UTILIZATION OF WIRELESS SIGNAL STRENGTH FOR MOBILE ROBOT LOCALIZATION IN INDOOR ENVIRONMENTS

by

Samuel L. Shue

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Electrical Engineering

Charlotte

2017

Approved by:

____

Dr. James Conrad

____

Dr. Asis Nasipuri

____

Dr. Thomas Weldon

____

Dr. Aidan Browne

ABSTRACT

SAMUEL L. SHUE. Utilization of wireless signal strength for mobile robot localization in indoor environments. (Under the direction of DR. JAMES CONRAD)

Localization is a key component of any mobile robot application. For any task a mobile robot might need to perform, precise knowledge of its pose within its environment is critical. Mobile robots employ a multitude of sensors to estimate position, orientation, and mapping of its environment. Distance to wireless beacons through signal strength decay can be integrated into a simultaneous localization and mapping (SLAM) algorithm of a mobile robot equipped with a wireless transceiver, with an emphasis on indoor environments. However, radio signal strength does not predictably attenuate indoors as it does in open environments due to signal interference, absorption, and reflection from objects within the environment, inflicting unexpected amplification or decay at the receiver known as multipath interference. This causes erroneous distance estimations due to the unexpected changes in signal strength attenuation.

In this research, models of radio propagation as it relates to the received signal strength indicator (RSSI) are explored along with localization techniques which utilize these models. For development and testing of RSSI-based localization techniques a simulation method has been described which utilizes a Markov chain to provide realistic multipath interference on simulated RSSI data. Using this simulation technique, a multipath filtration method is proposed and applied to a range-only SLAM algorithm.

ACKNOWLEDGEMENTS

I would like to thank Dr. James Conrad for not only fulfilling his role as my advisor throughout this research, but also for constantly supporting me throughout all of graduate school with various funding opportunities. I would also like to thank him for, despite my difficulties as an undergraduate student, giving me a second chance at UNC Charlotte. If it was not for him giving me the opportunity to prove myself, I would not be where I am today.

I want to thank my committee as well, Dr. Asis Nasipuri, Dr. Thomas Weldon, and Dr. Aidan Browne, for their guidance and review of this work.

I would also like to thank my parents, whose constant support both financially and emotionally has made it possible to get this far. I could not have done this without them.

I also want to thank Dr. Adam Harris and Dr. Ashish Panday, who have shown me what it is to be a PhD student, helped set my standards for research, provided technical advice, and friendship.

Finally, I would like to thank my friends, whose support over the past five years has helped me manage my stress and keep me encouraged. Kenny Hall, Jordan Brown, Tyler Major, Jeremy Sabo, Aaron McClure, Sultana Alimi, and Lauren Johnson; I could not have made it this far without you all.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

In recent years, mobile robotics has grown into an enormous research field for both industrial applications as well as domestic applications. These applications include autonomous vacuum cleaners, pool cleaners, lawn mowers, and self-driving vehicles, just to name a few. In all of these applications, knowledge of the robot's position within its operating environment is of critical importance. To accurately fulfill the requirements of any mobile robotics task, the robot must be aware of its position relative to the objects within the environment to properly interact with them. The robot must also be able to build a map to assist in navigating from point to point within that area in order to interact with the objects within the map. This presents a causality dilemma, in which the robot must know its own position to build a map, but also know the map to track its position. Algorithms that attempt to solve this problem are known as Simultaneous Localization and Mapping (SLAM) algorithms.

Robots use a variety of sensors to estimate their position, such as odometry measurements from the mobile base as well as range finding sensors or visual information from cameras to help identify recognizable landmarks within the environment. In this research, wireless signal strength is used as sensory information to estimate distance between transceivers. If the robot is equipped with a wireless transmitter, as the signal attenuates while traveling to receiver, distance to that receiver can be estimated based on the loss of signal strength. This loss of power between transceivers is represented by the Received Signal Strength Indicator (RSSI). Most wireless transceivers represent RSSI in -dBm (decibel milliwatts) which represents the amount of attenuation or lost power during transmission. If the operating environment of a wireless mobile robot is equipped with transceivers scattered throughout, the estimated distance to each transceiver from the robot can be used to refine the position estimation of the robot.

There are several mathematical models which can be used for estimating distance from RSSI. These models largely do not account for the multipath fading effect. Multipath fading occurs when a signal takes multiple paths from the transmitter to the receiver. Depending on the phase of signals when they arrive together, they can either cause either large attenuation or large amplification of the signal. When the signal is suddenly amplified, the model will not hold and will generate highly erroneous readings. Some models account for multipath, but are intended for long range transmission, such as transmission from a cell tower. Multipath fading effects in short range indoor environments are much more difficult to predict due to the many factors that affect its behavior. For mobile robot operating environments, such as a warehouse or a private home, short range multipath is the most prevalent problem in RSSI distance estimation.

Many solutions exist to mitigate the effects of multipath attenuation. However, most of these rely on some modification to the wireless modulation hardware. Common strategies involve analyzing the signal before the demodulation phase, attempting to identify multipath components and removing them, leaving the primary path in tact. Ultra-Wide band techniques utilize various frequencies for transmitting the same message and observing how each frequency is affected by the transmission, allowing the detection and compensation for multipath interference. Other techniques will utilize antenna arrays or directional antennas to avoid multipath interference. While these methods have proven effective, they all require some specialty modification to the transmission and reception hardware or utilize non-standard communication protocols. This limits the use of these techniques in existing networks, and an increase in implementation costs due to hardware obscurity. For this reason, it would be desirable to find a multipath solution that utilizes common hardware. Most wireless devices provide the user with RSSI values for each transmission. If multipath interference could be filtered or mitigated for RSSI-based ranging techniques, localization could be accomplished in a more inexpensive and accessible manner.

1.1 Objective of this Work

In this work, wireless transceivers will be utilized as landmarks within the EKF SLAM algorithm with a focus on indoor environments. EKF SLAM uses the robot's odometry information along with detected landmarks within the environment to estimate the position and orientation of the robot as well as building a map using range data typically from a laser rangefinder such as LIDAR. EKF SLAM has been proven to be an effective method for SLAM, however, it scales poorly when the number of detected landmarks within an environment becomes large, causing high memory consumption. It also encounters the landmark association problem, where a newly observed landmark may be improperly identified as a previously recorded landmark. Utilizing wireless transceivers as landmarks largely solves these problems, since the number of transceivers is finite, limiting the number of landmarks to the number of devices within the network. Generally, there are not enough network devices where the memory consumption becomes a problem for most modern computers and embedded devices. The landmark association problem is entirely solved as each landmark has a network address associated with it. However, including wireless ranging into the algorithm introduces the multipath problem in exchange for providing solutions for EKF SLAM's problems.

1.2 Contribution

This research provides an in-depth study on signal strength-based localization techniques. Statistical properties for wireless propagation are explored along with mathematical models which relate path loss to distance. Localization techniques for RSSI signals are discussed with respect to their effectiveness in indoor applications.

A novel method for simulating multipath interference on RSSI signals is presented here as well. This method utilizes RSSI data collected over a range of distances within a variety of indoor environment types. This data is then used to train a Markov chain which produces RSSI over distance graphs perturbed with multipath interference. This data is useful when testing RSSI-ranging localization algorithms in environments where the geometry of the map isn't important, but realistic interference is.

Using this simulation method, a multipath-rejection scheme is described for use in EKF range-only (RO) SLAM applications. In EKF RO-SLAM, wireless beacons are often used as landmarks. However, correction of the robot's position estimate may become erroneous when relying on distance measured from RSSI signals undergoing multipath attenuation. Here, the motion information provided by the robot's odometry sensors is used to detect multipath and provide an estimate of the true distance at that time.

1.3 Organization

This dissertation is organized into seven sections. The first chapter provides the motivation for the work presented here. The second chapter provides a background on radio propagation and models. The third and fourth chapters discuss radio signal strength-based localization methods in the static and dynamic cases, respectively. The fifth chapter covers the collection of RSSI data and simulation of RSSI data using the collected data. The sixth chapter describes a novel method for mitigating multipath interference in distance estimation and the utilization of this method in a range-only SLAM algorithm. The final chapter concludes by discussing the shortcomings of current radio propagation models, the strengths and weaknesses of the simulation method, and the results of the multipath mitigation technique in the SLAM simulation.

CHAPTER 2: MOTIVATION

It is a common practice to estimate distance between radio transceivers using the radio signal itself. Often in wireless networks - in particular, wireless sensor networks (WSN) and robotics applications - the information sensed by the wireless device is only useful if the location is known as well. Using the transmission signal itself for distance estimation is practical as it reduces footprint size and device expense by removing the need for additional sensors, such as GPS or ultrasonic. Many methods exist which manage to estimate distance from wireless transmissions, such as: time delay on arrival (TDOA), angle of arrival (AOA), signal strength fingerprinting, and signal strength decay.

Time delay on arrival utilizes a similar technique as used in GPS signals, where the message is time stamped from the transmitter and the receiver observes the time of the message reception. The two time stamps give the time-of-flight of the packet, and when multiplied by the transmission speed, gives the estimated distance between transceivers [10]. Angle of arrival techniques utilize antenna arrays or directional antennas along with TDOA techniques to determine distance and the angle of the transmitted signal. This method is especially useful for not only determining the distance from a transmitter, but also the location. While these techniques are among the more successful, they also have strict hardware requirements. TDOA requires precise and costly oscillators for detecting the delay time in a radio transmission. Additionally, all clocks within the network have the be precisely synchronized, which becomes increasingly more difficult in a purely wireless network, due to the uncertain delay time between the signals being transmitted to synchronize the network clocks (This is achieved more easily when the anchor nodes are connected through a wired connection, which is common in TDOA-based wireless localization kits). Angle of arrival also requires specialty antennas, which increases size and expense. One of the primary philosophies of WSNs is that each device in the network (node) be as small and inexpensive as possible for scalability purposes, as network sizes tend be rather large.

Signal strength-based methods are desirable because of their modest hardware requirements. Signal strength methods at best only require an omni-directional antenna, but is still optional in most scenarios and only serves to increase accuracy. The omni-directional antenna is an antenna which emits a, mostly, uniform radiation pattern, which is useful to assume that the signal strength is uniform at a certain distance from a transmitter, regardless of the orientations of the devices. Signal strength methods are used by measuring the decay of the transmitted signal between devices, in a metric known as received signal strength (RSS) or the received signal strength indicator (RSSI). Both names are used interchangeably in literature but refer to the same metric. RSSI is measured in negative decibel milliwatts (-dBm), where 0 dBm represents a perfectly transmitted signal (physically impossible) and values greater than -100 dBm represent a highly-attenuated signal.

One of the most popular RSSI-based methods for localization is fingerprinting. RSSI fingerprinting involves recording a map of measured signal strengths throughout an area of interest. Once the map has been obtained, the position of a mobile device moving throughout the map can be determined by matching the series of RSSI observation with how those changes fit on the map. While fingerprinting methods are effective, they also have many drawbacks. The mapping process can be quite tedious, and the accuracy of the localization method is heavily tied to the resolution of the map. Increasing map RSSI resolution in turn increases the amount of time it takes to build the map. The map is also highly sensitive to environmental changes. If multiple new wireless devices begin transmitting in the area, the interference will generate faulty measurements on the mobile device, leading to inaccurate matches on the map. RSSI is also affected by reflective and impeding materials within the environment, and if objects composed of materials with these properties change location, the radiation map will change as well, prompting remapping of the area to maintain accuracy.

RSSI can also be used to measure distance as a function of the signal attenuation. If the transmission power, the gains of both the transmitter and receiver antenna, the frequency, and received power are known, the distance can be inferred. As an RF signal propagates through the air, the signal attenuates, allowing the distance to be estimated. However, there are other factors which affect RSSI values apart from the distance between transceivers.

CHAPTER 3: RADIO PROPAGATION

In order to utilize signal strength decay for distance estimation, knowledge of how radio wave propagate through the environment is necessary. Within this chapter, a background of how radio waves propagate and the factors which affect that propagation are discussed.

3.1 Multipath Fading

As mentioned before, there are several factors which can affect signal transmission power loss other than the impedance of the air. Reflective materials can cause the signal strength to behave irregularly, no longer dependent on distance between transceivers alone. While difficult to model, there are several properties that are used to model the way radio waves are affected while propagating through an environment [1] [11]:

- Reflection - Radio waves which "bounce" off a large surface, usually with a size much greater than the wavelength, and change direction. In outdoor applications, the ocean and ionosphere can serve as a reflector for long-range transmissions. In indoor applications, such as office buildings or warehouses, metallic objects will serve as reflectors for short range, high frequency transmissions (WiFi, Bluetooth) [12].

- Absorption - the opposite of reflection; radio waves collide with an object and do not pass through or bounce, but are absorbed into the object.

- Attenuation - The scaling of the amplitude of the signal.

- Dispersion - The spreading of the signal in time.

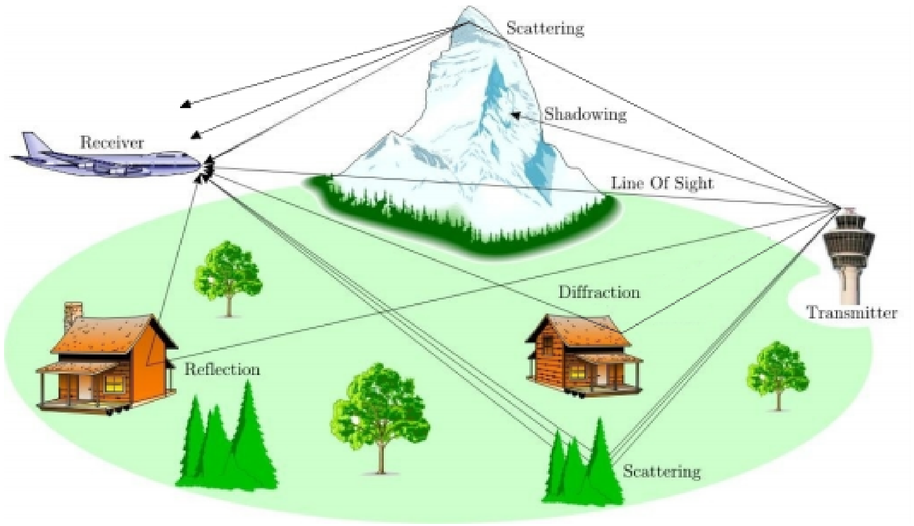


Figure 3.1: Reflections, diffraction, scattering, and fading [1].

- Diffraction - The bending of the wave around the edges of smaller object, usually those comparable in size to the transmitting wavelength. Diffraction may occur when radio waves strike a well-defined obstacle such as a mountain range or the edge of a building. When the wave hits the sharp edge of an object, secondary waves are formed on the other side.

- Refraction - Radio waves can also be diffracted, just as light causes objects to appear distorted as it passes through water due to refraction. In certain mediums, radio waves can pass through and emerge at a different angle than it entered.

- Scattering - Scattering is similar to refraction but tends to be more unpredictable. This occurs when radio waves collide with objects similar in size to the wave-length (or an order of the wave-length) and of an uneven geometry. This causes the signal to scatter into many different directions after a collision. Examples of objects which cause scattering include: lamp posts, forests, street signs, and foliage.

All the aforementioned properties which can affect a transmission channel lead to the fading. Radio wave reflections, scattering, etc. lead to multiple paths for the signal to traverse from transmitter to receiver. The receiver sees the superposition of the multiple signals, which depending on the phase and amplitude, can result in an amplification or attenuation of the received signal. The variation of attenuation or amplification on the received signal is referred to as fading.

Fading can be classified into multiple different categories. One category of fading classification is large-scale and small-scale. Large scale fading affects the signal attenuation over large distances between transmitter and receiver, usually caused by obstacles (e.g., forests, hills, buildings) obscuring the line-of-sight path. The effects of large scale fading are commonly known as shadow fading [13]. Small-scale fading refers to drastic sudden changes in amplitude and phase which occur when the displacement between transceivers is varied by distances less than the wavelength, causing signal spreading and Doppler spread.

Based on coherence time, fading can be described as fast or slow. Coherence time is the minimum time required for the magnitude or phase change of a channel to become uncorrelated from its previous value [14]. Slow fading occurs when amplitude and phase change remains relatively constant over the period of use. This occurs when a large object, such as a mountain, blocks the primary signal path between transceivers. Fast fading occurs when the amplitude and phase change rapidly over the period of use. This form of fading is commonly seen in highly reflective indoor applications.

A primary cause of fast fading is multipath interference. As mentioned before, in reflective environments, radio wave may find multiple paths from transmitter to receiver. This is referred to as multipath interference or multipath fading. Depending on the phase and amplitude of the multiple signals, the result on the overall received signal may be attenuated or amplified, as seen in Figure 3.2.

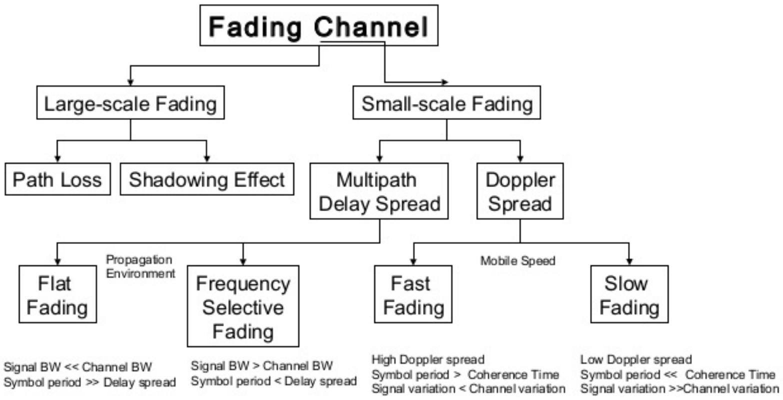


Figure 3.2: Fading channel classifications [2].

While multipath fading is present in nearly all environments, it is particularly noticeable in indoor environments, due to the short transmission distances, numerous objects, and generally reflective materials (metal filing cabinets, steel support beams). These sudden attenuations or amplifications cause indoor distance estimation based on RSSI decay very difficult, as the signal strength is not what is to be expected at a given distance. Most models do not have a method to account for indoor multipath, where the number of paths are numerous; however, some models exist which can account for multipath in outdoor environments, where the number of paths is mostly predictable and the line-of-sight path has not experienced as much attenuation as the multiple paths.

3.2 Multipath Channel Models

Multipath interference is deterministic when all properties of the environment are known [15]. Since only information about the transmitter/receiver hardware and possibly only basic environment details are available, statistical models are used to attempt to capture the multipath effect. In the following sections, multiple models

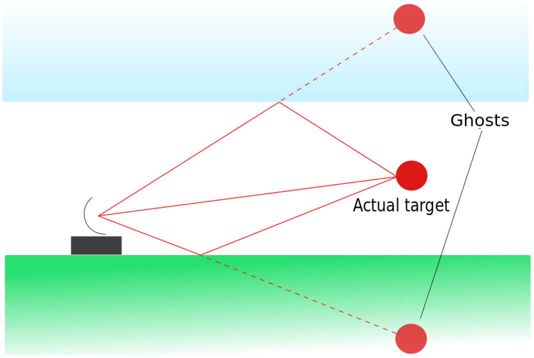


Figure 3.3: An example of multipath propagation [3].

are presented to describe multipath interference under various conditions.

