博士論文

Practical wireless localization algorithms for IoT

(IoT システムにおける実用的な無線位置推定アルゴリズム)

グエン　カム　リー

Practical wireless localization algorithms for IoT
IoT システムにおける実用的な無線位置推定アルゴリズム

by

Cam Ly Nguyen
グエン　カム　リー

A Doctor Thesis
博士論文

Submitted to
the Graduate School of the University of Tokyo
on March 19, 2019
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and
Technology
in Computer Science

Thesis Supervisor: Hiroshi Imai　今井浩
Professor of Computer Science

ABSTRACT

Wireless localization is a fundamental task which refers to extracting geo-locating information of an object based on its wireless signals to multiple known devices. It plays important roles in the Internet of Things (IoT) due to its numerous important applications, particularly industrial applications, commercial environments, public safety settings, and everyday life and defense/security systems. Rapid technological development in IoT offers new opportunities for wireless localization. New network systems, applications, IoT equipped devices, etc. produce unconventional localization problems which remain mostly unexplored.

The main purposes of this dissertation are to propose and motivate new localization problems, as well as to develop practical techniques to resolve the problems. It focuses on a proposal of a new localization problem related to scenarios where the device positions are known *a priori*, however, the device IDs are not, and therefore need to be matched using radio frequency methods. The problem, called *WLMP: the wireless localization matching problem*, is motivated through various real-world applications including, but not limited to, disaster prevention wireless sensor networks (WSNs), indoor positioning, and smart lighting and heating systems. We propose several practical techniques to resolve the WLMP in different scenarios and network paradigms. Extensive computer simulations and real experiments in various environments validate that the proposed techniques can achieve high localization accuracy satisfied the requirements of real-world applications.

The dissertation consists of three parts. The first part investigates major wireless localization systems and techniques. It studies the properties and applicability of every technique to derive low-cost techniques that are most suitable for almost all IoT devices. The second part exploits low-cost localization techniques through two different case studies. The first case study illustrates that localization in multi-hop networks can determine coarse location of wireless devices, thus substantiating the validity of low-cost localization. The second case study demonstrates that Received Signal Strength Indicator (RSSI)-based localization, which is one of the low-cost localization methods, can achieve satisfactory accuracy for some specific applications, thus validating the applicability of low-cost localization. The third part, which is the main part, therefore, proposes low-cost localization techniques to resolve the WLMP in different scenarios and network paradigms. It first proposes maximum-likelihood matching localization algorithms called *MLMatch* and *MLMatch3D* for resolving this problem based on RSSI values under mesh network paradigms. The main benefit of these algorithms is that they do not rely on propagation models, i.e. independent of wireless environments. Therefore, they are more practical than traditional localization techniques that requires a time-consuming campaign to survey radio environment to produce an accurate propagation model. Extensive experiments in different environments show that the proposed method can achieve high localization accuracy, especially 100% accuracy in various environments. On the other hand, in the scenario where an accurate propagation model is available, the dissertation further proposes a machine learning-based method called *MLRefine* to improve the accuracy of the above localization algorithms. Experimental results show that MLRefine can improve up to 10% of localization accuracy. Consequently, the WLMP can be resolved efficiently using mesh network models. In practice, however, some sensor networks are not built as mesh networks, i.e. sensor nodes are not connected with each other. Instead, they are connected to a central unit, e.g. a wireless concentrator. Using a mobile central unit, we propose novel methods called *LEMOn* and *LEMOn-M* to resolve the traditional localization problem as well as the WLMP. Various simulations show that the two algorithms achieve a very high localization accuracy even in harsh radio environments. In conclusion, the WLMP can be solved efficiently in the context of different network models.

This dissertation advances the state of art on wireless localization for IoT in several dimensions. First, it defines and promotes unexplored localization problems that are applicable in many real-world applications. Second, it bridges gaps between theory and practice by exploring features that are hard to get practically and substitute by new features that can be obtained easily in practice for a certain network system. It then proposes practical algorithms for resolving defined problems. Besides, the proposed

methods are easily deployable in different environments because they do not rely on a specific environment/hardware. Third, the stability of the proposed methods is not only proved theoretically but also evaluated through various experiments. Last but not least, extensive real experiments in various environments illustrate that the proposed localization methods can achieve enough localization accuracy for real applications, thus validating the practicability of the proposed localization techniques.

Acknowledgements

I express my deepest acknowledgment to people who have contributed to this dissertation and supported me through my Ph.D. Without any of them, this work would not have been completed.

First of all, I offer my sincerest appreciation to my supervisor, Professor Hiroshi Imai, who has supported me throughout my dissertation with his patience and knowledge. This dissertation would not have been possible unless his valuable guidance and encouragement.

I would like to show my gratitude to Dr. Orestis Georgiou and Dr. Yusuke Doi, who have helped me with generating research ideas, paper writing. Without their support, original research ideas would not have been realized. I am grateful Dr. Vorapong Suppakitpaisarn and Dr. Hidefumi Hiraishi, who have helped me to configure the dissertation.

I would like to thank all of my co-authors through my doctoral course: Dr. Orestis Georgiou, Dr. Yusuke Doi, Dr. Vorapong Suppakitpaisarn, Dr. Aftab Khan, Dr. Usman Raza, and Mr. Yuki Yonezawa. Without their valuable support and contribution, our papers would not have been published and this dissertation would not have been completed.

I am grateful to the Directors of Toshiba Research and Development Center for their financial support. I would also thank colleagues who have helped to conduct the experiments for their tireless support.

Many thanks to all the research colleagues especially Mr. Shinya Murai, Mr. Mitsuru Kanda, Mr. Takafumi Sakamoto, Dr. William H. Thompson, Dr. Thomas Hone, Dr. Tuan Thanh Ta, Dr. Ryan Paderna, Dr. Masashi Ito, and Dr. Makoto Higaki for their valuable advice and suggestions on conducting research and paper writing.

I also would like to thank my family and friends especially my husband and my parents who have unconditional helped me to maintain my mental and physical health as well as my inspiration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

With the improvement of wireless communication technology in these decades, the Internet of Things (IoT) is a promising technology and has been widely adopted in constructing smart cities, buildings, and houses. There are numerous key issues in IoT including wireless localization. Localization, which refers to extracting geo-location information of an object, has therefore been well researched and developed. However, with the rapid development of IoT and thus new network paradigms, there are new localization problems and applications that haven't been defined. The main purposes of this dissertation are to propose and to develop new classes of wireless localization that haven't yet been explored. Proposed localization methods can be widely applied in various real-world applications for IoT, and are compatible with many network paradigms.

This chapter investigates perspectives and opportunities for wireless localization for IoT in Sections 1.1 and 1.2. It also highlights issues and challenges in developing wireless localization systems in practice in Section 1.3. It then summarizes the main contributions of this dissertation in Section 1.4. Finally, abbreviations, acronyms, notations and symbols used in this dissertation are summarized in Sections 1.5 and 1.6.

1.1 A Brief History and Future Perspectives of The Internet of Things

The Internet of Things (IoT) is a promising technology which is composed of three paradigms: Internet-oriented, things-oriented and semantic-oriented [1]. Typically, an IoT system consists of low power devices which interact with each other through the Internet. The main goal of IoT is to ensure every device, including sensors, smart-phones, wearable sensors, tablets, transportation system, etc., can connect with each other through a common interface. This allows machine-to-machine (M2M) communication without human intervention [2] and can be attained equipping things with sensors that can connect to the Internet.

The phrase Internet of Things was first coined by Procter & Gamble executive Kevin Ashton in 1999 in the context of supply chain management. However, the actual ideas of connected devices have been conceptualizing for decades. In 1990, a toaster, one of "Things", was connected and controlled via the Internet. Since then, many companies, including industry, utilities, and logistics companies, have begun connecting their machines to each other. The term Internet of Things, where "Things" indicates everything including devices, assets, people, etc., is considered to be innate around 2008 when there were more connected devices than people. Today, the definition covered a wider range of applications including health care, utilities, transport, etc. [3]. Although the definition of "Things" has changed from time to time, the main goal of making things of inter-connecting with each other without the aid of human remains the same. For instance, an agriculture IoT [4], can enable seamless data collection from various sensor types, i.e., cameras, drones, and soil sensors, even in the face of power and Internet outages.

Recently, the number of IoT devices has been increasing exponentially. There are nine billion interconnected devices in 2013 and it is expected to reach 24 billion devices by 2020. This amounts to 1.3 trillion US Dollars revenue opportunities according to the GSMA [5]. This is partly because IoT can enhance the efficiency of various systems including delivery, retail, marketing, health, and smart services, etc [1]. Consequently, IoT is expected to have great potential for future technology.

The increase in the number of IoT devices offers new opportunities for new research and development related to IoT. With the expansion of the hype around IoT, new applications are being developed. For instance, smart house, smart city and smart grids use IoT solutions to solve various problems such as automatic control and management, optimizing electricity consumption and supplement, etc. In most applications, it is necessary to relate the geolocation to the user or the source of data. Consequently, the vision of research and development of localization or navigation using IoT is great.

1.2 Opportunities and Vision of Wireless Localization for IoT

Wireless localization plays an important role in the development of IoT. A localization system extracts geo-location information of an object based on its wireless signals to multiple known devices. The opportunities for localization systems are great because of their numerous important applications, particularly industrial applications, commercial environments, public safety settings, everyday life, and deference/security systems. The expansion of low-cost IoT hardware enables us to track and locate many things, and not only limited to high-cost items. The tracking of people in a building allows businesses to optimize product placements. This also allows people to navigate themselves in an unknown environment. Besides this, localization systems that use low-cost IoT can substitute traditional technologies such as security gates, high-cost positioning and navigation systems.

Wireless localization also plays an important role in wireless sensor networks (WSNs), which are a crucial component of IoT. A WSN uses numerous sensors to execute different functions of environmental sensing such as humidity sensing, temperature sensing, pressure sensing. Sensed data is then sent to a central unit such as a backhaul server, or a data fusion through wireless connections. The central unit is then process received data to make the data available to the user. To make the data useful, it is necessary to relate the stream of sensed data to the location of the corresponding sensor node. Therefore, wireless localization is crucial in most WSNs.

The next-generation of IoT network further brings new challenges and opportunities for wireless localization. With the proliferation of IoT devices, new network models are being researched and developed to enable anything can interconnect [2]. For instance, cooperative communication and networking, which is predicted to be a key component in next-generation wireless networks, has been developed to improve quality of service [6]. Wireless network systems that use mobile-agents, such as UAVs, as relay agents, data collector offers one promising solution to enhance inter-device connectivity [7]. Further, the expansion of IoT systems offers new opportunities for wireless localization through developing new applications. Consequently, it is necessary to establish new research related to localization to target new applications and also to adapt to the change of network models.

In conclusion, the opportunities for the research of localization using IoT are great, ranging from improving conventional techniques to developing totally new techniques/systems.

1.3 Challenges in Developing Practical Wireless Localization Systems

The previous sections expose that the opportunities for wireless localization for IoT are great. However, there are many issues and challenges in developing localization systems in practice. This section highlights the major issues and challenges in developing practical wireless localization systems. It also illustrates our solutions for those issues, thus demonstrating the practicability of our proposed localization systems.

- *Cost effectiveness*: Cost is one of the major challenges to the application of wireless localization. Some localization systems require specific wireless hardware to be equipped with IoT devices, and extra wireless infrastructure. This not only adds cost to the localization services but also make the devices bigger and heavier. As discussed in Section 1.1, IoT devices are often equipped with low-cost and tiny wireless hardware, therefore localization systems that require additional wireless hardware would be inappropriate for most IoT applications. We overcome this issue by using existing infrastructure and wireless hardware.

- *Energy constraint*: Since most IoT devices are constrained on energy consumption, energy efficiency is also a major challenge of wireless localization. Many existing localization systems consume high energy to provide accuracy. For instance, a Global Positioning System (GPS) module embedded to a wireless device consumes about 30 mA at 3.3 V. However, since localization is often not a primary task of most of the user devices, and an IoT device is often constrained on energy, high power consumption can limit the usage of localization. Consequently, it is necessary to constrain the energy consumption of the localization service. To deal with this problem, our localization methods do not use additional hardware that consumes energy.

- *Environment dependency*: Wireless localization is heavily affected by the surrounding environment. The accuracy of a localization system change with the variation of the environment, especially when the surrounding environment changed significantly. Therefore, to enhance accuracy, most localization systems requires a priori measurements to be performed at the same environment of interest, or when the environment changed significantly. This adds manual cost to the system, thus limiting the adoption of localization. Besides, obstacles such as wall also reduce the accuracy of localization systems that assume the existent of a line of sight (LoS) path between two wireless devices. To overcome these issues, we develop localization methods that independent from the environment parameters. Further, we also develop a method that can detect non-line of sight (NLoS) signals.

- *Accuracy*: There is a trade-off between localization accuracy and other features including hardware cost, manual cost, energy consumption as aforementioned. Therefore, cost and energy constraints limit the accuracy of a localization system. To improve the localization accuracy without increasing the cost, we observe new features that can be obtained practically in a specific system and application. These features are used as additional input information to perform accurate localization. Consequently, our localization systems can achieve high localization accuracy that is enough for the requirement of real-world applications.

1.4 Dissertation Contribution and Structure

As discussed in previous sections, developing new wireless localization techniques has great opportunities and potential in the development of IoT. New network systems, applications, IoT equipped devices, etc. produce unconventional localization problems which remain mostly unexplored. Further, there are many issues and challenges in developing practical localization systems used in IoT.

This dissertation proposes and motivates new localization problems, as well as proposes and develops practical techniques to resolve the problems. It focuses on a proposal of a new localization problem related to scenarios where the device positions are known *a priori*, however, the device IDs are not, and therefore need to be matched using radio frequency methods. The problem, called *WLMP: the wireless localization matching problem*, is motivated through various real-world applications including, but not limited to, disaster prevention wireless sensor networks (WSNs), indoor positioning, and smart lighting and heating systems. We propose several practical techniques to resolve the WLMP in different scenarios and network paradigms. Extensive computer simulations and real experiments in various environments validate that the proposed techniques can achieve high localization accuracy satisfied the requirements of real-world applications.

The remainder of this section summarizes our main contributions, organized in this dissertation in the same order as described below.

1.4.1 Investigation of Major Localization Systems and Techniques

In Chapter 2, we investigate major wireless localization systems and techniques. We analysis the advantages and disadvantages of each system and techniques to derive the most suitable techniques for most IoT devices.

We studied from this investigation that:

- Radio frequency (RF) localization is well-suited for many IoT devices.

- Among RF measurement techniques, Received Signal Strength Indicator (RSSI) and range-free measurement techniques are applicable for most IoT devices because they are more economical than other measurement techniques.

- While range-free techniques are suitable for multi-hop, i.e. large scale, networks, RSSI-based techniques are more suited for small-scale networks, for instance, indoor localization.

- The main disadvantage of these techniques is the low localization accuracy.

1.4.2 Exploiting Low-Cost Localization

Chapter 2 demonstrates that low-cost localization is suitable for many IoT devices. We illustrate the validity of low-cost localization by using two different case studies.

Localization in Multi-Hop Networks

The first case study, described in Chapter 3, investigates the localization accuracy of wireless localization method used a hop-count technique which is a range-free technique. This technique is often used in large-scale or multi-hop networks. In this study, we first survey the state-of-the-art localization methods for multi-hop networks to derive high potential methods. We then propose a maximum-likelihood-based multihop localization algorithm called *kHopLoc* that is strong in both isotropic and anisotropic network deployment regions [8]. During an initial training phase, a Monte Carlo simulation is utilized to produce multihop connection density functions [9]. Then, sensor node locations are estimated by maximizing local likelihood functions of the hop counts to anchor nodes. Finally, we perform simulations to evaluate and to compare localization accuracy between the proposed kHopLoc and conventional localization methods. The proposed kHopLoc outperforms conventional methods in varying network configurations and connection link-models.

We studied from this work that:

- Coarse positions of wireless devices can be determined using the proximity measurement technique, which is known as the lowest ranging accuracy among radio frequency measurement techniques. This enables us to assume that localization of wireless devices can be realized using RSSI measurement techniques, which is known as providing better accuracy.

- The proposed kHopLoc enhances accuracy because it is provided full statistical information for the multihop connection probabilities, which is obtained through an off-line training phase. This training phase is conducted using simulation data and Monte Carlo simulation method, thus not requiring extra data collection cost. We thus further extend this idea to improve the accuracy of different localization problems, which is demonstrated in Chapter 6.

Indoor Binary Localization

After demonstrating the validity of RF-based localization, we explore new potential applications related to RSSI-based wireless localization, which is described in Chapter 4. We propose a new approach towards RSSI based wireless localization for scenarios where, instead of absolute positioning of an object, only the information whether an object is *inside* or *outside* of a specific area is required [10]. This is motivated through a number of applications including, but not limited to, *a) security:* detecting whether an object is removed from a secure location, *b) wireless sensor networks:* detecting sensor movements outside of a network area, and *c) computational behavior analytics:* detecting customers leaving a retail store. The result of such detection systems can naturally be utilized in building a higher level contextual understanding of a system or user behaviors. We use a supervised learning method to overcome issues related to RSSI based localization systems including multi-path fading, shadowing, and incorrect model parameters (as in unsupervised methods). Moreover, to eliminate the cost of collecting training data, we employ a detection method called One-Class SVM (OC-SVM) which requires only one class of data (positive data, or target class data) for training. Since this class of data can be obtained during the operation of the wireless network system, our method does not require a priori measurements. We derive a mathematical approximation of accuracy which utilities the characteristics of wireless signals as well as OC-SVM. Based on this we then propose a novel mathematical formulation to find the optimal placement of devices. This enables us to optimize the placement without performing any costly experiments or simulations. We validate our proposed mathematical framework based on simulated and real experiments.

We studied from this work that:

- RSSI-based localization can achieve satisfactory localization accuracy for some specific applications. This motivates us to further develop new applications related to RSSI-based localization.

- Low-cost device placement guidelines to improve the localization accuracy is realizable without performing a priori measurements. This suggests that the accuracy of a localization system can be improved using low-cost methods.

In summary, Chapter 2 derives appropriate localization techniques suitable for most IoT devices. Chapter 3 demonstrates the validity of low-cost RF-based localization through a proposed method called kHopLoc. Chapter 4 illustrates the applicability of RSSI based localization through a proposed method called WiLAD.

1.4.3 The Wireless Localization Matching Problem

After demonstrating the validity and applicability of low-cost RF-based localization, we propose and motivate a new class of wireless localization called *WLMP: Wireless Localization Matching Problem* that has not been yet explored. The problem is applicable in numerous real-world applications related to the Internet of Things (IoT).

The Wireless Localization Matching Problem

In Chapter 5, we propose a new class of wireless localization, called *WLMP: wireless localization matching problem* belonging to the Internet of Things, specifically related to scenarios where the device positions are known *a priori*, however, the device IDs are not [11, 12]. These positions and device IDs, therefore, need to be matched using radio frequency positioning methods, which are more time and cost efficient as compared to manual installation. Immediate examples of real-world applications include but are not limited to smart lighting and heating. We propose maximum-likelihood matching algorithms called MLMatch and MLMatch3D for resolving this problem based on measured RSSI values. Since the search space of node-to-position permutations grows factorially with the number of target devices, we propose several searching methods including Mixed Integer Programming, LP relaxation to reduce computation time. The MLMatch3D algorithm further addresses the problem whereby nodes are located at multiple rooms and/or floors of a building. This algorithm first utilizes a Graph Partitioning method to determine in which room a node is located, followed by MLMatch for finding room specific positions corresponding to each node. In addition, this chapter analyses the stability of these algorithms with respect to different wireless fading models as well as compares the performance of these algorithms in various environments via numerical simulations. Finally, we report on experiments performed in different environments using up to 33 wireless devices in order to demonstrate the problem and validate our results. We demonstrate that the proposed algorithm can achieve high localization accuracy satisfied the requirements of real-world applications.

The main contributions of this work are to:

- Define and motivate the WLMP which is a new class of wireless localization.
- Propose and analytically study the performance of MLMatch and MLMatch3D as candidate solutions to the WLMP.
- Experimentally validate MLMatch and MLMatch3D.

Improved Localization Accuracy: Predicting and Refining RSSI Measurements

After proposing algorithms for resolving the WLMP, we propose a machine learning-based method called *MLRefine* to improve the accuracy of the algorithms. Chapter 6 describes the proposed MLRefine, which uses machine learning methods to model the relationship between accurate values and features extracted from in silico RSSI values [13]. MLRefine then applies the trained model to features extracted from real RSSI measurement values to return a predicted set of refined RSSI values. The idea of MLRefine is actually similar to the idea of the kHopLoc described in Chapter 3, which extracts useful information using Monte Carlo and simulation measurement data for use with real measurements.

The main contributions of this work are to:

- Propose a novel method called *MLRefine* to reduce the magnitude of noisy factors attached to RSSI values.
- Substantiate MLRefine through various simulations and experiments.
- Quantify the impact of MLRefine by combining with state-of-the-art localization estimators.

LEMOn: Wireless Localization for IoT Employing A Location-Unaware Mobile Unit

Chapters 5 and 6 proposed localization techniques and an improved technique to resolve the proposed WLMP under a scenario that sensor nodes connect with each other. In practice, however, there are network systems that sensor nodes connect to a central unit, e.g. a wireless controller or a wireless concentrator, etc., rather than with each other. Considering that the central unit equipped with mobile robot, in Chapter 7, we address the problem of localizing sensor nodes using a mobile wireless unit [14, 15].

Existing localization methods that use a mobile wireless unit assume an accurate knowledge of the location of the mobile unit and a precise propagation model of the actual radio environment. By getting rid of these two requirements, our proposed localization algorithms make mobility-assisted localization far more practical as we do not need to equip the mobile unit with a GPS or run a time-consuming campaign to survey radio environment. LEMOn, i.e. Localization Employing a location-unaware MObile unit, estimates the position of target nodes by using known locations of a small set of fixed anchor nodes while receiving messages sent from a mobile unit from unknown arbitrary locations. LEMOn-M, i.e. LEMOn for the WLMP, on the other hand, solves the WLMP by mapping an arbitrary number of target nodes to the known set of locations. Both algorithms first estimate an inter-node distance using a similarity between RSSI of beacons received from the mobile unit. Conventional location estimators are then employed to localize target nodes with an unknown location. Obvious examples of real-world applications include but are not limited to UAV-assisted wireless sensor works and indoor IoT systems.

The main contributions of this work are to:

- Define and motivate new localization problems that are hybrid between static localization and mobility-assisted localization,
- Propose and analytically study the performance of LEMOn and LEMOn-M as candidate solutions to the above problems,
- Highlight some potential applications related to real-world scenarios.

Finally, Chapter 8 concludes the dissertation. The dissertation advances the state of art on wireless localization for IoT in several dimensions. First, it defines and promotes unexplored localization problems that are applicable in many real-world applications. Second, it bridges gaps between theory and practice by exploring features that are hard to get practically and substitute by new features that can be obtained easily in practice for a certain network system. It then proposes practical algorithms for resolving defined problems. Besides, the proposed methods are easily deployable in different environments because they do not rely on a specific environment/hardware. Third, the stability of the proposed methods is not only proved theoretically but also evaluated through various experiments. Last but not least, extensive real experiments in various environments illustrate that the proposed localization methods can achieve enough localization accuracy for real applications, thus validating the practicability of the proposed localization techniques.

1.5 List of Abbreviations and Acronyms

IoT	Internet of Things
WSN	Wireless Sensor Network
NW	Network
AP	Access Point
GPS	Global Positioning System
IR	Infrared Radiation
RF	Radio Frequency
AoA	Angle of Arrival
ToF	Time of Flight
ToA	Time of Arrival
TDoA	Time Difference of Arrival
RSSI	Received Signal Strength Indicator
UWB	Ultra Wide Band
DSSS	Direct-Sequence Spread Spectrum
MHz	Megahertz
dBm	Decibel-miliwatt
SNR	Signal to Noise Ratio

LoS	Line of Sight
NLoS	No Line of Sight
SDP	Semidefinite Programming
LP	Linear Programming
MIP	Mixed Integer Programming
MILP	Mixed Integer Linear Programming
MIQP	Mixed Integer Quadratic Programming
ILP	Integer Linear Programming
LSAP	Linear Sum Assignment Problems
SVM	Support Vector Machine
OC-SVM	One-Class Support Vector Machine
GA	Genetic Algorithm
BF	Brute Force algorithm
3D	3-Dimensional
CDF	Cumulative Distribution Function
PDF	Probability Density Function
LOOCV	Leave-One-Out Cross Validation

kHopLoc	k-hop Localization
WiLAD	Wireless Localization through Anomaly Detection
WLMP	Wireless Localization Matching Problem
MLMatch	Maximum Likelihood Matching
MLRefine	Machine Learning based Refining technique
LEMOn	Localization Employing a Mobile unit
LEMOn-M	LEMOn for the wireless localization matching problem

1.6 List of Notations and Symbols

$\mathcal{N}$	Gaussian distribution
$\mathcal{N}(0, \sigma^2)$	Gaussian distribution with 0 mean and σ^2 variance
$\boldsymbol{x}$	a vector
$\lvert x \rvert$	absolute-value norm of x
$\lVert \boldsymbol{x} \rVert$	Euclidean norm of vector $\boldsymbol{x}$
$\approx$	approximate
$X \sim \mathcal{N}$	random variable X has the probability distribution $\mathcal{N}$
$\sum$	summation
$\prod$	product
$\mathbb{R}$	real numbers
$\forall$	for all
$A \succeq B$	$A - B$ is a positive semidefinite matrix
lg	$\log_{10}$, logarithm to base 10
ln	$\log_e$, natural logarithm

P_0	reference power
η	path loss exponent
n_i	the i-th wireless node
r	an RSSI value (in dB microwatt)
$r_{i,j}$	RSSI value between nodes n_i and n_j
d	a distance (in meter)
$r_{i,j}$	separation distance between nodes n_i and n_j
$p_{i'}$	the i'-th position

X	a random variable
F_X	complementary cumulative distribution function of X
$\mathcal{A}$	a set
A	the number of elements in set $\mathcal{A}$
$\boldsymbol{V}$	a two-dimensional domain
V	area of domain $\boldsymbol{V}$

Chapter 2

Investigation of Wireless Localization Systems and Techniques

The previous chapter illustrates great opportunities for wireless localization used in IoT. It also discusses major requirements on developing localization systems for IoT. This chapter reviews existing localization systems and techniques. By discussing their advantages and disadvantages, we derive localization techniques appropriate for many IoT devices. This enables us to focus on particular techniques to develop our localization systems presented in the next chapters.

This chapter is organized as follows. In Section 2.1, we introduce an overview of state-of-the-art wireless localization systems. In Section 2.2, we present a review on major measurement Radio Frequency measurement techniques. In Section 2.3, we introduce major RSSI-based location estimators. In Section 2.4, we discuss major sources of localization error, as well as a well-known lower bound on the error. Finally, Section 2.5 concludes the chapter.

2.1 Wireless Localization Systems

This section introduces an overview of state-of-the-art wireless localization systems, which is shown in Table 2.1.4. We focus on the Radio Frequency based localization systems.

2.1.1 Global Positioning System

The Global Positioning System (GPS) [16] is a navigation system to estimate the location of an object that connects to a cellular network. It performs well in many outdoor environments. However, it does yet not work in indoor environments because it relies on Time of Arrival (ToA) measurement technique that disallows the presence of obstacles, e.g. walls, between the satellite and the object. GPS, thus, is infeasible for indoor localization systems. Moreover, a GPS module embedded or attached to each IoT device adds significant production costs and power requirements (about 30mA at 3.3V). Due to constraints on energy and cost in many IoT systems such as wireless sensor networks (WSNs), GPS is not suited for those systems.

2.1.2 Ultrasound Based Localization System

An Ultrasound-based localization system estimates the position of an object through ultrasound signals emitted from the object to multiple location-known receivers [17]. It makes use of the Time of Flight (ToF) of the ultrasound signals from the object to each receiver to calculate the distance between them. Then the position of the object is estimated through these distances and known-locations of receivers. Ultrasound signals cannot travel through walls but reflect off most obstructions. Ultrasound-based localization in indoor systems is termed as high accuracy systems having centimeter level location accuracy due to the slow propagation speed of signals, which is about 340 m/s. However, these systems require specific hardware, i.e. ultrasonic transducers, which are not always equipped with ordinary IoT devices, especially cheap wireless sensor nodes.

2.1.3 Infrared Radiation Based Localization System

Infrared Radiation (IR) localization systems are one of the most common localization systems, which make use of the scanning sweep method [18]. They use infrared light pulses to locate objects in indoor. Typically, there are IR landmarks deployed at many places in a building. When an IR tag or sensor pulses, it is read by the IR landmarks. Then, the position of the IR tag is determined using line-of-sight (LoS) pulses and known locations of the landmarks. The main advantage of IR based systems is that the devices are small, lightweight, and easy to carry out. However, the IR based indoor system has expensive system hardware and maintenance cost. Similar to ultrasound localization systems, IR based localization systems require specific hardware, i.e. IR emitter and receiver, thus being inappropriate for many IoT systems.

2.1.4 Radio Frequency Based Localization System

Radio Frequency (RF) localization systems are the most common localization systems for IoT networks, which make use of properties of wireless signals [19]. Wireless devices in a network system send signals to each other. A localization system uses either Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), or Angle of Arrival (AoA) of these signals to determine positions of location-unknown devices. We will discuss detailed properties of these measurement techniques in Section 2.2. The main advantage of RF based systems is to determine the position of almost every wireless devices without installing extra software or embedding extra hardware. Beside this, signal strength based systems do not require line-of-sight (LoS). Therefore, RF-based localization techniques are suitable for almost IoT networks. The main disadvantage of RF based systems is, however, low accuracy. Signal attenuation of the static environment like a wall, furniture, and movement of people caused large localization error. We will further discuss advantages and limitations of RF-based localization techniques in Sections 2.2 and 2.4.

2.2 Measurement Methods Used for RF Based Localization

A Radio Frequency (RF) based wireless localization system based on various measurement methods. This section presents a review of major measurement RF measurement techniques as shown in Table 2.2. Our main focus is put on the Received Signal Strength Indicator (RSSI) measurement techniques.

There are four major measurement methods, including Angle of Arrival (AoA) measurements, propagation time measurements, Received Signal Strength Indicator (RSSI) measurements and connectivity measurements [20]. Table 2.2 summarizes the characteristics of these measurements. As every measurement method has advantages and disadvantages, appropriate methods should be chosen considering the network architecture, wireless hardware, and requirements of target

System	Technique	Remarks	Accuracy	Applicability
GPS	Use signals' travel time from satellites and a wireless object to determine the position of the object.	Satellite-based positioning. High production cost and power consumption	6 m- 10 m	Appropriate for outdoor.
Ultrasound	Use travel time and/or angle of ultrasounds from an emitter to receivers to determine the position of the emitter.	Sensitive to environmental. Require specific hardware.	3 cm- 1 m	Appropriate for both indoor and outdoor.
Infrared radiation	Use travel time of infrared light pulses from an IR tag to IR landmarks to determine the position of the tag.	Short range detection. Require specific hardware.	1 m- 2 m	Appropriate for indoor.
Radio Frequency	Use either strength, travel time, or angle of signals between wireless devices to determine their positions.	Compatible to every wireless network. No specific hardware.	In meters	Appropriate for both indoor and outdoor.

Table 2.1: Comparison of common localization systems.

applications. The remainder of this section briefly discusses these techniques and their limitations in different IoT applications.

2.2.1 Angle of Arrival Measurements

The Angle of Arrival (AoA) measurement method refers to the technique for estimating the direction of propagation of a radio-frequency wave incident on an antenna array or determined from maximum signal strength during antenna rotation [21]. It is also known as the bearing measurement or the direction of arrival measurement. The AoA measurement allows us to calculate the angle at which the signal arrives from a target node, i.e. the transmitter, to an anchor node, i.e. the receiver. Then, the target node is likely to lie in the straight line having a certain angle from the anchor node. Thereby, the locations of target nodes can be estimated using at least two anchor nodes. A small error in measurement could cause a large localization error. Measurement is further sophisticated by the presence of the multi-path effect of the deployed environment. A multi-path component from the transmitted signal may appear as a signal coming from an entirely different direction and consequently causes a very large error in measurement accuracy [22]. In addition, using NLoS measurements, which often appear in environments having many obstacles, as input in the localization, a great additive localization error will arise [23]. Therefore, the AoA measurement method is of limited interest in the localization of IoT networks and consequently being an inappropriate choice of localization measurements.

2.2.2 Propagation Time Measurements

Propagation time measurement methods refer to the technique for estimating the time of propagation of a Radio Frequency signal traveling between two devices [24]. These methods can be divided into two categories: Time of Flight (ToF) measurement methods and Time Difference of Arrival (TDoA) measurement methods [25].

Time of Flight

Time of Flight (ToF) uses the signal's travel time from a transmitter, which is usually a target node, to a receiver, which is usually anchor node, to measure the distance between them [26]. Given flight time t, the distance between the two wireless nodes can be calculated by $d = c \times t$, where c is the speed of RF signal (approximately equals to the speed of light, 3×10^8 m/s). In order to properly localize with ToF, there must be at least three anchor nodes. When the distances from different anchors are known, the location of the target can be determined using either multilateration methods, maximum likelihood methods, or least-squares methods [25], which estimate the position of a target node from distance measurements to known anchor nodes. ToF can be calculated using either one-way Time of Arrival (ToA) or two-way ToA. One-way ToA measures the Time of signal pulses transmitted by the transmitter arrival at receivers to determine their ToF. It, therefore, requires perfect synchronization between the target node and anchor nodes. This technique is more common in cellular networks since the target nodes are typically synchronized to anchor nodes, i.e. base stations. On the other hand, two-way ToA measures ToA of signal pulses from a node to another node and back to determine the ToF of these signals. Therefore, it does not require synchronization between nodes. However, the main disadvantage of

AoA	Uses the angle between the target node and the anchor node to construct the line between them, then uses their intersection to estimate the location of the target node.	Requires new hardware, i.e. antenna arrays, resulting in additional costs and larger node sizes.	Inappropriate for systems using tiny nodes due to large size.
One-way ToA	Uses distance between target and anchors derived from the signal's travel time	Needs perfect synchronization between target nodes and anchor nodes. Needs high-resolution clocks.	More appropriate for cellular networks
Two-way ToA	Uses distance between target and anchors derived from signal's travel time	Needs high-resolution clocks and Ultra Wide Band (UWB) techniques	Inappropriate for network systems that do not use UWB technology
TDoA	Uses difference between ToAs in multiple anchors	Needs perfect synchronization between anchors	
RSSI	Inter-node distance is estimated based on the attenuation of the received signal strength	Needs an accurate propagation model for reliable distance estimation.	Appropriate for low-cost IoT networks that require a coarse estimate.
RSSI profiling	Uses fingerprint RSSI information of measured radio signal at different geographical locations.	Requires a training stage to obtain a database. The database may be unreliable if the environment changes.	Mostly used in wireless local area networks.
Range-free	Uses attributes of network and devices, for instance, network connectivity, geometric relationship, and hop count between nodes, etc.	Requires network connectivity information	Mostly used in large-scale wireless sensor networks.

Table 2.2: Comparison of common measurement techniques used for Radio Frequency based localization.

this method is that signal pulses need to be sent consecutively between nodes which may cause latencies for applications where devices move quickly.

Time Difference of Arrival

Time Difference of Arrival (TDoA)-based localization techniques use a hyperbolic method to estimate the location of a target node [27]. Similar to ToF techniques, TDoA uses the travel time from a transmitter, which is usually a target node, to receivers, which are usually anchor nodes, to measure distance. The basic idea of these techniques is to estimate the location of the transmitter by means of the difference in arrival time of the signal at separated multiple receivers. In order to calculate the time difference, the synchronization between the receivers is required [28]. However, the synchronization between the transmitter and receivers is not required because TDoA techniques do not use the distance between them.

Although the principle of time based localization is simple, measuring ToF, TDoA for wireless signals is extremely challenging in various dimensions. Since the speed of RF signals is extremely fast, i.e. one nanosecond translates to 0.3 meter, a time based localization system requires fine resolution of the timestamps. Two mostly used technologies for ToF systems are Ultra Wide Band (UWB) impulse [29] and Direct-Sequence Spread Spectrum (DSSS) [30], which provide very fine time resolution. Besides this, the localization system need to process the delay introduced by the hardware. In addition, large errors occur when the Line of Sight (LoS) signals are undetectable [31]. Due to the these challenges, ToF, TDoA based localization techniques are inappropriate for many IoT systems, for instance systems that use only ordinary low-cost wireless hardware or systems that require low-power consumptions.

2.2.3 Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) uses the received power level of a signal sent from a transmitter, which is measured at the receiver, to determine the distance between them. An RSSI value, which is a measurement of the received signal power, is available in most of the wireless receivers. It can be used for distance estimation due to the property of signal attenuation over distance [32]. Therefore localization of unknown target nodes can be realized by means of RSSI measurements. For instance, in free space propagation, the relationship between an RSSI value r and transmission distance d is expressed through Friis free-space propagation model as follows [33].

$$r = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2}, \tag{2.1}$$

where P_t is the transmitted power, G_t is the transmitted antenna gain, G_r is the receiver antenna gain and λ is the wavelength of the transmitter signal in meters. However, this equation is an over-idealization. In practice, the signal attenuation due to path loss is often more complicated than that suggested by (2.1) due to surrounding environments. For instance, the singular path loss function in (2.1) which diverges at $d = 0$ can replaced by multi-slope and non-singular alternatives in some specific environments [34].

In the following, we introduce two representative propagation models that are used in this dissertation.

Log-Distance Path Loss Model

The log-distance path loss model, which is an extension to Friis free space model, is widely used and adopted by the scientific community [35]. While Friis model is limited to free space without any obstruction LoS path between the transmitter and the receiver, log-distance path loss model is a representative propagation model for most environments [35]. Received power observed at the receiver, denoted as r, is modeled as follows (in dBm).

$$r = P_0 - 10\eta \log_{10}(d) + X \tag{2.2}$$

where d is the separation distance in meter between the transmitter and the receiver, P_0[dBm] is the reference power in dB milliwatts at the reference distance of 1 meter from the transmitter, η is the path loss exponent that measures the rate at which the received signal strength decreases with the distance. P_0 is environment and hardware dependent, while η is environment dependent. For instance, the value of path loss exponent η depends on the measured environment. For instance, it is approximately 2 in free space, and from 3 to 5 in shadowed urban cellular radio [36]. X is a random variable characterizing multi-path fading, and shadowing effects, etc. X is often assumed to follow a zero-mean log-normal distribution, namely Gaussian distribution in dB. In other words, $X \sim \mathcal{N}(0, \sigma^2)$, where σ^2 is the variation of random variable X. Since this model is commonly used in literature in most environments [35], we will use it in our analysis and evaluation simulations in the rest of this dissertation.

Non-Singular Path Loss Model

While log-distance path loss model encapsulated in Formula (2.2) is commonly used to describe the propagation loss, in practice, propagation is often more complicated than the log-distance model. For instance, the singular path-loss function in (2.2) which diverges at $d = 0$ can be replaced by multi-slope and non-singular alternatives [34]. Besides, the fluctuation of the received power, which is denoted as random variable X in (2.2) does not always follow Gaussian distribution model. For instance, the short-term signal variation can be described by several other distributions such as Hoyt, Rayleigh, Rice, Nakagami-m, and Weibull [37]. In this dissertation, we employ the following model in our evaluation simulations to verify the robustness of our proposed localization methods. In this model, propagation path loss follows a non-singular model and the random variable follows a Rayleigh fading model. The RSSI values $\bar{r}$ relates to distance d as follows.

$$\bar{r} = P_0 - 10 \log_{10}(\epsilon + d^{\eta}) + \mathcal{X} \tag{2.3}$$

where $\epsilon > 0$, and $\mathcal{X}$ is a random variable with density

$$\begin{aligned} f_{\mathcal{X}}(x) &= \mathbb{P}[\mathcal{X} = x] = \frac{\mathrm{d}}{\mathrm{d}x}\mathbb{P}\big[10 \log_{10} |h|^2 \leq x\big] \\ &= \lambda 10^{x/10} \exp\Big(- \lambda 10^{x/10}\Big) \frac{\ln 10}{10} \end{aligned} \tag{2.4}$$

where we have assumed that the channel gain $|h|^2$ is an exponentially distributed random variable with mean $1/\lambda$. Recent indoor measurements at 2.4GHz [38] have confirmed the above model.

The major merits of RSSI-based localization techniques are the unnecessity of specific hardware and low power consumption. However, the major demerits of these methods are the requirement of many prior measurements to be done after performing localization because they are highly sensitive to environments.

2.2.4 Range-Free Localization Techniques

Range-free schemes realize localization through the attributes of network and devices, for instance, network connectivity, geometric relationship, and hop count between the target node and anchor nodes, etc [39, 40]. There are various approaches related to connectivity-based techniques such as Distance Vector hop (DV hop) [39], centroid algorithm [19], multi-hop techniques [41], and Approximate Point in Triangulation test (APIT) [39]. In DV hop localization, a target node first counts the minimum numbers or hop to the anchor nodes, then calculates the distance to anchor nodes using the hop-count, and finally localizes itself using estimated distances and known-location of anchor nodes. Centroid algorithm, which is proximity based scheme, outputs the averaged coordinates of multiple nearby anchor nodes as the coordinates of the target node. Multi-hop techniques, which are centralized schemes, first compute a connectivity graph and then use a Multidimensional Scaling (MDS) method to perform localization. APIT makes use of the area-based scheme to perform localization by dividing the are into triangular regions between anchor nodes. It then estimates each target node's position by determining whether the target node is inside or outside the triangle regions.

Similar to RSSI-based localization techniques, the main advantages of range-free schemes are the specific hardware-free and low power consumption. However, the main disadvantage of these schemes is the low localization accuracy. Besides, these techniques are only suitable for high-density multi-hop networks, i.e. large-scale networks, but are inappropriate for one-hop networks.

In conclusions, RSSI-based localization techniques are well suited for many IoT applications because they satisfy the requirements of many IoT systems on using ordinary low-cost hardware, and on low power consumption. However, the aforementioned disadvantages of RSSI-based techniques should be alleviated so as to enhance applicability in practice.

2.3 RSSI-Based Location Estimators

An RSSI-based location estimator determines the geo-location of unknown wireless nodes (called *target nodes*) using RSSI values between nodes and provided locations of known wireless nodes (called *anchor nodes*). This section presents a review on major state-of-the-art RSSI-based location estimators shown in Table 2.3. RSSI-based location estimators can be divided into two schemes: RSSI profiling-based schemes and propagation model-based schemes. As every localization method has advantages and disadvantages, appropriate methods should be chosen considering the network architecture, environments, and requirements of target applications.

2.3.1 RSSI Profiling-Based Localization Techniques

RSSI profiling-based techniques [42], which are also known as supervised techniques or fingerprinting techniques, estimate the location of a wireless device by means of the map of RSSI measurements. The map is built through either a real-time phase by using some sniffing devices [43], or a priori measurements [42]. The target node, whose position is unknown, determines its position by referring

Method	Summary	Advantages	Disadvantages	Applicability
RSSI-profiling	Estimates the location of a wireless device by means of the map of RSSI measurements	High accuracy	Requires a prebuilt RSSI map	Most suited for small indoor environments
Propagation model-based, non-cooperative	Uses RSSI and propagation model to estimate distances between the target node and anchor nodes, then locates the target node	Does not require a prebuilt RSSI map & Low computation complexity	Low accuracy & Requires an accurate propagation model	Most suited for tracking problem in both indoor and outdoor environments
Propagation model-based, cooperative	Uses RSSI and propagation model to estimate distances between almost all sensor nodes, then locates all target nodes simultaneously	Requires smaller number of anchor nodes than non-cooperative methods	Requires an accurate propagation model & High computation complexity	Most suited for WSNs, IoT networks.
Mobility-assisted	Uses a wireless mobile unit, which is aware of its location (e.g. equipped with GPS), to assists the target wireless nodes in localizing themselves	Usable in spare networks & Does not require anchor nodes	Requires accurate propagation model & Requires accurate location information of the mobile unit	Most suited for UAV-enabled WSNs

Table 2.3: Comparison of common RSSI-based location estimators.

to the RSSI map. It calculates RSSIs to anchor nodes and then matches the position from the corresponding map by means of a supervised machine learning techniques such as k nearest neighbors (kNN) [44], support vector machine (SVM) [45].

RSSI profiling-based techniques can alleviate multi-path shadowing, fading and NLoS effects caused by obstacles, thus enhancing accuracy. However, the main drawback of these methods is the requirement of expensive a priori measurements to build an RSSI map. For instance, sniffing devices or reference points should be distributed throughout the coverage area. Therefore, these techniques are suited for localization in small indoor environments but are inappropriate for many low-cost IoT applications such as applications that do not allow human intervention.

2.3.2 Propagation Model-Based Localization Techniques

Propagation model-based techniques determine the location of target nodes by means of the propagation model of wireless signals rather than search for the best-fit location from the RSSI database. The distance between two wireless nodes can be obtained from the RSSI value between them using a given propagation model and the model parameters. The model and its parameters are often obtained through a priori measurements.

A location estimator is then applied to determine the location of the target nodes using estimated distances and known-locations of anchor nodes. There are various location estimators, which can be broadly categorized into three groups: Non-cooperative methods, cooperative methods, and mobility-assisted methods [46] [47, 48]. In non-cooperative localization, distance measurements are made only between anchor nodes and target nodes. Each target node estimates its distance to the anchor nodes using measured RSSI values, then use localization algorithm such as multilateration [49] to locate itself. These methods are suitable for the target tracking problem, or server-client network models.

Cooperative localization methods, where distance measurements between target nodes are also made, estimate all node positions simultaneously rather than localizing each target node individually. These methods enhance localization accuracy compared to non-cooperative techniques as they use more measurements. These methods are, therefore, more suitable for wireless sensor mesh networks [50]. There are numerous localization algorithms such as Multidimensional Scaling (MDS) [51], Semidefinite Programming (SDP) [52], stochastic optimization (e.g. Simulated Annealing (SA) [53]), etc. On the other hand, mobility-assisted localization methods use a wireless mobile unit, which is aware of its location (e.g. equipped with GPS), to assists the location-unknown wireless nodes in localizing themselves [47, 48]. A mobile unit could be an Unmanned Aircraft Vehicle (UAV), a drone or a ground vehicle, etc. A wireless node could be a wireless sensor node or any device that equipped with a wireless hardware. A wireless node is often assumed having a fixed position, i.e. its position does not change frequently. These mobility-assisted localization methods are advanced to the aforementioned localization methods that reliable inter-node distance information is hard to obtain [48]. However, obtaining accurate positions of the mobile unit is a big challenge. The main advantage of propagation model-based techniques is that they do not require a prebuilt RSSI map, thus reducing manual cost. However, these techniques encounter the following challenges. First, in practice, RSSI values often do not strictly follow a specific propagation model. These values fluctuate due to the surrounding environments, especially those having irregular obstacles inside. This induces inaccuracy on distance measurement. Second, the estimation of model parameters is also a very difficult task. For instance, the path loss exponent in the log-distance model formulated as (2.2) is environment dependent, which is usually close to 2 for free space wireless propagation and from 3 to 5 in shadowed urban areas [36]. Besides this, the reference power, P_0 is a function of the calibration of both the transmitter and receiver. Transmitted powers and received powers vary from device to device. This is even truer in IoT systems where each device is equipped with a different type of wireless hardware. Wireless nodes might be designed to measure and report their own calibration data to each other. This, however, complicates the hardware design. Consequently, to obtain accurate model parameters, a priori measurements are often conducted at the environment of interest [20]. This process raises the manual cost and is challenging when wireless nodes are deployed in inaccessible areas such as post-disaster areas.

