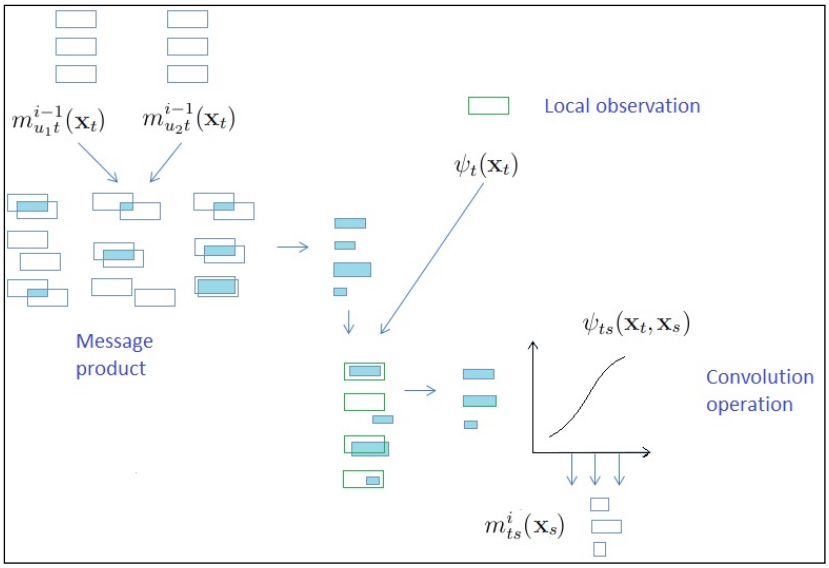
# Data fusion and collaborative state estimation in wireless sensor networks

Hiba Haj Chhadé

Par **Hiba HAJ CHHADÉ**

*Data fusion and collaborative state estimation in wireless sensor networks*

Thèse en cotutelle présentée pour l'obtention du grade de Docteur de l'UTC




Soutenue le 1er juin 2015
**Spécialité** : Technologies de l'Information et des Systèmes

D2207



A thesis

presented in fulfilment of the requirements for
the degree of Doctor of Philosophy
in
Computer Engineering

by: Hiba Haj Chhadé

June 1st 2015

# Data Fusion and Collaborative State Estimation in Wireless Sensor Networks

Supervisors:

Fahed Abdallah
Imad Mougharbel
Amadou Gning

Jury members:

Mr. Fahed Abdallah, University of Technology of Compiègne
Mr. Imad Mougharbel, Lebanese University
Mr. Amadou Gning, University College London
Mr. Hichem Snoussi, University of Technology of Troy
Mr. Ali El Hajj, American University of Beirut
Mr. Mohammad Hajjar, Lebanese University
Mr. Enrico Natalizio, University of Technology of Compiègne
Mme. Lyudmila Mihaylova, University of Sheffield

# ACKNOWLEDGEMENTS

This PhD dissertation represents not only my efforts, but also the time, energy, support, and encouragement of my mentors, colleagues, and friends. I am grateful for this opportunity to acknowledge all those who made this thesis possible.

First and foremost I thank my supervisor at the UTC, Mr. Fahed Abdallah, a person who is exceptional in all ways, scientifically, professionally and especially on the human side, for sharing his vision and experience, for putting me on track in various fields, and finally for treating me as family throughout these past years. I would also like to thank my supervisor at the Lebanese University, Mr. Imad Mougharbel, for sharing his ideas and experience, for giving me the freedom to explore my own ideas and for all the support he showed during my three-years PhD trip.

I give acknowledgement to my adviser, Mr. Amadou Gning, for sharing his knowledge and talent, for making complex material accessible and for all the help and goodwill he showed whenever I met a difficulty. I also acknowledge the supports from the Lebanese CNRS and from the U.K. Engineering and Physical Sciences Research Council (EPSRC) for the support via the Bayesian Tracking and Reasoning over Time (BTaRoT) grant which was only possible through the efforts and collaboration of Mrs. Lyudmila Mihaylova from the University of Sheffield.

Looking back, I owe a tremendous debt of gratitude to the director and my professor at the faculty of engineering, Mr. Mohammad Hamdan, for his continuous efforts towards improving the learning experience at the faculty of engineering and for his help and his guidance during the past years. I humbly thank these and all my professors and advisors for the time and energy they invested in me.

I am greatly indebted to my close friends who have enriched my graduate experience and my life tremendously, for during the past three years they became my family and we spent unforgettable moments together.

Words cannot express my gratitude to my family: to my husband who encouraged and supported me from the very beginning to fulfil my ambitions, to my brother who encourages me to be the best I can be, to my lovely sweet sister who is my closest friend, to the newest member in my family, to my angel Léa who gave me a reason to wake up every morning with a will to live, laugh and progress, and finally, and most importantly, to my mother who made me the person I am, who supported me, who taught me the meaning of ambition, hope and will, to my mother who sacrificed and worked hard for us to be, who lived for and through us; to you mother, I dedicate this work with love and gratitude hoping that I will manage to make you happy and proud.

**Abstract**

The aim of the thesis is to develop fusion algorithms for data collected from a wireless sensor network in order to locate multiple sources emitting some chemical or biological agent in the air. These sensors detect the concentration of the emitted substance, transported by advection and diffusion, at their positions and communicate this information to a treatment centre. The information collected in a collaborative manner is used first to locate the randomly deployed sensors and second to locate the sources. Applications include, amongst others, environmental monitoring and surveillance of sensitive sites as well as security applications in the case of an accidental or intentional release of a toxic agent. However, the application we consider in the thesis is that of land mine detection and localisation. In this approach, the land mines are considered as sources emitting explosive chemicals. The thesis includes a theoretical contribution where we extend the Belief Propagation algorithm, a well-known data fusion algorithm that is widely used for collaborative state estimation in sensor networks, to the bounded error framework. The novel algorithm is tested on the self-localisation problem in static sensor networks as well as the application of tracking a mobile object using a network of range sensors. Other contributions include the use of a Bayesian probabilistic approach along with data analysis techniques to locate an unknown number of vapour emitting sources.

**Titre de la thèse en français**

Fusion de données et estimation collaborative d'état dans les réseaux de capteurs sans fil.

**Résumé de la thèse en français**

L'objectif de la thèse est de développer des algorithmes de fusion de données recueillies à l'aide d'un réseau de capteurs sans fil afin de localiser plusieurs sources émettant un agent chimique ou biologique dans l'air. Ces capteurs détectent la concentration de la substance émise, transportée par advection et diffusion, au niveau de leurs positions et de communiquer cette information à un centre de traitement. L'information recueillie de façon collaborative est d'abord utilisée pour localiser les capteurs déployés au hasard et ensuite pour localiser les sources. Les applications comprennent, entre autres, la surveillance environnementale et la surveillance de sites sensibles ainsi que des applications de sécurité dans le cas d'une libération accidentelle ou intentionnelle d'un agent toxique. Toutefois, l'application considérée dans la thèse est celle de la détection et la localisation de mines terrestres. Dans cette approche, les mines sont considérées comme des sources émettrices de produits chimiques explosifs. La thèse comprend une contribution théorique où nous étendons l'algorithme de propagation de la croyance, un algorithme de fusion de données bien connu et largement utilisé pour l'estimation collaborative d'état dans les réseaux de capteurs, au cadre des méthodes à erreurs bornées. Le nouvel algorithme est testé sur le problème de l'auto-localisation dans les réseaux de capteurs statiques ainsi que l'application de suivi d'un objet mobile en utilisant un réseau de capteurs de distance. Autres contributions comprennent l'utilisation d'une approche probabiliste Bayésienne avec des techniques d'analyse de données pour localiser un nombre inconnu de sources émettrices de vapeur.

# LIST OF FIGURES

# LIST OF TABLES

# CONTENTS

# INTRODUCTION

## *0.1 Theme and objectives*

A wireless sensor network (WSN) consists of a number of spatially distributed sensor nodes which have capabilities of monitoring a physical condition and of cooperating by passing data through the network. Initially motivated by military applications (*e.g.* surveillance of a battlefield), the properties associated with WSNs, such as their autonomy, low cost, small size, diversity and effectiveness in hardly accessible regions, expanded their use to consumer and industrial applications. However, the advantages of WSNs are limited by a number of critical issues, the most popular of which relies probably in the inherent restrictions on bandwidth and energy consumptions. Typically, ubiquitous sensing might result in overwhelming volumes of raw data. In order to extract the small amount of useful information from the network without communicating large volumes of possibly irrelevant raw data, local processing of the information at the sensor nodes is often useful, this leads to the need of distributed algorithms when performing estimation and/or inference.

Another important difficulty associated with WSNs applications is caused by the fact that data gathered by a sensor node is usually noise-corrupted and requires special caution when performing tasks such as decision making or inference. In fact, many problems have inherent uncertainties, a famous example is that of self-localisation of a mobile robot which receives noisy measurements from its wheels encoders and gets inaccurate views from its range finders. Uncertainties are typically modelled as statistical and thus probabilistic methods offer an attractive framework for addressing the problem. In these approaches, attributes of interest are represented using collections of random variables. Often probability distributions associated to these random variables are complex and in the presence of non-linearities, sequential Monte Carlo methods offer an attractive solution. However, these techniques commonly require high computational complexity due to the large number of samples (or particles) needed to represent complex probability density functions (PDFs). On the other hand, interval estimation is another promising methodology which allows complex PDFs to be estimated using fewer *box-particles* and is also independent of non-linearities.

In this thesis, we are interested in deriving fusion algorithms for data collected from a WSN while respecting energy and bandwidth restrictions, and taking into account the noisy aspect of the sensed data. Bayesian inference forms the central theme of this manuscript since it acts as the hub to which various analytical and computational procedures attach in order to solve a variety of problems. Mainly, our work can be structured into two parts.

The first part deals with the problem of inference in distributed systems, such as sensor networks, where the probability model is stored in a distributed fashion. In these applications, each node (*e.g.* sensor) receives information about its local environment and has a local set of attributes (described in terms of random variables) for which it needs to compute a posterior density. Thus each node only stores a relevant portion of the model and has to collaborate with other nodes in the network in order to compute its marginal

posterior of interest, given *all* the observations available to the system [82].

An attractive formalism allowing to exploit conditional independence properties of a joint PDF associated with large scale systems is possible through the use of probabilistic graphical models [62]. Built upon the graphical model, the message passing (or Belief Propagation) approach relies on exchange of messages between neighbour graph nodes. These messages (characterising local/marginal PDFs) are iteratively exchanged between neighbour nodes until convergence is reached. Each node is then able to estimate a specific marginal of the joint posterior density. This approach has many advantages: it is general and can be applied to any network without necessitating knowledge of the overall graphical structure (since each node only knows about its immediate neighbours), and is a distributed algorithm by construction.

Belief Propagation (BP) can be implemented in a parametrised manner (*e.g.*, based on Gaussian PDFs) or in a non-parametric way (*e.g.*, based on sample representation of PDFs). Whilst in the parametric implementation, parameters of the PDF of interest are estimated, *e.g.*, the mean and variance of the Gaussian PDF, in the non-parametric implementation, the PDF is unknown and represented using samples. The Non-parametric BP (NBP) [99] is a particle-based implementation of the BP algorithm [101] and is especially useful for representing joint and marginal probability distributions that are multi-modal, non-standard Gaussian and in the presence of non-linearities.

The second part of the thesis considers the problem of detecting and localising several sources emitting a chemical compound in the air using exclusively concentration measurements provided by a network of wireless sensors assumed capable of sensing the concentration of the emitted compound at their positions. Various applications are associated with this scenario especially in environmental monitoring and security fields (*e.g.* in the case of accidental or intentional release of a toxic agent) [110, 69]. Another application of interest in this thesis is that of land mines detection and localisation [57].

The localisation problem is formulated as Bayesian inference and a Markov Chain Monte Carlo (MCMC) sampler allows to estimate the properties of the posterior PDF of interest. The performance of the probabilistic scheme is also compared to that of an optimisation least squares approach for two scenarios typically associated with land mine detection and localisation [57]: the first, considers the mines as instantaneous point sources (it assumes that the evaporation of the explosive chemical is induced), the second considers the mines as continuous release point sources that were buried long enough for a steady state concentration profile to be established.

## *0.2 Contributions*

### *Part I*

The efficiency of the NBP depends on the number of particles, and typically the imprecision on the available information requires a large number of samples (or particles) which leads to high computational complexity. NBP's operations turn out to be complex and computationally demanding. To deal with the high number of particles, interval-based estimation has arisen as a promising methodology and has been widely employed in state estimation problems [39, 55]. The bounded error framework has proven to be robust to non-linearities and able to estimate complex PDF using a small number of box-particles.

We propose a novel non-parametric Belief Propagation approach, which we call box Belief Propagation (box-BP). In the box-PP, a collection of box particles [41] represents the information exchange between nodes of the network. Adopting an approximation of

the Bayesian solution with mixtures of uniform PDFs, having boxes as support [42], we provide the theoretical Bayesian derivation of the box-BP and the corresponding implementation.

Furthermore, the performance of the novel box-BP is validated on localisation applications described by static and dynamic graphical models. The first version of the box-BP [23] is focused on a sensor calibration problem, *i.e.*, finding the coordinate of all *static* sensors in the network knowing a prior position of a few of them, for this scenario the efficiency of the box-BP is demonstrated over simulated data. The second application considers the localisation of a single mobile target moving within a network of wireless range sensors, it is more general and proposes a solution to the more difficult, dynamic cases, with dynamic posterior PDFs (*e.g.*, for sequentially estimating the state of a moving object). Additionally, results are reported over simulated and real data.

In both cases, the novel box-BP is compared with the standard particle-based Nonparametric Belief Propagation (NBP). As the NBP, the box-BP is aimed for cases of non-linearities, multi-modal PDFs and, compared with the NBP, has several key advantages. First, it is shown through the Bayesian derivation that it requires simpler and faster computations: this is essentially due to the fact that, in comparison with NBP's representation with samples, it uses box-particles which are interpreted as an approximation of PDFs based on mixtures of uniform distributions. Secondly, it needs much less box particles to estimate the PDFs which is doubly advantageous in terms of memory optimisation and reduction of energy and bandwidth requirements for the messages sent in the network. Thirdly, the box-BP has properties inherent to bounded errors approaches, *e.g.* it can deal with quantised measurements and bounded noises with unknown statistics, and is able to propagate, in time, supports that are guaranteed to contain the real solutions [39].

Hence, the contribution of part I is threefold: first, a novel box-Belief Propagation (box-BP) approach is proposed where the BP is performed based on box representations of the state PDFs; second, the effectiveness of the box-BP approach is demonstrated over simulated data in the case of static graphical models; third, the advantages of the box-BP are proven, using real dataset, for a more general case including a dynamic structure of the graph.

### *Part II*

All existing methods for land mine detection consider a known number of sources in a specific region [7, 57]. However this information rarely exists for real applications. One original contribution of this part of the thesis is to present a solution for an *unknown number* of land mines or vapour-emitting sources. We propose a two-step method where the objective is to estimate the number of sources and then localise them. In the first step, the set of concentration measurements which have been made by the detection system are grouped in a matrix and a PCA scheme is used in order to determine the number of sources. Once this number is known we are then able, in the second step, to locate explosive sources and to estimate the emission rate of each source. Thus, in difference to most of existing methods (which solve the problem of localising a single source), we solve the problem of localising *multiple* anti-personnel land mines.

Hence, the main novelty in part II consists in considering the case of multiple land mines whose number is unknown a priori.

## 0.3 Thesis outline

As mentioned earlier, the manuscript revolves around two main parts and is organised in six chapters.

Chapter 1 provides an overview of the prior work and the concepts relevant to the thesis. It basically recalls the Monte Carlo sample-based methods as well as an introduction of graphical models and message-passing algorithms.

Chapters 2, 3 and 4 constitute the core of part I and revolve around the three contributions mentioned in the previous section. Chapter 2 presents the main theoretical contribution of this part: the derivation of the box-BP where the BP is performed based on box representations of the state PDFs. The nodes' instantaneous measurement are fused following a graphical model representation and it is shown that the box-BP provides an efficient solution to the Bayesian inference problem. The derivation of the box-BP relies on the interpretation of a box as a uniform PDF defined over its support, and shows how to implement the BP in the bounded error framework. In chapter 3, the performance of the novel box-BP is tested on the problem of calibration in wireless sensor networks. That is the positioning of a number of randomly deployed sensors, according to some reference defined by a set of anchor nodes for which the positions are known a priori. The effectiveness of the new message-passing scheme is demonstrated over simulated data on a static graphical structure. Chapter 4 investigates the problem of continuously localising an object moving within a wireless sensor network. A wide range sensors are to be deployed in a field of interest such that, at any given time, the moving object can be detected by only a small number of time-of-flight sensors. A cooperative and distributed solution is needed to achieve the common localisation task using, exclusively, the available range measurements. Based on local exchange of information, the Belief Propagation (BP) approach provides a powerful solution to sequentially fuse data collected in a distributed fashion from the sensor nodes. Both the NBP and the box-BP schemes are compared for this scenario over real and simulated data sets.

Part II is organised in two chapters. Chapter 5 provides a literature review on the use of sensor networks for source characterisation in several application fields, as well as an overview of advection-diffusion models for the cases of instantaneous and continuous release point sources. It ends with formulating the forward problem aiming at predicting the concentration measurements at the sensors' positions knowing the sources parameters. Chapter 6 then considers the problem of localising an unknown number of land mines using, exclusively, concentration measurements provided by a wireless sensor network. The two-step approach (mentioned earlier) consisting in sequentially determining the number of sources, then localising them is employed to solve the inverse problem, corresponding to the forward model formulated in Chapter 5. Two technical solutions are reported, a probabilistic Bayesian approach based on a Markov Chain Monte Carlo (MCMC) sampling scheme, and an optimisation least squares technique. The effectiveness of these schemes is tested on simulated data for two scenarios, assuming, respectively, the land mines as instantaneous and continuous release sources.

A general conclusion recapitulating the basic concepts and contributions of the thesis, as well as future work and perspectives, is at last given.

Much of the research in this dissertation has been published or submitted in the form of journal and conference papers. Results reported in chapters 2 and 3 are adapted from [23] while some of the results provided in chapter 6 have been established in [21, 22].

# Part I

# INFERENCE ON GRAPHICAL MODELS

# 1. BACKGROUND

## *1.1 Introduction*

Message-passing algorithms built upon probabilistic graphical models have become important tools to infer in WSNs applications [82]. Furthermore, in order to increase the decision accuracy within a stochastic environment, filtering and inference issues should be involved.

In this chapter, we provide a short overview of concepts relevant to the subsequent parts of the thesis, as well as an introduction of the prior work related to the methods and algorithms derived later on. The first part of the thesis considers the problem of inference on arbitrary graphical models where we derive a novel message-passing algorithm defined in the bounded error framework. A graphical model is a statistical tool which can be associated with a distributed system, where the probability model is stored in a distributed fashion, such as sensor networks. A general description of sensor networks and the common issues involved with them (typically communication and energy constraints) is reported in section 1.2. Section 1.3 then defines the popular problem of filtering which will be seen afterwards as a particular case of the inference problem defined on a graphical model considered in this first part of the dissertation. Different non-parametric representations of probability quantities are employed in the probabilistic procedures we consider, or derive, throughout this manuscript, amongst which are the Monte Carlo particle-based representations, that are briefly explained in section 1.4. Finally, basic knowledge of graphical models and message-passing procedures is available through sections 1.5, 1.6 and 1.7 of this introductory chapter.

## *1.2 Sensor networks: a general overview*

Sensor networks have become a popular solution in many civilian and military applications [93, 110], they are useful whenever there is a need to monitor or to control physical quantities, such as temperature, brightness or pressure. In typical scenarios, sensors acquire data from their environment and often communicate the information collected to a processing centre.

Due to recent advances in wireless technology, networks of wireless sensors present several advantages over their wired counterparts [70], mainly: they are cheaper because they require much less infrastructure, they are freed from the constraint of wiring which considerably limited the deployment of a large number of nodes, they are able to convey information to all nodes connected to the network and, finally, they are self-configurable and self-organising networks with almost no need for direct human intervention.

With these features, WSNs are quickly and cheaply deployed in environments which are dangerous for people, such as regions of conflicts [84], or areas which are difficult to access, as in environmental monitoring applications [69]; WSNs also make it possible to deploy a network consisting of a large number of sensors to monitor wider areas.

A WSN consists of a large number of units called sensor nodes. A node is generally

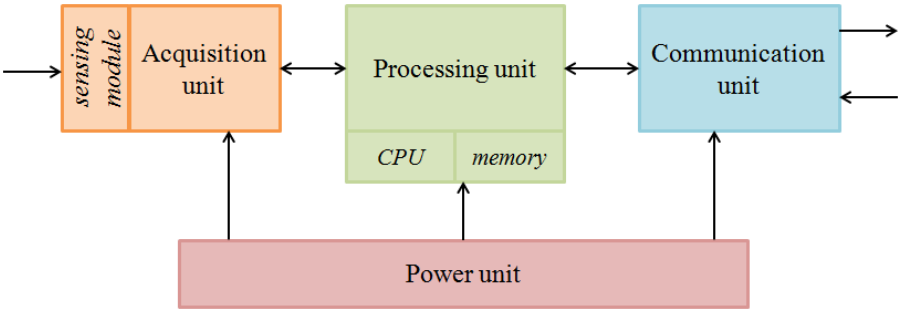



*Fig. 1.1:* Main components of a sensor node.

composed of three main units [10, 51]: an acquisition unit, a processing unit and a communication unit. The acquisition unit obtains numerical measures on environmental parameters depending on the sensing ability of the sensor (*e.g.* seismic, optical, acoustic or meteorological) and forwards the recorded information to the processing unit. The processing unit is composed of a CPU and a specific operating system allowing the sensor node to perform some local computations. The communication unit sends and receives data via a radio communication medium thus enabling the sensor to exchange information with other nodes of the network. These three units are powered by a self-contained battery as depicted in figure 1.1. Other components may be added to a sensor node, such as a positioning module (GPS) or a mobility component to make it mobile.

Despite the advantages of WSNs, a common concern in almost the majority of the work on sensor networks, is that of reducing energy consumption. Indeed, since wireless sensor networks are usually intended to raise information in hostile or inaccessible environments without human intervention, it is hard to find a source of energy other than self-contained batteries; therefore, the lifetime of a sensor node is equal to the lifetime of its battery. Consequently, limitations on the available energy should be considered when performing tasks such as estimation and inference in WSNs. Typically, communication requires many times more the amount of power required for sensing or computation [75]. Algorithms for routing and distributed processing [5, 64, 89], only based on information gathered by neighbour nodes, i.e. within limited ranges, have arisen as an important key for energy optimisation of a WSN.

## 1.3 Bayesian filtering

A dynamic system is a physical system that evolves over time and can be described by a set of variables grouped into a state vector. State estimation methods aim at studying the behaviour of a dynamic system by calculating, at each time step, an estimate of the state vector, usually through combining information provided by different sensors. Filtering refers to the problem of sequentially estimating the states, or hidden variables, of a system as a set of observations become available [109] (this problem is illustrated in figure 1.2). Consider a dynamic model described by the following system:

$$\begin{cases} \mathbf{x}_k = \mathbf{f}(\mathbf{x}_{k-1}, \mathbf{v}_k), & (1.1) \\ \mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{w}_k), & (1.2) \end{cases}$$

where:

- $\mathbf{x}_k \in \mathbb{R}^{n_x}$, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ are respectively the state and the observation vectors at time

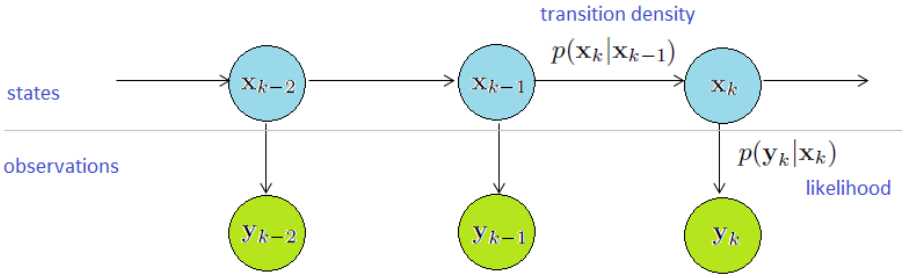



*Fig. 1.2:* The general problem of filtering.

step $k$.

- $\mathbf{f}$ is a transition function which allows to predict the state vector $\mathbf{x}_k$ at time step $k$ in terms of the previous state $\mathbf{x}_{k-1}$ and an i.i.d. process noise sequence, $\mathbf{v}_k \in \mathbb{R}^{n_x}$; equation (1.1) is usually referred to as the prediction model.
- $\mathbf{g}$ is a function that defines the relation between the state and the measurement vectors ($\mathbf{x}_k$ and $\mathbf{y}_k$ respectively) taking into consideration a measurement noise, $\mathbf{w}_k \in \mathbb{R}^{n_y}$, denoting an i.i.d. measurement noise sequence; equation (1.2) is often called the observation model.
- $\mathbf{X}_k = \{\mathbf{x}_k, \mathbf{x}_{k-1}, \ldots, \mathbf{x}_1\}$ and $\mathbf{Y}_k = \{\mathbf{y}_k, \mathbf{y}_{k-1}, \ldots, \mathbf{y}_1\}$ are vectors grouping, respectively, the states and the measurements up to time step $k$.

Within the Bayesian framework, a description of the state given the measurements, up to time step $k$, is provided by the posterior probability distribution $p(\mathbf{X}_k|\mathbf{Y}_k)$. Furthermore, in Bayesian filtering it is often enough determining, at each time step $k$, the posterior *marginal* distribution denoted as $p(\mathbf{x}_k|\mathbf{Y}_k)$ and given by:

$$p(\mathbf{x}_k|\mathbf{Y}_k) = \frac{1}{\alpha_k} p(\mathbf{x}_k|\mathbf{Y}_{k-1}) p(\mathbf{y}_k|\mathbf{x}_k), \tag{1.3}$$

where $\alpha_k = \int p(\mathbf{y}_k|\mathbf{x}_k) p(\mathbf{x}_k|\mathbf{Y}_{k-1}) d\mathbf{x}_k$ is a normalisation factor, $p(\mathbf{y}_k|\mathbf{x}_k)$ is the likelihood, $p(\mathbf{x}_k|\mathbf{Y}_{k-1})$ is the prior and can be expressed as:

$$p(\mathbf{x}_k|\mathbf{Y}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1}) p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1}) d\mathbf{x}_{k-1}. \tag{1.4}$$

In equation (1.4), the PDF $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is called the transition density (and is associated with the prediction model) while $p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})$ is the posterior at time step $k$. The recursion given by expressions (1.3) and (1.4), corresponding respectively to the time update and the measurement update steps, allows to sequentially compute the posterior marginals $p(\mathbf{x}_k|\mathbf{Y}_k)$, for each $k$; the iterative process is initialised with a prior $p(\mathbf{x}_0)$, which can be chosen as an uninformative uniform PDF over some region of the state space.

## 1.4 Monte Carlo methods

Let $p(\mathbf{x})$ denote a probability density function (PDF) defined for $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{x}_p^i \sim p(\mathbf{x})$, $i = 1, \ldots N$, an i.i.d. sample of size $N$. Monte Carlo methods rely on the fact that $p(\mathbf{x})$ can be empirically represented by a set of independent samples drawn from $p$ [58]. A

representation of $p(\mathbf{x})$, denoted as $\hat{p}(\mathbf{x})$, is given by the following expression:

$$\hat{p}(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta_{\mathbf{x}_p^i}(\mathbf{x}), \tag{1.5}$$

where $\delta_{\mathbf{a}}(\mathbf{x})$ denotes the Dirac delta function located at $\mathbf{a} \in \mathbb{R}^n$. The expectation of any function $\mathbf{f}$ of $\mathbf{x}$, under the distribution $p$, can also be approximated as follows:

$$\mathbb{E}_p(\mathbf{f}(\mathbf{x})) = \int \mathbf{f}(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N}\mathbf{f}(\mathbf{x}_p^i). \tag{1.6}$$

It can be easily demonstrated that this estimate is unbiased [58].

Since, in some cases, it is not possible to draw samples directly from $p(\mathbf{x})$, such as when this distribution is not given in a closed parametric form, Monte Carlo approximation allows to represent $p$ using a collection of weighted particles, or samples, using a procedure such as *importance sampling*.

Let $q(\mathbf{x})$ denote a *proposal distribution* (the next section shows how it is chosen) that can be evaluated and sampled, and let $\mathbf{x}_q^i \sim q(\mathbf{x}),\ i = 1, \ldots, N$, be an i.i.d. sample for $q$. Assuming that $q(\mathbf{x})$ is absolutely continuous with respect to $p(\mathbf{x})$, *i.e.*, if $q(\mathbf{x}_0) = 0$, for some $\mathbf{x}_0 \in \mathbb{R}^n$, then $p(\mathbf{x}_0) = 0$ as well. The relative likelihood of having drawn $\mathbf{x}_q^i$ from $p(\mathbf{x})$ versus $q(\mathbf{x})$ is given by:

$$\omega^i \ \propto \ \frac{p(\mathbf{x}_q^i)}{q(\mathbf{x}_q^i)}. \tag{1.7}$$

The quantities $\omega^i,\ i = 1, \ldots, N$, are called *weights* and are normalised as to obtain $\sum_i \omega^i = 1$; to compute these weights, only the ratio of $p$ to $q$ needs to be known. The resulting weighted collection of samples $\{\omega^i, \mathbf{x}_q^i\}$ represents the distribution $p(\mathbf{x})$ and similarly to equation (1.6) an approximation of the expectation of a function $\mathbf{f}(\mathbf{x})$ is given by

$$\int \mathbf{f}(\mathbf{x})p(\mathbf{x})d\mathbf{x} \approx \frac{1}{N}\sum_{i=1}^{N}\omega^i\mathbf{f}(\mathbf{x}_q^i). \tag{1.8}$$

### 1.4.1 Particle filtering

Different approaches for solving the filtering problem defined in section 1.3 have been proposed in the literature. Amongst the most popular filtering procedures we find the standard Kalman filter (KF) and its variants such as the extended Kalman filter (EKF) [109, 111], these algorithms make an assumption of Gaussian posterior PDFs to approximate the recursive Bayesian estimation described previously. Particle filters make no assumptions on the form of the probability densities of interest, they provide a Monte Carlo based procedure for sequentially estimating the posterior marginal distributions $p(\mathbf{x}_k|\mathbf{y}_k, \mathbf{y}_{k-1}, \ldots, \mathbf{y}_1)$ at each time step.

The most basic form of particle filtering operates as follows: given a collection of weighted samples $\{\omega_{k-1}^j, \mathbf{x}_{k-1}^j\}$ representing the posterior $p(\mathbf{x}_{k-1}|\mathbf{Y}_{k-1})$ at time $k-1$, in the time update step, these particles are propagated through the prediction model by sampling from the conditional distribution $p(\mathbf{x}_k|\mathbf{x}_{k-1})$:

$$\mathbf{x}_k^j \sim p(\mathbf{x}_k|\mathbf{x}_{k-1}^j), \tag{1.9}$$

the proposal distribution is thus chosen to be the transition probability function. The

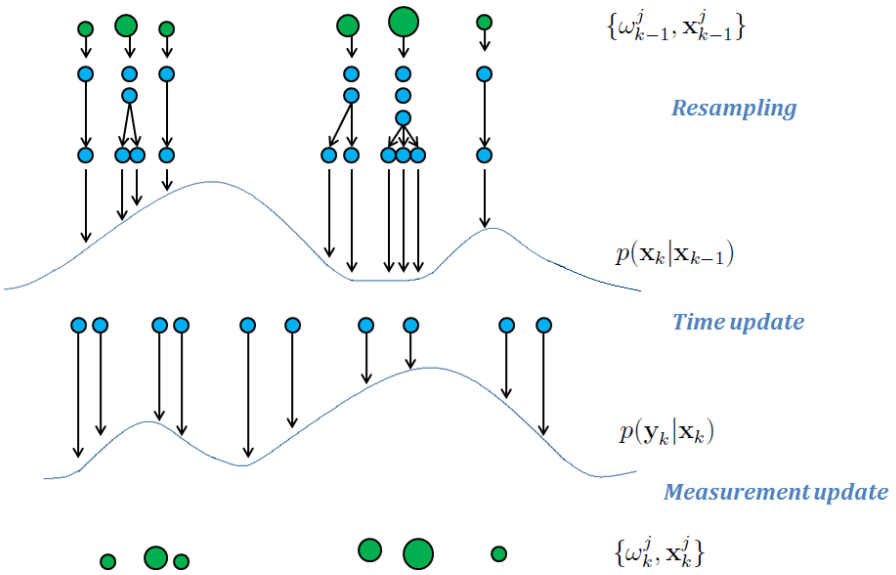



*Fig. 1.3:* Particle filtering.

resulting collection of particles $\{\omega_{k-1}^j, \mathbf{x}_k^j\}$ represents the prior at time $k$; the next step, known as the measurement update step, is to correct the weights using the likelihood at $k$ according to what follows

$$\omega_k^j \propto \omega_{k-1}^j p(\mathbf{y}_k|\mathbf{x}_k). \tag{1.10}$$

The weighted collection of samples $\{\omega_k^j, \mathbf{x}_k^j\}$ now represents the posterior $p(\mathbf{x}_k|\mathbf{Y}_k)$ at time $k$. This iterative procedure is initialised by drawing samples $\mathbf{x}_1^j \sim p(\mathbf{x}_1)$ from some prior distribution $p(\mathbf{x}_1)$.

A problem often encountered with this procedure is particle depletion: that is one or more particles end up dominating with a relatively great weight as we walk through the iterations. A popular solution for this problem is resampling. It consists in replacing the collection of weighted particles by a set of equally weighted particles, before propagating the particles through the transition probability as shown in figure 1.3. Intuitively, several copies of a single particle would exist, their number is proportional to the initial particle's weight. These copies, when propagated through the transition density, result in different new particles and thus adds variety to the samples representing the prior at time $k$.

## 1.5 Graphical models

Probabilistic graphical models [62] are special graphs that combine probability with graph theories in order to visualise conditional independence properties of joint PDFs [82]. They are important and useful tools to reason with probability models defined in large spaces [12]. Typically, given a set of random variables, conditional independence properties are crucial for understanding the underlying structures of a joint PDF. Graphical models offer a framework to exploit these conditional independence properties and to represent joint probability distributions as a product of local factors where each factor depends only on a subset of variables. This factorisation offers many computational advantages, mainly a more compact representation of the PDF and an accelerated inference [82].

Graphical models have important characteristics [12] that make them widely employed in fields such as statistics, statistic physics, computer vision and machine learning, where

the problems of uncertainty and complexity are often encountered: (a) they represent the structure of a probability model; (b) they offer a visualisation of the conditional independence properties of the model; (c) they translate complex computations, required to perform tasks such as inference, into visible graphical manipulations.

Formally, if $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ denotes a set of random variables, $\mathbf{x}_i,\ i = 1, \ldots, N$, $p(\mathbf{X})$ a joint PDF defined over $\mathbf{X}$, a probabilistic graphical model is a tool to represent the underlying structure of the joint PDF $p$; it is a special graph $G = (V, E)$ where $V$ is a set of nodes or vertices, and $E$ is a set of edges, each node $i \in V$ is associated with a random variable $\mathbf{x}_i$ and each edge $(i, j)$ represents a probabilistic relation between the random variables $\mathbf{x}_i$ and $\mathbf{x}_j$. The graph structure describes the independence relationships between the variables qualitatively. The parameters $\Theta$ will be used to quantify the interdependency between nodes $i$ and $j$.

Different types of graphical models exist: directed graphical models and undirected graphical models. Directed graphical models, also known as Bayesian Networks (BNs), express causal relationships between random variables. Undirected graphical models, also called Markov Random Fields (MRFs), encode constraints and correlations between variables [97]. BNs are popular in statistics, artificial intelligence and machine learning societies while MRFs are widely employed in computer vision and statistical physics [11]. A third class, known as factor graphs, generalises both directed and undirected types of graphical models [63, 100].

### *1.5.1 Directed graphical models*

Directed graphical models, *i.e.* Bayesian Networks, are a class of graphical models that uses directed acyclic graphs (DAGs) to express causal relationships between random variables [11]. They characterise the way a joint PDF defined over a set of random variables, factors into a product of conditional probabilities imposed by the structure of the graph.

Let $p(\mathbf{X})$ denote a joint probability distribution defined over a set $\mathbf{X}$ of $N$ random variables, $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Using the chain rule, $p$ can be expressed as:

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = p(\mathbf{x}_1) \prod_{j=2}^{N} p(\mathbf{x}_j | \mathbf{x}_{j-1}, \ldots, \mathbf{x}_1). \tag{1.11}$$

Many graphs, such as static or dynamic sensor networks, are sparse. This implies that, for the associated probability models, the probabilities of some of the variables $\mathbf{x}_j$ are not dependent of all of their predecessors (i.e. $\mathbf{x}_1$, $\mathbf{x}_2$, $\ldots$, $\mathbf{x}_{j-1}$) but only depend on some of them. This restricted set of variables is called the set of parents of $\mathbf{x}_j$ and is denoted by $pa(\mathbf{x}_j)$. Using these independence properties, the following equality holds:

$$p(\mathbf{x}_j | \mathbf{x}_{j-1}, \ldots, \mathbf{x}_1) = p(\mathbf{x}_j | pa(\mathbf{x}_j)),$$

and hence, equation (1.11) can be put into the following form:

$$p(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N) = \prod_{j \in \{1,...,N\}} p(\mathbf{x}_j | pa(\mathbf{x}_j)). \tag{1.12}$$

The directed graph describing this joint probability distribution is defined by two sets: the set of nodes associated with the random variables, and the set of edges representing a direct *causal connection* between the variables. Indeed, an arrow from node $i$, associated with random variable $\mathbf{x}_i$, to node $j$ associated with $\mathbf{x}_j$, encodes the fact that a value taken by $\mathbf{x}_j$ is dependent of the value taken by $\mathbf{x}_i$. The DAG structure provides

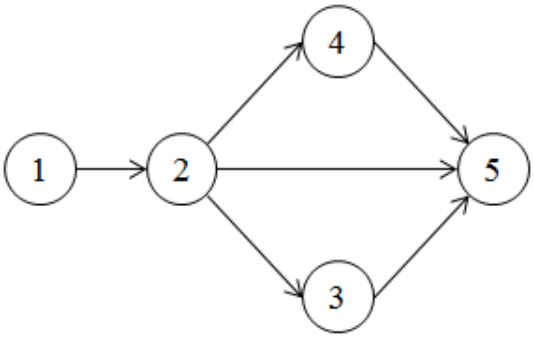



*Fig. 1.4:* Example of a Bayesian network: the DAG encodes the structure of the distribution $p(\mathbf{X}) = p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5)$.

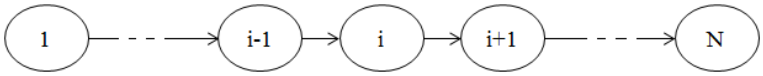



*Fig. 1.5:* Example of a Bayesian network: a first order Markov chain.

a qualitative description of the model, the set of quantitative parameters $\Theta$, which specifies the forms of the relationships between the random variables, groups the quantities $\theta_{\mathbf{x}_i|pa(\mathbf{x}_i)} = p(\mathbf{x}_i|pa(\mathbf{x}_i))$. Figure 1.4 illustrates an example of a BN associated with a joint PDF defined over $N = 5$ random variables.

Conversely, having the DAG describing the structure of a joint PDF, the conditional independence properties, *i.e.* the factorised form of the model, can be inferred. In fact, each node/variable is independent of its non-descendants [1] in the graph given the state of its parents. The joint PDF can be factorised into a product of conditional probability distributions where each node is only dependent of its parents. In the example of figure 1.4, the joint PDF factors into:

$$p(\mathbf{X}) = p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5) = p(\mathbf{x}_5|\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4)p(\mathbf{x}_4|\mathbf{x}_2)p(\mathbf{x}_3|\mathbf{x}_2)p(\mathbf{x}_2|\mathbf{x}_1)p(\mathbf{x}_1).$$

To ensure that this factorisation holds, an important restriction on the topology of the graph must be maintained, that is, no path, from any node of the graph, can lead back to the node itself along the directed edges; this property is guaranteed by the structure of the acyclic graph.

A final remark is that, although the arrows represent a direct causal connection between the variables, information can propagate in *any direction*. In fact, a causal relation from node $i$ to node $j$ implies that any information on $\mathbf{x}_i$ may modify the knowledge one has on $\mathbf{x}_j$. Conversely any knowledge on $\mathbf{x}_j$ may reflect an information about $\mathbf{x}_i$ [97].

**Example 1.5.1.** A Markov chain is a simple example of a directed graphical model. A set of random variables $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ define a first order Markov chain if

$$p(\mathbf{x}_{i+1}|\mathbf{x}_i, \mathbf{x}_{i-1}, \ldots, \mathbf{x}_1) = p(\mathbf{x}_{i+1}|\mathbf{x}_i), \ \forall i \in \{1, \ldots, N-1\}. \tag{1.13}$$

The corresponding graph is given in figure 1.5.

[1] In graph theory, the set of descendants of a node comprises the set of nodes that can be reached from this node on a direct path [11].

### 1.5.2 Undirected graphical models

Markov Random Fields (MRFs) are a class of graphical models that uses an undirected graph to encode conditional independence relationships between random variables [11].

