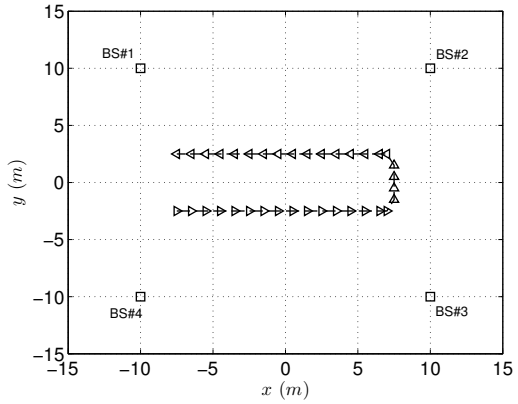
Reliable and accurate knowledge of train position and speed plays an important role in avoiding collisions and optimizing the traffic by increasing lines capacities. In classical train control systems, localization is based on track side equipments. The new trend in design of train control systems consists in integrating on-board solutions so as to reduce the need of track-side equipments with their inherent roll-out and maintenance costs and to simplify the deployment of new technologies and configuration changes.

Several methods for on-board position measurements are discussed in [133]. These methods are tachometers, Inertial Navigation Systems (INS) (e.g. accelerometers and gyroscopes), Doppler effect and GPS (and possibly other GNSSs such as Galileo).

The GPS solution provides a good precision when sufficient non-obstructed satellite signals are available. When the GPS fails (e.g. tunnels, valleys, some urban areas), other solutions are required. In [134], dead reckoning the train position from on-board sensors (odometers and accelerometers) is performed during the GPS failures where a data fusion approach based on Kalman filtering is developed. All the above-mentioned technologies but the GPS are DR as the current position is estimated based on a previous position and an estimation of the




(a)




(b)

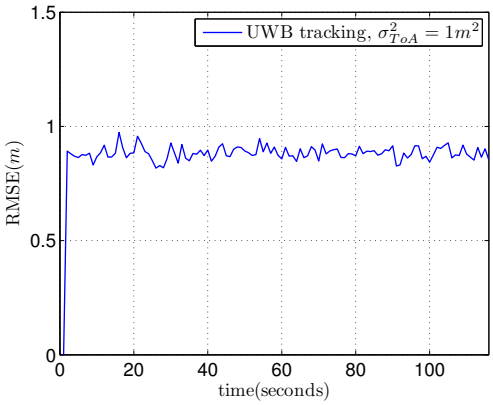
***Figure 5.15*** — Anchor nodes locations and two mobile terminal trajectories.

traveled distance over elapsed time. The performance of DR deteriorates with time due to error accumulation and a practical solution is needed to reduce this error especially if the GPS failure lasts for a long time duration.

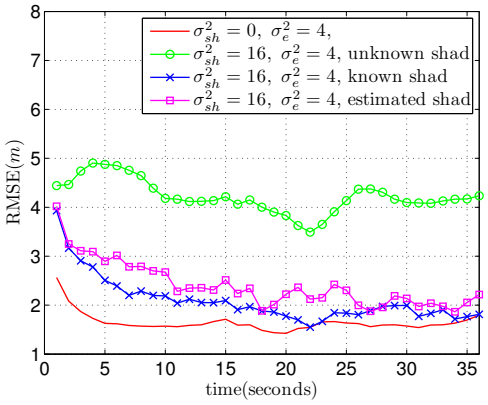
In this section, we consider the use of RSS measurements in radio communication systems for on-board train positioning. A radio communication system can be either dedicated for communication between the train and the railway regulation control centers (e.g. GSM-R) or a public network offering services to passengers. This solution does not suffer from error accumulation of DR and can be integrated with on-board sensors that can give more information and thus improve the accuracy. The Bayesian map estimation algorithm will be used for this purpose and will be validated via Monte Carlo simulations.

### Motion model

The train is constrained to move on a known railway track and the position is defined as the Cartesian coordinate of a reference point belonging to the train. We assume that there is no track branching, and the railway can be seen as a parametric curve of one parameter that




(a)




(b)

***Figure 5.16*** — (a) RMSE of the position tracking using ToA measurements. (b) RMSE of the position tracking using RSS measurements.

we call rail coordinate. We assume that the length of the rail (in meters $(m)$) between two rail coordinates $r_1$ and $r_2$ is equal to the absolute value of $r_1 - r_2$. The train position can be described by the rail coordinate as there is a direct mapping to the Cartesian coordinates. The train speed is the derivative of the rail coordinate.

We define the state vector $\mathbf{s}_k$ comprising the scalar rail coordinate $r_k$ and the scalar speed $v_k$. It evolves according to the following model :

$$\mathbf{s}_k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \mathbf{s}_{k-1} + \begin{bmatrix} 0 \\ T \end{bmatrix} (a_{k-1} + q_k) \tag{5.74}$$

where $a_{k-1}$ is the acceleration vector provided by an accelerometer and $q_k$ is a Gaussian variable that accounts for the acceleration estimation errors. We take the standard deviation of $q_k$ equal to $1m/s^2$ and the mean equal to zero (the accelerometer bias is assumed to be known, otherwise it can be added to the state vector and tracked by the particle filter). The time step is $T = 0.5s$.

The train moves on a linear rail of length $4km$ and there are two base stations located at $[-1390, -1032]^T$ and $[1589, -688]^T$ as shown in Figure 5.17. The train is moving from the

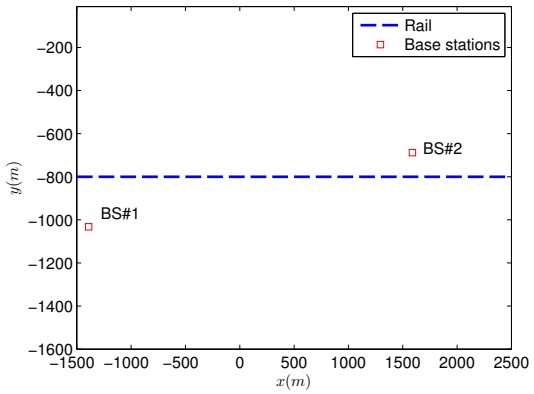



***Figure 5.17*** — Rail and base stations deployment.

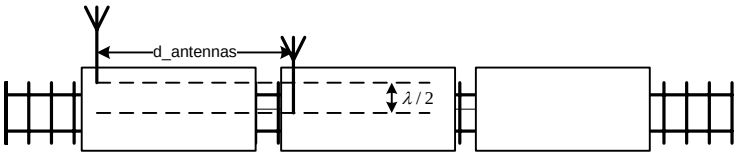



***Figure 5.18*** — Train antennas with a displacement of $\lambda/2$.

left to the right. The speed is linearly increasing with time from $200km/h$ to $250km/h$. The train takes $64s$ to travel the $4km$.

### Observation model

We consider several mobile stations having their antennas placed on the top of different carriages of the train for a better sensitivity and penetration losses avoidance. The antennas are sufficiently spaced (several meters to several tens of meters depending on the train size) so that the shadowing values are different at two antennas' positions for the same time instant. We denote $N_{MS}$ the number of these antennas. The RSS measurement model is given by (5.61), and the error $\mathbf{e}_k$ is a zero mean white Gaussian process with a diagonal covariance matrix and diagonal entries equal to $4dB^2$. In fact, to have a white process, the error remaining after averaging out the fast fading has to be white, and this can be obtained by a displacement of the antennas of about $\lambda/2$ perpendicular to the motion direction, as shown in Figure 5.18.

For the path loss, we consider the Macro-cell system simulation model defined in [131] with omnidirectional base station antennas.

The shadowing has a standard deviation of $\sigma_{sh} = 4dB$ and an exponentially decreasing correlation with $d_{corr} = 200m$. The cross-correlation coefficient from two different base stations is constant and equal to 0.5, for simplicity. The initial atlas mean is equal to zero and the atlas covariance matrix is constructed according these parameter values.

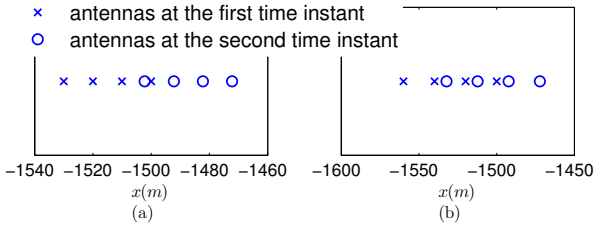



***Figure 5.19*** — Antennas positions at two consecutive time instants with separations of (a) $10m$ and (b) $20m$.

### Particle filter implementation

For the particle filter implementation, we take $N = 100$ particles. This number is sufficient since the dimension of the state vector is low (equal to two). We also replace the Gaussian process $p(\mathbf{\Omega}_k|\mathbf{\Omega}_{0:k-1}, \mathbf{s}^i_{0:k})$ in (5.73) by a first-order process $p(\mathbf{\Omega}_k|\mathbf{\Omega}_{k-1}, \mathbf{s}^i_{k-1:k})$. At low train speeds, it might be useful to consider higher orders of this process as the antennas at two non-consecutive time steps can be overlapping as shown in Figure 5.19(b).

### Numerical results

In Figure 5.20, we plot the RMSE of the position tracking. The position RMSE of the DR solution based on the accelerometer observations is increasing with time. The initial position and speed are perfectly known.

The antennas placed on the train are equidistant. We can see that with 4 antennas, the performance is better when the distance between two consecutive antennas is $20m$ (d_anntennas in Figure 5.18). A possible justification of this result is that the antennas are overlapping at two consecutive time steps when the separation is equal to $20m$ leading to a better estimation of the shadowing, while this overlap does not occur for a separation of $10m$ as shown in Figure 5.19. Moreover, the shadowing values affecting the different antennas measurements are less correlated for larger separations, and thus, leading to a higher diversity. We remark that the position RMSE begins to decrease after about $40s$ as the train approaches $BS\#2$ since the path loss decreases logarithmitically with the distance which is better estimated near a base station.

**Tracking using TDoA measurements** Now, we consider a first estimation of the shadowing atlas performed by a train measuring the time of flight of the signals emitted by the two base stations. These measurements are made by one mobile station. By assuming that the base stations are synchronized, the probability distribution of the position can be computed over time based on the following measurements :

$$\begin{aligned} z^1_k &= \|\mathbf{x}_k - \mathbf{x}^{BS1}\| + bias + \tilde{e}_{1,k} \\ z^2_k &= \|\mathbf{x}_k - \mathbf{x}^{BS2}\| + bias + \tilde{e}_{2,k} \end{aligned} \tag{5.75}$$

where *bias* is a bias value since the train equipment clock is not synchronized with base stations clocks, and $\tilde{e}_{i,k}$ is a zero mean Gaussian error of standard deviation equal to $20m$. The RMSE of the position estimated over time based on $\mathbf{z}_k = [z_k^1, z_k^2]^T$ is plotted in Figure 5.21(a) where the train speed is linearly increasing from $100km/h$ to $150km/h$ and the initial position is perfectly known.

The RSS measurements are made by four receivers having antennas spacing of 20 meters, and are used for estimating the atlas. In Figure 5.21(b), we plot the RMSE of one shadowing map estimation. The atlas estimation can be processed either on-line or off-line where the measurements made during the train journey are saved and processed by a server at the railway station. The initial distribution of the atlas can be also downloaded at the railway station.

After this estimation of the shadowing, a second train passes the railway and its position is tracked based on the RSS measurements performed by four equipments having an antennas spacing of 20 meters. The probability distribution of the atlas $\mathbf{\Lambda}$ computed after the first train passing is used during this operation. In Figure 5.22, the RMSE of the position tracking is plotted. The train speed is linearly increasing from $200km/h$ to $250km/h$. We plot also the position RMSE when the initial distribution of $\mathbf{\Lambda}$ is used. We can see that there is much improvement when the shadowing maps are estimated. Thus, the RSS-based tracking can be a good candidate for an on-board train positioning equipment.

For the error models assumed in this simulation, the time-of-flight measurements result in a better tracking accuracy than the RSS measurements (a RMSE of $7m$ for TDoA versus a RMSE of $10m$ for RSS) even though one receiver is used in the TDoA tracking. But in fact, timing measurements are more difficult to obtain and require maintaining a synchronization accuracy of the order of a few nanoseconds.

## 5.6 Conclusion

In this chapter, position tracking based on RSS measurements was addressed. The goal was to study the impact of the random shadowing on the positioning accuracy, and to develop

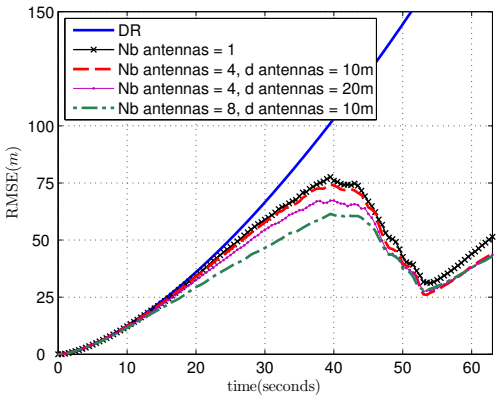
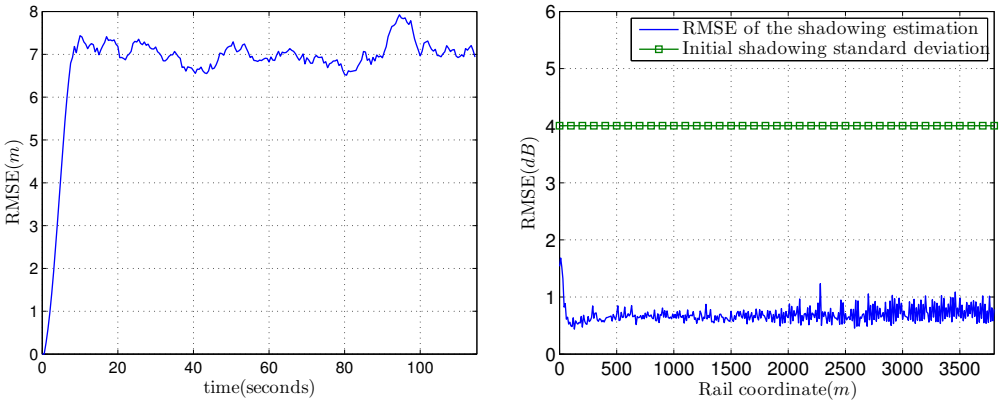



***Figure 5.20*** — RMSE of the train position tracking using RSS measurements only.




(a) (b)

***Figure 5.21*** — (a) RMSE of the train poistion tracking using TDoA measurements. (b) RMSE of the shadowing estimation.

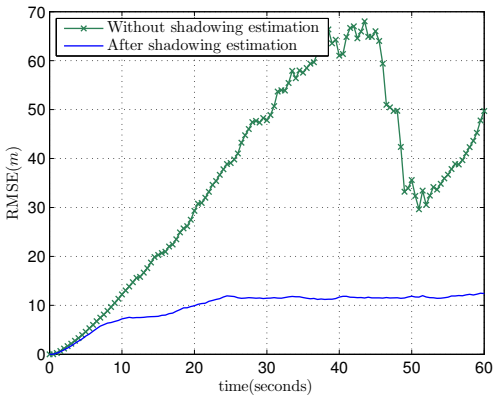



***Figure 5.22*** — RMSE of the train position tracking using RSS measurements.

solutions to mitigate it.

The shadowing maps corresponding to the different BSs were assumed to be realizations of Gaussian random fields with exponentially decreasing correlation functions of known parameters. These parameters can be obtained by a calibration procedure either in the considered area or in another area sharing similar physical properties.

We started by studying the improvements brought by the shadowing knowledge to the localization accuracy. The considered scenario was an indoor area with a small number of BSs (4 BSs). The simulation results showed that in the non-tracking position estimation (i.e., the position at a given time instant is computed using only the measurements made at this time instant), the knowledge of the shadowing does not always result in high accuracy improvements. This result was justified by the fact that, due to the presence of the shadowing, two distant positions might have close likelihood values, and thus, a small perturbation in the observations might result in a high localization error. On the other hand, in the Bayesian tracking solution, the accuracy improvement is much higher. This solution efficiently exploits the observations made at previous time instants using a motion model.

Then, we developed a solution for jointly tracking the position and the shadowing when this latter is unknown. An AR model was developed to describe the shadowing process. This solution was implemented by means of a Rao-Blackwellized particle filter which allows a reduction of the required number of particles. To exploit all the previous measurements, the order of the AR model increases with time, which involves increasing complexity and memory needs. This order can be reduced according to several factors, such as complexity and memory limitations, the a priori knowledge of the itinerary, the map constraints, the update time step or the accuracy of INS information. We applied this solution to the particular case of straight line trajectories, which is relevant for map-based car navigation systems, or train position estimations. We also showed that using high order AR-models can improve the positioning accuracy for some trajectories, such as the the trajectories with a U-turn. The accuracy delivered by this solution can still low compared with the case of perfectly known shadowing.

Finally, we developed a Bayesian shadowing map estimation solution which can be useful for reducing the calibration of the fingerprinting method. This solution estimated the maps by using unlabeled traces, which are sequences of RSS and other positioning observations made by mobile stations at unknown positions. This solution was applied to indoor tracking and outdoor train tracking. Significant positioning accuracy improvements were observed when the maps were first estimated using RSS and timing observations.

CHAPTER 6

# Conclusions and Future Research

In this thesis, some topics of wireless localization and tracking have been explored from a statistical signal processing perspective. These topics belong to the categories of cooperative static localization and non-cooperative dynamic localization. Several theoretical results have been derived and several localization and tracking algorithms have been developed. In this chapter, we restate the main contributions of the thesis with concluding remarks and we discuss future research ideas.

