A Framework for Cooperative Localization in Ultra-Wideband Wireless Networks

by

Jaime Lien

S.B. Electrical Engineering, MIT (2005)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2007
(June 2007)

Author .....
Department of Electrical Engineering and Computer Science
May 25, 2007

Certified by .....
Moe Z. Win
Associate Professor
Thesis Supervisor

Accepted by .....
Arthur C. Smith
Chairman, Department Committee on Graduate Students

A Framework for Cooperative Localization in Ultra-Wideband Wireless Networks

by

Jaime Lien

Submitted to the Department of Electrical Engineering and Computer Science
on May 25, 2007, in partial fulfillment of the requirements for the degree of Master of Engineering in Electrical Engineering and Computer Science

Abstract

Location-aware technologies have the potential to revolutionize computing, cellular services, sensor networks, and many other commercial, military, and social applications. In wireless networks, accurate information about an agent's location can give meaning to observed data and facilitate the agent's interactions with its surroundings and neighbors. Determining the location of one or more agents, known as localization or positioning, is a fundamental challenge. Most existing localization methods rely on existing infrastructure and hence lack the flexibility and robustness necessary for large ad-hoc networks. In this thesis, we describe a framework for localization that overcomes these limitations by utilizing cooperation: the agents in the network work together to determine their individual locations. We derive a practical algorithm for cooperative localization by formulating the problem as a factor graph and applying the sum-product algorithm. Each agent uses relative positioning measurements and probabilistic location information from its neighbors to iteratively update its location estimate. We investigate the performance of this algorithm in a network of ultra-wideband (UWB) nodes, which are well-suited for localization due to their potential to measure inter-node distances with high accuracy. Realistic models of UWB ranging error, based on an extensive measurement campaign in several indoor environments, are incorporated into the localization algorithm. Using the experimental data and simulations, we quantify the benefits that cooperation brings to localization.

Thesis Supervisor: Moe Z. Win

Title: Associate Professor

Acknowledgements

I would like to gratefully acknowledge the many people who contributed to this thesis.

First and foremost, I wish to thank my advisor, Professor Moe Win, for his encouragement and guidance over the past year. From him, I have gained invaluable insight and self-confidence, not just with respect to my work, but also my future. Professor Win has been a wonderful mentor and friend. I cannot express the full extent of my gratitude for his support.

I am greatly indebted to Henk Wymeersch for his cheerful and patient guidance throughout my time with the group. Without his generous help, technical expertise, and Belgian supercluster of computers, this thesis would not have been possible. I also wish to thank Henk for the group lunches, movie events, gourmet chocolate, and unending stream of entertaining stories.

Professor Seth Teller and his students, particularly Jun-geun Park, freely shared their time and insight throughout our measurement campaign. I wholeheartedly thank them for their collaboration.

I am grateful to all the members of the W Group–Pedro, Yuan, Wes, Andrew, Ae, Wee Peng, and Tony–for their friendship and support. I have truly enjoyed getting to know each of you, both in and outside of work. Thank you for sharing your knowledge, for making the office a fun place to be, and for accompanying me on searches for free food.

To all my friends, old and new, thank you for sharing both the ups and the downs of this process with empathy and laughter.

Finally, I wish to thank my parents, Meze, Wes, and Chris, for their unwavering support in everything I do.

Contents

List of Figures

Abbreviations

AoA	angle-of-arrival
GPS	Global Positioning System
LOS	line-of-sight
NLOS	non-line-of-sight
RSS	received signal strength
SPA	sum-product algorithm
TDoA	time-difference-of-arrival
ToA	time-of-arrival
UWB	ultra-wideband

Chapter 1

Introduction

Location-aware technologies have the potential to revolutionize computing, cellular services, sensor networks, and many other commercial, military, and social applications. In particular, location awareness is a key enabling factor for wireless networks, which are comprised of multiple untethered agents. Accurate information about an agent's location gives meaning to observed data and facilitates the agent's interactions with its surroundings and neighbors. For example, new-generation cellular services provide the capability for users to determine their location relative to friends or landmarks [1]. The ability to pinpoint the origin of a cell phone call could also dramatically increase the efficiency of emergency 911 services [2]. In some cases, the function of the network necessitates that each node have an accurate estimate of its spatial coordinates within an absolute or relative map. A set of wireless sensors may be used to detect variations in temperature or barometric pressure [3] across different regions of an environment. Because such data has meaning only in relation to the site where it was collected, knowledge of the location of each sensor is essential. Other possible applications for location-awareness include search-and-rescue [4], military target tracking [5], healthcare monitoring [6],

and logistics [7].

The problem of determining the location of one or more agents, known as localization or positioning, is a fundamental challenge. Typically, only a small fraction of the nodes in the network, known as anchors, have prior knowledge about their location, as shown in Figure 1.1. Numerous solutions have been proposed in existing literature; however, most existing localization methods are ill-suited for the majority of wireless networks. The ad-hoc and often dynamic nature of wireless networks means that localization methods can rely on minimal, if any, infrastructure, human maintenance, and a priori location information. A centralized computing unit is often infeasible; hence, distributed algorithms are required. As wireless networks may contain hundreds or thousands of agents, localization methods must also be scalable. Moreover, the agents that comprise such networks are typically limited in terms of their processing and communication capabilities. Noise and environmental factors also present a challenge to robust, accurate localization.

In this thesis, we describe an algorithm for localization that addresses the challenges above through cooperation amongst agents in the network. The nodes in the network share information to determine each individual location. An algorithm for cooperative localization is derived by applying the sum-product algorithm to a factor-graph representation of the problem. Using probabilistic models of relative positioning measurements, the algorithm allows agents to exchange and incorporate location information from other agents.

The algorithm provides a framework for cooperative localization using any type of relative positioning measurements. Due to its fine time resolution, ultra-wideband (UWB) technology is particularly well-suited for such positioning measurements, especially in dense multipath or harsh environments [8]. In this work, we present the results of an extensive experimental campaign to

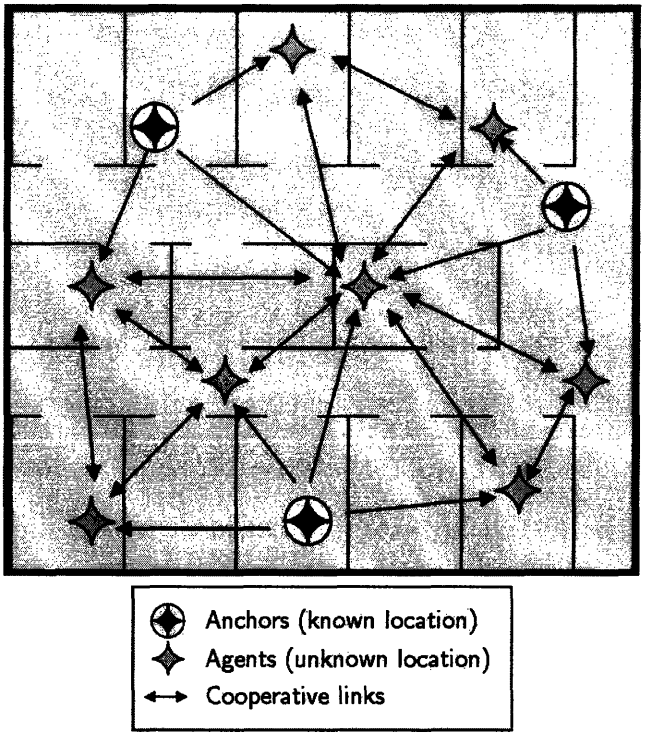


Figure 1.1: Localization for an indoor wireless network

produce models of UWB ranging measurements, which are then incorporated into the cooperative localization algorithm. Using the experimental data and simulations, the cooperative localization algorithm is shown to provide greater robustness and accuracy than traditional localization techniques.

The main contributions of the thesis are as follows:

- We derive a centralized algorithm for cooperation localization using factor graphs and the sum-product algorithm.
- After performing a graph transformation, we derive a distributed algorithm for cooperative localization with improved computational complexity.
- We discuss the incorporation of practical location information into the algorithm, the representation of messages transmitted between agents, and possible adjustments to the algorithm to reduce computation.

- We present results from an experimental campaign to develop realistic and tractable models of UWB ranging measurements.
- We quantify the benefits that cooperation brings to localization using simulations that incorporate the experimental data.

The rest of this thesis is organized as follows. In Chapter 2, we present an overview of current research in the area of localization, including existing systems and algorithms. We also discuss the use of UWB for localization. Chapter 3 explains the theoretical background and derivation of the cooperative localization algorithm. In Chapter 4, we cover practical issues related to the implementation of the algorithm. Chapter 5 focuses on the UWB measurement campaign and our resulting ranging error models. Results based on these models and simulations are presented in Chapter 6. Finally, Chapter 7 contains concluding remarks.

Chapter 2

Background and prior work

In this chapter, we present an overview of localization and ultra-wideband technology. Section 2.1 explains how localization is accomplished and introduces a set of criteria with which to classify localization methods. Localization systems, both implemented in current systems and proposed in the literature, are also discussed. Section 2.2 provides a background on ultra-wideband technology, including its traditional use as a means for communication. We then explain why ultra-wideband is well-suited for localization.

2.1 Overview of localization techniques

2.1.1 Signal metrics

Localization is accomplished using signals passed between an agent and existing infrastructure or from agent to agent. Information about the receiver's location relative to the transmitter (or vice versa) can be extracted from these signals using a variety of metrics. In the following section, we discuss time of arrival and time difference of arrival, received signal strength, angle of arrival, hop count, connectivity, and fingerprinting.

Time of arrival and time difference of arrival Time-based metrics use the measured signal propagation time to estimate the distance between a transmitter and a receiver. Assuming a constant known propagation speed s, the estimated distance is given by $\hat{d} = s \cdot \hat{t}$, where $\hat{t}$ is the measured duration of the signal propagation. If the clocks at the transmitter and receiver are synchronized, the transmitter can include a timestamp that allows the receiver to measure the signal flight time based on the time of arrival (ToA). Alternatively, the roundtrip flight time can be measured, eliminating the need for clock synchronization at the expense of increased communication. Time difference of arrival (TDoA) techniques determine the position of a transmitter relative to two receivers with known location [9]. The difference in the signal's arrival time at each receiver defines a hyperbola upon which the transmitter is located, with loci at each receiver. To calculate TDoA, the receiver clocks must be synchronized with each other but not necessarily with the transmitting agent.

Because time-based positioning measurements depend on the direct path signal from transmitter to receiver, they are subject to errors caused by multipath and non-line-of-sight (NLOS) conditions. Multipath refers to a phenomenon in which signals reflect off of surroundings and arrive at the receiver via multiple indirect paths. The superposition of these arriving paths results in fading, complicating detection of the direct path. NLOS conditions, created by physical obstructions of the direct path, may produce a number of effects. As the signal propagates through barriers, its speed decreases as a function of the composition of the obstruction (e.g. material, thickness, etc), which are typically unknown to the agent. Consequently, there is a discrepancy between the true propagation speed and the constant s used for the distance calculation. The direct path may also be attenuated or, in extreme NLOS

conditions, undetectable compared to late-arriving paths in the received signal. Such cases may cause detection algorithms to select an indirect path, producing erroneous time-based positioning measurements and, in particular, positively biased ToA-based range measurements. The accuracy of time-based signal measurements increases with the bandwidth of the signal [10].

Received signal strength The distance between a transmitter and receiver can also be estimated based on received signal strength (RSS) [11]. The relation between the received signal power P_{rec} and the propagation distance d is described by the path-loss model

$$P_{\text{rec}}(d) = P_0 - 10n_p \log \frac{d}{d_0}$$

where n_p is the path-loss exponent and P_0 is the power received at a reference distance d_0. These parameters represent the need for prior information about the channel and transmission power in order to calculate RSS-based range measurements. Because signal propagation through unknown materials other than air causes the power attenuation to deviate from the path loss model, shadowing is a major source of error for RSS measurements [7]. RSS measurements are also subject to errors due to multipath.

Angle of arrival The angle at which a signal arrives at a receiver provides information about the receiver's position relative to the transmitter [12]. Angle of arrival (AoA), also known as direction of arrival, can be measured with an array of antennae in a fixed orientation, e.g. linearly. Based on the signals arrival time at each antenna, the direction of arrival can be inferred. The need for multiple antennae makes AoA unattractive for size- or cost-constrained applications, such as unobtrusive sensors or small mobile units. Because AoA is

based on multiple ToA measurements, ToA errors due to NLOS and multipath affect AoA position estimates. Signal reflections arriving from indirect paths further complicate AoA estimation.

Connectivity If an agent is able to receive signals from a transmitter with known location, the agent's position is within the transmitter's range of communication. Information about an agent's location can thus be inferred through its connectivity to other agents or beacons [11]. If the agent is connected to multiple nodes, its possible location is constrained to the intersection of their communication areas. Hence, the greater the number of connections to an agent, the more tightly it can constrain the space of possible locations. As the counterpart to connectivity, disconnectivity may provide information about an agent's location given that it cannot receive from a known transmitter. For example, the agent may be likely to be outside of the transmitter's communication range, or in an area with obstructed line-of-sight to the transmitter.

Hop-count In a network of wireless agents, nodes can communicate only with neighbors within their range of communication. In order for a node to send information to a disconnected node, the message must be routed via a multihop path. If the average hop distance is known, then the message receiver can estimate its distance from the sender through the number of hops [13]. Hop count provides a very rough estimate of distance between sender and receiver, especially in ad-hoc networks, as it is very unlikely that the multihop path follows a straight line and contains hops of the same distance.

Fingerprinting Fingerprinting, or pattern matching, provides information about the location of a transmitter by comparing the received signal waveform or characteristics, such as ToA, AoA, and RSS, to a database [4]. For each

fixed beacon located throughout the environment, a database is compiled prior to the deployment of the system consisting of waveforms or characteristics of signals originating at known locations in the vicinity of the beacon. After the system is deployed, a beacon performs a pattern matching algorithm on the signal received from an agent. The database entry that most closely matches the received signal indicates the agent's probable location.

2.1.2 Localization algorithm classifications

The signal metrics described above provide information about the relative positions of transmitters and receivers. Localization algorithms then integrate these measurements into an estimate of the agent's location in a map or coordinate system. The following classifications facilitate an understanding of the types of localization methods.

Centralized versus distributed In centralized localization algorithms, each node in the network routes its collection of relative positioning measurements to a central processing unit. The processor can then determine the locations of all the nodes using the entire set of measurements; for example, by optimizing some cost function over the set of measurements. Moreover, the processor has information about nodes that are disconnected. However, routing information from every node to the central computer is likely to result in heavy communication traffic and power usage. Hence, centralized algorithms become impractical as the number of network nodes increases. Distributed algorithms (e.g. GPS), on the other hand, do not rely on a central unit. Instead, each node can use only the location information that it has collected or received from neighboring nodes to determine its coordinates. Consequently, nodes have no information about disconnected nodes unless it is routed to them through

neighbors. Distributed algorithms are scalable, making them attractive for large networks.

System-level versus agent-level Algorithms that provide system-level location-awareness enable users to obtain information about the locations of all the agents from one node or station. Agents may either route measurement information to the central unit where all locations are calculated, as in a centralized algorithm, or each individual node may may be calculated its location in a distributed manner and then route the estimate to the system station. Such localization systems may be useful when an administrator wishes to coordinate the actions of multiple units, e.g. in a military scenario. Alternatively, algorithms may provide agent-level location-awareness; the objective of these algorithms is for each agent to individually discover and maintains information about its own location. For example, a cell phone user may wish to know his location in an unfamiliar building, but he has no need to locate every other person in the cellular network.

Infrastructure-dependent versus ad-hoc Some localization algorithms depend on infrastructure established prior to network deployment. Existing infrastructure may include beacons fixed at known location that act as positional references to agents in the network [14]. These algorithms may be possible when location-awareness is planned for a given building; however, they are infeasible in situations where agents are deployed with little prior notice in an unknown environment. Such networks necessitate ad-hoc localization algorithms. Ad-hoc localization requires no existing infrastructure, enabling rapid and adaptable deployment of location-aware wireless networks.

Absolute versus relative Localization algorithms make produce location estimates in an absolute or a relative map. Absolute localization produces global coordinates. Relative localization provides information about where an agent is located in the context of its neighbors or local environment, such as in a specific building. Localization systems that provide relative position information may tell users their location in terms of an indoor coordinate map, or which room or area of the building they are located. If positioning measurements are made relative to nodes with known global coordinates, both relative and absolute localization is possible.

Cooperative versus non-cooperative In a non-cooperative algorithm, each agent receives location information only from reference beacons [15]. In order for all agents to obtain sufficient information for localization, non-cooperative algorithms necessitate either a high density of beacons or long-range, high-power beacon transmissions. Both options are undesirable for ad-hoc and cost- and power-constrained networks. Additionally, dependence on beacons impairs the robustness of the system; if even one beacon is compromised, several agents may be unable to localize. These limitations are addressed by algorithms that allow agents to cooperatively determine their location. In cooperative algorithms, agents communicate both with beacons and with other agents, exchanging information about their relative positions. Inter-agent communication removes the need for all agents to be within range of one or more beacons. Moreover, each agent has access to more relative measurements and hence information about its position, increasing the potential accuracy and robustness of the system.

2.1.3 Existing localization systems and algorithms

A wide variety of localization techniques are available with existing technology or have been proposed in literature.

Global coordinates can be calculated using terrestrial or satellite-based systems. The Global Positioning System (GPS) consists of 24 satellites that transmit radio signals containing information about their position to receivers on the earth. The receiver then estimates its distance from the transmitting satellite using the time of arrival of the signal. Knowing its distance from at least three GPS satellites, the receiver can calculate its global coordinates through trilateration with an accuracy of meters [16]. Because the receiver clock is not synchronized with the satellites, a fourth signal corrects for the timing offset. GPS-based systems work well only when the user is in line-of-sight of at least three satellites; hence, such systems are impractical in dense or harsh environments, such as in urban environments, under forest canopies, and indoors. The 3G communication system provides localization capability for cellular phones using their radio-frequency communication signals. Base stations can either determine the absolute coordinates of mobile stations in the vicinity or provide reference points from which mobile stations can determine their own location. Coarse location information in the 3G system is accomplished by identifying the cell in which the mobile station is located, i.e. through proximity to base stations of known location. For finer position estimates, the 3G system is also able to use TDoA and assisted GPS [17].

Current indoor positioning techniques provide relative localization in GPS-denied areas for applications such as ubiquitous computing [18] and logistics. Such systems include Cricket [19], which calculates the distance from a user to a beacon using the time-of-flight of transmitted radio-frequency and ultrasonic signals. RADAR [20] provides indoor location information by perform-

ing trilateration based on radio-frequency RSS. Radio-frequency identification (RFID) allows the localization of tagged items based on their proximity to known RF signal receivers.

Existing localization systems, including those described above, are infrastructure-dependent and non-cooperative. The dependence on infrastructure precludes the use of such systems for ad-hoc wireless networks, which may be deployed in an unknown environment with no prior notice. To address these weaknesses, recent literature has focused increasingly on ad-hoc, cooperative localization algorithms [7]. In contrast to traditional localization algorithms, which have relied on the trilateration of every node using beacons, ad-hoc cooperative algorithms take advantage of network connections by using inter-agent relative measurements to improve their location estimates. We consider here only distributed algorithms, as the nature of ad-hoc networks often prohibits a centralized computation unit and extensive routing of information.

