2008

Two Dimensional Localization of Passive UHF RFID Tags

Bhavik Contractor
Wright State University

Follow this and additional works at: http://corescholar.libraries.wright.edu/etd_all
Part of the Computer Engineering Commons

Repository Citation

This Thesis is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact corescholar@www.libraries.wright.edu.
Two Dimensional Localization of Passive UHF RFID Tags

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Computer Engineering

By

BHAVIK VENILAL CONTRACTOR
B.E., Mumbai University, India, 2003

2008
Wright State University
Dayton, Ohio 45435-0001
I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Bhavik Venilal Contractor ENTITLED Two Dimensional Localization of Passive UHF RFID Tags BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Computer Engineering.

Jack S.N. Jean, Ph. D.
Thesis Director

Thomas A. Sudkamp, Ph. D.
Department Chair

Committee on Final Examination

Jack S.N. Jean, Ph. D.

Bin Wang, Ph. D.

Krishnaprasad Thirunarayan, Ph. D.

Meilin Liu, Ph. D.

Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies
ABSTRACT


The advent of GPS has redefined the need of a positioning system in today’s ubiquitous computing world. While GPS works satisfactorily and is quite a norm in an outdoor environment, it fails to work indoors due to the inherent complexity of an indoor environment. There is an ever increasing need to develop an indoor positioning system and a lot of research has been done to solve the problem of indoor localization. These solutions differ on the basis of cost, dependency on environment, line of sight requirements and so on.

Passive RFID (Radio Frequency Identification) tags pose an interesting solution to the problem of indoor localization, given the ease of deployment and the cost effective infrastructure. It is less expensive to tag items with RFID tags, than to attach them with sensor nodes. In this thesis, the problem of using the RFID technology for two dimensional indoor localization is studied. A relatively inexpensive technique requiring just one RFID reader and multiple passive RFID tags is adopted. The idea is to use multilateration among the passive tags to solve the problem of localization. A ranging technique is developed to establish a relationship between signal strength and distance. The concept of Received Signal Strength calibration is used to develop the ranging technique, and to account for the effects of dynamic environmental conditions on localization. Finally, an error map matching technique is employed to counter the errors in localization.
Contents

1 Introduction 2
1.1 RFID 3
1.1.1 RFID Components 3
1.1.1.1 RFID Reader 3
1.1.1.2 RFID Tag 3
1.1.2 RFID in Today’s World 4
1.2 Need for an Indoor Positioning System 5
1.2.1 Related Work 6
1.2.1.1 RFID-Radar 6
1.2.1.2 Cricket 6
1.2.1.3 RADAR 7
1.2.1.4 Landmarc 7
1.3 Infrastructure Used 8
1.3.1 Alien ALR-9800 Reader 8
1.3.2 Alien ALR-9610-BC Reader Antennas 10
1.3.3 Alien Gen2 Squiggle Tags 11
1.4 Software Tool 11
1.5 Thesis Organization 13

2 Ranging 14
2.1 Ranging Procedure 14
2.2 Getting the RSS Information 15
2.2.1 RF Attenuation .................................................. 15
2.2.2 Using the Software to obtain RSS information .......... 15
2.3 Radio Propagation Model ......................................... 16
2.3.1 Log-distance Path Loss Model ............................... 18
2.4 RSS Calibration .................................................... 20
2.4.1 Curve Fitting ................................................... 20
2.4.2 Calculations Involved ......................................... 21

3 Localization Process .................................................. 25
3.1 Localization Techniques ............................................ 25
3.1.1 Triangulation ................................................... 25
3.1.1.1 Lateration .................................................. 25
3.1.1.2 Angulation .................................................. 26
3.1.2 Scene Analysis .................................................. 27
3.1.3 Proximity ....................................................... 27
3.2 Multilateration ...................................................... 28
3.3 Determining the Location of an Unknown Tag ............... 29
3.4 Initial Validation Results .......................................... 33

4 Map Matching Technique and Software Tools ..................... 36
4.1 Map Matching Technique .......................................... 36
4.2 Software Tools .................................................... 41
4.2.1 Modes of Operation ............................................ 41
4.2.1.1 RSS Calibration Mode ................................. 41
4.2.1.2 Localization Mode ....................................... 41
4.2.2 Windows Program ............................................. 42
4.2.2.1 Structural Description ................................. 42
4.2.2.2 Functional Description ................................. 44
4.2.3 Matlab Program ............................................... 46
4.2.3.1 RSS Calibration Mode .................................. 46
List of Figures

1.1 Backscatter Modulation [Alien Technology 2004] .............................. 4
1.2 Alien ALR-9800 Reader .................................................. 8
1.3 Alien ALR-9610-BC Antenna .............................................. 10
1.4 Radiation Pattern of a Circular Antenna .................................... 11
1.5 Alien Gen2 Squiggle Tag .................................................. 11
1.6 GUI used in the Experiment ................................................ 12

2.1 Experimental Setup used in the Thesis ...................................... 17
2.2 RSS plot at the two antennas ................................................ 23
2.3 Plot of actual vs. calculated distances of the landmarks from the two antennas 24

3.1 Trilateration example in a 2D space ........................................ 26
3.2 Angulation example in a 2D space, using angles relative to a 0° reference vector [Karl and Willig 2005] .................................. 27
3.3 Locating the unknown Tag using multilateration ......................... 29
3.4 Geometry involved in Tag Landmark distance measurement .......... 30
3.5 A Map of the nine landmark Tags ........................................ 33
3.6 Plot of actual vs. calculated co-ordinates of the landmarks .......... 35

4.1 Plot of error map at calculated locations of 9 landmark Tags ........... 38
4.2 Plot of error map at calculated locations of 13 landmark Tags .......... 40
4.3 Class View ................................................................. 43

5.1 Plot of calculated vs. actual x co-ordinates without Map Matching .... 50
5.2 Plot of calculated vs. actual x co-ordinates with Map Matching . . . . . . 51
5.3 Plot of calculated vs. actual y co-ordinates without Map Matching . . . . 52
5.4 Plot of calculated vs. actual y co-ordinates with Map Matching . . . . . . 53
5.5 3-D error plot without Map Matching . . . . . . . . . . . . . . . . . . . 54
5.6 3-D error plot with Map Matching . . . . . . . . . . . . . . . . . . . . . 55
5.7 CCDF plot of the error for the two cases . . . . . . . . . . . . . . . . . . 56

A.1 FlowChart for the RSS collection procedure . . . . . . . . . . . . . . . . 64
List of Tables

2.1 Path Loss Exponent and Std deviation for different environments [Rappaport 2001] .......... 19
2.2 Calculated RSS parameters ................................................. 22
5.1 Error Comparison for the two techniques .................................. 57
I wish to thank and acknowledge those people who contributed to this thesis.

Dr. Jack Jean, my advisor and mentor, has been an inalienable part of this project. Not only did he provide me with valuable insights in every step of the process, but also the strength and self belief when the going got tough. Looking back, working under his capability was the best learning experience I could have asked for.

Additionally, I would take this opportunity to thank the members of my thesis committee, Dr. Krishnaprasad Thirunarayan, Dr. Bin Wang and Dr. Meilin Liu. I am also grateful to Dr. Robert Helt for being a constant source of encouragement.

This thesis would be incomplete without a special mention of my parents, Usha and Venilal Contractor. Their hard work makes me the person I am today, and this thesis is dedicated to them. My sisters, Sonal Patel and Pooja Contractor, are a constant source of joy in my life, I wish to thank them for all their support.

A special thanks to my friends, Pratik Desai, Jasmine Soni, Shirish Ingawale, Vivek Karekar, Binu Eapen, and Sushant Taksande, who found time from their busy schedules to help me out during the final phase of the thesis.

Finally, I would like to thank almighty for the courage and endurance to make this possible.
1

Introduction

In this thesis, the problem of locating passive RFID tags inside a room was studied. The key features of the technique are:

- The relatively inexpensive setup, where only one RFID reader and multiple passive tags are used, and
- The ability to dynamically adapt to changes in environment.

The technique utilizes the location information of priorly placed reference tags. The location information of these reference tags is known and hence are referred to as landmark tags. The landmark tags serve a three-fold purpose: to help calibrate the RSS (Received Signal Strength) parameters of a radio propagation model in order to account for the dynamic environment, to serve as reference tags for the lateration procedure used to determine the location of an unknown tag, and finally to build an error map, which can be used to refine the localization results.

Section 1 explains the basics of a RFID system and its importance in today’s ubiquitous computing world. Section 2 discusses the need of an indoor positioning system and provides a brief survey of the available indoor localization techniques. Section 3 mentions the hardware used in the experiments. Section 4 describes a software tool developed for the experiments. Section 5 provides an outline of the thesis.
1.1 RFID

RFID stands for Radio Frequency Identification. It is a technology, which uses RF (Radio Frequency) communication between a RFID reader and a RFID tag to identify the tag.

1.1.1 RFID Components

An RFID system basically consists of two components.

1.1.1.1 RFID Reader

An RFID reader consists of a transmitter, a receiver, and a microprocessor. The reader unit also contains an antenna as a part of the entire system, which may be embedded inside the reader or externally connected to the reader. The reader antenna transmits the RF wave generated inside the reader to interrogate for tags, and also receives the responses from the tags. Some readers use a single antenna while other readers use a pair of antennas, where one antenna can act as a transmitter and the other as a receiver. However, typically only one antenna is active at a time.

