LOCATION ESTIMATION IN SENSOR NETWORKS

by

Neal Patwari

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in The University of Michigan
2005

Doctoral Committee:

Professor Alfred O. Hero III, Chair
Professor Susan Murphy
Professor Atul Prakash
Professor Demosthenis Teneketzis

ACKNOWLEDGEMENTS

This work would not have been possible without the support of my colleagues, friends, and mentors. Specifically, I would like to thank my advisor, Alfred Hero, who has given me all the support and guidance I needed as a doctoral student. I am very grateful to have had his trust in my ability, and I have often benefitted from his insight and advice. I would also like to thank the excellent teachers from whom I have had the benefit of learning from, in particular, my committee members. I also owe my committee a great deal of thanks for their time and advice, which is reflected in this thesis.

This thesis was also made possible by collaboration and consultation with many colleagues. I would like to thank Norm Adams, Doron Blatt, Jose Costa, Panna Felsen, Adam Pacholski, David Thorsley, Yi Wang, and Jionglin Wu at the University of Michigan; Brian Sadler and Ananthram Swami at the Army Research Lab; Mark Crovella and Anukool Lakhina at Boston University; Greg Durgin at Georgia Tech; Radha Poovendran at the University of Washington, Seattle; and Paul Barford, Rob Nowak, and Mike Rabbat at the University of Wisconsin.

I owe a huge gratitude to my former colleagues and collaborators at Motorola Labs: Anthony Allen, Monique Bourgeois, Neiyer Correal, Spyros Kyperountas, Danny McCoy, Matt Perkins, Qicai Shi, and Yanwei Wang. Special thanks to Bob O'Dea, who spearheaded wireless sensor network systems research at Motorola Labs, who supported my return to grad school, and who has generously provided access and

permission to report measured data in this thesis.

In particular, collaborative work with Alfred Hero, Josh Ash, Randy Moses, Spyros Kyperountas, and Neiyer Correal led to publication of the article, "Locating the Nodes" in the July 2005 IEEE Signal Processing Magazine, parts of which are dispersed throughout this thesis. Further, collaboration with Jose Costa led to publication of two papers on the distributed weighted multi-dimensional scaling method for sensor localization, part of which is presented in Section 4.4.

Finally, I would like to acknowledge my friends and family, who have supported me through these efforts.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

CHAPTER I

Introduction

Networks, both wired and wireless, have for decades been growing in size and complexity. The number of computers connecting the global Internet are counted in the billions; wireless sensor networks of thousands or tens of thousands of nodes have been proposed, and these numbers will continue to grow over time as wireless sensors become cheaper and more ubiquitous. The huge size of these networks makes human network moderation and monitoring extraordinarily difficult. In the global Internet, there is no single entity that is able to control or even monitor the entire internetwork. In wireless networks, the size and the ad-hoc, even temporary nature of the network, precludes requiring a human administrator to set up each node in the network. In both wired and wireless cases, there is a great need for automated network self-configuration and monitoring, and this need will grow over time as networks grow larger.

The network self-configuration and self-monitoring problem of estimating the *location* of *sensors* is considered in this thesis. The use of both terms 'location' and 'sensor' must be explained to understand the generality of the problem considered.

First, 'sensor' is used to refer to a device which is connected, via wire or RF communication medium, to other devices in the network. A sensor is also referred to

as a 'node' in a graph with edges that represent the pair-wise communications possible in the network. This representation of a sensor network is depicted in Fig. 1.1. In this framework, each node measures some quantity relating either to its environment, or to the operation of the node itself, such as the traffic level passing through the node. Whether or not this measurement requires a physical sensor integrated circuit (IC) or software embedded in the node, this measurement is said to be made by a 'sensor'. The sensor is central to the purpose and operation of the network, and because of its importance, in this thesis, the word 'sensor' is used to refer to the entire device. Thus *sensor network* generally refers to any communication network in which each node, device, or router, measures some local quantity of interest.

Second, *location* refers to coordinates which describe each node in the network. These location coordinates to be estimated may be either *physical location*, or *data location*:

- *Physical Location*: The sensor's physical coordinates, that is, where it exists in space, are estimated.
- *Data Location*: Non-physical coordinates for the sensor's measured data are estimated. These coordinates describes where, in a particular space, the measured data vector lies.

Estimating a sensor's physical coordinates is intuitively important: when sensor data is reported, it should be accompanied with an indication of where in space that data was recorded. Data location coordinates are made necessary by the bandwidth and energy limitations of a network. In such sensor networks, large quantities of sensor data are recorded and are available, but typically are not sent through the network, due to the communication constraints. Instead, summary statistics may be typically

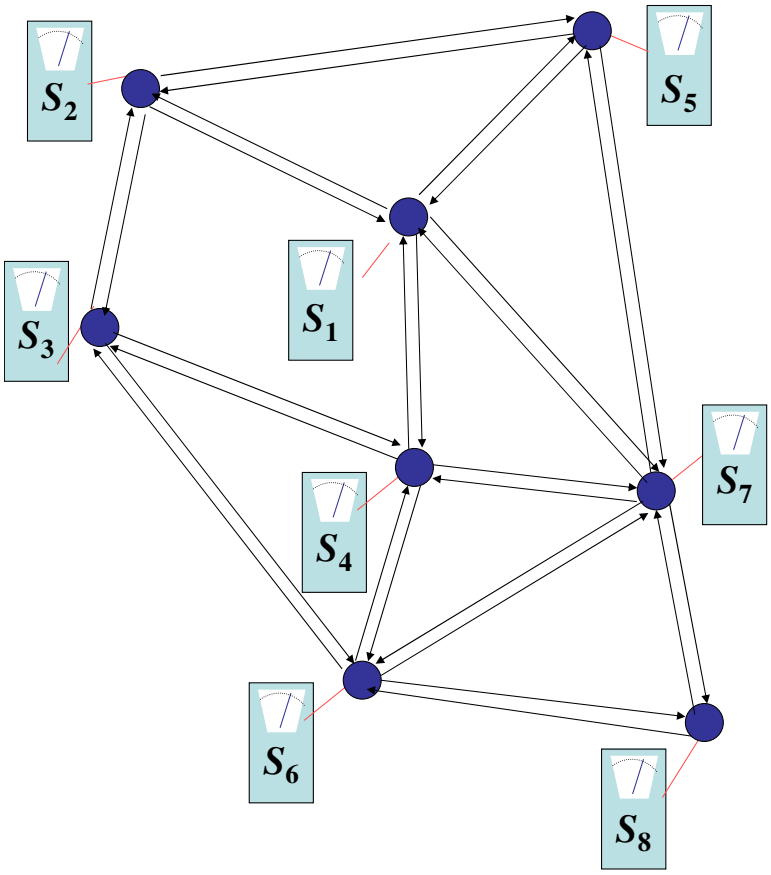


Figure 1.1: This thesis considers sensor networks, in which nodes (◯) have wired or wireless communication channels, shown as arrows, with some other nodes. Each node also has a sensor, shown as $S_1, \ldots, S_8$, which may measure the physical environment, or the node itself.

communicated, and occasionally when the situation requires it, full data is sent. Data coordinates are a summary statistic, a lower-dimensional representation of the full data available, which preserve information about the relationships between sensors' data.

To estimate these physical or data location coordinates, two types of sensor measurements are used in this thesis. First, *pair-wise measurements* are considered, in which two sensors measure some quantity which directly relates to the relative locations of the two sensors. For example, if one sensor measures the received signal strength (RSS) of the transmission made by a second sensor, and the RSS is known to decay with distance, this is a pair-wise measurement which gives some indication of relative location between sensors.

Second, *sensor data measurements* are considered, which refer to measurements made at a *single* sensor of the environment near itself. For example, if a sensor is placed at each city in the U.S. to measure the daily temperature, humidity, and rainfall over the course of a year, the sensor is recording a sensor data measurement. The sensor data measurement is taken at a single location; however, cities which are physically close to one another, *eg.*, Ann Arbor and Detroit, will be likely to have very correlated sensor data measurements. In general, in random fields correlated over space and time, the sensor data measurement also provides some information about the relative location of sensors.

In summary, this thesis considers *wired* and *wireless* networks comprised of sensors which make *pair-wise* or *sensor data* measurements, which are then used to estimate *physical* or *data* location coordinates. This unified approach allows these topics to be studied together as the problem of location estimation in sensor networks.

To introduce the particular issues which motivate studying these topics, in the

next sections, wireless sensor networks are introduced in Section 1.1 and the need for data localization in wired networks is introduced in Section 1.2. Following this, the location estimation problem in sensor networks is formally stated in Section 1.3. The specific contributions presented in this thesis are listed in Section 1.4, and Section 1.5 presents the outline and the table of notation used in this thesis.

1.1 Wireless Sensor Networks

Dramatic advances in radio frequency (RF) and micro-electro-mechanical systems (MEMS) IC design have made possible the use of large networks of wireless sensors for a variety of new monitoring and control applications. For example, smart structures will actively respond to earthquakes to make buildings and bridges safer, and constantly monitor for cracks or structural problems [73]. Precision agriculture will reduce costs and environmental impact by watering and fertilizing exactly where necessary, and will improve quality by monitoring storage conditions after harvesting [102, 39]. Condition-based maintenance will direct equipment servicing exactly when and where needed based on data from wireless sensors. Traffic monitoring systems will better control stoplights and inform motorists of alternate routes in case of traffic jams. Environmental monitoring networks will sense air, water and soil quality and identify the source of pollutants in real time. A wide variety of such applications have been enabled by the promise of inexpensive networks of wireless sensors, as described in review articles in [1, 26, 38, 95].

Automatic estimation of physical location of the sensors in these wireless networks is a key enabling technology. The overwhelming reason is that a sensor's location must be known for its data to be meaningful. If a system is set up to respond locally to changes in sensor data, then it must know where those changes are occurring. In

many cases, location itself is the data that needs to be sensed - localization can be the driving need for wireless sensor networks in applications such as warehousing and manufacturing logistics, in which radio tagged parts and equipment must be able to be accurately located at all times. Also, sensor location information, if it is accurate enough, can be extremely useful for scalable, 'geographic' routing algorithms.

To make these applications viable with possibly vast numbers of sensors, device costs will need to be low (from a few dollars to a few cents depending on the application), sensors will need to last for years or even decades without battery replacement, and the network will need to organize without significant human moderation. Traditional physical localization techniques are not well suited for these requirements. Including GPS on each device is cost and energy prohibitive for many applications, not sufficiently robust to jamming for military applications, and limited to outdoor applications. Local positioning systems (LPS) [125] rely on high-capability base stations being deployed in each coverage area, an expensive burden for most low-configuration wireless sensor networks.

Instead, this thesis considers the problem in which some small number m of sensors, *reference nodes*, obtain their coordinates - either via GPS, or from a system administrator during startup - and the rest, n *unknown-location nodes*, must determine their own coordinates. If sensors were capable of high-power transmission, they would be able to make measurements to multiple reference nodes, and positioning techniques such as multi-lateration or multi-angulation could be applied. These direct techniques have been studied for decades within and outside of the signal processing research community [121]. However, low-capability, energy-conserving devices will not include a power amplifier, will lack the energy necessary for long-range communication, or may be limited by regulatory constraints on transmit power. Instead,

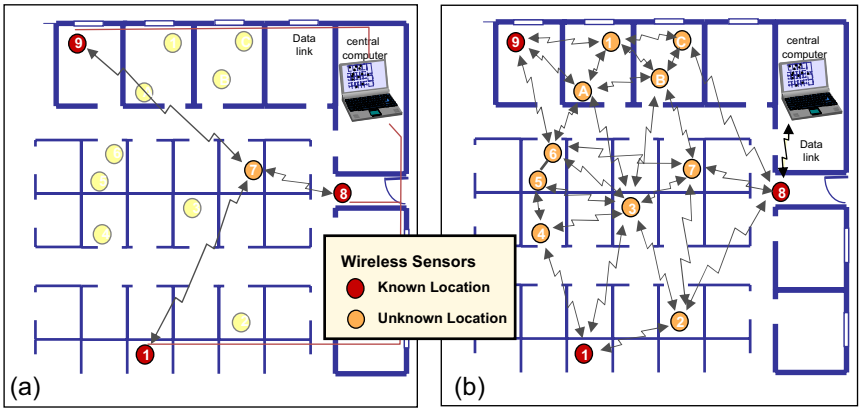


Figure 1.2: In cooperative localization (b), measurements made between any pairs of sensors can be used to aid in the location estimate. Traditional multi-lateration or multi-angulation (a) is a special case in which measurements are made only between an unknown-location sensor and known-location sensors.

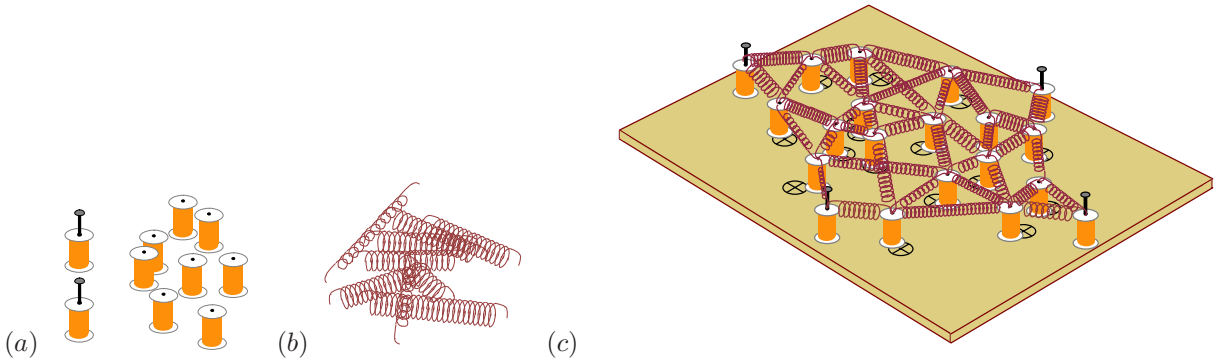


Figure 1.3: Cooperative localization is analogous to finding the resting point of (a) masses (spools of thread) connected by a network of (b) springs. First, reference nodes are nailed to their known coordinates on a board. Springs have a natural length equal to measured ranges and can be compressed or stretched. They are connected to the pair of masses whose measured range they represent. After letting go, the equilibrium point (c) of the masses represent a minimum-energy localization estimate; the actual node locations are indicated by $\otimes$.

wireless sensor networks, and thus localization techniques, will be multi-hop (a.k.a. 'cooperative' localization), as shown in Fig. 1.2. Rather than solving for each sensor's position one at a time, a location solver, analogous to the system of masses connected by springs shown in Fig. 1.3, will estimate all sensor positions simultaneously.

Such localization systems are an extension of techniques used in or proposed for wireless local area networks (WLAN) and cellular mobile station (MS) location, as described elsewhere in this issue. Unknown-location devices are still allowed to make measurements with known-location references, but in cooperative localization, they additionally are allowed to make measurements with other unknown-location devices. The additional information gained from these measurements between pairs of unknown-location devices enhances the accuracy and robustness of the localization system. In the considerable literature, such systems have alternatively been described as 'cooperative', 'relative', 'distributed', 'GPS-free', 'multi-hop', or 'network' localization; 'self-localization'; 'ad-hoc' or 'sensor' positioning; or 'network calibration'. In this article, 'cooperative' localization [105] is used to emphasize the communication and measurements between many pairs of sensors required to achieve localization for all sensors.

1.1.1 Motivating Application Example: Animal Tracking

If cooperative localization can be implemented in wireless networks as described above, many compelling new applications can be enabled. For the purposes of biological research and animal behavior studies, it is very useful to track animals over time and over very wide spaces [3, 61]. Such tracking can answer questions about animal behavior and interactions within their own and with other species. Using current practices, tracking is a very difficult, expensive process, and requires bulky tags that rapidly run out of energy. A typical practice is to attach VHF transmitter

collars to animals to be tracked, and then triangulate their location by driving (or flying) to various locations with a directional antenna. Alternatively, GPS-based collars can be used, but are limited by cost concerns, and offer only a short lifetime due to high energy consumption. Using wireless sensor networks can dramatically improve the abilities of biological researchers (as demonstrated by 'ZebraNet' [61]). Using multi-hop routing of location data through the sensor network enables low transmit powers from the animal tags. Furthermore, inter-animal distances, which are of particular interest to animal behaviorists, can be estimated using pair-wise measurements and cooperative localization methods, without resorting to GPS. The end result of the longer battery lifetimes is less frequent re-collaring of the animals being studied.

1.1.2 Motivating Application Example: Logistics

As another example, consider deploying a sensor network in an office building, manufacturing floor, and warehouse. Sensors already play a very important role in manufacturing. Monitoring and control of machinery has traditionally been wired, but making these sensors wireless reduces the high cost of cabling and makes the manufacturing floor more dynamic. Automatic localization of these sensors further increases automation.

Also, boxes and parts to be warehoused and factory and office equipment can all be tagged with sensors when first brought into the facility. These sensors monitor storage conditions (temperature, humidity) and help control the HVAC system. Sensors on mobile equipment report their physical location when the equipment is lost or needs to be found (*e.g.* during inventory), and even contact security if they are about to 'walk out' of the building. Knowing where parts and equipment are when they are critically needed reduces the need to have duplicates as back-up, savings

which could pay for the wireless sensor network itself.

Radio-frequency identification (RFID) tags, such as those now required by Walmart on pallets and cartons entering in its warehouses [65], represent a first step in warehouse logistics. RFID tags are only located when they pass within a few feet of a reader, thus remaining out of access most of their time in the warehouse. Networked wireless sensors, however, can be queried and located as long as they are within range (on the order of 10 m) of the closest other wireless sensor.

Costs of Commercial Logistics Solutions

Local positioning systems (LPS) are also proposed for logistics applications. For LPS, sensors are active, using signals transmitted to or received from high-capability base stations to locate them. One issue with LPS is the cost of deploying base stations which cover an area of interest.

For example, a company called Detection Systems, Inc. (now owned by Bosch Security Systems) deployed a LPS for a personal security application system on a small college campus. The base stations use measured RSS to calculate and report the location of a radio tag when its 'Alarm' button is pushed, so that police can be dispatched to the exact location of the person requesting assistance [10]. The system was successful in terms of meeting the accuracy requirements of the campus police, and had a demonstrated track record of success. However, the system was a test system, for which Detection Systems bore the $400,000 cost of deployment. It was probably this high cost of deployment which limited the adoption of this system at other campuses.

Another LPS is sold by WhereNet Corp., which is marketed as an "active RFID" solution. WhereNet has agreements with automobile manufacturers (including Ford Motor) to provide localization solutions for its logistics or "supply-chain inventory

visibility" applications. WhereNet's active tages transmit signals to multiple base stations, which locate them based on time-difference of arrival measurements. It has been reported that WhereNet's system suffers in heavy multipath environments like manufacturing floors. The cost of a WhereNet deployment, according to WhereNet, is around $350,000 to $500,000 for a 1 million square foot warehouse and 2,000 assets to track [78]. This price tag, about $175 to $250 per asset, is high compared to what a cooperative localization system would cost per asset, because cooperative localization systems would not require extensive installed infrastructure.

In addition, the theoretical accuracy of cooperative localization increases with the density of sensors. This is shown in Section 3.4 for a general class of location estimation problems. This is analogous to Metcalfs Law, which holds that the value of a network increases with the number of nodes. Thus, having heterogeneous sensors of varied purposes, all participating in the same network, will help drive localization errors down.

1.1.3 Pair-wise and Sensor Data Measurements

Sensors may have be able to measure various pair-wise or sensor data measurements. Wireless sensor networks are generally envisioned for applications which monitor wide areas for changes in the physical environment. Thus recording sensor data over time is a primary requirement of sensors; using the same sensor data measurement to extract location information would be a desirable extra feature. If the field being measured is correlated over space, and the correlation is isotropic, *i.e.*, independent of direction, then sensor data measurements over time provide information about which sensors are likely to be physically close to other sensors. Such information can be used to get a coarse estimate of physical coordinates. This problem is considered in Section 4.6.

While such use of sensor data measurements is possible in some applications, it is restricted due to its requirement of an isotropic correlated random field. More generally, pair-wise measurements are used for physical location estimation, in which one sensor measures the signal transmitted from a second sensor. Pair-wise measurements include the measurement of signal properties such as angle-of-arrival (AOA), time-of-arrival (TOA), or received signal strength (RSS). These measurements might be made using acoustic or RF signals. One contribution of this thesis is to report extensive pair-wise measurements of TOA and RSS in RF channels for wireless sensor networks, and models derived from those measurements, in Chapter II.

1.2 Internet Data Localization

The task of monitoring the Internet for anomalies such as worm outbreaks, denial-of-service attacks, and intrusion attempts, involves the processing of large quantities of data recorded at many nodes across the network. The sensor network framework is applicable here because software on routers and other computers (firewalls, end users, etc.) in the Internet can measure varying quantities of interest which might be affected when an anomalous event occurs. The huge quantity of information available (the header and data of all packets recorded in transit) is far greater than can be transferred to a central location for processing; furthermore, privacy concerns preclude setting up a system to record full traffic data. The decentralized reduction of the huge quantity and high-dimensionality of internet traffic data is a significant challenge which research must overcome in order to detect anomalous events on the internet with high reliability and with a low rate of false-alarms.

Estimating data location is the means considered in this thesis to reduce the dimensionality of traffic data recorded at each node on the network while preserving

information about the relationships between data recorded at different nodes. In this problem, considered in Chapter V, sensor data measurements are made at different routers on a wired Internet backbone network, in order to estimate 2-D data location coordinates which, plotted on a map, provide information about the relationships between the different backbone routers. Such maps show dramatic spatial changes when anomalous events occur on the backbone. This data localization problem is also referred to as 'cooperative' localization, since routers communicate and make calculations with the other nearby routers in order to calculate a network-wide set of data location coordinates.

1.3 Problem Statement

Before going into detail, it is useful to formally state the cooperative sensor location estimation problem. Throughout this thesis, 'cooperative' localization is considered as shown in Figs. 1.2 and 1.3. In every problem considered in this thesis, a network of N total sensors is considered. In the network, the objective is to estimate the coordinates of n of the sensors, given *a priori* the coordinates of m of the sensors, where $N = n + m$. In other words, for the 2-D localization problem, a total of $2n$ unknown-location node parameters must be estimated, $\boldsymbol{\theta} = [\boldsymbol{\theta}_x, \boldsymbol{\theta}_y]$, where

$$\boldsymbol{\theta}_x = [x_1, \ldots, x_n], \quad \boldsymbol{\theta}_y = [y_1, \ldots, y_n] \tag{1.1}$$

given the known reference coordinates $[x_{n+1}, \ldots, x_{n+m}, y_{n+1}, \ldots, y_{n+m}]$, and at least one of a variety location measurements. The location of sensor i is also referred to as $\mathbf{z}_i$ where $\mathbf{z}_i = [x_i, y_i]^T$. While the 2-D case is considered in this thesis, extension to 3-D appends a third coordinate to each sensor location vector [44].

Measurements can be either *pair-wise* or *sensor data*. Pair-wise measurements $X_{i,j}$ could be any physical reading that indicates distance or relative position, eg.

time-of-arrival (TOA), angle-of-arrival (AOA), received signal strength (RSS), or connectivity (whether or not two devices can communicate). Alternatively, a sensor data measurement at sensor i at time t is denoted $v_i(t)$. Both types of measurements can contain, directly or indirectly, information about the relative location of the sensors in the network.

In the case of pair-wise measurements, it is not assumed that all $\binom{N}{2}$ pairs make measurements. Let the set $H(i)$ be the set of sensors with which sensor i makes measurements. Clearly, $i \notin H(i)$, and $H(i) \subset \{1, \ldots, n+m\}$. Note that these pair-wise measurements could be done via different modalities - eg. RF, infrared (IR), acoustics [77, 6], or a combination of these [47]. Finally, TOA can be measured using different signaling techniques, such as direct-sequence spread-spectrum (DS-SS) [75, 96] or ultrawideband (UWB) [25, 40, 41]. These measurement methods are discussed in Chapter II. The main message of Chapter II is that these pair-wise measurements are adversely affected by the physical environment, which cannot be changed. Since the exact layout of the environment in which the sensor network is deployed determines the pair-wise measurements, and every area's layout can't *a priori* be known, these environmental effects are considered to be random, and statistical models are created for the pair-wise measurements.

In addition to the above problem statement applicable throughout this thesis, there are some additional extensions which will be noted as needed. We list them here to survey the scope of the estimation problems considered.

1.3.1 Imperfect Prior Knowledge

For some applications, nodes have some, but imperfect *a priori* coordinate knowledge. For these nodes, it is desirable to estimate the node coordinate taking into account its imperfect prior information. When we consider imperfect prior knowledge,

we assume that the first n sensors have some finite (not perfect) location information. We consider this generalization as an additional capability of the distributed weighted multi-dimensional scaling (dwMDS) algorithm presented in Section 4.4. The appropriate performance bounds in the case of imperfect prior information are mean-squared error (MSE) bounds, which have have been considered in [77, 80].

1.3.2 Solely Relative Location

Other localization research has focused on truly 'relative' location, *i.e.*, when no references exist ($m = 0$), and an arbitrary coordinate system can be chosen. For example, in geographic routing, only distances and angles between nodes are important, so no absolute coordinates are necessary. This is often called 'beacon-free' sensor localization [116]. In this thesis, beacon-free localization is explicitly considered using the dwMDS method in Section 4.4, and is a straightforward extension of the manifold learning-based algorithms considered in Sections 4.3, 4.5, and 4.6.

1.4 Contributions

The following lists particular novel contributions of my thesis research, which part of the thesis they are covered, and any publications reporting them.

- **Models**: Measurement-verified statistical models of pair-wise measurements of TOA, RSS, and connectivity in RF channels (Chapter II and [93, 87, 92])
- **Measurements**: Pair-wise RF channel TOA and RSS measurements in wireless sensor networks operating in a variety of environments (Section 2.6 and [93, 94, 87, 92])
- **Performance Bounds**: Lower bounds on estimator variance for a variety of pair-wise measurement modalities, derived from the appropriate models (Chap-

ter III and [87, 92, 88]), and for a sensor data measurements given a particular random field model (Chapter 4.6 and [89])

- **Theory**: Proof that increasing the number of sensors in the network, given pair-wise measurements and the specific sufficient conditions listed, will strictly decrease the lower bound on average estimator variance (Theorem III.5, Appendix A.2, and [92])

- **Algorithms**: Introduction and testing of the distributed weighted multi-dimensional scaling (dwMDS) localization algorithm (Section 4.4 and [27, 28]) and a Laplacian Eigenmap-based localization algorithm (Section 4.5 and [90])

- **Adaptive Methods**: Development of two adaptive neighbor selection algorithms to reduce sensor localization bias, in particular, caused by neighbor selection in noise (Sections 4.4.3 and 4.5.3, and [27, 28, 90])

- **Physical Localization from Sensor Data Measurements**: Application of manifold learning methods to estimate physical sensor location coordinates based on sensor data measurements (Section 4.6 and [89])

- **Internet Traffic Visualization**: Developed tools which use sensor data measurements to estimate 2-D data location coordinates for visualization of extremely high-dimensional space-time Internet traffic, for the purposes of detecting traffic anomalies (Chapter V and [91])

1.5 Outline of Thesis

The goal of this thesis is to explore the sensor location estimation problem from a statistical signal processing perspective. First, Chapter II introduces various pairwise measurement modalities, presents statistical models which describe them, and

the measurement campaigns conducted to build and verify the models. From these pair-wise measurement models, Cramér-Rao bounds are derived in Chapter III which lower bound the covariance achievable by any unbiased location estimator. Next, in Chapter IV, particular location coordinate estimators are introduced and compared to the lower bound. A general review of localization algorithms in the literature is presented. Then, manifold learning is introduced, and three new manifold learning-based localization algorithms are introduced and evaluated. The last of those three new algorithms uses sensor data, rather than pair-wise measurements, as relative location information. Then, in Chapter V, sensor data localization is applied to a sensors on the backbone routers of Abilene, an internet backbone network.

For convenience, a table of notation used through the thesis is included in Table 1.1.

Notation	Description
$N = n + m$	Total number of sensors
n	Number of sensors with imperfect or no *a priori* coordinate information
m	Number of sensors with perfect *a priori* coordinate knowledge ('anchor' nodes)
$\mathbf{z}_i$	Actual length D coordinate vector of sensor i, $i = 1 \dots n + m$
$P_{i,j}$	Power received (dBm) of signal at sensor i transmitted by sensor j
Δ_0, Π_0	Π_0 is free-space received power at reference distance Δ_0 (typically 1m)
n_p	RF Path loss exponent
σ_{dB}	Standard deviation of RSS error in (dB)
$T_{i,j}$	Time delay between transmission of signal by sensor j and arrival at sensor i
v_p	Speed of propagation (m/s)
μ_T	Mean of time delay error in (s)
σ_T	Standard deviation of time delay error in (s)
$A_{i,j}$	Angle-of-arrival of signal at sensor i transmitted by sensor j
σ_α	Standard deviation of AOA error in (rad)
$Q_{i,j}$	Connectivity (a.k.a. proximity), 1 if i in range of j, 0 otherwise
P_0	Minimum received power for successful reception
d_0	Distance at which mean received power = P_0
R	Threshold distance for neighborhood selection (often = d_0).
$v_i(t)$	Sensor data measurement taken by sensor i and time t
τ	Time duration during which sensors record data
$\mathbf{v}_i$	Data measurement vector, $\mathbf{v}_i = [v_i(1), \dots v_i(\tau)]$
$\delta_{i,j}$	Measured / estimated distance between sensors i and j
$\hat{\mathbf{z}}_i$	Estimated coordinate of sensor i, $i = 1 \dots n$
Z	Actual coordinate matrix, $[\mathbf{z}_1, \dots, \mathbf{z}_N]$
S	Global objective function to be minimized
S_i	Local objective function to be minimized at sensor $i = 1 \dots n$
$\mathbf{z}_i^{(k)}$	Estimated coordinates of sensor i at iteration k
$\mathcal{N}_i$	Set of sensor numbers which are considered neighbors of i
$w_{i,j}$	Weight given to the measured range between sensors i and j

Table 1.1: Symbols used through thesis.

CHAPTER II

Localization Measurements and Models

The key to developing reliable localization systems which use pair-wise measurements is to accurately represent the severely degrading effects of the channel in which the pair-wise measurements are made. Propagation of RF signals in real-world environments, full of obstructions, reflectors, people and objects in motion, make this representation challenging. This chapter discusses extensive RF measurement campaigns the author has conducted in order to characterize localization measurements. Models are presented and tested using the measured data. These measurements, and others from the literature that are referenced, allow the formation of statistical models for TOA, RSS, AOA, QRSS, and connectivity measurements, which are introduced in Sections 2.2 through 2.5. These models form the basis for the later chapters of this thesis.

Generally, range and angle measurements used for localization are impacted by both *time-varying errors* and *static, environment-dependent errors*. Time-varying errors (e.g. due to additive noise and interference) can be reduced by averaging multiple measurements over time. Environment-dependent errors are the result of the physical arrangement of objects (e.g. buildings, trees, and furniture) in the particular environment which the sensor network is operating. Since the environment

is unpredictable, these errors are unpredictable and must be modeled as random. However, in a particular environment, objects are predominantly stationary, and thus for a network of mostly stationary sensors, environment-dependent errors will be largely constant over time.

The majority of applications of wireless sensor networks involve mostly stationary sensors. Because some time delay is acceptable in these applications, each pair of sensors will make multiple measurements over time and average the results together to reduce the impact of time-varying errors. This thesis creates a model of the statistics of pair-wise measurements *after* time-averaging, in order to determine the performance of localization in wireless sensor networks.

Section 2.1 begins by discussing the general methodology and goals of these measurement experiments. In order to present complete coverage of this variety of possible pair-wise measurement modalities, parts of the description of the models presented in this chapter were written collaboratively by the author and A.O. Hero at the University of Michigan; J.N. Ash and R.L. Moses at Ohio State University; and S. Kyperountas and N.S. Correal at Motorola Labs, and initially published in [86].

2.1 Measurement Characterization

Ideally, statistical characterization of sensor network measurements would proceed as follows: deploy K wireless sensor networks, each with N sensors positioned with the identical geometry in the same type of environment, but each network in a different place. For example, one might test a sensor network deployed in a grid, in K different office buildings. In each deployment, make many measurements between all possible pairs of devices. Repeat each pair-wise measurement over a short time period and compute the time-average. Then, the joint distribution (conditional on

the particular sensor geometry) of the time-averaged measurements could be characterized. To our knowledge, no such wide-scale pair-wise measurements have been attempted, due to the huge scale of the task. First, a large K would be required to characterize the joint distribution. Secondly, the result would only be valid for that particular N and those particular sensor coordinates - each different geometry would require a different measurement experiment!

Measurements made to date have made simplifying assumptions about the measurement model. Typically, it is assumed that measurements in a network are independent and from the same family of distributions [68, 77, 25, 69, 52, 83]. The independence assumption, which says that observing an error in one link does not provide any information about whether or not errors occur in different links, is a simplifying assumption [94]. Large obstructions may affect a number of similarly-positioned links in a network. Considering correlations between links would make the analysis more difficult, but as a future research topic, conducting measurements to characterize the effects of link dependencies would be basic research which would enable even more accurate analysis and prediction in wireless sensor networks.

The second simplifying assumption is the choice of a family of distributions. Each measurement is usually subtracted from its ensemble mean, and then the assumption is made that the error (the difference) is characterized by a particular parameterized distribution, such as a Gaussian, log-normal, or mixture distribution. The measurements are then used to estimate the parameters of the distribution, such as the variance. By this method, one set of parameters can be used to characterize the whole set of measurements.

The author has conducted several measurement experiments in order to determine accurate statistical models for RSS and TOA measurements in indoor wireless sensor

networks. These measurement experiments are presented in Section 2.6. However, it is useful to first have a general introduction to the sources of error and difficulties of each type of measurement, before being able to understand a particular measurement campaign. Thus, Sections 2.2 through 2.5 are used to present an introduction into the various measurement models, prior to the measurement-based validation in Section 2.6. Each measurement modeling section (Sections 2.2 through 2.5) discusses four sub-topics: 'Major Sources of Error', 'Statistical Model', and 'Calibration and Synchronization'.

2.2 Received Signal Strength

Received signal strength (RSS) is defined as the voltage measured by a receiver's received signal strength indicator (RSSI) circuit. Often, RSS is equivalently reported as measured power, *i.e.*, the squared magnitude of the signal strength. The RSS of acoustic, RF, or other signals can be considered. Wireless sensors communicate with neighboring sensors, and RSS of RF signals can be measured by each receiver during normal data communication, without presenting additional bandwidth or energy requirements. Because RSS measurements are relatively inexpensive and simple to implement in hardware, they are an important and popular topic of localization research. Yet, RSS measurements are notoriously unpredictable. If they are to be useful and part of a robust localization system, their sources of error must be well-understood.

2.2.1 Major Sources of Error

In free space, signal power decays proportional to d^{-2}, where d is the distance between transmitter and receiver. In real-world channels, multipath signals and shadowing are two major sources of environment-dependence in the measured RSS

[101, 52]. Multiple signals with different amplitudes and phases arrive at the receiver, and these signals add constructively or destructively as a function of the frequency, causing frequency-selective fading. The effect of this type of fading can be diminished by using a spread-spectrum method (eg. direct-sequence or frequency hopping) which averages the received power over a wide range of frequencies. Spread-spectrum receivers are an acceptable solution, since spread-spectrum methods also reduce interference in the unlicensed bands in which wireless sensors typically operate. The measured received power using a wideband method (as the bandwidth $\rightarrow \infty$) is equivalent to measuring the sum of the powers of each multipath signal [36].

Assuming that frequency-selective effects are diminished, environment-dependent errors in RSS measurements are caused by shadowing, *i.e.*, the attenuation of a signal due to obstructions (furniture, walls, trees, buildings, and more) that a signal must pass through or diffract around in its path between the transmitter and receiver. Shadowing is also called medium-scale fading [52]. As discussed at the start of this section, these shadowing effects are modeled as random - as a function of the environment in which the network is deployed. A RSS model considers the randomness across an ensemble of many deployment environments.

2.2.2 Statistical Model

Typically, the ensemble mean received power in a real-world, obstructed channel decays proportional to d^{-n_p}, where n_p is the 'path-loss exponent', typically between 2 and 4 [101, 52]. The ensemble mean power at distance d is typically modeled as

$$\bar{P}(d) = \Pi_0 - 10 n_p \log \frac{d}{\Delta_0} \tag{2.1}$$

where Π_0 is the received power (dBm) at a short reference distance Δ_0.

The difference between a measured received power and its ensemble average, due

to the randomness of shadowing, is modeled as log-normal (*i.e.*, Gaussian if expressed in dB). The log-normal model is based on a wide variety of measurement results [52, 101, 35, 93, 92] and analytical evidence [29]. This model is tested via experimental measurements in Section 2.6. The standard deviation of received power (when received power is expressed in dBm), σ_{dB}, has units of (dB) and is relatively constant with distance. Typically, σ_{dB} is as low as 4 and as high as 12 [101]. Thus, the received power (dBm) at sensor i transmitted by j, $P_{i,j}$ is distributed as

$$f\left(P_{i,j} = p|\boldsymbol{\theta}\right) = \mathcal{N}\left(p; \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|), \sigma_{dB}^2\right), \tag{2.2}$$

where $\mathcal{N}(x; y, z)$ is our notation for the value at x of a Gaussian p.d.f. with mean y and variance z, $\boldsymbol{\theta}$ is the coordinate parameter vector from (1.1), and the actual transmitter-receiver separation distance $\|\mathbf{z}_i - \mathbf{z}_j\|$ is given by

$$\|\mathbf{z}_i - \mathbf{z}_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}, \tag{2.3}$$

for a two-dimensional location coordinate $\mathbf{z}_i = [x_i, y_i]^T$.

2.2.3 Estimating Range from RSS

The 'range', *i.e.*, the estimated distance between devices i and j, can be estimated from $P_{i,j}$. First, the maximum likelihood estimate of range is presented. The log-likelihood of $P_{i,j}$ given $d_{i,j} = \|\mathbf{z}_i - \mathbf{z}_j\|$ is,

$$\log f\left(P_{i,j}|\boldsymbol{\theta}\right) = c_1 - \frac{\left[P_{i,j} - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)\right]^2}{2\sigma_{dB}^2} \tag{2.4}$$

where c_1 is a constant independent of $\boldsymbol{\theta}$. Because of the quadratic form, it is clear that the maximum of the log-likelihood occurs when $P_{i,j} = \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$, where the $\bar{P}$ is given in (2.1). As a direct result, the distance $\delta_{i,j}^{MLE}$ which best estimates $\|\mathbf{z}_i - \mathbf{z}_j\|$ in the maximum-likelihood sense is,

$$\delta_{i,j}^{MLE} = \Delta_0 10^{\frac{\Pi_0 - P_{i,j}}{10 n_p}}. \tag{2.5}$$

Note that $\delta_{i,j}^{MLE}$ has a log-normal distribution since $\log \delta_{i,j}^{MLE}$ has a Gaussian distribution, and that

$$E\left[\delta_{i,j}^{MLE}\right] = C\|\mathbf{z}_i - \mathbf{z}_j\|, \tag{2.6}$$

where

$$C = \exp[\gamma/2], \quad \text{where } \gamma = \left(\frac{10n_p}{\sigma_{dB}\log 10}\right)^2. \tag{2.7}$$

The parameter C is a multiplicative bias factor. For typical channels (like those reported in [101]), $C \approx 1.2$, adding 20% bias to the range.

Motivated by (4.3), a bias-corrected estimator (a pseudo-MLE) of distance can be defined just by dividing the MLE by C,

$$\delta_{i,j}^{BC} = \frac{\Delta_0}{C} 10^{\frac{\Pi_0 - P_{i,j}}{10n_p}}. \tag{2.8}$$

The most important result of the log-normal model is that RSS-based range estimates (from either estimator above) have variance proportional to their actual range. This is not a contradiction of the earlier statement that σ_{dB} is constant with range. In fact, constant standard deviation in dB means that the multiplicative factors are constant with range. For example, consider a multiplicative factor of 1.5. At an actual range of 100m, one would measure a range of 150m, an error of 50m; at 10m, the measured range would be 15m, an error of 5m, a factor of 10 smaller. This is why RSS errors are referred to as multiplicative, in comparison to the additive TOA errors presented in Section 2.3. Clearly, RSS is most valuable in high-density sensor networks.

2.2.4 Calibration and Synchronization

As made clear by the path loss exponent model in (2.1), the measured RSS is a function of the path loss exponent n_p. This can be estimated along with the

coordinates, as an unknown parameter, as briefly described in Section 4.2.

In addition, measured RSS is also a function of the calibration of both the transmitter and receiver. Depending on the expense of the manufacturing process, RSSI circuits and transmit powers will vary from device to device. Also, transmit powers can change as batteries deplete. Sensors might be designed to measure and report their own calibration data to their neighbors.

Alternatively, each sensor's transmit power can be considered an unknown parameter to be estimated. This means that the unknown vector $\boldsymbol{\theta}$ described in Section 1.3 is augmented to include the actual transmit power of each sensor along with its coordinates. Or, analogous to time-difference of arrival (TDOA) measurements described in Section 2.3, one can consider only the differences between RSS measured at pairs of receivers [72]. The RSS difference between two sensors indicates information about their relative distance from the transmitter, and removes the dependency on the actual transmit power. As another alternative, each sensor might retain only the order statistics of RSS - an ordered list from highest RSS to lowest RSS from neighbor's transmissions [128]. The discussion of localization algorithms is left until Chapter IV.

2.3 Time-of-Arrival

Time-of-Arrival (TOA) is the measured time at which a signal (RF, acoustic, or other) first arrives at a receiver. The measured TOA is the time of transmission plus a propagation-induced time delay. This time delay, $T_{i,j}$, between transmission at sensor i and reception at sensors j, is equal to the transmitter-receiver separation distance, $\|\mathbf{z}_i - \mathbf{z}_j\|$ divided the propagation velocity, v_p. This speed for RF is approximately 10^6 times as fast as the speed of sound – as a rule of thumb, for acoustic propagation,

1 ms translates to 1 ft (0.3 m), while for RF, 1 ns translates to 1 ft.

The cornerstone of time-based techniques is the receiver's ability to accurately estimate the arrival time of the line-of-sight (LOS) signal. This estimation is hampered both by additive noise and multipath signals.

2.3.1 Major Sources of Error: Additive Noise

Even in the absence of multipath, the accuracy of the arrival time is limited by additive noise. Estimation of time-delay in additive noise is a relatively mature field [16]. Typically, the TOA estimate is the time that maximizes the cross-correlation between the received signals and the known transmitted signal. This estimator is known as a simple cross-correlator (SCC). The generalized cross-correlator (GCC) derived by Knapp and Carter [64] (the maximum likelihood estimator (MLE) for the TOA) extends the SCC by applying pre-filters to amplify spectral components of the signal that have little noise and attenuate components with large noise. As such, the GCC requires knowledge (or estimates) of the signal and noise power spectra.

For a given bandwidth and signal-noise ratio (SNR), our time-delay estimate can only achieve a certain accuracy. The Cramér-Rao bound (CRB) provides a lower bound on the variance of the TOA estimate in a multipath-free channel. For a signal with bandwidth B in (Hz), when B is much lower than the center frequency, F_c (Hz), and signal and noise powers are constant over the signal bandwidth [103],

$$\text{var}(TOA) \geq \frac{1}{8\pi^2\, B\, T_s\, F_c^2\, \text{SNR}}, \tag{2.9}$$

where T_s is the signal duration (s), and SNR is the signal to noise power ratio. By designing the system to achieve sufficiently high SNR, the bound predicted by the CRB (2.9) can be achieved in multipath-free channels. Thus (2.9) provides intuition about how signal parameters like duration, bandwidth, and power affect our ability to

accurately estimate the TOA. For example, doubling either the transmission power or the bandwidth will cut ranging variance in half. This CRB on TOA variance is complementary to the bound that will be presented in Section 3.1 for location variance, because the location variance bound requires, as an input, the variance of the TOA estimates.

2.3.2 Major Sources of Error: Multipath

TOA-based range errors in multipath channels can be many times greater than those caused by additive noise alone. Essentially, all late-arriving multipath components are self-interference that effectively decrease the SNR of the desired LOS signal. Rather than finding the highest peak of the cross-correlation, in the multipath channel, the receiver must find the first-arriving peak, because there is no guarantee that the LOS signal will be the strongest of the arriving signals. This can be done by measuring the time that the cross-correlation first crosses a threshold. Alternatively, in template-matching, the leading edge of the cross-correlation is matched in a least-squares sense to the leading edge of the auto-correlation (the correlation of the transmitted signal with itself) in order to achieve sub-sampling time resolutions [96]. Generally errors in TOA estimation are caused by two problems:

- *Early-Arriving Multipath*: Many multipath signals arrive very soon after the LOS signal, and their contributions to the cross-correlation obscure the location of the peak from the LOS signal.

- *Attenuated LOS*: The LOS signal can be severely attenuated compared to the late-arriving multipath components, causing it to be 'lost in the noise' and missed completely, causing large positive errors in the TOA estimate.

In dense sensor networks, in which any pair of sensors can measure TOA, there is the distinct advantage of being able to measure TOA between nearby neighbors. As the path length decreases, the LOS signal power (relative to the power in the multipath components) generally increases [83]. The measurement study in [83] is particularly valuable in verification of this claim, because it presented synchronized indoor TOA measurements which specifically measured the received power in the LOS signal and then compared it to the received power measured at later time delays. These measurements were made on a large number of links in an office building, and it was shown that the relative LOS signal power is high at low path lengths, and slowly decreasing with increasing path length. Thus, the severely attenuated LOS problem is especially severe in networks with large inter-sensor distances.

While early-arriving multipath components cause smaller errors than late-arriving multipath, they are very difficult to combat. Generally, wider signal bandwidths are necessary for obtaining greater temporal resolution. The peak width of the autocorrelation function is inversely proportional to the signal bandwidth. A narrow autocorrelation peak enhances the ability to pinpoint the arrival time of a signal and helps in separating the LOS signal cross-correlation contribution from the contributions of the early-arriving multipath signals. Wideband direct-sequence spread-spectrum (DS-SS) or ultra-wideband (UWB) signals (see sidebar on UWB) are popular techniques for high-bandwidth TOA measurements. However, wider bandwidths require higher speed signal processing, higher device costs, and possibly higher energy costs. Standards proposed to the IEEE 802.15 Alternative Physical Layer Task Group 3a quote receiver power consumptions on the order of 200 mW [58], about 5 times the quoted power consumption of IEEE 802.15.4 Low-Rate Wireless Personal-Area Networks receivers. Although high-speed circuitry typically means higher energy-

consumption, the extra bandwidth can be used to lower the time-average power consumption. Transferring data packets in less time means spending more time in standby mode.