2.4 Understanding Localization Error of RSSI-Based Localization

The previous section showed that propagation model-based localization techniques are suitable for most WSNs, which are crucial parts of IoT. This section describes major sources of localization error and introduces the famous Cramer-Rao lower bound of localization error covariance.

2.4.1 Major Sources of Error

In practice, multi-path fading and shadowing are two main factors caused localization error. Since the distance between a pair of wireless nodes is typically unchanged, multiple measurements between them may not reduce these effect unless using specific hardware or techniques. For instance, multiple antennas can reduce the effect of shadowing [54], a spread-spectrum method can reduce the effect of frequency-selective fading [50]. However, an ordinary wireless node is not always equipped with those hardware devices. Typically, the standard deviation of X, denoted as σ_X, (cf. (2.2)) which characterizes the above effects, is as low as four and as high as 12 [50]. This value causes large ranging error, which is illustrated through the following calculation.

A Ranging Method and Ranging Error

Using the log-distance path loss model described by Formula (2.2), the distance between the transmitter and the receiver can be estimated via the inversion of the formula as follows.

$$\hat{d} = 10^{\frac{P_0 - r}{10\eta}} \tag{2.5}$$

Assuming that $X \sim \mathcal{N}(0, \sigma_X^2)$, the root mean squared ranging error ε_X due to X can thus be calculated as follows.

$$
\begin{aligned}
\varepsilon_X &= (\mathbb{E}[(\hat{d}-d)^2])^{1/2} = \left(\mathbb{E}(d^2(10^{\frac{X}{10\eta}}-1)^2)\right)^{1/2} \\
&= d\Big[\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-X^2}{2\sigma^2}} (e^{\frac{X\ln 10}{10\eta}}-1)^2 \mathrm{d}X\Big]^{1/2} \\
&= d\Big[\frac{1}{\sqrt{2\pi}\sigma}\big(e^{(\frac{\sqrt{2}\sigma\ln 10}{10\eta})^2}\int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}(X-\frac{2\sigma^2\ln 10}{10\eta})^2}\mathrm{d}X \\
&\quad - 2e^{(\frac{\sigma\ln 10}{10\sqrt{2}\eta})^2}\int_{-\infty}^{\infty} e^{-\frac{1}{2\sigma^2}(X-\frac{\sigma^2\ln 10}{10\eta})^2} + e^{-\frac{X^2}{2\sigma^2}}\,\mathrm{d}X\big)\Big]^{1/2} \\
&= d\big[e^{(\frac{\sqrt{2}\sigma\ln 10}{10\eta})^2} - 2e^{(\frac{\sigma\ln 10}{10\sqrt{2}\eta})^2} + 1\big]^{1/2}
\end{aligned}
\tag{2.6}
$$

Thus ranging methods (2.5) are subject to errors that increase exponentially with the signal fluctuations. For instance, considering two nodes with a separation distance of 100m, path loss exponent $\eta = 2$, when $\sigma_X = 3$, $\sigma_X = 6$ and $\sigma_X = 12$ the expectation of the instantaneous distance estimation error is averagely 38 m, 102 m, and 642 m, respectively. In another environment, $\eta = 3$, and $\sigma_X = 10$, the average error is around 125 m.

2.4.2 Cramer-Rao Lower Bound of Localization Error Covariance

As the wireless localization problem is formulated as a parameter estimation problem, the localization error is often analyzed through estimation theory. Cramer-Rao Lower Bound (CRLB) [55] is the most famous estimation technique. We briefly introduce the intuition of CRLB analysis which is detailed in [50].

To calculate a CRLB, we need the statistical model of the random measurements. We denote the model by $g(Y|\theta)$, where Y is the random measurement and θ are the parameters to be estimated from the measurements. Any unbiased localization estimator $\hat{\theta}$ must satisfy the following.

$$
Cov(\hat{\theta}) \geq [E\{-\bigtriangledown_\theta (\bigtriangledown_\theta \ln g(Y|\theta))^T\}]^{-1} \tag{2.7}
$$

where $Cov(\hat{\theta})$ is the covariance of the estimator, $E[\cdot]$ is expected value, $\bigtriangledown_\theta$ is the gradient operator of the vector θ, superscript ^{T} is transpose, and $\ln g(Y|\theta)$ is the log-likelihood function.

The random measurements Y is often characterized by the random variable X in (2.2).

The CRLB for RSSI-based wireless localization mainly depends on the following factors:

- Positions of the wireless nodes,
- Characteristic of the measurement environment, i.e. the path loss exponent η, and
- Standard deviation of the measurement noise σ.

The CRLB analysis suggests that for a certain deployment of a network system, the lower bound on error covariance is essentially related to the ratio η/σ, where η and σ are, respectively the path loss exponent and the standard deviation of the random variable X (cf. Equation (2.2)). Intuitively, if σ is large, i.e. the log-likelihood function is broad, then the variance of the localization error is large. Conversely, if σ is small, i.e. the shape log-likelihood function is sharp, then the variance of the localization is small.

CRLB analysis gives insights on studying parameters elements that affect the localization error. Therefore, it is useful to optimize the settings of the IoT network in order to achieve the best localization accuracy. CRLB analysis, however, has some limitation. First, it can only bound the variance of localization error after the localization has been executed. Second, it does not consider the inaccuracy of propagation model parameters, e.g. parameters P_0, η. It, thus, does not return the mean error of the estimator, especially when estimated values of these parameters are incorrect. In practice, however, it is crucial to estimate the localization error before the execution of the location estimation, considering all possible inaccuracies.

2.5 Conclusions

In this chapter, we have presented a brief overview of wireless localization systems and techniques with the focus on techniques that are suitable for most IoT devices. We first discussed major state-of-the-art localization systems, including GPS, ultrasound-based systems, Infrared Radiation (IR) based systems, and Radio Frequency (RF) based systems. Based on their characteristics, we concluded that RF-based localization is suitable for numerous IoT devices. We then introduced RF measurement techniques and their properties including Angle of Arrival (AoA), Time of Flight (ToF), Time Difference of Arrival (TDoA), Received Signal Strength Indicator (RSSI), and range-free techniques which are often used to realize wireless localization. Since almost all IoT devices equipped with ordinary wireless hardware are able to measure RSSI information and connectivity information, range-free localization and RSSI-based localization is low-cost compared to methods using AoA or time-based measurements. The main disadvantage of these low-cost methods is their low localization accuracy. In the next two chapters, we will exploit low-cost localization through two different case studies on localization: one uses connectivity information and the other one uses RSSI measurements.

Chapter 3

Exploiting low-cost Localization: Maximum Likelihood-based Multi-hop Localization in Wireless Sensor Networks

The previous chapter introduces existing localization systems and techniques. Besides some localization techniques that can achieve high accuracy using specific hardware and technologies, there are low-cost localization methods that do not require any specific hardware or technologies. This chapter and the next demonstrate that localization techniques that do not require these hardware devices and technologies can achieve accuracy that meets the application needs. This validates the practicality and the applicability of low-cost localization techniques. This chapter illustrates that a range-free localization technique, which is known as low-accurate technique and is a low-cost technique, can produce coarse localization. Besides, the next chapter demonstrates that another low-cost localization techniques using RSSI can achieve accuracy that meets some specific localization needs.

This chapter presents the validity of low-cost localization through a proposed localization algorithm called *kHopLoc* which stands for k hop localization. kHopLoc is a maximum-likelihood-based multihop localization algorithm for use in Wireless Sensor Networks (WSNs) that is strong in both isotropic and anisotropic network deployment regions. Compared to other multi-hop localization algorithms, the proposed kHopLoc algorithm can provide coarse locations of target nodes, and achieves higher accuracy in varying network configurations and connection link-models.

The rest of this chapter is organized as follows. In Section 3.1, we introduce multi-hop localization techniques and related state-of-the-art works. In Section 3.2 we describe the network and system model. In Section 3.3 we present the details of our proposed kHopLoc algorithm. In Section 3.4 we evaluate the performance of kHopLoc through numerical simulations. Finally, Section 3.5 concludes the chapter.

3.1 Introduction

Wireless Sensor Networks (WSN) are composed of a set of spatially distributed wireless nodes, with sensing and transceiving capabilities, tasked with monitoring physical or environmental conditions, such as temperature, sound, pressure, radiation, etc. Data collected is then wirelessly passed through the network to a main gateway for storing and processing. WSNs are essential for applications such as environmental monitoring, target tracking, disaster relief and rescue operations [56, 57]. Nowadays, they are also becoming an indispensable part of smart technologies with applications in smart cities and smart buildings [58].

In view of the Internet of Things future vision, almost every device will soon have transceiving capabilities, be packed with sensors, connected to a network and producing huge data sets. As is often the case however, location information is vital for the insightful processing of this data. GPS modules may be embedded to each sensor node enabling it to autonomously discover its location both accurately and on demand, however this does not come without a cost to the manufacturer (and hence the user) and the node's power source - about 30mA at 3.3V. Moreover, in some extreme instances such as sand storms and blizzards, or simply when operating indoors, satellite signals cannot reach the sensor nodes.

To alleviate such problems, cooperative schemes have been developed to estimate the locations of sensor nodes with the assistance of nodes which have perfect location information [50,59]. The nodes whose locations are known and the nodes whose locations are unknown are usually called *anchor nodes* and *target nodes* respectively and the localization techniques can be broadly classified into two schemes: *range-based* and *range-free* schemes. Range-based schemes [60] assume that the distance or angle between anchor nodes and target node can be measured based on signal measurements such as received signal strength indication (RSSI), Time of Arrival (ToA), or Angle of Arrival (AoA). In large-scale WSNs where signal range is limited however, range based schemes typically require a lot of anchor nodes to produce accurate results. On the other hand, range-free schemes [39] estimate inter-node distances based on hop count information, thus all target nodes can be localized with fewer anchor nodes.

Conventional range-free approaches [61–63] usually consider isotropic WSNs where sensor nodes are uniformly distributed in a regular region (e.g. a square domain), thus the distance between two nodes is assumed to be proportional to their hop count. The celebrated DV-hop algorithm [61] estimates the average one-hop distance, and then multiplies this by the hop count to at least three anchor nodes before trilaterating. Although improved DV-hop algorithms have been suggested in the literature [62,63], performance gains have been limited, even more so when used in anisotropic WSNs where factors such as irregular radio propagation, obstacles, nonuniform node distributions degrade the hop-distance proportionality assumption. DV-hop-like variants which are anisotropic network compatible have also been proposed [64,65], the main idea usually being to reduce the estimation error by reducing the effect of unreliable anchors. Thus, these algorithms require an increased number of anchor nodes, possibly as many as range-based schemes. Other approaches [66, 67] attempt to use approximate shortest paths to reduce the effect of anisotropic networks, yet [67] under performs in irregular-shaped regions as shown in [66] which comes with large communication and computational overhead.

We propose a *maximum likelihood based multihop localization algorithm* called *kHopLoc* which achieves good performance in both isotropic and anisotropic WSNs. The algorithm first runs a training phase during which a Monte Carlo simulation is utilized to produce accurate multihop connection probability density functions (described later). In its second phase, the algorithm constructs likelihood functions for each target node based on their hop counts to all reachable anchor nodes which it then maximizes to produce localization information. Unlike most DV-hop algorithms which use only first order statistics, the proposed kHopLoc algorithm generates and uses the full multihop density distributions (even for anisotropic networks) thus constructing accurate likelihood functions and in turn localization results. In addition, our algorithm's communication cost is about half of most DV-hop-like algorithms and computational cost is much smaller than [66] for example.

3.2 Network Definitions and System Model

Consider a WSN of N total sensor nodes, M of which are anchor nodes (i.e. have perfect location information). The remaining $N-M$ sensors are target nodes (i.e. locations are unknown). All nodes are randomly distributed in some subset of $\mathbb{R}^2$ and are equipped with isotropic antennas. Due to phenomena such as fading and multi path, the communication model adopted may not be well described by the simplistic disk model (where two nodes connect if they are within a finite range of each other). Instead, we adopt a *random connection model* where nodes connect with a distance dependent probability while also accommodating for environmental parameters such as path loss exponent [68]. Namely, we consider two such communication models: *Rayleigh fading* communication model and *Quasi Unit Disk Graph* (QUDG) communication model [69]. Moreover, we consider isotropic networks in which the sensor nodes are distributed in a square, and anisotropic networks where sensor nodes are deployed in irregular shaped regions.

3.2.1 Rayleigh Fading Communication Model

The *pair connectedness function* H defined as the probability that two nodes are directly connected. One way of formulating H is thus the complement of the information outage probability with respect to a mutual information threshold ϑ. For a narrow band transmission subject to small-scale fading

$$H = P\left(\log_2(1 + \text{SNR} \cdot X) > \vartheta\right) = F_X\left(\frac{2^{\vartheta} - 1}{\text{SNR}}\right), \tag{3.1}$$

where X is a random variable defining the gain of the wireless channel between the two nodes, F_X is its complementary cumulative distribution function (CCDF), and SNR is the long-term average received signal-to-noise ratio. Since the nodes are equipped with isotropic antennas, the SNR $\propto d^{-\eta}$, where d is the distance between the two nodes, and η is the path loss exponent. Adopting a Rayleigh fading model for the small-scale fading gain $X \sim \exp(1)$ implies that the CCDF of X is written as $F_X(x) = \exp(-x)$, and the pair connectedness function can be expressed as

$$H = \exp(-\beta d^{\eta}), \tag{3.2}$$

where β is a constant depending on the node transmission power, wavelength, information threshold etc. and can be understood as an effective communication range $r_0 = \beta^{-1/\eta}$. Notice that in the theoretical limit of $\eta \to \infty$, we have that $r_0 \to 1$ and H converges to the unit disk model. Without loss of generality, we will henceforth set $\beta = 1$.

3.2.2 Quasi Unit Disk Graph Communication Model

In the QUDG communication model [69], the probability H that two nodes a distance d apart are directly connected is

$$H = \begin{cases} 1 & \text{if } d < d_{max}/DOI \\ \frac{DOI(d_{max}-d)}{d_{max}(DOI-1)} & \text{if } d \in [d_{max}/DOI, d_{max}] \\ 0 & \text{if } d > d_{max}/DOI \end{cases} \tag{3.3}$$

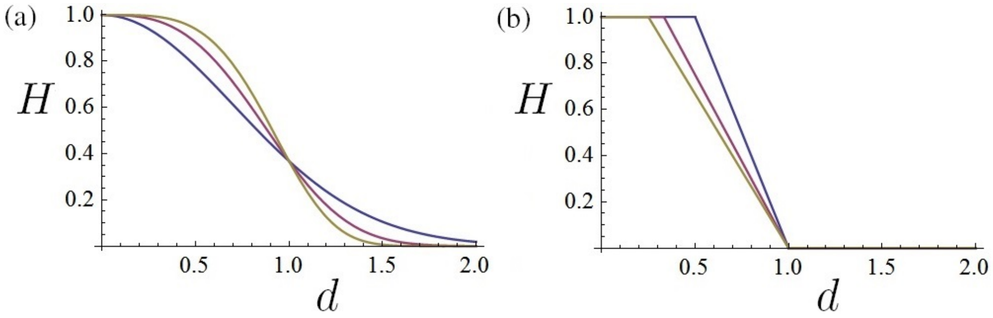


Figure 3.1: Plots of the pair connectedness function H using (a) the Rayleigh fading communication model (3.2) for parameters $\beta = 1$ and $\eta = 2, 3, 4$, and (b) the Quasi Unit Disk Graph communication model (3.3) for parameters $d_{max} = 1$ and $DOI = 2, 3, 4$.

where $d_{max} > 0$ and $DOI > 1$ are the maximum successful transmission distance and the degree of radio irregularity respectively.

During computer simulations, the network graph edges are formed if a random number $\zeta \in [0, 1]$ is less than the calculated H of the respective communication model. Two nodes sharing a successful link are called one-hop neighbours. More generally, two nodes with the minimum hop count of $k \geq 1$ (measured along the shortest path) are called k-hop neighbours. The two pair connectedness functions are plotted in Fig. 3.1.

3.3 KHopLoc Algorithm Description

Our proposed kHopLoc algorithm is composed of three simple steps. In the first step, each target node counts the minimum number of hops to each anchor node. In the second step, the conditional probability density function is generated for the probability density function $p(d|k)$ of the Euclidean inter-node distance d given hop count from target to anchor node equal to $k = 1, 2, \ldots$. This is performed locally at one or several nodes using Monte Carlo simulations. Finally, in the third step, each target node calculates its position by maximising the joint conditional density function to all reachable anchors. We now describe each of these steps in more detail.

3.3.1 Step 1: Minimum Hop-Count to Anchor Nodes

In this initial step, a classic distance vector exchange routine takes place (as in most DV-hop type localization algorithms [61]) where all target nodes in the network get distances, in hops, to their reachable anchors i.e. at most K-hops away. This can be initiated by anchors nodes which broadcast beacons to be flooded throughout the network containing the anchor's location with a hop-count value initialized to one. Each node maintains a table containing the coordinates (x_i, y_i) of anchor node i and the minimum hop distance to it h_i, and exchanges such updates only with its immediate neighbours.

3.3.2 Step 2: Multihop Connection Probability Density Function

The conditional probability density function $p(d|k)$ of inter-node distance d given the minimum hop count k, is to the best of our knowledge the necessary ingredient which differentiates our algorithm from all others. From Bayes theorem we have that $p(d|k) = p(k|d)p(d)/p(k)$. We may therefore calculate $p(d|k)$ indirectly through $p(k|d)p(d)$ and $p(k)$. Since $p(k)$ is independent of the inter-node distance, it does not affect the maximization calculation which follows and so will later be ignored.

The conditional probability of the target-to-anchor hop count being k, given that they are a distance d apart is given by $p(k|d)$ and can be approximately formulated in closed form as given by [9, 70]. Similarly, the probability distribution of the distance between two random points $p(d)$ can be formulated as in [71]. However, the probability density function in [70] consists of multiple integrals and assumes very dense networks, making it difficult to calculate, even more so by the sensor's weak processing unit. Therefore, in order to make progress we propose here a method for producing the probability density function $p(k|d)p(d)$ and then flooding the network with the required fitting parameters. First, Monte Carlo simulations are employed to calculate the discrete values of $p(k|d)p(d)$, which are then fitted into a continuous functions.

Monte Carlo Simulation

Let $K < N$ and $D \gg 1$ be the maximum allowed hop count and maximum allowed Euclidean distance between a target and anchor node respectively. Roughly, D should be at least larger than Kd_{max} or K/β. Partition the disk of radius D into L concentric shells (like an archery target) of widths $2\delta = D/L$ indexed by $l = 1, 2 \ldots L$ such that each shell has central radius of $d_l = \delta(2l + 1)$. We want to calculate the value of $p(k|d_l)p(d_l)$ for each $k = 1, 2, \ldots K$ and each d_l with $l = 1, 2 \ldots L$. The Monte Carlo simulation can now be performed at one or more sensor nodes (preferably ones with significant processing power) or a central server e.g. the gateway as follows:

(1) Generate random coordinates with intensity $\rho = N/A$ inside some predefined region. If the total number of nodes N and the WSN deployment region shape and area A are known then this is easy. If the WSN region shape is unknown, then a large square region can be used - it is shown that this does not affect the results significantly (see Fig. 3.5). If the density of nodes is unknown, then it can be estimated by the method described below in subsection 3.3.2.

(2) Generate communication links between nodes based on their mutual distances and appropriate connection probability function H (e.g. (3.2) or (3.3) or other).

(3) For each $k \in [1, K]$ and $l \in [1, L]$, calculate the discretized cumulative probability $p(k|d_l)p(d_l) \approx P(k|(d_l - \delta < d \leq d_l + \delta))P(d_l - \delta < d \leq d_l + \delta)/2\delta$.

(4) Repeat steps (1) - (3) above several times in a Monte Carlo fashion in order to refine the estimated $p(k|d_l)p(d_l)$.

Fitting: For each $k \in [1, K]$, fit the discrete probability distribution $p(k|d)p(d)$ to the following function of d

$$p(k|d)p(d) = \exp(-A(k)(d - B(k))^2 + C(k)), \tag{3.4}$$

where $A(k), B(k), C(k)$ are functions of k, e.g. polynomials of degree $\wp \gg 1$. In the simulations that follow we use $\wp = 4$. The Gaussianity of (3.4) was inspired by the extensive simulations results and analysis presented in [72].

Density Estimation

This section describes a simple method of estimating node density ρ, a prerequisite for performing the Monte Carlo simulations and building the said distributions. In a uniformly distributed network, the node density is defined as $\rho = N/A$, where A is the WSN deployment region area, thus it can be approximated by the average number of one-hop neighbours N_e divided by it's average communication area A_e. The former can be simply determined by each sensor counting the number of 1-hop neighbours from each node while the latter can be calculated by

$$A_e = \int_0^{2\pi} \int_0^{\infty} rH(r)\mathrm{d}r\mathrm{d}\theta, \tag{3.5}$$

where $H(r)$ is the pair connectedness function of two nodes whose relative distance is r (see (3.2) and (3.3)). The estimated density ρ_e is then sent to the nearest processor which performs an average to obtain a refined estimate.

3.3.3 Step 3: Maximum Likelihood Based Multihop Localization

This subsection describes our proposed kHopLoc algorithm utilizing maximum likelihood methods. First we introduce the likelihood function and then we address a method to maximize it. Consider a target node X and let h_i be the hop count measure along the shortest path from X to anchor node $i \in [1, M]$. The likelihood function of X at the yet undetermined coordinate (x, y) is defined as

$$L(x, y) = p(x, y|h_1, h_2, ...h_M). \tag{3.6}$$

It follows that the best estimate of the true location of node X is the value that maximizes it's likelihood

$$(x^*, y^*) = \arg\max_{(x,y)} L(x, y). \tag{3.7}$$

Assuming that the probability density functions $p(x, y|h_i)$ and $p(x, y|h_j)$ are mutually independent for $i \neq j$, equation (7.5) can be written as

$$L(x, y) = \prod_{i=1}^{M} p(x, y|h_i) = \prod_{i=1}^{M} p(d_i|h_i), \tag{3.8}$$

where $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$ and (x_i, y_i) are the coordinates of anchor node i. Invoking Bayes theorem $p(d_i|h_i) = p(h_i|d_i)p(d_i)/p(h_i)$ yields

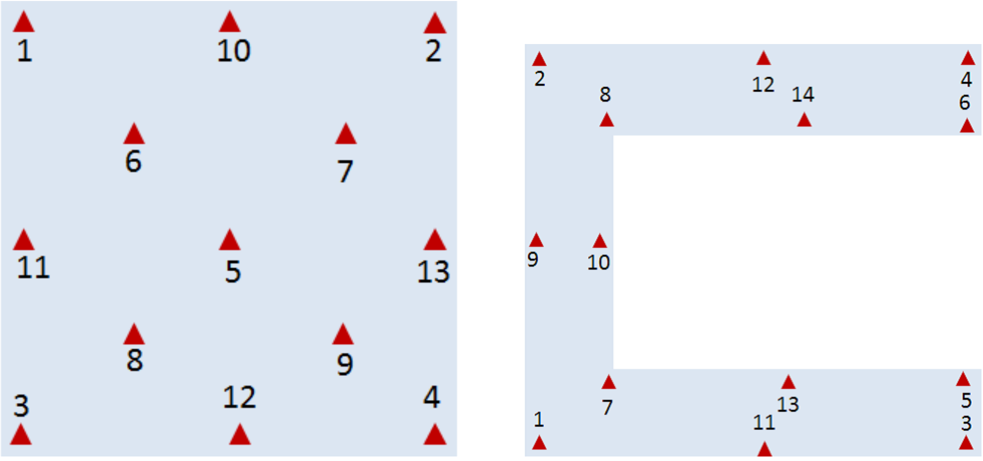
$$L(x, y) = \prod_{i=1}^{M} \frac{p(h_i|d_i)p(d_i)}{p(h_i)}. \tag{3.9}$$

Substituting equation (3.9) back into equation (3.7) , we obtain

$$(x^*, y^*) = \arg\max_{(x,y)} \frac{\prod_{i=1}^{M} p(h_i|d_i) \cdot \prod_{i=1}^{M} p(d_i)}{\prod_{i=1}^{M} p(h_i)}. \tag{3.10}$$

Because $p(h_i)$ is independent of (x, y), the product in the denominator can be eliminated from equation (3.10) such that

$$(x^*, y^*) = \arg\max_{(x,y)} \prod_{i=1}^{M} p(h_i|d_i)p(d_i). \tag{3.11}$$



(a) Isotropic network

(b) Anisotropic network

Figure 3.2: Illustration of fixed anchor node locations.

Finally, substituting our fitted equation (3.4) into equation (3.11) gives the result

$$\begin{aligned}(x^*, y^*) &= \arg\max_{(x,y)} \prod_{i=1}^{M} \exp(-A(h_i)(d_i - B(h_i))^2 + C(h_i)) \\ &= \arg\min_{(x,y)} \sum_{i=1}^{M} A(h_i)(d_i - B(h_i))^2.\end{aligned} \tag{3.12}$$

The left hand side of equation (3.12) can be easily calculated using gradient descent method or Newton method for example and can be performed by each target node independently.

3.4 Performance Evaluation and Analysis

In this section, we demonstrate using computer simulations the effectiveness of kHopLoc, and also compare to the original DV-hop [61] algorithm and a recent improved variant [66] which we will refer to as the ASP algorithm for short. To evaluate the performance of the kHopLoc in isotropic networks and anisotropic networks, we deploy sensor nodes randomly in a 10 × 10 square-shaped region (such that $A = 100$) and in a 10 × 10 C-shaped of width 2 (such that $A = 52$), where communication probabilities in isotropic networks and anisotropic networks are assumed to follow Raleigh fading communication model and QUDG communication model, respectively. In Rayleigh fading communication model, we assume the path loss exponent to be $\eta = 2$ and parameter $\beta = 1$ in formula (3.2). In the QUDG communication model, we assume the maximum connection distance $d_{max} = 1$, and the degree of irregularity $DOI = 1.5$. Moreover, to study the characteristic localization errors we simulate results in both isotropic and anisotropic WSN deployment regions with variable *a*) anchor node positions (fixed and random), *b*) number of anchor nodes, and *c*) node densities. Communication and computation overhead costs are also compared and discussed.

3.4.1 Localization Error

Fig. 3.3 depicts the average localization errors of DV-hop algorithm and kHopLoc by varying the number of anchor nodes and total sensor nodes in a square-shaped region under Rayleigh fading communication model. In all cases, the error based on kHopLoc is significantly smaller than that of the DV-hop algorithm with performance gains ranging between 20 and 40%. Generally, the average localization error decreases with number of anchor nodes. Fixed anchor nodes (at strategic locations as in Fig. 3.2(a)) in general provides better performance than randomizing anchor nodes, especially when the number of anchor nodes is small. This is clearly due to the controlled avoidance of overlaps between anchor nodes which can lead to duplicate mutual information.

Fig. 3.4 illustrates two example network topologies (a sparse regime at $N = 200$, and a dense regime at $N = 700$), and highlights the localization error of each node. There are exactly $M = 13$ fixed anchor nodes in all cases. The green lines connecting the nodes and blue circles describe communication links and localization errors in which the radius are proportional to the localization errors. In kHopLoc (panels (a) and (c)), errors of nodes having few links tend to be big, thus the average error decreases in the dense regime of $N = 700$. On the other hand, in the DV-hop algorithm, localization errors of nodes near the border tend to be significantly larger.

Fig. 3.5 depicts the average localization error of the DV-hop algorithm [63], the ASP algorithm [66], and kHopLoc, for varying number of anchor nodes and total sensor nodes in C-shaped anisotropic region under the QUDG communication model. In all cases, the error based on kHopLoc is smaller than the other algorithms. Notice that there are two result curves for kHopLoc: when the density and shape of the deployment region are known (green curve), and when both the density and shape of the region are unknown (purple curve). The performance of the case when the region shape is known is better than the other case since the Monte Carlo simulation phase of the algorithm over a known shape region produces more precise distributions and thus results for the multihop connection probability than when the region is assumed to be a square. Significantly, it is worth noting that the average localization error due to kHopLoc continues to decrease for the other two algorithms seems to saturate after 300 nodes. This demonstrates the benefits of using the full statistical hop-distribution generated during the first step of our algorithm (see section 3.3.2) rather than just first order statistics such as the mean one-hop Euclidean distance.

Fig. 3.6 illustrates network topologies (a sparse regime at $N = 200$, and a dense regime at $N = 700$), and highlights the localization errors of each node. There are exactly 14 fixed anchor nodes in the anisotropic C-shaped WSN deployment regions as shown in Fig. 3.2(b). The green lines connecting the nodes and blue circles describe communication links and localization errors in which the radius are proportional to the localization errors. Similar to isotropic network, in kHopLoc,the main source of localization errors is due to nodes having few one-hop links. The reason for this is that nodes with fewer links tend to require a larger than average hop-count to reach anchors, thus making it difficult to estimate these node locations accurately. As can be seen, this improves significantly in the dense regime. On the other hand, localization errors in the other two algorithms seem not to improve with node density. The reason for this is that these algorithms suffer from inaccurate inter-node distance estimations.

3.4.2 Overhead Analysis

This section discusses the communication and computational costs of the three localizations algorithms under investigation: the DV-hop algorithm, the ASP algorithm, and kHopLoc

Computational Cost

For calculating location of one target node, DV-hop algorithm costs $\mathcal{O}(M^8)$ if using normal matrix multiplication. This cost comes from the matrix operations necessary to estimated the least square error [62]. The ASP algorithm costs consist of additional computation costs of $\mathcal{O}(\Delta)$ (where Δ is the number of tagged partitions for calculating Riemann sum) for calculating node density and $M(M-1)/2 \cdot \mathcal{O}(\epsilon^{-3/2})$ for estimating distances between the target node and M anchor nodes, where $M(M-1)/2$ is number of compound shortest paths, and $\mathcal{O}(\epsilon^{-3/2})$ (in which ϵ is the upper bound of the norm of the gradient) is the number of iterations of Newton method [73] for calculating the optimal central angles of the virtual holes [66]. Consequently, the total computation complexity of the ASP algorithm is $\mathcal{O}(M^8 + \Delta + M^2 \cdot \epsilon^{-3/2})$.

On the other hand, the cost of kHopLoc consists of Monte Carlo simulation cost and the MLE localization cost. The former computational cost requires generating the random network, finding the shortest k-hop paths and fitting. Thus, the cost comes up to $\mathcal{O}(I \cdot N^3 + K \cdot L \cdot \epsilon^{-3/2})$, where I is number of Monte Carlo iterations and $\mathcal{O}(N^3)$ is the cost for calculating all-pairs shortest paths if using for example the Floyd Warshall algorithm [74], and $\mathcal{O}(K \cdot L \cdot \epsilon^{-3/2})$ is the fitting cost (see (3.4)) if using Newton method. However, this computation can be done just once at a central node (or some server e.g. at the gateway) and then flooded through the network thus incurring an additional communication cost of $\mathcal{O}(N)$. Or otherwise, it is done before deploying the sensor nodes and then derived parameters $(A(k), B(k))$ are included into each sensor nodes, thus no additional communication cost occurs.

The MLE cost is $\mathcal{O}(M \cdot \epsilon^{-3/2})$, where $\mathcal{O}(\epsilon^{-3/2})$ is the number of iteration of Newton method and $\mathcal{O}(M)$ is cost for calculating the value of function (3.12) in each iteration. Therefore the total cost amounts to $\mathcal{O}(I \cdot N^3 + M \cdot \epsilon^{-3/2})$ Obviously, the MLE localization cost is smaller than that of the ASP algorithm.

Communication Cost

The communication costs of the DV-hop algorithm and the ASP algorithm are bounded by $2\mathcal{O}\left(M(N-M)\right)$, where M and $N-M$ are number of anchor and target nodes respectively. This is because these algorithms perform flooding twice - first for the minimum hop count estimation, and second to broadcast the average one-hop distance.

The communication cost of kHopLoc however is due to the initial hop count calculation giving $\mathcal{O}(M(N-M))$ (i.e. similar to [67]). When the density ρ is unknown, additional communication costs of $2\mathcal{O}(N)$ can be incurred, in which $\mathcal{O}(N)$ occurs when the nodes pass theirs 1-hop neighbour number to the central node. After the central runs the Monte Carlo simulations and fits the said distributions, it then broadcast the results (parameters $A(k), B(k)$) to all nodes, thus costing another $\mathcal{O}(N)$.

3.5 Conclusions

We have proposed a maximum likelihood-based multihop localization algorithm called kHopLoc for use in Wireless Sensor Networks (WSNs). The main advantage of the algorithm is the use of a Monte Carlo initial training phase to generate the multihop connection probability density functions. These are then used to build likelihood functions whose maxima estimate each target node location.

Since the algorithm uses full statistical information for the multihop connection probabilities, localization results are significantly (about $20-40\%$) more accurate for both in isotropic and anisotropic networks. We have validated these results through computer simulations and discussed how and why some localization errors appear. Finally, we have discussed the communication and computational costs of kHopLoc compared to conventional ones. Moreover, like most range-free algorithms, kHopLoc can be used in conjunction with GPS and/or range-based localization schemes to improve performance and energy consumption in WSNs.

We studied from this work that

- Coarse positions of wireless devices can be determined using the proximity measurement technique, which is known as the lowest ranging accuracy among radio frequency measurement techniques. This enables us to assume that localization of wireless devices can be realized using RSSI measurement techniques, which is known as providing better accuracy.

- The proposed kHopLoc enhances accuracy because it is provided full statistical information for the multihop connection probabilities, which is obtained through an off-line training phase. This training phase is conducted using simulation data and Monte Carlo simulation method, thus not requiring extra data collection cost. We thus further extend this idea to improve the accuracy of different localization problems, which is demonstrated in Chapter 6.

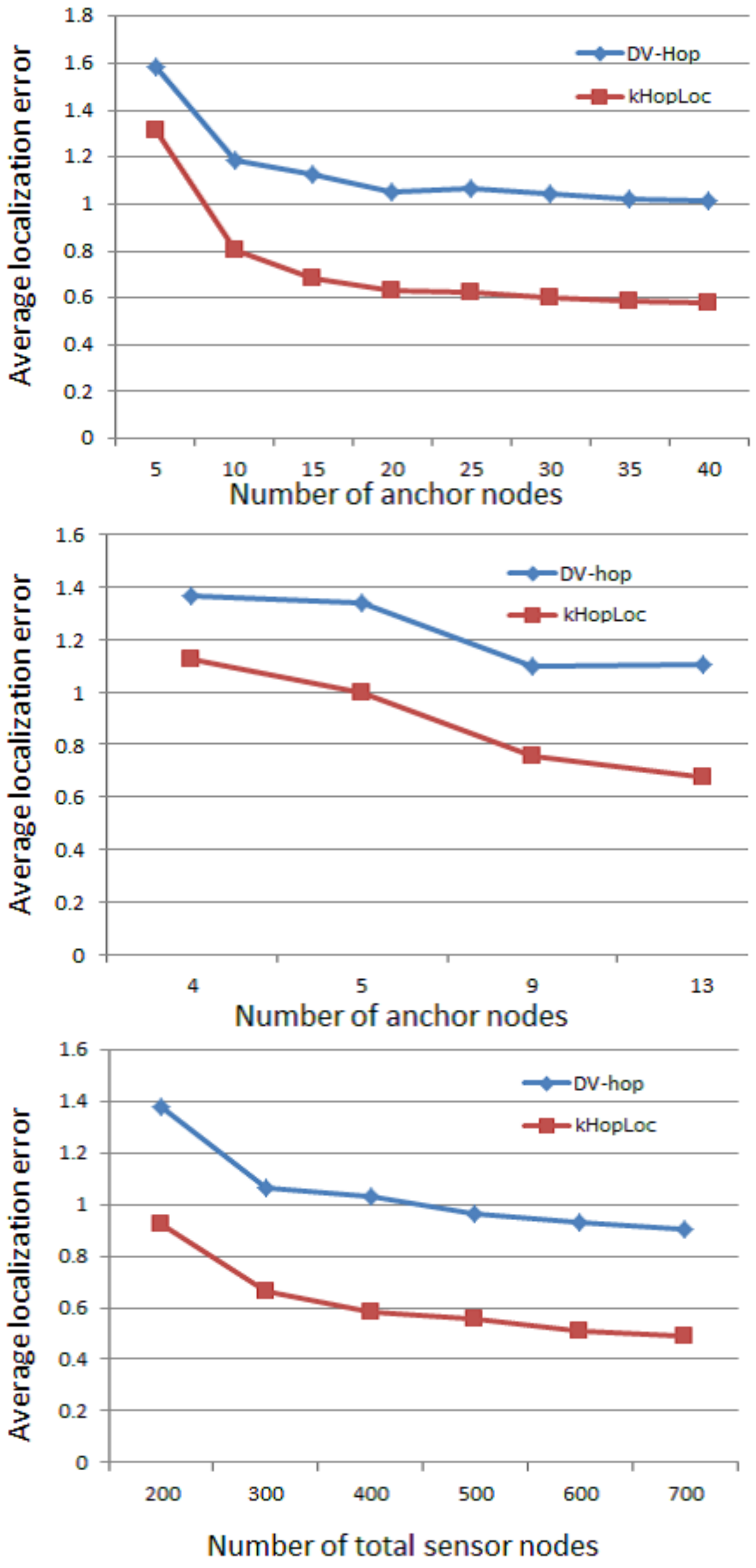


Figure 3.3: Average localization error of DV-hop and kHopLoc in isotropic networks with random target node locations. *Top:* $N = 300$ with random anchor node locations. *Middle:* $N = 300$ with fixed anchor node locations as in Fig. 3.2(a). *Bottom:* $M = 13$ fixed anchor node locations and $N \in [200, 700]$.

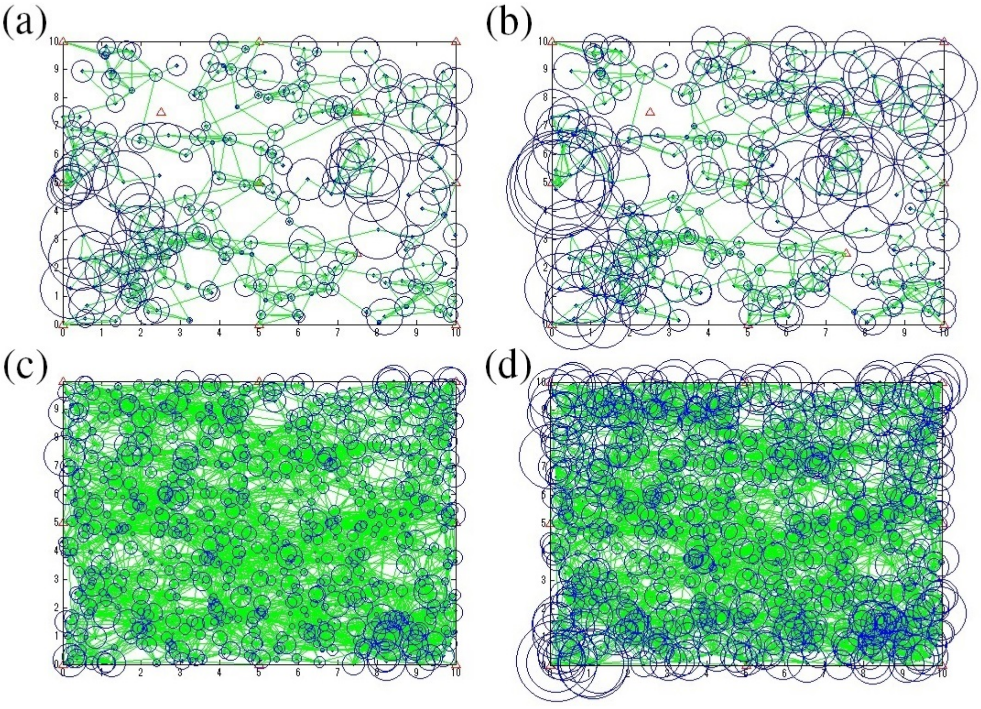


Figure 3.4: Example isotropic WSN topologies and localization errors shown as disks. In all cases there are $M = 13$ fixed anchor nodes as in Fig 3.2(a). Panels (a) and (c) use kHopLoc and $N = 200, 700$ sensor nodes. Panels (b) and (d) use DV-hop algorithm and $N = 200, 700$ sensor nodes. It is clear that kHopLoc has smaller localization errors, particularly near the domain boundary.

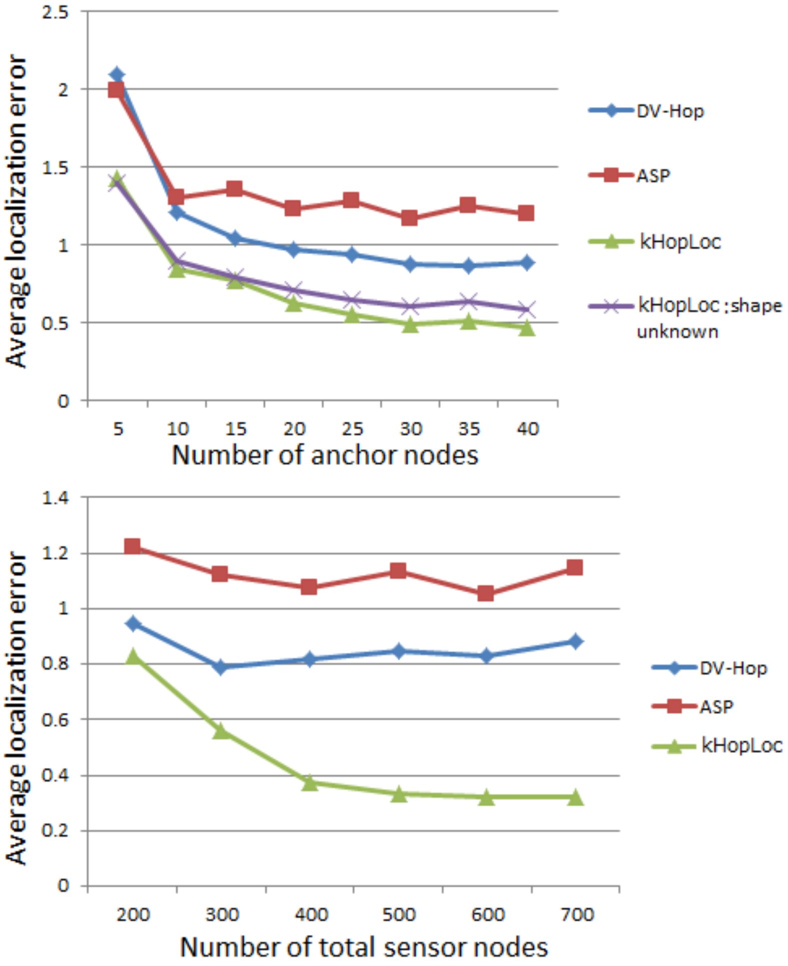
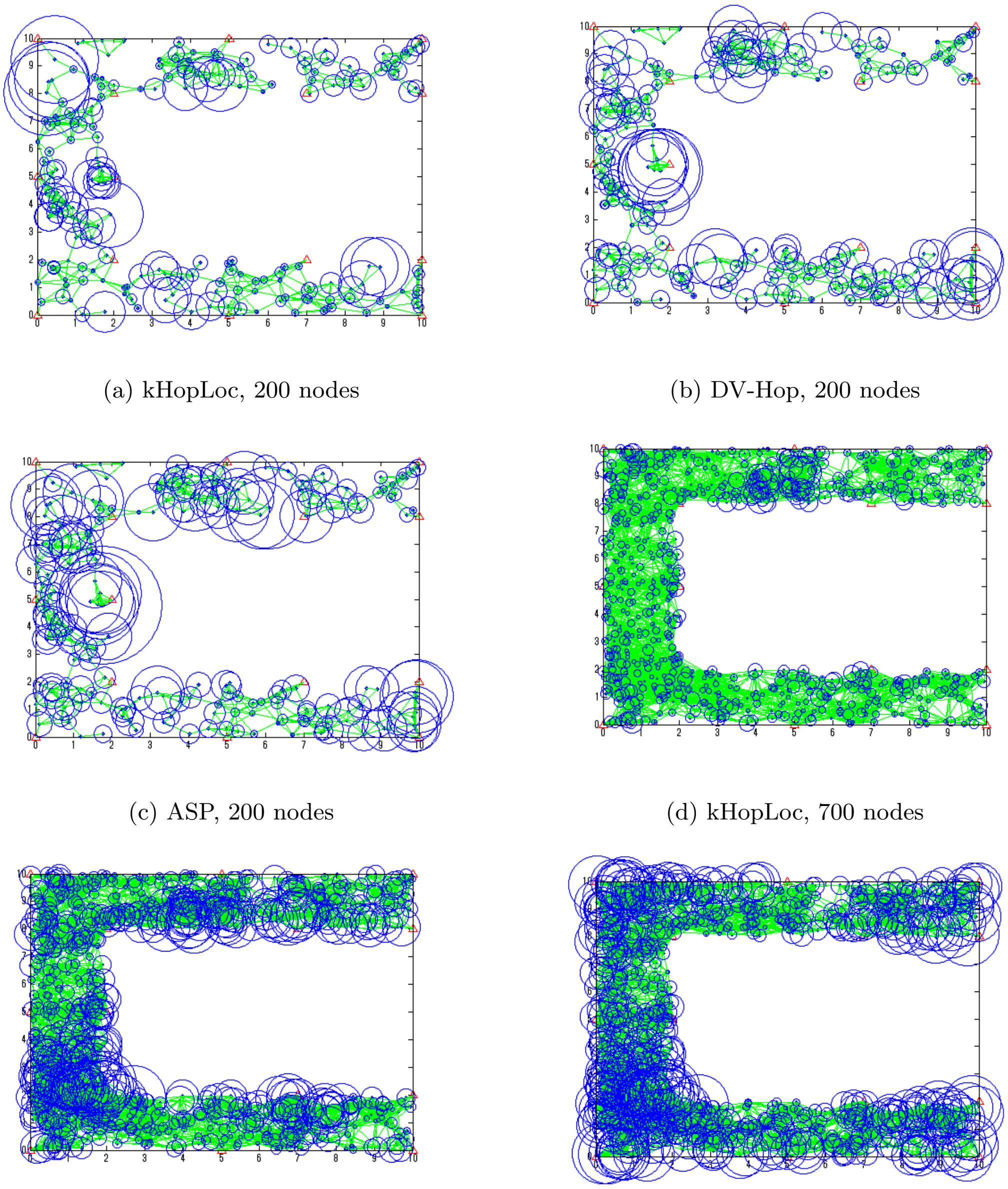


Figure 3.5: Average localization error of DV-hop, ASP, and kHopLoc in anisotropic networks with random target node locations. *Top:* $N = 300$ with random anchor node locations. *Bottom:* $M = 14$ fixed anchor node locations (as in Fig. 3.2(b)) and $N \in [200, 700]$.



(a) kHopLoc, 200 nodes

(b) DV-Hop, 200 nodes

(c) ASP, 200 nodes

(d) kHopLoc, 700 nodes

(e) DV-Hop, 700 nodes

(f) ASP, 700 nodes

Figure 3.6: Example anisotropic WSN topologies and localization errors shown as disks. In all cases there are $M = 14$ fixed anchor nodes as in Fig 3.2(b). It is clear that kHopLoc has smaller localization errors, particularly near the domain boundary.

Chapter 4

Exploiting Low-cost Localization: Indoor Binary Localization

In the previous chapter, we demonstrate that low-cost localization techniques that use ordinary wireless hardware can provide coarse locations of target wireless nodes. We also showe that a simple technique based on Monte Carlo simulation, which does not increase extra cost, can improve localization accuracy. This demonstrates the potency of low-cost localization. In this chapter, we illustrate the applicability of low-cost localization through a new localization scheme called *WiLAD*, which stands for Wireless Localization through Anomaly Detection.