1.1.1.2 RFID Tag

An RFID tag generally consists of a small antenna and a micro chip. The micro chip stores some information regarding the tag, usually an ID. The micro chip also performs the function of modulating the RF signal. The antenna is used to receive and transmit the RF signal. There are two kinds of RFID tags:

- Passive Tag: A passive tag does not have power of its own and it uses the RF signal from the reader antenna to power the internal circuitry. The communication with the reader is achieved by a process called backscatter modulation. Backscatter modulation is the process in which the tag antenna modulates the RF signal received from the reader in patterns of ones and zeros, which may represent the tag ID. The modulated signal is transmitted back to the reader. Figure 1.1 shows the backscatter modulation process.
1.1. RFID

1.1.2 RFID in Today’s World

RFID is similar in some ways to bar code technology in that the tags contain ID and other data readable by electronic equipment [Alien Technology 2004]. However RFID tags have many advantages over bar codes, listed as follows:

- RFID tags have a long read range when compared with bar codes, and they do not require line of sight.
- They are more robust to damage done due to improper handling, and can work in harsh environment conditions.
- The data in a tag can be reprogrammed and the tag can be reused. They can also be provided with enhanced security features, which can prevent duplication of tags and
1.2. NEED FOR AN INDOOR POSITIONING SYSTEM

The process of tracking moving inventory can be completely automated using RFID, which can otherwise be a very laborious task.

Due to the above mentioned and many other advantages of RFID, the technology is finding more employment in various industries. Currently the technology is widely used in various applications such as asset tracking, automatic toll collection, theft protection, health care systems, livestock identification, etc.

1.2 Need for an Indoor Positioning System

The GPS (Global Positioning System) has led to a sort of revolution ever since its deployment. After the downing of Korean Flight 007 in 1983 -a tragedy that might have been prevented if its crew had access to better navigation tools- President Ronald Reagan issued a directive that guaranteed that GPS signals would be available at no charge to the world [Smithsonian ]. Since then GPS has been extensively used in a host of applications like tracking, navigation, land surveying, etc. Today, there are mobile phones, cameras, and wristwatches available which are equipped with GPS technology. This trend shows that there has been a steady increase on the reliance on a positioning system.

However, GPS works poorly when operated indoors and there is a need for a better indoor positioning system. It would be desirable for a fireman to track his position in a building during a rescue operation. Searching for a particular book in a library is often a laborious work, even when the books are placed in a serial order, and especially when the book gets misplaced and is no longer at the correct serial number position. Another troublesome task is that of finding one’s luggage at the airport. An indoor positioning system can be employed to solve the above mentioned problems and tagging objects with passive RFID tags does not seem to be a very expensive approach.
1.2. NEED FOR AN INDOOR POSITIONING SYSTEM

1.2.1 Related Work

A lot of research has been done to solve the problem of indoor localization. A few are listed here.

1.2.1.1 RFID-Radar

Trolley Scan (Pty) Ltd developed a new technology called RFID-Radar, to locate RFID tags with a high level of accuracy. RFID-Radar combines the distance information of a tag from a reader along with the angle of arrival information to locate the tag. The system measures the path length for signals travelling from the transponder to the reader to determine the range. The system also has to perform a calibration process using a single transponder, to record the delays through the antennas, and the time to travel through the antennas and the cable. The angle is measured by comparing the signals arriving at two identical receivers with closely spaced antennas. The system has been able to achieve a distance accuracy of 0.5 meters and a direction accuracy of within 1 degree. They also have a larger range. [Trolley Scan (Pty) Ltd ; Collins 2008].

1.2.1.2 Cricket

Cricket uses lateration technique to determine the location of a listener device, whose position is to be estimated [Priyantha 2005; Hightower and Borriello 2001]. Active beacons are mounted on walls and/or ceilings, and are capable of transmitting an RF signal along with an ultrasound signal. The RF signals are embedded with the location information of the beacon. The listening device first detects the RF signal from a beacon and then waits for the ultrasound signal. Since RF signals travel faster than ultrasound signals, the listener can calculate the time difference of arrival and determine its distances from multiple beacons. The distance information is then fed to a host computer which can compute the location of the listener using lateration. The Cricket system was able to achieve a position accuracy of 10 cm [Priyantha 2005]. The Bat Ultrasonic Location System works on a similar principle [Harter et al. 1999].
1.2. NEED FOR AN INDOOR POSITIONING SYSTEM

1.2.1.3 RADAR

Paramvir Bahl and Venkata Padmanabhan at Microsoft developed the RADAR localization system, for locating and tracking users inside a building. The system is based on the IEEE 802.11 wireless network technology [Hightower and Borriello 2001], and it basically uses a scene analysis technique. The RSS (Received Signal Strength) readings of a mobile device are computed at three base stations. The RSS readings are then compared with a previously generated RSS map composed of RSS samples at different locations. A close match is found for each base station and the principle of triangulation is applied to determine the location of the mobile device. Nearest neighbor technique is used to determine the closest match.

The system explores two different techniques to obtain the RSS map. In the first approach, RSS samples are calculated at different locations using empirical measurements and stored in a database. In the second approach, the RSS map is created by developing a Radio propagation model. While the second approach is less accurate, it has acceptable results and is more cost effective than the first approach, since it does not require detailed empirical measurements to generate a signal strength map. The technique was able to obtain a median resolution of 4.3 meters using the Radio propagation model and a resolution of 2.94 meters using the empirical model [Bahl and Padmanabhan 2000].

1.2.1.4 Landmarc

Landmarc is an RFID based localization technique which uses active reference RFID tags to determine the location of an unknown active tag. An approach similar to that of RADAR is used to localize the tag, wherein the RSS readings of an unknown tag at four or more readers are compared with the previously obtained RSS samples of reference tags, and a closest neighbor technique is used to determine the location of the unknown tag. Landmarc was able to achieve the localization with a distance error of one meter on average using one reference tag per square meter. The system requires at least three readers and many reference active tags, which makes it an expensive setup [Ni et al. 2003].
1.3  Infrastructure Used

The experiments in this thesis are performed using the RFID development kit manufactured by Alien Technology. This section describes the key components in detail.

1.3.1  Alien ALR-9800 Reader

The Alien ALR-9800 works in a frequency band of 915 MHz, also known as UHF (Ultra High Frequency) band. The reader is the heart of the RFID system. The reader can read any EPC Class 1 Generation 1 or 2 tag and communicate the information to a host computer [Alien Technology 2007a]. The host computer is connected to the reader via a RS-232 cable. The reader consists of a CPU which executes all the commands issued by the host computer. Figure 1.2 shows the picture of an Alien Reader.

![Alien ALR-9800 Reader](image)
The reader also consists of an antenna panel which can be used to connect up to four antennas. The antennas are connected to the reader using coaxial cables. The antennas work in a pair so at least two antennas are required to constitute a working RFID system. Antenna 0 is paired with antenna 1, while antenna 2 is paired with antenna 3. When antenna 0 is the transmitter, antenna 1 acts as the receiver and vice versa. Similar is the case with antenna 2 and antenna 3. The section below describes the antennas used.
1.3. INFRASTRUCTURE USED

1.3.2 Alien ALR-9610-BC Reader Antennas

Two circular polarized antennas (ALR-9610-BC) as shown in Figure 1.3 are used in this thesis.

Figure 1.3: Alien ALR-9610-BC Antenna

They use circular polarization to distribute the radio waves in a radially symmetrical pattern. This makes the detection process somewhat independent of the tag orientation [Alien Technology 2004; Texas Instruments 2005]. However the downside is that the output power is reduced when compared to linear antennas and consequently the read range is also reduced. Since the experiments in this thesis were conducted in a small grid, the choice of circular antennas was deemed to be an appropriate one.

To ensure that the radiations from the reader antennas is uniformly distributed over the interrogation area, the antennas are mounted at a angle. The angle is determined by manually rotating the antennas about the vertical axis, until an orientation is found which leads to the detection of all the landmark tags.
1.3.3 Alien Gen2 Squiggle Tags

Alien technology’s EPC compliant Gen2 Squiggle tags are used for our experiments. Figure 1.5 shows the cross section of a Gen2 tag. These are passive tags and do not have any power of their own. Tags are powered by the RF waves transmitted by the reader antenna and communication with the reader is achieved by the process of backscatter modulation. Since it uses the reader RF energy, a passive RFID tag cannot work outside a reader’s detection range. The advantage with passive tags is that they are very inexpensive when compared to active tags and have a longer life time. Although circular antennas are said to be less dependent on tag orientation, it was found via experiments, that horizontal tag placement yielded us the best results, in terms of consistency with the ranging procedure (chapter 2) used in this thesis. The observation is in line with the recommendations made by Texas Instruments in their manual [Texas Instruments 2005].

1.4 Software Tool

This section gives an overview of the software tool developed in this thesis. A detailed description of the software tool and its working is provided in Chapter 4. The software tool was developed using Microsoft’s .Net development framework, and is used in combination
with Matlab. The .Net program is used for the data collection process, while Matlab is used for the data processing part. Figure 1.6 shows the GUI (Graphical User Interface) window.

The tool can be operated in two modes.

- RSS Calibration Mode: This mode is used to calibrate the RSS parameters for the ranging technique described in Chapter 2. In this mode, RSS data is collected from the landmark tags and fed into a text file. Once the data collection process is done, a Matlab code is called by the .NET program to calculate the RSS parameters. The Matlab program also builds an error map, which is used to refine the localization error.