Finally, note that time delays in the transmitter and receiver hardware and software add to the measured TOA. While the nominal delays are typically known, variance in component specs and response times can be an additional source of TOA variance.

2.3.3 Statistical Model

Measurements have shown that for short-range measurements, measured time delay can be roughly modelled as Gaussian [77, 68, 17, 25],

$$f\left(T_{i,j} = t|\boldsymbol{\theta}\right) = \mathcal{N}\left(t; \|\mathbf{z}_i - \mathbf{z}_j\|/v_p + \mu_T, \sigma_T^2\right), \tag{2.10}$$

where μ_T and σ_T^2 are the mean and variance of the time delay error, $\boldsymbol{\theta}$ is defined in (1.1), $\|\mathbf{z}_i - \mathbf{z}_j\|$ is given in (2.3), and v_p is the propagation velocity. Wideband DS-SS measurements reported in [92] supported the Gaussian error model and showed $\mu_T = 10.9$ ns and $\sigma_T = 6.1$ ns. UWB measurements done in on a mostly-empty Motorola factory floor showed $\mu_T = 0.3$ ns and $\sigma_T = 1.9$ ns. This mean error μ_T can be estimated (as a nuisance parameter) by the localization algorithm so that it can be subtracted out [25].

However, the presence of large errors can invalidate the Gaussian model. These errors make the tails of the distribution of measured TOA heavier than Gaussian, and have been modeled using a mixture distribution: with a small probability, the TOA measurement results from a different, higher-variance distribution, as described in [13] and [51]. Localization systems should be designed to be robust to these large errors, also called non-line-of-sight (NLOS) errors. For TOA measurements made over

time in a changing channel, the TOAs which include excess delays can be identified and ignored [13]. Even in static channels, if the number of range measurements to a device are greater than the minimum required, the redundancy can be used to identify likely NLOS errors [20, 4]. Localization algorithm robustness is further addressed in Chapter IV.

2.3.4 Calibration and Synchronization

If wireless sensors have clocks that are accurately synchronized, then the time delay is determined by subtracting from the measured TOA the known transmit time. Sensor network clock synchronization algorithms have reported precisions on the order of 10μs [113]. Because of the difference in propagation speed, such clock accuracies are adequate for acoustic signals [47], but not for RF signals.

For time-of-arrival in asynchronous sensor networks, a common practice is to use *two-way* (or *round-trip*) TOA measurements. In this method, a first sensor transmits a signal to a second sensor, which immediately replies with its own signal. At the first sensor, the measured delay between its transmission and its reception of the reply is twice the propagation delay plus a reply delay internal to the second sensor. This internal delay is either known, or measured and sent to the first sensor to be subtracted. Multiple practical two-way TOA methods have been reported in the literature [63, 69, 75, 40]. Generally each pair of sensors measures round-trip TOA separately in time. But, if the first sensor has adequate signal processing capability, multiple sensors can reply at the same time, and two-way TOAs can be estimated simultaneously using multi-user interference cancellation [63].

The state of each sensor's clock (its bias compared with absolute time) can also be considered to be an unknown parameter and included in the parameter vector $\boldsymbol{\theta}$. In this case, one-way TOA is measured and input to a localization algorithm which

estimates both the sensor coordinates and the biases of each sensor's clock [68]. The difference between the arrival times of the same signal at two sensors is called the time-difference of arrival (TDOA). A TDOA measurement does not depend on the clock bias of the transmitting sensor. TDOA methods have been used in source localization for decades for locating asynchronous transmitters, and has application in GPS and cellular localization. Under certain weak conditions, it has been shown that TOA with clock bias (treated as an unknown parameter) is equivalent to TDOA [111].

2.3.5 Ultra-Wideband and TOA

Ultra Wideband (UWB) communication employs narrow pulses of very short (sub-nanosecond) duration that result in radio signals that are broadly spread in the frequency domain. The article by Gezici et. al. [45] provides a detailed overview of UWB-based localization. A signal is considered to be UWB if either its *fractional bandwidth*, the ratio of its bandwidth to its center frequency, is larger than 0.2, or it is a multiband signal with total bandwidth greater than 500 MHz. In 2003, the U.S. Federal Communications Commission (FCC) approved the commercialization and operation of UWB devices for public safety and consumer applications. Among the envisaged applications are wireless networking and localization. Standardization of UWB is underway, including the development of a high bit rate UWB physical layer that supports peer-to-peer ranging, in IEEE task group 802.15.3a, and in IEEE task group 802.15.4a [58].

The very high bandwidth of UWB leads to very high temporal resolution, making it ideal for high precision radiolocation applications. Implementations of UWB-based range measurements, reported in [69, 41, 25, 40], have demonstrated RMS ranging errors of 0.4 to 5 feet (0.12 to 1.5 m).

2.4 Angle-of-Arrival

By providing information about the direction to neighboring sensors rather than the distance to neighboring sensors, angle-of-arrival (AOA) measurements provide localization information complementary to the TOA and RSS measurements discussed above.

There are two common ways that sensors measure AOA (as shown in Figure 2.1). The most common method is to use a *sensor array* and employ so-called *array signal processing* techniques at the sensor nodes. In this case, each sensor node is comprised of two or more individual sensors (*e.g.,* microphones for acoustic signals or antennas for RF signals) whose locations with respect to the node center are known. A four-element Y-shaped microphone array is shown in Figure 2.1(a). The AOA is estimated from the differences in arrival times for a transmitted signal at each of the sensor array elements. The estimation is similar to time-delay estimation discussed in Section 2.3, but generalized to the case of more than two array elements. When the impinging signal is narrowband (that is, its bandwidth is much less than its center frequency), then a time delay τ relates to a phase delay ϕ by $\phi = 2\pi f_c \tau$ where f_c is the center frequency. Narrowband AOA estimators are often formulated based on phase delay. See [123, 115, 82] for more detailed discussions on AOA estimation algorithms and their properties.

A second approach to AOA estimation uses the RSS ratio between two (or more) directional antennas located on the sensor (see Figure 2.1(b)). Two directional antennas pointed in different directions such that their main beams overlap can be used to estimate the AOA from the ratio of their individual RSS values.

Either AOA approach requires multiple antenna elements, which can contribute to

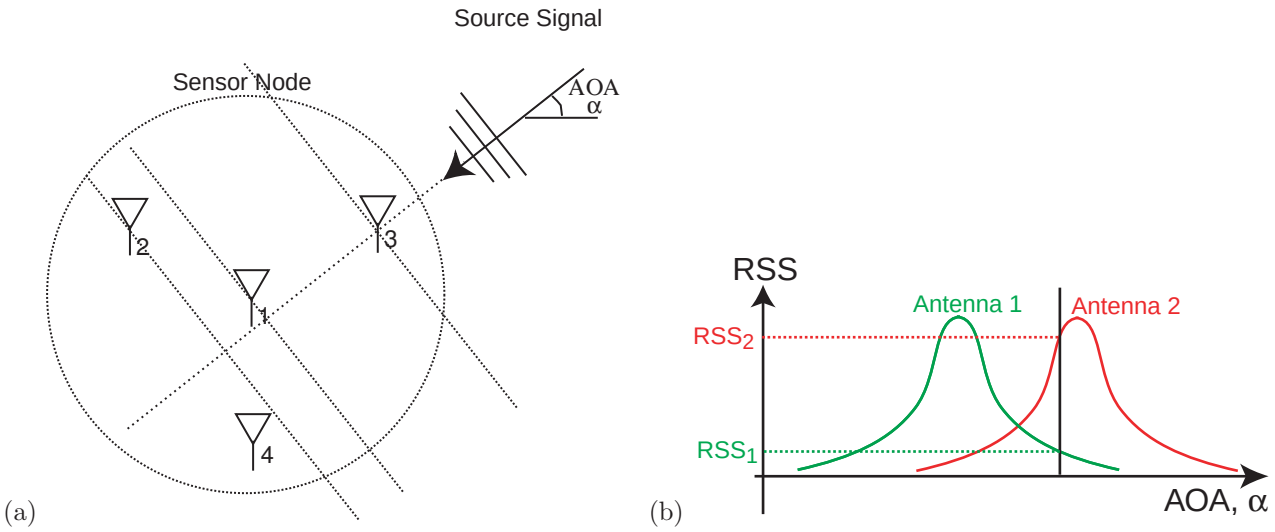


Figure 2.1: Angle-of-arrival (AOA) estimation methods. (a) AOA is estimated from the time-of-arrival differences among sensor elements embedded in the node; a 4-element Y-shaped array is shown. (b) AOA can also be estimated from the received signal strength (RSS) ratio RSS_1/RSS_2 between directional antennas.

sensor device cost and size. However, acoustic sensor arrays may already be required in devices for many environmental monitoring and security applications, in which the purpose of the sensor network is to identify and locate acoustic sources [19]. Locating the sensors themselves using acoustics in these applications is a natural extension. RF antenna arrays imply large device size unless center frequencies are very high. However, available bandwidth and decreasing manufacturing costs at millimeter-wave frequencies may make them desirable for sensor network applications. For example, at 60 GHz, higher attenuation due to oxygen absorption helps to mitigate multipath, and accurate indoor AOA measurements have been demonstrated [126].

2.4.1 Major Sources of Error and Statistical Model

AOA measurements are impaired by the same sources discussed in the TOA section above - additive noise and multipath. The resulting AOA measurements are typically modeled as Gaussian, with ensemble mean equal to the true angle to the source and standard deviation σ_α. Theoretical results for acoustic-based AOA esti-

mation show standard deviation bounds on the order of $\sigma_\alpha = 2^\circ$ to $\sigma_\alpha = 6^\circ$ depending on range [23]. Estimation errors for RF AOA on the order of $\sigma_\alpha = 3^\circ$ have been reported using the RSS ratio method [7].

2.4.2 Calibration and Synchronization

It is not likely that sensors will be placed with known orientation. When sensor nodes have directionality, the network localization problem must be extended to consider each sensor's orientation as an unknown parameter, to be estimated along with position. In this case, the unknown vector $\boldsymbol{\theta}$ (see Section 1.3) is augmented to include the orientation of each sensor.

2.5 Quantized RSS and Connectivity

2.5.1 Connectivity Measurements

It is common for localization research to consider connectivity (a.k.a. proximity) measurements as a simple, inexpensive, backward-compatible location measurement. Whether or not devices have accurate RSS measurement circuitry on their receivers, two devices can determine whether or not they can communicate. Two sensors are *not* considered to be connected solely based on the distance between them – two sensors are connected if the receiving sensor can successfully demodulate packets transmitted by the other sensor. The receiver fails to successfully demodulate packets when the received signal strength (RSS) is too low. Since RSS is a random variable due to the unpredictability of the fading channel, and connectivity is a function of RSS, connectivity is also a random variable.

Specifically, the connectivity measurement of sensors i and j, $Q_{i,j}$, is modeled as

a binary quantization of RSS,

$$Q_{i,j} = \begin{cases} 1, & P_{i,j} \geq P_1 \\ 0, & P_{i,j} < P_1 \end{cases} \tag{2.11}$$

where $P_{i,j}$ is the received power (dBm) at sensor i transmitted by sensor j, and P_1 is the receiver threshold (dBm) under which packets cannot be demodulated.

In reality, being in-range of another device (transmitting a packet which the other device correctly demodulates) is not a step function of received power. An additional source of variation in proximity measurements is the randomness of packet errors given the received power level. Thus in reality, given received power $P_{i,j}$, proximity $Q_{i,j} \in \{0, 1\}$ is a binary random variable, such that

$$\mathcal{P}[Q_{i,j} = 1 | P_{i,j}] = \mathcal{P}\left[\text{No Packet Error} | P_{i,j}\right] \tag{2.12}$$

where the probability of a packet error is a function of the type of signalling and forward error correction (FEC) used, packet length, and whether the receiver is coherent or non-coherent. If (2.12) was used to define proximity, all of these details of the transceiver implementation would be required in order to calculate the performance of localization based on connectivity measurements. Instead, by using (2.11) we can present results that are independent of signaling, packet length, and receiver implementation. Since (2.11) removes some variability from the measurement model, the calculated lower bounds are conservative: they do in fact provide a lower bound for a connectivity-based localization system.

Finally, note that the assumption that proximity is a step function of RSS will not loosen the bound significantly for digital receivers in typical fading channels. For digital receivers, there is a large range of received powers for which the probability of packet error is very close to zero, and a large range of power for which

the probability is very close to one. The range of power for which the probability of packet error is neither close to one or zero is small in comparison. Fig. 2.2 plots $\mathcal{P}[\text{No Packet Error}|P_{i,j}]$ from (2.12) for a packet of 200 bits and a coherent BPSK receiver without FEC. For comparison, Fig. 2.2 also plots the CDF of received power under a log-normal model with standard deviation of 8 dB, which is a typical value for indoor channels [101]. It can be seen that the variation caused by the fading channel is significantly more severe than that caused by the randomness of packet errors given the received power level.

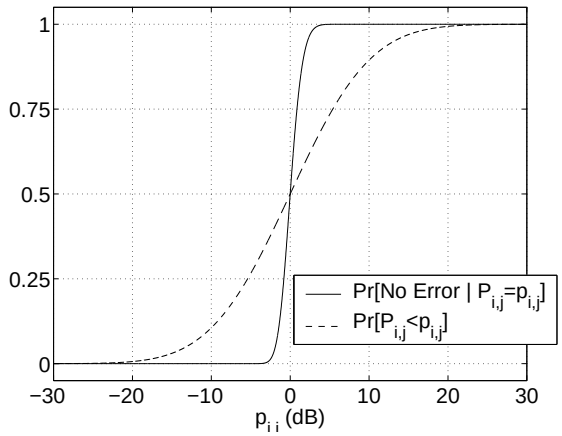


Figure 2.2: Two plots relating to the variation in proximity measurements: (- - - -) the CDF of $P_{i,j}$ in dB above $\bar{P}_{i,j}$, and (——) the probability of no packet error given $P_{i,j}$ in dB above P_{thr} (for a packet of 200 bits and a coherent BPSK receiver without FEC).

Given the definition of proximity in (2.11) and the model for $P_{i,j}$ in (2.2), it can be shown that the probability mass function of $Q_{i,j}$ given the coordinates of devices i and j is

$$\mathcal{P}[Q_{i,j} = s|\mathbf{z}_i, \mathbf{z}_j] = s + (-1)^s \Phi\left[g_{i,j}(1)\right], \tag{2.13}$$

$$g_{i,j}(s) = \sqrt{\gamma} \ln \frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{d_s}, \tag{2.14}$$

$$\gamma = \left(\frac{10 n_p}{\sigma_{dB} \log 10}\right)^2 \tag{2.15}$$

where $s \in \{0, 1\}$, and d_s is the range at which the the mean received power is P_s, ie.,

the communication 'range'. Specifically, from (2.1),

$$d_s = \Delta_0 10^{\frac{\Pi_0 - P_s}{10 n_p}}. \tag{2.16}$$

The function $\Phi(x)$ is the CDF of a univariate zero-mean unit-variance Normal distribution.

2.5.2 Quantized RSS Measurements

As noted, connectivity measurements are just a binary quantization of RSS measurements. For more generality, this thesis considers arbitrary K-level quantized received signal strength (QRSS) system. For example, consider a transmitter which has the option of using a power amplifier (PA). The transmitter could send a packet twice - once using the PA, and once without. Assuming a stationary channel during the two measurements, one of three results are possible, essentially resulting in a single 3-level QRSS measurement.

Also, consider that real-world RSS measurements are always going to be quantized. If there are very many levels, then the effect of the quantization is minimal. However, if an A/D converter is used to quantize an analog measure of received power, the complexity of the A/D increases linearly with the number of levels. Determining the acceptable granularity of measured RSS will help minimize receiver complexity.

Expanding on the model for connectivity measurements, define K levels numbered 0 through $K-1$. For a particular system, define the (increasing) threshold powers P_s, $s \in \{1, \ldots, K-1\}$ to be the minimum power in level s. Similarly, define d_s to be the path length at which the mean received power is equal to P_s, as in (2.16). Thus, a measurement of $Q_{i,j} = s$ for $s \in \{0, \ldots, K-1\}$ would occur if $P_{i,j} \in [P_s, P_{s+1})$.

Similarly, $Q_{i,j} = s$ would occur if $\delta_{i,j} \in (d_{s+1}, d_s]$, where

$$\delta_{i,j} = \Delta_0 10^{\frac{\Pi_0 - P_{i,j}}{10 n_p}} \tag{2.17}$$

Since there is no lower bound for 'out-of-range' power (the lower bound of level $s = 0$), define $P_0 = -\infty$(dBm). Similarly, since it is preferable not to define a finite maximum measured power (the upper bound of level $s = K - 1$), define $P_K = \infty$(dBm). Using (2.16), this implies that $d_0 = \infty$ and $d_K = 0$. Note P_s are increasing in s, but d_s are decreasing in s.

Now the probability mass function of QRSS measurements can be written as,

$$\mathcal{P}[Q_{i,j} = s | \mathbf{z}_i, \mathbf{z}_j] = \Phi\left[g_{i,j}(s+1)\right] - \Phi\left[g_{i,j}(s)\right], \tag{2.18}$$

where $g_{i,j}(k)$ is given in (2.14) and the convention is used for $0 < d < \infty$, $\ln \frac{d}{0} = \infty$ and that $\ln \frac{d}{\infty} = -\infty$.

2.5.3 Calibration and Synchronization

In general, QRSS and connectivity have identical calibration and synchronization issues are RSS. As noted, QRSS is a more realistic representation of RSS, since RSS will be generally be quantized to some extent. However, in addition to knowing transmit power, an optimal location estimator would need to know the thresholds between the quantization levels in each receiver. In general, these are not exactly known. In reality, sub-optimal algorithms will be used for localization, which do not try to know or estimate the receiver quantization function.

2.6 Channel Measurement Experiments

Figure 2.3: Photo of measurement area looking above cubicle walls.

In this section, two sets of multipoint-to-multipoint (M2M) wideband channel measurements are presented. These measurements were conducted in the Motorola facility in Plantation, Florida. Here, the term 'multipoint-to-multipoint' emphasizes that the channel between every pair of sensors is measured. In contrast, channel measurements made for cellular deployment or WLAN deployment, measure the channel between a base station and many possible mobile station locations, and are thus point-to-multipoint. The first campaign measures RSS only, while the second campaign uses more sophisticated hardware and measures both TOA and RSS simultaneously.

The objective of these campaigns are first, to verify the models presented in this chapter, and provide estimates of the model parameters. Second, the measurements are directly useful as input into localization algorithms to test what their performance would have been had they been deployed in the measured environment. The model verification and parameter estimation is presented in this chapter. The measurement-based localization performance of the algorithms presented later in this thesis are

presented after those algorithms have been introduced in Chapter IV.

In both campaigns, the measurement environment is an office area partitioned by 1.8m high cubicle walls, with hard partitioned offices, external glass windows and cement walls on the outside of the area. There are also metal and concrete support beams within and outside of the area. Offices are occupied with desks, bookcases, metal and wooden filing cabinets, computers and equipment. Since the areas are open plan, it is difficult to define the 'room dimensions', but this is typical of office environments in modern office buildings.

2.6.1 Measurement Campaign I: RSS

The initial measurement campaign used off-the-shelf measurement equipment to measure received signal strength in a narrowband channel at 925 MHz, within the 900-928 MHz ISM band in the US. The measurement system consists of a HP 8644A signal generator transmitting a CW signal at 925 MHz at an output level of 0.1 mW and a Berkeley Varitronics Fox receiver. A $\lambda/4$ dipole with Roberts balun resonant at 925 MHz is positioned at a height above the floor of 1 meter at both the transmitter and receiver. The antennas are both stationary during each measurement and have an omnidirectional radiation pattern in the horizontal plane and a vertical beamwidth of 30^o. The Fox receiver was set to average received power over one second. The campaign is conducted during evenings and on weekends to ensure that the channel is mostly static during the measurements. Two meter tall Hayworth partitions and ceiling-height interior walls divide the area into cubicles, lab space, and offices. To simulate a system in which reference devices are placed approximately every 15 m in the indoor environment, they are placed in a 4 by 4 grid in the measurement area (see map in Fig. 2.4), thus $m = 16$.

Forty locations are chosen for the unknown-location devices ($n = 40$) in the

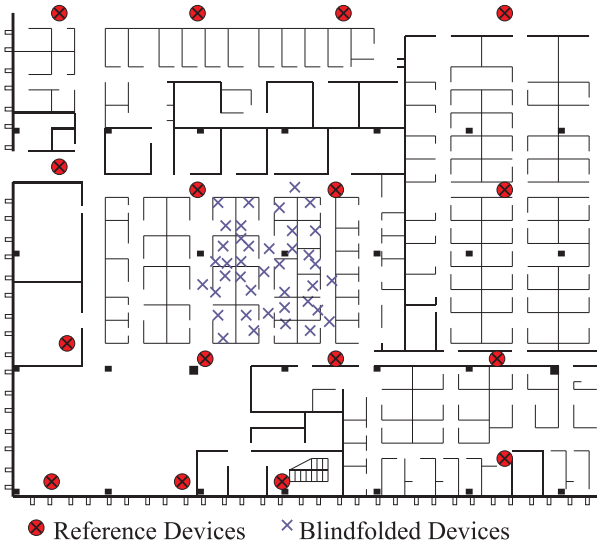


Figure 2.4: Floor plan of measurement area for Measurement Campaign I showing sensor locations.

center quadrant (16 m by 14 m). The center quadrant consists of four columns of cubicles and the hallways that separate them. Two or three unknown-location device locations are chosen for each cubicle, and a few locations put into the hallways. This density or greater might be expected, for example, in a location and tracking system in which each employee places a tag on two or three valuable things that he or she works with, such as computers and accessories, electronic equipment, briefcases, wireless phones, notebooks, tools, or key rings. Together, there are $N = 56$ total node (reference and unknown-location device) locations.

First, the transmitter is placed at location 1, and received power readings are taken and recorded at locations 2 through 56. Next, the transmitter is moved to location 2, and power readings are taken at locations 1 and 3 through 56. This process continues until power measurements have been made between each pair of devices (in both directions), for a total of $56 \times 55 = 3080$ RSS measurements. The measured received powers, plotted in Fig. 2.6, fit the channel model of Eq. 2.1 with a d_0 of 1 m, n_p of 2.98. The histogram of the residuals, $r^R_{i,j} = P_{i,j} - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$ shows

a Gaussian PDF with nearly zero mean and a standard deviation of $\sigma_{dB} = 7.38$.

The log-normal distribution of the RSS measurements is verified by examining the residuals $r_{i,j}^R$ using a quantile-quantile plot in Fig. 2.5. If the data is log-normal, it would be expected that the data points to follow closely to the straight line. They are very close, within the $[-2, 2]$ quantile range. Outside of that range, the data is somewhat heavier-tailed.

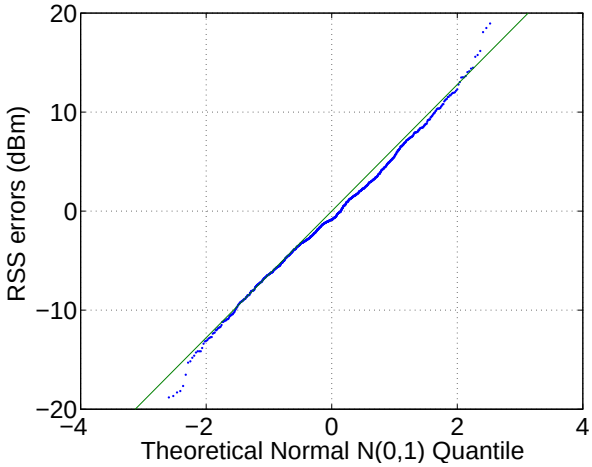


Figure 2.5: Q-Q plot of residuals $P_{i,j} - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$ for RSS data (in dBm), compared to a Gaussian quantile.

Using a Kolmogorov-Smirnov (KS) test, the hypothesis is tested: $H_0 : r_{i,j}^R \sim \mathcal{N}(\bar{r}^R, S_R^2)$ vs. $H_1 : r_{i,j}^R$ is not Gaussian, where $\bar{r}^R$ is the sample mean of $r_{i,j}^R$ and S_R^2 is the sample variance. The KS test yields a p-value of 0.28. One would accept H_0 at a level of significance of $\alpha = 0.05$. Note that the KS test p-value is particularly affected by the worst fit between measurements and model, which occurs further than 2 quantiles away from the mean. Thus, except for the behavior of the far tails shown in Fig. 2.5, the data shows even higher fit with the log-normal model than indicated by $p = 0.28$.

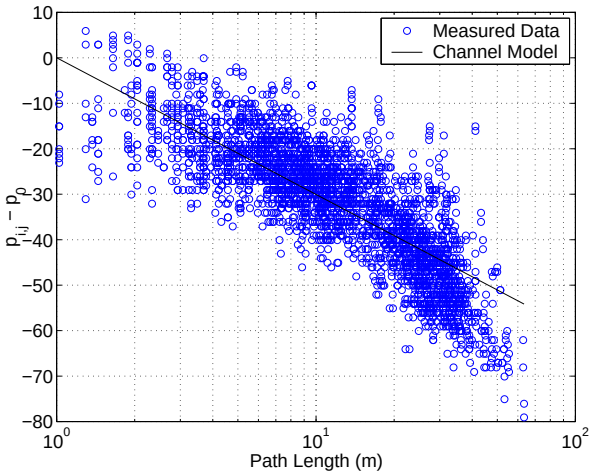


Figure 2.6: RSS Measurements in Experiment I compared to path loss exponent model in (2.1) shows agreement with $n_p = 2.98$, with standard deviation $\sigma_{dB} = 7.38$.

2.6.2 Measurement Campaign II: RSS and TOA

In the second measurement experiment, a more sophisticated measurement system is used in order to simultaneously measure RSS and TOA. While measurement of round-trip TOA can, in fact, be implemented in relatively inexpensive hardware, the author knows of no off-the-shelf commercial hardware that provides an all-purpose testbed. In this experiment, since it was hoped to sample the channel such that different LOS estimation methods could be tested, hardware that implemented a particular LOS estimation method and reports the TOA could not be used. Instead, a more capable system was used, basically involving software radios which are synchronized to GPS, programmed to meet the measurement needs, and which record complete channel data, which is then post-processed using the 'template-matching' LOS estimation algorithm presented in Section 2.3.2.

The measurement system comprises a wideband direct-sequence spread-spectrum (DS-SS) transmitter (TX) and receiver (RX) (Sigtek model ST-515). The TX and RX are battery-powered and are placed on carts. The TX outputs an unmodulated pseudo-noise (PN) code signal with a 40 MHz chip rate and code length 1024. The

center frequency is 2443 MHz, and the transmit power is 10 mW. Both TX and RX use 2.4 GHz sleeve dipole antennas kept 1m above the floor. The antennas have an omnidirectional pattern in the horizontal plane and a measured antenna gain of 1.1 dBi. The RX records I and Q samples at a rate of 120 MHz, downconverts, and correlates them with the known PN signal and outputs a power-delay profile (PDP). An example PDP is shown in Fig. 2.7. Noise and ISM-band interference is ensured to have minimal effect by by maintaining an SNR > 25 dB throughout the campaign.

For TOA, wireless sensors will likely make two-way (round-trip) measurements as discussed in Section 2.3. However, for the purpose of these measurements, two-way TOA measurements are not necessary. Instead, the TX and RX are carefully synchronized using off-the-shelf time-synchronization equipment. Since they are carefully calibrated, one can essentially simulate what would happen in a two-way TOA measurement using one-way TOA measurements. Essentially, this method measures *more* than is necessary to model two-way TOA measurements; however, the one-way measurements were easier to implement using off-the-shelf equipment.

Both TX and RX are synchronized by 1 pulse per second (1PPS) signals from two Datum ExacTime GPS and rubidium-based oscillators. On each of the eight days of the campaign, a procedure is followed to ensure a stable time base. After an initial GPS synch of the ExacTimes, GPS is disconnected and the rubidium oscillators provide stable 1PPS signals. The frequencies of the two rubidium oscillators are off very slightly, thus the 1PPS signals drift linearly, on the order of ns per hour. By periodically measuring and recording the offset between the two 1PPS signals using an oscilloscope, the effect of the linear drift can be cancelled. A time base with a standard deviation of between 1-2 ns is achieved. Since the noise variances add together, the variance of the time base ($\leq 4\text{ns}^2$) is a relatively small source of error

in the total measured TOA variance (37ns^2) which will be explained in Section 2.6.2.

Forty-four device locations are identified and marked with tape. The smallest rectangle that encompasses the 44 locations is 14m by 13m. The M2M measurements are conducted by first placing the TX at location 1 while the RX is moved and measurements are made at locations 2 through 44. Then the TX is placed at location 2, as the RX is moved to locations 1 and 3 through 44. At each combination of TX and RX locations, the RX records five PDPs. Since reciprocity is expected, there are a total of 10 measurements for each link. All devices are in range of all other devices. Over the course of the 8-day campaign, a total of 44*43*5 = 9460 measurements are taken.

TOA from Experiment II

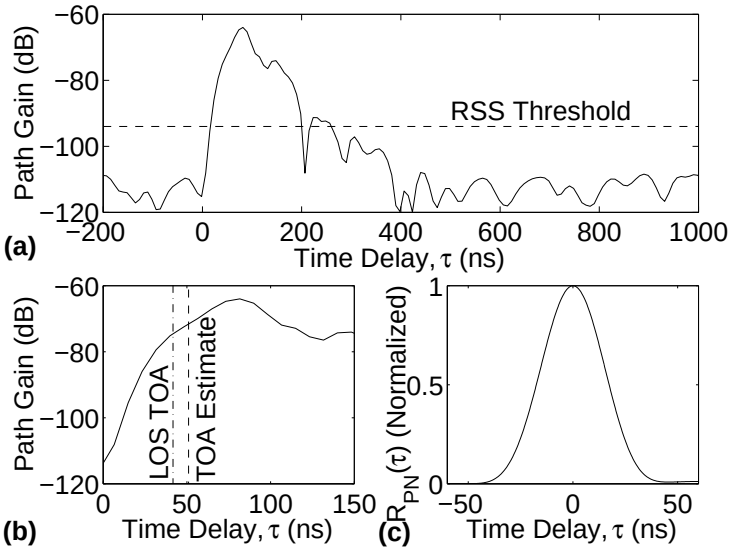


Figure 2.7: (a) Measured PDP with TX at 1 and RX at 24 and threshold (- - -) above which received power is integrated to calculate RSS. (b) Leading edge of same PDP showing LOS TOA $= \|\mathbf{z}_1 - \mathbf{z}_{24}\|/v_p$ $(\cdot - \cdot - \cdot)$ and estimated TOA (- - -). (c) Autocorrelation of PN signal $R_{PN}(\tau)$ used in template-matching [96].

The wideband radio channel impulse response (CIR) is modeled as a sum of attenuated, phase-shifted, and time-delayed multipath impulses [52][101]. The PDP

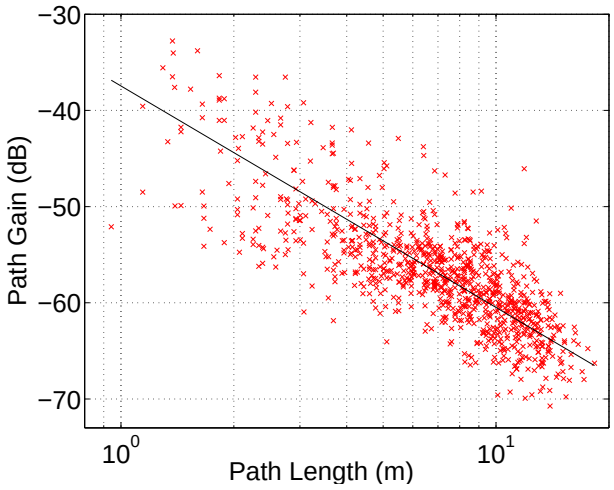


Figure 2.8: Measured wideband path gain (x) as a function of path length. Linear fit (—) is with $d_0 = 1$m, $n_p = 2.3$, and $\sigma_{dB} = 3.92$.

output of the Sigtek measurement system, due to its finite bandwidth, replaces each impulse of the CIR with the autocorrelation function of the PN signal $R_{PN}(\tau)$ shown in Fig. 2.7(c), an approximately triangular peak $2/R_C = 50$ns wide. In high SNR, low multipath cases, TOA estimates can be more accurate than $2/R_C$. However, a wider peak permits more multipath errors since the line-of-sight (LOS) component, with TOA $\|\mathbf{z}_i - \mathbf{z}_j\|/v_p$, can be obscured by non-LOS multipath that arrive $< 2/R_C$ seconds after the LOS TOA. If the LOS component is attenuated, it can be difficult to distinguish the LOS TOA. In Fig. 2.7(a), the PDP is seen to contain several multipath within the first 200ns. Inspecting the PDP immediately after $\tau = 0$, as shown in Fig. 2.7(b), the LOS path at 42ns is visible but attenuated compared to a later multipath which appears to arrive at 80ns.

The template-matching method [96] provides a TOA estimation algorithm which is robust to such attenuated-LOS multipath channels. In template-matching, samples of the leading edge of the PDP are compared to a normalized and oversampled template of $R_{PN}(\tau)$ shown in Fig. 2.7(c). The TOA estimate $\tilde{t}_{i,j}$ is the delay that minimizes the squared-error between the samples of the PDP and the template. In

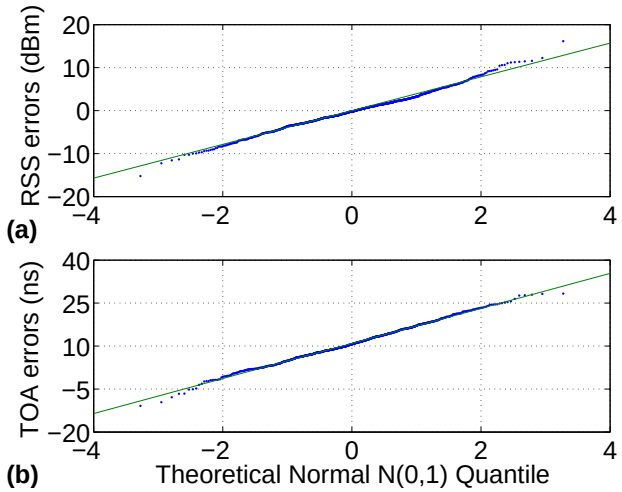


Figure 2.9: Q-Q plot of (a) $P_{i,j} - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$ for RSS data, and (b) $T_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\|/v_p$ for TOA data, compared to a Gaussian quantile.

Fig. 2.7(b), the template-matching TOA estimate $\tilde{t}_{1,24} = 51$ns is in error by +9ns. If a local maximum was necessary to identify the LOS path, the error would have been much greater.

The average of the 10 $\tilde{t}_{i,j}$ measurements for the link between i and j is called $T_{i,j}$. This set of time-averaged measurements, $\{T_{i,j}\}$, is a reasonable representation of the TOAs that the studied 44-node sensor network would measure.

Consider the error in the measurement $r^T_{i,j}$,

$$r^T_{i,j} = T_{i,j} - \frac{\|\mathbf{z}_i - \mathbf{z}_j\|}{v_p}. \tag{2.19}$$

Since non-LOS multipath are delayed in time, $r^T_{i,j}$ usually has a positive mean. In these measurements, the average of $r^T_{i,j}$ for all pairs (i,j), $\mu_T = 10.9$ ns. The measured standard deviation, σ_T, is 6.1 ns.

Next, the parameters for the measurement-based RSS model is shown, before performing hypothesis tests on the measured data in Section 2.6.2.

RSS from Experiment II

It has been shown that a wideband estimate of received power, $p_{i,j}$, can be obtained by summing the powers of the multipath in the PDP [101]. To distinguish between noise and multipath, only power above a noise threshold is summed, as shown in Fig. 2.7(a). This wideband method reduces the frequency-selective fading effects. Define the time-average RSS measurement $P_{i,j}$ to be the geometric mean of the 10 $p_{i,j}$ measurements for the link between i and j. As described in Section 2.2, time-averaging reduces fading due to motion of objects in the channel, and shadowing effects remain the predominant source of error.

The measured $P_{i,j}$ match the log-normal shadowing model in (2.2) with $n = 2.30$ and $\sigma_{dB} = 3.92$ dB, using $d_0 = 1$m. The low variance may be due to the wide bandwidth, averaging, and homogeneity of the measured cubicle area.

In Figure 2.10, the error between range measurements and real distances, i.e., $\delta_{i,j}^{BC} - \|\mathbf{z}_i - \mathbf{z}_j\|$, is plotted. Note that the standard deviation of the RSS-based range estimator error increases steadily with distance. But, most importantly, the error as a percentage of actual range is often high: there are several range errors larger than 100% of the actual range.

RSS Model Verification from Experiment II

The log-normal and Gaussian distributions of the RSS and TOA measurements are verified by examining the residuals $r_{i,j}^R \triangleq P_{i,j} - \bar{P}(\|\mathbf{z}_i - \mathbf{z}_j\|)$ and $r_{i,j}^T \triangleq T_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\|/v_p$ via quantile-quantile (Normal probability) plots in Fig. 2.9. Both RSS and TOA data fit the models well between the -2 and +2 quantiles. Using a Kolmogorov-Smirnov (KS) test, the hypothesis is tested: $H_0 : r_{i,j}^R \sim \mathcal{N}(\bar{r}^R, S_R^2)$ vs. $H_1 : r_{i,j}^R$ is not Gaussian, where $\bar{r}^R$ is the sample mean of $r_{i,j}^R$ and S_R^2 is the sample variance. An

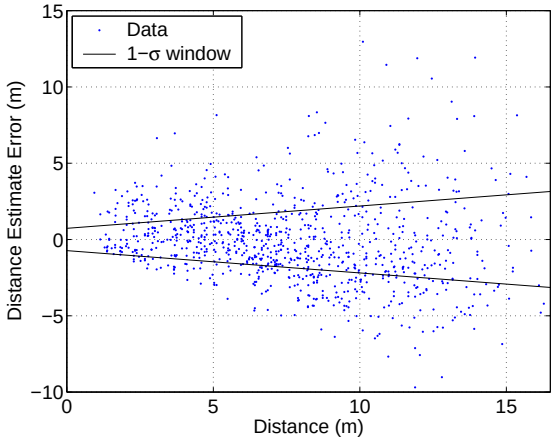


Figure 2.10: Plot of distance measurement errors Vs. distance. The $1-\sigma$ interval superimposed on the plot was obtained from the ML fit of the error measurement model $\mathcal{N}(\|\mathbf{z}_i - \mathbf{z}_j\|, \big(a\,\|\mathbf{z}_i - \mathbf{z}_j\| + b\big)^2)$.

identical test is conducted on $r_{i,j}^T$ for the TOA measurements. For the RSS and TOA residuals, the KS tests yield p-values of 0.09 and 0.50, respectively. In both cases, one would decide to accept H_0 at a level of significance of $\alpha = 0.05$.

However, the low p-value for the RSS data indicates that log-normal shadowing model in (2.2) may not fully characterize the data. In fact, if in H_0 a 2-component Gaussian mixture distribution is used (with parameters estimated from $r_{i,j}^R$ via the MLE), the KS test yields a p-value of 0.88. A topic for future research is to investigate whether the potential benefits of using a mixture distribution in the channel model would justify its additional complexity. The experimental results reported in the next sections do assume the log-normal shadowing model. An avenue of future research remains to investigate the difference in lower bounds and maximum likelihood estimation performance when assuming a log-normal mixture model as compared to a log-normal model for RSS.

2.6.3 Data Availability

Measurement-based verification is essential for testing localization algorithms and determining measurement models, yet such measurement sets are difficult to record.

To help overcome this difficulty, the TOA and RSS measurements from Experiment II have been made freely available online in Matlab data file format [84].

2.7 Conclusion

The models presented in Sections 2.2 through 2.5 have been shown through experiments, both those the author has conducted and those reported in the literature, to be good approximations for the very complicated real-world behavior of pair-wise measurements in unpredictable RF channels. Wireless sensor networks designed to use these pair-wise measurements should consider these sources of error, error models, and calibration and synchronization issues.

In particular, these error models are sufficient to find Cramér-Rao bounds on localization performance in cooperative localization, lower bounds which are not a function of the particular localization algorithm employed, and thus can be used to very quickly judge the precisions possible from various measurement modalities. The models in this chapter are fundamental to the performance limits presented in the following chapter.

CHAPTER III

Localization Bounds

3.1 Limits on Localization Covariance

The Cramér-Rao bound provides a means for calculating a lower bound on the covariance of any unbiased location estimator which uses connectivity, QRSS, RSS, TOA, or AOA measurements. Such a lower bound provides a useful tool for researchers and system designers. Without testing particular estimation algorithms, a designer can quickly find the 'best-case' using particular measurement technologies. Researchers who are testing localization algorithms, like those presented in Chapter IV, can use the CRB as a benchmark for a particular algorithm. If the bound is nearly achieved, then there is little reason to continue working to improve that algorithm's accuracy. Furthermore, the bound's functional dependence on particular parameters helps to provide insight into the behavior of cooperative localization.

The bound on estimator covariance is a function of the following:

1. Number of unknown-location and known-location sensors,
2. Sensor geometry,
3. Whether localization is in two or three dimensions,
4. Measurement type(s) implemented (*i.e.*, RSS, TOA, or AOA),

5. Channel parameters (such as σ_{dB} and n_p in RSS, σ_T in TOA, or σ_α in AOA measurements),

6. Which pairs of sensors make measurements (network connectivity),

7. 'Nuisance' (unknown) parameters which must also be estimated (such as clock bias for TOA or orientation for AOA measurements).

Note that, in collaboration with J.N. Ash at Ohio State University, we have developed a publicly available, multi-featured Matlab-based code and GUI for the calculation of the localization CRB, shown in Figure 3.2 [5]. The GUI was developed to illustrate the use of a variety of Cramér-Rao bounds on localization variance developed jointly at OSU and the University of Michigan. The code can determine bounds when any combination of RSS, TOA, and AOA measurements are used. It allows the inclusion of device orientation and clock biases as unknown 'nuisance' parameters. Sensors can be arranged visually using the GUI and the bound can be calculated. For each sensor, the GUI displays the CRB by plotting the lower bound on the 2-σ uncertainty ellipse. The tool also includes the ability to run Monte-Carlo simulations which estimate sensor parameters and coordinates using the maximum-likelihood estimator (MLE) that will be discussed in Chapter IV. The Monte-Carlo coordinate estimates are plotted on screen for comparison with the covariance bound. This is a very useful graphical tool to help make concrete the analysis presented in this section.

Furthermore, the author has made public more basic Matlab functions for directly calculating the CRB [84]. Given the above list of inputs, these Matlab functions will directly calculate and return the lower bound on the variance. These functions allow a user to calculate bounds for comparison with their own localization simulation

results. In addition to TOA, RSS, and AOA, this code additionally calculates bounds for QRSS and connectivity. These code and tools have been made publicly available in order to make the bound more readily available to researchers and system designers from a variety of disciplines.

In this section, analytical results for the CRB are presented. First, it is shown that 'cooperative' location estimation is one of a larger class of 'self-calibration' estimators which use some prior information, and pair-wise estimates, to infer device parameters. A result is proven regarding sufficient conditions for a strictly decreasing variance bound as more sensors are added to the network. This result is an important theoretical contribution of this thesis.

Next, the localization CRB derivation is presented for the different pair-wise measurement methods considered in this thesis. The objective is both to show that the lower bound is simple to calculate and that it provides a means to compare the three measurement methods presented in Chapter II.

To keep the formulation short, two simplifying assumptions are made. First, 2-D (rather than 3-D) localization is addressed. Second, the unknown channel and device parameters (orientation for AOA, transmit powers and n_p for RSS and QRSS, and clock biases for TOA) are assumed to be known. Analysis of bounds without these assumptions are left to the references [44, 76, 77, 17, 68] which have presented details of these analytical CRBs, for a variety of different measurement types.

Furthermore, to keep this section short and to the point, the detailed derivations are presented separately in Appendix A.

3.1.1 What is the Cramér-Rao Bound?

The Cramér-Rao bound (CRB) provides a lower bound on the variance achievable by any unbiased location estimator [122]. The bound is useful as a guideline:

knowing the best an estimator can possibly do helps us judge the estimators that are implemented. Essentially, the CRB is a general uncertainty principle which is applied in this chapter to location estimation.

A detailed derivation of the CRB is provided in [122]. This section purely serves as an overview for those without *a priori* knowledge of the CRB. The most important benefit of the CRB is that the lower bound on estimation variance can be calculated without ever considering a single estimation method. All that is needed to calculate a CRB is the statistical model of the random measurements, *i.e.*, $f\left(\mathbf{X}|\boldsymbol{\theta}\right)$, where $\mathbf{X}$ is the random measurement, and $\boldsymbol{\theta}$ are the parameters that are to be estimated from the measurements. Any unbiased estimator, $\hat{\boldsymbol{\theta}}$, must satisfy

$$\mathrm{Cov}(\hat{\boldsymbol{\theta}}) \geq \left\{\mathrm{E}\left[-\nabla_{\boldsymbol{\theta}}(\nabla_{\boldsymbol{\theta}} \ln f\left(\mathbf{X}|\boldsymbol{\theta}\right))^{T}\right]\right\}^{-1}, \tag{3.1}$$

where $\mathrm{Cov}(\hat{\boldsymbol{\theta}})$ is the covariance of the estimator, $\mathrm{E}[\cdot]$ indicates expected value, $\nabla_{\boldsymbol{\theta}}$ is the gradient operator w.r.t. the vector $\boldsymbol{\theta}$, and superscript ^{T} indicates transpose.

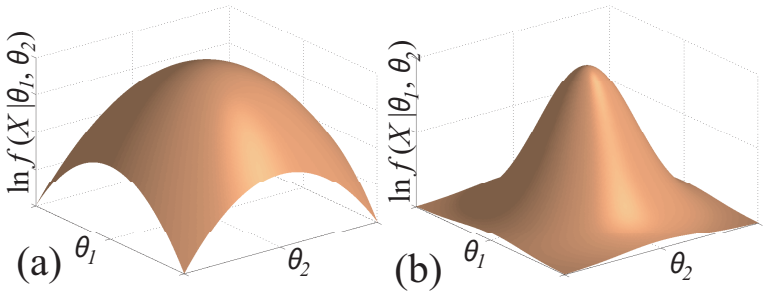


Figure 3.1: Example log-likelihood functions for two-parameter estimation with (a) small and (b) large curvature. The variance bound will be higher in example (a) than in (b).

The bound is very similar to sensitivity analysis, applied to random measurements. The CRB is based on the curvature of the log-likelihood function, $\ln f\left(\mathbf{X}|\boldsymbol{\theta}\right)$. Intuitively, if the curvature of the log-likelihood function is very sharp like the example plot in Fig. 3.1(b), then the optimal parameter estimate can be accurately

identified. Conversely, if the log-likelihood is broad with small curvature like the graph in Fig. 3.1(a), then estimating the optimal will be more difficult.