In the two-step localization algorithm Hop-TERRAIN/Refinement [13], nodes first estimate their distance from beacons based on the number of hops. Beginning with this rough position estimate, nodes then use range measurements from their neighbors to iteratively refine their location estimate. The refinement step performs trilateration by finding the least-squares solution to a set of linearized equations. The Ad-Hoc Localization System (AHLoS) [21] is an iterative algorithm in which nodes within range of three beacons determine their position through time-based trilateration. Agents that have determined their location then act as beacons for neighboring nodes. Location-awareness thus propagates throughout the network in an ad-hoc, distributed fashion. Several variants of the Ad-Hoc Positioning System (APS) have been proposed to iteratively estimate the location of each node in the network using AoA [12] or distance-vector routing [22]. In the algorithm Cooperative Localization

with Optimum of Estimate (CLOQ) [23], agents use statistical information about the ranging error to select the three best anchors for trilateration. Each localized agent then acts as an anchor for the next iteration of the algorithm. Statistical information about ranging measurements is also utilitized in [24] to determine the maximum likelihood estimate of each agent's position given the observed measurements. In [6], multidimensional scaling is used to localize sections of the map, which are then aligned using anchors are references. Kernal methods are utilized to statistically classify the RSS measurements received by each agent in [25]. These classifications define areas relative to other nodes in which the agents are likely to be located. Probabilistic information about each node's position relative to its location of deployment enables agents to cooperatively infer their locations without the use of anchors in [26]. In the field of robotics, Bayesian formalizations of localization [27] have led to methods in which agents localize themselves with respect to others utilizing Kalman filters [28] or particle filters [29].

The cooperative algorithms described above have shown promising results for localization in ad-hoc networks. However, a theoretical framework for cooperative localization is still needed. The framework should provide a distributed algorithm for integrating all prior knowledge and relative measurements into location information. A theoretical derivation for cooperative localization would enable both mathematical analysis, such as convergence issues and bounds on accuracy, as well as practical application. The authors in [30] describe a specialized framework for localization based on nonparametric belief propagation (NBP), a message-passing algorithm for performing inference on a graphical problem. Each node maintains a potential representing its location belief. Particle-based messages passed between nodes allow the receiver to update its location belief using range measurements relative to the sender. The

nodes iteratively estimate their coordinates as messages propagate throughout the network.

2.2 Ultra-wideband technology

Ultra-wideband (UWB) signals have a number of characteristics that make them an attractive candidate for communication systems in general and particularly for localization. UWB signals are defined by having a bandwidth that is 20% of the center frequency or greater than 500 MHz. Because the power is spread over a large bandwidth, UWB communication systems have low probability of detection, efficient power consumption, and minimal interference to other systems [31]. Moreover, the wide bandwidth can easily include low frequencies that enable superior signal penetration through obstacles, leading to robust communications in dense environments.

UWB is particularly attractive for localization because the fine time resolution of UWB signals means that multipath components can be easily resolved [32]. The immunity of UWB signals to multipath [33] enables highly accurate time-based range measurements even in dense and cluttered environments [8]. Algorithms to accurately estimate ToA of UWB signals range from correlation techniques to super-resolution techniques that have improved performance in NLOS conditions [34]. The distance between transmitter and receiver is then calculated using the speed of light as the signal propagation speed. Moreover, NLOS conditions can be identified [35, 36] and the errors mitigated in UWB range estimation using prior information [24, 37]. Because the accuracy of ToA (and hence AoA) and TDoA measurements scales with the bandwidth of the signal, UWB localization systems benefit more from these metrics than RSS, connectivity, and other observed signal characteristics [10].

UWB signals have the added advantage of simultaneously accomplishing both communication and accurate ranging. Nodes that exchange information can thus derive information about their relative positions without the need for further transmissions, increasing efficiency and reducing power usage in the network. Because of these characteristics, UWB has attracted increasing attention as a framework for location-awareness, as evidenced by the recent development of UWB-specific localization algorithms [15, 38, 39].

Chapter 3

Algorithm derivation

In this chapter, we present the theoretical foundations of the cooperative localization algorithm. Localization is mathematically formalized as an inference problem in Section 3.1. The problem can be graphically represented and solved using factor graph theory and the sum-product algorithm, as explained in Section 3.2. We can therefore derive an algorithm for cooperative localization by mapping the physical network to a factor graph. Applying the sum-product algorithm to this factor graph in Section 3.3 results in a centralized localization algorithm. In Section 3.4, we transform the factor graph in order to derive a distributed version of the cooperative localization algorithm.

3.1 Mathematical problem statement

Consider a network of N nodes, labeled $1, 2, \ldots N$, existing in a D-dimensional environment with a predetermined coordinate system. The location of each node i is described by a D-dimensional vector of coordinates, denoted by x_i. Each location x_i is associated with an a priori probability distribution $p(x_i)$.

Within the network, nodes are able to communicate with each other by transmitting packets of information. We denote the set of nodes from which

node i can receive and decode transmissions by Γ_i. Note that the communication link may not be omni-directional; hence, $j \in \Gamma_i$ does not imply $i \in \Gamma_j$. Using packets received from $j \in \Gamma_i$, node i is able to make measurements of signal metrics, which contain information about its position relative to node j. These measurements, represented by the vector $z_{j \to i}$, may include time of arrival, angle of arrival, or other metrics described in Section 2.1.1. We denote the set of all measurements made in the network by $\mathcal{Z} = \left\{ \{z_{j \to i}\}_{\forall j \in \Gamma_i} \right\}_{i=1,\ldots,N}$.

Our objective is to derive a distributed algorithm that enables each node i to determine its a posteriori location distribution $p(x_i \mid \mathcal{Z})$. These distributions can then be used to estimate the nodes' coordinates and provide any other pertinent location information (e.g. confidence in these estimates). In developing the localization algorithm, we need to answer the following questions:

- How should we incorporate the a priori information and observations?
- Which nodes should cooperate? What information needs to be transmitted between cooperating nodes?
- How can cooperation be carried out in a power-efficient manner?

Our approach revolves around a representation of the problem as a factor graph. Factor graphs provide a framework for calculating the marginals of a function in an efficient and distributed manner. In the next section, we present an overview of factor graphs and the algorithm to compute marginal functions, as well as the reasoning behind this approach. These details will provide the foundation for the rest of the chapter, when we will use factor graphs to derive the desired localization algorithm.

3.2 Factor graphs and the sum-product algorithm

3.2.1 Factor graphs

Factors graphs provide an intuitive way to represent and understand multivariable functions. A factor graph expresses a *global function* as the product of factors, or *local functions* [40]. By illustrating which factors depend on which variables, a factor graph shows how the variables of the global function are interdependent through shared local functions.

Consider a global function $F(\cdot)$ that can be expressed as the product of local functions $f_j(\cdot)$:

$$F(x_1, \ldots, x_N) = \prod_{j \in J} f_j(X_j)$$

where J is a set of indices, X_j is a subset of $\{x_1, \ldots, x_N\}$, and $f_j(X_j)$ is a function of the elements of X_j. The factor graph of $F(\cdot)$ is a bipartite graph containing a *factor vertex* for each factor $f_j(\cdot)$ and a *variable vertex* for each variable x_i. Factor vertex f_j and variable vertex x_i are connected by an (undirected) edge if and only if $f_j(\cdot)$ is a function of x_i, i.e. $x_i \in X_j$. Hence, the set of vertices adjacent to a variable vertex x_i, denoted by $\eta(x_i)$, contains all factors taking x_i as an argument, and all the variables in set $\eta(f_j) = X_j$ are interdependent according to the function $f_j(\cdot)$. The factor graph thus illustrates the relation between all the variables of $F(\cdot)$ via shared local functions.

Factor graphs are particularly useful in understanding inference problems, where the global function $F(\cdot)$ represents some joint probability distribution of several random variables. Connections in the resulting factor graph encode the interdependency of these random variables.

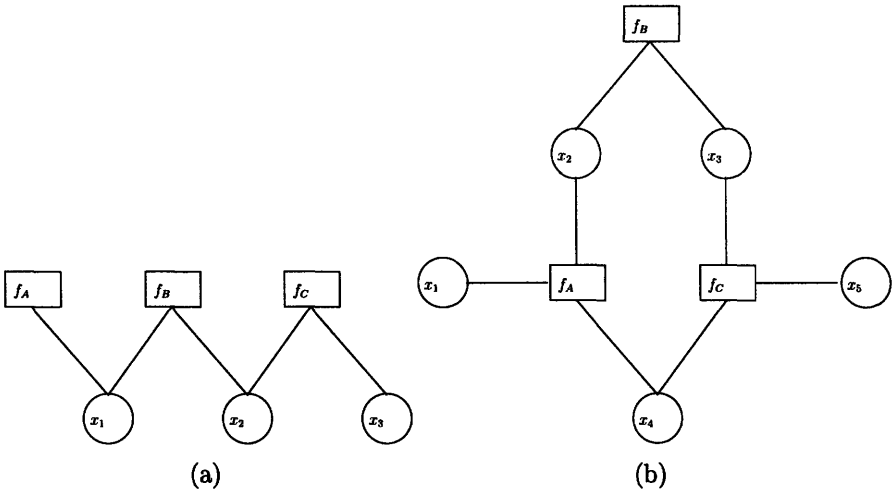


Figure 3.1: Factor graph examples

Figure 3.1 shows some examples of simple factor graphs. The graph in (a) represents the function $F(x_1, x_2, x_3) = f_A(x_1) \cdot f_B(x_1, x_2) \cdot f_C(x_2, x_3)$. The factor graph for $F(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_4) \cdot f_B(x_2, x_3) \cdot f(x_3, x_4, x_5)$ is shown in (b).

3.2.2 The sum-product algorithm

The factorization of the global function $F(\cdot)$, expressed in the factor graph, facilitates computation of the N marginal functions of $F(\cdot)$, denoted by $g_i(x_i)$ for $i = 1, \ldots, N$. Each marginal function is given by

$$g_i(x_i) = \sum_{\sim\{x_i\}} F(x_1, \ldots, x_N)$$

where the notation $\sim \{x_i\}$ indicates that the summation is performed over every variable except x_i[1]. When $F(\cdot)$ is a joint probability distribution, $g_i(x_i)$ is the distribution of the individual random variable x_i.

The marginal functions can be calculated by performing the *sum-product*

[1]When considering continuous functions, summations are replaced by integrations.

algorithm (SPA), also known in some contexts as belief propagation, on the factor graph. The SPA applies the distributive law to the factorization of $F(\cdot)$, enabling parts of the marginal function to be computed locally. These local pieces of information are then transmitted to other parts of the factor graph via messages passed along edges of the graph. When message-passing terminates, each variable vertex x_i has received the information necessary to calculate $g_i(x_i)$. The SPA thus provides an efficient and distributed way to calculate all marginal functions of $F(\cdot)$ simultaneously.

Given a cycle-free factor graph, the SPA defines the messages that should be transmitted along each edge the graph. Two types of messages are described: those transmitted from a variable vertex to a factor vertex, denoted by $\mu_{x\to f}(\cdot)$, and those transmitted from a factor vertex to a variable vertex, denoted by $\mu_{f\to x}(\cdot)$. Each message is a function of the associated variable x. The algorithm initiates at the leaves of the graph, where the message from the terminal vertex to an adjacent vertex is given by

$$\mu_{x_i\to f_m}(x_i) = 1$$

$$\mu_{f_n\to x_j}(x_j) = f_n(x_j)$$

for variable vertices and factor vertices respectively. Non-terminal vertices then use the following *update rule* to calculate the outgoing message along an edge based on messages incoming along the other incident edges:

$$\mu_{x_i\to f_m}(x_i) = \prod_{f_n\in\eta(x_i)\setminus f_m} \mu_{f_n\to x_i}(x_i) \tag{3.1}$$

$$\mu_{f_n\to x_j}(x_j) = \sum_{\sim\{x_j\}} f_n(X_n)\cdot \prod_{x_k\in X_n\setminus x_j} \mu_{x_k\to f_n}(x_k) \tag{3.2}$$

Message-passing terminates when a message has been sent in both directions

along every edge of the graph. Each marginal function $g_i(x_i)$ is then equal to the product of the ingoing and outgoing messages passed along any edge incident to variable vertex x_i:

$$g_i(x_i) = \mu_{x_i \to f_m}(x_i) \cdot \mu_{f_m \to x_i}(x_i), \ f_m \in \eta(x_i)$$

Figure 3.2 illustrates the application of the SPA on the factor graph for $F(x_1, x_2, x_3) = f_A(x_1) \cdot f_B(x_1, x_2) \cdot f_C(x_2, x_3)$. The numbered arrows in (a) indicate the order in which messages propagate. Two example messages, labeled in (b), are given by

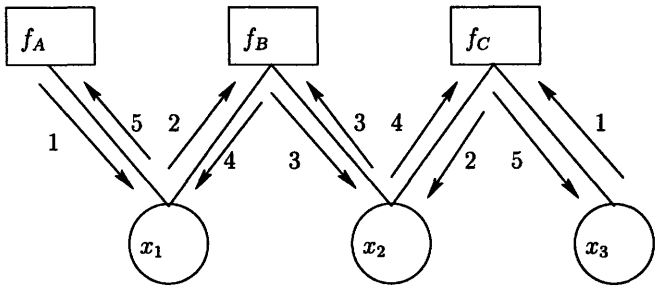
$$\mu_{f_B \to x_2}(x_2) = \sum_{x_1} f_B(x_1, x_2) \mu_{x_1 \to f_B}(x_1)$$

$$\mu_{x_2 \to f_B}(x_2) = \mu_{f_C \to x_2}(x_2)$$

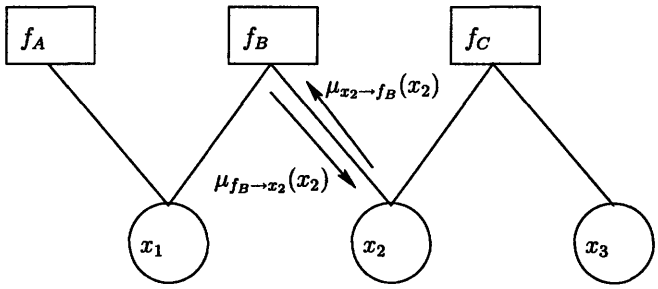
After all messages have been passed, the marginal function of x_2 can be calculated as

$$g_x(x_2) = \mu_{f_B \to x_2}(x_2) \cdot \mu_{x_2 \to f_B}(x_2) = \mu_{f_C \to x_2}(x_2) \cdot \mu_{x_2 \to f_C}(x_2)$$

Scaling the global function and messages in the SPA has no effect on the algorithm except to scale the resulting marginal functions. Consequently, when the SPA is used to perform inference (e.g. when calculating marginal distributions from a joint distribution), it is often desirable to normalize the messages and interpret them as probability distributions. We shall discuss message interpretation in much greater detail in Section 4.3, when we apply factor graphs and the SPA to the localization problem.



(a) Propagation of SPA messages



(b) Example messages

Figure 3.2: Application of the SPA

3.2.3 Cyclic factor graphs

The SPA is proven to produce the exact marginal functions on any factor graph without cycles. When a factor graph does contain cycles, the initialization and termination steps described above are no longer valid. Instead, the SPA can be adapted into an iterative algorithm, also known as loopy belief propagation. To initialize the algorithm, the messages incoming along certain edges are set to unity. The algorithm then proceeds with messages computed according to the same update rules 3.1 and 3.2. Due to the structure of the graph, message-passing is cyclic and thus iterative. After initialization, each vertex sends an outgoing message. When these messages are received, they update the outgoing messages, trigger a new round of messages that replace the previous ones. We will represent the iterations by adding a time superscript to the message notation: $\mu^t_{x_i \to f_m}(x_i)$ and $\mu^t_{f_n \to x_j}(x_j)$. Because no natural termination occurs, the cyclic SPA iterates some stopping criterion is met. The marginal

function at each variable vertex is then given by one of the following:

- the product of the incoming and outgoing messages along an incident edge
- the product of all outgoing messages from that vertex

The two expressions above can be shown to be equal if the outgoing messages reflect the most recent incoming messages.

The results of the SPA on a cyclic factor graph are not guaranteed to be exact marginal functions. In fact, convergence of the loopy algorithm is not well-understood and remains an open area of research [41]. Extensive simulations have shown, however, that the adapted SPA can achieve good performance despite cycles in the graph [42].

3.2.4 Factor graph transformations

A number of transformations can be made to a factor graph without changing the represented global function. Below, we describe a few transformations that will be utilized later in the localization algorithm derivation.

- Any factor vertex representing a constant function can be eliminated from the graph.
- Any set M of factor vertices can be merged, forming a single vertex whose function f_M is the product of the merged factors:

$$f_M(X_M) = \prod_{m \in M} f_m(X_m)$$

  where X_M is the union of the arguments of the factors. Each edge incident on an original factor vertex is replaced with an edge incident on the merged vertex.

- Similarly, a factor vertex can be split into multiple vertices whose product equals the original function.

The vertex-merging and vertex-splitting transformations can be used to remove or introduce cycles in a factor graph. Consequently, such transformations may result in different marginal functions and affect the convergence of the SPA, even though the global function remains the same.

3.2.5 Factor graphs for localization

The objective of our work, as described in Section 3.1, is to determine the posterior distribution $p(x_i \mid \mathcal{Z})$ for the location of every node $i = 1, \ldots, N$. As these distributions are the marginal functions of the joint probability distribution $p(x_1, \ldots, x_N \mid \mathcal{Z})$, factor graphs and the SPA are a natural choice to approach the problem. Visualizing the joint distribution as a factor graph provides insight about how the measurements $\mathcal{Z}$ create interdependency between unknown locations x_i, as well as how a priori information affects the desired marginal distributions. Moreover, as we shall see in the following section, the physical network topology can be clearly mapped onto the factor graph. Consequently, we can interpret the SPA messages as information that nodes should exchange in order to cooperatively determine their locations. We will use the theoretical SPA messages to derive the content of physical messages transmitted by agents in the network. These physical transmissions form the basis of a cooperative localization algorithm that calculates all the desired location distributions simultaneously.

3.3 Centralized algorithm derivation

We first consider a centralized algorithm for cooperative localization, intended to be executed on a central processing unit. In this section, we derive the centralized localization algorithm using a factor graph of the joint posterior distribution $p(x_1, \ldots, x_N \mid \mathcal{Z})$. By applying the SPA to this factor graph, we obtain an algorithm that computes the marginal distributions $p(x_i \mid \mathcal{Z})$ for $i = 1, \ldots, N$.

3.3.1 Available information

In order to execute the centralized algorithm, the central processing unit first collects all available location information and measurements from the network. The processor therefore has access to the following information:

- the identities of nodes $j \in \Gamma_i$ from which each node i received transmissions
- the identities of nodes $k \notin \Gamma_i$ from which each node i did not receive transmissions
- all measurements $\mathcal{Z}$
- all a priori distributions $p(x_i)$, $i = 1, \ldots N$

3.3.2 Factor graph

To develop the desired factor graph, we begin with a factorization of the joint distribution $p(x_1, \ldots, x_N \mid \mathcal{Z})$. The factorization follows from the assumptions below:

1. The locations of all nodes are a priori mutually independent:

$$p(x_1, \ldots, x_N) = \prod_{i=1}^{N} p(x_i)$$

2. Conditioned on the locations of the nodes, all measurements are mutually independent:

$$p(\mathcal{Z} \mid x_1, \ldots, x_N) = \prod_{i=1}^{N} \left\{ \prod_{j \in \Gamma_i} p(z_{j \to i} \mid x_1, \ldots, x_N) \prod_{k \notin \Gamma_i} f_D(x_i, x_k) \right\}$$

where $f_D(x_i, x_k)$ is a function constraining the possible locations of nodes i and k given that node i cannot receive transmissions from k.