3.2.1 Rayleigh Fading Model

The Rayleigh fading model is a fading model which describes the multipath effect in an environment in which signals are sufficiently scattered. When all signals are scattered to the point where each signal has an equal probability of reaching the receiver from all angles, the central limit theorem states that the channel can be modelled as a complex-valued Gaussian random process due to the delays associated with each path [11]. The envelope amplitude in a highly scattered multipath environment is described by the following Rayleigh distribution function:

$$p(r) = \left\{ \begin{array}{cc} \frac{r}{\sigma^2} e^{\frac{r^2}{2\sigma^2}} & \text{for } r \leq 0 \\ 0 & \text{otherwise} \end{array} \right\} \tag{3.1}$$

Where r is the amplitude of the received signal and $2\sigma^2$ is the pre-detection mean power $(E\{r^2\} = 2\sigma^2)$ of the multipath signal [2]. Note that this distribution models the amplitude and not the received power.

3.2.2 Ricean Fading Model

The Ricean model is an extension of the Rayleigh model, except a dominant line of sight component is present within a high scattering environment. The PDF for a Ricean fading distribution is as follows:

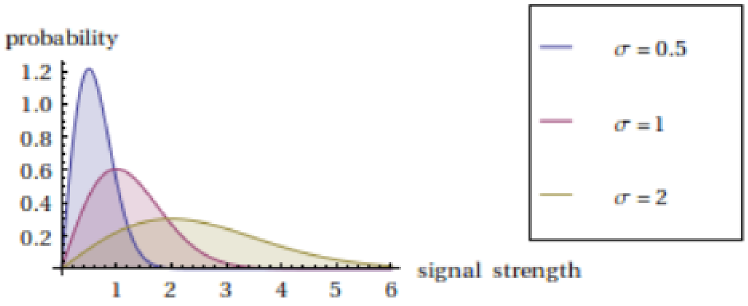


Figure 3.4: Rayleigh fading model for various parameter values [1].

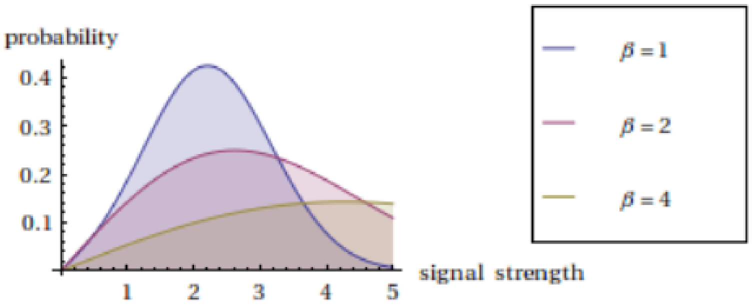


Figure 3.5: Ricean fading model for various parameter values [1].

$$p(r) = \left\{ \begin{matrix} \frac{r}{\sigma^2} e^{\frac{r^2+A^2}{2\sigma^2}} I_0(\frac{rA}{\sigma^2}) & \text{for } r \leq 0 \\ 0 & \text{otherwise} \end{matrix} \right\} \tag{3.2}$$

Where I_0 is the 0^th order Bessel function of the first kind,

$$I_0(x) = \frac{1}{\pi} \int_0^\pi e^{xcos(\theta)} cos(n\theta) d\theta \tag{3.3}$$

And A is the peak amplitude of the line-of-sight path and σ^2 is the RMS value of all the other multipath components, modeled as noise. The Bessel function models the line-of-sight path by generating a decaying sinusoid, and the rest of the PDF models the multipath interference on that signal.

3.3 Path Loss Models

Path loss can be described as the ratio between transmit power and received power. The linear path loss is defined by the following relationship:

$$P_L = \frac{P_t}{P_r} \tag{3.4}$$

Where Pt is the transmitted power, Pr is the received power, and P_L is the path loss of the channel, usually measured in milliwatts. Often, path loss is described in decibel milliwatts (dBm). Taking the log10 of P_L multiplied by 10 yields the dBm representation.

$$PLdB = 10log_{10}(P_L) \tag{3.5}$$

The following expression returns the power from dBm to milliwatts:

$$P_L = 1mW10^{\frac{PLdB}{10}} \tag{3.6}$$

Decibel milliwatts are absolute units since they are referenced to the watt, unlike the decibel (dB), which is a dimensionless unit, usually used to represent a gain factor. dBm is used rather than watts because gain and loss in each stage of an RF system is multiplicative, and when using the logarithm of the gain of each stage, they can be added due to the log rule (log(mn) = log(m) + log(n)), making the calculation much easier to follow. [6] Since dBm refers to a gain in milliwatts, 0 dBm represents 1 mW. A 3dB gain is approximately double the power. Therefore, each 3 dBm is approximately 2 mW of power. A 3dB loss in power is roughly half the power, making -3 dBm correspond to 0.5 mW (the half-power point). For each loss or gain of 3 dBm the power is either doubled or halved, respectively [16].

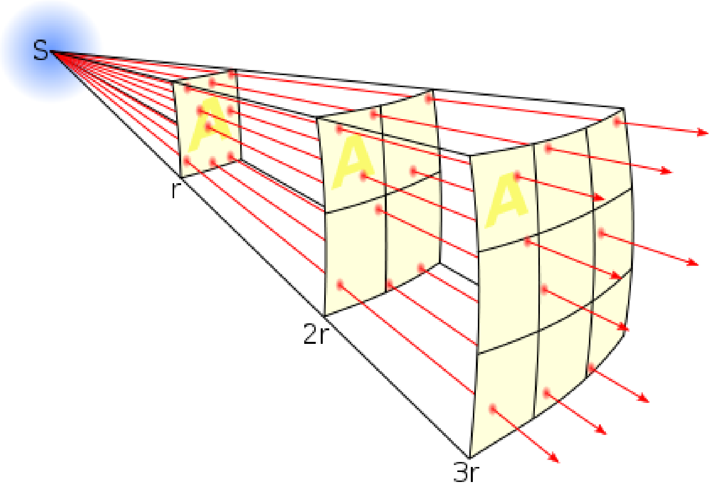


Figure 3.6: Graphical illustration of the inverse square law. The area A receives exponentially less rays each multiple of the distance r [4].

3.3.1 Inverse Square Law

The inverse square law is used to describe the spreading of electromagnetic energy in free space, described by the following relationship:

$$intensity \propto \frac{1}{distance^2} \tag{3.7}$$

This states that a physical quantity or intensity radiating from a point source is inversely proportional to the square of the distance from the source. The energy per unit of area perpendicular to the source will exponentially decrease as the distance increases. For example, an object would receive one fourth the energy it would at a distance d away from the source at a distance $2d$ 3.6. illustrates this concept graphically. This is the basic principle which governs the relationship between estimating distance from RSSI measurements [4] [11].

Relating the inverse square law to the transmission power in milliwatts, the inverse square law can be expressed as:

$$I = \frac{P}{4\pi r^2} \tag{3.8}$$

Where I is the intensity in power per unit area, P is the radiated power, and r is the distance from the point source. Assuming the radiator uniformly radiates power in a spherical pattern around the point, the intensity per unit area is equal to the amount of power in the surface area of the sphere at a distance r. The $4\pi r^2$ denominator comes from the formula for the surface area of a sphere, $A = 4\pi r^2$ [4].

3.3.2 Friis Transmission Equation

The Friis transmission equation is a common model used by RF engineers which gives the power received by one antenna from another a given distance away under ideal conditions. It is a type of free space model, which assumes the region between the transmitter and receiver are free of all objects that might absorb or reflect RF energy. It also assumes that the atmosphere and earth are sufficiently far away such that any reflection from them is negligible. This equation still maintains the intensity-squared distance relationship, but includes additional parameters based on the transmission/reception hardware and frequency.

$$\frac{P_r}{P_t} = G_t G_r (\frac{\lambda}{4\pi R})^2 \tag{3.9}$$

Where G_t and G_r are the transmitter and receiver gains, respectively, λ is the wavelength, and R is the distance between the antennas. The formula gives the ratio of the received power to the output power, expressed in the form of a traditional gain. It can be seen when compared to equation for P_L, the gain will be small, since it is expressed with respect to the attenuation of the signal. The ideal conditions for Friis transmission formula to hold are as follows [17]:

- $R >> \lambda$. If $R < \lambda$ then the power received would be greater than the transmission power, a physical impossibility.

- P_r and P_t should have loss which incurs through the connecting cables. The antennas should be impedance matched with their transmission lines should be conjugate matched.

- The antennas are aligned with the same polarization

- The bandwidth is narrow enough that a single value for the wavelength can be assumed.

3.3.3 Free Space Path Loss Model

While the Friis transmission model includes variables for antenna gain, isotropic radiation from a point, and distance between transceivers; it does not include any other factors for system losses. The free space path loss model includes a variable L to describe these losses. The factor usually takes on a value greater than 1, but if no losses are present, L is set to 1 and the model is equivalent to the Friis transmission model. Some texts refer to the free space path loss model as the Friis free space propagation model [11].

$$P_r(d) = \frac{P_t G_t G_\gamma \lambda^2}{(4\pi d)^2 L} \tag{3.10}$$

Both the free space and Friis transmission models relate the frequency as well as the distance to the path loss equation, making them significantly different from the inverse-square law. While it is a common misconception that the path loss through the medium is frequency dependent, λ's presence in these equations represents the antenna's ability to properly receive the energy at a transmitted frequency. These equations only hold for omni-directional stick antennas, and not dish antennas [2].

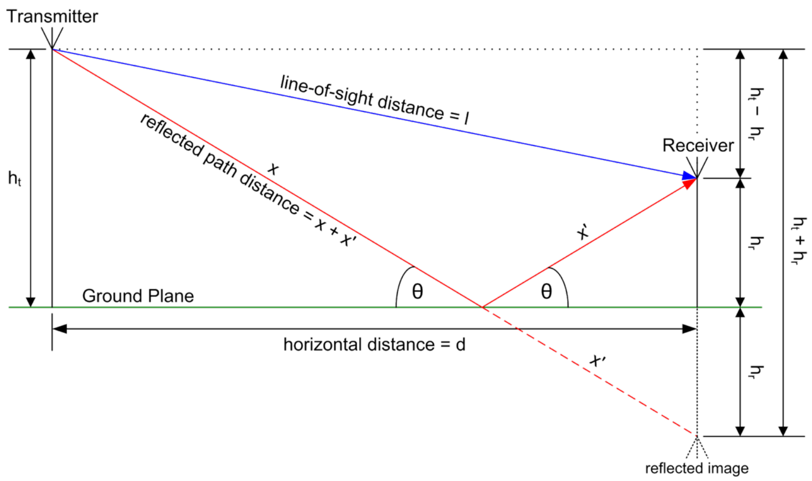


Figure 3.7: The Two-Ray Ground-Reflectance Model [5].

3.3.4 Two-Ray Ground Reflectance Model

The Friis transmission equation and free space path loss models account for transceiver hardware properties but do not account for environment properties, which as discussed earlier, play a large role in radio propagation. Several models attempt to capture these properties by modeling flat fading effects and multipath interference.

The two-ray ground reflectance model is used for outdoor, long-range transmissions, particularly for cellular communications. This model accounts for multipath interference when there is a predominant secondary path in addition to the primary path. Figure 3.7 depicts the geometry of the two-ray ground-reflectance scenario.

The model for the two-ray ground-reflectance model is described by the following expression:

$$P_r(dB) = P_t G_t G_r (h_t^2 h_r^2)/d^4 \tag{3.11}$$

Where P_t is the transmitter's power, G_t and G_r are the transmitter and the receiver antennae gain in dB, ht and hr are the height of the transmitter and receiver antennae in meters, respectively, and d is the distance between transmitters in meters.

While this model has been proven effective in cellular applications, it is not practical for any other environment. In cellular transmissions, there are not many significant reflectors apart from the earth, creating two predominant waves. Other environmental properties will reflect other multipath waves, but the attenuation inflicted on these additional paths is so great, their effects on the primary signal and reflected signal are insignificant. The two-ray model does not hold up whatsoever in indoor applications as there are many paths similar in magnitude to the line-of-sight path rather than two.

3.3.5 Okumura Model

A propagation model which accounts for multipath interference is the Okumura Model. The Okumura model was designed with propagation throughout an urban environment in mind, and includes multiple parameters to represent the type of environment, such as residential, suburban, urban, or open area. The model performs well in cities with many structures, but not tall, blocking structures. The model is presented below:

$$L = L_{FSL} + A_{MU} - H_{MG} - HBG - \sum K_{correction} \tag{3.12}$$

Where L is the median path loss, L_{FSL} is the free space loss, A_{MU} is the median attenuation, H_{MG} is the mobile station antenna height gain, H_{BG} is the base station antenna height gain, and $K_{Correction}$ is the correction gain factor, used to compensate for un-explicitly modeled parameters. The Okumura model has also been extended to incorporate diffraction, refraction, and scattering effects caused by urban structures [18]. A general form of the model is as follows:

$$PL = A + Blogd + C \tag{3.13}$$

Where PL is the path loss, and A,B, and C are variables which model the frequency and antenna height [1]. These variables are often empirically determined, but generic values are available for the environments types mentioned above.

The Okumura models where designed with urban propagation in mind, and do not perform well in other environments, particularly indoor, just like the two-ray ground reflection model. These models are also only function with a limited frequency range, and do not perform well at frequencies beyond 1500 MHz [18].

3.3.6 Log-Distance Path Loss Model

The log-distance path loss model is a propagation model that attempts to model the flat fading effects in any environment. This has become the most ubiquitous model used in indoor distance estimation and serves as the basis for more complex models. Rather than identifying multiple parameters describing the hardware and channel, the log-distance path loss model is an empirical model. The parameters of the model are set according to measured data to capture the characteristics of the environment and the transceiver. The most common form of the model is as follows:

$$RSSI = 10nlog_10(d) + A + X_g \tag{3.14}$$

Where n is the path-loss exponent, d is the distance between transceivers in meters, d_0 is the reference distance, A is the RSSI value at d_0, and X_g is random, zero-mean Gaussian noise. When no fading is present, this variable is 0. In the case of slow fading or shadowing, the variable may have a standard distribution in dB following a Gaussian distribution which sometimes lends this model to be referred to as the log-normal shadow model. In the case of fast fading when a Gaussian distribution does not properly model the noise, X_g should be replaced by either a Rayleigh or Ricean distributed random variable [1].

Naturally, when an RSSI value is measured, the resulting distance can be estimated by re-arranging the equation and dropping the random noise variable as it only serves

Table 3.1: Example path loss exponent values.

Environment	Path-Loss Exponent(n)
Free Space	2.0
Urban Area Cellular Radio	2.7 3.5
Indoor Residential	1.4 1.8

to model the measured RSSI value.

$$d = 10^{(RSSI-A)/10n} \tag{3.15}$$

The value for the path loss exponent, n, is determined based on the environment type. Often, an approximate value can be used if the environment falls into a common type. Table 3.1 contains n values for common operation environments. The path loss exponent may also be solved for by taking sample RSSI values at known distances and reverse solving for n at various points and averaging the results.

The entire model is balanced around the reference distance, d_0, which is often taken as the one meter mark for indoor applications and one kilometer for long range outdoor applications. These values are also selected due to the convenience that diving by one of the unit being reported in d causes the variable to be ignored. This reference distance also determines the value for A, which acts as an offset for the entire model, representing the flat fading induced by the environment.

CHAPTER 4: WIRELESS LOCALIZATION: STATIC CASE

Location estimation is a common application within wireless networks and robotics. While applications involving mobile robots also provide motion, which can be helpful when localizing, in this chapter localization schemes which only involve signal strength are discussed. While these techniques do not take advantage of the motion provided by a mobile robot, they are still applicable to those applications. Here, common localization techniques are described and assessed with their effectiveness in indoor applications and their robustness when it comes to multipath-affected distance estimations.

To track the location of a wireless device networked with other wireless transceivers distributed over an area of interest, distance between devices must be estimated. Several methods have been proposed to determine the distance between devices, including data packet time of flight [19], ultra wide-band transceivers which utilize multiple frequency spectra [20], ultrasonic time of flight [21], RSSI fingerprinting [22], and, most commonly, RSSI distance estimation [23]. Each of the aforementioned techniques have some specific hardware requirements to perform accurate localization, such as directional antennas or multiple antennas, or require precise clock synchronization, such as any time-of-flight based technique. RSSI based estimation techniques have the most flexible requirements, which is typically only an antennae with an omnidirectional radiation pattern for more accurate results. Between the RSSI-based techniques, fingerprinting and RSSI ranging, fingerprinting requires a detailed map of signal strength across the area of interest to be created first, which can be a very tedious task, and the map many not be reliable depending on other interfering wireless devices entering the area. RSSI distance estimation is calculated by measuring the amount of degradation of the signal between the two devices and comparing it to a calibrated model which relates the signal loss to distance [3] [23].

4.1 Trilateration

Possibly the most common method of localizing a wireless node is through the use of trilateration. Trilateration utilizes estimated distance to at least three wireless devices with known locations in order to determine the position of an unknown fourth device. In the most simplistic of scenarios, trilateration is accomplished by using the distances to each device with a known location, referred to as "anchor nodes", as radii for circles representing the possible locations of the device we wish to localize, referred to as the "mobile node", with respect to that individual anchor. When at least three circles are formed around anchor nodes, ideally there should be a position at which each circle overlaps, resulting in the position of the mobile node. Figure 3.a displays the ideal trilateration scenario. In the ideal scenario, the following equations describe how to calculate the trilaterated position [6]:

$$(X_1 - X_4)^2 + (Y_1 - Y_4)^2 = r_1^2 \tag{4.1}$$

Where (X1,Y1) and (X4,Y4) represent the Cartesian coordinates for anchor node 1 and mobile node 4, respectively, and r1 represents the estimated Euclidean distance between the two nodes. This equation can be rearranged to the following form:

$$(X_1 - X_4)^2 + (Y_1 - Y_4)^2 - r_1^2 = 0 \tag{4.2}$$

This equation can be repeated for each remaining anchor node and formed into the following system of equations:

$$\begin{bmatrix} (X_1 - X_4)^2 + (Y_1 - Y_4)^2 \\ (X_2 - X_4)^2 + (Y_2 - Y_4)^2 \\ (X_3 - X_4)^2 + (Y_3 - Y_4)^2 \end{bmatrix} - \begin{bmatrix} r_1^2 \\ r_2^2 \\ r_3^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.3}$$

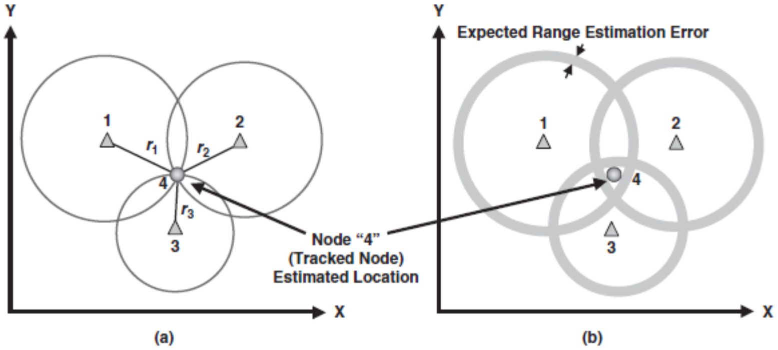


Figure 4.1: Ideal trilateration scenario (a) and trilateration scenario with expected range estimation error (b) [6].