The CRB is limited to unbiased estimators. Such estimators provide coordinate estimates that, if averaged over enough realizations, are equal to the true coordinates. Although unbiased estimation is a very desirable property, some bias might be tolerated in order to reduce variance – in such cases, the bound can be adapted [54].

3.2 Decreasing Bound on Self-Calibration Estimators

A fundamental question regarding 'cooperative' localization is whether or not adding unknown-location sensors to the network increases the precision of the location estimates. This is a critical question, because it relates to the claim that more dense sensor networks will provide better localization estimates.

In this section, a single additional unknown-location sensor is added to an existing N-sensor network. It is shown that, given the sufficient conditions presented, the lower bound on the covariance of original N sensors' coordinates strictly decreases. While this doesn't show that the covariance approaches zero (or any particular limit), it does show that adding additional sensors does, in fact, allow more precise localization estimates.

In fact, this theorem applies more generally to a larger class of network estimation problems called 'self-calibration' estimators. In network self-calibration problems each device in a network has a parameter (or parameters) which must be determined. In these problems, information comes both from measurements made between pairs of devices and a subset of devices which know *a priori* their parameters. A network self-calibration estimator estimates the remaining, unknown device pa-

rameters. Another example, besides sensor localization, is the problem of distributed clock synchronization in a network, which could be achieved by devices observing pair-wise timing offsets when just a small number (≥ 1) of devices are synchronous.

3.2.1 Self-Calibration Estimation

Specifically, consider a vector of device parameters $\boldsymbol{\gamma} = [\gamma_1, \ldots, \gamma_{n+m}]$. Here, each device is assumed to have one parameter, but note that the results would equally apply if γ was a vector of parameters. Devices $1 \ldots n$ are unknown devices (previously called unknown-location devices, but now more general than just location) and devices $n+1 \ldots n+m$ are reference devices. The unknown parameter vector is $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_n]$ where $\theta_i = \gamma_i$ for $i = 1 \ldots n$. Note $\{\gamma_i : i = n+1 \ldots n+m\}$ are known. Devices i and j make pair-wise observations $X_{i,j}$ with density $f_{X|\boldsymbol{\gamma}}(X_{i,j}|\gamma_i, \gamma_j)$. Devices are permitted to make incomplete observations, since two devices may be out of range or have limited link capacity. Let $H(i) = \{j : \text{device } j \text{ makes pair-wise observations with device } i\}$. By convention, a device cannot make a pair-wise observation with itself, so that $i \notin H(i)$. By symmetry, if $j \in H(i)$ then $i \in H(j)$.

By reciprocity, it is assumed that $X_{i,j} = X_{j,i}$, thus it is sufficient to consider only the lower triangle of the observation matrix $\mathbf{X} = ((X_{i,j}))_{i,j}$ when formulating the joint likelihood function. In practice, if it is possible to make independent observations on the links from i to j, and from j to i, it is assumed that a scalar sufficient statistic can be found. Finally, it is assumed $\{X_{i,j}\}$ are statistically independent for $j < i$. This assumption can be somewhat oversimplified (see [94] for the RSS case) but necessary for analysis. Using measurements like those presented in Section 2.6 remains important to verify true performance.

The log of the joint conditional pdf is

$$l(\mathbf{X}|\boldsymbol{\gamma}) = \sum_{i=1}^{m+n} \sum_{\substack{j \in H(i) \\ j<i}} l_{i,j}, \quad \text{where } l_{i,j} = \log f_{X|\boldsymbol{\gamma}}(X_{i,j}|\gamma_i, \gamma_j). \tag{3.2}$$

The CRB on the covariance matrix of any unbiased estimator $\hat{\boldsymbol{\theta}}$ is $\text{cov}(\hat{\boldsymbol{\theta}}) \geq \mathbf{F}_{\boldsymbol{\theta}}^{-1}$, where the Fisher information matrix (FIM) $\mathbf{F}_{\boldsymbol{\theta}}$ is defined as,

$$\mathbf{F}_{\boldsymbol{\theta}} = -\text{E}\nabla_{\boldsymbol{\theta}}(\nabla_{\boldsymbol{\theta}} l(\mathbf{X}|\boldsymbol{\gamma}))^T = \begin{bmatrix} f_{1,1} & \cdots & f_{1,n} \\ \vdots & \ddots & \vdots \\ f_{n,1} & \cdots & f_{n,n} \end{bmatrix} \tag{3.3}$$

As derived in Appendix A, the diagonal elements $f_{k,k}$ of $\mathbf{F}_{\boldsymbol{\theta}}$ reduce to a single sum over $H(k)$, since there are $\text{card}\{H(k)\}$ terms in (3.2) which depend on $\theta_k = \gamma_k$. The off-diagonal elements can be further reduced: when $k \neq l$, there is at most one summand in (3.2) which is a function of both k and l.

$$f_{k,l} = \begin{cases} -\sum_{j \in H(k)} \text{E}\left[\frac{\partial^2}{\partial \theta_k^2} l_{k,j}\right], & k = l \\ -\text{I}_{H(k)}(l)\text{E}\left[\frac{\partial^2}{\partial \theta_k \partial \theta_l} l_{k,l}\right], & k \neq l \end{cases} \tag{3.4}$$

where $\text{I}_{H(k)}(l)$ is an indicator function, 1 if $l \in H(k)$ or 0 otherwise.

3.2.2 Conditions for a decreasing CRB

Intuitively, as more devices are used for location estimation, the accuracy increases for all of the devices in the network. For an N device network, there are $O(N)$ parameters, but $O(N^2)$ variables $\{X_{i,j}\}$ used for their estimation. The analysis of this section gives sufficient conditions to ensure the CRB decreases as devices are added to the network. Consider a network of n unknown devices and m reference devices, and define $N = n+m$. Now consider adding one additional unknown device. For the n and $(n+1)$ unknown device cases, let $\mathbf{F}$ and $\mathbf{G}$ be the FIMs defined in (3.3), respectively. (In other words, $\mathbf{F}$ is an $n \times n$ matrix, and $\mathbf{G}$ is an $n+1 \times n+1$ matrix.)

Theorem III.5. *Let* $[\mathbf{G}^{-1}]_{ul}$ *be the upper left* $n \times n$ *block of* $\mathbf{G}^{-1}$*. If for the* $(n+1)$ *unknown device case:*

Condition 1*:* $\frac{\partial}{\partial \theta_{n+1}} l_{k,n+1} = \pm \frac{\partial}{\partial \theta_k} l_{k,n+1}, \forall k = 1 \ldots n$ *and*

Condition 2*: device* $n+1$ *makes pair-wise observations between itself and at least one unknown device and at least two devices, in total;*

then two properties hold:

Property 1*:* $\mathbf{F}^{-1} - [\mathbf{G}^{-1}]_{ul} \geq 0$ *in the positive semi-definite sense, and*

Property 2*:* $tr\,\mathbf{F}^{-1} > tr\,[\mathbf{G}^{-1}]_{ul}$.

Theorem III.5 is proven in Appendix A.2. The Gaussian and log-normal distributions in Sections 2.2, 2.3, and 2.4 meet Condition (1). Property (1) implies that the additional unknown parameter introduced by the $(n+1)^{st}$ unknown device does not impair the estimation of the original n unknown parameters. Furthermore, Property (2) implies that the sum of the CRB variance bounds for the n unknown parameters strictly decreases. Thus when an unknown device enters a network and makes pair-wise observations with at least one unknown device and at least two devices in total, the bound on the average variance of the original n coordinate estimates is reduced. Note that Properties (1) and (2) of Theorem 1 would be trivially satisfied by the data processing theorem if adding a device into the network did not increase the number of parameters.

3.3 The Cooperative Localization CRB

In this section, the self-calibration lower bound analysis in Section 3.2 is applied specifically to the localization estimation problem originally stated in Section 1.3. In particular, 2-D coordinate estimation is considered, when the measurements $X_{i,j}$ are RSS, QRSS, connectivity, TOA, or AOA. As it turns out, the formulation of the

variance bounds for these various measurements is remarkably similar. Particular differences can be seen that show how localization performance varies by measurement type.

Detailed derivations are done separately for each measurement type in the Appendices. However, in order to readily be able to compare the bounds over all measurement types, a unified calculation method is presented. Specifically, calculation of the CRB for estimation of $\boldsymbol{\theta}$ as given in (1.1) is described in three steps. The reader should refer to the Appendix for the detailed derivation of each lower bound (TOA, RSS, AOA, and K-level QRSS).

3.3.1 Calculate Fisher information sub-matrices

First, form three $n \times n$ matrices: $\mathbf{F}_{xx}$, $\mathbf{F}_{xy}$, and $\mathbf{F}_{yy}$. As introduced in Section 1.3, n is the number of unknown-location sensors. The k, l element, for $k, l \in \{1, \ldots, n\}$ of each matrix is calculated as follows:

$$
\begin{aligned}
\left[\mathbf{F}_{xx}\right]_{k,l} &= \begin{cases} \gamma \sum_{i \in H(k)} h_{k,i}(x_k - x_i)^2 / \|\mathbf{z}_k - \mathbf{z}_i\|^s & k = l \\ -\gamma\, I_{H(k)}(l) h_{k,l}(x_k - x_l)^2 / \|\mathbf{z}_k - \mathbf{z}_l\|^s & k \neq l \end{cases} \\
\left[\mathbf{F}_{xy}\right]_{k,l} &= \begin{cases} \gamma \sum_{i \in H(k)} h_{k,i}(x_k - x_i)(y_k - y_i) / \|\mathbf{z}_k - \mathbf{z}_i\|^s & k = l \\ -\gamma\, I_{H(k)}(l) h_{k,l}(x_k - x_l)(y_k - y_l) / \|\mathbf{z}_k - \mathbf{z}_l\|^s & k \neq l \end{cases} \\
\left[\mathbf{F}_{yy}\right]_{k,l} &= \begin{cases} \gamma \sum_{i \in H(k)} h_{k,i}(y_k - y_i)^2 / \|\mathbf{z}_k - \mathbf{z}_i\|^s & k = l \\ -\gamma\, I_{H(k)}(l) h_{k,l}(y_k - y_l)^2 / \|\mathbf{z}_k - \mathbf{z}_l\|^s & k \neq l \end{cases}
\end{aligned}
\tag{3.6}
$$

Here, γ is a channel constant, and s is an exponent, which are both a function of the measurement type and are given in Table 3.1, and $I_{H(k)}(l)$ is the indicator function, (which allows us to include the information only if sensor k made a measurement with sensor l), $I_{H(k)}(l) = 1$ if $l \in H(k)$, or 0 if not. Also, $h_{k,l}$ is a loss term due to quantization, which is equal to one for TOA, RSS, and AOA, since they are assumed

to use unquantized measurements, and for QRSS, $h_{k,l}$ is given by

$$h_{i,j} = \frac{1}{2\pi} \sum_{s=0}^{K-1} \frac{\left[\exp\left(-\frac{1}{2}g_{i,j}^2(s+1)\right) - \exp\left(-\frac{1}{2}g_{i,j}^2(s)\right)\right]^2}{\Phi\left(-g_{i,j}(s+1)\right) - \Phi\left(-g_{i,j}(s)\right)} \tag{3.7}$$

where $g_{i,j}(s)$ was given in (2.14), $\Phi(x)$ is the CDF of the standard normal, and distance thresholds d_s are given by (2.16). Note that this can be used for any K-level QRSS measurements, but in particular, 2-level QRSS represents connectivity measurements. For $K = 2$, (3.7) simplifies considerably,

$$h_{i,j} = \frac{\exp[-g_{i,j}^2(1)]}{2\pi\Phi\left[-g_{i,j}(1)\right]\left\{1 - \Phi\left[-g_{i,j}(1)\right]\right\}}. \tag{3.8}$$

3.3.2 Merge sub-matrices to form the FIM

Next, form the $2n \times 2n$ Fisher information matrix (FIM) $\mathbf{F}$ corresponding to the $2n$ coordinates in $\boldsymbol{\theta}$ that need to be estimated. For TOA, RSS, or QRSS, select $\mathbf{F} = \mathbf{F_{TR}}$, while for AOA, select $\mathbf{F} = \mathbf{F_A}$, where

$$\mathbf{F_{TR}} = \begin{bmatrix} \mathbf{F}_{xx} & \mathbf{F}_{xy} \\ \mathbf{F}_{xy}^T & \mathbf{F}_{yy} \end{bmatrix}, \quad \mathbf{F_A} = \begin{bmatrix} \mathbf{F}_{yy} & -\mathbf{F}_{xy} \\ -\mathbf{F}_{xy}^T & \mathbf{F}_{xx} \end{bmatrix}, \tag{3.9}$$

where $\mathbf{F}_{xx}$, $\mathbf{F}_{xy}$, and $\mathbf{F}_{yy}$ are given in (3.6), and the superscript ^{T} indicates matrix transpose.

3.3.3 Invert the FIM to get the CRB

The CRB matrix is equal to $\mathbf{F}^{-1}$, the matrix inverse of the FIM. The diagonal of $\mathbf{F}^{-1}$ contains $2n$ values which are the variance bounds for the $2n$ parameters of $\boldsymbol{\theta}$. To say this more precisely, let an estimator of sensor i's coordinates be $\hat{\mathbf{z}}_i = [\hat{x}_i, \hat{y}_i]^T$. If the location variance of the estimator is defined to be σ_i^2,

$$\sigma_i^2 \triangleq \text{tr}\left\{\text{cov}_{\boldsymbol{\theta}}(\hat{\mathbf{z}}_i)\right\} = \text{Var}_{\boldsymbol{\theta}}(\hat{x}_i) + \text{Var}_{\boldsymbol{\theta}}(\hat{y}_i), \tag{3.10}$$

then the Cramér-Rao bound asserts that,

$$\sigma_i^2 \;\geq\; \left(\mathbf{F}^{-1}\right)_{i,i} + \left(\mathbf{F}^{-1}\right)_{i+n,i+n}. \tag{3.11}$$

Table 3.1: Differences in CRB by Measurement Type.

	Channel Constant γ	Exponent s	FIM $\mathbf{F}$	Quantization Loss h
TOA	$\gamma = 1/(v_p\sigma_T)^2$	$s = 2$	$\mathbf{F} = \mathbf{F_{TR}}$	$h_{k,l} = 1$
RSS	$\gamma = \left(\frac{10n_p}{\sigma_{dB}\log 10}\right)^2$	$s = 4$	$\mathbf{F} = \mathbf{F_{TR}}$	$h_{k,l} = 1$
QRSS	$\gamma = \left(\frac{10n_p}{\sigma_{dB}\log 10}\right)^2$	$s = 4$	$\mathbf{F} = \mathbf{F_{TR}}$	$h_{k,l}$ given in (3.7)
Connectivity	$\gamma = \left(\frac{10n_p}{\sigma_{dB}\log 10}\right)^2$	$s = 4$	$\mathbf{F} = \mathbf{F_{TR}}$	$h_{k,l}$ given in (3.8)
AOA	$\gamma = 1/\sigma_\alpha^2$	$s = 4$	$\mathbf{F} = \mathbf{F_A}$	$h_{k,l} = 1$

3.3.4 Results Seen from the CRB

Even without calculating the CRB for a particular sensor network geometry, the scaling characteristics of the variance bound can be explored. What happens when the geometry and connectivity of the network is kept constant, but the dimensions of the network are scaled up proportionally?

- *TOA*: TOA bounds will remain constant with a scaling of the dimensions. Note that since $s = 2$ for TOA, the fractions in (3.6) are unitless - if units of the coordinates were (ft) or even (cm) instead of (m), the ratios would be identical. Instead, the units come from the standard deviation of ranging error, $v_p\sigma_T$.

- *RSS, and AOA*: These bounds on standard deviation are proportional to the size of the system. Since $s = 4$ for RSS and AOA, the geometry ratios in (3.6) have units of 1/distance^2, so the variance bound (the inverse) takes its units of distance^2 directly from this ratio. Note that the channel constant γ is unitless for both RSS and AOA.

- *QRSS*: The CRB for quantized RSS and connectivity have the same proportionality as RSS, except for the terms $\{h_{i,j}\}$. However, the only dependency of $h_{i,j}$ is on the ratio between inter-sensor distances and the threshold distance, $\|\mathbf{z}_i - \mathbf{z}_j\|/d_s$. Thus if the threshold distances d_s are scaled proportionally with the size of the network, then the $h_{i,j}$ terms remain constant, and allow QRSS bounds to scale proportional to the size of the network, just like RSS.

Of course, channel parameters will change slowly as the path lengths change (TOA measurements over kilometer links would have higher variance than over 10 m links), but these scaling characteristics are good first order approximations.

For connectivity measurements, the term $h_{i,j}$ in has a maximum of $2/\pi \approx 0.64$ when $g_{i,j}(1) = 0$, which happens when devices i and j are separated by approximately the threshold distance d_1. Compared to the FIM for RSS measurements, the connectivity measurements have the same FIM except for the term $h_{i,j}$. Thus, 2-level quantized measurement of RSS in this (best) case contains 64% of the information in the unquantized RSS measurement. The inverse of the Fisher information is the variance bound, and in this case, it would be $pi/2 \approx 1.57$ times the variance bound for the unquantized RSS measurement. This analysis also shows explicitly that the information is highest when two devices are separated by approximately the threshold distance. Setting the threshold distance for connectivity-based localization is more explicitly explored in the examples in Section 3.4.

Finally, note that the bound on standard deviation of localization error is proportional to $\sqrt{1/\gamma}$. It makes sense that the localization error is proportional to σ_T for TOA and σ_α for AOA. It is not as obvious, but from the CRB for RSS, it is seen that the proportionality is to σ_{dB}/n_p for RSS-based localization. An RSS-based localization system operating in a high path-loss exponent environment, while requiring

higher transmit powers from sensors, also allows more accurate sensor localization. For example, the author has measured $n_p \approx 5$ when placing wireless sensors on a lawn, due to both the ground-level antennas and the absorption of the grass. Such high n_p can be viewed as a favorable environment for accurate localization.

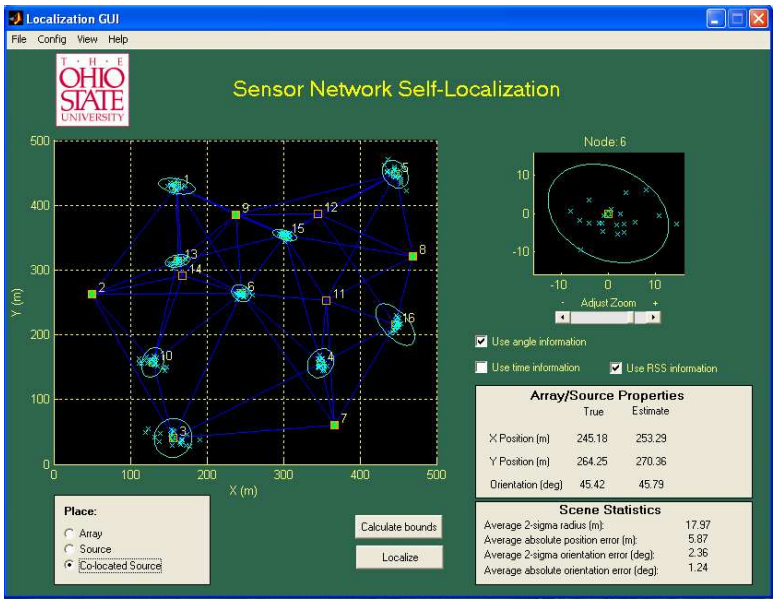


Figure 3.2: Lower bounds and Monte-carlo ML estimates can be calculated interactively using this Matlab-based GUI which is freely available online [5]. Sensors can be placed arbitrarily, and their capabilities and *a priori* location information given. The user may select any combination of AOA, TOA, and RSS measurements.

3.4 Numerical Examples

Figures 3.3 and 3.7 show two examples which are used throughout this thesis: the single unknown-location sensor example, and the $\sqrt{N}$ by $\sqrt{N}$ grid example. Other, more random deployments are also tested, but are presented along with simulation results in Chapter IV. For this chapter, two simple geometries are used in order to make observations about the relative performance of the many measurement modalities.

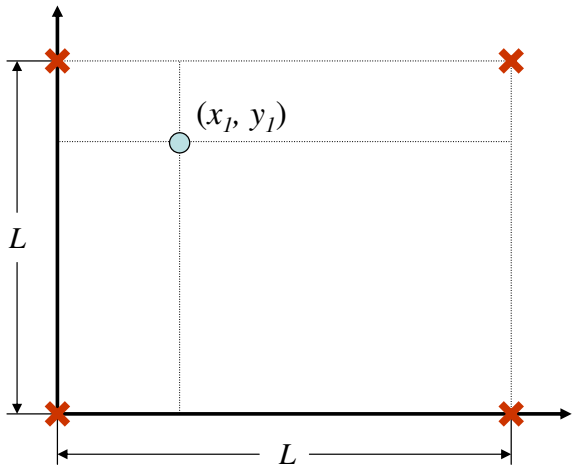


Figure 3.3: Diagram showing layout of the $N = 5$ sensors from the single unknown-location sensor example described in Section 3.4.1, with four reference sensors ($\times$) at the corners of a $L \times L$ square area and one unknown-location sensor ($\bullet$) at $\mathbf{z}_1 = [x_1, y_1]^T$.

3.4.1 Single Unknown-Location Node Example

Although this thesis considers predominantly 'cooperative' localization, the CRB derived here applies equally well to the traditional multi-lateration techniques in which one unknown-location device must be able to make measurements with multiple reference devices. In this example, the network consists of unknown-location device 1 and reference devices $2 \ldots m+1$, and device 1 makes measurements with all m reference devices. Specifically, bounds are calculated for the case in which $m = 4$ reference devices are located in the corners of a L meter by L meter square area, as shown in Fig. 3.3. For now, $L = 1$m is assumed, but the scaling properties of the measurement modalities described in Section 3.3.4 describe the CRB as a function of L. The unknown-location device makes measurements with all four reference devices. Further, the value of the parameter ratio, $\sigma_{dB}/n_p = 1.7$, is used to match the values obtained from Measurement Experiment II. For the case of RSS measurements, the lower bound for σ_1 is calculated and plotted as a function of unknown-location device location in Fig. 3.4(a). The minimum of the CRB for σ_1 for the case of RSS

measurements is 0.27 m, and the average bound within the square is 0.305 m.

For proximity or QRSS measurements, the bound is a function of the threshold distances $\{d_s\}$. For a system using proximity measurements, designers can select the threshold distance d_1 by changing either the sensitivity of the receiver or the transmit power level. For a K-level QRSS system, a designer must additionally set d_s for $s = 2 \dots K-1$, which can be done either by design of the A/D converter in the receiver, or by design of the power amplifier transmit power levels. Other system considerations, for example, network connectivity and energy efficiency, must also be considered when setting these parameters. In this example, in order to present a universal lower bound, d_1 and $d_2 \dots d_{K-1}$ as the distances that minimize the lower bound σ_1. Specifically, this bound is minimized for the case when the unknown-location device is located in the center of the square area, ie., $\mathbf{z}_1 = [0.5, 0.5]^T$ m.

Proximity

Since the unknown-location device is located equidistant from all of the reference devices, the analytical expression for the CRB simplifies considerably. In particular, the CRB in the proximity case is minimized when d_1 is equal to the distance between the unknown-location device and any of the reference devices, ie., $d_1 = 1/\sqrt{2}$ m. In this case, the CRB is given by,

$$\sigma_1^2 \geq \frac{\pi}{4} \left(\frac{\sigma_{dB} \log 10}{10 n_p} \right)^2 . \tag{3.12}$$

For $\sigma_{dB}/n_p = 1.7$, the bound on the standard deviation σ_1 is 0.3477 m. This is verified graphically in Fig. 3.4(b), which plots the bound on the standard deviation of unbiased location estimates as a function of the location of the unknown-location device, while the design parameter $d_1 = 1/\sqrt{2}$ m is kept constant. Furthermore, the average standard deviation bound within the square is 0.45 m. Note that the average

standard deviation bound using proximity measurements is 48% worse as compared to the bound obtained using RSS measurements.

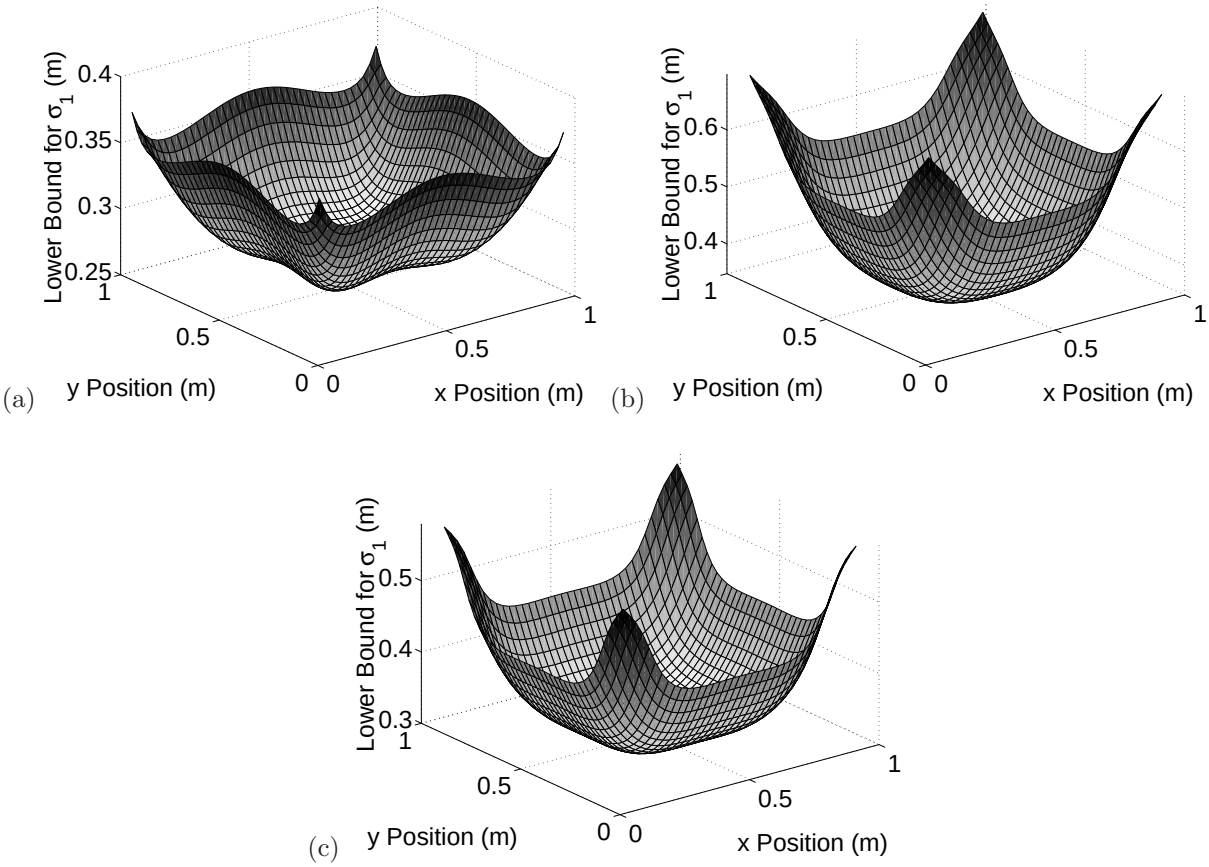


Figure 3.4: Lower bound for σ_1 (m) for the single unknown-location device system vs. the coordinates of the unknown-location device, in a channel with $\sigma_{dB}/n_p = 1.7$, for (a) RSS, (b) proximity with $d_1 = 1/\sqrt{2}$ m and (c) 3-level QRSS with $d_1 = 0.90$ m and $d_2 = 0.56$ m.

Three-Level QRSS

Next, consider the performance of the system in the case of $K = 3$ QRSS measurements. Again, the system is optimized to minimize the CRB when the unknown-location device is located at $\mathbf{z}_1 = [0.5, 0.5]^T$ m. It can be shown that the CRB as a

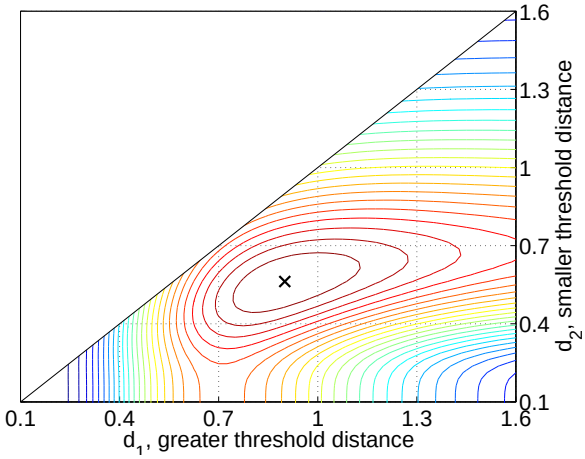


Figure 3.5: The Fisher information $f_{1,1}$ for the case of 3-level QRSS when a single unknown-location device is located in the center of a 1m by 1m square with reference devices in each corner. The FIM is plotted as a function of the two thresholds, d_2 and d_1, which separate the three QRSS regions. The maximum of $f_{1,1}$ (x) is at $d_1 = 0.90$ m and $d_2 = 0.56$ m. Since $d_2 \leq d_1$, only half of the plot is shown.

function of the two threshold distances, d_1 and d_2, is given by,

$$\sigma_i^2 \geq \frac{2}{f_{1,1}} \tag{3.13}$$

$$f_{1,1} = \frac{2b}{\pi}\left\{\frac{\exp\left[-b\ln^2\frac{d_a}{d_1}\right]}{\Phi\left(-\sqrt{b}\ln\frac{d_a}{d_1}\right)} + \frac{\exp\left[-b\ln^2\frac{d_a}{d_2}\right]}{\Phi\left(-\sqrt{b}\ln\frac{d_a}{d_2}\right)} + \frac{\left[\exp\left(-\frac{b}{2}\ln^2\frac{d_a}{d_2}\right) - \exp\left(-\frac{b}{2}\ln^2\frac{d_a}{d_1}\right)\right]^2}{\Phi\left(-\sqrt{b}\ln\frac{d_a}{d_2}\right) - \Phi\left(-\sqrt{b}\ln\frac{d_a}{d_1}\right)}\right\}$$

where $d_a = 1/\sqrt{2}$ m, the distance between the unknown-location device and any reference device. The term $f_{1,1}$ is equal to $\mathbf{F}_{xx}$, ie., the Fisher information for the x-coordinate. Note the notation $\ln^2 x$ is used to indicate $(\ln x)^2$. From (3.13), d_1 and d_2 are selected to maximize $f_{1,1}$. For a range of d_1 and d_2, the Fisher information $f_{1,1}$ is plotted in Fig. 3.5.

For three cases, the three-level QRSS Fisher information $f_{1,1}$ reverts to the two-level proximity Fisher information. These cases are (1) when $d_1 = d_2$, (2) when d_1 is very large, and (3) when d_2 is very small. Intuitively, in any of these three cases,

there are effectively only two levels, and the system reverts to a proximity system. Fig. 3.5 shows this graphically. The value of $f_{1,1}$ along the diagonal $d_1 = d_2$ can be seen to be the same as $f_{1,1}$ along the horizontal line at the lowest d_2, and along the vertical line at the highest d_1.

The maximum of $f_{1,1}$ occurs for $d_1 = 0.90$ m and $d_2 = 0.56$ m. For these two parameters, the bound on the standard deviation σ_i is 0.3076 m when the unknown-location device is at $\mathbf{z}_1 = [0.5, 0.5]^T$ m. The CRB as a function of unknown-location device location is plotted in Fig. 3.4(c). Furthermore, the average standard deviation bound within the square is 0.37 m.

Note that in any of the three cases presented in this section, the CRB scales with the size of the system. If instead the square area had sides of length L, and the threshold distances d_s were scaled by L, then the bound on σ_i would also be scaled by L.

TOA

The case of localization using TOA measurements when there is one unknown-location device and m reference devices has been well-studied in the literature [121, 111, 114], and the results in [114] are specifically used to validate the bound presented in this thesis. From the CRB, the lower bound when there is $n = 1$ unknown-location device and m reference devices is,

$$\sigma_1^2 = v_p^2 \sigma_T^2 m \left[\sum_{i=2}^{m} \sum_{j=i+1}^{m+1} \left(\frac{d_{1\perp i,j} \|\mathbf{z}_i - \mathbf{z}_j\|}{\|\mathbf{z}_1 - \mathbf{z}_i\| \, \|\mathbf{z}_1 - \mathbf{z}_j\|} \right)^2 \right]^{-1}, \tag{3.14}$$

where $d_{1\perp i,j}$ is the shortest distance from $\mathbf{z}_1$ to the line which intersects $\mathbf{z}_i$ and $\mathbf{z}_j$. The ratio $d_{1\perp i,j} \, \|\mathbf{z}_i - \mathbf{z}_j\| / (\|\mathbf{z}_1 - \mathbf{z}_i\| \, \|\mathbf{z}_1 - \mathbf{z}_j\|)$ has been called the geometric conditioning $\mathcal{A}_{i,j}$ of device 1 w.r.t. references i and j [114]. $\mathcal{A}_{i,j}$ is the area of the parallelogram specified by the vectors from device 1 to i and from device 1 to j, normalized by the

lengths of the two vectors. The geometric dilution of precision (GDOP), defined as $\sigma_1/(v_p\sigma_T)$, is

$$GDOP = \sqrt{\frac{m}{\sum_{i=2}^{m}\sum_{j=i+1}^{m+1}\mathcal{A}_{i,j}^2}} \tag{3.15}$$

which exactly matches the result in [114].

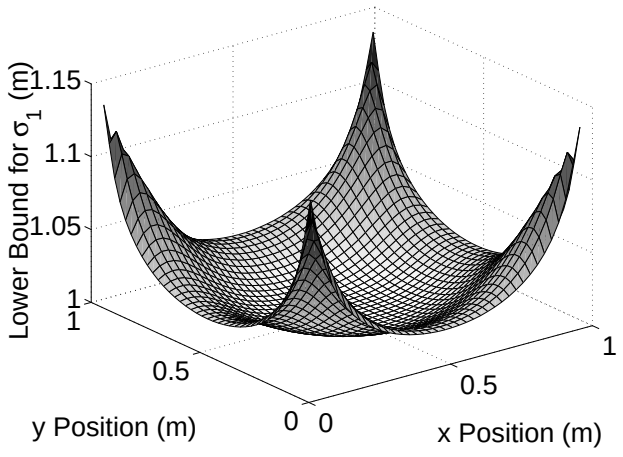
Fig. 3.6 plots σ_1 as a function of $\mathbf{z}_1 = [x_1, y_1]^T$ for case when the $m = 4$ reference devices are located in the corners of a 1 m by 1 m square. Note in the TOA case, σ_1 is proportional to $v_p\sigma_T$, thus $v_p\sigma_T = 1$ was chosen in Fig. 3.6 so that the bound could be calculated more easily for arbitrary $v_p\sigma_T$. Finally, σ_1, does not scale if the size of the square was L by L meters rather than 1 by 1 m.

3.4.2 $\sqrt{N}$ by $\sqrt{N}$ Grid Example

Consider a sensor network in a L by L area, with N sensors arranged into $\sqrt{N}$ rows and $\sqrt{N}$ columns, as shown in Fig. 3.7. The four sensors in the corners are reference nodes, while the remaining $N-4$ are unknown-location nodes. This section considers what happens to the localization variance bound as $\sqrt{N}$ increases, for $L = 20$m, when measurements are:

1. RSS with $\sigma_{dB}/n_p = 1.7$, obtained from Measurement Experiment II [92],
2. TOA with $\sigma_T = 6.1$ ns and $v_p = 3 \cdot 10^8 \frac{\mathrm{m}}{\mathrm{s}}$, also obtained from Measurement Experiment II, and
3. AOA with $\sigma_\alpha = 5^\circ$ (see Section 2.4.1).

As presented above, the lower bound on localization standard deviation for RSS, TOA, and AOA are proportional to these three channel parameters. This simulation is begun with the full-connectivity assumption, *i.e.*, that each sensor makes measurements with every other sensor in the network. The RMS value of the localization



(b)

Figure 3.6: σ_1 (m) for the example system vs. the coordinates of the single unknown-location device, when measurements are TOA with $v_p\,\sigma_T = 1$m.

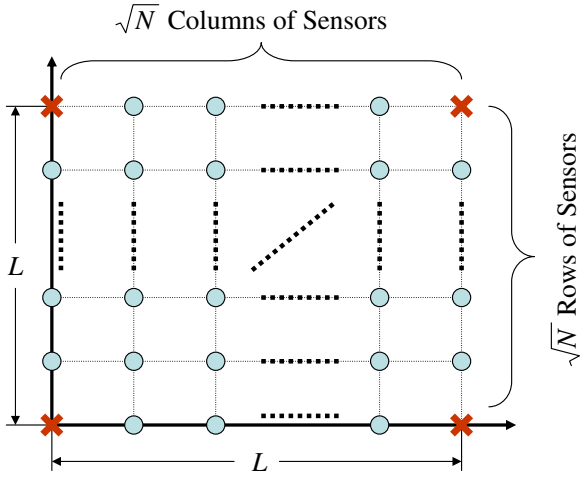


Figure 3.7: Diagram showing layout of the N sensors from the example described in Section 3.4.2, with four reference sensors ($\times$) and $N-4$ unknown-location sensors ($\bullet$) in a $L \times L$ square area.

bound is calculated, *i.e.*, $\sqrt{\frac{1}{n}\text{tr}\,\mathbf{F}^{-1}}$, which gives an average of the bound over the entire $K^2 - 4$ unknown-location sensors. The result is shown in the solid lines in Fig. 3.8 labelled as '$r = \infty$'. Next, the simulation is changed to consider the realistic case in which each sensor only makes measurements to those sensors located within $r = 10$ m of itself[1]. In this case, the bound is shown as dotted lines in Fig. 3.8 and labelled as '$r = 10$ m'.

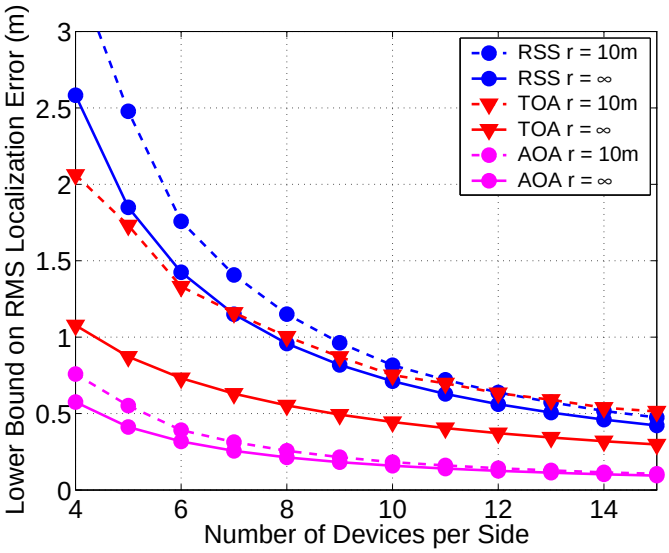


Figure 3.8: Lower bounds for localization standard deviation for the example described in Section 3.4.2 when measurements are RSS (with $\sigma_{dB}/n_p = 1.7$ [92]), TOA (with $\sigma_T = 6.3$ns [92]), and AOA (with $\sigma_\alpha = 5°$). Parameter r is the radius of connectivity - only pairs of sensors closer than r make measurements, and for $r = \infty$, all pairs make measurements.

Comparing performance of the measurement methods for the chosen parameters, AOA outperforms TOA and RSS, while RSS can perform as well as TOA at high sensor densities. Of course, these comparisons are based on the chosen values of the measurement parameters and the chosen geometry shown in Fig. 3.7. As described in Section 3.3.4, these bounds are proportional to $1/\sqrt{\gamma}$ where γ is the channel constant given in Table 3.1. For example, if it was assumed that $\sigma_\alpha = 10°$ instead of 5°, the standard deviation bound for AOA would be twice that shown in Fig. 3.8. Note that

[1]Of course, sensors will not really know *exactly* which sensors are within 10 m, but the connectivity implied by the 10 m radius provides a realistic test.

RSS and AOA bounds decrease more rapidly than TOA as the density increases. Also for RSS and AOA, the difference between the $r = 10$m and $r = \infty$ lines decreases dramatically as density increases. At high densities, the results show that little additional information comes from the distant sensors (> 10 m). For TOA, however, even distant sensors' measurements can provide significant localization information.

Connectivity Bound in Grid Example

For QRSS measurements, the lower bound is a function of distance thresholds. Consider the CRB for connectivity, *i.e.*, 2-level QRSS, in the $\sqrt{N}$ by $\sqrt{N}$ grid example described above, as $\sqrt{N}$ increases. If designing a system for a particular density of devices, N, a system designer should optimize the threshold for that particular density. What is the optimal threshold as a function of N? This relationship is determined by simulation as shown in Fig. 3.9.

First, an optimal d_1 for each $\sqrt{N} = 3 \ldots 10$ is found by calculating $\bar{\sigma}$ for thresholds in the range $[0.1, 0.8]$ in increments of 0.01. The results, displayed in Fig. 3.9, show the optimal d_1 decreasing from 0.64 for $\sqrt{N} = 3$ to 0.21 for $\sqrt{N} = 10$. Note, however, that the ratio of the minimum $\bar{\sigma}$ and the distance between devices in the grid $(\sqrt{N} - 1)^{-1}$ is almost constant. That is, $(\sqrt{N} - 1) \min_{d_1} \bar{\sigma} \approx 0.60$ for each $\sqrt{N}$ in the range tested. Since the standard deviation bound scales proportionally with system size, as a rule of thumb, the lower bound on the RMS standard deviation is about 60% of the distance between devices.

Quantization Levels in QRSS in the Grid Example

In this section, CRB results for K-level QRSS are presented as a function of K. This analysis is done for the 5 by 5 grid example ($N = 25$) with $L = 1$ m. Results have already been presented for the case of $K = 2$ (connectivity) and the case of

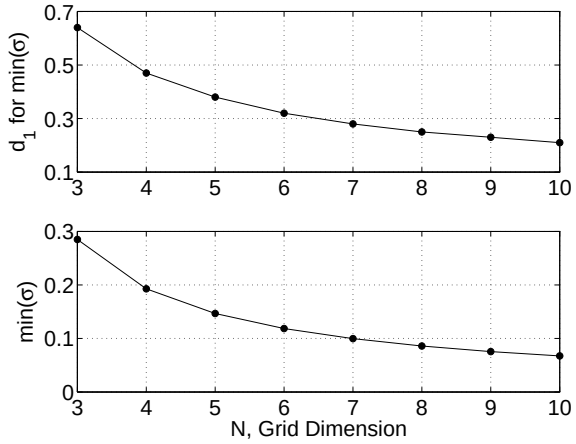


Figure 3.9: For proximity measurements in the $\sqrt{N}$ by $\sqrt{N}$ grid example, the minimum $\bar{\sigma}$ vs. $\sqrt{N}$, and the d_1 used to achieve that minimum.

unquantized RSS measurements, which should match the asymptotic performance of QRSS as $K \to \infty$.

Ideally, for a particular K, the $K-1$ thresholds $\{P_s\}_{s=1\ldots K-1}$ which minimize the bound on $\bar{\sigma}$ should be found. However, as K becomes large, finding a minimum in this $K-1$ dimensional space becomes difficult. In addition, in a real low-cost implementation, the quantization design of RSS is unlikely to be non-uniform. Specifically, RSS is often quantized on a log scale with a constant granularity in (dB). Thus it is reasonable to limit the search space to two parameters: the mean of P_s, ie. $\bar{P} = \frac{1}{K-1}\sum_{s=1}^{K-1} P_s$, and $\Delta P \triangleq P_{s+1} - P_s, \forall s = 1 \ldots K-2$. Equivalently, the geometric mean of d_s can be used, ie. $d_{mean} \triangleq (\prod_{s=1}^{K-1} d_s)^{1/(K-1)}$, and the parameter d_{ratio}, defined as $d_s/d_{s+1}, \forall s = 1 \ldots K-2$, can be used.

For these two parameters over wide ranges, the CRB is calculated for K-level QRSS. At each K, the d_{mean} and d_{ratio} which minimize the bound for $\bar{\sigma}$ are found. This search is repeated for each K, for $K = 4 \ldots 10$. The minimum bound for $\bar{\sigma}$ is plotted as a function of K in Fig. 3.10(a). Also shown in Fig. 3.10(a) is the RSS result for the same example, which gives the asymptotic limit for QRSS. The results

show that for $K \geq 5$, the QRSS bound is within 10% of the bound for RSS. Thus, K-level QRSS rapidly approaches the limits of RSS as K increases. The values of d_{mean} and d_{ratio} which minimize the bound at each K are plotted in Fig. 3.10(b). Note that the value of d_{mean} is relatively constant (between 0.44 and 0.45 m) for $K > 3$. Thus for a particular geometry of devices, there may be a rule-of-thumb for the selection of d_{mean} (or equivalently $\bar{P}$) regardless of the value of K.

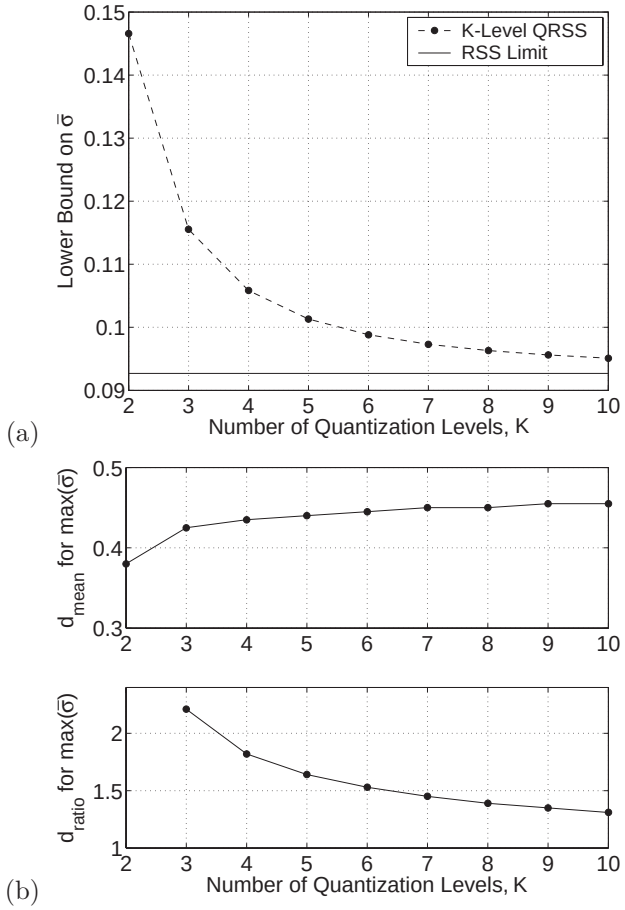


Figure 3.10: For K-level QRSS in the 5 by 5 grid example, (a) the lower bound on $\bar{\sigma}$ as a function of K, and (b) the d_{mean} and d_{ratio} which result in those minima.