Let $G = (V, E)$ be an undirected graph representing a joint probability distribution, $p(\mathbf{X})$ defined over a set of $N$ random variables $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$, $A$, $B$ and $D$ refer to three disjoint subsets of $V$, $\mathbf{X}_A, \mathbf{X}_B, \mathbf{X}_D$ the sets of random variables associated with the nodes in sets $A$, $B$ and $D$ respectively. If every path between $A$ and $B$ passes through some node in $D$, $D$ is said to separate $A$ and $B$ [100]. This graphical criterion, known as *graph separation* [82], encodes conditional independence between $\mathbf{X}_A$ and $\mathbf{X}_B$ given $\mathbf{X}_D$, thus:

$$p(\mathbf{X}_A, \mathbf{X}_B|\mathbf{X}_D) = p(\mathbf{X}_A|\mathbf{X}_D)p(\mathbf{X}_B|\mathbf{X}_D). \tag{1.14}$$

For a node $i \in V$, let $\Gamma_i \subset V$ refer to the set of neighbours of $i$, $\Gamma_i$ hence consists of the nodes of $G$ connected to $i$ by an edge, and designates a set of variables denoted as $\mathbf{X}_{\Gamma_i} = \{\mathbf{x}_j | j \in \Gamma_i\}$. The graph structure suggests that any random variable $\mathbf{x}_i$ is conditionally independent of all other variables of the model given $\mathbf{X}_{\Gamma_i}$:

$$p(\mathbf{x}_i|\mathbf{X}_{V\backslash i}) = p(\mathbf{x}_i|\mathbf{X}_{\Gamma_i}), \tag{1.15}$$

where $V\backslash i$ refers to the set of nodes of $G$ except for node $i$.

To specify the forms of the relationships between random variables, a parametrisation of MRFs is obtained by referring to the Hammersley and Clifford theorem. Recall beforehand that, in graph theory, a clique is a set of fully connected nodes in a graph (see figure 1.6 for an example).

**Theorem 1.5.1.** Hammersley and Clifford Theorem: *(adapted from [100])*
*Let $G = (V, E)$ be an undirected graphical model, where each vertex $i \in V$ corresponds to a random variable $\boldsymbol{x}_i$, and $C$ denotes a set of cliques of $G$; then, a probability distribution defined as the product of normalised positive functions (symmetric in their arguments) defined on the cliques is always Markov with respect to the graph.*

$$p(\boldsymbol{X}) \propto \prod_{c \in C} \psi_c(\boldsymbol{X}_c). \tag{1.16}$$

*Alternatively, any positive joint density function $p(\boldsymbol{X}) > 0$, $\forall \boldsymbol{X}$, which is Markov with respect to the graph, implies that there exist positive functions $\psi_c$ (symmetric in their arguments) such that the equality in equation (1.16) holds.*

According to theorem 1.5.1, the joint distribution can be expressed as a product of factors, each of which is defined on a clique of the graph.

$$p(\mathbf{X}) = \frac{1}{Z} \prod_{c \in C} \psi_c(\mathbf{X}_c), \tag{1.17}$$

where $C$ is a set of cliques of $G$, $\mathbf{X}_c$ designates the set of random variables associated with $c \in C$, $\psi_c(\mathbf{X}_c)$ is a factor associated with $c$ and is referred to as a *potential function*, and $Z$ is a normalisation constant. Figure 1.7 shows an example of a MRF with $N = 5$ nodes/variables. One possible parametrisation of the graph is given by

$$p(\mathbf{X}) = \frac{1}{Z}\psi_{235}(\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5)\psi_{245}(\mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5)\psi_{12}(\mathbf{x}_1, \mathbf{x}_2).$$

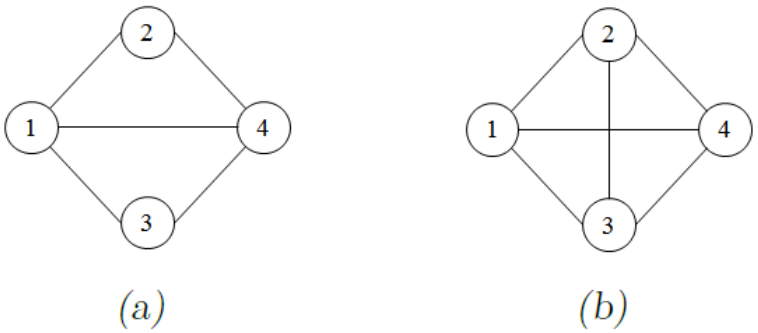



*Fig. 1.6:* (a) this configuration does not define a clique because no edge interconnects nodes 2 and 3. (b) an example of a clique.

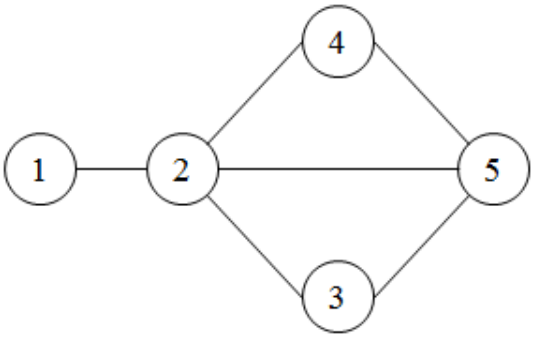



*Fig. 1.7:* Example of a Markov Random Field.

**Example 1.5.2.** *Pairwise Markov Random Fields*
Pairwise MRFs are a restricted class of the undirected graphical models family, in which, the cliques are exclusively pairs of nodes of the graph connected by an edge. Figure 1.8 illustrates an example of a pairwise MRF. The joint PDF can be expressed as

$$p(\mathbf{X}) = \frac{1}{Z}\psi_{12}(\mathbf{x}_1, \mathbf{x}_2)\psi_{23}(\mathbf{x}_2, \mathbf{x}_3)\psi_{24}(\mathbf{x}_2, \mathbf{x}_4)\psi_{25}(\mathbf{x}_2, \mathbf{x}_5).$$

**Example 1.5.3.** For the example given in figure 1.7, choosing the cliques to be exclusively pairs of nodes connected by an edge, a different parametrisation of the MRF can be obtained:

$$p(\mathbf{X}) = \frac{1}{Z}\psi_{12}(\mathbf{x}_1, \mathbf{x}_2)\psi_{23}(\mathbf{x}_2, \mathbf{x}_3)\psi_{24}(\mathbf{x}_2, \mathbf{x}_4)\psi_{25}(\mathbf{x}_2, \mathbf{x}_5)\psi_{35}(\mathbf{x}_3, \mathbf{x}_5)\psi_{45}(\mathbf{x}_4, \mathbf{x}_5).$$

The parametrisation of a MRF is not unique.

### *1.5.3 The problem of inference*

In a typical scenario, two sets of random variables are considered: the set $\mathbf{X}$ and the set $\mathbf{Y}$ of respectively hidden and observable variables, and it is usually desired to obtain an information about the hidden quantities given the observations. Consequently, the set of nodes $V$, associated with the random variables describing the system, is split into two disjoint sets, $V_x$ and $V_y$, associated respectively with the set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$

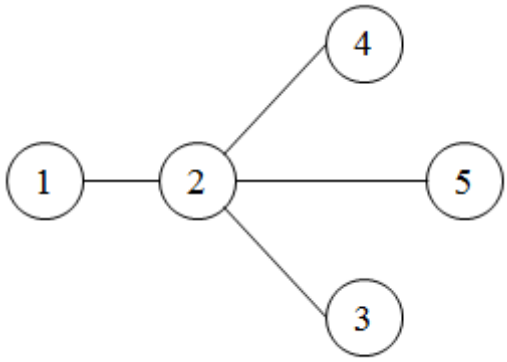



*Fig. 1.8:* Example of a pairwise MRF.

of the hidden variables and the set $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_M\}$ of the observable ones. Nodes in sets $V_x$ and $V_y$ are referred to as hidden (or latent) and evidence nodes respectively. The problem of inference on a graphical model refers to finding the posterior distribution, denoted as $p(\mathbf{X}|\mathbf{Y})$, of some, or all, hidden variables given the observations [97, 117].

Without loss of generality, let us develop further the expression of the posterior distribution $p(\mathbf{X}|\mathbf{Y})$ in the case of an undirected graphical model whose cliques are chosen to be exclusively pairs of nodes connected by edges, *e.g.* the case of a pairwise MRF. Let us also assume that the set $\mathbf{Y}$ of observations is actually a collection of local observations $\mathbf{y}_i$ of individual node variables $\mathbf{x}_i$, $i \in V$, corrupted by some noise [51] and that $\mathbf{y}_i$ is independent of all variables in the model given $\mathbf{x}_i$. Using the Hammersley and Clifford theorem, the probability distribution $p(\mathbf{X}, \mathbf{Y})$ can be factorised into a product of factors, or potential functions, which, under the previous assumptions, can be partitioned into two groups: the first group corresponds to edges existing between the hidden variables and their associated local observations and the second group corresponds to edges interconnecting hidden variables. Hence, the posterior distribution can be expressed as a product:

$$p(\mathbf{X}|\mathbf{Y}) = \frac{p(\mathbf{X}, \mathbf{Y})}{p(\mathbf{Y})} \propto \prod_{(i,j)\in E} \psi_{ij}(\mathbf{x}_i, \mathbf{x}_j) \prod_i \psi_i(\mathbf{x}_i, \mathbf{y}_i). \tag{1.18}$$

Figure 1.9 illustrates the problem of inference on a graphical model defined in this section. In this example, the vector of hidden variables is $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7)$ and the vector grouping the observations is $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_3, \mathbf{y}_5)$. Typically, nodes associated with $\mathbf{Y}$ should have been added to the graph, with edges interconnecting these evidence nodes with their corresponding hidden variables; for simplicity reasons, and since the observations are local to their associated hidden nodes, nodes for which an observation is available are shaded.

## 1.6 Belief Propagation algorithm

The Belief Propagation (BP), also known as the sum-product algorithm, is a statistical estimation tool for solving inference problems defined on graphical models. It exploits the conditional independence relationships encoded by the graph in order to calculate, in an exact or an approximated way, the marginal posterior distribution at a node of the model. In our exposition of the BP, we focus on pairwise MRF, and hence, we reason within the assumptions made in section 1.5.3. However, the methods underlying BP as

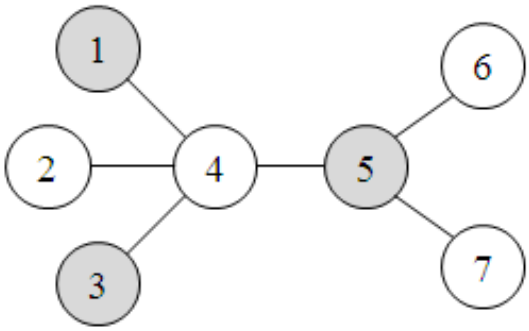



*Fig. 1.9:* The inference problem refers to determining the posterior distribution $p(\mathbf{X}|\mathbf{Y})$. Shaded nodes are associated with variables for which an observation is available.

described in this section are also adapted to message-passing in Bayesian Networks (BNs) and in factor graphs [108, 18, 100].

The inference goal considered here is to compute the posterior *marginal* distribution associated with a node $t$ given the set of observations $\mathbf{Y}$:

$$p(\mathbf{x}_t|\mathbf{Y}) = \int_{\mathbf{X}\setminus\mathbf{x}_t} p(\mathbf{X}|\mathbf{Y}) d\mathbf{X}\setminus\mathbf{x}_t. \tag{1.19}$$

Based on the factorisation (1.18), this marginalisation task is captured by BP in an iterative fashion: BP [112] iteratively exchanges information between nodes of a graphical model through a message-passing procedure, the most common form of which is a parallel update scheme where, at each iteration, each node calculates outgoing messages to all of its neighbours simultaneously. A message, denoted by $m^i_{ts}(\mathbf{x}_s)$, is sent from a node $t$ to one of its neighbours $s$, at an iteration $i$ of BP, and is given by the following expression:

$$m^i_{ts}(\mathbf{x}_s) \propto \int \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s)\psi_t(\mathbf{x}_t) \prod_{u\in\Gamma_t\setminus s} m^{i-1}_{ut}(\mathbf{x}_t) d\mathbf{x}_t, \tag{1.20}$$

where $\psi_t(\mathbf{x}_t, \mathbf{y}_t)$ is replaced by $\psi_t(\mathbf{x}_t)$ for the simplicity of the notation. Note that the index $i$ is henceforth employed to denote the iteration number while $t$ and $s$ will refer to two neighbouring nodes. Figure 1.11 illustrates BP's message update step for the sample graph given in figure 1.10.

Let us denote by $R^i_{ts}(\mathbf{x}_t)$ the product of messages incoming to node $t$, at iteration $i$ of BP, from the set of its neighbours except for node $s$:

$$R^i_{ts}(\mathbf{x}_t) \propto \prod_{u\in\Gamma_t\setminus s} m^i_{ut}(\mathbf{x}_t). \tag{1.21}$$

The quantity

$$M^i_{ts}(\mathbf{x}_t) \propto \psi_t(\mathbf{x}_t) R^i_{ts}(\mathbf{x}_t) \tag{1.22}$$

is referred to as the partial belief, it combines, at an iteration $i$, all the available information about $t$, from $t$ itself and $t$'s neighbouring nodes (except for the receiving node $s$).

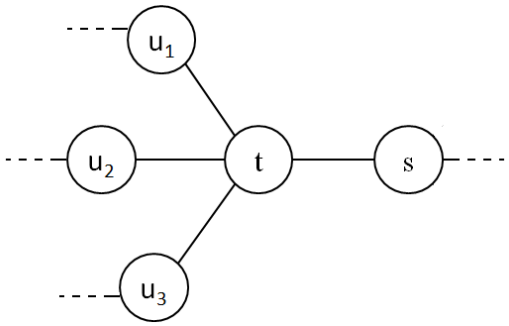



*Fig. 1.10:* A sample graph.

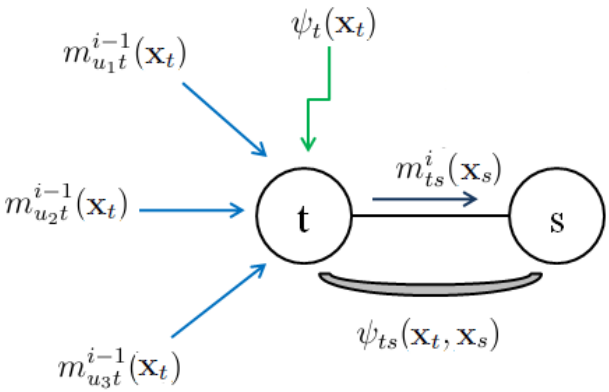



*Fig. 1.11:* Message update.

These two quantities will be helpful when we demonstrate the theoretic derivation of the box-BP in the next chapter.

Next to message-passing, the belief at iteration $i$, at node $t$, is computed as follows:

$$q^i_t(\mathbf{x}_t) \;\propto\; \psi_t(\mathbf{x}_t) \prod_{u \in \Gamma_t} m^i_{ut}(\mathbf{x}_t). \tag{1.23}$$

Beliefs are PDFs and hence normalised in order to integrate to unity [51].

Figure 1.12 illustrates BP's belief computations for the example in figure 1.10. The belief at a node $t$ approximates the posterior marginal of interest $p(\mathbf{x}_t|\mathbf{Y})$. In fact, under some conditions the belief converges to the exact posterior marginal distribution [51].

### 1.6.1 Loopy Belief Propagation

If the graphical model is tree-structured (i.e. is loop free), as in the case of a pairwise MRF, the Belief Propagation is guaranteed to converge after a finite number of iterations (at most equal to the length of the longest path in the graph [51]). In this case, the belief at a node $t$ will be exactly equal to the posterior marginal of interest $p(\mathbf{x}_t|\mathbf{Y})$. Nevertheless, BP may also be applied to arbitrary graphical models. The same local Belief Propagation equations are iterated overlooking the presence of loops in the graph. This iterative procedure is referred to as *loopy BP*. In this case, the convergence of the sequence of messages is not guaranteed. Under some conditions however, fixed points will appear and in practice, these fixed points (beliefs) constitute reasonable approximations of the exact posterior marginal distribution associated with the node. Loopy BP is explained in details in [18].

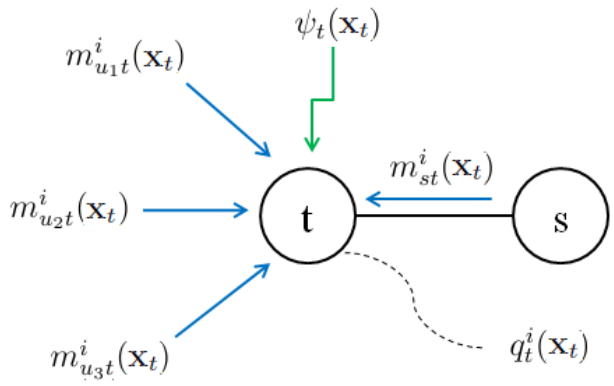



*Fig. 1.12:* Belief computation.

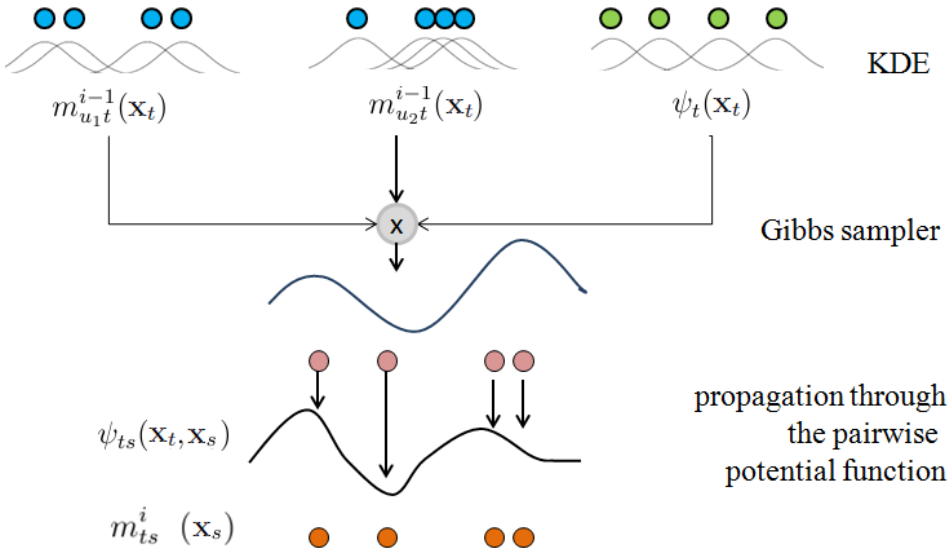



*Fig. 1.13:* Non-parametric Belief Propagation.

## 1.7 Non-parametric Belief Propagation

The BP algorithm is a tool for performing exact or approximate marginalisation on arbitrary graphical models. This algorithm relies on two operations [79]: the first is a product of a collection of messages, the second is a convolution operation of this message product with a pairwise potential function as shown in equation (1.20). In many applications of graphical models, e.g. in computer vision [101], the hidden variables of interest are described by continuous non-Gaussian distributions; furthermore, the models governing information exchange between nodes may be seriously non-linear which makes BP's operations intractable. Non-parametric Belief Propagation (NBP) [99] is an attractive solution in the presence of such complex situations; it is considered, on one hand, as a generalisation of particle filtering applied to arbitrary graphical models and on the other hand as a stochastic approximation of BP [51, 101, 80].

The basic idea behind NBP is to use sample-based representations to approximate the operations of BP [90]. We explain the computations required by NBP by considering each operation of the message update step (see equation (1.20)) stated earlier. Recall that $t$ and $s$ refer to the sending and the receiving nodes respectively, while $\mathbf{x}_t$ and $\mathbf{x}_s$ are their associated random variables.

### 1.7.1 The message product operation

The first operation in the message update procedure is a message product which makes it possible to compute the partial belief at the sending node $t$ according to equation (1.22). In NBP, messages $m^i_{ut}(\mathbf{x}_t)$ are represented by a collection $\{\omega^j_{ut}, \mathbf{x}^j_{ut}\}$, $j = 1, \ldots, N$, of weighted samples. A representation of $m^i_{ut}(\mathbf{x}_t)$ is hence given by

$$\hat{m}^i_{ut}(\mathbf{x}_t) = \sum_{j=1}^{N} \omega^j_{ut} \delta(\mathbf{x} - \mathbf{x}^j_{ut}), \tag{1.24}$$

where $\delta$ is the Dirac impulse function. The product $\prod_{u \in \Gamma_t \setminus s} \hat{m}^i_{ut}(\mathbf{x}_t)$ is virtually guaranteed to be *zero* everywhere since the event of drawing exactly the same samples from all of the messages is far unlikely; it is thus necessary to have continuous estimates of the messages. Such an estimation can be determined using non-parametric density estimation methods [98]. In [51], the effect of each sample is smoothed onto a nearby region using a Gaussian Kernel $K_h(.)$, so the message is estimated by a mixture of $N$ Gaussian distributions:

$$m^i_{ut}(\mathbf{x}_t) = \sum_{j=1}^{N} \omega^j_{ut} K_h(\mathbf{x} - \mathbf{x}^j_{ut}). \tag{1.25}$$

Performing the product of $d$ messages, each represented by a mixture of $N$ Gaussian distributions, and noting that the product of two Gaussian distributions is Gaussian [97], the result is a Gaussian mixture containing $N^d$ components. To avoid complex computations, the author in [51] proposes to sample exactly $N$ particles from this product of Gaussian mixtures using a Markov Chain Monte Carlo (MCMC) method, namely the Gibbs sampler. The message product, *i.e.* the partial belief, is thus represented by a collection $\{\Omega^j_{ts}, \mathbf{x}^j_{ts}\}_{j=1}^N$ of weighted particles. If $N$ is chosen sufficiently large, this collection of samples is guaranteed to accurately characterise the true product [98].

### 1.7.2 The convolution operation

The second operation in the message update expression is a convolution through which the available information about the sending node $t$, represented by the partial belief, is propagated through the pairwise potential function $\psi_{ts}(\mathbf{x}_t, \mathbf{x}_s)$, specifying the relationship between $\mathbf{x}_t$ and $\mathbf{x}_s$, in order to form a message providing some information about the receiving node $s$' local state.

The partial belief is represented by a collection $\{\Omega^j_{ts}, \mathbf{x}^j_{ts}\}_{j=1}^N$ of weighted particles. In order to approximate the integral in equation (1.20), a direct application of Monte Carlo approximation given in equation (1.6) can be employed. However, the pairwise potential function $\psi_{ts}(\mathbf{x}_t, \mathbf{x}_s)$ can have a marginal influence on $\mathbf{x}_t$ [100], *i.e.* the integral

$$\zeta_{ts}(\mathbf{x}_t) \;=\; \int \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) d\mathbf{x}_s, \tag{1.26}$$

is not equal to unity. NBP accounts for this influence by incorporating $\zeta_{ts}(\mathbf{x}_t)$ into the message product operation, the samples are thus drawn from the product

$$\zeta_{ts}(\mathbf{x}_t) M^i_{ts}(\mathbf{x}_t) \;\propto\; \zeta_{ts}(\mathbf{x}_t) \psi_t(\mathbf{x}_t) \prod_{u \in \Gamma_t \setminus s} m^i_{ut}(\mathbf{x}_t). \tag{1.27}$$

To approximate the integral in equation (1.20), the resulting particles are then propagated through the pairwise potential function $\psi_{ts}(\mathbf{x}_t, \mathbf{x}_s)$. Figure 1.13 summarises NBP's

computations.

## 1.8 Conclusion

This chapter aimed at familiarising the reader with the material discussed in subsequent chapters of the thesis by providing an overview of the related prior work. It evoked general issues involved with sensor networks applications. It also presented a short explanation of Monte Carlo non-parametric representations of PDFs which are widely used in environments presenting complex and unknown uncertainties. The notion of a probabilistic graphical model as well as the problem of inference which is considered in the first part of the thesis were also defined. The general Belief Propagation message-passing algorithm as well as a non-parametric variant of it employing particle-based representations of probability distributions were also emphasised. This algorithm, known as Non-parametric Belief Propagation (NBP), provides a solution to the problem of inference on arbitrary graphical models associated with systems comprising high-dimensional random variables and/or continuous non-Gaussian uncertainties and severe non-linearities. Each iteration of NBP used an efficient sampling procedure to update kernel-based estimates of the true continuous messages.

However, on one hand, the storage of a sufficiently large number of weighted particles in order to accurately represent a PDF, and on the other hand, the overhead assigned to computing Kernel density estimates and sampling from products of Gaussian mixtures, increases the complexity associated with NBP's computations. This difficulty was the motivation behind deriving a novel scheme of message-passing in the next chapter. This procedure, which we will call box-BP, is applicable in the presence of complex uncertainties and non-linearities, and uses interval-based approximations of the probability quantities.

# 2. BELIEF PROPAGATION IN INTERVAL ANALYSIS FRAMEWORK

## *2.1 Introduction*

In the presence of complex statistical uncertainties and non-linearities, non-parametric particle methods offer impressive advantages over their parametric counterparts. We saw in the previous chapter how the NBP approximates the standard Gaussian BP algorithm using Monte Carlo methods and how it can be efficiently employed to infer on arbitrary graphical models in such complex situations. However, this algorithm requires intensive computations and the use of a large number of samples to accurately approximate the operations of the BP. In the context of temporal inference problems, *i.e.* filtering, the box particle filter [42] achieved a significant efficiency compared with the generic sampling importance resampling particle filter. In fact, the estimation accuracy provided by this last, using thousands of particles, can be achieved by the box-PF using only a few particles [41].

Inspired by the box particle filter (box-PF) which combines interval analysis with particle filtering to solve temporal inference problems, this chapter presents an algorithm that uses bounded error methods for solving the inference problem in an arbitrary graphical model. By using interval methods, the overhead associated with the computation of non-parametric density estimates and the procedure of sampling from the product of messages in the NBP is avoided. This results in simpler computations in addition to reducing the memory storage required since only a few box-particles can be used to accurately approximate a probability density function. This chapter is organised as follows. In section 2.2 we provide an overview of the fundamental concepts from the interval analysis framework. We then present a brief depiction of the box-PF in section 2.3. Later, we show how to implement the Belief Propagation procedure when the information is represented using boxes. The corresponding theoretical derivation, presented in section 2.4, is based on interpreting a box as a uniform PDF. Finally, the box-PF is seen as a special inference problem defined on a temporal graphical model in section 2.5 and a conclusion of this chapter is given in section 2.6.

## *2.2 Bounded error methods*

For systems presenting complex or unknown statistical bounded uncertainties, interval estimation is a promising methodology. Initially introduced to propagate rounding errors in mathematical computations [76], applications of the interval analysis framework to state estimation problems have been widely investigated [1, 40, 2, 55].

An interval is a closed and connected subset of $\mathbb{R}$ defined as

$$[x] = [\underline{x}, \overline{x}] = \{x \in \mathbb{R} \mid \underline{x} \leq x \leq \overline{x}\}, \tag{2.1}$$

where $\underline{x}$ and $\overline{x}$ refer to the minimal and maximal bounds of $[x]$ respectively.

A box is a vector $[\mathbf{x}]$ in $\mathbb{R}^n$ and can be seen as a Cartesian product of $n$ intervals:

$$[\mathbf{x}] = [x_1] \times \ldots \times [x_n]. \tag{2.2}$$

Henceforth, $\mathbb{IR}$ refers to the set of intervals in $\mathbb{R}$, $\mathbb{IR}^n$ the set of boxes in $\mathbb{R}^n$, and $|[x]|$ denotes the length of an interval $[x] \subset \mathbb{R}$.

### 2.2.1 Operations on intervals and boxes

Set-theoretic operations, such as intersection and union, are applicable to intervals. The intersection of two intervals is always an interval, whilst their union is not necessarily an interval. The *interval union*, denoted by $\sqcup$, of two intervals $[x],\ [y] \subset \mathbb{R}$ is defined as:

$$[x] \sqcup [y] = [[x] \cup [y]], \tag{2.3}$$

where $\cup$ refers to the set-theoretic union operation. The symbol [.] denotes the *interval hull* operator returning, for any set $S$ in $\mathbb{R}$, the smallest interval enclosing $S$ [56].

Binary operations $\{+, -, \times, /\}$ can be extended to intervals [56]. Let $\diamond$ refer to any binary operation and $[x]$ and $[y]$ denote any two intervals in $\mathbb{R}$, the resulting interval $[z] = [x] \diamond [y]$ is defined by the following:

$$[z] = [x] \diamond [y] = [\{x \diamond y \mid x \in [x], y \in [y]\}]. \tag{2.4}$$

If the binary operation $\diamond$ is continuous, *e.g.* in the case of the usual arithmetic operations, the set $\{x \diamond y \in \mathbb{R} | x \in [x], y \in [y]\}$ is an interval. Thus,

$$[x] \diamond [y] = \{x \diamond y \in \mathbb{R} | x \in [x], y \in [y]\}.$$

**Example 2.2.1.** The extension to intervals of the usual arithmetic operations $\{+, -, \times, /\}$ is given by the following expressions where $[x] = [\underline{x}, \overline{x}]$ and $[y] = [\underline{y}, \overline{y}]$ are two intervals in $\mathbb{R}$.

$$\begin{array}{rcl} [x] + [y] & = & [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \\ [x] - [y] & = & [\underline{x} - \overline{y}, \overline{x} - \underline{y}], \\ [x] \times [y] & = & [\min(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}), \max(\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y})], \\ \dfrac{[x]}{[y]} & = & [x] \times [1/\overline{y}, 1/\underline{y}],\ 0 \notin [y]. \end{array} \tag{2.5}$$

Elementary functions, such as exp, ln, cos and sin, can be also easily extended to intervals. Let $f : \mathbb{R} \longmapsto \mathbb{R}$ be a function, its interval counterpart will be denoted by $[f]$ and is defined as

$$[f]([x]) = [\{f(x) | x \in [x]\}]. \tag{2.6}$$

If $f$ denotes a continuous and monotonic function, $[f]([x])$ is simply equal to $f([x])$.

**Example 2.2.2.** ln is a continuous and strictly increasing function over $\mathbb{R}^{+*}$.

$$\forall [x] \subset \mathbb{R}^{+*}, [\ln]([x]) = [\ln(\underline{x}), \ln(\overline{x})].$$

All operations on intervals can be extended to boxes.

**Example 2.2.3.** Let $S \subset \mathbb{R}^n$, $[\mathbf{x}]$ and $[\mathbf{y}]$ two boxes in $\mathbb{R}^n$.

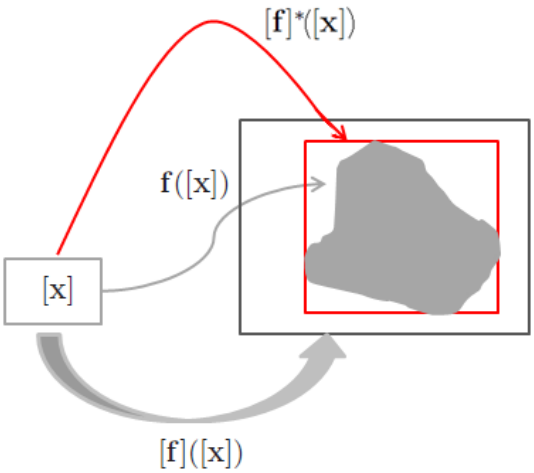



*Fig. 2.1:* The resulting image of a box $[\mathbf{x}]$ by a function $\mathbf{f}$, an inclusion function $[\mathbf{f}]$ for $\mathbf{f}$ and the minimal inclusion function $[\mathbf{f}]^*$.

a. The interval hull of $S$, denoted by $[S]$, is the smallest box of $\mathbb{R}^n$ enclosing $S$.

b. The interval union of $[\mathbf{x}]$ and $[\mathbf{y}]$ is defined as:

$$[\mathbf{x}] \sqcup [\mathbf{y}] = ([x_1] \sqcup [y_1]) \times \ldots \times ([x_n] \sqcup [y_n]).$$

c. Binary operations on real vectors and functions of real vectors may be extended to interval vectors. For instance, $[\mathbf{x}] + [\mathbf{y}] = ([x_1] + [y_1]) \times \ldots \times ([x_n] + [y_n])$.

### 2.2.2 Inclusion functions

Let $\mathbf{f}$ be a function $\mathbf{f} : \mathbb{R}^n \longmapsto \mathbb{R}^m$, $[\mathbf{x}]$ a box in $\mathbb{R}^n$. The image $\mathbf{f}([\mathbf{x}])$ of $[\mathbf{x}]$ by $\mathbf{f}$ is usually not a box (see figure 2.1), and its expression may be quite difficult to obtain. An *inclusion function* approximates $\mathbf{f}([\mathbf{x}])$ [41]. If $[\mathbf{f}]$ denotes an interval function from $\mathbb{IR}^n$ to $\mathbb{IR}^m$ then, by definition, $[\mathbf{f}]$ is said to be an inclusion function for $\mathbf{f}$ if

$$\mathbf{f}([\mathbf{x}]) \subseteq [\mathbf{f}]([\mathbf{x}]), \quad \forall [\mathbf{x}] \in \mathbb{IR}^n. \tag{2.7}$$

Inclusion functions can be very pessimistic, an inclusion function $[\mathbf{f}]$ is *minimal* if, for any $\mathbf{x}$, $[\mathbf{f}]([\mathbf{x}])$ is the interval hull of $\mathbf{f}([\mathbf{x}])$. The minimal inclusion function for $\mathbf{f}$ is unique and will be denoted by $[\mathbf{f}]^*$.

Two main purposes of interval analysis [56] are, first, finding an inclusion function $[\mathbf{f}]$ such that, for most $\mathbf{x}$, $[\mathbf{f}]([\mathbf{x}])$ is close to $[\mathbf{f}]^*([\mathbf{x}])$, and second, finding it with a convenient computational time. Several algorithms have been proposed to reduce the size of the box enclosing $\mathbf{f}([\mathbf{x}])$. The next section presents one such method.

### 2.2.3 Constraints satisfaction and contraction methods

#### 2.2.3.1 Constraints satisfaction problems (CSPs)

Let $\mathbf{x} = (x_1, x_2, \cdots, x_n)^T$ denote a vector of $n$ variables, $x_i \in \mathbb{R}$, $i \in \{1, \ldots, n\}$, $\mathbf{f} = (f_1, f_2, \cdots, f_m)^T$ a multivalued function which satisfies

$$\mathbf{f}(\mathbf{x}) = \mathbf{0}, \tag{2.8}$$

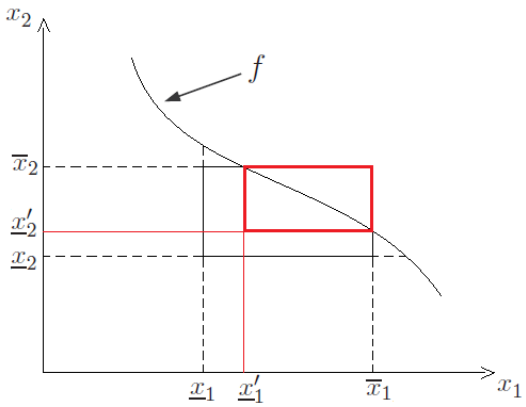



*Fig. 2.2:* An example of a CSP with two variables and a single constraint.

such that the components $f_j, \quad j \in \{1, \ldots, m\}$, of $\mathbf{f}$ represent constraints linking the variables $\{x_i\}_{i=1}^n$. The vector $\mathbf{x}$ belongs to a known prior domain $[\mathbf{x}] \in \mathbb{IR}^n$. A constraint satisfaction problem (CSP) aims at finding the *smallest* box enclosing the set of all $\mathbf{x} \in [\mathbf{x}]$ which satisfy the constraints $\mathbf{f}$. A CSP is commonly denoted as $\mathcal{H}$ and can be formulated as:

$$\mathcal{H} : (\mathbf{f}(\mathbf{x}) = \mathbf{0}, \mathbf{x} \in [\mathbf{x}]). \tag{2.9}$$

The solution set of the CSP $\mathcal{H}$ is given by

$$\mathbf{S} = \{\mathbf{x} \in [\mathbf{x}] \mid \mathbf{f}(\mathbf{x}) = \mathbf{0}\} \tag{2.10}$$

and is not necessarily a box. Solving a CSP refers to finding the smallest box $[\mathbf{x}'] \subset [\mathbf{x}]$ such that $\mathbf{S} \subseteq [\mathbf{x}'] \subseteq [\mathbf{x}]$ [56]. Figure 2.2 illustrates an example of a CSP with two variables $x_1, \ x_2$, and a single constraint $f$. The initial box is $[\underline{x}_1\overline{x}_1] \times [\underline{x}_2\overline{x}_2]$. The solution of the CSP is given by the box $[\underline{x}'_1\overline{x}_1] \times [\underline{x}'_2\overline{x}_2]$ and is highlighted in red, it is included in the initial box and it satisfies the constraint $f$.

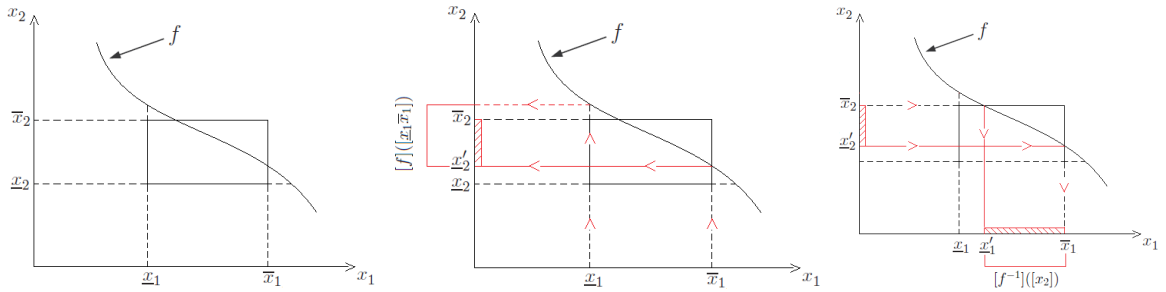
#### 2.2.3.2 Contractors

Contracting $\mathcal{H}$ refers to replacing $[\mathbf{x}]$ by a smaller domain $[\mathbf{x}]'$ such that $\mathbf{S} \subseteq [\mathbf{x}]' \subseteq [\mathbf{x}]$. Several contracting methods are described in [56] (chapter 4), each of which is suitable for different types of CSPs. We will present a general contraction method known as *Constraints Propagation* (CP), this technique is simple, efficient and most importantly independent of non-linearities [41].

The CP method is based on the use of *primitive constraints*. A constraint is said to be primitive if it involves a single binary operation (such as $+, -, \times, /$) or a single elementary function (such as sin, cos, ln, exp). Constraint Propagation technique proceeds by contracting $\mathcal{H}$ with respect to each primitive constraint until convergence to a minimal domain. This method can be implemented using a forward/backward propagation (FBP) procedure [39] which consists of two steps: the forward propagation and the backward propagation. The first step considers the direct forms of the equations. The second uses the inverse of the functions that appear in the equations.

**Example 2.2.4.** We consider the problem illustrated in figure 2.2 and contract the initial domain $[x_1] \times [x_2] = [\underline{x}_1\overline{x}_1] \times [\underline{x}_2\overline{x}_2]$ using the CP technique.

a. *Forward propagation:* Consider the direct form of the constraint $x_2 = f(x_1)$,




*(a)* Initial domain. *(b)* Forward propagation. *(c)* Backward propagation.

*Fig. 2.3:* An example of CP contraction technique.

$[x_2] \leftarrow [\underline{x}_2\overline{x}_2] \cap [f]([\underline{x}_1\overline{x}_1])$.

b. *Backward propagation:* Consider the inverse form of the constraint $x_1 = f^{-1}(x_2)$, $[x_1] \leftarrow [\underline{x}_1\overline{x}_1] \cap [f^{-1}]([x_2])$.

Figure 2.3 shows the operations of CP for this example.

**Example 2.2.5.** *This example is adapted from [41].*

Consider a three dimensional CSP with a single constraint $z = x\exp(y)$. The initial domains are given by $[x] = [1, 7]$, $[y] = [0, 1]$ and $[z] = [0, 3]$. The constraint involves one arithmetic operation ($\times$) and one elementary function (exp), it is not primitive. Introducing an auxiliary variable $a$, the constraint is decomposed into two primitive constraints:

$$\begin{aligned} a &= \exp(y), \\ z &= a \cdot x. \end{aligned} \tag{2.11}$$

An initial domain of $a$ is merely $[a] = [0, +\infty]$. The forward propagation step is achieved via the following two steps, denoted by $F_1$ and $F_2$:

- $F_1$: $[a] \leftarrow [a] \cap [\exp]([y]) = [0, +\infty] \cap [1, e] = [1, e]$
- $F_2$: $[z] \leftarrow [z] \cap [x] \cdot [a] = [0, 3] \cap [1, 7] \cdot [1, e] = [1, 3]$.

The backward propagation step is achieved via three steps, $B_3$, $B_4$ and $B_5$:

- $B_3$: $[x] \leftarrow [x] \cap ([z]/[a]) = [1, 7] \cap [1, 3]/[1, e] = [1, 3]$
- $B_4$: $[a] \leftarrow [a] \cap ([z]/[x]) = [1, e] \cap [1, 3]/[1, 3] = [1, e]$
- $B_5$: $[y] \leftarrow [y] \cap [\ln]([a]) = [0, 1] \cap [0, 1] = [0, 1]$.

The domains of the variables are reduced to $[x] = [1, 3]$, $[y] = [0, 1]$ and $[z] = [1, 3]$. The FBP alternates between forward and backward propagation; in this example, it can be checked that the previous domains, obtained after one forward-backward propagation cycle, will no longer change after another iteration of FBP.