- Localization Mode: This mode is used to perform the actual localization of an unknown tag. In this mode, RSS values for an unknown tag are obtained and fed to a Matlab program. The Matlab program computes the x and y co-ordinates of the unknown tag and displays it on the screen.

![Figure 1.6: GUI used in the Experiment](image)
1.5 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 describes the ranging procedure used to obtain the relationship between distance and RSS. Chapter 3 explains the localization technique used in the thesis to localize an unknown tag. Chapter 4 describes the map matching technique used to refine the results obtained from the localization technique. Chapter 4 also gives a detailed description of the software tools developed. Chapter 5 summarizes the results obtained. Chapter 6 concludes the thesis along with directions for future work.
2

Ranging

This chapter discusses the ranging technique employed in this thesis to calculate the distance of an unknown tag from an antenna. Section 1 gives a brief introduction of the basic types of ranging techniques used. Section 2 details the procedure used in the thesis to get the RSS information of an unknown tag. In Section 3, the Radio propagation model used in the thesis is explained. Section 4 talks about the RSS calibration procedure developed to model the propagation of radio waves.

2.1 Ranging Procedure

The most important aspect of this thesis is to develop a method to determine the physical distances between a tag and the reader antennas. This process is called ranging. There are many ranging techniques used to calculate the physical distance between two sensor nodes. However, the two basic techniques used are Received Signal Strength (RSS) and Signal Propagation Time [Lynch ].

- Signal Propagation Time: One way to obtain the physical distance is to measure the propagation time of a signal. However, radio waves, like all forms of electromagnetic waves, travel at the speed of light and it is difficult to measure the signal propagation time.

- Received Signal Strength: RSS (Received Signal Strength) is one of the simplest methods to obtain the distances between a transmitter (reader antennas, in our case) and a
2.2. GETTING THE RSS INFORMATION

receiver (tags). The method is relatively inexpensive and does not require line of sight. However, RSS readings are prone to many errors due to environmental factors such as shadowing, multi-path, orientation of tags, movement around the devices, etc.

2.2 Getting the RSS Information

In our application, the RSS information is not available directly. Instead the RF attenuation feature of the Alien reader acts as the RSS information for the tags.

2.2.1 RF Attenuation

The Alien reader ALR-9800, uses software control to provide power attenuation to the signals transmitted by the antennas [Alien Technology 2007b]. According to [Alien Technology 2006], “Attenuation is similar to a kink in a water hose - the more the attenuation, the more restrictive the flow. This is a negative figure - the higher the number’s magnitude, the lower the power.”

2.2.2 Using the Software to obtain RSS information

The Alien reader can set the RF attenuation factor from 0 to 150 in increments of 10 with each increment representing one dB reduction in power. Attenuation value of 0 indicates no attenuation i.e., maximum power transmitted by the antenna and consequently it implies that even the tags at the farthest distance and within the antenna read range can be detected. Attenuation value of 150 (-15dB) indicates maximum attenuation i.e., minimum power, and hence ideally only the tags which are closer to the antenna should respond. The software tool developed in this thesis, uses an API (Application Programming Interface) provided by Alien Technology to iteratively increase the value of RF attenuation by 10 starting from 0. During each iteration, the surrounding is interrogated for tags and if a tag fails to respond to an interrogation, the RF attenuation value of the previous iteration is converted to a value in dB (i.e. divided by 10) and recorded as the RSS for that corresponding tag; whereas, if a tag is detected it is simply ignored. If the RF attenuation value reaches 150, and the reader
is still able to detect a tag then the RSS for that particular tag is recorded as 15. Thus, the tags which are read with a lower value of RSS will ideally be farther away from the antenna than the tags which are read with a higher value of RSS. The above process is performed twice. When antenna 1 is set as a transmitter antenna 0 is set as a receiver and vice versa. Thus each tag will have two sets of RSS information.

2.3 Radio Propagation Model

According to [WikipediaA ] “ A radio propagation model, also known as the Radio Wave Propagation Model or the Radio Frequency Propagation Model, is an empirical mathematical formulation for the characterization of radio wave propagation as a function of frequency, distance and other conditions. A single model is usually developed to predict the behavior of propagation for all similar links under similar constraints. Created with the goal of formalizing the way radio waves are propagated from one place to another, such models typically predict the path loss along a link or the effective coverage area of a transmitter.”

The experiments for the theses were conducted in a closed laboratory with lot of computers and electrical equipments around. All tags are horizontally attached to paper boxes which stand on the experimental table roughly facing the antennas. Figure 2.1 shows the experimental setup.
2.3. RADIO PROPAGATION MODEL

Figure 2.1: Experimental Setup used in the Thesis
One biggest constraint in using RSS as a ranging method is that it is very difficult to predict the propagation of RF waves inside a building, particularly given the kind of settings in which the experiments in the theses were performed and hence the need of a Radio Wave Propagation model.

The most basic model of radio wave propagation involves “free space” radio wave propagation. However the “free space” propagation model is hardly followed in real world [Roberts ; Rappaport 1996]. The RF propagation indoors, is affected by many factors such as reflection, diffraction and scattering. Thus the RF signal at a given point can be a combination of signals reflected from various sources (multipath scattering), as well as scattered and diffracted signal components [Roberts ; Rappaport 1996]. The propagation models basically try to model the large scale fades and small scale fades. Large scale (fading) models predict the behavior of radio waves averaged over distances, and are useful in estimating the radio coverage area of a transmitter. Small scale (fading) models describe the rapid change in signal strength over a small distance (a few wavelengths) or time interval [Rappaport 2001; Girod 1999].

### 2.3.1 Log-distance Path Loss Model

In an indoor environment, the Log-distance Path Loss Model, Equation (2.2), has been used by researchers to model the path loss [Rappaport 2001]. The model is composed of two parts.

- The first part is the Path Loss model represented by Equation (2.1), which models the rate of decay of power with respect to distance using the path loss exponent parameter ($n$).

\[
PL = Pl_0 + 10n \log_{10}(d/d_0) \tag{2.1}
\]

where,

- $PL$ (dB) = total path loss at a distance $d$.
- $d_0$ = a reference distance, usually 1m.
2.3. RADIO PROPAGATION MODEL

\( P_l_0 \) (dB) = path loss at distance \( d_0 \).

\( n \) = path loss exponent.

- The second part is \( X_g \), which is a random Gaussian variable with a standard deviation of \( \sigma \), modeling the randomness in the observed power loss at the same distance (between the transmitter and the receiver). This phenomenon is called log-normal shadowing.

Combining the two parts, the complete equation for log distance path loss model is as follows.

\[
P_l = P_l_0 + 10n \log_{10}(d/d_0) + X_g
\]

(2.2)

For our application, \( P_l = -\text{RSS in dB} \).

\( d \) = distances of a tag from an antenna in meters.

Table 2.1 shows the typical values of \( n \) and \( \sigma \) obtained using empirical measurements for different environments.

<table>
<thead>
<tr>
<th>Environment</th>
<th>Frequency(Mhz)</th>
<th>( n )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retail Store</td>
<td>914</td>
<td>2.2</td>
<td>8.7</td>
</tr>
<tr>
<td>Grocery Store</td>
<td>914</td>
<td>1.8</td>
<td>5.2</td>
</tr>
<tr>
<td>Office, Soft Partition</td>
<td>900</td>
<td>2.4</td>
<td>9.6</td>
</tr>
<tr>
<td>Office, Hard Partition</td>
<td>1500</td>
<td>3.0</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Table 2.1: Path Loss Exponent and Std deviation for different environments [Rappaport 2001].

However, as discussed earlier, RF propagation is different for different environments. In fact even in the same room, the RF propagation parameters (\( P_l_0 \) and \( n \)) may change over time. It was observed that using the same parameters for both the antennas did not yield a good estimate of the distances. Hence there was a need to calibrate the parameters \( P_l_0 \) and \( n \) to account for the different antennas and the changes in environment. This is called RSS calibration.
2.4 RSS Calibration

Jihoon Kang et al [Kang et al. 2007], proposed a RSS self-calibration protocol for their WSN (Wireless Sensor Network) localization to account for errors in RSS readings, caused by factors such as multi-path and dynamic environment. The same approach is used to model the relationship between RSS and distance in our application.

2.4.1 Curve Fitting

The localization technique employed by us uses nine landmarks whose distances are already known. \( P_l (-\text{RSS}) \) readings at these locations are collected and plotted on the Y axis and the corresponding distances on the X axis, and regression analysis technique is employed to get a curve fit.

A method of least squares is used to derive the calibration parameters. According to [WikipediaB], “The method of least squares, also known as regression analysis, is used to model numerical data obtained from observations by adjusting the parameters of a model to get an optimal fit of the data. The best fit is that instance of the model for which the sum of squared residuals has its least value, a residual being the difference between an observed value and the value given by the model.”

Equation (2.1) is used in the calibration process. Since \( X_g \) is a random variable with a zero mean, it was ignored and instead average of ten RSS samples was taken for each landmark tag to account for the randomness.