Where the solution to this equation yields the coordinates X4,Y4. However, due to inaccuracies in the equipment, resolution of the RSSI values, fading effects from multipath, or the empirically calculated path loss exponent, distance estimation to each anchor node is usually afflicted by some error. Figure 4.1 shows a trilateration scenario with distance estimation error. Therefore, it is unlikely there is a solution for the equation above. Rather, we need to examine the error terms instead.

$$abs\left(\begin{bmatrix}(X_1-X_4)^2+(Y_1-Y_4)^2\\(X_2-X_4)^2+(Y_2-Y_4)^2\\(X_3-X_4)^2+(Y_3-Y_4)^2\end{bmatrix}-\begin{bmatrix}r_1^2\\r_2^2\\r_3^2\end{bmatrix}\right)=\begin{bmatrix}e_1^2\\e_2^2\\e_3^2\end{bmatrix}=E \qquad (4.4)$$

Where abs is the absolute value function, e2n is the squared error term for the nth anchor node, and E is the error vector. When the equation is in this form, the solution becomes finding the (X_4, Y_4) coordinates which minimize the error vector. This is usually accomplished by utilizing a non-linear least squares method to find that value [24].

4.1.1 Non-Linear Least Squares Trilateration

The non-linear least squares model attempts to minimize the sum of the squares of the distances errors. The sum of the square errors from the equations above can also be expressed as follows:

$$F(\theta) = F(x, y) = \sum_{i=1}^{n} f_i(x, y)^2 \tag{4.5}$$

Where,

$$f_i(x, y) = f_i(\theta) := d_i(\theta) - r_i = sqrt(((x - x_i)^2 + (y - y_i)^2)) - r_i \tag{4.6}$$

And r_i is the estimated distance between (x_i, y_i) and the mobile node, (x, y), and n is the number of anchor nodes. Because $F(\theta)$ is a non-linear function, the first order partial derivative is taken with respect to x and y to linearize it. Differentiating for x yields:

$$\frac{\partial F(\theta)}{\partial x} = 2\sum_{i=1}^{n} \frac{\partial f_i(\theta)}{\partial x} = 2\sum_{i=1}^{n} \frac{\partial d_i(\theta)}{\partial x} \tag{4.7}$$

Similarly, differentiating for y yields:

$$\frac{\partial F(\theta)}{\partial y} = 2\sum_{i=1}^{n} \frac{\partial f_i(\theta)}{\partial y} = 2\sum_{i=1}^{n} \frac{\partial d_i(\theta)}{\partial y} \tag{4.8}$$

To find the solution which minimizes the squared error, we must solve for:

$$\nabla F = 2J(\theta)^T f(\theta) = 0 \tag{4.9}$$

Where ∇F is the linearized vector of partial derivatives, $J(\theta)$ is the Jacobian,

$$J(\theta) = \begin{bmatrix} \frac{\partial d_1(\theta)}{\partial x} & \frac{\partial d_1(\theta)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial d_i(\theta)}{\partial x} & \frac{\partial d_i(\theta)}{\partial y} \end{bmatrix} \tag{4.10}$$

And $f(\theta)$ is the error function vector,

$$f(\theta) = \begin{bmatrix} f_1(\theta) \\ f_2(\theta) \\ \vdots \\ f_n(\theta) \end{bmatrix} \tag{4.11}$$

And n is the number of anchor nodes. The product of $J(\theta)^T f(\theta)$ yields the following matrix:

$$J(\theta)^T f(\theta) = \begin{bmatrix} \sum_{i=1}^{n} \frac{(x-x_1)\partial f_i(\theta)}{\partial x} \\ \sum_{i=1}^{n} \frac{(y-y_1)\partial f_i(\theta)}{\partial x} \end{bmatrix} \tag{4.12}$$

Which gives us the derivatives of $F(\theta)$, the squared error function, with respect to x and y. At this point, Newton's method can be used to iteratively solve for ∇F.

4.1.2 Newton's Method

Newton's method utilizes the following equation to find the roots or zero values of a function:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{4.13}$$

Where $f()$ is a function defined over the set of real numbers, x_n is the current "guess" for the approximation of the roots of the function, and xn+1 is the improved estimate. The function divided by its derivative gives the tangent line of the function, and the root of the function is found by finding the x intercept of the tangent line. The closer the tangent x intercept gets to the true x intercept of the function, the better the approximation of the root of the function. This process can be repeated until a desired threshold accuracy is achieved. Figure 4.2 illustrates the iterative process of Newton's method.

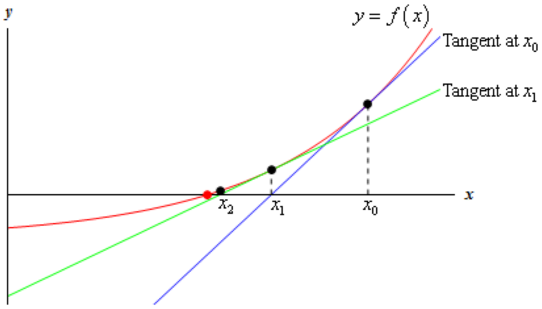


Figure 4.2: Newton's Method.

Newton's method for a vector function is of the following form:

$$x_{k+1} = x_k - \left[J(x_k) \right]^{-1} f(x_k) \tag{4.14}$$

Where x is the current set of parameters, k is current iteration, f is the function we wish to minimize, and J is the Jacobian of that function. For two dimensional trilateration, the function we wish to minimize is the first order derivative of the squared error function. This function can be described as:

$$g(x) = J(\theta)^T f(\theta) = 0$$

Which when applied to Newton's method gives:

$$\theta_{k+1} = \theta_k - \left[J(\theta_k)^T J(\theta_k) \right]^{-1} J(\theta_k)^T f(\theta) \tag{4.15}$$

Where k is current iteration, θ is the set of parameters, $f()$ is the error function vector, and J is the Jacobian of $f()$.

4.1.3 Weighted Non-Linear Least Squares Trilateration

When each observation of the non-linear least squares algorithm are not equally reliable, weights may be assigned to each element of the squared error vector. In most circumstances this is related to the inverse of the variance of the data to reflect the uncertainty. Weights must be within the range of $0 \leq W \leq 1$. The closer the weighting value is to 0, the lesser the impact on the sum of the squared error for the particular measurement being devalued. The non-linear least squares equation can be modified to the following form:

$$F(\theta) = F(x, y) = \sum_{i=1}^{n} W_i f_i(x, y)^2 \tag{4.16}$$

Where W_i is the weight associated with the nth anchor node.

Trilateration using non-linear least squares has proven to be an effective localization method, it is still very susceptible to multipath interference. The weighted method devalues longer range estimations, which is valid still according to the inverse-square law and due to the logarithmic attenuation reduces the possible resolution of from the log-distance model. Multipath will induce unexpected attenuations of the signal, which the inverse-weighting method helps reduce the impact of that particular signal when it comes to satisfying the least squares error; however, multipath interference may also induce an unexpected amplification, which the weighting method will then over-value due to the shorter distance the model will produce.

4.2 Maximum Likelihood Localization

While trilateration shows weaknesses when it comes to indoor multipath attenuation, there are other methods which determine the location of an unlocalized node through the use of anchor nodes in a similar fashion. Here, maximum likelihood estimation is used to determine the location of the an unlocalized node by maximizing the the likelihood of the multipath probability functions described in Section 3.2. This is most commonly applied to the Rayleigh and Ricean distributions which describe multipath fading under non-line of sight and line of sight conditions, respectively. However, since these distributions model the envelope amplitude of the signal and not the average received power, modifications must be made to suit RSSI applications.

4.2.1 Maximum Likelihood Estimation

Suppose there is a sample of independent and identically distributed random values, $x_1, x_2, ..., x_n$, belonging to a probability density function, $f(X|\theta)$, where θ is a vector containing parameters of the distribution. Given the samples X are representative of the population described by $f(X|\theta)$, the values of the parameters θ which maximize the likelihood for each sample are desired.

To find the values of θ, the joint density function must first be obtained for all observations:

$$f(x_1, x_2, ..., x_n|\theta) = f(x_1|\theta)f(x_2|\theta)...f(x_n|\theta) \tag{4.17}$$

This function is then modified to give the likelihood function, which is similar to the joint density, however, θ is viewed as the function's variable and the samples, $x_1, x_2, ..., x_n$, as constant parameters of the function.

$$L(\theta; x_1, x_2, ..., x_n) = f(x_1, x_2, ..., x_n|\theta) = \prod_{i=1}^{n} f(x_i|\theta) \tag{4.18}$$

Often, due to the impracticality of performing operations on products of functions, the natural log of the likelihood function will be used. Taking the natural log of the likelihood function preserves the order of the function, but converts the product into a summation. If a maximum exists for the likelihood function, it will still exist for the same values of θ for the log-likelihood function.

$$\ln L(\theta; x_1, x_2, ..., x_n) = \sum_{i=1}^{n} \ln f(x_i|\theta) \tag{4.19}$$

The MLE estimate of θ can be found from this point.

$$\theta_{MLE} = argmax(\ln L(\theta; x_1, x_2, ..., x_n)) \tag{4.20}$$

Most commonly, the value which maximizes the function is found where the partial derivative of $\ln L(\theta; x_1, x_2, ..., x_n))$ with respect to θ is equal to 0. This method is another motivating factor for using the log likelihood function, as taking the partial derivative of the likelihood function would result in an extremely long expression due to having to use the chain rule for each function in the product.

4.2.2 MLE using Rayleigh and Ricean Fading

The amplitude of the transmitted signal envelope can be described by various fading models introduced in section **??**. The PDFs of these models must be modified to represent the average power. This is accomplished by establishing the following relationship:

$$Y = \frac{X^2}{2} \tag{4.21}$$

or alternatively:

$$X = \sqrt{2Y} \tag{4.22}$$

Where Y is the average power and X is the amplitude. The PDF for average power is expressed by:

$$P(y) = \frac{dx}{dy} = \frac{1}{\sqrt{2}} y^{-\frac{1}{2}} \tag{4.23}$$

Starting with the PDF for average power, x is replaced by the Ricean distribution:

$$P(y) = \frac{df_{Ricean}(x)}{dy} \tag{4.24}$$

Apply chain rule:

$$P(y) = f_{Ricean}(x)\frac{dx}{dy} \tag{4.25}$$

Substitute x for $\sqrt{2y}$

$$P(y) = f_{Ricean}(\sqrt{2y})\frac{d\sqrt{2y}}{dy} \tag{4.26}$$

Which yields the following PDF for average power:

$$P(y) = f_{Ricean}(\sqrt{2y})\frac{1}{\sqrt{2y}} \tag{4.27}$$

When expanded contains the following:

$$P(y) = \frac{1}{\sigma^2}e^{\frac{2y+A^2}{2\sigma^2}}I_0(\frac{\sqrt{2y}A}{\sigma^2}) \tag{4.28}$$

From this point, the PDF must be modified once more. The goal is to obtain a likelihood function where the input is the distance and the RSSI values are the parameters. To do this, the power variable, Y can be related to distance based on the Friis transmission equation from Section **??**. Since maximum likelihood estimation will attempt to estimate the parameters of the transmission equation, it can be simplified to an exponential decay function with two parameters:

$$P = Cd^{-n} \tag{4.29}$$

Where P represents the path loss, C is a constant containing the transmitter and receiver antenna gains and frequency, d is the distance between transceivers, and n is the attenuation rate. The Ricean distribution models multipath interference with two parameters, A and σ^2 which model the amplitudes of the primary, line of sight path and the interference from all other paths, respectively. These two parameters are replaced by two reduced Friis transmission equations to represent the averaged power from each path.

$$A = ad^{-b} \tag{4.30}$$

$$\sigma^2 = \alpha d^{-\beta} \tag{4.31}$$

Plugging in for the following parameters gives a conditional distribution function for received power given the distance:

$$P(y|d) = \frac{1}{\alpha d^{-\beta}} e^{\frac{2y+a^2d^{-2b}}{2\alpha d^{-\beta}}} I_0(\frac{\sqrt{2y}ad^{-b}}{\alpha d^{-\beta}}) \tag{4.32}$$

From this PDF the ML estimates of the parameters $\theta = [a, b, \alpha, \beta]$ may be obtained for RSSI measurements and distances $y_k, d_k, fork = 1, 2, ..., n$. The likelihood function is as follows:

$$L(\theta; d_k, y_k) = \prod_{k-1}^{n} \frac{1}{\alpha d_k^{-\beta}} e^{\frac{2y_k+a^2d_k^{-2b}}{2\alpha d_k^{-\beta}}} I_0(\frac{\sqrt{2y}ad_k^{-b}}{\alpha d_k^{-\beta}}) \tag{4.33}$$

To find the parameters of θ which maximize the likelihood of the measurements, d_k, y_k, the partial derivative with respect to each parameter in θ must be taken and set equal to 0, as the slope will be 0 at the peak of the likelihood curvature.

Once the parameters have been estimated, the likelihood function can be modified for the location of the unlocalized node. The distance parameter which was taken as a conditional for the PDF can be expanded using the euclidean distance:

$$d_k = sqrt(x_k - x_u)^2 + (y_k - y_u)^2 \tag{4.34}$$

Where x_k and y_k refers to the x and y known location of the kth anchor node, respectively, and u indicates the parameter of the likelihood being estimated. By substituting equation 4.34 into 4.33, and taking the partial derivative with respect to each parameter, x_k, y_k. The following expressions yields the most likely values for x_k, y_k, thus giving the estimated location of the unlocalized node.

$$\frac{\partial l(x_u, y_u)}{\partial x_u} = \sum_{k=1}^{n_s} \left\{ \begin{array}{c} \frac{-\beta(x_{rk}-x_u)}{(x_{rk}-x_u)^2+(y_{rk}-y_u)^2} + \\ \frac{y_k \beta}{\alpha}(x_{rk} - x_u)[(x_{rk} - x_u)^2 + (y_{rk} - y_u)^2]^{\frac{\beta}{2}-1} + \\ \frac{a^2}{\alpha}(-b + \frac{\beta}{2}(x_{rk} - x_u)[(x_{rk} - x_u)^2 + (y_{rk} - y_u)^2]^{-b+\frac{\beta}{2}-1} - \\ \frac{I_1(\frac{\sqrt{2y_k} a d_K^{-b}}{\alpha d_k^{-\beta}})}{I_0(\frac{\sqrt{2y_k} a d_K^{-b}}{\alpha d_k^{-\beta}})} \frac{a\sqrt{2y_k}}{\alpha}(-b + \beta))(x_{rk} - x_u)[(x_{rk} - x_u)^2 + (y_{rk} - y_u)^2]^{\frac{-b+\beta}{2}-1} \end{array} \right\} = 0 \tag{4.35}$$

$$\frac{\partial l(x_u, y_u)}{\partial y_u} = \sum_{k=1}^{n_s} \left\{ \begin{array}{c} \frac{-\beta(y_{rk}-y_u)}{(x_{rk}-x_u)^2+(y_{rk}-y_u)^2} + \\ \frac{y_k \beta}{\alpha}(y_{rk} - y_u)[(x_{rk} - x_u)^2 + (y_{rk} - y_u)^2]^{\frac{\beta}{2}-1} + \\ \frac{a^2}{\alpha}(-b + \frac{\beta}{2}(y_{rk} - y_u)[(x_{rk} - x_u)^2 + (y_{rk} - y_u)^2]^{-b+\frac{\beta}{2}-1} - \\ \frac{I_1(\frac{\sqrt{2y_k} a d_K^{-b}}{\alpha d_k^{-\beta}})}{I_0(\frac{\sqrt{2y_k} a d_K^{-b}}{\alpha d_k^{-\beta}})} \frac{a\sqrt{2y_k}}{\alpha}(-b + \beta))(y_{rk} - y_u)[(x_{rk} - x_u)^2 + (y_{rk} - y_u)^2]^{\frac{-b+\beta}{2}-1} \end{array} \right\} = 0 \tag{4.36}$$

The same principle can be applied to the Rayleigh distribution as well. The Rayleigh distribution is a reduced form of the Ricean distribution, obtained by removal of the line of sight component. By setting A from Equation 4.32 to 0, we obtain a Rayleigh distribution:

$$P(y) = \frac{1}{\sigma^2} e^{\frac{-y}{\sigma^2}} \tag{4.37}$$

By substituting the Friis transmission equations from 4.31 for the envelope amplitude formula:

$$P(y) = \frac{1}{\alpha d^{-\beta}} e^{\frac{-y}{\alpha d^{-\beta}}} \tag{4.38}$$

Maximum likelihood (ML) estimates for position based on distance are determined through the same method for the Ricean fading model. This process may also be applied to Gamma, Nagakami, or any other PDF fading model. However, Ricean and Rayleigh are the most accepted models, so the others are not explicitly defined here. Results from this method conducted by Tanikawara et al. and Hara et al. indicate a positioning resolution no greater than 0.5 meters under ideal circumstances and approximately 4 meters under highly reflective conditions [25] [26].

MLE has also been applied to other models, such as the Gamma distribution [25] and modified forms of the log-distance model [27]. While these methods are interesting and prove more effective than basic trilateration using a standard attenuation model, such as Friis or log-distance, they still produce resolution no better than half meter in best case scenarios, limiting their usefulness in precision applications.