## 2.3 The box Particle Filter

Sequential Bayesian filtering (see section 1.3) consists in estimating the states of a system as a set of observations become available [109]. Unlike the standard Bayesian solutions such as the Kalman Filter (KF) [111] and its variants [59], sequential Monte Carlo [58]

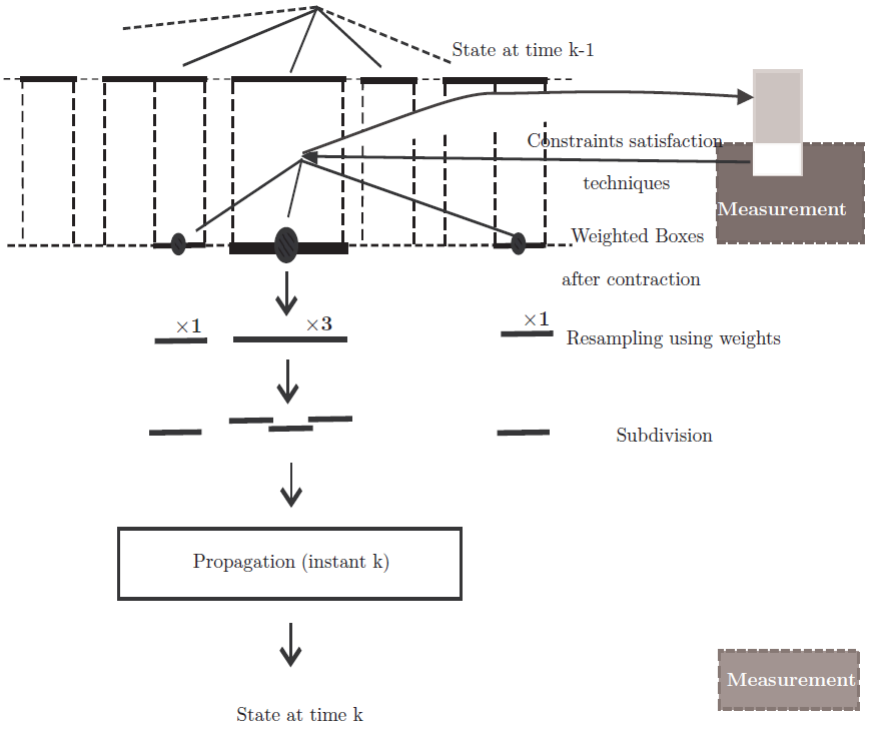



*Fig. 2.4:* The different steps of the box-PF (adapted from [41]).

techniques, also known as Particle Filtering (PF), make no assumptions on the form of the probability densities of interest. We have seen in section 1.4.1 that PFs [6] use sample-based representations in order to construct Monte Carlo approximations of the required integrals (e.g. in equation (1.4)) in the case of seriously non-linear and complex posterior distributions. However, the efficiency of particle filters (PFs) depends on the number of samples and on the proposal functions used for importance sampling. In fact, the imprecision on the available information requires potentially a large number of particles and thus a high computational complexity. The aim of the box-PF is to generalise particle filtering into the bounded error context; instead of propagating a large number of weighted point particles representing the posterior distribution, the Box-PF propagates a small number of weighted box particles through bounded error models and within the interval analysis framework.

The box-PF operates in four steps, for which the Bayesian justification was established by interpreting each box as a uniform PDF in [41]. These steps are: box particle initialisation, time update, measurement update and resampling.

a) **Box particle initialisation:** A prior bounded state space region is split into $N$, equally weighted and mutually disjoint, boxes $\{[\mathbf{x}_0^{(\ell)}]\}_{\ell=1}^{N}$. This initialisation using boxes allows to explore a large prior uncertainty region using only a few box particles.

b) **Time update step:** Knowing the cloud of box particles $\{[\mathbf{x}_{k-1}^{(\ell)}]\}_{\ell=1}^{N}$ representing the state at time step $k-1$, and assuming that the system noise is enclosed in $[\mathbf{v}_k]$, the boxes at step $k-1$ are propagated using interval analysis tools through the transition function $\mathbf{f}$ (refer to equation (1.1))

$$[\mathbf{x}_k^{(\ell)}] = [\mathbf{f}]([\mathbf{x}_{k-1}^{(\ell)}], [\mathbf{v}_k]) \quad for\ \ell = 1, \ldots, N,$$

where $[\mathbf{f}]$ is an inclusion function for $\mathbf{f}$ (refer to section 2.2.2 to recall the definition of an inclusion function).

c) **Measurement update step:** Similarly to particle filtering, the weights of the predicted box particles are updated using the new measurement at time step $k$. For the box-PF, likelihood factors are calculated using *innovation quantities* [42]: the innovation for the $\ell$-th box particle reflects the proximity between the measured box and the predicted box measurement. Hence, the innovation can be represented using the *intersection* between these two boxes. For each box particle, the predicted box measurement has the following expression (see equation (1.2)):

$$[\mathbf{y}_k^{(\ell)}] = [\mathbf{g}]([\mathbf{x}_k^{(\ell)}], [\mathbf{w}_k]), \quad \text{for} \quad \ell = 1, \ldots, N,$$

where $[\mathbf{g}]$ is an inclusion function for $\mathbf{g}$ and $[\mathbf{w}_k]$ is a bounded measurement noise. The innovation is given by:

$$[\mathbf{r}_k^{(\ell)}] = [\mathbf{y}_k^{(\ell)}] \cap [\mathbf{y}_k].$$

In the bounded error context, the likelihood is calculated based on the following idea: if the predicted box measurement does not intersect with the corresponding measured box, this box particle has a likelihood factor equal to zero. In contrast, if the predicted box measurement is included in the corresponding measured box, this box particle has a likelihood close to one [42].

Furthermore, a *contraction step* (see section 2.2.3) is performed in order to eliminate the inconsistent part of the box particles with respect to the measured boxes, and to preserve an appropriate size of the boxes. The *box likelihood* is thus given by:

$$L_k^{(\ell)} = \prod_{j=1}^{n_x} L_k^{(\ell),j},$$

where $n_x$ represents the dimension of the state and the likelihood factor according to a dimension $j$ of $x$ is given by

$$L_k^{(\ell),j} = \frac{|[\tilde{\mathbf{x}}_{k+1}^{(\ell)}(j)]|}{|[\mathbf{x}_{k+1}^{(\ell)}(j)]|}.$$

The term $[\tilde{\mathbf{x}}_{k+1}^{(\ell)}(j)]$ represents the new $\ell$-th box particle after the contraction step.

d) **Resampling step:** Similarly to the PF algorithm, a resampling step is also added in order to introduce variety into the box particles. Different resampling algorithms exist; in [41] for instance, the multinomial resampling combined with a new subdivision step is employed, in this scheme, after the resampling step, each box is divided by its corresponding number of realisations in order to obtain smaller boxes around regions with high likelihoods.

The computations of the box-PF are recapitulated in figure 2.4.

## 2.4 Theoretic derivation of the box-Belief Propagation

This section presents the main theoretical contribution of part I of the thesis: the derivation of the box-Belief Propagation (box-BP), which is a message-passing algorithm used

to infer on graphical models when probability quantities are represented using boxes. The box-BP is indeed an extension of the BP algorithm into the bounded error framework.

Inspired by the theoretic derivation of the box-PF presented in [41, 42], we show the theoretic derivation of BP in interval analysis framework by interpreting a box as a uniform PDF. Note, beforehand, that an advantage offered by this interpretation is that, while strictly speaking, a set of samples constitute a "representation" of a PDF, a set of boxes constitute a direct approximation of the PDF.

Going back to equation (1.20) representing BP's message update (for a message sent from a node $t$ to its neighbour $s$, at iteration $i$ of BP), recall that this iteration can be divided into two operations: the message product, or the computation of the partial belief given in equation (1.22), and the convolution operation. We demonstrate the extension to intervals of each operation in the following two subsections. A summary of the operations of the novel box-BP is illustrated in figure 2.5.

### 2.4.1 The message product operation

Let $\Gamma_t$ denote the set of neighbours of the sending node $t$, $u_l \in \Gamma_t \backslash s$ a node connected to $t$ by an edge, except for the receiving node $s$, and $d = card(\Gamma_t \backslash s)$; then the message product can be expressed as:

$$R_{ts}^{i-1}(\mathbf{x}_t) \propto \prod_{u \in \Gamma_t \backslash s} m_{ut}^{i-1}(\mathbf{x}_t) = \prod_{l=1}^{d} m_{u_l t}^{i-1}(\mathbf{x}_t). \tag{2.12}$$

In interval analysis framework, the messages are represented using $N$ weighted boxes, thus, a message $m_{u_l t}(\mathbf{x}_t)$ received by node $t$ from its neighbour $u_l$, $l = 1, \ldots, d$, is represented by a cloud of boxes $\{\omega_{u_l t}^{p_l}, [\mathbf{x}_{u_l t}^{p_l}]\}_{p_l=1}^{N}$. Note that, for simplicity reasons, in this representation we skip the index $i$ referring to the message passing iteration number on the particles and their weights. We are also assuming that all messages are represented using the same number of boxes, $N$. This reasoning may however be easily extended to a more general case with varying number of box particles per node/message.

Since each box is interpreted as a uniform probability distribution, we have:

$$m_{u_l t}^{i-1}(\mathbf{x}_t) = \sum_{p_l=1}^{N} \omega_{u_l t}^{p_l} U_{[\mathbf{x}_{u_l t}^{p_l}]}(\mathbf{x}_t), \text{ for } l = 1, \ldots, d, \tag{2.13}$$

where $U_{[\mathbf{x}]}$ denotes the uniform PDF over the box $[\mathbf{x}]$.

Let $P$ denote the vector of indexes $(p_0, p_1, \ldots, p_d)$ and $I = \{1, 2, \ldots, N\}$. Replacing expression (2.13) in equation (2.12), we obtain

$$\begin{aligned} R_{ts}^{i-1}(\mathbf{x}_t) &\propto \prod_{l} \Big( \sum_{p_l=1}^{N} \omega_{u_l t}^{p_l} U_{[\mathbf{x}_{u_l t}^{p_l}]}(\mathbf{x}_t) \Big) \\ &\propto \sum_{P \in I^d} \omega_{u_1 t}^{p_1} \ldots \omega_{u_d t}^{p_d} U_{[\mathbf{x}_{u_1 t}^{p_1}]} \ldots U_{[\mathbf{x}_{u_d t}^{p_d}]}(\mathbf{x}_t). \end{aligned} \tag{2.14}$$

A uniform PDF $U_{[\mathbf{x}]}$ is constant over its support (and equal to $1/|[\mathbf{x}]|$) and equal to *zero* everywhere else. Note that, by abuse of notation, $|[\mathbf{x}]|$, where $\mathbf{x} \in \mathbb{IR}^{n_x}$ and $n_x$ is the state space dimension, denotes the product $\prod_{\ell=1}^{n_x} |[\mathbf{x}(\ell)]|$, *e.g.* $|[\mathbf{x}]|$ denotes the area of the rectangle defined by $\mathbf{x}$ in 2D, the volume of the rectangular cuboid in 3D, etc.

The product $U_{[\mathbf{x}_{u_1t}^{p_1}]} \dots U_{[\mathbf{x}_{u_dt}^{p_d}]}(\mathbf{x}_t)$ is then different to *zero* if $\mathbf{x}_t$ belongs to the intersection of the supports of its terms. This product may hence be modelled using a uniform PDF given by

$$\begin{aligned} U_{[\mathbf{x}_{u_1t}^{p_1}]} \dots U_{[\mathbf{x}_{u_dt}^{p_d}]}(\mathbf{x}_t) \;&=\; U_{[\mathbf{x}_{u_1t}^{p_1}]\cap\dots\cap[\mathbf{x}_{u_dt}^{p_d}]} \\ &\times \frac{|[\mathbf{x}_{u_1t}^{p_1}] \cap \dots \cap [\mathbf{x}_{u_dt}^{p_d}]|}{|[\mathbf{x}_{u_1t}^{p_1}]| \dots |[\mathbf{x}_{u_dt}^{p_d}]|}. \end{aligned} \tag{2.15}$$

Then

$$\begin{aligned} R_{ts}^{i-1}(\mathbf{x}_t) \;\propto\; &\sum_{P\in I^d} \omega_{u_1t}^{p_1} \dots \omega_{u_dt}^{p_d} U_{[\mathbf{x}_{u_1t}^{p_1}]\cap\dots\cap[\mathbf{x}_{u_dt}^{p_d}]} \\ &\times \frac{|[\mathbf{x}_{u_1t}^{p_1}] \cap \dots \cap [\mathbf{x}_{u_dt}^{p_d}]|}{|[\mathbf{x}_{u_1t}^{p_1}]| \dots |[\mathbf{x}_{u_dt}^{p_d}]|}. \end{aligned} \tag{2.16}$$

The number of possible assignments of $P$ is $N^d$. Thus, the sum above contains at most $N^d$ terms. In practice, this number is much less since some assignments result in empty intersections while others result in coincident boxes.

Let us denote by $Q$ the set of assignments of $P$, for which $[\mathbf{x}_{u_1t}^{p_1}] \cap \dots \cap [\mathbf{x}_{u_dt}^{p_d}] \neq \emptyset$. We will also adopt the following notations for an assignment $P^k \in Q$, where $k = 1, \dots, card(Q)$:

$$\omega_{u_1t}^{p_1} \dots \omega_{u_dt}^{p_d} \times \frac{|[\mathbf{x}_{u_1t}^{p_1}] \cap \dots \cap [\mathbf{x}_{u_dt}^{p_d}]|}{|[\mathbf{x}_{u_1t}^{p_1}]| \dots |[\mathbf{x}_{u_dt}^{p_d}]|} = \omega_{ts}^k, \tag{2.17}$$

$$[\mathbf{x}_{u_1t}^{p_1}] \cap \dots \cap [\mathbf{x}_{u_dt}^{p_d}] = [\mathbf{x}_{ts}^k]. \tag{2.18}$$

The message product given in equation (1.21) is now represented by the cloud of boxes $\{\omega_{ts}^k, \mathbf{x}_{ts}^k\}$. Based on equation (2.16), we can define an algorithm to perform the message product operation.

**Example 2.4.1.** Algorithm 1 describes the method for combining *two* messages according to equation (2.16); note that both operations, ($\times$) and ($\cap$) are associative operations. In this algorithm, each message is represented using a certain number of weighted boxes, *e.g.*, $\{\omega_1^i, [\mathbf{x}_1^i]\}_{i=1}^{N_1}$, $\{\omega_2^j, [\mathbf{x}_2^j]\}_{j=1}^{N_2}$ denote the collection of weighted boxes representing message 1 and message 2 respectively. The cloud $\{\omega_{res}^k, [\mathbf{x}_{res}^k]\}_{k=1}^{N_{res}}$ denotes the resulting set of boxes and the corresponding weights according to equations (2.17) and (2.18), *iscoincident()* is a function that takes as arguments two boxes and returns 1 if the boxes are coincident according a threshold and 0 if not.

#### *2.4.1.1 Computation of the partial belief*

The partial belief is now given by the following expression:

$$M_{ts}^{i-1}(\mathbf{x}_t) \propto \psi_t(\mathbf{x}_t) \sum_{k=1}^{card(Q)} \omega_{ts}^k U_{[\mathbf{x}_{ts}^k]}(\mathbf{x}_t). \tag{2.19}$$

where, for simplicity and without loss of generality, $\psi_t(\mathbf{x}_t)$ replaces $\psi_t(\mathbf{x}_t, \mathbf{y}_t)$ under the assumption of an observation $\mathbf{y}_t$ of the same nature as the state $\mathbf{x}_t$ as stated in section 1.7. Within the bounded error framework, we assume that $\mathbf{y}_t$ is known within a bounded box $[\mathbf{y}_t]$ and the potential $\psi_t(\mathbf{x}_t)$ is approximated equal to the uniform PDF $U_{[\mathbf{y}_t]}(\mathbf{x}_t)$.

**Algorithm 1** Message product

**Input:** $\{\omega_1^i, [\mathbf{x}_1^i]\}_{i=1}^{N_1}$, $\{\omega_2^j, [\mathbf{x}_2^j]\}_{j=1}^{N_2}$
**Output:** $\{\omega_{res}^k, [\mathbf{x}_{res}^k]\}_{k=1}^{N_{res}}$

```
k ← 0
for i = 1, ..., N_1 do
  for j = 1, ..., N_2 do
    if [x_1^i] ∩ [x_2^j] ≠ ∅ then
      k = k + 1
      [x_res^k] = [x_1^i] ∩ [x_2^j]
      ω_res^k = ω_1^i × ω_2^j × |[x_1^i]∩[x_2^j]| / (|[x_1^i]|.|[x_2^j]|)
    end if
  end for
end for
loop
  k_1, k_2 ∈ {1, ..., k}
  if iscoincident([x_res^{k_1}], [x_res^{k_2}]) then
    keep only one occurrence
    sum up the corresponding weights
  end if
end loop
Weights normalisation: ω_res^k ⟵ ω_res^k / sum(w_res)
N_res = length(ω_res)
```

In the general case, the potential function $\psi_t(\mathbf{x}_t, \mathbf{y}_t)$ can be represented with the uniform PDF $U_{[\mathbf{y}_t]}(\mathbf{g}(\mathbf{x}_t))$, where $\mathbf{g}$ is a function linking the local observation $\mathbf{y}_t$ at node $t$ to the local state $\mathbf{x}_t$ (observation model), $\mathbf{y}_t = \mathbf{g}(\mathbf{x}_t)$. Using contraction techniques [56], the resulting weighted boxes $\{\omega_{ts}^k, [\mathbf{x}_{ts}^k]\}_{k=1}^{card(Q)}$ can be contracted and re-weighted. Under the assumption of an observation of the same nature as the state, the contraction reduces into an intersection between the resulting boxes and the observation. *Algorithm 2* summarises the procedure of contracting the boxes $\{\omega_{ts}^k, [\mathbf{x}_{ts}^k]\}_{k=1}^{card(Q)}$ using the measurement $[\mathbf{y}_t]$, the resulting weighted boxes are denoted by $\{\omega_{ts,c}^k, [\mathbf{x}_{ts,c}^k]\}_{k=1}^{N_c}$ where subscript $c$ stands for contracted. In this procedure $[\mathbf{g}]$ denotes an inclusion function for $\mathbf{g}$.

As a final remark, note that the belief at node $t$ as given in equation (1.23) can be computed using the same procedure derived in this section.

**Algorithm 2** Contraction using the local observation

**Input:** $\{w_{ts}^k, [\mathbf{x}_{ts}^k]\}_{k=1}^{card(Q)}$, $[\mathbf{y}_t]$
**Output:** $\{w_{ts,c}^k, [\mathbf{x}_{ts,c}^k]\}_{k=1}^{N_c}$
**for** $k = 1, \dots card(Q)$ **do**
*Predicted measurement:* $[\mathbf{y}^k] = [\mathbf{g}]([\mathbf{x}_{ts}^k]$
Innovation: $[\mathbf{r}^k] = [\mathbf{y}^k] \cap [\mathbf{y}_t]$
*Box particle contraction:*
**if** $[\mathbf{r}^k] \neq \emptyset$ **then**
contract $[\mathbf{x}_{ts}^k]$ using CP algorithm
the resulting box is denoted $[\mathbf{x}_{ts,c}^k]$
**else**
$[\mathbf{x}_{ts,c}^k] = \emptyset$
**end if**
*Re-weighting:* $w_{ts,c}^k \longleftarrow w_{ts}^k \times \frac{|[\mathbf{x}_{ts,c}^k]|}{|[\mathbf{x}_{ts}^k]|}$
*Weights normalisation:* $w_{ts,c}^k \longleftarrow w_{ts,c}^k / sum(w_{ts,c})$.
**end for**
$N_c = length(w_{ts,c} \neq 0)$

### 2.4.2 The convolution operation

The next step in the message update iteration is the convolution of the partial belief with the pairwise potential function:

$$\begin{aligned} m_{ts}^i(\mathbf{x}_s) &= \int \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) M_{ts}^{i-1}(\mathbf{x}_t) d\mathbf{x}_t \\ &= \int \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) \sum_{k=1}^{N_c} \omega_{ts,c}^k U_{[\mathbf{x}_{ts,c}^k]}(\mathbf{x}_t) \\ &= \sum_{k=1}^{N_c} \omega_{ts,c}^k \int_{[\mathbf{x}_{ts,c}^k]} \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) U_{[\mathbf{x}_{ts,c}^k]} d\mathbf{x}_t \\ &= \sum_{k=1}^{N_c} \omega_{ts,c}^k \frac{1}{|[\mathbf{x}_{ts,c}^k]|} \int_{[\mathbf{x}_{ts,c}^k]} \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) d\mathbf{x}_t. \end{aligned} \tag{2.20}$$

To evaluate the integral in equation (2.20), consider the general case where the potential $\psi_{ts}(\mathbf{x}_t, \mathbf{x}_s)$ can be represented as a transition function by which we can pass from $\mathbf{x}_t$ to $\mathbf{x}_s$, i.e. $\mathbf{x}_s = f(\mathbf{x}_t, \mathbf{v}; \mathbf{e})$ where $\mathbf{v}$ denotes a noise and $\mathbf{e}$ refers to a vector of constants (e.g. some known parameters of the model), and let $[f]$ be an inclusion function for $f$. We assume that the noise $\mathbf{v}$ is bounded in the box $[\mathbf{v}]$.

Then, by definition of an inclusion function, we have

$$\forall \mathbf{x}_t \in [\mathbf{x}_{ts,c}^k], \quad \mathbf{x}_s \in [\mathbf{f}]([\mathbf{x}_{ts,c}^k], [\mathbf{v}]; [\mathbf{e}]) \; for \; k \in \{1, \dots, N_c\},$$

i.e.

$$\psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) U_{[\mathbf{x}_{ts,c}^k]}(\mathbf{x}_t) = 0, \quad \forall \mathbf{x}_s \notin [\mathbf{f}]([\mathbf{x}_{ts,c}^k], [\mathbf{v}]; [\mathbf{e}]). \tag{2.21}$$

Equation (2.21) shows that for any transition function $\mathbf{f}$, using interval analysis techniques, the support for the PDF terms $\int_{[\mathbf{x}_{ts,c}^k]} \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) U_{[\mathbf{x}_{ts,c}^k]} d\mathbf{x}_t$ can be approximated by

$[\mathbf{f}]([\mathbf{x}_{ts,c}^k],[\mathbf{v}];[\mathbf{e}])$ and thus:

$$\int_{[\mathbf{x}_{ts,c}^k]} \psi_{ts}(\mathbf{x}_t,\mathbf{x}_s)U_{[\mathbf{x}_{ts,c}^k]}d\mathbf{x}_t \;\approx\; U_{[\mathbf{f}]([\mathbf{x}_{ts,c}^k],[\mathbf{v}],[\mathbf{e}])}. \tag{2.22}$$

Note that, it is shown in [41] that the approximation (2.22) can be done more precisely at a computation cost using a mixture of uniform boxes, *e.g.*, a cloud of box particles. For simplicity and without loss of generality only one box is used in this chapter.

Based on equations (2.20) and (2.22), we can perform the message update procedure, for a message sent from node $t$ to its neighbour $s$, according to Algorithm 3: once the incoming messages to the sending node $t$ are combined as depicted in Algorithm 1 and contracted using Algorithm 2, the collection of weighted boxes $\{\omega_{ts,c}^k,[\mathbf{x}_{ts,c}^k]\}_{k=1}^{N_c}$, combining all information about $\mathbf{x}_t$, are propagated through the model $\mathbf{x}_s=\mathbf{f}(\mathbf{x}_t,\mathbf{v},\mathbf{e})$ to obtain an information about $\mathbf{x}_s$. The resulting set $\{\omega_{ts,r}^\ell,[\mathbf{x}_{ts,r}^\ell]\}_{\ell=1}^{N}$, where the subscript $r$ stands for "resulting", represent the message sent from $t$ to $s$. Note that for the first iteration of message-passing between neighbouring nodes, messages are initialised using the local marginal PDF since, obviously, no previous messages exist. If no observation is available for the sending node $t$, no message can be sent, at the first iteration, from $t$ to any of its neighbours. If, however, an observation is available for $t$, we use it to generate uniformly $N$ boxes mutually disjoint and equally weighted with $\frac{1}{N}$.

**Algorithm 3** Message update

$MsgCounter \leftarrow 0$
**for** $u \in \Gamma_t$ and $u \neq s$ **do**
  **if** a message is incoming from $u$ to $t$ **then**
    $MsgCounter \leftarrow MsgCounter + 1$
  **end if**
**end for**
Message product
**if** $MsgCounter == 0$ **then**
  **if** an observation is available for $t$ **then**
    use the measurement to generate uniformly $N$ boxes mutually disjoint and weighted equally with $\frac{1}{N}$
  **else**
    no message can be forwarded from $t$ to $s$
  **end if**
**else**
  use Algorithm 1 and the associativity of $(\cap)$ to combine the messages
  contract the resulting boxes using the observation according to Algorithm 2
  the resulting cloud of boxes is $\{\omega_{ts,c}^k,[\mathbf{x}_{ts,c}^k]\}_{k=1}^{N_c}$
**end if**
propagate the boxes through the model $\mathbf{x}_s = \mathbf{f}(\mathbf{x}_t,\mathbf{v};\mathbf{e})$, get the resulting boxes $\{\omega_{ts,r}^\ell,[\mathbf{x}_{ts,r}^\ell]\}_{\ell=1}^{N}$
Weights normalisation: $\omega_{ts,r}^\ell \leftarrow \omega_{ts,r}^\ell / sum(\omega_{ts,r})$.

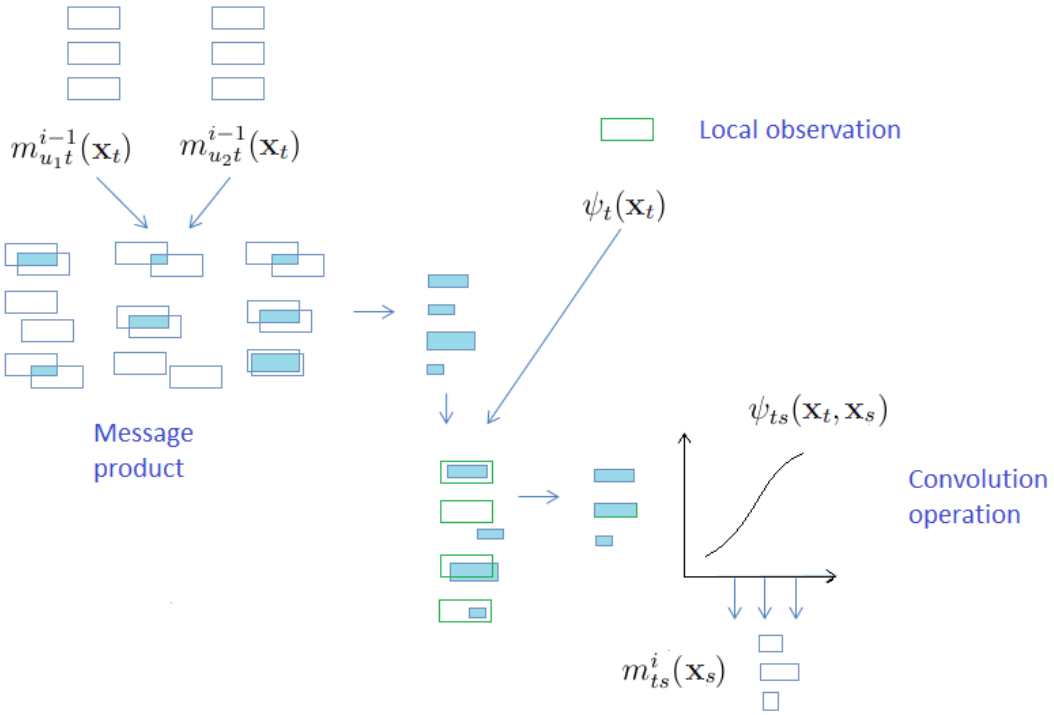



*Fig. 2.5:* The box-BP algorithm.

## 2.5 The box-BP: a generalisation of the box-PF

Same as the NBP generalises the PF to general inference problems on arbitrary graphical models, wed can demonstrate that the box-BP is also a generalisation of the box-PF.

Consider the special case of a temporal Markov chain. At each time step $k$ an observation $\mathbf{y}_k$ is available (refer to figure 1.2). A convenient parametrisation of the graph is as follows:

$$\psi_{k-1,k}(\mathbf{x}_{k-1}, \mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1}), \tag{2.23}$$

$$\psi_k(\mathbf{x}_k, \mathbf{y}_k) = p(\mathbf{y}_k|\mathbf{x}_k). \tag{2.24}$$

In the first equation, we recognise the transition probability density while in the second the likelihood at time instant $k$.

In the first iteration, a message $m_{01}(x_1)$ will be sent from node $\mathbf{x}_0$ to node $\mathbf{x}_1$, it is initialised using a prior domain $[\mathbf{x}_0]$ for $\mathbf{x}_0$: $[\mathbf{x}_0]$ is split into $N$, equally weighted and mutually disjoint, boxes $\{[\mathbf{x}_0^\ell]\}_{\ell=1}^N$. This corresponds to the box-particle initialisation step.

To form the message $m_{01}(\mathbf{x}_1)$, the resulting boxes are propagated through the transition function describing the pairwise potential function $\psi_{01}(\mathbf{x}_0, \mathbf{x}_1) = p(\mathbf{x}_1|\mathbf{x}_0)$ according to Algorithm 3. Thus the message $m_{01}(\mathbf{x}_1)$ corresponds to the prior for $\mathbf{x}_1$. This corresponds to the time update step.

The belief at node $\mathbf{x}_1$ is computed by the product of the only message received by $\mathbf{x}_1$ from $\mathbf{x}_0$, that is the prior at time instant 1, with $\psi_1(\mathbf{x}_1, \mathbf{y}_1) = p(\mathbf{y}_1|\mathbf{x}_1)$ which is the likelihood, the result is the posterior at time instant 1. In this step, innovation quantities are calculated, boxes are contracted and re-weighted according to Algorithm 2. This corresponds to the measurement update step. In order to introduce variety into the box particles, a resampling step can be added. Note that the partial belief at time instant 1 is equal to the belief at this node.

In general, the interpretation of the messages and (partial) beliefs at time step $k$ is given by

$$M_{k-1,k}(\mathbf{x}_{k-1}) = p(\mathbf{x}_{k-1}|\mathbf{y}_{k-1}, \ldots, \mathbf{y}_1), \quad \text{the posterior at } k-1,$$
$$m_{k-1,k}(\mathbf{x}_k) = p(\mathbf{x}_k|\mathbf{y}_{k-1}, \ldots, \mathbf{y}_1), \quad \text{the prior at } k.$$

In terms of messages and beliefs, the box particle filter can be described as follows: Given a collection of weighted samples $\{\omega_{k-1}^{\ell}, \mathbf{x}_{k-1}^{\ell}\}_{\ell=1}^{N}$ representing the partial belief $M_{k-1,k}(\mathbf{x}_{k-1})$, which is the posterior at time step $k-1$, the message $m_{k-1,k}(\mathbf{x}_k)$ is obtained by propagating $\{\mathbf{x}_k^{\ell}\}_{\ell=1}^{N}$ through the transition density, given by the process model and representing the pairwise potential function. The resulting cloud of boxes represents the prior at time step $k$. Next, the belief at node $\mathbf{x}_k$ is computed by multiplying the only message received by $\mathbf{x}_k$ with the local likelihood $\psi_k(\mathbf{x}_k, \mathbf{y}_k) = p(\mathbf{y}_k|\mathbf{x}_k)$. The result is the posterior at $k$.

Consequently, the box-particle filter is seen as a special case of the box-Belief Propagation, applicable to temporal inference problems. Equivalently, the box-BP generalises the box-PF techniques to arbitrary structured graphical models.

## 2.6 Conclusion

This chapter presented the main theoretical contribution of part I of this dissertation. It introduced a novel scheme of message-passing using interval representation of probability quantities. The purpose was to derive a BP applicable for graphical models associated with general statistical probability densities while avoiding the overhead of storing thousands of particles and the use of Kernel density estimates and the Gibbs sampler at each iteration of the NBP introduced in section 1.7. The efficiency of this novel box-BP will be tested in the following two chapters, on two different applications: the problem of self localisation in static WSNs and that of continuously localising a mobile target moving within a network of wireless range sensors.

# 3. SELF LOCALISATION IN WIRELESS SENSOR NETWORKS

## *3.1 Introduction*

Many industrial, scientific and domestic applications employ sensor networks whenever there is a need to monitor, and possibly control, physical quantities. Sensors acquire information from their environment (*e.g.* temperature or pressure, depending on their sensing ability) and usually communicate the data collected to a processing centre. However, a vast majority of applications in sensor networks deploys a large number of sensors *randomly*, usually due to the hostility of the area to be monitored, or its immensity. The sensors are thus randomly dropped, from a flying agent for instance, and fall into the region of interest with practically no information about their positions. The localisation of the sensors is often necessary to make the data collected informative.

In this chapter, we consider the problem of self localisation in wireless sensor networks. Since all sensors cannot be equipped with a positioning unit, *e.g.* GPS module, due to cost and energy constraints, each sensor is alternatively equipped with a transmitter-receiver module and communicates with neighbouring sensors. Hence, each sensor is within the vicinity of only a few nodes of the network, and its position can be estimated by exchanging information with several nearby sensor nodes, with which a communication can be established. An attractive formalism to represent this system while visualising the local independence conditional relationships and the underlying structure of the associated joint PDF is possible through the use of graphical models. A literature review on self localisation in sensor networks is provided in section 3.2 then the focus is oriented towards MDS (multidimensional scaling) based localisation algorithms in section 3.3. Since the problem of auto-localisation in wireless sensor networks is pre-eminently a distributed inference problem, we formulate it as inference on a graphical model [52] and use the BP algorithm to perform the task of inference in section 3.4, the effectiveness of the novel box-BP approach is then tested on simulated data. Section 3.5 at last concludes this chapter.

## *3.2 Literature review*

The problem of auto-localisation in WSNs have been widely investigated and a great deal of methods were proposed in the literature, most of which rely on the premise that a sensor node is able to communicate with its nearby counterparts in the network. Several classifications of localisation methods have been proposed. We cite for instance *range-based* versus *range-free* approaches. Range-based methods assume that sensors have capabilities to estimate distances with their neighbours, using technologies such as RSSI (Received Signal Strength Indicator), TOA (Time of Arrival) and TDoA (Time Difference of Arrival) [52], or angles, using AoA (angle of Arrival) [89]. These techniques vary in their accuracy and complexity and can be divided into radio frequency (RF) ranging and acoustic ranging. The RF ranging relies on the premise that, by measuring the received signal strength, a receiver can determine its distance to a transmitter. The second class of rang-

ing schemes measures the time difference of arrival of emitted and received signals [44]. In contrast to range-based algorithms, range-free [47] schemes do not need point-to-point distance or angle estimation for positioning. Approaches in [37, 94] for instance do not involve any range or angle measurement but rather rely on measures of connectivity.

Another classification subdivides the approaches for self localisation in wireless sensor networks to *anchor-based* and *anchor-free*. In both schemes, nodes must themselves determine their respective positions through cooperation techniques. In fact, the goal of a self localisation algorithm is to calculate the coordinates of each sensor based on proximity measures. The computed coordinates can be global, which requires the position of a number of anchors to be known a priori, these approaches are known as anchor-based [9]. Other methods create a relative map without the use of anchors and are called anchor-free [5]. In range-based techniques, the placement of the anchors can often have a significant impact on the solution: it was found that the location accuracy improves if the anchors form a convex polygon around the network [64]. Additional anchors placed at the centre of the network are nonetheless useful. Note that, other algorithms, related to auto-localisation in WSNs, were proposed to allow each node to locate its neighbours *qualitatively*; work in [48] offers one such example of a localised algorithm whose purpose is, for each node, to classify its neighbours into one of three categories: very close, near and far, rather than estimating distances separating them.

Each sensor has limited resources (e.g. bandwidth, battery energy, memory capacity, emission power), it can detect and communicate with other nodes in the network only within some maximum span. Early approaches to solving the problem of localisation in wireless sensor networks were proposed in a *centralised* environment. However, this strategy does not comply the principal energy constraint and is not appropriate for large-scale networks. Indeed, local data processing has low energy cost, (see [26] for details about energetic cost of RF communication in function of distance). In addition, the reliability of the centralised approach is low because a failure in the main processing unit causes the entire system to fail. Therefore, *distributed* approaches [52, 37] were proposed. A third approach is to assist the nodes of the network by a mobile anchor enabling them to locate themselves. This approach [9] has many advantages in terms of energy and location accuracy but its implementation can be expensive and is not always feasible as some areas can be hostile to moving robots.

We can also distinguish *optimisation* methods [37, 94] and *probabilistic* schemes [88]. While optimisation approaches provide a single optimal solution that minimises or maximises some criterion, the solution in probabilistic methods takes the form of a probability distribution therefore allowing to quantify the uncertainty on the estimated positions. Work in [88] presents a probabilistic method for locating sensors based on AoA technology, in this article the error on the measured angles is modelled using a Gaussian PDF. However, for some environments, uncertainties which are usually modelled as probabilistic can be complex; non-parametric methods offer an attractive framework to address these kind of problems [49]. In [52] for instance, the auto-localisation problem is formulated as inference on a graphical model. The Non-parametric Belief Propagation (NBP) algorithm described in section 1.7 is used to combine the information obtained from a global positioning system, with measures of relative distances between neighbouring sensors. Recall that the NBP algorithm is itself a variant of BP, where a set of particles is used to represent probability quantities in a non-parametric way.

Furthermore, in some problems, uncertainties or noise characteristics are unknown or complex; instead, only minimum and maximum values of the noise are available, *e.g.* quantised measurements. The interval analysis framework offers promising methodologies

for reasoning in the presence of unknown or complex statistical but bounded noises [41].

In this chapter, we propose a Bayesian probabilistic scheme for self localisation in WSNs based on a variant of BP algorithm where information is represented using a collection of boxes (intervals in the case of real variables). The use of this approach involves simplicity in modelling the information and memory optimisation. In fact, we will demonstrate that the use of interval representations in our approach offers many advantages, basically:

a. a reduction of the memory space required to store a PDF (thousands of particles are needed to efficiently represent a PDF using Monte Carlo methods while only a few box-particles are required to approximate a probability distribution).

b. energy saving since less information is exchanged between communicating sensors, this also implies a reduction of the required bandwidth.

c. using interval techniques results in simpler and faster computations and thus more time saving.

d. a set of boxes constitute a direct approximation of the PDF while a set of samples constitute a representation of a PDF. This fact allows to save the energy and time needed to estimate the PDF using Kernel Density Estimation (KDE) techniques in the case of the standard sample-based NBP. This also introduces less approximations since the kernel is artificially added to the particles.

## 3.3 Multidimensional scaling for self localisation in WSNs

Multidimensional scaling (MDS) is a data analysis technique that transforms proximity information into a geometric embedding [29]. Formally, given $n$ points in a $p$-dimensional space $E$ and a measure of dissimilarity reflecting, for a pair $(i, j),\ i, j \in \{1, \ldots, n\}$, the remoteness between points $i$ and $j$, the purpose of the classical Multidimensional scaling is to re-place these $n$ points in a new subspace (which is often Euclidean) of a size $q \leq p$ in such a manner that the inter-point distances in this subspace approximate at best the dissimilarities.

MDS is at the heart of several approaches for self localisation in WSNs, the most basic form of which is the MDS-MAP(C) algorithm introduced in [94] in a centralised context. In a typical scenario for auto-localisation of wireless sensors, each sensor has a limited communication range allowing it to only compute the distances separating it from its neighbours. In [94], MDS is used to solve the localisation problem after a preliminary treatment which makes it possible to determine a dissimilarity matrix. The dissimilarity matrix is in fact a distance matrix constructed by calculating the shortest path between each pair of nodes in the network; for instance, the algorithm of Dijkstra can be employed to determine optimal paths between pairs of nodes along the arcs of the network within a time complexity $O(N^3)$. MDS-MAP(C) operates in three steps:

a. Compute the dissimilarity, *i.e.* distance, matrix given the network connectivity information.

b. Use classical MDS to derive relative node coordinates that fit those distances. The network is reconstructed by only considering the $q$ eigenvectors having the highest energy ($q = 2$ or 3 depending on whether the localisation procedure operates in 2D or 3D). Note that MDS offers the best approximation by a lower rank matrix in terms of least squares [37].

c. Reset the absolute positions of the nodes using the known coordinates of a number $m$ of anchors within $O(m) + O(n)$ time complexity.

MDS-MAP(C) offers a good performance on networks with relatively *uniform node density*. On irregular networks, its performance degrades since the shortest path does not correspond well to the Euclidean distance. To tackle this difficulty, the same reference [94] proposes a variant of the MDS-MAP(C) employing a distributed implementation of it. The resulting algorithm is referred to as MDS-MAP(P) and operates by building, for each node, a local map of the small sub-network within the vicinity of it, and then merging all local maps together to form a global one.

Other approaches employing MDS for auto-localisation in WSNs can be found in [27, 31, 106]. In [27] for instance, a scalable and distributed weighted-multidimensional scaling (dwMDS) algorithm is proposed to adaptively emphasise the most accurate range measurements and thus account for communication constraints within the network. The sensors localisation task is formulated in [31] as a non-linear least squares (NLS) problem and is robustly initialised using MDS. A different type of MDS (called ordinal MDS) is used in[106] for self localisation in WSNs, the ordinal MDS differs from its classical counterpart by that it only requires, for each pair of nodes, a monotonicity constraint between the actual Euclidean distance and the shortest path distance.