## 6.1 Conclusions

- The conditions of unique solvability in cooperative localization based on ranging measurements have been studied. These conditions are derived using results from graph rigidity theory and semidefinite programming (SDP) modeling of the localization problem. The results of graph rigidity theory are restricted to generic networks while those of SDP are not. An iterative SDP-based algorithm has been developed. It enhances the unique solvability sufficient conditions and improves the detection of uniquely solvable nodes. The conditions derived from graph rigidity theory only require the knowledge of the network connectivity while those derived from SDP require the knowledge of the true distance values between the nodes.
- The global identifiability defines the conditions of existence of consistent estimators (i.e., estimators that become unbiased and their variance tends to the Cramér-Rao bound (CRB) at high SNR). The following correspondence has been derived between the global identifiability in cooperative localization and the global rigidity : When the inter-node distances are globally identifiable from the ranging pair-wise measurements and the network has at least $d+1$ anchor nodes in general positions, global identifiability is equivalent to global rigidity. Thus, the global identifiability in generic networks can be checked using the connectivity property without the need to know the positions of the target nodes.
- A relationship has been derived between the Fisher information matrix (FIM) and the rigidity matrix. It allows us to deduce the following results :
  – For a generic network lying in a $d$-dimensional space and having at least $d$ anchor

nodes, local identifiability of all target nodes positions is equivalent to network rigidity.
  - When a $d$-dimensional network is rigid but not globally rigid and has at least $d$ anchor nodes, the FIM is non-singular, which means that a lower bound on the variance of any unbiased estimator can be computed, even though no such estimator does exist. The errors due to the non-global identifiability of the positions can be very high and the mean square error of the estimators can be much higher than the theoretical lower bounds.
- The application of weighted least-squares estimators to cooperative localization has been presented and its deterministic stability conditions (i.e., conditions of uniqueness of the global optimum) have been derived.
- Probabilistic inference in graphical models has been discussed and its application to cooperative localization using the nonparametric belief propagation algorithm has been studied. A new variant of this algorithm has been developed. This variant improves the accuracy and allows using the connectivity information for mitigating the flip ambiguity. This algorithm can be applied to any kind of pair-wise measurements provided that a probabilistic model is available. The accuracy of the developed algorithm has been studied via Monte Carlo simulations where the pair-wise measurements are distance measurements affected by additive errors.
- The multi-link shadowing correlation information has been shown, via a Monte Carlo simulation, to be useful for flip ambiguity mitigation.
- Concerning the non-cooperative dynamic localization, the RSS measurements have been exploited. These measurements are affected by a random shadowing that highly degrades the performance. The following work has been achieved in this topic :
  - The accuracy brought by the knowledge of the shadowing maps has been investigated using Monte Carlo simulations for an indoor deployment scenario with a small number of BSs. Simulation results have shown that the position tracking accuracy is much improved compared to the case of unknown shadowing. Simulation results have also shown that tracking brings more improvement compared to non-tracking position estimation. The position tracking under known shadowing has been processed using a regularized particle filter. This solution allows reducing the number of particles needed to avoid the divergence caused by the shadowing randomness.
  - A solution for improving the accuracy when the shadowing is unknown has been developed. It performs a joint tracking of the position and the shadowing and takes advantage form the shadowing spatial correlation. This solution is implemented using a Rao-Blackwellized particle filter. The shadowing process is modeled using an auto-regressive (AR) model. The application to vehicle tracking in a macro-cellular system showed that using AR orders higher than one can result in accuracy improvements, but this improvement comes at the expense of an increased complexity. In the absence of map constraints and when the motion model is not assisted by INS data, a divergence of the filter can be observed due to the high variance of the shadowing.
  - A solution for on-line estimation of the shadowing maps has been developed. This solution uses unlabeled traces, which are sequences of RSS and other positioning

measurements made at unknown positions. The maps are modeled using basis expansion and are estimated using a linear MMSE estimator for each sampled trajectory of particles. Simulation results have shown that the integration of accurate ranging measurements such as ToA and TDoA results in a better maps estimation and improves the RSS tracking accuracy. This solution has been applied to train tracking and has been shown to be a good candidate for on-board tracking equipment.

## 6.2 Future research

The research in wireless localization is a relatively new and wide research area. The following are some of the topics of interest : Design of waveforms that are suitable for estimating location dependent parameters from pair-wise measurements, design of algorithms for estimating these parameters, channel modeling and statistical modeling of the dependence of the estimated parameters on the positions, design of algorithms for estimating the positions and mitigating the induced propagation impairments, computation of the lower bounds of the accuracy achievable by estimators, derivation of the unique solvability conditions and flip ambiguity mitigation, etc.

In this section, we provide some future research ideas that are related to the topics explored in this thesis :

- The unique solvability and identifiability conditions have been studied in the case of ranging measurements. Other kinds of measurements can be addressed such as AoAs, spatially correlated data observations and timing measurements when there are several groups of nodes synchronized with each other.
- A quantitative comparison of the different cooperative localization algorithms in terms of accuracy and complexity under different deployment scenarios and network topologies and different error models.
- Comparison of the accuracy of the algorithms with the CRB, and study of other lower bounds when the CRB is too optimistic. This comparison is of great importance as it allows us to select the most suitable pair-wise measurements when several of them are available, and to decide whether it is worthwhile to pay additional effort for improving the accuracy achieved by an algorithm.
- Study of the improvement brought by the cooperation between the static or dynamic target nodes to the estimation of the shadowing maps. More specifically, study of the effect of density of cooperating nodes when they are performing RSS measurements only or RSS and low accuracy timing measurements.
- Construction of the shadowing maps of static nodes when their positions are not perfectly known.
- Study of the effect of the deviation of the assumed statistical models from the true ones, and on-line calibration or estimation of the models parameters in the Bayesian tracking solutions.

APPENDIX A

# Fisher Information Matrix in Cooperative Network Localization

The Fisher information matrix (FIM) $\mathbf{J}(\theta)$ of the target nodes coordinates is derived here. These unknown coordinates are considered as deterministic parameters. The notations of section 3.3 are used here.

The pair-wise measurements $y_{u,v}$ are random variables issued from the likelihood distributions $p(y_{u,v}|d_{u,v})$ in which $d_{u,v}$ is the only unknown parameter. Since the measurements are independent, the joint likelihood function factorizes as

$$p(\mathbf{y}|\theta) = \prod_{\substack{(u,v)\in E,\\ u<v \text{ and } m<v}} p(y_{u,v}|d_{u,v}), \tag{A.1}$$

and the joint log-likelihood function can be written as

$$\log p(\mathbf{y}|\theta) = \sum_{\substack{(u,v)\in E,\\ u<v \text{ and } m<v}} \log p(y_{u,v}|d_{u,v}). \tag{A.2}$$

The independent likelihood functions are assumed to verify the following regularity condition :

$$\int \nabla_\theta p(y_{u,v}|d_{u,v})\mathrm{d}y_{u,v} = \int (\nabla_\theta \log\, p(y_{u,v}|d_{u,v}))p(y_{u,v}|d_{u,v})\mathrm{d}y_{u,v} = \mathbf{0}. \tag{A.3}$$

Hence $E\{\nabla_\theta \log\, p(y_{u,v}|d_{u,v})\} = \mathbf{0}$.

Under this assumption, the FIM is given by

$$\mathbf{J}(\theta) = E\left\{\nabla_\theta \log\, p(\mathbf{y}|\theta)\,(\nabla_\theta \log\, p(\mathbf{y}|\theta))^T\right\}. \tag{A.4}$$

Let $(u,v)$ and $(k,l)$ be two couples of nodes indices verifying $u<v$ and $k<l$. For $(u,v)\neq(k,l)$ we have

$$\begin{aligned} E\{\nabla_\theta \log\, p(y_{u,v}|d_{u,v})(\nabla_\theta \log\, p(y_{k,l}|d_{k,l}))^T\} &= E\{\nabla_\theta \log\, p(y_{u,v}|d_{u,v})\}E\{\nabla_\theta \log\, p(y_{k,l}|d_{k,l})\}^T \\ &= \mathbf{0}, \end{aligned} \tag{A.5}$$

and for $(u,v)=(k,l)$ we have

$$E\{\nabla_\theta \log\, p(y_{u,v}|d_{u,v})(\nabla_\theta \log\, p(y_{u,v}|d_{u,v}))^T\} = E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\}\nabla_\theta d_{u,v}\,(\nabla_\theta d_{u,v})^T. \tag{A.6}$$

Thus, the FIM can be written as

$$\begin{aligned}\mathbf{J}(\theta) &= \sum_{\substack{(u,v)\in E,\\ u<v \text{ and } m<v}} \mathbf{J}_{u,v}(\theta)\\ &= \sum_{\substack{(u,v)\in E,\\ u<v \text{ and } m<v}} E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\}\nabla_\theta d_{u,v}\,(\nabla_\theta d_{u,v})^T.\end{aligned} \tag{A.7}$$

Now, we provide the computation of $E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\}$ for three kinds of measurements in additive noise.

## A.1 Case of distance measurements with additive Gaussian noise

The distance observations are modeled as

$$y_{u,v} = d_{u,v} + e_{u,v} \tag{A.8}$$

where $e_{u,v}$ is Gaussian distributed with mean equal to zero and variance equal to $\sigma^2_{u,v}$ (i.e., $e_{u,v}\sim\mathcal{N}(0,\sigma^2_{u,v})$). Thus

$$p(y_{u,v}|d_{u,v}) = \frac{1}{\sqrt{2\pi}\sigma_{u,v}}\exp\left(-\frac{(y_{u,v}-d_{u,v})^2}{2\sigma^2_{u,v}}\right), \tag{A.9}$$

and we can show that

$$E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} = \frac{1}{\sigma^2_{u,v}}. \tag{A.10}$$

## A.2 Case of RSS measurements with additive Gaussian noise

The RSS observations in decibel ($dB$) are modeled as

$$y_{u,v} = f(d_{u,v}) + e_{u,v} \tag{A.11}$$

where $f(d_{u,v})$ is a deterministic path loss function and $e_{u,v}\sim\mathcal{N}(0,\sigma^2_{u,v})$. Thus

$$p(y_{u,v}|d_{u,v}) = \frac{1}{\sqrt{2\pi}\sigma_{u,v}}\exp\left(-\frac{(y_{u,v}-f(d_{u,v}))^2}{2\sigma^2_{u,v}}\right), \tag{A.12}$$

and we can show that

$$E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} = \frac{(f'(d_{u,v}))^2}{\sigma^2_{u,v}}. \tag{A.13}$$

The path loss function must be differentiable in order to verify the regularity condition. In general, this function is decreasing with $d_{u,v}$, and by assuming a strictly decreasing one, we obtain $f'(d_{u,v}) \neq 0$.

## A.3 Case of distance measurements with additive noise distributed according to a mixture of Gaussian distributions

The distance observations are modeled as

$$y_{u,v} = d_{u,v} + e_{u,v} \tag{A.14}$$

where $e_{u,v}$ is issued from a mixture of Gaussian distributions :

$$e_{u,v} \sim \sum_{i=1}^{L} w_i \mathcal{N}(b_i, \sigma_i^2) \tag{A.15}$$

where $w_i$ are positive real weights having the sum equal to one (i.e., $\sum w_i = 1$) and $b_i$ is a positive bias corresponding to an excess delay due to NLOS propagation. Thus

$$p(y_{u,v}|d_{u,v}) = \sum_{i=1}^{L} \frac{w_i}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(y_{u,v} - d_{u,v} - b_i)^2}{2\sigma_i^2}\right), \tag{A.16}$$

and we can show that

$$E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} = E\left\{\frac{1}{p^2(y_{u,v}|d_{u,v})}\left(\frac{\partial p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\}, \tag{A.17}$$

and since $p(y_{u,v}|d_{u,v}) > 0$ for all $y_{u,v} \in \mathbb{R}$ and $\frac{\partial p(0|d_{u,v})}{\partial d_{u,v}} \neq 0$, we obtain

$$E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} \neq 0. \tag{A.18}$$

APPENDIX B

# Derivation of the Linear MMSE Estimator and Kalman Filter

## B.1 Conditional distribution of multivariate Gaussian vectors

Let $\mathbf{v} = [\mathbf{x}^T \ \mathbf{y}^T]^T$ be an $l \times 1$ jointly multivariate Gaussian distributed vector of mean $[E\{\mathbf{x}\}^T \ E\{\mathbf{y}\}^T]^T$ and partitioned covariance matrix

$$\mathbf{C}_v = \begin{bmatrix} \mathbf{C}_{xx} & \mathbf{C}_{xy} \\ \mathbf{C}_{yx} & \mathbf{C}_{yy} \end{bmatrix} \tag{B.1}$$

where $\mathbf{C}_{xx} = E\{\mathbf{x}\mathbf{x}^T\}$, $\mathbf{C}_{xy} = E\{\mathbf{x}\mathbf{y}^T\}$ and $\mathbf{C}_{yx}$ and $\mathbf{C}_{yy}$ are defined accordingly. The joint probability density function of $\mathbf{v}$ can be written as

$$p(\mathbf{x},\mathbf{y}) = \frac{1}{(2\pi)^{l/2}\det^{1/2}(\mathbf{C}_v)}\exp\left(-\frac{1}{2}\begin{bmatrix} \mathbf{x}-E\{\mathbf{x}\} \\ \mathbf{y}-E\{\mathbf{y}\} \end{bmatrix}^T \mathbf{C}_v^{-1} \begin{bmatrix} \mathbf{x}-E\{\mathbf{x}\} \\ \mathbf{y}-E\{\mathbf{y}\} \end{bmatrix}\right) \tag{B.2}$$

**Theorem B.1** ([34]). *The conditional distribution $p(\mathbf{y}|\mathbf{x})$ is also Gaussian and we have*

$$E\{\mathbf{y}|\mathbf{x}\} = E\{\mathbf{y}\} + \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}(\mathbf{x}-E\{\mathbf{x}\}) \tag{B.3}$$

$$\mathbf{C}_{y|x} = \mathbf{C}_{yy} - \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}\mathbf{C}_{xy}. \tag{B.4}$$

In this theorem, $\mathbf{C}_{xx}$ is assumed to be non-singular, otherwise an element of $\mathbf{x}$ is a linear combination of the other elements of $\mathbf{x}$ of and can be discarded.

## B.2 Linear minimum mean square error estimator

Let $\theta$ be an unknown vector of size $l \times 1$ drawn from a Gaussian distribution of mean $\mu_\theta$ and covariance $\mathbf{C}_\theta$. The prior Gaussian distribution of $\theta$ is denoted by $\mathcal{N}(\mu_\theta, \mathbf{C}_\theta)$. Let $\mathbf{y}$ be

an observation vector of size $m \times 1$ that linearly depends on $\theta$ according to

$$\mathbf{y} = \mathbf{H}\theta + \mathbf{w} \tag{B.5}$$

where $\mathbf{w}$ is a noise vector of distribution $\mathcal{N}(\mathbf{0}, \mathbf{C}_w)$ and independent of $\theta$.

The minimum mean square error (MMSE) estimator minimizes the mean square error

$$\text{mse}(\mathbf{y}) = \int \|\hat{\theta}(\mathbf{y}) - \theta\|^2 p(\theta|\mathbf{y}) \text{d}\theta \tag{B.6}$$

and is equal to the mean of the posterior distribution $p(\theta|\mathbf{y})$

$$\hat{\theta}_{MMSE} = \int \theta p(\theta|\mathbf{y}) \text{d}\theta \tag{B.7}$$

We can write

$$\begin{bmatrix} \theta \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_l & \mathbf{0} \\ \mathbf{H} & \mathbf{I}(m) \end{bmatrix} \begin{bmatrix} \theta \\ \mathbf{w} \end{bmatrix} \tag{B.8}$$

where $I_l$ is the identity matrix of dimension $l$. Since $\theta$ and $\mathbf{w}$ are independent of each other and each one is Gaussian distributed, they are jointly Gaussian distributed. The vector $[\theta^T \ \mathbf{y}^T]^T$ is also jointly Gaussian distributed since its a linear transformation of a joint Gaussian vector, and its mean vector is $[\mu_\theta^T \ \mu_\theta^T \mathbf{H}^T]^T$ and its covariance matrix is

$$\begin{bmatrix} \mathbf{C}_\theta & \mathbf{C}_{\theta y} \\ \mathbf{C}_{y\theta} & \mathbf{C}_{yy} \end{bmatrix} \tag{B.9}$$

where $\mathbf{C}_{\theta y} = \mathbf{C}_\theta \mathbf{H}^T$ and $\mathbf{C}_{yy} = \mathbf{H}\mathbf{C}_\theta \mathbf{H}^T + \mathbf{C}_w$. The mean and covariance of $p(\theta|\mathbf{y})$ can be computed according to equations (B.3) and (B.4) resulting in

$$E\{\theta|\mathbf{y}\} = \mu_\theta + \mathbf{C}_{\theta y}\mathbf{C}_{yy}^{-1}(\mathbf{y} - \mathbf{H}\mu_\theta) \tag{B.10}$$

and

$$\mathbf{C}_{\theta|y} = \mathbf{C}_\theta - \mathbf{C}_{\theta y}\mathbf{C}_{yy}^{-1}\mathbf{C}_{y\theta}. \tag{B.11}$$

Thus, the MMSE estimator of $\theta$ is

$$\hat{\theta}_{MMSE} = \mu_\theta + \mathbf{C}_{\theta y}\mathbf{C}_{yy}^{-1}(\mathbf{y} - \mathbf{H}\mu_\theta) \tag{B.12}$$

We can notice that this estimator is linear with respect to $\mathbf{y}$ and is called linear MMSE estimator.

The mean square error (MSE) matrix which is the covariance of the error vector $\hat{\theta}_{MMSE} - \theta$ is equal to $\mathbf{C}_{\theta|y}$.

## B.3 Sequential linear MMSE

We consider that observations arrive sequentially at discrete time instants. The observation at time $n$ is written as

$$\mathbf{y}_n = \mathbf{H}_n\theta + \mathbf{w}_n. \tag{B.13}$$

The noise process $\mathbf{w}_n$ is assumed to be white with respect to the time domain (i.e, independent at different time instants) and issued from a known Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{C}_{w,n})$.

Assume that we have already computed the posterior Gaussian distribution $p(\theta|\mathbf{y}_{0:n-1})$, where $\mathbf{y}_{0:n-1} = [\mathbf{y}_0^T, \cdots, \mathbf{y}_{n-1}^T]^T$, and we denote by $\mathbf{m}_{n-1}$ its mean vector and $\mathbf{C}_{n-1}$ its covariance matrix. When the observation $\mathbf{y}_n$ becomes available, the posterior distribution $p(\theta|\mathbf{y}_{0:n})$ can be updated as follows. The joint distribution $p(\theta, \mathbf{y}_n|\mathbf{y}_{0:n-1})$ is Gaussian with mean vector $[\mathbf{m}_{n-1}^T \ \mathbf{m}_{n-1}^T\mathbf{H}_n^T]^T$ and covariance matrix

$$\begin{bmatrix} \mathbf{C}_{n-1} & \mathbf{C}_{n-1}\mathbf{H}_n^T \\ \mathbf{H}_n\mathbf{C}_{n-1} & \mathbf{H}_n\mathbf{C}_{n-1}\mathbf{H}_n^T + \mathbf{C}_{w,n} \end{bmatrix}. \tag{B.14}$$

Thus, the mean and covariance of $p(\theta|\mathbf{y}_{0:n-1}, \mathbf{y}_n)$ are computed according to equations (B.3) and (B.4) resulting in

$$\mathbf{m}_n = \mathbf{m}_{n-1} + \mathbf{Q}_n(\mathbf{y}_n - \mathbf{H}_n\mathbf{m}_{n-1}) \tag{B.15}$$

and

$$\mathbf{C}_n = \mathbf{C}_{n-1} - \mathbf{Q}_n\mathbf{H}_n\mathbf{C}_{n-1}, \tag{B.16}$$

where $\mathbf{Q}_n$ is the gain matrix given by

$$\mathbf{Q}_n = \mathbf{C}_{n-1}\mathbf{H}_n^T(\mathbf{H}_n\mathbf{C}_{n-1}\mathbf{H}_n^T + \mathbf{C}_{w,n})^{-1}. \tag{B.17}$$

## B.4 Kalman filter

We define the hidden state vector $\mathbf{s}_n$ at time $n$ that evolves according to the linear Gauss-Markov model

$$\mathbf{s}_n = \mathbf{A}_n\mathbf{s}_{n-1} + \mathbf{b}_n \tag{B.18}$$

where $\mathbf{b}_n$ is a white Gaussian process of distribution $\mathcal{N}(\mathbf{0}, \mathbf{B}_n)$, and the initial state $\mathbf{s}_0$ is also Gaussian distributed and independent of $\mathbf{b}_n$.

The following noisy observation of the state $\mathbf{s}_n$ is made at time $n$ :

$$\mathbf{y}_n = \mathbf{H}_n\mathbf{s}_n + \mathbf{w}_n \tag{B.19}$$

where $\mathbf{w}_n$ is a white Gaussian noise of distribution $\mathcal{N}(\mathbf{0}, \mathbf{C}_{w,n})$.