A large point of contention in localization techniques for indoor environments utilizing distributions is which model properly describes the multipath effect in indoor, short range measurements. These models are most often applied in long-range transmissions, where multipath scattering is caused by building and forests. When applied to indoor applications, the same principles should theoretically be true. Some papers claim that Ricean fading properly models indoor line of sight transmissions. The work by Tanikawara et al [25]. used the ML localization approach with both Ricean and Rayleigh models. Their results conclude that the Ricean-based distribution provided most accurate results when used in an experiment conducted in an open room. The higher accuracy was attributed to the line of sight component. The work by Hara et al [26]. conducted a similar ML localization experiment using an exponential distribution loosely based on the Rayleigh distribution, concluding that the power decay followed an exponential distribution, which could be said about both Rayleigh and Ricean fading. The work by O'Hollaren and Shell [28] matched RSSI data to both Ricean and Rayleigh distributions to determine if two transmitters where in line of sight, concluding a noticeable change when the line of sight component of the Ricean distribution was not present. However, the line of sight component was still present when thin partitions where place between transceivers, causing the authors to conclude the environment material has a strong effect on the model accuracy. The work by Puccinelli and Haenggi [15] explores the probabilistic fading models in relation to the deterministic nature of multipath fading in indoor environments. Their results show that despite a line of sight component existing in the environment they tested in, it did not stand out from the other paths due to the geometry of the environment. Ultimately, these distributions do not hold for every type of indoor environment. Multipath interference is deterministic based on the geometry and materials of the environment, and without that information a reliable statistical model cannot exist.

CHAPTER 5: WIRELESS LOCALIZATION: DYNAMIC CASE

While localization techniques have been proven effective for static cases involving distance estimation based on signal attenuation purely, they still have limited resolution due to the lack of attenuation models which account for multipath interference, especially those which describe attenuation well indoors. For mobile applications involving a robotic platform, the motion provided from the odometry may be used to help determine the positions of an unknown node. This motion may also eliminate the need for anchor nodes as well, as the known motion can be used to localize other nodes within the network.

5.1 Particle Filter

Particle filters are a genetic-type algorithm used for state estimation. Particle filters represent the state by generating multiple "particles", in which each particle represents a hypothesis of the current state values. Each iteration, the particle goes through a prediction step, where each hypothesis is updated according to a system model. In the next step a measurement is taken. An expected measurement value is then computed according to each predicted hypothesis value and an error is computed. A re-sampling phase is then executed based on a probability distribution computed from the error values for each particle. If a particle filter contained N number of particles, M particles would be pseudo-randomly selected according the error distribution. As the filter iterates, the hypothesis with incredibly inaccurate values will have a low chance of selection, and will be eliminated over time, while those with values closer to that of the true state of the system will have lower error values, giving a higher change of selection each iteration.

The particle filter is defined in Algorithm 1. Here, all particles, m, are contained within the set, X_t. At the beginning of the algorithm, X_t and $\bar{X}_t$ are initialized as empty sets, which are populated as particles are drawn from the previous set, X_{t-1}. The prediction is performed in the sample process, where x_t is calculated based on x_{t-1} and u_t. The Bayesian posterior is also calculated for each particle, $p(x_t|u_t, x_{t-1}^{[m]})$. The weights are also calculated for each particle for the measurement vector z_t. The importance weight is assigned from the distribution $p(z_t|x_t)$, which uses the measurement vector and the calculated posterior for x_t. The set $\bar{X}_t$ is then populated with M hypothesis, x_t, and associated importance weights, w_t. The next step re samples M particles from $\bar{X}_t$ according to the importance weights assigned.

Algorithm 1 The Particle Filter (X_{t-1}, z_t)[29]

```
$\bar{X}_t = X_t = \emptyset$
for $m = 1$ to $M$ do
    sample $x_t^{[m]}$ $p(x_t|u_t, x_{t-1}^{[m]})$
    $w_t^{[m]} = p(z_t|x_t^{[}m])$
    $\bar{X}_t = \bar{X}_t + < X_t^{[m]}, w_t^{[m]}$
for $m = 1$ to $M$ do
    draw $i$ with probability $\approx w_t^{[}i]$
    add $x_t^{[i]}$ to $X_t$
return $X_t$
```

This algorithm's process is graphically depicted in Figure 5.1, which describes the process of localizing a mobile robot within known map. In the first image, (a), a map of an interior building is shown, filled with uniformly scattered black dots. Each dot represents a particle or hypothesis of the state, or position, of the robot. Over time, as the robot moves, the measured motion is applied to each particle. The particles "condense" over iterations as the erroneous particles' predicted measurements generate a low importance weight when compared to the sensor measurements. In Figure 5.1 image b, it can be seen there are two primary particle clusters. As the robot has moved down the hallway, the symmetrical appearance of the hallway has introduced two nearly equally likely position for the robot to exist. The particles for representing the likely incorrect location are eliminated as the robot travels into a room with a geometrically unique structure compared to the room opposite on the map in Figure 5.1:c.

The particle filter's strength lies in its ability to represent any kind of distribution. The particle filter also performs reasonably well with non-linear systems. The drawbacks of using a particle filter are the convergence time and large number of particles. If the initial condition of the state is not well known, or is uniform over a large space, the filter will take some time to converge before the state estimate is reliable. Also depending on a large solution space, it may take a large number of particles to represent the distribution well. If the space is large and the initial condition is unknown, a large number of particles may also be required such that the convergence rate does not take an inordinate amount of time.

5.2 Probability Grids

Probability grid is another technique that has been used extensively for localization applications in both robotics and wireless sensor networks. Probability grids function by the discretization of a continuous probability density function into grid space, where each space in the grid represents some input value to the probability function.

For the case presented in [8], wireless beacons are localized using probability grids. In that work, an empirical conditional probability density function, $p(r_s|m_i)$, is created based on gathered values. This function assigns a probability for a measured RSSI value, m_i, given a certain distance. Each point on the gird is assigned a probability from the following equation:

$$\gamma_s = \frac{p(r_s|m_i)}{2\pi r_s} \tag{5.1}$$

Where r_s is the range from the robot to the beacon, and $r_s = ||x_b - x_s||$ and x_s is the

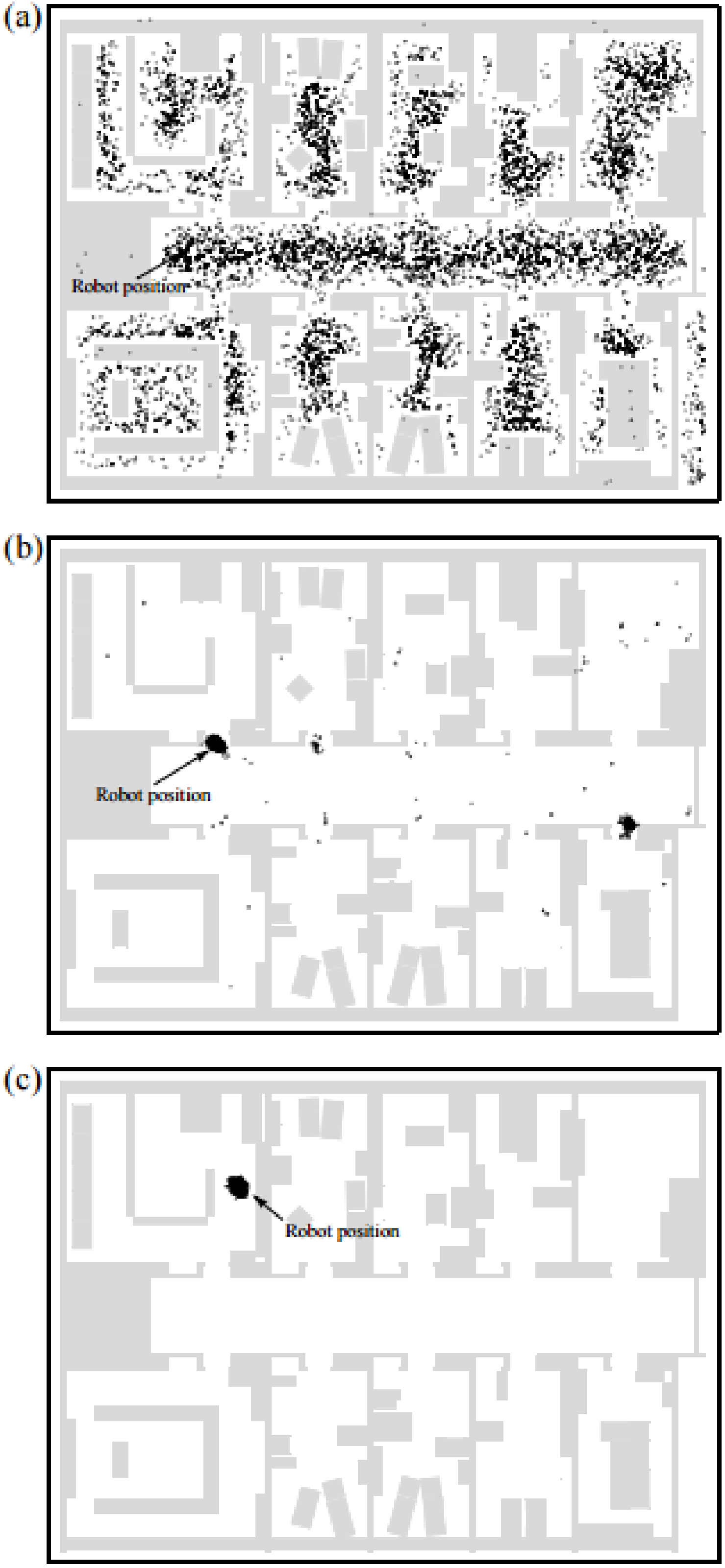


Figure 5.1: Particle filter for mobile robot localization [7].

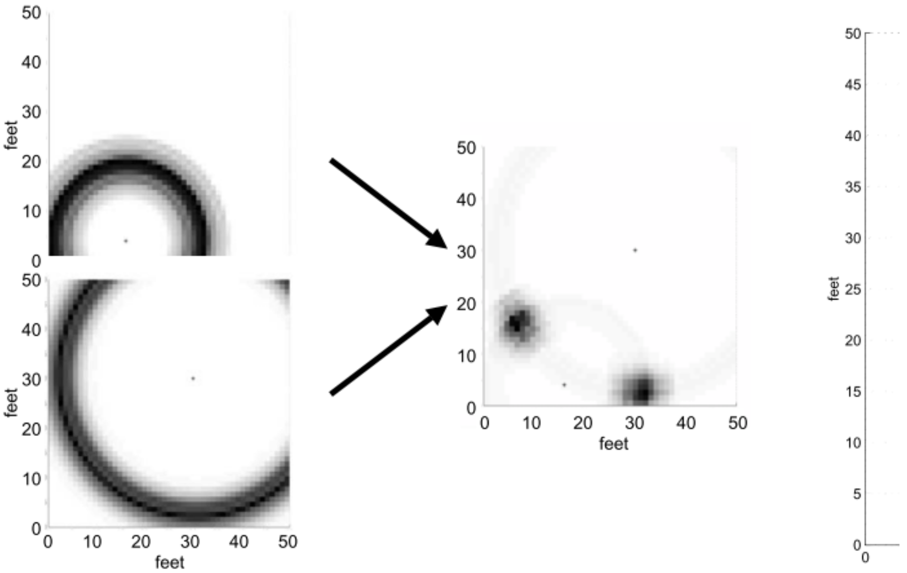


Figure 5.2: Probability grid for localizing wireless beacons [8].

center grid location in the world coordinate frame and x_b is the location of the robot. After assigning a γ value to each grid location, the entire grid is normalized using:

$$P_s = \frac{\gamma_s}{\alpha} \tag{5.2}$$

Where $\alpha = \sum_{N_s}^{s=1} \gamma_s$. This creates proper probability values in the newly created grid by normalizing over the sum of all values within the grid.

After the first grid is created, a ring appears expressing all locations where the beacon may reside statistically. Each time the robot moves, its position is updated and a new probability gird is calculated. The new grid and the grid from the previous time step are added together and normalized using the process described above. As each ring of probability is superimposed on top of each other, the overlapping regions develop higher probability of containing the location of the beacon. This process is described in 5.2.

5.3 Simultaneous Localization and Mapping

The particle filter and probability grid applications presented here are useful when either the location of the robot is well known, or the locations of anchor nodes are well known. However, if neither are known a prior, then a simultaneous localization and mapping (SLAM) algorithm can be used to determine both. While SLAM algorithms have been adapted to utilize wireless distance measurements, it is prudent to discuss the general applications initially, then introduce their applicability as it pertains to wireless beacon localization.

SLAM algorithms are designed to provide a device, usually a robot, with a map of its environment and its position within that map. Many applications, such as path planning, are dependent on this information. Many consider SLAM to be the first step in achieving true autonomy due to the amount of other applications which are dependent on it. Currently, in most industrial and home automation applications, various modifications are made to the environment to provide assistance in localization of a robot. In industrial applications, magnetic strips and lines on the floors provide a track for autonomous robots to follow. Visual landmarks are also often installed within an area of operation for an autonomous robot to identify, and, using previous knowledge of the landmark's position and estimated distance to that landmark, localize itself within that environment. SLAM algorithms attempt to remove the need for pre-configured environments by localizing themselves and building the map without any prior information about the environment.

The SLAM problem is a very active research field and still a largely unsolved problem. The SLAM problem entails tracking the position and orientation of a robot within a map, while also exploring and building that map. This presents a causality dilemma; how can a robot be localized without a map, and how can a map be built if the one building the map is uncertain of its own position. SLAM operates on the notion of no known or very little a priori information. Because of this, uncertainties concerning both the position and orientation of the robot affect the certainties of building the map, and the uncertainties within the map affect the correction methods used to counteract the uncertainties in the robot's motion. In most robotic localization applications, range sensory information can be used to correct noise and errors from the robot's motion, by comparing expected range value to objects in the map with measured range values within the real world. However, when the map information is uncertain, and the robot's motion is uncertain, it becomes quite difficult to keep errors from accruing over time and rendering the entire application useless.

5.4 SLAM Techniques

There are many different approaches to the SLAM problem, many of them differing based on the type of sensors being used and the filtering method being used. Most commonly, range finding sensors such as LIDAR have been used, both for populating the map with obstacles as well as the identification of landmarks. More recently, due to the increase of processing speed and parallel computation, the use of cameras and depth images have been used for landmark identification and map construction. The filtering techniques behind the SLAM algorithms are how each algorithm attempts to handle the uncertainty in landmark identification, map construction, and the robot's motion. Graph SLAM, Extended Kalman Filter SLAM, GMapping [30], FastSLAM 1.0 and 2.0 [31] are a few of the more popular SLAM algorithms, just to name a few. What is interesting about each of these algorithms is that they each use a different filtering method to handle uncertainties. Each method brings its own set of strengths and weaknesses, however [31] [32].

5.5 EKF SLAM

The Extended Kalman Filter (EKF) SLAM is one of the most common and reliable implementations of SLAM. It can be adapted for multiple types of landmarks, can be quickly computed, and has a relatively straight-forward implementation. As the name implies, it uses the EKF at its core for handling the uncertainties associated with SLAM. EKF is a recursive Bayesian filter which models each input and prediction with a Gaussian noise. To understand how EKF SLAM functions, the EKF must first be introduced. The following sections begin by introducing the regular Kalman filter, then the extended Kalman filter, and finally followed by the particulars of EKF SLAM.

5.5.1 The Kalman Filter

The Kalman filter is an algorithm which uses a series of noisy measurements over time to estimate the state of unknown variables of a system. The Kalman filter models the noise of the signals as a Gaussian distribution, and each iteration the state variables are updated based on the certainty of the current measurement and the previous state variable. The Kalman filter can be described as having two phases: a prediction phase and a measurement phase. During the prediction phase, the state variables are updated according to the state transition model, which predicts what the variables should be during the next time step, and the error covariance matrix is updated, which reflects the confidence of each state variable estimate. The measurement phase attempts to correct the prediction of the update phase from sensor measurements and accounting for the noise from the sensor and the state transition [33].

5.5.1.1 The Prediction Phase

There are two equations for the update phase, one for each variable which represents the state of the filter, the first being the state variable vector, x:

$$x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k \tag{5.3}$$

Where $x_{k|k-1}$ is the estimate of the state variables at time step k, given the previous estimate at $k-1$, F_k is the state transition matrix at time step k, $x_{k-1|k-1}$ is the state variable last time step, B_k is the control input model, and uk is the control input. The B matrix and u vector are optional, given a control input is available:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{5.4}$$

The Pk|k-1 is the covariance matrix at step k given the previous estimate at k-1, Pk-1|k-1 is the covariance matrix of the previous iteration, and Qk is the covariance matrix of the process noise, which represents how much noise is generated in the transition between states [34].

5.5.1.2 The Measurement Phase

The measurement phase begins by calculating the error between the estimation and the state variable:

$$y_k = z_k - H_k x_{k|k-1} \tag{5.5}$$

Where yk is the error vector, zk is the measurement vector, and Hk is the observation model, which maps the state variable vector into the space of the measurement vector. The innovation covariance matrix, Sk, is calculated next, which relates the covariance of the state variables to the measurement vector and accounts for the noise of the measurement vector:

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{5.6}$$

Where Rk is the covariance matrix of the noise of the measurement vector, zk . The Kalman gain, Kk, is then determined by multiplying the current state covariance matrix, the observation model, and the innovation matrix:

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{5.7}$$

The Kalman gain applies an optimal weight to the measurement error, which is then used to correct the current state estimation, $x_{k|k}$:

$$x_{k|k} = x_{k|k-1} + K_k y_k \tag{5.8}$$

Finally, the covariance matrix, Pk|k, is then updated, by scaling the covariance values associated with the observation model by the value determined by the Kalman gain through the following equation:

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{5.9}$$

Where I is the identity matrix. As this filter runs through multiple iterations, it estimates the state variables defined in x though the transitions defined by the state transition model and the measurement vector. The covariance matrix is constantly updated to represent the variance of each state variable, which also represents the confidence in the current measurement. Algorithm 2 shows the entirety of the Kalman filter [34].

Algorithm 2 The Kalman Filter ($x_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k$)

```
KF_Prediction:
$x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k$
$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
KF_Measurement:
$y_k = z_k - H_k x_{k|k-1}$
$S_k = H_k P_{k|k-1} H_k^T + R_k$
$K_k = P_{k|k-1} H_k^T S_k^{-1}$
$x_{k|k} = x_{k|k-1} + K_k y_k$
$P_{k|k} = (I - K_k H_k) P_{k|k-1}$
return $x_{k|k}, P_{k|k}$
```

5.5.2 The Extended Kalman Filter

The Kalman filter is a linear state estimator, so it does not hold for non-linear systems, which very few systems are linear. However, many modifications of the

Kalman filter exist to account for non-linear systems, such as the Extended Kalman filter, Unscented Kalman filter, and Hybrid Kalman filter. The most common of these non-linear forms is the Extended Kalman filter. The Extended Kalman filter accounts for non-linearities by linearizing the system through taking the first-order Taylor series expansion of the state transition and observation models [29]. This process gives the Jacobian of F and H, which contain the first order derivative of a vector function of several variables. The Jacobians are used to define the curvature of the linearized function models, describing how the probability mass is spread during the prediction and measurement phases. The Jacobians are defined as follows:

$$F_{k-1} = \frac{\partial f}{\partial x}|_{x_{k-1|k-1},u_k} \tag{5.10}$$

$$H_k = \frac{\partial h}{\partial x}|_{x_{k-1|k-1}} \tag{5.11}$$

The state transition matrix and observation matrix are replaced by nonlinear functions which describe the state transitions and observation models, f and h, in the prediction and measurement phases. From these functions the Jacobians are calculated, as the values of F and H will change depending on the time of the linearization. The extended Kalman filter in its complete form is shown in Algorithm 3. In the section covering EKF SLAM, examples of the f and h functions can be found along with Jacobian calculations based on these functions [29].