## 3.4 Self localisation in WSNs as inference on a graphical model

In this section, the problem of self localisation in a WSN is formulated as inference on a graphical model whose nodes are associated with randomly deployed sensors. Only few nodes of the network, called anchors, have a prior information about their positions; these sensors are placed manually at well known locations or equipped with a GPS module and randomly dropped, along with other sensors, if manual placement is impractical. Each sensor is able to communicate with nearby nodes and to estimate the distance separating them. The purpose is to form a global map of the nodes by combining the information available at the anchors, *e.g.* obtained from a global positioning system, with measures of relative distances, by exchanging local messages between neighbouring sensors using the box-BP scheme introduced in chapter 2. After convergence of this iterative procedure, each sensor is left with an estimation of its 2D position. This formulation (using a graphical model) is compliant with the distributed aspect of the problem we are considering since it allows to exploit the local independence conditional relationships between the sensor nodes. The use of a message-passing BP scheme is also advantageous since BP is a distributed algorithm which can be employed to infer on a graphical model even when the whole structure of the graph is not known a priori. In the following, the performance of the box-BP will be also compared to that of the standard particle-based NBP and the MDS-MAP(C) algorithm reported in the previous section.

### 3.4.1 Problem formulation

A number $n$ of sensors is randomly deployed in a planar region. Each sensor computes noisy measurements of its distances from its neighbouring counterparts, the estimated distance is supposed symmetric for both communicating nodes. Two sensors are able to communicate if the distance separating them is less than some maximum range denoted as $R$, in this case they are said to be neighbours. The position of a sensor $t$ is denoted as $\mathbf{x}_t, \quad t = 1, ..., n$, and only a small number of anchor nodes have significant a priori information, $p_t(\mathbf{x}_t)$, about their positions.

Let $d_{ts}$ denote the noisy measurement of the distance between sensors $s$ and $t$, then:

$$d_{ts} = ||\mathbf{x}_t - \mathbf{x}_s|| + v_{ts}, \quad v_{ts} \sim p_v(\mathbf{x}_t, \mathbf{x}_s), \tag{3.1}$$

where $v_{ts}$ refers to a noise, $p_v(\mathbf{x}_t, \mathbf{x}_s)$ denotes a noise probability distribution and $||\ .\ ||$ is the Euclidean distance between $t$ and $s$. Note that both sensors $t$ and $s$ are assumed to have the same estimation $d_{ts}$ of the distance separating them. The initialisation of the sensor network for the self localisation procedure is recapitulated hereafter, mainly, each sensor node $t$ has to:

a. obtain local information $p_t(\mathbf{x}_t)$, if available.

b. broadcast its ID and listen for other sensors' broadcasts.

c. estimate the distance $d_{ts}$ separating it from any sensor $s$ amongst the set of received broadcasts/IDs.

d. communicate with neighbouring nodes to symmetrise the estimated distances.

Consider a binary random variable $o_{ts}$ indicating whether the distance $d_{ts}$ is observed or not, *i.e.* whether a communication between sensors $t$ and $s$ was established or not, thus, according to the assumption above:

$$P(o_{ts} = 1) = \mathbf{1}_{||\mathbf{x}_t - \mathbf{x}_s|| \leq R}. \tag{3.2}$$

The goal is, for each sensor node $t$ of the network, to compute its marginal distribution of interest $p(\mathbf{x}_t|\mathbf{X}\backslash\mathbf{x}_t, \{d_{ts}\}_{(t,s):o_{ts}=1})$, where $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ is a vector grouping the positions of all the sensors in the network.

Looking back at equation (3.1), for the joint probability distribution $p(\mathbf{X}, \{d_{ts}\}_{(t,s):o_{ts}=1})$, the quantities $d_{ts}$ only depend on the random variables $\mathbf{x}_t$ and $\mathbf{x}_s$, and thus, using the chain rule, this joint PDF can be factorised as follows:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \{d_{ts}\}_{(t,s):o_{ts}=1}) = \prod_{(t,s):o_{ts}=1} p(d_{ts}|\mathbf{x}_t, \mathbf{x}_s) \prod_t p_t(\mathbf{x}_t). \tag{3.3}$$

The undirected graph describing this joint probability distribution is defined by the following two sets: the set $V$ of nodes associated with each sensor and the set $E$ of edges defined as $E = \{(t,s), o_{ts} = 1\}$. Each edge thus models an established neighbourhood between two sensors. This factorisation indeed justifies the intuitive representation of this system by a graph wherein each edge represents an actual physical link established between neighbouring sensors/nodes allowing them to communicate and to mutually detect each other.

### 3.4.2 Implemented solutions

Based on equation (3.3), the set of potential functions associated with the sensor network, *i.e.* the corresponding graphical model, can be divided into two subsets: the set of pairwise potential functions $\psi_{ts}(\mathbf{x}_t, \mathbf{x}_s)$ defined over two neighbouring/communicating nodes and associated with the set $E$ of edges of the graph, and the set $\psi_t(\mathbf{x}_t)$ of local likelihoods at

each sensor, where:

$$\begin{aligned} \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s) &= p(d_{ts}|\mathbf{x}_t, \mathbf{x}_s) \\ &= p_v(d_{ts} - ||\mathbf{x}_t - \mathbf{x}_s||), \quad (t,s) \in E, \\ \psi_t(\mathbf{x}_t) &= p_t(\mathbf{x}_t), \quad t \in V, \end{aligned} \tag{3.4}$$

then, equation (3.3) can be rewritten as:

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_n, \{d_{ts}\}_{(t,s):o_{ts}=1}) = \prod_t \psi_t(\mathbf{x}_t) \prod_{t,s} \psi_{ts}(\mathbf{x}_t, \mathbf{x}_s). \tag{3.5}$$

Both algorithms, the novel box-BP with interval-based representations of probability quantities and the standard particle-based NBP, are tested for the scenario described here above. For the novel box-BP algorithm, message passing between two neighbouring nodes, namely $t$ and $s$, is conducted as follows: for the first iteration, messages are initialised using the local observations, only anchor nodes are thus able to send informative messages to their neighbours, since they are the only nodes with significant a priori information, $p_t(\mathbf{x}_t)$, about their positions. The prior information being represented by a box, $N$ boxes $[\mathbf{x}_t^j]^0, j = 1, \ldots, N$, mutually disjoint and weighted equally with $\frac{1}{N}$ are uniformly generated from the available observations at the anchors.

**Algorithm 4** BP's message update for each node $t$ in the sensor network

$\Gamma_t \leftarrow \{u \in V, o_{tu} = 1\}$
$s \in \Gamma_t$
$MsgCounter \leftarrow 0$
**for** $u \in \Gamma_t$ and $u \neq s$ **do**
  **if** a message is incoming from $u$ to $t$ **then**
    $MsgCounter \leftarrow MsgCounter + 1$
  **end if**
**end for**
Message product
**if** $MsgCounter == 0$ **then**
  **if** $t$ is an anchor node **then**
    use the available measurement at $t$ to generate uniformly $N$ boxes mutually disjoint and weighted equally with $\frac{1}{N}$
  **else**
    no message can be forwarded from $t$ to $s$
  **end if**
**else**
  **for** $i = 1, \ldots, MsgCounter$ **do**
    combine the messages sequentially using the series of intersections in Algorithm 1
    contract the resulting boxes by intersecting them with the available observation at $t$ (if any)
  **end for**
**end if**
propagate the boxes through the model (3.6)
normalise the weights

At an iteration $i$, the message product at node $t$ is represented by a collection $\{\omega_t^{(i)}, [\mathbf{x}_t]^{(i)}\}$

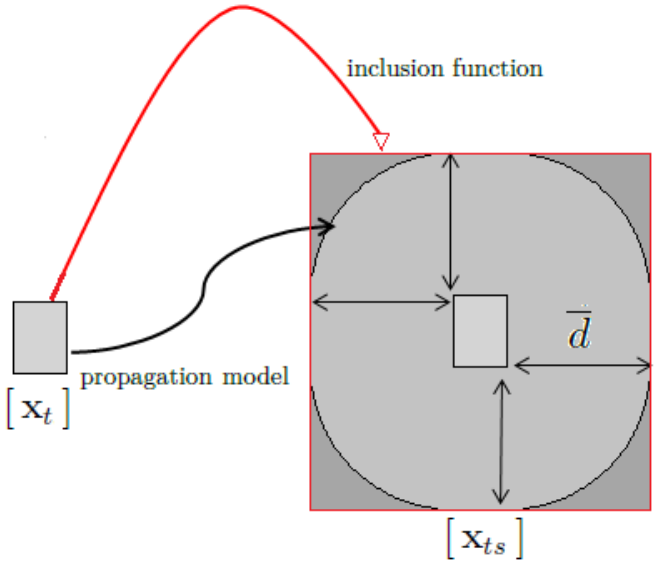



*Fig. 3.1:* The propagation model: the small box $[\mathbf{x}_t]$ represents the position of sensor $t$, the red box an inclusion function for the model (3.6) which is merely a disk equation.

of $N$ weighted box-particles, obtained using Algorithm 1 (see section 2.4.1). These boxes are contracted and re-weighted using the local evidence at $t$ as described in Algorithm 2 in the same section. Note that in this application, we are in the special case of an observation $y_t$ of the same nature as the state $x_t$, the contraction reduces into an intersection between $[y_t]$ and $[x_t]^i$. The forwarded message from $t$ to $s$, separated by noisy distance $d_{ts}$, is then computed by propagating these boxes $\{[\mathbf{x}_t]^{(i)}\}$ through the following model, as shown in the approximation (2.22):

$$\mathbf{x}_{ts}^{(i)} = \mathbf{x}_t^{(i)} + d_{ts}\begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix}, \tag{3.6}$$

where $\theta \in [0, 2\pi]$. Algorithm 4 recapitulates BP's message update operation for each sensor $t \in V$.

Figure 3.1 illustrates the result of propagating a box through the model (3.6). Let $[\mathbf{x}_t] = [\underline{x}_t \;\; \overline{x}_t] \times [\underline{y}_t \;\; \overline{y}_t]$ and $[d_{ts}] = [\underline{d} \;\; \overline{d}]$. Then:

$$\begin{aligned} [d_{ts}] \times cos[\theta] &= [\underline{d} \;\; \overline{d}] \times [-1 \;\; 1] \\ &= [-\overline{d} \;\; \overline{d}], \end{aligned}$$

$$\begin{aligned} [x_t] + [d_{ts}] \times cos[\theta] &= [\underline{x}_t \;\; \overline{x}_t] + [-\overline{d} \;\; \overline{d}] \\ &= [\underline{x}_t - \overline{d} \;\; \overline{x}_t + \overline{d}]. \end{aligned}$$

Similarly,

$$[y_t] + [d_{ts}] \times sin[\theta] = [\underline{y}_t - \overline{d} \;\; \overline{y}_t + \overline{d}].$$

The implementation of NBP for self localisation in sensor networks is described next:

for the first iteration, $N$ particles $\{\omega_t^0, \mathbf{x}_t^0\}$ are sampled from $p_t(\mathbf{x}_t)$; at an iteration $i$ of NBP, the message product at node $t$ is represented by a collection $\{\omega_t^i, \mathbf{x}_t^i\}$ of weighted particles. The weights are corrected using the local evidence at $t$. The forwarded message from $t$ to $s$, separated by noisy distance $d_{ts}$, is then computed by propagating the samples $\{\mathbf{x}_t^i\}$ through the model (3.6) where $\theta \sim U([0, 2\pi])$. In our implementation of the NBP, we used the NBP toolbox available from the work of Leonid Sigal [97].

### *3.4.3 Simulation results*

In this section, we present some simulation results. Both box-BP and NBP algorithms are tested on simulated scenarios and compared to the MDS-MAP(C) described in section 3.3. In our simulations, 100 sensors are randomly deployed in a planar region $L \times L$. We study the performance of the auto localisation algorithms with two different configurations of the anchors: a grid-like and a random placement, we also study the effect of the number of the anchors on the resulting estimation accuracy.

#### *3.4.3.1 Grid-like placement of the anchors*

Basically, two parameters can have impact on the solution provided by a self localisation algorithm: the range within which the sensors can communicate and the number of anchor nodes. Varying the communication range $R$ leads to a variation of the number of neighbourhoods established. Thus, by increasing $R$, more links between nearby nodes are established and a more complex and dense graph is obtained. In [52], only three anchor nodes are used for calibration (since three is the minimal required number of anchors to be able to determine the global coordinates of the sensors in a two dimensional space), in order to obtain accurate results for NBP, the authors chose to increase the number of neighbourhoods/links established between the sensors. This resulted in a very dense graphical model and thus complex computations and more energy and time consumption. Drawing on scenarios presented in [37, 64], we choose to place nine anchors in a grid-like position, eight of them are located on the contour of the region, these can be placed manually, and one anchor is at the centre of the network. In fact, positioning algorithms perform better when anchors surround the nodes with unknown positions [64]; intuitively, nodes at the edges of the graph are less likely to be connected making their localisation more difficult. The range of communication is set to $R = L/4$, the noise standard deviation is set to $0.005L$. Figure 3.2 shows the distribution of the sensors where anchor nodes are marked by circles, while figure 3.3 shows the corresponding graphical model for $R = L/4$.

Table 3.1 summarises the results obtained for both NBP and box-BP algorithms. The simulations were carried out using Matlab on Intel Core i7-3520M processor (2.90GHz-4MB Cache, Dual-core) for $L = 100m$. The error refers to the root mean squared error (RMSE), it shows the mean distance between the true and the estimated position of a sensor and is given as a percentage of $L$. For the box-BP algorithm, the error is calculated using the distance between the centre of the boxes and the true position of the sensors. Note that all three algorithms achieve comparable accuracy (see table 3.1). However, in order to represent the posterior PDF, NBP stores 200 weighted particles, which corresponds, in our 2D application to a total of 600 floating points values. The box-BP algorithm uses only 9 box particles, that is equivalently 45 floating points values. This reduced storage capability is a great enhancement in terms of energy saving and bandwidth needed for the information exchanged in the network. Furthermore, the box-PF is

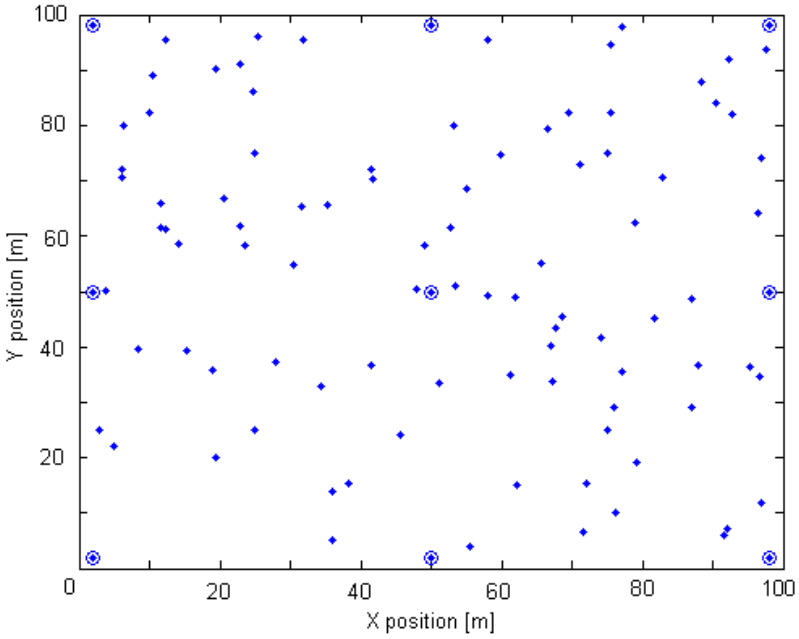



*Fig. 3.2:* A scenario with 100 sensors/nodes and 9 grid-like placed anchors.

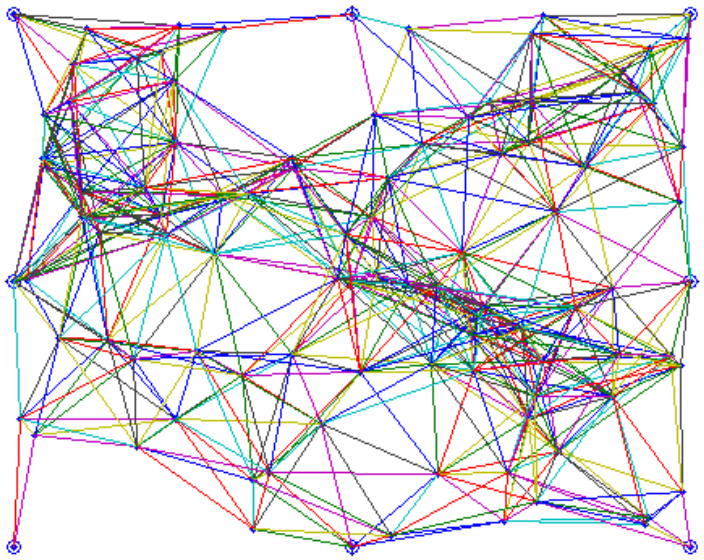


*Fig. 3.3:* The corresponding graph for $R = L/4$. The sensors form the nodes and an edge indicates an established communication between two sensor nodes.

| algorithm | error (%) | No of particles | time (sec) |
|---|---|---|---|
| box-BP | 2.11 | 9 | 14.54 |
| NBP | 2.08 | 200 | 159.4 |
| MDS-MAP(C) | 2.95 | N/A | 0.28 |

*Tab. 3.1:* Simulation results for a grid-like placement of the anchors.

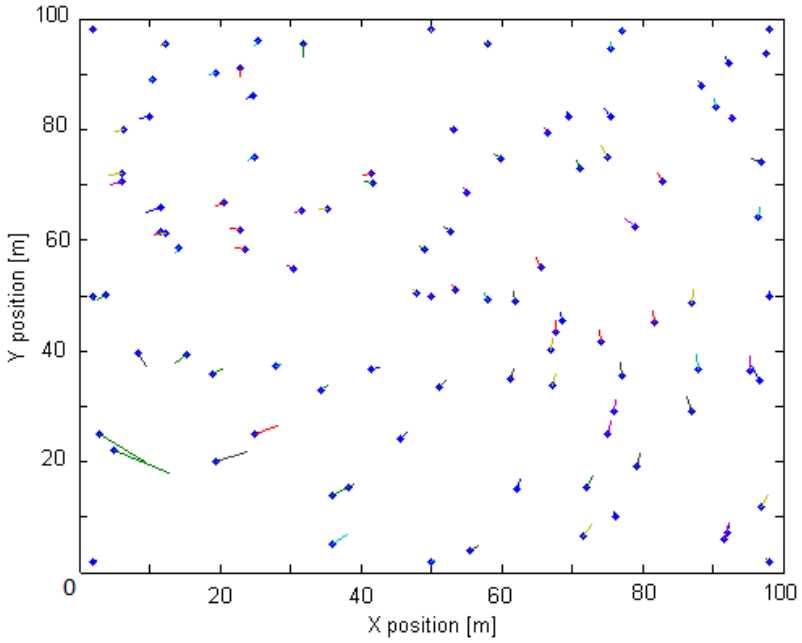



*Fig. 3.4:* Results for the NBP algorithm in the case of a grid-like placement of the anchors.

about 10 times faster than NBP (see table 3.1). Regrading the centralised MDS-MAP(C) algorithm, the computational time is naturally less than that of the iterative BP procedures, however the centralised aspect of the MDS-MAP(C) does not comply with the network energy constraint. It also requires the knowledge of the whole network structure (estimating the distance between each pair of nodes) without offering an improved accuracy. Figures 3.4, 3.5 and 3.6 illustrate the results obtained for the NBP, box-BP and MDS-MAP(C) algorithms respectively. The dots denote the true sensors positions whereas the lines indicate the error on these positions (the distance between the true and the estimated positions).

#### *3.4.3.2 Random placement of the anchors*

We now test both message-passing algorithms on different anchor layouts and compare their performance to the MDS-MAP(C) procedure. Rather than placing the anchors in a grid-like position so as to enclose the randomly spread sensors, the anchors are placed randomly as shown in figure 3.7. We also study the performance of the localisation algorithms while varying the number of anchors within the range 6 to 9. The results are grouped in table 3.2 and table 3.3 for the box-BP and the NBP respectively.

Table 3.2 shows that the novel box-BP algorithm presents comparable results in term of the root mean squared error for different number of anchors. However, this error rate is achieved within less time, i.e. less iterations, if the number of anchors is increased. The

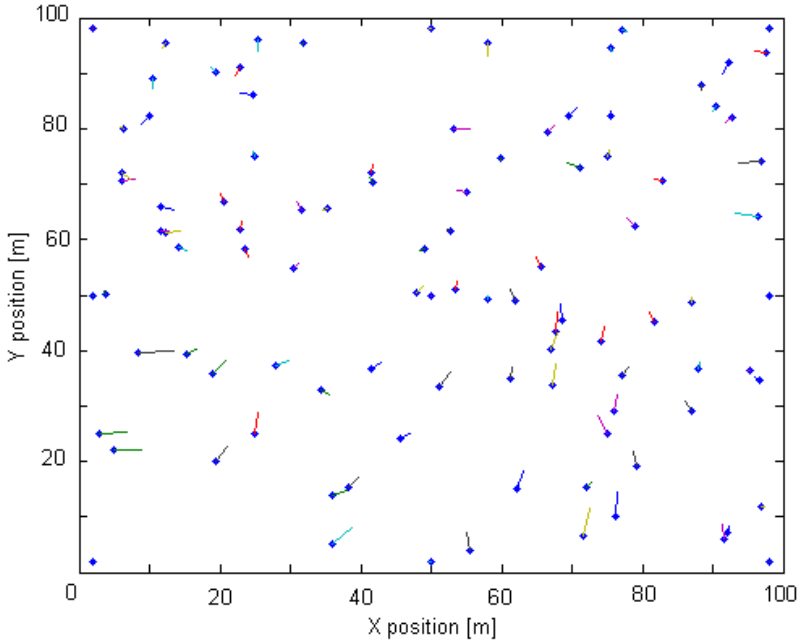



*Fig. 3.5:* Results for the box-BP algorithm in the case of a grid-like placement of the anchors.

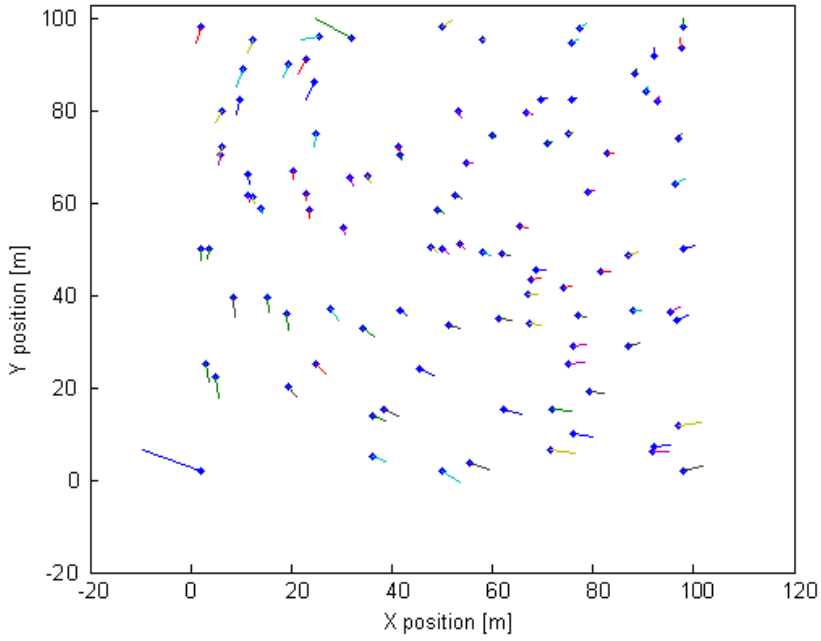



*Fig. 3.6:* Results for the MDS-MAP(C) algorithm in the case of a grid-like placement of the anchors.

| No of anchors | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| No of particles | 9 | 9 | 9 | 9 |
| error (%) | 2.53 | 2.08 | 2.08 | 2.06 |
| time (sec) | 22.80 | 22.87 | 15.72 | 14.27 |

*Tab. 3.2:* Results for the box-BP algorithm in the case of randomly spread anchors.

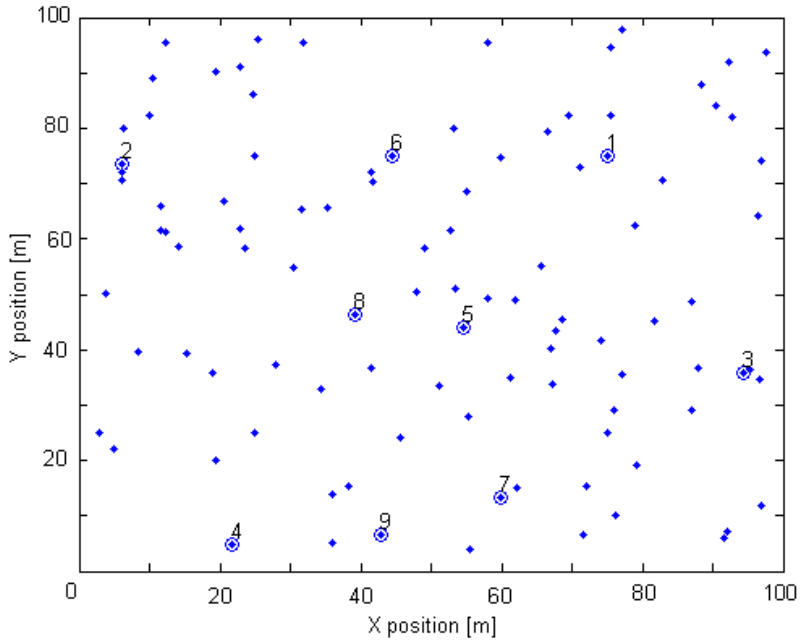



*Fig. 3.7:* A scenario with 100 sensors and 9 randomly spread anchors. The anchors are marked as circles.

| No of anchors | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| No of particles | 200 | 200 | 200 | 200 |
| error (%) | 6.89 | 11.05 | 12.23 | 10.72 |
| time (sec) | 244.33 | 299.96 | 220.41 | 220.87 |

*Tab. 3.3:* Results for the NBP algorithm in the case of randomly spread anchors.

only condition is for the anchors to be uniformly deployed across the region. Figure 3.8 shows the results obtained using the box-BP algorithm in the case of a layout with 6 anchor nodes.

By examining table 3.3, the NBP turns out to be more sensitive to the anchors layout. Figure 3.9 shows the results obtained using the NBP algorithm for a 6-anchor layout. It can be seen that the sensors on the edge of the region could not be well positioned, the error on the position of some of these reached around $L/3$ (which justifies the large value of the root mean squared error noticed in table 3.3), this is observed for all layouts (defined by different numbers of anchor nodes).

Finally, by examining table 3.4 the MDS-MAP(C) centralised scheme offers an accuracy that is comparable to that provided by the distributed box-BP algorithm, again with practically no difference between different anchor layouts. To sum up, the novel box-BP algorithm offers an accuracy that is comparable to that of the centralised MDS-MAP(C) without requiring the whole structure of the network and using local information exchange

| No of anchors | 6 | 7 | 8 | 9 |
|---|---|---|---|---|
| error (%) | 1.73 | 1.70 | 1.71 | 1.71 |
| time (sec) | 0.30 | 0.29 | 0.30 | 0.30 |

*Tab. 3.4:* Results for the MDS-MAP(C) algorithm in the case of randomly spread anchors.

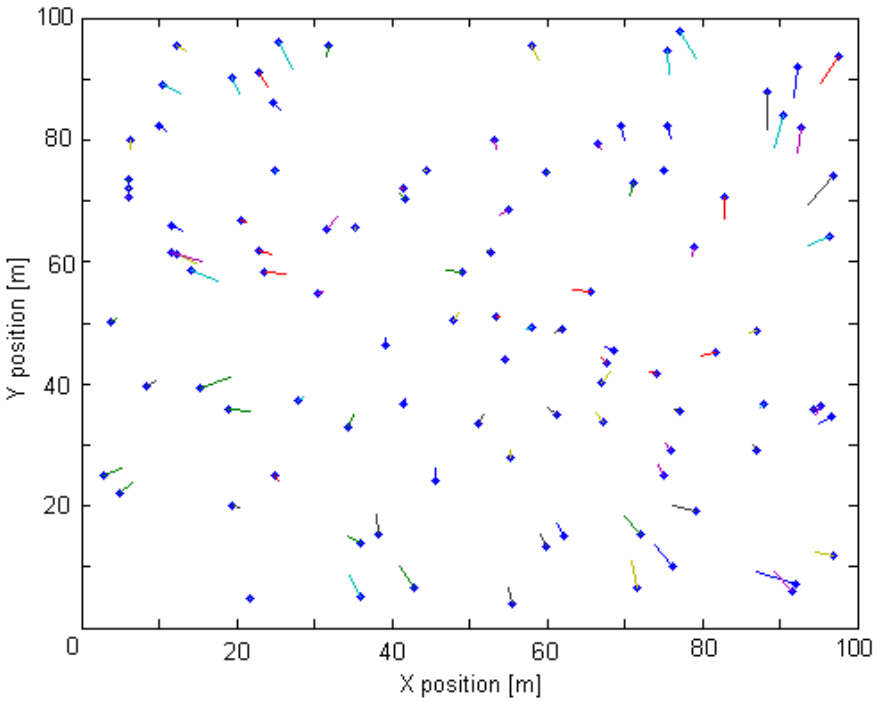



*Fig. 3.8:* Results for the box-BP algorithm in the case of 6 randomly spread anchors.

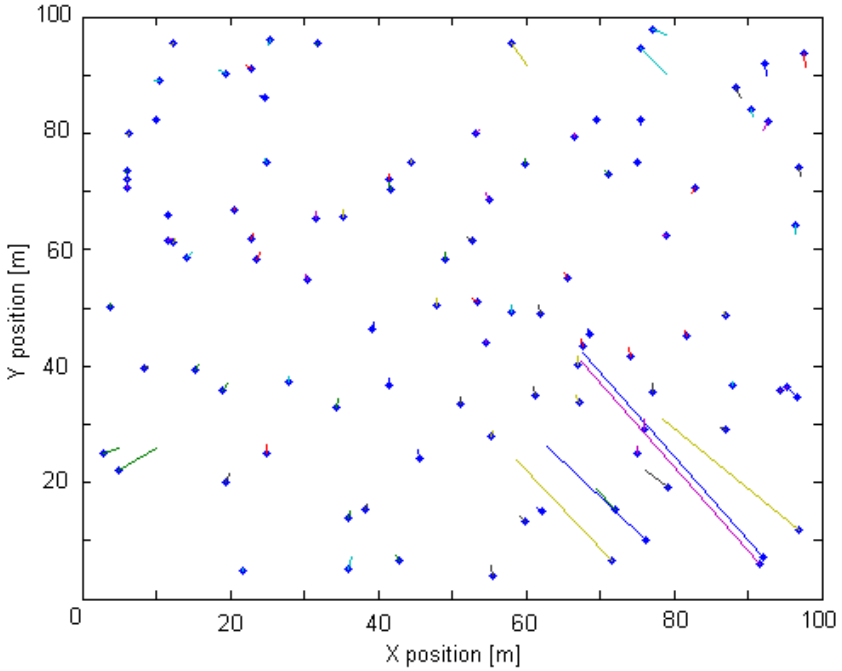



*Fig. 3.9:* Results for the NBP algorithm in the case of 6 randomly spread anchors.

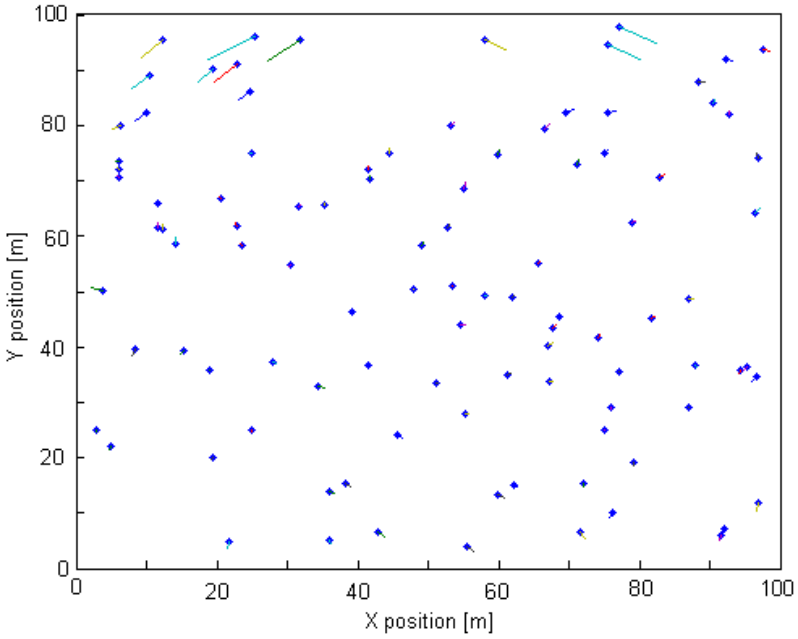



*Fig. 3.10:* Results for the MDS-MAP(C) algorithm in the case of 6 randomly spread anchors.

in a distributed scheme, which is compliant with the energy constraint in a WSN. The box-BP also offers a better performance than that of the standard NBP while requiring less memory storage, bandwidth and computational time as was discussed previously.

## 3.5 Conclusion

In this chapter, we tested the box-BP algorithm, that uses interval representations of probability quantities, on the problem of self localisation in wireless sensor networks. The simulation results showed that for a grid-like placement of the anchors the estimation accuracy provided by NBP algorithm is achieved by the novel box-BP using much less particles and within less computational time. For a random placement of the anchors, the NBP algorithm failed to accurately locate sensors on the edge of the network. The box-BP provided in this case more accurate results. The advantages offered by the new algorithm are reducing the required storage memory, bandwidth and energy needed to exchange information between nodes of a network. Another advantage of the box-BP over the standard NBP is decreasing the complexity of the computations and the mean computational time. Both BP schemes were also compared to the standard centralised MDS-MAP(C) localisation algorithm. The accuracy offered by the novel box-BP is comparable to that of the centralised approach without the necessity to knowing the whole structure of the network and within a distributed context which complies the energy constraint in a WSN.

Nevertheless, in this chapter, the performance of the box-BP was tested on a static wireless sensor network and over simulated data. In the next chapter, we consider a different application involving a dynamic network and test the novel message-passing algorithm on the application of continuously localising a mobile object over simulated and real datasets.

# 4. LOCALISATION OF A MOBILE OBJECT IN A WIRELESS SENSOR NETWORK

## *4.1 Introduction*

Wireless sensor networks (WSNs) built up with cheap and simple sensor nodes have become a popular solution in a range of areas involving monitoring and decision making [66]. From environmental monitoring [69] to surveillance [110], WSNs applications also include target detection, localisation and tracking [66, 92, 71, 77].

Many applications need to locate people or objects, *e.g.*, location aware end-users, intrusion detection and emergency services for finding persons in need, especially in rescue operations based on mobile phones [91]. Localisation algorithms can be classified into different categories as was discussed in the previous chapter. In this chapter, we consider a range-based algorithm for mobility tracking of a single target using information collected from a network of range sensors. At each time step, the target can be detected only by a small number of time-of-flight (or range) sensors, and estimates of the distances separating them from the target are calculated. Thus, at each time instant, the position of the target can be evaluated knowing only the positions of the sensors within the vicinity of it. Since usually even the sensors are not perfectly localised, *i.e.* their positions are known up to some error, the sensors' coordinates are described using random variables, and it is desired to improve the accuracy on these positions while performing the common task of mobility tracking. In order to exploit the local independence conditional relationships, we will formulate the localisation problem as inference on a dynamic graphical model and use a BP scheme to continuously localise the mobile object. Using the BP, we will also be able to reduce the error on the sensors' positions.

In the following, we provide an overview of the prior related work in section 4.2 and then formulate our problem as inference on a graphical model in section 4.3. The box-BP will be employed to perform the mobility tracking task and its performance will be compared to that of the standard NBP. The effectiveness of the novel interval-based message-passing procedure will be demonstrated on simulated and real datasets. Similarly to what was observed in the previous chapter (over the static network), the box-BP will offer an impressive reduction of the memory, bandwidth and computational time required to achieve the localisation goal. Conclusions about this chapter are drawn at last in section 4.4.

## *4.2 Literature review*

Location awareness is fundamental in many applications of WSNs. Localisation and object tracking in sensor networks have become important research problems and have been widely studied. For instance the purpose of the work of Ermis *et al* [34, 35, 36] is to detect localised distributed events or sources in a WSN while minimising inter-network communications cost. The problem of detecting and localising a source emitting a signal with unknown power is considered in [34], whilst [35] develops a multi-target detection al-

gorithm and [66, 71, 96] present multiple target tracking results. In [66], multiple objects moving in a sensor field are tracked by fulfilling several tasks in a collaborative signal processing framework. Decision making requires target detection, classification, localisation and future position prediction. The localisation process employs commonly data provided by a number of sensors such as acoustic, seismic or thermal sensing modalities and is based on the time-varying space-time signature an object generates when moving in a geographical region. In [71] light sensors (along with light sources) are used to track multiple moving objects. A distributed particle filter (DPF) is proposed in [96] to track multiple targets in a WSN. In this approach, the sensors are first clustered into disjoint and uncorrelated cliques. Next, unlike a centralised particle filter (CPF), a DPF updates the weights of the particles, resulting from the prediction step of a particle filter [58], sequentially or in parallel, using only local measurements (*i.e.* observations within a clique of sensors).

In [8, 33] sequential Monte Carlo (SMC) localisation is presented that improves the approach introduced in [50]. The SMC algorithm described in [50] considers a scenario with mobile nodes and anchors, in an environment where no range measurements can be obtained and under the only assumption that all nodes and anchors have a known maximum speed and the same radio range. In [33], range measurements are included into the recursive SMC computations. Other SMC algorithms are developed in [73, 74]. In contrast to [6], the algorithm introduced in [8] remains range-free and works on constricting the area from which the SMC procedure draws samples. In [77], an algorithm is proposed in the bounded error framework for self-localisation of a mobile node in an ad-hoc sensor network, while in [92] the purpose is to solve the problem of tracking multiple robots by formulating it as inference on a graphical model and by using a variant of Belief Propagation (BP) algorithm, called Non-parametric Belief Propagation (NBP) [101]. While providing more accurate location estimations, the NBP requires more computations. This observation was the motivation behind using the box-BP approach we introduced in chapter 2 to localise an object moving within a WSN. The objectives are to achieve the accuracy provided by the NBP within less computational time. A similar approach is given in [119] where a message-passing algorithm is used to solve the problem of tracking multiple targets. This algorithm propagates information from future data about past hypotheses using particle representations of the messages.

The scenario we consider is similar to what was proposed in [91, 105]. The purpose is to locate an object using exclusively its distance measurements to other known points or anchors with no assumptions or limitations on the object movement. We assume that a set of range sensors are deployed in the field of interest and are able to detect and estimate the distance separating them from an object that falls into their sensing range, *i.e.*, their maximal detectable distance. At each time step, the moving object will be in the range of a limited set of sensors in the network. The sensors provide noisy measurements of the distance separating them from the target. Since the dynamics of the moving object is not known, the localisation at each step is based exclusively on exchanging information with the sensors that are able to detect the target. We formulate the problem as inference on a graphical model and employ a variant of Belief Propagation (BP) algorithm, the box-BP which uses box representations for probability quantities, to localise the target over time.

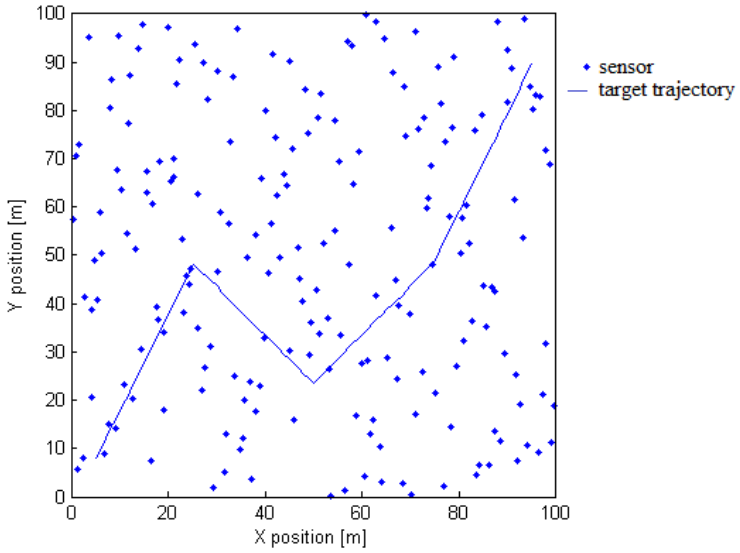



*Fig. 4.1:* Simulated scenario.

## 4.3 Localisation of a mobile target as inference on a graphical model

### 4.3.1 Simulated Scenario

In this section, we consider the problem of localisation of a mobile target, moving within a region of interest, in the presence of randomly distributed sensors and data arriving sequentially in time. A number $n$ of range sensors are deployed in the considered region and are assumed to be able to detect and estimate the distance separating them from an object that falls into their sensing range. This range denotes the maximal detectable distance and will be referred to as $R$. At each time step (whose value will be denoted as $\tau$), the mobile object will be in the range of a limited set of sensors. Thus, estimates of the positions of the moving object will computed each $\tau$ *sec* using only available measurements of the distances separating it from the limited number of sensor nodes that were able to detect it.

The position of the target at time $k\tau$, $k \in \mathbf{N}$, is denoted as $x_{t,k}$ while that of the $j^{th}$ sensor is referred to as $\mathbf{x}_{s_j}$, $\quad j = 1, \ldots, n$. The sensors are static, their positions are fixed in time and are assumed to be known *a priori* (up to some error).

Figure 4.1 shows a simulated scenario with $n = 200$ sensors deployed in a planar $L \times L$ region with $L = 100$ $m$. The trajectory of the mobile target is also shown.

The inference purpose in this application is to determine the posterior distributions $p(\mathbf{x}_{t,k}|\{\mathbf{x}_{s_j}\}_{j=1}^{n}), \quad k = 1, \ldots, T$ where $T$ refers to the number of time steps.