Equation (2.1), when \( d_0 = 1 \text{m} \), is further simplified as follows

\[
P_l = P_{l0} + ND
\]

where,

\[
N = 10n.
\]

\[
D = \log_{10}(d).
\]
Equation (2.3), now takes the form of a simple straight line equation

\[ y = mx + c \] (2.4)

So, for Figure 2.2, if we use the least squares technique and find a best fit for the graph then the y-axis intercept will correspond to the parameter \( P_l_0 \) and the slope of the line will be the parameter \( N \).

### 2.4.2 Calculations Involved

The line should then be fitted to the plotted points in Figure 2.2 in such a way that the sum of squares of the residuals (difference between the plotted points and the line) is minimized. The steps below shows the calculations involved in deriving the parameters of the line. We will use Equation (2.4) to simplify the calculations. The sum of squares of residuals is expressed as

\[ \varepsilon = \sum_{i=1}^{n} [y_i - (c + mx_i)]^2 \] (2.5)

The minimum of the sum of squares is found by setting the gradient to zero. Since there are two parameters \( c \) and \( m \), there will be two gradient equations

\[ \frac{\partial \varepsilon}{\partial c} = -2 \sum [y_i - (c + mx_i)] = 0 \] (2.6)

\[ \frac{\partial \varepsilon}{\partial m} = -2 \sum x_i[y_i - (c + mx_i)] = 0 \] (2.7)

where \( \sum_{i=1}^{n} \) is replaced by \( \sum \)
2.4. RSS CALIBRATION

Solving Equations (2.6) and (2.7), we get

\[
\begin{align*}
c &= \frac{\left(\sum y\right)\left(\sum x^2\right) - \left(\sum x\right)\left(\sum xy\right)}{n\left(\sum x^2\right) - \left(\sum x\right)^2} \quad (2.8) \\
m &= \frac{n\left(\sum xy\right) - \left(\sum x\right)\left(\sum y\right)}{n\left(\sum x^2\right) - \left(\sum x\right)^2} \\
\end{align*}
\]

If \(x_i\) corresponds to the distances of the nine landmark tags \((d)\) and \(y_i\) corresponds to \(P_l\) then \(c\) corresponds to \(P_{l0}\) and \(m\) corresponds to \(N\).

The empirical values of \(P_{l0}\) and \(N\) obtained from the above calculations leads to a nice curve fit as shown in Figure 2.2.

Next, to verify the accuracy of the calibration, the distances of the landmarks from the two antennas were calculated using the derived equations. The calculated distances compares fairly well with the actual distances as shown in Figure 2.3.

The derived equations are then used in determining the distances of the unknown tags from the two antennas and subsequently localize the tags. Table 2.2 shows the RSS paramters obtained by the above calculations.

<table>
<thead>
<tr>
<th>Antenna</th>
<th>(P_{l0}) (dB)</th>
<th>(N) (10n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antenna 0</td>
<td>-17.43</td>
<td>21.85</td>
</tr>
<tr>
<td>Antenna 1</td>
<td>-17.32</td>
<td>25.15</td>
</tr>
</tbody>
</table>

Table 2.2: Calculated RSS paramters.

The distances estimated from the above equations will still have some errors, which we will try to account for by using the multilateration and the map matching techniques. The next chapter focuses on the multilateration technique to determine the location of an unknown tag.
2.4. RSS CALIBRATION

Figure 2.2: RSS plot at the two antennas

(a) Antenna 0

(b) Antenna 1
Figure 2.3: Plot of actual vs. calculated distances of the landmarks from the two antennas
3

Localization Process

In the previous chapter, the ranging technique used to obtain the distance information between an unknown tag and reader antennas was discussed. This chapter describes the approach used to determine the two dimensional location of an unknown target tag. Section 3.1 discusses the commonly used localization techniques. Section 3.2 details the multilateration technique employed in this thesis to solve the localization problem. Section 3.3 describes the calculations involved in the process. In section 3.4, the localization process is applied on the landmark tags to test the validity of the process.

3.1 Localization Techniques

Three basic techniques have been widely used in locating an unknown sensor node.

3.1.1 Triangulation

Triangulation applies the properties of a triangle to a set of known geometric information between anchor nodes and an unknown target node to compute the location of the target node. The discussion here is limited to two dimensional space.

3.1.1.1 Lateration

Lateration exploits the distance information between anchor nodes and the target node. If the distances of a target node from three anchors are known, then the circles drawn around
these anchors, using the respective distances as radius, will intersect at a unique point, which will be the location of the target node. However, two basic conditions should be satisfied in order to employ the lateration technique.

- There should be at least three anchor nodes, whose locations are known.
- The three anchor nodes should not lie in a straight line.

Figure 3.1 illustrates the concept. This technique is widely used in various localization technologies. GPS [Dana] and Cricket [Priyantha 2005] are some of the examples.

![Figure 3.1: Trilateration example in a 2D space](image)

### 3.1.1.2 Angulation

Angulation is similar to lateration, except that angles are used instead of distances in determining the position of a target node. In general, if two angle measurements and one distance measurement is known then the location of the target node can be determined [Hightower and Borriello 2001] as shown in Figure 3.2. This technique has been explored by Nasipuri and Li [Nasipuri and Li 2002; Karl and Willig 2005] in their paper on Wireless Localization. The VHF Omnidirectional Ranging aircraft navigation system uses a different example of the angulation technique [Hightower and Borriello 2001].
3.1. LOCALIZATION TECHNIQUES

The advantage of the above two techniques is that they are relatively simple to implement. However, the disadvantage is that the accuracy can be affected by environmental changes.

3.1.2 Scene Analysis

The scene analysis technique uses properties of a scene obtained from a particular frame of reference and compares it with previously recorded information to draw conclusions about the location of a target [Hightower and Borriello 2001]. The scene can comprise of visual images [Karl and Willig 2005] or any other measurable characteristic like RSS information [Bahl and Padmanabhan 2000].

The disadvantage is that it requires offline collection of data, which may be very time consuming. Furthermore, if changes in environment alter the perceived properties of the scenes, the offline data collection process has to be repeated.

3.1.3 Proximity

This technique is used to obtain coarse-grain information about a target node, like whether it is in the proximity of another sensor node. Although the technique does not provide
3.2 Multilateration

In this thesis, lateration technique has been used to determine the location of a target tag. However, as described in the previous chapter, the calculated distances between the target and the landmarks are not accurate. The distances are just estimates $D_i$ with some unknown error $\varepsilon$, i.e., $D_i = D_{\text{actual}} + \varepsilon$. Using just three landmarks and the erroneous distances to calculate the location of the unknown tag will, in general, lead to some errors in the localization results. To make the location calculation less sensitive to errors in individual distance measurements, more than three landmarks (multilateration) are used. This will turn the lateration problem into an over determined system of linear equations, for which a solution can be computed that minimizes the mean-square error. Figure 3.3 shows the idea of multilateration as used in this thesis. As explained later in this chapter, the multilateration process effectively becomes a triangulation calculation.

Normally, in a lateration problem, the anchors are active devices, which are capable of passing information to the unknown tag, from which the distance can be deduced and the lateration technique applied. But in this thesis, only two reader antennas are used, which act as our active devices. Therefore the concept here is to determine the target’s location with reference to $N$ passive landmark tags and not with reference to the two antennas. The approach serves to solve two problems:

- It reduces the cost of the system. The cost of employing one more reader antenna is far more than using multiple tags as landmarks.
- Due to the directionality aspect of the tag antenna, which is printed dipole antenna, the
detection of the tag by three non collinear reader antennas, is difficult [Bekkali et al. 2007; Griffin et al. 2005].

3.3 Determining the Location of an Unknown Tag

This section explains the technique employed to determine the distances between an unknown tag and landmark tags, thereby effectively causing the landmark tags to act as anchors. The idea was demonstrated by Bekkali et al [Bekkali et al. 2007], in their paper on RFID Indoor Positioning. The paper has been extensively referenced for this section of the thesis.

A map generated from landmark tags, whose locations are known is used to determine the location of the unknown tag. All the distances are measured in meters. The setup uses the grid shown in Figure 3.3. It consists of two RFID antennas, A0 and A1, placed at location A0 (0, 0) and A1 (1.6764, 0) respectively. Note that the co-ordinates of the antennas are not required for the localization process; just the knowledge of the distance between the antennas is good enough. Both the antennas are connected to the RFID reader. The antennas alternatively interrogate the surrounding to detect the presence of tags. The distance between the RFID antennas and the detected tags is then determined using the Ranging technique described in Chapter 2.
3.3. DETERMINING THE LOCATION OF AN UNKNOWN TAG

The next step is to calculate the distance between a target tag and a landmark tag. Figure 3.4 demonstrates the geometry involved in computing the distance $D_i$ between a target tag and a landmark.

![Figure 3.4: Geometry involved in Tag Landmark distance measurement](image)

$|d_i^k| i = 1...n; k = 1,2$ is the calculated distance between the $i$th landmark and $k$th antenna (A0 and A1).

$|d_i^k| k = 1,2$ is the calculated distance between the target tag and the $k$th antenna.

$|d_{12}|$ is the distance between the antennas A0 and A1.