WiLAD is a special case of wireless localization for scenarios where, instead of absolute positioning of an object, only the information whether an object is *inside* or *outside* of a specific area is required. This is motivated through a number of applications including, but not limited to, *a) security:* detecting whether an object is removed from a secure location, *b) wireless sensor networks:* detecting sensor movements outside of a network area, and *c) computational behavior analytics:* detecting customers leaving a retail store. The result of such detection systems can naturally be utilized in building a higher level contextual understanding of a system or user behaviors. Through experiments conducted in a real store, we demonstrate that a simple and low-cost RSSI-based localization technique can achieve satisfactory localization accuracy. Besides this, we propose low-cost optimization methods to improve accuracy.

This chapter is organized as follows. In Section 4.1, we highlight related works and motivate WiLAD through potential real-world applications. In Section 4.2, we describe the system model, and related background. In Section 4.3, we present the details of our proposed localization technique. In Section 4.4, we propose a mathematical formulation for estimating the accuracy under some assumptions, aimed at providing meaningful insights towards achieving optimal accuracy. In Section 4.5, we propose optimization methods to improve the localization accuracy. In Section 4.6, we evaluated the performance of WiLAD in real environments and to validate our proposed optimization through experiments in a real store environment. Finally, Section 4.7 concludes the chapter.

4.1 Introduction

Wireless localization, which has been of great interest over the past few years [20], refers to extracting geo-location information of an object based on its wireless signals to multiple known devices. There are numerous important applications, particularly industrial applications, commercial environments, public safety settings, everyday life and defense/security systems [75]. Solutions for deriving the location information can be categorized into two groups as unsupervised methods and supervised methods [20]. Unsupervised methods also known as triangulation methods estimate the distance from a number of known devices (anchors) and multilaterate the location of target objects [76]. These methods are subject to errors that are caused by various factors including noise, multi-path fading, shadowing and Non-Line of Sight (NLoS). Moreover these are sometimes costly and time-consuming since model parameters need to be adjusted for specific environments. On the other hand supervised methods such as fingerprinting compare signal features to a pre-generated database in order to identify the most likely location of target objects [77]. Received Signal Strength Indicator (RSSI)-based location fingerprinting is commonly used for this method. There are various fingerprinting-based localization algorithms such as probabilistic methods [78], k-nearest-neighbor (kNN), Neural Networks, Support Vector Machine (SVM) [79], and Smallest M-vertex Polygon (SMP) [80]. These methods often perform better than unsupervised methods, however they are computationally expensive and time consuming since signal fingerprints are required to be collected in advance.

These localization methods provide absolute positions of target objects. However in some applications, the absolute positions are not always required. For example, in some scenarios a target object is only required to be detected whether it is *inside* or *outside* of a specific place. In security, it is crucial to detect whether an object (for example an object that can transmit wireless signals such as a smart-phone, a tablet) is removed from a secure location. In the context of customer analytics (for example in retail), the main interests are in the number of people entering or leaving the store and the time they spend purchasing/viewing products. Such analytics can be enabled assuming that people have access to wireless devices such as smart phones. With pre-designated zones in the store, a more thorough and complete understanding of consumer behavior can be established. For example, analysis of customers entering and spending time at a particular section can be made where a new product has been recently launched. Such localization systems can also find an application in the sensor networks domain where a sensor node is required to be detected if it goes *out* of its specific (i.e. usual) area. The information can be used such as to detect when some phenomenon happened (earth quake, landslide).... Such kind of applications motivate us to define and develop a new class of localization, WiLAD (Wireless localization through Anomaly Detection) which classifies the target object into two types of area: *target area* (i.e. inside) and *non-target area* (i.e. outside). Due to the fact that the non-target area can be too large making data collection practically infeasible, collecting training data in only the target area, which is normally small, can significantly reduce data collection costs compared with conventional fingerprinting methods in which training data for all classes is required. Therefore, these methods cannot be directly utilized in our localization system which requires only training data in one class. Besides, other information such as locations of anchors and model parameters (particularly needed in the case of unsupervised methods), are also not required, thus deployment requirements are minimal.

In order to identify objects of a specific class amongst all objects, we approach this with a one-class (or unary) classification mechanism. This is performed by learning from a training set containing only the objects of that class [81]. Among one-class classification methods, one-class Support Vector Machine (OC-SVM) is known to outperform other methods in several datasets [81]. Therefore we employ OC-SVM in our localization system. To the best of our knowledge, this is the first such attempt to perform wireless localization.

Besides, other works related to improving localization accuracy, such as localization accuracy estimation [75], anchor placement optimization [82,83] has also attracted significant attention in recent times. For example, [75] derive the bound of localization accuracy for RSSI measurements. This gives a useful insight in to localization performance and deployment issues of a localization system, which could help in designing an efficient localization system. In [82,83], the authors set out to find the optimal number and placement of the anchor nodes in a given area for improving localization accuracy. These methods rely on real experiments or simulations for specific environment (specifically requiring experiments or simulations for each of the given areas) thus raising both the cost and complexity. We derive approximation formulation of accuracy which utilizes the characteristics of wireless signals as well as OC-SVM. Moreover, based on the formulation we then propose a novel mathematical framework to find the optimal placement of devices (anchor devices as well as target areas for target objects). This mathematical formulation enables us to optimize device placement without performing any costly experiment or simulation.

Our main contributions are summarized as following:

- Propose a method to estimate the detection accuracy.
- Propose methods to improve the accuracy including a novel method for optimizing placement of devices.
- Validate proposed methods via numerical simulations as well as real experiments.

4.2 Preliminaries

4.2.1 System Model

Consider a wireless network system of k anchors (here referred to as access points or APs) positioning at $\boldsymbol{a}_1, \boldsymbol{a}_2, ..., \boldsymbol{a}_k$, and a target object normally moving or staying around an area called *target area*. The target object is equipped with a radio transceiver, and broadcasts beacon signals at a set interval of time. Each AP then receives the signals and retrieves RSSIs (Received Signal Strength Indicators), followed by sending values of RSSIs to a backhaul server. The server then uses collected RSSIs to determine whether the target object stays inside its target area or not (i.e. non-target area).

4.2.2 Propagation Models

RSSI $r_{\boldsymbol{t},i}$ between a target object positioning at $\boldsymbol{t}$ and the i-th AP positioning at $\boldsymbol{a}_i$ is related through the log-distance propagation model (in dBm) [35]. one

$$r_{\boldsymbol{t},i} = P_0 - 10\eta \lg \|\boldsymbol{t} - \boldsymbol{a}_i\| + \mathcal{X} \tag{4.1}$$

where η and P_0which are constants, are the path loss exponent and the reference power at the distance of 1 meter respectively, $\mathcal{X}$ is a random variable characterizing the effects due to multi-path fading and noisy measurements. $\|\boldsymbol{x}\|$ is Euclidean norm of a vector $\boldsymbol{x}$, thus $\|\boldsymbol{x} - \boldsymbol{y}\|$ is the distance between two positions $\boldsymbol{x}$ and $\boldsymbol{y}$. We denote $\log_{10} x$ as $\lg x$ for simplicity. The signal fluctuations $\mathcal{X}$ due to multi-path fading and noise depend on the wireless propagation environment. For example, the long-term signal variation is known to follow the Log-normal distribution, whereas the short-term signal variation can be described by several other distributions such as Hoyt, Rayleigh, Rice, Nakagami-m, and Weibull.

4.2.3 One-Class Support Vector Machine (OC-SVM)

OC-SVM (a particular type of supervised learning) tries to identify objects of a specific class amongst all objects, by learning from a training set containing only the objects of that class. We briefly introduce OC-SVM [84] as follows. Suppose the training target class is $\hat{\boldsymbol{r}}_1, \hat{\boldsymbol{r}}_2, ..., \hat{\boldsymbol{r}}_s$, where $\hat{\boldsymbol{r}}_j \in \mathcal{R}^k, \forall j \in [1, s]$. In the input space, OC-SVM aims to determine a hyperplane to separate the target class and the origin of the input space with the maximum margin:

$$\begin{aligned} \min \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 - \rho + \frac{1}{\vartheta \cdot s}\sum_{1\le j\le s} \xi_j \\ \text{s.t.} \quad & \boldsymbol{w} \cdot \hat{\boldsymbol{r}}_j \ge \rho - \xi_j \\ & \xi_j \ge 0, \quad \forall j \end{aligned} \tag{4.2}$$

where parameter $\vartheta \in (0, 1)$ is used to trade off the sphere volume and the errors $\sum_{1\le j\le s} \xi_j$, s is the size of the training data. For a test sample $\hat{\boldsymbol{r}}_{\boldsymbol{t}}$ if

$$\boldsymbol{w} \cdot \hat{\boldsymbol{r}}_{\boldsymbol{t}} \ge \rho, \tag{4.3}$$

it is classified into the target class, otherwise, it belongs to the non-target class. In practice, ϑ is automatically calculated if provided the fraction of training error (called ν). The inner product is normally calculated using a kernel. The Radial basis function kernel, also called the RBF kernel, or Gaussian kernel is widely used, which is defined as follows.

$$\boldsymbol{x} \cdot \boldsymbol{y} = \exp\left(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|^2\right) \tag{4.4}$$

where γ is a constant. The kernel is the indicator of similarity between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$.

4.3 Localization Anomaly Detection Method

Our proposed framework for anomaly detection in localization system has two main phases: training phase and decision phase. In the training phase, data in the target class collected beforehand is used to train an OC-SVM. In the detection phase, the trained model is used to determine whether the target object is inside the target area or outside using the data collected in real-time.

To improve the accuracy, before passing to the OC-SVM, we perform feature extraction as follows.

4.3.1 Feature Extraction

As described in Section 4.2.2, a single signal fluctuation normally follows a non-Gaussian distribution, in which, in extreme cases it is possible that the absolute value of random variable $\mathcal{X}$ becomes very large, i.e., RSSI between nodes is small even when their distance is close. Such fluctuations can have a significant effect on the detection accuracy. Therefore to improve detection accuracy, we average N successive RSSI values between the target object an each APs, in which N can be empirically selected depending on the applications. Therefore, the availability of multiple independent RSSI measurements enables the use of the Center Limit Theorem (CLT), and thus the modeling of fluctuation by a Gaussian distribution. The averaged RSSIs between the target object positioning at $\boldsymbol{t}$ and the i-th APs positioning at $\boldsymbol{a}_i$ follows:

$$\bar{r}_{\boldsymbol{t},i} = P_0 - 10\eta \lg \|\boldsymbol{t} - \boldsymbol{a}_i\| + X, \tag{4.5}$$

where X is a random variable (with a Gaussian distribution).

Secondly, in order to achieve a generalized applicability and a scalable method, we standardize our averaged data to minimize cross-environmental RSSI magnitude variance. Namely, in the training phase each averaged RSSI is subtracted by the mean from each feature type, then divided by its standard deviation. On the other hand, in the detection phase, the averaged RSSIs is subtracted by the mean from the corresponding features in training data.

4.3.2 Parameter Settings for OC-SVM

To enhance the system accuracy, it is fundamental to choose appropriate parameters for the OC-SVM. While in binary SVM, the training data in both classes are available thus the parameters can be optimized using such as cross validation, in OC-SVM the parameters are difficult to be optimized since data in non-target class is unavailable. The RBF's parameter γ is therefore set to be at its default value, i.e., $\gamma = 1/k$, where k is the number of features which is equal to number of APs. This is because, Section 4.4 shows that the value of γ does not effect the OC-SVM strongly if the data is standardized.

4.4 Probability of Successful Detection

In this section we propose a mathematical formulation for estimating the accuracy under some assumptions, aimed at providing meaningful insights towards achieving optimal accuracy. Given the fraction of training error ν, the accuracy is related to the probability of successful detection (here called *detection rate*) when the target node goes outside of its target area.

4.4.1 Formulation

For simplicity, we propose a method for calculating the detection rate under the following assumptions: Firstly, we assume that the target area is small, so we can consider that it is a point positioning at $\boldsymbol{t}_{in}$. Note that if the target is not that small, we can approximately consider $\boldsymbol{t}_{in}$ as the middle point of the target area. For instance, $\boldsymbol{t}_{in}$ is illustrated by symbol X in Fig. 4.4 which is the middle point of a particular target area. Secondly, due to the averaging process described in section 4.3, fluctuation of each averaged RSSI can be assumed as following Gaussian distribution with 0 mean, σ^2 variance, namely, $X \sim \mathcal{N}(0, \sigma^2)$. Due to Equation (4.5), the averaged RSSI between target object positioning at $\boldsymbol{t}_{in}$ and the i-th AP follows $\mathcal{N}(P_0 - 10\eta \lg \|\boldsymbol{t}_{in} - \boldsymbol{a}_i\|, \sigma^2)$. Thus the value of the i-th feature corresponding to the j-th training data is

$$\hat{r}_{j,i} = \frac{\bar{r}_{j,i} - P_0 + 10\eta \lg \|\boldsymbol{t}_{in} - \boldsymbol{a}_i\|}{\sigma} = \frac{X}{\sigma}. \tag{4.6}$$

Hence each training vector $\hat{\boldsymbol{r}}_j$ consists of k components following $\mathcal{N}(0, 1)$, where k is the number of APs. We then estimate the margin of an OC-SVM trained by training data $\hat{\boldsymbol{r}}_1, ..., \hat{\boldsymbol{r}}_s$. The first constraint of OC-SVM given by Formula (4.2) can be written as follows.

$$\begin{aligned} &\exp\left(-\gamma \|\boldsymbol{w} - \hat{\boldsymbol{r}}_j\|^2\right) \geq \rho - \xi_j \\ \Leftrightarrow\ &\|\boldsymbol{w} - \hat{\boldsymbol{r}}_j\|^2 \leq -\frac{\ln(\rho - \xi_j)}{\gamma} \end{aligned} \tag{4.7}$$

Since the objective of an OC-SVM is to minimize the sum $\frac{1}{2}\left\|\boldsymbol{w}\right\|^2 - \rho + \frac{1}{\vartheta \cdot s}\sum_{1\leq j\leq s}\xi_j$, namely approximately minimize $\|\boldsymbol{w}\|$ and $\xi_j - \rho$. Thus $\|\boldsymbol{w} - \hat{\boldsymbol{r}}_j\|^2$ (the left side of (4.7)) and $\|\boldsymbol{w}\|$ should take small values. Moreover the average of training data $\hat{\boldsymbol{r}}$ is $\mathbf{0}$, consequently $\boldsymbol{w}$ is approximately also $\mathbf{0}$.

Substituting $\boldsymbol{w} = \mathbf{0}$ in Equation (4.3), a vector $\hat{\boldsymbol{r}}_{\boldsymbol{t}}$ can then be classified as in the target class if:

$$\begin{aligned} \exp\left(-\gamma \left\|\boldsymbol{w} - \hat{\boldsymbol{r}}_{\boldsymbol{t}}\right\|^2\right) \geq \rho \\ \Leftrightarrow \left\|\hat{\boldsymbol{r}}_{\boldsymbol{t}}\right\|^2 \leq \delta, \end{aligned} \tag{4.8}$$

where $\delta = -\ln(\rho)/\gamma$.

Since vector $\hat{\boldsymbol{r}}_j$ has k components, in which each component $\hat{r}_{j,i}$ follows $\mathcal{N}(0,1)$ and are independent to each other, thus $\left\|\hat{\boldsymbol{r}}_j\right\|^2$ follows chi-squared distribution with k degrees of freedom. As fraction of training error is ν, there is $1-\nu$ fraction of training data satisfying Equation (4.8). Thus,

$$\delta = F_{\chi 2}^{-1}(1-\nu) \tag{4.9}$$

where $F_{\chi 2}(x)$ is the cumulative distribution function (CDF) of variable $\chi 2$ following chi-squared distribution with k degrees of freedom, evaluated at x, and $F_{\chi 2}^{-1}(x)$ is its inverse function. Equation (4.9) shows that the value of δ only depends on ν, thus is a constant.

We now calculate the *detection rate* of a specific position $\boldsymbol{t}$, which is the probability that the trained OC-SVM classifies a vector $\hat{\boldsymbol{r}}_{\boldsymbol{t}}$ as non-target class when the target object positioned at $\boldsymbol{t}$ is outside its target area. Averaged RSSI between $\boldsymbol{t}$ and AP $\boldsymbol{a}_i$ can be described as follows.

$$\bar{r}_{\boldsymbol{t},i} = P_0 - 10\eta \lg \|\boldsymbol{t} - \boldsymbol{a}_i\| + X. \tag{4.10}$$

Utilizing Equation (4.6), the standardized vector $\hat{\boldsymbol{r}}_{\boldsymbol{t}}$ has i-th component having the following value:

$$\begin{aligned} \hat{r}_{\boldsymbol{t},i} &= \frac{\bar{r}_{\boldsymbol{t},i} - P_0 + 10\eta \lg \|\boldsymbol{t}_{in} - \boldsymbol{a}_i\|}{\sigma} \\ &= \frac{10\eta}{\sigma} \lg \frac{\|\boldsymbol{t}_{in} - \boldsymbol{a}_i\|}{\|\boldsymbol{t} - \boldsymbol{a}_i\|} + \frac{X}{\sigma}, \end{aligned} \tag{4.11}$$

which follows $\mathcal{N}(\frac{10\eta}{\sigma} \lg \frac{\|\boldsymbol{t}_{in} - \boldsymbol{a}_i\|}{\|\boldsymbol{t} - \boldsymbol{a}_i\|}, 1)$ Thus we have,

$$\left\|\hat{\boldsymbol{r}}_{\boldsymbol{t}}\right\|^2 = \sum_{i\in[1,k]} \left(\frac{10\eta \lg \frac{\|\boldsymbol{t}_{in} - \boldsymbol{a}_i\|}{\|\boldsymbol{t} - \boldsymbol{a}_i\|} + X}{\sigma}\right)^2 \tag{4.12}$$

which follows non-central chi-squared distribution with k degrees of freedom and non-centrality parameter $\lambda_{\boldsymbol{t}}/\sigma^2$, where,

$$\lambda_{\boldsymbol{t}} = \sum_{i\in[1,k]} \left(10\eta \lg \frac{\|\boldsymbol{t}_{in} - \boldsymbol{a}_i\|}{\|\boldsymbol{t} - \boldsymbol{a}_i\|}\right)^2. \tag{4.13}$$

The target object is classified as non-target area (cf. (4.8)) iff: $\left\|\hat{\boldsymbol{r}}_{\boldsymbol{t}}\right\|^2 > \delta$.

Therefore, the detection rate, i.e., probability that the target object t (called $R(t)$) is classified as non-target area is:

$$\begin{aligned} R(\boldsymbol{t}) &= \mathbb{P}[\left\|\hat{\boldsymbol{r}}_{\boldsymbol{t}}\right\|^2 > \delta] \\ &= 1 - P(\delta; k, \lambda_{\boldsymbol{t}}/\sigma^2) \\ &= Q_{k/2}(\sqrt{\lambda_{\boldsymbol{t}}}/\sigma, \sqrt{\delta}), \end{aligned} \tag{4.14}$$

where δ can be calculated using Equation (4.9), $P(\delta; k, \lambda_{\boldsymbol{t}}/\sigma^2)$ is the CDF evaluated at δ, of a random variable following non-central chi-squared distribution centering at $\lambda_{\boldsymbol{t}}/\sigma^2$ and having k- degrees of freedom. This CDF can be calculated by Marcum Q-function $Q_{k/2}(\sqrt{\lambda_{\boldsymbol{t}}}/\sigma, \sqrt{\delta})$ which is proved to be monotonic [85]. Moreover as δ is a constant, $R(\boldsymbol{t})$ is a monotonic function of $\sqrt{\lambda_{\boldsymbol{t}}}/\sigma$.

The detection rate of a domain D, which is the probability that a trained OC-SVM classifies the target object as non-target area, when the target object positioned at an arbitrary point inside domain D, is:

$$R(D) = \frac{1}{V_D} \int_D R(\boldsymbol{t}) d\boldsymbol{t}, \tag{4.15}$$

where V_D is the volume of domain D.

4.4.2 Stability of The Proposed Formulation

In practice, the signal attenuation due to path loss and its fluctuations due to multi-path fading are often more complicated than suggested by Formula (4.5). To investigate the appropriateness of the proposed Equation (4.14) as well as to analyze factors that affect the detection accuracy, we conduct Monte Carlo simulations under two different propagation models: One following Formula (4.5) and the other following a more advanced propagation model described below.

Advanced Propagation Model

We simulate a propagation environment experiencing Rayleigh fading, non-singular path loss. The RSSI values r under this propagation model are generated via:

$$r = P_0 - 10 \lg(\epsilon + d^\eta) + \mathcal{X} \tag{4.16}$$

where $\epsilon > 0$, d is the distance between two wireless devices, and $\mathcal{X}$ is a random variable with density:

$$f_{\mathcal{X}}(x) = \lambda 10^{x/10} \exp\Big(- \lambda 10^{x/10}\Big) \frac{\ln 10}{10} \tag{4.17}$$

Recent indoor measurements at 2.4GHz [38] have confirmed the above model. where λ is a constant, and here we set $\lambda = 0.561$ because in this case the mean of $\mathcal{X}$ is zero [11, 12]. A meaningful correspondence between $\mathcal{X}$ and our simplified Gaussian approximation X can be established ($\sigma = 5.57$).

Parameter Settings

Assuming that there are three APs located at positions having coordinates of $[0, 0], [0, 10], [10, 0]$ (in meters). The $\boldsymbol{t}_{in}$ corresponding to the target area is set at $[5, 5]$. The fraction of training error ν is set as 0.1. We calculate and compare the detection rate $R(\boldsymbol{t})$ at 20 positions of $\boldsymbol{t}$ that are approximately 3-30m from $\boldsymbol{t}_{in}$.

For each position of the target object, we generated 1000 sets of data, in which each set contained 3 RSSIs from the target object to three APs. In each random realization and for each pair of target object and AP, RSSI is generated randomly under two propagation models given by (4.5) Utilizing random variable $X \sim \mathcal{N}(0, \sigma^2)$ and (4.16) and the random variable X with its probability density described by (4.17) and $\epsilon = 0.1$, and common parameters $\sigma = 0.57$, $\eta = 2$ and $P_0 = -30dBm$. The detection rate $R(\boldsymbol{t})$ for each position $\boldsymbol{t}$ is the percentage of data classified as non-target area.

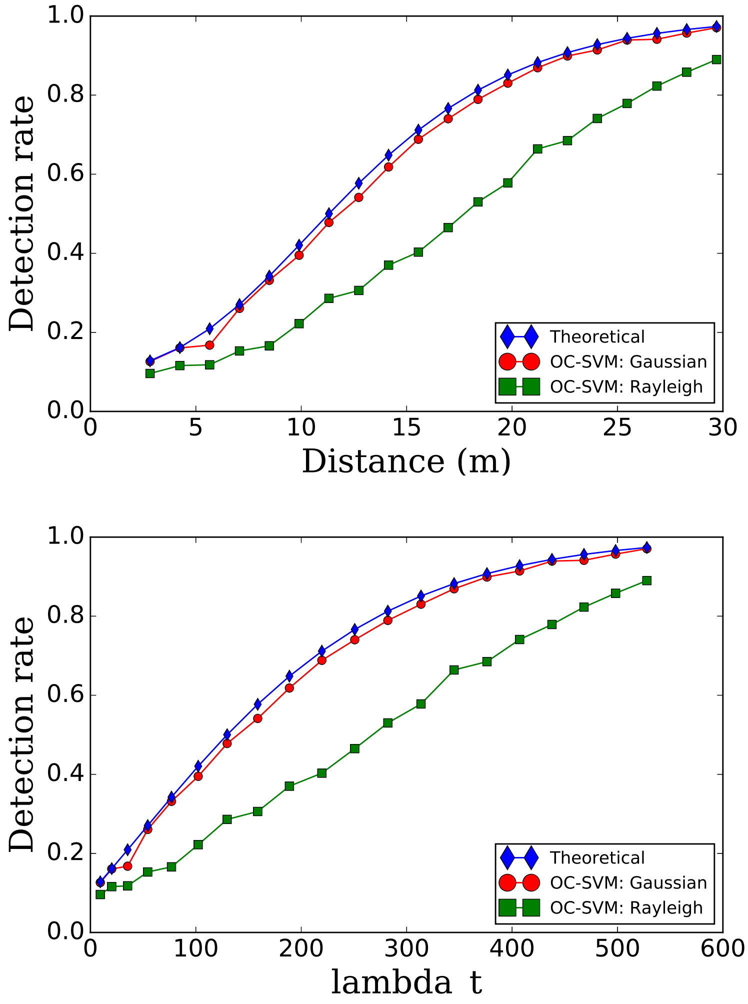


Figure 4.1: Detection rate when the target object moved out of its target area: Blue diamonds describe the rate calculated by the proposed Equation (4.14), red circles and green square illustrate the detection rate achieved using simulation under the assumption that RSSI follows Equations (4.5) and (4.16) respectively. X-axis corresponds to the distance from the target object to the target area in a) (the top figure), and to the λ_t in b) (the bottom figure).

Results

When the signal attenuation due to path loss follows Friis model described by Equation (4.5), and its fluctuations following Gaussian distribution, the detection rate $R(\boldsymbol{t})$ calculated by simulation are closed to the proposed formula (4.14) (shown in Figure 4.1). It indicates that under the assumption that random variables X follows Gaussian distribution, our proposed formula (4.14) is accurate.

When RSSIs follow a different model (e.g. following (4.16)), detection rate $R(\boldsymbol{t})$ by simulations is not close to the proposed Equation (4.14) as it has been derived under the assumption that RSSIs strictly follow Equation (4.5) for simplicity. On the other hand, Fig. 4.1 b) verifies the our claim that the detection rate is a monotonic function of $\lambda_{\boldsymbol{t}}$ regardless of propagation models. It indicates that the accuracy (i.e. detection rate) can be optimized by maximizing $\lambda_{\boldsymbol{t}}$, which provides meaningful insights towards optimizing placement of APs, as well as target area of the object as discussed in Section 4.5.2.

4.5 Optimization Methods

In this section, we propose two optimization methods – capable of improving the detection rate – Utilizing our proposed solution in Equations (4.14) and (4.15). These equations illustrate that for any domain D and any position of the APs, the detection rate $R(D)$ increases when σ decreases. In this work, we hypothesize that σ can be decreased by averaging successive RSSI, as also discussed in Section 4.5.1. Moreover, we also propose a novel mathematical formulation to find optimal placement of APs as well as target areas which maximizes the detection rate of a specific domain D, detailed in Section 4.5.2. The proposed formulation is environment independent, i.e. it can be used in any environment, enabling us to establish optimized placement of APs/target-areas without performing any costly experiments.

4.5.1 Averaging

Averaging successive RSSI can reduce the standard deviation of the signal resulting in less fluctuation and enabling an improved detection accuracy. To illustrate this, we pick signal fluctuations experiencing Rayleigh fading, namely probability density $\mathcal{X}$ following Equation (4.17), for example. The probability density of $\mathcal{X}$ is illustrated by Fig. 4.2 a), and its standard deviation is approximately 5.56. Fig. 4.2 b) shows the probability distribution of averaged five time series successive RSSIs and its standard deviation is now approximately 1.8, which is much smaller than 5.56, the standard deviation of the single signal fluctuation.

4.5.2 Device Placement Optimization

In this section, we discuss the proposed solution to the optimization problem of placing APs/target-areas in order to maximize the detection rate.

Problem Definition

In the context of our application case-study, a store would like to locate a set A_k comprised of k access points (APs) and a set T_m consisting m target objects. Each AP can choose its position from a set A_K consisting K candidate positions, and each target object can choose its target area from a set T_M with M candidate areas. Note that any two APs/objects cannot choose the same candidate position/area. The objective is to choose appropriate positions/areas for APs/objects to maximize the detection rate especially when a target object goes out of the store.

Proposed Method

We first define some symbols. Utilizing a set A_k of APs, we denote $R(D|T_m, A_k)$ as the detection rate of domain D given target areas T_m, which is the value we would like to maximize; $R(\boldsymbol{t}|T_m, A_k)$ as the detection rate when any target object in T_m reaches position $\boldsymbol{t}$; and $R(\boldsymbol{t}|\boldsymbol{t}_{in}, A_k)$ as the detection rate if the target object with its target area being $\boldsymbol{t}_{in}$ reaches the position $\boldsymbol{t}$.

We solve the problem under assumptions described in Section 4.4 and that the store is separated by walls that absorb wireless signals. Therefore the detection rate $R(\boldsymbol{t}|T_m, A_k)$ at any position $\boldsymbol{t} \in D$ (where D is the domain outside the store, see Fig. 4.4) is not smaller than the detection rate $R(\boldsymbol{t}_d)$ of the position $\boldsymbol{t}_d$ which

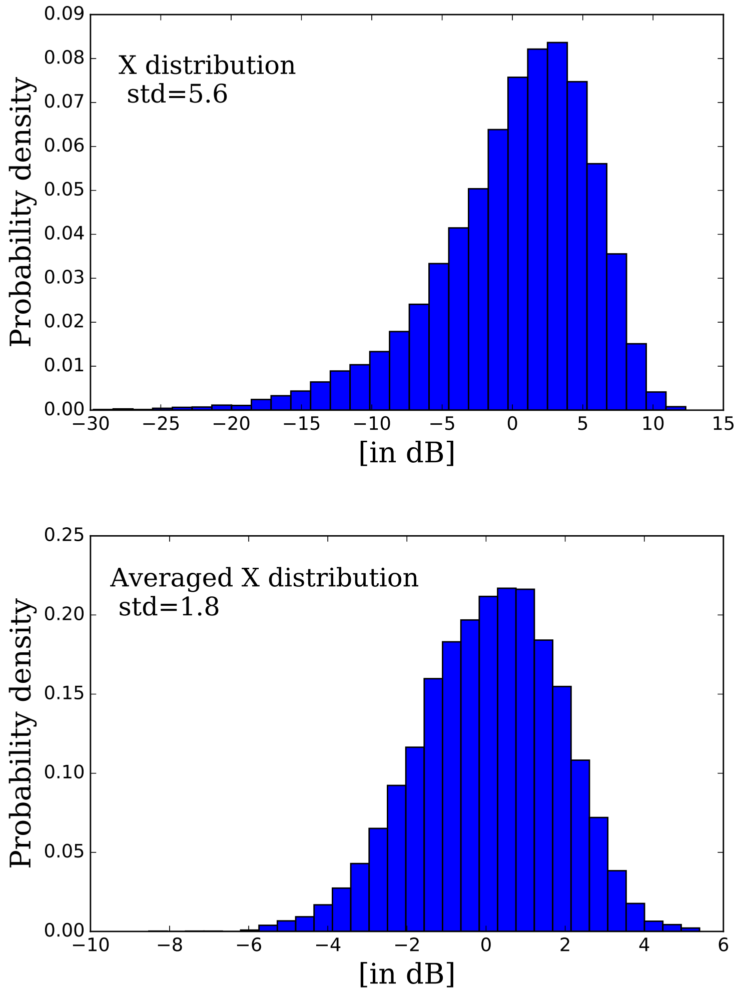


Figure 4.2: Probability density of the RSSI fluctuations due to multi-path fading and noise ($\mathcal{X}$). a) Single RSSI (the top figure), b) averaged 5 successive RSSI (the bottom figure)

is the position in the middle of the gate (see Fig. 4.4). We have:

$$\begin{aligned} R(\boldsymbol{t}|T_m, A_k) \geq R(\boldsymbol{t}_d|T_m, A_k), \quad \forall \boldsymbol{t} \in D \\ \Rightarrow R(D|T_m, A_k) \geq R(\boldsymbol{t}_d|T_m, A_k). \end{aligned} \tag{4.18}$$

therefore maximizing the detection rate $R(D|T_m, A_k)$ is approximately maximizing the detection rate $R(\boldsymbol{t}_d|T_m, A_k)$. Moreover, the store would like to detect if any target object goes outside, thus $R(\boldsymbol{t}_d|T_m, A_k)$ can be defined as:

$$R(\boldsymbol{t}_d|T_m, A_k) = \min_{\boldsymbol{t}_{in} \in T_m} R(\boldsymbol{t}_d|\boldsymbol{t}_{in}, A_k) \tag{4.19}$$

Consequently the objective of the problem is maximizing the right side of Equation (4.19). Therefore, the objective of the problem can be written as follows:

$$A_k^*, T_m^* = \underset{A_k \subset A_K, T_m \subset T_M}{\operatorname{argmax}} \min_{\boldsymbol{t}_{in} \in T_m} R(\boldsymbol{t}_d|\boldsymbol{t}_{in}, A_k) \tag{4.20}$$

Since $R(\boldsymbol{t}_d|\boldsymbol{t}_{in}, A_k)$ is a monotonic function of $\lambda(\boldsymbol{t}_d, \boldsymbol{t}_{in}, A_k)/\sigma^2$ (see Section 4.4), where

$$\lambda(\boldsymbol{t}_d, \boldsymbol{t}_{in}, A_k) = \sum_{\boldsymbol{a} \in A_k} (10\eta \lg \frac{\|\boldsymbol{t}_{in} - \boldsymbol{a}\|}{\|\boldsymbol{t}_d - \boldsymbol{a}\|})^2 \tag{4.21}$$

η is a constant reducing Equation (4.20) to:

$$A_k^*, T_m^* = \underset{A_k \subset A_K, T_m \subset T_M}{\operatorname{argmax}} \min_{\boldsymbol{t}_{in} \in T_m} \sum_{a \in A_k} (\lg \frac{\|\boldsymbol{t}_{in} - \boldsymbol{a}\|}{\|\boldsymbol{t}_d - \boldsymbol{a}\|})^2 \tag{4.22}$$

Thus the optimal positions of APs and target areas can be calculated easily and efficiently – without doing costly experiments or simulations – in order to maximize the detection rate when an object is moved out of the store.

4.6 Experimental Investigation of WiLAD

In order to evaluate the performance of WiLAD in real environments and to validate our optimization methods described in Section 4.5, we performed experiments in a real store environment (see Fig. 4.3).

4.6.1 Experimental Setup

The experiments were conducted at a store in which the area inside and outside the store is approximately 120m^2 and 40m^2 respectively (area of Z5 in Fig. 4.4). The store is separated between inside and outside by concrete walls. Inside the store there are some obstacles such as goods shelves (1.6 m of height), tables (0.8 m of height; see Figs. 4.3 and 4.4). We used multiple Tessera RL7023 Stick-L acting as APs as well as target objects in the experiment, using 920 MHz band. These IEEE802.15.4 standardized devices operate at 926.9 MHz and house a patch antenna transmitting at 13 dB. Four APs were placed at positions described by red points labeled as A1 to A3 (0.8 m of height) and A4 (2 m of height) in Fig. 4.4 . Other three RL7023 Stick-L acting as target objects, could move around their target areas described by blue rectangles labelled as Z1 to Z3 in Fig. 4.4. During the experiment, the target objects broadcasted beacons every second; the APs after receiving the beacons and getting the RSSI would send it to a server for post-processing.

To collect data, we first divided the space of the store into five zones, illustrated by Z1 to Z5 in Fig. 4.4. Z1 to Z3 (shown as the blue rectangles) are target areas corresponding to three target objects. Z4 is the remaining area inside the store and Z5 is the area outside of the store. We then installed wireless devices (RL7023 Stick-L) collecting 4 to 9 sets of data in each zone, and each set containing approximately 400 subsets of data (where each subset contains four RSSIs from the target object to four APs). The experiment was conducted at different times of the day covering a range of business hours from less busy (few people in the store) to very busy (many people in the store).

4.6.2 Evaluation Methodology

Cross Validation

For each pair of target area and non-target area, we used a Leave-One-Out Cross Validation (LOOCV) scheme and calculated the evaluation value (i.e. detection rate or F-measure). For instance, consider that Z1 is the target area, we used 7 of the total 8 sets of data collected at Z1 as the training data, and the remaining set as the test data (positive data), and also the data in non-target area as the test data (negative data).

Detection Rate

Assuming the scenario described in Section 4.5, we estimate the detection rate showing the percentage of detections made by the OC-SVM when a target object goes out of the store, namely, $R(Z5)$. We calculate the detection rate as follows. For each target object (target area), similar to LOOCV described in the previous section, we use $S - 1$ data sets (S is the number of data set for the target area) as the training data. We then use the trained OC-SVM to calculate the percentage of successful detections when the target object stays in Z5 followed by repeating this calculation S times for other target objects and averaging the results. Detection rate is used in Experiment 1 below.

F-Measure

F-measure [86] is defined as follows.

$$\begin{aligned} F &= 2 \cdot \frac{precision \cdot recall}{precision + recall} \\ \text{where} \quad precision &= \frac{tp}{tp + fp} \\ recall &= \frac{tp}{tp + fn}, \end{aligned} \tag{4.23}$$

where tp (true positive) is the number of positive data (i.e. data in target area) classified as target-area; fp (false positive) is the number of positive data (i.e. data in target area) classified as non-target area; fn (false negative) is the number of negative data (i.e. data in non-target area) classified as target area. F-measure is used to evaluate the performance of WiLAD in Experiment 2 below.

4.6.3 Experiment 1

In order to validate our proposed optimization method for installation points given by Equation (4.22) in Section 4.5, we vary the value of $k \in [1, 3], m \in [1, 3]$ which are the number of APs and number of target objects that the store would like to setup, respectively. k APs can choose their positions from 4 candidates depicted by red points in Fig. 4.4. m target objects can choose their areas from 3 areas Z1, Z2, Z3 (namely $K = 4, M = 3$). Similar to the problem described in Section 4.5, the objective is to choose the best combination of k AP positions and m target areas that maximizes the detection rate when one of the target object goes outside the store.

To compare the solutions based on the proposed Equation (4.22) and our experimental solutions, for each values of k, m, we first list all feasible solutions, then sort the list using Equation (4.22) as well as based on the detection rate by experiment described in Section 4.6.2. Here, we set a small fraction of training error ($\nu = 0.02$) to enlarge the sphere volume of the OC-SVM. We then calculate the Pearson's correlation coefficient and its p-value between the two lists.

These results are shown in Table 4.1. For each pair k, m, we list the optimal solution based on the experimental setup followed by the number describing its order based on the theoretical representation (i.e., the proposed Equation (4.22)). Text in bold describe solutions that have the same order in both the experiments and the proposed formulation. For example, when $k = 1$, and $m = 2$, the best solution is A[1], Z[1,2] which means that the detection rate is maximum if the AP is set at A1, and two target objects are set at Z1 and Z2. It is ranked 1 based on our proposed formula and the experimental evaluation. For all values of k, m,

Table 4.1: Experiments vs Theoretical results

k, m	**Solutions**	**Correlation**	**p-value**
1, 1	**A[3], Z[3] (1)**	0.79	2e − 3
1, 2	**A[1], Z[1,2] (1)**	0.66	0.02
1, 3	A[4], Z[1,2,3] (2)	0.60	0.40
2, 1	**A[3,4], Z[3] (1)**	0.92	1e − 7
2, 2	A[1, 3], Z[1, 3] (2)	0.90	3e − 7
2, 3	**A[1, 3], Z[1, 2,3] (1)**	0.89	0.02
3, 1	**A[1, 3, 4], Z[3] (1)**	0.76	4e − 3
3, 2	A[2, 3, 4], Z[2, 3] (6)	0.80	2e − 3
3, 3	**A[1, 2, 3], Z[1, 2, 3] (1)**	1.00	0.00

the correlation coefficient ranges from 0.60 to 1.00 with corresponding $p \leq 0.05$ in most cases, showing that our proposed approach is appropriate. Table 4.1 also shows that 67% of the optimal solutions by the proposed formulation match the optimal solutions by experiments. In some cases, where the proposed formulation produces a different solution can be attributed to various environmental factors that RSSIs experience including multi-path fading, shadowing, and NLoS.

4.6.4 Experiment 2

To evaluate the performance of the proposed WiLAD system in Section 4.3 and to validate our approach described in Section 4.5.1, we performed the experiment under the following scenario. There are three target objects with target areas namely Z1, Z2, Z3. Using three APs positioning at A1, A2, A3 (see Fig. 4.4), we are mainly interested in detecting whether a target object stays inside its target area or goes out of that area. Note that the positions of target areas as well as APs are chosen using the results of the previous experiments: optimal solution for $k = m = 3$.

For each target object, we define its non-target area as, 1) the remaining area of its target area located *inside* the store (i.e. non-target area of the first object is Z2+Z3+Z4. The main purpose is to estimate the decision accuracy of WiLAD under the assumption that the object stays inside the store which is one of the *non-target areas* and 2) the remaining area of its target area (i.e. which is the non-target area of the first object i.e., Z2+...+Z5). This is because, we are interested in estimating the decision accuracy of WiLAD under the assumption that the object stays inside the store or outside the store, called *combined* non-target area. In each pair of target and non-target areas, we calculate the F-measure (see Section 4.6.2) using two type of data: 1) raw data (i.e. use single RSSI, namely set $N = 1$, where N is number of data to be averaged, see Section 4.3) and 2) averaged RSSI using 5 successive data points (i.e. $N = 5$); here called *raw data* and *averaged data* respectively. We set the fraction of training error as 0.1 (i.e. $\nu = 0.1$).

The mean F-measure depicted in Figure 4.5 shows that the averaged data provides significantly better results compared with the raw data in every case (t-test, $p \leq 0.05$). The overall results achieved are always greater than 0.75 illustrating a highly reliable system, further proving our arguments given in Section 4.5.1.

Average detection rate depicted in Figure 4.6 illustrates that the farther the target object goes out of its target area, the higher percentage that WiLAD can detect. Especially, it can detect with the accuracy of 100% when a target

Figure 4.3: Photos of the store, and a close-up of an RL7023 Stick-L as an AP.

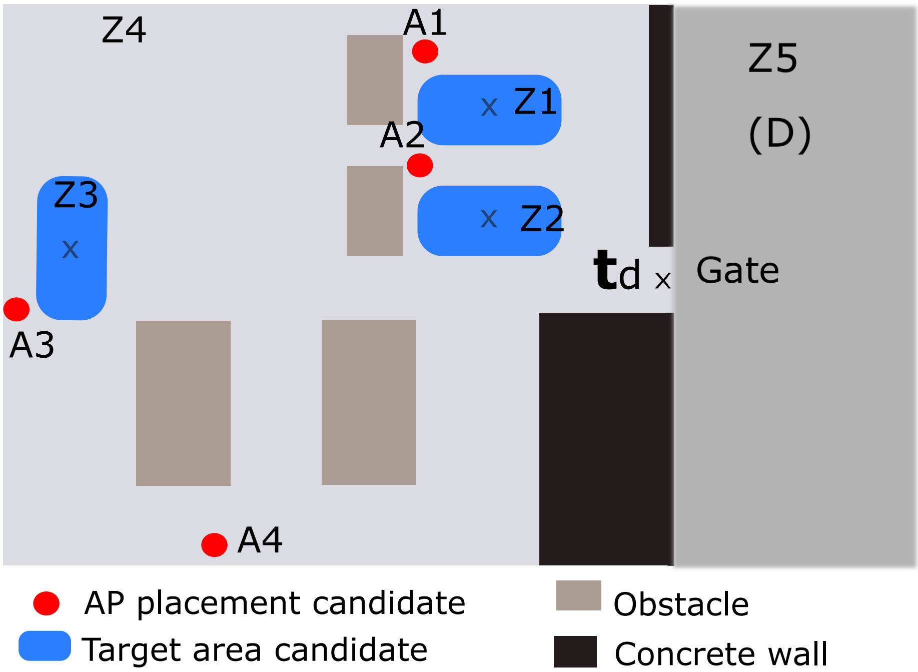


Figure 4.4: Layout of the experimental environment.

node goes outside of the store, thus demonstrating that WiLAD can archive high accuracy enough for security/defense applications, i.e. detecting a device moving out of a store.

4.7 Conclusions

We have proposed a new type of RSSI-based localization method called WiLAD, and in particular addressing the problem of determining whether an object is inside its target area or not. Examples of such scenarios are commonly found in real life, for instance in security, in outlier detection of a wireless sensor network, or in customer analytics. We employed a one class classifier (OC-SVM) to classify an object in either target or non-target areas using its RSSIs to a number of known access points. We also derived an approximation formulation for estimating the accuracy and used it to derive a mathematical framework to optimize device

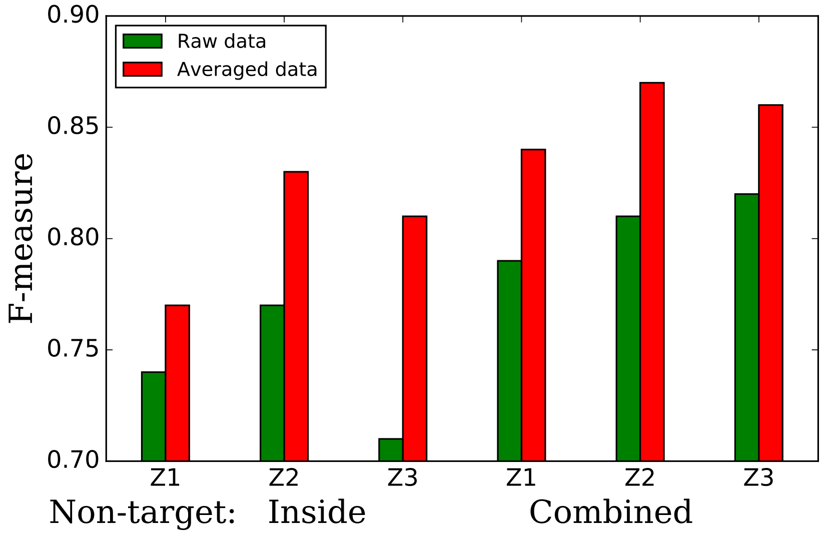


Figure 4.5: F-measure under different pair of target area and non-target area.

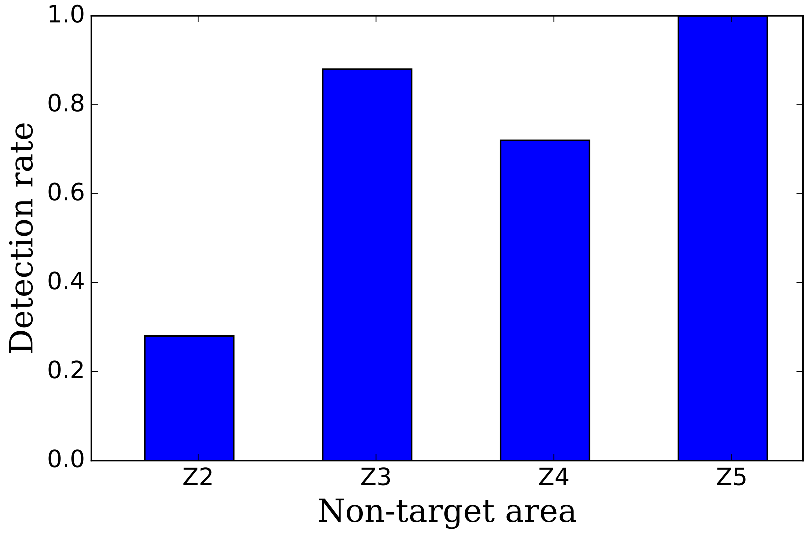


Figure 4.6: Average detection rate in each zone, where Z1 is the target area, and non target areas are Z2, Z3, Z4, Z5.

placements. Finally, we validated our approach through experiments in a real store. The results showed that 67% of the optimal solutions by the proposed method match optimal solutions by experiments. Moreover, the achieved F-measures are always greater than 0.75 illustrating a high reliability. Especially, WiLAD can detect a device moving out of a store with an accuracy of 100%, thus satisfying the requirement of security/defense applications.

We studied from this work that

- RSSI-based localization can achieve satisfactory localization accuracy for some specific applications. This motivates us to further develop new applications related to RSSI-based localization.

- Low-cost device placement guidelines to improve the localization accuracy is realizable without performing a priori measurements. This suggests that the accuracy of a localization system can be improved using low-cost methods.

We, therefore, use RSSI measurements to develop new localization systems described in the next three chapters.