3.5 Bias Sensitivity

The CRB applies to estimators which are unbiased, that is, that the mean of a parameter estimate is equal to the actual value. This section investigates how the

lower bound might change if an estimator was permitted to have a small amount of bias.

Note that a constant bias does not affect the bound - if a parameter estimator has a known constant bias (constant across the range of the parameter), the bias can simply be subtracted out. However, if an estimator has a non-zero bias gradient, *i.e.*, a changing bias as a function of the actual parameter value, the bias cannot simply be negated. While zero bias gradient is a desirable estimator characteristic, a small amount of bias gradient is often acceptable in exchange for a decrease in the estimator variance. In particular, the uniform Cramér-Rao bound (UCRB) analysis of [54] is used in this section. The bias sensitivity index η is defined as [54],

$$\eta = \left| \frac{\partial}{\partial \delta} B(\boldsymbol{\theta}, \delta) \Big|_{\delta=0} \right| \tag{3.16}$$

where δ is the limit imposed on $|\nabla_{\boldsymbol{\theta}} \mathbf{b}_{\boldsymbol{\theta}}|$, where $\nabla_{\boldsymbol{\theta}}$ is the gradient w.r.t. the parameter vector $\boldsymbol{\theta}$, $\mathbf{b}_{\boldsymbol{\theta}}$ is the bias vector as a function of $\boldsymbol{\theta}$, and $B(\boldsymbol{\theta}, \delta)$ is the UCRB, the lower bound on the variance possible from any estimator with bias gradient norm less than or equal to δ, as defined in [54]. Intuitively, η represents how quickly the variance bound can be reduced as a function of the bias gradient, if a small bias gradient is allowed.

In particular, the bias sensitivity index is calculated for the single unknown-location node example discussed at length in Section 3.4.1. There are just two parameters, x_1 and x_2. Define η_x and η_y to be the bias sensitivity indices of x_1 and x_2, respectively. By direct application of the result in [54],

$$\eta_x = 2\sqrt{1 + \frac{f_{xy}^2}{f_{yy}^2}} \tag{3.17}$$

$$\eta_y = 2\sqrt{1 + \frac{f_{xy}^2}{f_{xx}^2}}, \tag{3.18}$$

where f are the (scalar) elements of F, the Fisher information matrix for the single unknown-location node coordinates, which, since there are only two unknown coordinates, is given by

$$\mathbf{F} = \begin{bmatrix} f_{xx} & f_{xy} \\ f_{xy} & f_{yy} \end{bmatrix}, \tag{3.19}$$

where the elements of F are given in (3.6).

A value of $\eta = 2$ means that the variance bound is insensitive to δ near $\delta = 0$, while very high η indicates that even a small bias gradient can dramatically change the variance bound. Plots of η_x and η_y are shown in contour plots in Fig. 3.11 for the cases when measurements are TOA, RSS, and 3-level QRSS. The TOA and RSS results are independent of the channel parameter γ, but the 3-level QRSS result is a function of the chosen distance thresholds. In the 3-level QRSS plots, the thresholds $d_1 = 0.90$ m and $d_2 = 0.56$ m are used, to follow those chosen in Fig. 3.5 in Section 3.4.1.

When measurements are TOA, the values of η_x and η_y are small throughout the area. Even at the corners, both are less than about 2.2, only 10% higher than the minimum of 2. Even at the corners, there is only a small change in the variance bound when a small bias gradient is permitted.

When measurements are RSS, the bias sensitivity index η is very high when the sensor is on the edge of the square area and very close to one of the reference sensors, as high as 6.5 in Fig. 3.11(b-c). This is a result of the accuracy of RSS being proportional to the distance between sensors. As the unknown-location sensor gets very close to a reference sensor, its distance from that reference can be very accurately determined. In comparison, the information obtained from RSS measurements with the distant reference devices is very small. In effect, the unknown sensor can be

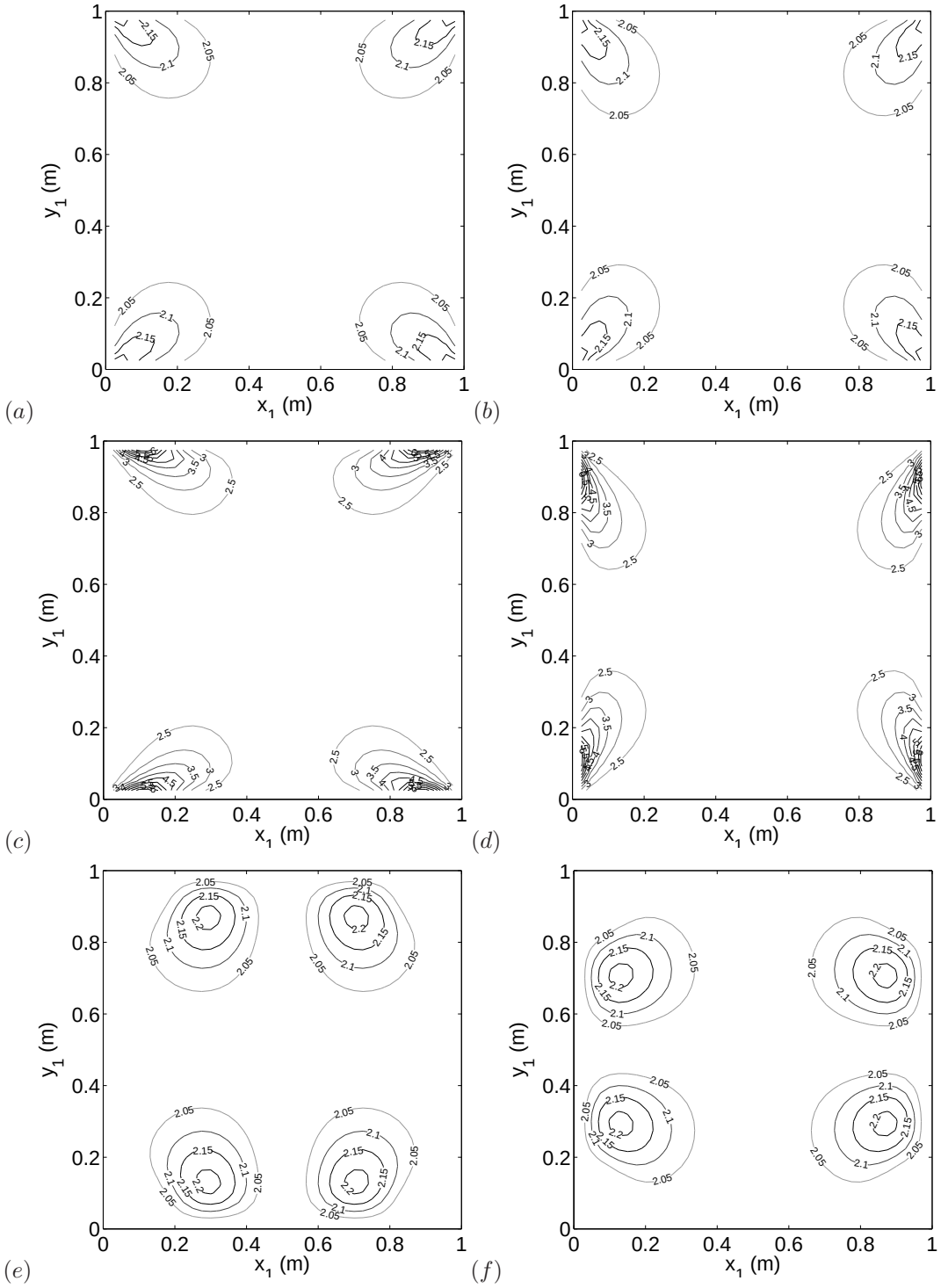


Figure 3.11: For the one-blindfolded sensor example, contour plots of the bias sensitivity index η for the estimation of (a,c,e) x_1 and (b,d,f) y_1, when measurements are (a-b) TOA, (c-d) RSS, and (e-f) 3-level QRSS with thresholds $d_1 = 0.90$ m and $d_2 = 0.56$ m.

accurately estimated to be on a circle with a particular radius around the nearby reference sensor. When the unknown sensor is close to a reference sensor and also on the $y_1 = 0$ or $y_1 = 1$ edge of the square area, essentially, there is very little y_1 information (f_{yy}), but a high quantity of x_1 information (f_{xx}). As a result, in this case, η_x (Fig. 3.11(c)) is very high. Incidentally, the bias sensitivity indices for the case of AOA measurements are the same as for RSS measurements, except with η_x and η_y exchanged. For AOA measurements, η_x is given in Fig. 3.11(d) while η_y is given in Fig. 3.11(c). This equality is due to the similar, and complementary, forms of the Fisher information for the cases of AOA and RSS, as presented in Section 3.3.

However, note that the very high bias sensitivity in the RSS case is partially due to the assumption that there is no quantization - that all RSS values, even very high RSS values, can be measured. In fact, all RSS circuits will have some degree of quantization. In particular, practical receivers have a maximum measurable RSS quantization level, essentially countering the effect described above when the unknown sensor approaches a reference sensor. This is seen in Fig. 3.11(e-f), which shows the η results for 3-level QRSS. In this case, there is very little bias, and a maximum η of about 2.2 is seen, and this maximum is located away from the corners. When RSS is quantized, the references contribute more equally to the location information of the unknown location sensor, even when that sensor is very close to one of the references.

This bias gradient and variance tradeoff for localization based on RSS measurements is explored further in Section 4.2, in which the bias gradient of the maximum likelihood estimator for the single unknown-location node example is calculated and discussed.

3.5.1 Discussion

Note that this chapter has not addressed how tight the localization CRB is, that is, for realistic signal-to-noise ratios, whether or not the lower bound can be nearly achieved. The discussion in Chapter IV introduces particular localization algorithms, simulates their performance, and shows that in many cases, the estimator performance can be very near the CRB.

CHAPTER IV

Localization Algorithms

Up until this chapter, models for measurements and the performance bounds which result from them have been presented without discussing a single sensor location estimator. This chapter introduces several such estimators, and uses the bounds presented in the previous chapter as a comparison. This chapter will also present experimental measurements from Section 2.6 to test how well these new estimators would have performed in a real sensor network.

Section 4.1 provides a brief review of the now extensive literature in sensor localization algorithms. Then, Section 4.2 presents maximum likelihood estimators (MLEs) for sensor localization when measurements are RSS and TOA as a baseline for comparison.

Then Section 4.3 introduces manifold learning. The contribution of this thesis to sensor localization algorithms is the application of multiple different manifold learning methods. It isn't necessarily intuitive that manifold learning methods are applicable to sensor localization. However, as will be described in detail in Section 4.3, these dimensionality reduction methods attempt to find low dimensional coordinates which best represent original high-dimensional coordinates in terms of some metric. Manifold learning methods, in particular, emphasize the local structure

between nearest neighbors, when finding the low-dimensional representation. Thus manifold learning is a tool which provides the desired behavior – finding 2-D (or 3-D) coordinates from higher-dimensional measured data, and emphasizing the distances between pairs of sensors which are in close proximity. As discussed in Chapter II, measurements between nearby sensors often provide more location information than measurements between distant sensors.

In particular, three new manifold learning-based approaches to sensor localization are presented in this thesis:

1. Section 4.4 introduces the distributed weighted multi-dimensional scaling (dwMDS) algorithm for use when pair-wise measurements are either TOA or RSS. This algorithm was developed in collaboration between the author and Jose A. Costa [27, 28].

2. Next, Section 4.5 introduces an adaptive Laplacian Eigenmap algorithm for sensor localization from pair-wise connectivity measurements [90].

3. Finally, Section 4.6 compares multiple manifold learning approaches when sensor data itself is used as location information. Rather than using range measurements between pairs of sensors, high-dimensional data from an isotropic, spatially correlated field is measured by each sensor, and used for localization.

Table 4.1 provides a quick reference for the manifold learning-based algorithms (and the MLE) which use pair-wise measurements for sensor localization. Manifold learning approaches to using sensor data to achieve localization are enumerated separately in Section 4.6. As a further note regarding Table 4.1, since there are general, decentralized algorithms for optimizing cost functions [99], many centralized optimizations could be accomplished by local collaboration. However, Table 4.1

labels each algorithm as centralized or decentralized based only on whether or not the decentralized implementation has been introduced and tested in this thesis.

	MLE	**dwMDS 2-Stage**	**Lapl. Eig. 2-Stage**	**Isomap**
Allows imperfect prior	No	Yes	No	No
Decentralized	No	Yes	No	No
Model-based	Yes	No	No	No
Measurement Modalities				
• Presented in thesis	TOA/RSS	TOA/RSS	Connectivity	Connectivity
• Could be used with	Any	Connectivity	TOA/RSS	TOA/RSS
Iterative Optimization	Yes	Yes	No	No
Computational Complexity	$\mathcal{O}\left(LN^2\right)$	$\mathcal{O}\left(LKN\right)$	$\mathcal{O}\left(KN^2\right)$	$\mathcal{O}\left(N^3\right)$

Table 4.1: Comparison of algorithms discussed in Chapter IV.

4.1 Overview of Localization Algorithm Research

The literature in sensor co-operative localization algorithms is extensive and growing. Signal processing, statistics, and computer science communities have published extensively in this area. This section provides an overview from a signal processing perspective, of the relevant techniques and methods which have been useful. More complete reviews and algorithm comparisons have been presented in [80, 106, 116].

While positioning and navigation have a long history, to enable cooperative localization, there is a need to extend existing methods by finding ways to use measurements (of range or angle) between pairs of unknown-location nodes. The challenge is to allow sensors which are not in range of any known-location devices to be located, and further, to improve the location estimates of all sensors.

If each sensor was in range of multiple reference nodes, each sensor's location could be calculated directly and independently. For example, in [14], nodes measure RSS received from a dense network of reference nodes. Each sensor estimates its location to be the mean of the locations of the in-range reference nodes. In most wireless sensor networks, though, to minimize installation expenses, reference nodes will be

sparse, and low-energy sensors generally will not be in range of enough references (3 or 4 for 2-D or 3-D localization, respectively).

Cooperative localization algorithms can be generally divided into centralized algorithms, which collect measurements at a central processor prior to calculation, and distributed algorithms, which require sensors to share information only with their neighbors, but possibly iteratively.

4.1.1 Centralized Algorithms

If the data is known to be described well by a particular statistical model (eg. Gaussian or log-normal), then the maximum likelihood estimator can be derived and implemented [92, 77], as will be presented in Section 4.2 for the cases of RSS and TOA measurements. One reason that these estimators are used is that their variance asymptotically (as the signal-to-noise (SNR) ratio goes high) approaches the lower bound given by the CRB (in Section 3.1). While this property is desirable, note that there may be no practical way to increase the SNR due to severe channel degradations discussed in Chapter II. As indicated by the name, the maximum of a likelihood function must be found. There are two difficulties with this approach:

1. *Local Maxima*: Unless the MLE is initialized to a value close to the correct solution, it is possible that our maximization search may not find the global maximum.

2. *Model Dependency*: If measurements deviate from the assumed model (or model parameters), the results are no longer guaranteed to be optimal.

One way to prevent local maxima is to formulate the localization as a convex optimization problem. In [33], convex constraints are presented that can be used to require a sensor's location estimate to be within a radius r and/or angle range

$[\alpha_1, \alpha_2]$ from a second sensor. In [68], linear programming using a 'taxi metric' is suggested to provide a quick means to obtain rough localization estimates. More general constraints can be considered if semi-definite programming (SDP) techniques are used [12]. One difficulty which must be overcome in both techniques is their high computational complexity. Toward this end, a distributed SDP-based localization algorithm was presented in [11]. Also, very simple constraints can be used by using the order statistics of RSS measurements - at a receiver, store only a list of neighbors, listed in decreasing order of RSS. This non-parametric method allows simple half-plane constraints and has been used in a coding-theoretic approach to sensor localization [128].

Multi-dimensional scaling (MDS) algorithms (and Isomap [118]) are formulated as sensor localization algorithms in the setting of a least-squares (LS) problem [110, 109]. These formulations are described in detail in Section 4.3.3. The major result is that in classical MDS, the LS solution is found by eigen-decomposition, which does not suffer from local maxima. In order to linearize the localization problem, the classical MDS formulation works with squared distance rather than distance itself, and the end result is very sensitive to range measurement errors.

While MDS and Isomap have complexity $\mathcal{O}(N^3)$, where $N = n + m$ is the total number of sensors, other manifold learning methods, such as local linear embedding (LLE) [104], are also based on eigen-decomposition, but of sparse matrices, and are $\mathcal{O}(KN^2)$, where K is the number of neighbors. Manifold learning will likely play an important role when using either pair-wise measurements [90, 27] or sensor data measurements [89]. Three approaches are presented in this thesis in Sections 4.4 through 4.6. Also adapted from the statistical learning area, 'supervised learning' approaches localization as a series of detection problems [112]. The covered area is

split into smaller, overlapping regions, and based on the measurements, each region detects whether or not the sensor is within its boundaries.

4.1.2 Distributed Algorithms

There are two big motivations for developing distributed localization algorithms. First, for some applications, no central processor, or none with enough computational power, is available to handle the calculations. Second, when a large network of sensors must forward all measurement data to a single central processor, there is a communication bottleneck and higher energy drain at and near the central processor.

Distributed algorithms for cooperative localization generally fall into one of two categories:

1. *Network Multilateration*: Each sensor estimates its multi-hop range to the nearest reference nodes. These ranges can be estimated via the shortest path between the sensor and reference nodes, *i.e.*, proportional to the number of hops, or the sum of measured ranges along the shortest path [81, 108, 79]. Note that finding the shortest path is readily distributed across the network. When each sensor has multiple range estimates to known positions, its coordinates are calculated locally via multilateration [121, 15].

2. *Successive Refinement*: These algorithms try to find the optimum of a global cost function, eg., least squares (LS), weighted LS (WLS) [27], or maximum likelihood (ML). Each sensor estimates its location and then transmits that assertion to its neighbors [2, 105, 107]. Neighbors must then recalculate their location and transmit again, until convergence. A device starting without any coordinates can begin with its own local coordinate system and later merge it with neighboring coordinate systems [124]. Typically, better statistical performance

is achieved by successive refinement compared to network multilateration, but convergence issues must be addressed.

Bayesian networks (or factor graphs) provide another distributed successive refinement method to estimate the probability density of sensor network parameters. These methods are particularly promising for sensor localization - each sensor stores a conditional density on its own coordinates, based on its measurements and the conditional density of its neighbors [57]. Alternatively, particle filtering (or Monte-Carlo estimation methods) methods have each sensor store a set of 'particles', *i.e.*, candidate representations of its coordinates, weighted according to their likelihood [22, 56]. These methods have been used to accurately locate and track mobile robots [120], and they will likely find application in future sensor localization and tracking research.

4.1.3 Centralized / Decentralized Comparison

Both centralized and distributed algorithms must face the high relative costs of communication. The energy required per transmitted bit could be used, depending on the hardware and the range, to execute 1,000 to 30,000 instructions [19]. Centralized algorithms in large networks require each sensor's measurements to be passed over many hops to a central processor, while distributed algorithms have sensors send messages only one hop (but possibly make multiple iterations). The energy-efficiency of centralized and distributed estimation approaches can be compared [99]; in general, when the average number of hops to the central processor exceeds the necessary number of iterations, distributed algorithms will likely save communication energy costs.

There may be hybrid algorithms which combine centralized and distributed fea-

tures in order to reduce the energy consumption beyond what either one could do alone. For example, if the sensor network is divided into small clusters, an algorithm could select a processor from within each cluster to estimate a map of the cluster's sensors. Then, cluster processors could operate a distributed algorithm to merge and optimize the local estimates, such as described in [60]. Such algorithms are a promising open topic for future research.

4.2 Maximum Likelihood Estimator

The maximum likelihood estimator (MLE) is a centralized and model-based localization algorithm which could be used to estimate location using TOA, RSS, connectivity, or AOA measurements, as long as a statistical model is available. In this section, the MLE is presented for the TOA and RSS cases, with a brief discussion of the case of connectivity measurements at the end of this section. A discussion of the MLE when measurements are AOA was presented in [77]. None of the presented MLEs have analytical solutions, and optimization of the likelihood functions has been implemented using an iterative, conjugate-gradient method.

4.2.1 TOA

Recall that in TOA measurements, the time delay between sensors i and j, $T_{i,j}$, is modelled as Gaussian, as given by (2.10). In general, the maximum likelihood estimator finds the parameters which maximizes the likelihood function, or equivalently, minimizes the negative of the log-likelihood function. Thus, when measurements are TOA, the MLE is

$$\{\hat{\mathbf{z}}_i\} = \arg\min_{\{\mathbf{z}_i\}} \sum_{i=1}^{N} \sum_{\substack{j \in H(i) \\ j<i}} \left[v_p(T_{i,j} - \mu_T) - \|\mathbf{z}_i - \mathbf{z}_j\| \right]^2 . \tag{4.1}$$

Recall that $\mathbf{z}_i$ is the coordinate of sensor i, $H(i)$ is the set of sensors with which sensor i made measurements, v_p is the speed of propagation, and μ_T is the mean of TOA error.

4.2.2 RSS

The MLE for the RSS case is [93],

$$\{\hat{\mathbf{z}}_i\} = \arg\min_{\{\mathbf{z}_i\}} \sum_{i=1}^{N} \sum_{\substack{j \in H(i) \\ j<i}} \left(\ln \frac{(\delta_{i,j}^{MLE})^2}{\|\mathbf{z}_i - \mathbf{z}_j\|^2} \right)^2 \tag{4.2}$$

where $\delta_{i,j}^{MLE}$ is a function of the measured received power $P_{i,j}$ as given in (2.17) (specifically, the MLE of distance given $P_{i,j}$). Unlike the MLE based on TOA measurements, the RSS MLE is readily shown to be biased. Specifically, for a single reference and single unknown-location device, the range estimate between the two devices is $\delta_{1,2}$. Using (2.2), the mean of $\delta_{1,2}$ is given by

$$E[\delta_{1,2}] = C \left\| \mathbf{z}_1 - \mathbf{z}_2 \right\| \tag{4.3}$$

where C is the multiplicative bias factor given in (2.7). For typical channels (like those reported in [101]), $C \approx 1.2$, adding 20% bias to the range. Motivated by (4.3), a pseudo-MLE can be defined,

$$\{\hat{\mathbf{z}}_i\} = \arg\min_{\{\mathbf{z}_i\}} \sum_{i=1}^{N} \sum_{\substack{j \in H(i) \\ j<i}} \left(\ln \frac{(\delta_{i,j}^{BC})^2}{\|\mathbf{z}_i - \mathbf{z}_j\|^2} \right)^2, \tag{4.4}$$

where $\delta_{i,j}^{BC}$ is given in (2.8) and is the MLE of distance divided by the bias factor C.

Note that the channel parameters n_p in (4.4) and μ_T in (4.1) can also be estimated as nuisance parameters. In this case, they are also optimized during the cost minimization.

Even in the bias-reduced estimator in (4.4), there remains residual bias. Consider $m = 4$ and $n = 1$. Place the reference devices at the corners of a 1 m by 1 m square and the unknown-location device within the square, the case drawn in Fig. 3.3. The bias gradient norm of $\hat{\mathbf{z}}_1$ is calculated via simulation [54] from (4.4) as a function of the unknown-location device's coordinates and displayed in Fig. 4.1.

The gradient of the bias could be used in the uniform CRB to calculate the achievable variance of the biased estimator [54] as compared to all other estimators with same or less bias gradient norm. Fig. 4.1 shows that the bias gradient is high (with norm ≈ 1) at the corners of the square. Expression (4.4) shows that the MLE

tries to force the ratio $\delta_{1,j}^{BC}/\|\mathbf{z}_1 - \mathbf{z}_j\|$ close to 1. If the unknown-location device is very close to one reference device and far away from the others, then measurements from the other three reference devices provide relatively little information regarding the placement of the unknown-location device. In the limit as the unknown-location device approaches a reference device, it can only be localized to a circle around that reference. Thus no unbiased estimator is possible. The MLE in (4.4) approaches a constant in the limit, and thus the bias gradient norm approaches 1.

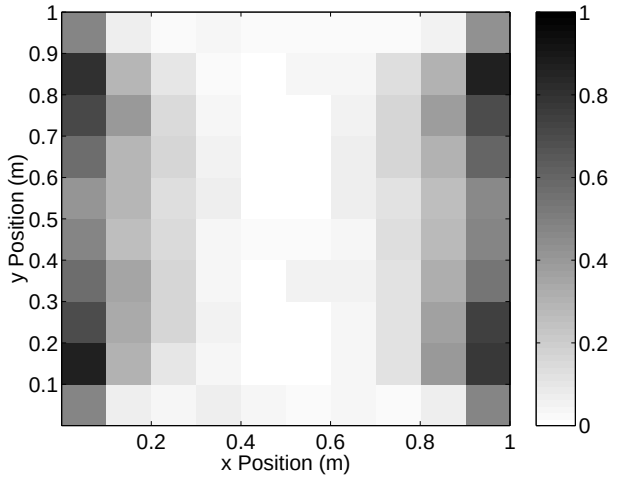


Figure 4.1: Bias gradient norm of the RSS MLE of x_1 from (4.4) for the single unknown-location sensor example (see Fig. 3.3).

4.2.3 Connectivity

When measurements are connectivity, the likelihood function is very difficult to optimize. When 'connected' sensors are closer to each other than the connection radius R, there is little motivation to move them closer; alternatively, when 'non-connected sensors are further than the connection radius R, there is little motivation to move them further apart. The cost function suffers from many local optima separated by high 'barriers' in which a gradient-based iterative optimization procedure will be likely to become stuck. Since gradient-based iterative algorithms have great

difficulty converging to the global optimum, the MLE has not been implemented for the case when measurements are connectivity. Instead, this thesis has developed a different estimator, which obtains a global optimum directly via eigen-decomposition, as presented in Section 4.5.

4.2.4 Simulation Results

The MLE is tested for the case of RSS measurements via simulation on a 5×5 grid of sensors, of which the four in the corners are reference sensors, as shown in Fig. 3.7 with $N = 25$ and $L = 1$m. As described in Section 3.3.4, the results can be seen in units of L, since they would scale proportionally to L. A channel with $\sigma_{dB}/n_p = 1.7$ is used to match the measured channel parameters from Measurement Experiment II (Section 2.6.2), and 200 Monte Carlo simulations are run. In these simulations, all sensors are assumed to be in range of all other sensors, and thus make RSS measurements with every other sensor.

For comparing different estimators, let the mean bias $\bar{b}$ and the RMS standard deviation $\bar{\sigma}$ of the estimator be defined as:

$$\bar{b} = \frac{1}{n}\sum_{i=1}^{n} \|\bar{\mathbf{z}}_i - \mathbf{z}_i\| \tag{4.5}$$

$$\bar{\sigma} = \sqrt{\frac{1}{n}\mathrm{tr}\,\mathbf{C}} \tag{4.6}$$

where n is the number of unknown-location devices, $\bar{\mathbf{z}}_i$ is the mean of all of the estimates of sensor i over all trials of the simulation, $\mathbf{z}_i$ is the actual location of sensor i, and $\mathbf{C}$ is the covariance of the coordinate estimates over all trials of the simulation.

First, simulation results of the pseudo-MLE (4.4) are shown in Fig. 4.2(a). For this estimator, $\bar{b} = 0.033$ and $\bar{\sigma} = 0.0957$. Comparing the estimator and the CRB, the lower bound on $\bar{\sigma}$ is 0.0925. Next, the MLE from (4.2) is tested, and the results

are shown in Fig. 4.2(b). For this estimator, $\bar{b} = 0.006$ and $\bar{\sigma} = 0.1029$. The bias has been reduced by a factor of 5, and the standard deviation has increased by 7.5%.

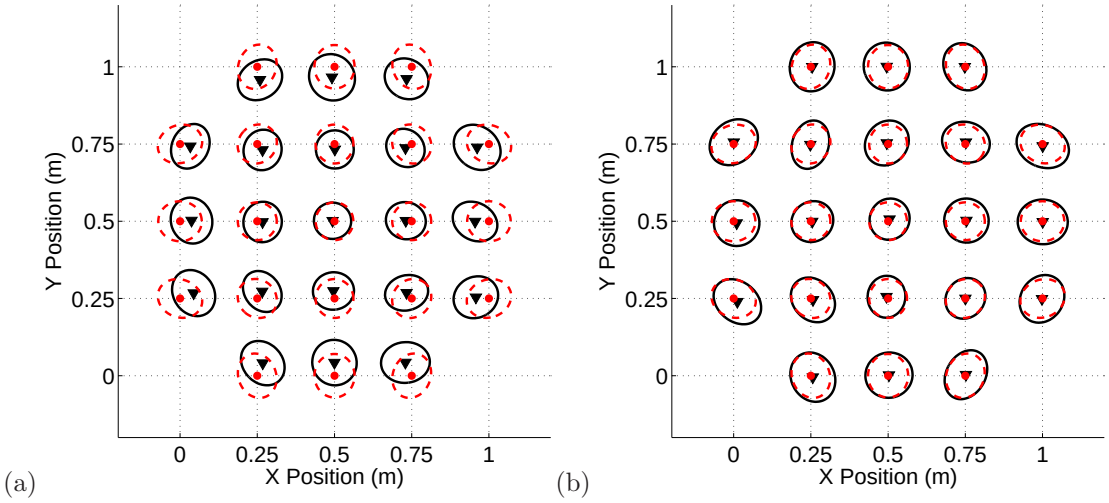


Figure 4.2: RSS pseudo-MLE and MLE performance: mean (▼) and 1-σ uncertainty ellipse (—-) for each unknown-location sensor compared to the true location (●) and CRB on the 1-σ uncertainty ellipse (- - - -), when reference sensors are located at each ×.

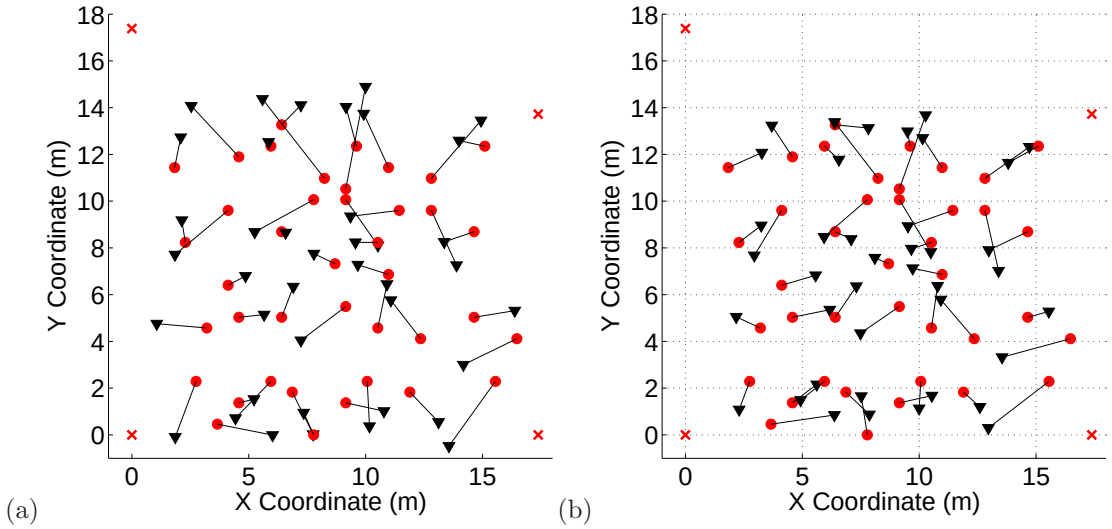
The fact that the estimator is so close to the bound even at realistic values of the signal-to-noise ratio is promising, as it means that the lower bound presented in Chapter III is a reasonably tight lower bound.

4.2.5 Experimental Results

In this section, the RSS MLE and pseudo-MLE and TOA MLE are tested on the recorded RSS and TOA data sets from Measurement Experiments I and II. For all of these experimental results, since measurements were made between each pair of devices, the full connectivity assumption is employed.

First, the measured RSS data set from Measurement Experiment I is used in the RSS MLE and pseudo-MLE. The location estimates are shown in Fig. 4.3. As a summary, RMS location error of the estimates is 2.16m for the MLE and 1.87m for the pseudo-MLE.

Next, the RSS and TOA data sets recorded in Measurement Experiment II are



(a) (b)

Figure 4.3: Estimated coordinates (▼) connected to actual coordinates (●) for the 40 unknown-location sensors from Measurement Experiment I using (a) MLE and (b) pseudo-MLE. Reference devices are located at each ×.

used to test the MLEs. The results are shown in Fig. 4.4. For the TOA measurements, the RMS error is 1.15 m, while for the RSS measurements, the RSS MLE has RMS error 2.20 m. The estimates from the pseudo-MLE using RSS measurements have virtually identical results, with RMS error 2.21 m, and are not plotted separately.

Note that in all cases, the localization errors of the sensors in the center of the measurement area are generally lower than those at the edge. This indicates that, for a particular sensor, there may be a significant advantage to having sensors on all sides of itself.

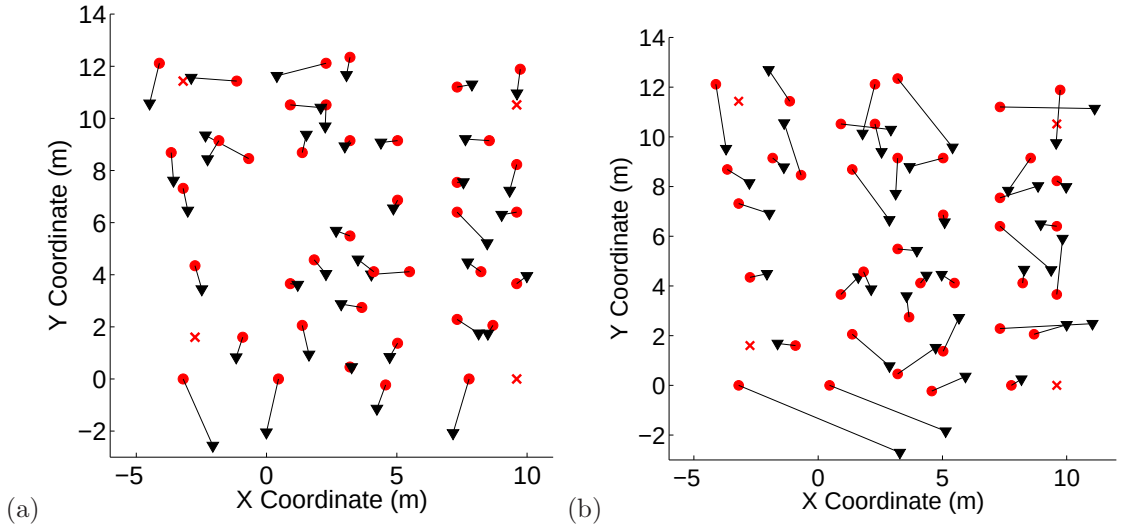


Figure 4.4: Estimated coordinates (▼) connected to actual coordinates (●) for the 40 unknown-location sensors from Measurement Experiment II using (a) TOA measurements and the TOA MLE, (b) RSS measurements and the RSS MLE. Reference devices are located at each ×.

4.3 Introduction to Manifold Learning Methods

The MLE depends on a parametric model and is consequently subject to model mismatch and slow convergence. In contrast, algorithms presented in this chapter are based on non-parametric models and manifold learning algorithms. Manifold learning refers to a variety of non-linear data dimension reduction methods which can extract lower-dimensional non-linear subspaces from very high-dimensional data. For example, consider Fig. 4.5. Three-dimensional data is shown to lie approximately within a curved, 2-D surface. Principal components analysis (PCA) would calculate a 2-D plane which, in a least-squares sense, would best represent the data points. But since the data clearly does not lie in a plane, the PCA output would have significant errors. In comparison, manifold learning algorithms are formulated assuming only that locally (for a point and its nearest neighbors) that data lie in an approximately linear subspace. Globally, the subspace can be curved. Manifold learning methods output 2-D coordinates for each point which preserve the relative distances between

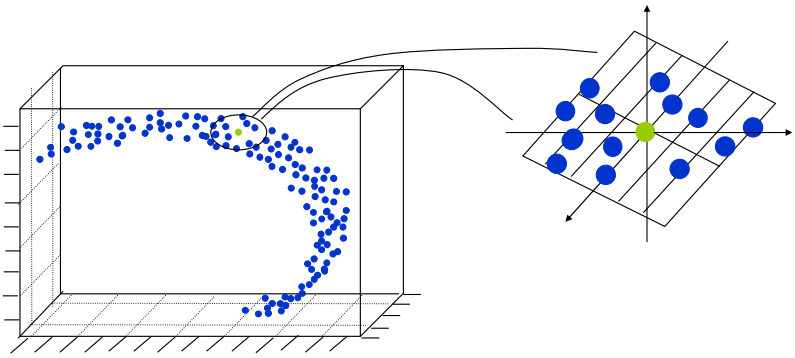


Figure 4.5: Manifold learning methods *do not* assume that data lies in a globally linear subspace; however, a data point and its nearest neighbors lie approximately within a *locally linear* subspace.

points and their nearest neighbors.

Generally, manifold learning methods comprise these six steps:

1. **Input Points**: Collect N high-dimensional data points.
2. **Compute Distances**: Compute distances between all pairs of points.
3. **Select Neighbors**: For each point, find which few other points are closest.
4. **Build Graph**: Calculate a graph which encodes the neighbor relationships.
5. **Find Low-D Coordinates**: Find the low-dimensional coordinates which best preserve the neighbor relationship graph (in some sense).
6. **Low-D Transform**: Possibly rotate, scale, or translate the low-D coordinates to match some *a priori* information, or for ease of viewing.

These steps in particular are discussed in the next few sections, and for the remainder of this thesis, since each localization problem considered uses these steps.

Further, a contribution of this chapter is to consider iterative neighbor selection, in which, after completing Step 5 (Find Low-D Coordinates) the algorithm returns to Step 3 (Select Neighbors) to adjust each point's neighbor relationship, and re-

completes Steps 4 through 6. These iterative algorithms are presented in Sections 4.4 and 4.5.

Finally, note that there is a duality between data localization and physical sensor localization. In data localization, low-dimensional coordinates are found for measured high-dimensional data vectors. The resulting low-D coordinates are non-physical in the sense the data do not have mass at the coordinates calculated for them. In physical sensor localization, physically meaningful coordinates of sensors are estimated for sensors which do exist at those coordinates. However, when sensors measure data that has spatial relationships, there can be significant overlap, even duality, between the two localization problems.

For example, consider the image data localization problem presented by Tenenbaum et. al. in [118], shown in Fig. 4.6, which computed 2-D coordinates for face images which were taken from different azimuth and elevation angles. While Tenenbaum framed the problem as data localization, it could equivalently be considered to be sensor localization. Rather than estimating coordinates for images (data), we want to know the coordinates of the cameras (sensors) which took the pictures. Certainly, there is a duality in this problem between data and physical sensor localization.

One major difference between the two types of localization is that in physical sensor localization, the first step (input points) is often skipped. In the algorithms introduced in Sections 4.4 and 4.5, the distances between sensors are directly measured via the radio channel.

The next section discusses methods of neighbor selection (Step 3). Then, Sections 4.3.2 through 4.3.4 discuss how Steps 4 and 5 differ in various manifold learning algorithms.

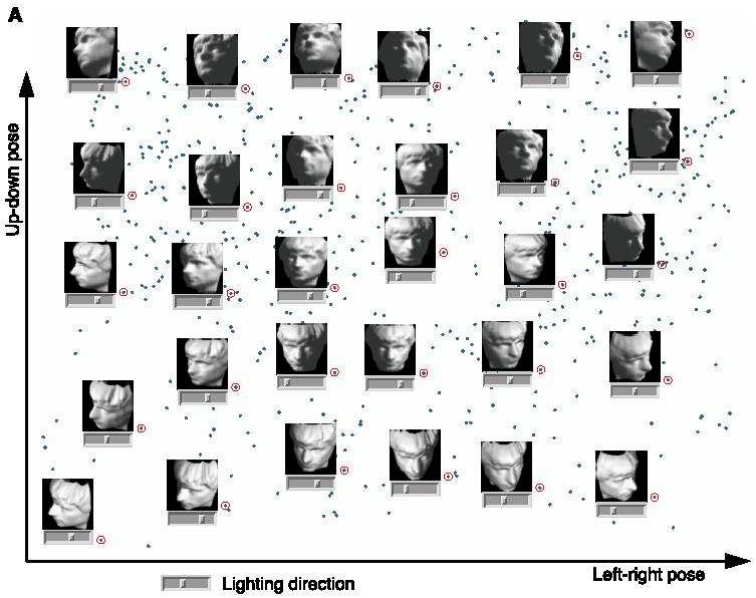


Figure 4.6: Tenenbaum et. al. in [118] present this example of localizing face images in a database, but it equivalently serves to localize the cameras which took each photo.

4.3.1 Neighbor Selection Methods

Neighbor selection uses the pair-wise distances $\{\delta_{i,j}\}$ to determine which sensors each sensor considers to be its neighbors. We denote $\mathcal{N}_i$ to be the set of neighbors of sensor i.

In the *R-radius* method, all pairs (i, j) with $\delta_{i,j} < R$ for some constant radius R, are considered to be neighbors. Since $\delta_{i,j} = \delta_{j,i}$, the R-radius neighbor relation is symmetric.

However, if the radius R is set too low, the neighborhood graph may end up sparse or disconnected, while setting it too high would result in large neighborhoods which may violate the local linearity assumption. Thus variants of the K-nearest-neighbors (KNN) method, which are non-parametric neighbor selection methods, are presented. These methods are:

1. *K-nearest-neighbors* (KNN): Sensor j is a neighbor of sensor i, *i.e.*, $j \in \mathcal{N}_i$, if and only if $\delta_{i,j}$ is one of the K smallest of the set $\{\delta_{i,k}\}_{k \neq i}$. Clearly, $|\mathcal{N}_i| = K$.

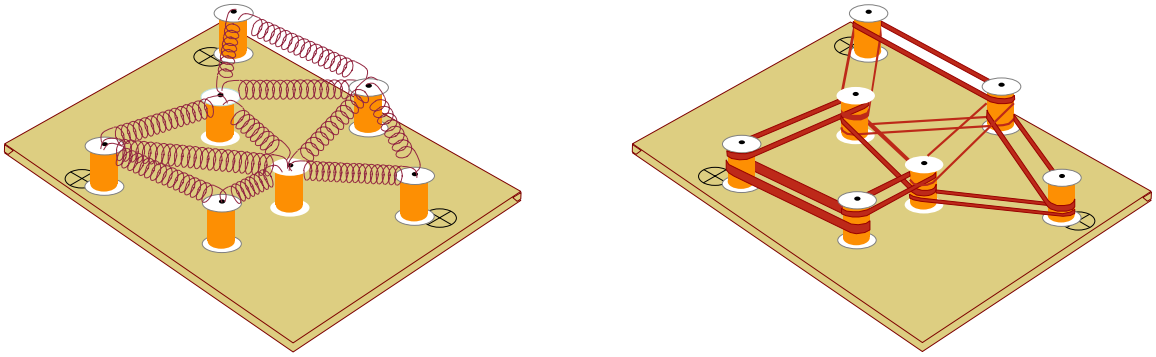
Note KNN is not a symmetric relation.

2. *Symmetric K-nearest-neighbors* (SKNN): To achieve a symmetric neighbor relation, $j \in \mathcal{N}_i$, if and only if $\delta_{i,j}$ is one of the K smallest of *either* set $\{\delta_{i,k}\}_{k \neq i}$ or one of the K smallest of set $\{\delta_{k,j}\}_{k \neq j}$. In other words, the KNN relation is OR'ed together with its transpose. For SKNN, $|\mathcal{N}_i| \geq K$.

3. *'Take Pity' K-nearest-neighbors* (TPKNN): Initialize $\mathcal{N}_i$ using KNN for all sensors i. If sensor k has less than L_{min} sensors which consider it as a neighbor, *i.e.*$|\{i : k \in \mathcal{N}_i\}| < L_{min}$, then it is considered a 'lonely' sensor. For each lonely sensor k, its L_{min} closest neighbors are forced to 'take pity' on sensor k and include k in their neighbor list. The TPKNN relation is not symmetric, but it is used to avoid particular sparsity issues.

4.3.2 Typology of Manifold Learning Algorithms

Once neighbors are selected, an algorithm (*eg.* Isomap or Laplacian Eigenmap) encodes the neighbor and distance information into a graph, and then calculates low-dimensional coordinates which best represents the graph (Steps 4 and 5 from the list at the start of Section 4.3). Manifold learning algorithms are generally one of two types: *distance-based* and *similarity-based* methods. These two types are contrasted by analogy in Fig. 4.7. As the name would indicate, the distance-based methods encode information regarding the distances between points in the graph. The similarity-based methods encode inverse distance, or some decreasing function of distance. The operation of Isomap, a distance-based algorithm, is presented in Section 4.3.3, and the operation of Laplacian Eigenmap, a similarity-based algorithm, is presented in Section 4.3.4.



(a) (b)

Figure 4.7: Physical analogy of manifold learning algorithms based on (a) distances and (b) similarity between sensors. Sensors (spools) are connected by (a) springs which have natural length equal to the measured distance, or by (b) rubber bands with different weights (thickness of the band). In distance-based methods, sensors can push *and* pull their neighbors, while in similarity-based methods, sensors can only pull. After the spools are released and the system comes to rest (optimizes), the group is scaled and rotated to match the *a priori* known coordinates $\otimes$.

4.3.3 Isomap

Classical MDS finds the coordinates $\{\mathbf{z}_i\}$ which minimize the following cost function:

$$S_{MDS} = \sum_{i,j} \left(\delta_{i,j}^2 - \|\mathbf{z}_i - \mathbf{z}_j\|^2\right)^2 \tag{4.7}$$

where $\delta_{i,j}$ is a measured distance between sensors i and j. Because the distances are squared before taking the difference, the cost is a quadratic function of the coordinates. In particular, the minimum of S_{MDS} can be found directly from the singular value decomposition of the appropriate transform of the $N \times N$ distance matrix $D = [[\delta_{i,j}]]_{i,j}$, as derived in detail in [30]. This eigen-decomposition operation has computational complexity $\mathcal{O}(N^3)$.