3. Conditioned on the locations of the nodes, the relative measurement $z_{j \to i}$ depends only on the receiver's and transmitter's locations, x_i and x_j respectively:

$$p(z_{j \to i} \mid x_1, \ldots, x_N) = p(z_{j \to i} \mid x_i, x_j)$$

Using these assumptions, we can now factorize the joint distribution as follows. According to Bayes' Rule,

$$p(x_1, \ldots, x_N \mid \mathcal{Z}) = \frac{p(x_1, \ldots, x_N) \cdot p(\mathcal{Z} \mid x_1, \ldots, x_N)}{p(\mathcal{Z})}$$

Because $p(\mathcal{Z})$ is a constant normalizing factor that does not depend on the locations $x_1, \ldots, x_N$, it can be ignored without affecting the SPA.

$$p(x_1, \ldots, x_N \mid \mathcal{Z}) \propto p(x_1, \ldots, x_N) \cdot p(\mathcal{Z} \mid x_1, \ldots, x_N)$$

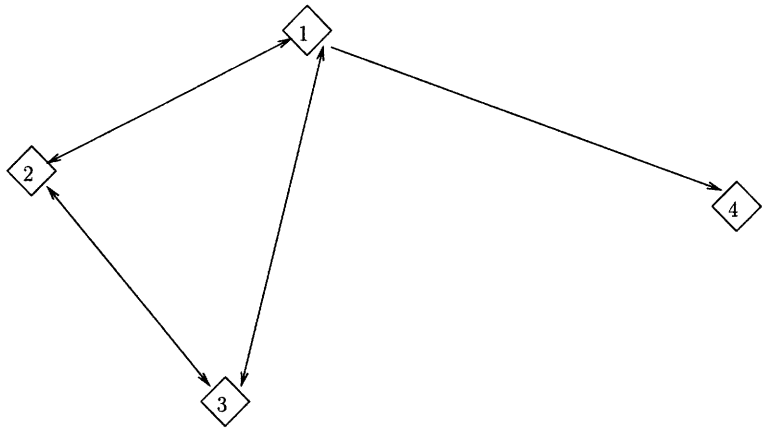
Applying assumptions 1-3 in order,

$$\begin{aligned} p(x_1,\ldots,x_N \mid \mathcal{Z}) &\propto \prod_{i=1}^{N} p(x_i)\cdot p(\mathcal{Z} \mid x_1,\ldots,x_N) \\ &\propto \prod_{i=1}^{N}\left(p(x_i)\prod_{j\in\Gamma_i} p(z_{j\to i} \mid x_1,\ldots,x_N)\prod_{k\notin\Gamma_i} f_D(x_i,x_k)\right) \\ &\propto \prod_{i=1}^{N}\left(p(x_i)\prod_{j\in\Gamma_i} p(z_{j\to i} \mid x_i,x_j)\prod_{k\notin\Gamma_i} f_D(x_i,x_k)\right) \end{aligned} \tag{3.3}$$

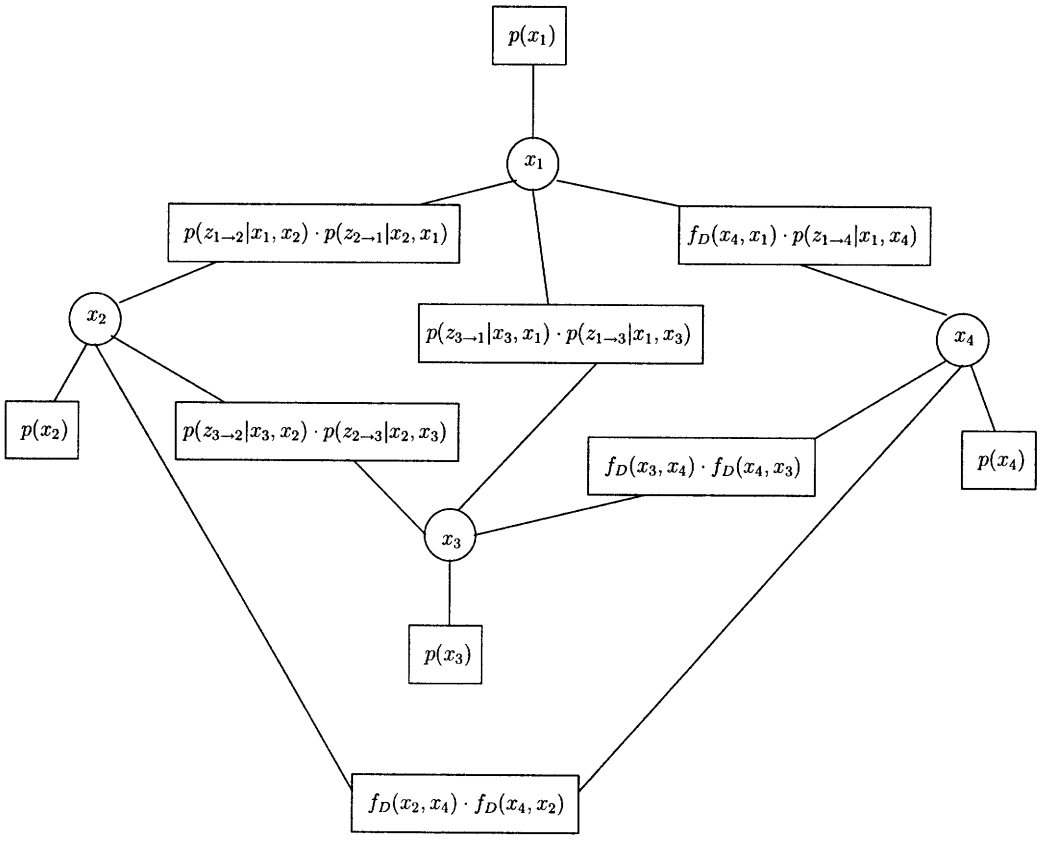
The joint distribution (and hence the corresponding factor graph) is highly dependent on the physical network topology that it represents. To illustrate, Figure 3.3(a) depicts a simple example network with nodes labeled 1-4[2]. The communication links, depicted by arrows, are omnidirectional except between nodes 1 and 4 (the latter can receive from the former, but not vice versa). The factor graph for the corresponding joint distribution $p(x_1,x_2,x_3,x_4 \mid \mathcal{Z})$ is shown in Figure 3.3(b). Each location x_i and measurement $z_{j\to i}$ is represented by a variable vertex. The a priori distribution of x_i is represented by the adjacent factor vertex $p(x_i)$. Variable vertices $x_i$and x_j and measurements $z_{j\to i}$ and $z_{i\to j}$ (if they exist) share a mutually adjacent factor vertex $h_{i,j}(x_i,x_j)$ that has one of the following forms, depending on the physical network topology:

- If nodes i and j can both receive communication from the other, then $h_{i,j}(x_i,x_j) = p(z_{i\to j} \mid x_i,x_j)p(z_{j\to i} \mid x_j,x_i)$.
- If node i is able to receive communication from node j but not vice versa, then $h_{i,j}(x_i,x_j) = p(z_{j\to i} \mid x_j,x_i)f_D(x_j,x_i)$.
- If nodes i and j cannot communicate at all, then $h_{i,j}(x_i,x_j) = f_D(x_i,x_j)f_D(x_j,x_i)$.

[2]Continuing previous terminology, we will use the term node to refer to a component of the physical network and the term vertex to refer to a component of the factor graph.



(a) Physical network



(b) Corresponding factor graph

Figure 3.3: An example network with corresponding factor graph

3.3.3 Resulting algorithm

Due to cycles in the graph, the SPA is initialized by setting all the incoming messages to unity. The update rule then gives the following theoretical messages:

$$\mu^t_{x_i \to h_{i,j}}(x_i) = p(x_i) \cdot \prod_{k/j} \mu^{t-1}_{h_{i,k} \to x_i}(x_i)$$

$$\mu^t_{h_{i,j} \to x_j}(x_j) = \sum_{x_i} h_{i,j}(x_i, x_j) \mu^t_{x_i \to h_{i,j}}(x_i)$$

Note that for each location variable x_i, a unique message must be sent to every other location variable x_j, $j \neq i$. Messages are iteratively calculated, sent, and updated until termination criteria is met. After message-passing terminates at time T, the marginal functions $p(x_i \mid \mathcal{Z})$ are given by the product

$$p(x_i \mid \mathcal{Z}) = \mu^T_{x_i \to h_{i,j}}(x_i) \cdot \mu^T_{h_{i,j} \to x_i}(x_i)$$

for any $j \neq i$.

A total of $2N^2 N_{\text{it}}$ messages are calculated in the centralized localization algorithm, where N is the number of nodes and N_{it} is the number of iterations. The order of complexity of the algorithm is dependent on the message representation, as discussed in Section 4.3.

3.4 Distributed algorithm derivation

In many practical situations, a centralized localization algorithm is infeasible or undesirable, due to the need for a centralized processing unit and extensive routing of information. In this section, we develop a distributed localization algorithm, in which computation is performed by individual network nodes.

We transform the factor graph from Section 3.3.2 to reflect the information available to each node. The theoretical SPA messages are adapted into feasible physical messages to be transmitted between nodes, producing a distributed algorithm for localization.

3.4.1 Available information

In the distributed algorithm, computation is performed by the individual network nodes instead of a signal processing unit. Unlike a centralized processor, each node has access to only limited information, which must be taken into account when developing the algorithm. The following information is available to node i:

- the identities of nodes $j \in \Gamma_i$ from which it can receive transmissions
- the measurements it has made from received signals: $z_{j\to i}$ for $j \in \Gamma_i$
- its a prior distribution $p(x_i)$

Notice the key differences between the information available to each node and the information available to a centralized processor in Section 3.3.1. First, node i does not have any information about disconnected nodes $k \notin \Gamma_i$[3]. It additionally does not know measurements $z_{i\to j}$ made from its own transmissions, nor any measurements in which it is neither the transmitter or receiver.

3.4.2 Factor graph

Regardless of how of the joint distribution $p(x_1, \ldots, x_N \mid \mathcal{Z})$ is marginalized, the factorization remains as in Equation 3.3. We will therefore transform

[3]We assume for now that information is not forwarded across multiple hops in the physical network.

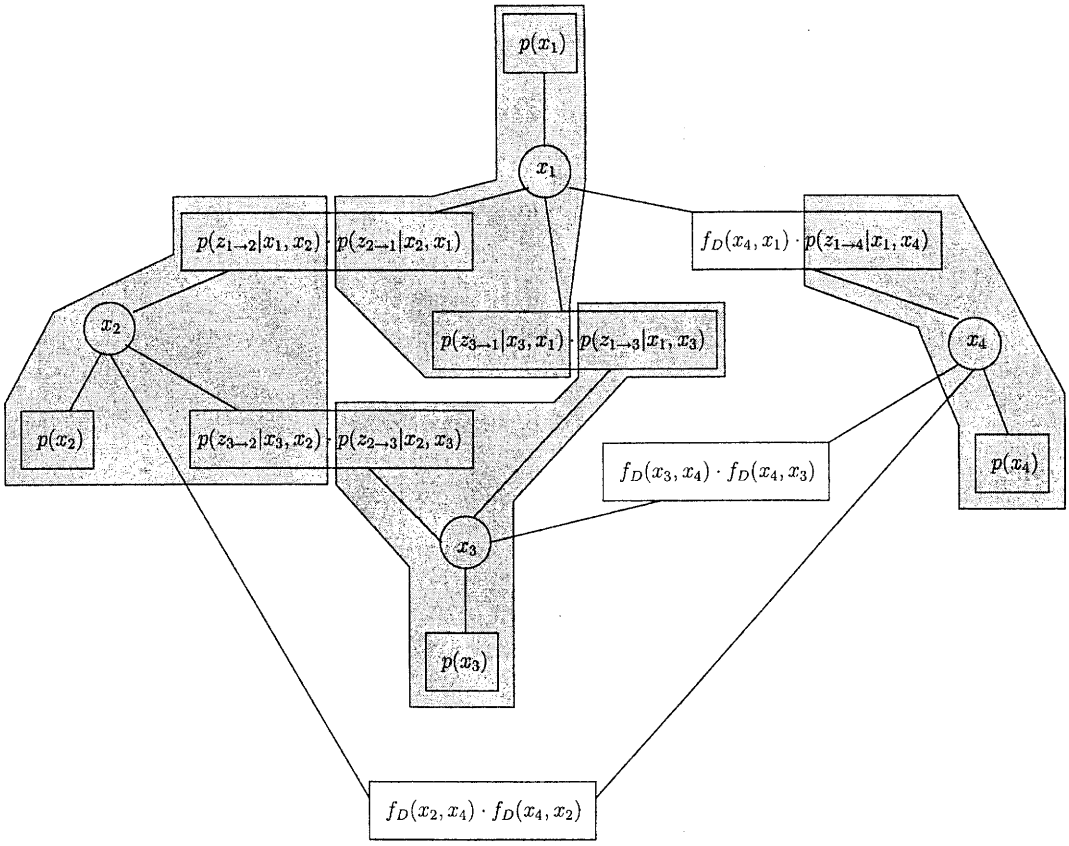
the factor graph in Figure 3.3(b), keeping the global function constant but accounting for the following:

- Each node has a limited set of available information.

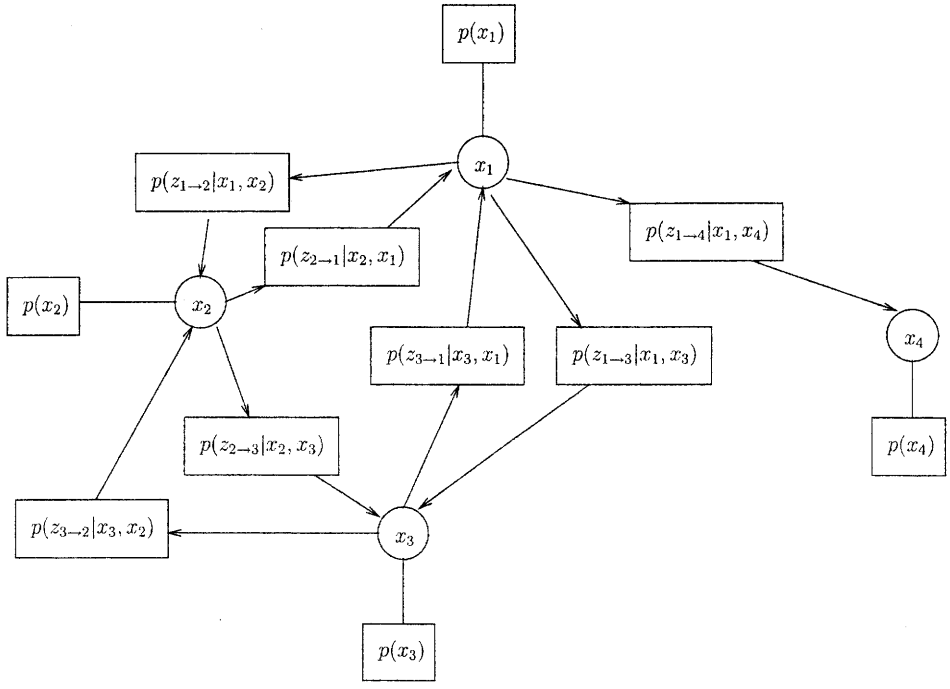
- Cooperation takes the form of information exchanged between physical nodes. Hence, the *theoretical* SPA messages that are passed between vertices of the factor graph must be adapted into *physical* messages that can feasibly be transmitted as packets between nodes in the network.

Figure 3.4(a) depicts the same factor graph as Figure 3.3(b), with the information available to each node represented by the shaded areas. The vertices and edges of the shaded area corresponding to node i will be termed the *domain* of node i. Note that factor vertex $h_{ij}(x_i, x_j)$ depends on both $z_{j\to i}$, which is available to x_i, and $z_{j\to i}$, which is available to x_i. Computation involving this vertex is thus shared between two nodes. In order to distribute computation and reduce the amount of internode communication, we need to associate each vertex with a single node. Consequently, we apply the vertex-splitting transformation, separating $h_{i,j}(x_i, x_j) = p(z_{i\to j} \mid x_i, x_j)p(z_{j\to i} \mid x_j, x_i)$ into two vertices, $p(z_{i\to j} \mid x_i, x_j)$ and $p(z_{j\to i} \mid x_j, x_i)$. The former is now available only to node j and the latter only to node i. Additionally, because nodes are unaware of disconnected nodes, we can consider the factors $f_D(\cdot, \cdot)$ to contain no information. Hence, these factor vertices can be removed from the graph.

The transformed factor graph is shown in Figure 3.4(b). The flow of information in the physical network is represented by directed edges between the nodes.



(a) Domains of nodes



(b) Transformed factor graph

Figure 3.4: Factor graph for distributed algorithm

3.4.3 Resulting algorithm

Applying the SPA to the transformed factor graph produces theoretical messages μ passed between vertices of the graph. These theoretical messages will be used to derive physical messages, which we shall denote by m, that are passed between nodes in the network.

The SPA messages of the factor graph in Figure 3.4(b) are as follows:

$$\mu^t_{x_i \to p_{i\to j}}(x_i) = p(x_i) \cdot \prod_{k \in \Gamma_i} \mu^{t-1}_{p_{k\to i} \to x_i}(x_i) \tag{3.4}$$

$$\mu^t_{p_{i\to j} \to x_j}(x_j) = \sum_{x_i} p(z_{i\to j} \mid x_i, x_j) \mu^t_{x_i \to p_{i\to j}}(x_i) \tag{3.5}$$

We first discuss message 3.4. The message $\mu^t_{x_i \to p_{i\to j}}(x_i)$ represents information computed within the domain of node i and passed to the domain of node j. We can thus interpret this theoretical message as a physical message that should be transmitted from node i to j at time t:

$$m^t_{i\to j}(x_i) = \mu^t_{x_i \to p_{i\to j}}(x_i)$$

By normalizing the outgoing message, we can interpret it as the distribution $p(x_i \mid \mathcal{Z})$ at time t given all the information obtained up to time $t-1$. Notice that the message does not depend on j. In fact, the transmitted message from node i to any connected node is identical. Node i can therefore *broadcast* this message instead of sending a unique outgoing message to each receiver. To emphasize this fact, we denote the broadcast message as

$$m^t_i(x_i) = m^t_{i\to j}(x_i)$$

Accordingly, the amount of necessary computation is greatly reduced compared

to the centralized algorithm, in which each node must calculate a distinct message for each other node in the network.

SPA message 3.5, $\mu^t_{p_{i\to j}\to x_j}(x_j)$, uses information available only to node j and remains entirely within the domain of j. Message computation is done internal to node j using the received physical message $m_i(x_i)$, according to

$$\mu^t_{p_{i\to j}\to x_j}(x_j) = \sum_{x_i} p(z_{i\to j} \mid x_i, x_j) m^t_i(x_i)$$

The internal message can be interpreted as information about location x_j that is obtained from the distribution $m^t_i(x_i)$ of its neighbor and the corresponding received measurement $z_{i\to j}$. To emphasize the origin of the received information and to simply notation, we will denote the internal message as

$$\mu^t_{x_i\to x_j}(x_j) = \mu^t_{p_{i\to j}\to x_j}(x_j)$$

To account for cycles in the factor graph, the SPA is initialized by setting all theoretical incoming messages $\mu_{x_k\to x_i}(x_i)$ equal to unity. Hence, the distributed localization algorithm begins at time $t = 0$ with each node i broadcasting its a priori distribution $p(x_i)$. Node i then listens for any messages broadcast by its neighbors, receiving $m^t_j(x_j)$ for all $j \in \Gamma_i$. For each received message $m^t_j(x_j)$, node i calculates the internal message

$$\mu^t_{x_i\to x_j}(x_j) = \sum_{x_i} p(z_{i\to j} \mid x_i, x_j) m^t_i(x_i)$$

The new outgoing message is then given by the product of all internal messages with the a priori distribution:

$$m^{t+1}_i(x_i) = p(x_i) \cdot \prod_{k\in\Gamma_i} \mu^t_{x_k\to x_i}(x_i)$$

The algorithm iterates until termination conditions are met at time T. The marginal posterior distribution of each node i is then given by the most recent outgoing message,

$$p(x_i \mid \mathcal{Z}) = m_i^T(x_i)$$

In the distributed algorithm, a total of $N(|\Gamma|_{\text{avg}} + 1)N_{\text{it}}$ messages are computed, where $|\Gamma|_{\text{avg}}$ is the average number of neighbors for each node. The number of messages is notably reduced from the centralized algorithm by a factor $\frac{|\Gamma|_{\text{avg}}+1}{2N}$, which is much less than $\frac{1}{2}$ for typical large ad-hoc networks, because incoming messages are received only from connected nodes and outgoing messages are broadcast. The distributed localization algorithm thus increases computational efficiency while decreasing dependence on infrastructure.