Chapter 5

The Wireless Localization Matching Problem

In the previous chapter, we demonstrate that low-cost RSSI-based localization can achieve satisfactory accuracy of some specific applications. The rapid development of IoT open new opportunities for developing new applications for low-cost wireless localization. We propose a new class of wireless localization of devices belonging to the Internet of Things, called *WLMP: the Wireless Localization Matching Problem.* Immediate examples of real-world applications include but are not limited to smart lighting and heating. This chapter and the next two chapters define, motivate and propose solutions for WLMP. This chapter proposes localization algorithms for resolving WLMP under mesh network paradigms. Chapter 6 presents a method for improving localization accuracy. Chapter 7 proposes a new solution for WLMP, which uses a mobile unit assisting node localization.

WLMP relates to scenarios where the device positions are known a priori, however, the device IDs are not. These positions and device IDs, therefore, need to be matched using radio frequency positioning methods, which are more time and cost efficient as compared to manual installation. In this chapter, we propose maximum-likelihood matching algorithms called MLMatch and MLMatch3D for resolving this problem based on measured RSSI values. Since the search space of node-to-position permutations grows factorially with the number of target devices, we propose several searching methods including Mixed Integer Programming, LP relaxation to reduce computation time. The MLMatch3D algorithm further addresses the problem whereby nodes are located at multiple rooms and/or floors of a building. This algorithm first utilizes a Graph Partitioning method to determine in which room a node is located, followed by MLMatch for finding room specific positions corresponding to each node. In addition, this chapter analyzes the stability of these algorithms with respect to different wireless fading models as well as compares the performance of these algorithms in various environments via numerical simulations. Finally, we report on experiments performed indoors and outdoors using up to 33 wireless devices in order to demonstrate the problem and validate our results. We demonstrate that the proposed algorithm can achieve high localization accuracy satisfied the requirements of real-world applications.

This chapter is organized as follows. In Section 5.1, we define and motivate our proposed WLMP by giving potential real-world applications. In Section 5.2, we describe the system and propagation models. In Section 5.3, we give brief introductions to two branches of combinatorial optimization used later in the chapter. In Sections 5.4 and 5.5, we present the details of our proposed MLMatch and MLMatch3D algorithms. In Section 5.6, we study the overhead and stability analysis of the algorithms. In Section 5.7, we study and compare the algorithms' performance utilizing various searching methods via numerical simulations. In Section 5.8, we evaluate the performance of MLMatch through real experiments conducted in Japan. Finally, Section 5.10 concludes this chapter.

5.1 Definition and Motivation of The WLMP

We propose, analyze, and experimentally demonstrate a new localization problem found in many large scale WSN deployments: *the Wireless Localization Matching Problem* (WLMP) [11]. Unlike most localization problems studied in the literature, here the locations of the sensor nodes are known *a priori*. What is unknown however is the unique ID of the wireless sensor node which is located at each position, or in other words, which node is which. Essentially, the WLMP is a spatially embedded version of the celebrated *matching problem* of probability theory, first proposed by Pierre-Remond Montmort in 1714 [87]: A deck of N cards labeled 1 through N is shuffled randomly. A match occurs when the number on the card matches the card's position in the deck. What is the probability of $k \in [0, N]$ matches? By analogy, the complexities of localizing a device in space in conventional localization problems [46,88] are replaced in the WLMP by those of correctly matching (assigning) N unique device IDs to a discrete set of N device positions.

We motivate the WLMP through a number of real-world application scenarios. In disaster prevention WSNs, sensors are often deployed randomly (e.g., sensors are air-dropped from an airplane) [46], and then their locations are obtained via high resolution satellite photos from the sky. In indoor position tracking systems where cameras provide node positions (e.g., the location of parcels in department stores or warehouses). A final example is related to the mass installation of equipment in factories or office buildings. Whilst the equipment positions are well known from the floor plan blueprints (often found in out-of-reach positions, e.g., behind ceiling panels or rooftops), the specific equipment ID may be difficult to be recorded by installation engineers since it is often printed on small wireless devices used to wirelessly control said equipment. In fact, many wireless light bulbs for example have their ID number printed inside the bulbs, mainly for aesthetic reasons. In practice, to check the ID number of a wireless light bulb after installation, an engineer needs to turn on/off a specific bulb in order to find its position and make a record of it. Either way, even if a specific equipment ID is visible and easily readable, recording it manually is still time and therefore cost consuming. Such scenarios are further intensified when different (sub-)contractors are employed for the installation of heating, ventilation, and air conditioning (HVAC) units, thermostats, security units, lighting, or some other fixed infrastructure IoT. Consider the example of wireless controlled *smart* lighting which is currently commercially available. Installation typically requires manual calibration via a smart phone app, which assigns smart-bulbs to specific rooms (e.g., bulb 001 to living-room, bulb 002 to kitchen, bulb 003 to bedroom, etc.) in a user-editable floor plan. While this is easily performed in residential properties, manual calibration in commercial and industrial buildings containing hundreds of lamps [89] is a taxing and lengthy exercise. This is further complicated since it is difficult to request a specific device to be installed at a specific position, especially when lighting is delivered and installed in bundle. Recalling that lighting in commercial buildings contributes to about 38% of their total energy output, that lighting is responsible for 19% of global electricity consumption and accounts for 6% of total greenhouse emissions, and finally that smart-lighting can reduce these numbers by up to 40% [90], it becomes paramount to devise methods to facilitate the wide-spread adoption of WSNs and green IoT. The WLMP is one such challenge.

Indeed, we are currently seeing more and more smart sensor devices being introduced to the consumer market in the form of place & play clusters of products which monitor and control our homes, our health, our sleep etc. with promises of a better and more efficient lifestyle. A major enabler for these products has been the power efficient ZigBee and Bluetooth 4.x wireless technologies which threads together the Low-Energy IoT. Already, research projects are exploring how such sensors can transition from place & play add-on services to built-in infrastructure and how these can bridge e-health and the IoT [91]. In view of these trends, we expect that the WLMP will soon become widespread in many IoT applications.

The WLMP is a bipartite matching problem between N position candidates and N nodes (the devices) which utilize RF localization methods. In addition, 3D-WLMP is a special case of the WLMP in which the nodes can be located at multiple rooms and floors.

We use RSSI between each of the $N(N-1)/2$ pairs of nodes to calculate the best matching between nodes and position candidates. We chose RSSI-based ranging methods because most WSNs employ cheap devices with low-profile hardware (e.g. compatible with ZigBee and Bluetooth 4.x wireless technologies), no precise clocks, or ultra wide-band (UWB) capabilities. Our first proposed algorithm called MLMatch is divided into two phases: i) a likelihood calculation phase and ii) a maximum likelihood searching phase. In the first phase, the likelihood of a match is reduced into a simple formula which is independent from environment based constants (e.g., path-loss exponent, transmission power) thus rendering our approach applicable to generic propagation environments. In the second phase, to target various applications, we propose several searching methods including exact, approximation and heuristic methods. Our second proposed algorithm called MLMatch3D provides a solution for the 3D-WLMP under the assumption that the signal absorption loss by room wall is unknown and/or uncertain. It also contains two phases: i) a node-room determining phase and ii) a node-position matching phase. In the first phase, the algorithm first constructs a graph consisting of vertices and edges corresponding to nodes and RSSIs, then uses our proposed novel class of Graph Partitioning to determine in which room a node is located. In the second phase, in each room it employs MLMatch to calculate the best matching between nodes and positions for each room. The latter process can of course be trivially parallelised for each room.

The main contributions of this chapter are as follows.

1. We define and motivate the WLMP and 3D-WLMP.
2. We propose and analytically study the performance of MLMatch and MLMatch3D as candidate solutions to the WLMP and 3D-WLMP respectively.
3. We experimentally validate MLMatch and MLMatch3D.

5.2 System and Propagation Models

5.2.1 Problem Definitions

WLMP

Consider a wireless mesh network of N wireless nodes which are labeled $n_1, n_2, ..., n_N$, and are located at N candidate positions labeled $p_1, p_2, ..., p_N$. It is unknown however in which position $p_{i'}$ node n_i is located. Each node is equipped with a

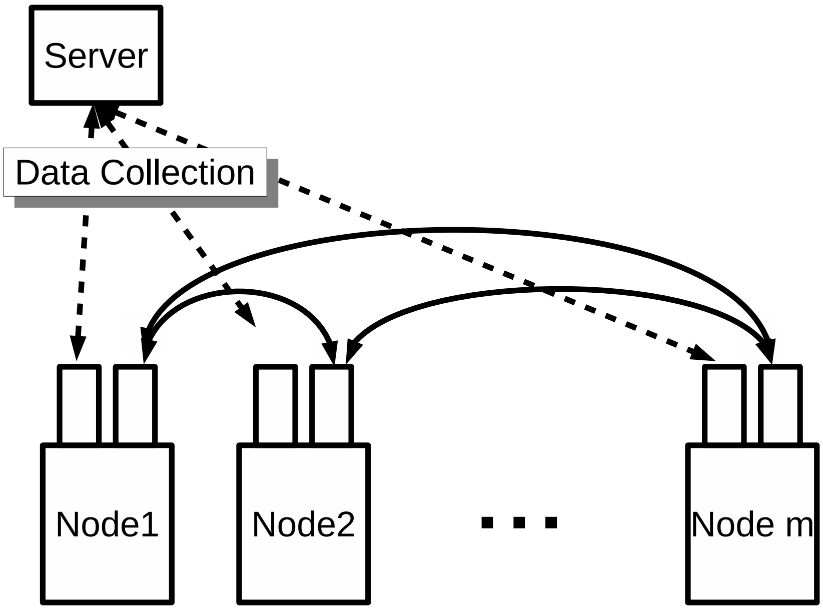


Figure 5.1: The WLMP system setup. Each node is equipped with a radio transceiver thus can obtain RSSI from other nodes by exchanging messages each other. RSSIs are then sent to a backhaul server.

radio transceiver and can exchange messages from other nodes. Thus RSSI between each pair of nodes can be obtained, and sent to a backhaul server for post-processing. After collecting all RSSIs from the wireless nodes, the server saves them in a matrix $\mathbf{R}$ with entries r_{ij} equal to the RSSI between two nodes n_i and n_j. The server is then tasked with matching the node labels n_i to their correct positions $p_{i'}$ using $\mathbf{R}$ and the known distances between positions $d_{i'j'} = |p_{i'} - p_{j'}|$ obtained from a floor plan or similar. Figure 5.1 represents an example of the WLMP system setup. Note that only one node can be located at each position.

3D-WLMP

In 3D-WLMP, the mesh network consists of N wireless nodes and N positions. The N positions are located at l different rooms and/or floors (called rooms for short) labeled $L_1, L_2, \ldots, L_l$. It is known *a priori* that a particular position $p_{i'}$ is located at a particular room L_k, however it is unknown in which position $p_{i'}$ or room L_k node n_i is located at. Similar to the WLMP, RSSI between each pair of nodes can be obtained and used to resolve the matching problem.

5.2.2 Propagation Models

RSSI r is related with distance d, and obstacles such as wall, between wireless devices through the following formula.

$$r = f(d, T) + X \tag{5.1}$$

where $f(d, T)$ is a function characterizing the path loss depending on distance d and transmission coefficient of walls T. When there are no walls between the wireless devices, namely line-of-sight (LoS) $f(d, T) = f(d)$. In Equation (5.1) the parameter X is a random variable characterizing the effects due to multi-path fading and noisy measurements.

LoS Propagation Models

There are various path loss models which are appropriately used in different environments. The log-distance model is widely used and adopted by the scientific community [35]. Here, RSSI r is related with distance d between wireless devices through the log-distance path loss model (in dBm).

$$f(d) = P_0 - 10\eta \log_{10}(d) \tag{5.2}$$

where η is the path loss exponent usually set to 2 for free space wireless propagation and closer to 4 for cluttered or urban environments. P_0 is the reference power at one meter, measured in dB milliwatt. In practice, the signal attenuation due to path-loss are often more complicated than that suggested by (5.2). The singular path-loss function in (5.2) which diverges at $d = 0$ is often replaced by multi-slope and non-singular alternatives [34].

NLoS Propagation Model

When the wireless signal is propagated though a wall, the RSSI r is related with distance d between wireless devices by [92]

$$f(d, T) = P_0 - 10\eta \log_{10} d + 10\eta \log_{10} T \tag{5.3}$$

where T is transmission coefficient through the wall, having value depending on various parameters such as the wall material, thickness. The term $-10\eta \log_{10} T$ is also known as signal absorption loss by a wall, called *wallLoss* for short.

Fading and Noise

The signal fluctuations due to multi-path fading and noise depend on the wireless propagation environment. For example, the long-term signal variation is known to follow the Lognormal distribution, whereas the short-term signal variation can be described by several other distributions such as Hoyt, Rayleigh, Rice, Nakagami-m, and Weibull. Under those distributions, in extreme cases it is possible that the absolute value of random variable X becomes very big, i.e., RSSI between nodes is small even when their distance is close. Such fluctuations can have a significant effect on localization accuracy. Averaging successive RSSI values might be used to weaken these effects, thus improve localization accuracy. Therefore, the availability of multiple independent RSSI measurements enables the use of the Center Limit Theorem (CLT), and thus the modeling of the random variable X by a Gaussian distribution.

Assuming that the distribution of random variable X is characterized by a function $g(X)$, then for a given distance d between a pair of sensor nodes, the probability that RSSI equals a specific value r is

$$\begin{aligned} \mathbb{P}[\text{RSSI}{=}r|\text{distance}{=}d] &= \mathbb{P}\Big[X = r - f(d, T)\Big] \\ &= g(r - f(d, T)). \end{aligned} \tag{5.4}$$

Especially, when the propagation model follows the log-distance model and the random variable X is Gaussian distributed with zero mean and σ standard deviation, we have that the propagation model is given by

$$r = P_0 - 10\eta \log_{10} d + 10\eta \log_{10} T + X \tag{5.5}$$

where $T \in (0,1]$, with $T = 1$ if there are no wall between the wireless devices. Equation (5.4) can be rewritten as follows

$$\begin{aligned}\mathbb{P}[\mathrm{RSSI}=r|\mathrm{distance}=d] &= \mathbb{P}\Big[X=r-P_0+10\eta\log_{10}(d/T)\Big]\\ &= \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(r-P_0+10\eta\log_{10}(d/T))^2}{2\sigma^2}}\end{aligned} \tag{5.6}$$

5.3 Introduction to Combinatorial Optimization

Combinatorial optimization is concerned with the study of efficient algorithms for finding an optimal object from a finite set of objects. In this section, we briefly introduce two branches of combinatorial optimization including Mixed Integer Programming (MIP) and Graph Partitioning used later in our proposed algorithms described in sections 5.4 and 5.5.

5.3.1 Mixed Integer Programming

MIP refers to a mathematical optimization program in which some or all the variables are restricted to integer values. There are a number of MIP computational solvers specializing in linear objectives and linear constraints (called MILP: Mixed Integer Linear Programming), and others for quadratic objectives and/or quadratic constraints (called MIQP: Mixed Integer Quadratic Programming) [93]. A general MILP optimization problem is formulated as follows:

$$\begin{aligned}\min \quad & \boldsymbol{c}^T\boldsymbol{x}\\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\\ & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\\ & \text{and some or all } x \text{ must take integer values.}\end{aligned} \tag{5.7}$$

On the other hand, a MIQP models are of the form:

$$\begin{aligned}\min \quad & \boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \boldsymbol{q}^T\boldsymbol{x}\\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}\\ & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\\ & \boldsymbol{x}^T\boldsymbol{Q}_i\boldsymbol{x} + \boldsymbol{q}_i^T\boldsymbol{x} \leq \boldsymbol{b}_i\\ & \text{and some or all } x \text{ must take integer values.}\end{aligned} \tag{5.8}$$

MIP methods find application in industrial and agricultural production plants (when only integral quantities can be produced), scheduling of transport routes, and resource allocation problems in cellular and D2D networks [94].

The integrality constraints above allow MIP models to capture the discrete nature of some decision. For instance, in the WLMP, a variable whose values are restricted to 0 or 1 (Binary Integer Programming), can be used to decide whether a node located at a specific position or not. The integer nature of MIPs makes them non-convex optimization problem and thus NP-hard in general [95].

To target applications of WLMP that restrict computational time, we develop herein two approximation searching methods (described in more detail in section 5.4.2) utilizing a Linear Programming relaxation (LP relaxation) technique. The LP relaxation technique of a Binary Integer Linear Program is to replace the constraint that each variable must be 0 or 1 by a weaker constraint, namely that each variable belongs to the continuous interval $[0,1]$. This technique transforms a MILP which is NP-hard into a Linear Programming (LP) problem that is solvable in polynomial time. The solution to the relaxed Linear Program can be used to gain information about the solution to the original MILP. However, not only may this solution not be optimal, it may not even be feasible, that is it may violate some of the constraints.

5.3.2 Graph Partitioning

The Graph Partitioning problem is the problem that a weighted graph $G = (V, E)$ with V vertices and E edges, is parted into two or more smaller components as to satisfy a set of specific properties. For example, a $k-$way partition divides the vertex set into k smaller components. A *good* partition is defined as one in which the sum of edges weights connecting the parted components is small. There are a number of Graph Partitioning classes studied in literatures such as the balanced Graph Partitioning problem which tries to divide the vertices of a graph into almost equal size components [96], and the capacitated Graph Partitioning problem in which the maximum of number of nodes in each component is limited [97].

In view of the WLMP, we define a new Graph Partitioning problem in which the number of vertices in each component is restricted to be a pre-defined number. The problem is applied within the MLMatch3D algorithm (see section 5.5) to determine in which room a node is located.

5.4 MLMatch Algorithm

In this section, we describe our proposed MLMatch algorithm for solving WLMP and 3D-WLMP under the assumption that transmission coefficient through the wall T is known *a priori*. The problem is a bipartite matching problem between the N node positions and the N node IDs. Note however that a simple threshold based sorting algorithm would suffer from degenerate solutions and inaccuracies since it would only perform pairwise comparisons between each of the RSSI estimated distances and the real ones. For example, a simple threshold based sorting algorithm could first calculate the $N-1$ pair distances for each node using the known positions $p_{i'}$ (e.g., using the floor blueprint), then sort the distances in ascending order, and finally compare these with those obtained from the RSSI estimated pair distances. Comparison is achieved by calculating the norm of the difference between the two lists for each node position and ID, thus giving a fitness measure. If this difference is less than some small threshold then a match is made. If not all pairs are matched successfully, then the threshold is increased slightly until all are matched. Degenerate solutions and inaccuracies would certainly arise however if the real pair distances are fairly similar to one another (e.g., when nodes are almost symmetrically deployed on a grid for instance). An algorithm that takes into account non-local statistical information is required.

We therefore propose and formulate a likelihood function of the bipartite match between all sensor nodes and their all their positions. We call each possible matching as a *hypothesis* h, where $h(i) = i'$ if node n_i locates at position $p_{i'}$. Note that h is a permutation of the label set $\{1, 2 \ldots, N\}$, and that there are $N!$ possible permutations.

5.4.1 Hypothesis Likelihood Calculation

For an RSSI matrix $\mathbf{R}$, we wish to find the hypothesis that maximises the likelihood function over the hypothesis set H which can be formulated as follows

$$\begin{aligned} h^* &= \operatorname*{argmax}_{h\in H} \mathbb{P}[h|\mathbf{R}] = \frac{\mathbb{P}[\mathbf{R}|h]\mathbb{P}[h]}{\mathbb{P}[\mathbf{R}]} \\ &= \operatorname*{argmax}_{h\in H} \mathbb{P}[\mathbf{R}|h] \end{aligned} \tag{5.9}$$

since $\mathbb{P}[\mathbf{R}]$ and $\mathbb{P}[h]$ are constants. We next make use of the known distances between positions $d_{i'j'} = |p_{i'} - p_{j'}|$ and the independence of the individual RSSI measurements

$$\begin{aligned} \mathbb{P}[\mathbf{R}|h] &= \prod_{i<j} \mathbb{P}[r_{ij}|d_{h(i)h(j)}, T_{h(i)h(j)}] \\ &= \prod_{i<j} g(r_{ij} - f(d_{h(i)h(j)}, T_{h(i)h(j)})) \end{aligned} \tag{5.10}$$

Substituting this back into (5.9) we arrive at

$$h^* = \operatorname*{argmax}_{h\in H} \sum_{i<j} \Big(\ln g(r_{ij} - f(d_{h(i)h(j)}, T_{h(i)h(j)})) \Big) \tag{5.11}$$

Especially, when the propagation model follows the log-normal models, and variable X follows Gaussian distribution, the formula can be reduced to

$$\begin{aligned} h^* &= \operatorname*{argmin}_{h\in H} \sum_{i<j} \left(r_{ij} - P_0 + 10\eta \log_{10} \frac{d_{h(i)h(j)}}{T_{h(i)h(j)}} \right)^2 \\ &= \operatorname*{argmin}_{h\in H} \sum_{i<j} \left(r_{ij} \ln \frac{d_{h(i)h(j)}}{T_{h(i)h(j)}} \right) \end{aligned} \tag{5.12}$$

since $\sum_{i<j} \ln r_{ij}^2, \sum_{i<j} \ln d_{h(i)h(j)}^2, \sum_{i<j} \ln T_{h(i)h(j)}^2, P_0$ and η are constants, and $\ln x$ is the natural logarithm of x. Note that the hypothesis which has the largest likelihood is also the hypothesis in which the Pearson correlation between RSSIs and the logarithm of the distances divided by transmission coefficient is smallest. Note also that if some RSSI values are missing, we may simply ignore them from the computation of (7.5).

Equation (7.5) demonstrates that h^* is independent of the transmit power and other specifics of the wireless propagation environment η and σ. This simplification is due to the maximum likelihood formulation of (5.9) and the Gaussian approximation of X in (5.4), thus enabling the calculation of h^* for any wireless propagation environment without the need to estimate or input further system parameters. For this reason, we will employ formula (7.5) in the following simulations and experiments. This approach is beneficial towards the implementation of MLMatch in real environments, however may be inaccurate in some extreme conditions or when X is far from Gaussian distributed. We will study this aspect both experimentally, and via numerical simulations.

5.4.2 Computing Likelihoods

Equation (5.11), as well as (7.5) is a combinatorial optimization problem of N variables. Solution methodologies for problems can be categorised into three groups: exact, heuristic, and approximation methods. Unlike to exact methods, heuristic and approximation methods typically do not guarantee convergence to the optimal solution however offer significant benefits in terms of computation time which is much less than exact search methods, especially when the number of nodes is large. This makes them appropriate for large networks.

Exact Methods

To improve on the performance of trivial brute-force search, we formulate the problem as an Integer Quadratic Programming (IQP) problem as follows

$$\begin{aligned}
\min \quad & \sum_{i,j,i',j'} c_{i,j,i',j'} z_{i,i'} z_{j,j'} \\
\text{s.t.} \quad & \sum_{i'} z_{i,i'} = 1, \quad \forall i \\
& \sum_{i} z_{i,i'} = 1, \quad \forall i' \\
& z_{i,i'} = 0 \text{ or } 1, \quad \forall i, i'
\end{aligned} \tag{5.13}$$

where $c_{i,j,i',j'}$ are constant values obtained by real measurements, substituted by $\ln g(r_{ij} - f(d_{i',j'}, T_{i',j'}))$ or $r_{ij} \ln(d_{i',j'}/T_{i',j'})$ if utilizing (5.11) or (7.5), respectively. The variable $z_{i,i'}$ equals 1 if node n_i locates at positions $p_{i'}$, otherwise $z_{i,i'} = 0$ (the last constraint). For each node n_i, there is only one candidate position $p_{i'}$ such that the node matches the position, $\sum_{i'} z_{i,i'} = 1$ (the first constraint). Similarly, for each position $p_{i'}$, there is only one node n_i matches the position (the second constraint). The IQP problem defined above is a $(0-1)$ Quadratic Programming of N^2 variables that can be solved by Mixed Integer Programming (MIP) solvers such as CPLEX [93].

We also propose Mixed Integer Linear Programming (MILP) formulation such that it can be addressed by MIP solvers that do not support solving IQP.

$$\begin{aligned}
\min \quad & \sum_{i,j,i',j'} c_{i,j,i',j'} x_{i,j,i',j'} \\
\text{s.t.} \quad & x_{i,j,i',j'} \leq (z_{i,i'} + z_{j,j'})/2, \quad \forall i, j, i', j' \\
& \sum_{i'} z_{i,i'} = 1, \quad \forall i \\
& \sum_{i} z_{i,i'} = 1, \quad \forall i' \\
& x_{i,j,i',j'} = 0 \text{ or } 1, \quad \forall i, j, i', j' \\
& 0 \leq z_{i,i'} \leq 1, \quad \forall i, i'
\end{aligned} \tag{5.14}$$

where $z_{i,i'} = 1$ if the node n_i locates at positions $p_{i'}$, otherwise $z_{i,i'} < 1$ (the last constraint). Due to the fourth constraint and the first constraint, and the negative values of parameters c , $x_{i,j,i',j'} = 1$ if and only if $z_{i,i'} = z_{j,j'} = 1$, otherwise $x_{i,j,i',j'} = 0$. Hence the term $x_{i,j,i',j'}$ can substitute the product $z_{i,i'} z_{j,j'}$ given in the IQP form. Note that if some values of c are positive, subtracting a constant from every c is required before passing to a MILP solver.

Approximation Methods

Since IQP and MILP problems are NP-complete, the computational time needed is at worst exponential. Thus, exact methods are not appropriate for large networks. Instead, we proposed approximation methods by utilizing LP relaxation techniques. Note however that unless $x_{i,j,i',j'}$ is an integer, $x_{i,j,i',j'}$ cannot be substituted for the product $z_{i,i'}z_{j,j'}$. We therefore propose ILP formulation having larger number of constraints.

$$
\begin{aligned}
\min \quad & \sum_{i<j}\sum_{i'\neq j'} c_{i,j,i',j'}x_{i,j,i',j'} \\
\text{s.t.} \quad & \sum_{i'\neq j'} x_{i,j,i',j'} = 1, \quad \forall i < j \\
& \sum_{i\neq j} x_{i,j,i',j'} = 1, \quad \forall i' < j' \\
& \sum_{j'} x_{i,j,i',j'} - \sum_{k'} x_{i,k,i',k'} = 0, \quad \forall j < k, \text{ and } \forall i, i' \\
& x_{i,j,i',j'} - x_{j,i,j',i'} = 0, \quad \forall i < j, i' \neq j' \\
& 0 \leq x_{i,j,i',j'} \leq 1, \quad \forall i < j, i' \neq j' \\
& x_{i,j,i',j'} \text{ is integer}, \quad \forall i < j, i' \neq j'
\end{aligned}
\tag{5.15}
$$

where $x_{i,j,i',j'} = 1$ if the two nodes n_i, n_j locate at positions $p_{i'}, p_{j'}$, otherwise $x_{i,j,i',j'} = 0$ (the two last constraints). This problem is and ILP problem of N^4 variables. For each pair of nodes n_i, n_j, there is only one pair of candidate positions $p_{i'}, p_{j'}$ such that node pair matches the position pair, $\sum_{i'\neq j'} x_{i,j,i',j'} = 1$ (the first constraint). Similarly, for each pair of candidate positions $p_{i'}, p_{j'}$, there is only one matched pair of node IDs n_i, n_j (the second constraint). For every node n_j there is one candidate position $p_{j'}$ such that n_j matches $p_{j'}$. Therefore, for each pair of node n_i and candidate position $p_{i'}$, if n_i matches $p_{i'}$, $\sum_{j'} x_{i,j,i',j'} = 1, \forall j$. Otherwise, if n_i does not match $p_{i'}$, $\sum_{j'} x_{i,j,i',j'} = 0, \forall j$. Consequently, $\sum_{j'} x_{i,j,i',j'} - \sum_{k'} x_{i,k,i',k'} = 0, \forall j, k$ and $\forall i, i'$ (the third constraint). Finally, if a pair of node n_i, n_j matches the pair of candidate position $p_{i'}, p_{j'}$, i.e., $x_{i,j,i',j'} = 1$, then pair of node n_j, n_i also matches pair of candidate position $p_{j'}, p_{i'}$, and vice versa. Therefore, $x_{i,j,i',j'} - x_{j,i,j',i'} = 0, \forall i < j, i' \neq j'$ (the forth constraint).

We next illustrate our two approximation methods basing mainly on LP relaxation. The first method, called *LP method* for short, first solves the LP problem obtained by relaxing the last constraint given in (5.15). Since the solutions may not be integers, it then utilizes the following method to obtain approximate solutions: For all i, i' we let

$$
z_{i,i'} = \frac{1}{N-1}\sum_{\substack{j,j' \\ j\neq i}} x_{i,j,i',j'} \tag{5.16}
$$

where $0 \leq z_{i,i'} \leq 1$ acts as an indicator for the event that node n_i matches the position $p_{i'}$. The closer $z_{i,i'}$ is to one, the higher the probability that node n_i is positioned at $p_{i'}$. We can assemble all this information in a matrix $\boldsymbol{Z}$ with entries $z_{i,i'}$. The approximate method then continues to find a bipartite matching between nodes and positions (namely a hypothesis h where $h(i) = i'$ if node n_i is positioned at $p_{i'}$) that maximize the sum of z over all nodes

$$
h^* = \operatorname*{argmax}_{h\in H} \sum_i z_{i,h(i)}, \tag{5.17}
$$

where H is a set of all hypothesis. The maximal hypothesis h^* can be found efficiently by using the *Hungarian algorithm* for solving Linear Sum Assignment Problems (LSAP) [98]. Note that the original LSAP tries to find the matching that minimizes the cost sum. Hence, by setting the cost matrix as $-z_{i,i'}$, an equivalent problem is obtained.

To check optimality, letting $matchSum = \frac{1}{N}\sum_i z_{i,h^*(i)}$, then it follows that $0 < matchSum \leq 1$ with $matchSum = 1$ iff all solutions obtained by the LP relaxation methods are integer, namely, optimal solutions for (5.15) are obtained.

This LP relaxation method however may be inaccurate, i.e., it produces non optimal solutions in some extreme conditions especially when nodes are positioned in highly symmetric layouts and/or in noisy networks. As a remedy to this and to obtain more accurate results, we propose an approximation algorithm called *MILP method*, in which some variables are constrained to be integers. The algorithm first runs the LP method as described above. If $matchSum = 1$, i.e. optimal solutions are obtained, the algorithm halts and output its solutions. Otherwise, a second run is proposed where the method solves the following MILP problem which includes an additional constraint: $x_{0,1,i',j'}$ is an integer $\forall i', j'$. Thus this is a MILP problem of N^2 integer variables and $N^4 - N^2$ real variables. The algorithm finally concludes with a runs Hungarian algorithm (same as in the LP method) to obtain approximate solutions. Note that this algorithm is faster than the original ILP problem since it has less integers than ILP, but its computation time is at worst exponential.

Heuristic Methods

Combinatorial optimization search methods such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Simulated Annealing (SA) [99] can significantly speed up the computation compared to exact methods. However, it is well known that such methods can fall into local optima, especially if a traditional hill climbing algorithm is assumed. There are many ways of countering these falls including mutation in combination with crossover techniques and noise perturbations which are designed to move the particle populations away from local optima. We will compare the performance of heuristic methods and approximation methods via numerical simulations in section 5.7.

5.5 MLMatch3D Algorithm

In many 3D localization applications, it is difficult to obtain accurate value of transmission coefficient through a wall T since it depends on various parameters such as wall material, thickness, and number of layers [100]. The coefficient T also varies even when wireless signals transmit through the same wall. Measurements have shown that RF signal absorption loss by a wall whose average value is 20 dBm, varies from 10 dBm up to 40 dBm [101]. It is therefore useful to solve the 3D-WLMP under the assumption that T is unknown. To this end we propose an algorithm called MLMatch3D.

The formula (7.5) can be rewritten as

$$\begin{aligned} h^* &= \operatorname*{argmin}_{h\in H} \sum_{i<j} \left(r_{ij} \ln \frac{d_{h(i)h(j)}}{T_{h(i)h(j)}} \right) \\ &= \operatorname*{argmin}_{h\in H} \sum_{\substack{i<j \\ q(h(i))=q(h(j))}} \left(r_{ij} \ln d_{h(i)h(j)} \right) + \sum_{\substack{i<j \\ q(h(i))\neq q(h(j))}} \left(r_{ij} \ln \frac{d_{h(i)h(j)}}{T_{h(i)h(j)}} \right) \end{aligned} \tag{5.18}$$

where we have defined $q(i') = k$ if position $p_{i'}$ belongs to room L_k. Since T only affects the second term in (5.18), we would like to remove this term from the calculation after determining which room L_k a wireless node n_i is located in. We call this process the *node-room determining* problem for short. The problem then reduces to the original WLMP given in section 5.4 and can thus be solved by MLMatch.

We demonstrate our proposed method for solving the node-room determining problem via a Graph Partitioning method. First, we construct a weighted graph (V, E) in which, V consists of vertices $i, (1 \leq i \leq N)$ corresponding to the wireless nodes, and E consists of edges (i, j) connecting two vertices i and j, having non-negative weight $w_{i,j} = r_{i,j} - r_{min}$, where r_{min}(dBm) is the minimum value of RSSI that a wireless node can measure. We set $w_{i,j} = 0$ if there is no connection between nodes between the pair $n_i \sim n_j$. Typically, the RSSI between nodes located in different rooms is small compared to RSSI between nodes located in the same room. Thus, the weight of edges connecting nodes in different rooms will be smaller than the weight of edges connecting nodes in the same room. The Graph Partitioning problem therefore aims to partition the node set of a graph so that the total nodes within each set of the partition is constrained by the *a priori* known number of nodes in each room. The objective is to minimize the sum of the weights of edges having their endpoints in different sets of the partition. We propose ILP formulation for this problem:

$$\begin{aligned}
\min \quad & \sum_{i<j} w_{i,j} y_{i,j} \\
\text{s.t.} \quad & \sum_{k} z_i^k = 1, \quad \forall i \\
& \sum_{i} z_i^k = F_k, \quad \forall k \\
& y_{i,j} \geq z_i^k - z_j^k, \quad \forall i, j, k \\
& y_{i,j}, z_i^k \in \{0, 1\}, \quad \forall i, j, k,
\end{aligned} \tag{5.19}$$

where z_i^k and $y_{i,j}$ are variables, such that $z_i^k = 1$ if vertex i lies in set number k (i.e., node n_i is located in room L_k), and $z_i^k = 0$ otherwise. Further, we have that $y_{i,j} = 1$ if vertices i and j lie in different sets (i.e., nodes n_i, n_j are located in different rooms), and $y_{i,j} = 0$ otherwise (the last constraint). For each vertex i, there is only one set that i belongs to (the first constraint). For each partitioning set, there are exactly F_k vertices (the second constraint), where F_k is the number vertices that lie in set k (i.e., the number of wireless nodes located at room L_k). If two vertices i, j lie in different sets, then $y_{i,j}$ must equal 1 (the third constraint).

This is a $N^2 + N$ variables ILP problem which can be solved using a MIP solver [93]. To target application in which computation time is restricted, we propose a Genetic Algorithm implementation for solving this, which is illustrated in detail in section 5.7.1.

By utilizing the node-room determining method described above, the 3D-WLMP can be solved even when parameter T is unknown or its value is not a constant. Moreover, since the search space of WLMP is parted into smaller sets, the computation time for hypothesis searching can be reduced significantly. On the other hand, when parameter T is close to 1, namely wireless signal absorption loss by the wall $wallLoss$ is small, the method may not work well since it is hard to distinguish between signals travelling through walls from LOS signal. Besides, when T can be obtained accurately, ignoring RSSI between nodes located in different rooms from the calculation can actually reduce the amount of useful information and may in effect reduce accuracy of MLMatch. For these reasons, we recommend to use node-room determining methods when the parameter variation of T is large and/or when the network is large.

5.6 Overhead and Stability Analysis

5.6.1 Overhead Analysis

This subsection discusses computation and communication costs of the proposed MLMatch algorithm.

- Computation cost: The algorithm is divided into two parts, the likelihood calculation part, and the search part. The likelihood calculation of each hypothesis is $\mathcal{O}(N^2)$, where N is the number of nodes. In the search phase, a brute force search would entail $\mathcal{O}(N!)$ complexity, a GA search would depend on initial parameters, LP method would entail in polynomial time, whilst ILP, IQP, MILP would generally be faster than brute force search, and at worst exponential.

- Communication cost: The algorithm requires two transmission broadcasts by each node. Each node after receiving RSSI from all other $N-1$ nodes saves the values locally and forwards them to a central server.

5.6.2 MLMatch Stability

In this subsection, we study the features that can affect the performance of MLMatch under ideal conditions, i.e., assuming that the algorithm can produce global optimal solutions the random variable X follows a perfect Gaussian distribution model, and there are no absorption losses between wireless nodes. Namely, we study the probability that the correct hypothesis h_t is more likely than some other hypothesis h. We therefore compare the likelihoods of the two hypothesis and arrive at

$$\begin{aligned} P_c(h_t,h) =& \mathbb{P}\Big[\sum_{i<j}\big(r_{ij}-P_0+10\eta\log_{10} d_{i'j'}\big)^2 \\ &\qquad > \sum_{i<j}\big(r_{ij}-P_0+10\eta\log_{10} d_{ij}\big)^2\Big] \\ =& \mathbb{P}\Big[\sum_{i<j}\big(X_{ij}+10\eta\log_{10}(d_{i'j'}/d_{ij})\big)^2 > \sum_{i<j} X_{ij}^2\Big] \\ =& \mathbb{P}\Big[\sum_{i<j} 20\eta X_{ij}\log_{10}(d_{i'j'}/d_{ij}) \\ &\qquad > -\sum_{i<j}\big(10\eta\log_{10}(d_{i'j'}/d_{ij})\big)^2\Big] \end{aligned} \tag{5.20}$$

where we have used the fact that

$$r_{ij} = P_0 - 10\eta\log_{10} d_{ij} + X_{ij}. \tag{5.21}$$

Letting $E(h) = \sqrt{\sum_{i<j}\big(\log_{10}(d_{i'j'}/d_{ij})\big)^2}$, the right hand side of the final inequality in (5.20) equals to $-(10\eta E(h))^2$. Moreover, since $X_{ij} \sim \mathcal{N}(0,\sigma^2)$ we have that

$$20\eta X_{ij}\log_{10}(d_{i'j'}/d_{ij}) \sim \mathcal{N}(0,(20\eta\sigma\log_{10}(d_{i'j'}/d_{ij})^2), \tag{5.22}$$

and by the central limit theorem (CLT) we arrive at

$$W = \sum_{i<j} 20\eta X_{ij} \log_{10}(d_{i'j'}/d_{ij}) \sim \mathcal{N}(0, (20\eta\sigma E(h))^2). \tag{5.23}$$

Therefore, we can rewrite $P_c(h_t, h)$ as

$$\begin{aligned} P_c(h_t, h) &= \mathbb{P}\Big[W > -\big(10\eta E(h)\big)^2\Big] \\ &= \frac{1}{2} + \frac{1}{2}\operatorname{erf}\Big[\frac{10\eta E(h)}{\sqrt{8}\sigma}\Big]. \end{aligned} \tag{5.24}$$

Finally, the probability that the correct hypothesis h_t has maximum likelihood over all other hypothesis can be obtained via the theory of order statistics:

$$P_c(h_t, H) = \mathbb{P}\big[h_t = h^*\big] = \prod_{\substack{h\in H,\\ h\neq h_t}} P_c(h_t, h) \tag{5.25}$$

Equations (5.24) and (5.25) suggest that that probability that correct hypothesis h_t has maximum likelihood and is therefore resolved by MLMatch is an exponential function of η/σ. The larger this ratio, the more accurate we expect MLMatch to be. Nodes having distance-closed neighbors can be estimated incorrectly in high probability. In highly symmetric layouts however, $E(h)$, where h is a hypothesis describing a reverse layout corresponding to h_t, can be very close to 0 thus reducing the accuracy of the algorithm. As a mitigation strategy, we propose the external input of at least one node-position pair as to avoid such shortcomings.

5.7 Simulations

In order to substantiate our proposed methods, we have performed a series of numerical simulations in various environments utilizing Monte Carlo simulations. The 1st to 3rd simulations demonstrate the performance of MLMatch described in section 5.4 in various environments, as well as compare its performance utilizing different searching methods described in section 5.4.2. The 4th simulation demonstrates performance of MLMatch and MLMatch3D under various 3D environments.

5.7.1 Implementations and Computer Specifications

Implementations for Searching Methods

To compare the performance of three different searching methods, we have implemented brute-force search for exact method, LP method and MILP method given in section 5.4.2 for approximation methods, and Genetic Algorithm (GA) for heuristic method. For the approximation approach, we have used the software package CyLP which is a Python interface to open source COIN-OR's Linear and mixed-integer program solvers. Further, we have used the primal-dual method for solving LP problem and, Branch & Bound method for solving MILP problem. For the heuristic approach, we have implement GA for the Traveling Salesman Problem (TSP) [102], where the value of $\sum_{i<j}\Big(r_{ij}\ln d_{h(i)h(j)}\Big)$ is substituted for the total travel distance used in TSP, as the evaluation value of a hypothesis h.

Implementations for Graph Partitioning

In general, a GA simulates the survival of the fittest among individuals over consecutive generation for finding optimal solution of a problem. To solve the Graph Partitioning problem described in section 5.5, a GA first codes each individual (here called *ind*) as an N-length vector of components in terms of alphabets $1, 2, ..., l$, where N and l are the number of wireless nodes and the number of rooms, respectively. Each individual corresponds to a partition, in which the i-th component corresponds to wireless node n_i and its alphabet corresponds to the room that it belongs to. For instance, $ind[i] = k$ indicates that nodes n_i locates at room L_k, e.g., an individual $ind = 1122$ corresponds to a partition in which nodes n_1, n_2 and n_3, n_4 belongs to room L_1 and room L_2, respectively. Note that number of each alphabet $k \in [1, l]$ appearing in an individual equals to the number of nodes belonging to room L_k. The GA then utilizes celebrated selection, crossover, mutation operators to search the fittest individual corresponding to the best partition, where the fitness of an individual is evaluated as follows

$$\text{fitness}(ind) = \sum_{\substack{i<j \\ ind[i] \neq ind[j]}} w_{i,j}, \tag{5.26}$$

which is the sum the weights of edges having their endpoints in different sets of the partition.

Computer Specifications

The simulations describe in this section were performed in a computer processor with 4 cores with 2533.492 MHz CPU, cache size of 4 MB and 8 GB memory.

5.7.2 Parameter Settings

Parameter Settings for Propagation Models

We simulate a propagation environment experiencing Rayleigh fading and a non-singular path-loss model. The RSSI values $\bar{r}$ under this propagation model are generated via

$$\bar{r} = P_0 - 10\log_{10}(\epsilon + d^{\eta}) + \mathcal{X} \tag{5.27}$$

where $\epsilon > 0$, and $\mathcal{X}$ is a random variable with density

$$\begin{aligned} f_{\mathcal{X}}(x) = \mathbb{P}[\mathcal{X} = x] &= \frac{\mathrm{d}}{\mathrm{d}x}\mathbb{P}\big[10\log_{10}|h|^2 \leq x\big] \\ &= \lambda 10^{x/10} \exp\Big(-\lambda 10^{x/10}\Big)\frac{\ln 10}{10} \end{aligned} \tag{5.28}$$

where we have assumed that the channel gain $|h|^2$ is an exponentially distributed random variable with mean $1/\lambda$. Recent indoor measurements at 2.4GHz [38] have confirmed the above model. The mean and variance of $\mathcal{X}$ are given by $\mathbb{E}[\mathcal{X}] = -\frac{10(\gamma+\ln\lambda)}{\ln 10}$, and $\text{var}(\mathcal{X}) = \frac{50(6\gamma^2+\pi^2+6(2\gamma+\ln\lambda)\ln\lambda)}{3(\ln 10)^2}$, respectively, where $\gamma \approx 0.577$ is Euler's constant.

We argue that the proposed Gaussian model for X, and the singular path-loss model encapsulated in (5.5) is a sufficient and convenient approximation of the propagation environment for addressing the WLMP using MLMatch. Indeed, while such approximations are known to induce significant errors in conventional

RF based localization techniques [46], and in cellular networks throughput calculations [34], these errors are diluted by the maximum likelihood formulation and the fact that, unlike other localization problems, the WLMP has a finite and discrete solution set. Hence, the mean of $\mathcal{X}$ is zero only when $\lambda = e^{-\gamma} \approx 0.561$ in which case the variance reduces to $\mathrm{var}(\mathcal{X}) = \frac{50\pi^2}{3(\ln 10)^2} \approx 31.025$. Therefore, a meaningful correspondence between $\mathcal{X}$ and our simplified Gaussian approximation X can be established with respect to their first two central moments only when $\lambda = e^{-\gamma}$, and $\sigma = \sqrt{\frac{50\pi^2}{3(\ln 10)^2}} \approx 5.570$.

Parameter Settings for Genetic Algorithm

In each GA simulation run, crossover rate and mutual rate, are randomly generated in the range of $[0.6, 0.95]$ and $[0.005, 0.01]$, respectively. In each selection stage, the best individual among random k individuals is advanced to the next stage. Furthermore, the value of k is generated randomly from the range of $[3, 13]$. Each GA simulation run halts when the evaluation value stays the same for 200 times or the number of iterations reaches 10^4. To prevent GA falling into local optima, for each data set, the best solution among 10 is selected as the output.

Parameter Settings in Each Simulation

To test the robustness of MLMatch described in section 5.4 against different wireless propagation models and to compare the performance of different searching methods, we have performed four simulations employing Monte Carlo simulations.

Simulation 1: In the first simulation, for the number of nodes N ranging from 3 to 20, we numerically randomly generate a set of $N = 100$ different sensor node positions in a 20×20 m^2 square domain. In each random realization and for each pair of nodes, we generate 4 separate pairs of RSSI values. Two using (5.5), and two using (5.27), for different values of $\eta = 3$, and 4, with common parameters $\sigma = 5.570$, $\lambda = e^{-\gamma}$, and $\epsilon = 0.1$. We run MLMatch as defined in (7.5) (i.e., assuming normally distributed RSSI fluctuations) and measure the average accuracy over the N random realizations. This way we can test the robustness of MLMatch against other propagation models. A brute force search approach was performed for $N \leq 13$ followed by a GA method for $14 \leq m \leq 20$.

Simulation 2: In the second simulation, 10 nodes were randomly generated in a 10×20 m^2 rectangle domain under two types of layouts: random layouts (Figure 5.3a)) and highly symmetric layout (Figure 5.4a)). In each random realization and for each pair of nodes, random RSSI values were also generated using (5.5) and for different values of σ in the chosen uniformly from $\{2, 2.5, 3, ..., 10.5\}$, with common parameters $\eta = 3$. We run MLMatch utilizing three methods: brute force (BF), LP method and MILP method. This way we can test the robustness of MLMatch against search methods.

Simulation 3: In the third simulation, for N ranging from 3 to 20, we randomly generate a set of $N = 100$ different node positions in a 10×20 m^2 rectangle domain. For each pair of nodes, RSSI is generated randomly under the model of equation (5.5), in which the path-loss exponent η and the standard deviation of the noisy channel σ are set as 3 and 5.57, respectively. We run MLMatch utilizing two searching methods: LP method, and Genetic Algorithm (GA).This way we can test the robustness of MLMatch against other search methods.