The goal of the Kalman filter is to recursively update the posterior distribution $p(\mathbf{s}_n|\mathbf{y}_{1:n})$. Here we present a modified version that computes the posterior distribution of the history of states $p(\mathbf{s}_{0:n}|\mathbf{y}_{1:n})$. In fact, this posterior distribution is Gaussian since the state space evolution and the observation are linear and the noise processes are Gaussian.

Assume that at time $n-1$ the distribution $p(\mathbf{s}_{0:n-1}|\mathbf{y}_{1:n-1})$ have been computed and denote by $\mathbf{m}_{n-1|n-1}$ and $\mathbf{C}_{n-1|n-1}$ its mean and covariance, respectively. The distribution $p(\mathbf{s}_{0:n}|\mathbf{y}_{1:n})$ is computed in two steps : prediction and correction.

### B.4.1 Prediction

The predictive distribution factorizes as
$p(\mathbf{s}_{0:n}|\mathbf{y}_{1:n-1}) = p(\mathbf{s}_{0:n-1}|\mathbf{y}_{1:n-1})p(\mathbf{s}_n|\mathbf{s}_{n-1})$ and its mean and covariance are

$$\mathbf{m}_{n|n-1} = [\mathbf{m}_{n-1|n-1}^T \quad (\mathbf{A}_n\mathbf{m}_{n-1|n-1})^T]^T \tag{B.20}$$

and

$$\mathbf{C}_{n|n-1} = \begin{bmatrix} \mathbf{C}_{n-1|n-1} & \mathbf{C}_{n-1|n-1}\mathbf{A}_n^T \\ \mathbf{A}_n\mathbf{C}_{n-1|n-1} & \mathbf{A}_n\mathbf{C}_{n-1|n-1}\mathbf{A}_n^T + \mathbf{B}_n \end{bmatrix}. \tag{B.21}$$

### B.4.2 Correction

We can write $p(\mathbf{s}_{0:n}, \mathbf{y}_n|\mathbf{y}_{1:n-1}) = p(\mathbf{y}_n|\mathbf{s}_n)p(\mathbf{s}_{0:n}|\mathbf{y}_{1:n-1})$. The observation equation (B.19) can be written as

$$\mathbf{y}_n = \tilde{\mathbf{H}}_n\mathbf{s}_{0:n} + \mathbf{w}_n \tag{B.22}$$

where $\tilde{\mathbf{H}}_n = [\mathbf{0} \ \cdots \ \mathbf{0} \ \mathbf{H}_n]$ and the number of zero columns is equal to the size of $\mathbf{s}_{0:n-1}$. The mean of $p(\mathbf{s}_{0:n}, \mathbf{y}_n|\mathbf{y}_{1:n-1})$ is

$$\tilde{\mathbf{m}}_{n|n} = [\mathbf{m}_{n|n-1}^T \ (\tilde{\mathbf{H}}_n\mathbf{m}_{n|n-1})^T]^T \tag{B.23}$$

and its covariance

$$\tilde{\mathbf{C}}_{n|n} = \begin{bmatrix} \mathbf{C}_{n|n-1} & \mathbf{C}_{n|n-1}\tilde{\mathbf{H}}_n^T \\ \tilde{\mathbf{H}}_n\mathbf{C}_{n|n-1} & \tilde{\mathbf{H}}_n\mathbf{C}_{n|n-1}\tilde{\mathbf{H}}_n^T + \mathbf{C}_{w,n} \end{bmatrix}. \tag{B.24}$$

The mean and covariance of $p(\mathbf{s}_{0:n}|\mathbf{y}_{1:n})$ can be computed according to equations (B.3) and (B.4) resulting in

$$\mathbf{m}_{n|n} = \mathbf{m}_{n|n-1} + \mathbf{Q}_n(\mathbf{y}_n - \tilde{\mathbf{H}}_n\mathbf{m}_{n|n-1}) \tag{B.25}$$

and

$$\mathbf{C}_n = \mathbf{C}_{n|n-1} - \mathbf{Q}_n\tilde{\mathbf{H}}_n\mathbf{C}_{n|n-1}, \tag{B.26}$$

where $\mathbf{Q}_n$ is the Kalman gain matrix given by

$$\mathbf{Q}_n = \mathbf{C}_{n|n-1}\tilde{\mathbf{H}}_n^T(\tilde{\mathbf{H}}_n\mathbf{C}_{n|n-1}\tilde{\mathbf{H}}_n^T + \mathbf{C}_{w,n})^{-1}. \tag{B.27}$$

APPENDIX C

# Proof of the Proposition of Section 5.4.2

The same notations of section 5.4.2 are used here.

The $(i, j)$ entry of $\mathbf{R}_L^{-1}$ is :

$$\mathbf{R}_L^{-1}(i,j) = \frac{(-1)^{i+j}}{\det(\mathbf{R}_L)} \det(\mathbf{M}_{j,i}) \tag{C.1}$$

$\mathbf{M}_{j,i}$ is the matrix obtained by eliminating the $j^{th}$ row and the $i^{th}$ column of $\mathbf{R}_L$ and has a size of $(L-1) \times (L-1)$. For the $(i, j)$ couple verifying $|i-j| > 1$, the $i-1$ to $i+1$ rows of $\mathbf{M}_{j,i}$ are :

$$\begin{array}{ccc|ccc} e^{-\gamma(x_{i-1}-x_1)} & \dots & 1 & e^{-\gamma(x_{i+1}-x_{i-1})} & \dots & e^{-\gamma(x_L-x_{i-1})} \\ e^{-\gamma(x_i-x_1)} & \dots & e^{-\gamma(x_i-x_{i-1})} & e^{-\gamma(x_{i+1}-x_i)} & \dots & e^{-\gamma(x_L-x_i)} \\ e^{-\gamma(x_{i+1}-x_1)} & \dots & e^{-\gamma(x_{i+1}-x_{i-1})} & 1 & \dots & e^{-\gamma(x_L-x_{i+1})} \\ & \overset{\leftrightarrow}{i-1} & & & \overset{\leftrightarrow}{L-i} & \end{array} \tag{C.2}$$

The rank of the first $i-1$ columns is equal to one and the rank of the last $L-i$ columns is equal to one. Thus the rank of this submatrix is equal to two, and as a result the rank of $\mathbf{M}_{j,i}$ is equal to at most $L-2$ and its determinant is null.

APPENDIX D

# Résumé en français

## D.1 Géolocalisation dans les systèmes sans fil : Techniques et applications

### D.1.1 Introduction

Les progrès que les systèmes de communication sans fil ont connu ainsi que leur grande disponibilité ont permis l'apparition de plusieurs solutions de géolocalisation sans fil. Ces solutions peuvent compléter ou remplacer la solution fondée sur le système GPS dans les situations où cette dernière n'est pas fiable ou opérationnelle, et ont ouvert la voie au développment d'une variété d'applications et de services de géolocalisation.

Dans ce chapitre, nous commençons par présenter brièvement l'évolution des systèmes de localisation sans fil. Ensuite, nous discutons les principaux applications et services basés sur l'information de position. Puis, nous présentons les techniques fondamentales de localisation géométrique qui sont accomplies en deux phases : dans la première phase, des mesures de paramètres topo-dépendants sont effectuées sur les signaux reçus. Chaque mesure définit un lieu géométrique. Dans la deuxième phase, la position est calculée comme étant l'intersection des différents lieux. Les mesures les plus attractives sont celles qui permettent l'estimation de la distance entre un émetteur et un récepteur. Ces mesures comprennent les mesures du temps de propagation et les mesures de la puissance, et sont présentées avec les sources de bruit et leurs modélisations statistiques. Une technique de localisation qui a soulevée beaucoup d'intérêt à cause de sa précision est la technique de fingerprinting, dans laquelle les mesures sont comparées aux entrées d'une base de données, où une entrée est une empreinte de mesures à un emplacement connu. Les entrées de la base de donnée sont obtenues soit par des campagnes de mesures soit à l'aide d'outils de calibrage. Cette technique est également décrite. Une classification des algorithmes de localisation basées sur les méthodes de mesure ou sur le type d'information utilisée est aussi fournie dans ce chapitre

Pour des raisons de concision, nous allons juste présenter les notions que nous allons utiliser dans ce résumé.

### D.1.2 Méthodes fondamentales de localisation

Nous définissons un nœud cible comme étant un dispositif dont la position est inconnue, et un nœud d'ancrage comme étant un dispositif dont la position est connu, ou en d'autres termes, dont les coordonnées dans un espace Euclidien de dimension 2 ou 3 sont connues. Les méthodes fondamentales de localisation calculent la position d'un seul nœud cible en deux étapes :

- Dans la première étape, des métriques sont mesurées sur les signaux échangés entre le nœud cible et plusieurs nœuds d'ancrage. Des informations concernant la position de la cible par rapport au nœud d'ancrage peuvent être extraites de ces métriques.
- Dans la deuxième étape, les mesures sont agrégées et traitées par un algorithme de localisation ou de fusion de données qui fournit en sortie la position.

En supposant que les mesures sont parfaites, chaque métrique définit une contrainte sur les coordonnées du nœud cible, ou en d'autres termes une région de faisabilité, et la position peut être calculée sans ambiguïté quand ces régions se croisent en un seul point. Dans les applications réelles, les mesures sont bruitées. Par conséquent, les régions de faisabilité estimées ne contiennent pas nécessairement la vraie position, et leur intersection peut être l'ensemble vide. Dans ce cas, des techniques d'estimation appropriées sont nécessaires pour définir l'algorithme de localisation.

### D.1.3 Mesures de distances et sources d'erreurs

Les deux principales métriques qui permettent d'estimer la distance entre un émetteur et un récepteur sont le temps de propagation (ToA) et la puissance reçue (RSS).

#### Mesure du temps de propagation

La distance est obtenue en multipliant le temps de propagation par la vitesse de propagation. Il existe deux techniques de ToA :

- Technique unidirectionnelle qui est basée sur une seule transmission et nécessite une synchronisation des horloges de l'émetteur et du récepteur.
- Technique bidirectionnelle qui utilise deux transmissions pour surmonter le besoin de synchronisation.

Pour la plupart des applications 'indoor', l'erreur de mesure ToA ne doit pas dépasser quelques nanosecondes. Les principales sources d'erreurs sont les erreurs de synchronisation, les dérives des horloges, les bruits additifs, les trajets multiples et le blocage du trajet direct (NLoS).

**Utilisation de signaux ultra-large bande** Les signaux ultra-large bande sont construits à partir d'impulsions de courtes durées qui sont largement étalées dans le domaine fréquentiel. Ainsi, ils permettent de mesurer le temps de propagation avec une bonne précision en présence des bruits additifs. Les courtes durées des impulsions permettent aussi de séparer les trajets multiples.

**Modélisation statistique des erreurs** Des mesures effectuées à des petites portées ont montré que, pour des propagations en vue directe (LoS), les erreurs peuvent être modélisées par des variables aléatoires gaussiennes. Dans le cas des propagations NLoS, d'autres modèles ont été proposés pour tenir compte des erreurs qui peuvent prendre des valeurs importantes. Des exemples de ces modèles sont le mélange de gaussiennes et le mélange de distributions gaussiennes et exponentielles.

#### Mesure de la puissance du signal reçue

La mesure de la puissance du signal reçue (RSS) peut facilement être obtenue sans aucune complexité supplémentaire au niveau du matériel. L'estimation de la distance à partir de cette mesure repose sur le principe que la puissance moyenne reçue est décroissante avec la distance séparant le nœud de transmission du nœud de réception. Ainsi, un modèle d'atténuation de la puissance est nécessaire pour l'estimation de la distance à partir de la valeur de RSS. Un tel modèle qui est largement utilisé est donné par

$$P_r(d) = P_0 - 10n_p \log_{10} d + \epsilon \tag{D.1}$$

où $P_r(d)$ (en $dB$) est la valeur de RSS à une distance $d$, $P_0$ est la valeur moyenne de RSS à une distance de $1m$, $n_p$ est l'exposant d'atténuation, et $\epsilon$ est une variable aléatoire gaussienne centrée de variance $\sigma_{sh}^2$ représentant l'évanouissement à grande échelle ou masquage. Le principal inconvénient de cette technique est la grande variance de la variable aléatoire $\epsilon$, ce qui réduit la précision de l'estimation de la position. En outre, cette précision dépend de la valeur de la distance puisque la fonction de log dans (D.1) varie lentement pour les grandes valeurs de distance.

### D.1.4 Localisation coopérative et poursuite

Les méthodes de localisation que nous avons décrites dans la section D.1.2 considèrent un nœud cible connecté à (ou en communication avec) plusieurs nœuds d'ancrage, et la position est calculée grâce aux mesures effectuées avec ces ancres. Dans de nombreux scénarios, les nœuds cibles peuvent ne pas être connectés à un nombre suffisant de nœuds d'ancrage. De plus, même si un nombre suffisant de points d'ancrage est disponible, la précision de positionnement peut ne pas répondre aux besoins des applications envisagées. Pour surmonter ces problèmes, d'autres méthodes de localisation ont été développées en exploitant d'autres types de mesures telles que les mesures intra-nœud dans le cas des nœuds en déplacement, et les mesures inter-nœuds entre les paires de nœuds cibles. Selon le type de mesures utilisées, les algorithmes de localisation peuvent être classés dans l'une des quatre classes suivantes :

1. Localisation non-coopérative et statique : la position est calculée uniquement à partir des mesures effectuées avec des nœuds d'ancrage. Si le nœud cible est en mouvement, la position à un instant donné est calculée sans l'aide d'un modèle de mouvement ou des mesures effectuées aux instants précédents.
2. Localisation non-coopérative et dynamique : les algorithmes de cette catégorie effectuent une poursuite de la position dans le temps. La poursuite permet d'améliorer la précision

en exploitant les mesures effectuées à plusieurs instants. La dépendance entre ces mesures est dérivée d'un modèle de mouvement définissant la dépendance entre les positions à des instants différents. Des mesures intra-nœud peuvent être utilisées pour améliorer la précision du modèle de mouvement. Ces mesures peuvent être délivrées par des capteurs de navigation inertielle (INS) tels que des accéléromètres et des gyroscopes.

3. Localisation coopérative et statique : la localisation coopérative permet de contourner le besoin d'une grande densité de nœuds d'ancrage et la puissance de transmission élevée pour la connectivité à longue portée. Elle est basée sur l'utilisation de mesures effectuées entre des paires de nœuds cibles, en plus de celles avec les nœuds d'ancrage. Les positions de plusieurs nœuds cibles sont calculées simultanément.
4. Localisation coopérative et dynamique : une opération de poursuite des cibles mobiles est effectuée et la coopération est maintenue en exploitant les mesures inter-nœuds.

## D.2 Rigidité, identifiabilité et localisabilité en localisation coopérative

### D.2.1 Introduction

Dans la localisation coopérative, on cherche à attribuer des positions aux nœuds cibles, dans un espace Euclidien, qui soient conformes à un ensemble de mesures prises par paire entre les nœuds. En général, les mesures ne sont pas disponibles pour toutes les paires de nœuds, et le graphe du réseau n'est pas complètement connecté. Par conséquent, on peut ne pas avoir suffisamment d'informations pour estimer les différentes positions sans ambiguïtés. Même avec la connaissance exacte des distances entre les nœuds, des ambiguïtés existent quand il y a plusieurs solutions possibles vérifiant les contraintes. La théorie de la rigidité graphique et la programmation semi-définie définissent des conditions suffisantes pour la solvabilité unique des positions des nœuds cibles, en supposant des mesures de distance sans erreur. Dans l'estimation probabiliste, où les mesures sont affectées par des fluctuations aléatoires, l'ambiguïté peut être définie par la théorie de l'identifiabilité. L'objectif de ce chapitre est d'étudier, pour les réseaux statiques, les conditions de solvabilité unique lorsque les valeurs de distances sont exactes, et les conditions d'identifiabilité lorsque ces valeurs sont bruitées. En étudiant les propriétés d'identifiabilité, les positions inconnues sont traitées comme des paramètres déterministes, et les mesures sont supposées suivre des modèles probabilistes connus.

Nous commençons par examiner la théorie de la rigidité graphique dans le contexte de la localisation coopérative. Puis, le bruit de mesure est pris en compte, et nous établissons une relation entre la rigidité et la matrice d'information de Fisher (FIM), et les conditions de non-singularité de la FIM sont ainsi déduites. L'identifiabilité est également étudiée dans cette section où nous établissons des correspondances entre la rigidité et l'identifiabilité. Ces correspondances permettent de vérifier des propriétés d'identifiabilité en utilisant des outils de rigidité graphique. La programmation semi-définie (SDP) est aussi étudiée, où la méthode

de l'état de l'art est décrite, et une nouvelle technique est proposée pour améliorer la détection des nœuds ayant des solutions uniques.

### D.2.2 Théorie de la rigidité graphique

Dans cette théorie, on cherche à déterminer si un graphe a une réalisation unique dans un espace de dimension donnée. Les résultats de cette théorie ont été appliqués à la localisation coopérative, et fournissent des conditions suffisantes pour garantir des solutions de position uniques. Dans cette section, nous présentons certains de ces résultats (une étude plus détaillée est fournie dans le mémoire), et nous commençons par fournir quelques notations et définitions préliminaires.

#### Définitions

Définissons un réseau $\mathcal{L}$ de dimension $d$ et de taille $N$ nœuds. Ce réseau est formé de $m < N$ nœuds d'ancrage numérotés de 1 à $m$ et de $n = N - m$ nœuds cibles numérotés de $m+1$ à $N$. La position du nœud $i$ est reprèsentée par le vecteur colonne $\mathbf{x}_i \in \mathbb{R}^d$. La distance entre les nœuds $i$ et $j$ est notée $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$, ou en d'autres termes la norme Euclidienne de $\mathbf{x}_i - \mathbf{x}_j$. Dans ce chapitre, nous supposons que $\mathbf{x}_i \neq \mathbf{x}_j$ pour tout $1 \leq i < j \leq N$.

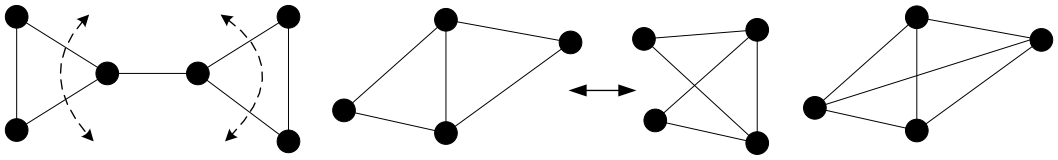
Un graphe $G = (V, E)$, ayant $V = \{1, 2, \cdots, N\}$ comme ensemble de sommets et $E = \{(i,j)|\ i,j \in V$ et $i$ et $j$ sont des voisins$\}$ comme ensemble d'arêtes est associé au réseau $\mathcal{L}$ comme suit. Le sommet $i$ est associé au nœud $i$, et deux sommets sont reliés par une arête s'ils correspondent à des noeuds d'ancrage ou à des nœuds effectuant une mesure de distance entre eux. Les termes nœud et sommet seront utilisés indifféremment.

Un structure $(G, \mathbf{p})$ de dimension $d$ est définie comme étant un ensemble constitué d'un graphe $G = (V, E)$ et d'une fonction $\mathbf{p}$ de $V$ dans $\mathbb{R}^d$ telle que $\mathbf{p}(i) \neq \mathbf{p}(j)$ pour tout $(i,j) \in E$.