Chapter 4

Algorithm implementation

In this chapter, we consider how the algorithm is implemented. We describe the practical information contained in the a priori information and measurement models of the algorithm in Sections 4.1-4.2 respectively. In Section 4.3, we examine how messages should be represented in order to be transmitted between nodes. Finally, we discuss how the algorithm terminates in Section 4.4, and how to use the resulting marginal functions to produce estimates of the nodes' coordinates in Section 4.5.

4.1 A priori information

Before localization, each node i uses any available location information about its location x_i to form the a priori probability distribution function $p(x_i)$. For anchors, such as GPS-enabled nodes or nodes that have been placed in determined positions by a system administrator, $p(x_i)$ can be described as a Dirac delta function shifted to the known coordinates. Agents, which have no prior information about their location, may be represented with a uniform $p(x_i)$ over the entire map. The available prior location knowledge may also contain more complicated information; for example, if nodes have a map of the

building layout, $p(x_i)$ may convey the fact that the node cannot be in a wall or a restricted area. In wireless sensor networks, a node may use its collected data to infer information about its location; for example, if a weather-sensing node detects heavy rainfall, it may shape $p(x_i)$ to reflect a greater probability of being in a location that is known to receive more rain.

Because any uniform function may be removed from a factor graph, as described in Section 3.2.4, the a priori distribution of agents may be ignored. Practically, this means nodes with no a priori location information are *censored*; they do not transmit outgoing messages until they have received nontrivial location information from neighbors. As a result, message-passing propagates from nodes with prior information, i.e. anchors, outward to agents many hops away. This strategy greatly reduces the amount of traffic, especially in the first few iterations of cooperation.

4.2 Measurements and measurement models

Node i is able to obtain relative measurements $\{z_{j\to i}\}_{\forall j\in\Gamma_i}$ either prior to or during localization, possibly using the message broadcast by node j. These measurements relate the location x_i to location x_j through the probabilistic model $p(z_{i\to j} \mid x_i, x_j)$. The value of the distribution $p(z_{i\to j} \mid x_i, x_j)$ conveys the likelihood of variable arguments x_i and x_j, given the observed measurement $z_{i\to j}$. Good knowledge of $p(z_{i\to j} \mid x_i, x_j)$ improves the performance of the algorithm.

Below, we discuss the incorporation of measurements and measurement models appropriate for UWB signals.

Ranging

As mentioned in Section 2.2, UWB transmissions have the potential to provide accurate and high resolution range measurements. Therefore, for the remainder of this thesis, we will consider $z_{j\to i}$ to contain a measurement $\hat{d}$ of the distance $d = \|x_j - x_i\|$ from j to i. The corresponding distribution, $p(\hat{d} \mid d)$, describes the likelihood that the true distance between nodes i and j is d, given that the measured distance is $\hat{d}$. If the ranging measurements were known to be exact, the distribution $p(\hat{d} \mid d)$ would be a Dirac distribution $\delta(d - \hat{d})$. Realistically, though, there is some uncertainty about the accuracy of the measured range. To develop realistic distributions $p(\hat{d} \mid d)$ for UWB radio nodes, we conducted an extensive measurement campaign, presented in Chapter 5.

NLOS identification

Information about whether a signal was LOS or NLOS may affect the distribution $p(z_{i\to j} \mid x_i, x_j)$. For example, the range measured in a NLOS scenario will include a positive bias that is not present in the LOS scenario [43]. The identification of NLOS conditions [35, 36] can determine which distribution $p(\hat{d} \mid d)$ to use, as discussed further in Chapter 5. Previous work has shown that NLOS identification can mitigate errors in UWB range estimation for localization [24, 37].

Connectivity

If node i is able to receive communication from node j, it may be able to constrain its location to a certain area (for example, within the radius of communication of node j). Disconnectivity may provide less information. A node i may be unable to receive from another node k if a physical blockage

completely obstructs the signal path, even if node k is within communication range.

4.3 Message representation

In the localization algorithm, nodes convey information to each other by broadcasting messages. The messages, which can be interpreted as probability distributions, must be represented in a manner that can be transmitted as packets by physical nodes. The message representation determines the communication and computational costs of the localization algorithm. In this chapter, we discuss two possibilities for representing the physical messages: probability mass functions (pmfs) and particle representations.

4.3.1 Probability mass functions

To represent a continuous distribution over an environment, we can discretize the environment and simply represent the values of the distribution at each discrete point. The resulting discrete function is called a probability mass function. The effectiveness and computational complexity of utilizing a pmf representation is directly related to the resolution of discretization. Because the messages are of dimension D^2, the complexity increases exponentially with the resolution along each dimension. This is a problem for fine-grained localization and highly accurate ranging over a large map. The finite number of discrete points implies a finite number of distances on which to evaluate $p(\hat{d} \mid d)$. If the resolution of discretization is too coarse compared to the resolution of the range estimates $\hat{d}$, then the discrete ranging model distribution may fail to represent the true, continuous distribution. Moreover, the final location estimate will have a degree of uncertainty associated with the resolution

of the map.

4.3.2 Particle representation

In this section, we consider representing the physical message distributions with a finite set of samples. The general use of particle filters for localization is discussed in [27]. Our discussion pertains to a two-dimensional environment in which ranging measurements are used, although the particle techniques can be extended to higher dimensions and other signal metrics.

Sampling methods

A set of samples $\left\{x^{(r)}\right\}_{r=1,\ldots,R}$ with associated weights $\left\{w^{(r)}\right\}_{r=1,\ldots,R}$ is said to represent a continuous distribution $p(x)$ if

$$\sum_{r=1}^{R} w^{(r)} = 1$$

and

$$\sum_{r=1}^{R} w^{(r)} f(x^{(r)}) \approx \int f(x)p(x)\, dx$$

for any integrable function $f(x)$. There are a number of methods to produce such samples from a distribution $p(x)$. In *uniform sampling*, R independent samples are drawn directly from $p(x)$, each with equal weight $\frac{1}{R}$. This technique works well for distributions such as the uniform and Gaussian, but it is often hard to directly sample an arbitrary target distribution $p(x)$ [44]. *Importance sampling* addresses these difficulties by using a distribution $q(x)$ that is easy to sample from and that is nonzero everywhere that $p(x)$ is nonzero [45]. Equal-weight samples $\left\{x^{(r)}\right\}_{r=1,\ldots,R}$ are drawn from the sampler distribution

$q(x)$. Each sample $x^{(r)}$ is then assigned a weight $w^{(r)}$ according to [46]

$$w^{(r)} \propto \frac{p(x^{(r)})}{q(x^{(r)})}$$

Hence, samples that are more likely to be produced by $q(x)$ than by $p(x)$ are given a lower weight, and vice versa. The closer the sampler distribution $q(x)$ is to the target distribution $p(x)$, the more accurate the sample representation. After all weights have been calculated, they are normalized by their sum.

Messages

To implement particle representations, we must determine how to represent the broadcast message[1]

$$m_i^t(x_i) = \prod_{k \in \Gamma_i} \mu_{x_k \to x_i}^{t-1}(x_i)$$

as a set of samples and weights $\left\{x_i^{(r)}, w^{(r)}\right\}_{r=1,\ldots,R_{\text{ext}}}$. In addition, we need a method to calculate samples $\left\{x_j^{(r)}, w^{(r)}\right\}_{r=1,\ldots,R_{\text{int}}}$ of the internal message

$$\mu_{x_i \to x_j}^t(x_j) = \sum_{x_i} p(\hat{d} \mid d = \|x_i - x_j\|) m_i^t(x_i)$$

using the received samples of $m_i^t(x_i)$. Note that the number of samples representing the broadcast message, R_{ext}, may differ from the number R_{int} used for the internal message calculation, allowing the algorithm's communication cost to be tuned independently of the computational cost.

To represent the broadcast message $m_i^t(x_i)$, we use importance sampling. The target density $m_i^t(x_i)$ is the product of $|\Gamma_i|$ distributions, the internal messages $\mu_{x_k \to x_i}^{t-1}(x_i)$ for $k \in \Gamma_i$. Each distribution k is represented by samples

[1] We remove the a priori distribution $p(x_i)$, assuming it is uniform. However, it is straightforward to include a non-uniform distribution in this discussion.

and weights $\left\{x_{i,k}^{(r)}, w_k^{(r)}\right\}_{r=1,\ldots,R_{\text{int}}}$. For the sampler distribution $q(x_i)$, we use the sum of the distributions,

$$q(x_i) = \sum_{k \in \Gamma_i} \mu_{x_k \to x_i}^{t-1}(x_i)$$

To sample $q(x_i)$, we draw $\frac{\alpha R_{\text{int}}}{|\Gamma_i|}$ from each distribution $\mu_{x_k \to x_i}^{t-1}(x_i)$, producing samples $\left\{x_i^{(r)}\right\}_{r=1,\ldots,\alpha R_{\text{int}}}$, where $\alpha \geq 1$ and can be tuned for complexity (similar to [30]). The weights of the samples are then given by

$$w_i^{(r)} \propto \frac{m_i(x_i^{(r)})}{q(x_i^{(r)})} = \frac{\prod_{k \in \Gamma_i} \mu_{x_k \to x_i}(x_i^{(r)})}{\sum_{k \in \Gamma_i} \mu_{x_k \to x_i}(x_i^{(r)})}$$

In order to evaluate $\mu_{x_k \to x_i}(x_i^{(r)})$, we convert the sample representation of $\mu_{x_k \to x_i}(x_i)$ into a smooth distribution,

$$\mu_{x_k \to x_i}(x_i) \approx \sum_{r=1}^{R_{\text{int}}} w_k^{(r)} \mathcal{N}(x_{i,k}^{(r)}, \Sigma)$$

where $\mathcal{N}(x, \Sigma)$ is a Gaussian with mean x and covariance Σ. For regularization, we follow a suggestion in [30] and use

$$\Sigma = \frac{\text{wcov}(\left\{x_{i,k}^{(r)}\right\}_{r=1,\ldots,R_{\text{int}}})}{R_{\text{int}}^{1/3}}$$

where wcov$(\cdot)$ is the weighted covariance of its arguments. The regularization of $\mu_{x_k \to x_i}(x_i)$ enables the computation of $\mu_{x_k \to x_i}(x_i^{(r)})$ (and hence $w_i^{(r)}$) for any $x_i^{(r)}$. We thus obtain αR_{int} weighted samples of $m_i(x_i)$, which can be independently resampled to produce equal-weight samples $\left\{x_i^{(r)}\right\}_{r=1,\ldots,R_{\text{ext}}}$. The computation of $m_i(x_i)$, represented by R_{ext} samples, is of $\mathcal{O}(\alpha R_{\text{int}}^2 |\Gamma_i| + R_{\text{ext}})$ complexity.

We now consider the internal message

$$\mu_{x_i \to x_j}(x_j) = \sum_{x_i} p(\hat{d} \mid d = \|x_i - x_j\|) m_i(x_i)$$

Our goal is to obtain a sample representation $\left\{x_j^{(r)}, w^{(r)}\right\}_{r=1,\ldots,R_{\text{int}}}$ of $\mu_{x_i \to x_j}(x_j)$, given R_{ext} equal-weight samples $x_i^{(r)}$ of $m_i(x_i)$. We first produce R_{int} equal-weight samples of $m_i(x_i)$ by augmenting the original set $\left\{x_i^{(r)}\right\}_{r=1,\ldots,R_{\text{ext}}}$ with the necessary number of copies. The desired samples $x_j^{(r)}$ are related to the given samples $x_i^{(r)}$ by $d = \|x_i - x_j\|$ in the distribution

$$p(\hat{d} \mid d = \|x_i - x_j\|)$$

Hence, we can use samples of d from the ranging distribution $p(\hat{d} \mid d)$ to obtain a sample $x_j^{(r)}$ for every given sample $x_i^{(r)}$. Because $p(\hat{d} \mid d)$ may be difficult to sample from, we use importance sampling. We first draw R_{int} samples of the angle $\theta^{(r)}$ from a uniform distribution on $(0, 2\pi]$, as well as R_{int} samples $d^{(r)}$ from a sampler distribution $q(d \mid \hat{d})$. Each sample $x_j^{(r)}$ is then given by

$$x_j^{(r)} = x_i^{(r)} + d^{(r)} \cdot \begin{bmatrix} \cos\theta^{(r)} \\ \sin\theta^{(r)} \end{bmatrix}$$

The weight $w^{(r)}$ of each sample $x_j^{(r)}$ is given by

$$w^{(r)} \propto \frac{p(\hat{d} \mid d^{(r)})}{q(d^{(r)} \mid \hat{d})}$$

The R_{int}-sample representation of the internal message $\mu_{x_i \to x_j}(x_j)$ requires $\mathcal{O}(R_{\text{int}})$ computational time.

The full localization algorithm is therefore of complexity $\mathcal{O}(\alpha R_{\text{int}}^2 |\Gamma|_{\text{avg}} N N_{\text{it}} + R_{\text{ext}} N N_{\text{it}})$. The terms $|\Gamma|_{\text{avg}}$ and N are properties of the network. However,

α, R_{int}, R_{ext}, and $N_{\text{it}}$are tunable parameters of the algorithm, which may be adjusted to achieve the desired trade-off between complexity and performance.

4.4 Convergence

Ideally, the localization algorithm would stop when it met some convergence criteria. While convergence of the loopy SPA algorithm remains an open area of research [41], extensive simulations have shown it can achieve good performance despite cycles in the graph [42]. Possible stopping criterion include a threshold for the variance of the posterior location distribution of each node. Nodes with "peaky" location distributions stop refining (but continue to broadcast) their beliefs. Alternatively, the algorithm can simply be run for a predetermined number of iterations.

4.5 Location estimation

The estimated coordinates of the node may be given by the mode of the posterior distribution, corresponding to the maximum a posteriori (MAP) estimator. Alternatively, the mean of the posterior distribution provides the minimum mean square error (MMSE) estimate. The posterior distribution may also be used to measure the level of confidence in the estimate, e.g. using the variance of the distribution.

Chapter 5

UWB ranging measurement campaign

In order to develop realistic and tractable models of UWB range measurements, an extensive experimental campaign was carried out. In Section 5.1, the purpose of the campaign is explained in detail. Section 5.2 describes the set-up of the experiment. The resulting range models are presented in Section 5.3.

5.1 Purpose

Our algorithm incorporates a model $p(z_{i\rightarrow j} \mid x_i, x_j)$ that determines the likelihood of the unknown positions x_i and x_j given the measurement $z_{i\rightarrow j}$. The performance of the algorithm is heavily dependent on how well this model represents the true distribution of $z_{i\rightarrow j}$ given the location of the sending and receiving nodes. Measurement models are also typically used to simulate ranging measurements to evaluate the performance of localization algorithms. Hence, these models affect both the true performance of the algorithm as well as simulated performance evaluation. Because realistic ranging models have generally

been unavailable, authors have thus far resorted to models derived from highly idealized signals [7, 47, 48]. This leads to misleading results for the localization algorithms that build upon these ranging models, as well as unrealistic performance characteristics based on simulation.

The purpose of our measurement campaign was to develop realistic yet tractable ranging error models for commercial UWB radios. Ranging measurements were collected in a variety of indoor environments around the MIT campus, including offices, hallways, and hangars, and in line-of-sight (LOS) and non-line-of-sight (NLOS) conditions. On the resulting database of ranging measurements, we applied learning algorithms to model the ranging error and derived tractable models using Gaussian mixtures densities. These UWB ranging models are later incorporated into the localization algorithm in Chapter 6.

5.2 Experimental set-up

For the experiment, two Time Domain Corporation PulsOn® 210 radios [49] were used. These commercial UWB radios, shown in Figure 5.1, have a center frequency of approximately 4.7 GHz with bandwidth 3.2 GHz and comply with FCC power regulations. Each radio is of dimensions 16.5cm x 10.2cm x 5.1cm, a feasible size for practical localization systems. The node is able to transmit and receive packets through an omni-directional antenna. Ranging is accomplished using the roundtrip time-of-flight; one node sends a request, to which the other returns a reply. The roundtrip time-of-flight is calculated at the requester using the time-of-arrival and an estimate of the electrical delay, the amount of time the responder takes to process the packet, which is included in the response. In the next ranging request, the range estimate $\hat{d}$

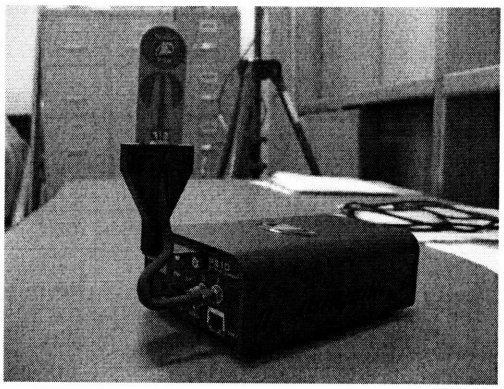


Figure 5.1: Time Domain Corporation PulsOn® 210 radio

is piggybacked in the range packet, so that both nodes know their internode distance. To account for the nature of realistic localization systems, which may be composed of off-the-shelf parts, no modifications were made to the hardware or embedded and host software. Range measurements were collected and used as is.

Our measurements thus specifically characterize off-the-shelf, FCC-compliant ranging devices that can be readily deployed in practical localization systems, unlike previous UWB experimental campaigns that employed equipment or data processing that may not be realistically feasible. For example, measurements made with a vector network analyzer and low noise amplifiers [50, 51] may result in a high signal-to-noise ratio that fails to represent real-world scenarios, where such hardware often cannot be accommodated. These experiments also make use of extensive post-processing to analyze data collected in the frequency domain. Furthermore, [51] assumes ideal detection and channel estimation, which may not be true of practical systems. Many other UWB range models are based on measurement campaigns to date have been undertaken with the goal of characterizing channel parameters such as path loss, fading, and delay spread, independent of the effect of the measurement device and methods [52, 53].

Location	Signal characterization	Min-max separation d_{sep}
LIDS 6th-floor hallway	LOS	0.25-19.25 m
LIDS 6th-floor office and lobby	NLOS (concrete wall)	2.75-9.00 m
CSAIL 3rd-floor hallway	LOS	0.25-13.50 m
CSAIL 3rd-floor hallway	NLOS (glass doors)	1.50-13.50 m
Aero/Astro hangar	LOS	0.25-12.00 m

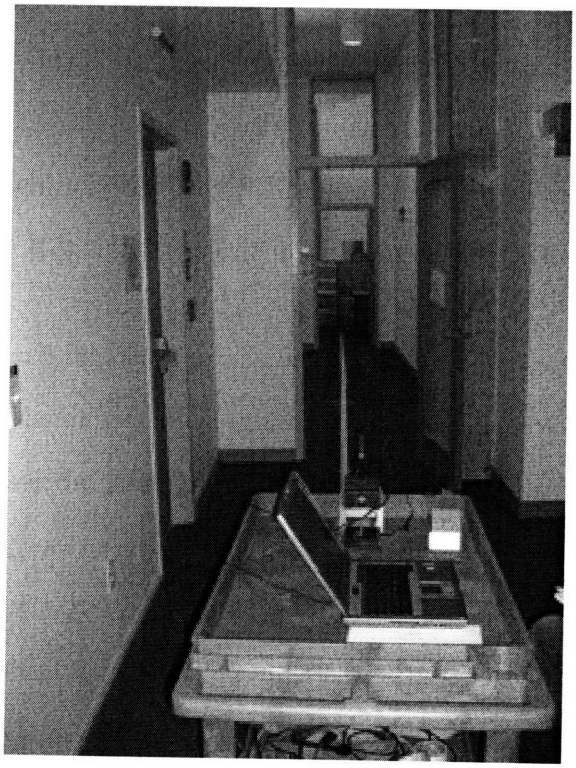
Table 5.1: Environments used for measurement campaign

A series of five campaigns were performed in different indoor environments around the MIT campus, as described in Table 5.1. Two environments, CSAIL-NLOS and LIDS-NLOS, were characterized by non-line-of-sight conditions, while the line-of-sight between the nodes was unobstructed in CSAIL-LOS, LIDS-LOS, and Hangar-LOS. In each environment, the nodes were placed 89 centimeters above the ground. At each distance d_{sep} of separation, 1000 ranging measurements were collected. We perform no averaging of measurements, unlike [50,51]. The separation of the nodes was increased in increments of 25 centimeters. The experimental set-up in two environments is shown in Figure 5.2.