#### 4.3.1.1 Implemented Solutions

Let $c_{j,k}$ denote a boolean variable which indicates whether or not the target t is in the range of sensor $j$ at time $k\tau$. Thus,

$$c_{j,k} = \begin{cases} 1 & if \quad ||\mathbf{x}_{t,k} - \mathbf{x}_{s_j}|| \leq R, \\ 0 & otherwise. \end{cases}$$

In the following, $\Gamma_k$ designates the set of sensors that are able to detect the target at

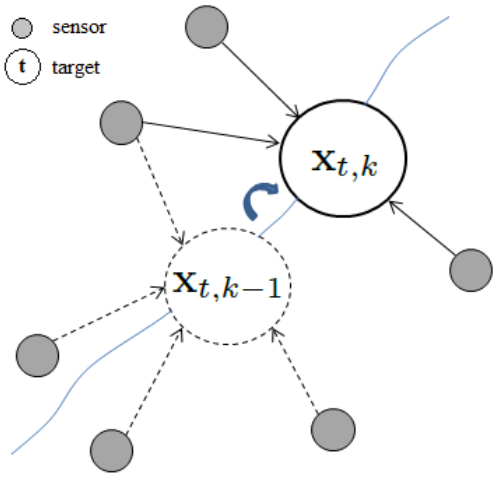



*Fig. 4.2:* The directed acyclic graphs representing the joint probability distribution at two consecutive time steps $k-1$ and $k$.

time $k\tau$, i.e. $\Gamma_k = \{s_\ell,\ c_{\ell,k} = 1\}$. The noisy measurement of the distance between the target at $k\tau$ and a sensor $s_\ell \in \Gamma_k$ is denoted as $d_{t\ell,k}$ and is given by

$$d_{t\ell,k} = ||\mathbf{x}_{t,k} - \mathbf{x}_{s_\ell}|| + v_{t\ell,k}, \quad v_{t\ell,k} \sim p_v(\mathbf{x}_{t,k}, \mathbf{x}_{s_\ell}), \tag{4.1}$$

where $v_{t\ell,k}$ denotes a statistical measurement noise of a probability distribution $p_v(\mathbf{x}_{t,k}, \mathbf{x}_{s_\ell})$, and $||\ .\ ||$ denotes the Euclidean distance between the target t at $k\tau$ and $s_\ell \in \Gamma_k$.

Given the position $\mathbf{x}_{s_\ell}$ of sensor $s_\ell$, the target will be at position

$$\mathbf{x}_{t,k} \ = \ \mathbf{x}_{s_\ell} \ + \ ||\mathbf{x}_{t,k} - \mathbf{x}_{s_\ell}|| \times \begin{pmatrix} \cos(\theta_{t\ell,k}) \\ \sin(\theta_{t\ell,k}) \end{pmatrix},$$

where $\theta_{t\ell,k}$ is the angle between $\mathbf{x}_{s_\ell}$ and $\mathbf{x}_{t,k}$. Thus, the joint probability distribution $p(\mathbf{x}_{t,k}, \mathbf{x}_{s_1} \ldots, \mathbf{x}_{s_n})$ can be factorised as follows:

$$\begin{aligned} p(\mathbf{x}_{t,k}, \{\mathbf{x}_{s_j}\}_{j=1}^n) &= p(\mathbf{x}_{t,k} | \{\mathbf{x}_{s_j}\}_{j=1}^n) \prod_j p(\mathbf{x}_{s_j}) \\ &= p(\mathbf{x}_{t,k} | \{\mathbf{x}_{s_\ell}\}_{s_\ell \in \Gamma_k}) \prod_j p(\mathbf{x}_{s_j}), \end{aligned} \tag{4.2}$$

where $p(\mathbf{x}_{s_j})$ refers to the prior probability distribution of sensor $j$'s position.

According to (4.2), the directed graph describing this joint probability distribution is a graph whose nodes represent the target at time $k\tau$ and the set $\Gamma_k$ of its neighbouring sensors, and whose edges are directed from nodes $\{s_\ell \in \Gamma_k\}$ to the node associated with the target.

Figure 4.2 shows the directed graphs representing the joint PDF in (4.2) at two consecutive time steps, $k-1$ and $k$. Note that if the dynamics of the target is known, thus a function linking $\mathbf{x}_{t,k-1}$ to $\mathbf{x}_{t,k}$ is available, information would also propagate between time steps which translates graphically into an edge directed from node $t_{k-1}$ towards $t_k$. Another point is that although the arcs of the graph are directed, information circulates in either direction and the operations of BP as described in the previous sections can be used.

In our simulations, we fixed the time period to $\tau = 1$ *sec*; hence, each second, new

links are established between the target and the sensors that, at the considered time instant, have the moving object within their sensing range, while links that were made at the previous time step would be broken when the moving object steps out of their range: the resulting graphical model is dynamic as can be seen in Figure 4.2. In order to solve the inference problem for the simulated scenario described here (which consists in estimating the posteriors $p(\mathbf{x}_{t,k}|\{\mathbf{x}_{s_j}\}_{j=1}^n)$), two algorithms, the novel NBP with box representations – referred to as box-BP– and NBP with sample representation – referred to simply as NBP – are compared. During a time period $\tau$, three iterations of message-passing are run, messages are exchanged along the edges of the graph and the target node computes an estimation of its current position. Recall that the BP is a tool to perform marginalisation of joint PDFs represented by graphical structures, and that it allows to estimate the sensors' positions wile performing the mobility tracking task. However, in our simulations, the purpose is to continuously localise the mobile object and thus the marginal PDFs of interest are given by $p(\mathbf{x}_{t,k}|\{\mathbf{x}_{s_j}\}_{j=1}^n)$. An example of message-passing computations is given in the following.

**Algorithm 5** BP's message update for mobility tracking at time instant $k\tau$

$\Gamma_k = \{s_\ell,\ c_{\ell,k} = 1\}$
*Message passing from the sensors to the mobile target*
**for** $s_\ell \in \Gamma_k$ **do**
use the available observation at sensor $s_\ell$ to generate uniformly $N$ boxes mutually disjoint and weighted equally with $\frac{1}{N}$
propagate the boxes through the model $\mathbf{x}_{s_\ell,t_k}^{(i)} = \mathbf{x}_{s_\ell}^{(i)} + d_{tl,k}[\cos(\theta) \quad \sin(\theta)]^T$
**end for**
combine the messages sequentially using the series of intersections in Algorithm 1
normalise the weights
the resulting boxes referred to as $\{[\mathbf{x}_{t,k}]^{(i)}, \omega_{t,k}\}_{i=1}^N$ represent the belief at the target at time $k\tau$
*Message passing from the target to the sensors*
**for** $s_\ell \in \Gamma_k$ **do**
propagate the resulting boxes through the model $\mathbf{x}_{t_k,s_\ell}^{(i)} = \mathbf{x}_{t,k}^{(i)} + d_{tl,k}[\cos(\theta) \quad \sin(\theta)]^T$
combine the messages sequentially using the series of intersections in Algorithm 1
contract the resulting boxes using the observation at sensor $s_\ell$
normalise the weights
the resulting boxes $\{[\mathbf{x}_{s_\ell}]^{(i)}, \omega_{s_\ell}\}_{i=1}^N$ now represent the position of sensor $s_\ell$
**end for**

In this example, message-passing between two neighbouring nodes (which will be denoted as 1 and 2 for the simplicity of the notation) is explained. For the novel box-BP algorithm, the prior information at each sensor is represented by a box, $N$ boxes mutually disjoint and weighted equally with $\frac{1}{N}$ are uniformly generated from the available information about sensors positions. For NBP, $N$ particles are sampled from the prior $p(\mathbf{x}_{s_\ell})$. At an iteration $i$, in the bounded error context, the message product at node 1 is represented by a collection $\{\omega_1^{(i)}, [\mathbf{x}_1]^{(i)}\}$ of $N$ weighted box-particles obtained using Algorithm 1. The forwarded message from 1 to 2, separated by noisy distance $d_{12}$, is denoted as $\mathbf{m}_{12}^{(i)}$ and is represented by the cloud of boxes $\{\omega_{12}^{(i)}, \mathbf{x}_{12}^{(i)}\}$ obtained, as shown in (2.22), by propagating the collection $\{[\mathbf{x}_1]^{(i)}\}$ through the following model

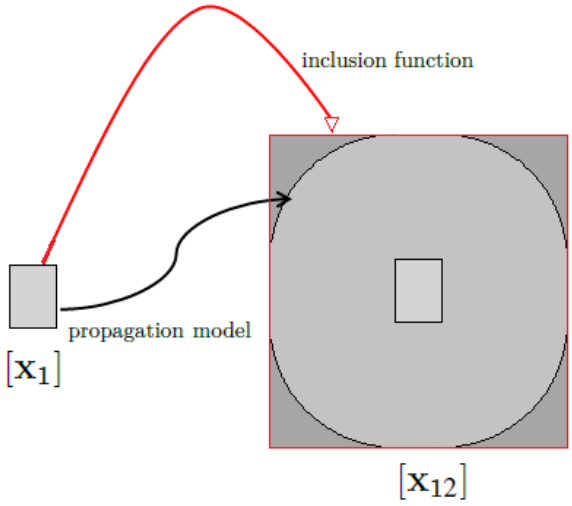



*Fig. 4.3:* The result of propagating a box through the disk equation given by (4.3). An inclusion function for this propagation model is also given.

(which is merely a disk equation):

$$\mathbf{x}_{12}^{(i)} = \mathbf{x}_1^{(i)} + d_{12}[\cos(\theta) \quad \sin(\theta)]^T, \tag{4.3}$$

where $\theta \in [0, 2\pi]$.

Figure 4.3 illustrates the result of propagating a box through the model (4.3). In this example, the position of the sending node is $[\mathbf{x}_1]$, the receiving node 2 is at a distance $d_{12} = [\underline{d} \quad \overline{d}]$ from the sending node 1.

Algorithm 5 describes the message passing procedure to perform the mobility tracking task.

For NBP algorithm, the message product at the sending node 1 being represented by a collection $\{\omega_1^i, \mathbf{x}_1^i\}$ of weighted particles, the forwarded message from 1 to 2, separated by noisy distance $d_{12}$, is computed by propagating the samples $\{\mathbf{x}_1^i\}$ through the model (4.3) where $\theta \sim U([0, 2\pi])$.

Furthermore, both probabilistic algorithms are also compared to the optimisation approach introduced by Sayed *et al.* in [91] and to which we will be referring as *Sayed05*. A brief review of the Sayed05 algorithm is given in the next subsection.

#### 4.3.1.2 The Sayed05 Localisation Algorithm

Sayed05 is an algorithm for 2D landmark-based localisation that was proposed in [91] in response to a mandate issued by the Federal Communications Commission (FCC) that aimed to solve a public safety problem: the need to determine the geographic coordinates of a mobile subscriber in a cellular network knowing that an important proportion of emergency calls originates from cell phones. This algorithm is based on the knowledge of distances separating the target, *i.e.* the non-localised object, from three known points or stations also called landmarks. This problem is commonly known as trilateration and is illustrated in Figure 4.4. The Sayed05 algorithm works as follows. Let $\mathbf{r}_t = [x_t, \ y_t]^T$ denote the position of the target and $\mathbf{r}_i = [x_i, \ y_i]^T, \ i = 1, 2, 3$ the coordinates of three landmarks. The squared distance between $\mathbf{r}_t$ and $\mathbf{r}_i$ is given by

$$d_i^2 = (x_i - x_t)^2 + (y_i - y_t)^2, \ i = 1, 2, 3. \tag{4.4}$$

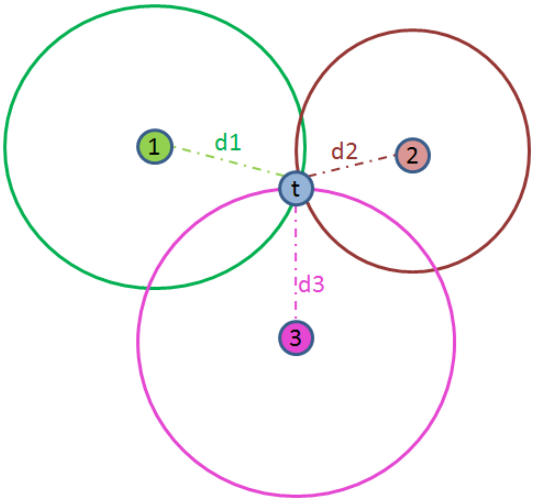



*Fig. 4.4:* Trilateration.

Subtracting the first equation ($i$ =1) from the $i$-th equation ($i$=2, 3) in (4.4), we obtain

$$d_i^2 - d_1^2 = \tilde{x}_i^2 - 2\tilde{x}_i\tilde{x}_t + \tilde{y}_i^2 - 2\tilde{y}_i\tilde{y}_t, \quad i = 2, 3, \tag{4.5}$$

where $\tilde{\mathbf{r}} = [\tilde{x},\ \tilde{y}]^T = \mathbf{r} - \mathbf{r}_1$. The constraints in (4.5) can be expressed in matrix notation as

$$\begin{pmatrix} \tilde{x}_2 & \tilde{y}_2 \\ \tilde{x}_3 & \tilde{y}_3 \end{pmatrix} \begin{pmatrix} \tilde{x}_t \\ \tilde{y}_t \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \tilde{x}_2^2 + \tilde{y}_2^2 + d_1^2 - d_2^2 \\ \tilde{x}_3^2 + \tilde{y}_3^2 + d_1^2 - d_3^2 \end{pmatrix}, \tag{4.6}$$

or, in a compact form,

$$\mathbf{H}\tilde{\mathbf{r}}_t = \mathbf{b}. \tag{4.7}$$

The algorithm then uses a least squares approach to determine $\tilde{\mathbf{r}}_t = \mathbf{b}\mathbf{H}^{-1}$.

#### *4.3.1.3 Simulation Results*

The simulations are run under Matlab on Intel Core i7-3520M processor (2.90 GHzâĂŞ4 MB Cache, Dual-core). The sensing range is set to $R = 15\ m$. We run two series of simulations: the first series considers a white Gaussian noise, of standard deviation $\sigma$, on the measured distances; the second considers a bounded measurement noise modelled as a uniform PDF defined over $[-a\ \ a]$, ($\sigma, a \in \mathbf{R}^{+*}$). The performance of both box-BP and NBP are compared to that of the Sayed05 localisation algorithm that was briefly recapitulated in the previous subsection. We also study the effect of the measurement noise by varying $\sigma$ and $a$ defined earlier. Tables 4.1 and 4.2 summarize the results obtained for all three algorithms in the case of a Gaussian measurement noise and a bounded noise respectively. The error refers to the root mean squared error, that is the mean distance between the true and the estimated positions. The time refers to the mean computational time per time step.

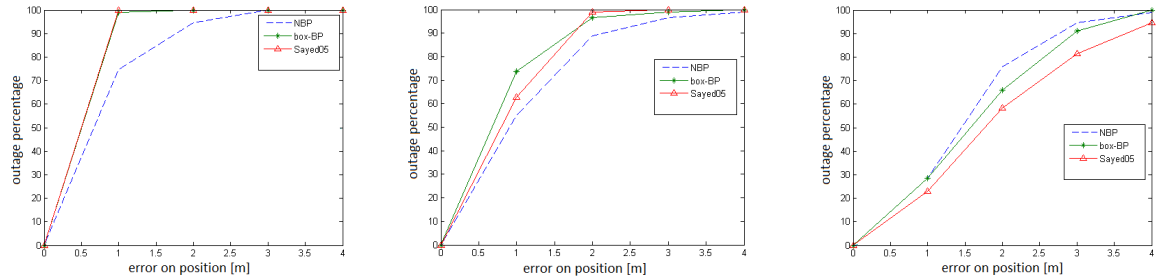
These tables show that the novel box-BP algorithm provides a better accuracy (or a similar accuracy in the case of a Gaussian measurement noise for large $\sigma$) than the NBP, the optimisation Sayed05 algorithm while offering similar performance to the probabilistic NBP and box-BP algorithms for low measurement noise, provides less accurate estimates as the noise magnitude increases. An interpretation of these results is based on the fact

| | | $\sigma = 0.5\ m$ | | $\sigma = 1\ m$ | | $\sigma = 1.5\ m$ | |
|---|---|---|---|---|---|---|---|
| Algorithm | $N^o$ of particles | error [m] | time [ms] | error [m] | time [ms] | error [m] | time [ms] |
| box-BP | 9 | 0.3284 | 2 | 0.8287 | 2.8 | 1.6878 | 3.3 |
| NBP | 200 | 0.8317 | 59.1 | 1.0704 | 57.1 | 1.5069 | 58.4 |
| Sayed05 | N/A | 0.1923 | 0.18 | 0.8567 | 0.17 | 1.9835 | 0.16 |

*Tab. 4.1:* Simulation results in the case of a Gaussian measurement noise.

| | | $a = 0.5\ m$ | | $a = 2\ m$ | | $a = 4\ m$ | |
|---|---|---|---|---|---|---|---|
| Algorithm | $N^o$ of particles | error [m] | time [ms] | error [m] | time [ms] | error [m] | time [ms] |
| box-BP | 9 | 0.3808 | 1.8 | 0.6964 | 2.3 | 1.3885 | 3 |
| NBP | 200 | 0.9325 | 59.2 | 2.2367 | 57.7 | 1.5032 | 57.3 |
| Sayed05 | N/A | 0.2339 | 0.18 | 0.9488 | 0.16 | 2.0321 | 0.24 |

*Tab. 4.2:* Simulation results in the case of a bounded measurement noise.

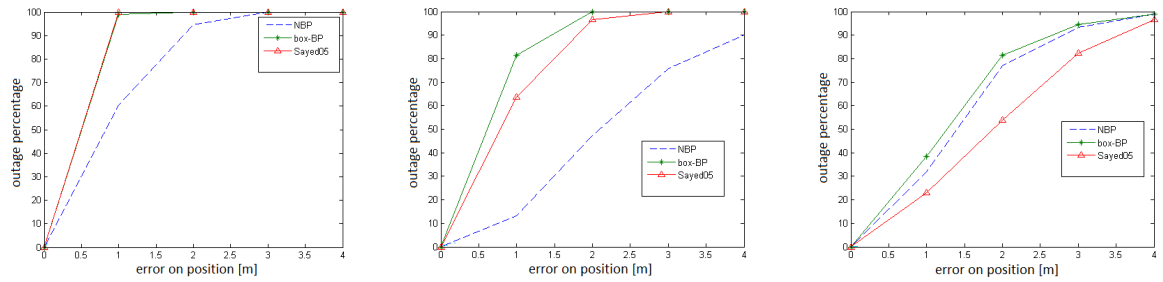



*(a)* $\sigma = 0.5\ m$. *(b)* $\sigma = 1\ m$. *(c)* $\sigma = 1.5\ m$.

*Fig. 4.5:* Cumulative distribution function of the error: case of a Gaussian measurement noise.

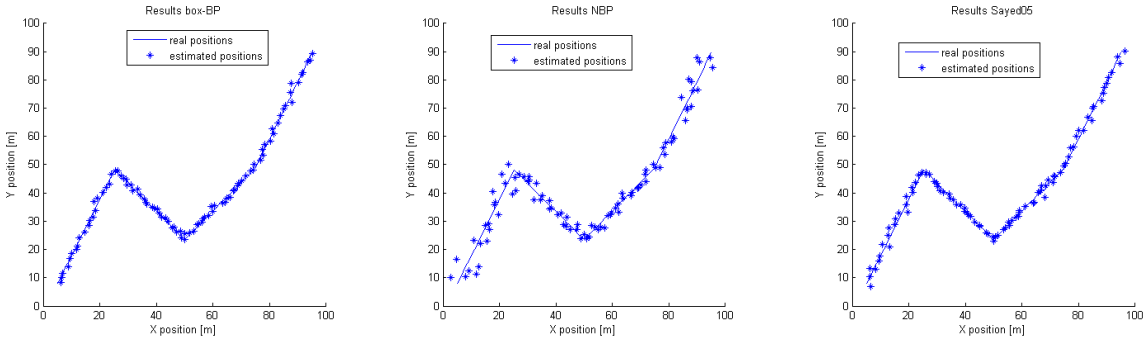
that the box-BP provides a direct approximation of the posterior distribution using a mixture of uniform PDFs, and not a representation of it using particles as in the case of NBP. To calculate the messages and beliefs, there is no need to introduce kernels, which is not necessarily optimal for the NBP. Also, compared with the NBP, the box-BP and the Sayed05 algorithms are simpler and introduce less approximations. Furthermore, the novel box-BP uses only 9 box particles to calculate the probability quantities, that is equivalently 45 floating points values, versus 200 weighted particles for NBP, which corresponds, in our 2D application to a total of 600 floating points values. This reduced storage requirement is a great enhancement in terms of energy saving and bandwidth needed for the information exchanged in the network. The box-PF is *about* 30 *times faster* than the NBP. While the mean computational time for the Sayed05 optimisation algorithm is about 10 times less than that of the box-BP, the performance of this optimisation technique degrades in noisy environments since it does not take into account a measurement error. These results can be seen by examining Figures 4.5 and 4.6 illustrating the empirical cumulative distribution function of the error for all three algorithms and for both Gaussian and bounded measurement error environments respectively.

Figure 4.7 illustrates the real trajectory and the estimated trajectories obtained for box-BP, NBP and Sayed05 algorithms, respectively. The presented results correspond to a bounded error environment with $a = 2\ m$.




(a) $a = 0.5\ m$. (b) $a = 2\ m$. (c) $a = 4\ m$.

*Fig. 4.6:* Cumulative distribution function of the error: case of a bounded measurement noise.




(a) box-BP. (b) NBP. (c) Sayed05.

*Fig. 4.7:* Real and estimated trajectories for box-BP, NBP and Sayed05 (case of a bounded measurement noise with $a = 2m$).

### 4.3.2 Results Obtained on Real Data

In this section, the box-BP algorithm is tested on real data available from the UTIAS multi-robot cooperative localisation and mapping dataset [65]. This data set is produced using 5 robots moving in an indoor workspace with an area of $15 \times 8\ m^2$ and where 15 landmarks were placed as illustrated in Figure 4.11. The landmarks consisted of cylindrical tubes of 30 $cm$ in height. Figure 4.8 shows the cylindrical landmarks and the robots.

The robots are identical in construction, equipped with a monocular camera serving as a primary sensing module and interfacing with a netbook. The camera is mounted in a way that the robot's body frame and the measurement coordinate frame are coincident in two dimensions, as shown in Figure 4.9.

During the experiments the robots were driving to randomly generated waypoints in the workspace. When a robot or landmark falls in a robot's field of view, a range and bearing measurements are taken. Throughout the data collection process, the groundtruth data for the robots and the landmarks were made available through the use of a 10-camera Vicon motion capture. Figure 4.10 illustrates a scene of the experiments conducted while producing the UTIAS dataset.

The original dataset is transformed as in [25] in order to adapt to the scenario we consider in this paper: the localisation of a single moving target using exclusively range measurements from landmarks. We thus extract from the original dataset, data associated with poses, of any of the five robots, for which range measurements to landmarks are available. A table of real groundtruth data for a single target is obtained, along with a table indicating for each pose/step identifiers of the landmarks within the vicinity of the

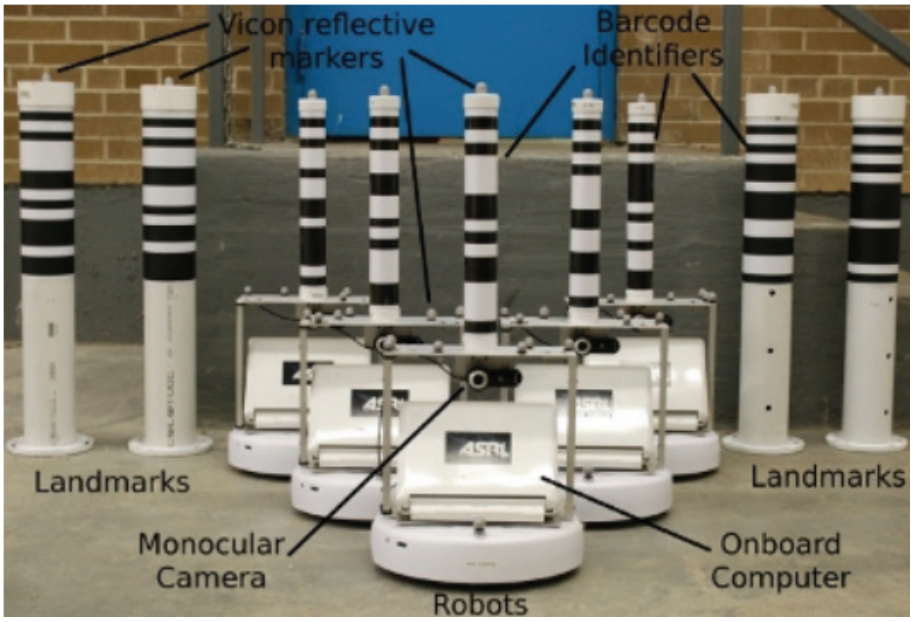



*Fig. 4.8:* The cylindrical landmarks and robots used for the generation of the UTIAS dataset.

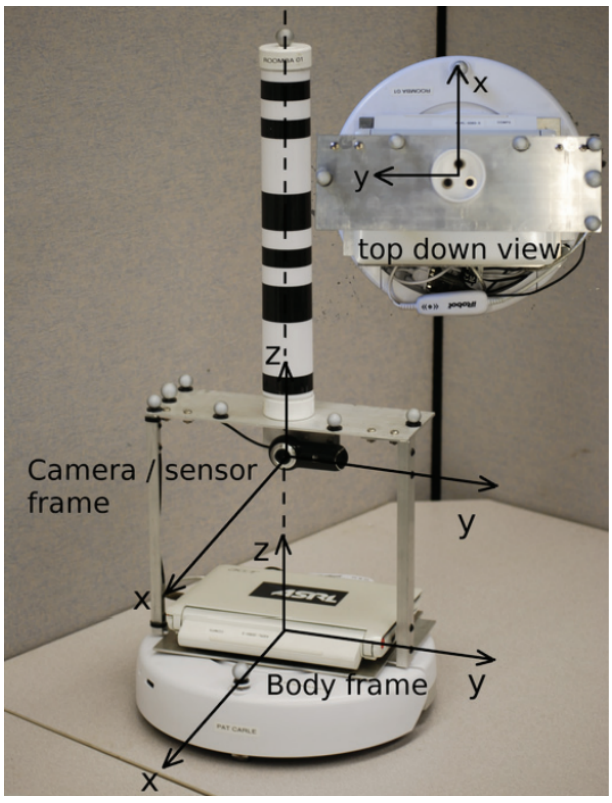



*Fig. 4.9:* The workspace for the real data set.

*Fig. 4.10:* The workspace for the real dataset.

| Algorithm | $N^o$ of particles | error [m] | time [ms] |
|---|---|---|---|
| box-BP | 9 | 0.6599 | 2.1 |
| NBP | 200 | 1.1139 | 54.1 |
| Sayed05 | N/A | 3.9447 | 0.28 |

*Tab. 4.3:* Results obtained on a real data set.

non localised object and a real noisy measurement of the distance separating them. The performance of the box-BP is again compared to that of the NBP and Sayed05 algorithms. The results are grouped in Table 4.3.

The cumulative distribution function of the error is given in Figure 4.12.

The novel box-BP algorithm outperforms both NBP and Sayed05 algorithms. The performance of the Sayed05 algorithm turns out to degrade in noisy environments; though this optimisation algorithm requires less computations than the remaining two probabilistic algorithms. The advantages of storage, memory and bandwidth reduction offered by the box-BP over the NBP remain outstanding as with the simulated scenarios.

## 4.4 Conclusion

In this chapter, we apply the novel box-BP to perform the localisation of a moving target, at each time step, using only range measurements from landmarks. The effectiveness of the new box-BP algorithm is shown on simulated and on real data. Its performance is also compared to that of a probabilistic and an optimisation algorithms, the NBP and the Sayed05 algorithms, respectively. The simulation results shows that the estimation accuracy provided by the novel box-BP algorithm is better (or similar) to that of the NBP, and is achieved using much less particles and within less computational time. The performance of the novel algorithm also remains outstanding even in noisy environments unlike the fast optimisation Sayed05 algorithm. The accuracy of the box-BP turns out to be better on the real data set. In summary, the advantages offered by the new algorithm are a reduction of the required storage memory, bandwidth and energy needed to exchange

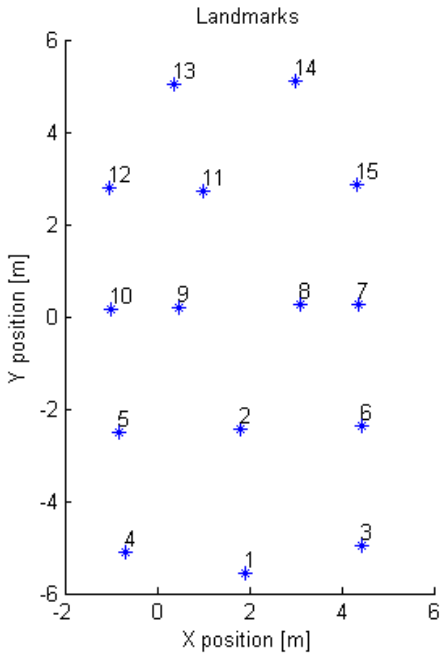



*Fig. 4.11:* Positions of the landmarks in the real workspace.

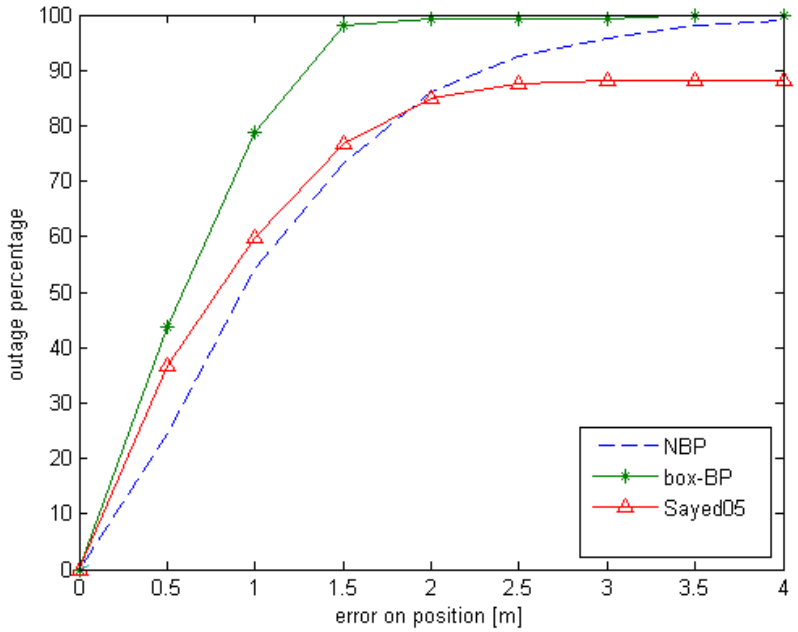



*Fig. 4.12:* Cumulative distribution function of the error for the real data set.

information between the nodes as compared to the NBP and a better accuracy in noisy scenarios when compared to the Sayed05 algorithm. The box-BP in fact introduces less approximations than the NBP (which uses kernels to approximate the PDFs) and thus offer a better accuracy, it is also applicable in the presence of quantised measurements, this would require less bandwidth for information exchange since float type measurements are often replaced by integers.

# Part II

# SOURCE CHARACTERISATION PROBLEM

# 5. LITERATURE REVIEW ON SOURCE CHARACTERISATION AND ADVECTION-DIFFUSION MODELS

## *5.1 Introduction*

Recently, wireless sensor networks (WSNs) are being successfully used in source identification applications, which are approaches for automatic detection and characterisation of sources of dispersion (*i.e.* sources releasing a chemical or biological agents into their environment) based on data collected from a WSN. In this thesis, we are interested in the application of land mine detection and localisation given the humanitarian aspect of this topic, and we refer, throughout the following chapters, to the task of determining the land mines/ sources unknown parameters, including their positions and emission rates, as the problem of *source characterisation*.

An anti-personnel mine is a device designed to injure or to kill whomever comes into contact with it through direct pressure or a trip-wire [46]. Land mine detection, localisation and clearance is of great importance due to the danger that buried land mines still represent to people all over the world. It is pointed out in [87] that about 50–100 million anti-personnel mines exist in more than 80 countries and that more than 20,000 people are killed or injured every year due to the explosion of buried land mines. Though the Ottawa treaty prohibited in 1997 the use of this weapon, some countries have not signed the treaty, and nearly two million land mines are laid per year. The dangers are particularly acute for those responsible for localising and decommissioning land mines. To limit the number of victims, land mine detection and clearance actions have taken place since the end of the Second World War [16, 54].

Historically, the most common method for land mine detection is metal detection. Although this technique has proved to be effective with primary land mines, it fails with many modern land mines, which are fabricated from sophisticated non-metallic materials, such as plastic and wood [46], making them invisible to the metal detector. Therefore, many other methods have been developed. These include the use of trained dogs and several physical detection techniques based on ground penetrating radar (GPR), X-ray, infrared (IR) imaging [67, 81], neutron activation (TNA) and nuclear quadrupole resonance (NQR) [15, 87]. However, a common problem with all these techniques is that the probability of false positives is high [38]. Other approaches employ unmanned vehicles for landmine detection [43]. This technique requires sophisticated and rather expensive equipment and control.

In this second part of the thesis, we address the problem of localising an unknown number of land mines using data collected from a network of wireless sensors capable of detecting the concentration of the explosive chemicals in the air. The motivation behind the proposed framework is that buried land mines can be considered as vapour-emitting sources, based on the observation that some explosive chemicals, such as trinitrotoluene (TNT) or dinitrotoluene (DNT), leak out from buried land mines into the surrounding environment and are transported through the air by mechanisms such as advection and diffusion; the idea of using a WSN for land mine localisation then seems intuitively feasible.

the idea is to deploy a network of wireless sensors, capable of sensing the concentration of the explosive chemical at their positions, in the contaminated region. Sensors for these types of chemical explosive materials exist [118], thus, having a vector of concentration measurements provided by a WSN, by expressing the concentration of the explosive as a function of the land mines' locations and by solving the inverse problem, we will show in the following chapters that the proposed framework is able to detect, locate and find the emission rates of several land mines.

One primary advantage of the proposed technique consists in the fact that existing land mine detection and localisation techniques (*e.g.* those reported here above) involve either human (or animals) intervention or the use of bulky and expensive equipments. This translates into threatening risks on the operators lives and/or high operational costs. Using a network of low-cost small-sized wireless sensors allows an automatic and less expensive detection of buried land mines. Furthermore, it was suggested in [32] that using this technique reduces false positives since the sensors respond to the key element of a landmine, the explosive chemical.

In this chapter, we first provide a literature review on the use of sensor networks for source characterisation in section 5.2. Then, an overview of advection-diffusion models in the cases of instantaneous and continuous release point sources is available in section 5.3. The forward model, aiming at predicting the concentration measurements at the sensors' positions given the sources parameters, is formulated in section 5.4. Section 5.5 then concludes the chapter.

## 5.2 Related works

The advances in sensing technologies [38] increase the use of sensor networks in a vast range of applications [7, 93, 68]. Recently, wireless sensor networks (WSNs) have become popular in source identification applications. In fact, risk management applications in the fields of environment [13, 61, 69] and security [93, 110] rely on data collected from a WSN in order to characterise a source of dispersion, e.g., in the case of an accidental or intentional release of a chemical or biological substance in the air. In [110], an algorithm is derived to detect CO2 leaks at several potential locations at a carbon sequestration site. The aim in [69] is to study the emissions of a number of contaminant sources, located at well-known positions, at a large lead-zinc smelter. In [13, 61, 93], probabilistic Bayesian approaches are used to determine the unknown position and possibly other model parameters, such as the emission rate and the diffusion coefficient, of a single dispersion source using data collected from a WSN. In [83], a recursive algorithm based on a state space representation of the system is developed to estimate a single diffusion source position and to track its intensity in time using concentration measurements provided by a sensor network. In [85] theoretical results are derived characterising the accuracy of the location estimate of a single gas emitting source using a network of binary sensors. The measurements are quantised and a single bit of information is generated depending on whether the sensed value is lower or higher than some threshold. In [86] a computation method is proposed to overcome the difficulties associated with the choice of an adequate dispersion model and the calculation of the likelihood function in a Bayesian framework in order to solve the problem of localisation of a source of toxic release.

For landmine detection, the idea is to spread wireless sensors over the contaminated field, *e.g.* using a flying robot, these sensors measure the concentration of the explosive material at their positions and communicate with a processing unit. The measured concentrations depend on the land mines and sensors positions and on the perturbation

caused by multiple environmental effects, *e.g.* the wind. The complexity of the problem is seen through the following reasons:

a. the sensors positions are unknown since they are randomly dropped. This problem was however dealt with in chapter 3,

b. the mines are buried and diffusion constants are different in the ground and in the air,

c. diffusion constants are not known accurately,

d. environmental disturbance is hardly modelled,

e. an error exists between the real and the measured concentrations.

The idea of using a sensor network for land mine localisation is addressed in [32, 57], but both consider the case of a single land mine. In [32], the problem of localisation of a single land mine is considered using an analytical solution of the inverse problem, not taking into account a model or measurement noise. In [57], a maximum likelihood estimation algorithm is derived in order to locate a single land mine and find its emission rate. The performance of the estimator is evaluated by computing the Cramer–Rao bound.

The case of parameter estimation for multiple sources is briefly addressed in [113] and [95] in a Bayesian probabilistic and an optimisation least squares frameworks and for a known number of sources. However, in the context of land mine detection, this information rarely exists for real applications. A more difficult case when the number of sources is unknown is addressed in [114, 115, 116]. In [114, 116], the problem is formulated as a generalised parameter estimation problem, where the number of sources is included in the vector of unknown parameters. This approach implements a reversible-jump MCMC algorithm and requires intensive computations since the dimensionality of the unknown parameters' vector is variable. In [115], Yee formulates the problem of characterising an unknown number of sources as a model selection problem. While this approach is less complex than the previous one, it is also computationally demanding.

The application of interest in this chapter is the localisation of multiple anti-personnel land mines using a WSN, where we consider the case of an unknown number of sources. In our method, the objective is to estimate the number of sources and then characterise them. Briefly speaking, the set of concentration measurements which have been made by the detection system are grouped in a matrix and a PCA scheme is used in order to determine the number of sources. The use of this strategy makes the problem of localisation less complex and more efficient. The localisation problem is addressed after in a probabilistic Bayesian framework and a Markov chain Monte Carlo (MCMC) algorithm, namely, that of slice sampling, is used in order to sample from the posterior density of interest. This probabilistic approach is tested and compared to an optimisation technique, *i.e.,* the popular least squares approach [104]. The advantages and limitations of both techniques are discussed in detail. There are two main advantages of using a probabilistic approach. First, the solution provided takes on the form of a probability distribution, so the uncertainty on the estimated position can be quantified [61], rather than approximated, *e.g.*, by computing the Cramer–Rao bound, as in [57]. Another important advantage of the proposed Bayesian technique is that it overcomes the convergence problems (to local minima) that the least squares approach could face.

## 5.3 Advection-diffusion models

Processes that move a compound from the location at which it is generated through the air (or water) are called mass transport processes and can be divided into two categories: diffusion and advection. Diffusion works to eliminate discontinuities in concentration profiles by transporting a compound from a region where its concentration is high to a region of a lower concentration through the action of random motions. A simple example is that of a cigarette smoke which diffuses into the air and spreads throughout the room or a drop of ink which, once dropped in a glass of water, diffuses into the whole glass. Advection on the other hand refers to the transport of a compound due to the mean fluid velocity or flow. For instance, a wind blowing to the north carries a compound present in the atmosphere towards the north by advection.

In the following, we start by considering the case of a single emitting source located at $(x_s, y_s, z_s)^T$ in an unbounded domain $(x, y, z)$. We deal first with the case of an instantaneous point release source: we derive the expression of the concentration $C(\mathbf{r}, t)$ at position $\mathbf{r} = (x, y, z)^T$ at time $t$ with an assumption that the emitted compound is exclusively transported by diffusion (*i.e.* the advection time scale is slow compared to the diffusive time scale); then using the previous result we generalise to the case of a transport by advection and diffusion. Second, we derive a concentration profile by solving the advection-diffusion equation in the case of a continuous release source emitting a substance at a constant rate.

### 5.3.1 Instantaneous release source

#### 5.3.1.1 Transport by diffusion

Consider that an instantaneous release of a total mass $m$ $[Kg]$ occurs at time $t_0$ at point $\mathbf{r}_s = (x_s, y_s, z_s)^T$. The emitted compound molecules are diffused in the atmosphere, the variation in time and space of the compound concentration $C(\mathbf{r}, t)$ $[Kg/m^3]$ is governed by the equation of mass conservation [60]:

$$\frac{\partial C}{\partial t} = -\nabla \mathbf{q}, \tag{5.1}$$

where

$$\mathbf{q} = -\mathbf{K}.\nabla C \quad [Kg/s/m^2] \tag{5.2}$$

is the mass flux per unit area and $\mathbf{K}$ is a tensor grouping the diffusivities $(m^2/s)$ in the $x$, $y$ and $z$ directions and is assumed to be diagonal, *i.e.* $\mathbf{K} = diag(K_x, K_y, K_z)$. Under the assumption $K_x = K_y = K_z = K$, equation (5.1) becomes

$$\frac{\partial C}{\partial t} = K\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2}\right), \tag{5.3}$$

and is called the diffusion equation.