In Isomap [118], the distances $\delta_{i,j}$ measured between non-neighbors are ignored. Essentially, long-distance measurements are assumed to be completely unreliable. Then, a neighbor graph is built which connects all pairs (i, j) of neighboring sensors

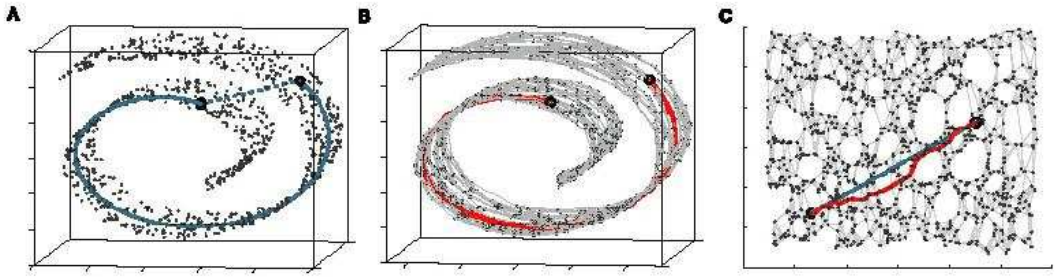


Figure 4.8: Example taken from [118] which demonstrates the Isomap algorithm on a 'Swiss Roll'.

with an edge of length $\delta_{i,j}$. Finally, $\delta_{k,l}^{SP}$ is set equal to the sum length along the shortest path on the neighbor graph between sensors k and l, for all pairs $(k,l) \in \{1,\ldots,N\}^2$. Isomap proceeds just as classical MDS, replacing in (4.7) the direct distances $\{\delta_{i,j}\}$ with the shortest path distances $\{\delta_{i,j}^{SP}\}$. The general idea is that the shortest path on the neighborhood distance graph is a good approximation to the shortest distance on the manifold, as demonstrated in Fig. 4.8 [118]. In terms of calculation, the Isomap algorithm adds to the MDS complexity of $\mathcal{O}(N^3)$, an additional requirement of finding of shortest path. Using Dijkstra's algorithm, this is an $\mathcal{O}(N^2)$ operation, and this calculation is readily decentralized in a wireless network.

An algorithm called MDS-Map, introduced by Shang et. al. [110], applies the Isomap algorithm when measurements are connectivity to sensor localization. Simulation results will be shown as a comparison to the Laplacian Eigenmap-based algorithm developed in Section 4.5. The distributed weighted MDS (dwMDS) algorithm, introduced in Section 4.4, is also a distance-based manifold learning method.

4.3.4 Laplacian Eigenmap

The Laplacian Eigenmap method is a similarity-based manifold learning method that considers the minimization of the cost S_{LE} [8]:

$$S_{LE} = \sum_{i,j} w_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|^2 \tag{4.8}$$

subject to the translation and scaling constraints,

$$\sum_i \mathbf{z}_i = \mathbf{0} \quad \text{and} \quad \sum_i \|\mathbf{z}_i\|^2 = 1. \tag{4.9}$$

The weights (or similarities) $w_{i,j}$ have an inverse relationship with distance - for more distant neighbors, the weights are set smaller. For non-neighbors, $w_{i,j}$ is set to zero as an approximation, since the distance is large for non-neighbors. Also, the Laplacian Eigenmaps method requires symmetric weights, *i.e.*, $w_{i,j} = w_{j,i}$.

The minimum of cost S_{LE} without any constraints would occur when all the coordinates $\mathbf{z}_i$ were equal. The constraints in (4.9) remove the translation ambiguity by setting the origin as the center, and counteract the tendency to put all points at the origin by mandating a unit norm average coordinate.

The benefit of the formulation in (4.8) and (4.9) is that the globally optimum solution can be found via eigen-decomposition. Defining the $N \times N$ weight matrix $W = [[w_{i,j}]]_{i,j}$ and its column sums (or row sums, since W is symmetric) $u_i = \sum_{j=1}^{N} w_{i,j}$, the graph Laplacian L is given by,

$$L = \text{diag}[u_1, \ldots, u_N] - W, \tag{4.10}$$

where $\text{diag}[u_1, \ldots, u_N]$ is the diagonal matrix with $\{u_i\}$ on its diagonal. Matrix L is sparse, since $w_{i,j} = 0$ for non-neighbors, and each row or column has at most $K + 1$ non-zero elements. The eigen-decomposition of L is the set of $(\lambda_k, \mathbf{v}_k)$, for eigenvalues λ_k and eigenvectors $\mathbf{v}_k$, $k = 1 \ldots N$. Here, it is assumed w.l.o.g. that the

eigenvectors are sorted in increasing order by magnitude of eigenvalue. As presented in detail by Belkin and Niyogi in [8], the $\mathbf{v}_k$ for $i = 2 \dots d+1$ provide the optimal lowest-cost, d-dimensional solution to (4.8). Specifically,

$$\hat{\mathbf{z}}_i = [\mathbf{v}_2(i), \dots, \mathbf{v}_{d+1}(i)], \tag{4.11}$$

where $\mathbf{v}_k(i)$ is the ith element of the kth eigenvector.

Finding the smallest eigenvalues and eigenvectors of a sparse and symmetric matrix is a computational problem which has been studied for decades for problems in physics and chemistry [32, 9], and can be solved using distributed algorithms for parallel processing. In particular, if sensors select local cluster-heads, the distributed algorithm can use data-distribution techniques and block-Jacobi preconditioning methods to reduce communication. Due to the sparsity of the graph Laplacian matrix, the computational complexity of the eigen-decomposition is $\mathcal{O}(KN^2)$, where K is the number of neighbors of each sensor.

The locally linear embedding (LLE) and the Hessian-based LLE (HLLE) methods are also similarity-based manifold learning algorithms. The HLLE method [34] expands the optimization to attempt to preserve the local Hessian, *i.e.*, 2nd-order differences within local neighborhoods, within the final low-dimensional coordinate embedding.

4.4 Distributed Weighted Multi-dimensional Scaling

In this section, joint work of the author and Jose A. Costa is presented. In this work, a distributed weighted multi-dimensional scaling algorithm (dwMDS) is designed for sensor localization [27, 28]. The key features of dwMDS are:

1. The algorithm allows for a distributed implementation across the network with minimal communication requirements,

2. The algorithm accounts for prior information, which allows use of reference nodes with either perfect or imperfect prior coordinate estimates,

3. The algorithm uses a weighted cost function that allows pair-wise measurements that are believed to be more accurate to be weighted more heavily, and

4. The algorithm is based on iterative function optimization such that each iteration is guaranteed to improve the value of the cost function.

4.4.1 Distributed Weighted MDS Cost

The algorithm is inspired by classical multi-dimensional scaling (MDS), which was presented in Section 4.3.3. The proposed method minimizes the following variation on the classical MDS cost function,

$$S = 2 \sum_{1 \le i \le n} \sum_{i < j \le N} w_{i,j} \left(\delta_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\| \right)^2 + \sum_{1 \le i \le n} r_i \|\mathbf{z}_i - \overline{\mathbf{z}}_i\|^2 \tag{4.12}$$

where coordinates $\{\mathbf{z}_i\}$ are found which minimize S, given: pair-wise measurements $\delta_{i,j}$; an arbitrary weight $w_{i,j}$ assigned to quantify the predicted accuracy of measurement $\delta_{i,j}$; imperfect prior information that node i for $i = 1 \dots n$ are located at coordinate $\overline{\mathbf{z}}_i$ with accuracy r_i; and perfect prior information that node i for $i = n+1 \dots N$ are located exactly at $\overline{\mathbf{z}}_i$. If no measurement $\delta_{i,j}$ is available between i and j, or its accuracy is zero, then $w_{i,j} = 0$. Similarly, if no prior information exists for the coordinates of node i, then $r_i = 0$. It is assumed that $w_{i,j} \ge 0$, $w_{i,i} = 0$ and $w_{i,j} = w_{j,i}$, i.e., the weights are symmetric. In summary, the first n sensors have either no prior information or some, but imperfect prior information (unknown-location sensors); sensors $n + 1 \dots N$ have perfect prior coordinate information (reference sensors).

In comparison to classical MDS and Isomap, (4.12) differs in that:

- A penalty term accounts for prior knowledge about node locations,

- Rather than pure least squares, weighted least squares is possible, and
- The cost is the squared difference between distances, rather than the squared difference between *squared* distances.

Note that, under a Bayesian perspective, (4.12) can be interpreted as the posterior density of the nodes locations given the observed dissimilarities, $f(\{\mathbf{z}_i\}|\{\delta_{i,j}\})$, if the dissimilarities $\{\delta_{i,j}\}$ are assumed to be i.i.d. Gaussian with mean $\|\mathbf{z}_i - \mathbf{z}_j\|$ and variance $(2w_{i,j})^{-1}$ and the points $\{\mathbf{z}_i\}$ have a Gaussian prior with mean $\overline{\mathbf{z}}_i$ and variance $(2r_i)^{-1}$.

After simple manipulations, S can be rewritten as follows:

$$S = \sum_{i=1}^{n} S_i + c, \tag{4.13}$$

where local cost functions S_i are defined for each unknown-location node (*i.e.*, $1 \leq i \leq n$),

$$S_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{i,j}\left(\delta_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\|\right)^2 + \sum_{j=n+1}^{N} 2w_{i,j}\left(\delta_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\|\right)^2 + r_i\|\mathbf{z}_i - \overline{\mathbf{z}}_i\|^2, \tag{4.14}$$

and c is a constant independent of the nodes locations $\{\mathbf{z}_i\}$. As S_i only depends on the measurements available at node i and the positions of neighboring nodes, (i.e., nodes for which $w_{i,j} > 0$), it can be viewed as the local cost function at node i. Note that if $m = 0$ (i.e., no reference nodes are available) and $r_i = 0$, for all i (i.e., no prior information on the nodes locations), then $\partial S/\partial \mathbf{z}_i = 2\,\partial S_i/\partial \mathbf{z}_i$. This implies that the influence of $\mathbf{z}_i$ on the local cost S_i determines its influence on the global cost S. Motivated by this cost structure, an iterative scheme is proposed in which each sensor updates its position estimate by minimizing the corresponding local cost function S_i, after observing dissimilarities and receiving position estimates from its neighboring nodes.

4.4.2 dwMDS Algorithm

Unlike classical MDS, no closed form expression exists for the minimum of the cost function S or S_i. By assuming that each node has received position estimates from neighboring nodes, the algorithm minimizes $S_i = S_i(\mathbf{z}_i)$ iteratively using quadratic majorizing functions as in SMACOF (Scaling by MAjorizing a COmplicated Function [49]). This method has the attractive property of generating a sequence of non-increasing STRESS values.

A majorizing function $T_i(\mathbf{z}, \mathbf{y})$ of $S_i(\mathbf{z})$ is a function $T_i : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$ that satisfies: (i) $S_i(\mathbf{z}) \leq T_i(\mathbf{z}, \mathbf{y})$ for all $\mathbf{y}$, and (ii) $S_i(\mathbf{z}) = T_i(\mathbf{z}, \mathbf{z})$. This function can then be used to implement an iterative minimization scheme. Starting at an initial condition $\mathbf{z}_0$, the function $T_i(\mathbf{z}, \mathbf{z}_0)$ is minimized as a function of $\mathbf{z}$. The newly found minimum, $\mathbf{z}_1$, can then be used to define a new majorizing function $T_i(\mathbf{z}, \mathbf{z}_1)$. This process is then repeated until convergence (see [49] for details). The trick is to use a simple majorizing function that can be minimized analytically, e.g., a quadratic function. Following [49], S_i is rewritten as:

$$S_i(\mathbf{z}_i) = \eta_\delta^2 + \eta^2(\mathrm{Z}) - 2\,\rho(\mathrm{Z})\ ,$$

where $\mathrm{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$, and

$$\eta_\delta^2 = \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{i,j}\delta_{i,j}^2 + \sum_{j=n+1}^{N} 2w_{i,j}\delta_{i,j}^2, \tag{4.15}$$

$$\eta^2(\mathrm{Z}) = \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{i,j}\|\mathbf{z}_i - \mathbf{z}_j\|^2 + \sum_{j=n+1}^{N} 2w_{i,j}\|\mathbf{z}_i - \mathbf{z}_j\|^2 + r_i\|\mathbf{z}_i - \overline{\mathbf{z}}_i\|^2, \tag{4.16}$$

$$\rho(\mathrm{Z}) = \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{i,j}\delta_{i,j}\|\mathbf{z}_i - \mathbf{z}_j\| + \sum_{j=n+1}^{N} 2w_{i,j}\delta_{i,j}\|\mathbf{z}_i - \mathbf{z}_j\|. \tag{4.17}$$

Term (4.15) does not depend on $\mathbf{z}_i$ and term (4.16) is quadratic in $\mathbf{z}_i$. Only term (4.17) depends on $\mathbf{z}_i$ through a more complicated (sum of square roots) function.

Define $T_i(\mathbf{z}, \mathbf{y})$ as:

$$T_i(\mathbf{z}_i, \mathbf{y}_i) = \eta_\delta^2 + \eta^2(\mathrm{Z}) - 2\,\rho(\mathrm{Z}, \mathrm{Y})\ , \tag{4.18}$$

where

$$\rho(\mathrm{Z}, \mathrm{Y}) = \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{i,j} \frac{\delta_{i,j}}{\|\mathbf{y}_i - \mathbf{y}_j\|}(\mathbf{z}_i - \mathbf{z}_j)^T(\mathbf{y}_i - \mathbf{y}_j) + \sum_{j=n+1}^{N} 2w_{i,j} \frac{\delta_{i,j}}{\|\mathbf{y}_i - \mathbf{y}_j\|}(\mathbf{z}_i - \mathbf{z}_j)^T(\mathbf{y}_i - \mathbf{y}_j). \tag{4.19}$$

Using the fact that, by the Cauchy-Schwarz inequality,

$$\|\mathbf{z}_i - \mathbf{z}_j\| = \frac{\|\mathbf{z}_i - \mathbf{z}_j\|\,\|\mathbf{y}_i - \mathbf{y}_j\|}{\|\mathbf{y}_i - \mathbf{y}_j\|} \geq \frac{(\mathbf{z}_i - \mathbf{z}_j)^T(\mathbf{y}_i - \mathbf{y}_j)}{\|\mathbf{y}_i - \mathbf{y}_j\|},$$

it is easily seen that T_i majorizes S_i. Minimizing S_i through a majorizing algorithm is now a simple task of finding the minimum of T_i:

$$\frac{\partial T_i(\mathbf{z}_i, \mathbf{y}_i)}{\partial \mathbf{z}_i} = 0\ . \tag{4.20}$$

An expression for this gradient is given in [27]. If $\mathrm{Z}^{(k)}$ is the matrix whose columns contain the position estimates, $\mathbf{z}_i^{(k)}$, for all points $i = 1 \ldots n$ at iteration k, one can derive an update for the position estimate of node i using equation (4.20):

$$\mathbf{z}_i^{(k+1)} = a_i \left(r_i\, \overline{\mathbf{z}}_i + \mathrm{Z}^{(k)} \boldsymbol{b}_i^{(k)} \right)\ , \tag{4.21}$$

where

$$a_i^{-1} = \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{i,j} + \sum_{j=n+1}^{n+m} 2w_{i,j} + r_i, \tag{4.22}$$

and $\boldsymbol{b}_i^{(k)} = [b_1, \ldots, b_{n+m}]^T$ is a vector whose entries are given by:

$$\begin{array}{ll} b_j = w_{i,j}\left[1 - \delta_{i,j}/\|\mathbf{z}_i^{(k)} - \mathbf{z}_j^{(k)}\|)\right] & j \leq n,\ j \neq i \\ b_i = \sum_{\substack{j=1 \\ j\neq i}}^{n} w_{i,j}\delta_{i,j}/\|\mathbf{z}_i^{(k)} - \mathbf{z}_j^{(k)}\| + \sum_{j=n+1}^{N} 2w_{i,j}\delta_{i,j}/\|\mathbf{z}_i^{(k)} - \mathbf{z}_j^{(k)}\| & \\ b_j = 2w_{i,j}\left[1 - \delta_{i,j}/\|\mathbf{z}_i^{(k)} - \mathbf{z}_j^{(k)}\|\right] & j > n \end{array}\ . \tag{4.23}$$

Inputs: $\{\delta_{i,j}\}$, $\{w_{ij}\}$, m, $\{r_i\}$, $\{\overline{\mathbf{z}}_i\}$, ϵ, initial condition $\mathrm{Z}^{(0)}$
Initialize: $k = 0$, $S^{(0)}$, compute a_i from equation (4.22)
repeat
  $k \leftarrow k + 1$
  for $i = 1$ **to** n
    compute $\boldsymbol{b}_i^{(k-1)}$ from equation (4.23)
    $\mathbf{z}_i^{(k)} = a_i \left(r_i \overline{\mathbf{z}}_i + \mathrm{Z}^{(k-1)} \boldsymbol{b}_i^{(k-1)} \right)$
    compute $S_i^{(k)}$
    $S^{(k)} \leftarrow S^{(k)} - S_i^{(k-1)} + S_i^{(k)}$
    communicate $\mathbf{z}_i^{(k)}$ to neighbors of node i (i.e., nodes for which $w_{i,j} > 0$)
    communicate $S^{(k)}$ to node $i + 1 \,(\mathrm{mod}\, n)$
  end for
until $S^{(k-1)} - S^{(k)} < \epsilon$

Figure 4.9: Algorithm for distributed weighted multi-dimensional scaling

As the weights $w_{i,j}$ are nonzero only for nodes j in the neighborhood of node i, only the corresponding entries of vector $\boldsymbol{b}$ will be nonzero, and the update rule for $\mathbf{z}_i$ will depend only on this neighborhood (as opposed to the whole matrix $\mathrm{Z}^{(k)}$).

Note that unlike the centralized SMACOF algorithm described in [49], the computation of (4.21) does not require the evaluation of a $n \times n$ Moore-Penrose matrix inverse.

Summary and Comments

The proposed algorithm is summarized in Figure 4.9. Note the following comments:

1. The choice of weighting function $w_{i,j}$ should reflect the accuracy of measured dissimilarities, such that less accurate measurements are down-weighted in the overall cost function. If a noise measurement model is available, $w_{i,j}$ can be tailored to the variance predictions of the model. For example, one might select $w_{i,j} = 1/(c_1\delta_{i,j} + c_2)^2$ if the measurements are Gaussian distributed with standard deviation increasing linearly with the true distances, i.e., $\sigma = c_1\|\mathbf{z}_i - \mathbf{z}_j\| + c_2$. When a reliable model is not available, one can adopt a model-

independent adaptive weighting scheme. This is the approach adopted in this thesis. Inspired by the weighting frequently used in locally weighted regression methods (LOESS) [21], the following weight assignment is proposed:

$$w_{i,j} = \begin{cases} \exp\left\{-\delta_{i,j}^2/h_{i,j}^2\right\} & \text{, if } \delta_{i,j} \text{ is measured} \\ 0 & \text{, otherwise} \end{cases}, \tag{4.24}$$

where $h_{i,j} = \max\left[\{\delta_{i,k}\}_k \cup \{\delta_{k,j}\}_k\right]$. This choice of $w_{i,j}$, both equalizes the (nonzero) weight distribution in all sensors, and is symmetric. This LOESS-based scheme shows robust performance in the experiments reported in Section 4.4.5.

2. The question of how to adaptively choose the neighbors of each node (i.e., which weights are made positive) in order to decrease communication costs or improve localization performance is addressed in Section 4.4.3.

3. The values of r_i should be chosen according to the prior information on node i's location. For a Gaussian distributed prior with variance σ, one should choose $r_i = 1/(2\sigma^2)$. Prior information can have a dramatic impact on localization accuracy, as shown in [106]. If r_i's are very high (compared to the w_{ij}'s), then the solution to (4.12) will 'stretch' range measurements in order to place sensors with prior information at their *a priori* coordinates. Conversely, if r_i's are very low, then the solution will attempt to preserve range measurements and instead find a single global translation and rotation that results in agreement between estimates and prior coordinates. The localization performance as a function of r_i is investigated in Section 4.4.6.

4. Regarding the initialization of the algorithm, every node requires an initial estimate of its position. This can be done using the algorithms proposed in

[105] or [124]: each node builds its local coordinate system, which is then passed along the network until a rough global map of the network is built. In the experiments reported in Section 4.4.5, the algorithm was robust with respect to "rough" initial position estimates.

5. In the description of the algorithm, it was assumed for notational convenience, that the algorithm cycles through the network in an ordered fashion (i.e., messages are passed between nodes in the order $1, 2, \ldots, n$). However, many other non-cyclic update rules are possible. In particular, one possibility is for (spatial) clusters of sensors to iterate among themselves until their position estimates stabilize. These estimates can then be transmitted to the neighboring clusters, before starting a new iteration step.

6. Although the majorization approach used guarantees a non-increasing sequence of STRESS vales, it may converge to a local minimum of this cost function, instead of the global one, like any gradient search method. This behavior can be alleviated to some extent by using some of the advanced search techniques proposed in [49].

4.4.3 Neighbor Selection in Noise

In most sensor network research, each sensor selects its neighbors by choosing those devices which are within a threshold distance (R-radius neighbor selection). But, since the exact distance $\|\mathbf{z}_i - \mathbf{z}_j\|$ is not known, sensors must use noisy measurements $\delta_{i,j}$ to select neighbors. Range measurements, whether made via TOA, RSS, or proximity, are all subject to errors. In this section the biasing effects of selecting neighbors via noisy distance measurements are explored.

When distance is measured in noise, the act of thresholding neighbors based on the

measured distance will tend to select the devices with smaller measured distances. For example, consider two devices separated by distance R, when R is also the threshold distance. With some positive probability (due to noise), the measured distance, δ, will be greater than R, and the two will not be considered neighbors. Alternatively, if $\delta \leq R$, the two will be considered neighbors, and δ will be used in the localization algorithm. The problem is that the expected value of δ, for devices separated by R which consider themselves neighbors, is less than R. Thus, the measured distance is negatively biased because of the effect of thresholding. Note that using KNN for neighbor selection (see Section 4.3.1) effectively has an adaptive threshold, and thus does not avoid this biasing effect.

This bias has not been specifically addressed in the sensor localization literature, because its effects are not severe in certain systems. Some proposed sensor localization systems measure very accurate distances, eg., using TOA in UWB or a combination of RF and ultrasound media – for these systems, the effect of selecting neighbors based on measured distances will be minimal. Alternatively, if neighbors are selected based an independent means (eg., based on RSS or connectivity when range estimates are based on TOA), than the biasing effect is avoided[1]. Finally, when studies show results for the case in which all devices are connected to every other device, the thresholding step (and its biasing effect) is eliminated. In this thesis, the full-connectivity assumption is explicitly stated when it is used. The manifold learning-based localization algorithms presented in this and in the next sections use thresholding and noisy measurements, and are thus subject to the biasing effect, but are specifically designed to mitigate it.

The discussion is limited to RSS measurements in this section, since low device

[1]Note, however, that if both RSS and TOA are available, one may wish to use a combination of them; and even if not, RSS and TOA for a link are correlated because objects in the environment tend to degrade both measurements simultaneously.

costs and energy consumption are very attractive device characteristics of RSS, but the discussion is also applicable to systems which use noisy TOA-based range measurements for neighbor selection.

RSS-based Biasing Effect

When discussing thresholding based on RSS, a distinction must be made between the physical limits of the receiver and the threshold which is used to select neighbors, because generally, the two do not need to be the same. If a device has a large radio range in order to be robust to low device densities, a system designer may set a stricter threshold when there are very many devices with which the sensor can communicate. Denote P_0 to be the received power level below which a receiver cannot demodulate packets. This assumption that connectivity is a binary quantization of received power is an assumption addressed in Section 2.5.1. Denote P_R to be the received power level below which the transmitting device is not included as a neighbor. Clearly, $P_R \geq P_0$. Equivalently, distances d_0 and R can be defined from (2.5) to be the maximum-likelihood estimate of range given the received power $P_{i,j}$ is equal to P_0 and P_R, respectively. (d_0 is the same quantity defined for connectivity measurements in Section 2.5 by (2.16).)

Whether or not neighbors are selected based on connectivity (measured power is greater than P_0) or based on a power threshold (measured power is greater than P_R), the biasing effect will be the same. In following, P_R and R are used to indicate the thresholds (which may be set equal to P_0 and d_0 if desired).

Let $E\left[\delta_{i,j} | P_{i,j} > P_R\right]$ be the expected value of the range estimate between devices i and j given that the two are neighbors (i.e., the received power $P_{i,j}$ is greater than

P_R). Using the RSS measurement model (see Section 2.2), it can be shown that

$$E\left[\delta_{i,j}|P_{i,j} > P_R\right] = C\|\mathbf{z}_i - \mathbf{z}_j\| \frac{1 - \Phi\left(\sqrt{\gamma}\log\frac{\|\mathbf{z}_i-\mathbf{z}_j\|}{R} + \frac{1}{\sqrt{\gamma}}\right)}{1 - \Phi\left(\sqrt{\gamma}\log\frac{\|\mathbf{z}_i-\mathbf{z}_j\|}{R}\right)}, \tag{4.25}$$

where $\Phi(\cdot)$ is the cumulative distribution function of a standard Gaussian random variable, γ is the RSS channel parameter given in Table 3.1, and C is the constant (bias multiplier) given in (4.3). Equation (4.25) is plotted in Fig. 4.10 as a function of the ratio of the true distance to R. Ideally, the range estimator should have a mean value equal to the actual range. However, as the range increases, the expected value of $\delta_{i,j}$ (given that i and j are neighbors) deviates from linear and asymptotically becomes constant. There is a strong negative bias for devices separated by R or greater.

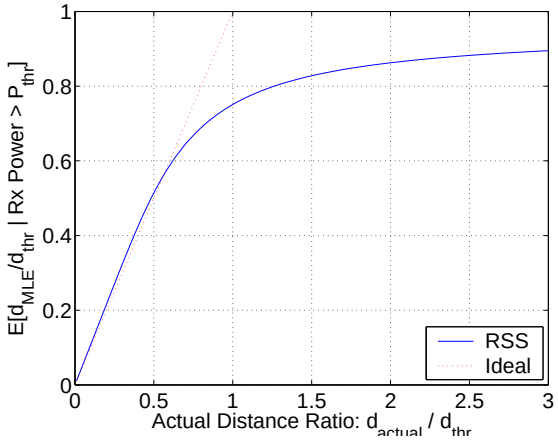


Figure 4.10: The expected value of the RSS-based estimate of range given that that two devices are neighbors (- - -), and the ideal unbiased performance ($\cdots\cdots$). The channel has $\sigma_{dB}/n_p = 1.7$ and $R = 1$ (or equivalently, distances are normalized by R).

4.4.4 Two-Stage Selection Algorithm

Motivated by the negative bias phenomenon displayed in Fig. 4.10, a two stage neighborhood selection process is proposed.

In the first stage, the dwMDS algorithm from Fig. 4.9 is run with a neighborhood structure based on the available range measurements, i.e., set $w_{i,j} = 0$ if $\delta_{i,j} > R$.

After convergence, this step provides an interim estimate $\{\hat{\mathbf{z}}_i\}$ of the sensors locations. With high probability, the predicted distances between the estimated sensor locations will be negatively biased.

In the second stage, these predicted distances from the estimated sensor locations are used to compute a new neighborhood structure, by assigning $w_{i,j} = 0$ if $\|\hat{\mathbf{z}}_i - \hat{\mathbf{z}}_j\| > R$. Some neighbors with low range measurements will be dropped, and some neighbors with possibly longer range measurements will be added. Then, using $\{\hat{\mathbf{z}}_i\}$ as an initial condition and the new neighborhood structure, the dwMDS algorithm is re-run, resulting in the final location estimates. Note that the predicted distances $\|\hat{\mathbf{z}}_i - \hat{\mathbf{z}}_j\|$ are used only to select neighbors (i.e., which weights are positive) – the measured ranges $\delta_{i,j}$ are still used to determine the positive weight values.

Note that this 2-stage algorithm does not imply twice the computation of a one-stage algorithm. The dwMDS algorithm is based on majorization, and each iteration brings it closer to convergence. Since the first stage only needs to provide coarse localization information, it does not need to be very accurate, and so the dwMDS algorithm can be stopped quickly with a large ϵ. Next, the second stage begins with very good (although biased) coordinate estimates, so the second run of the dwMDS algorithm will likely require fewer iterations to converge.

Note that for some of the devices which are considered neighbors in the 2nd stage of the algorithm, the measured range $\delta_{i,j}$ will actually be greater than R. Thus, to use this 2-stage algorithm, R must be sufficiently less than the physical communication limit of the devices, d_0, so that other range measurements can be considered. Considering the non-circular (real-world) coverage area of a device, d_0 can be considered to be the mean radius of the coverage area, while R should be set to the minimum radius of the coverage area.

4.4.5 Simulation Results

In this section, all the simulated data were generated from the RSS measurement model presented in Section 2.2, with channel parameter $\sigma_{dB}/n_p = 1.7$.

First dwMDS algorithm performance is demonstrated on a network of 7×7 sensors arranged on a uniform grid of unit area, as was drawn in Fig. 3.7, in which the four corner devices are reference nodes and the remaining 45 are unknown location devices, and $L = 1$ m. For all experiments on this configuration, $R = 0.4\,\text{m}$ is used (yielding an average of 14 neighbors per device). A Monte Carlo simulation is run with 200 trials to determine confidence ellipses, root-mean-square error (RMSE) and bias performance (per sensor) of the location estimates. The results are displayed in Figure 4.11, which plots the mean and 1-σ uncertainty ellipse of the estimator, compared to the actual device location and the Cramér-Rao lower bound (CRB) on the uncertainty ellipses (presented in Chapter III). Note that the CRB shown is calculated assuming full connectivity (all devices measure range to all other devices), and as such provides only a loose lower bound on the best performance achievable by any unbiased estimator. The first experiment is a baseline best-case scenario, in which perfect (noise-free) distance measurements are used to select neighborhoods (even though noisy measurements are used as measurements $\{\delta_{i,j}\}$). The baseline assumes that an oracle is available to declare when the true distance between i and j is less than a threshold, *i.e.*, $\|\mathbf{z}_i - \mathbf{z}_j\| < R$. This is shown in Figure 4.11(a), resulting in a RMSE of the location estimates of 0.090 m and an average bias of 0.019 m.

For the second experiment, the perfect connectivity knowledge is removed. Instead, RSS measurements are used to select neighbors, i.e., devices i and j are neighbors if $P_{i,j} \geq P_R$, or, equivalently, if $\delta_{i,j} \leq R$. The results are shown in Figure 4.11(b). The estimates are strongly pulled towards the center of the square, due

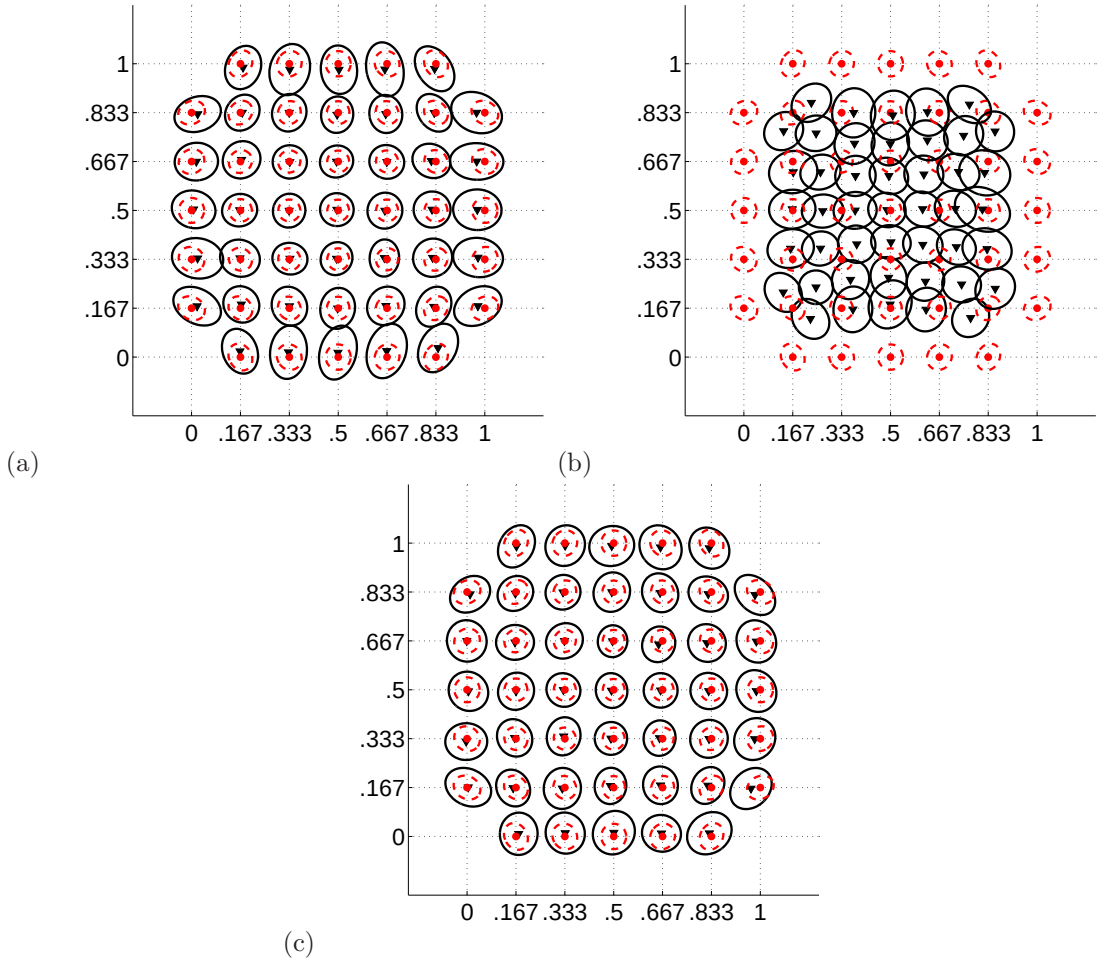


Figure 4.11: Comparison of dwMDS estimators when neighborhood selection is done using (a) oracle to obtain actual distances, (b) measured distances, or (c) adaptive, 2-stage selection algorithm. Estimator mean (▾) and 1-σ uncertainty ellipse (—) for each unknown-location sensor compared to the true location (·) and CRB on the 1-σ uncertainty ellipse (- - -).

to the negative bias of the range estimates which are 'selected' by the connectivity condition. Now, the RMSE is 0.162 m and the bias is 0.130 m.

A third experiment uses the 2-stage adaptive neighborhood selection method proposed in Section 4.4.4. The results are displayed in Figure 4.11(c), where it can be seen that this method succeeds in removing the negative bias effect. The bias has gone back down to 0.012 m, while the RMSE is 0.092m, just slightly higher than the baseline experiment using the oracle.

Comparing Figures 4.11(c) and (a), the localization errors of the two-stage algorithm are spread more evenly throughout the network compared to the first experiment – the errors for edge devices are reduced, while those in the center have increased. Based on the similarity of the RMSE in both experiments, it is believed that the 2-stage algorithm eliminates most of the neighbor selection bias. Additionally, by changing the neighbor lists (and therefore the cost function) and re-running the dwMDS algorithm, the 2nd iteration also provides the opportunity to break out local maxima, which are more likely to affect edge devices. Finally, the low variance achieved by the 2-stage algorithm is very close to the CRB which no unbiased location estimator can outperform, despite the fact that the CRB is an optimistic bound for the scenario considered here.

The influence of the threshold distance on the RMSE performance of the proposed algorithms has also been studied. Figure 4.12 shows a plot of the RMSE vs. threshold distance, for the 7×7 uniform grid example using adaptive neighborhood selection. It can be seen that there is an optimal threshold distance, $R = 0.5$ m, beyond which, no performance increase occurs. As R is increased beyond this optimal value, more distant sensors are included in the cost function. By the RSS measurement model, the accuracy of range measurements degrades quickly with distance, thus adding

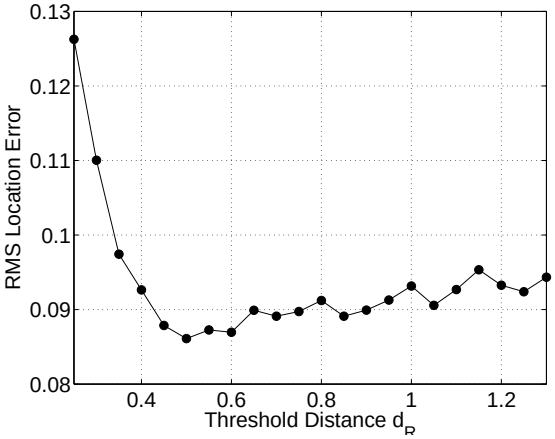


Figure 4.12: RMSE versus threshold distance for the 7×7 uniform grid example using adaptive neighborhood selection.

these far way sensors will not bring any gain to the estimation algorithm.

4.4.6 Influence of Prior Weight r_i

To test how the estimator performs as a function of r_i, the following simulations are run. Under the same 7×7 grid scenario, consider the four corner nodes to have imperfect information (rather than perfect *a priori* coordinate knowledge). In particular, the algorithm only has access to a noisy version of the actual coordinates of these nodes, perturbed by zero mean Gaussian noise with unknown variance σ_p^2. Figure 4.13 shows the resulting RMSE, obtained by running 5000 Monte-Carlo 2-stage dwMDS simulation trials for $\sigma_p = 0:025, 0:050, 0:100$ and setting $r_i = r$ for the corner nodes, where r is made to vary between 10^{-2} and 10^2.

When r is very small (10^{-2}), the RMSE levels off to a value that is constant for all values of σ_p. Essentially, the prior is only being used to rotate and translate the relative map produced by the dwMDS in order to best fit the observed positions of the corner sensors. This constant RMSE at low r is not too far from the minimal RMSE, so a reasonable non-parametric option might be to set $r = 10^{-2}$. On the other hand, for very high values of r, the observed positions of the corner sensors are

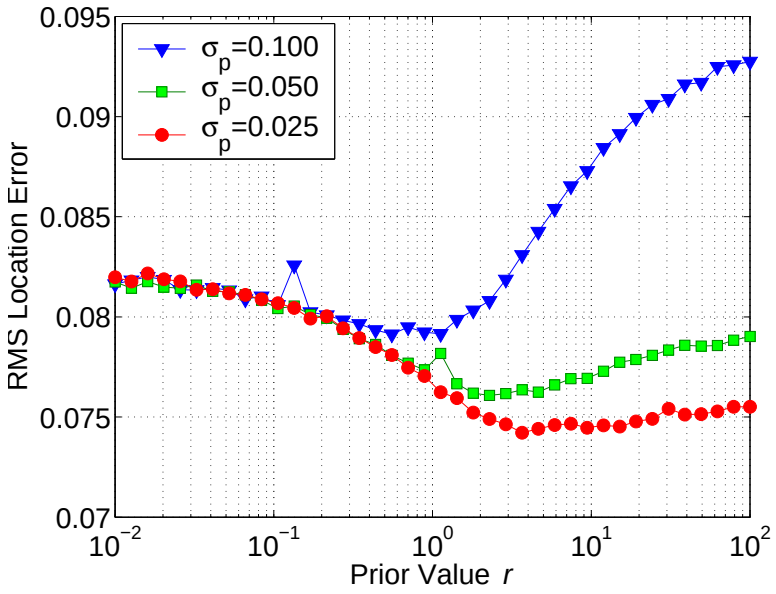


Figure 4.13: RMS location error versus prior weighting value r of the four corner nodes in the 7×7 grid example when the corner nodes coordinates are observed in different levels of noise σ_p.

treated as fact, and the map is stretched to place the corner sensors exactly where they were observed, to the detriment of fitting measured ranges.

For intermediate values of r, there is an optimal r which best weights the relative information in the prior coordinates with respect to the weights chosen for the measured ranges. As would be expected, the optimal r is inversely proportional to σ_p^2, although the exact dependence may depend on factors such as the distribution of n_i, the weighting scheme chosen, and the number of neighbors of each node. Further research should investigate these dependencies. However, the RMS location error near the optimal is a very shallow function of r – for all three plots, there is nearly an order of magnitude range within which the RMS location error is within 1% of its minimum. So, although simulation might be necessary to find the optimal r, as long as r is within the correct order of magnitude, the results will be nearly optimal. If r is set to an intermediate value, the resulting RMS location error will be lower than that possible using either:

- A method that uses prior coordinate information only to find the best rotation of a calculated relative map (i.e. [110]), which is analogous to very low r in the dwMDS method, or
- An MLE which assumes that reference sensor coordinates are known perfectly (i.e. [92], which is analogous to very high r.

4.4.7 Measurement Results

To test the performance of the proposed algorithm on real-world channel measurements, the RSS and TOA measurements presented in detail in Section 2.6.2 are used. Recall that this data set includes the RSS and TOA range measurements from a network of 44 devices in an office area. The bias-corrected pseudo-MLE from (2.8) is used to estimate range $\delta_{i,j}$ from the measured RSS. Of the $N = 44$ nodes, $m = 4$ are set to be reference nodes (*i.e.*with perfect prior information).

The performance of the dwMDS algorithm with adaptive neighborhood selection is compared to classical MDS and the MLE based solutions from [92]. Figures 4.14(a) and (b) show the location estimates using classical MDS (which used all the pairwise range measurements between sensors) and the dwMDS algorithm, for the RSS measurement data set. The true and estimated sensor positions are marked by 'o' and '▽', respectively, where the lines represent the estimation errors. The reference nodes are marked with an 'x'. It can be observed that the dwMDS algorithm does much better than classical MDS. In fact, the RMSE for the classical MDS solution is 4.30 m, while for dwMDS, using $R = 6$ m (yielding an average of 19 neighbors per sensor), it drops to 2.48 m. This error is 14% higher than the RMSE of 2.18 m achieved using a centralized MLE. However, note that the MLE uses all pairwise range measurements, and relies on previously estimated channel parameters. If R is

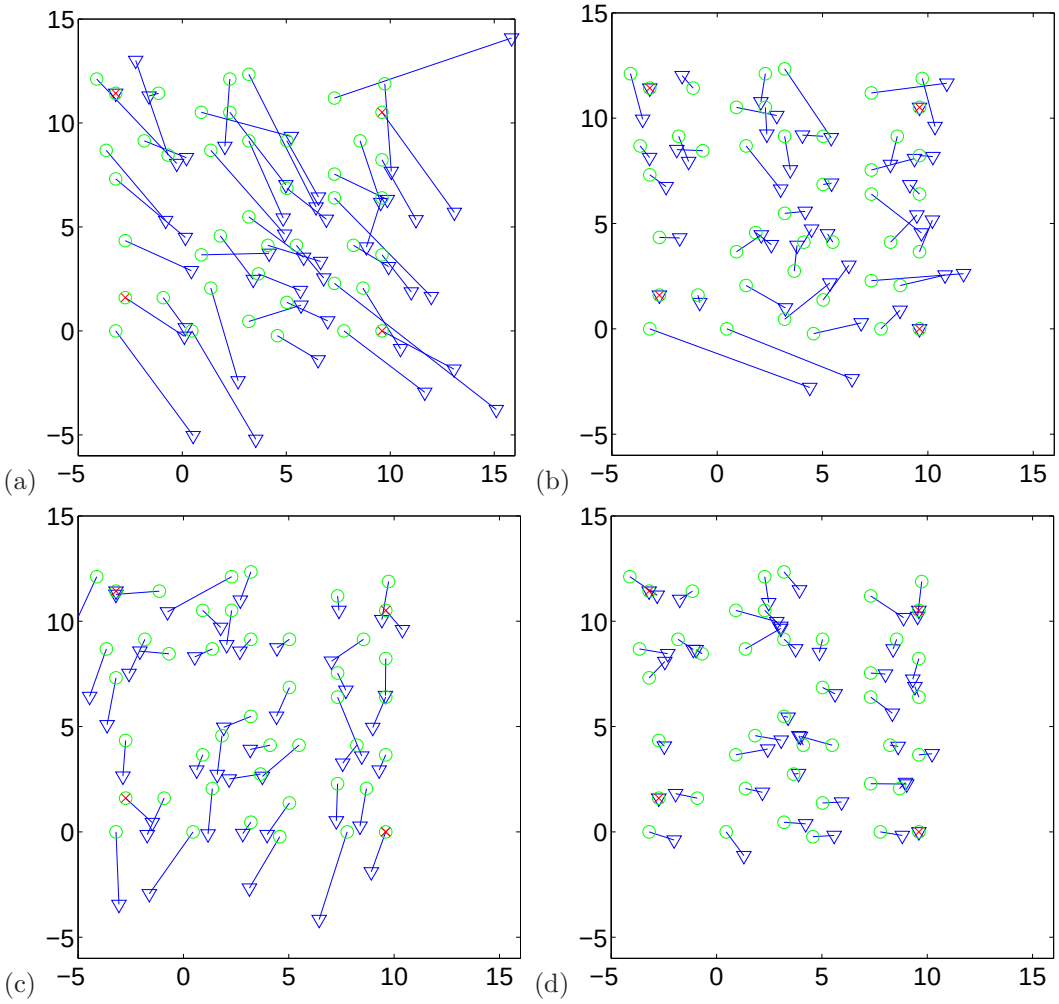


Figure 4.14: Location estimates using (a-b) RSS and (c-d) TOA range measurements from experimental sensor network, comparing results from (a & c) Classical MDS and (b & d) dwMDS estimators. True and estimated sensor locations are marked, respectively, by 'o' and '▽', while reference nodes are marked by 'x'. The dwMDS algorithm uses adaptive neighbor selection, with $R = 6\,\mathrm{m}$.

allowed to increase at the expense of increasing communication costs, the dwMDS algorithm can reach an RMSE as low as 2.27 m for $R = 8.5$ m, just 4% higher than the MLE.

Figures 4.14(c) and (d) show again the location estimates using classical MDS and the dwMDS algorithm, but this time for the TOA measurement data set. The RMSE for the classical MDS solution is 1.96 m and 1.12 m for the dwMDS algorithm using $R = 6$ m (wich yields an average of 17.5 neighbors per sensor). This error is slightly better than the RMSE of 1.23 m achieved using a centralized MLE. If R is allowed to increase at the expense of increasing communication costs, the dwMDS algorithm can reach an RMSE as low as 0.94 m for $R = 7.5$ m. Once again, note that the dwMDS algorithm, unlike the MLE estimator from [92] and Section 4.2, does not use all the pairwise range measurements and doesn't assume knowledge of the distribution of the range measurements.

4.5 Laplacian Eigenmap Connectivity-based Estimator

This section develops and compares localization algorithms which solely use connectivity measurements as location information. Recall that if a sensor i can successfully demodulate the packets transmitted by another sensor j, then the two are considered to be connected, *i.e.*, $Q_{i,j} = 1$. When received signal strength (RSS) is too low, packets can't be demodulated, and sensors will not be connected, *i.e.*, $Q_{i,j} = 0$. Connectivity was described in more detail in Section 2.5.1.