5.6 The EKF SLAM Algorithm

The EKF SLAM algorithm is one of the most traditional solutions for the SLAM problem today. It is one of the earliest algorithms to introduce the landmark concept, in which particularities in the environment are recorded and their positions are used to correct odemetric errors in the robot by estimating distance to the landmark in

Algorithm 3 The Extended Kalman Filter $(x_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k)$

```
EKF_Prediction:
$x_{k|k-1} = f(x_{k-1|k-1}, u_k)$
$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k$
EKF_Measurement:
$x_{k|k-1} = h(x_{k-1|k-1}, z_k)$
$S_k = H_k P_{k|k-1} H_k^T + R_k$
$K_k = P_{k|k-1} H_k^T S_k^{-1}$
$x_{k|k} = x_{k|k-1} + K_k y_k$
$P_{k|k} = (I - K_k H_k) P_{k|k-1}$ return $x_{k|k}, P_{k|k}$
```

each iteration in which it is observed. Odometry readings come from either wheel encoders on the robot counting rotations to measure velocity, accelerometers, gyros, and magnetometers for measuring inertial information, or both. Odometry readings are the base sensory information used in nearly all SLAM algorithms. If odometry sensory reading were reliable enough, there would be no need for SLAM and dead reckoning approaches could be used for localization entirely. However, due to wheel slippage and noisy readings from inertial sensors, odometry data is not reliable over long periods of time. The small errors will accrue over time into a large error, rendering the information useless. The EKF comes into play by utilizing landmarks as the input to the measurement vector to correct the accrued error from odometry. The algorithm tracks the position and orientation of the robot in the state variables vector, along with the (x, y) positions of the recorded landmarks. The prediction phase utilizes the odometry readings as inputs to the control vector and corrects these readings in the measurement phase with the estimated distance to the observed landmarks. Based on what are usually empirically gathered statistics, the EKF weights these predictions and measurements appropriately so the near-optimal estimate is acquired. The sequence in Figure 5.3 depicts this process, where robot (triangle) records the locations of three landmarks (stars) relative to its position within the map. The robot then moves some distance, measures distance to the recorded landmarks, then uses that information to correct the inaccuracies in the odometry reading.

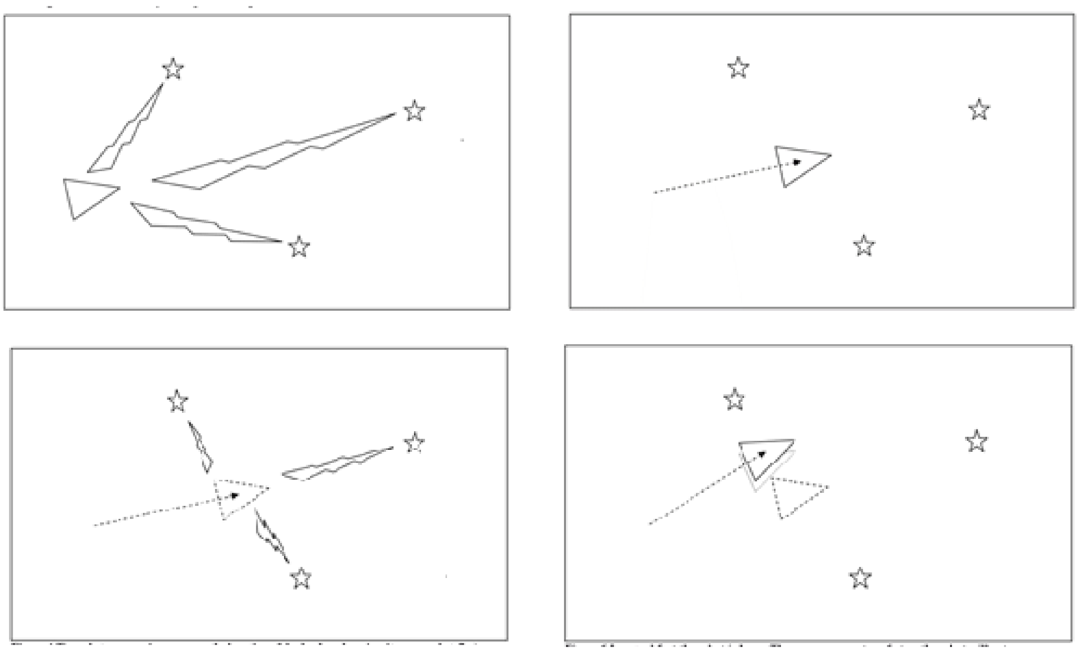


Figure 5.3: The Odometry-Landmark Correction process [9].

Apart from the EKF determining the system state estimates, there are a few other processes which comprise EKF SLAM. A large aspect of most landmark-based SLAM algorithms is the data association problem. This occurs when an observed landmark is mis-associated with a similar, but different, recorded landmark. Due to the fact that landmarks are usually determined from laserscan data or unique data sets within images, it is possible to mistake one of these for another data set with similar features. When this occurs the state variable estimations can become highly erroneous due to having the odometry measurements over-corrected for an incorrect distance measured to a previously recorded landmark. In the following sections, the EKF matrices are covered with respect to EKF SLAM. To be thorough, each matrix which must be configured for this application has its contents discussed in detail.

5.6.1 The State Vector

The state vector contains the parameters of the system that are important to the rest of the system and a few others that might be necessary for the algorithm to function. In the case of two-dimensional EKF SLAM, the state variables being estimated include the following: the robot's (x, y) pose, the robot's orientation, θ, and the (x, y) coordinates for each recorded landmark. Since the number of landmarks is not static, appending landmarks to this vector requires each matrix in the entire algorithm to expand with it, causing an exponential increase in memory usage. This is one of the largest problems with EKF SLAM. The state vector appears as such:

$$x = \begin{bmatrix} x_r \\ y_r \\ \theta_r \\ x_1 \\ y_1 \\ \vdots \\ x_n \\ y_n \end{bmatrix} \tag{5.12}$$

Where x_r, y_r is the position of the robot, θ_r is the orientation of the robot, x_i, y_i is the position of the ith observed landmark, and n is the total number of recorded landmarks. These position values are usually stored in meters or millimeters, and the orientation is usually stored in degrees or radians. The units selected for these parameters must remain consistent throughout the entire filter.

5.6.2 The Covariance Matrix

As the name indicates, the covariance matrix contains the covariances for each state variable with each other. The diagonal values of the covariance matrix contain the variance of each state variable. As the filter iterates, this value essentially reflects the system's confidence in the estimation. If the system is well modeled, this value will decrease over time as the system converges. Initially, the diagonals of this matrix need to be initialized with some value to avoid singularities in the calculation, despite

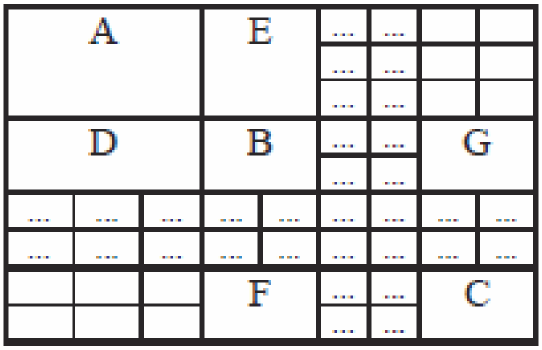


Figure 5.4: Sub-sections of the Covariance Matrix [9].

in the initial state there is no reason to assume uncertainty about the position. The following equation depicts the contents of the covariance matrix.

$$P = \begin{bmatrix} Var(x_r) & Cov(x_r,y_r) & Cov(x_r,\theta_r) & Cov(x_r,x_1) & Cov(x_r,y_1) & \dots & Cov(x_r,x_n) & Cov(x_r,y_n) \\ Cov(y_r,x_r) & Var(y_r) & Cov(y_r,\theta_r) & Cov(y_r,x_1) & Cov(y_r,y_1) & \dots & Cov(y_r,x_n) & Cov(y_r,y_n) \\ Cov(\theta_r,x_r) & Cov(\theta_r,y_r) & Var(\theta_r) & Cov(\theta_r,x_1) & Cov(\theta_r,y_1) & \dots & Cov(\theta_r,x_n) & Cov(\theta_r,y_n) \\ Cov(x_1,x_r) & Cov(x_1,y_r) & Cov(x_1,\theta_r) & Var(x_1) & Cov(x_1,y_1) & \dots & Cov(x_1,x_n) & Cov(x_1,y_n) \\ Cov(y_1,x_r) & Cov(y_1,y_r) & Cov(y_1,\theta_r) & Cov(y_1,x_1) & Var(y_1) & \dots & Cov(y_1,x_n) & Cov(y_1,y_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Cov(x_n,x_r) & Cov(x_n,y_r) & Cov(x_n,\theta_r) & Cov(x_n,x_1) & Cov(x_n,y_1) & \dots & Var(x_n) & Cov(x_n,y_n) \\ Cov(y_n,x_r) & Cov(y_n,y_r) & Cov(y_n,\theta_r) & Cov(y_n,x_1) & Cov(y_n,y_1) & \dots & Cov(y_n,x_n) & Var(y_n) \end{bmatrix} \tag{5.13}$$

The P matrix can be broken down into various sub-sections, clarifying the meaning of the matrix's contents. Figure 8 indicates the sub-sections of the P matrix.

Section A of the P matrix contains the 3x3 covariance matrix of the robot's position and orientation. Section B holds the covariance of the first recorded landmark. Section D contains the covariance of the robot's state with the first landmark, while Section E contains the transpose of D. Section F contains the covariance of the first landmark with the nth landmark, and G is the transpose of F. Finally, section C contains the covariance of the nth landmark.

5.6.3 The Prediction Model and the Jacobian

The prediction model in EKF SLAM is a non-linear function which estimates the state variables for the next iteration. The prediction model for a two-dimensional mobile robot is as follows:

$$f(x_{k-1|k-1}, u_k) = x_{k-1|k-1} + \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix}^T \begin{bmatrix} -\frac{v_t}{\omega_t} sin(\theta) + \frac{v_t}{\omega_t} sin(\theta + \omega_t \Delta t \\ \frac{v_t}{\omega_t} cos(\theta) - \frac{v_t}{\omega_t} cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{bmatrix} \tag{5.14}$$

$$f(x_{k-1|k-1}, u_k) = \begin{bmatrix} x_r & y_r & \theta_r & x_1 & y_1 & \dots & x_n & y_n \end{bmatrix}^T + \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix}^T \begin{bmatrix} -\frac{v_t}{\omega_t} sin(\theta) + \frac{v_t}{\omega_t} sin(\theta + \omega_t \Delta t \\ \frac{v_t}{\omega_t} cos(\theta) - \frac{v_t}{\omega_t} cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{bmatrix} \tag{5.15}$$

Where k is the current iteration of the filter, θ is the current orientation of the robot from the state vector, v_t is the linear velocity of the robot, ω_t is the rotational velocity of the robot, and Δt is the time interval in which linear and rotational velocities are measured. The matrix multiplied by the state prediction matrix simply gets the terms in the form of the state vector, where zeros are in the places of the x and y landmark variables. The control vector, u_k, is composed of the linear and rotational velocities,

$$u_k = \begin{bmatrix} v_t \\ \omega_t \end{bmatrix} \tag{5.16}$$

As $f(x_{k-1|k-1}, u_k)$ is a non-linear function, it must be linearized for use in the covariance update portion of the prediction stage.

$$\begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix}^T \begin{bmatrix} -\frac{v_t}{\omega_t}sin(\theta) + \frac{v_t}{\omega_t}sin(\theta + \omega_t \Delta t \\ \frac{v_t}{\omega_t}cos(\theta) - \frac{v_t}{\omega_t}cos(\theta + \omega_t \Delta t) \\ \omega_t \Delta t \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix} \tag{5.17}$$

Where F_k is the Jacobian of the motion model, and the matrices with 1's along the diagonal are $3x3N + 3$ where N is the number of recorded landmarks are used to get the linearized model into the following form where 1's are present along the diagonal of the matrix but the linearized motion equations are in the top $3x3$ section of the F_k matrix. This is for multiplication with the P matrix so in the prediction step the landmarks remain unaffected. The prediction update step with the Jacobian included is as follows:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{5.18}$$

Which is of the same form as equation 5.4, except here F_k represents the Jacobian from Equation 5.10.

5.6.4 The Observation Model and the Jacobian

The observation model is used to adapt the measurements into the form of the state vector. In the case of EKF SLAM, these measurements are usually the landmarks extracted from laserscan data. Laserscan data gives the landmarks in the form of range and orientation with respect to the robot's base reference frame. The observation model, $h(x_{k|k-1}, u_k)$, is responsible for getting landmarks from the form of x and y coordinates in the state variables vector to range and gearing in the base reference frame of the robot. In this form, a difference can be taken between the prediction and the measurement, and eventually determining the Kalman gain, which tells the filter how much to correct the state estimates based on the measurements. The measurement phase begins by taking the difference between the state and the measurement as follows:

$$y_k = z_k - h(x_{k|k-1}, u_k) \tag{5.19}$$

Where h() is the observation model,

$$h(x_{k|k-1}, u_k) = \begin{bmatrix} (\sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} \\ tan^{-1}(\frac{y_i - y_r}{x_i - x_r}) - \theta) \end{bmatrix} = \begin{bmatrix} range \\ bearing \end{bmatrix} \tag{5.20}$$

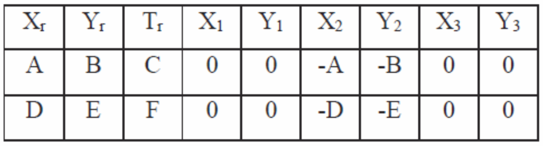
Where x_r and y_r are the robot's (x, y) coordinates from the state vector, $x_{k|k-1}$, x_i and y_i are the ith landmark coordinates from the state vector as well, θ is the robot's orientation within the map from the state vector, and i is the ith detected landmark. Taking the first order partial derivative of $h()$ with respect to x, y, and θ gives the following Jacobian, H:

$$H = \begin{bmatrix} \frac{x_r - x_i}{r} & \frac{y_r - y_i}{r} & 0 \\ \frac{y_r - y_i}{r^2} & \frac{x_r - x_i}{r^2} & -1 \end{bmatrix} \tag{5.21}$$

$H(1, 1)$ indicates the change in range with respect to x. $H(2, 1)$ indicates the change in bearing with respect to x. $H(1 : 2, 2)$ is the changes in range and bearing with respect to y. The last column shows the change in rotation. $H(1, 3)$ is zero because the orientation of the robot has no impact on the range to the landmark. However, H is the Jacobian of $h()$ and must be reshaped depending on the landmark which has been detected. Since it only affects the ith landmark and robot's orientation, given that H is of the form:

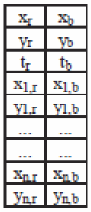
$$H = \begin{bmatrix} A & B & C \\ D & E & F \end{bmatrix} \tag{5.22}$$

Then the H matrix with in the filter will have the contents seen in Figure 5.5, using



X_r	Y_r	T_r	X_1	Y_1	X_2	Y_2	X_3	Y_3
A	B	C	0	0	-A	-B	0	0
D	E	F	0	0	-D	-E	0	0

Figure 5.5: Jacobian of the observation model [9].



x_r	x_b
y_r	y_b
t_r	t_b
$x_{1,r}$	$x_{1,b}$
$y_{1,r}$	$y_{1,b}$
...	...
...	...
$x_{n,r}$	$x_{n,b}$
$y_{n,r}$	$y_{n,b}$

Figure 5.6: Kalman gain matrix, K [9].

the 2nd landmark as an example:

The first three columns contain the full H matrix. In the columns containing the 2nd landmark, only the first two columns are used. This is because the orientation of the landmarks are not part of the state variables. The landmark values are also negated due to the change of perspective from the robot's base frame to the coordinates of the detected landmark.

5.6.5 The Kalman Gain

The Kalman gain determines how much of the measured values is used for correcting the state estimate. The following equation indicates how this value is determined:

$$K_k = P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1} \tag{5.23}$$

Where P is the covariance matrix, H is the Jacobian of the observation model, and R is the measurement error covariance matrix. The Kalman gain matrix has the form seen in Figure 5.6.

Where x, y, and t are the robot's position and orientation, and x_n and y_n denote the coordinates for the nth landmark. Each column represent the range and bearing of each variable, denoted by the subscripts r and b, respectively. The Kalman gain is used the following two equations for correcting the state estimates:

$$x_{k|k} = x_{k|k-1} + K_k y_k \tag{5.24}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{5.25}$$

5.6.6 The Noise Covariance Matrices

The Q and R matrices are used to model the uncertainty of the predicted state values and the measurement values. The Kalman filter assumes the noise associated with the predictions and the measurements are Gaussian distributed. The values in the Q and R matrices are usually assigned as the standard deviations of these values, often gathered from empirical trials. Q is associated with the prediction phase, and represents the variance from the control terms. Q needs to be the same size as the covariance matrix, P, for the matrix algebra to compute, however, as it is only associated with the control inputs, only the first three rows and columns need to have values associated; the rest of the matrix can be zeros. The R matrix is a two by two matrix, containing the variances along the diagonal and the other values being zero. In the case of using a laser rangefinder as the sensor, the resolution of the sensor is assigned as the first value on the diagonal, and the second diagonal value is set to the step size in the sweep, as these are associated with range and bearing measurement values, respectively.

5.6.7 EKF SLAM in Summary

EKF SLAM has been the standard in SLAM algorithms for years, until being largely replaced by the fastSLAM algorithms, which still utilizes the EKF, however, it utilizes it within the particles of a particle filter for dealing with the scaling problems which EKF SLAM encounters. Algorithm 4 shows the EKF SLAM algorithm as a whole.