Let us begin with a one-dimensional system in $x$, *e.g.* the total amount of substance $m$ is diffusing in a cylinder of an infinite length and a cross-sectional area $A_{yz}$ $[m^2]$. The diffusion equation reduces to

$$\frac{\partial C}{\partial t} = K\left(\frac{\partial^2 C}{\partial x^2}\right). \tag{5.4}$$

The boundary conditions for an unbounded domain are

$$C(x,t), \quad \frac{\partial C}{\partial x} \longrightarrow 0 \ \text{ if } \ x \longrightarrow \pm\infty,$$

while the initial conditions are

$$C(\mathbf{r},t_0) = C(x,t_0) = \frac{m}{A_{yz}}\delta(x-x_s).$$

The solution of equation (5.4) has the form [30]:

$$C(x,t) = \frac{B}{(t-t_0)^{(1/2)}} \exp\left(-\frac{(x-x_s)^2}{4K(t-t_0)}\right), \tag{5.5}$$

where $B$ is a constant to be determined using the following mass conservation condition:

$$\int_{-\infty}^{+\infty} C(x,t)dx = m.$$

Setting $\lambda^2 = x^2/4K(t-t_0)$ and integrating with respect to $\lambda$ we can find:

$$C(x,t) = \frac{m/A_{yz}}{(4\pi K(t-t_0))^{(1/2)}} \exp\left(-\frac{(x-x_s)^2}{4K(t-t_0)}\right). \tag{5.6}$$

In the case of a three-dimensional system, the initial and the boundary conditions for an unbounded domain become, respectively:

$$\begin{aligned} C(\mathbf{r},t_0) &= C(x,y,z,t_0) = m\delta(x-x_s, y-y_s, z-z_s), \\ C(\mathbf{r},t), \ \frac{\partial C}{\partial x}, \ \frac{\partial C}{\partial y}, \ \frac{\partial C}{\partial z} &\longrightarrow 0 \ \text{ if } \ x,y,z \longrightarrow \pm\infty. \end{aligned}$$

The solution to equation (5.3) can be found using the variable separation technique, thus, we try to determine a solution having the form

$$C(x,y,z,t) = C_1(x,t)C_2(y,t)C_3(z,t)$$

which satisfies the condition of mass conservation

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} C(x,y,z,t)dxdydz = m.$$

This expression can be easily deduced from expression (5.6) and is given by [30, 60]:

$$C(x,y,z,t) = \frac{m}{(4\pi K(t-t_0))^{(3/2)}} \exp\left(-\frac{(x-x_s)^2+(y-y_s)^2+(z-z_s)^2}{4K(t-t_0)}\right). \tag{5.7}$$

Note that the concentration is in fact expressed in terms of the distance $||\mathbf{r}-\mathbf{r}_s||$ between location $\mathbf{r}$ and the source.

Figure 5.1 illustrates the variation in space and in time of the concentration of the emitted chemical in function of the distance $||\mathbf{r}-\mathbf{r}_s||$, separating the point where the concentration is estimated from the source, according to equation (5.7). In this example, the parameters are fixed to $t_0 = 0$, $K = 25\ m^2/s$ and $m = 1\ mg$.

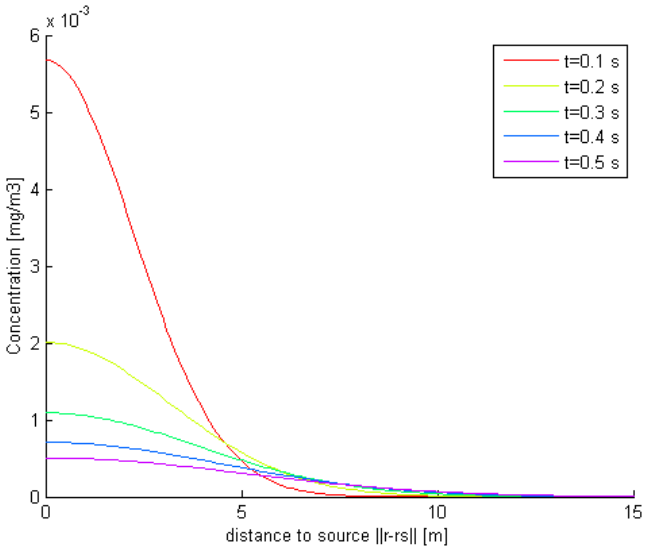



*Fig. 5.1:* Concentration profile: case of an instantaneous source and assuming the transport of the emitted material occurs exclusively by diffusion.

#### *5.3.1.2 Transport by advection-diffusion*

In this section, we consider the combined transport of the emitted substance by advection and diffusion for a single instantaneous point release. Considering a mean wind velocity $\mathbf{v} = (V_x, V_y, V_z)$, the mass flux per unit area is now given by:

$$\mathbf{q} = C\mathbf{v} - \mathbf{K}.\nabla C. \tag{5.8}$$

Under the same assumptions made in the previous section, the advection-diffusion equation is deduced from condition (5.1) and is written as:

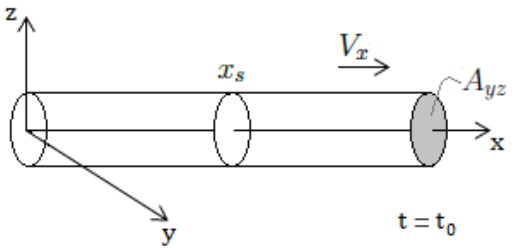
$$\frac{\partial C}{\partial t} + V_x\frac{\partial C}{\partial x} + V_y\frac{\partial C}{\partial y} + V_z\frac{\partial C}{\partial z} = K\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2}\right). \tag{5.9}$$

Again, we begin with a one-dimensional system as described in the previous section, a total mass release $m$ occurs at time $t_0$ at location $x_s$. The advection-diffusion equation reduces to

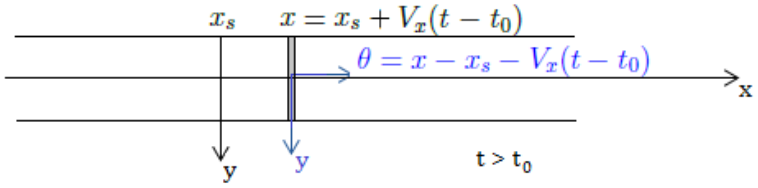
$$\frac{\partial C}{\partial t} + V_x\frac{\partial C}{\partial x} = K\left(\frac{\partial^2 C}{\partial x^2}\right), \tag{5.10}$$

with the same initial and boundary conditions for an unbounded domain.

To solve equation (5.10), let us change the frame of reference from the stationary coordinate system $(x, t)$ to the moving frame $(\theta, t)$ where $\theta = x - x_s - V_x(t - t_0)$ which moves with a velocity equal to that of the mean wind flow $V_x$ (see figure 5.2). Using the




*(a)* A one-dimensional system with an instantaneous release at $t_0$ at position $x_s$.




*(b)* The new frame of reference indicated in blue is moving with a velocity $V_x$.

*Fig. 5.2:* The stationary and the new moving coordinate systems.

chain rule of differentiation, we have

$$\begin{aligned}
\frac{\partial C(\theta,t)}{\partial t} &= \frac{\partial C}{\partial \theta}\frac{\partial \theta}{\partial t} + \frac{\partial C}{\partial t}\frac{\partial t}{\partial t} = -V\frac{\partial C}{\partial \theta} + \frac{\partial C}{\partial t}, \\
\frac{\partial C(\theta,t)}{\partial x} &= \frac{\partial C}{\partial \theta}\frac{\partial \theta}{\partial x} + \frac{\partial C}{\partial t}\frac{\partial t}{\partial x} = \frac{\partial C}{\partial \theta}, \\
\frac{\partial^2 C(\theta,t)}{\partial x^2} &= \frac{\partial^2 C}{\partial \theta^2}.
\end{aligned} \tag{5.11}$$

Replacing expressions (5.11) in the differential equation (5.10), we have

$$\frac{\partial C}{\partial t} = K\frac{\partial^2 C}{\partial \theta^2}, \tag{5.12}$$

with the initial conditions $C(\theta, t_0) = \frac{m}{A_{yz}}\delta(\theta - \theta_s)$, $\theta_s = 0$, and the same boundary conditions in $\theta$. The solution is merely given by equation (5.6):

$$C(\theta,t) = \frac{m/A_{yz}}{(4\pi K(t-t_0))^{(1/2)}}\exp\left(-\frac{\theta^2}{4K(t-t_0)}\right), \tag{5.13}$$

and thus, in function of $x$,

$$C(x,t) = \frac{m/A_{yz}}{(4\pi K(t-t_0))^{(1/2)}}\exp\left(-\frac{(x-x_s-V_x(t-t_0))^2}{4K(t-t_0)}\right). \tag{5.14}$$

In the case of a three-dimensional system, the solution to equation (5.9) can be deduced from the previous expression as shown previously (using variable separation technique) and is given by [60]:

$$C(x,y,z,t) = \frac{m}{(4\pi K(t-t_0))^{(3/2)}}\exp\left(-\frac{(x-x_s-V_xt)^2+(y-y_s-V_yt)^2+(z-z_s-V_zt)^2}{4K(t-t_0)}\right). \tag{5.15}$$

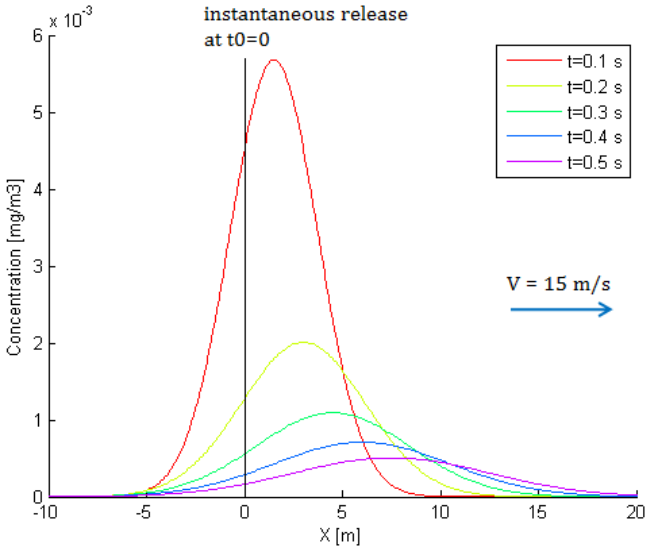



*Fig. 5.3:* Variation of the concentration in space and time in the case of a 1D instantaneous source and assuming the transport of the emitted material occurs by advection-diffusion.

Figure 5.3 shows the variation of the concentration profile in a one-dimensional system caused by an instantaneous point source located at $x_s = 0$ and emitting a total mass $m = 1mg$ at $t_0 = 0$, the wind flow is directed along the $x$-axis and its mean velocity is set to $V_x = 15m/s$.

#### 5.3.1.3 *Diffusion in a semi-infinite medium: method of reflection and superposition*

Consider the one-dimensional system in $x$ described here above. In the case of an unbounded domain we have yet considered, half of the emitted material moves in the direction of $x > x_s$ and the other half along $x < x_s$.

Assuming an instantaneous point source located at $x_s$ of a semi-infinite cylinder extending over $x > x_s$, *e.g.* imagine having an impermeable boundary at $x = x_s$, all of the emitted substance diffuses along the direction $x > x_s$. We can consider [30] that the solution for $x < x_s$ is reflected on the boundary $x = x_s$ and superposed on the original distribution for $x > x_s$. Note that the solution in the case of an infinite medium given in equation (5.6) is symmetrical about $x_s$, thus the concentration in a semi-infinite medium is given by:

$$C(x,t) = 2 \times \frac{m/A_{yz}}{(4\pi K(t-t_0))^{(1/2)}} \exp\left(-\frac{(x-x_s)^2}{4K(t-t_0)}\right). \tag{5.16}$$

The principle of reflection and superposition also allows us to deduce the solution to the advection-diffusion equation given in (5.9) for the case of a semi-infinite medium as illustrated through the following example.

**Example 5.3.1.** Consider a Cartesian coordinate system $(x, y, z)$ where the $x$-axis is chosen to be oriented in the direction of the mean wind velocity, the $y$-axis represents the horizontal cross-wind direction and the $z$-axis is oriented in the upward vertical direction. The mean wind velocity will be denoted as $\mathbf{v} = (V, 0, 0)$ $[m/s]$ where $V$ is constant. Suppose that an instantaneous point source is located at $(x_s, y_s, z_s)$ on an impermeable surface $z = z_s = 0$. The solution in the case of an infinite medium is given in expression (5.15)

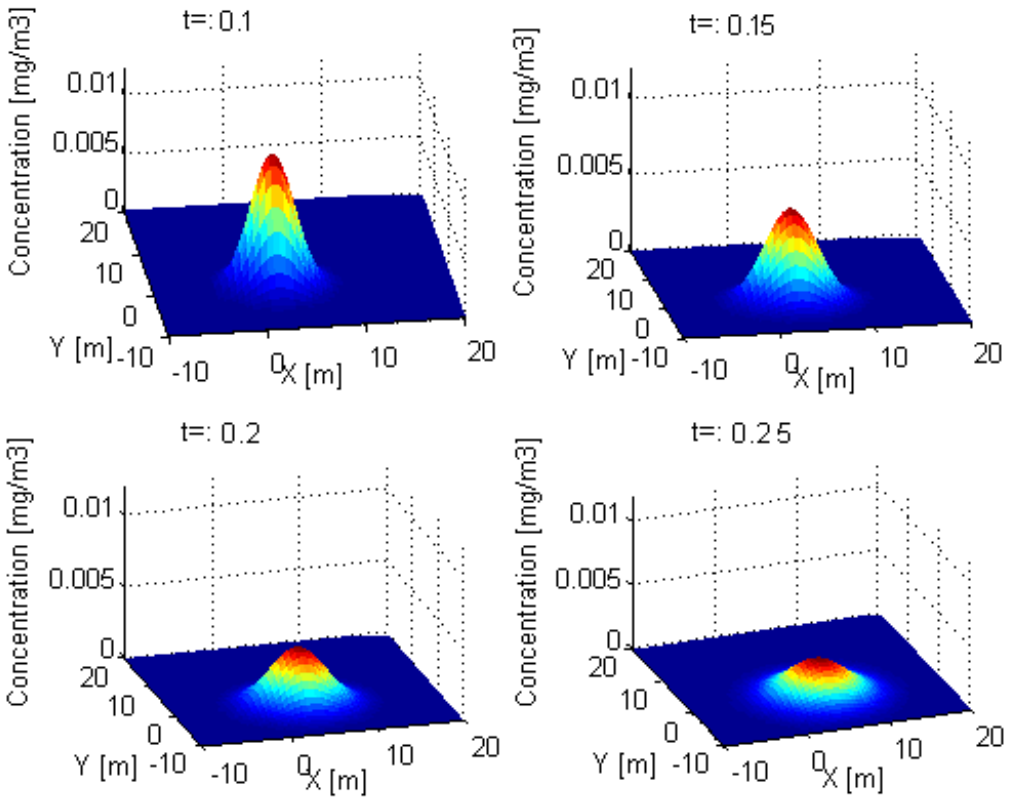



*Fig. 5.4:* Concentration profile in the case of an instantaneous source placed in a semi-infinite medium and assuming the transport of the emitted material occurs by advection-diffusion.

and can be written for this example as

$$C(x,y,z,t) = \frac{m}{(4\pi K(t-t_0))^{(3/2)}} \exp\left(-\frac{(x-x_s-Vt)^2+(y-y_s)^2+z^2}{4K(t-t_0)}\right). \qquad (5.17)$$

Using the principle of reflection and superposition, the solution for $z < 0$ is reflected on the impermeable boundary $z = z_s = 0$ and superposed on the concentration for $z > 0$, since the original solution is symmetric about $z_s = 0$, the resultant concentration in a semi-infinite medium is given by

$$C(x,y,z,t) = 2 \times \frac{m}{(4\pi K(t-t_0))^{(3/2)}} \exp\left(-\frac{(x-x_s-Vt)^2+(y-y_s)^2+z^2}{4K(t-t_0)}\right). \qquad (5.18)$$

Figure 5.4 shows the concentration profile at level $z = 0$ in the case of an instantaneous point source placed at $(0,0)$ on the impermeable surface $z = 0$. The wind velocity is chosen to be $V = 20m/s$ and the other parameters $K = 25m^2/s$, $t_0 = 0$ and $m = 1mg$.

### 5.3.2 Continuous release source

Let us first consider the case of a single source placed at position $\mathbf{r}_s = (x_s, y_s, z_s)^T$ on the impermeable plane boundary $z = 0$. The source emits a chemical with a constant rate $Q$ (g/s) starting at time $t_0$. The mean wind velocity is modelled as a constant

vector directed along the $x$-axis and of a constant magnitude $V$ (m/s). The differential equation [7, 72] governing the variation of the concentration $C(\mathbf{r}, t)$ of the emitted compound at time $t \geq t_0$ and at position $\mathbf{r} = (x, y, z)^T$ in the semi-infinite medium $z \geq 0$ is given by:

$$\begin{aligned} &\frac{\partial C}{\partial t} - K\left(\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2}\right) + V\frac{\partial C}{\partial x} \\ &\qquad = 2Q.u(t - t_0).\delta(x - x_s).\delta(y - y_s).\delta(z - z_s) \end{aligned} \tag{5.19}$$

where $K$ $(m^2/s)$ is the isotropic air diffusion coefficient; $\delta(.)$ denotes the Dirac delta function and $u(t - t_0)$ refers to the step function vanishing for $t < t_0$ and equal to unity for $t \geq t_0$. With an initial condition $C(\mathbf{r}, t) = 0, \quad \text{for} \quad t < t_0$, the solution [17] for $t \geq t_0$ at position $\mathbf{r}$ is given by:

$$\begin{aligned} C(Q, \mathbf{r}_s, \mathbf{r}, t, t_0) = &\frac{Q}{\pi^{\frac{3}{2}} K d} exp\left(\frac{V(x - x_s)}{2K}\right) \\ &\times \int_{\frac{d}{2\sqrt{K(t-t_0)}}}^{\infty} exp\left(-u^2 - \frac{V^2 d^2}{16K^2 u^2}\right) du \end{aligned} \tag{5.20}$$

where $d = d(\mathbf{r}_s, \mathbf{r}) = ||\mathbf{r}_s - \mathbf{r}||$ is the Euclidean distance between positions $\mathbf{r} = (x, y, z)^T$ and $\mathbf{r}_s = (x_s, y_s, z_s)^T$.

Under the assumptions of constant $V$ and $Q$, and after a sufficiently long time $(t \to \infty)$, a stationary concentration profile [7, 72], given by the following concentration, is established:

$$C_\infty(Q, \mathbf{r}_s, \mathbf{r}) = \frac{Q}{2\pi K d} exp\left(-\frac{V.\big(d - (x - x_s)\big)}{2K}\right) \tag{5.21}$$

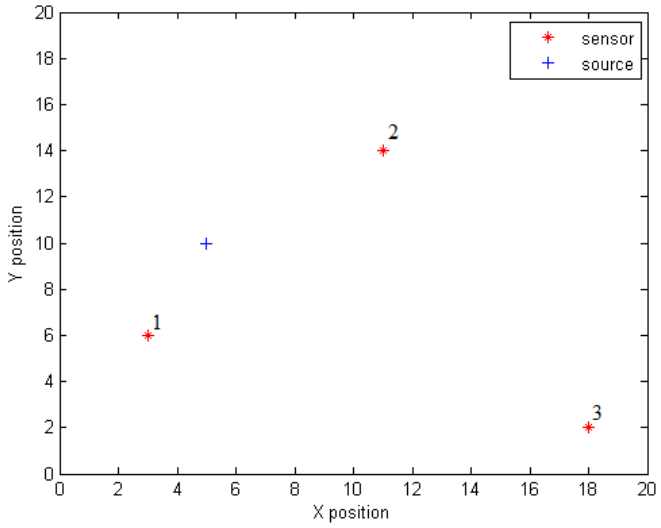



*Fig. 5.5:* An illustrative example with a single continuous point source.

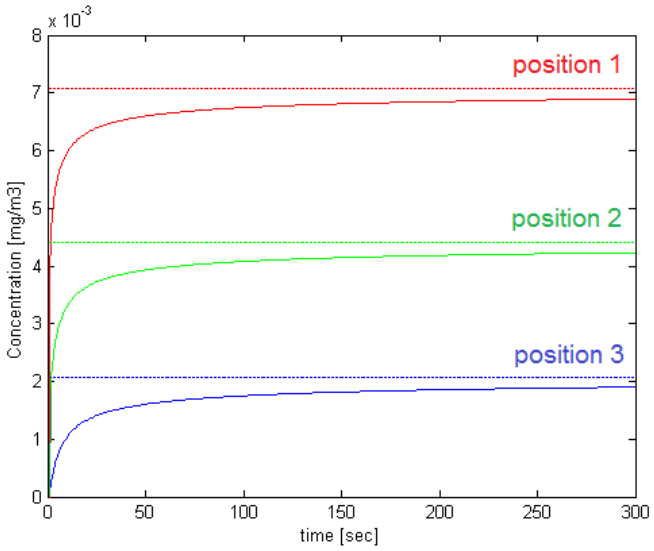



*Fig. 5.6:* Concentration profiles are given in solid line at positions 1, 2 and 3 for the example given in figure 5.5. Dotted lines show the stationary concentrations.

Figures 5.5 and 5.6 show respectively an illustrative example with one continuous point source placed at position $(x_s = 5, y_s = 5)$ in the plane $z = 0$, and the variation of the concentration at three different positions in a 20 m $\times$ 20 m planar region, *i.e.* at level $z = 0$, as given in equation (5.20). The parameters are set to $Q = 5mg/s$, $K = 25m^2/s$ and $V = 5cm/s$. The stationary concentrations determined using expression (5.21) are also shown in dotted lines. The graph shows that the concentration change within 5 minutes reaches 97.50, 95.96 and 91.40 percent of the stationary concentration that would be established at the positions 1, 2 and 3, respectively. Figure 5.7 illustrates the stationary concentration profile for this example.

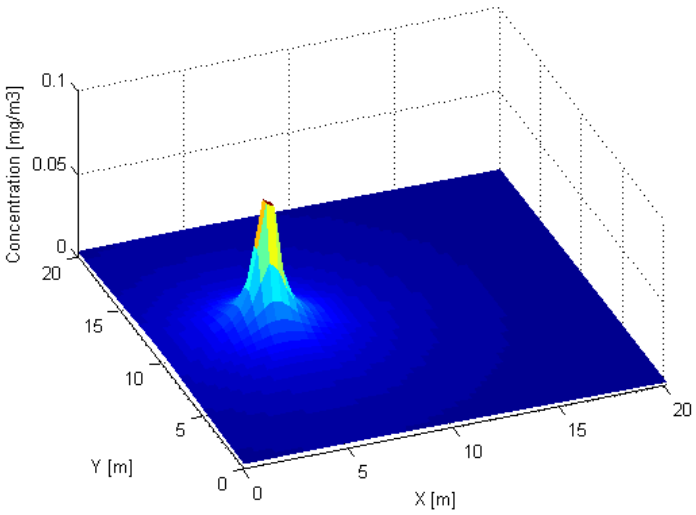



*Fig. 5.7:* The stationary concentration profile for the example in figure 5.5.

### 5.3.3 Case of multiple point sources

Let us consider now the case of $N$ point sources where the $i-th$ source, denoted as $S_i$, $i = 1, \ldots, N$, is at position $\mathbf{r}_{s_i} = (x_{s_i}, y_{s_i}, z_{s_i})^T$. Let us further assume that all

sources are on the same impermeable plane boundary $z = 0$ and that the wind velocity is directed along the $x$-axis.

Using the principle of superposition, and noting that the general advection-diffusion equation and its boundary conditions are linear, the *resultant* concentration [69] for $N$ instantaneous point sources each emitting a mass $m_i, i = 1, \ldots, N$ at time $t_0$ evaluated at position $\mathbf{r} = (x, y, z)^T$ is deduced from equation (5.18) by adding together (superposing) individual solutions and is hence given by:

$$C(x, y, z, t) = \sum_{i=1}^{N} \frac{m_i}{4(\pi K(t - t_0))^{(3/2)}} \exp\left(-\frac{(x - x_{s_i} - V(t - t_0))^2 + (y - y_{s_i})^2 + z^2}{4K(t - t_0)}\right). \tag{5.22}$$

Similarly, for the case of $N$ continuous point sources each emitting the compound at a constant rate $Q_i, i = 1, \ldots, N$, the resultant stationary concentration evaluated at position $\mathbf{r} = (x, y, z)^T$ can be written as:

$$C_r\left(\{Q_i, \mathbf{r}_{s_i}\}_{i=1}^{N}, \mathbf{r}\right) = \sum_{i=1}^{N} C_\infty(Q_i, \mathbf{r}_{s_i}, \mathbf{r}) \tag{5.23}$$

where $C_\infty(Q_i, \mathbf{r}_{s_i}, \mathbf{r})$ is given in equation (5.21).

## 5.4 The forward problem for land mine localisation

Recall that the purpose of part II of the thesis is to develop algorithms to locate vapour-emitting land mines exclusively using concentration measurements provided by a network of vapour detectors/sensors. The forward model is used to compute an estimated concentration of the explosive chemical at a certain location given a vector of parameters consisting of land mine locations, emission rates and environmental conditions, such as the diffusivity of the air and the wind velocity. It describes the transport of the explosive chemical emitted by the land mines due to the advection and diffusion processes. Note that numerical solutions for modelling the transport of TNT emanating from land mines [53] were proposed. In our approach, we model a land mine as a point source placed on an impermeable planar surface $z = 0$ and diffusing an explosive chemical, such as the TNT, and an analytical model for the transport of the explosive vapours is used. We keep the notations and assumptions of section 5.3.3.

Two possible scenarios are associated with the land mine localisation problem [57]:

a. the first scenario considers that after a long period of time, buried land mines no longer emit important amounts of explosive vapours and consequently the corresponding concentrations are not detectable. To overcome this problem, it was proposed to induce the evaporation of the explosive chemical (*i.e.* using microwaves). In this case, the land mines are considered as instantaneous point sources. This scenario is appropriate to model long-buried land mines.

b. the second scenario assumes that the land mines have been buried since a sufficiently long time for a stationary profile to be established. Here, the land mines are considered as continuous point sources and this scenario is more adapted to recently buried land mines.

Both scenarios will be considered in what follows. Furthermore, in the remaining sections, the sources/mines and the sensors are considered to be in the same plane $z = 0$, and we omit thus the third coordinate $z$ in all position vectors.

Consider $M$ sensors which are placed at known positions $\mathbf{r}_j = (x_j, y_j)^T,\ \ j = 1, \ldots, M$.

For scenario $(a)$, let us set $t_0 = 0$, the instant when the evaporation of the explosive chemical is induced and consider that the measurements are taken at time $t_1$. Referring to equation (5.22), the estimated concentration at position $\mathbf{r}_j$ can be written as:

$$C_j\left(\{m_i, \mathbf{r}_{s_i}\}_{i=1}^N, \mathbf{r}_j\right) = \sum_{i=1}^{N} m_i \times h_{ji}(\mathbf{r}_{s_i}, \mathbf{r}_j) \tag{5.24}$$

where

$$h_{ji}(\mathbf{r}_{s_i}, \mathbf{r}_j) = \frac{1}{4(\pi K t_1)^{(3/2)}} \exp\left(-\frac{1}{4Kt_1}\Big((x_j - x_{s_i} - Vt_1)^2 + (y_j - y_{s_i})^2\Big)\right). \tag{5.25}$$

For scenario $(b)$, we refer to equations (5.21) and (5.23), the estimated concentration at position $\mathbf{r}_j$ of sensor $j$ can be expressed as:

$$C_j\left(\{Q_i, \mathbf{r}_{s_i}\}_{i=1}^N, \mathbf{r}_j\right) = \sum_{i=1}^{N} Q_i \times g_{ji}(\mathbf{r}_{s_i}, \mathbf{r}_j) \tag{5.26}$$

where

$$g_{ji}(\mathbf{r}_{s_i}, \mathbf{r}_j) = \frac{1}{2\pi K d_j^i} exp\left(-\frac{V.\Big(d_j^i - (x_j - x_{s_i})\Big)}{2K}\right) \tag{5.27}$$

and $d_j^i = ||\mathbf{r}_{s_i} - \mathbf{r}_j||$.

### 5.4.1 Formulation of the forward problem

Let $\mathbf{P}$ denote the vector of parameters of the sources, *i.e.*, sources' positions and emission rates. For $N$ land mines,

$$\begin{aligned} \mathbf{P} &= [\{x_{s_i}, y_{s_i}, m_i\}_{i=1}^N]^T, \quad \text{(scenario } (a)\text{)} \\ \mathbf{P} &= [\{x_{s_i}, y_{s_i}, Q_i\}_{i=1}^N]^T, \quad \text{(scenario } (b)\text{)} \end{aligned}$$

The concentration measurements provided by the sensor network are grouped in an array denoted by $\mathbf{Y}^m = [C_1^m, \ldots, C_M^m]^T$.

Let $\mathbf{Y}^t = [C_1^t, \ldots, C_M^t]^T$ denote the vector of true concentrations at positions $\mathbf{r}_j$, and $\mathbf{E}^m = [e_1^m, \ldots, e_M^m]^T$ a vector of measurements error, thus:

$$C_j^m = C_j^t + e_j^m, \quad j = 1, \ldots, M \tag{5.28}$$

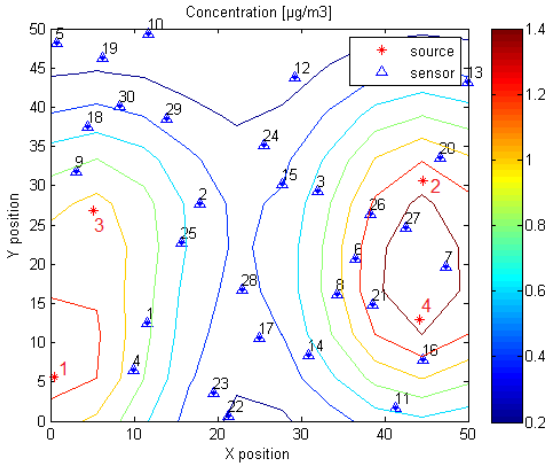
Furthermore, if $\mathbf{Y}^e = [C_1^e, \ldots, C_M^e]^T$ refers to the vector of *estimated* concentrations at $\mathbf{r}_j$, *i.e.*, the concentrations obtained by resolving the forward problem according to equations (5.24) or (5.26), and $\mathbf{E}^e = [e_1^e, \ldots, e_M^e]^T$ a vector of model error, then:

$$C_j^e = C_j^t + e_j^e, \quad j = 1, \ldots, M \tag{5.29}$$

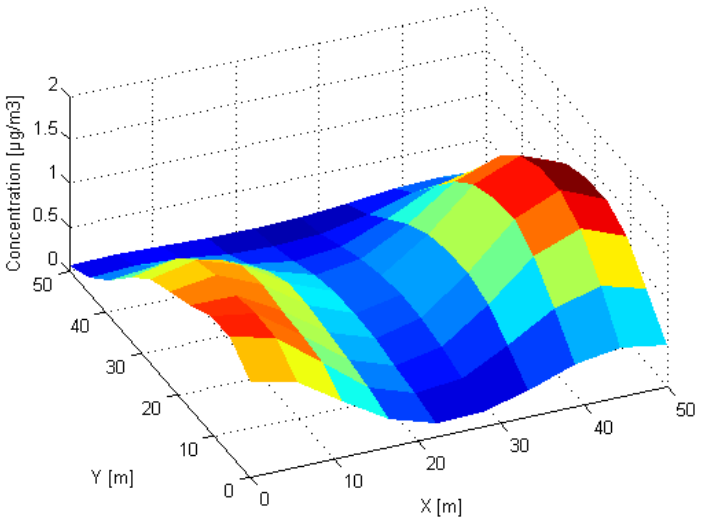
Based on equations (5.28) and (5.29), we can write:

$$C_j^m = C_j^e + e_j, \quad j = 1, \ldots, M \tag{5.30}$$

where $e_j$ denotes the quantity characterising the difference between measured and mod-




*(a)* Concentration profile at $t_1 = 2s$ for a scenario with $N = 4$ instantaneous point sources and $M = 30$ sensors.




*(b)* The corresponding perspective view.

*Fig. 5.8:* Illustration of the forward model for scenario $(a)$.

elled/estimated concentrations. Equivalently, in vector notation:

$$\begin{aligned} \mathbf{Y}^m &= \mathbf{Y}^t + \mathbf{E}^m \\ \mathbf{Y}^e &= \mathbf{Y}^t + \mathbf{E}^e \end{aligned} \tag{5.31}$$

and:

$$\mathbf{Y}^m = \mathbf{Y}^e + \mathbf{E} \tag{5.32}$$

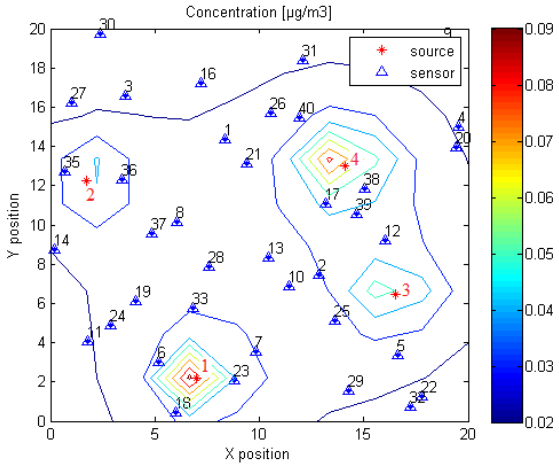
where $\mathbf{E}$ refers to an error. Based on (5.31) and (5.24) for scenario $(a)$, (5.26) for scenario $(b)$, we can define an operator $\mathbf{F}$, such that:

$$\mathbf{Y}^m = \mathbf{F}(\mathbf{P}, \mathbf{A}) \tag{5.33}$$

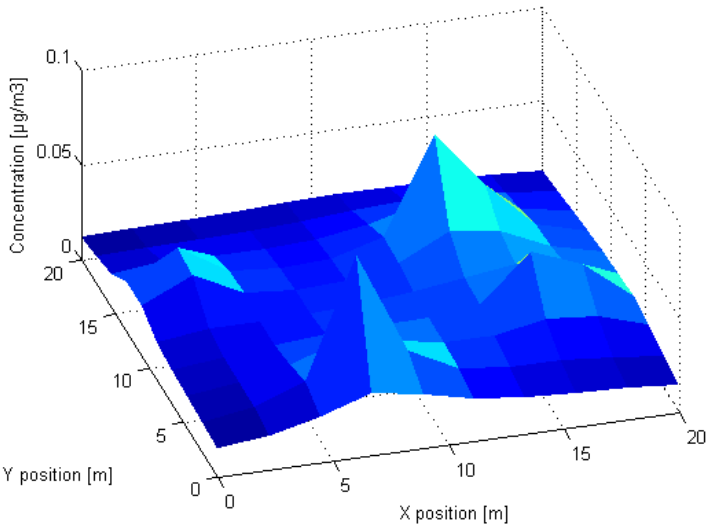
where $\mathbf{A}$ is a constant vector grouping information about the environment, sensors characteristics and model applicability, e.g., wind velocity, air diffusion coefficient and noise variances.

Figure 5.8 shows the concentration profile at $t_1 = 2s$ of the emitted chemical determined using equation (5.22) for a scenario with $N = 4$ instantaneous point sources and $M = 30$ sensors, which are randomly deployed in a $50 \times 50$ $m^2$ planar region. Emissions are randomly fixed to $m_1 = 9.78$ $\mu g$, $m_2 = 8.43$ $\mu g$, $m_3 = 7.58$ $\mu g$, and $m_4 = 9.94$ $\mu g$.

Figure 5.9 on the other hand shows the stationary concentration profile of the diffused explosive chemical determined using equation (5.23) for a scenario with $N = 4$ continuous point sources and $M = 40$ sensors, which are randomly deployed in a $20 \times 20$ $m^2$ planar region. Emission rates are randomly fixed to $Q_1 = 8.01$ $\mu g/s$, $Q_2 = 5.38$ $\mu g/s$, $Q_3 = 5.87$ $\mu g/s$ and $Q_4 = 9.39$ $\mu g/s$. For both examples, the environmental parameters are set to $V = 5cm/s$ and $K = 25m^2/s$.




*(a)* Stationary concentration profile for a scenario with $N = 4$ continuous point sources and $M = 40$ sensors.




*(b)* The corresponding perspective view.

*Fig. 5.9:* Illustration of the forward model for scenario (*b*).

## 5.5 Conclusion

This chapter formulates the forward problem for land mines localisation. This model describes the transport of the explosive chemical emitted by the source into the surrounding area due to advection and diffusion. It allows to predict the concentration at the sensors' positions if the sources parameters (coordinates and emissions) are known. A literature review on the solution of differential advection-diffusion equations was provided and two scenarios, applicable to long-buried and recent-buried land mines, were considered.

# 6. THE INVERSE PROBLEM FOR SOURCE CHARACTERISATION

## *6.1 Introduction*

In this chapter, we consider the problem of localising an unknown number of land mines exclusively using concentration measurements provided by a wireless sensor network. We adopt a two-step approach consisting in sequentially determining the number of sources, and then characterising them (*i.e.* computing their coordinates and emissions) by solving the inverse problem associated with the forward model formulated in the previous chapter. We start in section 6.2 with a general overview of inverse problems and their common solution techniques. The first step consisting in determining the unknown number of sources is dealt with in section 6.3 where we use the principal component analysis (PCA), a well known dimension reduction technique, on a matrix of concentration measurements. Once the number of sources is determined, the inverse problem describing the source characterisation problem is formulated in section 6.4. Two solution techniques are reported, a probabilistic Bayesian approach based on a Markov Chain Monte Carlo (MCMC) sampling scheme, and an optimisation least squares technique. The effectiveness of these schemes is tested on simulated data for two scenarios, considering respectively instantaneous and continuous release sources, in section 6.5. Section 6.6 at last recapitulates the conclusions drawn about this chapter.

## *6.2 Generalities on inverse problems*

### *6.2.1 General formulation*

The *inverse problem* can be defined as the process of inferring causes, conditioned on knowledge of the effects, as opposed to the forward or direct problem, allowing to determine the effects knowing the causes. Typically, the forward and corresponding inverse problems are linked via a possibly non-linear operator $\mathbf{H}$ which represents the relation between a system's input and output.

An example of an inverse problem, which will be of main concern in this chapter, is the one of parameter identification [104]. The corresponding forward problem consists of determining the output of a system knowing the system's parameters. If $\mathbf{P}$ denotes a vector of parameters which is linked by an operator $\mathbf{H}$ to vector $\mathbf{Z}$ grouping the system's outputs (possibly noise-corrupted), then the forward model can be written as $\mathbf{Z} = \mathbf{H}(\mathbf{P})$, and the inverse problem refers to solving $\mathbf{P} = \mathbf{H}^{-1}(\mathbf{Z})$.

A practical difficulty in the study of an inverse problem is that it is often ill-posed [61], meaning that an inverse transformation of the direct model may not exist, may not be unique, and might be unstable.

### *6.2.2 Common solution techniques*

The solution of an inverse problem can be approached in several distinct ways. Amongst the most general and popular techniques are least squares and regularisation approaches.

The key idea for both schemes is to reformulate the inverse problem as an optimisation problem, usually consisting of minimising a functional error between actual measurements and predicted ones obtained by resolving the direct problem. However, for regularisation, the objective function is better-conditioned through the use of regularisation parameters which allows a stable approximation of the inverse operator.

Probabilistic approaches can also be used to address the problem of parameter identification. A primary advantage of probabilistic methods is that the solution takes on the form of a probability distribution rather than a point solution, optimal in terms of a given criterion [61]. In this chapter, we compare the optimisation least squares and the probabilistic Bayesian approaches while citing the advantages and limitations of each scheme.

## 6.3 Determining the unknown number of sources

The purpose of this chapter is to determine unknown land mines/sources' parameters including their positions and emissions. We also consider the case where the number of buried land mines is unknown a priori. In this section, we use the principal component analysis (PCA) in order to determine the number of sources based on a set of measurements. PCA [3] is a popular statistical method that has been widely applied in the analysis of multidimensional data sets, which are usually represented by tables of observations of many possibly inter-correlated variables. Since the information provided by these variables is often redundant, PCA attempts to replace the original set of variables by a smaller number of new variables, called principal components, without losing too much information. This technique considers that the new variables are linear combinations of the original ones and that they are linearly uncorrelated.

Mathematically, PCA transforms the data to a new coordinate system, such that the original set of observations is expressed in terms of the principal components. A technique for performing PCA is to compute the covariance matrix calculated from the available measurements and then to determine its eigenvalues. The corresponding normalised eigenvectors, ordered according to decreasing eigenvalues, define the new coordinate system. A smaller dimensional coordinate system, which is supposed to conserve most of the information, can be obtained by only retaining the eigenvectors associated with the largest eigenvalues.

In our application, $M$ sensors are randomly deployed in a region of interest where the unknown number of vapour-emitting sources is denoted by $N$. Each sensor provides measurements of the concentration of the explosive material emitted by the sources and transported to the sensor's position due to advection and diffusion processes. Supposing that concentration measurements are taken on each sensor at $T$ time instants, the measurements are grouped in an $M \times T$ matrix $\mathbf{C} = (C_{jt})$, $j = 1, \ldots, M, t = 1, \ldots, T$, where the $j-th$ row groups the concentration measurements recorded by the $j-th$ sensor at different time instants and the $t-th$ column groups measurements recorded by all $M$ sensors at time instant $t$. We explain in the following how the matrix $\mathbf{C}$ is obtained for both scenarios $(a)$ and $(b)$.