From Figure 3.4, the distance between the target tag and the $i$th landmark tag can be calculated as

$$D_i = \sqrt{|d_i^k|^2 + |d_i^l|^2 - 2|d_i^l||d_i^k|\cos(\phi_i - \theta)} \quad (3.1)$$

where,

$$\phi_i = \cos^{-1}((|d_{12}|^2 + |d_i^l|^2 - |d_i^k|^2)/2|d_{12}||d_i^k|) \quad (3.2)$$
3.3. **DETERMINING THE LOCATION OF AN UNKNOWN TAG**

\[
\theta = \cos^{-1}(\frac{(|d_{12}|^2 + |d_{1t}|^2 - |d_{2t}|^2)/2|d_{12}||d_{1t}|)} (3.3)
\]

Using the above equations, distance of the target tag from each landmark can be calculated as \(D_i, i = 1,....n\). Now, if the position of landmarks is denoted as \(L_i\) and the position of the target tag as \(X\), then the distance \(D_i\) can be expressed as follows

\[|L_i - X| = D_i, i = 1, ...n \] (3.4)

The co-ordinates of the landmarks are denoted as \((l_{x_i}, l_{y_i})\) and the target tag as \((x, y)\). Now, to obtain the location of the target tag, we have to solve the over determined system of the non-linear equations of the form

\[(l_{x_i} - x)^2 + (l_{y_i} - y)^2 = D_i^2, i = 1, ...n \] (3.5)

The steps involved in solving the non-linear equations can be found in the book [Karl and Willig 2005]. They are summarized as follow.

Equation (3.5) is expanded as follows

\[(l_{x_1} - x)^2 + (l_{y_1} - y)^2 = D_1^2 \] (3.6)
\[(l_{x_2} - x)^2 + (l_{y_2} - y)^2 = D_2^2 \] (3.7)
\[\vdots \]
\[(l_{x_n} - x)^2 + (l_{y_n} - y)^2 = D_n^2 \] (3.8)

Since the aim is to calculate the co-ordinates \(x\) and \(y\), we will try to separate the terms \(x\) and \(y\) from the above equation. This can be achieved by subtracting the equation involving
the \(n\)th landmark from equations involving the remaining landmarks

\[
(lx_1 - x)^2 - (lx_n - x)^2 + (ly_1 - y)^2 - (ly_n - y)^2 = D_1^2 - D_n^2 \tag{3.9}
\]

\[
(lx_2 - x)^2 - (lx_n - x)^2 + (ly_2 - y)^2 - (ly_n - y)^2 = D_2^2 - D_n^2 \tag{3.10}
\]

\[
\vdots
\]

\[
(lx_{n-1} - x)^2 - (lx_n - x)^2 + (ly_{n-1} - y)^2 - (ly_n - y)^2 = D_{n-1}^2 - D_n^2 \tag{3.11}
\]

Simplifying the quadratic equation, we get

\[
2(lx_n - lx_1)x + 2(lx_n - ly_1)y = (D_1^2 - D_n^2) - (lx_1^2 - lx_n^2) - (ly_1^2 - ly_n^2)
\]

\[2(lx_n - lx_2)x + 2(lx_n - ly_2)y = (D_2^2 - D_n^2) - (lx_2^2 - lx_n^2) - (ly_2^2 - ly_n^2)\tag{3.12}\]

\[\vdots\]

\[2(lx_n - lx_{n-1})x + 2(lx_n - ly_{n-1})y = (D_{n-1}^2 - D_n^2) - (lx_{n-1}^2 - lx_n^2) - (ly_{n-1}^2 - ly_n^2)\tag{3.13}\]

The equations can now be written in matrix form as:

\[
\begin{bmatrix}
(lx_n - lx_1) & (ly_n - ly_1) \\
(lx_n - lx_2) & (ly_n - ly_2) \\
\vdots & \vdots \\
(lx_n - lx_{n-1}) & (ly_n - ly_{n-1})
\end{bmatrix}
\begin{bmatrix}
x \\ y
\end{bmatrix} =
\begin{bmatrix}
(D_1^2 - D_n^2) - (lx_1^2 - lx_n^2) - (ly_1^2 - ly_n^2) \\
(D_2^2 - D_n^2) - (lx_2^2 - lx_n^2) - (ly_2^2 - ly_n^2) \\
\vdots \\
(D_{n-1}^2 - D_n^2) - (lx_{n-1}^2 - lx_n^2) - (ly_{n-1}^2 - ly_n^2)
\end{bmatrix}
\]

\[2 \begin{bmatrix}
(lx_n - lx_1) & (ly_n - ly_1) \\
(lx_n - lx_2) & (ly_n - ly_2) \\
\vdots & \vdots \\
(lx_n - lx_{n-1}) & (ly_n - ly_{n-1})
\end{bmatrix}
\begin{bmatrix}
x \\ y
\end{bmatrix} =
\begin{bmatrix}
(D_1^2 - D_n^2) - (lx_1^2 - lx_n^2) - (ly_1^2 - ly_n^2) \\
(D_2^2 - D_n^2) - (lx_2^2 - lx_n^2) - (ly_2^2 - ly_n^2) \\
\vdots \\
(D_{n-1}^2 - D_n^2) - (lx_{n-1}^2 - lx_n^2) - (ly_{n-1}^2 - ly_n^2)
\end{bmatrix}
\tag{3.15}\]

The problem is now turned into an over determined system of linear equations of the form \(Ax = b\). A solution, that is the pair \((x, y)\), can be computed that minimizes the mean square
error $\| Ax - b \|_2$. The solution yields us the location $(x, y)$, which best satisfies the position constraints from all the $n$ landmarks, with minimum average error. The problem is solved using the matlab operator ‘\’ which gives the solution in least squares sense.

### 3.4 Initial Validation Results

In order to validate the technique, multilateration was applied on the nine landmark tags to compute their positions. Figure 3.5 shows the map of the nine landmark tags as placed in our experimental setup.

![Figure 3.5: A Map of the nine landmark Tags](image)

To apply the multilateration technique, each landmark tag was in turn treated as an unknown tag. One by one their locations were computed with reference to the remaining landmark tags. A surprise finding was that, for landmark tags, it is much better to use the
known distances \(d_i\) instead of the calculated distances in the calculations above, specifically Equations (3.1) and (3.2). This is because the measured distances do not misbehave consistently for different landmarks and that leads to a larger error in the calculated \((x, y)\) co-ordinates. Therefore Equations (3.1) and (3.2) are modified to use the known distances of landmarks to the antennas. As a result of this modification, only two measured distances (i.e. distances \(d_t\)) are used in the multilateration algorithm. That effectively converts the multilateration algorithm to a triangulation one. The calculated location results were plotted against the actual location information as shown in Figure 3.6.

As expected, there were errors in the calculated \((x, y)\) co-ordinates of the landmarks. The next chapter discusses the use of map matching technique to account for the localization errors.
3.4. **INITIAL VALIDATION RESULTS**

Figure 3.6: Plot of actual vs. calculated co-ordinates of the landmarks

(a) x co-ordinates

(b) y co-ordinates
4

Map Matching Technique and Software Tools

In this chapter, an error map matching technique is discussed to refine the localization results obtained in Chapter 3. The chapter also describes the details of the software tool developed in the thesis including a Windows program and a Matlab program.

4.1 Map Matching Technique

As described in Chapter 3, the localization results obtained are not accurate and contain some errors. In this section a Map Matching technique is developed to minimize the observed errors.

Each landmark tag is treated as an unknown tag and its location is computed using the localization technique. Since the actual location of the landmark tags is known, the error at each landmark tag is computed as:

\[
\begin{align*}
\text{error}_x &= x_{\text{actual}} - x_{\text{calculated}} \\
\text{error}_y &= y_{\text{actual}} - y_{\text{calculated}}
\end{align*}
\]

The errors are mapped to the calculated values of \( x \) and \( y \) co-ordinates of the landmark tags, which leads to two error maps, one for \( \text{error}_x \) and one for \( \text{error}_y \) for a total of nine
landmark tags. Note that the maps associate the $error_x(y)$ values with their corresponding calculated $x(y)$ values, not their actual $x(y)$ values. Figure 4.1 shows the maps in 3 dimension.

The error map data is stored in two Excel sheets. In one file, the calculated $x$ and $y$ co-ordinates of the landmark tags are stored. The computed $error_x$ and $error_y$ are stored in another file. A Matlab program is written to read these files and associate the errors with each landmark tag.

When an unknown tag is detected, its location is computed using the localization technique described in Chapter 3. The calculated location is then refined using the error map built above. The $x$ and $y$ co-ordinates of the unknown tag are compared with the above two maps, and a 2-D surface interpolation technique is applied to obtain an estimate of $error_x$ and $error_y$ terms for the unknown tag. The error terms are then added to the calculated $x$ and $y$ co-ordinates of the unknown tag to determine the final location.

The Matlab function ‘griddata’ is used to achieve the 2-D surface interpolation. According to [MathWorks], “$ZI = \text{griddata}(x, y, z, XI, YI)$ fits a surface of the form $z = f(x, y)$ to the data in the (usually) nonuniformly spaced vectors $(x, y, z)$. griddata interpolates this surface at the points specified by $(XI, YI)$ to produce $ZI$. The surface always passes through the data points. $XI$ and $YI$ usually form a uniform grid (as produced by meshgrid).” The griddata functionality in Matlab uses four different interpolation methods. These are: ‘linear’, ‘cubic’, ‘nearest’, ‘v4’. These methods are differentiated on the basis of the type of surface used to fit the data. The ‘cubic’ and ‘v4’ methods produce smooth surfaces, while ‘linear’ and ‘nearest’ methods produce discontinuous surface. For building the error map in this thesis, the ‘v4’ method is used, for the following two reasons:

- The method produces a smooth surface fit.
- The method is also able to interpolate the points outside the boundary. In our case, it is possible that a calculated value of $x$ and $y$ may lie outside the boundary.