Algorithm 4 EKF SLAM ($x_{k-1|k-1}, P_{k-1|k-1}, u_k, z_k$)

```
$z_0, R_0 = get\_measurements$
for $k = 1$ to $steps$ do
    $u_k, Q_k = get\_odometry$
    $x_{k|k-1}, P_{k|k-1} = EKF\_Prediction(x_k)$
    $z_k, R_k = get\_measurements$
    $DA_k = data\_association(x_{k|k-1}, z_k, R_k)$
    $x_{k|k-1}, P_{k|k-1} = append\_landmarks(x_{k|k-1}, z_k, R_k)$
    $x_k = EKF\_Update(x_{k|k-1}, z_k, R_k, DA_k)$
return $x_{k|k}, P_{k|k}$
```

5.7 Wireless Ranging in EKF SLAM

Wireless nodes can be used in SLAM algorithms to assist in localizing a robot, along with other nodes within the network. Typically, in traditional EKF SLAM approaches, the wireless nodes are used as landmarks. Using nodes as landmarks helps reduce some of the scaling problems in EKF SLAM by providing a static number of landmarks. This prevents EKF SLAM from continuously appending landmarks and exponentially consuming memory. If the number of nodes within the network is known at initialization, the appending step of the EKF can be avoided altogether by pre-allocating space for all landmarks within the system. Using nodes also completely solves the data association problem, as each node includes a network address, correctly identifying which measurement is associated with each landmark [35] [36].

5.7.1 Range-Only EKF SLAM

In the SLAM algorithm presented in 4, both the range and orientation of the landmark relative to the pose of the robot are measured. When utilizing wireless beacons as landmarks, only the distance to the landmark is available. However, range only measurements can still prove useful for refining the position of the robot within the world. These algorithms are referred to as range-only SLAM (RO-SLAM), and typically involve the use of wireless beacons as landmarks. Like any other form of SLAM, there have been many proposed solutions to the problem, however, EKF-based solutions are among the most common.

EKF RO-SLAM follows the same algorithm as EKF SLAM with only a few modifications to the measurement phase. Just as before the measurement phase takes the difference between the measured distance and the expected distance based on the current state estimation:

$$y_k = z_k - h(x_{k|k-1}, u_k) \tag{5.26}$$

However, in this case z_k is a scalar value rather than a vector. The observation function, $h()$, has the orientation estimation component removed:

$$h(x_{k|k-1}, u_k) = \begin{bmatrix} (\sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} \end{bmatrix} = \begin{bmatrix} range \end{bmatrix} \tag{5.27}$$

The Jacobian of $h()$ has the bottom row removed, giving the following:

$$H = \begin{bmatrix} \frac{x_r - x_i}{r} & \frac{y_r - y_i}{r} & 0 & \dots & -\frac{x_r - x_i}{r} & -\frac{y_r - y_i}{r} & \dots & 0 & 0 \end{bmatrix} \tag{5.28}$$

As the measurement vector is reduced to a scalar, the R matrix is also as it only needs to express the variance of the measurement noise.

5.7.2 RSSI EKF RO-SLAM

Landmarks in wireless beacon EKF RO-SLAM must go through an additional initialization phase before being appended to the state vector. In traditional EKF SLAM, landmarks must be observed multiple times before being appended to the state vector, as to avoid appending faulty observations as landmarks or legitimate landmarks which cannot be consistently observed due to obscuring factors in the environment such as clutter which limits visibility. Since orientation is not possible to observe from a single reading, the x, y position of the landmark must be inferred from multiple observations as the robot moves along a well estimated trajectory. At

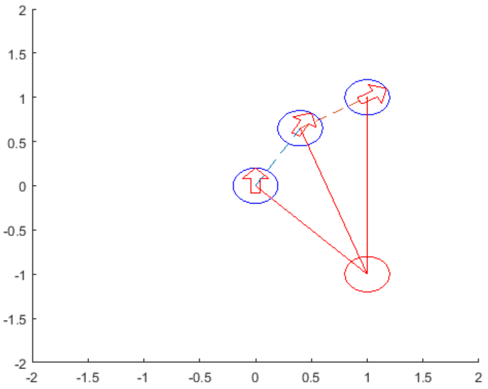


Figure 5.7: Wireless beacon EKF initialization.

multiple points along the robot's motion RSSI reading can be recorded and after a pre-determined number of measurements have been taken a trilateration localization method may be applied to determine an initial estimate of the beacon's x, y position. Figure 5.7 illustrates trilateration initialization by estimating distance at multiple points along a trajectory and performing trilateration to determine the position of the node [35].

While trilateration has been proven an effective technique, other, more computationally-intensive, yet more effective techniques have been used to initialize node locations. Particle filters have already been applied to localize nodes in a network from a robot's motion, and can easily be applied here. Probability graphs can also be used to find the location of a landmark. Once an unlocalized wireless beacon has passed some confidence/variance threshold, the landmark can be appended to the EKF state vector and covariance matrix, where the EKF will refine the estimated position.

Mitigation of multipath interference on the estimated distance from robot to wireless beacon landmark is the largest inhibiting factor when using RSSI in RO-SLAM. To reduce the multipath effect, Menegatti et al proposed a method to filter the impact multipath inteference has on the distance estimation model by limiting estimation based on odometry information. This method utilizes the linear displacement of the robot's odometry to generate an estimated RSSI value and combines that with the measured RSSI value to a particular node. If a robot moves some distance between RSSI measurements, the RSSI value cannot undergo an extreme change relative to the change in distance. Given the motion vector of the robot is relative the frame of the world or map, it is not known relative the node's position, however; the extreme cases can be considered. If a robot were to move 0.5 meters between readings the maximum expected reading would be the RSSI value associated with a 0.5 meter increase in distance and the minimum would be the RSSI value associated with a 0.5 meter decrease. Given a newly measured RSSI value and a displacement vector from odometery the filtered RSSI value is as follows:

$$RSSI_e = 10n \log d \pm u + A$$

$$RSSI_p = \frac{RSSI_e + RSS_m}{2}$$

$$d_f = 10^{\frac{RSSI_p - A}{10n}}$$

Where $RSSI_e$ is the expected RSSI at a given distance, d, plus or minus the displacement, u. If the change in RSSI is positive, the change is added, and if the change is negative, the change is subtracted. RSSIp is the predicted RSSI value found by averaging the expected RSSI with the measured RSSI. The filtered distance, d_f, is found by re-arranging the log-distance model and applying the predicted RSSI value. The filtered distance value behaves as a low-pass filter for RSSI values [35]. While this work has been one of the few to utilize this kind of approach, it has also been criticized for relying heavily on the odometry data.

CHAPTER 6: ANALYSIS AND SIMULATION OF MULTIPATH EFFECTS IN DISTANCE ESTIMATION

Multipath propagation is difficult to detect in indoor environments due to the amount of variables which contribute to the effect. In this chapter, multipath propagation effects are detected utilizing distribution specifics known about the signal and comparison of predicted values against measured values to detect the multipath interference. Utilizing a mobile robot navigating within a distributed network, detection of nodes undergoing multipath fading is used to devalue the distance estimation to those nodes by use of assigned weights, and increasing the distance estimation to nodes which seem to have minimal interference.

6.1 RSSI-Distance Recorded Data

To begin characterizing the multipath effect, signal attenuation data was gathered in various indoor environments. Tests were taken in classrooms, laboratories, and hallways within multiple different buildings.

Data was collected using XBee 802.15.4 2.4 GHz modules and Arduino Nano microcontrollers for managing transmissions. Omnidirectional, 2.1 dBi antennas by Digi International were used with the XBee modules. Each XBee was configured for transmitting at PL (Power Level) 0 and 2, where the output transmission power is -5dBm and +1dBm, respectively.

6.1.1 Data Collection Method

For the examples shown in this work, data was collected within the Energy Production and Infrastructure Center (EPIC) building on the campus of the University of North Carolina at Charlotte. Data was collected in various environment types, including large classrooms, hallways, and laboratories. An outdoor data set was also collected for a set with minimal multipath interference. This data was collected using XBee 802.15.4 radios and Digi 2.4 GHz Omnidirectional Dipole antennas with a 2.1 dBi Gain. The XBee radios were configured with boost mode disabled and power level set to 2 (1 dBm Gain). Each XBee module was interfaced with an Arduino Uno; one module being set to transmit, and the other configured to return the RSSI value of the received packet. Each node was placed on a 1 meter tall stalk and moved away from the transmitter at 0.25 meter increments up to 10 meters. At each increment, 5 readings where taken at 90 degree increments for a full rotation. Despite the antennas being omni-directional the radiation pattern is not perfectly symmetrical and the rotations attempt at capturing those variations. This data was recorded and placed within a spreadsheet. The simulation would then later use these spreadsheets to train the Markov chain. Figure 6.2 depicts an example of the collected RSSI-distance data. In this figure, the first column's data indicates the individual RSSI readings in -dBm at various distances in meters in blue dots, open red dots represent the average RSSI value at that distance, and the black line is the fitted log-distance model with the parameters used indicated in the plot's label. The middle graph is the variance of the data at each distance, and the last graph shows the mean square error between the predicted RSSI from the fitted log-distance model and the noisy measurement.

6.1.2 Collected Data in Various Environments

The following Figures 13 through 16 show the collected data in various environments. The figures are organized by similar environments.

From this data, it can be seen that some types of environments, such as hallways and classrooms exhibit similar multipath characteristics, while laboratories and large, open areas, such as building foyers, tend to vary largely between each other. Ultimately, the position of the node within the geometry of the room will greatly affect the distance at which the multipath effect occurs. Without knowledge of the environ-

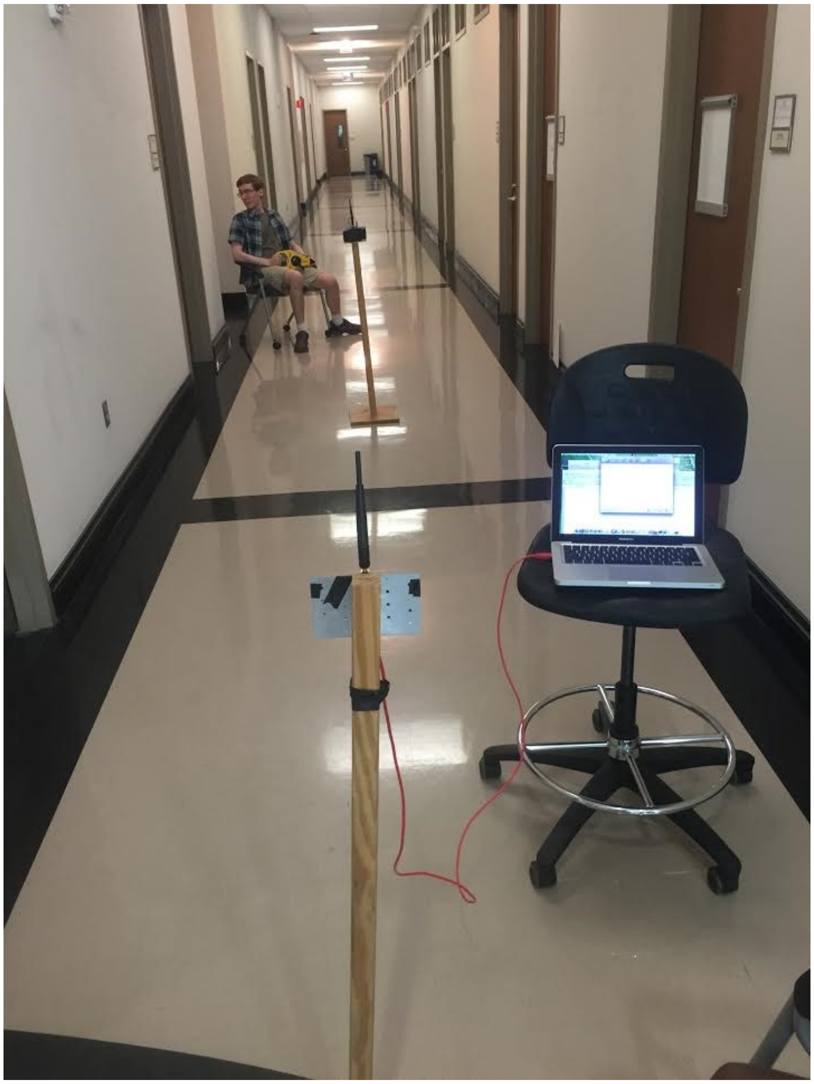


Figure 6.1: RSSI data collection method.

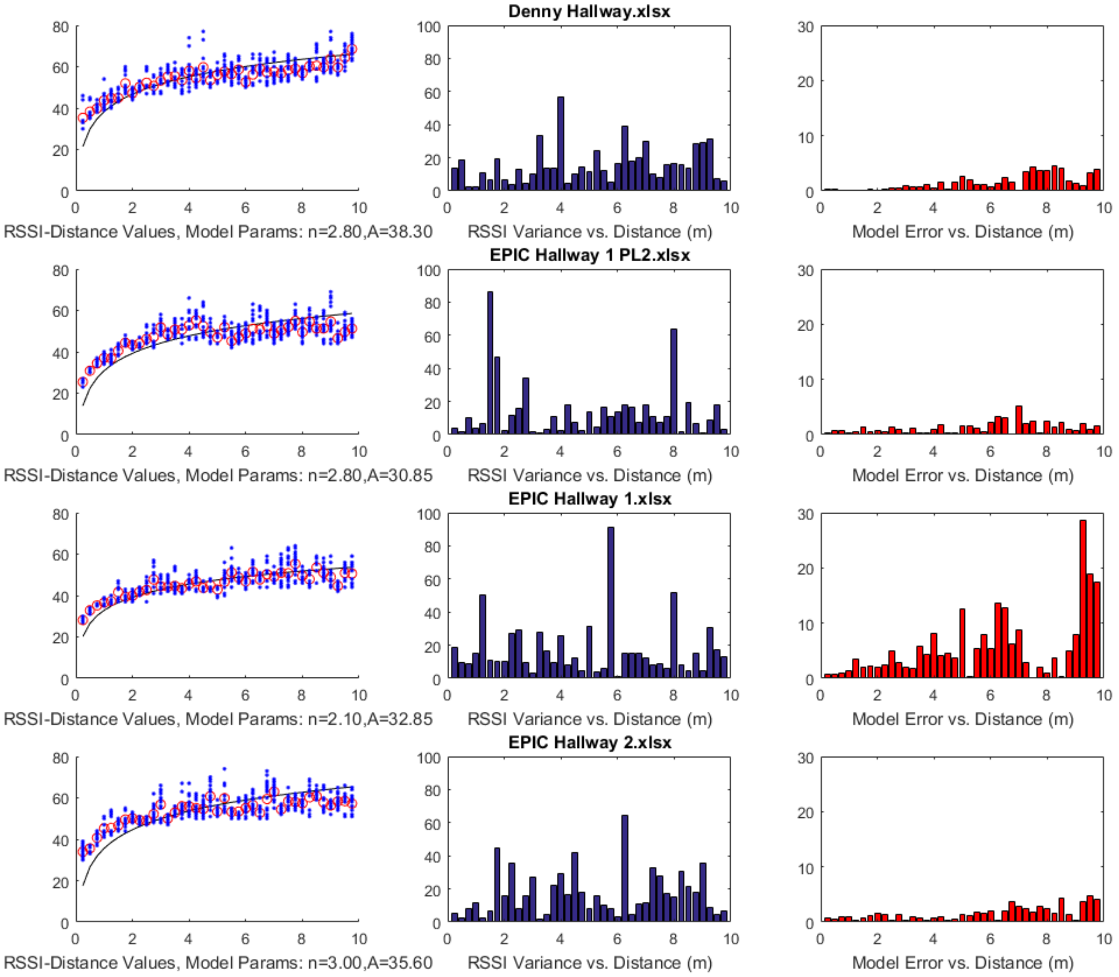


Figure 6.2: RSSI vs. Distance Measurements and Statistics for various hallways on the University of North Carolina at Charlotte's Campus.

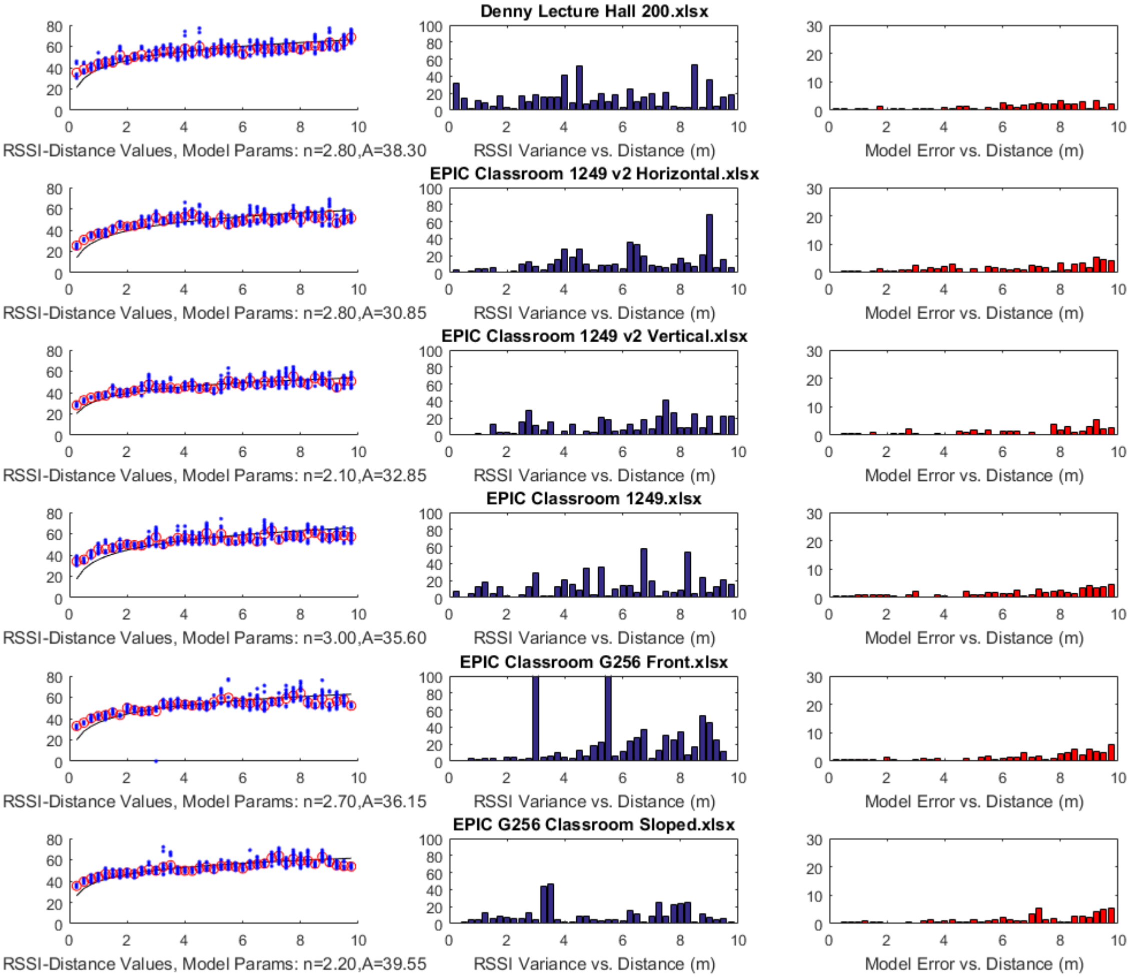


Figure 6.3: RSSI-Distance data collected from various classrooms around the UNCC campus displayed in the same format in Figure 6.2.