**Scenario (a)**

In this case, the emissions of the sources are considered to be induced and thus the land mines are considered as instantaneous sources. We assume that the evaporation of the chemical is induced $T$ times and that, each time, the concentration measurements are taken on the sensors after a constant period of time $t_1$. In the following, let

– $m_{it}$ denote the emission $[kg]$ of source $i, i = 1, \ldots, N$, induced at time $t, t = 1, \ldots, T$;

– $\mathcal{M} = (m_{it})$ an $N \times T$ matrix grouping the emissions of the sources, where each row $i$ is associated with source $i$ and each column $t$ groups the emissions of the sources at time $t$.

– $\mathcal{H} = (h_{ji})$ an $M \times N$ matrix grouping the factors $h_{ji}(\mathbf{r}_{s_i}, \mathbf{r}_j)$ associated with instantaneous source $i$ and sensor $j$, as given in equation (5.25)

If $\mathbf{C}_t$ denotes the $t-th$ column of matrix $\mathbf{C}$, *i.e.*, the vector of concentration measurements provided by the sensors at time $t$ and $\mathcal{M}_t$ denotes the $t-th$ column of matrix $\mathcal{M}$, *i.e.*, the vector of sources emission rates at time $t$, referring to Equation (5.24), we have in matrix notation:

$$\mathbf{C}_t = \mathcal{H} \times \mathcal{M}_t, \quad t = 1, \ldots, T. \tag{6.1}$$

Taking into account an additive noise, as in Equation (5.30), we obtain:

$$\mathbf{C} = \mathcal{H} \times \mathcal{M} + \boldsymbol{\epsilon}, \tag{6.2}$$

where $\boldsymbol{\epsilon} = (e_{jt}),\ 1 \leq j \leq M,\ 1 \leq t \leq T$ is an additive noise matrix.

**Scenario (b)**

For the case of continuous release point sources, since the land mines emissions are likely to change slowly over time, we model these emissions using piecewise constant functions. Referring to figure 5.6, the stationary concentration profile is established within only a few minutes; thus, the recorded concentration measurements can be considered as stationary concentrations established for the constant emission rates between time instants $t-1$ and $t$ similarly to what was suggested in [13]. Figure 6.1 recapitulates the assumptions we make for scenario $(b)$. It illustrates the variation of the emissions of three sources and indicates, using red boxes, the regions where stationary concentration profiles are established. The measurements grouped in matrix $\mathbf{C}$ are, for instance, taken at time instants $t_1$, $t_2$, *etc.* Note that these measurements can be chosen by simply examining the signals provided by the sensors and detecting stationary points, *i.e.* the steady-state concentration values.

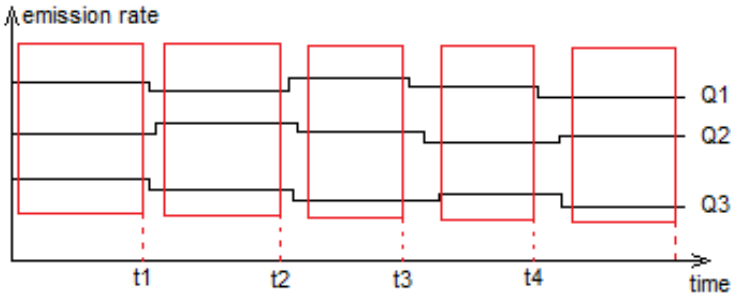



*Fig. 6.1:* The emission rates are piecewise constant, and a series of stationary concentration profiles are established.

For scenario $(b)$,

– $Q_{it}$ denotes the emission rate $[kg/s]$ of continuous source $i$ considered as constant between time instants $t-1$ and $t$;

– $\mathbf{Q} = (Q_{it})$ is an $N \times T$ matrix grouping the emission rates of the sources, where each row $i$ is associated with source $i$ and each column $t$ groups the emissions of the sources at time $t, t = 1, \ldots, T$;

– $\mathbf{G} = (g_{ji})$ refers to an an $M \times N$ matrix grouping the factors $g_{ji}(\mathbf{r}_{s_i}, \mathbf{r}_j)$ associated with continuous source $i$ and sensor $j$, as in equation (5.27).

Similarly, if $\mathbf{C}_t$ (respectively $\mathbf{Q}_t$) denotes the vector of measurements taken at the sensors (the sources' emission rates) at time $t$, looking back at equation (5.26), we can write

$$\mathbf{C}_t = \mathbf{G} \times \mathbf{Q}_t, \quad t = 1, \ldots, T \tag{6.3}$$

and again

$$\mathbf{C} = \mathbf{G} \times \mathbf{Q} + \boldsymbol{\epsilon}, \tag{6.4}$$

where $\boldsymbol{\epsilon}$ is an additive noise matrix.

Equations (6.2) and (6.4) shows that the concentrations measured on the sensors are linear combinations of the sources emissions/emission rates. In our application, the only available information is the matrix of concentrations $\mathbf{C}$. Performing the PCA technique on matrix $\mathbf{C}$ and retaining only the eigenvalues, which are larger than some threshold $\lambda_{th}$, allows to transform the data to a new coordinate system whose dimension is $d \leq M$. An ideal choice of the threshold would recover precisely the number of sources ($d = N$). The condition $d \leq M$ implies that the number of sensors cannot be less than the number of mines.

In our simulations, the number of concentration records is fixed to $T = 10$, and at each time instant, the emissions (emission rates) of the sources are drawn uniformly from the interval $[5 \;\; 10]\mu g$ ($\mu g/s$); and a white Gaussian noise of standard deviation $\sigma$ is added to the estimated concentrations. The sources and the sensors are randomly placed in an L $\times$ L planar region.

In order to develop the relationship when $N \leq M$, we studied what is the minimal number of sensors that can detect the number $N$ of the true hidden sources using the PCA technique for the simulation conditions described above. In our study, we fixed first the number $N$ of sources and initiated $M = N$. A hundred source-sensor configurations, obtained by randomly deploying $N$ sources and $M$ sensors in an L $\times$ L region, are generated. The PCA technique is tested on the different configurations, and the number of sensors is iteratively increased until the true number of sources is obtained for at least $\eta\%$ of the configurations. For the simulation conditions recapitulated in tables 6.1 and 6.3, an adequate threshold $\lambda_{th} = 10^{-7}$ is empirically determined.

| $T$ | $L$ $[m]$ | $K$ $[m^2/s]$ | $V$ $[cm/s]$ | $t_1$ $[s]$ | $\eta\%$ |
|---|---|---|---|---|---|
| 10 | 50 | 25 | 5 | 2 | 80 |

*Tab. 6.1:* Simulation parameters for scenario ($a$).

The results for scenario ($a$) are given in table 6.2.

| $N$ | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| $M$ | 6 | 11 | 22 | 45 | 100 | 206 |

*Tab. 6.2:* The minimal number of sensors required in order to detect the true number of instantaneous sources for the simulation conditions considered in table 6.1.

| $T$ | $L\ [m]$ | $K\ [m^2/s]$ | $V\ [cm/s]$ | $\eta\%$ |
|---|---|---|---|---|
| 10 | 20 | 25 | 5 | 90 |

*Tab. 6.3:* Simulation parameters for scenario $(b)$.

The results for scenario $(b)$ are given below in table 6.4. Note that for $T = 10$ and for the simulation conditions we fixed, the maximal detectable number is $N = 8$ sources.

| $N$ | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|
| $M$ | 5 | 10 | 16 | 26 | 35 | 48 | 71 |

*Tab. 6.4:* The minimal number of sensors required in order to detect the true number of continuous release sources for the simulation conditions fixed in table 6.3.

This technique is tested on the examples given in figures 5.8 and 5.9. First, a matrix of concentration measurements is obtained as described above and a principal component analysis is conducted.

The largest eigenvalues for scenario $(a)$ illustrated in figure 5.8 are $\lambda_1 = 0.1428 \times 10^{-4}$, $\lambda_2 = 0.0348 \times 10^{-4}$, $\lambda_3 = 0.0299 \times 10^{-4}$, $\lambda_4 = 0.0135 \times 10^{-4}$; their number is equal to four, which is the number of the sources. The sum of the remaining eigenvalues is $2.1818 \times 10^{-8}$.

For scenario $(b)$, we obtain $\lambda_1 = 0.1116 \times 10^{-3}$, $\lambda_2 = 0.0751 \times 10^{-3}$, $\lambda_3 = 0.0345 \times 10^{-3}$, $\lambda_4 = 0.0159 \times 10^{-3}$; the sum of the remaining eigenvalues is $4.1672 \times 10^{-8}$.

Thus, we were able at this first step to successfully determine the number of sources in the considered region. The next step is to localise the sources given concentration measurements.

## 6.4 Formulation of the inverse problem for source characterisation

Referring to Equation (5.33) where $\mathbf{A}$ is a constant vector, we can define an operator $\mathbf{H}$, such that:

$$\mathbf{Y}^m = \mathbf{H}(\mathbf{P}) \tag{6.5}$$

and introduce the inverse source characterisation problem as finding:

$$\mathbf{P} \approxeq \mathbf{H}^{-1}(\mathbf{Y}^m) \tag{6.6}$$

The exact solution for Equation (6.6) is usually not tractable. This is primarily due to the existence of model and measurement noises. In other words, the forward model is not perfectly known. Furthermore, due to the commutative property of the addition in Equations (5.24) and (5.26), the solution is not unique; for instance, in the case of two land mines, *i.e.*, $N = 2$, if $\mathbf{P}_1 = [\tilde{x}_{s_1}, \tilde{y}_{s_1}, \tilde{Q}_1, \tilde{x}_{s_2}, \tilde{y}_{s_2}, \tilde{Q}_2]^T$ is a solution to the inverse problem, then $\mathbf{P}_2 = [\tilde{x}_{s_2}, \tilde{y}_{s_2}, \tilde{Q}_2, \tilde{x}_{s_1}, \tilde{y}_{s_1}, \tilde{Q}_1]^T$ is also a solution. In general, any parameter vector obtained by only permuting the labels of the sources in a solution vector is also a solution. The estimation problem is thus over a set of land mines.

### 6.4.1 Bayesian Inference for Solving the Inverse Problem

In this section, the inverse problem is solved within a probabilistic Bayesian framework. Based on Equation (5.33), the problem consists in determining $\mathbf{P}$ having the vector of concentration measurements $\mathbf{Y}^m$ and some prior knowledge gathered in the constant vector

$\mathbf{A}$. In a Bayesian framework, this refers to finding the posterior distribution $p(\mathbf{P}|\mathbf{Y}^m, \mathbf{A})$. According to Bayes theorem:

$$p(\mathbf{P}|\mathbf{Y}^m, \mathbf{A}) = \frac{p(\mathbf{Y}^m|\mathbf{P}, \mathbf{A})p(\mathbf{P}|\mathbf{A})}{p(\mathbf{Y}^m|\mathbf{A})} \tag{6.7}$$

where $p(\mathbf{Y}^m|\mathbf{P}, \mathbf{A})$ is the measurement likelihood, $p(\mathbf{P}|\mathbf{A})$ is the prior distribution and $p(\mathbf{Y}^m|\mathbf{A})$ is the evidence. The evidence measures the suitability of the model (depending on the number of sources) to the available data [61]. The evidence values are calculated and compared for different models in order to determine the most probable number of sources. The higher the evidence, the better the model can predict the data. Since in our method, the number of sources is determined *a priori* as described in section 6.3, the evidence is considered as a normalisation factor.

Let us consider a bounded domain denoted as $\Omega$ for the land mine locations, *i.e.*, the sources lie within a bounded region $[x_{min} \;\; x_{max}] \times [y_{min} \;\; y_{max}]$, and the emissions/emission rates are also bounded within lower and upper bounds, $m_i \in [m_{min} \;\; m_{max}]$, $Q_i \in [Q_{min} \;\; Q_{max}] \;\; i = 1, \ldots N$. Choosing a non informative distribution for the prior, *i.e.*, a uniform PDF, we can write:

$$p(\mathbf{P}|\mathbf{Y}^m, \mathbf{A}) \propto 1_{\mathbf{P}\in\Omega} \;\; p(\mathbf{Y}^m|\mathbf{P}, \mathbf{A}) \tag{6.8}$$

where $1_{\mathbf{P}\in\Omega}$ denotes the indicator function taking on a value 1 if $\mathbf{P} \in \Omega$ and 0 if not. Additionally, if the measurement and model noises are assumed to be white and Gaussian, *i.e.*:

$$\begin{aligned} e_j^m &\sim N(0, \sigma_{m,j}^2), \\ e_j^e &\sim N(0, \sigma_{e,j}^2), \quad j = 1, \ldots, M \end{aligned}$$

then, it can be shown that:

$$p(\mathbf{Y}^m|\mathbf{P}, \mathbf{A}) \propto exp\left[-\frac{1}{2}\sum_{j=1}^{M} \frac{(C_j - C_j^e(\mathbf{P}))^2}{\sigma_{m,j}^2 + \sigma_{e,j}^2}\right] \tag{6.9}$$

Sampling directly from this distribution is difficult, and approximate numerical techniques must be used. A widely used approach for estimating the properties of the posterior distribution given in (6.8) is to perform Markov chain Monte Carlo (MCMC) sampling [4]. In MCMC algorithms, samples are drawn from the target distribution in the form of a Markov chain where each sample depends on the previous one in the chain. The earliest MCMC algorithm is the random walk Metropolis (RWM) [24]. Its basic principle is to sample a candidate value from a proposal distribution depending on the current position of the chain. The candidate is then accepted or rejected according to the Metropolis acceptance probability as will be defined using the following example. Consider sampling from a PDF $\pi(.)$. If $\mathbf{x}_{i-1}$ denotes the current state of the Markov chain, a trial state $\mathbf{z}$ is sampled according to $\mathbf{z} = \mathbf{x}_{i-1} + \mathbf{u}, \;\; where \;\; \mathbf{u} \sim N(0, \mathbf{\Sigma})$, for instance, and $\mathbf{\Sigma}$ denotes a covariance matrix. The candidate $\mathbf{z}$ is accepted or rejected according to the Metropolis acceptance probability $a$ given by:

$$a(\mathbf{x}_{i-1}, \mathbf{z}) = \begin{cases} \min\left[\frac{\pi(\mathbf{z})}{\pi(\mathbf{x}_{i-1})}, 1\right] & if \;\; \pi(\mathbf{x}_{i-1}) > 0 \\ 1 \;\; if \;\; \pi(\mathbf{x}_{i-1}) = 0. \end{cases}$$

If the candidate is accepted, the chain moves to $\mathbf{x}_i = \mathbf{z}$; otherwise, the chain remains at $\mathbf{x}_i = \mathbf{x}_{i-1}$. The procedure only requires the choice of a proposal function $f(\mathbf{.})$.

While the early RWM algorithm requires the proposal distribution $f$ to be symmetric, *i.e.*, $f(\mathbf{x}_{i-1}, \mathbf{z}) = f(\mathbf{z}, \mathbf{x}_{i-1})$, the Metropolis–Hastings (MH) algorithm [24] generalises the approach to non-symmetric proposals. Obviously, the choice of the proposal distribution is crucial to the algorithm convergence. Thus, several procedures were proposed in order to improve the algorithm's convergence. These include, for instance, the adaptive Metropolis algorithm [45], the differential evolution Markov chain Monte Carlo (DE-MC) [14] and the differential evolution adaptive Metropolis (DREAM) algorithm [107]. Another class of MCMC sampling techniques is the slice sampling technique [78] and will be used in this paper in order to draw samples from the posterior distribution given in Equation (6.8).

The slice sampling algorithm relies on the observation that sampling from a probability distribution, e.g., $\pi(.)$ in the case of a univariate distribution, can be done by drawing samples uniformly from the region under the plot of $\pi(.)$ [78]. It has an advantage over other MCMC methods, such as the Gibbs sampler and the RWM, in that the magnitude of the changes made to move from one element to the next in the chain is chosen adaptively.

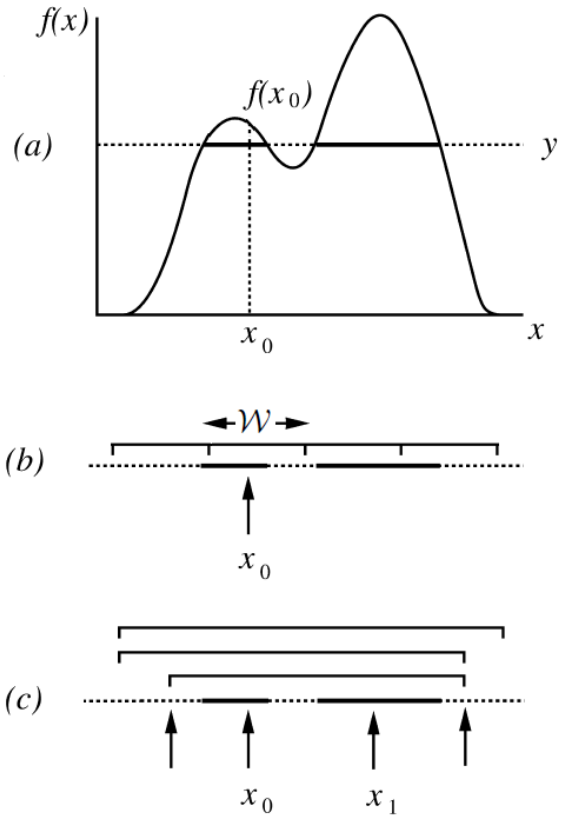



*Fig. 6.2:* Slice sampling (adapted from [78]).

Figure 6.2 illustrates the operations of the slice sampling algorithm in the case of a univariate target distribution $\pi(.)$. The procedure requires only the knowledge of a function $f(.)$ that is proportional to $\pi(.)$. It operates iteratively in three steps:

(a) Starting from the current position of the chain, denoted as $x_0$, and such that $f(x_0) > 0$, draw a value $y$ uniformly from the interval $[0, f(x_0)]$. The horizontal slice defined by $y$ consists of the values of $x$ for which $f(x) > y$ (see Figure 6.2a).
(b) Find an interval around $x_0$ comprising the majority, or the totality, of the slice defined in (a). Several methods can be used at this step. The approach adopted here (and

illustrated in Figure 6.2b) is called "stepping-out". It requires fixing, *a priori*, an interval width $\mathcal{W}$ and operates as follows: first, set an interval of width $\mathcal{W}$ randomly around $x_0$. Then iteratively expand this interval in steps of size $\mathcal{W}$ and stop when both interval ends become outside the slice $\{x, f(x) > y\}$.

(c) Draw a value $x_1$ from the part of the slice that is within the interval determined in (b). The technique used here is referred to as "shrinkage" (see Figure 6.2c) because it picks points uniformly from the determined interval, shrinks this last using points that are outside the slice, and stops whenever finding a point inside it.

Slice sampling can also be used to sample from multivariate distributions. This can be done by updating each variable in turn. It is useful though to note that slice sampling methods, which update all variables of a multivariate distribution simultaneously, do exist [78].

### 6.4.2 The Least Squares Technique for Source Characterisation

In this section, we formulate the source characterisation problem as an optimisation problem and propose to solve it using the least squares (LS) approach. LS is a popular method for solving the inverse problems [104]; it seeks an optimal point solution usually by minimising a quadratic error or cost function between actual measurements and synthetic ones estimated using the forward model.

Referring to Equation (5.32), we propose to solve the inverse source characterisation problem by minimising over $\mathbf{P}$, the vector of unknown parameters, the functional:

$$J = \left(\mathbf{Y}^m - \mathbf{Y}^e(\mathbf{P})\right)^T \left(\mathbf{Y}^m - \mathbf{Y}^e(\mathbf{P})\right) \tag{6.10}$$

The solution is thus given by:

$$\mathbf{P}^{opt} = \underset{\mathbf{P}}{\operatorname{argmin}}\ J \tag{6.11}$$

Despite the wide applicability, the ease of use and ease of understanding associated with the least squares technique, this method presents a well-known problem: it is sensitive to the convexity of the cost function [72] and can converge to local optima, thus diverging from the true solution.

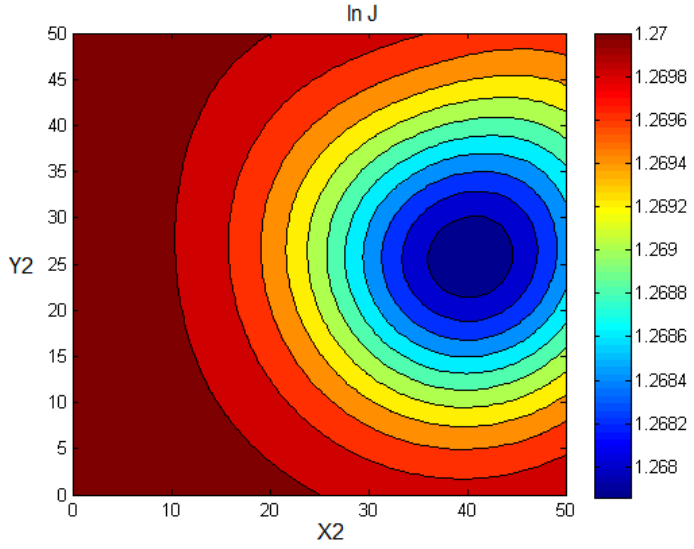



*Fig. 6.3:* Functional to be minimised in the logarithmic scale, case of instantaneous release sources.

Figure 6.3 illustrates the variation of the criterion $J$ in terms of Source 2 coordinates for the scenario shown in figure 5.8. The values of $J$ are calculated as a function of $(x_{s_2}, y_{s_2})$ after fixing the remaining unknown parameters, *i.e.*, $[\{x_{s_i}, y_{s_i}, Q_i\}_{i=1,3,4}, Q_2]^T$, to their true values. Note that the functional $J$ is convex and presents one global maximum near Source 2's true position. The least squares search algorithm is expected to provide accurate position estimates in this case.

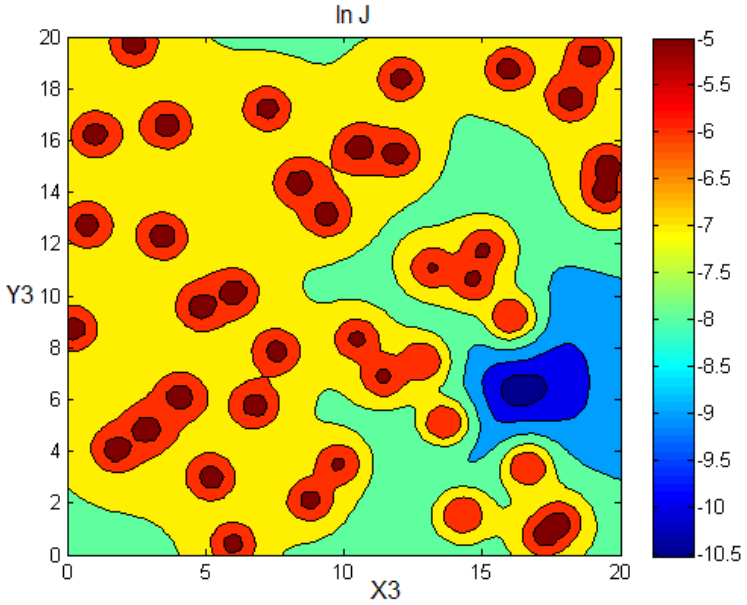



*Fig. 6.4:* Functional to be minimised in the logarithmic scale, case of continuous point sources.

Figure 6.4 illustrates the variation of the criterion $J$ in terms of Source 3 coordinates for the scenario shown in Figure 5.9. Again, the values of $J$ are calculated as a function of $(x_{s_3}, y_{s_3})$ after fixing the remaining unknown parameters to their true values. Note that the functional $J$ is not convex and presents local maxima at the sensors' positions. This will cause the convergence of the least squares search algorithm to a local minimum, as we show in the next section, where we test the least squares technique on a simulated scenario and compare its performance to the probabilistic Bayesian approach introduced earlier.

## 6.5 Case studies

### 6.5.1 Case of an instantaneous release

In this section, we consider the problem of localising $N = 4$ land mines by randomly deploying $M = 30$ sensors in a $50 \times 50$ $m^2$ planar region according to the scenario shown in figure 5.8. Model and measurement noises are considered to be white Gaussian with an identical standard deviation equal to $0.001\mu g/m^3$. We fixed the wind velocity to $V = 5cm/s$ and the air diffusion coefficient to $K = 25m^2/s$, as in [57].

Table 6.5 shows the true values of the unknown parameters to be determined; these are the sources positions and emissions.

| $i$ | $(x_{s_i}, y_{s_i})$ | $m_i$ $[\mu g]$ |
|---|---|---|
| 1 | $(0.53, 5.69)$ | 9.78 |
| 2 | $(44.61, 30.61)$ | 8.43 |
| 3 | $(5.11, 26.88$ | 7.58 |
| 4 | $(44.23, 12.96)$ | 9.94 |

*Tab. 6.5:* True parameters for scenario $(a)$.

#### 6.5.1.1 Probabilistic Bayesian Approach

The probabilistic Bayesian approach as described in Section 6.4.1 is tested first. The slice sampling scheme was used in order to draw $N_p = 4,000$ particles/samples from the posterior distribution defined in Equation 6.7.

Figures 6.5 and 6.6 show, respectively, the variation of the log-likelihood of the samples and the evolution of the Markov chain through the iterations. The dimension of the parameter vector is 12.

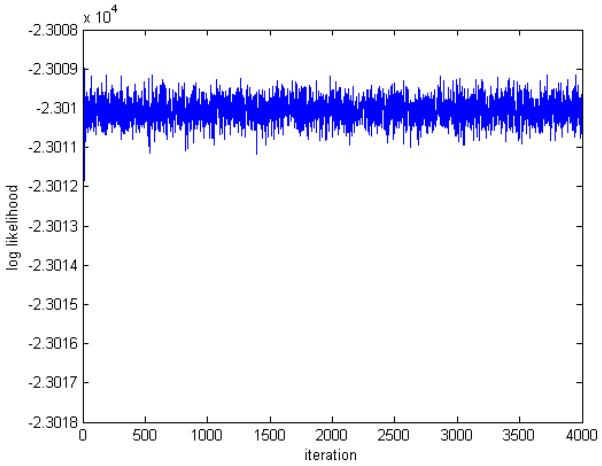



*Fig. 6.5:* Variation of the log-likelihood in the Markov chain for scenario $(a)$.

Note on the graphs of figure 6.6 that there is a transition phase (where the samples likelihood is low) before the chain converges to the posterior distribution of interest. This phase is referred to as the burn-in.

In theory, the effect of the initial values tends to zero if the Markov chain is run for an infinite amount of time. In practice, however, an infinite number of samples cannot be drawn, so it is generally assumed that only after a certain number of iterations, the chain reaches the target distribution. Thus, in order to minimise the effect of initial values on the posterior inference, an initial portion, $N_{burn}$, of a Markov chain samples is discarded and the remaining samples are used to estimate the properties of the posterior distribution. The number $N_{burn}$ of the iterations that will be discarded is called the burn-in number.

Figure 6.7 shows the normalised histograms of the samples corresponding to the different parameters. The empirical distributions of the parameters are also estimated (using kernel density estimation KDE) and shown in red on the same graphs. Note that the empirical PDFs are centred near the true values of the parameters.

Figure 6.8 illustrates the true and the estimated positions, which were determined using MCMC slice sampling algorithm in order to solve the Bayesian inference problem. The positions are estimated by computing the mean value. The simulations were carried

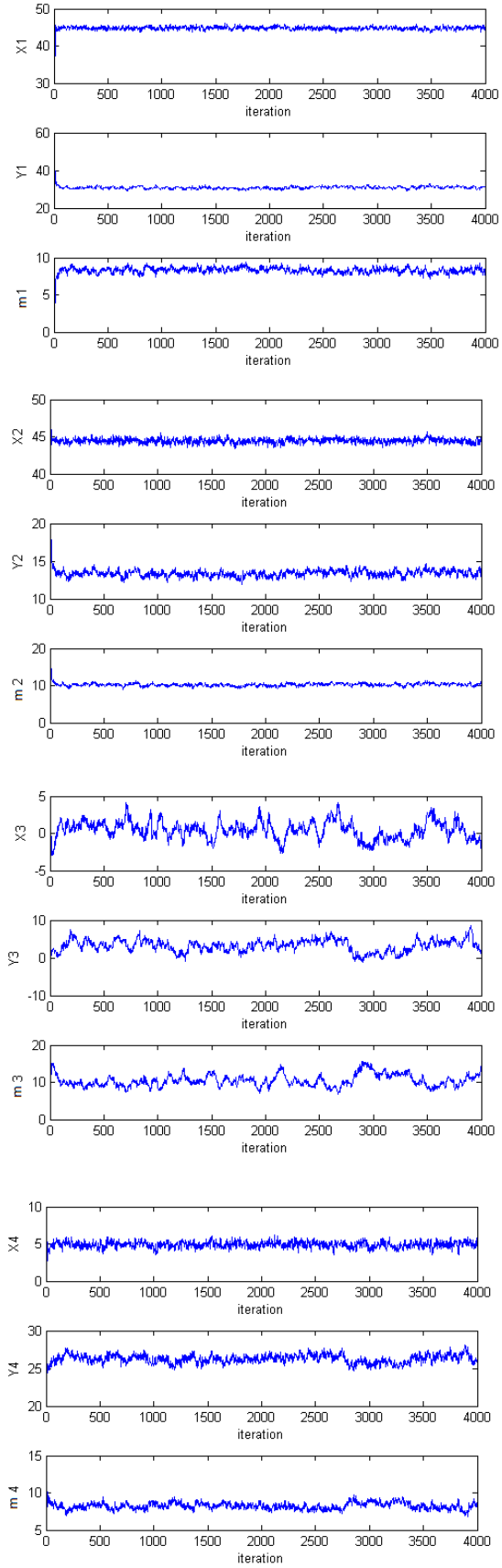



*Fig. 6.6:* Evolution of the Markov chain for scenario $(a)$.

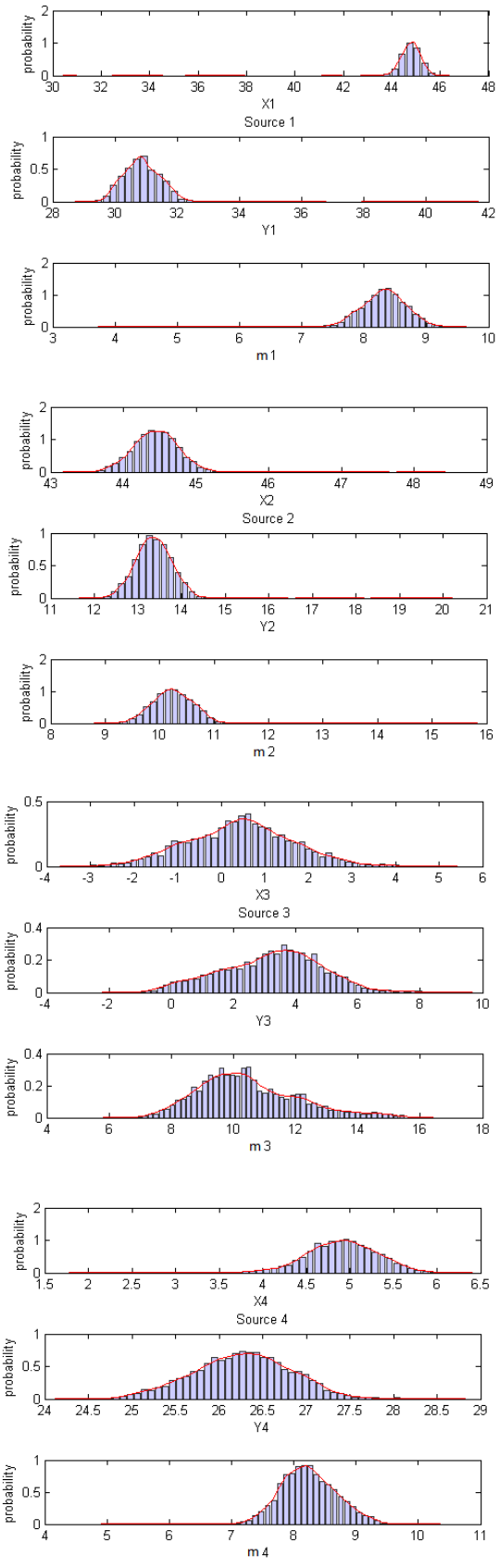



*Fig. 6.7:* Samples empirical distributions for scenario $(a)$.

out using MATLAB on an Intel Core i7-3520M processor (2.90 GHz, 4-MB Cache, Dual-core). The computational time for this approach is 244.60 s. Table 6.6 shows the estimated parameters using the slice sampling technique.

| $i$ | $(x_{s_i}, y_{s_i})$ | $m_i$ $[\mu g]$ |
|---|---|---|
| 1 | $(0.53, 3.65)$ | 10.08 |
| 2 | $(44.90, 30.82)$ | 8.33 |
| 3 | $(4.95, 26.30)$ | 8.20 |
| 4 | $(44.49, 13.29)$ | 10.22 |

*Tab. 6.6:* Estimated parameters using the slice sampling for scenario $(a)$.

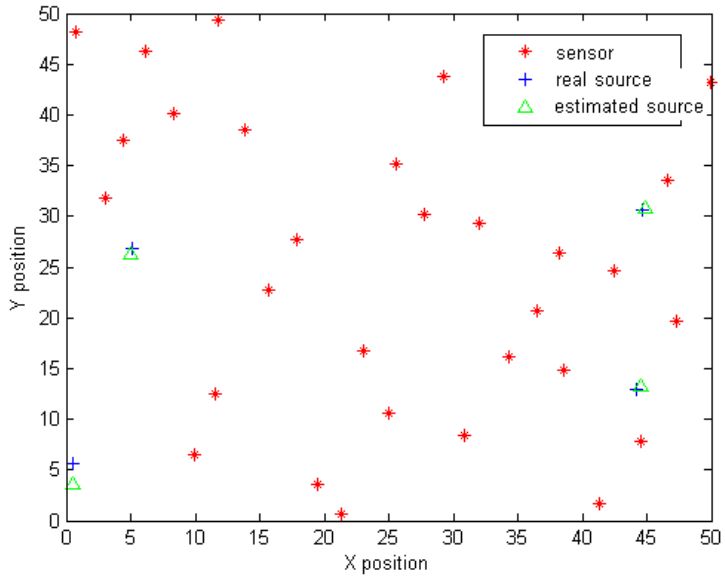



*Fig. 6.8:* Estimated positions using Bayesian inference and MCMC sampling in the case of instantaneous point sources.

### 6.5.1.2 Convergence Diagnostic

Two common critical issues when using an MCMC sampler in order to estimate the properties of a PDF are, first, how to decide when to stop sampling and use the available samples in order to estimate the characteristics of the posterior distribution of interest, and second, how to determine the number of iterations that correspond to the burn-in and should be discarded [28].

While it is difficult to predict the number of iterations $N_{iter}$ after which it is safe to stop sampling and the number $N_{burn}$ of initial samples to be discarded, diagnostic tools can be applied to the output of the MCMC samplers in order to address the convergence problem.

In order to decide if the resulting samples accurately estimate the posterior distribution of interest, we apply in this section a convergence diagnostic to the chain outputted by our sampler. We use the StatLib implementation of the Raftery and Lewis diagnostic (1992) [28]. This test requires as inputs a posterior quantile of interest $q$, an acceptable tolerance $r$ for $q$ and a probability $s$ of being within this tolerance. It outputs, amongst other parameters, the number of iterations $N_{iter}$ and burn-ins $N_{burn}$ necessary to satisfy

the specified conditions. The diagnostic was run on the resultant Markov chain for $q = 0.5$, $r = 0.01$ and $s = 0.95$, which means we want to measure the 0.5 quantile with an accuracy of 0.01. The output was a total number of iterations $N_{iter} = 3672$ to be run, of which the first $N_{burn} = 14$ samples correspond to the burn-in and should be discarded. Thus, we are 95% sure that the true quantile is within $\pm 0.01$ from the corresponding estimated value.

#### 6.5.1.3 Least Squares Technique

Next, the probabilistic Bayesian approach is compared to the generic least squares optimisation approach.

Looking back at Figure 6.3, the cost function to be minimised is convex and has one global minimum. Recall that this figure illustrates the variation of the functional $J$ given by equation (6.10) as a function of Source 3 coordinates after fixing the remaining parameters to their exact values.

Figure 6.9 illustrates the optimal land mine positions, resulting in minimising the functional $J$ given in equation (6.10). The least squares technique provides in this example an accurate estimation of the unknown parameters.

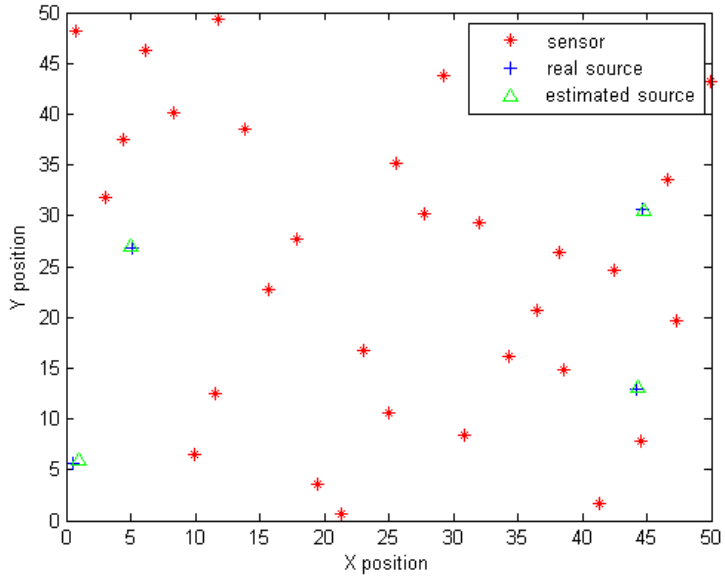



*Fig. 6.9:* Optimal solution provided by the least squares approach (scenario $(a)$).

Table 6.7 groups the optimal results obtained using the least squares technique. The computational time is 1.99 s.

| $i$ | $(x_{s_i}, y_{s_i})$ | $m_i$ $[\mu g]$ |
|---|---|---|
| 1 | $(0.95, 5.94)$ | 9.59 |
| 2 | $(44.78, 30.50)$ | 8.29 |
| 3 | $(5.04, 27.03)$ | 7.45 |
| 4 | $(44.35, 13.14)$ | 10.04 |

*Tab. 6.7:* Estimated parameters using the Least Squares search algorithm for scenario $(a)$.

Table 6.8 groups the mean squared errors on sources positions and emission rates for both probabilistic Bayesian and optimisation least squares approaches.

| Approach | Error on | |
|---|---|---|
| | **Position** | **Emission Rate** |
| Slice sampling | 1.207 | 0.140 |
| Least Squares | 0.088 | 0.020 |

*Tab. 6.8:* Mean squared errors on positions and emission rates in the case of instantaneous release sources.

Both the least squares technique and the Bayesian probabilistic approach manage to provide accurate estimates of the sources' parameters. The least squares approach however offers a better performance and requires less computational time. This was predicted since the cost function to be minimised was convex with one global minimum. Nevertheless, we will see in the next section that the performance of the least squares degrades when the cost function is no longer convex.

### *6.5.2 Case of a continuous release*

In this section, we consider the problem of localising $N = 4$ land mines, considered as continuous release point sources, by randomly deploying $M = 40$ sensors in a $20 \times 20$ $m^2$ planar region according to the scenario shown in Figure 5.9. Model and measurement noises are considered to be white Gaussian with an identical standard deviation equal to $0.001\mu g/m^3$. The wind velocity was set to $V = 5cm/s$ and the air diffusion coefficient to $K = 25m^2/s$, as in the previous scenario.

In this section, the aim is to localise the continuous release sources given concentration measurements.

Table 6.9 shows the true values of the unknown parameters to be determined; these are the sources positions and emission rates.

| $i$ | $(x_{s_i}, y_{s_i})$ | $Q_i$ $[\mu g/s]$ |
|---|---|---|
| 1 | $(6.99, 2.18)$ | 8.01 |
| 2 | $(1.73, 12.27)$ | 5.38 |
| 3 | $(16.52, 6.45)$ | 5.87 |
| 4 | $(14.11, 13.01)$ | 9.39 |

*Tab. 6.9:* True parameters.

#### *6.5.2.1 Probabilistic Bayesian Approach*

The probabilistic Bayesian approach explained in Section 6.4.1 is tested now. The slice sampling scheme is used in order to draw $N_p = 4,000$ particles/samples from the posterior distribution defined in the same section.

Figures 6.10 and 6.11 show, respectively, the variation of the log-likelihood of the samples and the evolution of the Markov chain through the iterations. In this example, the dimension of the parameter vector is 12.

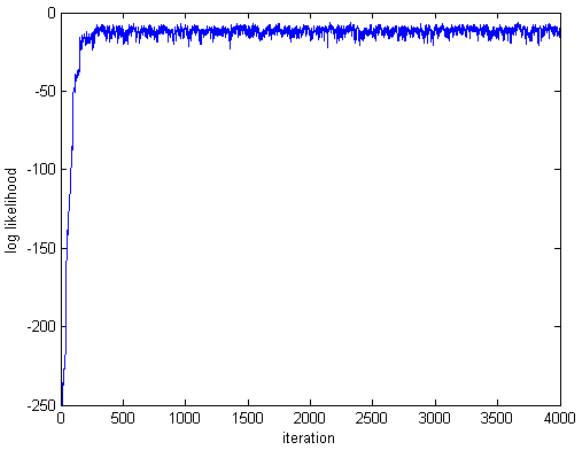



*Fig. 6.10:* Variation of the log-likelihood in the Markov chain for scenario (*b*).

Note again on the graphs of figures 6.10 and 6.11 the burn-in phase before the chain converges to the posterior distribution of interest, the initial samples likelihood is low.

Figure 6.12 shows the normalised histograms of the samples corresponding to the different parameters. The empirical PDFs of the parameters are also estimated and shown on the same graphs, they are centred near the true values of the parameters.