5.3 Ranging models

We observed that a histogram of the 1000 ranging measurements collected at each true distance d_{sep} typically contained one large peak near d_{sep}, plus a small set of outliers on each side of the peak. For example, Figures 5.3-5.5 show histograms of the measurements collected at $d_{sep} = 5.25$ m in each environment. The outliers are consistently centered at large distances from the main peak, sometimes producing negative range measurements. The fact that some measured ranges are significantly *less than* and greater than the true distance d_{sep} indicates that far-lying outliers are primarily caused by the ranging algorithm rather than multipath or NLOS effects. Further examina-

(a) LIDS-LOS



(b) CSAIL-LOS

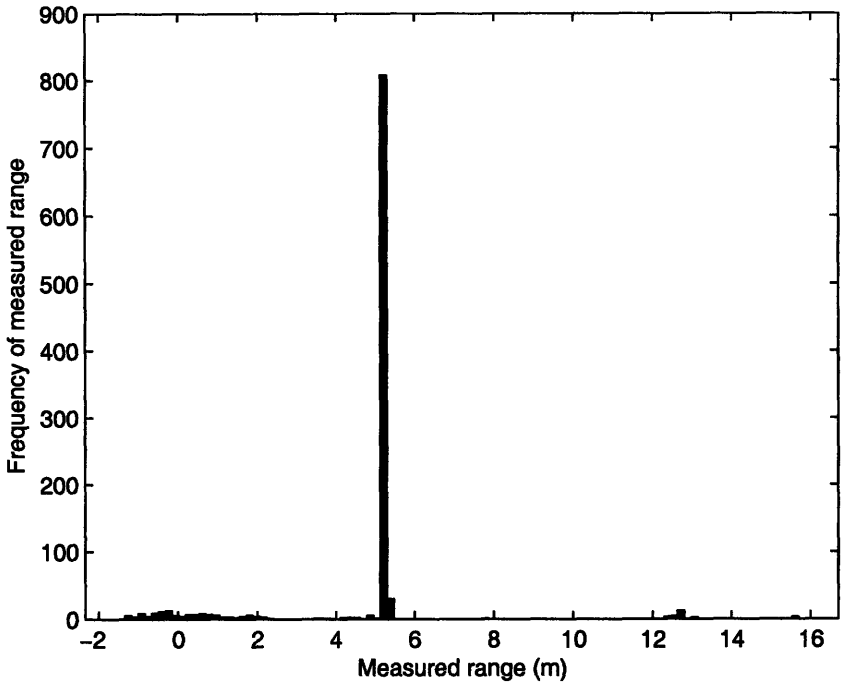
Figure 5.2: Experimental set-up in two environments

tion revealed that the electrical delay estimated by the Time Domain nodes under a proprietary algorithm was subject to high variance and possibly large errors. These findings indicate that the measurement devices and methods are important factors to take into consideration when characterizing UWB range measurements.

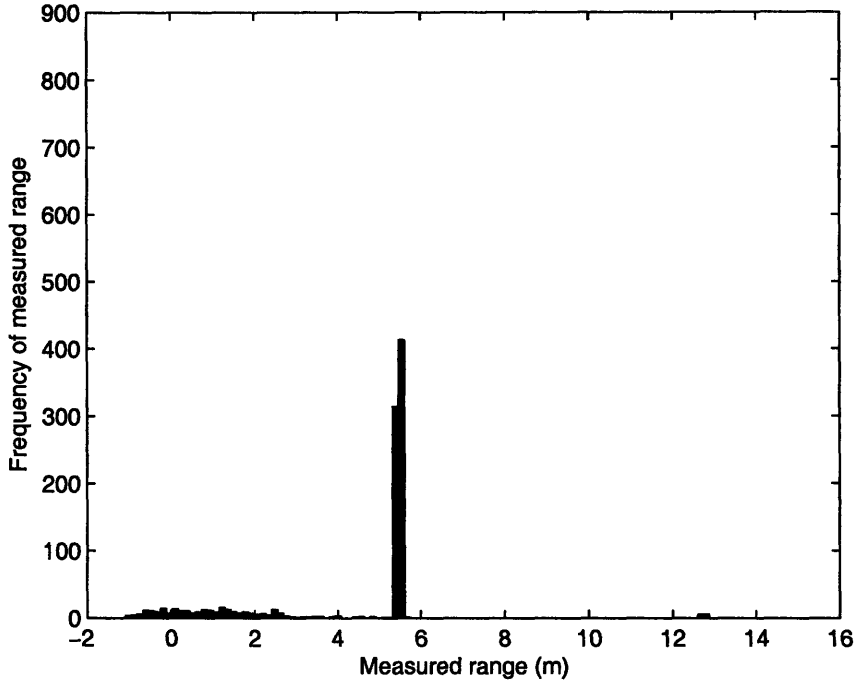
Figures 5.3-5.5 show that the ranging environment has a significant effect on the distribution of the measurements. The measurements collected in the CSAIL hallway, with no clutter, have much fewer outliers than those collected in the LIDS hallway, with adjacent concrete pillars and walls, and in the hangar, where large crates and other objects were nearby. The figures additionally show that NLOS measurements tend to have more outliers than LOS measurements collected in the same environment. Moreover, the variance of the main peak in LIDS-NLOS is greater than the variance of LIDS-LOS. Additionally, the nature of the blockage plays an important role in determining its effect on NLOS measurements. The glass doors in CSAIL caused much fewer outliers than the concrete wall in LIDS. These findings are corroborated by the results in other UWB measurement campaigns [51].

Using the histograms, we concluded that a reasonable underlying distribution for the measurements collected in a given environment at distance d_{sep} is a Gaussian mixture density with three components, labeled $l = 1, 2, 3$ for the lower outliers, main component, and upper outliers respectively. Each component is parametrized by a mean $M_l(d_{\text{sep}}, \mathcal{E})$, a variance $R_l(d_{\text{sep}}, \mathcal{E})$, and a weight $W_l(d_{\text{sep}}, \mathcal{E})$. Hence, the distribution of range measurements collected at true distance d_{sep} is given by

$$p(\hat{d} \mid d_{\text{sep}}, \mathcal{E}) = \sum_{l=1}^{3} P_{W,l}(d_{\text{sep}}, \mathcal{E}) \mathcal{N}(P_{M,l}(d_{\text{sep}}, \mathcal{E}), P_{R,l}(d_{\text{sep}}, \mathcal{E}))$$

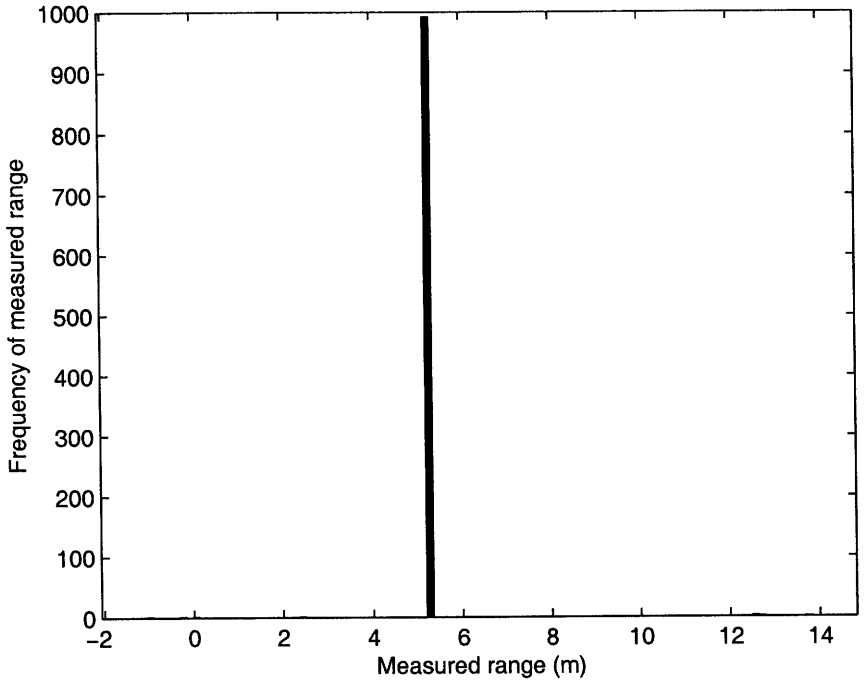


(a) LOS

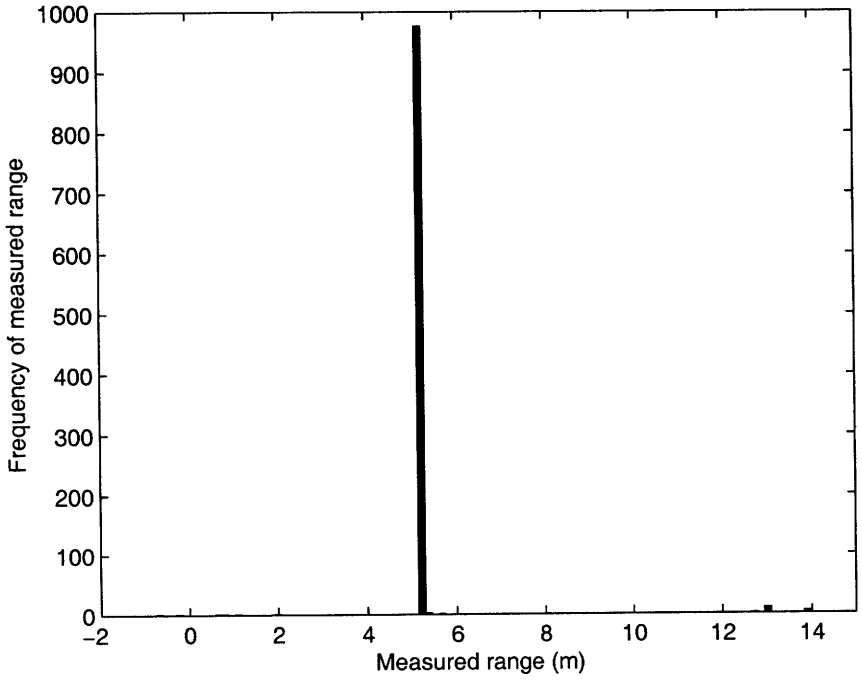


(b) NLOS

Figure 5.3: Histograms of UWB measurements collected in LIDS at $d_{\text{sep}} = 5.25$ m



(a) LOS



(b) NLOS

Figure 5.4: Histograms of UWB measurements collected in CSAIL at $d_{\text{sep}} = 5.25$ m

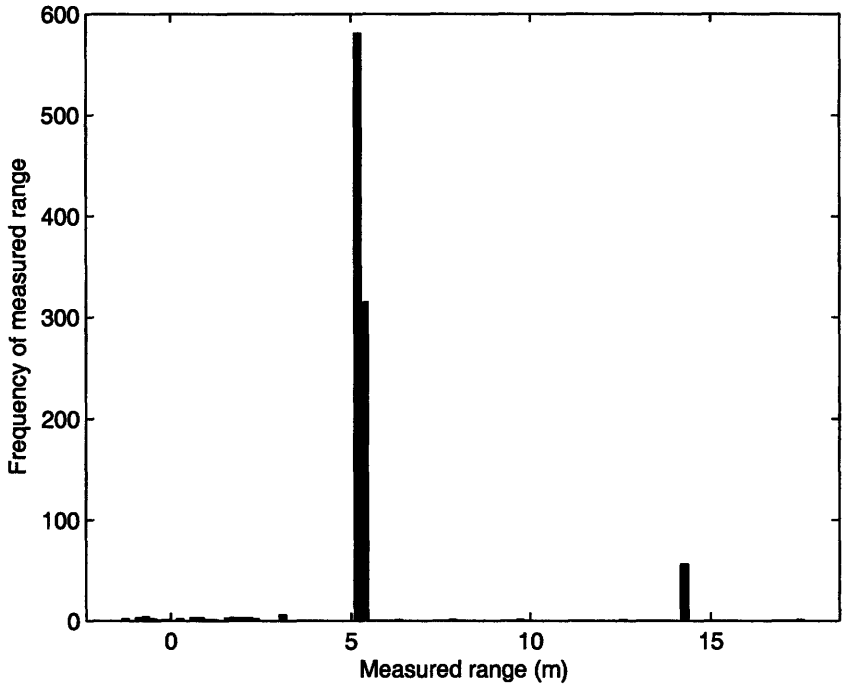


Figure 5.5: Histogram of UWB measurements collected in Hangar-LOS at $d_{\text{sep}} = 5.25$ m

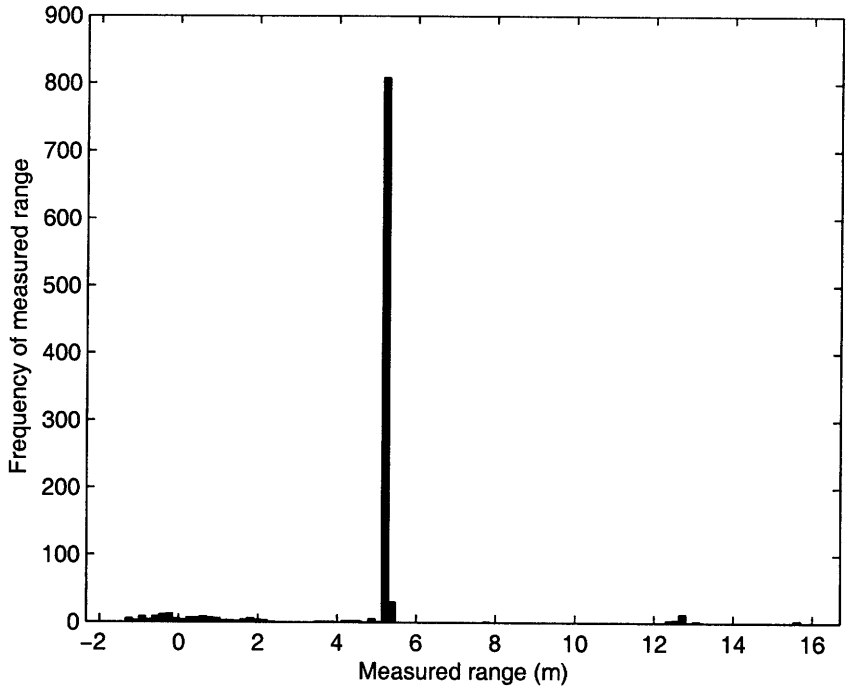
In order to determine the means $M_l(d_{\text{sep}}, \mathcal{E})$, variances $R_l(d_{\text{sep}}, \mathcal{E})$, and weights $W_l(d_{\text{sep}}, \mathcal{E})$, we applied the expectation-maximization (EM) algorithm [54,55] to the set of 1000 measurements collected at each distance d_{sep} in environment $\mathcal{E}$. The EM algorithm uses the maximum likelihood criteria to estimate the parameters of a Gaussian mixture. As an example, Figure 5.6 shows the histogram for $d_{\text{sep}} = 5.25$ m in LIDS-LOS with the resulting Gaussian mixture distribution.

Each environment $\mathcal{E}$ is thus characterized by a set of Gaussian parameters:

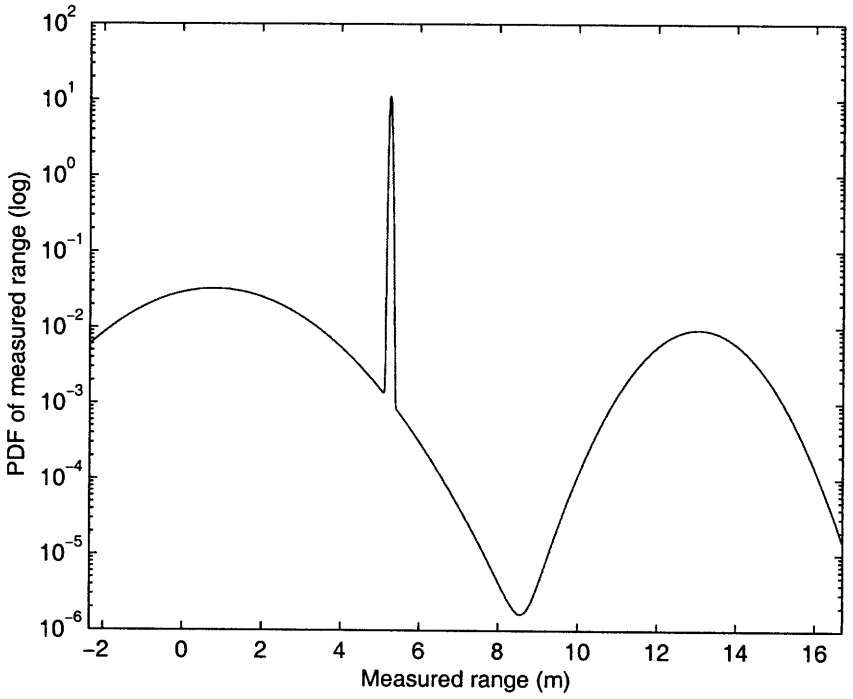
$$\left\{ \{M_l(d_{\text{sep}}, \mathcal{E})\}_{l=1,2,3}, \{R_l(d_{\text{sep}}, \mathcal{E})\}_{l=1,2,3}, \{W_l(d_{\text{sep}}, \mathcal{E})\}_{l=1,2,3} \right\}_{\forall d_{\text{sep}}}$$

For example, the set of parameters for LIDS-NLOS are plotted as a function of d_{sep} in Figure 5.7. Results presented later in the chapter (Figures 5.8-5.12) include plots of the parameters for all environments. The EM-determined Gaussian parameters capture the features of the histograms, with the main component $l = 2$ typically having high weight, very low variance, and a mean $M_2(d_{\text{sep}}, \mathcal{E})$ with very small bias. NLOS measurements, such as in Figure 5.9, exhibit a larger positive bias $M_2(d_{\text{sep}}, \mathcal{E}) - d_{\text{sep}}$ than the corresponding LOS measurements, Figure 5.8. This agrees with other UWB measurement campaigns and models [43]. The lower and upper outliers, represented by components $l = 1$ and $l = 3$ respectively, are characterized by smaller weights than $l = 2$, while the variances are greater. The means of the outlier components are offset by fairly constant biases $M_l(d_{\text{sep}}, \mathcal{E}) - d_{\text{sep}}$. Unlike [51], we find that the variance the measurements does not always increase with distance. Our results demonstrate that the effect of the surrounding environment may outweigh the effect of distance and LOS/NLOS conditions.