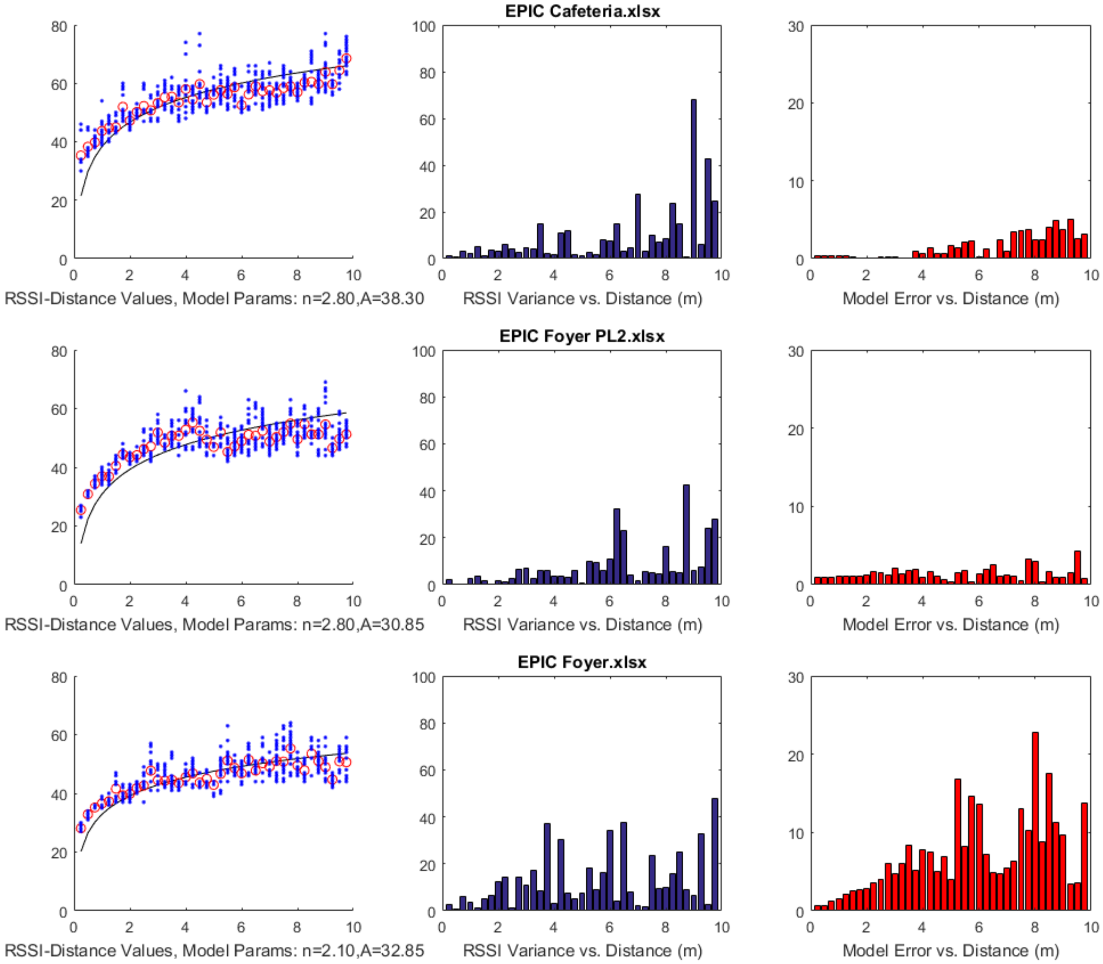


Figure 6.4: RSSI-Distance data collected from various large open rooms around the UNCC campus displayed in the same format in Figure 6.2.

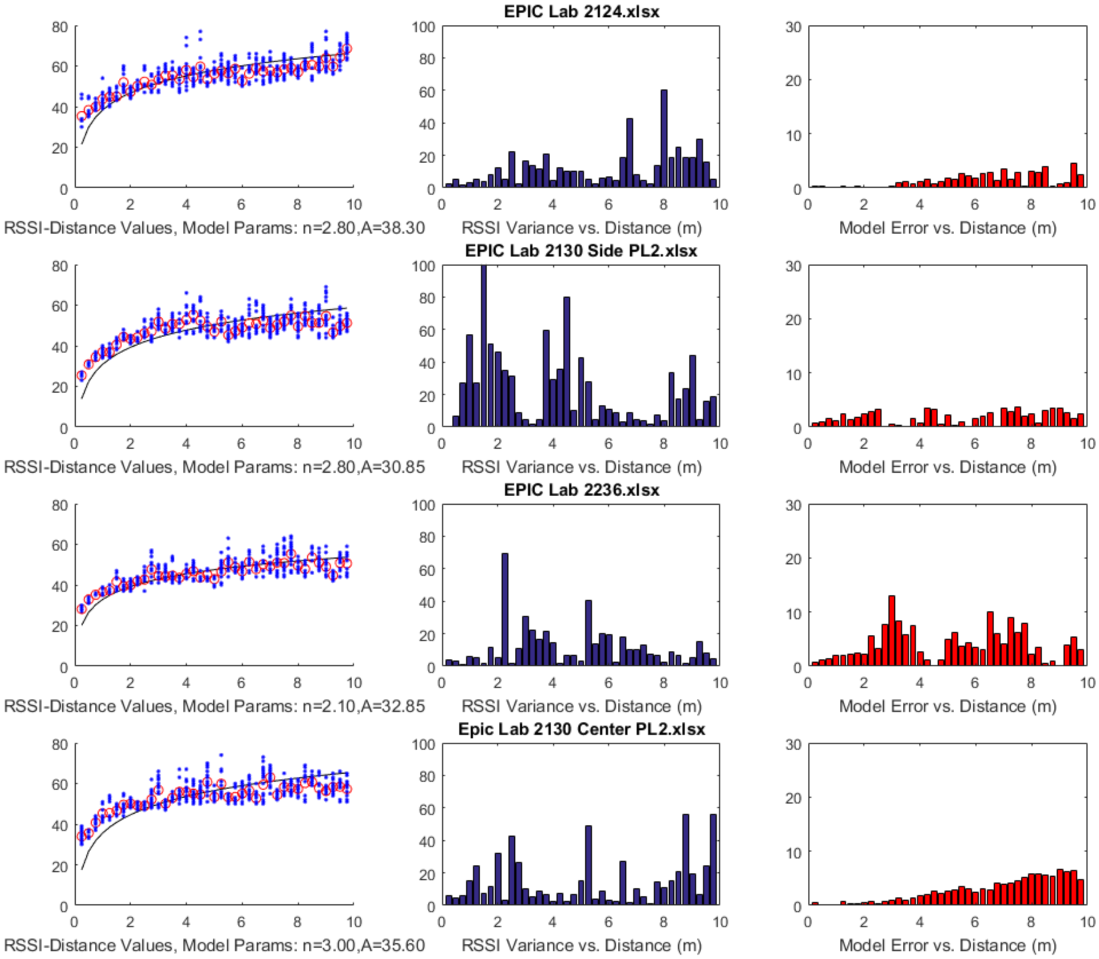


Figure 6.5: RSSI-Distance data collected from various research laboratories around the UNCC campus displayed in the same format in Figure 6.2.

ment's geometry or phase information of multipath signals, predicting the occurrence of multipath propagation is extremely difficult. However, with some knowledge of how the data is distributed within different environments, a confidence metric may be formed to express the quality of a distance estimate from RSSI data.

6.2 Simulating RSSI Data

Implementing an entire sensor network to test localization algorithms can be costly and time consuming. Therefore, it is practical to simulate a testing environment prior to implementation. However, simulation of RSSI can be unreliable due to over-idealized signal strength data, which does not accurately portray the multipath effect, or it can computationally expensive from ray-tracing techniques and geometrically complex environments. Here we present a simulation technique which accurately portrays the RSSI signal behavior within an environment of interest, given the application is only concerned with signal behavior and not the geometry of the environment. This simulation technique involves training a discrete Markov chain to generate RSSI data over distance, utilizing actual collected RSSI data in the environment of interest. This generated data contains RSSI over distance vectors, which are then assigned to a specified number of rays emitting from the node's location. This forms a discretized radiation pattern for each node which serves as a lookup table given a certain distance and angle from the node. All figures and simulations presented in this work are generated using MATLAB 2016B.

6.2.1 Markov Chain for Generating RSSI Data

Markov chains are procedural algorithms which determine the next state of a random variable from the current state. Each state has a probability associated with it to transition to a new state on the next iteration [37].

For this simulation, each combination of discrete RSSI value and distance value is modeled as a state. Each state has some probability of transitioning to a state 1 distance increment higher than the current state (0.25 m in this example, because each RSSI measurement was taken at 0.25 meter increments). The probability of transitioning to any state with a distance lower than the current state or higher than 1 increment of the current state is 0. The probability of transitioning to the next state one distance increment higher is determined by how often a new RSSI value is observed after the RSSI value of the current state. Following the Markov chain from distance = 0.25 and a randomly selected initial RSSI value from the set of RSSI values at 0.25 meters, the chain generates an RSSI-distance data set that contains the multipath characteristics of the environment from which it was collected. This data is utilized as an RSSI look-up table for a certain distance from the transmitter. The limitation of this method is the maximum distance that can be accurately simulated is dependent on the maximum distance from the source data collected in the actual environment. While the Markov process is used to model systems in which the present state is independent of the history of the system, as the current state is said to be representative of the conditions which have led to that point. While this is not true for predicting actual multipath interference, for the particular algorithms simulated here, a Markov chain can simulate the sudden changes in signal strength observed in the real environment. If the application being simulated requires a realistic history of signal strength, the Markov chain might not properly emulate the same values the actual environment may produce. However, in the simulation environment presented here, the details of the environment, such as geometry and material compositions, have been excluded, so the Markov chain provides the kind of multipath signal strength fluctuations needed to test the robustness of the algorithm.

Figure 6.7 graphically depicts an example of RSSI-Distance Markov chain at state, distance = 0.5, RSSI = 35. Each state is defined by a discrete RSSI/distance combination. The arrows and numbers indicate the probability of transitioning into the next state. As illustrated here, only states one distance increment above the current

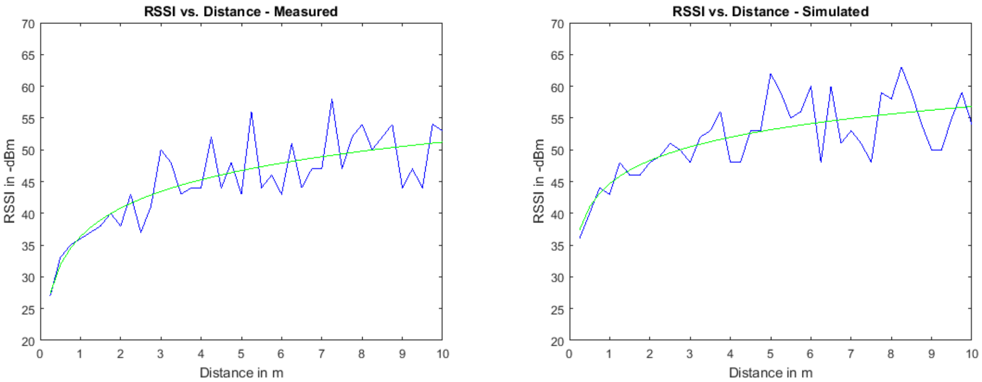


Figure 6.6: Measured and simulated RSSI data over distance.

state have some probability of transitioning to. The size of the distance increment is determined by the resolution of the collected RSSI over distance data. The probability of transitioning is calculated based on how often an RSSI value was observed on the next distance increment during the data collection process.

Figure 6.6 depicts an RSSI vs. distance graph generated by a trained Markov chain. The measured RSSI was data collected within a long hallway environment. The simulated data is produced by the Markov chain trained on the collected data from hallways. The blue line represents the average RSSI value collected at each distance increment, and the green line is the calibrated log-distance model for this data.

6.2.2 Determining the RSSI Value

Within the simulation environment each node is assigned a certain number of rays where each ray is composed of a Markov-generated RSSI-distance vector. Given a receiver node is a certain position, distance and orientation, away from the node the RSSI value is determined based on distance from the node and the distance to the nearest rays using a bilinear interpolation method. Figure 6.8 depicts the ray configuration for a node generated with 8 rays.

The following equation describes the RSSI-value assignment given a non-integer

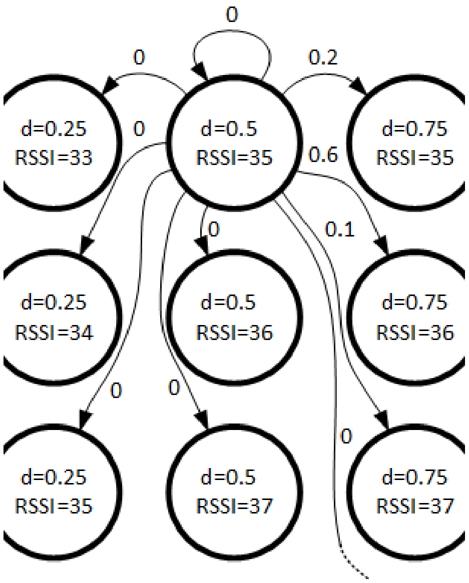


Figure 6.7: Example of RSSI-Distance Markov chain.

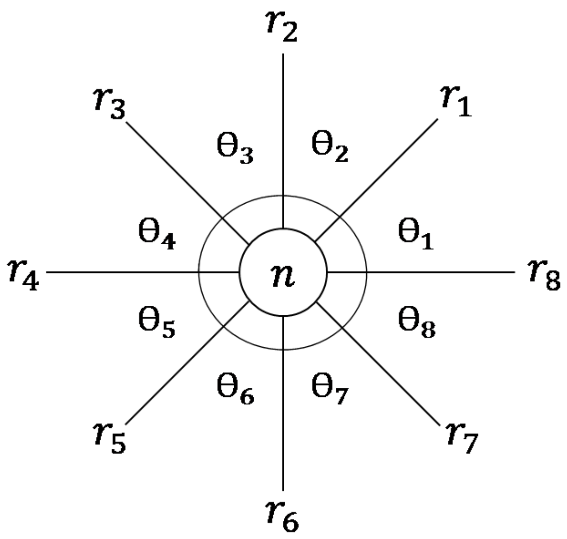


Figure 6.8: A Markov generated node with 8 rays. Each ray, r_n, corresponds to a generated RSSI-distance vector at an associated angle, θ.

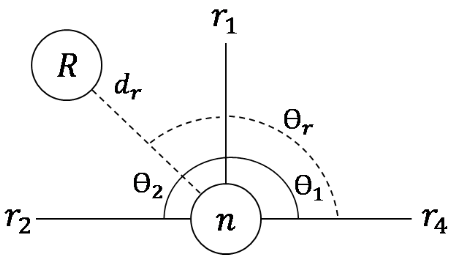


Figure 6.9: An Illustration of the bilinear interpolation process.

valued (x, y) position relative to the node:

$$RSSI(\theta_1, d) = \frac{d_2 - d}{d_2 - d_1} RSSI_m(\theta_1, d_1) + \frac{d_1 - d}{d_2 - d_1} RSS_m(\theta_1, d_2) \tag{6.1}$$

$$RSSI(\theta_2, d) = \frac{d_2 - d}{d_2 - d_1} RSSI_m(\theta_2, d_1) + \frac{d_1 - d}{d_2 - d_1} RSS_m(\theta_2, d_2) \tag{6.2}$$

$$RSSI(\theta, d) = \frac{\theta_2 - \theta}{\theta_2 - \theta_1} RSSI_m(\theta_1, d) + \frac{\theta_1 - \theta}{\theta_2 - \theta_1} RSS_m(\theta_2, d) \tag{6.3}$$

Where θ and d represent the angle and distance of the receiver node with respect to the transmitter node, and the sub-indexes of these variables represent the nearest discrete indexes generated by the Markov chain. The $RSSI_m$ function serves as a lookup table function for specified indexes. Depending on the number of rays specified, each ray has an angle value, θ, associated with it. Figure 6.8 graphically illustrates the bilinear interpolation process.

Using this equation to determine the RSSI value, Figure 6.9 displays radiation patterns for nodes generated with Markov chains trained on various environment types. In this figure, the position of the node fuses two RSSI values from rays r_1 and r_2, which are the weighted average between two RSSI readings from each ray based on distance. The two RSSI reading are averaged again based on weights determined from distances to each angle, θ_1 and θ_2.

Figure 6.10 depicts the radiation pattern for two nodes generated by the Markov chain in outdoor and indoor environments, respectively. Both figures have a node placed in the center of the graph, at (4,4) and the intensities along the z axis indicate the signal attenuation. The graph on the top was trained on data collected in an outdoor environment, where the multipath interference was minimal, while the graph on the bottom was trained on hallway data, exhibiting intense multipath fading interference.

6.3 Mobile Robot Simulation Environment

Prior to investing the time and effort into deploying a physical implementation of a robotic localization system, it is practical to simulate the algorithm being tested. Simulation allows for quick modification and testing, increasing the likelihood of the algorithm in question to be valid upon implementation.

Many robotics simulators exist, the most popular among being Gazebo. However, many of these simulators exist only for implementation using conventional robotics sensors, and do not include wireless ranging options, and even if included, they do not properly model indoor, short-range multipath fading. There are also simulators for modeling multipath fading indoors, however these simulators are not designed for simulating robotics applications as well and have high simulation run times, as they compute many details about the transmission through the channel, environment, and even aspects of the modulator and antenna. Even with the detail provided by these simulators, it is still difficult to model the RSSI effects in an urban indoor environment. Therefore, the RSSI simulation method described in the previous section is used, as it is statistically generated based on empirical data from the environment type desired.