This section emphasizes that connectivity is a noisy measurement, and as a result, the bias and variance of any connectivity-based coordinate estimator must be considered. In addition to accuracy and precision, the estimator should have low computational complexity so that sensor localization will scale well with the size

of the network. This section introduces a Laplacian Eigenmap based localization method which has both lower computational complexity and lower variance than MDS-based methods.

In this section, imperfect prior coordinate information is not considered - sensors either have no prior information (unknown-location nodes) or perfect prior information (reference nodes). However, allowing imperfect prior information would be a simple extension to the developed method.

Connectivity measurement-based localization algorithms, also called *range-free* localization, have found considerable application in ad hoc networks and wireless sensor networks, *eg.*, in [81, 110, 117]. In particular, connectivity-based localization via MDS was introduced in [110], which demonstrated that localization can be achieved without resorting to iterative optimization algorithms that don't always converge to the global maxima. The MDS-MAP method in [110] effectively applies the manifold learning technique called Isomap [118] to the connectivity-based sensor localization problem. The new method is compared to the MDS-MAP method in Sections 4.3.3 and 4.5.4.

4.5.1 Weight Selection

The selection of weights $w_{i,j}$ for neighboring sensors is critical to localization performance, as shown earlier for the dwMDS method in Section 4.4. In the original Laplacian Eigenmap (LE) method [8], weights are selected by looking at the local geometric structure of neighboring high-dimensional data points. In this connectivity-based problem, only a coarse idea of local structure is available, since measurements are binary. Also, the algorithm begins with 'distance' (connectivity) measurements rather than higher-dimensional coordinates themselves, so the weight selection method of [8] cannot be directly applied.

This section presents multiple methods of weight selection, and then compares them via simulation in Section 4.5.4.

First, in the *Equal Weights* method, set $w_{i,j} = Q_{i,j}$, *i.e.*, $w_{i,j} = 1$ if i and j are connected and 0 if not. As will be shown in Section 4.5.4, this is a poor weight selection method, because sensors with the most neighbors will tend to have too much 'pull', and will bias their neighbors' coordinate estimates too close to their own.

To counteract this tendency, two alternatives are offered which both affect the column sums of W, *i.e.*, $u_i = \sum_{j=1}^{N} w_{i,j}$. Note that u_i is analogous to the 'pull' of sensor i. In both alternative methods, $\{w_{i,j}\}$ are first set using the equal weights method. Then,

Equal Sum-of-Weights : Adjust the weights such that the new column sums $\tilde{u}_i = \mu_u$ for all $i = 1 \dots N$, where μ_u is the average of the original column sums, $\mu_u = \frac{1}{N} \sum_{i=1}^{N} u_i$.

Linear Sum-of-Weights : Adjust the weights such that the new column sums $\tilde{u}_i$ are linearly related to u_i. Specifically, let $\tilde{u}_i = \mu_u + \beta(u_i - \mu_u)/\sigma_u$ where σ_u is the standard deviation of $\{u_i\}_{i=1}^{N}$. In this chapter, slope $\beta = 0.1$ is used throughout.

Adjustment of weights to achieve desired column sums is described in Section 4.5.2. The W output by any neighbor weight selection method is then used to calculate coordinate estimates $\{\tilde{\mathbf{z}}_i\}$ via the Laplacian Eigenmap algorithm in Section 4.3.

4.5.2 Symmetric Adjustment of Weights

Matrix W must remain symmetric after any weight adjustment, since it describes a symmetric graph. If the weights in column i were just scaled by $\tilde{u}_i/u_i$, column i would have the desired sum $\tilde{u}_i$, but W would not remain symmetric. In this weight-

adjustment algorithm, we iteratively adjust $\{w_{i,j}\}$ until $\tilde{u}_i = u_i$. The inputs to the algorithm are: the original weights $\{w_{i,j}\}$; the desired sum of weights $\{\tilde{u}_i\}$ for $i = 1 \ldots N$; and a convergence threshold ϵ (here $\epsilon = 0.01$). The algorithm outputs the modified weight matrix. The steps are:

1. Calculate $u_i = \sum_{j=1}^{N} w_{i,j}$ for $i = 1 \ldots N$.
2. Define $\phi_i = \sqrt{\tilde{u}_i / u_i}$, for $i = 1 \ldots N$.
3. Assign $w_{i,j} = w_{j,i} := \phi_i w_{i,j} \phi_j \; \forall$ neighbors i, j. (In matrix notation, $W := \Phi W \Phi$ where $\Phi = \text{diag}[\phi_1, \ldots, \phi_N]$.)
4. If $\forall i$, $1 - \epsilon < \phi_i < 1 + \epsilon$, stop. Else go to 1.

The algorithm requires $\mathcal{O}(KN)$ multiplies, where K is the average number of neighbors. The convergence of this algorithm is not addressed here, except to note that in simulations, it typically converges in 5-10 iterations for $\epsilon = 0.001$.

4.5.3 Two-Stage Weight Selection

From experience with the dwMDS algorithm, it is apparent that localization estimates can be greatly improved by using a two-stage neighbor selection method. Thus the following two-stage algorithm for weight selection is tested:

1. Using the linear sum-of-weights method to set W, calculate the Laplacian Eigenmap coordinate estimates $\{\tilde{\mathbf{z}}_i\}$.
2. For the 2nd round, let the desired column sums $\tilde{\tilde{u}}_i$ be

$$\tilde{\tilde{u}}_i = \tilde{u}_i \sqrt{K_i / \tilde{K}_i} \tag{4.26}$$

where $\tilde{K}_i$ is the number of its neighbors j for which $\|\tilde{\mathbf{z}}_i - \tilde{\mathbf{z}}_j\| < R$, and R is the radius of coverage. Adjust W to meet $\{\tilde{\tilde{u}}_i\}$ as described in Section 4.5.2,

and then calculate final coordinate estimates $\{\hat{\mathbf{z}}_i\}_{i=1\ldots N}$ using the new weight matrix.

Intuitively, if few of the neighbors of sensor i are estimated to be within its communication range, then one can guess that sensor i's weights should be increased. The presented choices are by no means optimal, and other iterative algorithms or updates are certainly possible. This section simply demonstrates that the performance of this ad hoc two-stage method does in fact dramatically improve localization performance.

4.5.4 Simulation Results

In this section localization performance is tested using different estimators in multiple sensor geometries. For each test, 200 Monte-Carlo simulation trials were run in order to determine the mean coordinate estimate $\bar{\mathbf{z}}_i$ for $i = 1 \ldots n$, and the covariance matrix $\mathbf{C}$. In each simulation, the statistical model in Section 2.5.1 is used to randomly generate connectivity measurements in the sensor network. Each plot in Fig. 4.15 shows $\bar{\mathbf{z}}_i$ (▼) and the 1-σ covariance ellipse (—-) for each sensor. For comparison, plotted in gray (or red in the electronic version) is the actual sensor coordinate (●) and the Cramér-Rao bound (CRB) for the 1-σ covariance ellipse (- - -) [88]. For each test, the mean bias $\bar{b}$ and the RMS standard deviation $\bar{\sigma}$ summarize the performance of the localization estimator, which were defined in (4.5). Note all distances are in terms of L, the chosen scale of the network.

MDS-MAP

First MDS-MAP is tested in the 7 by 7 grid network, as drawn in Fig. 3.7, in which the four corner sensors are reference sensors, and the other 45 are unknown-location sensors. For a communication radius $R = 0.5$, the MDS method has standard deviation of location error $\bar{\sigma} = 0.218$ and a bias of $\bar{b} = 0.087$, as shown in Fig. 4.15(a).

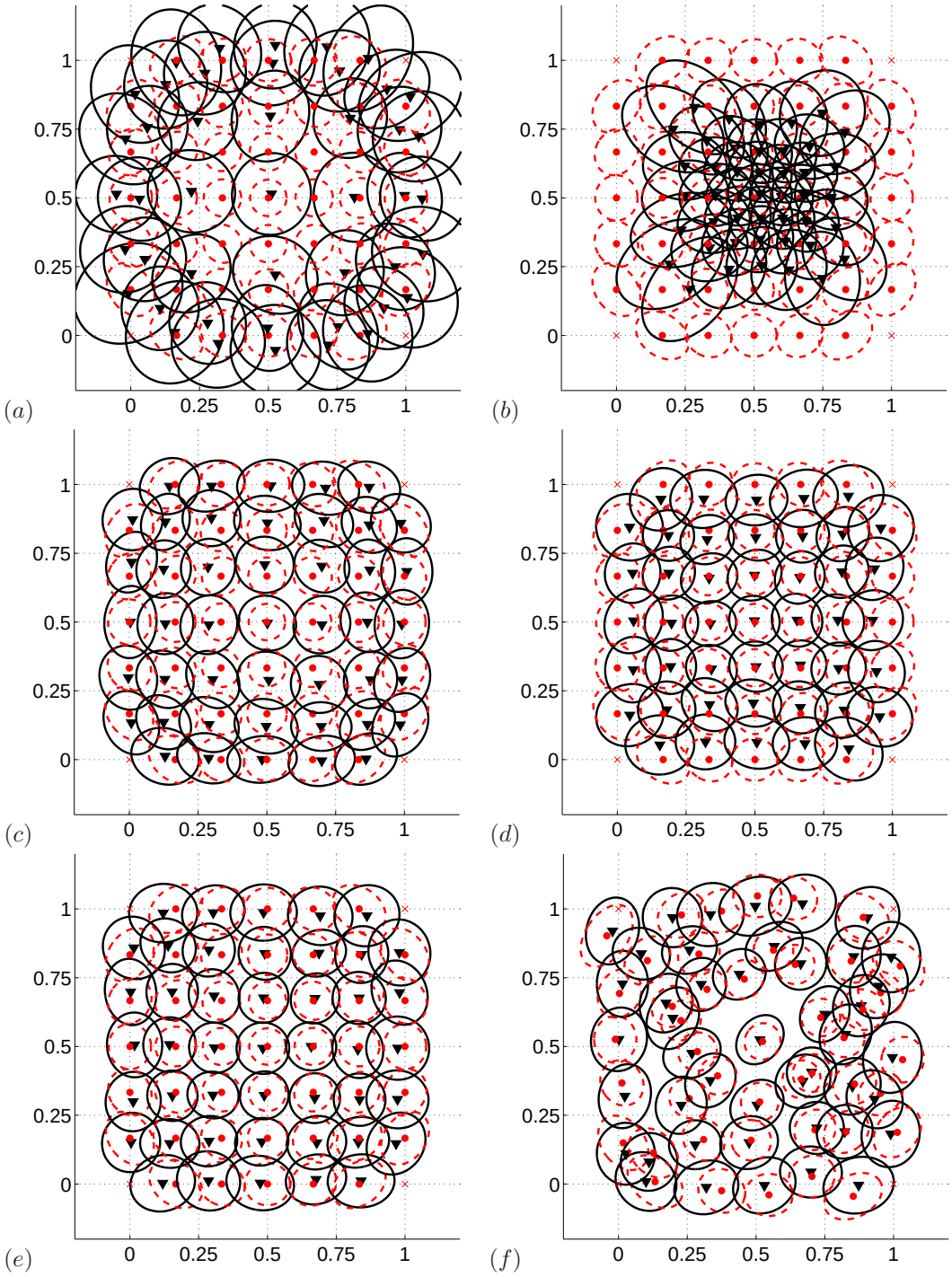


Figure 4.15: Estimator mean (▼) and 1-σ uncertainty ellipse (—-) for each unknown-location sensor compared to the true location (●) and CRB on the 1-σ uncertainty ellipse (- - - -), when reference sensors are located at each ×. All cases are $R = 0.5$ tests described in Section 5 and in Table 1.

Location Estimator	MDS-MAP	LE Eql. Wts.	LE Eql. $\sum$-Wts.	LE Lin. $\sum$-Wts.	LE 2-Stage Linear Sum-of-Weights			
Geometry	7 by 7 Grid					Grid+$\frac{Z}{4}$	Grid+$\frac{Z}{2}$	Unif. Rand
R = 0.3	$\bar{b} = 0.026$	$\bar{b} = 0.106$	$\bar{b} = 0.056$	$\bar{b} = 0.048$	$\bar{b} = 0.039$	$\bar{b} = 0.046$	$\bar{b} = 0.062$	$\bar{b} = 0.069$
	$\bar{\sigma} = 0.205$	$\bar{\sigma} = 0.191$	$\bar{\sigma} = 0.153$	$\bar{\sigma} = 0.153$	$\bar{\sigma} = 0.133$	$\bar{\sigma} = 0.142$	$\bar{\sigma} = 0.155$	$\bar{\sigma} = 0.126$
R = 0.4	$\bar{b} = 0.022$	$\bar{b} = 0.154$	$\bar{b} = 0.059$	$\bar{b} = 0.035$	$\bar{b} = 0.033$	$\bar{b} = 0.037$	$\bar{b} = 0.055$	$\bar{b} = 0.048$
	$\bar{\sigma} = 0.205$	$\bar{\sigma} = 0.188$	$\bar{\sigma} = 0.143$	$\bar{\sigma} = 0.144$	$\bar{\sigma} = 0.136$	$\bar{\sigma} = 0.139$	$\bar{\sigma} = 0.141$	$\bar{\sigma} = 0.127$
R = 0.5	$\bar{b} = 0.087$	$\bar{b} = 0.186$	$\bar{b} = 0.040$	$\bar{b} = 0.036$	$\bar{b} = 0.026$	$\bar{b} = 0.027$	$\bar{b} = 0.040$	$\bar{b} = 0.031$
	$\bar{\sigma} = 0.218$	$\bar{\sigma} = 0.189$	$\bar{\sigma} = 0.149$	$\bar{\sigma} = 0.146$	$\bar{\sigma} = 0.144$	$\bar{\sigma} = 0.147$	$\bar{\sigma} = 0.149$	$\bar{\sigma} = 0.140$
Figure	4.15(a)	4.15(b)	4.15(c)	4.15(d)	4.15(e)	4.15(f)	4.16(g)	4.16(h)

Table 4.2: Simulated performance of MDS-MAP and Laplacian Eigenmap (LE) using equal weights, equal sum-of-weights, linear sum-of-weights, and two-stage linear sum-of-weights, for different ranges R. The sensor geometries are the 7 by 7 grid, grid plus noise ($c = 2$ and $c = 4$), and uniform random.

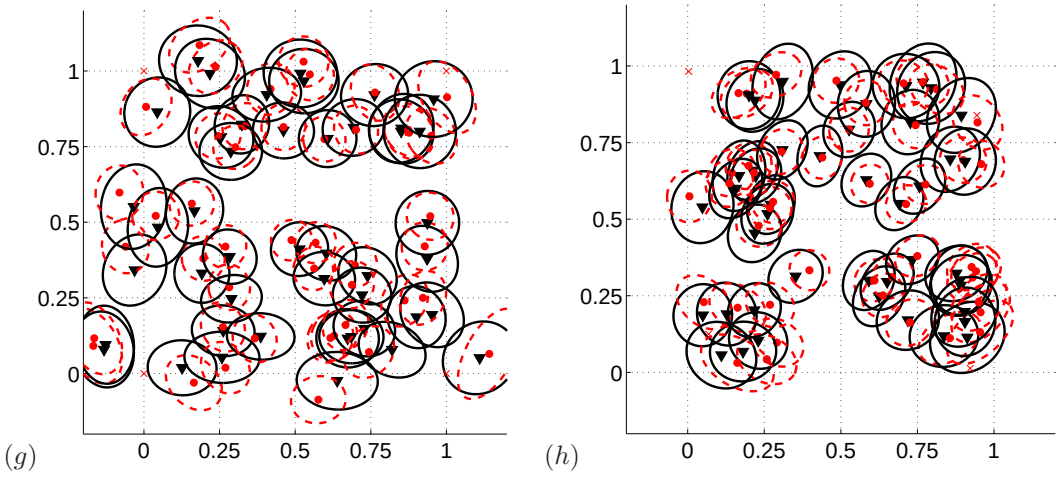


Figure 4.16: Continued from Figure 4.15.

At $R = 0.5$, almost all pairs of sensors are within 1 or 2 hops from each other. At lower radii R, the MDS-MAP achieves very low bias, as shown in Table 4.5.4, but the standard deviation of error is largely constant, consistently about twice the lower bound of the CRB.

Laplacian Eigenmap One-Stage

Equal Weights: Next, the Laplacian Eigenmap is tested using the equal weights method (as described in Section 4.5.1). The simulation results show a heavily biased estimator. For $R = 0.5$, the results are shown in Fig. 4.15(b), in which the mean bias $\bar{b} = 0.186$ and the standard deviation of location error $\bar{\sigma} = 0.189$. At $R = 0.3$ and $R = 0.4$, the biases $\bar{b}$ listed in Table 4.5.4 are lower but still very high.

Equal Sum-of-Weights: The performance of Laplacian Eigenmap, when weights are determined by the equal sum-of-weights method, is dramatically better than the equal weights method, as shown in Table 4.5.4. For $R = 0.5$, the results shown in Fig. 4.15(c) show that the edge nodes seem to have weights too high compared to the interior nodes, the opposite bias pattern compared to Fig. 4.15(b).

Linear Sum-of-Weights: The Laplacian Eigenmap with adjusted sum-of-weights reduces the bias compared to equal sum-of-weights. As shown in Table 4.5.4 and in Fig. 4.15(d), the bias has been reduced, especially at $R = 0.4$, even though the values of $\bar{\sigma}$ are largely unchanged.

Laplacian Eigenmap Two-Stage

Using the two-stage weight adjustment described in Section 4.5.3, bias is further reduced. Furthermore, as shown in Table 4.5.4, the variance for $R = 0.3$ and $R = 0.4$ is dramatically lower than the one-stage linear sum-of-weights method. These variances in the grid geometry are about 30-35% higher than the Cramér-Rao lower

bound, so even an efficient estimator would not reduce $\bar{\sigma}$ dramatically further.

However, we certainly don't expect that sensors will be arranged in a perfect grid. The true test of sensor localization is performance sensor placement is random, which is presented next. Each test shows the performance of the Laplacian Eigenmap two-stage weight selection method.

Grid Plus Noise: First, a Gaussian random vector is added to each unknown grid coordinate, *i.e.*, for $i = 1 \ldots n$, $\mathbf{z}_i = \dot{\mathbf{z}}_i + \mathbf{Z_i}/c$, where $\dot{\mathbf{z}}_i$ is the original coordinate on the 7 by 7 grid, and $\{\mathbf{Z_i}\}$ are independent Gaussian-distributed with mean zero and covariance $(1/6)^2\mathrm{I}_2$, and $c = 2$ or 4. Essentially, the standard deviation of the random addition is either one-fourth or one-half of the distance between grid nodes. Two geometries are generated from this model for $c = 2$ and $c = 4$, and simulation results are shown in Table 4.5.4 and in Fig. 4.15(f) and Fig. 4.16(g), respectively. For $R = 0.4$ and 0.5, the bias and standard deviation of location error increase only slowly. However, for $R = 0.3$, the bias and variance do increase considerably. Note that sensors actually located outside of the unit square $[0, 1]^2$ have noticeably higher bias and variance.

Uniform Random: Next, for $i = 1 \ldots N$, $\mathbf{z}_i$ are independently chosen from a uniform distribution over the unit square area, $[0, 1]^2$, as seen in Fig. 4.15(h). The sensors closest to each corner are selected as the 4 references, so in this test, even the references are randomly deployed. The resulting $\bar{\sigma}$ are lower than in the 7 by 7 grid or the grid plus noise geometries. Note that the CRB for $\bar{\sigma}$ is also about 15% lower for this deployment compared to the 7 by 7 grid, so it is legitimate to expect lower $\bar{\sigma}$. Essentially, sensors very close together can provide increased information about their relative location. However, the biases $\bar{b}$ are higher than the 7 by 7 grid, especially for $R = 0.3$.

4.5.5 Discussion

For random deployments, a low communication radius like $R = 0.3$ may cause some sensors to have very few neighbors, and localization performance will suffer. System designers should plan for the tendency of sensors outside of the convex hull of the reference nodes to experience higher localization errors. This behavior is a result of using similarity-based algorithms like Laplacian Eigenmaps, in which sensors only 'pull' closer together, as described in Section 4.3. It is difficult for such methods to accurately locate sensors outside of the reference nodes. If reference node locations can be placed, some should be placed on the edge of the desired coverage area.

Using a realistic statistical model for connectivity, simulations presented in this section show the potential of the Laplacian Eigenmap method to be a robust, low-bias and low-variance sensor location estimator. It does not suffer from local optima and it has low computational complexity compared to MDS-based estimators such as [110]. The presented two-stage weight-selection method is used to achieve low bias and standard deviation within 35% of the lower bound. However, general analysis of weight selection methods has not been attempted. Finally, distributed algorithms have not yet been presented for the proposed methods. These issues remain open for future research.

4.6 Sensor Field Data Localization

This section explores using sensor data measurements to provide sensor location information. When sensors are deployed in order to be able to measure and monitor a time-varying, spatially correlated field, it is proposed to use the measurements simultaneously to extract sensor location information. Localization using field data is possible when a high density of sensors in the network results in correlation in

data recorded at neighboring sensors. This high density and correlated field is already expected to be an attribute of many wireless sensor networks. Due to the randomness of deployment - sensors may be dropped from a helicopter or spread onto a field from a tractor - high average density is desirable to ensure complete coverage. Also, for the manufacture of inexpensive (sometimes unreliable) devices, redundancy increases reliability and robustness to sensor failures. Finally, correlation is useful to reduce data rate (and lengthen battery lifetime) via distributed compression [98].

A startup period is required in which sensor data is measured to establish correlations between the field measurements made at different sensors. For long-term monitoring applications using static sensors, a one-time setup delay can be readily justified. Furthermore, inexpensive devices are likely to additionally use RF proximity or RSS information to estimate sensor locations [88, 92]. In a real sensor network, field data might be useful as complementary information, used alongside RSS or other distance measurement. It might become an important means of verification, in a secure localization system, that a sensor which claims to be close to another neighbor, is actually a neighbor. However, this section considers localization performance using solely sensor field data. Localization using multi-modal measurements is an important topic of future research.

4.6.1 Application Examples

Consider the precision agriculture application. Sensors in the soil measuring soil pH level, salinity, nitrogen level, and moisture level, will allow farmers to customize the planting, watering, herbicide application, and fertilizing of their field so to maximize their crop and minimize their costs and negative environmental impacts [102]. Soil chemistry at various areas of the field changes over time due to weather, biological effects, watering or fertilization. Over days or months, the soil conditions

may be viewed as a random field, correlated in time and space. Similarly, consider acoustic sensor networks deployed over wide outdoor areas for the purpose of source localization. The ambient noise in the environment, caused by birds, wind, lightning, vehicles, and pedestrian traffic, will show significant spatial correlation, since sound attenuates with distance. Sensor self-location was presented for acoustic sensor networks when sound sources are deterministically inserted into the environment [76] or dynamically tracked through the environment [18]. In this section, such purposeful sources may be a part of the deployment, but more generally, the effect of the existing environmental sources is considered. It is assumed in this case that the acoustic field is a stationary and isotropic spatially correlated random field.

As another example, consider wireless sensors attached or built into to the structure of bridges or buildings to monitor vibrations and structural health. The vibrations experienced at nearby sensors should be strongly correlated, possibly time-delayed and noisy versions of the same signal. This observation has already been used for time-synchronization of wireless sensors in smart structures [70]. Using vibration signals for spatial synchronization, *i.e.*, localization, is a natural extension.

4.6.2 Localization Algorithm

This section describes how manifold learning algorithms are applied to estimate sensor location.

Sensor Data Measurement

Initially, each sensor $i = 1 \dots N$ records data $v_i(t)$, for time $t = 1 \dots \tau$. Let the vector $\mathbf{v}_i = [v_i(1), \dots, v_i(\tau)]$ be the data of sensor i. Then, after time τ, each sensor sends its data ($\mathbf{v}_i$) to its immediate neighbors. Define k_i as the number of sensors with which sensor i can directly communicate, and K as the desired number

of neighbors. If $k_i < K$, sensor i queries sensors that are one or more hops away from itself. If $k_i \geq K$, sensor i only receives data from those k_i sensors. Sensor i then calculates Euclidean distance in $\mathbb{R}^\tau$ between its own data and its neighboring sensors' data. Let these data distances be denoted $\{\delta_{i,j}\}_j$, where

$$\delta_{i,j} = \|\mathbf{v}_i - \mathbf{v}_j\|. \tag{4.27}$$

There are some specific items to note:

- Previously in this thesis, $\delta_{i,j}$ was used to denote *physical* distance measurement, and here it is used to denote distance between *data* measured at different sensors.
- Equation (4.27) uses Euclidean distance, but other applications may require and use other distance measures.
- If multi-modal sensors are used, this work might be extended by allowing $v_i(t)$ to be a vector measurement.

Neighbor Selection

Next, using the distances $\{\delta_{i,j}\}_j$, sensor i defines its neighbor list. In this section, the following neighbor selection methods are used: the LLE algorithm uses the K-nearest neighbors (KNN) selection method; the Isomap algorithm, which requires neighbor symmetry, uses the symmetric KNN (SKNN) neighbor selection method, and the HLLE, which does not require symmetry but is sensitive to sparse neighbor graphs, uses the take-pity KNN (TPKNN) neighbor selection method (with $L_{min} = 3$). These neighbor selection methods were defined in Section 4.3.1. As will be described in more detail when presenting the simulation results, the HLLE method does not work when there exists a sensor in the network which is not considered to be a neighbor of any other sensor. The TPKNN method is used to avoid such a situation.

Dimension Reduction

Following the neighbor selection, a manifold learning algorithm is applied (Isomap, LLE, or HLLE algorithm) to reduce the dimension of the sensor data $\{\mathbf{v}_i\}$. Detailed descriptions of these algorithms are given in [118, 104, 34], and these algorithms are reviewed in Section 4.3. These algorithms output a map with a mean coordinate of zero ($\sum_i \mathbf{z}_i = \mathbf{0}$) and possibly a non-physical scale, for example, $\sum_i \|\mathbf{z}_i\|^2 = 1$. The coordinate outputs also have some arbitrary rotation. Thus they serve only as a relative map of the sensors in the network.

Matching Prior Knowledge

Once the manifold learning algorithms output a relative map of sensor coordinates, the *a priori* known coordinates are used to find a rotation, scaling, and possible mirroring so that the reference coordinate estimates match their known coordinates in a least-squares sense. Since this final optimization involves only $m \ll n$ device coordinates, its calculation is much less complex than the original manifold learning on a single device. The outputs are a translation and $D \times D$ transformation matrix, which is used to produce the final coordinate estimates $\{\hat{\mathbf{z}}_i\}_{i=1}^n$. For the reference devices, *i.e.*, $i = n+1 \ldots N$, which are assumed to have perfect prior coordinates, their *a priori* coordinates are used rather than their estimated coordinates.

4.6.3 Simulation Model

In order to run simulations to test the above algorithms, a model is required for the time-varying, isotropic spatially correlated field. Unlike the RF measurements conducted and modelled in Chapter II, the author has not made or found reported in-depth sensor network measurements of the space-time field. Part of the difficulty is that each modality, eg., temperature, soil moisture, acoustic, or image sensors

would likely require a different space-time model.

Instead, for the purposes of simulations presented in this thesis, an isotropic multivariate Gaussian model is employed. This model is used in the precision agriculture literature [31, 74] to model soil chemistry parameters and soil moisture levels. While it is clear that many field measurements will have more complicated behavior, this model provides us with a means to show some analytical and simulation results and to explore the system performance as a function of a few parameters. In particular, the citations [31, 74] make the isotropic assumption, *i.e.*, that the covariance between two sensors' data is only a function of the distance between them. This would not hold though, if, for example, there was some angular dependence, such as a consistent wind direction which caused higher correlation in a particular angular direction. However, isotropic models form the building blocks for more sophisticated non-isotropic or non-stationary models [48].

In the model used in this chapter, at each time t, sensor i measures data $v_i(t)$. Here, it is assumed that the data measured by sensors $1 \dots N$ at time t, $\mathbf{v}(t) = [v_1(t), \dots, v_N(t)]^T$ are jointly Gaussian with mean $\boldsymbol{\mu}$ independent of the actual coordinates $\{\mathbf{z}_i\}$, and covariance matrix $\mathbf{R}(\boldsymbol{\theta})$,

$$[\mathbf{R}(\boldsymbol{\theta})]_{i,j} = \sigma^2 \varphi(\|\mathbf{z}_i - \mathbf{z}_j\|), \tag{4.28}$$

where $\varphi(d)$ is a normalized isotropic covariance function. Further, it is assumed that this covariance function is non-negative, thus $\varphi : [0, \infty) \to [0, 1]$. In the statistical literature, a popular model for φ is the powered exponential class [48],

$$\varphi(d) = \exp\left(-(d/\delta)^\alpha\right), \tag{4.29}$$

where $0 < \alpha \leq 2$ and $\delta > 0$ are constants.

It is assumed, for simulation purposes, that sensor data recorded at different time instants $\{\mathbf{w}(t)\}_{t=1}^{\tau}$ are i.i.d. In reality, sensor data is likely to be correlated in time, which would reduce the effective duration of the time sample.

4.6.4 Cramér-Rao Bound Formulation

Given the above space-time measurement model, a Cramér-Rao lower bound can be derived for the variance of any unbiased location estimator in a network of arbitrary geometry. This will be useful to judge the performance of the introduced location estimators.

The Fisher information matrix $\mathbf{F}$, when measurements follow the above multivariate Gaussian model, is [62],

$$[\mathbf{F}(\boldsymbol{\theta})]_{k,l} = \frac{\tau}{2}\mathrm{tr}\left[\mathbf{R}^{-1}(\boldsymbol{\theta})\frac{\partial \mathbf{R}(\boldsymbol{\theta})}{\partial \theta_k}\mathbf{R}^{-1}(\boldsymbol{\theta})\frac{\partial \mathbf{R}(\boldsymbol{\theta})}{\partial \theta_l}\right], \tag{4.30}$$

where $k, l \in \{1, \ldots, n\}$. Analytical simplification of (4.30) isn't generally feasible, and $\mathbf{F}(\boldsymbol{\theta})$ must be calculated. Let $\hat{x}_i$ and $\hat{y}_i$ be unbiased estimators of x_i and y_i. The trace of the covariance of the i^{th} location estimate (for $i = 1 \ldots n$) satisfies

$$\begin{aligned} \sigma_i^2 &\triangleq \mathrm{tr}\left\{\mathrm{cov}_{\boldsymbol{\theta}}(\hat{x}_i, \hat{y}_i)\right\} = \mathrm{Var}_{\boldsymbol{\theta}}(\hat{x}_i) + \mathrm{Var}_{\boldsymbol{\theta}}(\hat{y}_i) \\ &\geq \left[\mathbf{F}^{-1}(\boldsymbol{\theta})\right]_{i,i} + \left[\mathbf{F}^{-1}(\boldsymbol{\theta})\right]_{n+i,n+i} \end{aligned} \tag{4.31}$$

Note that the CRB is not a function of σ^2, and is inversely proportional to the length of the sampling interval τ.

4.6.5 Simulation Examples

In particular, for the simulations, the powered exponential covariance function (4.29) is used with $\delta = 1$ and $\alpha = 1$, and the time-duration is set to $\tau = 100$. For each test described below, a particular geometry of devices is fixed and 100 trials are run, from which the means and one-σ uncertainty ellipses of the 45 unknown-location

sensor location estimates are calculated. For the HLLE, the TPKNN neighbor selection method is used with $K = 11$ and $L_{min} = 3$; the LLE uses SKNN with $K = 7$, and for Isomap, KNN with $K = 7$ is used. (These values of are chosen by observing simulated performance over a range of K and selecting which values of K worked best for each method.)

Simulations are first run on the 7 by 7 grid example shown in Fig. 3.7. Four sensors in the corners are reference devices (coordinates are known exactly), and 45 additional sensors are unknown-location. The performance of the LLE, HLLE, and Isomap algorithms are compared to the CRB in Figs. 4.17(a),(b), and (c). While none of the estimators have variance approaching the CRB, the LLE and HLLE estimators have low variance compared to Isomap. While the HLLE is nearly unbiased, the LLE and Isomap estimates have high bias.

Next, the performance of the HLLE algorithm is tested when the sensor positions are perturbed from the grid locations. If $\boldsymbol{\theta}_1$ is the vector of unknown coordinates used above in the 7 by 7 grid example, then $\boldsymbol{\theta}_2 = \boldsymbol{\theta}_1 + \mathbf{e}$ is used, where $\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \sigma_e^2 \mathbf{I})$. Here, $\sigma_e = 1/18$, *i.e.*, 1/3 of the distance between devices in $\boldsymbol{\theta}_1$. The results for a particular deployment is shown in Fig. 4.17(d). There are sensors which have significant bias, and overall, estimator variances have increased. In particular, the HLLE tends to push apart the estimates of very close neighbors.

Finally, a random deployment is tested, *i.e.*, the case when $\mathbf{z}_i$ are uniformly distributed in $[0, 1]^2$ for all i. This random deployment is much more dispersive than the perturbed grid, since there is no guarantee of average device density throughout the square area. In Fig. 4.17(e) and (f), the performance of the Isomap and HLLE algorithms are shown for a particular realization of the uniformly random sensor deployment. Both estimators show bias and similar variances, but some of the Isomap

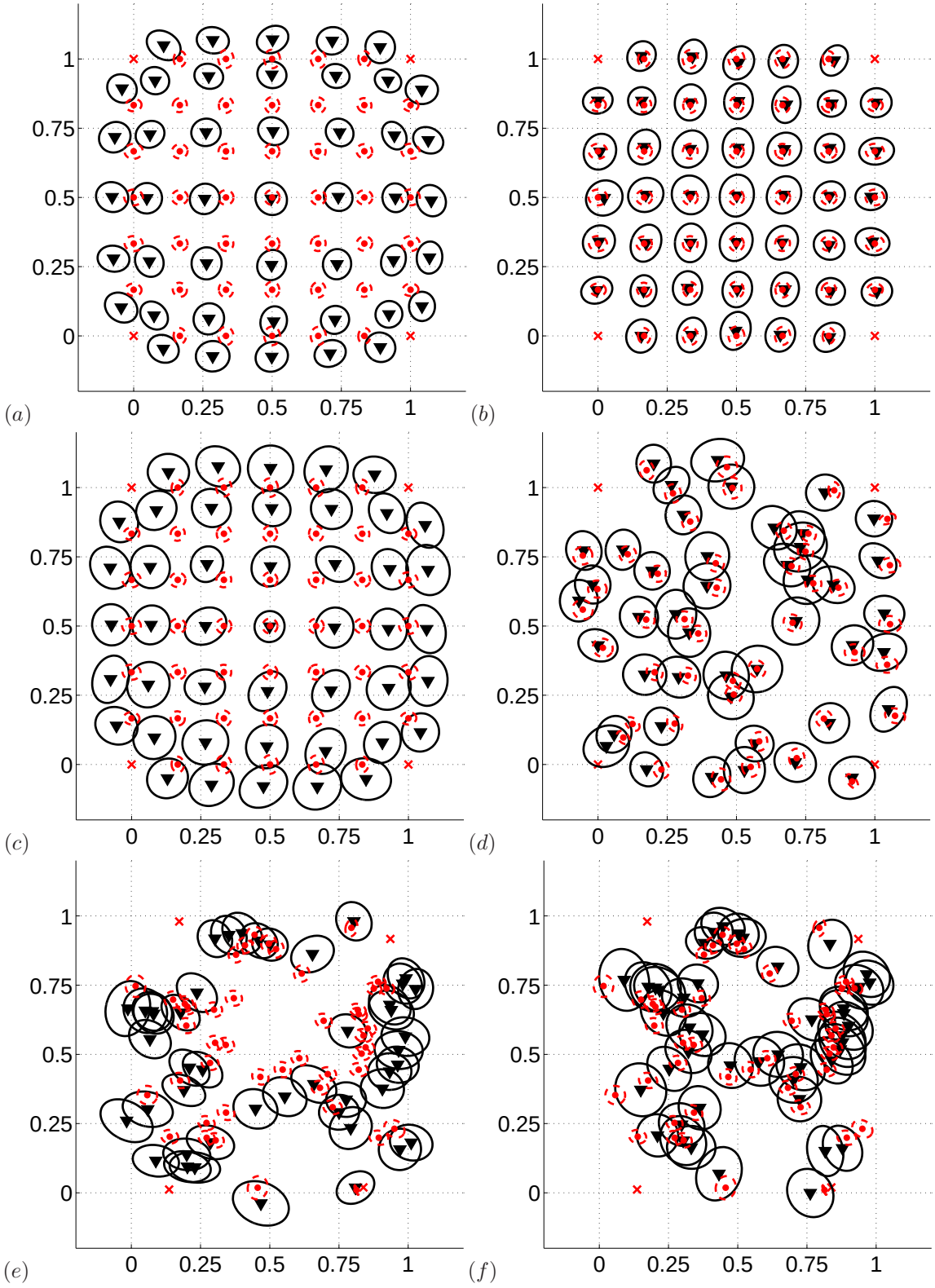


Figure 4.17: Estimator mean (▼) and 1-σ uncertainty ellipse (——-) for each unknown-location sensor compared to the true location (·) and CRB on the 1-σ uncertainty ellipse (- - - -), when the estimator is (a) LLE, (b) HLLE, (c) Isomap, (d) HLLE, (e) Isomap, and (f) HLLE. For HLLE, $K = 10$, and for LLE and Isomap, $K = 7$. One reference device (x) is in each corner, and the unknown-location devices are (a-c) located on a 7 by 7 grid, (d) permuted from the 7 by 7 grid by Gaussian random vectors, or (e-f) chosen from a uniform random distribution on $[0, 1]^2$.

estimates are severely biased and show much higher variance. The HLLE algorithm preserves the general shape of the network.

In the uniform random deployment, if the HLLE algorithm had used the KNN neighbor selection method, then its performance is not robust. This robustness issue was reported in [89] before its cause had been determined. The cause is the non-symmetric behavior of the neighbor relation, and the sensitivity of the HLLE method. As noted in Section 4.6.3, the HLLE fails in the 'lonely' sensor case, *i.e.*, when a sensor has very few devices (none, or just one) which consider it to be a neighbor. For this reason, TPKNN is used with $L_{min} = 3$ throughout the HLLE simulations reported in this section.

4.6.6 Discussion

Manifold learning approaches can be very useful for sensor localization in networks which wish to reduce overhead by using only the correlation in their data as location information. Furthermore, this estimation could be adapted to be performed in a scalable, distributed manner using the LLE and HLLE. While this decentralized version is not implemented in this section, similarity-based methods are particularly adaptable for distributed calculation, as described in Section 4.3.4. Finally, HLLE appears to show the best performance in terms of the tradeoff between estimator bias and variance. Future research must address bias issues in these manifold learning methods in a systematic way. Note that adaptive neighborhood selection methods such as explored in earlier sections have not been attempted in sensor data localization. The manifold learning approach taken here can be extended to provide non-linear interpolation for field estimation. Due to the distinct advantages of manifold learning, it is believed that future research applying these techniques to estimation in wireless sensor networks will be particularly fruitful.

CHAPTER V

Internet Data Localization

This chapter discusses the application of sensor data localization to visualize traffic data collected on routers across the Abilene backbone network, as reported in [91]. The goal of this data visualization is to enable network administrators to view very high-dimensional traffic data on a 2-D image in order to readily see any traffic anomalies that might be occurring, or to identify and classify anomalies that have occurred in past recorded traffic data sets. Traffic anomalies include port or network scans, denial of service (DoS) attacks, flash crowds, outages or network mis-configuration, and worm activity.

The assumption made in this chapter is that when traffic anomalies or intrusion attempts occur across the network, the distribution of network traffic across features changes. Monitoring the network for changes over features such as time, space (at various routers in the network), source and destination ports, IP addresses, and AS numbers, is an important part of anomaly detection. This chapter presents a manifold learning-based tool for the visualization of large sets of data which emphasizes the unusually small or large correlations that exist within the data set. This tool and other command-line tools have been implemented in C by the author, and are referred to as 'Map-tools'. In this chapter, Map-tools are employed to display

anomalous traffic recorded by NetFlow on the Abilene backbone network in January 2005. Furthermore, we present an online Java-based GUI which allows interactive demonstration of the use of the visualization method.

While this chapter differs from the previous ones in that it deals with a *wired* sensor network rather than a wireless one, it is analogous to a sensor network localization problem which deals with the estimation of 2-D coordinates, the same problem statement posed in Section 1.3 except that the 2-D domain is no longer physical space. The problem is especially similar to the one considered in Section 4.6, in which sensors measure data from a spatially correlated random field. The difference is, rather that estimating actual, physical sensor coordinates, we will estimate 2-D data coordinates. These 2-D data coordinates won't have a particular physical meaning, but if they change dramatically over time, it can indicate a traffic anomaly.

5.1 Introduction

Statistical intrusion and anomaly detection methods allow networks to be monitored for attacks for which attack signatures have not yet been developed. However, the huge quantity and high-dimensionality of internet traffic data are significant challenges which research must overcome in order to achieve high reliability and low false-alarm rates. Recently, subspace-based analysis of traffic data by Lakhina, Crovella, and Diot [67, 66] has shown that high-dimensional Abilene traffic measurements can be well-represented within a very low-dimensional subspace. The 'curse-of-dimensionality' can be avoided when high-dimensional data can be represented well in a low-dimensional subspace.

Sketching is a dimensionality reduction method which projects data onto random linear lower dimensional subspaces in a way that preserves inter-data distances, approximately, with high probability [46]. Sketching has been used to dramatically reduce the number of dimensions necessary to store multi-dimensional histograms [119]. Since the visualization presented in this chapter calculates distances between multi-dimensional histograms, sketching could presumably be used to reduce the storage and communication complexity of a distributed implementation of the proposed method. Sketching is not tested in the implementation presented in this thesis, but it would be a very valuable future addition.

In this chapter the distributed weighted MDS (dwMDS) method, presented in Section 4.4, is used to take very high-dimensional NetFlow traffic measurements from the Abilene backbone network and reduce their dimensionality to two dimensions. The resulting 2-D 'map' of the measurements provides a means for visualization of the relationships which exist in a set of traffic data. These relationships may be spatial, *eg.*, between measurements taken across a backbone network or between IP addresses, autonomous systems (AS), or origin-destination (OD)-flows; temporal, *eg.*, measurements taken at different times; or between different applications, as indicated by port numbers.

Such visualization is complementary to detection methods which rely on dimensionality reduction. Subspace-based detection [66] has been successfully used infer the presence of spatial traffic distribution anomalies in network-wide traffic measurements. This inference is done by quantifying the amount of traffic which *cannot* be represented within a low-dimensional subspace. The method we present in this chapter allows the visualization of the traffic which *can* be represented within a low-dimensional subspace. Furthermore, this work uses a non-linear dimensionality

reduction method rather than a linear method.

5.1.1 Visualization

In order to facilitate human moderation, the discussion in this chapter is limited to 2-D representation of the data. Data visualization will complement statistical detection methods by helping provide information to help a human moderator make a decision regarding whether or not an anomaly has occurred, and if so, to determine its temporal and spatial characteristics. To simplify the use of the visualization tool by a human moderator, 2-D display is preferred.

Other visualization methods have found use in network monitoring. For example, visualization of flows by application over time is commonly done using FlowScan [97]. Monitoring the number of flows over time using FlowScan is an excellent tool to identify DoS attacks. However, there is an increasing a number of ports at which attacks are possible. As attacks (such as the Slammer worm) exploit smaller user populations, even obscure services' traffic must be monitored. Dimensionality reduction is a means to monitor, separately, hundreds or thousands of traffic statistics but to minimize the complexity of the information display.

Furthermore, graph visualization provides information regarding the physical connections that exist in a network. Visualizations of the global internet, such as CAIDA's Skitter plot [24], are important statements about the interconnectivity of the global network. Other tools developed at CAIDA, such as Otter and Walrus, provide 2-D and 3-D visualization of network graphs. The visualization method presented in this chapter provides information not just about the connections that exist, but also the traffic correlations that exist. Connection distances match correlation - when correlation between two nodes is high, they are plotted close together, but when correlation is low, nodes are drawn further apart.

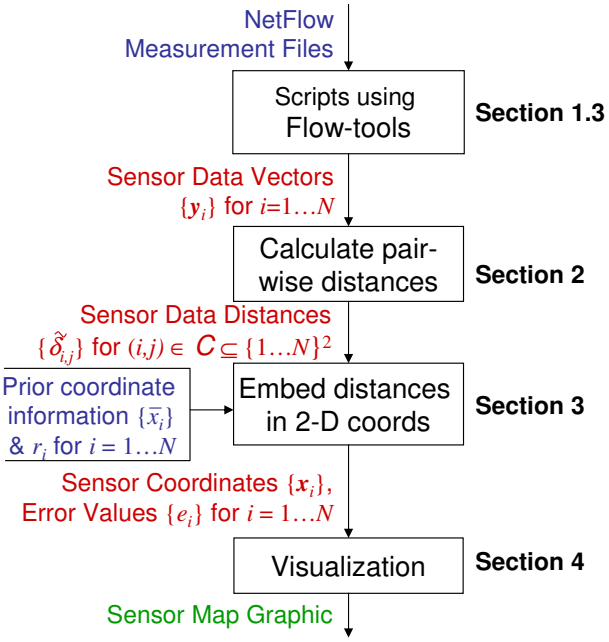


Figure 5.1: Flow chart of data visualization from NetFlow data input to data map output.

5.1.2 Framework

The dimensionality reduction problem is framed as a sensor data localization problem, very similar to the framework of Section 4.6. In this framework, 'sensors' are the hardware or software which record data, for example, on each router in a backbone network. The traffic data which they record can be of arbitrarily high dimension. For example, rather than counting the grand total number of flows (just one dimension), sensors could count the total number of flows from each source IP address (up to 2^{32} dimensions). The key to understanding a particular sensor data map is to know "Where are the sensors?", "What traffic statistic is recorded?", and "What are the dimensions?":

1. *Where are the sensors?*: Sensors can be 'located' at physical computers, *i.e.*, at backbone routers, or at IP addresses; or they can be 'located' at less physical concepts such as source or destination ports, or particular time periods. A

sensor attached to a particular source port monitors only traffic which matches its source port, and a sensor attached to a time monitors only traffic which arrives during that time period.

2. *What traffic statistic is recorded?*: Sensors might measure flows, packets, or octets, or some combination of the three.

3. *What are the dimensions?*: Sensors can divide traffic by source or destination IP address, port, or AS; time period; link or router; or some combination of them. Traffic statistics are then recorded for each dimension (port, IP address, AS, time period, link or router) separately.

For example, in Section 5.3, sensors are located at backbone routers, recording the number of flows from each source IP address. As another example, in Section 5.4, sensors are located on backbone routers, recording the total number of packets received in each of the past T 5-minute time intervals.