Simulation 4: In the forth and final simulation we test the robustness of MLMatch for the 3D-WLMP described in section 5.5. Here, we have generated

Table 5.1: Description of Simulations

	Simulation 1	**Simulation 2**	**Simulation 3**	**Simulation 4**
Area(m^2)	20×20	10×20	10×20	$4 \times$ 10×20 $\times 2$ rooms
Layout	random	random & high symmetry	random	random
# of Nodes	3 - 20	10	3 - 20	10×2
Path-loss exp.η	3 & 4	3	3	3
NoiseStd σ(dBm)	5.57	2 - 10.5	5.57	6
RSSI models	(5.5) & (5.27)	(5.5)	(5.5)	(5.5)
wallLoss (dBm)	x	x	x	30
wallLoss std(dBm)	x	x	x	0-17
# of Running	100	100	100	100
Search method	BF/GA	BF & LP & MILP	LP & GA	GA

20 nodes randomly in two rooms located at adjacent floors. Each room is $10 \times 20\text{m}^2$ large and 4 m height, contains 10 sensor nodes. Among the 10 sensor nodes, we fix one of the nodes, i.e., we provide its correct position $p_{i'}$. In each random realization and for each pair of nodes, we generate random RSSI values using (5.5) for $\eta = 3$, $\sigma = 6$ and use different values of signal absorption by wall $wallLoss$. The value of the parameter $wallLoss$ is chosen randomly from a Gaussian distribution with mean 30 dBm (namely $T = 0.1$) and standard deviation ranging from 0 to 17 dBm. For each realization, we run MLMatch and MLMatch3D utilizing a GA for hypothesis searching.

The values of these simulation parameters and methods used are summarized in Table 5.1.

5.7.3 Simulation Results

The results of the first simulation are plotted in Figure 5.2b) indicating a clear positive trend as the number of sensor nodes N increases, approaching 100% for larger values of $N \geq 20$. Note however that the simulation accuracy is quite low for $N \leq 5$. We expect that this is due to symmetric position layouts. Moreover, for small N the matching errors are statistically more pronounced since matching errors come at least in pairs. For example for $N = 5$, MLMatch accuracy can only attain values $\{0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, 1\}$, but not $\frac{4}{5}$. These are smoothed out in Figure 5.2b) by averaging over realizations. There is also a clear difference in performance for different path loss exponents η, as expected from the stability analysis of (5.24). Finally, MLMatch appears to be robust against the two fading models X and $\mathcal{X}$ despite them having noticeably different distributions (see Figure 5.2a)).

The results of the second simulation are plotted in Figures 5.3b) and 5.4b) indicating that the accuracy of MLMatch utilizing any of the investigated searching methods decreases as σ increases. In random layouts, while for any value of σ the accuracy of MILP method is very close to the accuracy of brute force method, the accuracy of LP method goes down comparing to the other two methods when σ becomes large. In highly symmetric layouts the accuracy of approximation methods, especially of the LP method, are worse than exact method. This is because, as LP relaxation allows some fractional values, the indicator that a node matches a position $z_{i,i'}$ tends to share its magnitude with the mirror of the position, thus reducing localization accuracy.

The LP method takes averagely 4 seconds to run, while the brute force method takes on average 133 seconds in the second simulation setup. The MILP method has a varying running time depending on the values of σ. When σ is large, the MILP running time increases and may become even larger than that of the brute force method. Our understanding is that when σ is large, especially in highly symmetric layouts, the probability that LP outputs the global optimum decreases and thus the algorithm has to perform further MILP computations which are CPU taxing. Hence, these simulation results illustrate the various trade-off between accuracy and computation time found in the WMLP.

The results of the third simulation are plotted in Figure 5.5. When the number of nodes $N \leq 11$ the accuracy of GA is higher than LP method. In contrast, for $N > 11$ the LP method tends to give better results. This is because for large N the search space increases exponentially, and hence it is easier for GA methods to fall into local optima if the sampling size is kept the same. Increasing number of iterations and/or the number of running GA (thus running time also increases) might alleviate this shortcoming. Recall that the running time by LP method increases polynomially with N, and sub-linearly for GA. Our simulations suggest that this transition is at about $N \approx 17$ nodes when the computation time of LP methods becomes larger than GA.

The results of the second and third simulations hint towards choosing the right searching method as appropriate for certain applications and certain networks. For example, if the application has a constraint running time, LP methods or GA with small number of iterations can be used. If the application requires high accuracy, exact methods or GA with large number of iterations might be more suited.

The results of the fourth simulation are plotted in Figure 5.6. When $wallLoss$ standard deviation is less than 7, the accuracy by MLMatch is higher than that of MLMatch3D, while for larger than 7, MLMatch3D tends to give better results. This is because by eliminating instances where RSSI is available for pairs of nodes in adjacent rooms, MLMatch3D actually loses useful information. This can reduce the algorithm's performance. Note however that when $wallLoss$ varies dramatically, utilizing RSSI between rooms actually can also influence negatively the matching accuracy.

Finally and as expected, the running time by MLMatch3D is much lower than MLMatch. This is because MLMatch3D divides the search space into smaller compartments.

5.8 Experimental Investigation of The WLMP

5.8.1 Prototype Implementation

In order to validate our proposed solution to the WLMP, we have performed a series of experiments in our premises in Japan, utilizing the 920MHz band. Figure 5.1 illustrates our network system consisting of a server and multiple wireless nodes. The server which is an ordinary PC, is used to collect RSSI data and run MLMatch algorithms. Each wireless node (shown in Figure 5.10) is a Raspberry-pi 2 acting as a host connected to a wireless transceiver of TESSERA RL7023 Stick-L (RL7023 for short) used to interchange message among nodes utilizing the 920MHz band (IEEE 802.15.4), and a Wi-Fi adapter of Planex GW-450D used to control the experiments: RSS data collection start/stop, and 920MHz wireless parameters control. RSSI value measurements are only done within RL7023 which is equipped with a RF of ADF7023-J operating at 926.9 MHz and housing a patch antenna transmitting at 13dB. The TESSERA RL78/G13 microcontroller of a RL7023 is programmed to do a simple Trickle [103] multicast where each RL7023 broadcasts at least one message within an pre-defined time interval. Once a RL7023 has received a message, it measures RSSI and reports to the host via USB serial port. The host then sends RSSI to the server immediately via equipped Wi-Fi adapter. We defined time interval as 10 seconds, namely a matrix $\mathbf{R}$ with $N \times N$ entries $r_{i,j}$ can be collected within 10 seconds. In Experiments 3, and 5 prototype Toshiba wireless devices were used as wireless nodes (not shown) instead of RL7023s, operating at three channels: 922.5MHz, 924.5MHz, and 927.7MHz.

5.8.2 Description of Experimental Setup

In all cases, the devices were placed at specific positions throughout the various experiments. The layouts of these six experiments are depicted in Figures 5.7 and 5.8, photographed in Figure 5.9, and described in Tab. 5.2. Especially, detailed results of Experiment 6 are described in Figure 5.12.

To verify the MLMatch algorithm described in section 5.4 in various constraints, we varied the number of nodes from 10 to 33 in different layouts, and in different deployment areas ranging from 9m^2 up to 4100m^2 in indoor and outdoor environments. To induce different levels of noise and path loss in our measurements, we experimented in different locations, different node positions, different environments, and different number of channels. For example, to induce variations in our measurements, we placed the wireless nodes inside ceiling tiles (Experiment 1), on tables (Experiments 2, 5), on the floor (Experiments 3, 6) and also on the grass outside (Experiment 4).

To verify the MLMatch3D algorithm described in section 5.5, we deployed 30 nodes in two adjacent office rooms (see Figure 5.8) partitioned by a concrete wall whose average absorption loss ($wallLoss$) is approximately 16 dBm, and the standard deviation of $wallLoss$ approximately 9 dBm. Not all 30 nodes where used simultaneously. Instead, from the 30 nodes, we randomly choose $N = \{7, 9, 11, \dots, 19\}$ nodes and run the two algorithms MLMatch and MLMatch3D separately. This was performed 50 times in order to calculate and compare the average localization accuracy of each algorithm.

Finally, our RSSI matrix $\mathbf{R}$ and path-loss model was constructed as follows. For each pair of nodes, the maximum value of RSSI over 10 measurements was collected. This approach gave the lowest values of σ compared to other RSSI values such as min, mean, median. Using these measurements we numerically fitted RSSI to (5.5) and obtained estimates of η and σ for each environment (see Tab. 5.2 and Figure 5.11). These were then used to calculate h^* using (7.5). In layouts which had a high degree of symmetry (e.g., Experiments 1, 4, and 5), we provided the algorithm with one true node-position in order to prune the solution set.

5.8.3 Experimental Results

Figure 5.7 depicts the estimation results of the MLMatch algorithm for the first five experiments. Red circles and blue circles describe nodes that are estimated correctly and incorrectly by MLMatch, respectively. In Experiments 1, and 5, MLMatch algorithm produced 33%, and 30% incorrect answers, namely, three nodes and ten nodes are estimated to be located at wrong candidate positions, respectively. In Experiments $2, 3$, and 4, MLMatch had 100% correct answers.

Table 5.2: Description of Experiments

	Experiment 1	**Experiment 2**	**Experiment 3**	**Experiment 4**	**Experiment 5**	**Experiment 6**
Environment	office	office	office	park	office	3D office
Node positions	ceiling	on desks	floor	ground	on desks	floor
Surroundings	many obstacles + people	many obstacles + people	few obstacles no people	no obstacles no people	many obstacles + people	no obstacles no people
Layout symmetry	high	medium	medium	very high	high	random
Number of Channels	1	3	1	1	3	1
Number of Nodes	10	12	14	16	33	7 - 19
Area scale($\mathbf{m^2}$)	200	270	4100	9	700	160
Path-loss exponential η	1.7	1.8	3.9	4.1	1.9	2.6
Std of Noise σ (dBm)	5.0	3.1	6.6	3.2	5.3	4.7
Ratio η/σ	0.34	0.57	0.59	1.28	0.37	0.55
Averaged *wallLoss* (dBm)	x	x	x	x	x	16
Std of *wallLoss* (dBm)	x	x	x	x	x	9
Searching method	BF	BF	LP	LP	GA	BF & GA
Algorithm accuracy	**67 %**	**100%**	**100%**	**100%**	**70%**	**55%-75%**

As suggested by our analysis in section 5.6.2, environments in which the path-loss exponent is small, signal fluctuations are high and MLMatch tends to produce inaccurate answers. This is indeed what we have observed in our experiments. Experiments 1 and 5 having low path-loss exponent to measurement noise data η/σ are 0.34 and 0.37, displayed poor results in terms of matching accuracy. In contrast, Experiment 2 which had a very similar path-loss exponent to Experiment 1 managed to localize correctly all devices by utilizing more channels and positioning the devices on desks as to avoid strong ceiling multi-path reflections. On the other hand, high path-loss exponents are capable of countering noisy environments such as in Experiment 3, further mitigated by an average level of layout symmetry. In Experiment 4, the path-loss exponent was high and the level of noise was low thus leading to accurate localization matching results. As a general rule of thumb, we have observed that a high ratio η/σ is a good indicator of the algorithm performance.

Figure 5.12 depicts the estimation results of the MLMatch3D vs MLMatch for the last Experiment 6. On average, MLMatch3D gave better accuracy as well as running time than MLMatch. This was because the standard deviation of $wallLoss$ in this case was quite large (approximately 9dBm). Focusing on MLMatch3D, the Graph Partitioning method gave 100% correct results, namely, the method could determine in which room a node is located correctly. In each room, the algorithm then utilizes MLMatch freely as to estimate in which location nodes are positioned. Therefore, MLMatch3D's accuracy equals to MLMatch in each room. From this we may transfer a lot of the engineering insights developed in Experiments 1-5 for MLMatch. For instance, the accuracy of MLMatch3D depends on the η/σ ratio and the degree of layout symmetry. Even when the ratio η/σ is quite high (0.55), the average accuracy by MLMatch3D is lower than in other experiments because of the layout symmetry effect causing degenerate solutions.

We can draw multiple conclusions from the above experiments:

- The proposed localization algorithms can achieve room-level localization accuracy of 100%.
- The proposed algorithms can achieve nearly 100% of localization accuracy in less noisy environments. Besides, they can determine coarse location of nodes in noisy environments.
- Relationship between the localization accuracy and the ratio η/σ is validated.

5.9 Applications

Similar to conventional RSSI-based localization techniques, the proposed localizations MLMatch and MLMatch3D do not always achieve localization accuracy of 100% in every environment. However, we demonstrate that the achievable accuracy is enough for various real-world applications. This section briefly describes some potential applications.

Room-level devices control and management applications: In many applications, for instance, smart house applications, wireless devices such as light bulbs, air conditioners are often controlled and managed at room or zone level. For instance, the followed commands are often found in smart home scenarios: "Turn off all light bulbs in the living room", "Turn on an air conditioner in the bedroom", etc. Since MLMatch3D can achieve localization accuracy of 100% at room-level, it can be applied in such applications to automatically match wireless devices with their positions. Similarly, some applications may require zone-level accuracy, for instance, to control a light bulb in the north of room number one to be turned on, etc. Since MLMatch and MLMatch3D can produce coarse location of wireless devices, they can be applied in those applications.

Localization in wireless sensor networks (WSNs): As described in Section 5.1, MLMatch can be combined with image processing techniques to determine the location of sensor nodes in a WSN. In many WSN applications, coarse locations of sensor nodes are enough for some applications such as sensed data processing, wireless routing, and wireless node clustering, etc.

Assistance to reduce manual calibration cost: As described in Section 5.1, manual calibration in commercial and industrial buildings containing hundreds of wireless devices, for instance, light bulbs, is a taxing and lengthy exercise. The proposed MLMatch and MLMatch3D, which can automatically determine the proximity of devices, can teach labors nearby devices which should be calibrated. This can reduce manual cost compared to the scenario where there is no prior information on the vicinity of these wireless devices.

5.10 Conclusions

While RF localization has come a long way [88, 104], there are many unconventional localization problems which remain unexplored. In this chapter, we have proposed a new type of localization problem for use in WSNs: the wireless localization matching problem (WLMP), which is a bipartite matching problem between perfectly known sensor node positions and their unknown IDs via wireless RF positioning methods. Examples of such scenarios are commonly found in real life, for instance during mass equipment (e.g., smart-lighting) installations in commercial or industrial buildings which are currently time and cost inefficient.

The solution to the WLMP is hindered by noisy wireless measurements and symmetric deployment layouts. We have therefore proposed the use of maximum likelihood-based algorithms called MLMatch and MLMatch3D, and also several maximum likelihood searching methods. The main advantage of the MLMatch algorithms is the use of likelihood estimation to reduce the error rate. Flexibility in searching methods is also an advantage as accuracy can be traded off with localization speed according to the application setting. In a 3D setting where nodes are located within a multi-room or multi-story building, the main advantage of MLMatch3D is that it can solve the 3D-WLMP even when the $wallLoss$ signal attenuation through walls is uncertain or unknown. Not only have we proposed and defined the WLMP, but we have also numerically and experimentally analyzed the main factors that affect the accuracy of our algorithms. Extensive experiments in various environments illustrated that the proposed algorithms can achieve high localization accuracy enough for a number of real-world IoT applications. For instance, the proposed MLMatch3D which can achieve 100% room-level of accuracy can be successfully applied into IoT devices control and management at room-level. We have also discussed some of the algorithms' limitations and suggested methods to improve their performance. In the next chapter, we will introduce a method that uses machine learning to improve the accuracy of the proposed algorithms.

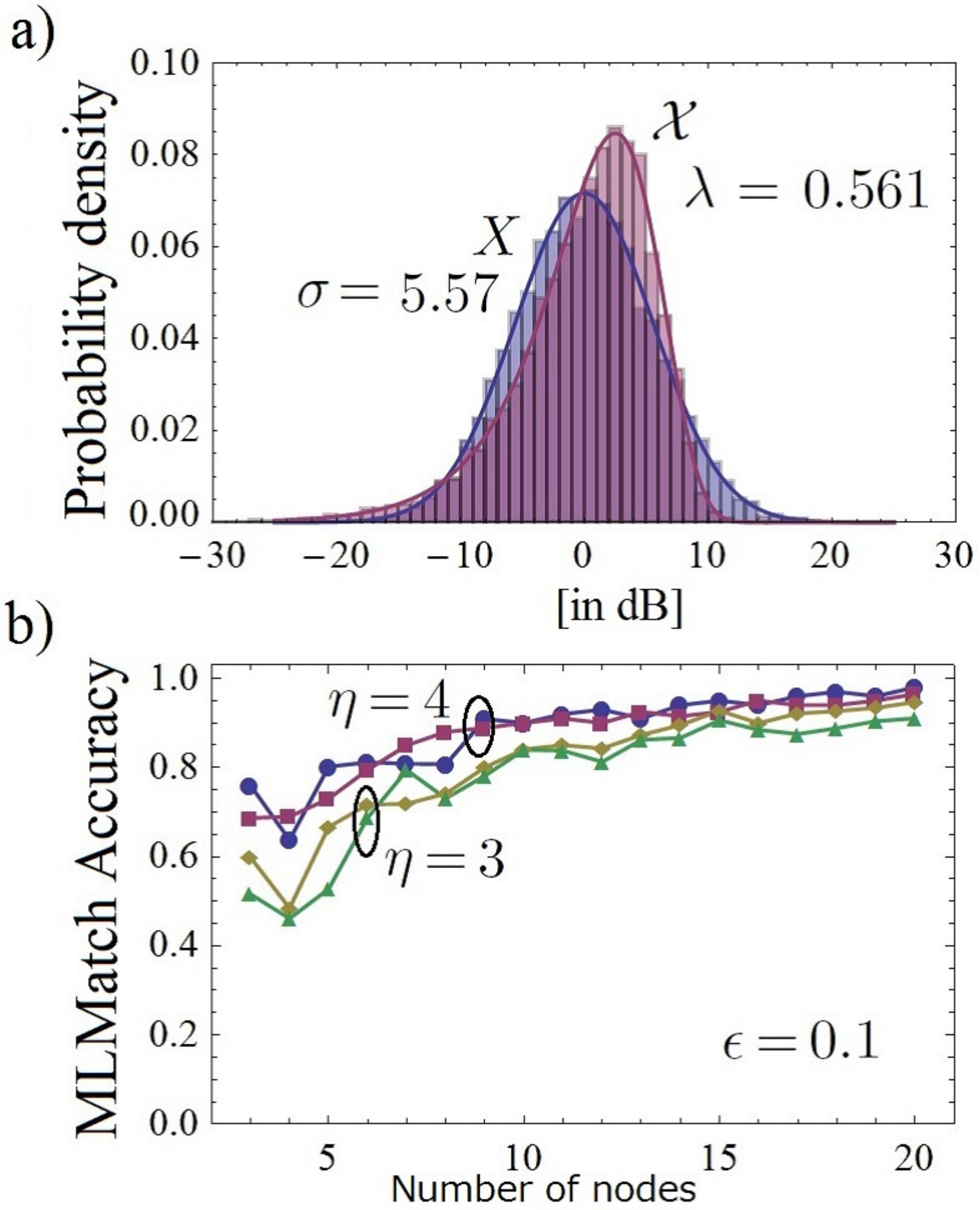


Figure 5.2: a) Probability density of the RSSI fluctuations due to multi-path fading and noise, modeled here as a normally distributed random variable X (blue), and also as a Rayleigh fading random variable $\mathcal{X}$ (purple). b) MLMatch accuracy as a function of N randomly positioned nodes in a 20×20 m^2 square domain, for different RSSI fluctuation models X (blue circles and yellow diamonds) and $\mathcal{X}$ (purple squares and green triangles) and path-loss exponents.

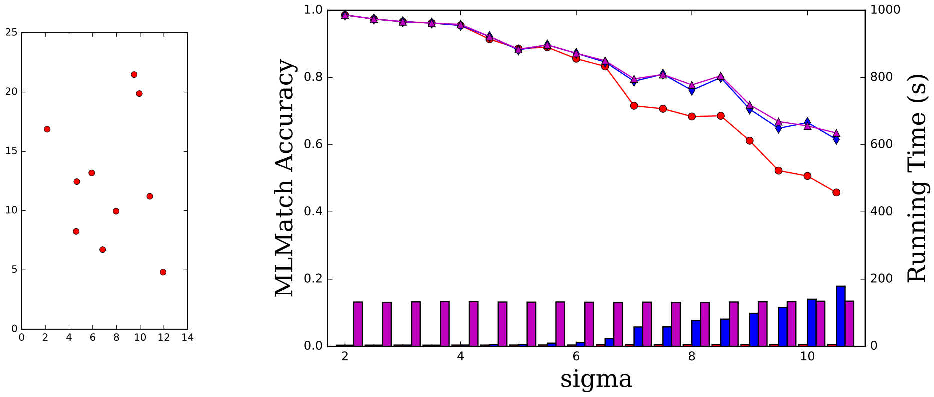


Figure 5.3: a) An example of a Random Layout (left), b) MLMatch accuracy (lines) and running time (bars) utilizing LP method (in red), MILP method (in blue) and brute force method (in purple). Nodes are randomly positioned in a $10 \times 20\text{m}^2$ rectangle domain. (right)

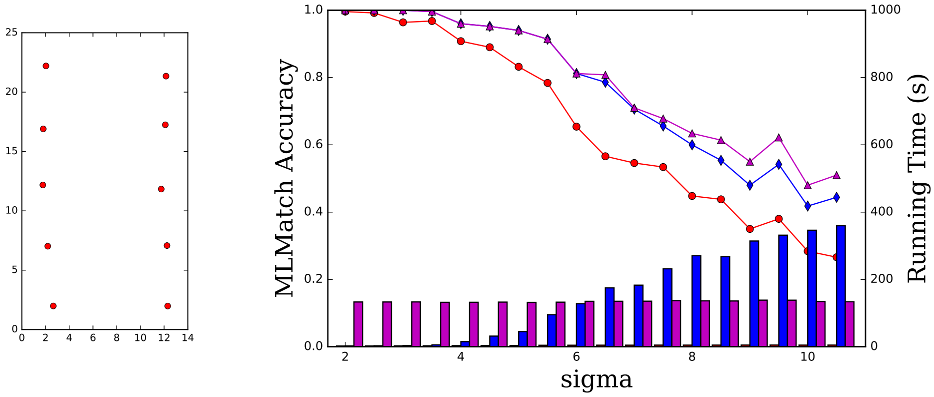


Figure 5.4: a) An example of a Highly Symmetric Layout (left), b) MLMatch accuracy (lines) and running time (bars) utilizing LP method (in red), MILP method (in blue) and brute force method (in purple). Nodes are randomly positioned in $10 \times 20\text{m}^2$ rectangle domain, under assumption of highly symmetric layout. (right)

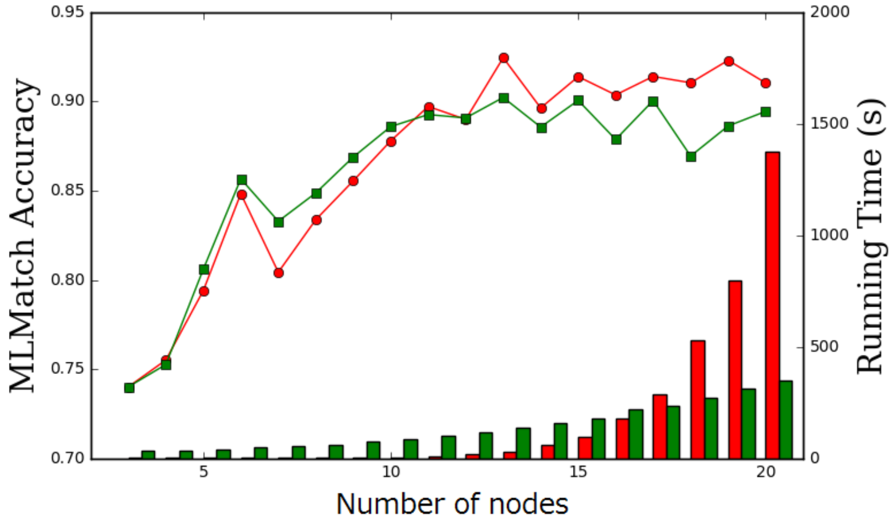


Figure 5.5: MLMatch accuracy (lines) and running time (bars) utilizing LP method (in red) and GA (in green).

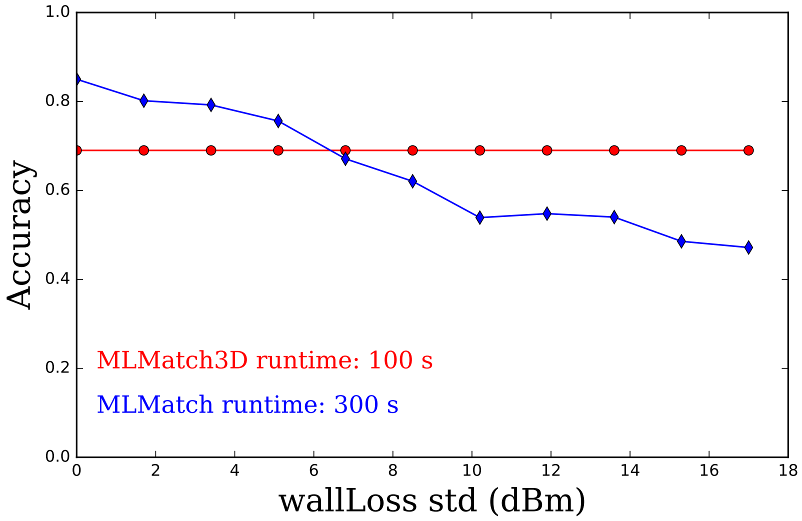
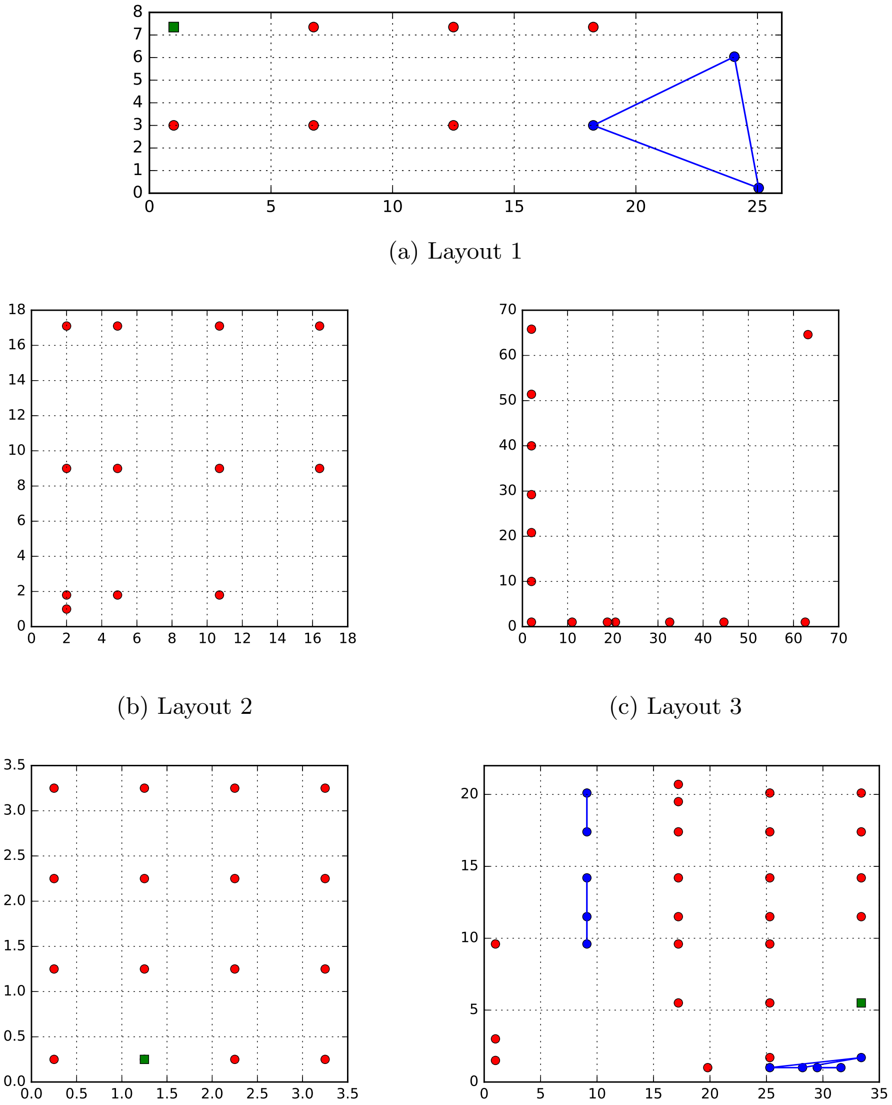


Figure 5.6: Accuracy and running time of MLMatch3D (in red) and MLMatch (in blue) through simulations.



(a) Layout 1

(b) Layout 2

(c) Layout 3

(d) Layout 4

(e) Layout 5

Figure 5.7: Layouts of deployed nodes and estimation results. Coordinates describe length in meter. Green rectangles depict fixed nodes. Red circles and blue circle depict nodes that are estimated correctly and incorrectly, respectively. Correct node positions and estimated positions are connected by blue lines.

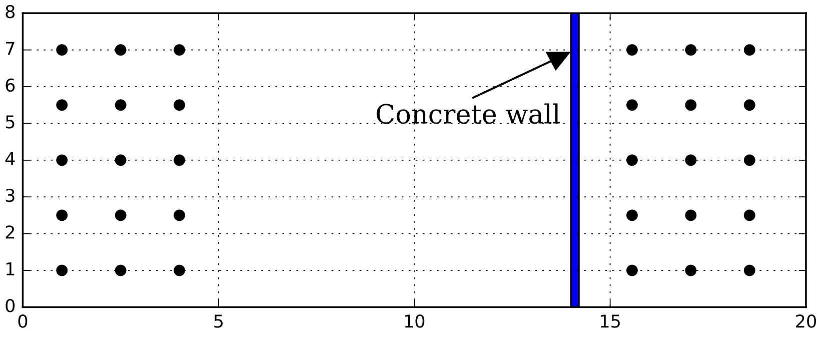


Figure 5.8: Layout of deployed nodes in Experiment 6.

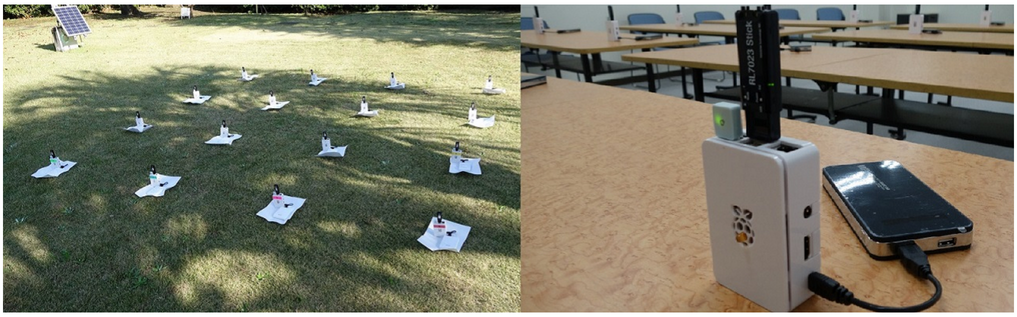


Figure 5.9: Photos of Experiments 4 and 5.

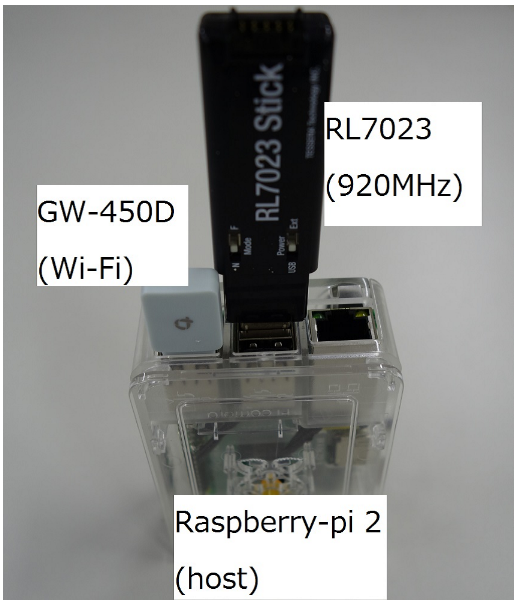


Figure 5.10: A wireless node equipped with RL7023 Stick-L

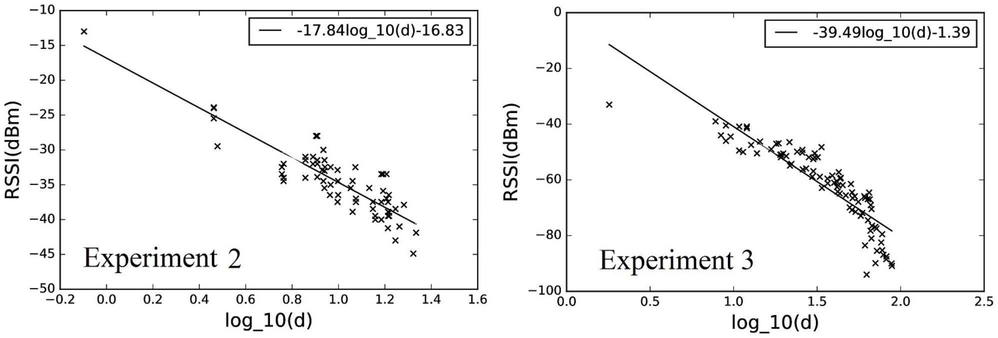


Figure 5.11: Linear fitting of the path loss exponent for Experiments 2 and 3.

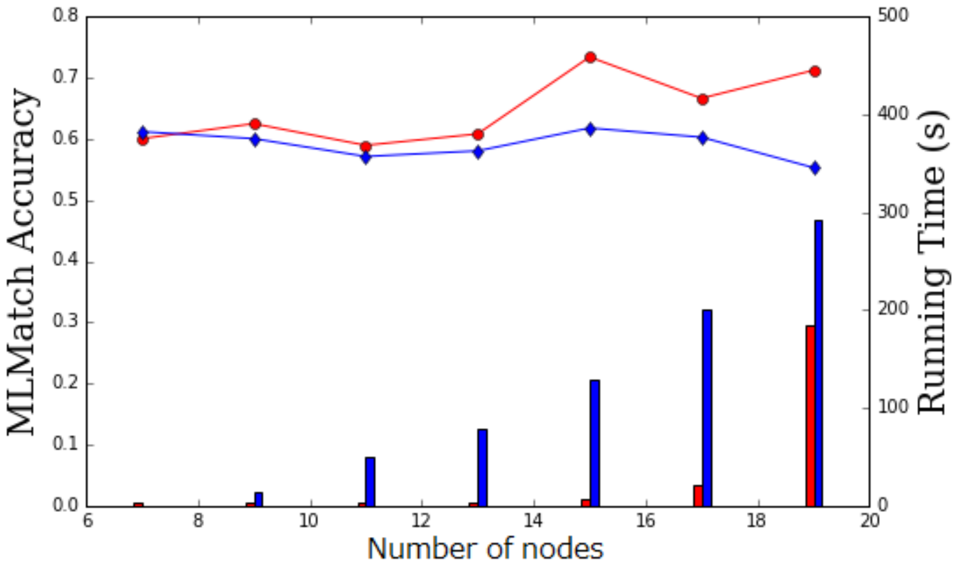


Figure 5.12: Accuracy (lines) and running time (bars) of MLMatch3D (in red) and MLMatch (in blue) through experiments.

Chapter 6

Improved Localization Accuracy: Predicting and Refining RSSI

In the previous chapter, we defined and promoted a new class of wireless localization called *WLMP* (i.e., the Wireless Localization Matching Problem). We also proposed algorithms called MLMatch to resolve the problem. Although MLMatch algorithms are practical because they do not require a priori measurements, their accuracy are related to the variation of the noisy factor. In many cases in practice, however, calibration of the propagation model which is measured during a priori measurements, can be conducted. Using the propagation model, we can improve the localization accuracy of MLMatch. In this chapter, we propose a method called *MLRefine* (i.e., Refining RSSI measurements using Machine Learning) to reduce the variation of the noisy factor, thus enhancing localization accuracies.

MLRefine uses machine learning methods to model the relationship between accurate values and features extracted from in silico RSSI values. MLRefine then applies the trained model to features extracted from real RSSI measurement values to return a predicted set of refined RSSI values. The refined RSSI values are shown through computer simulations and real experiments to improve localization accuracies of state-of-the-art location estimators including the proposed MLMatch described in the previous chapter.

This chapter is organized as follows. In Section 6.1, we illustrate the importance of refining RSSI measurements in wireless localization. In Section 6.2, we discuss conventional related work on noise reduction techniques and localization techniques. In Section 6.3, we define the RSSI refining problem and describe our proposed MLRefine. In Section 6.4, we substantiate MLRefine through computer simulations. In Section 6.5 we investigate and validate the performance benefits of MLRefine through real experiments. Finally, Section 6.6 concludes this chapter.

6.1 Motivation

RSSI based localization methods are subject to much errors due to background noise, wireless multi-path fading, shadowing, non line-of-sight (NLoS), path loss, etc. Hence, empirical models are often evoked to improve the accuracy of localization methods. These are sometimes costly and time-consuming since model parameters need to be adjusted for each specific WSN deployment environment. At their core, most range based RSSI localization methods require a conversion function f that relates the distance d between two wireless devices with the RSSI value r. Most range based RSSI localization methods require a ranging function f that relates the distance d between two wireless devices with the RSSI value r.

This can be expressed in dBm through the log-distance propagation model [35] as

$$r = f(d) + X = P_0 - 10\eta \log_{10} d + X \tag{6.1}$$

where P_0 is a known reference power value at distance of 1 meter from the transmitter. η is the path loss exponent usually set to 2 for free space wireless propagation or can be experimentally fitted via linear fitting (cf. Fig. 6.1a)). X is a random variable that captures the statistics of the RSSI spatio-temporal fluctuations and therefore the inherent error of each data measurement. X is normally assumed as a zero-mean Gaussian random variable (in dB), i.e. $X \sim \mathcal{N}(0, \sigma^2)$. RSSI based localization methods are subject to much errors due to background noise, multi-path fading, shadowing, etc. A calculation in Section 2.4 however can show that the ranging error per node-pair grows exponentially with the variance σ^2 of X, and specifically with the ratio σ/η (cf. Appendix).

Non-zero values of X cause ranging error and thus localization error. The Cramer-Rao lower bound (CRLB) can give a lower bound on the covariance of localization error for a given environment is proportional to the ratio σ/η (cf. Section 2.4.2). More generally, we have previously shown theoretically and experimentally that the accuracy of localization matching algorithms scales with the ratio η/σ (cf. Chapter 5). Hence, the lower the σ/η ratio is, the higher localization accuracy a localization estimator can achieve.

It follows that it would be useful to predict and remove the value of the random noise variable X attached to each measurement, thus decreasing the variance σ^2 and thereby improving the localization accuracy. Without information on surrounding environment, this is not trivial since it is unknown which factors affect a single signal. To achieve this, we start from the realization that RSSI measurements between different node pairs in a network correlate with each other. We therefore extract useful geometrical information from non-local RSSI measurements that when combined with local RSSI measurements can generate a refined predicted set of more accurate local RSSI measurements. To this end, we employ standard regression techniques trained on computer simulated RSSI data (in order to reduce training data collection time and cost) to build a prediction model, and then apply the model to real RSSI data to achieve more accurate RSSI measurements (here called *refined RSSI*). The refined RSSI data is general and can be used with any existing RSSI location estimator to improve its accuracy.

The main contributions of this chapter are to:

- Propose a novel method called *MLRefine* reducing the magnitude of noisy factors attached to RSSI values.
- Substantiate MLRefine through various simulations and experiments.
- Quantify the impact of MLRefine by combining with state-of-the-art localization estimators including MLMatch described in Chapter 5.

6.2 Related Work

Many range-based localization techniques consist of three phases: 1) the *calibration phase* during which the propagation model is calibrated using real RSSI measurements (cf. Figure 6.1a), 2) the *measurement phase* during which nodes broadcast packets that are used at the receiver end to estimate and collect RSSI values between all nearby nodes, and 3) the *localization phase* during which the collected RSSI values are used to estimate each node's location by means of the propagation model and a localization algorithm. The proposed MLRefine method sits between phases 2) and 3). This section discusses related work on noise reduction methods (Section 6.2.1) and gives background on localization algorithms used to quantify the impact of MLRefine (Section 6.2.2).

6.2.1 Noise Reduction Methods

Types of ranging noise can be classified into outlier noise and normal noise [105]. Outlier, which is often caused by malicious attacks, non-line-of-sight (NLoS), hardware malfunction, means abnormal measurements far beyond the normal range. There are numerous attempts that try to detect distance outliers whose distance measurement errors are significantly large and hence are most likely outliers (see [106] for a survey). For instance, Jian et al. [105] use triangle inequality to detect outlier distances that are a factor of ten away from accurate measurements. Xiao et al. [107] propose method to detect outlier distances and outlier nodes that can collude due to malicious attacks. Recently, Xiao et al. [108] use multi-norms regularized matrix completion to realize localization methods that are robust to outlier noise. All of these methods consider that normal distance measurements are highly accurate or contain a small Gaussian error. Therefore, they are inapplicable to RSSI-based localization in which normal RSSI measurements contain large error. Considering RSSI measurements, we have proposed a method that uses a Graph Partitioning to detect and remove NLoS signals that travel through walls in Chapter 5.

Normal noise, which is often caused by multi-path fading, shadowing, etc, is hard to be detected. Smoothing consecutive RSSI measurements between two wireless nodes can reduce the normal noise. For instance, averaging RSSI values from multiple antennas can reduce the effect of shadowing [54]. Time-consecutive RSSI measurements can be filtered to reduce noise using particle filters such as Hampel filter, Kalman filter [109], or using averaging technique (cf. Chapter 4). These methods, however, can only be realizable if there are transmissions between a pair of nodes.

The proposed method, MLRefine, is neither an outlier detection method nor a smoothing method. It uses correlation between RSSI measurements between different pairs of nodes to reduce normal noise. It, however, can also be combined with the above techniques to further reduce the effect of noise. For instance, outlier RSSI measurements are first detected and removed using a Graph Partitioning method as discussed in Chapter 5. A smoothing method such as averaging is then applied to reduce the effect of shadowing. MLRefine is, finally, applied to further reduce normal noise level.

6.2.2 Cooperative Localization Algorithms

Once the data is refined from noisy errors, a localization algorithm must be applied. In contrast to traditional multilateration techniques [49], cooperative localization methods estimate all node positions simultaneously using measurements between almost all nodes rather than localizing each node, thus enhancing accuracy [50]. There are many cooperative localization algorithms such as multidimensional scaling (MDS), semidefinite programming (SDP) [52], stochastic optimization (e.g. simulated annealing (SA)) [46], and matching methods (cf. Chapter 5). Among cooperative localization, SDP localization enhance accuracy

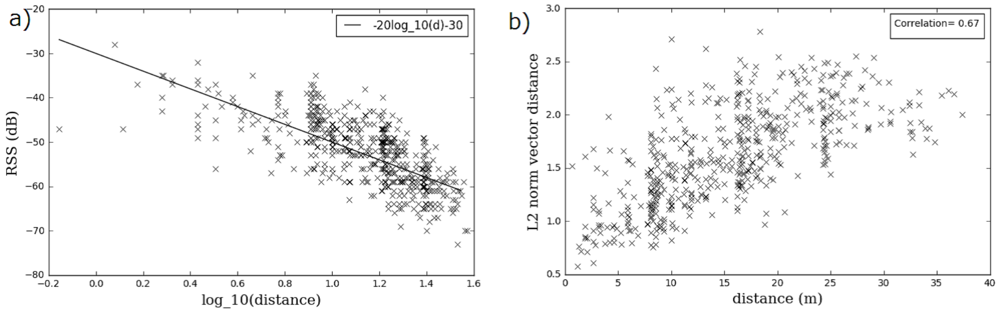


Figure 6.1: a) Linear fitting of the path loss exponent η and reference power P_0 via real measurements. b) Correlation between RSSI vectors similarity (here used Euclidean norm distance) and nodes separation distance.

and can provide deterministic solutions, while localization matching technique enhance accuracy when a set of node positions is provided. To quantify the impact of proposed MLRefine, we will, therefore, employ SDP method designed for resolving RSSI-based localization problems (i.e., Formulas (11) and (12) in Ref. [52]) and the proposed MLMatch that is described in Sections 5.4, 5.4.2. These localization algorithms will be applied to the refined datasets in Sections 6.4 and 6.5.

6.3 The MLRefine Algorithm

6.3.1 Problem Definition

Consider a WSN composed of m nodes that are labeled $1, 2, ..., m$ and equipped with a radio transceiver such that they can exchange messages with each other. RSSI measurements between nodes can be obtained and thus sent to a backhaul server for post-processing. The server then saves all the RSSI values in a square $m \times m$ matrix $\mathcal{R}$ with entries $r_{i,j}$ equal to RSSI value between nodes i, j. If some RSSI values are missing, their corresponding entries are saved as null values. This network model for localization is often used in literatures [50,52]. MLRefine is a method that enable the server to refine these measurements and output a refined RSSI square matrix $\mathcal{R}'$ with refined RSSI entries $r'_{i,j}$ that can provide more accurate ranging and therefore can improve localization estimators. In other words, $r'_{i,j}$ is closer to $\tilde{r}_{i,j} = f(d_{i,j})$ (here called *non-noisy RSSI value*, cf. Formula (6.1)) than $r_{i,j}$.

6.3.2 MLRefine Intuition

MLRefine exploits the inherent geometrical properties of the network deployment space. Namely, since wireless sensor nodes are located in Euclidean space, the pair-separation-distance between two nodes correlates strongly with various distances of other node pairs. Note that the latter can be estimated via RSSI values. In practice, if nodes i and j are close to each other, then the similarity between RSSI values $r_{i,k}$ and $r_{j,k}$ is high for an arbitrary node k. In contrast, if nodes i and j are far from each other, then the similarity between $r_{i,k}$ and $r_{j,k}$ is low for some node k. It follows that inter-node separation distance $d_{i,j}$, and therefore non-noisy RSSI value $\tilde{r}_{i,j}$, are correlated with some similarity metric between RSSI vectors $\boldsymbol{r_i}$ and $\boldsymbol{r_j}$, where $\boldsymbol{r_i} = [r_{i,1}, r_{i,2}, \ldots, r_{i,m}]$. Figure 6.1 b) illustrates the strong correlation between distances $d_{i,j}$ and the similarity of ex-

Table 6.1: Description of features

Feature type	**Feature name**	**Formula**
Vector similarity	Pearson correlation	$\frac{\text{cov}(\boldsymbol{r_i},\boldsymbol{r_j})}{\sigma_{\boldsymbol{r_i}}\sigma_{\boldsymbol{r_j}}}$
	Average Absolute value norm	$\frac{\|\boldsymbol{r_i}-\boldsymbol{r_j}\|_1}{\|\boldsymbol{r_i}\|_0}$
	Average Euclidean norm	$\frac{\|\boldsymbol{r_i}-\boldsymbol{r_j}\|_2}{\|\boldsymbol{r_i}\|_0}$
Other statistic values	Standard deviation of $\boldsymbol{r_i}, \boldsymbol{r_j}, \boldsymbol{r_i} - \boldsymbol{r_j}, \boldsymbol{r_i} + \boldsymbol{r_j}$	
	Average of $\boldsymbol{r_i}, \boldsymbol{r_j}$	
Raw RSSI value	RSSI value between 2 nodes	$r_{i,j}$

where $\|\boldsymbol{x}\|_p = (|x_1|^p + |x_2|^p + ...)^{1/p}$,
$cov(\boldsymbol{x}, \boldsymbol{y})$ is covariance of $\boldsymbol{x}$ and $\boldsymbol{y}$, $\sigma_{\boldsymbol{x}}$ is standard deviation of $\boldsymbol{x}$

perimentally measured RSSI vectors. Here, we used Euclidean norm distance the of two vectors $\boldsymbol{r_i}$ and $\boldsymbol{r_j}$ as the similarity metric. As a result we may conclude that RSSI vector similarities can be used as features to predict non-noisy RSSI values. However, the correlations between RSSI vector similarities and non-noisy RSSI values are not known *a priori* and are difficult to formulate. To avert this problem, MLRefine leverages standard machine learning regression methods that encode said mathematical relationships using computer generated training data.