Pour deux structures $(G, \mathbf{p})$ et $(G, \mathbf{q})$ ayant la même dimension, on a les définitions suivantes :

- $(G, \mathbf{p})$ et $(G, \mathbf{q})$ sont équivalentes si $\|\mathbf{p}(i) - \mathbf{p}(j)\| = \|\mathbf{q}(i) - \mathbf{q}(j)\|$ pour tout $(i,j) \in E$.
- $(G, \mathbf{p})$ et $(G, \mathbf{q})$ sont congruentes si $\|\mathbf{p}(i) - \mathbf{p}(j)\| = \|\mathbf{q}(i) - \mathbf{q}(j)\|$ pour tout $i, j \in V$.
- $(G, \mathbf{p})$ est globalement rigide si chaque structure qui est équivalente à $(G, \mathbf{p})$ est aussi congruente à $(G, \mathbf{p})$ (par exemple, la Figure D.1(c)).
- $(G, \mathbf{p})$ est rigide s'il existe un $\epsilon > 0$ de telle sorte que chaque structure $(G, \mathbf{q})$ équivalante à $(G, \mathbf{p})$ et satisfaisant $\|\mathbf{p}(i) - \mathbf{q}(i)\| < \epsilon$ pour tout $i \in V$ est congruente à $(G, \mathbf{p})$ (par exemple, la Figure D.1(b)).
- Un structure qui n'est pas rigide est flexible, c.-à-d., susceptible de déformations continues (par exemple, la Figure D.1(a)).

Une structure $(G, \mathbf{p})$ est appelée une structure de réalisation du réseau $\mathcal{L}$ si ces deux entités ont la même dimension et $\|\mathbf{p}(i) - \mathbf{p}(j)\| = d_{i,j}$ pour tous les $(i,j) \in E$, où $G$ est le graphe associé au réseau. Un structure $(G, \mathbf{p})$ est appelée une réalisation fondamentale du réseau $\mathcal{L}$ s'il s'agit d'une structure de réalisation vérifiant $\mathbf{p}(i) = \mathbf{x}_i$ pour tout $i \in V$.



(a) Flexible (b) Rigide (c) Globalement rigide

***Figure D.1*** — Trois structures bidimensionnelles.

Le problème de localisation est défini par

$$\begin{aligned} \text{déterminer} \quad & \mathbf{x}_j,\ j = m+1, \ldots, N, \\ \text{s.q.} \quad & \mathbf{x}_i,\ i = 1, \ldots, m, \\ & \|\mathbf{x}_u - \mathbf{x}_v\| = d_{u,v},\ \forall (u,v) \in E. \end{aligned} \tag{D.2}$$

Le réseau $\mathcal{L}$ est dit uniquement solvable si le problème (D.2) admet une solution unique, c.-à-d., tous les nœuds sont uniquement solvables.

**Théorème D.1.** *Le réseau $\mathcal{L}$ de dimension $d$ est solvable de manière unique si et seulement si toute structure de réalisation de $\mathcal{L}$ est globalement rigide et le réseau a au moins $d+1$ nœuds d'ancrage en positions générales.*

Les nœuds sont dit en positions générales si leurs vecteurs de position sont linéairement indépendants.

Un graphe $G' = (V'; E')$ est un sous-graphe de $G = (V, E)$ si $V' \subseteq V$ et $E' = \{(i,j) \in E | i,j \in V'\}$. La structure $(G', \mathbf{p})$ est alors une sous-structure de $(G, \mathbf{p})$. Un nœud est uniquement solvable si sa solution est la même pour chaque solution du problème (D.2). Une conséquence directe du théorème D.1 est le corollaire suivant :

**Corollaire D.1.** *Un nœud est uniquement solvable s'il appartient à une sous-structure globalement rigide ayant au moins $d+1$ nœuds d 'ancrage en positions générales, pour toute structure de réalisation de $\mathcal{L}$.*

La condition définie par ce corollaire est suffisante pour la solvabilité unique. Des conditions nécessaires et suffisantes ne sont pas encore connues. La théorie de la rigidité graphique fournit des outils pour tester les propriétés de rigidité et de rigidité globale. Maintenant, nous présentons la matrice de rigidité qui est un outil important et qui sera utilisée plus tard pour établir une relation entre la rigidité et la FIM.

### Matrice de rigidité

Un mouvement souple d'une structure $(G, \mathbf{p})$ est défini par la fonction $\mathbf{P}$ de $\mathbb{R}^+ \times V$ dans $\mathbb{R}^d$ vérifiant les conditions suivantes :

1. $\mathbf{P}(t, v)$ est dérivable par rapport à $t$, où $t$ peut être considéré comme une variable de temps.

2. $\mathbf{P}(0, v) = \mathbf{p}(v)$ pour tout $v \in V$.
3. $\|\mathbf{P}(t, v) - \mathbf{P}(t, u)\| = \|\mathbf{p}(v) - \mathbf{p}(u)\|$ pour toute $(u, v) \in E$ et tout $t \geq 0$.

Ainsi, pour $t = 0$, $(G, \mathbf{p})$ est rigide si et seulement si tout mouvement souple donne lieu à une structure congruente à $(G, \mathbf{p})$.

Le point 3 peut être réécrit comme

$$\|\mathbf{P}(t, v) - \mathbf{P}(t, u)\|^2 = \text{constante} \quad \text{par rapport à } t \tag{D.3}$$

et en dérivant par rapport à $t$, on obtient

$$[\mathbf{P}(t, v) - \mathbf{P}(t, u)]^T \left[\mathbf{P}'(t, v) - \mathbf{P}'(t, u)\right] = 0 \tag{D.4}$$

où $\mathbf{P}'(t, v)$ représente la vitesse instantanée de la fonction $\mathbf{P}$, et $^T$ est l'opérateur de transposition. A l'instant $t = 0$, définissons par $\mathbf{p}'(v) = \mathbf{P}'(0, v)$ :

$$[\mathbf{p}(v) - \mathbf{p}(u)]^T \left[\mathbf{p}'(v) - \mathbf{p}'(u)\right] = 0 \text{ pour tout } (u, v) \in E \tag{D.5}$$

qui peut s'écrire sous la forme

$$\begin{bmatrix} \mathbf{p}(v) - \mathbf{p}(u) \\ \mathbf{p}(u) - \mathbf{p}(v) \end{bmatrix}^T \begin{bmatrix} \mathbf{p}'(v) \\ \mathbf{p}'(u) \end{bmatrix} = 0 \text{ pour tout } (u, v) \in E. \tag{D.6}$$

Une affectation de vitesse $\mathbf{p}'(v)$, pour tout $v \in V$, vérifiant le système d'équations (D.6) est appelée mouvement infinitésimal de $(G, \mathbf{p})$. Ce système d'équations peut être utilisé pour déterminer une base pour tous les mouvements infinitésimaux.

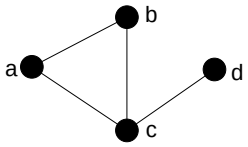
La matrice de rigidité de la structure $(G, \mathbf{p})$, dénotée par $M(G, \mathbf{p})$, est la matrice des coefficients de (D.6). Sa taille est $|E| \times d|V|$, où $|.|$ désigne la cardinalité d'un ensemble. Chaque arête indexe une ligne dans cette matrice, et chaque sommet indexe $d$ colonnes consécutives. Pour une ligne indexée par une arête $(u, v) \in E$, les entrées correspondant au sommet $u$ sont égales à $(\mathbf{p}(u) - \mathbf{p}(v))^T$ et celles correspondant au sommet $v$ sont égales à $(\mathbf{p}(v) - \mathbf{p}(u))^T$, et les autres entrées sont égales à zéro. Une structure $(G, \mathbf{p})$ et sa matrice de rigidité sont présentées par la Figure D.2, où $\mathbf{p}(i) = [x_i \ , \ y_i]^T \in \mathbb{R}^2$.

Les mouvements infinitésimaux d'une structure constituent l'espace nul de la matrice de rigidité. Cet espace nul est toujours à rang non-nul, car il contient les mouvements triviaux, c.-à-d., les rotations et les translations de la structure, et par conséquent, la matrice de rigidité ne peut pas avoir un rang plein. Pour une structure de dimension $d$ ayant au moins $d$ nœuds, qui ne sont pas tous alignés pour $d = 3$, il existe $d$ translations triviales indépendantes et $d(d-1)/2$ rotations triviales indépendantes. Ainsi, le nombre total de mouvements triviaux indépendants est $d(d+1)/2$.

Pour un certain nombre de nœuds $N \geq 2$, définissons

$$S(N, d) = \begin{cases} dN - d(d+1)/2 & \text{si } N \geq d+2 \\ N(N-1)/2 & \text{si } N \leq d+1 \end{cases}. \tag{D.7}$$

Remarquons que pour $N \geq d+2$, $S(N, d)$ est égal au nombre de colonnes de la matrice de rigidité moins le nombre des mouvements triviaux. Le rang minimum de l'espace nul de $M(G, \mathbf{p})$ est égal au nombre de mouvements triviaux.




(a)

$$\begin{bmatrix} x_a - x_b & y_a - y_b & x_b - x_a & y_b - y_a & 0 & 0 & 0 & 0 \\ x_a - x_c & y_a - y_c & 0 & 0 & x_c - x_a & y_c - y_a & 0 & 0 \\ 0 & 0 & x_b - x_c & y_b - y_c & x_c - x_b & y_c - y_b & 0 & 0 \\ 0 & 0 & 0 & 0 & x_c - x_d & y_c - y_d & x_d - x_c & y_d - y_c \end{bmatrix}$$

(b)

***Figure D.2*** — Une structure bidimensionnelle avec sa matrice de rigidité. La première ligne correspond à l'arête $(a, b)$, la deuxième à l'arête $(a, c)$, la troisième à l'arête $(b, c)$ et la quatrième à l'arête $(c, d)$

**Théorème D.2.** *Soit* $(G, \mathbf{p})$ *une structure de dimension* $d$ *avec* $N \geq 2$ *nœuds. Alors* $rang M(G, \mathbf{p}) \leq S(N, d)$*. En outre, si l'égalité est vérifiée, alors* $(G, \mathbf{p})$ *est rigide.*

En d'autres termes, lorsque l'égalité est vérifiée, les seuls mouvements possibles sont les mouvements triviaux, et la structure ne peut pas avoir une déformation continue. $(G, \mathbf{p})$ est dite être infiniment rigide si $\text{rang}M(G, \mathbf{p}) = S(N, d)$. D'après le théorème D.2, la rigidité infinitésimale est une condition suffisante de rigidité. Mais la rigidité infinitésimale n'est pas équivalente à la rigidité. La rigidité infinitésimale est équivalente à la rigidité pour une catégorie de structures appelées structures génériques.

### Structures génériques

Une structure $(G, \mathbf{p})$ est dite générique si sa matrice de rigidité a le rang maximal parmi toutes les structures possibles $(G, \mathbf{q})$ ayant la même dimension. Pour une structure de dimension $d$ constituée de $N$ nœuds, les coordonnées des nœuds peuvent être considérés comme un seul point de $\mathbb{R}^{Nd}$. Les structures génériques forment un ensemble ouvert et dense de $\mathbb{R}^{Nd}$, et les structures non-génériques forment un ensemble algébrique de $\mathbb{R}^{Nd}$. Ainsi, si nous tirons au hasard $N$ positions uniformément distribuées dans un ensemble ouvert de $\mathbb{R}^d$, la structure obtenue est générique avec une probabilité égale à un.

**Théorème D.3.** *Soit* $(G, \mathbf{p})$ *une structure générique de dimension* $d$*.* $(G, \mathbf{p})$ *est rigide si et seulement si* $(G, \mathbf{p})$ *est infiniment rigide.*

Pour un graphe $G$, le rang de $M(G, \mathbf{p})$ sera maximisé lorsque $(G, \mathbf{p})$ est générique, et donc, $\text{rang}M(G, \mathbf{p})$ est le même pour toutes les structures génériques qui partagent le même graphe dans une dimension donnée. Ce fait nous permet de présenter le théorème suivant :

**Théorème D.4.** *Si un graphe possède au moins une structure infiniment rigide, alors toutes ses structures génériques sont rigides.*

Ainsi, la rigidité est une propriété combinatoire du graphe dans toutes les dimensions.

**Test de rigidité générique** Etant donnée une dimension $d$, la rigidité générique n'est pas une caractéristique d'une structure spécifique, mais plutôt une caractéristique du graphe. Des algorithmes combinatoires de complexité polynomiale pour tester la rigidité ne sont connus que pour $d = 1$ et 2. En outre, le test de rang de la matrice de rigidité peut être utilisé dans toutes les dimensions.

**Test de rigidité globale générique** La rigidité globale est une propriété générique dans toutes les dimensions. Des algorithmes combinatoires de complexité polynomiale pour tester la rigidité globale ne sont connus que pour $d = 1$ et 2.

**Structures non-génériques** Il est NP-difficile de décider si une structure de dimension $d$ est rigide même pour $d = 1$.

#### Construction distribuée des réseaux rigides et globalement rigides

Les algorithmes pour tester la rigidité et la rigidité globale ne sont pas applicables d'une manière distribuée. Néanmoins, il existe des catégories de réseaux pour lesquels les propriétés de rigidité et de rigidité globale peuvent être vérifiées de façon distribuée. Mais ces réseaux ne constituent que des sous-ensembles des réseaux rigides et globalement rigides. Enfin, nous mentionnons la condition suivante nécessaire pour la solvabilité unique : Si un nœud est uniquement solvable, alors il existe $d + 1$ chemins disjoints connectant ce nœud aux nœuds d'ancrage, où deux chemins sont appelés disjoints s'ils ne partagent pas un nœud en commun. Cette condition peut être utilisée pour identifier des nœuds n'ayant pas de solution unique.

### D.2.3 La rigidité, la FIM et l'identifiabilité

Dans cette section, les positions inconnues des nœuds cibles sont considérées comme des paramètres déterministes, et les mesures sont considérées comme des réalisations de variables aléatoires. Avant de fournir les contributions originales de cette section, qui sont la correspondance entre la matrice d'information de Fisher (FIM) et la rigidité et les correspondances entre la rigidité et l'identifiabilité, nous commençons par présenter le calcul de la FIM.

Soit $\theta = [\mathbf{x}_{m+1}^T, \cdots, \mathbf{x}_N^T]^T$ le vecteur de paramètres de taille $nd$ composé des coordonnées des $n = N - m$ nœuds cibles.

La valeur scalaire $y_{u,v}$, correspondant à la mesure entre les nœuds $u$ et $v$, est supposée être la réalisation d'une variable aléatoire $\mathrm{y}_{u,v}$ de fonction de distribution connue paramétrée par la distance $d_{u,v}$ :

$$y_{u,v} \sim p_{\mathrm{y}_{u,v}|d_{u,v}} \tag{D.8}$$

où $d_{u,v}$ est le seul paramètre inconnu dans cette distribution, pour tout $(u, v) \in E$, $u < v$ and $m < v$. Nous supposons également que les mesures sont indépendantes. Soit $\mathbf{y}$ le vecteur

de toutes les mesures. Ce vecteur est une réalisation d'un vecteur aléatoire $Y$. La fonction de vraisemblance de $\theta$ peut être écrite comme

$$p(\mathbf{y}|\theta) = \prod_{\substack{(u,v)\in E \\ u<v,\ m<v}} p(y_{u,v}|d_{u,v}) \tag{D.9}$$

où les indices des fonctions de distribution sont supprimées afin de simplifier la notation.

Sous la condition de régularité, c.-à-d., $\log p(\mathbf{y}|\theta)$ est continûment dérivable par rapport à $\theta$, la FIM est donnée par

$$\mathbf{J}(\theta) = E_{Y|\theta}\left\{\nabla_\theta \log\, p(\mathbf{y}|\theta)\left(\nabla_\theta \log\, p(\mathbf{y}|\theta)\right)^T\right\} \tag{D.10}$$

où $E_{Y|\theta}\{.\}$ est la valeur de la moyenne conditionnelle, et $\nabla_\theta$ est le gradient par rapport à $\theta$.

Dans les sections suivantes, nous supposons que $E\left\{\left(\frac{\partial \log\, p(y_{u,v}|d_{u,v})}{\partial d_{u,v}}\right)^2\right\} \neq 0$ pour tout $d_{u,v} > 0$. Cette condition est vérifiée par les mesures de ToA affectées par des bruits additifs issus de lois gaussiennes ou de mélanges de gaussiennes, ou par les mesures de RSS affectées par des bruits additifs gaussiens (en dB).

### Correspondance entre la rigidité et la FIM

Soit $(G, \mathbf{p})$ la structure de réalisation fondamentale du réseau $\mathcal{L}$, et soit $M^{\ddagger}(G, \mathbf{p})$ la matrice obtenue de la matrice de rigidité $M(G, \mathbf{p})$ en supprimant toutes les colonnes correspondant aux nœuds d'ancrage. Nous avons démontré la proposition suivante :

**Proposition D.1.** *Les rangs de la FIM* $\mathbf{J}(\theta)$ *et de la matrice* $M^{\ddagger}(G, \mathbf{p})$ *sont égaux :*

$$rang\mathbf{J}(\theta) = rang M^{\ddagger}(G, \mathbf{p}). \tag{D.11}$$

En supposant que le réseau $\mathcal{L}$ de dimension $d$ a au moins $d$ nœuds d'ancrage et que la sous-structure de $(G, \mathbf{p})$ correspondant aux nœuds d'ancrage est générique, nous avons démontré le théorème suivant :

**Théorème D.5.** *La structure* $(G, \mathbf{p})$ *est infiniment rigide si et seulement si la FIM* $\mathbf{J}(\theta)$ *est non singulière.*

Puisque la rigidité est équivalente à la rigidité infinitésimale pour les réseaux génériques, on peut en déduire le corollaire suivant :

**Corollaire D.2.** *Un réseau générique de dimension d ayant au moins d nœuds d'ancres a un graphe rigide si et seulement si la FIM correspondante est non singulière.*

Ainsi, pour les réseaux génériques, la singularité de la FIM est une propriété graphique qui peut être testée sans avoir besoin de connaître la position des nœuds.

**Théorie d'identifiabilité**

Soit $\mathbf{y}$ le vecteur d'observation de taille $k$. Nous citons ci-dessous des définitions offertes par la théorie d'identifiabilité :

- Deux points de paramètres $\theta^0$ et $\theta^1$ sont dits équivalents selon l'observation si $p(\mathbf{y}|\theta^0) = p(\mathbf{y}|\theta^1)$ pour tout $\mathbf{y} \in \mathbb{R}^k$.
- Un point de paramètre $\theta^0$ est dit globalement identifiable s'il n'y a pas un autre $\theta \in \mathbb{R}^{nd}$ qui lui est équivalent.
- Un point de paramètre $\theta^0$ est dit localement identifiable s'il existe un voisinage ouvert de $\theta^0$ ne contenant aucun autre paramètre qui lui est équivalent.