Note that this framework can be used to describe the measurements in [67], in which sensors were located at all 10-minute time intervals over the course of a week, and sensors measured total octets on each link across the (Sprint-Europe or Abilene) backbone network.

Denote the data measured at sensor i as $\mathbf{v}_i$, where $i \in \{1, \ldots, N\}$, where N is the total number of sensors. The high-dimensional vector $\mathbf{v}_i$ is then defined by,

$$\mathbf{v}_i = [v_i(l_1), v_i(l_2), \ldots, v_i(l_{|\mathcal{L}|})], \tag{5.1}$$

where $\mathcal{L}$ is the set of possible dimensions (see #3 above) with $|\mathcal{L}|$ elements, and $l_k \in \mathcal{L}$ for all $l = 1 \ldots |\mathcal{L}|$. In many applications, $v_i(l_k) = 0$ for most of its elements l_k, thus $\mathbf{v}_i$ is best stored as a sparse vector. For example, the set of possible IP

addresses is much larger than the set of IP addresses observed in a particular traffic stream.

Map-tools uses the flow-tools package created by Mark Fullmer [42] to process NetFlow files in order to generate the data for the vectors $\{\mathbf{v}_i\}$. A portion of an example data set is shown in Table 5.1.

ATLA	CHIN	DNVR
130.14.24.0, 1545	129.25.0.0, 13913	129.25.0.0 14331
131.247.224.0, 1487	141.89.48.0, 8738	207.46.104.0 12142
128.61.64.0, 1197	207.46.104.0, 3708	207.46.248.0 7198
198.32.152.0, 1147	204.179.120.0, 3520	207.68.176.0 4968
164.111.192.0, 1139	203.250.224.0, 3441	64.4.16.0 4156
131.247.232.0, 1098	207.46.248.0, 3300	207.68.168.0 3707
⋮	⋮	⋮

Table 5.1: Example data: Top few lines of $\{\mathbf{v}_i\}$ for 3 Abilene routers, flows by source IP (last 11 bits zeroed) for 5 minutes ending 20 Jan 2005 01:00 UTD.

5.2 Algorithm and Implementation

After sensors record $\{\mathbf{v}_i\}_i$ as described in Section 5.1.2, the next step is to calculate distances between the data vectors. Often, two sensors will record different levels of total traffic even though their traffic is very correlated. In this chapter, unless otherwise noted, we normalize each data vector such that its sum is one:

$$\tilde{\mathbf{v}}_i = \frac{\mathbf{v}_i}{\|\mathbf{v}_i\|_1} \tag{5.2}$$

where $\|\mathbf{v}_i\|_1$ is the L_1 norm, *i.e.*, the total traffic measured at sensor i. The value $\tilde{\mathbf{v}}_i(l)$ thus is the fraction of traffic measured in dimension l. Although this normalization is arbitrary (another normalization could have been used), the normalization by L_1 norm is intuitively desirable since $\tilde{\mathbf{v}}_i$ represents the distribution or histogram of traffic across dimension.

5.2.1 Calculation of Distances

Let $\tilde{\delta}_{i,j}$ be the distance between the measurement vectors from sensors i and j. An arbitrary norm can be used; in this chapter, the Euclidean distance between $\tilde{\mathbf{v}}_i$ and $\tilde{\mathbf{v}}_j$ is used,

$$\tilde{\delta}_{i,j} = \Big[\sum_{l \in |\mathcal{L}|} \left(\tilde{\mathbf{v}}_i(l) - \tilde{\mathbf{v}}_j(l) \right)^2 \Big]^{\frac{1}{2}} \tag{5.3}$$

Then, define $D_{\mathbf{v}} = \sum_{(i,j)\in\mathcal{C}} \tilde{\delta}_{i,j}$, *i.e.*, the sum of all of the distances between neighbors. To ensure that different sensor maps are approximately the same size, we also normalize $\tilde{\delta}_{i,j}$ in order to achieve a desired constant sum, $D_{\mathbf{z}}$, for some $D_{\mathbf{z}} \in \mathbb{R}$. For each $(i,j) \in \mathcal{C}$, we define

$$\delta_{i,j} = \tilde{\delta}_{i,j} \frac{D_{\mathbf{z}}}{D_{\mathbf{v}}} \tag{5.4}$$

5.2.2 Neighbor Selection

Using the set of normalized distances $\{\delta_{i,j}\}$, we next determine the sensor neighbor relation. Intuitively, pairs of sensors which are 'close' to each other will consider each other to be neighbors. The symmetric K-nearest neighbors (SKNN) method is used to determine the neighbor set, since symmetry is required by the dwMDS.

5.2.3 Prior Coordinate Information

Prior coordinate information is an option in the dwMDS method. If there is no prior information for sensor i, we set $r_i = 0$. If all sensors have $r_i = 0$, the calculated output map can arbitrarily be translated, rotated, and flipped without affecting its cost S. The purpose of the map is to show the relationships between sensors' data, and as such, translation, rotation, and flipping do not change the meaning of the map. However, if the user will view many maps, for example, in sequence over time, it would be confusing if each subsequent map was nearly identical but with arbitrary

rotation. In this chapter, prior coordinate information is used purely to provide a stable orientation for a sensor map. This is accomplished by setting $r_i = 10^{-3}$ for all i, a very low value for the prior coordinate weight. As described in the simulations in Section 4.4.6, very low r results in the dwMDS algorithm optimizing the relative map of coordinates purely to match the measured distances, and a rotation and translation to make the relative map match the prior coordinates.

As a set of prior coordinates, when the sensors are on Abilene backbone routers as described in Section 5.3, the router coordinates plotted in Fig. 5.2(a).

5.2.4 Coordinate Embedding

Using the calculated distances $\{\delta_{i,j}\}$, for pairs (i, j) which are neighbors, the coordinate embedding is calculated in two dimensions using the dwMDS method described in Section 4.4. Without repeating the details, this means that two-dimensional coordinates $\{\mathbf{z}_i\}_{i=1\ldots N}$ are found which best represent the calculated distances and the prior coordinate information in a weighted-least-squares sense.

Map-tools uses a centralized implementation of the dwMDS algorithm, in which the data is collected and optimized at a single processor [85]. This is used to demonstrate the visualization method's capabilities, but a distributed implementation is an important topic for future research.

5.2.5 Error Metric

The squared error value e_i^2 for sensors $i \in \{1, \ldots N\}$ is defined as,

$$e_i^2 = \sum_{i,j} w_{i,j} \left(\delta_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\|\right)^2, \tag{5.5}$$

where $w_{i,j}$ is the neighbor weight used in the dwMDS algorithm, and is given in (4.24). The value of e_i helps quantify how much information was lost in the low dimensional representation of the sensor coordinates. It represents the quantity of

measurement distances $\{\delta_{i,j}\}_j$ which are not represented by the 2-D coordinates of itself, $\mathbf{z}_i$, and its neighbors j, $\{\mathbf{z}_j\}$. The value e_i^2 is analogous to the residual value in PCA and can be used to help decide whether or not the sensor's data is anomalous. In the case studies, we 'color' each sensor as a function of e_i: if e_i is low, we shade the sensor light gray, and if e_i is high, we shade the sensor dark gray. Examples are presented in the following section.

Note that the dwMDS method will output coordinates in an arbitrary dimension. Thus, although we limit ourselves to 2-D for the purposes of this thesis, 3-D visualization is equally possible with the presented method.

5.3 Router Map Case Studies

In this section several examples of Map-tools are presented which visualize backbone traffic on Abilene. The examples use NetFlow data recorded during January 2005, downloaded from the Abilene Observatory [59]. Note that, for privacy reasons, only the most significant 21 bits of IP addresses are available - the last 11 bits are zeroed out. Furthermore, NetFlow data is sampled at 1/100, so for each packet reported here, there were 99 more unrecorded.

When sensors are attached to routers in a backbone network, a 'router map' shows the spatial characteristics and correlations of the routers' traffic, rather than just the connectivity of the routers. In the following examples, sensors are routers, and we measure flows in a 5-minute period, separated by source IP address. For the number of neighbors, $K = 5$ is chosen, and the size of each sparse data vector is limited to 1000 – flows not from the top 1000 source IP addresses (/21) for a particular router are ignored. For this reason, source IP addresses are typically lost if they have less than 10-50 flows. Note that such thresholding will likely miss low-level distributed

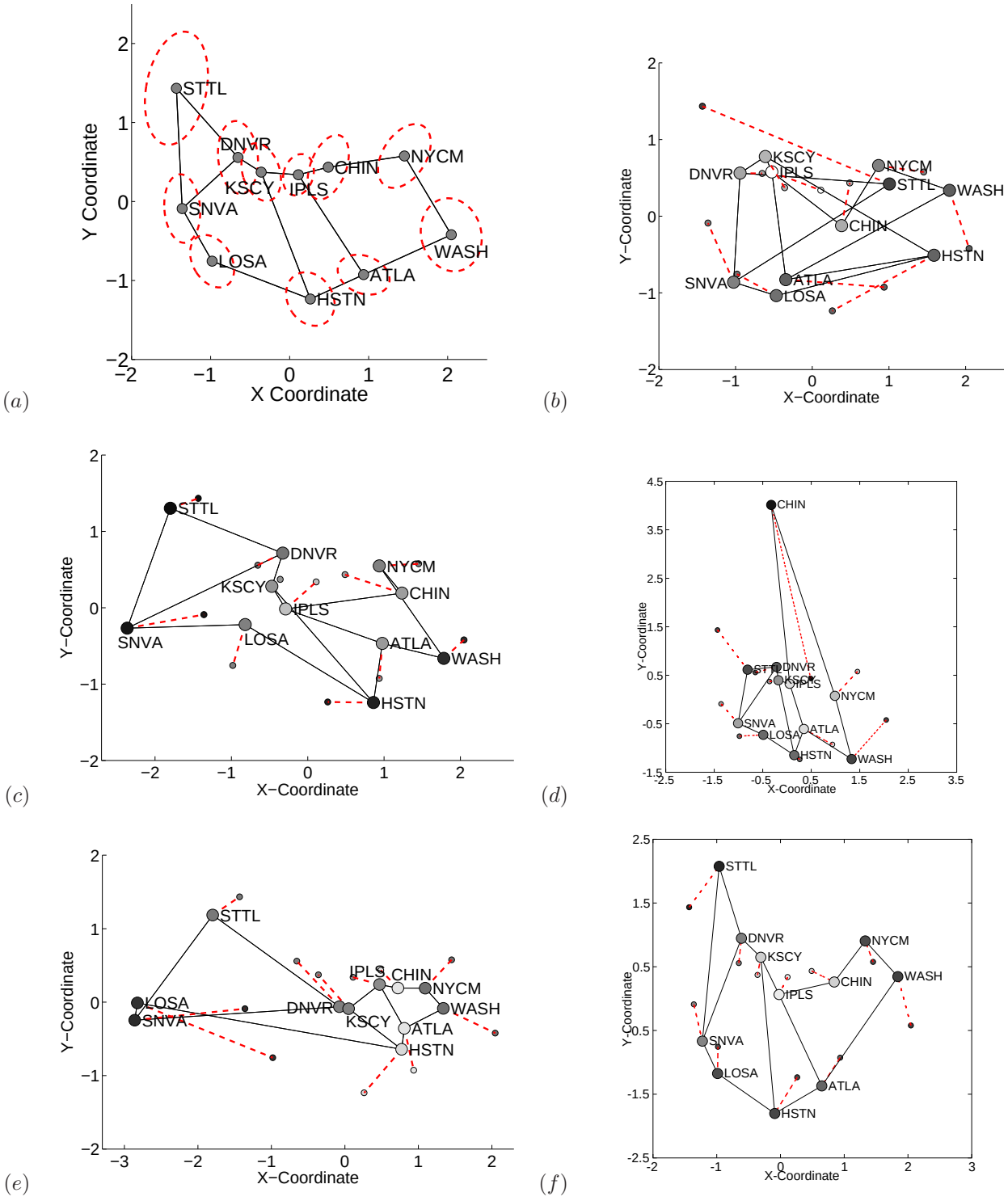


Figure 5.2: (a) Mean (●) and 1-σ uncertainty ellipse (- - -) of router maps from 2-Jan to 29-Jan. Figs. (b)-(f) show router coordinates (●) connected (- - -) to the mean (•) from (a), and shaded by error value e_i, during five anomalous events: (b) worm activity on 02-Jan 8:10, (c) outage on CHIN-IPLS link on 5-Jan 8:05, (d) network scan on 6-Jan 17:55, (e) port scan on 12-Jan 20:15, and (f) port scan on 20-Jan 01:00. All times are UTD, and figures show Abilene backbone links (—).

denial-of-service (DoS) attacks, which use a few flows each from many source IP addresses. Using a much higher number of flows will be critical for detection of widely distributed DoS attacks, and is left for future research.

5.3.1 Typical Activity

First, we attempt to characterize 'typical' router map behavior using the router maps for the four-week period 02-Jan to 29-Jan. Since a router map can be calculated for each 5 minute period (12 per hour), there are a total of $12*24*7*4 = 8064$ maps. We calculate the 8064 maps, and then calculate the sample mean and covariance of $\mathbf{z}_i$ for each router i, and show the results in Fig. 5.2(a). Although there are certainly attacks active during this 4-week period, averaging maps over a long period of time can provide intuition about what is typical behavior for the router map.

The typical router map in Fig. 5.2(a) both makes sense geographically and describes typical traffic patterns seen on Abilene. Much of Abilene traffic is East-West or West-East, and Northern routers (especially DNVR, KSCY, IPLS, and CHIN) bear much of this traffic. These routers have very correlated sensor data because a significant proportion of Abilene OD-flows pass through all of them.

While Fig. 5.2(a) shows an 'average' router map, there are some times when the map differs dramatically from the mean. In the following, five specific examples of such router maps are shown, and a description is given of the traffic in the network at that particular time.

5.3.2 Sunday, 02-Jan-2005 at 8:10 UTD

Fig. 5.2(d) shows the 02-Jan 8:10 router map during which there are about 1.3×10^4 flows going between two IP addresses: 129.171.184.0 (University of Miami, FL) and 64.4.16.0 (`hotmail.com` or `msn.com`). There are about 6000 flows originating from

the U. Miami address from a wide variety of source ports to destination port 80 (TCP) of the Microsoft address. Each flow contains 1-6 (for an average of 2) 40-byte packets. The hotmail.com address replies with 1500-byte packets, from source port 80 to a wide range of destination ports. While there are normally many flows from the Microsoft address, this traffic accounts for about 80% of the total flows coming from that source address. This traffic is suspected to be a result of worm activity, but more investigation is warranted. The map shows the source and destination, ATLA and STTL, being mapped very far from their mean location. Routers DNVR, KSCY, and IPLS are also affected by the anomalous traffic, and are grouped very close together. HSTN traffic is usally very similar to ATLA, but at this time it is very different, and so HSTN is placed very far away.

5.3.3 Wednesday, 5-Jan at 8:55 UTD

At this time, there is scheduled maintenance on the CHIN-IPLS link. Usually, IPLS and CHIN traffic are very similar, but during the downtime, much of the traffic on Abilene re-routes through different links, such as a more Southern route through WASH and ATLA. As a result, the router map in Fig. 5.2(c) shows a much larger distance between IPLS and CHIN, and a much flatter map, since traffic on the Southern routers are, temporarily, very correlated with Northern traffic.

5.3.4 Thursday, 6-Jan at 17:55 UTD

On Thursday, 6-Jan-2005, during the 5-minute period ending at 17:55 UTD, NetFlow data recorded on the CHIN router count a total of 9×10^4 single-packet flows (of 40-byte packets) from two source IP addresses in Taiwan to a small range of destination IP addresses in Hungary. This volume corresponds to about 25% of the typical flow volume on CHIN. The traffic from the two Taiwanese source IP addresses was

observed on CHIN and no other router, thus distances between sensor data recorded at CHIN and other routers are unusually high, and the 2-D coordinates for CHIN must be kept very distant from all other sensors. Also, because of the normalization done to calculate $\tilde{\delta}$ from δ, the rest of the map distances have shrunk to compensate. This is shown in Fig. 5.2(d).

5.3.5 Wednesday 12-Jan at 20:15 UTD

Fig. 5.2(e) shows a router map during a large anomaly of 7.1×10^4 flows at the STTL, LOSA, and SNVA routers on 12-Jan at 20:15 UTD. These flows are single, 29-byte UDP packet flows from source IP address 163.30.88.0 (possibly `tyc.edu.tw`) to destination IP address 134.71.24.0 (`csupomona.edu`, California Poly in Pomona). The packets are from source port 40150 to random destination ports. Since the traffic was observed on LOSA, SNVA, and STTL but no other router, these routers are placed far away to the West, while the rest of the routers, due to the constraint on total distances, are placed very close together.

5.3.6 Thursday, 20-Jan at 01:00 UTD

At this time, there are 14,000 29-byte packets from a 129.25.0.0 (Drexel U.) source IP address sent to a 131.252.120.0 (Portland State U.) destination (see Table 5.1). The packets are UDP with source port 3095 or 3096 to a wide range of random destination ports > 1024. These packets travel through the WASH, NYCM, CHIN, IPLS, KSCY, DNVR, and STTL backbone routers. Other routers (SNVA, LOSA, HSTN, and ATLA) do not see any flows from this source address at this time. Distances between the listed Northern routers and the other Southern routers are unusually high. In the router map shown in Fig. 5.2(f), there is a clear split in the map between the two sets of routers.

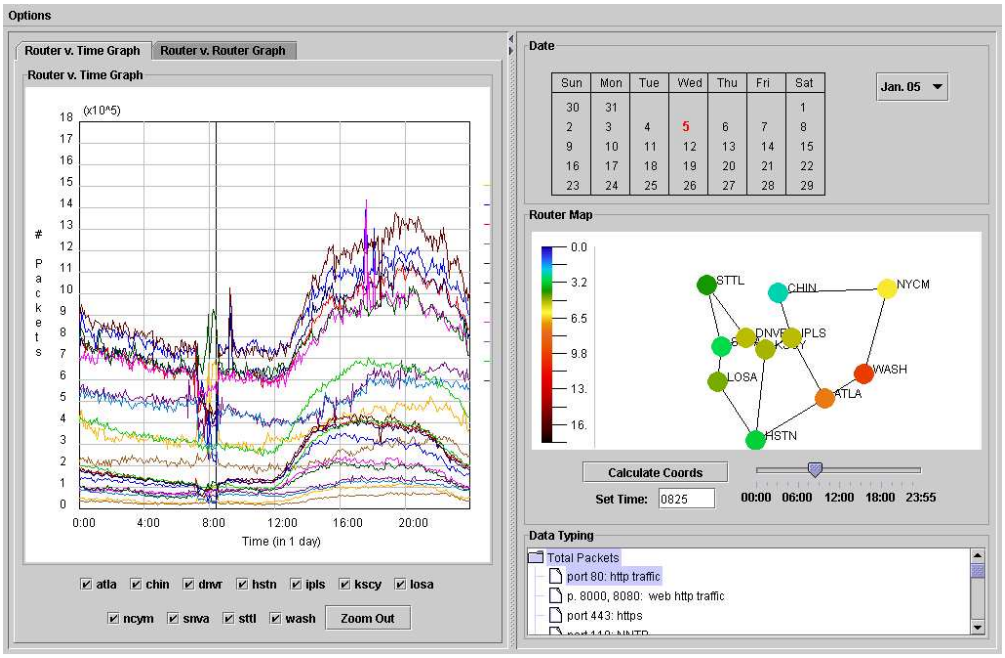


Figure 5.3: Total traffic and port 80 traffic on 05-Jan-2005, displayed using the visualization applet [85]. The router map is calculated for 08:25 UTD, during scheduled maintenance of the CHIN-IPLS link, during which traffic drops at CHIN and IPLS and increases dramatically at the HSTN and ATLA routers.

5.4 Map Web Applet

In collaboration between the author, Panna Felsen, a high school student from San Diego, and Adam Pacholski, a University of Michigan undergraduate student, we have developed a Java-based space-time visualization applet. This applet, publicly available online at [85], displays temporal and spatial traffic on the Abilene backbone, as shown in Fig. 5.3. The applet has a graph of the traffic for each router, and can plot traffic levels from 21 different sets of ports grouped by application.

In addition to displaying the traffic by port and time, the applet calculates a router map for each time. In this router map, the sensors are routers, and they measure total packets, by port and 5-minute time interval. The user can select which groups of port traffic to use, from among a preset list of 21 ports or sets of ports. The

user can simultaneously select the number of time intervals to use as dimensions. Rather than normalizing to calculate $\tilde{\mathbf{v}}_i(l)$ as described by (5.2), this applet uses $\tilde{\mathbf{v}}_i(l) = \mathbf{v}_i(l) - \overline{\mathbf{v}}_i(l)$, where $\overline{\mathbf{v}}_i(l)$ is the median of the past T_m time samples, where T_m is also user-adjustable. Using a filtered traffic stream emphasizes the changes that occur over time. When traffic changes over time, the router map on the applet shows where (which routers) the change is most dramatic. Details are available online [85].

5.5 Detection using Low-Dimensional Data

If there were no bandwidth or confidentiality restrictions on the internet, we could send all data recorded at network routers to a central processor which would then detect spatial and temporal anomalies. However, full data can quickly consume a significant percentage of the bandwidth of the network and cause bottlenecks near the central processor, where bandwidth requirements will be highest. In addition, in the event of a high-traffic anomaly, the bandwidth requirements to monitor the anomaly will increase, actually contributing to the network congestion.

Furthermore, if a central processor did have access to full high dimensional data, it would suffer from the 'curse of dimensionality'. Building a model to describe the behavior of each dimension and the inter-relationships between dimensions would require huge quantities of data as input and huge memory requirements at the central processor if such a model could be calculated. A central processor may need to reduce the data dimension as a first step, even given full data.

These bandwidth and detection issues motivate methods that use only low-dimensional representations of the router traffic data in order to detect traffic anomalies. If a detection method can accept data reduced in dimension by many orders of magnitude and still provide adequate power, then it is likely that it can be successfully

implemented across routers on the internetwork.

This section is meant to show an example traffic anomaly detection using low-dimensional data without fully exploring the power and tradeoffs of such detectors. There are many possible low-dimensional detection methods; see for example research by Lakhina, Crovella, and Diot [67, 66]. In this section, for visualization purposes, it is easiest to produce maps of coordinates in two dimensions. For detection purposes, the low-dimensional representation wouldn't be strictly limited to 2-D or 3-D. Detection performance as a function of quantization can be bounded and approximated analytically to help find an optimal tradeoff between bandwidth and performance [50]. Future research would be very valuable in this regard.

5.5.1 Map Anomaly Detector Formulation

Section 5.2 details the algorithm and implementation for finding 2-D location coordinates from high-dimensional data recorded at each router in a network. Further, router maps during 'normal' and 'anomalous' times were displayed and discussed. This section further formalizes the classification of router maps as either normal or anomalous. No attempt is made to derive a causal test in this section, since the data set (January, 2005) used in this Chapter is fully available. Online detection methods are discussed at the conclusion of this section.

The particular statistical test which is used in this section is known as the *multivariate t-test* or *Hotelling's T^2 test*. Intuitively, the router coordinates at one time are compared to a *library*, *i.e.*, router coordinates calculated at other times. If the current coordinates are far enough from the mean of the library of other coordinates, then the current traffic is labeled anomalous. Specifically, consider the current coordinates $\boldsymbol{\theta}(t)$ and the library of coordinates $\{\boldsymbol{\theta}(s)\}_{s \neq t}$, where $\boldsymbol{\theta} = [x_1, \ldots, x_N, y_1, \ldots, y_N]$ and the time s at which the coordinates were calculated is denoted as (s). We as-

sume that the $\{\boldsymbol{\theta}(s)\}_{s\neq t}$ are independent multi-variate Gaussian, *i.e.*, $\forall s \neq t, \boldsymbol{\theta}(s) \sim \mathcal{N}(\boldsymbol{\mu}, R)$. It is also assumed that the current sample has the same covariance, so $\boldsymbol{\theta}(t) \sim \mathcal{N}(\boldsymbol{\mu}_t, R)$. We wish to test the hypotheses,

$$H_0 : \boldsymbol{\mu} = \boldsymbol{\mu}_t \tag{5.6}$$

$$H_1 : \boldsymbol{\mu} \neq \boldsymbol{\mu}_t$$

Defining the sample mean and covariance to be

$$\hat{\boldsymbol{\mu}} = \frac{1}{\tau}\sum_s \boldsymbol{\theta}(s) \tag{5.7}$$

$$S = \frac{1}{\tau - 1}\sum_s (\boldsymbol{\theta}(s) - \hat{\boldsymbol{\mu}})(\boldsymbol{\theta}(s) - \hat{\boldsymbol{\mu}})^T, \tag{5.8}$$

the Hotelling's T^2 statistic for time t is [127],

$$T^2 = (\boldsymbol{\theta}(t) - \hat{\boldsymbol{\mu}})^T S^{-1} (\boldsymbol{\theta}(t) - \hat{\boldsymbol{\mu}}) \tag{5.9}$$

The statistic T^2 measures the squared distance between the the data at time t and the mean. The distance is a weighted distance, which takes into account the covariance of the data set. When T^2 is higher than a threshold, we decide H_1 and declare an anomaly. Otherwise, the measurement is concluded to be 'normal'.

Note that Hotelling's T^2 test has previously been proposed for intrusion and anomaly detection systems [127]. It is also an approximation for the generalized likelihood ratio test (GLRT) for the hypotheses described by 5.6. The approximation made is that the sample mean $\hat{\boldsymbol{\mu}}$ and covariance S, calculated using measurements from $1 \ldots \tau$, are approximately equal to the sample mean $\hat{\boldsymbol{\mu}}_1$ and covariance S_1 calculated excluding the measurement from time t. Since the measurement at time t has a different mean under H_1, it would be proper to exclude $\boldsymbol{\theta}(t)$ from the mean and covariance calculation. However, this is impractical from a computational

perspective, since it requires calculating τ different covariance matrices and their inverses.

5.5.2 Experimental Testing of the Map Anomaly Detector

The T^2 statistic of (5.9) is calculated for the four-week period starting on 2 Jan extending through the end of 29 Jan. First, router maps are calculated for each 5 minute period during the four weeks of January, using the total flows by source IP address data, as described in Section 5.3. There are a total of $\tau = 8064$ router maps. With 11 routers and 2-D coordinates, the parameter vector $\boldsymbol{\theta}$ is length 22, thus S is a 22 by 22 matrix. All of the calculations have been implemented in C code, and are part of the command line executable 'calcTstat'.

With regards to the threshold, we choose to set the threshold using the measured data to meet a particular number of alarms. In particular, a threshold was chosen to classify 0.4% (or 32 out of 8064) of the router maps as anomalies; this corresponds to a threshold of $\eta = 13.9^2$. Figure 5.4 plots $\sqrt{T^2}$ over the course of the four week period in January, and marks the times of the anomalous maps.

It is typical to evaluate the theoretical rate of false alarm, α, for a threshold η. It is known, given H_0, that $T^2 \sim \frac{2N(\tau-1)}{\tau-2N} F_{(2N,\tau-2N)}$ where $F_{(n_1,n_2)}$ is the F distribution with n_1 and n_2 degrees of freedom. Using this, at a threshold of $\eta = 7.47^2$ the false alarm rate would be $\alpha = 10^{-4}$, essentially making it unlikely that we would see any false alarms in our sample of 8064. The rate α diminishes quickly as a function of η, and at the chosen threshold $\eta = 13.9^2$, α is insignificant, less than 10^{-20}. Given the model, we would not expect to see any false alarms in the sample data set.

As the next step, each detected anomaly event is investigated. Sometimes, several consecutive (or nearly consecutive) router maps are marked as anomalies, and these are denoted here as one 'anomaly event'. A total of 15 anomaly events are detected,

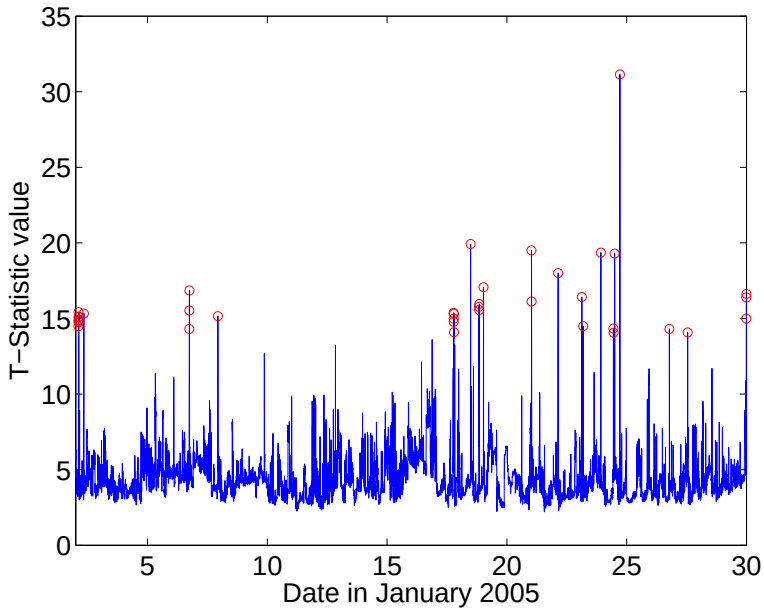


Figure 5.4: The square root of Hotelling's T^2 statistic, $\sqrt{T^2}$ for each 5-minute router map during 02-29 Jan 2005, with those exceeding the threshold $\sqrt{\eta} = 13.9$ marked with ○ as anomalies.

and painstakingly investigated manually by looking at the flows recorded during each event. These are detailed in Table 5.2. All of the router maps detected as anomalous did in fact see significant anomalous traffic at the time for which the router map was calculated. These anomalies involved were measured to be composed of between 13,000 and 130,000 flows, for an average of 58,000 flows. These size figures quoted are the total flows measured by NetFlow within the five minute measurement period - those which lasted longer than 5 minutes would have larger cumulative effect. Furthermore, since NetFlow samples 1/100 packets, and many of these anomalies involved flows with 1 packet per flow, the total number of flows on the network may have been 100 times greater. Thus, the false alarm rate is in fact zero.

5.5.3 Experimental Limitations

While the described detector shows a 0% false alarm rate in the above test, we can say nothing about the probability of miss. Identifying, classifying, and describing

Date	Time(s)	Size (Flows)	Label	Brief Description	Affected Routers
2005-01-02	2:45-3:20	30k-50k	DDOS	Single 29-byte UDP packet flows sent to Univ. in Sweden from two sources (Colorado and Taiwan)	DNVR, KSCY, IPLS, CHIN*
2005-01-02	8:10	13k	Worm activity	See Section 5.3.2	
2005-01-06	17:55-18:00	90k	Network scan	See Section 5.3.4	
2005-01-07	22:30	28k	Multiple	Single packet ICMP, TCP, and UDP flows exchanged between .NCTU.edu.tw and .edu.pl	NYCM, CHIN*
2005-01-17	18:35-18:55	110k-127k	Port scan	Single 40-byte TCP packet flows, scan of low dst ports of dst IP in .edu.pl from .NTU.edu.tw	CHIN
2005-01-18	11:40	50k	Multiple	Attack on mIRC (dst ports 6660-8) also using UDP, ICMP traffic to dst IP in .edu.pl from .NTU.edu.tw	CHIN
2005-01-18	20:05-20:20	42k	Multiple	Attack on mIRC (dst ports 6660-8) also using UDP, ICMP traffic to dst IP in .edu.pl from .NTU.edu.tw	CHIN
2005-01-19	0:35-0:45	22k	Worm activity	Single 60-byte TCP packets from multiple Miss. St. Univ. IPs to range of Microsoft IPs, dst port 113	STTL, DNVR, KSCY, IPLS, ATLA
2005-01-22	3:20	20k-130k	DDOS	Attack on telnet (dst port 23) using 40-byte TCP packets from 6-10 different source IPs	All
2005-01-23	3:05	44k	Port scan	Single 29-byte UDP packet flows from cwru.edu (Ohio) src port 49201 to NTU.edu.sg (Singapore)	STTL, DNVR, KSCY, IPLS
2005-01-23	4:30	45k	Port scan	Single 29-byte UDP packet flows from umaryland.edu src port 47508 to waseda.ac.jp (Japan)	LOSA, HSTN, ATLA, WASH
2005-01-23	22:05	22k	Port scan	Single 28-byte UDP and 40-byte TCP flows from src utoronto.ca to MOREnet (Missouri)	KSCY, IPLS
2005-01-24	12:00	117k	Port scan	Single 29-byte UDP packet flows from asu.edu src port 32769 to .ac.at (Austria)	DNVR, KSCY, IPLS, CHIN
2005-01-24	17:10	59k	Port scan	Single 40-byte TCP packet flows, random src & dst ports, from utoronto.ca to unm.edu (New Mexico)	STTL, DNVR, KSCY, IPLS
2005-01-29	23:45-23:55	40k	Port scan	Single 29-byte UDP packet flows from asu.edu src port 36203 to yonsei.ac.kr	SNVA, LOSA

Table 5.2: Dates and times of router maps detected as anomalous and the size (in total flows measured per five minutes), type of anomaly, brief description of the traffic, and the routers through which the anomalous traffic traveled. Description and classification are from the author's manual inspection; no verified log of the true anomalies is available.

anomalous traffic from manual inspection is a tedious and time-consuming process. Investigating all time instants (8064) for the measurement period would be a huge task. Clearly, though, at the given threshold level, some attacks are missed. For example, the link maintenance event discussed in Section 5.3.3 was not detected; neither were the port scans discussed in Sections 5.3.5 and 5.3.6. The T^2 value for these three events were 3.97^2, 13.23^2, and 5.02^2, respectively. The event in Section 5.3.5 was a large volume event and was just below the threshold of $\eta = 13.9^2$. Using a slightly lower threshold would have allowed detection of this event. However, the link maintenance event of Section 5.3.3 and a smaller size anomaly of Section 5.3.3 would not likely have been detected using the method described here.

Note that there are *almost always* small levels of worm, denial of service, port, and network scan activity taking place on the network. It is difficult to specify an exact level of anomalous traffic, below which a detector should not declare an anomaly, and above which it should. Without such a anomalous traffic level threshold, though, the detector will *almost never* see a false alarm.

One noticeable feature of Figure 5.4 is the daily periodicity of the T^2 statistic. Clearly, daily fluctuations in traffic distribution are contributing to the 'anomaly' value. This implies that further, more flexible, or more periodic means should be used to determine the mean and covariance of traffic. One obvious method would be to use an recursive least-squares (RLS) approach, which weights the most recent past more heavily than the distant past when characterizing statistical behavior.

A means to analyze the above experiment would be to compare it to other standard methods for traffic anomaly detection. Implementation of other detection algorithms and comparison would also be a valuable topic for future research.

5.6 Future Work and Conclusion

This chapter has introduced a visualization tool which can aid in the discovery of malicious activity and other traffic anomalies in high-dimensional NetFlow data. It is clear that manifold learning methods, and sensor data localization, can be used to provide information about relationships that exist in sets of sensor network data. Specifically, the router maps calculated on Abilene traffic data did show changes when anomalous traffic was present on the network. These changes were statistically significant in many cases and can be detected by looking at the (low-dimensional) coordinates rather than the high-dimensional traffic directly.

Future work will certainly attempt to further automate the detection and classification process, and compare the results to existing methods, in combination with extensive manual analysis of detected anomalies, as described in Section 5.5. Note that traffic anomaly detection methods which use full data, *i.e.*, [37], can outperform detection based on low-dimensional coordinate representation. Low-dimensional, or quantized versions of full data, will naturally degrade detector performance [50]. Thus comparisons should compare methods with equivalent bandwidth requirements.

Investigation is planned of other distance metrics which may better emphasize similarities in traffic distributions besides the L_2 norm. Other ML methods such as Isomap [118] and Laplacian Eigenmap [8] should also be tested. It is hoped that router maps, along with maps produced when other 'sensors' are used (such as port maps [91]), and other types of sensor maps may together serve as a step-by-step investigation aid, iteratively helping to locate a traffic anomaly in a very high-dimensional spatial and temporal data space.

CHAPTER VI

Conclusion

This thesis has explored topics in location estimation in sensor networks. Two types of sensor networks have been considered, wireless and wired. Also, two types of localization have been introduced: localization from pair-wise measurements, and sensor data localization. The former type of localization tries to estimate physical sensor location, while the latter type is a data visualization method, in which large quantities of data are summarized by low-dimensional coordinates. Both types of localization have been explored in wireless sensor networks. In wired networks, where the physical sensor locations are presumed known, the sensor data localization problem is explored for the purpose of statistical anomaly detection.

In localization from pair-wise measurements, this thesis has emphasized making and using models of those pair-wise measurements of TOA, RSS, connectivity, and QRSS. From those models theoretical location estimation performance bounds are derived. These bounds are used to judge multiple localization algorithms, and to develop low-complexity algorithms which can achieve performance near the bound. Manifold learning algorithms in particular have shown promising results, with variance close to the lower bound. Two manifold learning methods have been proposed for use when pair-wise measurements are TOA, RSS, and connectivity.

Sensor data localization has been applied in both wired and wireless sensor networks. When measured data comes from a time-varying, isotropic correlated field, manifold learning methods can be used to estimate physical sensor coordinates. This thesis has presented several examples of manifold learning methods, and their performance in a particular multi-variate Gaussian sensor field model. Sensor data localization has also been applied to the Abilene Internet backbone network, for the purposes of visualizing high-dimensional data. The resulting non-physical router map can then be 'watched' for the purposes of identifying and classifying the spatial characteristics of traffic anomalies occurring on the network.

For wireless sensor networks, this research has provided guidelines for the development of localization systems in sensor networks. First, the lower bounds developed are tools useful to compare localization algorithms. Often in the literature, algorithms are simulated and reported without comparing to any other localization algorithm. Now, simulation performance can be quickly evaluated using the appropriate lower bound as a benchmark.

Second, broad generalizations can be made about localization systems using different pair-wise measurement modalities. Actual localization performance will depend on many implementation details, including the localization algorithm used, size and density of the network, the quantity of prior coordinate information, the pair-wise measurement method chosen, and the accuracies possible from the measurements in the environment of interest (the γ of Table 3.1). However, as a generalization, it appears that TOA measurements will be most useful in low-density sensor networks, since they are not as sensitive to increases in inter-device distances as RSS and AOA. Both AOA and TOA are typically able to achieve higher accuracy than RSS; however, that accuracy can come with higher device costs. Because of their

scaling characteristics, localization based on RSS and AOA measurements can, without sacrificing much accuracy, avoid taking measurements on longer-distance links and focus on those links between nearest neighbors. RSS measurements will allow accurate localization in dense sensor networks, and will be very attractive due to their low costs to system designers. In reality, RSS will be quantized. If RSS is quantized down to connectivity, then performance will suffer significantly (typically 50-60% increase in standard deviation, as a rule of thumb), but using a small number of quantization levels (on the order of 10) will bring the performance limits of QRSS nearly to those of pure RSS.

Furthermore, localization algorithms can be made to nearly achieve the lower bounds. In particular, the dwMDS method is demonstrated to be a low complexity, decentralized localization algorithm applicable to both RSS and TOA measurements and capable of nearly achieving the lower bound even in realistic channels. Also, a similarity-based manifold learning algorithm, Laplacian Eigenmap, serves as a means to achieve localization using connectivity measurements, and has much lower variance than existing methods in the literature. Finally, note that iterative, adaptive neighbor selection will be a key part of achieving low bias using often poor pair-wise measurements.

6.1 Future Research Directions

There are many research questions raised by this research, and many others which are important and have not yet been addressed.

6.1.1 Research Directions and 'Dead Ends'

First, can be useful to discuss research areas which, in the author's opinion, should be emphasized or de-emphasized. In the area of wireless sensor networks, the em-

phasis of this thesis has been on low-cost methods. Research funding is generally more available to promote localization applications which require higher accuracy, such as in acoustic sensor networks used for security applications, or in RF ad-hoc communication networks useful in military applications. Also, researchers are very able to design a network to achieve high accuracy, which can be quantified, rather than design a network to achieve low cost, which is more difficult to quantify. However, it is important to recognize the commercial applications that are possible with lower location-accuracy, lower-cost sensors such as logistics, air/water/soil quality monitoring, precision agriculture, and 'smart' building monitoring and control. In such areas, it is unlikely that AOA will be measured due to the cost of antenna arrays. As Moore's law progresses, TOA will become more affordable, and will be more likely to be a good solution. For the next decade, at least, it is likely that RSS, combined with a robust location estimation method, is likely to provide a marketable localization solution.

Acoustic signals are natural for localization because of their slow propagation speed compared to RF. However, it is important to note that acoustic transmission requires high transmit energies. Battery powered sensors should not be required to transmit acoustic (or ultrasound) signals unless battery life or size is not an issue. When other environmental sources of acoustic signals are used for location information, this would allow acoustic sensors to be localized, as discussed below and in Section 4.6.

Finally, in Internet traffic anomaly detection, there is a critical lack of 'ground truth' data. It is nearly impossible to take a significant set of traffic measurements and determine and quantify all anomalous activity which happened within it. Thus, comparing different methods is difficult and somewhat qualitative. It would be ex-

tremely valuable for a validated database to be produced for a particular time period or data set and made publicly available to the research community. But such a task would need to be an ongoing one, since internet traffic and anomalous activity changes characteristics from month to month and year to year. Without such a database, quantitative evaluation of anomaly detectors remains elusive - researchers must be content to present and see presented somewhat subjective measures of the detector performance.

6.1.2 Directions in RF Channel Measurement

While simulation will be very valuable, the next step in cooperative localization research is to test algorithms using measured data. However, pair-wise measurements of RSS, TOA, and AOA in wireless sensor networks have only begun to be reported, largely because of the complexity of such measurement campaigns, as related in Section 2.6. To conduct pair-wise measurements in a N sensor network requires $\mathcal{O}(N^2)$ measurements, and multiple sensor networks must be measured in order to develop and test statistical models. Furthermore, for sensor data localization, data measurements across space and time must be recorded and reported. These measurements will become possible as wireless sensor networks themselves are deployed for particular applications such as precision agriculture and environmental monitoring. Despite the complexity, data from such measurement campaigns will be of key importance to sensor network researchers.

In particular, for pair-wise measurements, it will be critical to consider the link dependencies in a real RF environment of deployment. Currently, the assumption is commonly made that all $\binom{N}{2}$ links are independent - however, the dependencies will in fact have an effect on localization performance, in addition to energy consumption and routing algorithms.

Such measurements should also consider joint statistics of RSS, TOA, and AOA, since using multiple measurement modalities simultaneously may be complementary [17]. While this article has considered them separately, multiple modalities measured together may provide more information than just the sum of their parts. For example, together, angle and time (spatio-temporal) measurements can cross-check for NLOS errors - if at the leading edge in the receiver's cross-correlation, power from multiple angles are measured, then it is apparent that the leading edge is not a direct LOS signal. This example is part of a bigger issue of determining sufficient statistics of joint spatio-temporal-signal strength measurements, which is still an open research topic.

6.1.3 Location Security

Such multi-modal measurements are critical for location security, which has not been addressed in this thesis. To be robust to intruders, localization algorithms must be secure against 'Process of Measurement' (PoM) attacks [71] which try to compromise the measurements which are being used for localization. For example, an attacker might use a wormhole to trick another sensor that it is located somewhere where it is not actually located. Many such attacks are often simple. By using multiple modalities, localization can be made robust against simple attacks.

6.1.4 Multiple Access Interference

It has been assumed in this thesis that appropriate multiple access channel (MAC) layer mechanisms are used to avoid interference when making pair-wise measurements. For example, since low-power wireless sensors rarely transmit (in order to save energy), it might be appropriate to use carrier-sense multiple access (CSMA) in order to avoid most RF interference. Furthermore, spread-spectrum techniques are

required in unlicensed bands in which wireless sensors are likely to operate, and will help reduce (but not completely eliminate) interference power.

However, some sensor localization systems may choose to allow MAC layer interference during measurements. For example, research by Kim, Pals, et. al. [63] measures round-trip TOA by simultaneous reply of neighbors to a sensor's query. The sensor uses multiple-user interference (MUI) cancelation techniques to estimate the TOA of each arrival. Clearly, interference will degrade pair-wise measurements, but real systems must deal with interference, either from other systems or from other sensors in the same system.

6.1.5 Sensor Reliability

One of the aspects of wireless sensor networks is supposed to be their reliability to sensor mis-calibration and failure. In order to achieve inexpensive devices on the order of cents per device, testing and calibration will not be priorities. First, sensor transmit powers will not likely be calibrated. Unless a sensor has a feedback loop in its transceiver architecture like proposed in [75], the sensor transmit power might need to be estimated as well, with (possibly) prior information on the transmit power distribution of typical manufactured sensors at different stages in their battery charge. Different transmit powers (and receiver characteristics) cause the radio channel to be asymmetric - $P_{i,j} \neq P_{j,i}$. In the case of connectivity measurements, this asymmetry is an advantage because it serves to provide 3-level QRSS information rather than just two levels. When both devices agree on connectivity, two devices are either in-range or out-of-range; and when devices don't agree on connectivity, they are likely to be in some intermediate range. For localization based on RSS measurements, system designers should plan to estimate transmit powers of devices in order to be robust to sensor transmit powers. Such robust RSS-based algorithms

have been reported [72], and are an important consideration for future algorithm research.

Sensor 'failure' can mean many things. First, sensors that cannot power up or communicate effectively do not exist to the network. Sensor deployments must be dense enough that a given percentage of sensors failing does not disconnect the network. Second, sensor failure might mean wildly incorrect measurements, for example, from a uniform random distribution across a wide range. This is similar to the localization security issue discussed above. Algorithms which allow measurements to be 'weighted' low when neighbors see inconsistencies in their data, or ignored adaptively, are of key importance. The dwMDS method and the LE-based method discussed in Chapter IV both use adaptive neighborhood methods which update neighbor weights adaptively using multiple rounds of location estimation. Future research could quantify the ability of these algorithms to avoid a breakdown when a sensor's measurements fail in this way.

6.1.6 Sensor Mobility

This thesis estimated sensor coordinates over time in the context of Chapter V. In the localization in wireless sensor networks, the mobility of sensors was not explicitly considered. It is possible that tracking of mobile sensors could be done simply by re-initializing the measurement and localization process. However, this would not explicitly taken advantage of mobility to improve localization performance or to reduce bandwidth requirements over time.

Mobility creates the problem of locating and tracking moving sensors in real time, and also the opportunity to improve sensor localization. For the problem of passive tracking of sources in the environment, a review is presented in [19], but the problem of tracking active sensors has not been sufficiently addressed as a collaborative signal

processing problem. The sensor tracking problem is an important aspect of many applications, including the animal tracking and logistics applications discussed in Sections 1.1.1 and 1.1.2.