The ‘v4’ method uses the Biharmonic spline interpolation technique documented in [Sandwell 1987; MathWorks].
4.1. MAP MATCHING TECHNIQUE

Figure 4.1: Plot of error map at calculated locations of 9 landmark Tags
For each unknown tag, the griddata function is called twice, once for computing the $error_x$ term and once for $error_y$ term.

To validate the technique, the map matching technique was applied to the calculated co-ordinates of the landmark tags. As expected, the new location co-ordinates were without any errors. This is because the griddata interpolated surface always passes through the data points used to build the map. However, it is expected that when the technique is applied to an unknown tag, the results will still have some errors.

During experiments, it was observed that some points on the experimental setup grid would show unexpected RSS values. For instance, some points record an RSS value larger than expected and some points show RSS values lesser than expected. These points are called outliers and null spots respectively. The outliers are due to the multipath reflection from the surroundings whereas null spots are areas in the reader field that do not receive radio waves due to the shape of the waves. To account for some of these observed spots, we added four more landmark tags at these points. The newly added landmark tags will not participate in the calibration process described in Chapter 2; they will just be used for building the error map. Figure 4.2 shows the expanded error maps with additional landmark tags.
4.1. MAP MATCHING TECHNIQUE

(a) error map of x co-ordinate with 13 landmark tags

(b) error map of y co-ordinate with 13 landmark tags

Figure 4.2: Plot of error map at calculated locations of 13 landmark Tags
4.2 Software Tools

This section provides a description of the software tool developed in this thesis. Figure 1.6 shows the frontend GUI (Graphical User Interface) window.

4.2.1 Modes of Operation

The tool can be configured to work in two modes, RSS calibration mode and Localization mode. The choice can be made by checking the appropriate radio button.

4.2.1.1 RSS Calibration Mode

When the tool is configured in this mode, RSS parameters for the ranging technique described in Chapter 2 are computed. The user is allowed to enter an array of landmark tag IDs in the “Enter Tag ID” textbox. The tag IDs should be separated by ‘,’. The user can also input the number of iterations to be run. Once the “Start” button is clicked, a pair of RSS values (one for each antenna) is collected for each landmark tag. The data collected is also fed to an external file. Once the data collection process is done, a Matlab program is invoked by the Windows program. The Matlab program computes the RSS calibration parameters and also builds an error map. The calibration data and the error map data are fed into Excel files, which are later used when the program runs in the localization mode.

4.2.1.2 Localization Mode

In the localization mode, the user is allowed to enter one tag ID, whose location is to be computed. At this point, the tool is programmed to perform the localization of a single tag at a time. However, the program can be easily modified to locate multiple tags at a time.

Once the “Start” button is clicked, the localization process is started. The Windows program determines the RSS values of the unknown tag from the two antennas and calls a Matlab program. The Matlab program reads the RSS values and also the RSS calibration parameter values to compute the location of the unknown tag. The localization result is then refined, using the error map built in the RSS calibration mode.
As evident from the above discussion, each run of the program goes through two stages.

- The data collection stage, for which the Windows program is used.
- The data processing stage, for which a Matlab program is used.

The sections below, describe the operation of both the programs in detail.

4.2.2 Windows Program

The Windows program is developed using Microsoft’s .NET environment. The program uses the APIs provided by Alien technology, to control the Alien reader.

4.2.2.1 Structural Description

Figure 4.3 shows an overview of the class diagram.
Form1 is the main class in the program, which interacts with the Tags class and the alienReader class, to achieve the data collection process. The alienReader class declares an instance of the clsReader class available in Alien Technology’s .NET API as follows:

- private clsReader mReader = new clsReader();

The clsReader class is used to establish communication with the reader, which is connected to the host computer via a serial cable. The clsReader class provides a number of simple commands to control the functionality of the reader. A few important commands...
4.2. SOFTWARE TOOLS

used in our program are discussed below:

- **mReader.InitOnCom()**: Establishes the communication between the host computer and the reader using the COM port. The method takes an integer as the argument, which specifies the COM port number.

- **mReader.Disconnect()**: The method is used to disconnect the reader.

- **mReader.RFModulation**: This property allows the user to specify the modulation mode. For this thesis, the reader was configured to work in "DRM" (Dense reader Mode) mode. It has enhanced filtering capabilities for better operation in a noisy environment.

- **mReader.AcquireMode**: This property specifies the operation mode of the reader. The reader in this thesis was set to work in “Inventory” mode. In this mode, the reader uses an anti-collision algorithm, to perform tag interrogation. This is useful when there are many tags in front of the reader, which is precisely the case in our thesis.

- **mReader.AntennaSequence**: This property is used to specify the antenna sequence. It expects a string value.

- **mReader.RFAttenuation**: The property gets/sets the attenuation value, which is used to specify the level of attenuation that should be applied to the reader’s emitted power. The RF attenuation value can be varied from 0 to 150 in increments of ten with each increment representing a one dB reduction in power. As described in Chapter 2, attenuation value of 0 indicates no attenuation i.e., maximum power transmitted by the antenna, and an attenuation value of 150 (-15dB) indicates maximum attenuation i.e., minimum power. This property is used to determine the RSS value of the tag.

- **mReader.TagList**: This property gets the list of the tags detected by the reader.

### 4.2.2.2 Functional Description

The most important function of the Windows program is to obtain the RSS readings of the tags, from the two antennas.
Once a user enters the tag IDs in the GUI window and clicks on the “Start” button, the reader is initialized and the data collection process is started. Antennas connected on the ports 0 and 1 of the reader panel are used to transmit and receive the RF signal. Each antenna is alternatively selected to act as a transmitter.

Next, the attenuation value of the reader is changed iteratively from 0 to 150, in increments of 10, with each iteration representing a one dB reduction in power. During each iteration, the surrounding is interrogated for tags using the mReader.TagList command. The command returns a list of tags detected by the reader. For each iteration, the tag interrogation is repeated 20 times because sometimes the tags are slow to respond. It was found, via experiments that many tags would not respond to a tag interrogation during the first try but will respond during the subsequent trials. Thus to make sure that the tags are not missed, 20 trials are repeated for each iteration of attenuation value.

The Alien reader returns a list of tags which makes it complicated to separate tags on the basis of different attenuation levels. Therefore we use hash tables to aid us in determining the attenuation with which each tag is detected. Three hash tables are used: MyTable, MyDetectedTable, MyNotDetectedTable. The hash tables are global to the class Form1. The tag IDs inputted by the user are used to create an array of tag objects. A tag object has two important fields: tagID and attenuation value. The tag objects are then placed in the hash table called MyTable. The tag list detected by the reader is compared against the MyTable list to determine which tags were detected. If a tag is detected, it is removed from MyTable, placed inside MyDetectedTable and is ignored for the subsequent trials. If a tag is not detected even after 20 trials, it is confirmed to be not in the reading range of the reader. At the end of the 20 trials, the non-detected tags are placed in the MyNotDetected table, and they are assigned an attenuation value of the previous iteration. The attenuation value of the reader is then increased by 10 to perform the next iteration of tag interrogation. Before performing the next iteration, the tags from MyDetected table are moved to MyTable, so that only the tags which were detected in the previous iteration are interrogated. The process is repeated for 15 iteration, increasing the attenuation factor by 10 for each iteration. At the end of the 15 iterations, the MyNotDetected table will have set the
attenuation values for most of the tags. At this point, if MyTable is not empty, it means that some tags were detected even when maximum attenuation was applied to the reader’s signal. Therefore, these tags are assigned an attenuation value of 150. The attenuation values are then converted to dB to represent the RSS values.

The flowchart shown in Appendix A, illustrates the entire procedure of collecting the RSS values for the tags.

Once the data collection process is done and the RSS values of the tags are determined, the data is fed into a text file and the appropriate Matlab program is called depending upon the mode of operation.

### 4.2.3 Matlab Program

The Matlab Program is used to process the data collected by the Windows program. This section just provides a brief overview of the Matlab programs. The techniques relevant to the data processing part is already explained in detail in the previous chapters. There are two different sets of Matlab programs for the two operation modes.

#### 4.2.3.1 RSS Calibration Mode

In this mode, the Matlab program performs two functions.

**RSS Calibration:** The Matlab program “calibration.m” reads the RSS data of the 9 landmarks, collected by the Windows program, via a text file. The data is stored in two, (1 x 9) vectors, $R_1$ and $R_2$. $R_1$ represents the RSS values of the landmark tags with respect to Antenna 0, and $R_2$ represents the RSS values with respect to Antenna 1. The distance of the landmark tags are available in two, (1 x 9) vectors, $dist_1$ and $dist_2$, with respect to antenna 0 and antenna 1, respectively. Using the RSS values and the distance information, the RSS parameters for the two antennas are calculated using the ranging technique developed in Chapter 2. The RSS parameters are stored in a (2 x 2) matrix called $RSS_{param}$. Row 1 of the matrix represents the RSS parameters for antenna 0 and row 2 for antenna 1. The matrix is then saved onto an Excel file, called “calibration_param.xls”. The program then
builds the error map.

**Error Map Building:** The distance of the landmarks from the two antennas is calculated using the RSS values and the calibration parameters, and applying the ranging procedure. The calculated distances are stored in two, (1 x 9) vectors, called as *calc_dist1* and *calc_dist2*, representing the calculated distance of the landmark tags from the two antennas.