**Identifiabilité locale** Nous avons démontré la proposition suivante :

**Proposition D.2.** *Pour un réseau générique de dimension d ayant au moins d nœuds d'ancrage, l'identifiabilité locale est équivalente à la rigidité graphique.*

**Identifiabilité globale** Nous avons démontré la proposition suivante :

**Proposition D.3.** *Pour un réseau de dimension d ayant au moins $d+1$ nœuds d'ancrage en position générale, l'identifiabilité globale des positions des nœuds cibles implique la rigidité globale de toute structure de réalisation.*

Maintenant, nous allons montrer que lorsque les distances sont globalement identifiables à partir des mesures, la rigidité globale devient une condition suffisante pour l'identifiabilité globale. Pour cela, supposons que les valeurs de distance $d_{u,v}$ sont globalement identifiable pour toute $(u, v) \in E$, $u < v$ et $m < v$, c.-à-d., une valeur de distance $d^0_{u,v}$ n'est pas équivalente à aucune autre valeur de distance $d^1_{u,v}$

**Théorème D.6.** *Pour un réseau de dimension d ayant au moins $d+1$ nœuds d'ancrage en position générale, l'identifiabilité globale des positions des nœuds cibles est équivalente à la rigidité globale de toute structure de réalisation.*

Ainsi, afin de vérifier l'identifiabilité globale, il suffit de vérifier la rigidité globale. Pour les réseaux génériques, l'identifiabilité globale est une propriété graphique qui ne dépend pas des positions.

### D.2.4 Localisabilité par programmation semi-définie

Une approche alternative pour l'étude de la solvabilité unique est la programmation semi-définie (SDP). La méthode classique de SDP peut être utilisée pour identifier uniquement les de nœuds qui sont localisables de manière unique, c.-à-d., qui ont des solutions uniques dans toutes les dimensions. Nous avons développé un algorithme itératif basé sur la SDP qui améliore l'identification des nœuds à solution unique. Les algorithmes basés sur la SDP nécessitent la connaissance des valeurs des distances.

## D.3 Algorithmes de localisation coopérative

### D.3.1 Introduction

Dans ce chapitre consacré aux algorithmes de localisation coopérative, nous nous intéressons à l'estimation probabiliste, où les positions sont considérées comme des variables aléatoires et le problème de localisation est formulé comme une estimation dans les modèles graphiques probabilistes. Ce formalisme permet l'application de l'algorithme de propagation de croyance (BP), où chaque nœud calcule une fonction de densité de probabilité de ses coordonnées, de façon distribuée, en se basant sur l'information locale à priori, et sur les densités de probabilité fournie à chaque itération par les nœuds voisins. Cet algorithme produit des estimations des positions et des mesures des incertitudes. En outre, certaines ambiguïtés sur les positions peuvent être éliminées en utilisant l'algorithme BP et en supposant un modèle de connectivité probabiliste. L'algorithme BP est implémenté en utilisant une version particulaire où les messages, qui ne peuvent pas être calculés d'une façon analytique, seront approximés numériquement. Cette implémentation est connue sous le nom de propagation de croyance non paramétrique (NBP). La principale contribution de ce chapitre est le développement d'une nouvelle variante de la méthode NBP à deux phases. Cette solution améliore la précision de la localisation et permet de réduire la quantité de données échangées. En plus de l'estimation probabiliste, nous étudions les estimateurs de moindres carrés pondérés (WLS, Weighted Least Squares), et nous fournissons les conditions de leur stabilité déterministe en se basant sur les résultats de solvabilité unique fournis dans le chapitre précédent. Un aperçu de plusieurs algorithmes existants de localisation coopérative est fourni dans la version complète de ce chapitre.

### D.3.2 Définitions

Nous considérons un réseau $\mathcal{L}$ de taille $N$ composé de $m$ nœuds d'ancrage numérotés de 1 à $m$ et $n = N - m$ nœuds cibles numérotés de $m + 1$ à $N$ et dont les positions sont à estimer. Un graphe $G = (V, E)$ est associé au réseau. Le réseau est déployé dans un espace de dimension $d$ où le nœud $i$ est situé en $\mathbf{x}_i \in \mathbb{R}^d$. Les nœuds sont supposés être statiques.

Soit $\theta = [\mathbf{x}_{m+1}^T, \cdots, \mathbf{x}_N^T]^T$ le vecteur des positions inconnues. La mesure par paire entre les nœuds $i$ et $j$ est désignée par $\mathbf{y}_{i,j}$. Lorsque la distance est estimée à partir de $\mathbf{y}_{i,j}$, on la note $\tilde{d}_{i,j} = f(\mathbf{y}_{i,j})$. La vraie valeur de distance est $d_{i,j} = \|\mathbf{x}_i - \mathbf{x}_j\|$. Soit $\mathbf{y}$ le vecteur de toutes les mesures.

### D.3.3 Moindres carrés pondérés

Les estimateurs WLS ont été largement utilisés dans la localisation non coopérative et coopérative parce que leur formulation est simple et ne nécessitent pas d'hypothèses particulières sur le modèle de bruit. Un estimateur WLS cherche à minimiser une fonction de coût

de la forme :

$$\hat{\theta}_{WLS} = \underset{\theta \in \mathbb{R}^{dn}}{\arg\min} \sum_{\substack{(i,j)\in E \\ i<j,\ m<j}} w_{i,j}(\tilde{d}_{i,j} - \|\mathbf{x}_i - \mathbf{x}_j\|)^2 + \sum_{i>m} r_i \|\mathbf{x}_i - \bar{\mathbf{x}}_i\|^2, \tag{D.12}$$

où $w_{i,j}$ est un poids positif reflétant la précision de $\tilde{d}_{i,j}$, et les paramètres $\bar{\mathbf{x}}_i$ et $r_i$ modélisent la connaissance à priori sur $\mathbf{x}_i$. En l'absence de telle information, $r_i$ est égal à zéro. Nous noterons la fonction de coût (D.12) par $s(\theta)$.

Le problème (D.12) est un problème d'optimisation non-convexe. Les minimums locaux sont fréquents dans ce genre de problèmes et peuvent correspondre à de mauvaises configurations du réseau et à des erreurs d'estimation élevées. Un minimum de $s(\theta)$ peut être calculé en utilisant une solution numérique et itérative à partir de positions initiales. Les positions initiales doivent être choisies avec soin afin de réduire la probabilité de tomber dans un minimum local. Un exemple d'algorithme pour trouver des positions initiales est la SDP adaptée aux mesures bruitées.

#### La stabilité déterministe du WLS

Pour un problème d'estimation non linéaire exprimé comme un problème de minimisation d'une fonction de coût, la stabilité déterministe implique l'unicité du minimum global à condition que le bruit d'observation soit assez petit. Nous avons démontré que pour les réseaux génériques et uniquement solvable (c.-à-d., globalement rigide et ayant $d+1$ nœuds d'ancrage), la propriété de stabilité déterministe est vérifiée.

### D.3.4 Estimation probabiliste

L'estimation probabiliste exploite les modèles probabilistes disponibles sur les mesures (par exemple, les fonctions de vraisemblance) et sur les positions (par exemple, les informations à priori). Les algorithmes probabilistes peuvent être plus précis que les algorithmes non probabilistes (par exemple, WLS), car ils peuvent exploiter les informations à priori et peuvent atténuer l'effet de NLoS lorsqu'un modèle probabiliste de propagation NLoS est disponible. Ils sont aussi plus résistants à la présence de valeurs aberrantes, où une valeur aberrante est une observation qui se situe en dehors de la tendance générale d'une distribution. Afin d'appliquer ces algorithmes, on suppose que l'observation $\mathbf{y}$ est la réalisation d'une variable aléatoire $\mathbf{Y}$ de fonction de probabilité conjointe connue $p_{Y|\theta}(\mathbf{y}|\theta)$, qui sera écrite $p(\mathbf{y}|\theta)$. Pour un vecteur $\mathbf{y}$ donnée et $\theta$ variable, $p(\mathbf{y}|\theta)$ est appelée la fonction de vraisemblance de $\theta$.

Les estimateurs probabilistes peuvent être classés comme non bayésien ou bayésien, selon que $\theta$ est considéré comme un paramètre déterministe inconnu ou la réalisation d'une variable aléatoire de fonction de distribution à priori connue $p_\Theta(\theta)$.

### D.3.5 Algorithmes basés sur l'échange de messages

Dans le problème de localisation coopérative, sous l'hypothèse que les observations $\{\mathbf{y}_{i,j}\}$ et les positions $\{\mathbf{x}_i\}$ sont les réalisations de variables aléatoires indépendantes, et en considérant que les positions des nœuds d'ancrage sont des variables aléatoires de distributions à priori égales à la fonction delta de Dirac, la distribution à posteriori conjointe se factorise comme suit :

$$p(\mathbf{x}|\mathbf{y}) = \frac{1}{Z}\prod_{i\in V}\Phi_i(\mathbf{x}_i)\prod_{(i,j)\in E}\Psi_{i,j}(\mathbf{x}_i,\mathbf{x}_j), \tag{D.13}$$

où $\mathbf{x} = [\mathbf{x}_1^T, \cdots, \mathbf{x}_m^T, \theta^T]^T$, $\Phi_i(\mathbf{x}_i) = p(\mathbf{x}_i)$ est la fonction de distribution à priori de la position du nœud $i$, et $\Psi_{i,j}(\mathbf{x}_i,\mathbf{x}_j) = p(\mathbf{y}_{i,j}|\mathbf{x}_i,\mathbf{x}_j)$ est une fonction de potentiel par paire.

Le but des algorithmes basés sur l'échange de messages est de calculer, d'une manière distribuée, des fonctions distributions $b_i(\mathbf{x}_i)$ qui sont des approximations des distributions marginales $p(\mathbf{x}_i|\mathbf{y})$. Par définition, la distribution marginale est obtenue en intégrant (ou en faisant la somme de) la fonction de distribution conjointe sur toutes les variables sauf $\mathbf{x}_i$ :

$$p(\mathbf{x}_i|\mathbf{y}) = \int p(\mathbf{x}|\mathbf{y})\mathrm{d}\mathbf{x}_1\cdots\mathrm{d}\mathbf{x}_{i-1}\mathrm{d}\mathbf{x}_{i+1}\cdots\mathrm{d}\mathbf{x}_N. \tag{D.14}$$

Le calcul des distributions marginales a plusieurs avantages. Un de ces avantages est qu'il permet d'estimer les différentes variables aléatoires séparément, par exemple en calculant l'estimation MMSE (Minimum Mean Square Error) ou en calculant l'estimateur MAP (Maximum a Posteriori) local. Si les fonctions marginales sont disponibles à chaque nœud, les positions peuvent être estimées localement. Un autre avantage est que les fonctions marginales fournissent une représentation des incertitudes et peuvent être utilisés dans la détection des ambiguïtés potentielles.

L'algorithme basé sur l'échange de messages que nous considérons est la propagation de croyance, qui est aussi appelé algorithme somme-produit. Dans ce qui suit, nous présentons cet algorithme.

### D.3.6 Algorithme de propagation de croyance (BP)

Cet algorithme peut être mis en œuvre de manière itérative et distribuée, où les messages sont échangés en parallèle. Il peut procéder comme suit. Tout d'abord, tous les messages sont initialisés à une valeur arbitraire, par exemple

$$m_{j,i}^{(0)}(\mathbf{x}_i) = 1. \tag{D.15}$$

Ensuite, le message du nœud $j$ au nœud $i$ à l'itération $l$ est calculé comme suit :

$$m_{j,i}^{(l)}(\mathbf{x}_i) = \alpha\int \Phi_j(\mathbf{x}_j)\Psi_{i,j}(\mathbf{x}_i,\mathbf{x}_j)\prod_{k\in\eta(j)\backslash i} m_{k,j}^{(l-1)}(\mathbf{x}_j)\mathrm{d}\mathbf{x}_j, \tag{D.16}$$

où $\alpha$ est une constante multiplicative introduite pour contribuer à la stabilité des calculs, et $\eta(j)\backslash i$ est l'ensemble des voisins du nœud $j$ sans le nœud $i$. La croyance du nœud $i$ à

l'itération $l$ est donnée par

$$b_i^{(l)}(\mathbf{x}_i) = \frac{1}{Z}\Phi_i(\mathbf{x}_i) \prod_{j\in\eta(i)} m_{j,i}^{(l)}(\mathbf{x}_i). \tag{D.17}$$

Dans les graphes sans boucles (graphes de type arbre), le nombre d'itérations nécessaires pour la convergence est égal au diamètre du graphe, et les croyances calculées à la convergence sont égales aux fonctions de distribution marginale.

#### Graphes avec boucles

Lorsque l'algorithme de propagation de croyance est appliqué aux graphes contenant des boucles, il est appelé 'loopy BP'. Dans ce cas, les croyances calculées sont des approximations des fonctions de distribution marginale.

### D.3.7 Propagation de croyance non-paramétrique

L'intégrale (D.16) peut être calculée analytiquement lorsque les variables aléatoires sont discrètes et ont un nombre fini d'états ou lorsque la distribution conjointe a posteriori est gaussienne. Lorsque ces conditions ne sont pas vérifiées, comme dans le cas du problème de la localisation, cette équation ne peut pas être calculée d'une façon analytique et doit être remplacée par une approximation. Un algorithme connu qui calcule des approximations des messages et des croyances en appliquant des techniques stochastiques et de Monte-Carlo est la BP non paramétrique (NBP). La principale complexité de cet algorithme est l'échantillonnage du produit des messages reçus à un nœud donné, et il existe un compromis entre la complexité et le coût de l'échange de données d'une part, et la précision des approximations numériques d'autre part.

### D.3.8 Algorithme NBP à deux phases et compensation des ambiguïtés

#### Compensation des ambiguïtés

L'ambiguïté dans la localisation coopérative, qui correspond à des réalisations géométriques erronées, est un problème fondamental qui peut entraîner des grandes erreurs d'estimation. Elle est due soit à l'absence de mesures nécessaires pour atteindre une identifiabilité globale, soit à la topologie du réseau et aux bruits dans les mesures. Une solution existante pour résoudre l'ambiguïté est d'exploiter les informations de connectivité entre les différents nœuds. Le fait que deux nœuds ne soient pas voisins donne l'information complémentaire qu'il est probable qu'il soient loin l'un de l'autre. Ici, nous mettrons l'accent sur l'exploitation de l'information de connectivité en appliquant l'estimation probabiliste. Dans ce cas, un modèle de connectivité probabiliste doit être considéré. Nous définissons la fonction

$$p_o(\mathbf{x}_i, \mathbf{x}_j) = Q\left(\frac{10 n_p \log_{10}(d_{i,j}/R)}{\sigma_{sh}}\right), \tag{D.18}$$

pour modéliser la probabilité que deux nœuds soient connectés. Nous avons établi une telle fonction en se basant sur la puissance moyenne reçue, et en supposant que les différents liens sont affectés par des composantes de masquage indépendantes et distribuées selon une loi log-normale. Les informations de connexion peuvent être exploitées par les algorithmes basés sur l'échange de messages en définissant une fonction de compatibilité entre deux nœuds $i$ et $j$ qui ne sont pas des voisins directs par

$$\Psi_{i,j}(\mathbf{x}_i, \mathbf{x}_j) = 1 - p_o(\mathbf{x}_i, \mathbf{x}_j). \tag{D.19}$$

En échangeant des messages entre tous les nœuds qui ne sont pas des voisins directs, le graphe du réseau sera complètement connecté, ce qui entraîne une très grande complexité et des coûts de communication élevés. Pour résoudre ce problème, l'échange des messages aura lieu jusqu'aux voisins d'ordre $k$, où deux nœuds sont des voisins d'ordre $k$ si le chemin le plus court qui les relie à une longueur de $k$ arêtes.

#### Algorithme NBP à deux phases

Nous avons développé un algorithme basé sur la NBP qui améliore la précision et réduit la complexité et la taille des messages échangés. Cet algorithme est appelé TP-NBP et il procède comme suit. Dans la première phase, la NBP est appliquée en échangeant des messages seulement entre les voisins directs et sans prendre en compte les informations de connectivité. Puis dans une deuxième phase, un nouvel algorithme basé sur l'estimation dans l'espace d'état discret est appliqué. Cet algorithme est composé des étapes suivantes :

1. Pour chaque nœud, nous identifions les modes de la croyance et nous construisons un petit ensemble de points constitué de ces modes et de quelques points autour de chaque mode.
2. A ce stade, chaque nœud dispose d'un petit ensemble de points. Nous appliquons la version discrète de la BP pour déterminer à nouveau les croyances, où les positions inconnues prennent des valeurs dans les ensembles construits. Dans cette étape, on exploite l'information de connectivité et on échange des messages entre les nœuds qui ne sont pas des voisins directs.

### D.3.9 Quelques résultats numériques

Dans cette section, nous validons les avantages de la TP-NBP au moyen d'un exemple de simulation de Monte-Carlo.

#### Exemple : Mesure de distances affectées de bruits additifs gaussiens

Nous considérons des réseaux constitués de 4 nœuds d'ancrage et 16 nœuds cibles déployées dans une zone de dimensions $20m \times 15m$. Les nœuds d'ancrage sont situés comme indiqué dans la Figure D.3. Les positions des nœuds cibles sont tirées d'une façon aléatoire selon une loi uniforme à l'intérieur des carrés de taille $2m \times 2m$, un nœud par carré. Les centres

des carrés sont également indiqués par la Figure D.3. Nous divisons les nœuds cibles en deux catégories : les nœuds internes qui sont à l'intérieur de l'enveloppe convexe des 4 nœuds d'ancrage et les nœuds périphériques qui sont en dehors de cette enveloppe convexe. Les nœuds périphériques ont des probabilités plus élevées d'avoir des ambiguïtés sur leurs positions estimées que les nœuds internes, car ils ont moins de voisins, en moyenne. Pour la connectivité, nous considérons le modèle probabiliste de (D.18) avec $n_p = 3$, $\sigma_{sh} = 8dB$ et le rayon de connectivité $R$ varie de 4 à $10m$. De plus, nous ne considérons que les réseaux rigides qui ne peuvent pas avoir de déformation continue (c.-à-d., un réseau généré aléatoirement n'est conservé que s'il est rigide). Les observations sont des mesures de distance affectées par des erreurs additives gaussiennes de variance $\sigma^2$.

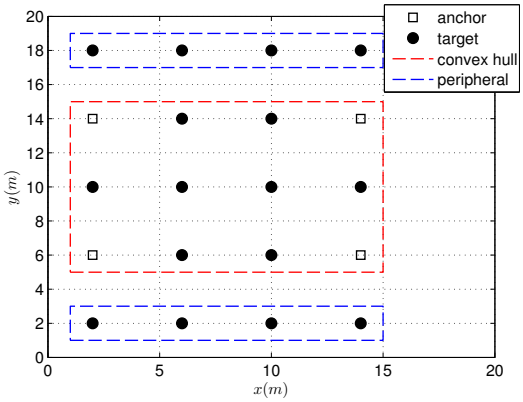



***Figure D.3*** — Positions des nœuds d'ancrage et des centres des carrés où les positions des nœuds cibles sont tirées selon une loi uniforme.

On considère une solution de référence centralisé qui est la solution WLS, où le point initial pour l'optimisation est obtenu en appliquant l'algorithme SDP adapté aux mesures bruitées. Cette solution sera appelée 'SDP + descent'.