For example, if a sensor makes multiple measurements to its neighbors as it moves across space, it has the opportunity to reduce environment-dependent errors (such as shadowing) by averaging over space. The multiple measurements are useful to help improve coordinate estimates for other sensors in the network, not just the mobile node. Researchers have tested schemes which use mobile sources and sensors to achieve cooperative localization [18, 43], however further opportunities to exploit mobility remain to be explored.

Consider the update rate, *i.e.*, how often location must be re-estimated. If sensors are completely stationary, localization might just be done at startup. In networks with some changes over time, it shouldn't be necessary to continually make and broadcast pair-wise measurements between stationary sensors. There might be a distributed algorithm which detects sensor motion (by monitoring for changing pair-wise measurements) and then updates the location estimates only of those sensors in motion, in order to save communication and computation. Such a distributed algorithm would be valuable in both wireless sensor networks and network visualization applications. The tradeoffs between measurement requirements, communication, and accuracy should be more explicitly explored.

Furthermore, latency, *i.e.*, speed of localization, becomes an issue, because long delays make location information obsolete. Speeding up distributed algorithms such as the dwMDS algorithm, or better understanding convergence rates, becomes important. What is the tradeoff between latency and accuracy in a communication-limited mobile network?

6.1.7 Directions: Physical Localization from Sensor Data

There are certainly limitations when estimating physical sensor locations using sensor field data measurements, as discussed in Section 4.6. First, this thesis has shown two different applications of sensor data - localization and visualization for anomaly detection. These are presented as separate applications. When doing localization from sensor data, it would be critical that no spatial anomaly existed in the field data - such an anomaly would clearly result in very poor location estimates. Similarly, for anomaly detection, coordinates of sensors shouldn't be used as instantaneous sensor positions. Only in the average, over time, might these coordinates indicate something about their sensor positions (see for example the average router map in Figure 5.2(a)).

Secondly, the sensor field used for localization must be both spatially isotropic and must be already a field which is important to the sensor network application. Localization will not be possible from field data which are dominated by non-spatial factors. For example, measuring poverty in counties across the U.S. would not provide spatial information, since its level is dominated by many other non-spatial factors.

However, particular sensor applications may be able to achieve localization from sensor field data. The most likely would be sensors that measure the acoustic field. Localization systems have already been designed for outdoor environmental monitoring sensor networks which are calibrated by driving a GPS-equipped vehicle (or by flying a helicopter) through the area of deployment while transmitting its known location to the sensors on the ground [18]. These studies use known-location sources for sensor localization; in general, unknown-location acoustic sources ('background noise') could provide a huge quantity of sensor location information. In collaboration with Norman H. Adams, a measurement campaign using acoustic sensors

(microphones) was conducted. The results indicated that simple Euclidean distance measures wouldn't be appropriate to judge the distance between two acoustic signals, which may have time offset and amplitude calibration issues. Furthermore, the measurements used directional microphones, which introduced anisotropy into the field measurements. Further measurements and an exploration of better distance measures will be important for future research in this area.

6.1.8 Routing in Wireless Sensor Networks

As another research direction, geographic routing is an application of sensor localization. The use of the coordinates of sensors can reduce routing tables and simplify routing algorithms. Localization errors, however, can adversely impact routing algorithms, leading to longer paths and delivery failures [53]. For the purposes of routing efficiency, actual geographical coordinates may be less useful than 'virtual' coordinates [100], *i.e.*, a representation of a sensor's 'location' in the graph of network connectivity. These virtual coordinates could be in an arbitrary dimension, possibly higher than 2 or 3. There are often paths in multi-hop wireless networks that consume less power than the shortest, straightest-line path between two nodes, and virtual coordinates may enable a better representation of the network connectivity. The virtual coordinate estimation problem is a dimension reduction problem which inputs each sensor's connectivity or RSS measurement vector and outputs a virtual coordinate in an arbitrary low dimension, optimized to minimize a communication cost metric. Such research could enable more energy-efficient scalable routing protocols for very large sensor networks.

6.1.9 Network Self-Administration

Finally, the sensor data localization framework is applicable to a variety of non-model-based data visualization and statistical detection and estimation problems. In general, it is desirable for wireless and wired networks to be self-configuring and self-monitoring. Due to the large scale of the Internet, and the proposed scale of wireless sensor networks, the reliance on human moderation becomes problematic. Networks that can use statistical learning techniques to model their own 'normal' behavior – either that of its measured data, or its traffic and routing characteristics – can then detect anomalous data or behavior, and possibly react to repair itself. Such self-monitoring networks are clearly a complicated and rewarding area of research.

APPENDICES

APPENDIX A

Derivations and Proofs

A.1 CRB for Network Self-Calibration

The diagonal elements, $f_{k,k}$, of $\mathbf{F}$ given in (3.3) are,

$$\begin{aligned} f_{k,k} &= \mathrm{E}\left(\tfrac{\partial}{\partial\theta_k} l(\mathbf{X}|\boldsymbol{\theta})\right)^2 = \mathrm{E}\Big(\sum_{j\in H(k)} \tfrac{\partial}{\partial\theta_k} l_{k,j}\Big)^2 \\ f_{k,k} &= \sum_{j\in H(k)} \sum_{p\in H(k)} \mathrm{E}\left(\tfrac{\partial}{\partial\theta_k} l_{k,j}\right)\left(\tfrac{\partial}{\partial\theta_k} l_{k,p}\right) \end{aligned}$$

Since $X_{k,j}$ and $X_{k,p}$ are independent random variables, and $E[\frac{\partial}{\partial\theta_k} l_{k,j}] = 0$, the expectation of the product is only nonzero for $p = j$. Thus $f_{k,k}$ simplifies to the $k = l$ result in (3.4). The off-diagonal elements similarly simplify,

$$f_{k,l} = \sum_{j\in H(k)} \sum_{p\in H(l)} \mathrm{E}\left(\tfrac{\partial}{\partial\theta_k} l_{k,j}\right)\left(\tfrac{\partial}{\partial\theta_l} l_{l,p}\right)$$

Here, due to independence and zero mean of the two terms, the expectation of the product will be zero unless both $p = k$ and $j = l$. Thus the $k \neq l$ result in (3.4).

A.2 Proof of Theorem III.5

Compare $\mathbf{F}$, the FIM for the n unknown device problem, to $\mathbf{G}$, the FIM for the $n+1$ unknown device case. Partition $\mathbf{G}$ into blocks,

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_{ul} & \mathbf{g}_{ur} \\ \mathbf{g}_{ll} & g_{lr} \end{bmatrix}$$

where $\mathbf{G}_{ul}$ is an $n \times n$ matrix, g_{lr} is the scalar Fisher information for θ_{n+1}, and $\mathbf{g}_{ur} = \mathbf{g}_{ll}^T$ are $n \times 1$ vectors with kth element,

$$\begin{aligned} \mathbf{g}_{ur}(k) &= \mathrm{I}_{H(n+1)}(k)\,\mathrm{E}\left(\tfrac{\partial}{\partial\theta_k} l_{k,n+1}^{n+1}\right)\left(\tfrac{\partial}{\partial\theta_{n+1}} l_{k,n+1}^{n+1}\right), \\ g_{lr} &= \sum_{j\in H(n+1)} \mathrm{E}\left(\tfrac{\partial}{\partial\theta_{n+1}} l_{n+1,j}^{n+1}\right)^2. \end{aligned}$$

Here, denote the log-likelihood of the observation between devices i and j in (3.2) as $l_{i,j}^n$ and $l_{i,j}^{n+1}$ for the n and $(n+1)$ unknown device cases, respectively. Similarly, let $l^n(\mathbf{X}|\boldsymbol{\gamma}_n)$ and $l^{n+1}(\mathbf{X}|\boldsymbol{\gamma}_{n+1})$ be the joint log-likelihood function in (3.2) for the n and $n+1$ unknown device cases, respectively. Then

$$l^{n+1}(\mathbf{X}|\boldsymbol{\gamma}_{n+1}) = \sum_{i=1}^{m+n+1} \sum_{\substack{j\in H(i)\\ j<i}} l_{i,j}^{n+1} = l^n(\mathbf{X}|\boldsymbol{\gamma}_n) + \sum_{\substack{j\in\\ H(n+1)}} l_{n+1,j}^{n+1}.$$

Since $l_{n+1,j}^{n+1}$ is a function only of parameters $\gamma_{n+1} = \theta_{n+1}$ and γ_j,

$$\frac{\partial^2}{\partial\theta_k\partial\theta_l} \sum_{\substack{j\in\\ H(n+1)}} l_{n+1,j}^{n+1} = \begin{cases} \mathrm{I}_{H(n+1)}(k)\frac{\partial^2}{\partial\theta_k^2} l_{n+1,k}^{n+1}, & l = k \\ 0, & l \neq k \end{cases}$$

Thus $\mathbf{G}_{ul} = \mathbf{F} + \mathrm{diag}(\mathbf{h})$, where $\mathbf{h} = \{h_1, \ldots, h_n\}$ and $h_k = \mathrm{I}_{H(n+1)}(k)\mathrm{E}(\frac{\partial}{\partial\theta_k} l_{n+1,k}^{n+1})^2$. Compare the CRB for the covariance matrix of the first n devices in the n and $n+1$ device cases, given by $\mathbf{F}^{-1}$ and $[\mathbf{G}^{-1}]_{ul}$, respectively. Here, $[\mathbf{G}^{-1}]_{ul}$ is the upper left $n \times n$ submatrix of G^{-1},

$$[\mathbf{G}^{-1}]_{ul} = \left\{\mathbf{G}_{ul} - \mathbf{g}_{ur} g_{lr}^{-1} \mathbf{g}_{ll}\right\}^{-1} = \{\mathbf{F} + \mathbf{J}\}^{-1}$$
$$\text{where} \quad \mathbf{J} = \mathrm{diag}(\mathbf{h}) - \tfrac{\mathbf{g}_{ur}\mathbf{g}_{ur}^T}{g_{lr}}$$

Both $\mathbf{F}$ and $\mathbf{J}$ are Hermitian. We know that $\mathbf{F}$ is positive semidefinite. Let $\lambda_k(\mathbf{F}), k = 1 \ldots n$ be the eigenvalues of $\mathbf{F}$ and $\lambda_k(\mathbf{F}+\mathbf{J}), k = 1 \ldots n$ be the eigenvalues of the sum, both listed in increasing order, then if it can be shown that $\mathbf{J}$ is positive semidefinite, then it is known [55] that:

$$0 \leq \lambda_k(\mathbf{F}) \leq \lambda_k(\mathbf{F}+\mathbf{J}), \forall k = 1 \ldots n \tag{A.1}$$

Since the eigenvalues of a matrix inverse are the inverses of the eigenvalues of the matrix,

$$\lambda_k\left(\{\mathbf{F}+\mathbf{J}\}^{-1}\right) \leq \lambda_k(\mathbf{F}^{-1}), \forall k = 1\ldots n, \tag{A.2}$$

which proves property 1 of Theorem III.5. If in addition, it can be shown that $\mathrm{tr}(\mathbf{J}) > 0$, then $\mathrm{tr}(\mathbf{F}+\mathbf{J}) > \mathrm{tr}(\mathbf{F})$, and therefore $\sum_{k=1}^{n}\lambda_k(\mathbf{F}+\mathbf{J}) > \sum_{k=1}^{n}\lambda_k(\mathbf{F})$. This with (A.1) implies that $\lambda_j(\mathbf{F}+\mathbf{J}) > \lambda_j(\mathbf{F})$ for at least one $j \in 1\ldots n$. Thus in addition to (A.2),

$$\lambda_j\left(\{\mathbf{F}+\mathbf{J}\}^{-1}\right) < \lambda_j(\mathbf{F}^{-1}), \text{for some } j \in 1\ldots n$$

which implies that $\mathrm{tr}\left(\{\mathbf{F}+\mathbf{J}\}^{-1}\right) < \mathrm{tr}(\mathbf{F}^{-1})$, which proves property 2 of Theorem III.5.

A.2.1 Showing positive semidefiniteness and positive trace of J

The diagonal elements of $\mathbf{J}$, $[\mathbf{J}]_{k,k}$ are,

$$[\mathbf{J}]_{k,k} = h_k - \mathbf{g}_{ur}^2(k)/g_{lr}. \tag{A.3}$$

If $k \notin H(n+1)$ then $h_k = 0$ and $\mathbf{g}_{ur}(k) = 0$, thus $[\mathbf{J}]_{k,k} = 0$. Otherwise, if $k \in H(n+1)$,

$$[\mathbf{J}]_{k,k} = \mathrm{E}\left(\frac{\partial l_{n+1,k}^{n+1}}{\partial\theta_k}\right)^2 - \frac{\left[\mathrm{E}\left(\frac{\partial l_{n+1,k}^{n+1}}{\partial\theta_k}\right)\left(\frac{\partial l_{n+1,k}^{n+1}}{\partial\theta_{n+1}}\right)\right]^2}{\sum_{j\in H(n+1)}\mathrm{E}\left(\frac{\partial l_{n+1,j}^{n+1}}{\partial\theta_{n+1}}\right)^2}.$$

Because of reciprocity, the numerator is equal to the square of the $j = k$ term in the sum in the denominator. Thus

$$[\mathbf{J}]_{k,k} \geq \mathrm{E}\left(\frac{\partial l_{n+1,k}^{n+1}}{\partial\theta_k}\right)^2 - \mathrm{E}\left(\frac{\partial l_{n+1,k}^{n+1}}{\partial\theta_k}\frac{\partial l_{n+1,k}^{n+1}}{\partial\theta_{n+1}}\right) = 0.$$

The equality will hold if k is the only member of the set $H(n+1)$. When condition (2) of Theorem III.5 holds, $[\mathbf{J}]_{k,k}$ will be strictly greater than zero. Thus tr$\mathbf{J} > 0$.

Next, it is shown that $\mathbf{J}$ is diagonally dominant [55], i.e.,

$$[\mathbf{J}]_{k,k} \geq \sum_{\substack{j=1 \\ j \neq k}}^{n} |[\mathbf{J}]_{k,j}| = \sum_{\substack{j=1 \\ j \neq k}}^{n} \frac{|\mathbf{g}_{ll}(k)\mathbf{g}_{ll}(j)|}{g_{lr}},$$

where $[\mathbf{J}]_{k,k}$ is given in (A.3). Since $H(n+1) \neq \emptyset$, thus $g_{lr} > 0$, and an equivalent condition is,

$$g_{lr} h_k \geq |\mathbf{g}_{ur}(k)| \sum_{j=1}^{n} |\mathbf{g}_{ur}(j)|. \tag{A.4}$$

If $k \notin H(n+1)$ then $h_k = 0$ and $\mathbf{g}_{ur}(k) = 0$, and the equality holds. If $k \in H(n+1)$, then

$$g_{lr} h_k = \mathrm{E}\left(\frac{\partial l_{k,n+1}^{n+1}}{\partial \theta_k}\right)^2 \sum_{j \in H(n+1)} \mathrm{E}\left(\frac{\partial l_{n+1,j}^{n+1}}{\partial \theta_{n+1}}\right)^2.$$

Because of condition (1) of Theorem III.5,

$$\mathrm{E}\left(\frac{\partial l_{k,n+1}^{n+1}}{\partial \theta_k}\right)^2 = \left|\mathrm{E}\left(\frac{\partial l_{k,n+1}^{n+1}}{\partial \theta_{n+1}} \frac{\partial l_{k,n+1}^{n+1}}{\partial \theta_k}\right)\right|$$

Thus

$$g_{lr} h_k = |\mathbf{g}_{ur}(k)| \left[\sum_{\substack{j \geq 1 \\ j \in H(n+1)}} |\mathbf{g}_{ur}(j)| + \sum_{\substack{j \leq 0 \\ j \in H(n+1)}} \left|\mathrm{E}\left(\frac{\partial l_{j,n+1}^{n+1}}{\partial \theta_{n+1}} \frac{\partial l_{j,n+1}^{n+1}}{\partial \theta_j}\right)\right| \right]$$

Since $\mathbf{g}_{ur}(j) = 0$ if $j \notin H(n+1)$, the first sum can include all $j \in 1 \ldots n$. Since the 2^{nd} sum is ≥ 0, (A.4) is true.

Diagonal dominance implies $\mathbf{J}$ is positive semidefinite, which proves (A.2). Note that if $H(n+1)$ includes ≥ 1 reference device, the 2^{nd} sum is > 0 and the inequality in (A.4) is strictly > 0,which implies positive definiteness of $\mathbf{J}$ and assures that the CRB will strictly decrease.

A.3 CRB for Location Estimation

A.3.1 RSS

For the elements of $\mathbf{F}_R$, using (2.2) and (3.2),

$$l_{i,j} = \log\left(\frac{10\log 10}{\sqrt{2\pi\sigma_{dB}^2}}\frac{1}{P_{i,j}}\right) - \frac{\gamma}{8}\left(\log\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{\delta_{i,j}^2}\right)^2.$$

Recall $\|\mathbf{z}_i - \mathbf{z}_j\| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. Thus,

$$\frac{\partial}{\partial x_j} l_{i,j} = -\frac{\gamma}{2}\left(\log\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{\delta_{i,j}^2}\right)\frac{x_j - x_i}{\|\mathbf{z}_i - \mathbf{z}_j\|^2}.$$

Note that $\frac{\partial}{\partial x_j} l_{i,j} = -\frac{\partial}{\partial x_i} l_{i,j}$, thus the log-normal distribution of RSS measurements meets condition (1) of Theorem III.5. The 2^{nd} partials differ based on whether or not $i = j$ and if the partial is taken w.r.t. y_i or x_i. For example,

$$\begin{aligned}
\frac{\partial^2 l_{i,j}}{\partial x_j \partial y_j} &= -b\frac{(x_i - x_j)(y_i - y_j)}{\|\mathbf{z}_i - \mathbf{z}_j\|^4}\left[-\log\left(\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{\delta_{i,j}^2}\right) + 1\right] \\
\frac{\partial^2 l_{i,j}}{\partial x_j \partial y_i} &= -b\frac{(x_i - x_j)(y_i - y_j)}{\|\mathbf{z}_i - \mathbf{z}_j\|^4}\left[\log\left(\frac{\|\mathbf{z}_i - \mathbf{z}_j\|^2}{\delta_{i,j}^2}\right) - 1\right]
\end{aligned}$$

Note that $\mathrm{E}[\log(\|\mathbf{z}_i - \mathbf{z}_j\|^2/\delta_{i,j}^2)] = 0$. Thus the FIM simplifies to take the form in (3.6) with $s = 4$ and $h_{k,l} = 0 \forall k, l$.

A.3.2 TOA

For the TOA case,

$$l_{i,j} = \left(-\log\sqrt{2\pi\sigma_T^2} - \frac{(T_{i,j} - \|\mathbf{z}_i - \mathbf{z}_j\|/v_p)^2}{2\sigma_T^2}\right) \tag{A.5}$$

taking the partial w.r.t. x_j

$$\frac{\partial}{\partial x_j} l_{i,j} = -\frac{1}{\sigma_T^2}\left(\frac{v_p T_{i,j}}{\|\mathbf{z}_i - \mathbf{z}_j\|} - 1\right)(x_j - x_i), \tag{A.6}$$

Note that in the TOA case it is also true that $\frac{\partial}{\partial x_j} l_{i,j} = -\frac{\partial}{\partial x_i} l_{i,j}$, meeting condition (1) of Theorem III.5. Two examples of the second partial derivatives are given by,

$$\begin{aligned}
\frac{\partial^2}{\partial x_j \partial y_j} &= -\frac{1}{\sigma_T^2 v_p^2} \frac{cT_{i,j}}{\|\mathbf{z}_i - \mathbf{z}_j\|} \frac{(x_i - x_j)(y_i - y_j)}{(x_i - x_j)^2 + (y_i - y_j)^2} \\
\frac{\partial^2}{\partial x_j \partial x_i} &= -\frac{1}{\sigma_T^2 v_p^2} \left[\frac{v_p T_{i,j}}{\|\mathbf{z}_i - \mathbf{z}_j\|} - 1 - \frac{v_p T_{i,j}}{\|\mathbf{z}_i - \mathbf{z}_j\|} \frac{(x_i - x_j)^2}{\|\mathbf{z}_i - \mathbf{z}_j\|^2} \right]
\end{aligned}$$

The 2nd partial derivatives depend on the term, $v_p T_{i,j}/\|\mathbf{z}_i - \mathbf{z}_j\|$, which has an expected value of 1, and the terms of $\mathbf{F}_R$ take the form in (3.6) with $s = 2$ and $h_{k,l} = 0 \forall k, l$.

A.3.3 QRSS

Next, the CRB is derived for the case when measurements are K-level QRSS. It has already been shown that the CRB for any self-calibration estimator is a function of the expected value of the second partial derivatives of the terms $\{l_{i,j}\}$ where,

$$l_{i,j} = \log \mathcal{P}[Q_{i,j} | \mathbf{z}_i, \mathbf{z}_j]. \tag{A.7}$$

The first partial derivatives for the QRSS case of $l_{i,j}$ with respect to x_i are

$$\frac{\partial}{\partial x_i} l_{i,j} = \frac{\frac{\partial}{\partial x_i} \mathcal{P}[Q_{i,j} | \mathbf{z}_i, \mathbf{z}_j]}{\mathcal{P}[Q_{i,j} | \mathbf{z}_i, \mathbf{z}_j]}.$$

Similarly,

$$\frac{\partial^2}{\partial x_i^2} l_{i,j} = \frac{\frac{\partial^2}{\partial x_i^2} \mathcal{P}[Q_{i,j} | \mathbf{z}_i, \mathbf{z}_j]}{\mathcal{P}[Q_{i,j} | \mathbf{z}_i, \mathbf{z}_j]} - \left(\frac{\frac{\partial}{\partial x_i} \mathcal{P}[Q_{i,j} | \mathbf{z}_i, \mathbf{z}_j]}{\mathcal{P}[Q_{i,j} | \mathbf{z}_i, \mathbf{z}_j]} \right)^2.$$

Thus,

$$\begin{aligned}
-\mathrm{E}\left[\frac{\partial^2}{\partial x_i^2} l_{i,j} \right] &= -\sum_{s=0}^{K-1} \frac{\partial^2}{\partial x_i^2} \mathcal{P}[Q_{i,j} = s | \mathbf{z}_i, \mathbf{z}_j] \\
&\quad + \sum_{s=0}^{K-1} \frac{\left(\frac{\partial}{\partial x_i} \mathcal{P}[Q_{i,j} = s | \mathbf{z}_i, \mathbf{z}_j] \right)^2}{\mathcal{P}[Q_{i,j} = s | \mathbf{z}_i, \mathbf{z}_j]}.
\end{aligned} \tag{A.8}$$

The first sum is a telescoping sum of $\frac{\partial^2}{\partial x_i^2}\Phi[\cdot]$ terms,

$$\begin{aligned}\sum_{s=0}^{K-1}\frac{\partial^2}{\partial x_i^2}\mathcal{P}[Q_{i,j}=s|\mathbf{z}_i,\mathbf{z}_j] &= \sum_{s=0}^{K-1}\frac{\partial^2}{\partial x_i^2}\Phi\left[g_{i,j}(s+1)\right]-\sum_{s=0}^{K-1}\frac{\partial^2}{\partial x_i^2}\Phi\left[g_{i,j}(s)\right]\\ &= \frac{\partial^2}{\partial x_i^2}\Phi\left[g_{i,j}(K)\right]-\frac{\partial^2}{\partial x_i^2}\Phi\left[g_{i,j}(0)\right]=0.\end{aligned}$$

To further evaluate (A.8), note that

$$\begin{aligned}\frac{\partial}{\partial x_i}\mathcal{P}[Q_{i,j}=s|\mathbf{z}_i,\mathbf{z}_j] &= \frac{\sqrt{\gamma}}{\sqrt{2\pi}}\frac{x_i-x_j}{\|\mathbf{z}_i-\mathbf{z}_j\|^2}\\ &\left[\exp\left(-\frac{\gamma}{2}\ln^2\frac{\|\mathbf{z}_i-\mathbf{z}_j\|}{d_{s+1}}\right)-\exp\left(-\frac{\gamma}{2}\ln^2\frac{\|\mathbf{z}_i-\mathbf{z}_j\|}{d_{s+1}}\right)\right].\end{aligned} \tag{A.9}$$

As a result of (A.9) and (A.9), (A.8) simplifies to

$$-\mathrm{E}\left[\frac{\partial^2}{\partial x_i^2}l_{i,j}\right]=\frac{\gamma}{2\pi}\frac{(x_i-x_j)^2}{\|\mathbf{z}_i-\mathbf{z}_j\|^4}h_{i,j}, \tag{A.10}$$

where $h_{i,j}$ takes the form of (3.7). The terms depending on other second partial derivatives are very similar to (A.10).

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Computer Networks (Elsevier) Journal*, vol. 38, no. 4, pp. 393–422, Mar. 2002.

[2] J. Albowicz, A. Chen, and L. Zhang, "Recursive position estimation in sensor networks," in *IEEE Int. Conf. on Network Protocols*, Nov. 2001, pp. 35–41.

[3] S. A. Altmann and J. Altmann, "The transformation of behaviour field studies," *Animal Behaviour*, vol. 65, pp. 413–423, 2002.

[4] J. N. Ash and R. L. Moses, "Outlier compensation in sensor network self-localization via the EM algorithm," in *IEEE ICASSP*, March 19-23, 2005.

[5] J. N. Ash. SeNeLEx: The Sensor Network Localization Explorer. [Online]. Available: http://www.ece.osu.edu/~ashj/localization/

[6] J. N. Ash and R. L. Moses, "Acoustic sensor network self-localization: Experimental results," in *Military Sensing Symposia*, 2003.

[7] J. N. Ash and L. C. Potter, "Sensor network localization via received signal strength measurements with directional antennas," in *Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing*, 2004.

[8] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, June 2003.

[9] L. Bergamaschi, G. Pini, and F. Sartoretto, "Computational experience with sequential and parallel,preconditioned jacobidavidson for large, sparse symmetric matrices," *J. Computational Physics*, vol. 188, no. 1, pp. 318–331, June 2003.

[10] J. E. Berube, "Personal security system with weighted receiver locations," US Patent 5,898,367, April 27, 1999, Detection Systems, Inc.

[11] P. Biswas and Y. Ye, "A distributed method for solving semidefinite programs arising from ad hoc wireless sensor network localization," Dept. of Computer Science, Stanford University, Tech. Rep., Oct. 2003.

[12] ——, "Semidefinite programming for ad hoc wireless sensor network localization," Dept. of Computer Science, Stanford University, Tech. Rep., Sept 2003.

[13] J. Borràs, P. Hatrack, and N. B. Mandayam, "Decision theoretic framework for NLOS identification," in *IEEE VTC 1998*, vol. 2, May 1998, pp. 1583–1587.

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

[15] J. J. Caffery, *Wireless Location in CDMA Cellular Radio Systems*. Kluwer Academic Publishers, 1999.

[16] G. Carter, *Coherence and Time Delay Estimation*. Piscataway, NJ: IEEE Press, 1993.

[17] A. Catovic and Z. Sahinoglu, "The Cramér-Rao bounds of hybrid TOA/RSS and TDOA/RSS location estimation schemes," Mitsubishi Electric Research Lab, Tech. Rep. TR-2003-143, Jan. 2004. [Online]. Available: http://www.merl.com/reports/docs/TR2003-143.pdf

[18] V. Cevher and J. H. McClellan, "Sensor array calibration via tracking with the extended Kalman filter," in *Fed. Lab. Symp. on Adv. Sensors*, Mar. 2001, pp. 51–56.

[19] J. C. Chen, K. Yao, and R. E. Hudson, "Source localization and beamforming," *IEEE Signal Processing Mag.*, vol. 19, no. 2, pp. 30–39, Mar. 2002.

[20] P.-C. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *IEEE Wireless Comm. and Networking Conf.*, Sept. 1999, pp. 316–320.

[21] W. Cleveland, "Robust locally weighted regression and smoothing scatterplots," *J. American Statistical Assoc.*, vol. 74, no. 368, pp. 829–836, 1979.

[22] M. Coates, "Distributed particle filters for sensor networks," in *IEEE Information Processing in Sensor Networks (IPSN)*, April 2004, pp. 99–107.

[23] S. L. Collier and D. K. Wilson, "Performance bounds for passive sensor arrays operating in a turbulent medium: Plane-wave analysis," *The Journal of the Acoustical Society of America*, vol. 113, no. 5, pp. 2704–2718, 2003. [Online]. Available: http://link.aip.org/link/?JAS/113/2704/1

[24] Cooperative Association for Internet Data Analysis (CAIDA). CAIDA web page. http://www.caida.org/.

[25] N. S. Correal, S. Kyperountas, Q. Shi, and M. Welborn, "An ultra wideband relative location system," in *IEEE Conf. on Ultra Wideband Systems and Technologies*, Nov. 2003.

[26] N. S. Correal and N. Patwari, "Wireless sensor networks: Challenges and opportunities," in *Proceedings of the 2001 Virginia Tech Symposium on Wireless Personal Communications*, June 2001, pp. 1–9.

[27] J. A. Costa, N. Patwari, and A. O. Hero III, "Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks," *IEEE/ACM Trans. Sensor Networks*, (to appear). [Online]. Available: http://www.eecs.umich.edu/~hero/comm.html

[28] ——, "Achieving high-accuracy distributed localization in sensor networks," *IEEE ICASSP'05*, March 2005.

[29] A. J. Coulson, A. G. Williamson, and R. G. Vaughan, "A statistical basis for lognormal shadowing effects in multipath fading channels," *IEEE Trans. on Veh. Tech.*, vol. 46, no. 4, pp. 494–502, April 1998.

[30] T. Cox and M. Cox, *Multidimensional Scaling*. London: Chapman & Hall, 1994.

[31] N. A. Cressie, *Statistics for Spatial Data*. New York: John Wiley & Sons, Inc., 1991.

[32] E. R. Davidson, "The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices," *J. Comput. Phys.*, vol. 17, no. 1, pp. 87–94, Jan. 1975.

[33] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, "Convex position estimation in wireless sensor networks," in *IEEE INFOCOM*, vol. 3, 2001, pp. 1655 –1663.

[34] D. L. Donoho and C. Grimes, "Hessian eigenmaps: new locally linear embedding techniques for high-dimensional data," Dept. of Statistics, Stanford University, Tech. Rep. TR2003-08, March 2003. [Online]. Available: http://www-stat.stanford.edu/research/

[35] G. Durgin, T. S. Rappaport, and H. Xu, "Measurements and models for radio path loss and penetration loss in and around homes and trees at 5.85 GHz," *IEEE Journal on Sel. Areas in Comm.*, vol. 46, no. 11, pp. 1484–1496, Nov. 1998.

[36] G. D. Durgin, *Space-Time Wireless Channels*. Prentice Hall PTR, 2002.

[37] C. Estan, S. Savage, and G. Varghese, "Automatically inferring patterns of resource consumption in network traffic," in *ACM SIGCOMM'03*, Aug. 2003, pp. 137–148.

[38] D. Estrin, Ed., *Embedded Everywhere: A Research Agenda for Networked Systems of Embedded Computers*. National Academy Press, 2001, report of the National Research Council Committee on Networked Systems of Embedded Computers. [Online]. Available: http://books.nap.edu/html/embedded_everywhere/

[39] B. J. Feder, "Wireless sensor networks spread to new territory," *New York Times*, July 26 2004. [Online]. Available: http://www.nytimes.com/2004/07/26/business/26sensor.html

[40] R. Fleming and C. Kushner, "Low-power, miniature, distributed position location and communication devices using ultra-wideband, nonsinusoidal communication technology," Aetherwire Inc., Semi-Annual Technical Report, ARPA Contract J-FBI-94-058, Tech. Rep., July 1995.

[41] R. J. Fontana and S. J. Gunderson, "Ultra-wideband precision asset location system," in *IEEE Conf. UWB Syst. & Tech.*, May 2002, pp. 147–150.

[42] M. Fullmer and S. Romig, "The OSU Flow-tools package and Cisco NetFlow logs," in *Large Installation System Administration (LISA) Conference 2000*, Dec. 2000, pp. 291–303.

[43] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor networks using a moving target," in *IEEE Information Processing in Sensor Networks (IPSN)*, April 2004, pp. 61–70.

[44] W. L. Garber, "Self-localization of sensor networks," Master's thesis, Ohio State University, 2003.

[45] S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. Poor, and Z. Sahinoglu, "Localization via ultra-wideband radios," *IEEE Signal Processing Mag.*, vol. 22, no. 4, pp. 70–86, July 2005.

[46] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss, "QuickSAND: Quick summary and analysis of network data," DIMACS, Tech. Rep. 2001-43, Nov. 2001. [Online]. Available: http://www.math.lsa.umich.edu/~annacg/ps.files/quickdimacstr.ps

[47] L. Girod, V. Bychkovskiy, J. Elson, and D. Estrin, "Locating tiny sensors in time and space: a case study," in *IEEE Int. Conf. on Computer Design*, 2002, pp. 214–219.

[48] T. Gneiting, "Compactly supported correlation functions," *J. Multivariate Analysis*, vol. 83, pp. 493–508, 2002.

[49] P. Groenen, *The majorization approach to multidimensional scaling: some problems and extensions*. DSWO Press, Leiden University, 1993.

[50] R. Gupta and A. O. Hero III, "High-rate vector quantization for detection," *IEEE Trans. Info Theory*, vol. 49, no. 8, pp. 1951–1969, Aug. 2003.

[51] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks," *IEEE Signal Processing Mag.*, vol. 22, no. 4, pp. 41–53, July 2005.

[52] H. Hashemi, "The indoor radio propagation channel," *Proceedings of the IEEE*, vol. 81, no. 7, pp. 943–968, July 1993.

[53] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *IEEE Mobicom 2003*, Sept. 2003.

[54] A. O. Hero III, J. A. Fessler, and M. Usman, "Exploring estimator bias-variance tradeoffs using the uniform CR bound," *IEEE Trans. on Sig. Proc.*, vol. 44, no. 8, pp. 2026–2041, Aug. 1996.

[55] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge Univ. Press, 1990.

[56] L. Hu and D. Evans, "Localization for mobile sensor networks," in *IEEE Mobicom 2004*, Sept. 2004, pp. 45–57.

[57] A. T. Ihler, J. W. Fisher III, and R. L. Moses, "Nonparametric belief propagation for self-calibration in sensor networks," in *IEEE ICASSP 2004*, May 2004.

[58] Institute of Electronics and Electrical Engineering. IEEE Wireless Standards Zone. http://standards.ieee.org/wireless/.

[59] Internet2. Abilene Observatory web page. [Online]. Available: http://abilene.internet2.edu/observatory/

[60] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling," in *IEEE INFOCOM'04*, 2004, pp. 2652–2661.

[61] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy efficient computing for wildlife tracking: Design tradeoffs and early experiences with ZebraNet," in *Conf. Architectural Support for Programming Lang. and Operating Systems (ASPLOS)*, October 2002.

[62] S. M. Kay, *Fundamentals of Statistical Signal Processing*. New Jersey: Prentice Hall, 1993.

[63] S. Kim, T. Pals, R. A. Iltis, and H. Lee, "CDMA multipath channel estimation using generalized successive interference cancellation algorithm for radiolocation," in *36th Conf. Information Sciences & Systems*, March 2002.

[64] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[65] J. Kumagai and S. Cherry, "Sensors and sensibility," *IEEE Spectrum*, vol. 41, no. 7, pp. 22–28, July 2004.

[66] A. Lakhina, M. Crovella, and C. Diot, "Characterization of network-wide anomalies in traffic flows," in *ACM/SIGCOMM Internet Measurement Conference*, Oct. 2004.

[67] ——, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM '04*, Aug. 2004.

[68] E. G. Larsson, "Cramér-Rao bound analysis of distributed positioning in sensor networks," *IEEE Signal Processing Letters*, vol. 11, no. 3, pp. 334–337, March 2004.

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

[70] Y. Lei, A. Kiremidjian, K. Nair, J. Lynch, and K. Law, "Algorithms for time synchronization of wireless structural monitoring sensors," *Earthquake Engineering and Structural Dynamics*, vol. 34, no. 6, pp. 555–573, 2005.

[71] Z. Li, Y. Zhang, W. Trappe, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *IEEE IPSN 2005*, April 2005.

[72] C. Liu, K. Wu, and T. He, "Sensor localization with ring overlapping based on comparison of received signal strength indicator," in *IEEE Mobile Ad-hoc and Sensor Systems (MASS)*, Oct. 2004.

[73] J. P. Lynch, "An overview of wireless structural health monitoring for civil structures," *Philosophical Transactions of the Royal Society of London, Series A, Mathematical and Physical Sciences*, 2005, (in press).

[74] D. Mallants, B. P. Mohanty, A. Vervoort, and J. Feyen, "Spatial analysis of saturated hydraulic conductivity in a soil with macropores," *Soil Technology*, vol. 10, pp. 115–131, 1997.

[75] D. D. McCrady, L. Doyle, H. Forstrom, T. Dempsy, and M. Martorana, "Mobile ranging with low accuracy clocks," *IEEE Trans. on Microwave Theory and Techniques*, vol. 48, no. 6, pp. 951–957, June 2000.

[76] R. L. Moses, D. Krishnamurthy, and R. Patterson, "An auto-calibration method for unattended ground sensors," in *ICASSP*, vol. 3, May 2002, pp. 2941–2944.

[77] ——, "A self-localization method for wireless sensor networks," *EURASIP Journal on Applied Sig. Proc.*, vol. 2003, no. 4, pp. 348–358, Mar. 2003.

[78] J. V. Murphy, "Everyone wants 20/20 vision when peering into the supply chain," *Global Logistics and Supply Chain Strategies*, June 2000. [Online]. Available: http://www.glscs.com/archives/6.00.20_20.htm?adcode=10

[79] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor network," in *2nd Intl. Workshop on Inform. Proc. in Sensor Networks*, April 2003.

[80] D. Niculescu, "Positioning in ad hoc sensor networks," *IEEE Network*, vol. 18, no. 4, pp. 24–29, July-Aug. 2004.

[81] D. Niculescu and B. Nath, "Ad hoc positioning system," in *IEEE Globecom 2001*, vol. 5, April 2001, pp. 2926 –2931.

[82] B. Ottersten, M. Viberg, P. Stoica, and A. Nehorai, "Exact and large sample ML techniques for parameter estimation and detection in array processing," in *Radar Array Processing*, S. S. Haykin, J. Litva, and T. Shepherd, Eds. Springer-Verlag, 1993, pp. 99–151.

[83] K. Pahlavan, P. Krishnamurthy, and J. Beneat, "Wideband radio propagation modeling for indoor geolocation applications," *IEEE Comm. Magazine*, pp. 60–65, April 1998.

[84] N. Patwari. Localization CRB Code and Channel Measurement repository. [Online]. Available: http://www.eecs.umich.edu/~hero/locate.html

[85] ——. Map-tools online supplement. [Online]. Available: http://www.engin.umich.edu/~npatwari/mnd05

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

[87] N. Patwari and A. O. Hero III, "Location estimation accuracy in wireless sensor networks," in *Asilomar Conf. on Signals and Systems*, Nov. 2002.

[88] ——, "Using proximity and quantized RSS for sensor localization in wireless networks," in *IEEE/ACM 2nd Workshop on Wireless Sensor Nets. & Applications*, Sept. 2003.

[89] ——, "Manifold learning algorithms for localization in wireless sensor networks," in *IEEE ICASSP*, May 2004.

[90] ——, "Adaptive neighborhoods for manifold learning-based sensor localization," in *IEEE SPAWC'05*, June 2005.

[91] N. Patwari, A. O. Hero III, and A. Pacholski, "Manifold learning visualization of network traffic data," in *ACM Mining Network Data (MineNet)'05*, Aug. 2005.

[92] N. Patwari, A. O. Hero III, M. Perkins, N. Correal, and R. J. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Trans. Sig. Proc.*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[93] N. Patwari, R. J. O'Dea, and Y. Wang, "Relative location in wireless networks," in *IEEE VTC*, vol. 2, May 2001, pp. 1149–1153.

[94] N. Patwari, Y. Wang, and R. J. O'Dea, "The importance of the multipoint-to-multipoint indoor radio channel in ad hoc networks," in *IEEE Wireless Comm. and Networking Conf.*, March 2002, pp. 608–612.

[95] M. Perkins, N. S. Correal, and R. J. O'Dea, "Emergent wireless sensor network limitations: a plea for advancement in core technologies," in *IEEE Sensors*, vol. 2, June 2002, pp. 1505–1509.

[96] B. B. Peterson, C. Kmiecik, R. Hartnett, P. M. Thompson, J. Mendoza, and H. Nguyen, "Spread spectrum indoor geolocation," *Journal of the Institute of Navigation*, vol. 45, no. 2, pp. 97–102, Summer 1998.

[97] D. Plonka, "FlowScan: A network traffic flow reporting and visualization tool," in *Large Installation System Administration (LISA) Conference 2000*, Dec. 2000, pp. 305–317.

[98] S. S. Pradhan, J. Kusuma, and K. Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 51–60, March 2002.

[99] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *IEEE Information Processing in Sensor Networks (IPSN)*, April 2004.

[100] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic routing without location information," in *IEEE Mobicom 2003*, Sept. 2003.

[101] T. S. Rappaport, *Wireless Communications: Principles and Practice*. New Jersey: Prentice-Hall Inc., 1996.

[102] P. C. Robert, Ed., *Int. Journal on Advances in the Science of Precision Agriculture*. Kluwer, 1998-2005. [Online]. Available: http://www.kluweronline.com/issn/1385-2256

[103] E. Robinson and A. Quazi, "Effect of sound-speed profile on differential time-delay estimation." *Journal of the Acoustical Society of America*, vol. 77, no. 3, pp. 1086–1090, 1985.

[104] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by local linear embedding," *Science*, vol. 290, pp. 2323–2326, Dec 2000.

[105] C. Savarese, J. M. Rabaey, and J. Beutel, "Locationing in distributed ad-hoc wireless sensor networks," in *ICASSP*, May 2001, pp. 2037–2040.

[106] A. Savvides, L. Girod, M. B. Srivastava, and D. Estrin, "Localization in sensor networks," in *Wireless Sensor Networks*, C. S. Raghavendra, K. M. Sivalingam, and T. Znati, Eds. Kluwer Academic Publishers, 2004.

[107] A. Savvides, C.-C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc networks of sensors," in *IEEE Mobicom*, July 2001.

[108] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the N-hop multilateration primitive for node localization problems," in *Intl. Workshop on Sensor Nets. & Apps.*, Sept. 2002, pp. 112–121.

[109] Y. Shang and W. Ruml, "Improved MDS-based localization," in *IEEE Infocom '04*, March 2004, pp. 2640–2651.

[110] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, "Localization from mere connectivity," in *Mobihoc '03*, June 2003, pp. 201–212.

[111] D.-H. Shin and T.-K. Sung, "Comparisons of error characteristics between toa and tdoa positioning," *IEEE Trans. Aerospace Electr. Systems*, vol. 38, no. 1, pp. 307–311, Jan. 2002.

[112] S. N. Simić, "A learning-theory approach to sensor networks," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 44–49, 2003.

[113] F. Sivrikaya and B. Yener, "Time synchronization in sensor networks: a survey," *IEEE Network*, vol. 18, no. 4, pp. 45–50, July-Aug. 2004.

[114] M. A. Spirito, "On the accuracy of cellular mobile station location estimation," *IEEE Trans. on Veh. Tech.*, vol. 50, no. 3, pp. 674–685, May 2001.

[115] P. Stoica and R. L. Moses, *Introduction to Spectral Analysis*. New Jersey: Prentice-Hall, 1997.

[116] G. Sun, J. Chen, W. Guo, and K. J. R. Liu, "Signal processing techniques in network-aided positioning," *IEEE Signal Processing Mag.*, vol. 22, no. 4, pp. 12–23, July 2005.

[117] N. Sundaram and P. Ramanathan, "Connectivity based location estimation scheme for wireless ad hoc networks," in *IEEE Globecom 2002*, vol. 1, Nov. 2002, pp. 143–147.

[118] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, Dec 2000.

[119] N. Thaper, S. Guha, P. Indyk, and N. Koudas, "Dynamic multidimensional histograms," in *ACM SIGMOD 2002*, June 2002, pp. 428–439.

[120] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, March 2002.

[121] D. J. Torrieri, "Statistical theory of passive location systems," *IEEE Trans. on AES*, vol. AES-20, no. 2, pp. 183–198, Mar. 1984.

[122] H. L. V. Trees, *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, 1968.

[123] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, Apr. 1988.

[124] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," in 34^{th} *IEEE Hawaii Int. Conf. on System Sciences (HICSS-34)*, Jan. 2001.

[125] J. Werb and C. Lanzl, "Designing a positioning system for finding things and people indoors," *IEEE Spectrum*, vol. 35, no. 9, pp. 71–78, Sept. 1998.

[126] H. Xu, V. Kukshya, and T. S. Rappaport, "Spatial and temporal characterization of 60 GHz indoor channels," in *IEEE VTC Fall*, vol. 1, Sept. 2000, pp. 6–13.

[127] N. Ye, X. Li, Q. Chen, S. M. Emran, and M. Xu, "Probabilistic techniques for intrusion detection based on computer audit data," *IEEE Trans. Systems, Man, Cybernetics Part A*, vol. 31, no. 4, pp. 266–274, 2001.

[128] K. Yedavalli, B. Krishnamachari, S. Ravula, and B. Srinivasan, "Ecolocation: A sequence based technique for RF-only localization in wireless sensor networks," in *IEEE IPSN 2005*, April 2005.

ABSTRACT

Location Estimation in Sensor Networks

by

Neal Patwari

Chair: Alfred O. Hero III

In wireless sensor networks, self-localizing sensors are required in a wide variety of applications, from environmental monitoring and manufacturing logistics to geographic routing. In sensor networks which measure high-dimensional data, data localization is also a means to visualize the relationships between sensors' high-dimensional data in a low-dimensional display. This thesis considers both to be part of the general problem of estimating the coordinates of networked sensors. Sensor network localization is 'cooperative' in the sense that sensors work locally, with neighboring sensors in the network, to measure relative location, and then estimate a global map of the network.

The choice of sensor measurement technology plays a major role in the network's localization accuracy, energy and bandwidth efficiency, and device cost. This thesis considers measurements of time-of-arrival (TOA), received signal strength (RSS), quantized received signal strength (QRSS), and connectivity. Extensive RF mea-

surement campaigns were conducted, and the statistical characterization and models which resulted from them are reported. From these models, Cramér-Rao lower bounds on the variance possible from unbiased location estimators are derived and studied. Next, several cooperative location estimation algorithms are developed and presented, for both centralized and distributed implementations. Manifold learning-based algorithms are shown to be particularly effective, in particular, when combined with adaptive neighbor selection methods. Finally, these cooperative localization algorithms are shown to be useful in internet traffic visualization to help show when an anomaly event (such as a port or network scan) is occurring, and to help answer questions about the place and features affected by the anomalous event.