We model how the distribution of ranging measurements varies with con-



(a) Histogram



(b) Distribution

Figure 5.6: Histogram and Gaussian mixture fit for true distance $d_{\text{sep}} = 5.25$ m in LIDS-LOS

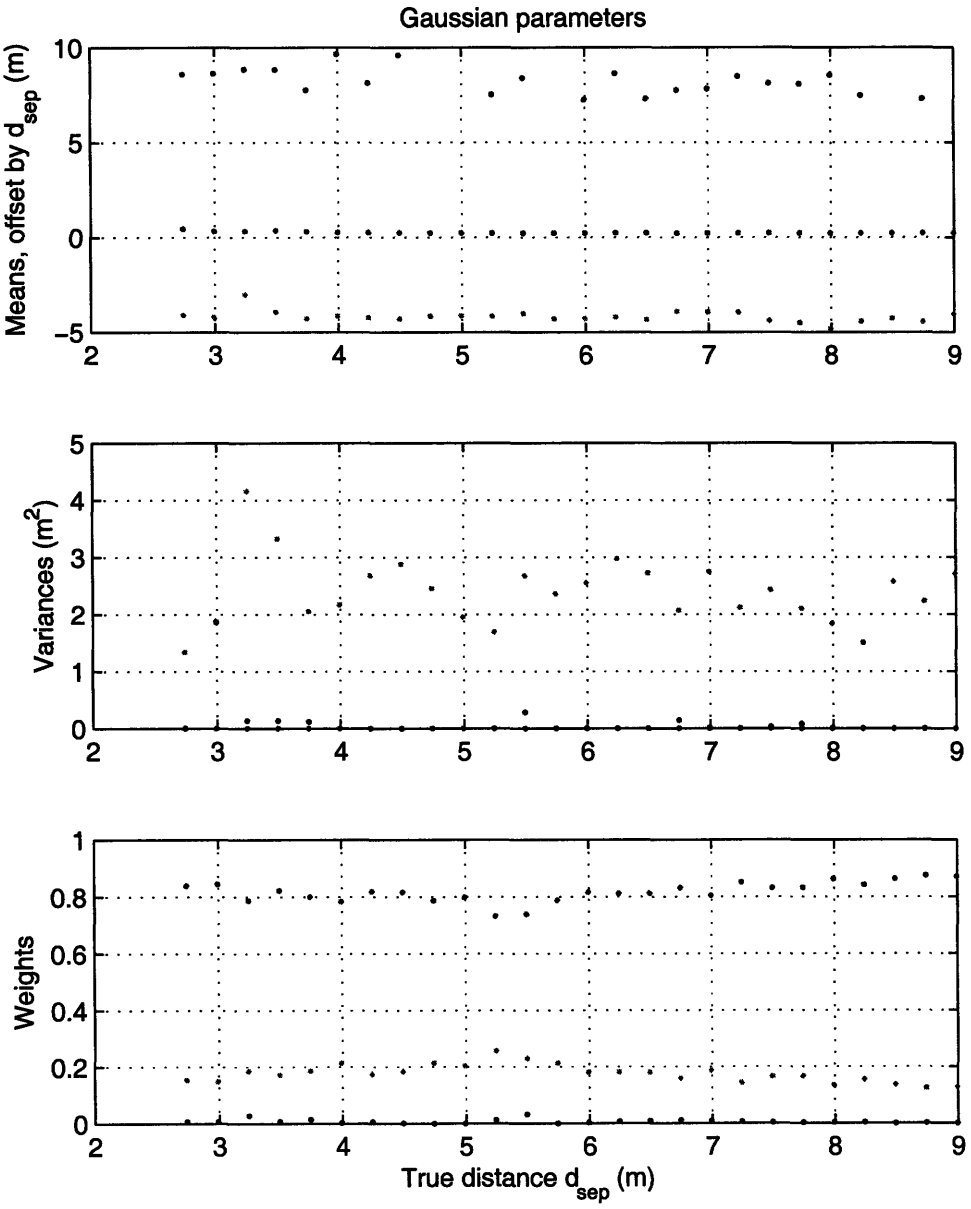


Figure 5.7: Gaussian parameters for LIDS-LOS

tinuous distance d by fitting quadratic polynomials $P_{\mathcal{E},M,l}(d)$, $P_{\mathcal{E},R,l}(d)$, and $P_{\mathcal{E},W,l}(d)$ to $\{M_l(d_{\text{sep}},\mathcal{E})\}_{\forall d_{\text{sep}}}$,$\{R_l(d_{\text{sep}},\mathcal{E})\}_{\forall d_{\text{sep}}}$, and $\{W_l(d_{\text{sep}},\mathcal{E})\}_{\forall d_{\text{sep}}}$ respectively. The coefficients for the resulting polynomials of form $a_2d^2+a_1d+a_0$ are shown in Table 5.2. Our final model of the UWB ranging measurements $\hat{d}$ as a function of true distance d in environment $\mathcal{E}$ is a Gaussian mixture described by

$$p(\hat{d} \mid d,\mathcal{E}) = \sum_{l=1}^{3} P_{\mathcal{E},W,l}(d)\mathcal{N}(P_{\mathcal{E},M,l}(d), P_{\mathcal{E},R,l}(d))$$

The Gaussian parameters with polynomial fits for each environment are displayed in Figures 5.8-5.12. With the polynomial coefficients as its only parameters, our UWB ranging model is tractable and easily implemented in low-complexity localization systems. Moreover, it accurately describes a practical ranging device operating in non-ideal, realistic environments.

	$P_{\mathcal{E},M,1}(d)$			$P_{\mathcal{E},M,2}(d)$			$P_{\mathcal{E},M,3}(d)$		
$\mathcal{E}$	a_2	a_1	a_0	a_2	a_1	a_0	a_2	a_1	a_0
LIDS-LOS	-0.0034	1.0692	-4.6091	-0.0003	1.0075	-0.0298	-0.0044	1.0911	7.5242
LIDS-NLOS	0.0088	0.8279	-3.5035	0.0099	0.8623	0.6908	0.0084	0.7164	9.5428
CSAIL-LOS	-0.0021	1.1966	-4.5550	-0.0006	1.0130	-0.0480	-0.0092	1.1022	7.5952
CSAIL-NLOS	0.0151	0.8682	-3.9301	-0.0002	1.0119	-0.0470	-0.0162	1.1666	7.6238
Hangar	0.0114	0.8411	-3.7557	-0.0005	1.0120	-0.0204	0.0293	0.9131	7.4155
	$P_{\mathcal{E},R,1}(d)$			$P_{\mathcal{E},R,2}(d)$			$P_{\mathcal{E},R,3}(d)$		
$\mathcal{E}$	a_2	a_1	a_0	a_2	a_1	a_0	a_2	a_1	a_0
LIDS-LOS	0.0048	-0.1004	2.8145	0	0	0.0007	-0.0044	0.0587	0.3481
LIDS-NLOS	-0.0143	0.1247	2.2029	0.0001	-0.0015	0.0056	0.0353	-0.4750	1.5721
CSAIL-LOS	0.0311	-0.2094	2.2466	0	0	0.0002	-0.0280	0.4123	-0.6412
CSAIL-NLOS	0.0373	-0.3649	2.7385	0	0	0.0002	-0.0134	0.1721	-0.1089
Hangar	0.0219	-0.1978	2.5155	-0.0002	0.0022	-0.0030	-0.0556	0.9958	0.3032
	$P_{\mathcal{E},W,1}(d)$			$P_{\mathcal{E},W,2}(d)$			$P_{\mathcal{E},W,3}(d)$		
$\mathcal{E}$	a_2	a_1	a_0	a_2	a_1	a_0	a_2	a_1	a_0
LIDS-LOS	-0.0022	0.0501	-0.0492	0.0024	-0.0519	1.0434	-0.0001	0.0018	0.0058
LIDS-NLOS	-0.0066	0.0695	0.0185	0.0068	-0.0707	0.9733	-0.0002	0.0012	0.0082
CSAIL-LOS	0.0003	-0.0002	0.0019	0.0002	-0.0073	1.0049	-0.0005	0.0075	-0.0068
CSAIL-NLOS	0.0003	-0.0018	0.0063	0.0002	-0.0045	0.9875	-0.0005	0.0063	0.0062
Hangar	-0.0004	0.0069	0.0179	0.0023	-0.0327	1.0147	-0.0019	0.0258	-0.0327