6.3.3 The MLRefine Algorithm

The proposed localization method is illustrated in Figure 6.2. The whole process is schematically broken down into four blocks: the propagation model calibration block, the training block, the refining block, and finally the localization algorithm. The calibration process will not be discussed since it can be chosen from any existing method such as linear fitting method (cf. Fig. 6.1). We also will not discuss the final block in detail since MLRefine is to a large extent agnostic to the chosen algorithm and should improve localization accuracy and performance regardless by decreasing the ratio σ/η.

Training Phase

The training phase, described by Algorithm 3, is designed to study the relationship between a non-noisy RSSI value and features extracted from other noisy RSSI values. As such, a key benefit of MLRefine is that it can be trained *in silico* (i.e., using computer simulated data). First, m pairs of coordinates are generated at random from some finite subspace $\mathcal{A}$ of $\mathbb{R}^2$ of comparable area to that of the intended WSN deployment. These coordinates represent the m sensor node locations. Inter-node separation distances $d_{i,j}$ are then calculated and converted into non-noisy RSSI values $\tilde{r}_{i,j}$ using the Propagation Model (6.1). Secondly, Gaussian noise is then added to all the non-noisy RSSI values $r_{i,j} = \tilde{r}_{i,j} + X$ and stored in a symmetric $m \times m$ matrix $\mathcal{R}$. In our simulations we have chosen the standard deviation of the added noise from the range of $\sigma \in [3, .., 7]$ dB as suggested from experiments in Chapter 5. Note that we are neither restricted to Gaussian noise nor specific range of σ; for instance in environments where distribution of X can be obtained accurately, it can be substituted for the chosen values of X. In the third step, *features* corresponding to each node pair i, j are then extracted. In machine learning, a feature is any measurable property or characteristic of a phenomenon being observed. Choosing informative and discriminating features is a

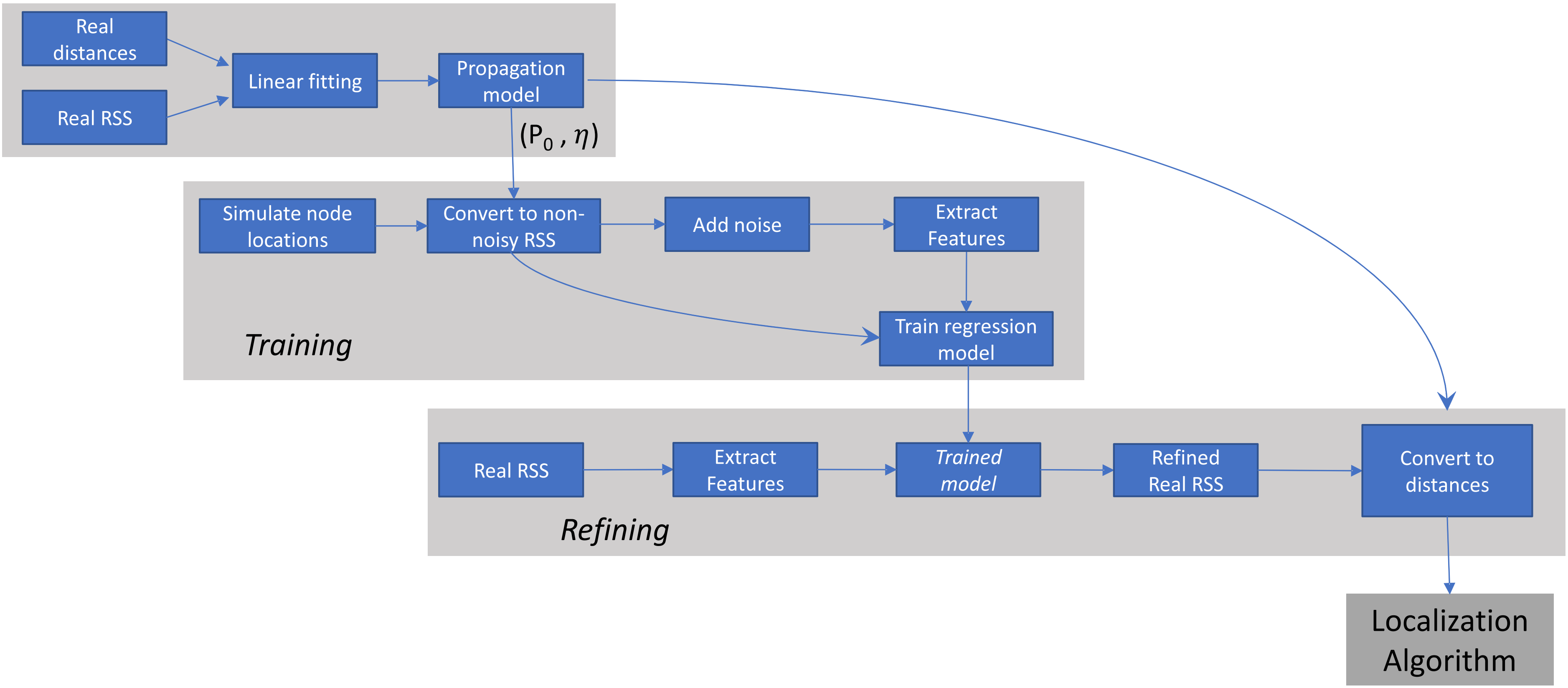


Figure 6.2: Schematic showing how different processes of MLRefine are inter-linked in each of the three blocks leading towards an localization estimator algorithm.

crucial step for effective algorithms in regression. Whilst there are learning techniques that can extract features automatically from raw datasets such as image, voice, etc. (e.g., CNN), efficient learning techniques for treating weighted graph data (i.e. wireless network data) is still unknown. Moreover, efficient learning from artificial data (in this case simulated RSSI data) is still a challenge. We therefore, heuristically extract features described in Table 6.1, in which vector similarities are intuitively correlated with non-noisy RSSI values (cf. Section 6.3.2). Namely, we extract the Pearson correlation, the average absolute and Euclidean norms. Other statistical values such as the standard deviation of vectors $\boldsymbol{r_i}$ and $\boldsymbol{r_j}$ and combinations of them are also extracted. All of these values are not strongly dependent on a single RSSI value but rather represent the whole network thus alleviating negative influence caused by the difference between a single simulated RSSI value and real or experimental RSSI one. Note that this is not a unique set of features since other statistical values could be used just as well.

Once features have been selected and extracted from the noisy RSSI matrix $\mathcal{R}$, we then use regression techniques to model the relationship between the non-noisy RSSI values $\tilde{r}_{i,j}$ and its corresponding features. There are a number of standard regression techniques suitable for this task. We ran linear regression, support vector machine, neural networks with various parameter settings, then compared the results on some sample sets. The results showed that except linear regression, other regression models gave similar results. This is because, the number of features is small. In the remaining of this chapter, we will report results of using a simple neural network model with a single hidden layer consisting of 10 hidden nodes, and activation function of ReLu.

Algorithm 1 Training phase

```
Require: $r = f(d)$ {propagation model}, $m$ {number of nodes}, $\mathcal{A}$ {nodes deployment area}
Ensure: $F$ {trained model}
  Training data generation
  for all $\sigma \in \boldsymbol{\sigma}$ { $\boldsymbol{\sigma}$ is a bounding range of $\sigma$ } do
    Generate $m$ random coordinates in Area $\mathcal{A}$
    for all $i, j \in \{1, \ldots, m\}$ do
      $\tilde{r}_{i,j} \leftarrow f(d_{i,j})$ {non-noisy RSSI value}
      $r_{i,j} \leftarrow \tilde{r}_{i,j} + X$, where $X \sim \mathcal{N}(0, \sigma^2)$ {noisy RSSI value}
    end for
    Feature Extraction
    for all $i, j \in \{1, .., m\}$ do
      $\boldsymbol{features_{i,j}} \leftarrow$ features described in Table 6.1
    end for
  end for
  Training
  Model Function $F$ satisfying $\tilde{r}_{i,j} \approx F(\boldsymbol{features_{i,j}}), \forall i, j$, using a regression method.
```

Refining Phase

The refining phase of MLRefine is described by Algorithm 4 and begins after a backhaul server has collected real RSSI measurements.To predict an accurate RSSI value corresponding to nodes i, j, MLRefine first extracts features from real

Algorithm 2 Refining phase

```
Require: $\mathcal{R}$\{measured RSSI matrix with entries $r_{i,j}$\}, $F$ \{trained model\}
Ensure: $\mathcal{R}'$ \{refined RSSI value matrix with entries $r'_{i,j}$\}
  for all $i, j \in \{1, \ldots, m\}$ do
    $\boldsymbol{features_{i,j}} \leftarrow$ features described in Table 6.1
    $r'_{i,j} \leftarrow F(\boldsymbol{features_{i,j}})$
  end for
```

RSSI values. This process is similar to the feature extraction process that is described in Section 6.3.3. MLRefine then applies the trained model F obtained from the training phase to the extracted features to yield the refined value $r'_{i,j}$. The refined RSSI values can be applied with the proposed MLMatch (cf. Chapter 5), or can be converted to distances using the propagation model (6.1) before the SDP localization algorithm [52] is applied to estimate nodes' locations. Alternatively, the refined RSSI dataset can be used in any off-the-shelf range based localization algorithm to improve multilateration accuracy.

6.4 MLRefine Performance Evaluation Via Simulations

In order to substantiate our proposed method, we have performed three computer simulations. We compare the variance fluctuations of raw RSSI values against the fluctuations of refined RSSI values, and we also contrast the final localization accuracy achieved when SDP localization method is applied (see Section 6.2.2).

6.4.1 Parameter Settings

Simulation 1: To test our proposed method against 1) different numbers of nodes $m \in \{10, ..., 100\}$ are deployed randomly in a 100×100m square domain, and 2) different noise distributions X. In each random realization and for each pair of nodes, we generate two separate pairs of RSSI values using Formula (6.1); one with a Gaussian distributed $X \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 5.57$ and one with a Rayleigh distributed X whose probability density function is given by

$$f_X(x) = \lambda 10^{x/10} \exp\big(-\lambda 10^{x/10}\big)\frac{\ln 10}{10}, \tag{6.2}$$

where $\lambda = 0.561$, in which case the standard deviation of X is $\sigma \approx 5.57$ (cf. Chapter 5). *Simulation 2*: To test MLRefine against different types of node distributions and different values of Gaussian noise variance we generate 49 nodes in a 100×100m square domain under two types of deployment layouts: 1) a random layout, and 2) a 7×7 grid layout with one grid length equals to 14m. In each random realization and for each pair of nodes, random RSSI values were generated using (6.1) with $X \sim \mathcal{N}(0, \sigma^2)$ and for different values of σ chosen uniformly from $\{3, 3.5, 4, 4.5, \ldots 7\}$. As discussed in Section 6.1, reducing the variation of signal fluctuation equivalent to reducing the lower bound of the variance of localization error. However, to quantify the impact of MLRefine, we feed SDP localization method, which is known as enhancing accuracy [52], with refined and unrefined (i.e. raw) RSSI data. *Simulation 3*: To test our proposed method's performance in anisotropic domain shape deployments against 1) different noise distributions: Gaussian and Rayleigh distributions given by Formula (7.10), and 2) different numbers of anchor nodes: from 20% to 50% of all the nodes, we generate 50 nodes randomly deployed in a C-shape domain (Figure 6.5 c). In this instance,

Table 6.2: Description of parameters in Simulations

	Simulation 1	Simulation 2	Simulation 3
# of Nodes	10 - 100	49	50
Area(m)	100×100	100×100	C-shape
Layout	Random	Random&Grid	Random
X distribution	Gaussian&Rayleigh	Gaussian	Gaussian&Rayleigh
Std of X	5.57	3.0 - 7.0	5.57
# of anchors	-	4	10 - 25

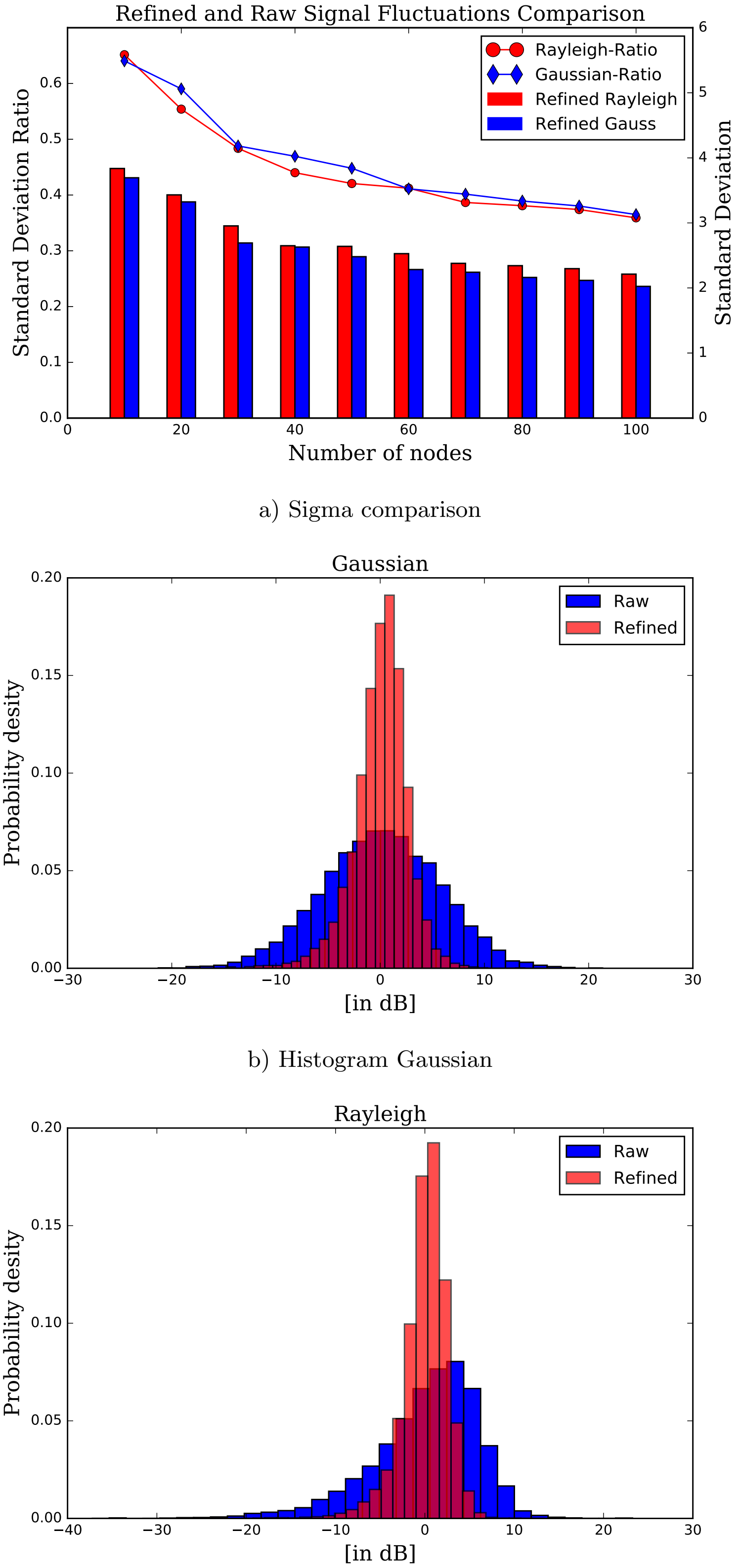
we use the SDP method with $a \in \{10, 12, 15, \ldots, 25\}$ anchors located randomly in the C-shaped WSN deployment region. All RSSI values in the above three simulations were generated using common parameters $P_0 = -30$ dB and $\eta = 2$. The values of all other parameters used are summarized in Tab. 6.2 For each set of parameters, simulations were performed 10 times to obtain statistical averages.

6.4.2 Results and Analysis

In every simulation, we first run the proposed MLRefine algorithm and measure the accuracy of the output refined values $r'_{i,j}$ compared to non-noisy RSSI values. Namely, we measure the standard deviation σ' of variable X' , where $X' = r'_{i,j} - P_0 + 10\eta \log_{10} d_{i,j}$. Note that we calculate standard deviation of X' under the assumption that the mean of X' is zero, which is equivalent to the mean squared error of refined RSSI values. We then compare σ' with σ, which is the standard deviation of input random variable X of raw (unrefined) RSSI values. As discussed in Section 6.1, we expect that the smaller of standard deviation of the RSSI values used for localization, the better the accuracy that a localization estimator can achieve. This is why it is meaningful here to compare σ' with σ. In addition to the fluctuations, we then also apply the SDP localization method and compare the corresponding localization errors using raw RSSI values and refined RSSI values.

The results are plotted in Figures 6.3,6.4,6.5 indicating that in any case, the value of σ' is much smaller than σ. The results of Simulation 1 (Figure 6.3 a) indicates that the standard deviation of refined RSSI values σ' decreases while the standard deviation of raw (unrefined) RSSI values σ is unchanged (equals to 5.57, cf. Section 6.4.1), hence the ratio $\sigma'/\sigma < 1$. Also, a clear positive trend is seen as the number of sensor nodes m increases. Figure 6.3 a) also validates the robustness of MLRefine against other random variable distribution models for wireless fading such as the Rayleigh distribution even though the training data is generated using a Gaussian distribution. Figures 6.3 b, c) illustrate that the refined distribution of variable X' is similar to the distribution of X, namely X' also follows Gaussian (or Rayleigh) when X follows Gaussian (or Rayleigh), but with smaller variation.

The results of Simulation 2 (Figure 6.4) illustrate the localization error of SDP method using raw vs. refined RSSI values when nodes are distributed randomly or in a grid layout. Observe that the localization error is much smaller for refined RSSI data compared to the raw (unrefined) RSSI values. Moreover, localization error significantly decreases when the standard deviation of raw RSSI values σ decreases, which suggest robustness of MLRefine especially in noisy networks. The figures also validate the robustness of MLRefine against other node distribution patterns such as a grid structure, even though the nodes are distributed under

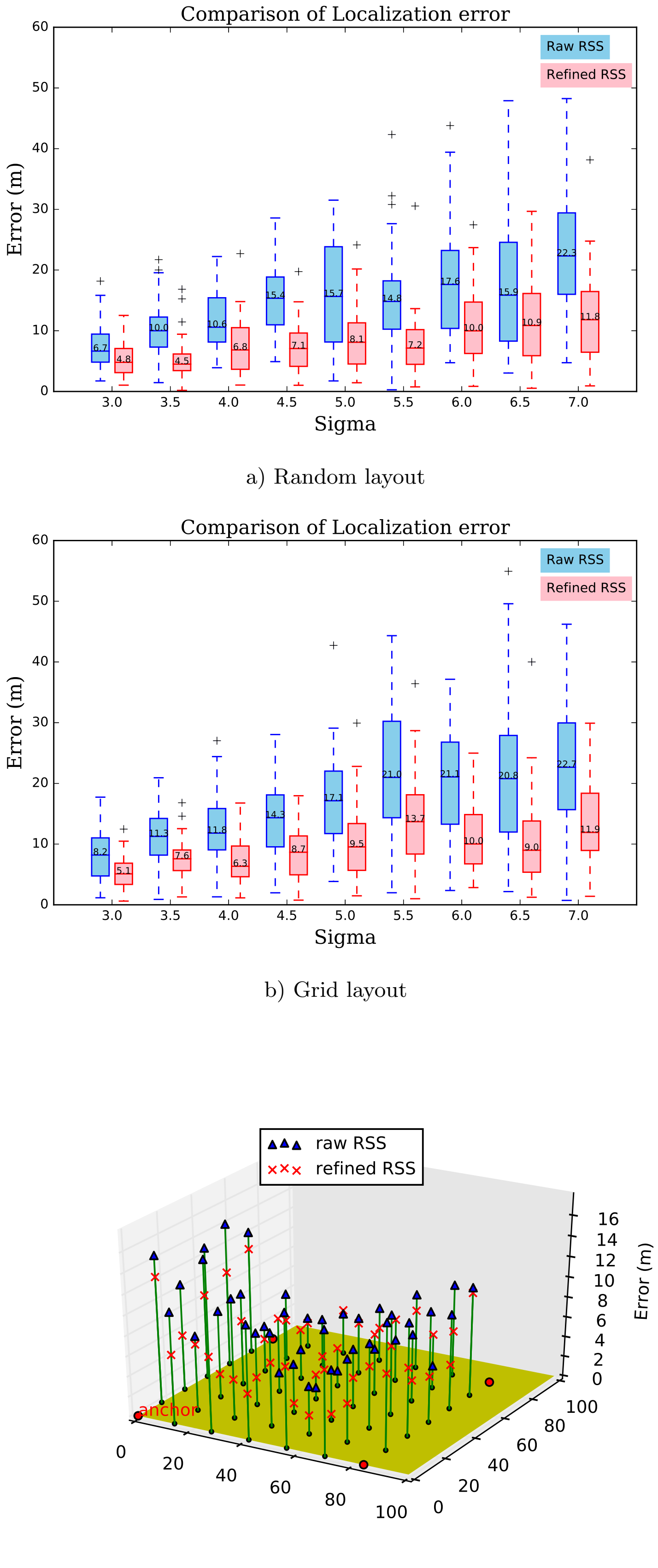


a) Sigma comparison

b) Histogram Gaussian

c) Histogram Rayleigh

Figure 6.3: Simulation 1 results: a) Comparison between fluctuations of raw (unrefined) RSSI values (i.e. σ) and fluctuation of refined values (i.e. σ'). Lines express ratio σ'/σ, bars express values of σ'. b, c) Probability density comparison between fluctuations of raw RSSI values (blue) and refined values (red) under Gaussian distribution (b) and Rayleigh distribution (c).

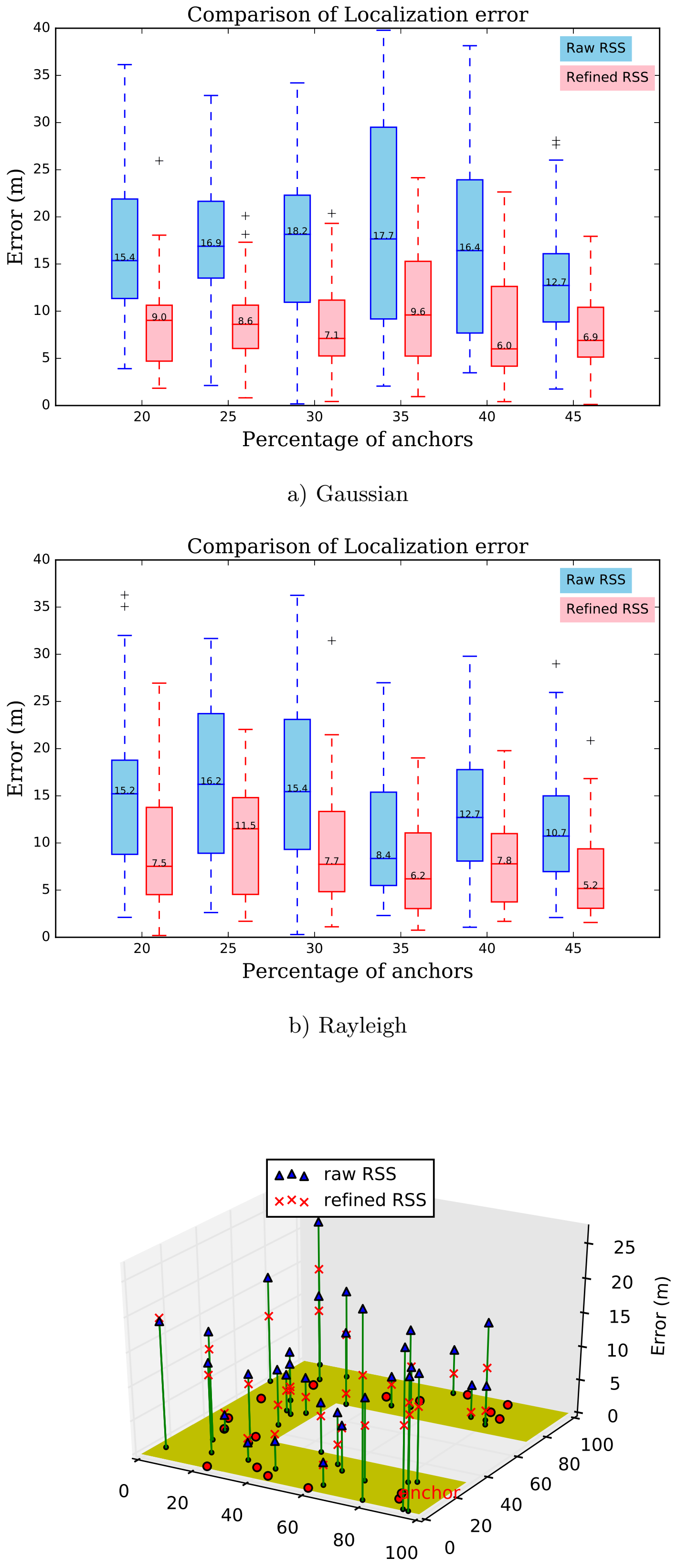


a) Random layout

b) Grid layout

c) Error distribution ($\sigma = 3$)

Figure 6.4: Simulation 2 results: a, b) Comparison of localization error using SDP method between raw RSSI values (in blue) and refined RSSI value (in red). c) Illustration of localization error in grid distribution with $\sigma = 3$.



a) Gaussian

b) Rayleigh

c) Error distribution (35% anchors)

Figure 6.5: Simulation 3 results: a, b) Comparison of localization error using SDP method between raw RSSI values (in blue) and refined RSSI value (in red). c) Illustration of localization error in grid distribution with C-shape domain with 17 anchors.

a random uniform spatial distribution during the training phase. The results of Simulation 3 (Figures 6.5) illustrate the localization error of SDP method using raw and refined RSSI values when nodes are distributed in a C-shape domain. Observe that MLRefine improves localization error consistently in these settings too and is impervious to using different fading models for noise and fading effects X despite our training phase using Gaussian noise, with uniform node deployments in a square domain. Further improvements could be anticipated if the training data utilized anisotropic C-shaped deployment regions. In conclusion, the results illustrate that MLRefine is robust in various environments despite the difference of distribution models between training data and raw RSSI values.

6.5 Experimental Investigation

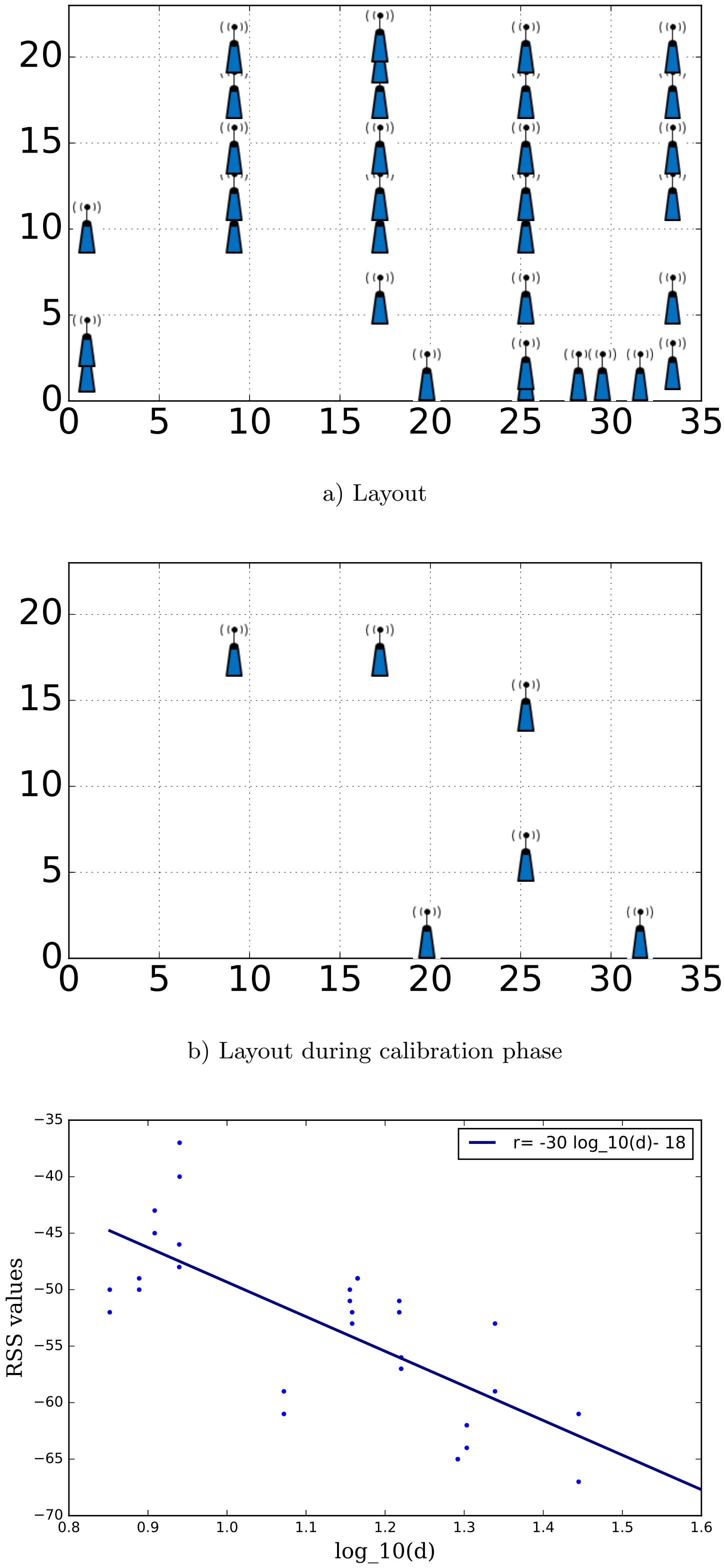
This section investigates and validates the performance benefits of MLRefine through real experiments. We compare the variance fluctuation of the experimentally measured raw RSSI values against the fluctuation of the refined RSSI values, and we also contrast the localization accuracy achieved when combined with MLMatch algorithm (see Section 6.2.2). The accuracy of the algorithm is defined as the ratio of number of correctly matched nodes to the number of total nodes, which is different from other localization methods.

6.5.1 Description of Experimental Setup

We have performed experiments in Toshiba premises in Japan, utilizing 33 Toshiba wireless devices operating at 920MHz band. The devices were placed at specific positions described in Figure 6.6 a). For further details of the experimental setup, prototype, and surrounding environment, readers are referred to Experiment 5 in Chapter 5 from which much of the data was obtained. *Calibration:* In order to determine the propagation model, we put 6 wireless devices located at positions described in Figure 6.6 b). The devices communicate with each other, thus RSSI value between each pair of nodes can be collected and sent to a server. The server uses a linear fit to calculate parameters P_0 and η (Figure 6.7 c).

6.5.2 Results and Analysis

Using data from a total of 33 wireless nodes we randomly select a subset and run MLRefine and compare refined RSSI values with the raw (unrefined) ones. Training was performed through computer simulations that have the same parameters as the experimental setup. Similar to our simulation results and analysis, we measure the expected accuracy of refined values $r'_{i,j}$ through the standard deviation of their fluctuations σ'. In addition, we run a localization matching algorithm (MLMatch) on both RSSI types: raw RSSI values and refined RSSI values. The results of this experiment are plotted in Figure 6.7 a) indicating that the value of σ' is much smaller than σ. Figure 6.7 a) depicts the distribution of variables X and X' illustrating that the distribution of unrefined (raw) signal fluctuations follows neither Gaussian nor Rayleigh distributions. This is due to the effect of the surrounding realistic indoor environment such as reflections and scattering. By truncating the original database we can study the experimental effects of MLRefine on smaller networks. We find that MLRefine can enhance the localization accuracy as seen in Figure 6.7 b) which illustrates the performance gains of the localization algorithm MLMatch running on refined RSSI datasets. These gains appear to onset when the number of nodes m is bigger than 17. For instance, the

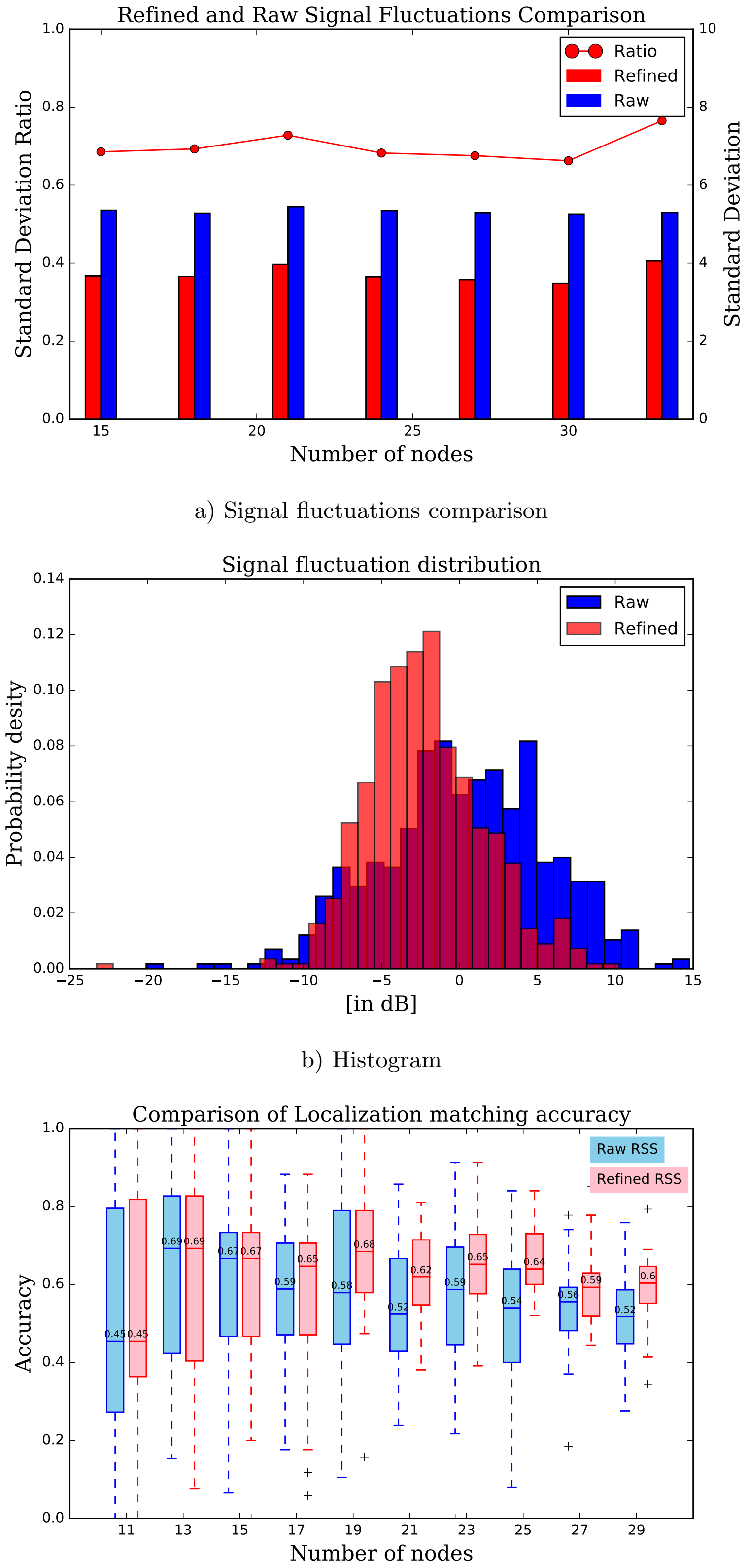


a) Layout

b) Layout during calibration phase

c) Linear fitting

Figure 6.6: Experimental setup: a) Layout of 33 wireless devices. b) Layout of 6 wireless devices during a priori measurements, i.e. calibration phase. c) Linear fitting using data from a priori measurements.



a) Signal fluctuations comparison

b) Histogram

c) MLMatch accuracy

Figure 6.7: Experimental results: a) Comparison between fluctuations of raw RSSI values (i.e. σ) and fluctuation of refined values (i.e. σ'). Lines express ratio σ'/σ, bars express values of σ(blue) and σ'(red). b) Probability density comparison between fluctuations of raw RSSI values (blue) and refine values (red). c) Localization matching accuracy comparison using raw RSSI values and refined RSSI values for different number of nodes.

difference of median is up to 10% for $m = 25$, up from 54% to 64%, respectively. This trend is similar to the results in Simulation 1 (cf. Figure 6.3 a), where the efficiency of MLRefine increased with the number of wireless devices being used.

6.6 Conclusions

In this chapter, we have proposed a method called *MLRefine* to refine raw RSSI data collected from wireless sensor networks in order to improve wireless ranging and therefore localization accuracy. MLRefine uses machine learning algorithms to extract and exploit the inherent spatial network geometrical correlations that are hidden in noisy RSSI datasets used for RF wireless localization. These correlations are captured by features that are then used to reduce the magnitude of RSSI fluctuations, in turn improving the localization accuracy. We note that training data can be generated off-line, thus minimizing the cost of collecting training data. We have validated the efficiency of MLRefine through extensive computer simulations but also real-world experiments. Especially, it can improve the localization accuracy of MLMatch described in the previous chapter up to 10%, thus improving the practicality of MLMatch.

Chapter 7

LEMOn: Wireless Localization for IoT Employing a location-unaware Mobile Unit

Chapters 5 and 6 propose localization techniques and an improvement technique to resolve the proposed WLMP under a scenario that sensor nodes connect with each other. In practice, however, there are network systems that sensor nodes connect to a central unit, e.g. a wireless controller or a wireless concentrator, etc., rather than with each other. Considering that the central unit equipped with mobile robot, in this chapter, we address the problem of localizing sensor nodes using a mobile wireless unit.

Existing localization methods that use a mobile wireless unit assume an accurate knowledge of the location of the mobile unit and a precise propagation model of the actual radio environment. By getting rid of these two requirements, our proposed localization algorithms make mobility-assisted localization far more practical as we do not need to equip the mobile unit with a global positioning system or run a time-consuming campaign to survey radio environment. *LEMOn* estimates the position of target nodes by using known locations of a small set of fixed anchor nodes while receiving messages sent from a mobile unit from unknown arbitrary locations. *LEMOn-M*, on the other hand, solves the localization matching problem by mapping an arbitrary number of target nodes to the known set of locations. Both algorithms first estimate an inter-node distance using a similarity between Received Signal Strength Indicators of beacons received from the mobile unit. Conventional location estimators are then employed to localize target nodes with an unknown location. Obvious examples of real-world applications include but are not limited to unmanned aerial vehicle assisted wireless sensor networks and indoor IoT systems. Various simulations show that the two algorithms achieve a very high localization accuracy even in harsh radio environments while static localization techniques fail.

The main contributions of this chapter are as follows:

- We define and motivate new localization problems that are hybrid between static localization and mobility-assisted localization.
- We propose and analytically study the performance of LEMOn and LEMOn-M as candidate solutions to the above problems,
- We highlight some potential applications related to real-world scenarios.

The remainder of this chapter is organized as follows. In Section 7.1, we briefly introduce two major RF-based localization techniques for IoT: localization techniques using mesh network paradigms and mobility-assisted localization techniques. In Section 7.2, we describe the system model and problem definitions.

In Section 7.3, we highlight conventional work on mobility-assisted localization and localization estimation algorithms with a focus on algorithms that are used in this chapter. In Section 7.4, we present the details of our proposed LEMOn and LEMOn-M techniques. In Sections 7.5 and 7.6, we evaluate the performance of the above algorithms through an extensive simulation-based study. In Section 7.7, we highlight potential real-world applications. Finally, we conclude this chapter with a summary of our results and discuss our future plans in Section 7.8.

7.1 Introduction

With improvements in wireless communication technology over the last decades, the Internet of Things (IoT) is now widely deployed in smart cities, buildings, and houses [1]. Typically, an IoT system consists of low cost and low power devices which interact with each other through the Internet. The main goal of IoT is to ensure every device, including sensors, smart-phones, wearable sensors, tablets, transportation system, etc., can connect with each other through a common interface. This allows machine-to-machine (M2M) communication without human intervention [2], thus can reduce manual cost. There are numerous key issues in IoT including wireless localization. Wireless localization, which refers to extracting geo-location information of an object, has therefore been well researched and developed [110]. Although a Global Positioning System (GPS) module can provide location information of a wireless device, this incurs additional cost and huge power consumption. GPS is thus not always suitable for many IoT applications. To alleviate such problems, several radio-frequency (RF) based localization methods have been developed to estimate the devices' location. These methods measure either devices' proximity (i.e. connectivity) [8], Angle of Arrival (AoA) [111], Time of Arrival (ToA) [112], Time Difference of Arrival (TDoA) [113], or Received Signal Strength Indicator (RSSI) [10, 11] to localize target devices deployed at unknown locations. In contrast to AoA measurements and time-based measurements that require additional hardware and techniques, RSSI can be obtained from almost all wireless hardware. Therefore, RSSI based localization is an cost-effective solution for most IoT applications. We, therefore, consider localization problems using RSSI. Typically, an RSSI value r in dBm is estimated through the log-distance propagation model [35].

$$r = P - 10\eta \log_{10} d + X \tag{7.1}$$

where d is the distance between the transmitter and the receiver, η is the path loss exponent, P is a reference power value measured in dBm at a distance of one meter from the transmitter, and X is a random variable characterizing the ranging error caused by multi-path fading and shadowing.

Depending on network paradigms, RF-based localization can be broadly divided into two categories: inter-node communication based localization schemes and mobility-assisted localization schemes. The former type of schemes (referred as *static localization* schemes in this chapter), first estimate the Euclidean distances between communicating nodes using inter-node RF measurements, then localize target devices using either multi-lateration methods [49] or cooperative localization methods [11,50,52,114]. These schemes are deemed suitable for mesh network paradigms. On the other hand, mobility-assisted localization schemes use a wireless mobile unit, which is aware of its location (e.g., equipped with GPS), to assist in estimating the location of target wireless nodes [47,48]. A mobile unit can use an Unmanned Aerial Vehicle (UAV), a drone, a ground vehicle, etc.

A node could be any device that equipped with a wireless hardware, position of which is often assumed to be fixed or changed only infrequently. We call these devices as *sensor nodes* to distinguish from the mobile unit. Unlike the power-constrained sensors, the energy of the mobile unit is typically assumed to be unconstrained. Therefore, these mobility-assisted localization methods are more advanced and practical than static localization methods as reliable inter-node distance is hard to obtain for the latter schemes [48]. For instance, while static localization techniques suffer from the problem of non-line-of-sight (NLoS), e.g. due to obstruction, line-of-sight (LoS) communication links can be established between the UAV and sensor nodes over time when UAV moves. Besides, in sparse node deployments, a nodes may not be able to inter-connect due to their energy constraints. A more powerful UAV, however, can often connect to every sensor node and can thus help localize these nodes. With the rapid development of UAV-assisted WSNs [115], mobility-assisted localization systems are attracting greater attention [116].

However, all of the above localization techniques suffer from the following limitations.

- *Difficulty in Estimating Accurate path loss exponent*: The value of path loss exponent η depends on the measured environment. For instance, it is approximately 2 for free space, and varies from 3 to 5 in shadowed urban cellular radio [36]. Estimating accurate values of these parameters is often prohibitively expensive in practice, and is even impossible in human-inaccessible environments, for instance, WSNs for post-disaster monitoring.

- *Difficulty in the Calibration Process*: An RSSI value is a function of the calibration of both the transmitter and receiver. Transmitted powers and received powers vary from device to device due to use of different hardware components. Wireless nodes might be designed to measure and report their own calibration data to each other. This, however, complicates the design [50].

- *Large Ranging Error*: Even if calibration process can be reliably done and path loss exponent is estimated accurately, there is a ranging error caused by multi-path fading and shadowing. These effects can be reduced only if sensor nodes are equipped with specific hardware or technologies. For instance, multiple antennas can mitigate the effect of shadowing [54] and a spread-spectrum method can reduce the effect of frequency-selective fading [50]. However, an ordinary wireless node is not always equipped with such hardware. A calculation in Section 2.4 illustrates that ranging error can be up to six times larger than the distance itself.

- *In Mobility-assisted Localization: Uncertainty of the Mobile Unit's Position*: Certain application deployment areas or environmental conditions can prevent from accurately determining position of the mobile unit [117]. These factors may include unavailability of the GPS in indoor environments or under poor weather conditions. Even if GPS is available, determining accurate position of the mobile unit is challenging in real-time, especially for a high-speed mobile unit. A standard non-differential GPS receiver has an error of 5 m to 10 m. In addition, delay caused by a low update rate, which varies from 1 to 10 Hz on average, and GPS response further increases the

Table 7.1: Comparison of localization techniques

	Static localization	**Mobility-assisted localization**	**LEMOn, LEMOn-M**
Method	Measure distance between sensor nodes using inter-node RF measurements	Measure distance between the mobile unit and a sensor node using their RF measurements	Measure distance between sensor nodes using RSSI measurements to the mobile unit
Advantages	Do not require the mobile unit	Do not require to fix anchor devices. Localization accuracy can be improved by increasing the number of beacons	Do not require calibration, propagation model parameter, and location of the mobile unit
Requirements	Require calibration, accurate propagation model and positions of anchor nodes	Require calibration, accurate propagation model, and accurate location of the mobile unit	Require a set of anchor nodes or a set of node positions
Applications	Localization in static WSNs	Localization in UAV assisted WSNs in environments that accurate location of the UAV can be obtained accurately	Localization using a mobile unit whose location is not obtained accurately

positioning error of a high-speed UAV. Moreover, the GPS position accuracy is known to deteriorate when fewer satellites are reachable [118]. To the best of our knowledge, these issues have not been considered and solved in mobility-assisted techniques.

To eliminate the above drawbacks of existing mobility-assisted localization methods and static localization methods, we propose two localization techniques that can combine the strength of both schemes. In our network model, we assume that a mobile unit broadcasts beacon messages (referred to as *beacons* for short) periodically to sensor nodes. Unlike conventional mobility-assisted localization techniques, we assume that neither the position of the mobile unit nor the signal propagation model is known. However, we assume that the following information is known: *1)* Position of some of the fixed sensor nodes also called *anchor nodes*, or *2)* A set of node positions, however, we do not know which node locates at which position. The former localization problem is often found in literature [50, 52]. On the other hand, the latter localization problem is studied in Chapter 5 and is called as the wireless *localization matching* problem. Using the RSSI values estimated from the beacon transmissions we propose two methods called *LEMOn*, i.e. Localization Employing a location-unaware MObile unit and *LEMOn-M*, i.e. LEMOn for localization matching to respectively resolve the above two problems. The main advantages of the proposed LEMOn and LEMOn-M are:

- *Suitable for both indoor and outdoor environments* as the location of the mobile unit is not required,
- *Easily applicable* as these methods do not require system calibration as well as a priori measurements to estimate the path loss exponent,
- *Robust to noise* as these methods use statistical features of RSSI measurements rather than a single or few RSSI measurement that indeed fluctuates a lot.

The characteristics of the above localization techniques are summarized in Table 7.1.