- A function “multilatation_landmarks” is called, which takes RSS values, along with the calculated and actual distance information of the landmarks as inputs and returns a matrix, composed of the calculated *x*, *y* co-ordinates of the requested landmarks. The “multilatation_landmarks” employs a for loop. During each iteration of the for loop, one landmark tag is treated as an unknown tag. The location information of the unknown landmark is then calculated using the multilateration technique, with the remaining eight landmark tags acting as reference tags. For calculating the location of the unknown landmark, its RSS values and *calc_dist* information is used. The for loop thus calculates the *x*, *y* values of all the landmarks and stores them in a (2 x 9) matrix, called *Ans_landmark*, where row 1 represents the calculated *x* co-ordinates and row 2 represents the calculated *y* co-ordinates of the 9 landmarks. The columns represent the 9 landmarks. The matrix is returned to the “calibration.m” function.

The actual *x*, *y* co-ordinates of the landmarks are already available to the program. Using the calculated values of the *x*, *y* co-ordinates, *error_x* and *error_y* is computed as:

\[
\text{error}_x = x - \text{calculated}_x;
\]

\[
\text{error}_y = y - \text{calculated}_y;
\]

The errors are stored in a (2 x 9) matrix called *error_mat*, where row 1 represents error in *x* and row 2 represents error in *y*. The *error_mat* is stored in an Excel file, “error_mat.xls”. Similarly, the matrix representing the calculated *x*, *y* co-ordinates is stored in “location_map.xls” file. Thus the error map data is created.
4.2. SOFTWARE TOOLS

4.2.3.2 Localization Mode

In this mode, the location information of an unknown tag is determined and the map matching technique is employed to refine the result. The Matlab program, “localize.m”, is used. The program reads the RSS information of the unknown tag and then reads the RSS calibration parameters from the Excel sheet “calibration_param.xls”. The ranging technique is then employed to determine the distance of the tag from the two reader antennas.

- The function “locate_unknown.m” is then called. The function already has the location information of the 9 landmark tags, which can be updated using an Excel sheet. The function takes the calculated distance information of the unknown tag as input and performs multilateration with respect to 9 landmark tags and computes the location $x, y$ of the unknown tag. The location information is returned to the “localize” program.

The program then calls the “interpolation.m” function. The input to the function is the calculated $x, y$ co-ordinates of the unknown tag.

- The function reads the error map data from the files “error_mat.xls” and “location_map.xls”. The data are put into vector variables, $error_x, error_y, calc_x$ and $calc_y$. The grid-data interpolation technique is used on the data to determine the error estimate for the unknown tag. The griddata function is called twice.

$$
interp_error_x = \text{griddata}(calc_x, calc_y, error_x, x, y);
$$
$$
interp_error_y = \text{griddata}(calc_x, calc_y, error_y, x, y);
$$
$$
x_{after\_interp} = x + interp\_error_x;
$$
$$
y_{after\_interp} = y + interp\_error_y;
$$

The refined location of the unknown tag is returned to the “localize” program, which displays the $x, y$ co-ordinates of the unknown tag on the screen. Both the results are displayed on the screen, the one obtained before the map matching being applied, and the one obtained after applying the map matching technique.
This chapter summarizes and analyzes the results obtained during the experiments. Section 1 describes the aim of the thesis. In Section 2, the results obtained are discussed using various analysis plots. Section 3 compares the results with a similar localization technique.

5.1 Aim

The thesis tries to solve the problem of indoor localization using the RSS information of passive tags. Since the experiments were performed inside a computer lab, the RSS readings of the tags are bound to be affected by the interference from various sources. This, coupled with the fact that relatively inexpensive passive RFID tags and only one reader is used, limits the accuracy of the localization technique. Consequently, we do not aim to compute the pin point location of the unknown tags; rather the aim is to localize a tag within a given region. It is also important to note that due to the directionality aspect of the tag antenna, which is printed dipole antenna, it is assumed that the tags are always facing the reader antennas.

5.2 Results

The experiments in this thesis were performed on a rectangular grid measuring 1.68 x 1.54 meters. Figure 2.1 shows the experimental setup used to validate the technique. A series of experiments were conducted by placing the unknown RFID tags at various points in
the grid, and applying the localization technique. Overall, the localization experiment was performed at 138 points. Two sets of results were obtained for each run of the experiment. The first set was obtained without applying the Map Matching technique, while the second set was obtained after applying the Map Matching technique.

5.2.1 Plotting X and Y co-ordinates

Figures 5.1 and 5.2 show the plot of calculated vs. actual values of $x$ co-ordinates, for the two cases. Although it is not very clear, it seems that the map matching technique was able to reduce the number of outliers.

![Figure 5.1: Plot of calculated vs. actual x co-ordinates without Map Matching](image)
Figure 5.2: Plot of calculated vs. actual x co-ordinates with Map Matching
Figures 5.3 and 5.4 show the plots of actual vs. calculated \( y \) co-ordinates. It seems that the map matching technique tends to reduce the error for the higher values of \( y \) co-ordinates. These are the points farther away from the reader.

Figure 5.3: Plot of calculated vs. actual \( y \) co-ordinates without Map Matching
Figure 5.4: Plot of calculated vs. actual y co-ordinates with Map Matching
5.2. RESULTS

5.2.2 Plotting Distance Errors

In order to get a better estimate of the localization results, 3-D plots of the distance error at the different experimental points are provided. We define the distance error as the Euclidean distance between the actual and the estimated location of the unknown tags.

$$\varepsilon = \sqrt{(x_{\text{actual}} - x_{\text{calculated}})^2 + (y_{\text{actual}} - y_{\text{calculated}})^2}$$

Figure 5.5 shows the 3-D error plot for the case without map matching and Figure 5.6 shows the plot for the map matching case.

Figure 5.5: 3-D error plot without Map Matching
Figure 5.6: 3-D error plot with Map Matching
Next the CCDF (Complimentary Cumulative Distribution Function) is plotted for the errors in both the cases. Here, the function $CCDF(e) = \text{Probability}(\text{Error} >= e)$. Figure 5.7 shows the plot along with the statistics for both the cases. The average error for the map matching case is 0.4574 meter, with a maximum error of 1.58 meters. The average error for the case without map matching is 0.4773 meter, with a maximum error of 1.4 meters.

![CCDF plot of the error for the two cases](image)

Although it is not very clear, it can be seen from Figures 5.5 and 5.6 that as the distance from the antennas increases, the map matching technique provides a better estimate of the
5.2. RESULTS

location. However, for points closer to the antenna, the case without map matching seems to provide better results. Overall, the average error in the map matching case is slightly better.

To illustrate the above point, we divide the region into two regions, one for the points close to the two antennas and the other with points far away from the antennas. The points that fall with actual $y \geq 2.0$ are considered as far points and the points that fall before $y < 2.0$ are considered as near points. We computed the average error in both the cases. Table 5.1 shows the results:

<table>
<thead>
<tr>
<th>Technique</th>
<th>Error in Near Region</th>
<th>Error in Far Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Map Matching</td>
<td>0.35</td>
<td>0.68</td>
</tr>
<tr>
<td>Map Matching</td>
<td>0.37</td>
<td>0.59</td>
</tr>
</tbody>
</table>

Table 5.1: Error Comparison for the two techniques.

Thus it can be deduced that the Map Matching technique works better for points farther away from the antenna, while the Non Map Matching technique works better for the near region. Thus the Map Matching technique does not always work. One reason is because due to the presence of outliers, there are points in the far region, which observe a signal strength reading higher than expected and as such may end up with a set of readings similar to points in the near region. The map matching technique tries to minimize the localization errors for some of these outliers. However, in doing so the map matching technique also ends up refining the localization results for the points in the near region, with a similar set of readings, which actually might not need a big refinement. This is a fundamental limitation in using just two reader antennas for the localization technique. Also, as explained by Eiman, et al [Elnahrawy et al. 2004]; localization techniques using RSS information are bound to have some limitations in terms of accuracy due to environmental factors. It is evident that the region farther away from the antenna is more prone to errors and the map matching technique aids us in countering the errors. Consequently, we provide users with two set of results. The table above also shows that for 75% of the region, the location accuracy is 0.35 meter, which is very encouraging. The major inaccuracy in the remaining 25% of the region is due to the heavy fluctuations of the received signal strength, which is
5.3. PERFORMANCE EVALUATION

difficult to model. One possible solution is to employ additional reader antennas.

To verify the effectiveness of the RSS calibration technique, we pick up a value of $n = 2.4$ from Table 2.1 in Chapter 2, for an office environment. The value of $P_{l_0}$ (reference path loss at 1 meter) is -15 dB. The average error obtained using the above parameter values is 0.7 meter without map matching and 0.56 meter with map matching. Thus the RSS calibration technique provides a considerable improvement in the localization results.