Table 5.2: Polynomial coefficients of ranging models

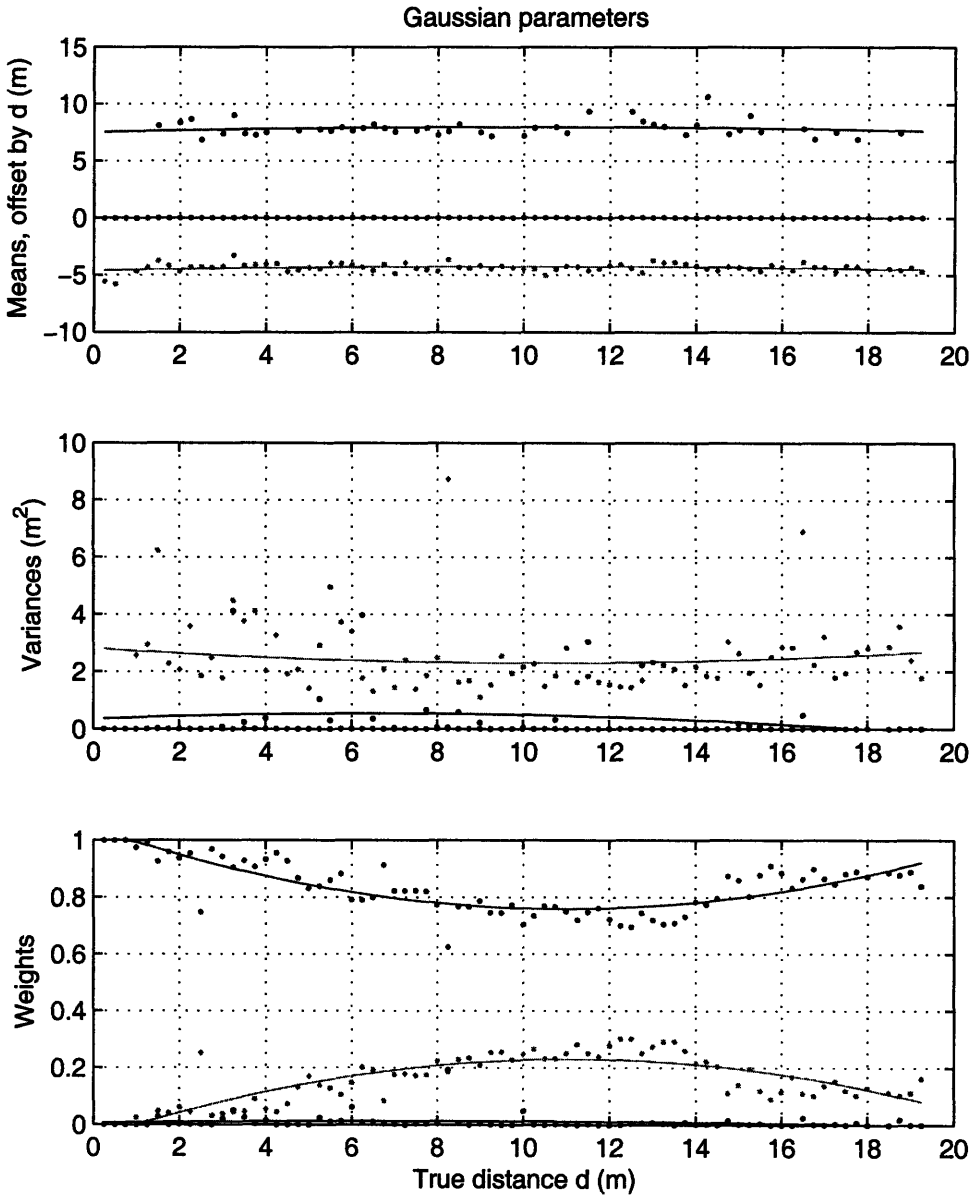


Figure 5.8: Gaussian parameters and polynomial fits for LIDS-LOS

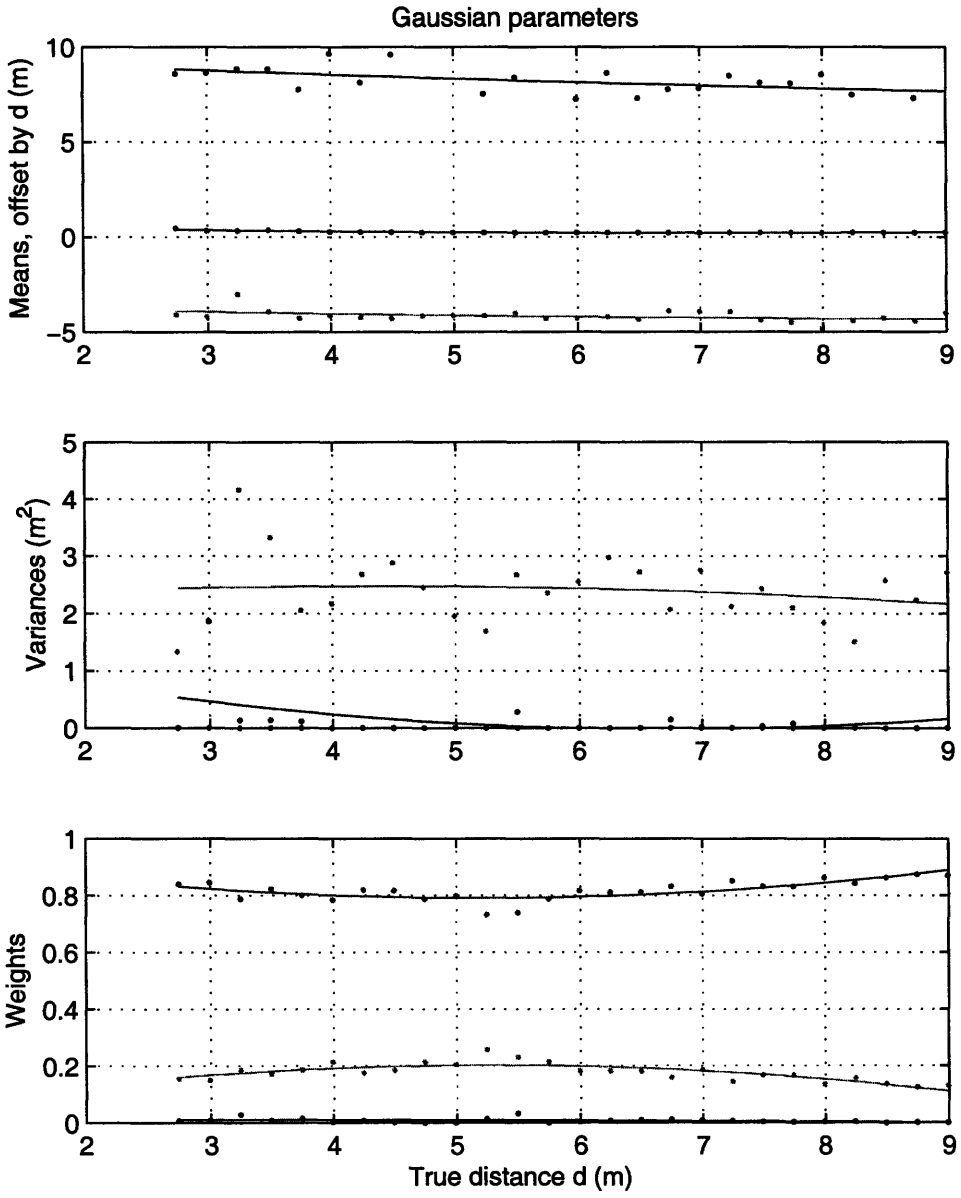


Figure 5.9: Gaussian parameters and polynomial fits for LIDS-NLOS

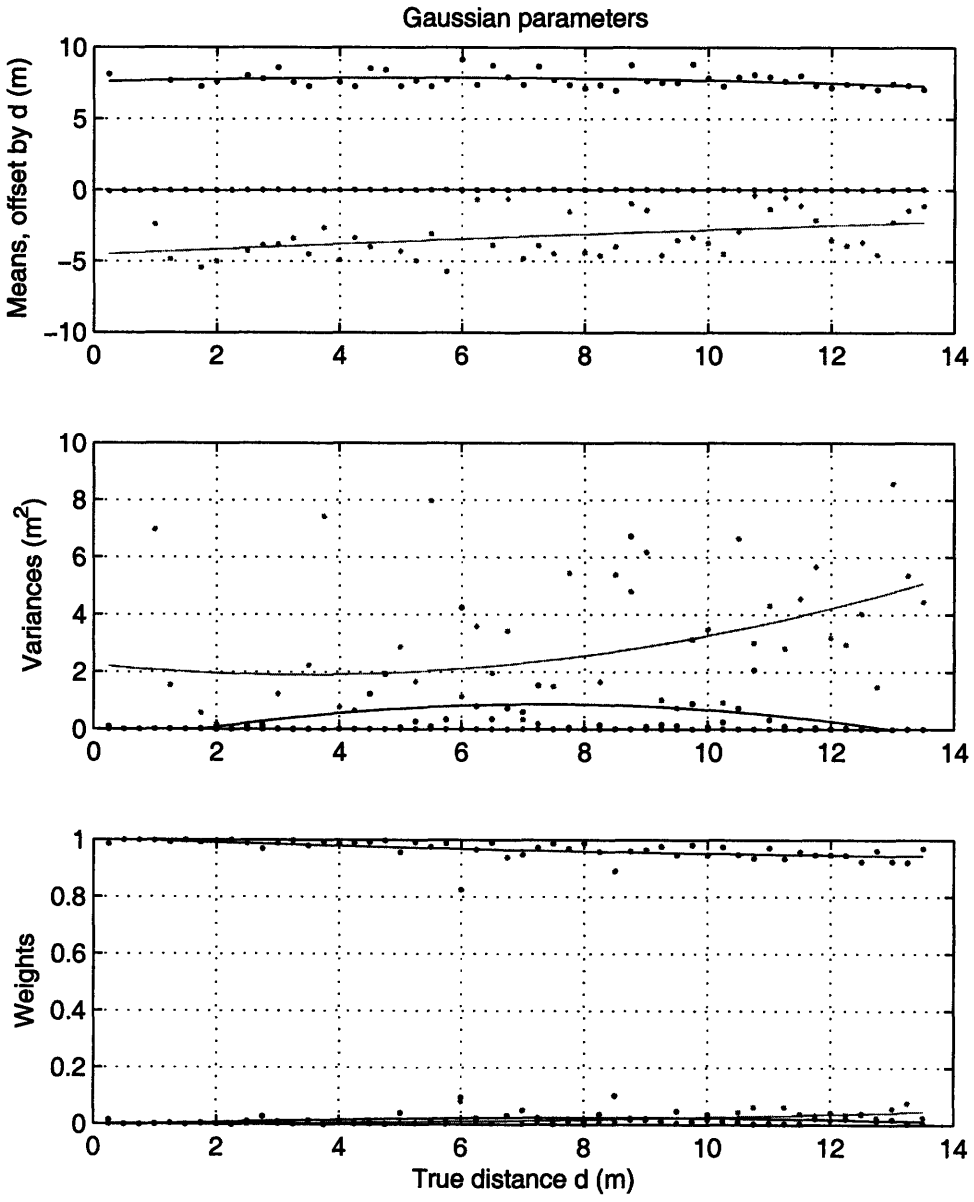


Figure 5.10: Gaussian parameters and polynomial fits for CSAIL-LOS

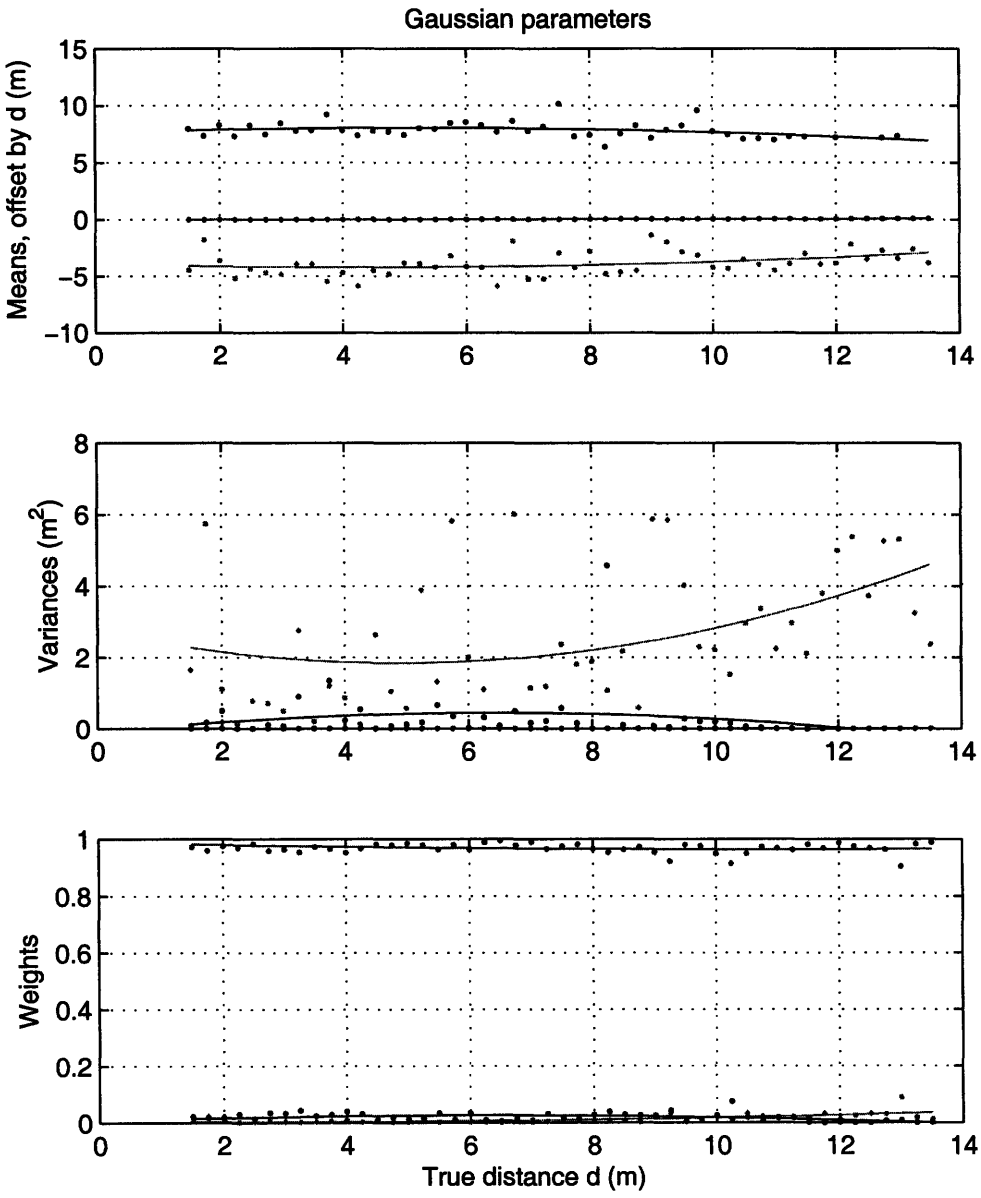


Figure 5.11: Gaussian parameters and polynomial fits for CSAIL-NLOS

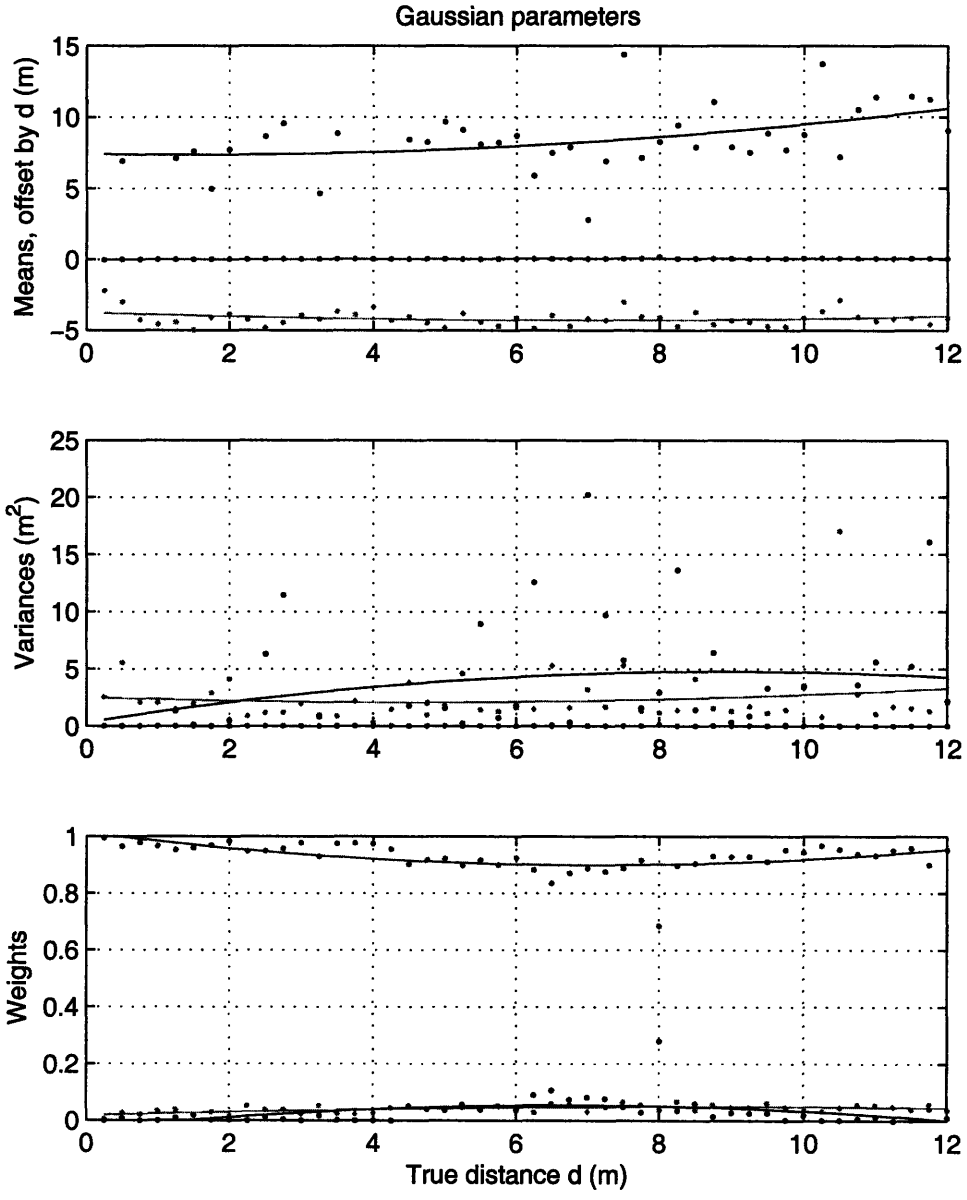


Figure 5.12: Gaussian parameters and polynomial fits for Hangar-LOS

Chapter 6

Performance analysis

In this chapter, we analyze the performance of the cooperative localization algorithm. A series of simulations incorporating realistic range measurements and the experimentally-developed range models were carried out, as described in Section 6.1. In Section 6.2, we present and analyze the results of the simulations.

6.1 Simulations

To characterize the performance of the cooperative localization algorithm, an extensive series of simulations was designed and executed. These simulations incorporated the range models that were experimentally developed in Chapter 5. The purpose of the simulations was to compare the distributed cooperative localization algorithm to two benchmark algorithms under the following varying conditions:

- number of anchor nodes
- LOS and NLOS conditions
- ability and inability to identify NLOS range measurements

We created 100 network topologies $T_1, \ldots, T_{100}$ by uniformly distributing the 100 agents across a 100 meter by 100 meter environment. Each topology T_i was superimposed on two maps, LOS and NLOS. In the former, no obstructions exist; hence all signals are LOS. We simulated a set of range measurements $\mathcal{Z}_i^{\text{LOS}}$ for each topology T_i in the LOS map. If nodes i and j were within 20 meters of each other, range measurements $\hat{d}_{i\to j}$ and $\hat{d}_{j\to i}$ were drawn from a combination of the LIDS-LOS, CSAIL-LOS, and Hangar-LOS distributions described in Section 5.3. For each communication link $i \to j$, we took 10 range measurements and threw away the outliers, a reasonable strategy for a realistic localization system. We then use one of the remaining measurements for $\hat{d}_{i\to j}$.

The second map was based on a simple indoor floor plan, shown in Figure 6.1. Another set of range measurements $\mathcal{Z}_i^{\text{NLOS}}$ was simulated for each topology T_i. For fair comparison, LOS range measurements were set equal to those in $\mathcal{Z}_i^{\text{LOS}}$, as described above. When nodes were separated by a wall, we considered the signal to be NLOS. NLOS range measurements were drawn from the LIDS-NLOS distribution up to distances of 10 meters. Again, multiple range measurements were taken for each link, with the outliers thrown away.

For each topology T_i with corresponding measurements $\mathcal{Z}_i^{\text{LOS}}$ and $\mathcal{Z}_i^{\text{NLOS}}$, we ran the localization algorithm, varying the following parameters with each trial:

Number of anchor nodes Simulations were performed with 4, 6, 8, 10, and 12 anchors, all uniformly distributed on the map.

NLOS identification In half of the simulations, we assumed perfect identification of NLOS signals. Identified LOS range measurements used the combined LOS range model for $p(\hat{d} \mid d)$. Identified NLOS measurements used the

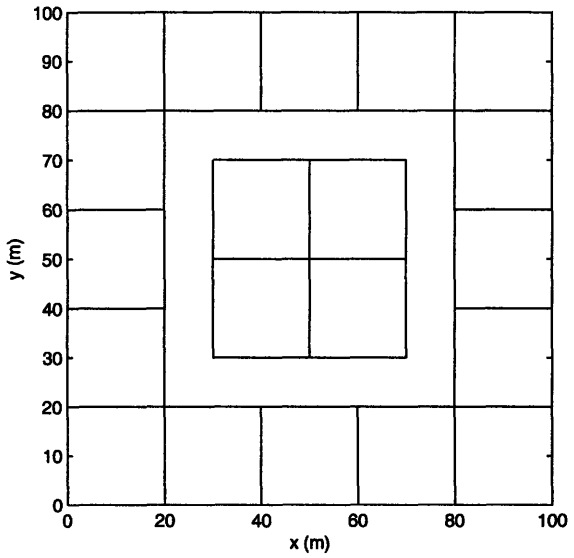


Figure 6.1: NLOS map

LIDS-NLOS distribution. In the other half of the simulations, no NLOS identification was used. The range model was then a combination of the LOS and NLOS distributions.

Localization algorithm We compared the performance of the distributed cooperative localization algorithm and a non-cooperative localization algorithm.

A smaller scale study of the following parameters was also performed.

Number of samples We performed a limited number of tests to determine how the performance of the algorithm changed with the number of samples transmitted per message and the number of samples used for internal computation, denoted R and R_{int} respectively.

Number of iterations We also examined how the accuracy of the algorithm increased with the number of iterations.

To account for the removal of outlier range measurements, we used only the main Gaussian component ($l = 2$) in the ranging model $p(\hat{d} \mid d)$. The sampler distribution $q(d \mid \hat{d})$ was a Gaussian with mean and variance given by fitting the measured range $\hat{d}$ to the polynomials $M_2(d)$ and $R_2(d)$ respectively. After the algorithm terminated, we estimated the location of each node using the mode of the appropriate marginal distribution. We then measured the error for each node, defined as the distance between the node's true and estimated locations.

6.2 Results

Figure 6.2 shows how the accuracy of the cooperative localization algorithm increases with each iteration, as information propagates out from the anchors. The estimated location of each node is connected to the true location by a gray line, which represents the error for that node. The result after the first iteration in Figure 6.2(a) is equivalent to the non-cooperative localization algorithm; agents receive information only from anchor nodes. Hence, all nodes outside of anchor range have no information about their location, as shown by their lack of a location estimate. The improvement between Figures 6.2(a) and (d) clearly demonstrates the benefit of cooperation for localization.

We can quantify the benefit of localization more systematically by examining the errors in all 100 network topologies for any combination of parameters. The outage probability $\mathcal{P}(e)$ at each error value e is defined as the probability of a node having error greater than e. Hence, the faster the function goes to 0, the better the performance of the algorithm. To calculate the outage probability $\mathcal{P}(e)$, we determined the number of nodes with error greater than e in $T_1, \ldots, T_{100}$ and normalized by the total number of nodes (100^2).

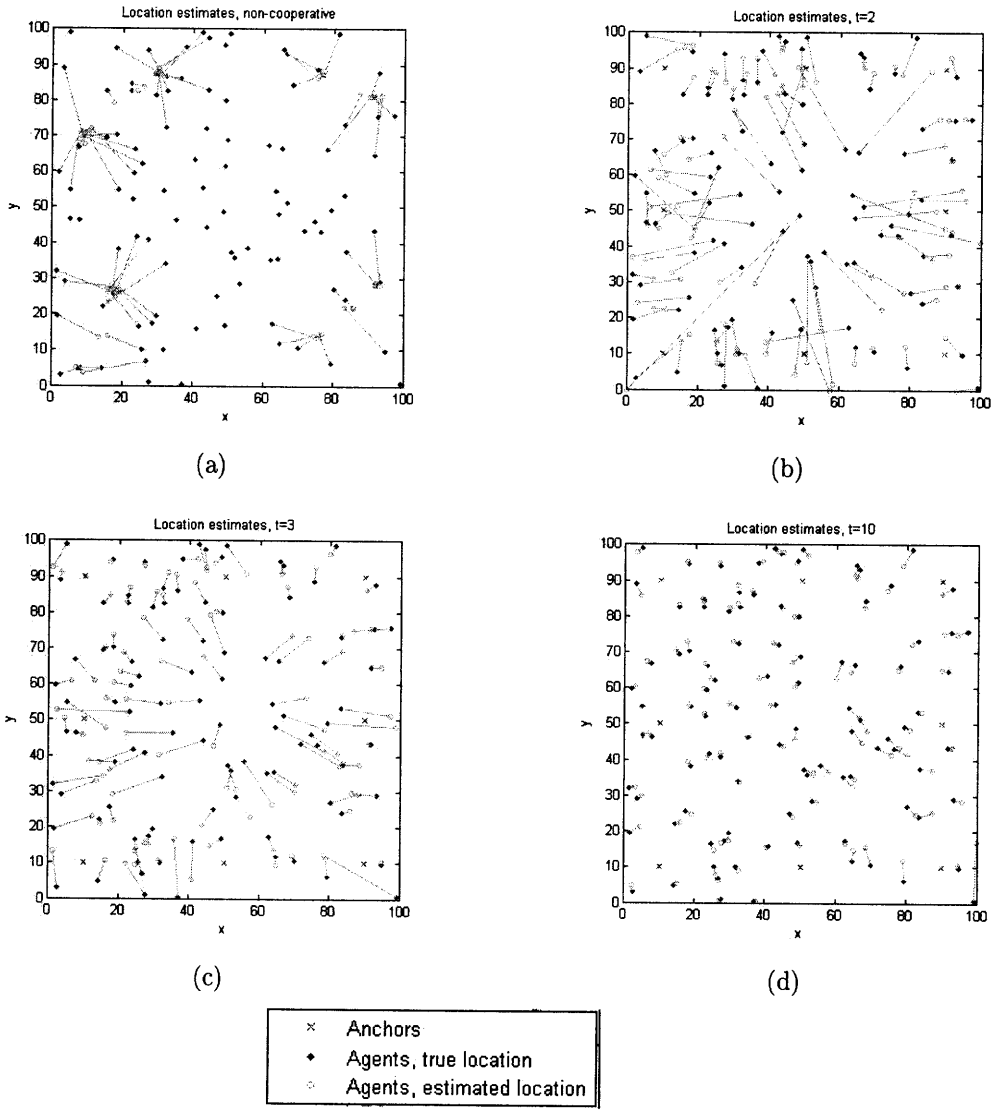


Figure 6.2: Evolution of estimated locations

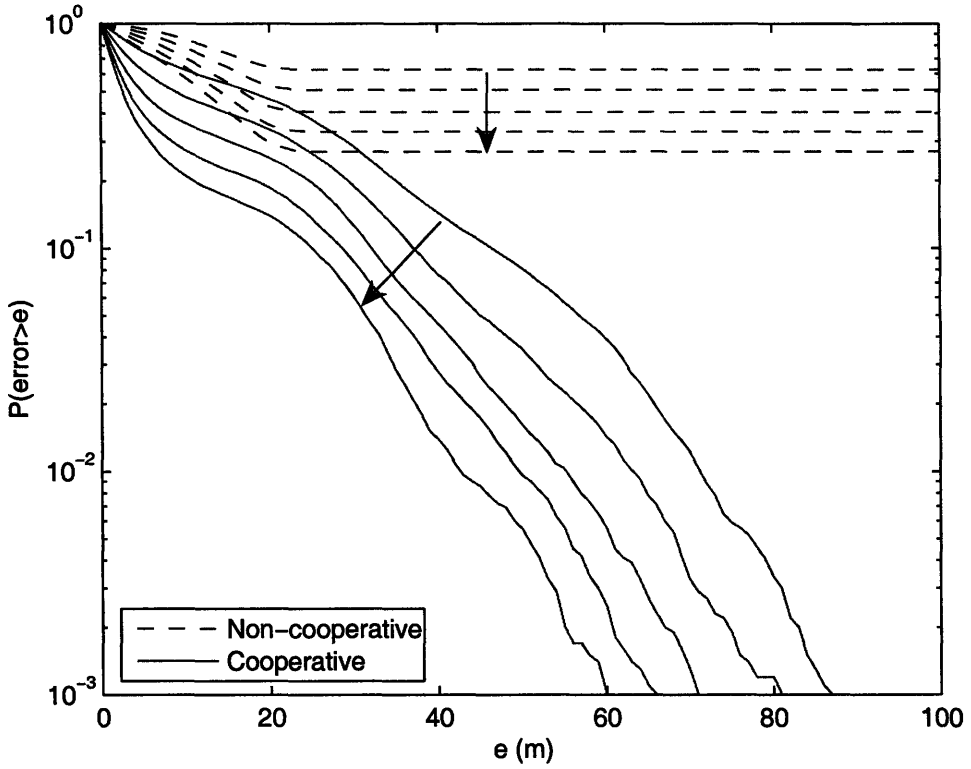


Figure 6.3: Outage probability with increasing number of anchors (4, 6, 8, 10, 12)

Figure 6.3 shows the outage probability for the NLOS map with no NLOS identification. The arrows indicate the increasing number of anchors (4, 6, 8, 10, and 12) for both the cooperative algorithm (after 7 iterations) and the non-cooperative algorithm. As expected, the performance of both algorithms improves as the number of anchors increases. More importantly, the cooperative algorithm consistently outperforms the non-cooperative algorithm. In fact, the non-cooperative is able to localize only a small fraction of the nodes, while the cooperative algorithm localizes almost all.

Figure 6.4 compares the performance of localization with and without NLOS identification in the NLOS environment. The identification of NLOS signals, along with the corresponding adjustment of the ranging distribution

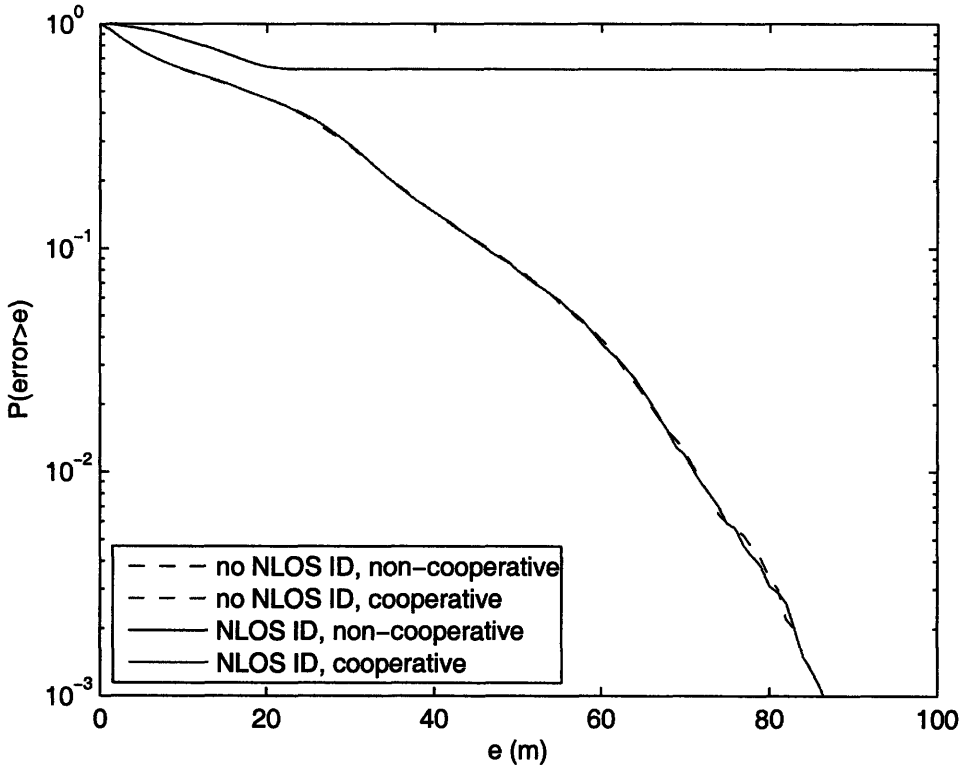


Figure 6.4: Outage probability with and without NLOS identification

$p(\hat{d} \mid d)$, has little effect on the outage curve. We believe the similarity in performance is due to the robustness of UWB ranging in both LOS and NLOS environments. These results confirm that ultra-wideband technology is an appropriate choice for localization.