Les Figures D.4 et D.5 montrent la variation de l'erreur quadratique moyenne (RMSE) avec la valeur de $R$ pour deux valeurs de la variance $\sigma^2$. La RMSE est obtenue en considérant 100 réalisations de réseaux et de bruits. Plusieurs méthodes sont comparées. NBP-MMSE et NBP-MAP considèrent la moyenne et le mode le plus probable de la fonction de croyance, respectivement, où les messages sont échangés entre les voisins directs seulement. NBP $k$-step correspond à l'application de la solution TP-BNP, où dans la deuxième phase, la version discrète de la BP est appliquée en utilisant la règle de somme-produit, et les messages sont échangés jusqu'aux voisins d'ordre $k$.

Plusieurs points peuvent être conclus de ces figures :
- NBP-MMSE est plus précise que NBP-MAP.
- NBP 1-step est plus précise que NBP-MMSE,.
- NBP 2-step et NBP 3-step améliore la précision pour les nœuds périphériques.
- Aucune amélioration n'est observée pour les valeurs élevées de $R$ puisque la probabilité d'avoir des ambiguïtés tend vers zéro.

Nous avons étudié un autre exemple où les mesures de distance sont affectées par une erreur additive avec une valeur aberrante pour modéliser les erreurs élevées. Les résultats des

simulations ont montré que les erreurs de l'algorithme TP-NBP peuvent être beaucoup plus faibles que ceux de l'algorithme NBP classique.

Nous avons aussi démontré, par des simulations de Monte-Carlo, que l'information sur la corrélation spatiale du masquage peut être utile pour résoudre les ambiguïtés.

## D.4 Poursuite de position basée sur les mesures de RSS

### D.4.1 Introduction

Dans ce chapitre, nous considérons que les terminaux mobiles (ou nœuds cibles) sont en mouvement et nous cherchons à suivre leurs positions au cours du temps. Comme dans le chapitre précédent, des solutions probabilistes bayésiennes sont élaborées, mais la coopération directe entre les stations mobiles n'est pas prise en compte. Notre objectif est d'exploiter la puissance du signal reçu (RSS) mesurée entre une station mobile et plusieurs stations de base (ou nœuds d'ancrage). Les obstacles dans l'environnement de propagation créent des évanouissements à long terme, ou phénomène de masquage (shadowing). Le masquage est généralement supposé suivre une distribution log-normale avec une corrélation spatiale.

Deux solutions de poursuite bayésienne sont développées dans ce chapitre pour exploiter efficacement les mesures et améliorer la précision en présence du masquage aléatoire :

- Dans la première solution, le masquage est suivi conjointement avec la position. A cet effet, on définit un modèle autorégressif de l'évolution temporelle du masquage avec le déplacement de la station mobile.
- Dans la deuxième solution, des cartes de masquages sont construites et mises à jour lors de la phase de poursuite. Cette solution permet de réduire l'effort nécessaire pour la méthode de fingerprinting, et les mesures réalisées par plusieurs stations mobiles se déplaçant dans la zone de déploiement peuvent être utilisées dans l'estimation des cartes.

Ces solutions sont implémentées en utilisant des méthodes de Monte Carlo séquentielles, connues sous le nom de filtres particulaires, et sont étudiées par des simulations dans plusieurs scénarios de déploiement.

Nous commençons par fournir la modélisation des mesures de RSS. Ensuite nous étudions la précision de la localisation lorsque les cartes de masquage sont parfaitement connues. Puis, la poursuite bayésienne est présentée, et la solution de poursuite conjointe est développée et appliquée à la poursuite des véhicules dans un système macro-cellulaire. Ensuite, la solution d'estimation des cartes est développée et appliquée à la localisation dans un environnement indoor et à la poursuite des trains.

### D.4.2 Modélisation des observations RSS

Nous considérons un déploiement d'un réseau composé de $N_{BS}$ stations de base de position connue. La mesure de RSS en décibel ($dB$) faite entre une station mobile à la position $\mathbf{x}$ et

la $i^{\text{ème}}$ station de base peut être écrite comme

$$y_i = P_i(\mathbf{x}) + \epsilon_i(\mathbf{x}) + e_i \tag{D.20}$$

où $P_i$ est une fonction déterministe qui tient compte de l'atténuation, la puissance rayonnée et les gains des antennes, $\epsilon_i$ est le masquage qui est invariant dans le temps, mais dépend de la position, et $e_i$ est une erreur rassemblant le masquage variable dans le temps causé par les obstacles mobiles et les erreurs d'estimation de RSS (par exemple, évanouissement à petite échelle, les effets de non-linéarité, etc.). Le masquage $\epsilon_i(\mathbf{x})$ est modélisée comme étant une réalisation d'une variable aléatoire gaussienne de moyenne nulle. Le masquage est aussi corrélé spatialement, c.-à-d., $E\{\epsilon_i(\mathbf{x}_p)\epsilon_i(\mathbf{x}_q)\} \neq E\{\epsilon_i(\mathbf{x}_p)\}E\{\epsilon_i(\mathbf{x}_q)\}$ où $\mathbf{x}_p$ et $\mathbf{x}_q$ sont deux positions quelconques. La corrélation du masquage à deux positions $\mathbf{x}_p$ et $\mathbf{x}_q$ est

$$E\{\epsilon_i(\mathbf{x}_p)\epsilon_i(\mathbf{x}_q)\} = \sigma_{sh}^2 \rho(\|\mathbf{x}_p - \mathbf{x}_q\|) \tag{D.21}$$

où $\rho$ est une fonction de corrélation isotrope, c.-à-d., ne dépend que de la distance entre les deux positions. Par définition $\rho(0) = 1$. Cette fonction de corrélation est supposée être connue pour toutes les stations de base.

Maintenant, nous développons deux modèles pour représenter le masquage : le premier décrit les cartes de masquage et relie la valeur de masquage à la position spatiale, et le deuxième décrit l'évolution temporelle du masquage pour une station mobile. Ces deux modèles seront utilisés dans les solutions développées par la suite.

### Modélisation des cartes de masquage

La station mobile est supposée résider dans une zone géographique $\mathcal{A} \subset \mathbb{R}^2$. Le masquage d'une station de base dans la zone $\mathcal{A}$ est une réalisation d'un champ aléatoire gaussien (GRF) que l'on appelle une carte de masquage. Une approximation de cette carte est donnée par

$$\epsilon_i(\mathbf{x}) \approx \sum_{k=1}^{L_{map}} \alpha_{i,k}\psi_k(\mathbf{x}). \tag{D.22}$$

où $\{\psi_k\}_{k=1}^{\infty}$ est une base complète définie sur $\mathcal{A}$, et $\{\alpha_{i,k}\}$ sont des coefficients distribuées selon une loi gaussienne.

$\alpha_i = [\alpha_{i,1}, \cdots, \alpha_{i,L_{map}}]^T$ est un vecteur gaussien ayant le vecteur $\mathbf{m}_i$ comme moyenne et $\mathbf{C}_i$ comme matrice de covariance. On applelle ce vecteur carte de masquage. Quand on dispose pas de l'information a priori sur la carte, $\mathbf{m}_i$ et $\mathbf{C}_i$ peuvent être calculés à partir des fonctions de base utilisées et du modèle de corrélation spatiale supposé.

Des mesures ont également montré que les masquages pour différentes stations de base peuvent être corrélés (c.-à-d., $E\{\epsilon_i(\mathbf{x})\epsilon_j(\mathbf{x})\} \neq E\{\epsilon_i(\mathbf{x})\}E\{\epsilon_j(\mathbf{x})\}$). Pour les $N_{BS}$ stations de base de la zone $\mathcal{A}$, nous définissons l'atlas de masquage comme étant l'ensemble des cartes de masquage obtenu par concaténation des vecteurs :

$$\Lambda = [\alpha_1^T, \cdots, \alpha_{N_{BS}}^T]^T. \tag{D.23}$$

On note $L_{atlas} = N_{BS} \times L_{map}$ la taille du vecteur atlas. Nous désignerons par $\mathbf{M}$ et $\Sigma$ sa moyenne et sa covariance, respectivement. Les mesures de RSS prises à des positions connues peuvent être utilisées pour mettre à jour $\mathbf{M}$ et $\Sigma$.

Pour une station mobile en mouvement, la corrélation spatiale est transformée en une corrélation temporelle. Ainsi, le masquage évolue avec le temps selon un processus gaussien. Maintenant, nous développons un modèle autorégressif (AR) pour décrire ce processus.

**Modèle autorégressif**

Nous considérons une station mobile qui se déplace et on désigne par $\mathbf{x}_k = [x_k, y_k]^T$ le vecteur de position à l'instant $kT$, où $k \in \mathbb{N}$ et $T$ est un pas d'échantillonnage. La corrélation du masquage à des instants $lT$ et $mT$ est

$$E\{\epsilon(\mathbf{x}_l)\epsilon(\mathbf{x}_m)\} = \sigma_{sh}^2 \rho(\|\mathbf{x}_l - \mathbf{x}_m\|). \tag{D.24}$$

Connaissant les positions $\mathbf{x}_{0:k} = [\mathbf{x}_0^T, \cdots, \mathbf{x}_k^T]^T$ et préalablement à toute observation, $[\epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_k)]^T$ est un vecteur gaussien de moyenne nulle. De plus, $p(\epsilon(\mathbf{x}_k)|\mathbf{x}_{0:k}, \epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_{k-1}))$ étant une distribution gaussienne, le processus $\epsilon(\mathbf{x}_k)$ peut être représenté par un modèle AR d'ordre-$k$ $AR(k)$ :

$$\epsilon(\mathbf{x}_k) = \mathbf{a}_k^T [\epsilon(\mathbf{x}_0), \cdots, \epsilon(\mathbf{x}_{k-1})]^T + \theta_k \tag{D.25}$$

où $\theta_k$ est une variable aléatoire gaussienne de variance $\sigma_{\theta_k}^2$. Les valeurs de $\mathbf{a}_k$ et $\sigma_{\theta_k}^2$ peuvent être mis à jour d'une manière récursive.

**Modèle AR pour plusieurs stations de base** Nous désignons par $\Omega_k = [\epsilon_1(\mathbf{x}_k), \cdots, \epsilon_{N_{BS}}(\mathbf{x}_k)]^T$ le vecteur de masquage des $N_{BS}$ stations de base à l'instant $k$. Avant toute observation, $\Omega_{0:k} = [\Omega_0^T, \cdots, \Omega_k^T]^T$ est un vecteur gaussien de moyenne nulle. Le processus gaussien $\Omega_k$ peut être également représenté par un modèle $AR(k)$ :

$$\Omega_k = \mathbf{A}_k^T \Omega_{0:k-1} + \Theta_k. \tag{D.26}$$

### D.4.3 Précision de la localisation lorsque le masquage est connu

Nous avons étudié l'effet de la connaissance des cartes de masquage sur la précision de la localisation statique dans un scénario de déploiement indoor. Les résultats des simulations ont montré que l'erreur d'estimation de position n'est pas très diminuée par rapport au cas où le masquage est inconnu. La justification de ce résultat est que, en raison du caractère aléatoire du masquage, des mesures à des points éloignés les uns des autres peuvent avoir des fonctions de distribution proches.

### D.4.4 La poursuite de la position

Dans cette section, nous traitons le problème de poursuite au cours du temps de la position d'une station mobile. La poursuite de position s'appuie sur les mesures obtenues à partir des

signaux radio ou les mesures d'un système de navigation inertielle (INS), en se basant sur un modèle de mouvement. Le modèle de mouvement décrit les caractéristiques du mouvement (par exemple, un piéton en indoor ne peut pas passer à travers les murs, un véhicule sur une route a une vitesse maximale limitée), et peut prendre en compte les lois cinématiques. Plus précisément, nous appliquons des algorithmes de poursuite bayésienne qui exploitent efficacement les mesures en mettant à jour, d'une manière récursive, la densité de probabilité a posteriori. Nous considérons deux scénarios : dans le premier, les cartes de masquage sont connues, et dans le deuxième, les cartes ne sont pas connues et le masquage est considéré comme une partie du vecteur d'état à estimer.

**Poursuite avec connaissance des cartes de masquage**

Nous avons considéré le même déploiement indoor que dans le cas du positionnement statique. Un modèle Markovien simple du mouvement est utilisé et la solution est mise en œuvre en utilisant des filtres particulaires. Les résultats de simulation ont montré que la précision est beaucoup améliorée par rapport au positionnement statique lorsque le masquage est connu. Nous avons aussi proposé un filtre particulaire régularisé qui permet de réduire le nombre de particules nécessaire pour obtenir une bonne précision.

**Poursuite conjointe de la position et du masquage**

Maintenant, nous supposons que le masquage est inconnu, mais sa distribution de probabilité et sa fonction de corrélation spatiale sont connues. Nous avons développé une solution de poursuite basée sur le filtrage bayésien, qui exploite efficacement les mesures en mettant à jour, de manière récursive, la distribution de probabilité a posteriori du vecteur d'état. La mise à jour est effectuée en prenant le masquage comme une partie de ce vecteur. Nous définissons le vecteur cinématique $\mathbf{c}_k$ à l'instant $kT$ comprenant la position $\mathbf{x}_k$ et éventuellement d'autres paramètres cinématiques (par exemple, vitesse). Ce vecteur est issu d'un processus de Markov connu de fonction de transition $p(\mathbf{c}_k|\mathbf{c}_{k-1})$ et de distribution initiale $p(\mathbf{c}_0)$. Le vecteur de masquage $\Omega_k$ est pris comme une partie du vecteur d'état caché $\mathbf{s}_k$ qui est définie par

$$\mathbf{s}_k = [\mathbf{c}_k^T \ , \ \Omega_k^T]^T. \tag{D.27}$$

Le processus de l'état n'est plus Markovien à cause de la corrélation du masquage, et il peut être décomposé comme suit :

$$p(\mathbf{s}_k|\mathbf{s}_{0:k-1}) = p(\mathbf{c}_k|\mathbf{c}_{k-1})p(\Omega_k|\mathbf{x}_{0:k}, \Omega_{0:k-1}). \tag{D.28}$$

Le vecteur $\Omega_k$ est représenté par le modèle AR précédemment décrit, ce qui permet le calcul de la moyenne et de la covariance de la distribution gaussienne $p(\Omega_k|\mathbf{x}_{0:k}, \Omega_{0:k-1})$.

A l'instant $kT$, la station mobile effectue des mesures de RSS, dénotées par le vecteur $\mathbf{y}_k$, avec $n_k$ stations de base. Le vecteur de mesures est égal à

$$\mathbf{y}_k = \mathbf{P}_k(\mathbf{x}_k) + \mathbf{J}_k\Omega_k + \mathbf{e}_k \tag{D.29}$$

où $\mathbf{P}_k$ est une fonction déterministe connue, $\mathbf{J}_k$ est une matrice de taille $n_k \times N_{BS}$ où $\mathbf{J}_k(i,j) = 1$ si la $i^{\text{ème}}$ mesure est effectuée avec la $j^{\text{ème}}$ station de base et 0 ailleurs, et $\mathbf{e}_k$ est une erreur gaussienne blanche par rapport au temps.

Le filtrage bayésien consiste à déterminer, de façon récursive dans le temps, la distribution a posteriori $p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k})$ afin d'appliquer un estimateur bayésien. Cette distribution est calculée selon

$$p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}) = p(\mathbf{s}_{0:k-1}|\mathbf{y}_{1:k-1})\frac{p(\mathbf{s}_k|\mathbf{s}_{0:k-1})p(\mathbf{y}_k|\mathbf{s}_k)}{p(\mathbf{y}_k|\mathbf{y}_{1:k-1})}. \tag{D.30}$$

Quand un processus d'ordre p ($AR(p)$) est considéré, $p(\mathbf{s}_k|\mathbf{s}_{0:k-1})$ est remplacée par $p(\mathbf{s}_k|\mathbf{s}_{k-p:k-1})$.

L'équation d'évolution du masquage (D.26) et l'équation d'observation (D.29) sont non-linéaires par rapport à $\mathbf{x}_k$, et par conséquent, la fonction à posteriori (D.30) ne peut pas être calculée analytiquement. Les filtres particulaires permettent le calcul numérique des solutions, avec un inconvénient de complexité : le nombre de particules doit en général augmenter de façon exponentielle avec la dimension du vecteur d'état. Dans notre cas, la dimension du vecteur d'état est élevée puisqu'il contient le masquage.

On peut remarquer que, conditionnellement à la connaissance de $\mathbf{x}_{0:k}$, les équations (D.26) et (D.29) sont linéaires par rapport à $\Omega_k$, ce qui permet d'appliquer un filtre de Kalman pour estimer le masquage. Ainsi, nous pouvons réduire le nombre de particules et limiter la complexité du filtre en appliquant une Rao-Blackwellisation, qui consiste à calculer la loi a posteriori d'une partie du vecteur d'état de façon analytique, dans notre cas avec le filtre de Kalman.

**Les résultats numériques :la poursuite de véhicules dans un système macro-cellulaire** Afin de montrer les améliorations apportées par l'algorithme, des simulations de Monte Carlo sont effectuées. A cet effet, nous considérons une trajectoire colinéaire avec une vitesse variable. Afin d'illustrer les avantages obtenus en utilisant des ordres AR supérieurs à 1, nous considérons également le cas d'un demi-tour.

Le déplacement du véhicule est modélisé par le processus de Markov linéaire (D.31). Le vecteur cinématique est $\mathbf{c}_k = [\mathbf{x}_k^T, \dot{\mathbf{x}}_k^T]^T$, et nous supposons que $\mathbf{a}_{k-1} = [a_{x,k-1}, a_{y,k-1}]^T$ est le vecteur accélération fourni par un accéléromètre et $\mathbf{q}_k$ est un vecteur gaussien qui tient compte des erreurs d'estimation de l'accélération. Nous prenons la covariance de $\mathbf{q}_k$ arbitrairement égale à $(0.5m/s^2)^2\mathbf{I}_2$.

$$\mathbf{c}_k = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{c}_{k-1} + \begin{bmatrix} 0 & 0 \\ T & 0 \\ 0 & 0 \\ 0 & T \end{bmatrix} (\mathbf{a}_{k-1} + \mathbf{q}_k) \tag{D.31}$$

Nous effectuons la poursuite avec une contrainte de carte, où la trajectoire appartient à une route droite avec deux voies. Nous considérons les deux trajectoires représentées par la Figure D.6. Trajectoire 1 est une ligne droite avec une vitesse moyenne de 57 $km/h$ et une vitesse maximale de 72 $km/h$. Trajectoire 2 contient un virage et les vitesses moyenne et

maximale sont 34 $km/h$ et 65 $km/h$, respectivement. Les positions de départ sont supposées être parfaitement connue. Pour le masquage, la fonction de corrélation est exponentielle avec $d_{corr} = 50m$. L'écart-type du masquage est $\sigma_{sh} = 8dB$.

Dans la Figure D.7, l'erreur quadratique moyenne de la position est tracée en fonction du temps pour Trajectoire 1 et pour un pas $T = 0.4$ secondes. Une grande amélioration peut être observée lorsque les mesures RSS sont exploitées. Ici, l'ordre AR-1 est optimal car la trajectoire est colinéaire. La nouvelle solution réduit la RMSE jusqu'à $10m$ par rapport à la solution dans laquelle le masquage est traité comme non-corrélé, mais les imprécisions restantes ne permettent pas d'éliminer son effet, et la RMSE reste élevée par rapport au cas où le masquage est parfaitement connu.