7.2 System Model and Problem Definitions

Consider a wireless network consisting of N *sensor nodes*, $n_1, n_2, ..., n_N$ that are deployed in a domain of interest $\mathcal{D}$. A *mobile unit* moving along an arbitrary trajectory inside a domain of interest $\mathcal{V}$ broadcasts beacons periodically to all sensor nodes. Each beacon includes its unique sequence number $k, k \in \{1, 2, 3, ..., K\}$ (K is the *number of beacons*). We assume that signals transmitted by the mobile unit are strong enough to reach all wireless sensor nodes. Each sensor node n_i receives the beacon transmissions, estimates their RSSI values, then constructs a vector $\boldsymbol{r_i} \in \mathbb{R}^K$ with entries $r_{i,k}$ equal to the RSSI value retrieved from the k-th beacon. Node n_i then sends its corresponding vector $\boldsymbol{r_i}$ back to a server (potentially via the mobile unit) for post-processing. The server is then tasked with estimating the locations of the target nodes. Without loss of generality, $\mathcal{D}$ and $\mathcal{V}$ are assumed as two-dimensional spaces, i.e. $\mathcal{D}, \mathcal{V} \subset \mathbb{R}^2$. We assume that either a wireless node or the mobile unit is equipped with an isotropic antenna. RSSI $r_{i,k}$ is, therefore, related with distance $d_{i,k}$ between a wireless nodes n_i and the mobile unit when it sends the k-th beacon through the log-distance propagation model [35].

$$r_{i,k} = P_i - 10\eta \log_{10} d_{i,k} + X_{i,k} \tag{7.2}$$

where η is the path loss exponent, which is approximated to 2 for free space, and a value between 3 and 5 for urban environments. Since wireless nodes can be deployed in any environment, we assume that η is an unknown constant but is identical for all nodes. P_i is a reference power corresponding to node n_i value at a distance of one meter from the transmitter, and X is a random variable characterizing the noise factors. The reference power P_i corresponding to node n_i is an unknown constant, as sensor nodes are assumed not to be calibrated. The distribution of X depends on the wireless propagation environment. For example, the long-term signal variation is known to follow the Log-normal distribution, whereas the short-term signal variation can be described by several other distributions, such as the Rayleigh distribution. For simplicity, in our analysis, X is assumed to follow the Log-normal distribution, namely Gaussian distribution in dB, i.e. $X \sim \mathcal{N}(0, \sigma_X^2)$, where the standard deviation σ_X can be as low as three (cf. Chapter 5) and as high as 12 [50].

We define two different problems, called the *localization* problem and the *localization matching* problem, which are motivated through real-world applications described in Section 7.7.

7.2.1 The Localization Problem

The localization problem can be defined in a similar way as in [50]. Given a set $\mathcal{A}$ comprising of A ($A < N$) *anchor nodes* with known-positions (acquired through GPS or or as a result of deployment process), the problem is to estimate the location of other sensor nodes, called *target nodes.*

7.2.2 The Localization Matching Problem

The wireless localization matching problem, is defined as follows. Given a set of N positions $p_1, p_2, ..., p_N$ where N sensor nodes are located, the problem is to correctly match each node label n_i with its correct position $p_{i'}$. In the beginning, it is unknown which position $p_{i'}$ node n_i is located at.

7.3 Background and Related Work

Since our proposed algorithms adopt a hybrid approach based on mobility-assisted and static localization techniques, this section highlights existing work on mobility-assisted localization as well as localization estimation algorithms for static networks. Section 7.3.1 describes a brief history of mobility-assisted localization and differentiates these localization techniques from our proposals. Section 7.3.2 provides a background on localization estimation algorithms for static networks, mainly focusing on the representative algorithms used in this chapter.

7.3.1 Mobility-Assisted Localization

Mobility-assisted localization schemes locate target nodes using mobile units that are aware of their own location and are capable of moving around the target nodes arbitrarily. The mobile units periodically broadcast beacons with their location, enabling the nearby target nodes to hear this information and estimate their own location [116].

RSSI-based mobility-assisted localization techniques are pioneered by Sichitiu and Ramadurai [47]. They use a wireless device carrying truck that broadcasts messages containing its known location to sensor nodes deployed in an outdoor environment. The nodes estimate RSSI values of the received messages to determine their distance to the truck. The nodes finally calculate their locations using a probabilistic method. Menegatti et al. [119] estimate the location of a robot while mapping the nodes simultaneously using RSSI measurements and odometry from the robot. They use a log-distance propagation model, which is calibrated before the experiments to calculate the distance between the robot and sensor nodes. Caballero et al. [120] use a robot equipped with Differential GPS (DGPS) moving in an outdoor parking lot to measure RSSI values from neighboring nodes. Similar to the above approaches, the propagation model is calibrated before the experiments.

Besides, mobility can be combined with time-based ranging techniques to localize the sensor nodes. For instance, Sun and Guo [121] use a mobile beacon traversing deployed area of sensor network and broadcasting location-containing packets. On receiving the beacon packets, nodes combine the received locations with the time of arrival of the packets to calculate its own location. Localization is performed using either non-parametric or parametric probabilistic estimation techniques.

In contrast to the ranging techniques described above, some range-free approaches [122–124] use connectivity information for locating nodes, thus no extra hardware or data communication is needed for the sensor nodes.

Recently, with the rapid research and development on UAV, there are numerous works on using a UAV to localize unknown devices [125, 126]. For instance, Villas et al. [126] use a UAV equipped with GPS broadcasting its geo-location when flying over the monitoring area. Using these 3D geo-locations and the corresponding RSSI values, sensor nodes can calculate its 3D location. Yang et al. [125] use a UAV carrying GPS and a camera to collect sensor node images. Locations of non-occluded nodes are then determined using image processing techniques. These nodes are then used as anchor nodes to localize occluded nodes using RSSI ranging localization techniques.

It is, however, worth noting that all of the above techniques require an accurate distance estimation method (i.e. calibration of the propagation model for ranging techniques, or connectivity model for range-free techniques), and an accurate mobile unit position, which can be hard to obtain in practice [117].

Recently, an RSSI-based localization method [15] is proposed that relaxes the requirement for the UAV to know its location. This work, however, can localize sensor nodes that are equipped with the exactly same hardware for the wireless devices. It is, therefore, less practical than the proposals in this chapter. To overcome such limitations, the proposed methods in this chapter, to the best of our knowledge, are the first ones that require neither calibration nor location of the mobile unit.

7.3.2 Location Estimation Algorithms

Location estimation algorithms (see [46] for a detailed survey) can be broadly divided into two categories: non-cooperative methods and cooperative methods. In non-cooperative localization, distance measurements are made only between anchor nodes and target nodes. Each target node estimates its distance to the anchor nodes using measured RSSI values, then uses localization algorithm such as multilateration [49] to locate itself. These methods are suitable for either target tracking problem, or mobility-assisted localization.

On the other hand, cooperative localization methods, where distance measurements between target nodes are also made, estimate all node positions simultaneously rather than localizing each target node individually. These methods enhance localization accuracy of non-cooperative techniques by using more measurements. These methods are, therefore, ideal for wireless mesh networks where nodes can communicate with each other [50]. There are numerous localization algorithms such as Multi-Dimensional Scaling (MDS) [51], Semi-Definite Programming (SDP) [52], stochastic optimization (e.g. simulated annealing (SA) [53]), and localization matching (cf. Chapter 5).

All above methods can determine nodes' location given estimated distances between them. We focus on proposing new distance estimation techniques rather than new location estimators. Therefore, we use the existing location estimators. We choose SDP localization method for resolving the localization problem because it can output deterministic solutions and can also enhance accuracy compared to other methods [52]. On the other hand, a localization matching method called MLMatch (cf. Chapter 5) is the unique location estimator that can solve the localization matching problem.

The remainder of this section depicts a high-level description of SDP localization and MLMatch, while more details can be found in [52] and Chapter 5.

SDP Localization Method

The problem can be stated as follows: given a set of anchors $\mathcal{A}$ with known locations $\boldsymbol{a}_j, (j \in \mathcal{A})$, a set of target nodes $\mathcal{T}$ whose locations are unknown, and some estimated distances $\hat{d}_{i,j}(i, j \in \mathcal{A} \cup \mathcal{T})$, find $\boldsymbol{x}_i, (i \in \mathcal{T})$ the location of target nodes, such that:

$$\begin{aligned} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| &= \hat{d}_{i,j}, i, j \in \mathcal{T} \\ \|\boldsymbol{x}_i - \boldsymbol{a}_j\| &= \hat{d}_{i,j}, i \in \mathcal{T}, j \in \mathcal{A} \end{aligned} \tag{7.3}$$

The problem in (7.3) can be reformulated as follows.

$$\min_{\boldsymbol{x}_i} \quad \sum_{i=1}^{N} \Big(\sum_{j \in \mathcal{A}} (\hat{d}_{i,j} - \|\boldsymbol{x}_i - \boldsymbol{a}_j\|)^2 + \sum_{j \in \mathcal{T}} (\hat{d}_{i,j} - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|)^2 \Big) \tag{7.4}$$

To solve the non-convex optimization problem in (7.4), Biswas et al. [52] use relaxation techniques and reformulate it as an SDP problem that can be solved in polynomial time. Note that the solutions may not be global optima.

MLMatch Method

MLMatch is formulated using a maximum likelihood technique and a statistical model between RSSI values and distances. Each possible matching between nodes and positions is formulated through a permutation h, where $h(i) = i'$ if node n_i is guessed to be located at position $p_{i'}$. Given an RSSI matrix $\mathcal{R}$ consisting of RSSI values between sensor nodes, MLMatch tries to find the best matching h^* whose likelihood is biggest. Namely, nodes n_i are most likely to be located at positions $h^*(i)$. Using the correlation between RSSI values and distances, MLMatch reduces the problem into a simple mathematical formulation as follows:

$$h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \sum_{i<j} \Big(r_{i,j} \ln d_{h(i),h(j)} \Big), \tag{7.5}$$

where $\mathcal{H}$ is the set consisting of all permutation h. It then applies an appropriate searching method, for instance, meta-heuristic or LP relaxation, to find the best matching h^*. Similar to SDP localization, MLMatch may not output the global optima.

7.4 Proposed Algorithms LEMOn and LEMOn-M

We illustrate algorithms for LEMOn and LEMOn-M. We first derive the relationship between an inter-node separation distance and the similarity between RSSI values in Section 7.4.1. We then illustrate localization methods using this similarity in Sections 7.4.2 and 7.4.3. Finally, we analyze factors that affect the localization accuracy in Section 7.4.4.

7.4.1 Distance Estimation

Inter-node distance estimation is fundamental in any localization method. The proposed algorithms make use of the correlation between inter-node separation distance and the similarity between RSSI values. Intuitively, if nodes n_i and n_j are close to each other, then the distance from the mobile unit to n_i is close to the distance from the mobile unit to n_j. In contrast, if nodes n_i and n_j are far from each other, their distances to the mobile unit are different from each other. Since an RSSI value $r_{i,k}$ correlates with the distance between the mobile unit and a sensor node n_i, it follows that inter-node distance $d_{i,j}$ between nodes n_i and n_j correlates with some similarity metric between RSSI vectors $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$, where $\boldsymbol{r}_i = [r_{i,1}, r_{i,2}, ..., r_{i,K}]$. Using standard deviation of the elements of vector $(\boldsymbol{r_i} - \boldsymbol{r_j})$ denoted as $s_{i,j}$, as a similarity metric, we prove that it is a monotonically increasing function of distance $d_{i,j}$ under some assumptions.

Theorem 1 *Standard deviation of the elements of vector $(\boldsymbol{r_i} - \boldsymbol{r_j})$ is approximately a monotonically increasing polynomial function of distance $d_{i,j}$ when K and $\mathcal{V}$ are large enough.*

Proof 1 *The mathematical proof of the above theorem is provided in the Appendix.*

Algorithm 3 LEMOn Algorithm

```
Require: $\boldsymbol{r_i}, \forall i \in \{1, ..., N\}$ {$\boldsymbol{r_i} = [r_{i,1}, r_{i,2}, ..., r_{i,K}]$}, $\boldsymbol{a_i}, \forall i \in \mathcal{A}$ {positions of
    anchor nodes}
Ensure: $\boldsymbol{x_i}, \forall i \in \mathcal{T}$ {positions of target nodes}
  $\boldsymbol{s} \leftarrow$ new Array; $\boldsymbol{d} \leftarrow$ new Array
  for all $i, j \in \mathcal{A}$ do
      $\boldsymbol{d}$.append($\|\boldsymbol{a_i} - \boldsymbol{a_j}\|$) {distance between two anchor nodes}
      $s_{i,j} \leftarrow \text{std}(\boldsymbol{r_i} - \boldsymbol{r_j})$
      $\boldsymbol{s}$.append($s_{i,j}$)
  end for
  $\alpha, \beta \leftarrow$ Linear_Regression($\boldsymbol{s}, \boldsymbol{d}$)
  for all $i \in \mathcal{A}, j \in \mathcal{A} \cup \mathcal{T}$ do
      $s_{i,j} \leftarrow \text{std}(\boldsymbol{r_i} - \boldsymbol{r_j})$
      $\hat{d}_{i,j} \leftarrow \frac{s_{i,j} - \alpha}{\beta}$
  end for
  for all $i \in \mathcal{T}$ do
      $\boldsymbol{x_i} \leftarrow$ position of node $n_i$ derived by SDP localization method
  end for
```

Theorem 1 suggests that $s_{i,j}$ can be used to estimate distance $d_{i,j}$. For the following reasons, we assume that $s_{i,j}$ is an approximately linear function of distance $d_{i,j}$, and is expressed through Equation (7.6). First, the accuracy of the assumption is illustrated through simulations (see Section 7.5). Second, a linear function has a minimal number of unknown parameters compared to other polynomial functions. Thus, these parameters are easy to estimate using measurements between anchor nodes.

$$s_{i,j} = \alpha + \beta \times d_{i,j} + Y \tag{7.6}$$

where α and β are constants, Y is a variable characterizing the error between the linearity, i.e. $\alpha + \beta \times d_{i,j}$, and $s_{i,j}$. Y is a function of the noise that is caused by RSSI noise (see the proof in Appendix) and the difference between $s_{i,j}$ and the linear function of the distance. Since Y is unknown, it is considered as a random variable. Using Equation (7.6), algorithms for resolving the two localization problems are described next.

7.4.2 LEMOn Algorithm

LEMOn is designed to solve the localization problem described in Section 7.2.1. It first calculates Euclidean distance $d_{i,j}$ and the similarity $s_{i,j}$ between each pair of anchor nodes n_i, n_j ($\forall i, j \in \mathcal{A}$), to derive values of α and β in (7.6) using the least squared linear regression technique. It then estimates distance $\hat{d}_{i,j}$ between every pair of sensor nodes n_i,n_j using the inversion of (7.6), namely:

$$\hat{d}_{i,j} = \frac{s_{i,j} - \alpha}{\beta} \tag{7.7}$$

It finally applies SDP localization method [52] (see Section 7.3.2) to estimate location of target nodes. A pseudo-code of LEMOn is described in Algorithm 3.

Algorithm 4 LEMOn-M algorithm

```
Require: $\boldsymbol{r_i}, \forall i \in \{1, ..., N\}$ $\{\boldsymbol{r_i} = [r_{i,1}, r_{i,2}, ..., r_{i,K}]\}$, $p_{i'}, \forall i' \in \{1, ..., N\}$
    {positions of sensor nodes in an arbitrary order}
Ensure: Permutation $h^*$ that node $n_i$ locates at position $p_{h^*(i)}$, $\forall i \in \{1, ..., N\}$
  $\mathcal{H} \leftarrow$ set of all permutations of a set of elements $1, 2, ..., N$
  $h^* \leftarrow \text{argmax}_{h \in \mathcal{H}} \sum_{i<j} \left(s_{i,j} d_{h(i),h(j)}\right)$ {$h^*$ is found using searching methods
    as in Chapter 5}
```

7.4.3 LEMOn-M Algorithm

LEMOn-M is designed to solve the localization matching problem described in Section 7.2.2. It is designed similar to the MLMatch algorithm described in Section 7.3.2. While MLMatch use a logarithmic relationship between an RSSI value and the corresponding distance, in LEMOn, this relation should be substitute by the linear relationship between the similarity $s_{i,j}$ and distance $d_{i,j}$. Consequently, the formula (7.5) (cf. Section 7.3.2) is substituted by (7.8).

$$h^* = \underset{h \in \mathcal{H}}{\text{argmax}} \sum_{i<j} \left(s_{i,j} d_{h(i),h(j)}\right). \tag{7.8}$$

LEMOn-M then uses the same searching methods as in MLMatch to find the best matching. A pseudo-code of LEMOn-M is described in Algorithm 4.

7.4.4 Limits on Localization Accuracy

Similar to other RSSI-based localization methods, the proposed algorithm cannot guarantee 100% accuracy. It is meaningful to understand factors causing localization error, and the impact of those factors on localization accuracy. This can give insights to optimize the localization accuracy.

As discussed, the proposed algorithms consist of two phases namely distance estimation phase and location estimation phase. Therefore, localization error can accumulate in both phases. The location estimation error is a function of number target nodes and anchor nodes, sensor geometry, and error of approximate solutions due to the location estimators. We recommend the readers to refer to Chapter 5 and [50] for further details.

As we focus on distance estimation techniques, we analyze factors that affect the distance estimation error. As discussed in Section 7.4.1 distance estimation error (which is denoted by the variable Y in Equation (7.6)) is a function of the difference of the similarity $s_{i,j}$ and the linear function of the distance $d_{i,j}$ and the noise due to the RSSI fluctuation. Theoretically, due to the proof of Theorem 1, the difference can be minimized if we increase the area of $\mathcal{V}$, i.e. making the mobile unit move in a large domain. In practice, however, the trajectory of the mobile unit is often pre-determined. In many applications, $\mathcal{V}$ typically equals to $\mathcal{D}$ which is the deployment area of sensor nodes (cf. Section 7.7). Besides, simulations in Sections 7.5 and 7.6 will show that $V = D$ is enough to realize accurate localization.

On the other hand, distance estimation error can be reduced by reducing the effect of RSSI noise, i.e. reducing the variation of random variable Z in Equation (7.12). Its variation can be reduced by increasing the number of beacons K. Note that, the distance estimation error cannot be reduced to zero even if K approaches infinity because the error is also affected by the difference described above. We call the localization accuracy that can be achieved when K approaches infinity the limitation of localization accuracy. In fact, there is a finite value of K, often called a threshold, that can achieve the limitation of localization accuracy. Finding this value can not only optimize the localization accuracy but can also optimize the number of beacons. This value, however, is difficult to be derived mathematically as it is dependent with other parameters. The value can be estimated through simulations as in Sections 7.5 and 7.6. All of the above arguments will be confirmed in the two sections.

7.5 Performance Evaluation of LEMOn Through Simulations

In order to substantiate the performance of LEMOn, we perform and analyze three simulations in various environments. Further, for reference, we also perform static cooperative localization methods for which RSSI values between individual nodes are used. Note that we do not benchmark our results against those of static methods because of significant differences in the structure and assumptions of our system. In static cooperative localization methods, we assume that all sensor nodes are perfectly calibrated and the path loss exponent is accurately estimated. We also assume that RSSI values between individual nodes are measured or known. In both cases, the SDP method is used as a localization estimator [52].

7.5.1 Propagation Models

Although our analysis described in Section 7.4.1 assume that the RSSI values follow the log-distance propagation model, we argue that it is also sufficiently accurate even if RSSI values follow other models. We substantiate our claim through numerical simulations shown below using a more advanced propagation model that is validated through indoor measurements at 2.4GHz [38]. We simulate a propagation environment experiencing Rayleigh fading and a non-singular path loss. The RSSI values $\bar{r}$ under this propagation model are generated via

$$\bar{r} = P_i - 10\log_{10}(\epsilon + d^{\eta}) + \mathcal{X} \tag{7.9}$$

where $\epsilon > 0$, and $\mathcal{X}$ is a random variable with density

$$\begin{aligned} f_{\mathcal{X}}(x) = \mathbb{P}[\mathcal{X} = x] &= \frac{\mathrm{d}}{\mathrm{d}x}\mathbb{P}\big[10\log_{10}|h|^2 \le x\big] \\ &= \lambda 10^{x/10}\exp\Big(-\lambda 10^{x/10}\Big)\frac{\ln 10}{10} \end{aligned} \tag{7.10}$$

In the rest of this chapter, we call this model *Rayleigh* model for short, while *Gaussian* model stands for the log-distance propagation model encapsulated in Equation (7.2).

7.5.2 Parameter Settings

Simulations 1 and 2 consider a UAV assisted WSNs where $N = 35$ sensor nodes are deployed randomly in a square domain $\mathcal{D} = 100 \times 100$ m^2. The mobile unit (i.e. the UAV) flies at a height of $h = 20$ m, randomly inside a domain $\mathcal{V} = \mathcal{D} = 100 \times 100$ m^2, which is an common case found in practice. The path loss exponent is set as $\eta = 3$, which is a common value noted by [36] for outdoor urban environments.

Simulation 1 investigates the localization error when numbers of beacons K varies. It validates the robustness of the proposed method under different propagation models.Using Rayleigh model, RSSI values between the devices are generated using Equation (7.9) with common parameters $\sigma_X = 5.57$, $\lambda = e^{-\gamma}$, and $\epsilon = 0.1$ (cf. Chapter 5). Using Gaussian model, RSSI values are generated using Equation (7.2) with σ_X also equal 5.57. We also use this parameter settings for all RSSI values that follow the Rayleigh model in the rest of the simulations. This way we can test the robustness of LEMOn against other propagation models. The positions of the mobile unit are also generated randomly in the domain $\mathcal{V}$. The number of beacons K is chosen in the range $\{100, 200, 300, ..., 1000\}$. This way we can observe the effect of K on the localization error.

Simulation 2 observes the localization error of LEMOn against different levels of signal noise. RSSI values are generated using a Gaussian model with σ_X is chosen in the range $\{3, 4, 5, 6, ..., 12\}$ which is as low as in Chapter 5 as high as in [127]. The number of beacons K is set as 200 and 800. This way we can test the robustness of LEMOn in a less noisy environment and very noisy environments, and also the effect of K on these scenarios.

Simulation 3 considered an indoor IoT network where sensor nodes are deployed in a room with area $\mathcal{D} = 10 \times 14$ m^2 and height $h = 3$ m. The path loss exponent is set as $\eta = 2.5$, which is a common value used for indoor environments [11, 36]. It investigates localization error of LEMOn against different numbers of sensor nodes. The number of nodes N is chosen from the set $\{15, 20, 25, ..., 45\}$ among which $A = 5$ or $A = 10$ nodes acted as anchors. RSSI values were generated using the Rayleigh model. The number of beacons K is set to 500.

In each simulation, the reference power P_i (cf. Equations (7.2), (7.9)) for each node is generated randomly. For each set of parameters, simulations are performed 20 times to obtain statistical averages.

The values of all other parameters used are detailed in Table 7.2.

7.5.3 Simulation Environment

We run our experiments with Python, a high-level programming language. In each realization, position of N sensor nodes are generated randomly under Poisson distribution in domain $\mathcal{D}$. Among these nodes, A nodes are chosen randomly acting as anchor nodes. Position of the mobile unit when it sends a beacon is also generated randomly in domain $\mathcal{V}$. An RSSI value $r_{i,k}$ is generated using either formulation (7.2) or (7.9), where random variable X follows Gaussian distribution or Rayleigh distribution, respectively. In particular, X is generated using *random.py* module, a Python pseudo-random number generator that implements various probability distributions.

7.5.4 Implementation of SDP Localization

In the proposed localization problem, we use Formula (6) of [52], which is designed to localize nodes when distances between some nodes are accurately known. On the other hand, in the static localization problem, namely localization using inter-node RSSI values, we use Formulas (11) and (12) from [52] that solve RSSI-based cooperative localization problems. We use MOSEK optimizer for Python [128] as the SDP solver.

Table 7.2: Description of parameter settings

	LEMOn			LEMOn-M		
	Sim. 1	**Sim. 2**	**Sim. 3**	**Sim. 4**	**Sim. 5**	**Sim. 6**
Localization algorithm	LEMOn	LEMOn	LEMOn	LEMOn-M	LEMOn-M	LEMOn-M
Area of $\mathcal{D}$ (m^2)	100×100	100×100	10×14	10×14	10×14	10×14
Area of $\mathcal{V}$	100×100	100×100	10×14	10×14	10×14	10×14
Height of the Mobile trajectory h (m)	20	20	3	3	3	3
Path loss exponent η	3	3	2.5	2.5	2.5	2 & 3
Propagation model	Gaussian &Rayleigh	Gaussian	Rayleigh	Gaussian &Rayleigh	Gaussian	Rayleigh
Standard deviation σ_X	5.57	$\{3, 4, ..., 12\}$	5.57	5.57	$\{3, 4, ..., 12\}$	5.57
Number of beacons K	$100, 200, ..., 1000$	200 & 800	500	$100, 200, ..., 1000$	200 & 800	500
Number of nodes N	35	35	$15, 20, ..., 45$	15	15	$11, 13, ..., 27$
Number of anchors A	10	10	5 &10	-	-	-

7.5.5 Result Analysis

In every simulation, we run LEMOn to estimate the location of target nodes using the mobile unit. For reference, we also run SDP localization scheme using inter-sensor nodes' RSSI, which is a static cooperative localization technique. We then output localization error that is the mean error (in meter). The results are illustrated by Figures 7.1, 7.2 indicating that LEMOn performs well in various environments. Localization errors from Simulation 1 are illustrated in Figure 7.1 a) indicating that the localization error decreases when the number of beacons K are increased to 400. The average localization error, however, does not vary when K varies from 400 to 1000. This confirms the argument on the threshold of K in Section 7.4.4. Besides, LEMOn performs very well regardless of the distribution of RSSI values such as the Gaussian model or the Rayleigh model. This simulation validates the robustness of LEMOn against other random variable distribution models for wireless fading such as the Rayleigh distribution.

Localization errors from Simulation 2 are illustrated in Figure 7.1 b) indicating that the localization error increases with the level of noise, especially when the value of K is not large enough, e.g. $K = 200$. However, for a large K (e.g., $K = 800$), LEMOn can achieve accuracy of less than 10 m in a noisy environment characterized by a $\sigma_X = 10$ when static localization technique fails (cf. Figure 7.2 b). This is because static localization technique that estimates inter-node separation distance directly from RSSI measurement between them suffer a large error. The calculation in Section 2.4 shows that RSSI-based ranging error in this environment is even larger than the inter-node separation distance. On the other hand, LEMOn uses statistical RSSI similarity, thus can suppress error caused by a single RSSI value. Simulation 2 validates the robustness of LEMOn in noisy environments. Simulations 1 and 2 confirm the argument in Section 7.4.4 regarding the impact of the number of beacons on localization accuracy. Besides, the localization error can be minimized to 5- 10 m even in noisy environments.

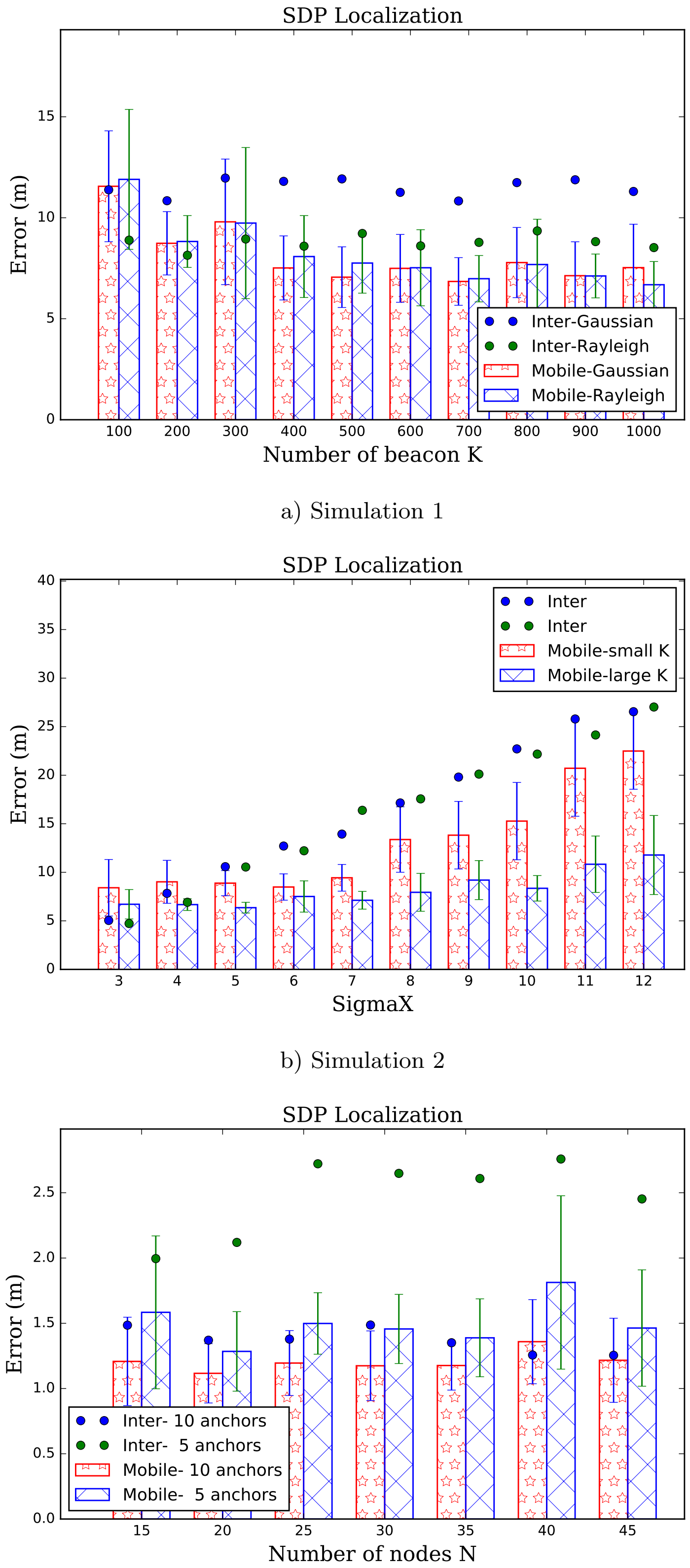
Localization errors from Simulation 3 are illustrated in Figure 7.1 c) indicating that the localization error does not vary a lot with the variation in the number of nodes N. On the other hand, decreasing number of anchors slightly reduces the localization accuracy. Especially, LEMOn performs much better than the static SDP localization when there is a small number of anchors (for example $A = 5$). This suggests that a high localization accuracy can be achieved using LEMOn even for a small number of anchor nodes. Besides, the localization error is around 1.5 m in this indoor environment.

The relationship between the similarity $s_{i,j}$ and inter-node distances $d_{i,j}$ is illustrated in Figure 7.2 c) indicating that the similarity is approximately linearly related to the distance. This confirms the argument on the linearity discussed in Section 7.4.1. Figure 7.2 c) also depicts that the linear fitting of distance between anchor nodes and their similarities $s_{i,j}$ is close to that of the target nodes. This enables node localization without a priori measurements.

All above results confirmed the robustness of LEMOn in various environments, and against different propagation models. Especially, it is robust even in very noisy environments.

7.6 Performance Evaluation of LEMOn-M Through Simulations

Similar to simulations in Section 7.5, we performed three simulations in different environments. Further, for reference, we also performed the static localization matching method, i.e. inter-sensor nodes RSSI values are used. In the static

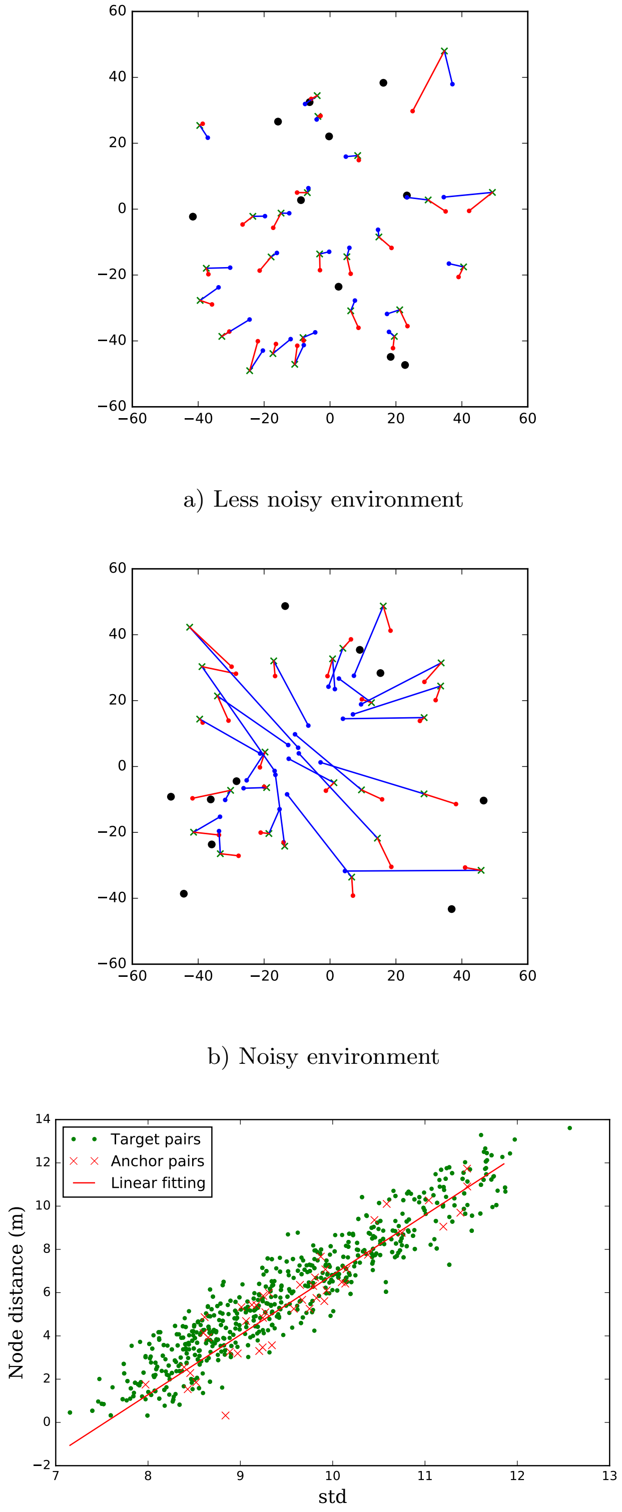


a) Simulation 1

b) Simulation 2

c) Simulation 3

Figure 7.1: Simulations 1-3 results: Average localization error obtained by LEMOn (bars) and by using static localization, i.e. using inter-node RSSIs (circles). a) Simulation 1: Impact of number of beacons on localization accuracy; b) Simulation 2: Impact of noisy level on localization accuracy; c) Simulation 3: Impact of number of nodes on localization accuracy.



a) Less noisy environment

b) Noisy environment

c) Relationship between RSSI similarity and distances

Figure 7.2: Simulations 1-3 results: a) A visualization of location estimation results by LEMOn (red triangles) and by using inter-node RSSI (blue dots) in a less noisy environment, and b) in a noisy environment; c) Relationship between RSSI similarity and distances.

localization matching method, we assume that all sensor nodes are perfectly calibrated. We also assume that RSSI values between all node-pairs are obtained.

7.6.1 Parameter Settings

Similar to Simulation 3 in Section 7.5, we consider an indoor IoT network where wireless nodes are deployed in a room with area $\mathcal{D} = 10 \times 14$ m^2 and height $h = 3$ m. The path loss exponent η is set to 2, 2.5, and 3 based on the real empirical measurements acquired from multiple indoor environments in Chapter 5. Use of multiple path loss exponents thus enables us to evaluate our protocols in different indoor environments. Wireless nodes attached to light bulbs and air conditioners serve as fixed objects, while a ground vehicle that moves around the room acts as a mobile unit. Furthermore, $\mathcal{V}$ is set equal to $\mathcal{D}$. Other parameters are set to exactly same as the three simulations described in Section 7.5 to evaluate the performance of LEMOn-M in different scenarios, and whether it performs similarly to LEMOn.

Simulation 4 evaluates the localization accuracy when the number of beacons K varies. It also validates the robustness of the LEMOn-M under different propagation models. The position of the mobile unit is also generated randomly in the domain $\mathcal{V}$. The number of beacons K is chosen from the set $\{100, 200, 300, ..., 1000\}$. RSSI values are generated randomly using the Gaussian model or the Rayleigh model. This way we can observe the effect of K on the localization error.

Simulation 5 observes the localization accuracy of LEMOn-M against different levels of signal noise. RSSI values are generated using a Gaussian model with σ_X chosen from the set $\{3, 4, ..., 12\}$. The number of beacons K is set to 200 and 800. This way we can test the robustness of LEMOn-M in environments characterized by different noise levels, and also the impact of K on these scenarios.

Simulation 6 studies localization accuracy of LEMOn-M against different numbers of sensor nodes. The number of nodes N is chosen in the range $\{11, 13, ..., 27\}$. RSSI values are generated using the Rayleigh model. The number of beacons K is set to 500. The path loss exponent η is set to 2 or 3, which is different than Simulations 4 and 5, in order to test the performance of LEMOn-M in other environments.

In each simulation, the reference power P_i (cf. Equations (7.2), (7.9)) for each node is generated randomly. For each set of parameters, simulations are run 20 times to obtain statistical averages. The values of all other parameters used are detailed in Table 7.2.

7.6.2 Results Analysis

In every simulation, we run LEMOn-M and to determine the best matching between nodes and positions using the mobile unit. For reference, we also run MLMatch localization using RSSI values between nodes that is the static localization matching technique. We then output localization accuracy that is defined as the ratio between the number of accurately matched nodes and the number of total nodes (cf. Chapter 5). The results are illustrated by Figure 7.3 indicating that LEMOn-M performs well in various environments. Localization accuracy from Simulation 4 are illustrated in Figure 7.3 a). Localization accuracy increases with increase in the number of beacons K up to 500. The average localization accuracy, however, does not vary when K increases further from 500 to 1000. Besides, LEMOn-M performs very well regardless the RSSI values follow the Gaussian model or the Rayleigh model. This validates the robustness of LEMOn-M against other random variable distribution models for wireless fading such as the Rayleigh distribution.

Localization accuracy from Simulation 5 are illustrated in Figure 7.3 b) indicating that the localization accuracy decreases when the level of noise increases, especially when K is not large enough, e.g. $K = 200$. However, when K is large (e.g. $K = 800$), LEMOn-M can achieve localization accuracy even in very noisy environment.

Localization accuracy from Simulation 6 are illustrated in Figure 7.3 c) indicating that the localization accuracy does not vary a lot with the variation of the number of nodes N.

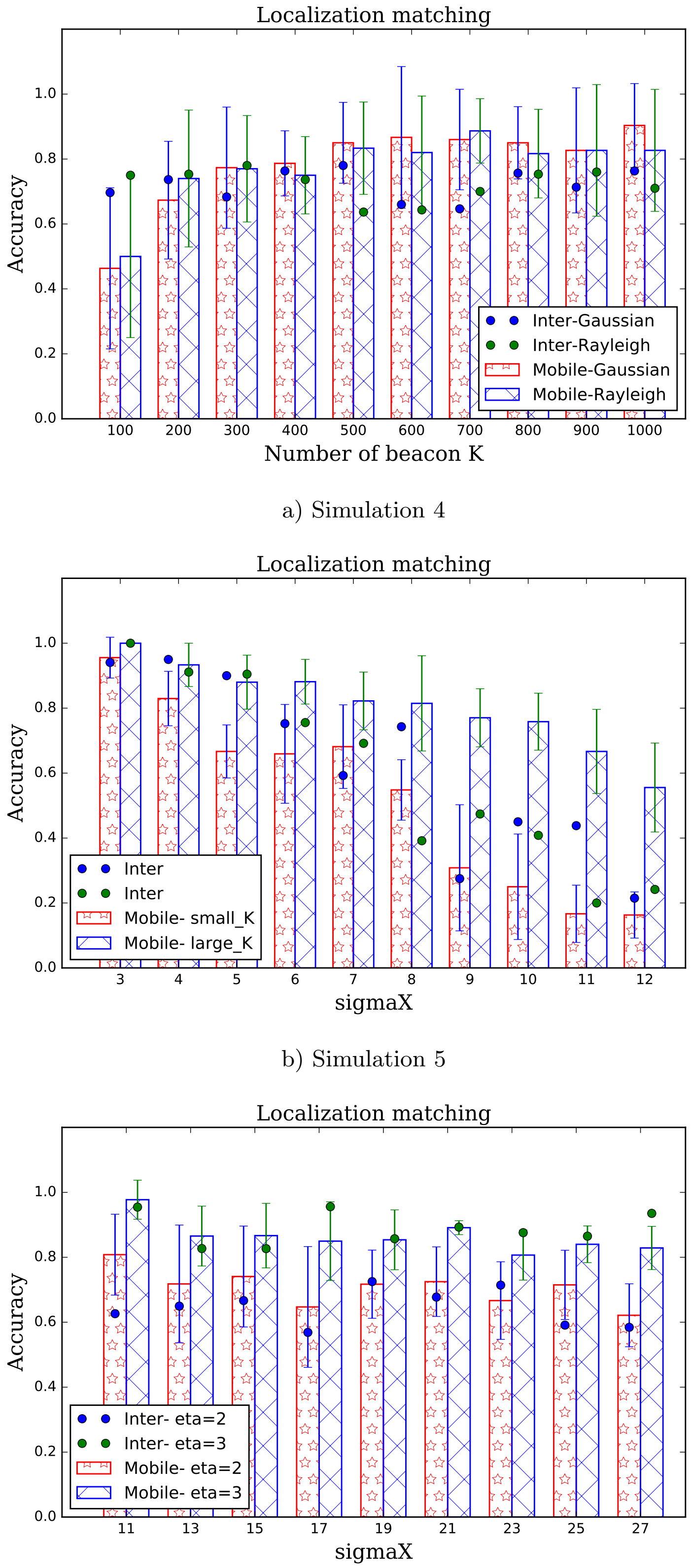
The trend of localization error by LEMOn-M is similar to that in LEMOn. Our results show that both algorithms can outperform static cooperative localization methods when there are sufficient number of beacon nodes K. Both methods are resilient against high noise level especially for sufficiently large K, which introduces diverse variations of distance between the mobile unit and sensor nodes. It helps in suppressing the effect of noise by shadowing and fading. On the other hand, in static localization techniques, multiple transmissions between two fixed nodes can reduce the effect only if nodes are equipped with specific hardware or technologies (cf. Section 7.1).

7.7 Applications

This section highlights some real-world applications of LEMOn and LEMOn-M.

7.7.1 UAV Assisted WSNs

The proposed algorithm can be used in a UAV assisted WSN which has various applications. For instance in agriculture, a UAV can control the amount of chemical sprayed over a piece of land [129], or to gather data from deployed wireless sensors [130]. For the purpose of post-disaster monitoring, UAVs are used to deploy a WSN in the area [131]. For data collection, a UAV is designed to collect data from a WSN efficiently [132], or to dispatches mobile agents that are used to collect data [133]. In all of these applications, it is necessary to collect sensed data from nodes and is necessary to relate the stream of data to the location of the corresponding sensor node. The UAV is, therefore, planned to fly along an optimal trajectory to collect the data from the WSN. To cover all sensor nodes without leaving any gap, the UAV is made to fly through an operation area parallel to the sensor node deployment plane [134, 135] (see Figure 7.4). Because a sensor node should detect whether the UAV is nearby, the UAV is assumed to transmit beacon messages periodically (e.g. every two seconds in [135]). It is often assumed that there is a small number of sensor nodes that know their positions either through GPS or deployment time configuration. These nodes are often used as a cluster head that collects data from nearby nodes. Since the energy supply of the UAV is not limited as that of sensor nodes, it is possible to assume that the transmit power of the UAV is large enough so that all the sensor can receive the beacons. Under this network model, we can obtain all necessary information to perform LEMOn. Further, simulation results in Section 7.5 show that LEMOn can provide an accuracy of 5-10 m which is similar to the accuracy that one can get from enhanced GPS.



a) Simulation 4

b) Simulation 5

c) Simulation 6

Figure 7.3: Average localization accuracy obtained by LEMOn-M (bars) and by using static localization, i.e. using inter-node RSSIs (circles). a) Simulation 4: Impact of number of beacons on localization accuracy; b) Simulation 5: Impact of noisy level on localization accuracy; c) Simulation 6: Impact of number of nodes on localization accuracy.

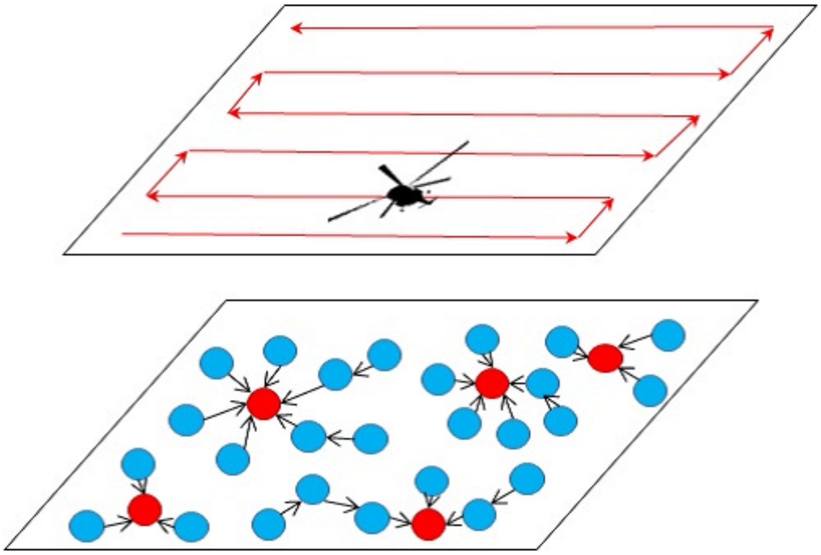


Figure 7.4: A UAV traveling around to collect data from sensor nodes

7.7.2 Wireless Localization for Indoor IoT

We consider an indoor IoT system where objects, for instance, light bulbs, air conditioners, TVs, fans, etc., are equipped with wireless transceivers (see Figure 7.5). To control each object, it is necessary to match its location and its ID (e.g., MAC address). Whilst the equipment positions of some fixed objects such as light bulbs, air conditioners, are well known from the floor plan blueprints (often found in out-of-reach positions, e.g., behind ceiling panels or rooftops), the specific equipment ID may not be recorded by installation engineers due to high manual labor cost (cf. Chapter 5).

Chapter 5 proposes a new problem called *the wireless localization matching problem (WLMP)* that automatically matches an object's position and its ID. The problem is resolved using RSSI values between all pairs of devices, and the set of positions of devices that are known from floor plan blueprints. However, in some network systems, collecting RSSI values between those devices complicates the network design. This is because wireless devices are often designed to communicate to some peculiar wireless controllers or sink nodes rather than to each other. The proposed solution in Chapter 5 is, therefore, inappropriate for typical sensor networked systems.

LEMOn-M can resolve the above problem. Moving the controller or the sink node around the floor and connecting it with a UAV or a ground robot such as vacuum cleaning robot can efficiently collect RSSI values from the wireless devices. A vacuum cleaning robot is often designed to move around a floor, thus making collection process of the RSSI values feasible.

Simulation results in Section 7.6 show that LEMOn-M outperforms in doing the localization matching in the wireless mesh networks , thus being feasible for real-world applications. Consequently, devices such as light bulbs and air conditioners can be matched to their exact position efficiently.

Furthermore, besides matching known fixed positions to the devices using LEMon-M, LEMon can be used to localize devices for which localization is not known at all. Simulation 3 in Section 7.5 shows that LEMOn achieves less than 2 meters of accuracy on average. This is sufficient for many indoor applications. Figure 7.6 illustrates an application use-case common to today's smart homes. Electronic devices are controlled through an Infrared Radiation (IR) controller.