Figure 6.13 illustrates the true and the estimated positions, which were determined using the slice sampling scheme. The computational time for this approach is 20.35 s. Table 6.10 shows the estimated parameters for the Bayesian probabilistic technique.

| $i$ | $(x_{s_i}, y_{s_i})$ | $Q_i$ $[\mu g/s]$ |
|---|---|---|
| 1 | $(7.06, 2.15)$ | 8.02 |
| 2 | $(1.68, 12.03)$ | 5.44 |
| 3 | $(16.27, 6.27)$ | 5.11 |
| 4 | $(14.10, 13.04)$ | 9.95 |

*Tab. 6.10:* Estimated parameters using the slice sampling for scenario (*b*).

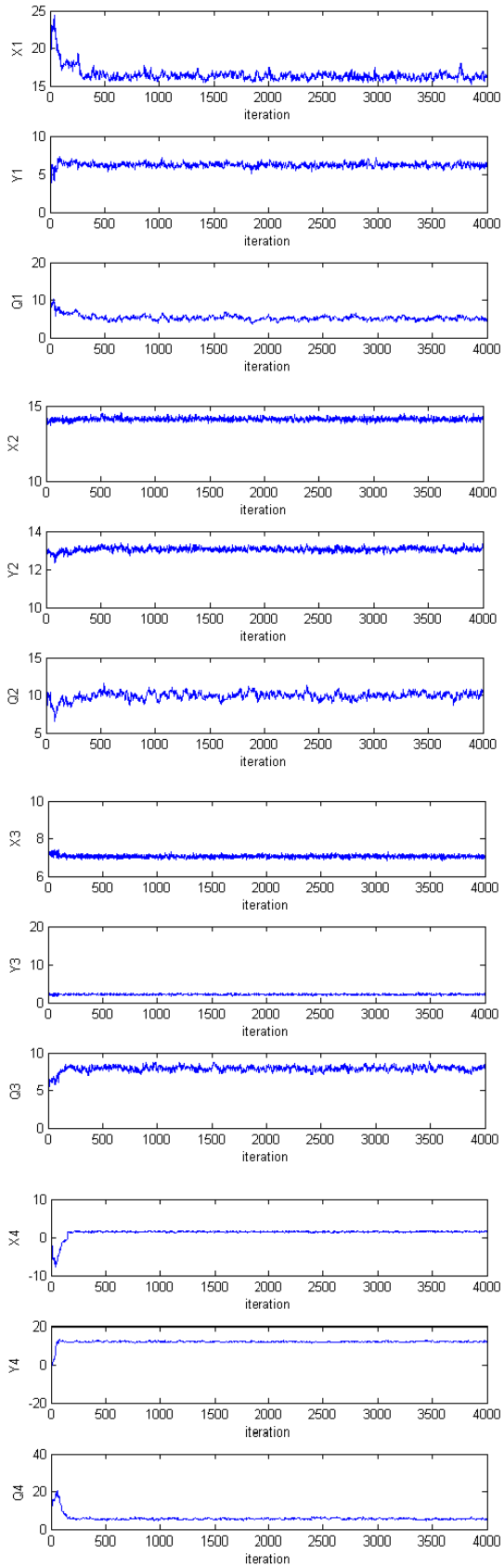



*Fig. 6.11:* Evolution of the Markov chain for scenario (*b*).

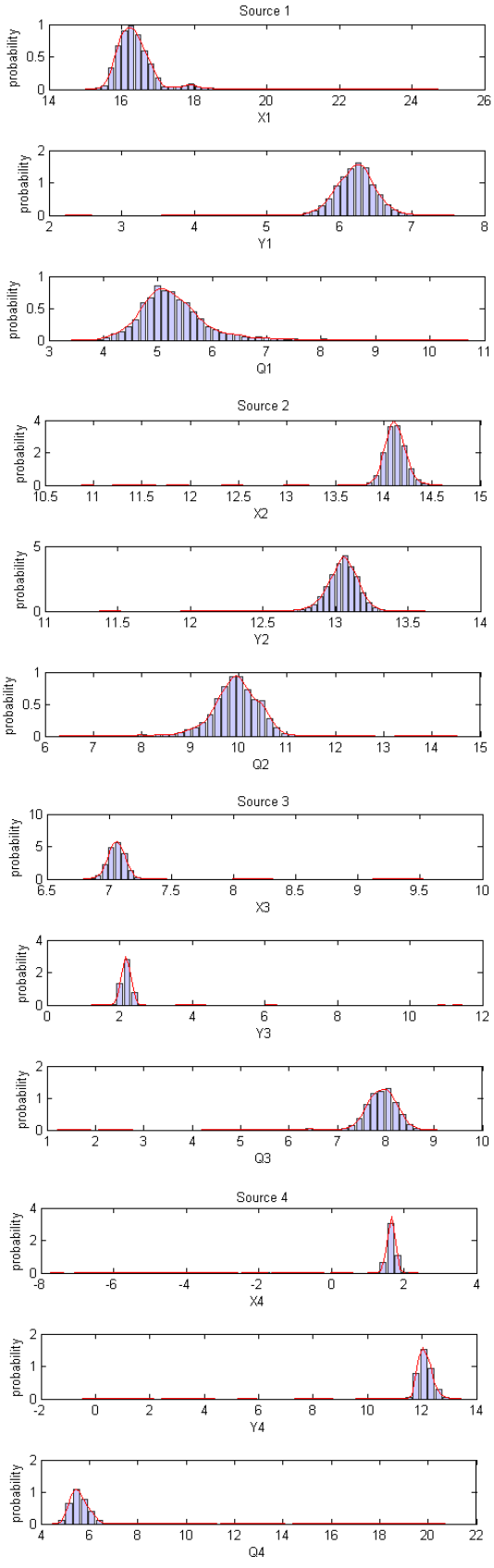



*Fig. 6.12:* Samples empirical distributions for scenario (*b*).

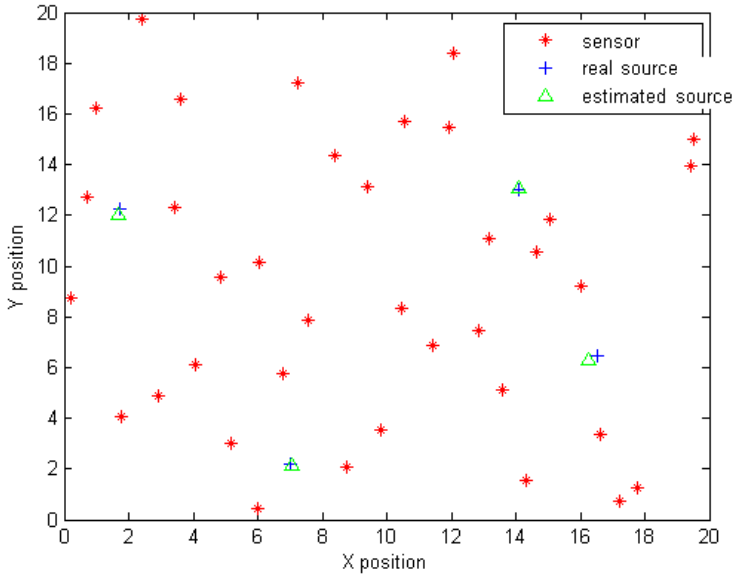



*Fig. 6.13:* Estimated positions of the continuous release sources using Bayesian inference and MCMC sampling.

#### 6.5.2.2 *Convergence Diagnostic*

In order to decide if the resulting samples accurately estimate the posterior distribution of interest, we apply a convergence diagnostic to the chain outputted by the slice sampler. Again, we use the Raftery and Lewis diagnostic.

The test was run on the resultant Markov chain for $q = 0.5$, $r = 0.01$ and $s = 0.95$, which means we want to measure the 0.5 quantile with an accuracy of 0.01. The output was a total number of iterations $N_{iter} = 2,655$ to be run, of which the first $N_{burn} = 10$ samples correspond to the burn-in and should be discarded. Thus, we are 95% sure that the true quantile is within $\pm 0.01$ from the corresponding estimated value.

#### 6.5.2.3 *Least Squares Technique*

Next, the probabilistic Bayesian approach is compared to the generic least squares optimisation approach.

Looking back at Figure 6.4, the cost function to be minimised is not convex and has multiple local minima. This figure illustrates the variation of the functional $J$ given by Equation (6.10) as a function of Source 3 coordinates after fixing the remaining parameters to their exact values. A least squares search algorithm might fall into some local minimum and, thus, diverge from the true global minimum located near Source 3's true position. Figure 6.14 shows the solution provided by the least squares technique when the vector of parameters is randomly initialised. The algorithm stopped at a local minimum, so the solution provided diverges from the true parameters.

The choice of the initial point for the search algorithm is crucial. If this start point is situated in the restrained convex region around the global minimum (see Figure 6.4), the least squares approach is likely to converge to the true solution. In order to overcome this problem, we choose the start position parameters to be the positions of the three sensors indicating the greatest concentration measurements. These sensors are likely to be the closest to the sources.

Figure 6.15 illustrates the optimal land mine positions, resulting in minimising the functional $J$ given in Equation (6.10). The start position parameters for the search algorithm are chosen to be the positions of the four sensors, indicating the maximal concentration measurements. The least squares technique provides in this case an accurate estimation of the unknown parameters.

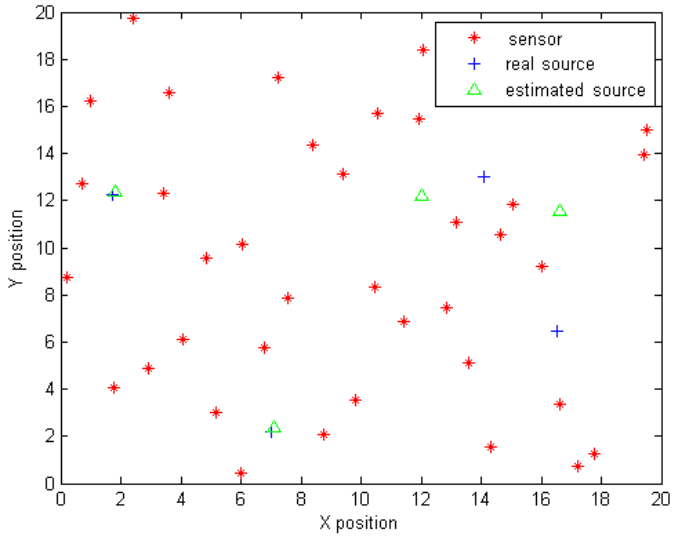



*Fig. 6.14:* Solution provided by the least squares approach for randomly chosen start parameters.

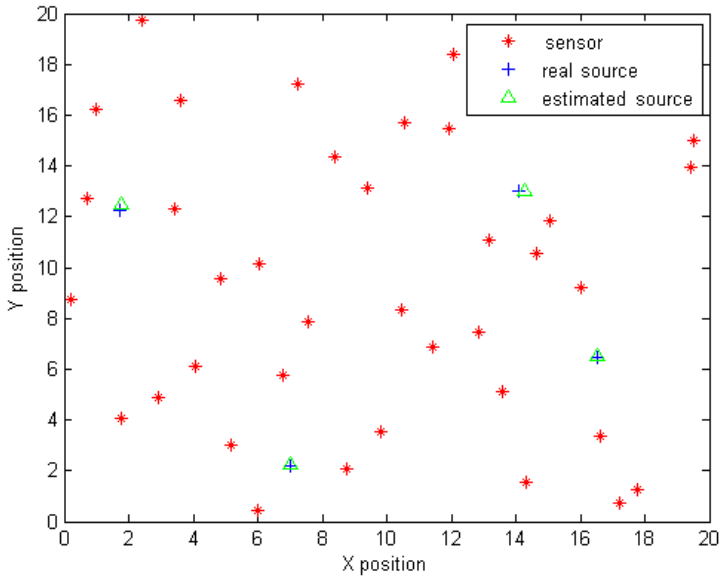



*Fig. 6.15:* Optimal solution provided by the least squares approach in the case of continuous release sources.

Table 6.11 groups the optimal results obtained using the least squares technique. The computational time is 6.34 s.

| $i$ | $(x_{s_i}, y_{s_i})$ | $Q_i$ $[\mu g/s]$ |
|---|---|---|
| 1 | $(6.99, 2.22)$ | 8.53 |
| 2 | $(1.78, 12.49)$ | 5.06 |
| 3 | $(16.54, 6.54)$ | 5.77 |
| 4 | $(14.26, 13.01)$ | 8.98 |

*Tab. 6.11:* Estimated parameters using the Least Squares search algorithm, case of continuous sources.

Table 6.12 groups the mean squared errors on sources positions and emission rates for both approaches.

| **Approach** | **Error on** | |
|---|---|---|
| | **Position** | **Emission Rate** |
| Slice sampling | 0.040 | 0.223 |
| Least Squares | 0.021 | 0.137 |

*Tab. 6.12:* Mean squared errors on positions and emission rates for scenario ($b$).

Both the least squares technique and the Bayesian probabilistic approach offer a similar performance provided an adequate choice of the start point for the optimisation search algorithm. The least squares approach also requires less computational time. However, it is important to note that, as the sensors are randomly deployed, even the ones with the greatest concentration measurements might not always be close enough to the sources, so as to find the global minimum. This optimisation technique is very sensitive to the choice of the initial point. On the other side, the probabilistic Bayesian approach together with an efficient sampling algorithm turns out to be more robust and less sensitive to the choice of the initial sample of the Markov chain.

## *6.6 Conclusion*

While previous work on land mine localisation using sensor networks solves the problem of locating a single source, this chapter considers the problem of locating several land mines. It also deals with the more difficult scenario of an unknown number of sources to be characterised. First, the PCA technique is used in order to determine the number of land mines. Second, the inverse problem consisting of locating and estimating the emission rates of the land mines is solved in a probabilistic Bayesian framework. In our simulations, we compare the results obtained using this approach with those provided by the least squares optimisation technique. Both methods localise successfully the sources and provide an accurate estimate of the emission rates of multiple land mines. The optimisation least squares offers a good performance when the functional to be minimised is convex as in the case of the instantaneous release point sources. For the scenario considering continuously emitting sources, the cost function presents several local minima. The main advantage of the probabilistic technique is that, using an efficient sampling scheme, it turned out to be less sensitive to the choice of the initial point of the chain, in contrast with the optimisation technique for which the choice of the start point of the search algorithm is crucial. The probabilistic method also makes it possible to quantify uncertainty

on the estimated positions, since a PDF of the unknown parameters is obtained, rather than a single optimal point solution.

# CONCLUSIONS AND PERSPECTIVES

In this thesis, we derived fusion algorithms for data collected from a WSN. A variety of problems were solved mainly in a Bayesian inference framework. Our work was structured in two parts.

In the first part, we dealt with the problem of inference on graphical models (describing distributed systems). We provided an overview of different classes of graphical models and introduced a message-passing algorithm which provides a distributed scheme for solving the inference problem in a Bayesian framework. This algorithm known as Belief Propagation (BP), also the sum of product algorithm, is a tool for performing exact or approximate marginalisation of joint PDFs represented by graphical structures.

However, for some applications, the problem is not always described by linear models and the uncertainty on the available information is statistically complex. These conditions make the computations of BP in its closed parametric form intractable. A first non-parametric approach known as the Non-parametric Belief Propagation (NBP) was introduced in [99] and utilised sample-based representations for probability quantities (as in particle filtering) in order to approximate BP's operations using Monte Carlo techniques. A novel BP scheme, inspired by the box-PF, was developed in this part of the dissertation. It is based on interval representations of the PDFs. This algorithm, which we called box Belief Propagation (box-BP) showed several advantages over the standard Monte Carlo based NBP when tested on both static and dynamic distributed systems. Mainly, it used a smaller number of box-particles to approximate the PDFs which leaded to simpler and faster computations, and to a more compact representation of the information to be exchanged wich in turn implies a reduction of the memory storage, bandwidth an energy consumption necessary to perform inference.

However, for all BP approaches stated earlier, we adopted a parallel update scheme in which, at each iteration, each node sends information to all of its neighbours simultaneously. As future work we can find an optimal message-passing scheme, or an optimal path for information circulation, that results in a reasonable approximation of the posterior PDF of interest with minimal communication requirements.

Perspectives to part I of the thesis also include the application of the box-BP on the localisation of multiple mobile targets, with communications between the mobile targets themselves and between the targets and stationary landmarks. In this case, the system is represented using a dynamic graphical model whose nodes are associated with the targets and the landmarks. At each time step, links are established between each target and the nodes that are within the vicinity of it and thus information can exchanged between a mobile target and its other counterparts in addition to the landmarks. This scenario extends the case we treated in chapter 4 to the problem of mobility tracking of several mobile targets using range measurements. Similarly to what was proposed in chapter 4, messages are exchanged exclusively between nodes representing the local states at one time instant $k$ (even though the graphical model associated with this system is dynamic).

Another perspective is to consider the dynamics of the robots known, thus at each time step a prediction of the future pose can be made and information will circulate

between time steps. For this scenario, the nodes of the graphical model describing the system include, in addition to the landmarks and the pose of the mobile targets at time instant $k$, the positions of the targets at time instants $k-1$ and/or $k+1$; *i.e.* the dynamic graphical model consists of nodes describing the system's states at consecutive time steps. Thus, range information is exchanged between nodes representing the local states at one time instant $k$, and message passing is also possible between nodes representing the local state of one mobile target at consecutive time steps since a model for the evolution of the targets is available.

The second part of the thesis considered the problem of characterising (*i.e.* determining the unknown parameters including the positions and emissions of) an unknown number of sources emitting a chemical/biological agent in their environment. The approach is based on deploying a number of sensors capable of estimating (up to an error level) the concentration of the emitted compound transported to its position due to advection and diffusion. Given a matrix of concentration measurements, we managed to determine the number of sources to be localised using a popular dimension reduction technique. Then the localisation problem was formulated as Bayesian inference and a MCMC sampler was used in order to estimate the properties of the PDF of interest. In this part, the application of interest is that of localisation of anti-personnel land mines. Since two scenarios can be typically associated with this application [57], the effectiveness of the proposed two-step technique was demonstrated on two distinct scenarios: the first considering instantaneous release sources, the second continuously emitting point sources. However, in our context, sources and sensors were assumed to be in the same plan.

For future work, a model considering a three-dimensional position for the sources and sensors can be employed since the land mines are buried and a third coordinate indicating the depth of the sources is necessary for a more realistic scenario. In this case, the emitted compound is transported by advection-diffusion via a two-layered medium (ground and air). The forward model that predicts the concentration of the emitted chemical at the sensors positions should thus take into consideration the existence of two different transport media with distinct diffusion coefficients.

The validation of the proposed scheme on real data sets is also important especially that using analytical models to describe the transport of the emitted chemical always necessitates fixing simplifying hypothesis (such as homogeneous and/or isotropic media, horizontal separating surface between the layers, constant wind velocity) which is not necessary compliant with the real scenario. Having a real dataset, a neural network could, for instance, be employed in order to predict concentration estimates at any position.

Furthermore, a recent article [102] presents a hierarchical model to find the ground-truth source bases using Non-negative Matrix Factorisation (NMF) [103]. Since in our approach, PCA was only used to determine the number of sources, the use of source separation techniques (amongst which are PCA and NMF) for source localisation forms an interesting subject of research. Looking back at equations (6.2) and (6.4), if we are able to reconstruct matrices $\mathcal{H}$ ($\mathbf{G}$) and $\mathcal{M}$ ($\mathbf{Q}$), having the concentration measurement at a sensor's position (due to all of the sources) it would be possible to determine the contribution of each source to the resultant measured concentration. Thus, the complex problem of localising multiple sources is reduced to several simpler tasks consisting of localising a single source.

# BIBLIOGRAPHY


[1] F. Abdallah, A. Gning, and Ph. Bonnifait. Adapting particle filter on interval data for dynamic state estimation. In *Proc. of the International Conf. on Acoustics, Speech, and Signal Processing*, volume 2, pages 1153–1156, Honolulu, HI, 2007.

[2] F. Abdallah, A. Gning, and Ph. Bonnifait. Box particle filtering for nonlinear state estimation using interval analysis. *Automatica*, 44(3):807–815, 2008.

[3] H. Abdi and L. J. Williams. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459, 2010.

[4] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

[5] D. Arbula. Distributed algorithm for anchor-free network localization using angle of arrival. In *IEEE International Symposium on Industrial Electronics 2008*, pages 792 – 797, Cambridge, 2008.

[6] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, february 2002.

[7] M. Badine and I. Mougharbel. Considerations for implementing a wireless network for landmine detection. In *IEEE 6th International Conference on Sciences of Electronic, Technologies of Information and Telecommunications (SETIT)*, pages 624–627, Sousse, Tunisia, march 2012.

[8] A. Baggio and K. Langendoen. Monte carlo localization for mobile sensor networks. *Ad Hoc Networks*, 6(5):718–733, 2008.

[9] J.M. Bahi, A. Makhoul, and A. Mostefaoui. A mobile beacon based approach for sensor network localization. In *Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 44–51, White Plains, NY, 2007.

[10] J.M. Bahi, A. Makhoul, and A. Mostefaoui. Improving lifetime and coverage through mobile beacon for high density sensor networks. In *Second International Conference on Sensor Technologies and Applications (SENSORCOMM)*, pages 335–341, Cap Esterel, 2008. IEEE.

[11] I. Ben-Gal. *Encyclopedia of Statistics in Quality and Reliability*, chapter Bayesian Networks. Wiley and Sons, 2007.

[12] C. M. Bishop. *Pattern Recognition and Machine Learning*, chapter 8: Graphical models. Springer, 2006.

[13] M. Borysiewicz, A. Wawrzynczak, and P. Kopka. Stochastic algorithm for estimation of the model's unknown parameters via Bayesian inference. In *Proceedings of the Federated Conference on Computer Science and Information Systems*, pages 501–508, Wroclaw, Poland, september 2012.

[14] C. J. F. Ter Braak. A Markov Chain Monte Carlo version of the genetic algorithm differential evolution: easy Bayesian computing for real parameter spaces. *Statistics and Computing*, 16:239–249, 2006.

[15] C. Bruschini and B. Gros. A survey of current sensor technology research for the detection of landmines. In *Proceedings of the International Workshop on Sustainable Humanitarian Demining (SusDem'97)*, volume 6, pages 18–27, Zagreb, Croatia, 1997.

[16] L. Cardona, J. Jiménez, and N. Vanegas. Landmine detection technologies to face the demining problem in antioquia. *Dyna*, 81(183):115–125, 2014.

[17] H. S. Carslaw and J. C. Jaeger. *Conduction of heat in solids*. Clarendon press, Oxford.

[18] P. Cheung-Mon-Chan. *Réseaux Bayésiens et Filtres Particulaires pour l'égalisation adaptative et le décodage conjoints*. PhD thesis, École Nationale Supérieure des Télécommunications, 2003.

[19] H. Haj Chhadé. Landmine detection and localization using a wireless sensor network. In *EYE LabSurfing Workshop*, Milan, May 2014.

[20] H. Haj Chhadé, F. Abdallah, I. Mougharbel, and H. Ghaziri. New approach for landmine localization using wireless biosensors. In *20th LAAS International Science Conference (LAAS'14)*, Beirut, march 2014.

[21] H. Haj Chhadé, F. Abdallah, I. Mougharbel, A. Gning, S. Julier, and L. Mihaylova. Localisation of an unknown number of land mines using a network of vapour detectors. *Sensors*, 14(11):21000–21022, 2014.

[22] H. Haj Chhadé, F. Abdallah, I. Mougharbel, A. Gning, L. Mihaylova, and S. Julier. Multiple land mines localization using a wireless sensor network. In *Proc. of the 17th International Conference on Information Fusion (FUSION)*, pages 1–7, Salamanca, July 2014. IEEE.

[23] H. Haj Chhadé, A. Gning, F. Abdallah, I. Mougharbel, and S. Julier. Non parametric distributed inference in sensor networks using box particles messages. *Mathematics in Computer Science*, 8(3-4):455–478, 2014.

[24] S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statician*, 49(4):327–335, 1995.

[25] S. Choi. Triangulation toolbox: Open-source algorithms and benchmarks for landmark-based localization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6440 – 6446, Hong Kong, 2014.

[26] B.W. Cook, S. Lanzisera, and K.S.J. Pister. Soc issues for rf smart dust. In *Proceedings of the IEEE*, volume 94, pages 1177 – 1196, 2006.

[27] J.A. Costa, N. Patwari, and A.O. Hero. Distributed weighted-multidimensional scaling for node localization in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 2(1):39–64, 2006.

[28] M. K. Cowles and B. P. Carlin. Markov Chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

[29] Michael A. A. Cox and Trevor F. Cox. *Springer Handbooks Comp.Statistics, Handbook of Data Visualization*, chapter Multidimensional Scaling, pages 315–347. Springer, 2008.

[30] J. Crank. *The mathematics of diffusion*. Clarendon press, Oxford.

[31] V. Deleskog, H. Habberstad, G. Hendeby, D. Lindgren, and N. Wahlstrom. Robust nls sensor localization using mds initialization. In *Proc. of the 17th International Conference on Information Fusion (FUSION)*, pages 1–7, Salamanca, 2014. IEEE.

[32] M.B. Dias. *Investigating The Viability Of Mems Vapor Sensors For Detecting Land Mines*. Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, 2000.

[33] B. Dil, S. Dulman, and P. Havinga. Range-based localization in mobile sensor networks. *Wireless Sensor Networks: Lecture Notes in Computer Science*, 3868:164–179, 2006.

[34] E. B. Ermis and V. Saligrama. Detection and localization in sensor networks using distributed fdr. In *40th Annual Conference on Information Sciences and Systems*, pages 699 – 704, Princeton, NJ, 2006.

[35] E. B. Ermis and V. Saligrama. Robust distributed detection with limited range sensors. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 1009–1012, Honolulu, HI, 2007.

[36] E.B. Ermis and V. Saligrama. Adaptive statistical sampling methods for decentralized estimation and detection of localized phenomena. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, volume 5, pages 1045–1048, Philadelphia, 2005.

[37] M. Essoloh, C. Richard, and H. Snoussi. Localisation distribuée dans les réseaux de capteurs sans fil par résolution d'un problème quadratique. In *Colloque GRETSI*, Troyes, 2007.

[38] D. L. García-González and R. Aparicio. Sensors: From biosensors to the electronic nose. *Grasas y Aceites*, 53(1):96–114, 2002.

[39] A. Gning and Ph. Bonnifait. Constraints propagation techniques on intervals for a guaranteed localization using redundant data. *Automatica*, 42(7):1167–1175, 2006.

[40] A. Gning, L. Mihaylova, and F. Abdallah. Mixture of uniform probability density functions for non linear state estimation using interval analysis. In *13th Conference on Information Fusion*, pages 1–8, Edinburgh, 2010.

[41] A. Gning, L. Mihaylova, F. Abdallah, and B. Ristic. *Integrated Tracking, Classification, and Sensor Management: Theory and Applications*, chapter Particle Filtering Combined with Interval Methods for Tracking Applications. John Wiley and Sons, 2012.

[42] A. Gning, B. Ristic, L. Mihaylova, and F. Abdallah. An introduction to box particle filtering. *IEEE Signal Process. Mag.*, 30(4):166–171, 2013.

[43] C. P. Gooneratne, S. C. Mukhopahyay, and G. Sen Gupta. A review of sensing technologies for landmine detection: Unmanned vehicle based approach. In *2nd International Conference on Autonomous Robots and Agents*, pages 401–407, Palmerston North, New Zealand, 2004.

[44] F. Gustafsson and F. Gunnarsson. Localization in sensor networks based on log range observations. In *10th International Conference on Information Fusion*, pages 1–8, Quebec, Que, 2007. IEEE.

[45] H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223–242, 2001.

[46] M. K. Habib. Mine detection and sensing technologies-new development potentials in the context of humanitarian demining. In *The 27th Annual Conference of the IEEE Industrial Electronics Society (IECON '01)*, volume 3, pages 1612–1621, Denver, CO, 2001.

[47] T. He, C. Huang, B.M. Blum, J.A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proc. of the 9th Annual International Conf. on Mobile Computing and Networking (MobiCom)*, pages 81–95, NY, 2003. ACM Press.

[48] K. Heurtefeux and F. Valois. Localisation collaborative pour réseaux de capteurs. In *Colloque Francophone sur l'Ingénierie des protocoles (CFIP)*, Les Arcs, France, 2008.

[49] P. Honeine, C. Richard, H. Snoussi, and M. Essoloh. Auto-localisation dans les réseaux de capteurs sans fil par régréssion de matrices de gram. In *Colloque GRETSI*, Dijon, 2009.

[50] L. Hu and D. Evans. Localization for mobile sensor networks. In *Tenth Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 45–57, New York, USA, 2004.

[51] A. T. Ihler. *Inference in Sensor Networks: Graphical Models and Particle Methods*. PhD thesis, Massachusetts Institue of Technology, 2005.

[52] A.T. Ihler, J.W. Fisher, R.L. Moses, and A.S. Willsky. Nonparametric belief propagation for self-localization of sensor networks. *IEEE journal on selected areas in communications*, 23(4):809–819, 2005.

[53] M. Irrazábal, S. P. Hernández-Rivera, and J. G. Briano. Modeling of TNT transport from landmines: Numerical approach. *Chemosphere*, 77:546–551, 2009.

[54] S.I. Ivashov, V.V. Razevig, A.P. Sheyko, and I.A. Vasilyev. A review of the remote sensing laboratory's techniques for humanitarian demining. In *International Conference on Requirements and Technologies for Detection, Removal and Neutralization of Landmines and UXO*, volume 1, pages 3–8, Brussels, Belgium, 2003.

[55] L. Jaulin. Nonlinear bounded-error state estimation of continuous-time systems. *Automatica*, 38(6):1079–1082, 2002.

[56] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter. *Applied Interval Analysis*. Springer-Verlag, 2001.

[57] A. Jeremić and A. Nehorai. Landmine detection and localization using chemical sensor array processing. *IEEE Transactions on signal processing*, 48(5):1295–1305, 2000.

[58] A. M. Johansen and A. Doucet. *A Tutorial on Particle Filtering and Smoothing: Fifteen years later*. The Oxford Handbook of Nonlinear Filtering, pp. 4-6, december 2009.

[59] S. Julier, J. Uhlmann, and H. Durrant-White. A new approach for filtering nonlinear systems. In *Proc. of the Amer. Control Conf.*, volume 3, pages 1628–1632, Seattle, WA, 1995.

[60] P. Kathirgamanathan, R. McKibbin, and R.I. McLachlan. Source term estimation of pollution from an instantaneous point source. *Res. Lett. Inf. Math. Sci. (R.L.I.M.S.)*, 3:59–67, 2002.

[61] W.A. Keats. *Bayesian inference for source determination in the atmospheric environment.* PhD thesis, Waterloo, Ontario, Canada, 2009.

[62] D. Koller and N. Friedman. *Probabilistic Graphical Models, Principles and Techniques*. The MIT Press.

[63] F.R. Kschischang, B.J. Frey, and H.A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498 – 519, 2001.

[64] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: a quantitative comparison. *Computer Networks: The International Journal of Computer and Telecommunications Networking - Special issue: Wireless sensor networks*, 43(4):499 – 518, 2003.

[65] K. Yk Leung, Y. Halpern, T. D Barfoot, and H. Ht Liu. The utias multi-robot cooperative localization and mapping dataset. *International Journal of Robotics Research*, 30(8):969–974, 2011.

[66] D. Li, K.D. Wong, Y. H. Hu, and A.M. Sayeed. Detection, classification, and tracking of targets. *IEEE Signal Processing Magazine*, 19(2):17–29, 2002.

[67] P. Lopez, M. Balsi, D.L. Vilarino, and D. Cabello. Antipersonnel mine detection on infrared images. In *IEEE International Symposium on Technology and Society, 2000. University as a Bridge from Technology to Society.*, pages 109–113, Rome, 2000.

[68] J. Lu, J. Gong, Q. Hao, and F. Hu. Space encoding based compressive multiple human tracking with distributed binary pyroelectric infrared sensor networks. In *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 180–185, Hamburg, Germany, 2012.

[69] E. Lushi and J.M. Stockie. An inverse Gaussian plume approach for estimating atmospheric pollutant emissions from multiple point sources. *Atmospheric Environment*, 44(8):1097–1107, 2010.

[70] A. Makhoul. *Réseaux de capteurs : localisation, couverture et fusion de données*. PhD thesis, Université de Franche-Comte, 2008.

[71] X. Mao, S. Tang, and M. Li. Multiple objects device-free passive tracking using wireless sensor networks. In *IEEE International Conference on Communications (ICC)*, pages 1–5, Kyoto, 2011.

[72] J. Matthes, L. Groll, and H. B. Keller. Optimal weighting of networked electronic noses for the source localization. In *IEEE Proceedings on Systems Communications*, pages 455–460, august 2005.

[73] L. Mihaylova, D. Angelova, D. R. Bull, and N. Canagarajah. Localization of mobile nodes in wireless networks with correlated in time measurement noise. *IEEE Transactions on Mobile Computing*, 10:44–53, 2011.

[74] L. Mihaylova, D. Angelova, and A. Zvikhachevskaya. Sequential Monte Carlo methods for localization in wireless networks. In P. Georgieva, L. Mihaylova, and L. C. Jain, editors, *Advances in Intelligent Signal Processing and Data Mining*, volume 410 of *Studies in Computational Intelligence*, pages 89–118. Springer Berlin Heidelberg, 2013.

[75] R. Min, M. Bhardwaj, S. Cho, N. Ickes, E. Shih, A. Sinha, A.Wang, and A. Chandrakasan. Energy-centric enabling tecumologies for wireless sensor networks. *IEEE Wireless Communications*, 9(4):28–39, 2002.

[76] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.

[77] F. Mourad, H. Snoussi, F. Abdallah, and C. Richard. Anchor-based localization via interval analysis for mobile ad-hoc sensor networks. *IEEE Transactions on signal processing*, 57(8):3226–3239, 2009.

[78] R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

[79] N. Noorshams and M. J. Wainwright. Belief propagation for continuous state spaces: Stochastic message-passing with quantitative guarantees. *Machine Learning Research*, 14:2799–2835, 2013.

[80] N. Noorshams and M. J. Wainwright. Stochastic belief propagation: A low-complexity alternative to the sum-product algorithm. *IEEE Transactions on Information Theory*, 59:1981–2000, 4.

[81] J. Paik, C. P. Lee, and M. A. Abidi. Image processing-based mine detection techniques: A review. *Subsurface Sensing Technologies and Applications*, 3(3):153–202, 2002.

[82] M. A. Paskin. *Exploiting Locality in Probabilistic Inference*. PhD thesis, University of california, Berkeley, 2004.

[83] S.S. Ram and V.V. Veeravalli. Localization and intensity tracking of diffusing point sources using sensor networks. In *IEEE Global Telecommunications Conference. (GLOBECOM '07)*, pages 3107–3111, Washington, DC, november 2007.

[84] M. Ridley, E. Nettleton, A. Goktogan, G. Brooker, S. Sukkarieh, and H. F. Durrant-Whyte. Decentralised ground target tracking with heterogeneous sensing nodes on multiple UAVs. *Information Processing in Sensor Networks*, 2634:545–565, 2003.

[85] B. Ristic, A. Gunatilaka, and R. Gailis. Cramer-rao bound for source estimation using a network of binary sensors. *ArXiv e-prints*, May 2014.

[86] B. Ristic, A. Gunatilaka, R. Gailis, and A. Skvortsov. Bayesian likelihood-free localisation of a biochemical source using multiple dispersion models. *ArXiv e-prints*, May 2014.

[87] L. Robledo, M. Carrasco, and D. Mery. A survey of land mine detection technology. *International Journal of Remote Sensing*, 30(9):2399–2410, 2009.

[88] P. Rong and M.L. Sichitiu. Angle of arrival localization for wireless sensor networks. In *Third annual IEEE Communications society on Sensor and Ad Hoc Communications and Networks*, number 1, pages 374–382, Reston, VA, 2006.

[89] C. Saad, A. Benslimane, and J.C. Konig. At-angle: a distributed method for localization using angles in sensor networks. In *IEEE symposium on computers and communications*, pages 1190–1195, Marrakech, 2008.

[90] S. Sanghavi, D. Malioutov, and A.S. Willsky. Belief propagation and LP relaxation for weighted matching in general graphs. *IEEE Transactions on Information Theory*, 57(4):2203–2212, 2011.

[91] A. H. Sayed, A. Tarighat, and N. Khajehnouri. Network-based wireless location: challenges faced in developing techniques for accurate wireless location information. *IEEE signal processing magazine*, 22(4):24–40, 2005.

[92] J. Schiff, E.B. Sudderth, and K. Goldberg. Nonparametric belief propagation for distributed tracking of robot networks with noisy inter-distance measurements. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1369–1376, St. Louis, MO, 2009.

[93] I. Senocak, N. W. Hengartner, M. B. Short, and W. B. Daniel. Stochastic event reconstruction of atmospheric contaminant dispersion using Bayesian inference. *Atmospheric Environment*, 42(33):7718–7727, 2008.

[94] Y. Shang, W. Rumel, Y. Zhang, and M. Fromherz. Localization from connectivity in sensor networks. *IEEE transactions on parallel and distributed systems*, 15(11):961 – 974, 2004.

[95] M. Sharan, S.K. Singh, and J.P. Issartel. Least square data assimilation for identification of the point source emissions. *Pure and Applied Geophysics*, 169:483–497, 2012.

[96] X. Sheng, Y. H. Hu, and P. Ramanathan. Distributed particle filter with gmm approximation for multiple targets localization and tracking in wireless sensor network. In *Fourth International Symposium on Information Processing in Sensor Networks (IPSN)*, pages 181–188, Los Angeles, California, USA, 2005.

[97] L. Sigal. *Continuous State Graphical Models for Object Localization, Pose Estimation and Tracking*. PhD thesis, Brown University, Providence, Rhode Island, 2008.

[98] B.W. Silverman. Density estimation for statistics and data analysis. In Chapman and Hall, editors, *Monograph on Statistics and Applied probability*, London, 1986.

[99] E. B. Sudderth, A. T. Ihler, M. Isard, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. *Commun. ACM*, 53(10):95–103, 2010.

[100] E. H. Sudderth. *Graphical Models for visual Object Recognition and Tracking*. PhD thesis, Massachusetts Institute of Technology, 2006.

[101] E.B. Sudderth, A.T. Ihler, W.T. Freeman, and A.S. Willsky. Nonparametric belief propagation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume I, pages 605–612, Madison, Wisconsin, June 2003.

[102] Q. Sun, F. Hu, and Q. Hao. Mobile target scenario recognition via low-cost pyroelectric sensing system: Toward a context-enhanced accurate identification. *IEEE Trans. Syst. Man Cybernetics: Syst.*, 44:375–384, 2014.

[103] Q. Sun, J. Lu, Y. Wu, H. Qiao, and X. Huang. Non-informative hierarchical bayesian inference for non-negative matrix factorization. *Signal Processing*, 44, 2014.

[104] A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2005.

[105] F. Thomas and L. Ros. Revisiting trilateration for robot localization. *IEEE Transactions on Robotics*, 21(1):93–101, 2005.

[106] V. Vivekanandan, BC Vancouver, and V.W.S. Wong. Ordinal mds-based localization for wireless sensor networks. In *Proc. of the IEEE 64th Vehicular Technology Conference (VTC)*, pages 1–5, Montreal, Que, 2006. IEEE.

[107] J. A. Vrugt, C.J.F. ter Braak, C.G.H. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon. Accelerating Markov Chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences and Numerical Simulation*, 10(3):271–288, 2009.

[108] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

[109] E.A. Wan and R. Van der Merwe. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium (AS-SPCC)*, pages 153–158, Lake Louise, Alta, 2000. IEEE.

[110] J. Weimer, B.H. Krogh, M.J. Small, and B. Sinopoli. An approach to leak detection using wireless sensor networks at carbon sequestration sites. *International Journal of Greenhouse Gas Control*, 9:243–253, 2012.

[111] G. Welch and G. Bishop. An introduction to the kalman filter. In *Annual Conference on Computer Graphics and Interactive Techniques*, Los Angeles, CA, USA, 2001. ACM Press, Addison-Wesley.

[112] J. S. Yedidia. Message-passing algorithms for inference and optimization. *Journal of Statistical Physics*, 145(4):860–890, 2011.

[113] E. Yee. Bayesian probabilistic approach for inverse source determination from limited and noisy chemical or biological sensor concentration measurements. In *Chemical and Biological Sensing VIII, A.W.Fountain III, Ed., of Proceedings of the SPIE*, volume 6554, page 12, 2007.

[114] E. Yee. Theory for reconstruction of an unknown number of contaminant sources using probabilistic inference. *Boundary-Layer Meteorology*, 127(3):359–394, 2008.

[115] E. Yee. Inverse dispersion for an unknown number of sources: Model selection and uncertainty analysis. *ISRN Applied Mathematics*, 2012:20 pages, 2012.

[116] E. Yee. Probability theory as logic: data assimilation for multiple source reconstruction. *Pure and Applied Geophysics*, 169:499–517, 2012.

[117] L. Ying, V. Chandrasekaran, A. Anandkumar, and A.S. Willsky. Feedback message passing for inference in gaussian graphical models. *IEEE Transactions on Signal Processing*, 60(8):4135–4150, 2012.

[118] L. Yu, Y. Huang, X. Jin, A.J. Mason, and X. Zeng. Ionic liquid thin layer EQCM explosives sensor. *Sensors and Actuators B: Chemical*, 140(2):363–370, 2009.

[119] C. Zhexu, C. Lei, M. Cetin, and A.S. Willsky. An efficient message passing algorithm for multi-target tracking. In *Proc. of the 12th International Conference on Information Fusion*, pages 826–833, Seattle, WA, 2009.