Pour Trajectoire 2, le pas est $T = 1$ seconde et la durée totale est de 30 secondes. Ainsi, un ordre égal à 30 est capable de prendre en compte tous les états précédents. Figure D.8 montre l'erreur quadratique moyenne pour ce trajet. A $t = 16$ secondes, lorsque le virage a lieu, l'erreur quadratique moyenne d'$AR(10)$ et $AR(30)$ devient inférieur à celle d'$AR(1)$. En effet, l'adjacence des deux parties de la trajectoire permet d'améliorer l'estimation du masquage en exploitant la corrélation. Quelques secondes après, la RMSE de l'$AR(10)$ commence à augmenter. Ceci s'explique par le fait que l'ordre n'est pas suffisant pour exploiter la corrélation entre le masquage à l'instant actuel et celui au début de la trajectoire.

### D.4.5 Estimation bayésienne des cartes

Les résultats des simulations présentées dans les sections précédentes ont montré que la connaissance du masquage peut considérablement améliorer la précision de la poursuite. La méthode classique pour construire les cartes est la méthode de fingerprinting, dans laquelle, les mesures sont collectées à des positions connues et enregistrées dans une base de données. Le problème avec cette solution est qu'elle nécessite beaucoup de temps et d'efforts pour construire la base de données. Dans cette section, nous proposons une solution pour l'estimation en ligne des cartes qui ne requière pas la connaissance parfaite des positions des stations mobiles. Les stations mobiles peuvent avoir des capacités de localisation autres que les mesures de RSS. Cette solution permet d'affiner et de mettre à jour les cartes de façon collaborative en exploitant les mesures effectuées par plusieurs stations mobiles. Cette solution procède par la mise à jour de la distribution de probabilité a posteriori de l'atlas de masquage :

$$p(\mathbf{\Lambda}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k}) = \int p(\mathbf{\Lambda}|\mathbf{s}_{0:k}, \mathbf{y}_{1:k}) p(\mathbf{s}_{0:k}|\mathbf{y}_{1:k}, \mathbf{z}_{1:k}) \mathrm{d}\mathbf{s}_{0:k} \tag{D.32}$$

où $\mathbf{y}_k$ sont les mesures de RSS et $\mathbf{z}_k$ sont d'autres mesures de positionnement, tels que ToA, INS et GNSS. L'équation (D.32) ne peut pas être calculée sous une forme analytique et les méthodes de Monte Carlo sont utilisées pour l'approximer numériquement. Nous avons appliqué cette solution à la poursuite en indoor et à la poursuite des trains via des simulations, et de grandes améliorations de précision ont été observées.

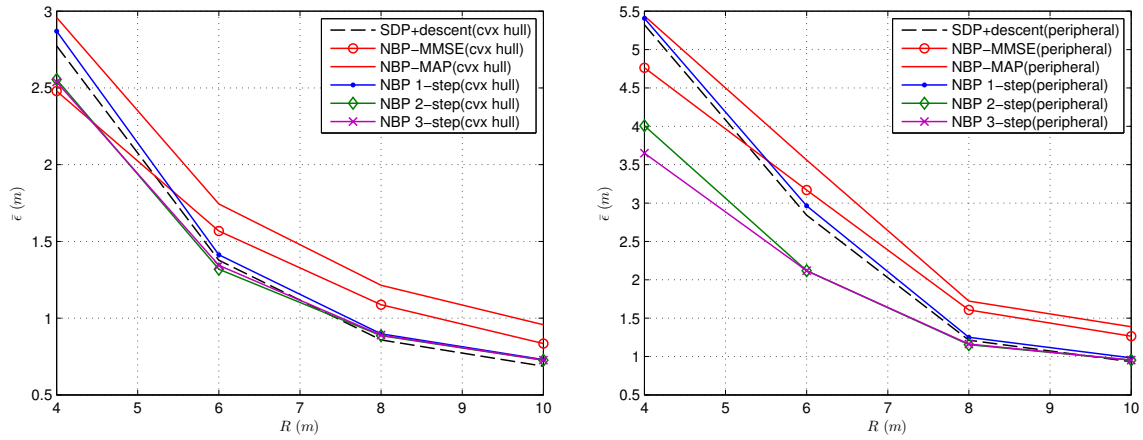



***Figure D.4*** — Variation de l'erreur quadratique moyenne avec $R$. $\sigma = 1.0$.

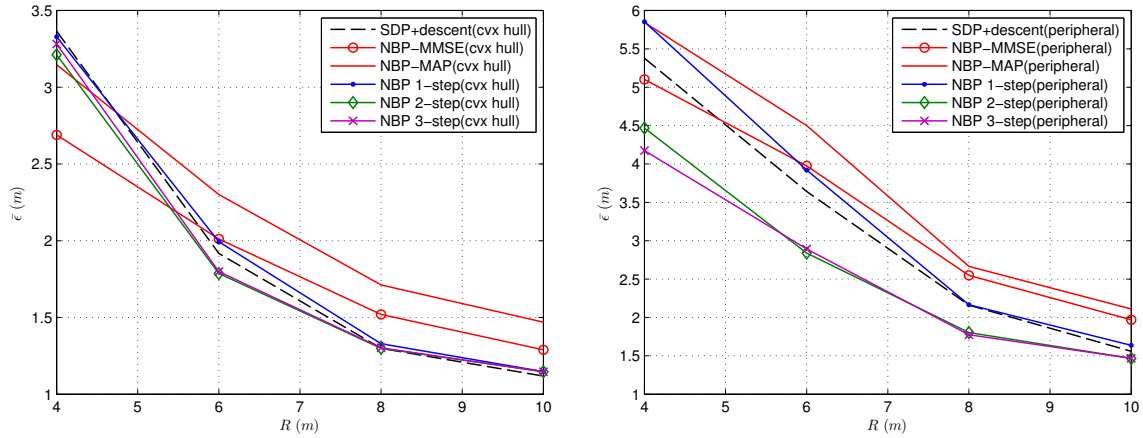



***Figure D.5*** — Variation de l'erreur quadratique moyenne avec $R$. $\sigma = 1.5$.

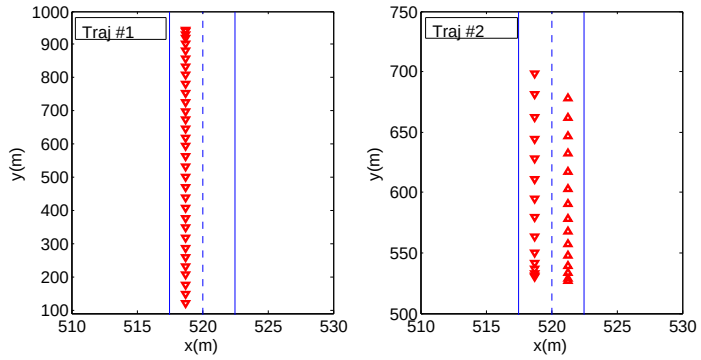



***Figure D.6*** — Les deux trajectoires considérées.

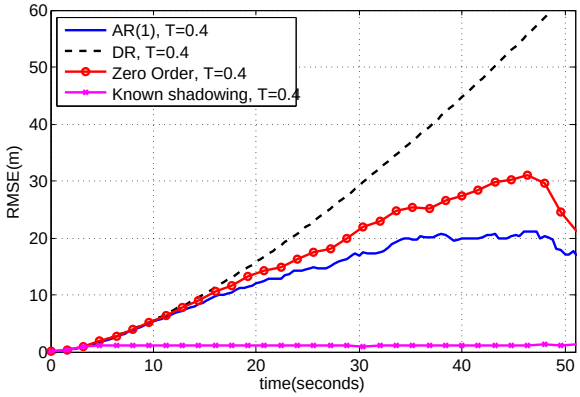



***Figure D.7*** — RMSE de la poursuite de position pour Trajectoire 1.

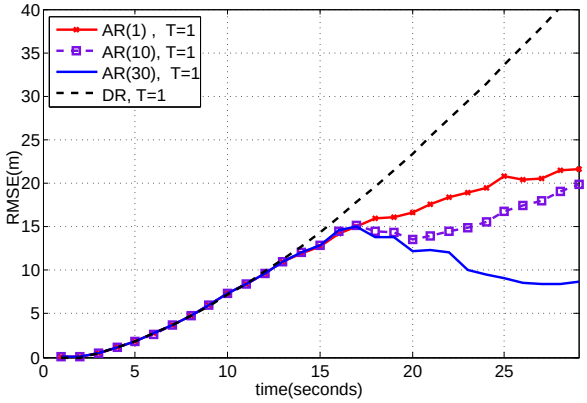



***Figure D.8*** — RMSE de la poursuite de position pour Trajectoire 2.

# Bibliography


[1] D. Goldenberg, A. Krishnamurthy, W. Maness, Y. Richard, Y. Young, A. Morse, A. Savvides, and B. Anderson, "Network Localization in Partially Localizable Networks," in *Proc. IEEE International Conference on Computer Communications, 2005*, ser. INFOCOM'05, Mar. 2005, pp. 313–326.

[2] H. Noureddine, D. Castelain, and R. Pyndiah, "Cooperative Network Localizability via Semidefinite Programming," in *Proc. IEEE Personal, International Symposium on Indoor and Mobile Radio Communications*, ser. PIMRC'011, Sep. 2011.

[3] H. Noureddine, N. Gresset, D. Castelain, and R. Pyndiah, "A New Variant of Nonparametric Belief Propagation for Self-Localization," in *Proc. IEEE Int. Conf. on Telecommunications*, ser. ICT'010, Apr. 2010, pp. 822–827.

[4] ——, "Auto-Regressive Modeling of the Shadowing for RSS Mobile Tracking," in *Proc. IEEE International Conference on Communications*, ser. ICC'011, Jun. 2011, pp. 1–5.

[5] H. Noureddine, D. Castelain, and R. Pyndiah, "Train Tracking and Shadowing Estimation based on Received Signal Strength," in *Proceedings of the Third International Conference on Communication Technologies for Vehicles*, ser. Nets4Cars/Nets4Trains'11. Springer-Verlag, 2011, pp. 23–33.

[6] N. Samama, *Global Positioning : Technologies and Performance.* Hoboken, NJ, USA : John Wiley, 2008.

[7] C. Hegarty and E. Chatre, "Evolution of the Global Navigation SatelliteSystem (GNSS)," *Proc. IEEE*, vol. 96, no. 12, pp. 1902 –1917, Dec. 2008.

[8] "What is galileo ?" http://www.esa.int/esaNA/galileo.html.

[9] M. Grewal, "Space-based augmentation for global navigation satellite systems," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 59, no. 3, pp. 497–503, Mar. 2012.

[10] T. Murphy and T. Imrich, "Implementation and Operational Use of Ground-Based Augmentation Systems (GBASs) : A Component of the Future Air Traffic Management System," *Proc. IEEE*, vol. 96, no. 12, pp. 1936 –1957, Dec. 2008.

[11] S. Feng and C. L. Law, "Assisted GPS and its impact on navigation in intelligent transportation systems," in *Proc. IEEE International Conference on Intelligent Transportation Systems*, 2002, pp. 926–931.

[12] M. Rabinowitz and J. Spilker, J.J., "A new positioning system using television synchronization signals," *IEEE Trans. Broadcast.*, vol. 51, no. 1, pp. 51–61, Mar. 2005.

[13] ——, "Position location using global positioning signals augmented by broadcast television signals," U.S. Patent US 7 733 270, Jun., 2010.

[14] G. Opshaug, D. Burgess, G. Flammer, S. Furman, A. Lee, M. Rabinowitz, and H. Samra, "Wide-lane pseudorange measurements using FM signals," U.S. Patent US 7 498 873, Mar., 2009.

[15] J. J. Caffery, *Wireless Location in CDMA Cellular Radio Systems.* Norwell, MA, USA : Kluwer Academic Publishers, 1999.

[16] A. 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.

[17] G. Sun, J. Chen, W. Guo, and K. Liu, "Signal processing techniques in network-aided positioning : a survey of state-of-the-art positioning designs," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 12–23, Jul. 2005.

[18] F. Evennou, "Techniques et technologies de localisation avancées pour terminaux mobiles dans les environnements indoor," Ph.D. dissertation, Université Joseph Fourier, 2007.

[19] "Skyhook's location technology." http://www.skyhookwireless.com/.

[20] "Navizon location solutions." http://www.navizon.com/.

[21] N. Patwari, J. Ash, S. Kyperountas, I. Hero, A.O., R. Moses, and N. Correal, "Locating the nodes : cooperative localization in wireless sensor networks," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 54–69, Jul. 2005.

[22] M. Bouet and G. Pujolle, "L-VIRT : Range-free 3-D localization of RFID tags based on topological constraints," *Comput. Commun.*, vol. 32, pp. 1485–1494, Aug. 2009.

[23] "Delivrable D1.1 : Scenarios and Parameters ," Project deliverable, 2011, FP7-ICT-2009-4 WHERE2.

[24] M. Laaraiedh, L. Yu, S. Avrillon, and B. Uguen, "Comparison of Hybrid Localization Schemes using RSSI, TOA, and TDOA," in *Proc. 17th European Wireless Conference*, Apr. 2011.

[25] "Enhanced 911." http://transition.fcc.gov/pshs/services/911-services/.

[26] P. Bellavista, A. Kupper, and S. Helal, "Location-Based Services : Back to the Future," *IEEE Pervasive Comput.*, vol. 7, no. 2, pp. 85–89, Apr./Jun. 2008.

[27] W. Zhang, X. Cui, D. Li, D. Yuan, and M. Wang, "The location privacy protection research in location-based service," in *Proc. IEEE International Conference on Geoinformatics*, Jun. 2010, pp. 1–4.

[28] L. Brunel, M. Plainchault, N. Gresset, A. Dammann, C. Mensing, and R. Raulefs, "Inter-cell interference coordination and synchronization based on location information," in *Proc. IEEE Workshop on Positioning Navigation and Communication*, ser. WPNC'010, Mar. 2010, pp. 224–232.

[29] S.-J. Kim, E. Dall'Anese, and G. Giannakis, "Cooperative Spectrum Sensing for Cognitive Radios Using Kriged Kalman Filtering," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Feb. 2011.

[30] I. Stojmenovic, "Position-based routing in ad hoc networks," *IEEE Commun. Mag.*, vol. 40, no. 7, pp. 128–134, Jul. 2002.

[31] Y. Nasser, H. Farhat, and J.-F. Helard, "Positioning information based technique in cooperative mimo-ofdm systems," in *Proc. IEEE Personal, International Symposium on Indoor and Mobile Radio Communications*, ser. PIMRC'09, Sep. 2009, pp. 2695–2699.

[32] S. Mumtaz, J. Bastos, J. Rodriguez, and C. Verikoukis, "Adaptive beamforming for OFDMA using positioning information," in *Wireless Advanced (WiAd), 2011*, Jun. 2011, pp. 7–14.

[33] L. Cong and W. Zhuang, "Hybrid TDOA/AOA mobile user location for wideband CDMA cellular systems," *Wireless Communications, IEEE Transactions on*, vol. 1, no. 3, pp. 439–447, Jul. 2002.

[34] S. M. Kay, *Fundamentals of Statistical Signal Processing : Estimation Theory.* Upper Saddle River, NJ, USA : Prentice-Hall, Inc., 1993.

[35] E. Robinson and A. H. Quazi, "Effect of sound-speed profile on differential time-delay estimation," *J. Acoust. Soc. Am.*, vol. 77, no. 3, pp. 1086–1090, Mar. 1985.

[36] D. Dardari, A. Conti, U. Ferner, A. Giorgetti, and M. Win, "Ranging With Ultrawide Bandwidth Signals in Multipath Environments," *Proc. IEEE*, vol. 97, no. 2, pp. 404–426, Feb. 2009.

[37] I. Güvenç and C.-C. Chong, "A Survey on TOA Based Wireless Localization and NLOS Mitigation Techniques," *IEEE Commun. Surveys Tuts.*, vol. 11, no. 3, pp. 107–124, quarter 2009.

[38] B. Denis and N. Daniele, "NLOS ranging error mitigation in a distributed positioning algorithm for indoor UWB ad-hoc networks," in *Proc. IEEE International Workshop on Wireless Ad-Hoc Networks*, May/Jun. 2004, pp. 356–360.

[39] T. Rappaport, *Wireless Communications : Principles and Practice.* Upper Saddle River, NJ, USA : Prentice Hall PTR, 2001.

[40] M. Laaraiedh, S. Avrillon, and B. Uguen, "Enhancing Positioning Accuracy through Direct Position Estimators Based on Hybrid RSS Data Fusion," in *Proc. IEEE Vehicular Technology Conference*, ser. VTC Spring 2009, Apr. 2009, pp. 1–5.

[41] P. Bahl and V. Padmanabhan, "RADAR : an in-building RF-based user location and tracking system ," in *Proc. IEEE International Conference on Computer Communications*, ser. INFOCOM'00, vol. 2, 2000, pp. 775–784.

[42] H. Laitinen, J. Lahteenmaki, and T. Nordstrom, "Database correlation method for GSM location," in *Proc. IEEE Vehicular Technology Conference*, ser. VTC Spring 2001, vol. 4, 2001, pp. 2504–2508.

[43] M. Triki and D. Slock, "Mobile Localization for NLOS Propagation," in *Proc. IEEE Personal, International Symposium on Indoor and Mobile Radio Communications*, ser. PIMRC'07, Sep. 2007, pp. 1–4.

[44] B. Ferris, D. Hähnel, and D. Fox, "Gaussian Processes for Signal Strength-Based Location Estimation," in *Robotics : Science and Systems*, G. S. Sukhatme, S. Schaal, W. Burgard, and D. Fox, Eds. The MIT Press, 2006.

[45] X. Chai and Q. Yang, "Reducing the Calibration Effort for Probabilistic Indoor Location Estimation," *IEEE Trans. Mobile Comput.*, vol. 6, no. 6, pp. 649–662, Jun. 2007.

[46] B. Denis, L. Ouvry, B. Uguen, and F. Tchoffo-Talom, "Advanced bayesian filtering techniques for uwb tracking systems in indoor environments," in *Proc. IEEE International Conference on Ultra-Wideband*, ser. ICUWB'05, Sep. 2005, pp. 638–643.

[47] N. Patwari, I. Hero, A.O., M. Perkins, N. Correal, and R. O'Dea, "Relative Location Estimation in Wireless Sensor Networks," *IEEE Trans. Signal Process.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[48] M. Win, A. Conti, S. Mazuelas, Y. Shen, W. Gifford, D. Dardari, and M. Chiani, "Network localization and navigation via cooperation," *IEEE Commun. Mag.*, vol. 49, no. 5, pp. 56–62, May 2011.

[49] I. Güvenç, C. Chong, F. Watanabe, and H. Inamura, "NLOS Identification and Weighted Least-Squares Localization for UWB Systems Using Multipath Channel Statistics," *EURASIP J. Adv. Signal Process.*, vol. 2008, Jan. 2008.

[50] K. Das and H. Wymeersch, "Censored Cooperative Positioning for Dense Wireless Networks," in *Proc. IEEE Personal, International Symposium on Indoor and Mobile Radio Communications Workshops*, Sep. 2010, pp. 262–266.

[51] B. Denis, M. Maman, and L. Ouvry, "On the Scheduling of Ranging and Distributed Positioning Updates in Cooperative IR-UWB Networks," in *Proc. IEEE International Conference on Ultra-Wideband*, ser. ICUWB'09, Sep. 2009, pp. 370–375.