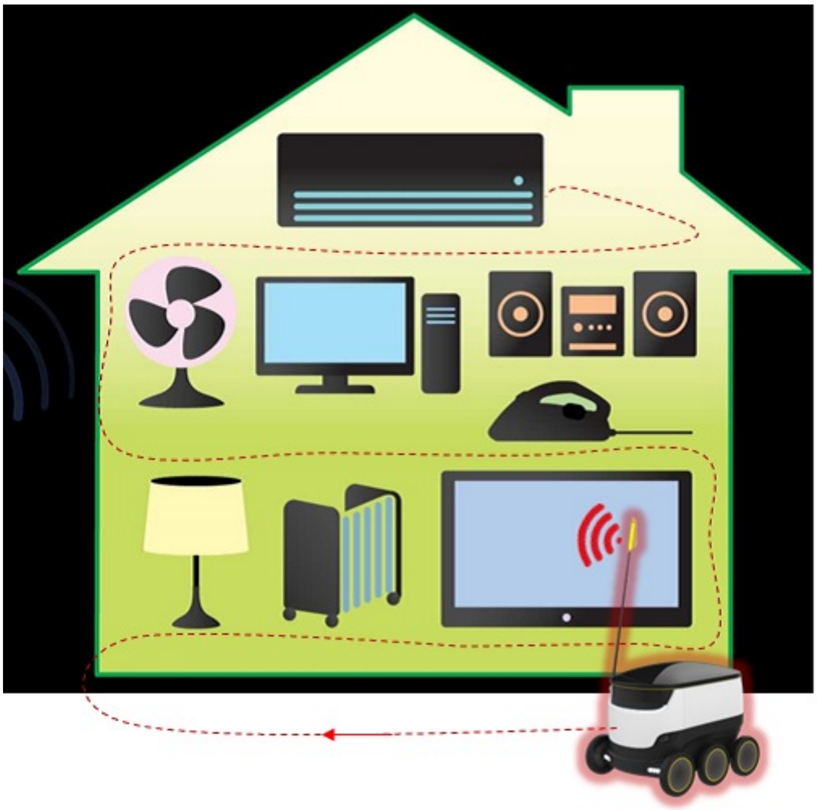


Figure 7.5: A wireless mobile unit going around to collect RSSIs from IoT devices

The emitting angle of an IR controller is often between ±10° and ±60°. We assume that emitting angle of the IR controller is ±25°, and the distance between the controller and an electronic device (e.g. a TV) is larger than 5 m. Then the IR controller can control the TV using an estimated position of TV that is less than 2m away from its real position.

7.8 Conclusions

This chapter proposes two novel localization methods namely LEMOn and LEMOn-M that use a location-unaware mobile unit to help estimating location of other wireless nodes. The mobile unit can be either an UAV, a ground robot, or a mobile access point. The method can, therefore, be used for many IoT systems, for instance, UAV assisted WSNs which have attracted great attention recently.

The proposed methods advance conventional localization methods in several dimensions. First, these can be used in both indoor and outdoor environments because he location of the mobile unit is not needed. Second, the techniques are hardware-independent and do not require complex calibration, making these suitable for a wide variety of emerging IoT applications. Third, we do not need to run lengthy campaigns to estimate parameters (such as path loss exponents) of each individual radio environment wherever the applications are to be deployed.

We extensively evaluated the performance of LEMOn and LEMON-M using simulations. LEMOn achieves a 5-10 m accuracy on average similar to a GPS in outdoor environments. In outdoors, LEMOn accurately localize within 2 m on average, which makes it suitable for many indoor applications. LEMOn-M is shown to outperform the static localization matching techniques even in very

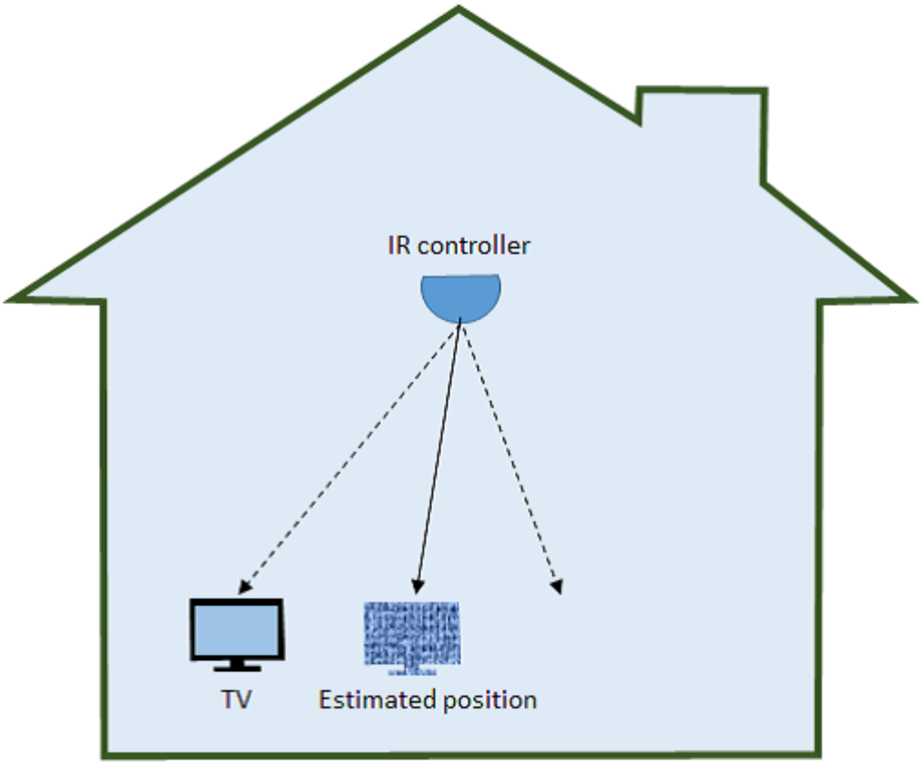


Figure 7.6: An example of application of indoor IoT localization

noisy environments where competing solutions fail. Consequently, LEMOn-M is a alternative solution of the proposed MLMatch described in Chapter 5 using star network models.

7.9 Appendix: Proof of Theorem 1

Notations, and assumptions used in the proof are first described. $s_{i,j}$ denotes standard deviation of vector $(\boldsymbol{r_i} - \boldsymbol{r_j})$. $\boldsymbol{c_k}$ denotes the position, i.e. coordinate, of the mobile unit when it sends the k-th beacons. $\boldsymbol{n_i}$ denotes the position of node n_i. $\|\boldsymbol{x} - \boldsymbol{y}\|$ is the Euclidean distance between two points locating at positions $\boldsymbol{x}$ and $\boldsymbol{y}$, e.g., $d_{i,j} = \|\boldsymbol{n_i} - \boldsymbol{n_j}\|$. $\ln \cdot$ denotes the nature logarithm. x is the magnitude of vector $\boldsymbol{x}$. V and D are the areas of $\mathcal{V}$ and $\mathcal{D}$. For simplicity, $\mathcal{V}$ and $\mathcal{D}$ are assumed to be parallel to each other; h denotes the distance between them.

Using Equation (7.2), we have

$$\begin{aligned} r_{i,k} - r_{j,k} &= P_i - P_j - 10\eta \log_{10} \frac{d_{i,k}}{d_{j,k}} + X_{i,k} - X_{j,k} \\ &= P_i - P_j - \frac{10\eta}{\ln 10}\big(\ln \frac{d_{i,k}}{d_{j,k}} + X_k\big) \end{aligned} \tag{7.11}$$

where, $X_k = \frac{(X_{i,k} - X_{j,k}) \ln 10}{10\eta}$. Let $\delta = \frac{10\eta}{\ln 10}$, then $X_k \sim \mathcal{N}(0, 2(\frac{\sigma_X}{\delta})^2)$. Since $(P_i - P_j)$ and δ are constant, the standard deviation $s_{i,j}$ of the vector $\boldsymbol{r}_i - \boldsymbol{r}_j$ divided by δ equals the standard deviation of a vector $\boldsymbol{r}$ whose entries equals

$r_k = \ln \frac{d_{i,k}}{d_{j,k}} + X_k$. Namely,

$$\begin{aligned}(\frac{s_{i,j}}{\delta})^2 &= \frac{1}{K}\sum_{1\le k\le K}\big(\ln\frac{d_{i,k}}{d_{j,k}} + X_k - \ln\frac{\bar{d}_i}{\bar{d}_j} - \bar{X}\big)^2\\ &= \frac{1}{K}\sum_{1\le k\le K}(X_k-\bar{X})^2\\ &+ \frac{1}{K}\sum_{1\le k\le K}2(X_k-\bar{X})(\ln\frac{d_{i,k}}{d_{j,k}} - \ln\frac{\bar{d}_i}{\bar{d}_j})\\ &+ \frac{1}{K}\sum_{1\le k\le K}(\ln\frac{d_{i,k}}{d_{j,k}} - \ln\frac{\bar{d}_i}{\bar{d}_j})^2\end{aligned} \tag{7.12}$$

where $\bar{d}_i, \bar{d}_j$, and $\bar{X}$ denote, respectively, the average value of distance $d_{i,k}, \forall k$, $d_{j,k}, \forall k$, and $X_k, \forall k$. The first term of the right side of (7.12) is the variance of X_k as K being large enough, thus equaling $2(\sigma_X/\delta)^2$. We denote Z the second term of the right side of (7.12). Due to Central Limit Theorem (CLT), Z is a random variable following zero mean Gaussian distribution. Consequently, (7.12) equals

$$\begin{aligned}(\frac{s_{i,j}}{\delta})^2 &\approx Z + \frac{2\sigma_X^2}{\delta^2} + \frac{1}{K}\sum_{1\le k\le K}(\ln\frac{d_{i,k}}{d_{j,k}} - \ln\frac{\bar{d}_i}{\bar{d}_j})^2\\ &\approx Z + \frac{2\sigma_X^2}{\delta^2} + \lim_{K\to+\infty}\frac{1}{K}\sum_{1\le k\le K}(\ln\frac{\|\boldsymbol{c_k}-\boldsymbol{n_i}\|}{\|\boldsymbol{c_k}-\boldsymbol{n_j}\|} - \ln\frac{\bar{d}_i}{\bar{d}_j})^2\\ &\approx Z + \frac{2\sigma_X^2}{\delta^2} + \int_{\mathcal{V}}(\ln\frac{\|\boldsymbol{x}-\boldsymbol{n_i}\|}{\|\boldsymbol{x}-\boldsymbol{n_j}\|} - \ln\frac{\bar{d}_i}{\bar{d}_j})^2 d\boldsymbol{x}\end{aligned} \tag{7.13}$$

Using the assumption that $\mathcal{V}$ is large enough, $\mathcal{V}$ can approximately be assumed to be a disc having radius of $R = \sqrt{V/\pi}$, and centered at the projection of the midpoint of $\boldsymbol{n_i}$ and $\boldsymbol{n_j}$ onto the plane $\mathcal{V}$ (see Figure 7.7). Under this assumption, $\bar{d}_i$ equals $\bar{d}_j$ therefore $\ln\frac{\bar{d}_i}{\bar{d}_j}$ can be ignored. Let θ be the angle between vector $\boldsymbol{x}$ and the projection of vector $\boldsymbol{n_j} - \boldsymbol{n_i}$ onto domain $\mathcal{V}$, distances between $\boldsymbol{x}$ and $\boldsymbol{n_i}, \boldsymbol{n_j}$ are formulated as (7.14)

$$\begin{aligned}\|\boldsymbol{x}-\boldsymbol{n_i}\|^2 &= h^2 + x^2 + (d_{i,j}/2)^2 + xd_{i,j}\cos\theta\\ \|\boldsymbol{x}-\boldsymbol{n_j}\|^2 &= h^2 + x^2 + (d_{i,j}/2)^2 - xd_{i,j}\cos\theta\end{aligned} \tag{7.14}$$

Using Taylor series expansion of $\ln(a+x)$ when $x < a$, i.e. $\ln(a+x) \approx \ln a + x/a$, (7.15) is obtained.

$$\begin{aligned}&\frac{\ln\|\boldsymbol{x}-\boldsymbol{n_i}\|}{\ln\|\boldsymbol{x}-\boldsymbol{n_j}\|} = \frac{1}{2}(\ln\|\boldsymbol{x}-\boldsymbol{n_i}\|^2 - \ln\|\boldsymbol{x}-\boldsymbol{n_j}\|^2)\\ &\approx \frac{1}{2}\big(\ln(h^2+x^2+(d_{i,j}/2)^2) + \frac{xd_{i,j}\cos\theta}{h^2+x^2+(d_{i,j}/2)^2}\\ &- (\ln(h^2+x^2+(d_{i,j}/2)^2) - \frac{xd_{i,j}\cos\theta}{h^2+x^2+(d_{i,j}/2)^2})\big)\\ &\approx \frac{xd_{i,j}\cos\theta}{h^2+x^2+(d_{i,j}/2)^2}\end{aligned} \tag{7.15}$$

By substituting (7.15) into (7.13) we have:

$$
\begin{aligned}
&(\tfrac{s_{i,j}}{\delta})^2 \\
&\approx Z + \frac{2\sigma_X^2}{\delta^2} + \frac{1}{\pi R^2}\int_0^R x dx \int_{-\pi}^{\pi} (\frac{x d_{i,j}\cos\theta}{h^2+x^2+(d_{i,j}/2)^2})^2 d\theta \\
&\approx Z + \frac{2\sigma_X^2}{\delta^2} + \frac{1}{\pi R^2}\int_0^R \frac{\pi x^3 d_{i,j}^2}{(h^2+x^2+(d_{i,j}/2)^2)^2} dx \\
&\approx Z + \frac{2\sigma_X^2}{\delta^2} + \frac{d_{i,j}^2}{2R^2}\int_0^{R^2} \frac{y}{(y+h^2+(d_{i,j}/2)^2)^2} dy \\
&\approx Z + \frac{2\sigma_X^2}{\delta^2} + \frac{d_{i,j}^2}{2R^2}\int_0^{R^2} (\frac{1}{y+h^2+(d_{i,j}/2)^2} \\
&- \frac{h^2+(d_{i,j}/2)^2}{(y+h^2+(d_{i,j}/2)^2)^2}) dy \\
&\approx Z + \frac{2\sigma_X^2}{\delta^2} + \frac{d_{i,j}^2}{2R^2}\big(\ln\frac{R^2+h^2+(d_{i,j}/2)^2}{h^2+(d_{i,j}/2)^2} \\
&- \frac{R^2}{R^2+h^2+(d_{i,j}/2)^2}\big)
\end{aligned}
\tag{7.16}
$$

Let $x = \big(\frac{d_{i,j}}{2R}\big)^2$, (7.16) can be rewritten as

$$
\begin{aligned}
s_{i,j} \approx \delta\big[Z + \frac{2\sigma_X^2}{\delta^2} + \frac{x}{2}\big(\ln\frac{1+(h/R)^2+x}{(h/R)^2+x} \\
- \frac{1}{1+(h/R)^2+x}\big]^{1/2}
\end{aligned}
\tag{7.17}
$$

Since R is large enough, x is smaller than 1, and is close to 0. Therefore, applying Taylor series expansion around $x = 0$ to (7.17), similarity $s_{i,j}$ approximates

$$
\begin{aligned}
s_{i,j} \approx \delta\big[(Z + \frac{2\sigma_X^2}{\delta^2})^{1/2} + \frac{1}{4(Z+\frac{2\sigma_X^2}{\delta^2})^{1/2}} \\
\times \big(\ln\frac{1+(h/R)^2}{(h/R)^2} - \frac{1}{1+(h/R)^2}\big)x + \mathcal{O}(x^2)\big]
\end{aligned}
\tag{7.18}
$$

Consequently, similarity $s_{i,j}$ is a monotonically increasing polynomial function of distance $d_{i,j}$. It, therefore, can be used to estimate distance $d_{i,j}$.

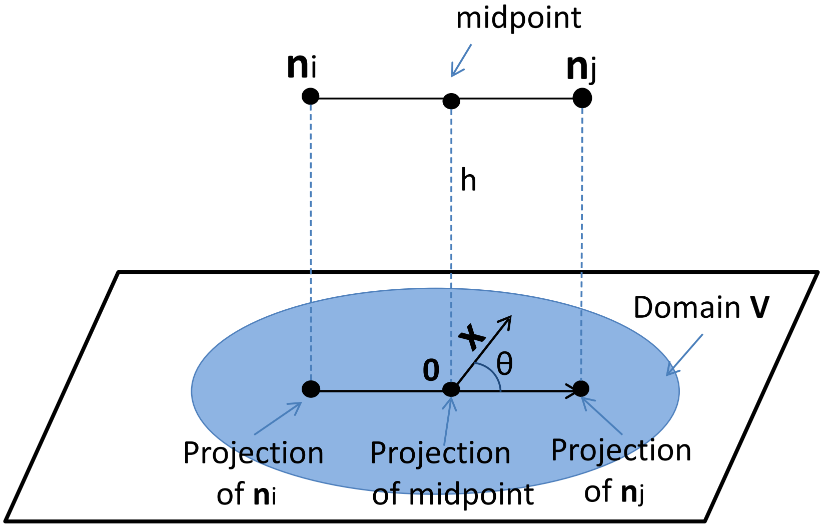


Figure 7.7: An example of geometric relationship between domain $\mathcal{V}$ and sensor nodes.

Chapter 8

Conclusion and Future Perspectives

8.1 Conclusion

Wireless localization refers to extracting geo-locating information of an object based on its wireless signals to multiple known devices. It plays important roles in many Internet of Things (IoT) systems due to its numerous important applications, particularly industrial applications, commercial environments, public safety settings, everyday life and defense/security systems. However, although wireless localization has come a long way, there are many unconventional localization problems which remain mostly unexplored.

The main purpose of this dissertation is to propose and motivate a new localization problem, as well as proposes and develops practical techniques to resolve the problem. The localization problem relates to scenarios where the device positions are known *a priori*, however, the device IDs are not, and therefore need to be matched using radio frequency methods. The problem, called *WLMP: the wireless localization matching problem*, is motivated through various real-world applications including, but not limited to, disaster prevention wireless sensor networks (WSNs), indoor positioning, and smart lighting and heating systems [11, 12].

To build accurate and practical localization systems, we have approached the WLMP step by step: investigation phase, preliminary study phase, and WLMP resolving phase. In the investigation phase, we first studied characteristics of wireless hardware equipped with most IoT devices. We then study the properties of major wireless localization systems and techniques to derive the most suitable techniques for IoT devices. As a result, we concluded that low-cost radio frequency (RF), especially Received Signal Strength Indicator (RSSI) measurement technique, is appropriate for most IoT devices due to their simplicity, small energy cost, and low processing overhead than other measurement techniques.

In the preliminary study phase, we exploited low-cost localization techniques through two different case studies. In the first case study, we proposed a maximum likelihood-based multihop localization method, called *kHopLoc: k-hop localization*, that is from 20% to 40% more accurate than conventional methods, thus confirming that localization accuracy of low-cost localization can be improved [8]. Besides, extensive simulations in various environments demonstrated that this range-free technique can produce coarse location of wireless devices, thus substantiating the validity of low-cost localization. In the second case study, we proposed an RSSI-based localization method, called *WiLAD: wireless localization through anomaly detection*, that used one-class classification [10]. We also proposed a low-cost devices placement optimization method that was in contrast to conventional methods that relied on expensive simulations or experiments. Experiments in a real store verified that the proposed WiLAD can detect whether a wireless device is taken out of a store with an accuracy of approximately 100%, thus satisfying requirements of defense/security applications. Besides, the proposed devices placement method was substantiated to improve the localization accuracy without additional cost. In conclusion, the above two case studies validated the practicability of low-cost RF localization: it can achieve high accuracy enough for some specific applications, and localization accuracy can be improved without additional cost.

In the WLMP resolving phase, which is the main phase, we proposed low-cost localization methods to resolve the WLMP in different scenarios and network paradigms [11, 12]. We first proposed practical methods, called *MLMatch: maximum likelihood-based localization matching* and *MLMatch3D: MLMatch for 3 dimensions* that resolved the WLMP under mesh network paradigms. The main advantage of the MLMatch algorithms is the use of likelihood estimation to reduce the error rate. Flexibility in searching methods is also an advantage as accuracy can be traded off with localization speed according to the application setting. In a 3D setting where nodes are located within a multi-room or multi-story building, the main advantage of MLMatch3D is that it can solve the 3D-WLMP even when the signal attenuation through walls is uncertain or unknown. Not only have we proposed and defined the WLMP, but we have also numerically and experimentally analyzed the main factors that affect the accuracy of our algorithms. Extensive experiments in various environments illustrated that the proposed algorithms can achieve high localization accuracy enough for a number of real-world IoT applications. For instance, the proposed MLMatch3D which can achieve 100% room-level of accuracy can be successfully applied into IoT devices control and management at room-level.

We then proposed a method called *MLRefine* to refine raw RSSI data collected from wireless sensor networks in order to improve localization accuracy [13]. MLRefine uses machine learning algorithms to extract and exploit the inherent spatial network geometrical correlations that are hidden in noisy RSSI datasets used for RF wireless localization. These correlations are captured by features that are then used to reduce the magnitude of RSSI fluctuations, in turn improving the localization accuracy. We note that training data can be generated off-line, thus minimizing the cost of collecting training data. We have validated the efficiency of MLRefine through extensive computer simulations but also real-world experiments. Especially, it can improve the localization accuracy of MLMatch up to 10%, thus improving the practicality of MLMatch.

Finally, we proposed a new approach for resolving the WLMP under star network models, which is in contrast with mesh network models assumed above [15]. We addressed the problem of localizing sensor nodes using a mobile wireless unit. This, in contrast to existing works, does not assume the mobile unit knows its own location, thus can be used in indoor. We proposed a new ranging method that uses the similarity between sets of RSSI values rather than using conventional propagation models. The proposed localization algorithms, therefore, require neither a priori measurements nor accurate positions of the mobile unit, thus, can be deployed practically. Extensive simulations in different environments validated high localization accuracy of the proposed algorithms even under noisy channel conditions. Especially, localization method for the WLMP outperformed that in mesh network models, thus being applicable.

The above localization algorithms are suitable for many IoT applications due to several reasons. First, they use RSSI values which can be read using any wireless hardware. Second, they do not require propagation model parameters, thus expensive prior measurements are not required. Third, they use only input features that can be obtained practically. Finally, they can achieve high localization accuracy that is enough for many real-world IoT applications.

8.2 Future Perspectives

The proposed WLMP can be applied in numerous real-world IoT applications. The initial purpose of the WLMP is to reduce the manual installation cost required by smart lighting systems which are currently commercially available. Recalling that lighting in commercial buildings contributes to about 38% of their total energy output, that lighting is responsible for 19% of global electricity consumption and accounts for 6% of total greenhouse emissions, and finally that smart-lighting can reduce these numbers by up to 40% [90], it becomes paramount to devise methods to facilitate the wide-spread adoption of WSNs and green IoT. The WLMP can be further applied into other IoT applications, especially heating, ventilation, and air conditioning (HVAC) units, thermostats, security units, lighting, or some other fixed infrastructure IoT. Recalling that the number of IoT devices has been increasing exponentially recently, for instance, there are nine billion interconnected devices in 2013 and it is expected to reach 24 billion devices by 2020, and this amounts to 1.3 trillion US Dollars revenue opportunities [5], opportunities for applications related to the IoT are great. The WLMP is one such challenge.

Besides, this dissertation opens several avenues for future research:

1. *Combination with image processing techniques*: The WLMP combined with image processing techniques can be further applied into ordinary WSNs to derive the position of sensor nodes. For instance, in disaster prevention WSNs, sensors are often deployed randomly (e.g., sensors are air-dropped from an airplane) [46], and their set of locations are obtained using a camera-equipped unmanned aerial vehicle (UAV) and image processing techniques.

2. *Combination with unmanned aerial vehicle control techniques*: Wireless localization techniques that use a mobile unit can be combined with UAVs control techniques in UAV-enable WSNs not only to enhance localization accuracy, but also to enhance connectivity of sensor nodes, or to enhance the energy efficiency of the UAV.

3. *Combination with data processing techniques*: Outliers caused by obstacles, multi-path, and noisy environments can decrease the proposed algorithms' accuracy. These deteriorating effects should be detected by future algorithms and removed efficiently. In addition, advanced data processing techniques can be evoked to counter the effects of outliers due to sudden signal fluctuations, fading, interference, etc.

4. *Generalization to large-scale networks*: Although this dissertation focused on developing localization methods for small-scale networks, i.e. one hop networks, this can be generalized into large-scale networks, i.e. multi-hop networks.

5. *Combination with different measurement methods*: Hybrid wireless localization techniques that combine RSSI with other RF measurement methods such as ToA, TDoA, and AoA are attracted great attention. Similarly, combining RSSI with such measurement techniques can improve the localization accuracy for the WLMP.

References

[1] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, “A survey on 5G networks for the Internet of Things: Communication technologies and challenges,” *IEEE Access*, vol. 6, pp. 3619–3647, 2018.

[3] E. Borgia, “The Internet of Things vision: Key features, applications and open issues,” *Computer Communications*, vol. 54, pp. 1–31, 2014.

[4] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. N. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, “Farmbeats: An IoT platform for data-driven agriculture.,” in *NSDI*, pp. 515–529, 2017.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of Things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[6] A. A. Khan, M. H. Rehmani, and A. Rachedi, “Cognitive-radio-based Internet of Things: Applications, architectures, spectrum related functionalities, and future research directions,” *IEEE wireless communications*, vol. 24, no. 3, pp. 17–25, 2017.

[7] N. H. Motlagh, M. Bagaa, and T. Taleb, “UAV-based IoT platform: A crowd surveillance use case,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[8] C. Nguyen, O. Georgiou, and Y. Doi, “Maximum likelihood based multihop localization in wireless sensor networks,” in *2015 IEEE International Conference on Communications (ICC)*, pp. 6663–6668, IEEE, 2015.

[9] O. Georgiou and C. Nguyen, “Multihop connectivity of ad hoc networks with randomly oriented directional antennas,” *IEEE Wireless Communications Letters*, vol. 4, no. 4, pp. 369–372, 2015.

[10] C. L. Nguyen and A. Khan, “Wilad: Wireless localisation through anomaly detection,” in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–7, IEEE, 2017.

[11] C. L. Nguyen, O. Georgiou, Y. Yonezawa, and Y. Doi, “The wireless localisation matching problem,” *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1312–1326, 2017.

[12] C. L. Nguyen, O. Georgiou, Y. Yonezawa, and Y. Doi, “The wireless localisation matching problem and a maximum likelihood based solution,” in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2017.

[13] C. L. Nguyen, O. Georgiou, and V. Suppakitpaisarn, “Improved localization accuracy using machine learning: Predicting and refining RSS measurements,” in *IEEE GLOBECOM 2018 Workshops: Machine Learning for Communications (MLComm 2018)*, p. to appear, IEEE, 2018.

[14] C. L. Nguyen and U. Raza, “LEMOn: Wireless localization for IoT employing a location-unaware mobile unit,” *IEEE Access*, p. to appear, 2019.

[15] C. L. Nguyen and U. Raza, “Localization of WSNs using a location-unaware UAV,” in *IEEE International Conference on Communications (ICC)*, p. to appear, IEEE, 2019.

[16] P. Misra and P. Enge, “Global positioning system: signals, measurements and performance (second edition),” *Massachusetts: Ganga-Jamuna Press*, 2006.

[17] J. Qi and G.-P. Liu, “A robust high-accuracy ultrasound indoor positioning system based on a wireless sensor network,” *Sensors*, vol. 17, no. 11, p. 2554, 2017.

[18] G. Monaci and A. Pandharipande, “Indoor user zoning and tracking in passive infrared sensing systems,” in *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pp. 1089–1093, IEEE, 2012.

[19] N. Bulusu, J. Heidemann, and D. Estrin, “GPS-less low-cost outdoor localization for very small devices,” *IEEE personal communications*, vol. 7, no. 5, pp. 28–34, 2000.

[20] A. K. Paul and T. Sato, “Localization in wireless sensor networks: A survey on algorithms, measurement techniques, applications and challenges,” *Journal of Sensor and Actuator Networks*, vol. 6, no. 4, p. 24, 2017.

[21] J. Xu, M. Ma, and C. L. Law, “Cooperative angle-of-arrival position localization,” *Measurement*, vol. 59, pp. 302–313, 2015.

[22] G. Mao, *Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking: Monitoring and Surveillance Techniques for Target Tracking.* IGI Global, 2009.

[23] L. Xiong, “A selective model to suppress NLOS signals in angle-of-arrival (AOA) location estimation,” in *Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on*, vol. 1, pp. 461–465, IEEE, 1998.

[24] G. Shen, R. Zetik, and R. S. Thoma, “Performance comparison of TOA and tdoa based location estimation algorithms in LOS environment,” in *Positioning, Navigation and Communication, 2008. WPNC 2008. 5th Workshop on*, pp. 71–78, IEEE, 2008.

[25] I. Guvenc and C.-C. Chong, “A survey on TOA based wireless localization and NLOS mitigation techniques,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, 2009.

[26] Y. Qi, H. Kobayashi, and H. Suda, “On time-of-arrival positioning in a multipath environment,” *IEEE Transactions on Vehicular Technology*, vol. 55, no. 5, pp. 1516–1526, 2006.

[27] A. Makki, A. Siddig, M. M. Saad, J. R. Cavallaro, and C. J. Bleakley, "Indoor localization using 802.11 time differences of arrival.," *IEEE Trans. Instrumentation and Measurement*, vol. 65, no. 3, pp. 614–623, 2016.

[28] F. Gustafsson and F. Gunnarsson, "Positioning using time-difference of arrival measurements.," in *ICASSP (6)*, pp. 553–556, Citeseer, 2003.

[29] N. S. Correal, S. Kyperountas, Q. Shi, and M. Welborn, "An UWB relative location system," in *Ultra Wideband Systems and Technologies, 2003 IEEE Conference on*, pp. 394–397, IEEE, 2003.

[30] B. B. Peterson, C. Kmiecik, R. Hartnett, P. M. Thompson, J. Mendoza, and H. Nguyen, "Spread spectrum indoor geolocation," *Navigation*, vol. 45, no. 2, pp. 97–102, 1998.

[31] R. Dobbins, S. Garcia, and B. Shaw, "Software defined radio localization using 802.11-style communications," *Worcester Polytechnic Institute*, 2011.

[32] C. Alippi and G. Vanini, "A RSSI-based and calibrated centralized localization technique for wireless sensor networks," in *Pervasive computing and communications workshops, 2006. PerCom workshops 2006. Fourth annual IEEE international conference on*, pp. 5–pp, IEEE, 2006.

[33] H. T. Friis, "A note on a simple transmission formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[34] X. Zhang and J. G. Andrews, "Downlink cellular network analysis with multi-slope path loss models," *IEEE Transactions on Communications*, vol. 63, no. 5, pp. 1881–1894, 2015.

[35] A. Bose and C. H. Foh, "A practical path loss model for indoor WiFi positioning enhancement," in *Information, Communications & Signal Processing, 2007 6th International Conference on*, pp. 1–5, IEEE, 2007.

[36] J. Miranda, R. Abrishambaf, T. Gomes, P. Gonçalves, J. Cabral, A. Tavares, and J. Monteiro, "Path loss exponent analysis in wireless sensor networks: Experimental evaluation," in *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on*, pp. 54–58, IEEE, 2013.

[37] G. Tzeremes and C. Christodoulou, "Use of Weibull distribution for describing outdoor multipath fading," in *Antennas and propagation society international symposium, 2002. IEEE*, vol. 1, pp. 232–235, IEEE, 2002.

[38] O. Georgiou, K. Mimis, D. Halls, W. H. Thompson, and D. Gibbins, "How many Wi-Fi APs does it take to light a lightbulb?," *IEEE Access*, vol. 4, pp. 3732–3746, 2016.

[39] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th annual international conference on Mobile computing and networking*, pp. 81–95, ACM, 2003.

[40] S. P. Singh and S. Sharma, "Range free localization techniques in wireless sensor networks: A review," *Procedia Computer Science*, vol. 57, pp. 7–16, 2015.

[41] A. Savvides, H. Park, and M. B. Srivastava, "The bits and flops of the n-hop multilateration primitive for node localization problems," in *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pp. 112–121, ACM, 2002.

[42] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 269–280, ACM, 2012.

[43] P. Krishnan, A. Krishnakumar, W.-H. Ju, C. Mallows, and S. Gamt, "A system for lease: Location estimation assisted by stationary emitters for indoor rf wireless networks," in *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 1001–1011, IEEE, 2004.

[44] Y. Xie, Y. Wang, A. Nallanathan, and L. Wang, "An improved k-Nearest-Neighbor indoor localization method based on Spearman distance.," *IEEE Signal Process. Lett.*, vol. 23, no. 3, pp. 351–355, 2016.

[45] K. Shi, Z. Ma, R. Zhang, W. Hu, and H. Chen, "Support vector regression based indoor location in IEEE 802.11 environments," *Mobile Information Systems*, vol. 2015, 2015.

[46] G. Mao, B. Fidan, and B. D. Anderson, "Wireless sensor network localization techniques," *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[47] M. L. Sichitiu and V. Ramadurai, "Localization of wireless sensor networks with a mobile beacon," in *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pp. 174–183, IEEE, 2004.

[48] N. B. Priyantha, H. Balakrishnan, E. D. Demaine, and S. Teller, "Mobile-assisted localization in wireless sensor networks," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 1, pp. 172–183, IEEE, 2005.

[49] Y. Zhou, J. Li, and L. Lamont, "Multilateration localization in the presence of anchor location uncertainties," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, pp. 309–314, IEEE, 2012.

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

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

[52] P. Biswas, T.-C. Lian, T.-C. Wang, and Y. Ye, "Semidefinite programming based algorithms for sensor network localization," *ACM Transactions on Sensor Networks (TOSN)*, vol. 2, no. 2, pp. 188–220, 2006.

[53] A. A. Kannan, G. Mao, and B. Vucetic, “Simulated annealing based localization in wireless sensor network,” in *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pp. 2–pp, IEEE, 2005.

[54] S. Hamdoun, A. Rachedi, and A. Benslimane, “Comparative analysis of RSSI-based indoor localization when using multiple antennas in wireless sensor networks,” in *Selected Topics in Mobile and Wireless Networking (MoWNeT), 2013 International Conference on*, pp. 146–151, IEEE, 2013.

[55] H. L. Van Trees, *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory.* John Wiley & Sons, 2004.

[56] S. Tilak, N. B. Abu-Ghazaleh, and W. Heinzelman, “A taxonomy of wireless micro-sensor network models,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 2, pp. 28–36, 2002.

[57] I. F. Akyildiz and M. C. Vuran, *Wireless sensor networks*, vol. 4. John Wiley & Sons, 2010.

[58] G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy, “Smart objects as building blocks for the Internet of Things,” *Internet Computing, IEEE*, vol. 14, no. 1, pp. 44–51, 2010.

[59] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” *Proceedings of the IEEE*, vol. 97, no. 2, pp. 427–450, 2009.

[60] B. Dil, S. Dulman, and P. Havinga, “Range-based localization in mobile sensor networks,” in *European Workshop on Wireless Sensor Networks*, pp. 164–179, Springer, 2006.

[61] D. Niculescu and B. Nath, “DV based positioning in ad hoc networks,” *Telecommunication Systems*, vol. 22, no. 1-4, pp. 267–280, 2003.

[62] H. Wu and R. Gao, “An improved method of DV-Hop localization algorithm,” *Journal of computational Information systems*, vol. 7, no. 7, pp. 2293–2298, 2011.

[63] X. Chen and B. Zhang, “Improved DV-Hop node localization algorithm in wireless sensor networks,” *International Journal of Distributed Sensor Networks*, vol. 2012, 2012.

[64] B. Xiao, L. Chen, Q. Xiao, and M. Li, “Reliable anchor-based sensor localization in irregular areas,” *Mobile Computing, IEEE Transactions on*, vol. 9, no. 1, pp. 60–72, 2010.

[65] X. Liu, S. Zhang, J. Wang, J. Cao, and B. Xiao, “Anchor supervised distance estimation in anisotropic wireless sensor networks,” in *Wireless Communications and Networking Conference (WCNC), 2011 IEEE*, pp. 938–943, IEEE, 2011.

[66] S. Lee, C. Park, M. J. Lee, and S. Kim, “Multihop range-free localization with approximate shortest path in anisotropic wireless sensor networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 80, 2014.

[67] Y. Wang, X. Wang, D. Wang, and D. P. Agrawal, “Range-free localization using expected hop progress in wireless sensor networks,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 20, no. 10, pp. 1540–1552, 2009.

[68] J. Coon, C. Dettmann, and O. Georgiou, “Full connectivity: corners, edges and faces,” *Journal of Statistical Physics*, pp. 1–21, 2012.

[69] D. Gao, P. Chen, C. H. Foh, and Y. Niu, “Hop-distance relationship analysis with quasi-UDG model for node localization in wireless sensor networks,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, pp. 1–11, 2011.

[70] G. Mao, Z. Zhang, and B. D. Anderson, “Probability of k-hop connection under random connection model,” *Communications Letters, IEEE*, vol. 14, no. 11, pp. 1023–1025, 2010.

[71] J. Philip, *The probability distribution of the distance between two random points in a box.* KTH mathematics, Royal Institute of Technology, 2007.

[72] Z. Zhang, G. Mao, and B. D. Anderson, “On the hop count statistics in wireless multihop networks subject to fading,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 7, pp. 1275–1287, 2012.

[73] C. Cartis, N. I. Gould, and P. L. Toint, “On the complexity of steepest descent, Newton’s and regularized Newton’s methods for nonconvex unconstrained optimization problems,” *SIAM Journal on Optimization*, vol. 20, no. 6, pp. 2833–2852, 2010.

[74] B. Korte, J. Vygen, B. Korte, and J. Vygen, *Combinatorial Optimization.* Springer, 2002.

[75] M. Stella, M. Russo, and D. Begusic, “RF localization in indoor environment,” *Radioengineering*, vol. 21, no. 2, pp. 557–567, 2012.

[76] J. Yang, H. Lee, and K. Moessner, “Multilateration localization based on singular value decomposition for 3D indoor positioning,” in *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*, pp. 1–8, IEEE, 2016.

[77] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, “Zee: zero-effort crowdsourcing for indoor localization,” in *Proceedings of the 18th annual international conference on Mobile computing and networking*, pp. 293–304, ACM, 2012.

[78] P. Kontkanen, P. Myllymaki, T. Roos, H. Tirri, K. Valtonen, and H. Wettig, “Topics in probabilistic location estimation in wireless networks,” in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 2, pp. 1052–1056, IEEE, 2004.

[79] C.-L. Wu, L.-C. Fu, and F.-L. Lian, “WLAN location determination in e-home via support vector classification,” in *Networking, sensing and control, international conference on*, vol. 2, pp. 1026–1031, IEEE, 2004.

[80] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, “On indoor position location with wireless LANs,” in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th International Symposium on*, vol. 2, pp. 720–724, IEEE, 2002.

[81] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *The Knowledge Engineering Review*, vol. 29, no. 03, pp. 345–374, 2014.

[82] K. Farkas, "Placement optimization of reference sensors for indoor tracking," *Acta Polytechnica Hungarica*, vol. 12, no. 2, pp. 123–139, 2015.

[83] O. Baala, Y. Zheng, and A. Caminada, "The impact of AP placement in WLAN-based indoor positioning system," in *Networks, 2009. ICN'09. Eighth International Conference on*, pp. 12–17, IEEE, 2009.

[84] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[85] Y. Sun, Á. Baricz, and S. Zhou, "On the monotonicity, log-concavity, and tight bounds of the generalized Marcum and Nuttall Q-functions," *IEEE Transactions on Information Theory*, vol. 56, no. 3, pp. 1166–1186, 2010.

[86] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.

[87] P. R. de Montmort, *Essai d'analyse sur les jeux de hazards. 2nd edition.* Paris: Quillau, 1714.

[88] J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni, "A survey on wireless indoor localization from the device perspective," *ACM Comput. Surv.*, vol. 49, pp. 25:1–25:31, June 2016.

[89] M. Ashe, M. de Monasterio, M. Gupta, and M. Pegors, "2010 US lighting market characterization," *Report to US Department of Energy*, 2012.

[90] M. Castro, A. J. Jara, and A. F. Skarmeta, "Smart lighting solutions for smart cities," in *Advanced information networking and applications workshops (WAINA), 2013 27th international conference on*, pp. 1374–1379, IEEE, 2013.

[91] N. Zhu, T. Diethe, M. Camplani, L. Tao, A. Burrows, N. Twomey, D. Kaleshi, M. Mirmehdi, P. Flach, and I. Craddock, "Bridging e-health and the Internet of Things: The sphere project," *IEEE Intelligent Systems*, vol. 30, no. 4, pp. 39–46, 2015.

[92] D. Pena, R. Feick, H. Hristov, and W. Grote, "Measurements and modeling of propagation losses in brick and concrete walls at the 900-Mhz band," *Transactions on Antennas and Propagation*, vol. 51, pp. 31–39, 2003.

[93] C. Bliek, P. Bonami, and A. Lodi, "Solving mixed-integer quadratic programming problems with ibmcplex: A progress report," in *Proceedings of the Twenty-Sixth RAMP Symposium*, pp. 171–180, 2014.

[94] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1801–1819, 2014.

[95] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Pptimization: algorithms and complexity.* Courier Corporation, 1982.

[96] K. Andreev and H. Räcke, "Balanced graph partitioning," in *Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, pp. 120–124, ACM, 2004.

[97] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey, "The node capacitated graph partitioning problem: A computational study," *Mathematical Programming*, vol. 81, no. 2, pp. 229–256, 1998.

[98] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[99] K. Chandrasekar and N. V. Ramana, "Performance comparison of GA DE PSO and SA approaches in enhancement of total transfer capability using facts devices," *Electrical Engineering and Technology*, vol. 7, no. 4, pp. 493–500, 2012.

[100] R. Wilson, "Propagation losses through common building materials 2.4 GHz vs 5 GHz," *E10589 Magis Networks, Inc*, pp. 1–27, 2002.

[101] B. Chomba, D. Konditi, D. Nyaanga, and J. Githeko, "Effects of varying distance on wireless signal propagation in indoor and outdoor built sites," *International Journal of Engineering Research in Africa*, vol. 6, pp. 75–89, 2011.

[102] J. J. Grefenstette, R. Gopal, B. J. Rosmaita, and D. VanGucht, "Genetic algorithms for the traveling salesman problem," in *Proceedings of the First International Conference on Genetic Algorithms and their Applications*, pp. 160–168, ACM, 1985.

[103] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, (Berkeley, CA, USA), pp. 2–2, USENIX Association, 2004.

[104] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using WiFi," in *ACM SIGCOMM Computer Communication Review*, vol. 45, pp. 269–282, ACM, 2015.

[105] L. Jian, Z. Yang, and Y. Liu, "Beyond triangle inequality: Sifting noisy and outlier distance measurements for localization," in *2010 Proceedings IEEE INFOCOM*, pp. 1–9, March 2010.

[106] H. Abukhalaf, J. Wang, and S. Zhang, "Outlier detection techniques for localization in wireless sensor networks: A survey," *International Journal of Future Generation Communication and Networking*, vol. 8, no. 6, pp. 99–114, 2015.

[107] Q. Xiao, K. Bu, Z. Wang, and B. Xiao, "Robust localization against outliers in wireless sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 9, no. 2, p. 24, 2013.

[108] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2409–2419, 2018.

[109] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, "Fusion of WiFi, smartphone sensors and landmarks using the kalman filter for indoor localization," *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[110] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1327–1346, 2016.

[111] S. Wielandt and L. D. Strycker, "Indoor multipath assisted angle of arrival localization," *Sensors*, vol. 17, no. 11, p. 2522, 2017.

[112] T. Van Haute, B. Verbeke, E. De Poorter, and I. Moerman, "Optimizing time-of-arrival localization solutions for challenging industrial environments," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1430–1439, 2017.

[113] Z. Han, C. S. Leung, H. C. So, and A. G. Constantinides, "Augmented lagrange programming neural network for localization using time-difference-of-arrival measurements," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 8, pp. 3879–3884, 2018.

[114] S. Čapkun, M. Hamdi, and J.-P. Hubaux, "GPS-free positioning in mobile ad hoc networks," *Cluster Computing*, vol. 5, no. 2, pp. 157–167, 2002.

[115] C. Y. Tazibt, M. Bekhti, T. Djamah, N. Achir, and K. Boussetta, "Wireless sensor network clustering for UAV-based data gathering," in *Wireless Days, 2017*, pp. 245–247, IEEE, 2017.

[116] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. K. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2220–2243, 2016.

[117] L. Jayatilleke and N. Zhang, "Landmark-based localization for unmanned aerial vehicles," in *Systems Conference (SysCon), 2013 IEEE International*, pp. 448–451, IEEE, 2013.

[118] J. D. Barton, "Fundamentals of small unmanned aircraft flight," *Johns Hopkins APL technical digest*, vol. 31, no. 2, pp. 132–149, 2012.

[119] E. Menegatti, A. Zanella, S. Zilli, F. Zorzi, and E. Pagello, "Range-only slam with a mobile robot and a wireless sensor networks," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pp. 8–14, IEEE, 2009.

[120] F. Caballero, L. Merino, P. Gil, I. Maza, and A. Ollero, "A probabilistic framework for entire wsn localization using a mobile robot," *Robotics and Autonomous Systems*, vol. 56, no. 10, pp. 798–806, 2008.

[121] G.-L. Sun and W. Guo, "Comparison of distributed localization algorithms for sensor network with a mobile beacon," in *Networking, Sensing and Control, 2004 IEEE International Conference on*, vol. 1, pp. 536–540, IEEE, 2004.

[122] A. Galstyan, B. Krishnamachari, K. Lerman, and S. Pattem, "Distributed online localization in sensor networks using a moving target," in *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, pp. 61–70, IEEE, 2004.

[123] K.-F. Ssu, C.-H. Ou, and H. C. Jiau, "Localization with mobile anchor points in wireless sensor networks," *IEEE transactions on Vehicular Technology*, vol. 54, no. 3, pp. 1187–1197, 2005.

[124] G. Teng, K. Zheng, and W. Dong, "Adapting mobile beacon-assisted localization in wireless sensor networks," *Sensors*, vol. 9, no. 4, pp. 2760–2779, 2009.

[125] X. Yang, Z. Gao, and Q. Niu, "Unmanned aerial vehicle–assisted node localization for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, p. 1550147717749818, 2017.

[126] L. A. Villas, D. L. Guidoni, and J. Ueyama, "3D localization in wireless sensor networks using unmanned aerial vehicle," in *Network Computing and Applications (NCA), 2013 12th IEEE International Symposium on*, pp. 135–142, IEEE, 2013.

[127] T. Rappaport, *Wireless communications: Principles and practice.* Prentice Hall Communications Engineering and Emerging Technologies Series, Prentice Hall, 2nd ed., 2002. Includes bibliographical references and index.

[128] A. MOSEK, "The mosek python optimizer API manual version 7.1 (revision 62)," 2017.

[129] F. G. Costa, J. Ueyama, T. Braun, G. Pessin, F. S. Osório, and P. A. Vargas, "The use of unmanned aerial vehicles and wireless sensor network in agricultural applications," in *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pp. 5045–5048, IEEE, 2012.

[130] J. Polo, G. Hornero, C. Duijneveld, A. García, and O. Casas, "Design of a low-cost wireless sensor network with UAV mobile node for agricultural applications," *Computers and electronics in agriculture*, vol. 119, pp. 19–32, 2015.

[131] G. Tuna, T. V. Mumcu, K. Gulez, V. C. Gungor, and H. Erturk, "Unmanned aerial vehicle-aided wireless sensor network deployment system for post-disaster monitoring," in *International Conference on Intelligent Computing*, pp. 298–305, Springer, 2012.

[132] I. Jawhar, N. Mohamed, and J. Al-Jaroodi, "UAV-based data communication in wireless sensor networks: Models and strategies," in *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on*, pp. 687–694, IEEE, 2015.

[133] M. Dong, K. Ota, M. Lin, Z. Tang, S. Du, and H. Zhu, "UAV-assisted data gathering in wireless sensor networks," *The Journal of Supercomputing*, vol. 70, no. 3, pp. 1142–1155, 2014.

[134] S. Rashed and M. Soyturk, "Analyzing the effects of UAV mobility patterns on data collection in wireless sensor networks," *Sensors*, vol. 17, no. 2, p. 413, 2017.

[135] H. Okcu and M. Soyturk, “Distributed clustering approach for UAV integrated wireless sensor networks,” *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1-3, pp. 106–120, 2014.