5.3 Performance Evaluation

In order to evaluate the performance of our system, we compare it to another RFID based localization system, which also depends upon the RSS information of the tags to solve the problem of localization. The Landmarc system [Ni et al. 2003], as explained in Chapter 1, compares the RSS information of an unknown active tag with priorly generated RSS map, to deduce the location of the unknown tag. The system uses four RFID readers along with one active reference RFID tag per square meter. The system was able to achieve an average accuracy of 1 meter with worst case error of 2 meters. In this thesis we were able to achieve an average location accuracy of 0.45 meter, with a worst case error of 1.5 meters. The results are very encouraging considering the inexpensive setup with which the results were achieved. The system, like Landmarc, is also able to account for dynamically changing environmental conditions. Landmarc, however, is able to cover a wider range due to the use of active RFID tags.
6

Conclusion and Future Work

Passive RFID tags pose an interesting solution to the problem of indoor localization, given the ease of deployment and the cost effective infrastructure. Airline companies have already started tagging suitcases with RFID tags while the construction industry is employing RFID tags to keep track of equipments. Adding context awareness to these tagged objects will only help increase productivity.

In this thesis, an indoor localization technique using passive RFID tags was studied and implemented based on an idea presented by Bekkali et al [Bekkali et al. 2007]. A ranging technique was developed to correlate the distance information of the tags with the signal strength information. The ranging technique uses the concept of RSS calibration to model the propagation of RF waves in an indoor environment.

The technique described in [Bekkali et al. 2007] is used to determine the inter-tag distance between an unknown tag and a landmark tag. Multilateration is performed on the unknown tag with respect to the landmark tags, to obtain the location of the unknown tag. It was found out that for the set up in this thesis, the multilateration technique was not more accurate than the triangulation technique.

A map matching technique is implemented to improve the localization results. One by one, each landmark tag is treated as an unknown tag and the localization technique discussed above is applied to obtain the calculated locations for all the landmark tags, in terms of x and y co-ordinates. The calculated locations are compared with the actual locations to built an error map. The error map is built for x co-ordinates as well as y co-
ordinates. The calculated location of an unknown tag is then refined by applying a 2-D surface interpolation technique on the error map.

A software tool is developed using the Microsoft .NET environment and Matlab. The .NET program collects the RSS data from the tags and feeds it to the appropriate Matlab program for post-processing. The software tool can be configured to work in two modes:

- **RSS Calibration Mode**: Used to calibrate the RSS parameters for the two antennas, and to generate the error maps to be used for the Map Matching technique.

- **Localization Mode**: Used to localize an unknown tag.

It is observed that the RSS calibration technique provides a better estimate of the Radio propagation model for an indoor environment. The calibration parameters are different for both the antennas. The RSS calibration technique also provides better adaptability to the dynamically changing environment.

It is also observed that as the distance from the reader increases, the RSS readings are more prone to fluctuations due to environmental factors such as multipath reflections, interference from other signals in the environment, etc. Therefore, the localization error increases as the distance from the reader increases. The error map matching technique improves the localization results, for the points which are farther away from the reader and with more errors.

**Future Work**

This thesis presented a prototype of an Indoor Localization system using passive RFID tags. Localization techniques using RSS information are bound to have some limitations due to environmental factors [Elnahrawy et al. 2004; Savvides et al. 2001; Whitehouse et al. 2007]. The paper presented by Eiman, et al [Elnahrawy et al. 2004], shows that these limitations are fundamental to localization techniques using signal strength information. Even with these limitations, the thesis was able to achieve fairly acceptable results, and demonstrate that it is feasible to employ passive RFID tags to solve the problem of indoor
Localization. The following advancements if made possible can lead to a more accurate and practical localization system.

- The RFID tags used in this thesis have a limited range of 3-4 meters. RFID companies are striving to provide passive RFID tags with a wider detection range. This will aid in designing a more practical localization system.

- In this thesis, only two RFID antennas are employed, which leaves us with just a pair of RSS information for each tag to perform the localization. Adding more antennas will provide us with more data to analyze and will certainly aid in improving the localization results.

- One more limiting factor was the fact that only 15 discrete power levels were available to estimate the signal strength of the tags. RFID companies are now offering to provide users with software, that provides RSS information of the tags directly. This will not only help in providing better signal strength information but also help in speeding up the localization process. The current technique employed in this thesis requires the program to go through 15 iterations to determine the signal strength of a tag. Alien technology recently released one such software, which directly provides RSS information of the tags.

- The localization technique developed in this thesis assumes that the RFID tags are always facing the reader antennas. A more robust technique can be designed to account for different orientations of the tags. RFID companies are doing more research to provide tags with context information. Alien technology is planning a software release in the future, which will provide distance information of the tags. According to a news article in RFID journal [RFIDJournal 2008], “To determine a tag’s distance from a reader, the Alien software uses an algorithm that counts the wavelengths of a tag’s response to interrogation. The distance of any tag within a reader’s interrogation zone can be determined with an error margin of 10 percent. (If a tag is 10 feet from the reader, for example, the software may come up with a calculated distance of 9 or 11 feet, instead).” Such capabilities, if available in the future, will greatly aid in solving the problem of
indoor localization using cheap passive RFID tags.
Appendix

A.1 Flow Chart for the Data Collection Procedure

The flowchart below describes in detail the RSS collection procedure, employed by the Windows program to determine the signal strength of the tags.
Figure A.1: FlowChart for the RSS collection procedure

Start

int reads = 0;
int iterations = value entered by the user;

Is reads < iterations?

Yes

reads++

Int antSeq = 0

antSeq = 2?

Yes

reads++;
alien.readerSetting() is called:
Sets the reader configuration.
alien.setAntSeq(Convert.ToString(antSeq)) is called:
Sets the appropriate antennas as the transmitter and receiver.

No

Done

A (from page ii)

addingTags() is called:
An array of tag objects tagArray is created depending upon the tag IDs inputted by the user. The array may contain one element or multiple elements depending upon the operation mode selected by the user.
The tag object has 3 important fields. They are:
1) int attenuation
2) string ID
3) bool detection

B (on page ii)
A.1. FLOW CHART FOR THE DATA COLLECTION PROCEDURE

Page ii

**Initialization:**
Three hash tables are initialized:
- MyTable, MyDetectedTable, MyNoDetectedTable.
For each hash table, a tag ID is the key and the tag object is the value.

**creatingHash()** is called:
The tags in the tagArray are added to the hash table, MyTable.
This table represents the tag IDs for which the user wants to compute the RSS values.

**GettingTags()** is called:

**C**
(on page iii)

Function call

**A**
(on page i)

**C_return**
(on page iii)

antSeq++;
gettingTags():
Attenuation value of the reader is changed through 15 iterations, from 0 to 150, incrementing the attenuation by 10 during each iteration. Whenever a tag is not detected, during an iteration, it is assigned the attenuation value of the previous iteration.

At this point the attenuation has exceeded the highest value possible, i.e. 150. If there are still tags left in MyTable, assign them an attenuation value of 150 (160 - 10), i.e. maximum RSS.
A.1. FLOW CHART FOR THE DATA COLLECTION PROCEDURE

Flow Chart Diagram:

1. **F** (on page viii)
   - Check if \( \text{tries} < 20 \) and \( \text{allDetected} \neq \text{true} \)
   - If true, go to **G** (on page vii)
   - If false, increment \( \text{tries} \)

2. **G** (on page vii)
   - Process: \( \text{process(alienDetectedTagList, attn1)} \)
   - Check if all tags are detected
   - If not detected, return

3. **G** (on page vii)
   - GenerateArrayList: \( \text{alienDetectedTagList} = \text{alien.get_Tags(attn1)} \)
   - Function call

4. **H**
   - If \( \text{allDetected} \neq \text{false} \)
     - Call function: \( \text{alien.get_Tags(attn1)} \)
     - The function in turn invokes the Alien API "mReader.TagList" and returns a list of tagIDs detected for the attenuation value attn1.
   - If not detected, go back to **F**

**Legend:**
- **ArrayList** (technology)
- **Function call** (software instruction)

---

**Note:**
- The diagram represents a flow chart for data collection, detailing the process of identifying and processing detected tag IDs with an attenuation value.
bool processTags(ArrayList\ntagList, int attn)

ICollection MyKeys;
ICollection MyDetectedKeys;
Tags temp_tag = new Tags("");
MyKeys = MyTable.Keys;
object Key → 1st Key in MyKeys

I (on page vi)

Are we done searching all the Keys in MyKeys?

Yes

No

Does tagList.Contains(Key.ToString())? (i.e. was the tag detected?)

temp_tag =
(\Tags)MyTable[Key.ToString()];
temp_tag.detection = true;

Does MyDetectedTable.ContainsKey(Key)? (i.e. was the tag detected in previous try?)

No

MyDetectedTable.Add(Key.ToString(), temp_tag);
Adds the detected tag to the MyDetectedTable

Next

Key
MyTable ← MyTable - MyDetectedTable;
Remove tags from MyTable, which are also in MyDetectedTable. Thus we have separated the tags which were not detected for the particular iteration of the tag interrogation. MyTable will now contain the non-detected tags.

Is MyTable empty? processTags(,) returns control.

Yes return true;

No return false;

G_return
addToLists:

D

For each tag Key in MyTable

Is the tag Key already contained in MyNotDetectedTable?

No

temp_tag = (Tag) MyTable[Key.ToString()];
temp_tag.detection = false;
temp_tag.attenuation = attnl;
MyNotDetectedTable.Add(Key.ToString(), temp_tag);

D_return
F (from page iii)

D (Instance 2) (on page vii)

D_return (Instance 2) (from page vii)

MyTable.Clear();
MyTable ← MyDetectedTable
Move the tags from the
MyDetectedTable to MyTable so
they can be queried in the next
iteration

attnl = attnl + 10;

For_loop_A (on page iii)
References