Figure 6.6 compares localization with random anchor placement to localization with a predetermined configuration. The anchor configuration, shown in Figure 6.5, was chosen based on a small-scale study of the performance of different anchor placements.We found that the larger the coverage of the anchors, the better the localization performance. In addition, placing anchors as far away from each other as possible ensures that all nodes are within a few hops of an anchor. Placing each node in the vicinity of an absolute reference reduces the possibility of a large cluster of neighbors achieving correct rela-

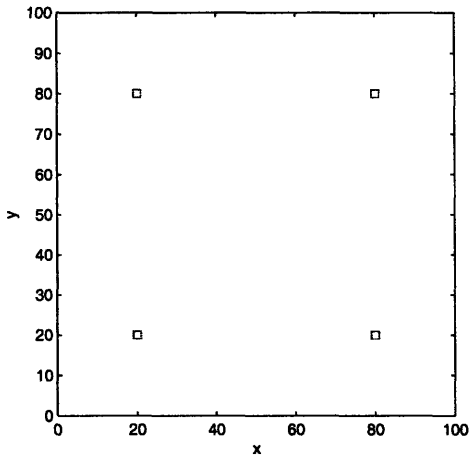


Figure 6.5: 4-anchor configuration

tive location estimates, but not absolute. Evidently, a good choice of anchor configuration can significantly improve the performance of the localization algorithm. The performance gain for the cooperative algorithm is much larger than that of the non-cooperative.

Finally, Figure 6.7 shows how the performance of the algorithm changes with the number of samples used. This small-scale study was run only on topology T_1 for 8 anchors, with no NLOS identification in the NLOS map. The heavy blue plots indicate the performance for $R = 100$ transmitted samples and $R_{\text{int}} = 500$ internal samples, the settings used for all previous results. In Figure 6.7(a), only R_{int} changes, while R remains constant at 100. Varying only the number of internal samples has little (and inconsistent) effect, suggesting that changing R_{int} alone may cause negligible performance gain. In Figure 6.7(b), we vary both R and R_{int}. In this case, the number of nodes with small error (less than 10 meters) clearly increases with the number of samples. More transmitted and internal samples thus results in better localization, at the cost of greater computational complexity. Once again, the performance gained by the cooperative algorithm is much greater than that by the non-cooperative.

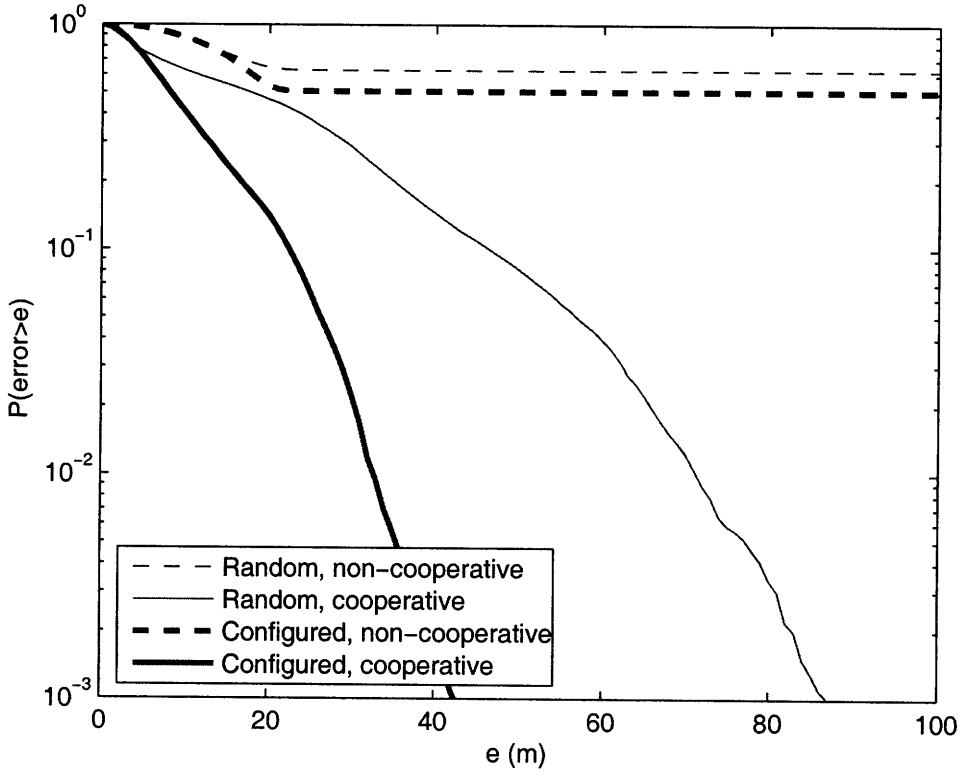
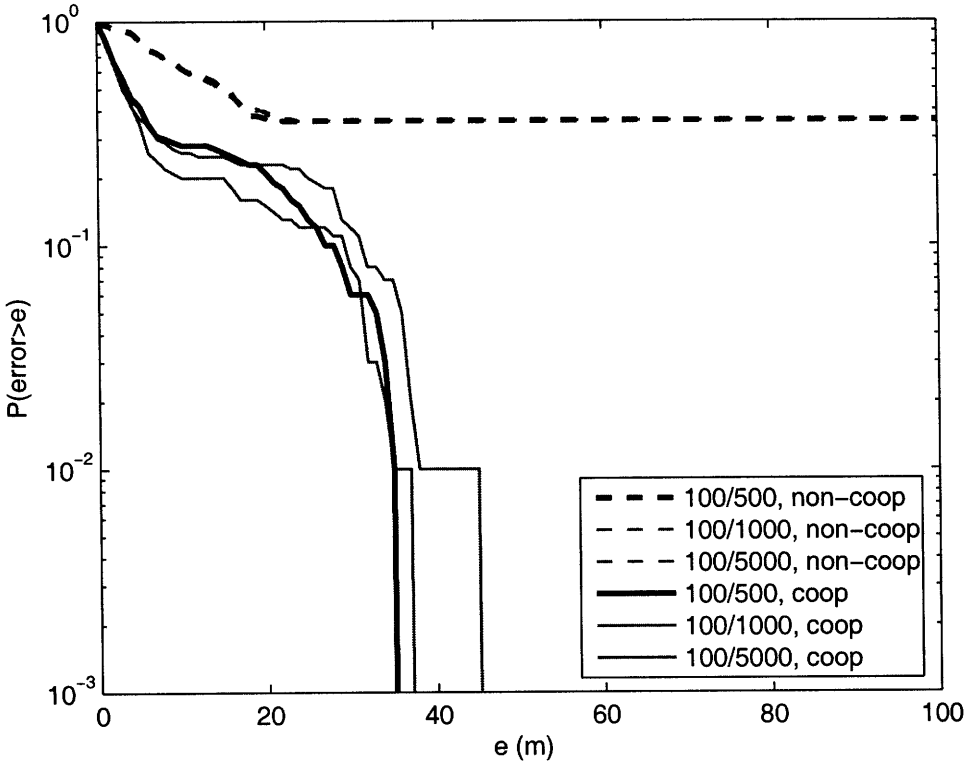
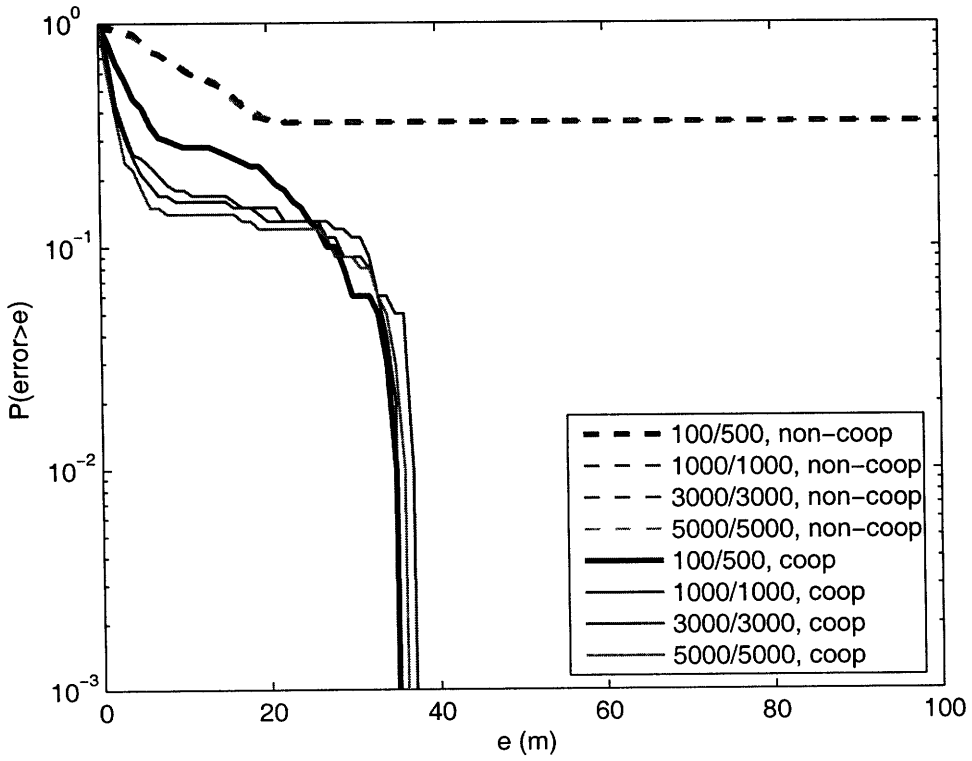


Figure 6.6: Outage probability for random and configured anchor placements



(a) Internal samples



(b) Transmitted and internal samples

Figure 6.7: Outage probability with different numbers of samples

Chapter 7

Conclusions

In this thesis, we derived an algorithm for cooperative localization in wireless networks. By formulating the localization problem as a factor graph and applying the sum-product algorithm, we obtained a theoretical solution for the a posteriori marginal distribution of each node's location. This theoretical algorithm was then adapted in a practical, distributed localization algorithm in which nodes cooperate by broadcasting messages. Several details related to algorithm's implementation were explored, including the incorporation of practical information and the representation of messages.

We demonstrated that ultra-wideband radios are capable of achieving high-resolution ranging measurements. Through an extensive measurement campaign in multiple environments, we developed realistic yet tractable models of UWB ranging error. These models were incorporated into a series of simulations that characterized the performance of cooperative localization. Simulation results showed that cooperation increases the accuracy and robustness of localization.

There are several possibilities for future research directions. First, convergence of the algorithm is of great importance. Specifically, knowing the

conditions under which the algorithm (a) converges and (b) converges to the correct distribution would be useful in assessing the algorithm's reliability. Additionally, a comprehensive study of the algorithm's numerical parameters (e.g. the number of samples R and the number of iterations) may reveal how the performance improves with increasing computation or complexity. The effects of unreliable transmissions and interfering communication traffic are practical considerations that should be taken into account in future work. Another area of interest is anchor placement; for example, how anchors should be deterministically placed in the environment to optimize the localization of agents. Finally, the cooperative localization algorithm can be extended to account for mobile networks, potentially compromised nodes, and other realistic scenarios. After showing that cooperation improves localization both in theory and in practice, we believe the full potential of cooperative localization has yet to be explored.

Bibliography

[1] A. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, 2005.

[2] J. J. Caffery and G. L. Stuber, "Overview of radiolocation in CDMA cellular systems," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 38–45, Apr. 1998.

[3] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications.* New York, NY, USA: ACM Press, 2002, pp. 88–97.

[4] K. Pahlavan, X. Li, and J. P. Makela, "Indoor geolocation science and technology," *IEEE Communications Magazine*, vol. 40, no. 2, pp. 112–118, Feb. 2002.

[5] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, Aug. 2003.

[6] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 4, pp. 2652–2661, Mar. 2004.

[7] N. Patwari, J. N. Ash, S. Kyperountas, A. O. I. Hero, R. L. Moses, and N. S. Correal, "Locating the nodes: cooperative localization in wireless sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, July 2005.

[8] J.-Y. Lee and R. A. Scholtz, "Ranging in a dense multipath environment using an UWB radio link," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 9, pp. 1677–1683, Dec. 2002.

[9] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements," *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 6, 2003.

[10] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 70–84, July 2005.

[11] N. Patwari and A. Hero III, "Using Proximity and Quantized RSS for Sensor Localization in Wireless Networks," *IEEE/ACM 2nd Workshop on Wireless Sensor Nets. & Applications*, pp. 20–29, 2003.

[12] D. Niculescu and B. Nath, "Ad hoc positioning system (APS) using AOA," *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 3, pp. 1734–1743, Mar./Apr. 2003.

[13] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks," in *Proceedings of the General Track: 2002 USENIX Annual Technical Conference*. Berkeley, CA, USA: USENIX Association, 2002, pp. 317–327.

[14] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 7, no. 5, pp. 28–34, 2000.

[15] D. B. Jourdan, J. J. J. Deyst, M. Z. Win, and N. Roy, "Monte carlo localization in dense multipath environments using UWB ranging," in *Ultra-Wideband, 2005. ICU 2005. 2005 IEEE International Conference on*, Sept. 2005, pp. 314–319.

[16] R. Bajaj, S. L. Ranaweera, and D. P. Agrawal, "GPS: location-tracking technology," *Computer*, vol. 35, no. 4, pp. 92–94, Apr. 2002.

[17] Y. Zhao, "Standardization of mobile phone positioning for 3g systems," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 108–116, July 2002.

[18] J. Hightower and G. Borriello, "Location systems for ubiquitous computing," *Computer*, vol. 34, no. 8, pp. 57–66, Aug. 2001.

[19] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, "The cricket location-support system," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM Press, 2000, pp. 32–43.

[20] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," vol. 2, Tel Aviv, Israel, 2000, pp. 775–784.

[21] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking.* New York, NY, USA: ACM Press, 2001, pp. 166–179.

[22] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks," *Telecommunication Systems*, vol. 22, no. 1, pp. 267–280, 2003.

[23] N. Alsindi, K. Pahlavan, and B. Alavi, "A Novel Cooperative Localization Algorithm for Indoor Sensor Networks," *Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on*, pp. 1–6, 2006.

[24] B. Denis and N. Daniele, "NLOS ranging error mitigation in a distributed positioning algorithm for indoor UWB ad-hoc networks," in *International Workshop on Wireless Ad-Hoc Networks*, May/June 2004, pp. 356–360.

[25] X. Nguyen, M. Jordan, and B. Sinopoli, "A kernel-based learning approach to ad hoc sensor network localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 1, no. 1, pp. 134–152, 2005.

[26] L. Fang, W. Du, and P. Ning, "A beacon-less location discovery scheme for wireless sensor networks," *IEEE Conference on Computer Communications (Infocom)*, 2005.

[27] V. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, vol. 2, no. 3, pp. 24–33, July/Sept. 2003.

[28] S. I. Roumeliotis and G. A. Bekey, "Collective localization: a distributed kalman filter approach tolocalization of groups of mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, San Francisco, CA, USA, 2000, pp. 2958–2965.

[29] A. Howard, M. Mataric, and G. Sukhatme, "Localization for mobile robot teams: A distributed MLE approach," *Experimental Robotics VIII*, vol. 5, pp. 146–155, 2003.

[30] A. T. Ihler, J. W. I. Fisher, R. L. Moses, and A. S. Willsky, "Nonparametric belief propagation for self-localization of sensor networks," vol. 23, no. 4, Apr. 2005, pp. 809–819.

[31] L. Yang and G. Giannakis, "Ultra-wideband communications: an idea whose time has come," *IEEE Signal Processing Magazine*, vol. 21, no. 6, pp. 26–54, 2004.

[32] M. Z. Win and R. A. Scholtz, "On the robustness of ultra-wide bandwidth signals in dense multipath environments," *IEEE Communications Letters*, vol. 2, no. 2, pp. 51–53, Feb. 1998.

[33] M. Win, R. Scholtz, and M. Barnes, "Ultra-wide bandwidth signal propagation for indoor wirelesscommunications," *IEEE International Conference on Communications*, vol. 1, 1997.

[34] N. Alsindi, X. Li, and K. Pahlavan, "Performance of TOA estimation algorithms in different indoor multipath conditions," *IEEE Wireless Communications and Networking Conference*, vol. 1, pp. 495–500, Mar. 2004.

[35] S. Gezici, H. Kobayashi, and H. V. Poor, "Nonparametric nonline-of-sight identification," *IEEE 58th Vehicular Technology Conference*, vol. 4, pp. 2544–2548, Oct. 2003.

[36] I. Guvenc, C.-C. Chong, and F. Watanabe, "NLOS identification and mitigation for UWB localization systems," in *IEEE Wireless Communications and Networking Conference*, Kowloon, China, Mar. 2007, pp. 1571–1576.

[37] Y. Qi, H. Kobayashi, and H. Suda, "Analysis of wireless geolocation in a non-line-of-sight environment," *IEEE Transactions on Wireless Communications*, vol. 5, no. 3, pp. 672–681, 2006.

[38] K. Yu and I. Oppermann, "Performance of UWB position estimation based on time-of-arrival measurements," in *International Workshop on Ultra Wideband Systems. Joint UWBST and IWUWBS 2004.*, Oulu, Finland, May 2004, pp. 400–404.

[39] B. Denis, L. Ouvry, B. Uguen, and F. Tchoffo-Talom, "Advanced bayesian filtering techniques for UWB tracking systems in indoor environments," in *2005 IEEE International Conference on Ultra-Wideband*, Sept. 2005.

[40] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[41] Y. Weiss and W. T. Freeman, "Correctness of Belief Propagation in Gaussian Graphical Models of Arbitrary Topology," *Neural Computation*, vol. 13, no. 10, pp. 2173–2200, 2001.

[42] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *IEEE International Conference on Communications: Technical Program*, vol. 2, Geneva, Switzerland, May 1993, pp. 1064–1070.

[43] B. Alavi and K. Pahlavan, "Modeling of the TOA-based distance measurement error using UWB indoor radio measurements," *IEEE Communications Letters*, vol. 10, no. 4, pp. 275–277, Apr. 2006.

[44] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[45] A. Doucet, S. Godsill and C. Andrieu, "On sequential monte carlo sampling methods for bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[46] P.M. Djuric, J.H. Kotecha, Jianqui Zhang, Yufei Huang, T. Ghirmai, M.F. Bugallo, J. Miguez, "Particle filtering," *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[47] J. Schiff and D. Antonelli, "In-network nonparametric loopy belief propagation on sensor networks for ad-hoc localization," *University of California, Berkeley*, 2006.

[48] R. Moses and R. Patterson, "Self-calibration of sensor networks," *SPIE vol. 4743: Unattended Ground Sensor Technologies and Applications IV*, vol. 4743, 2002.

[49] Time Domain Corporation, *P210 Reference Design.*

[50] C. Gentile and A. Kik, "A comprehensive evaluation of indoor ranging using ultra-wideband technology."

[51] B. Denis, J.Keignart, and N. Daniele, "Impact of NLOS propagation upon ranging precision in UWB systems," in *Proceedings of the Conference on Ultra Wideband Systems and Technology*, Nov. 2003, pp. 379–383.

[52] D. Cassioli, M. Win, and A. Molisch, "The ultra-wide bandwidth indoor channel: from statistical model to simulations," *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 6, pp. 1247–1257, 2002.

[53] A. Molisch, K. Balakrishnan, D. Cassioli, C. Chong, S. Emami, A. Fort, J. Karedal, J. Kunisch, H. Schantz, and K. Siwiak, "A comprehensive model for ultrawideband propagation channels," *submitted to IEEE Trans. Antennas Prop*, 2006.

[54] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, Nov. 1996.

[55] J. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," *manuscript, International Computer Science Institute*, 1998.