[52] D. Moore, J. Leonard, D. Rus, and S. Teller, "Robust Distributed Network Localization with Noisy Range Measurements," in *ACM Conference on Embedded Networked Sensor Systems*, Nov. 2004, pp. 50–61.

[53] A. Kannan, B. Fidan, and G. Mao, "Robust Distributed Sensor Network Localization Based on Analysis of Flip Ambiguities," in *Proc. IEEE Global Communications Conference*, ser. GLOBECOM'08, Nov./Dec. 2008, pp. 1–6.

[54] B. Anderson, P. Belhumeur, T. Eren, D. Goldenberg, A. Morse, W. Whiteley, and Y. Yang, "Graphical properties of easily localizable sensor networks," *Wirel. Netw.*, vol. 15, no. 2, pp. 177–191, Feb. 2009.

[55] A. Ihler, I. Fisher, J.W., R. Moses, and A. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," *IEEE J. Sel. Areas Commun.*

[56] B. Hendrickson, "Conditions for Unique Graph Realizations," *SIAM J. Comput.*, vol. 21, no. 1, pp. 65–84, Feb. 1992.

[57] D. Jacobs and B. Hendrickson, "An Algorithm for Two-dimensional Rigidity Percolation : The Pebble Game," *J. Comput. Phys.*, vol. 137, no. 2, pp. 346–365, Nov. 1997.

[58] W. Whiteley, "Rigidity and scene analysis," in *Handbook of discrete and computational geometry*, J. E. Goodman and J. O'Rourke, Eds. Boca Raton, FL, USA : CRC Press, Inc., 1997, pp. 893–916.

[59] R. Connelly, "Generic Global Rigidity," *Discrete Comput. Geom.*, vol. 33, no. 4, pp. 549–563, Apr. 2005.

[60] B. Jackson, "Notes on the rigidity of graphs," Oct. 2007.

[61] S. Gortler, A. Healy, and D. Thurston, "Characterizing Generic Global Rigidity," *http ://arxiv.org/abs/0710.0926*, 2007.

[62] T. Eren, D. K. Goldenberg, W. Whiteley, Y. R. Yang, A. S. Morse, B. D. O. Anderson, and P. N. Belhumeur, "Rigidity, Computation, and Randomization in Network Localization," in *Proc. IEEE International Conference on Computer Communications*, ser. INFOCOM'04, Mar. 2004, pp. 2673–2684.

[63] G. Rote, F. Santos, and I. Streinu, "Pseudo-triangulations — a survey," in *Surveys on Discrete and Computational Geometry—Twenty Years Later*, ser. Contemporary Mathematics, R. P. Eli Goodman, János Pach, Ed. American Mathematical Society, Dec. 2008.

[64] L. Asimov and B. Roth, "The rigidity of graphs. II." *J. Math. Anal. Appl.*, vol. 68, pp. 171–190, 1979.

[65] H. Gluck, "Almost all simply connected closed surfaces are rigid," in *Geometric Topology*, ser. Lecture Notes in Mathematics, 1975, vol. 438, pp. 225–239.

[66] G. Laman, "On graphs and rigidity of plane skeletal structures," *J. Engineering Math.*, vol. 4, pp. 331–340, 1970.

[67] B. Jackson and T. Jordán, "Connected rigidity matroids and unique realizations of graphs," *J. Comb. Theory Ser. B*, vol. 94, no. 1, pp. 1–29, May 2005.

[68] S. Even and E. Tarjan, "Network Flow and Testing Graph Connectivity," *SIAM J. Comput.*, vol. 4, no. 4, pp. 507–518, 1975.

[69] J. Saxe, "Embeddability of weighted graphs in k-space is strongly NP-hard," Computer Science Department, Carnegie Mellon University, Tech. Rep., 1979.

[70] Z. Yang, Y. Liu, and X. Li, "Beyond Trilateration : On the Localizability of Wireless Ad Hoc Networks," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1806–1814, Dec. 2010.

[71] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I.* Wiley-Interscience, 2001.

[72] A. Savvides, W. Garber, R. Moses, and M. Srivastava, "An Analysis of Error Inducing Parameters in Multihop Sensor Node Localization," *IEEE Trans. Mobile Comput.*, vol. 4, no. 6, pp. 567–577, Nov. 2005.

[73] Y. Shen, H. Wymeersch, and M. Win, "Fundamental Limits of Wideband Localization : Part II : Cooperative Networks," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 498–5000, Oct. 2010.

[74] T. Rothenberg, "Identification in Parametric Models," *Econometrica*, vol. 39, no. 3, pp. 577–591, May 1971.

[75] P. Biswas, "Semidefinite Programming Approaches to Distance Geometry Problems," Ph.D. dissertation, Department of Electrical Engineering, Stanford University., 2007.

[76] A. M. So and Y. Ye, "Theory of Semidefinite Programming for Sensor Network Localization," in *ACM-SIAM Symposium on Discrete Algorithms*, Jan. 2005, pp. 405–414.

[77] P. Biswas, T. Lian, T. 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.

[78] Z. Zhu, A. So, and Y. Ye, "Universal rigidity : Towards accurate and efficient localization of wireless networks," in *Proc. IEEE International Conference on Computer Communications*, ser. INFOCOM'010, Mar. 2010, pp. 1–9.

[79] P. Agrawal and N. Patwari, "Correlated Link Shadow Fading in Multi-hop Wireless Networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4024–4036, Aug. 2009.

[80] N. Patwari and P. Agrawal, "Effects of Correlated Shadowing : Connectivity, Localization, and RF Tomography," in *Proc. IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN'08, Apr. 2008, pp. 82–93.

[81] N. Patwari and I. Hero, A.O., "Manifold learning algorithms for localization in wireless sensor networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP'04, May 2004, pp. 857–860.

[82] R. Moses, D. Krishnamurthy, and R. Patterson, "A Self-Localization Method for Wireless Sensor Networks," *Eurasip J. on Applied Signal Processing, Special Issue on Sensor Networks*, vol. 2003, no. 4, pp. 148–158, 2003.

[83] G. Destino and G. De Abreu, "Weighing strategy for network localization under scarce ranging information," *IEEE Trans. Wireless Commun.*, vol. 8, no. 7, pp. 3668–3678, Jul. 2009.

[84] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from Mere Connectivity," in *Proc. ACM International Symposium on Mobile Ad Hoc Networking & Computing*, 2003, pp. 201–212.

[85] J. Costa, N. Patwari, and A. O. H. Iii, "Distributed Weighted-Multidimensional Scaling for Node Localization in Sensor Networks," *ACM Trans. Sen. Netw.*, vol. 2, no. 1, pp. 39–64, 2005.

[86] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proc. ACM International Workshop on Wireless Sensor Networks and Applications*, ser. WSNA'02, 2002, pp. 112–121.

[87] A. Kannan, G. Mao, and B. Vucetic, "Simulated Annealing based Wireless Sensor Network Localization with Flip Ambiguity Mitigation," in *Proc. IEEE Vehicular Technology Conference*, ser. VTC Srping 2006, vol. 2, May 2006, pp. 1022–1026.

[88] V. Savic and S. Zazo, "Nonparametric Boxed Belief Propagation for Localization in Wireless Sensor Networks," in *Proc. IEEE International Conference on Sensor Technologies and Applications*, Jun. 2009, pp. 520–525.

[89] C. Pedersen, T. Pedersen, and B. Fleury, "A Variational Message Passing Algorithm for Sensor Self-Localization in Wireless Networks," in *Proc. IEEE International Symposium on Information Theory*, Jul. 2011.

[90] M. Welling and J. Lim, "A distributed message passing algorithm for sensor localization," in *Proc. International Conference on Artificial Neural Networks*, ser. ICANN'07. Berlin, Heidelberg : Springer-Verlag, 2007, pp. 767–775.

[91] D. Niculescu and B. Nath, "Ad hoc positioning system (APS)," in *Proc. IEEE Global Communications Conference*, ser. GLOBECOM'01, vol. 5, 2001, pp. 2926–2931.

[92] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," in *Proceedings of the General Track of the annual conference on USENIX.*, 2002, pp. 317–327.

[93] S. Severi, G. Abreu, G. Destino, and D. Dardari, "Efficient and accurate localization in multihop networks," in *Conference Record of Forty-Third Asilomar Conference on Signals, Systems and Computers*, ser. ACSSC'09, Nov. 2009, pp. 1071–1076.

[94] K. Whitehouse and D. Culler, "A robustness analysis of multi-hop ranging-based localization approximations," in *Proc. IEEE International Conference on Information Processing in Sensor Networks*, ser. IPSN'06, 2006, pp. 317–325.

[95] S. Severi, G. Abreu, and D. Dardari, "A Quantitative Comparison of Multihop Localization Algorithms," in *Proc. IEEE Workshop on Positioning Navigation and Communication*, ser. WPNC'010, Mar. 2010, pp. 200–205.

[96] A. Savvides, C. Han, and M. Strivastava, "Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors," in *Proc. International Conference on Mobile Computing and Networking.* ACM, 2001, pp. 166–179.

[97] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in Distributed Ad-Hoc Wireless Sensor Networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP'01, 2001.

[98] Y. Shang and W. Ruml, "Improved MDS-Based Localization," in *Proc. IEEE International Conference on Computer Communications*, ser. INFOCOM'04, vol. 4, Mar. 2004, pp. 2640–2651.

[99] S. Capkun, M. Hamdi, and J.-P. Hubaux, "Gps-free positioning in mobile ad-hoc networks," in *Proc. IEEE 35th Annual Hawaii International Conference on System Sciences*, ser. HICSS'01, Jan. 2001.

[100] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for Dimensionality Reduction and Data Representation," *Neural Computation*, vol. 15, pp. 1373–1396, 2002.

[101] L. Doherty, K. Pister, and L. El Ghaoui, "Convex position estimation in wireless sensor networks," in *Proc. IEEE International Conference on Computer Communications*, ser. INFOCOM'01, vol. 3, 2001, pp. 1655–1663.

[102] N. Priyantha, H. Balakrishnan, E. Demaine, and S. Teller, "Anchor-free distributed localization in sensor networks," in *SenSys.* ACM, 2003, pp. 340–341.

[103] S. Basu and Y. Bresler, "The Stability of Nonlinear Least Squares Problems and the Cramér-Rao Bound," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3426–3436, Dec. 2000.

[104] E. L. Lehmann, *Elements of Large-Sample Theory.* New York : Springer, 1998.

[105] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms.* Cambrdige University Press, 2003.

[106] C. M. Bishop, *Pattern Recognition and Machine Learning.* Secaucus, NJ, USA : Springer-Verlag New York, Inc., 2006.

[107] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference.* Hanover, MA, USA : Now Publishers Inc., 2008.

[108] Y. Weiss, "Belief Propagation and Revision in Networks with Loops," MIT, Cambridge, MA, USA, Tech. Rep.

[109] E. Sudderth, A. Ihler, W. Freeman, and A. Willsky, "Nonparametric Belief Propagation," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*, vol. 1, Jun. 2003, pp. 605–612.

[110] A. Ihler, E. Sudderth, W. Freeman, and A. Willsky, "Efficient multiscale sampling from products of gaussian mixtures," in *Neural Information Processing Systems Conference.* MIT Press, 2003.

[111] V. Savic, H. Wymeersch, F. Penna, and S. Zazo, "Optimized edge appearance probability for cooperative localization based on tree-reweighted nonparametric belief propagation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, ser. ICASSP'011, May 2011, pp. 3028–3031.

[112] V. Savic and S. Zazo, "Pseudo-junction tree method for cooperative localization in wireless sensor networks," in *Proc. IEEE International Conference on Information Fusion*, ser. FUSION'011, Jul. 2010, pp. 1–8.

[113] ——, "Nonparametric belief propagation based on spanning trees for cooperative localization in wireless sensor networks," in *Proc. IEEE Vehicular Technology Conference*, ser. VTC Fall 2010, Sep. 2010, pp. 1–5.

[114] M. Carreira-Perpinñán, "Mode-finding for mixtures of Gaussian distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1318–1323, Nov. 2000.

[115] D. Schieferdecker and M. Huber, "Gaussian mixture reduction via clustering," in *Proc. IEEE International Conference on Information Fusion*, ser. FUSION'09, Jul. 2009, pp. 1536–1543.

[116] J. Yedidia, W. Freeman, and Y. Weiss, "Bethe free energy, kikuchi approximations and belief propagation algorithms," 2000.

[117] C. Pedersen, T. Pedersen, and B. Fleury, "Exploiting network topology information to mitigate ambiguities in VMP localization," in *Proc. IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, Dec. 2011, pp. 57–60.

[118] A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks," in *Proc. IEEE Hawaii International Conference on System Sciences*, ser. HICSS'02, Jan. 2002, pp. 3866–3875.

[119] A. Goldsmith, L. Greenstein, and G. Foschini, "Error statistics of real-time power measurements in cellular channels with multipath and shadowing," *IEEE Trans. Veh. Technol.*, vol. 43, no. 3, pp. 439 –446, Aug. 1994.

[120] M. Gudmundson, "Correlation model for shadow fading in mobile radio systems," *Electronics Letters*, vol. 27, no. 23, pp. 2145–2146, 7 1991.

[121] T. Sorensen, "Slow fading cross-correlation against azimuth separation of base stations," *Electronics Letters*, vol. 35, no. 2, pp. 127 –129, Jan. 1999.

[122] H. Koorapaty, "Barankin bounds for position estimation using received signal strength measurements," in *Proc. IEEE Vehicular Technology Conference*, ser. VTC Spring 2004, vol. 5, May 2004, pp. 2686–2690.

[123] A. Doucet, "On sequential simulation-based methods for bayesian filtering," Tech. Rep., 1998.

[124] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund, "Particle filters for positioning, navigation, and tracking," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 425–437, Feb. 2002.

[125] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," in *IEE Proc. Radar and Signal Processing*, vol. 140, no. 2, Apr. 1993, pp. 107–113.

[126] K.-J. Yang and Y.-R. Tsai, "Location tracking in mobile networks under correlated shadowing effects," in *Proc. IEEE Wireless Communications and Networking Conference*, ser. WCNC'09, Apr. 2009, pp. 1 –5.

[127] J. Guillet and F. LeGland, "Using noisy georeferenced information sources for navigation and tracking," in *Proc. IEEE Nonlinear Statistical Signal Processing Workshop*, Sep. 2006, pp. 156–159.

[128] H. Durrant-Whyte and T. Bailey, " Simultaneous Localisation and Mapping (SLAM) : Part I The Essential Algorithms," *IEEE Robot. Autom. Mag.*, vol. 2, 2006.

[129] H. Rue and L. Held, *Gaussian Markov Random Fields : Theory and Applications*, ser. Monographs on Statistics and Applied Probability. London : Chapman & Hall, 2005, vol. 104.

[130] M. McGuire, K. N. Plataniotis, and A. N. Venetsanopoulos, "Environment and movement model for mobile terminal location tracking," *Wirel. Pers. Commun.*, vol. 24, no. 4, pp. 483–505, 2003.

[131] 3GPP, "TR 25.814, Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA) (Release 7)," Sep. 2006.

[132] M. Tao and K.-N. Lau, "Location Determination and Location Tracking in Wireless Networks," U.S. Patent US 024 633A1, Nov., 2005.

[133] A. Mirabadi, N. Mort, and F. Schmid, "Application of sensor fusion to railway systems," in *IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems.*, Dec. 1996, pp. 185–192.

[134] P. Ernest, R. Mazl, and L. Preucil, "Train locator using inertial sensors and odometer," in *Proc. IEEE Intelligent Vehicles Symposium*, Jun. 2004, pp. 860–865.

## Abstract

Advancements in information technology and communication systems enabled the development of a wide variety of location based applications and services. The global navigation satellite systems are among the fundamental localization solutions. In harsh environments (e.g., urban canyons and indoor areas), these solutions do not provide a good accuracy or even become unavailable. In order to offer accurate and ubiquitous localization solutions, wireless communication systems have been considered, where several location dependent parameters of the transmitted signals can be measured and exploited (e.g., the time-of-arrival (ToA), the received signal strength (RSS)). In this work, the topic of wireless localization is explored from a statistical signal processing perspective with a focus on two axes. The first axis is cooperative localization applied to ad-hoc networks, where the nodes perform pair-wise ranging measurements (i.e., ToA or RSS) between each other in order to simultaneously estimate their positions. The unique solvability conditions are studied based on the two approaches of graph rigidity and semidefinite programming, and the identifiability conditions are derived. Location estimation solutions are considered with a focus on probabilistic estimation and its application in Markov random fields using the nonparametric belief propagation (NBP) algorithm. The second axis is mobile terminals tracking based on RSS measurements. These measurements are affected by a shadowing phenomenon. The improvement brought by the knowledge of the shadowing maps to the position estimation accuracy is studied. The classical solution for obtaining these maps is fingerprinting, which can be costly in time and effort. Solutions are developed to overcome these difficulties. Several solutions are proposed and investigated via Monte Carlo simulations in different deployment and application scenarios, and several theoretical and practical results are derived.



## Résumé

Les avancements des technologies de l'information et des systèmes de communication ont permis le développement d'une grande variété d'applications et de services de géolocalisation. Les systèmes de positionnement par satellites figurent parmi les solutions principales de localisation. Dans des environnements difficiles (par exemples, les canyons urbains ou à l'intérieur des bâtiments), ces solutions ne fournissent pas une bonne précision, ou même deviennent indisponibles. Afin d'offrir des solutions de localisation précises et disponibles quelque soit l'environnement, les systèmes de communication sans fil ont été utilisés, où plusieurs paramètres topo-dépendants des signaux transmis peuvent être mesurés et exploités (par exemple, le temps d'arrivée (ToA), la puissance du signal reçu (RSS)). Dans ce travail, la localisation dans les systèmes sans fil est étudié d'un point de vue traitement statistique du signal, et en explorant deux axes. Le premier axe concerne la localisation coopérative appliquée aux réseaux ad-hoc, où les différents nœuds effectuent des mesures de distance par paire (c.à.d. ToA ou RSS) afin d'estimer simultanément leurs positions. Les conditions de solvabilité unique sont étudiées en s'appuyant sur les deux approches de la rigidité graphique et la programmation semi-définie, et ainsi les conditions d'identifiabilité sont déduites. Les solutions d'estimation de la position sont considérées en se concentrant sur l'estimation probabiliste et son application dans des champs aléatoires de Markov et ce en utilisant l'algorithme de propagation de croyance non-paramétrique (NBP). Le deuxième axe concerne la poursuite des terminaux mobiles en se basant sur des mesures RSS. Ces mesures sont affectées par un phénomène de masquage (shadowing). L'amélioration apportée à la précision de positionnement par la connaissance des cartes de shadowing est étudiée. La solution classique pour l'obtention de ces cartes est le 'fingerprinting', qui peut être coûteux en temps de collecte de mesures. Des solutions sont développées afin de surmonter ces difficultés. Plusieurs solutions sont proposées et étudiées par des simulations de Monte Carlo pour différents scénarios d'application et de déploiement, et plusieurs résultats théoriques et pratiques sont obtenus.



**Keywords:** Cooperative localization, position tracking, graph rigidity, semidefinite programming, identifiability, Fisher information, Markov random field, nonparametric belief propagation, Monte Carlo methods, Bayesian filtering, particle filters, Rao-Blackwellization.