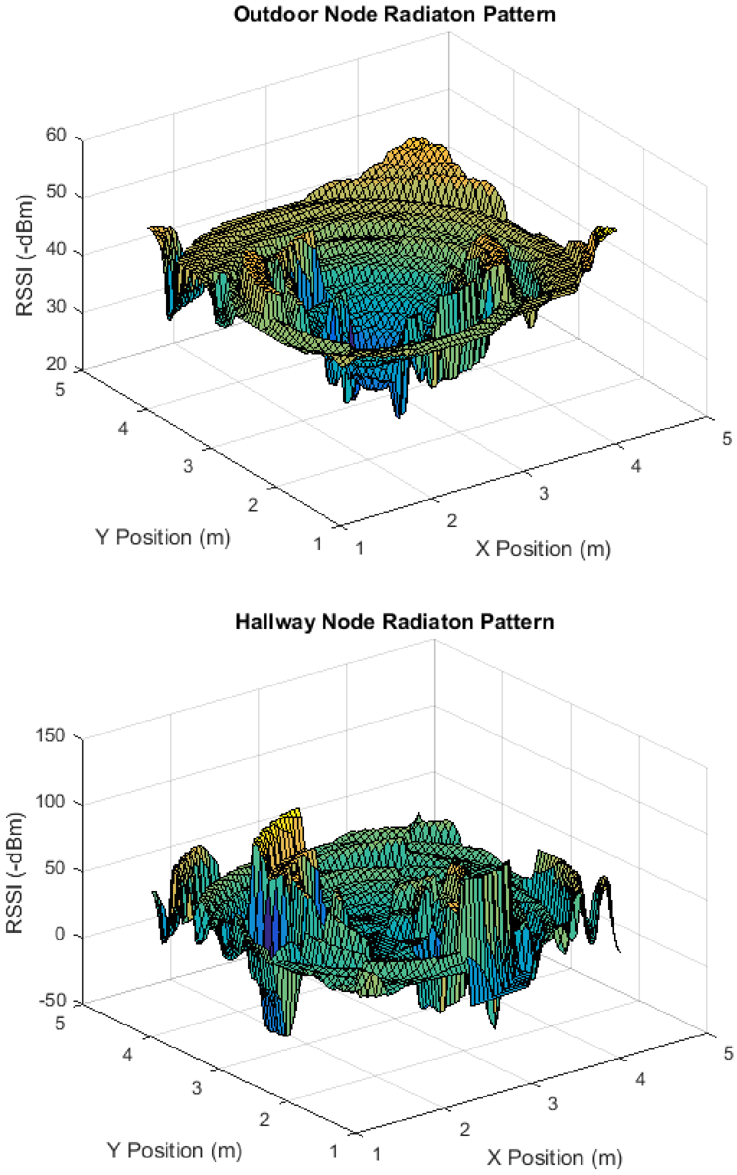


Figure 6.10: Simulated radiation patterns from Markov chains trained on various environment data.

CHAPTER 7: SIMULATION OF RSSI RO-EKF SLAM

In this chapter a method for mitigation of multipath interference is presented and simulated within an EKF RO-SLAM implementation. The following section discuss the details of the simulation environment, multipath filtration method, RO-SLAM configuration, and simulation results.

7.1 Simulation Environment

For the work described in this research, the simulation method described in Chapter 6 is used for generating and measuring RSSI data from wireless beacons. While mapping of the environment is an aspect of the overall goal of SLAM, construction of the map itself is trivial once the landmarks have been defined and localized. For that reason, the environment details such as obstacles and walls have been excluded to reduce complexity of algorithm testing and simulation runtime. Here only the estimation of the robot's pose and landmark localization are concerned.

The simulation environment begins by defining the robot's initial condition and wireless beacon locations. Since the environment is excluded from containing any obstacles, the locations are fairly arbitrary. Each beacon has an RSSI radiation pattern generated by the Markov RSSI model. The robot also has a pre-defined path defined for it to traverse, taking measurements to each landmark at each step and using the EKF RO-SLAM algorithm to refine its position. The robot's true position is tracked by the simulation environment which is perturbed with Gaussian noise to simulate the inaccuracies of the odometry sensors.

In this simulation, nodes are initialized with pre-defined location variables as well within the state vector. In a full RO-SLAM algorithm, some initialization method would have to be employed to find the initial x, y position from multiple range measurements taken over time. To simulate the initialization process, the actual value is perturbed by some random noise when added to the state vector.

7.2 Multipath Mitigation Method

To filter multipath interference a method similar to Menegatti et al. [35] presented, discussed in Section 5.7.2. This method is desirable as it takes advantage of the odometric data provided by the mobile base. However, in the method presented by Menegatti, the estimation of the measured RSSI is "evened out" with an maximum or minimum expected change in RSSI based on the odometry data and the attenuation model. This method will still generate rapid changes within the distance estimation when operating within an area susceptible to intense multipath interference.

The method proposed in this work attempts to estimate distances to each node based on a predicted distance estimation using the estimated landmark and robot positions from the EKF state vector. In contrast to Menegatti's work, the actual distance change from odometry relative to the landmark is estimated rather than using the maximum or minimum estimated change possible based on the motion in any direction within the world frame.

The multipath mitigation method presented here is dubbed the trust filter, as the odometry is used to determine whether the RSSI-distance estimate can be trusted due to multipath interference. The odometry measurements are projected onto a line which passes through the robot's current estimated position and the landmark position for each landmark. While odometry has been proven unreliable for long time periods due to sensor measurement drift, it is reliable for short measurements. Multipath interference in most indoor applications is manifested by huge changes over small distances, as it is a form of small-scale fading. The distance estimates may vary wildly as the transceivers are displaced by changes no longer than the wavelength of the frequency. The "Trust" factor is a weight used for averaging the measured distance from RSSI readings and a predicted distance from odometry changes. The measured distance from odometry is taken to get the change in distance relative to the landmark. The change in distance between time steps of the landmarks are compared to the relative odometry readings. If the difference between the readings is low, the trust factor is high, incorporating more of the measured distance into the distance estimate. If the difference is great, then the trust factor is low, using more of the predicted distance to determine the estimated distance.

Initially, the odometry, the control vector u_t, is transformed from a displacement relative to the world plane to a displacement relative to the robot and the ith landmark:

$$u_t = [\Delta d \Delta\theta]^T \tag{7.1}$$

$$u_{oi} = \sqrt{(x_i - (x_r + \Delta d \cos\Delta\theta + \theta_r))^2 + (y_i - (y_r + \Delta d \sin\Delta\theta + \theta_r))^2} - \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2} \tag{7.2}$$

Where Δd is the change in distance from the odometry, $\Delta\theta$ is the change in orientation from odometry, x_r and y_r are the x, y position of the robot from the state vector, x_i, y_i are the x, y position of the ith landmark. It is important to note that within the simulation presented here, RSSI measurements to each node are made available after each control vector is calculated. The predicted distance is calculated by applying the relative odometry to the previous distance estimation, $t-1$, for the ith node.

$$d_{pi}^{t} = d_{ei}^{t-1} + u_{oi} \tag{7.3}$$

The measured distance, d_m, is determined from the RSSI value using the log-distance path loss model:

$$d_{mi}^{t} = 10^{\frac{RSSI_i - A}{10n}} \tag{7.4}$$

The difference between the predicted change and the change in measured distance is calculated for each node.

$$\Delta v_i = u_{oi} - (d_{mi}^{t} - d_{mi}^{t-1}) \tag{7.5}$$

Where $RSSI_i$ is the average path loss expressed in -dBm for node i, A is the RSSI value at the reference distance (usually 1 meter), and n is the path loss exponent. The estimated distance The trust factor is calculated using the following expression.

$$T_i = e^{-k\Delta v_i}, k = \frac{-\ln(0.5)}{x_0} \tag{7.6}$$

Where T is the trust factor ranging from 1 to 0 as $\delta d \to \infty$, k is the decay rate factor which ensures the point x_0 is the value of 0.5. The point x_0 is set to a value which represents an acceptable resolution.

$$d_{ei}^{t} = (1 - T_i)d_{pi}^{t} + T_i d_{mi}^{t} \tag{7.7}$$

The estimated distance, d_{ei}^{t}, is then used as the input to the RO-SLAM algorithm for the ith landmark.

7.3 Simulation Results

The RO-EKF SLAM algorithm is described in Section 5.7.1. The block diagram for the system being simulated is described in Figure 7.1. Figure 7.2 depicts the simulation environment and trajectories. The robot followed a predetermined square path four times, represented by the black line. The gray line represents the robot's estimated location based on odometry readings alone, which is the dead reckoned position. The red path is the robot's location according to the EKF SLAM algorithm. The blue circle with the red arrow represents the robot's estimated pose at the end of the simulation. Each landmark's true position is represented by a blue circle, and the estimated positions as magenta circles. The error over time from the SLAM algorithm is shown in Figure 7.3 where the localization error is represented over time by the red line and for comparison, the dead-reckoned error is shown in gray. The longer the simulation runs, the greater the error from the dead reckoned position becomes while the EKF position remains fairly consistent. The results of the trust filter are shown in Figures 7.4 - 7.7. The first graph in each figure depicts the distance estimation error over time from the robot to the node. The true distance is shown in blue, the distance from the log-distance model is shown in green, the yellow shows the distance estimated from the robot's dead-reckoned position, and the red line represents the estimated distance from the trust filter. The second graph shows the velocities of the distance estimation over time, which is used for determining the trust factor. The trust factor value is shown in the third graph, with the green line representing the weight at that iteration and the blue line being the result of an averaging filter. The averaging filter prevents the estimation from associating an erroneous reading too quickly. Finally, in the fourth graph, the error over time from the estimated distance to the node is shown in red. All vertical axis data is in meters, and horizontal axis data represents the simulation iteration.

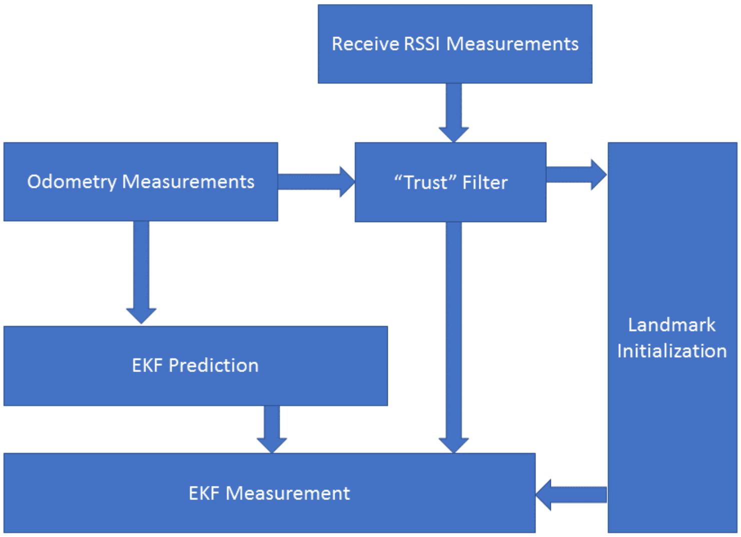


Figure 7.1: The algorithm block diagram.

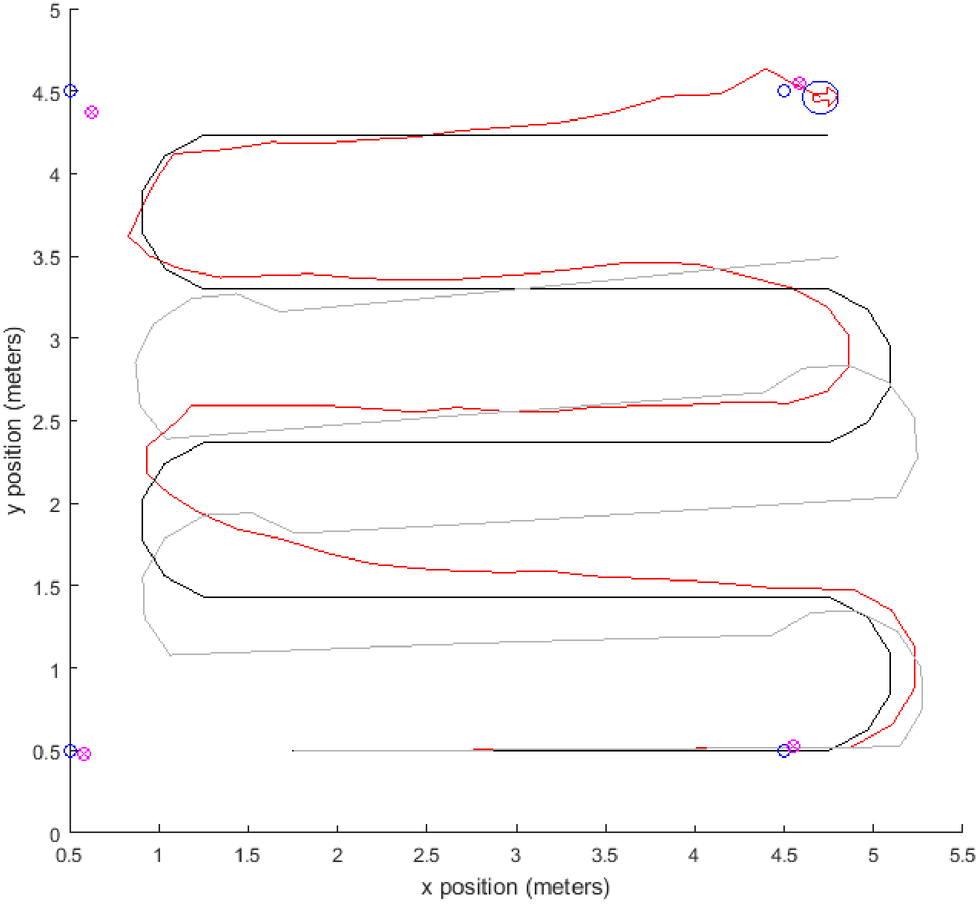


Figure 7.2: The simulation results.

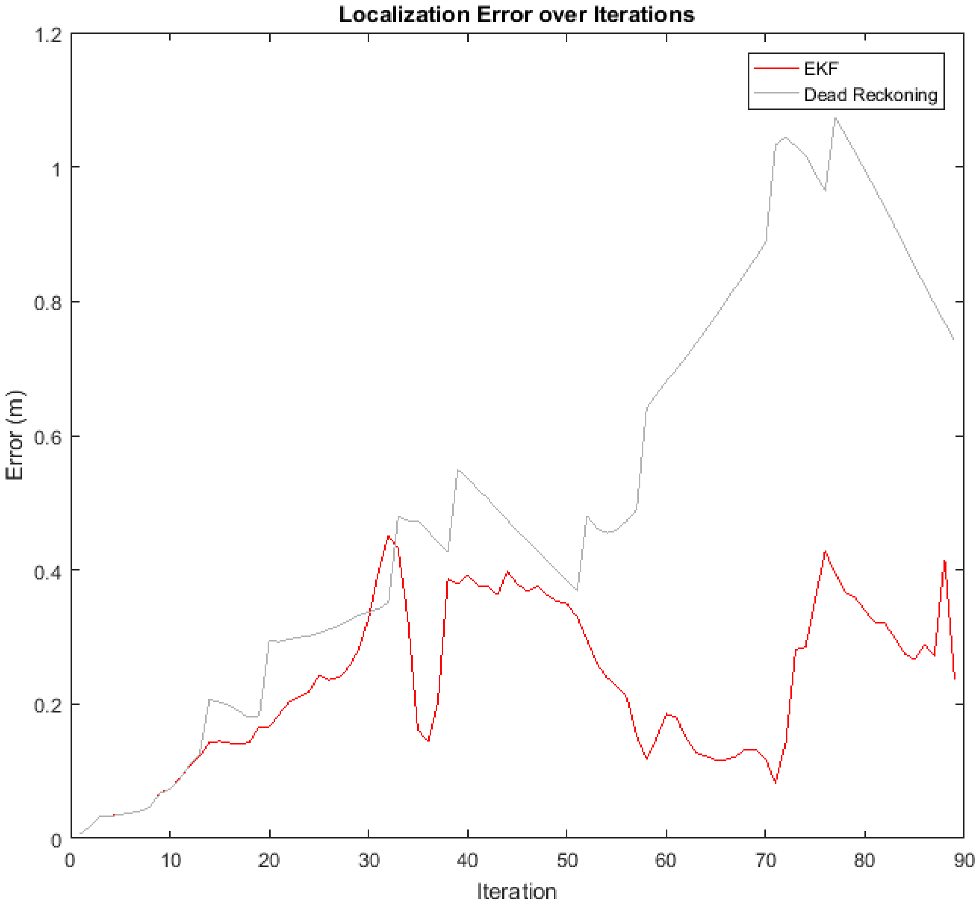


Figure 7.3: Localization error over path traversal.

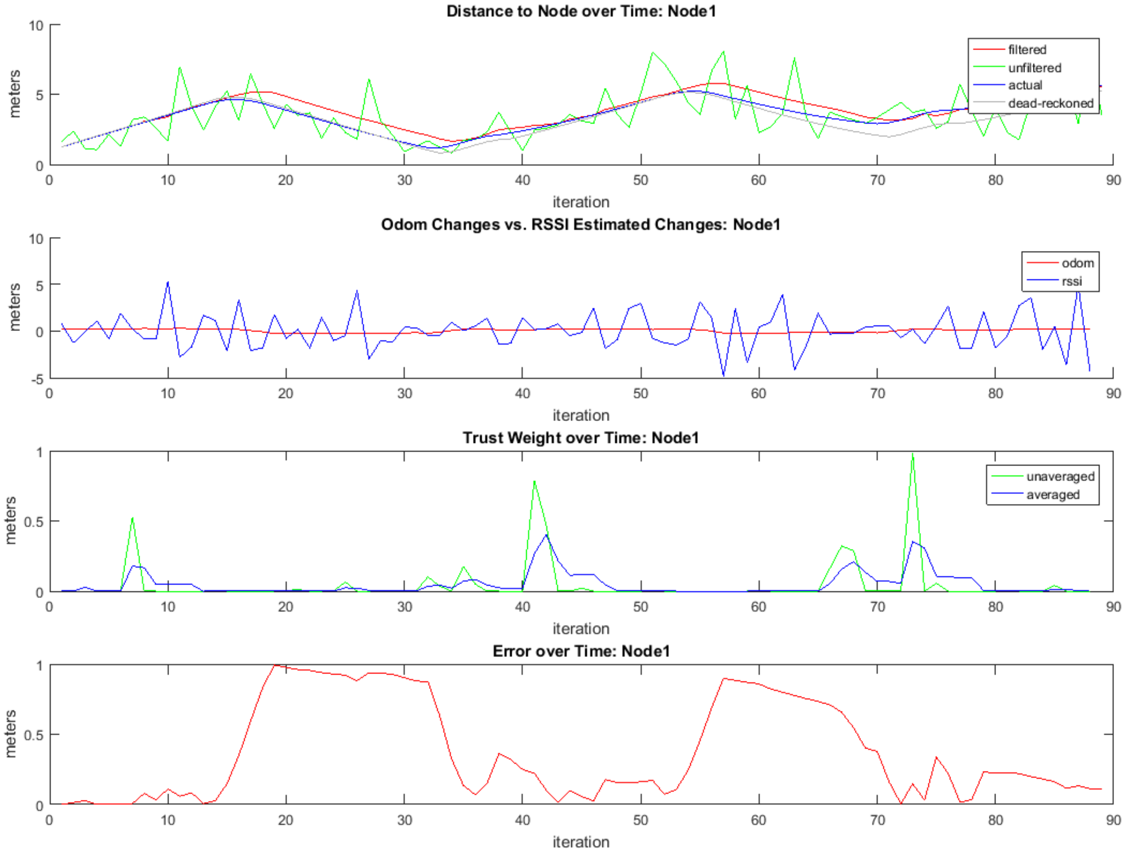


Figure 7.4: Estimated distance, relative displacements, trust weight, and distance estimation error over time for the first landmark.

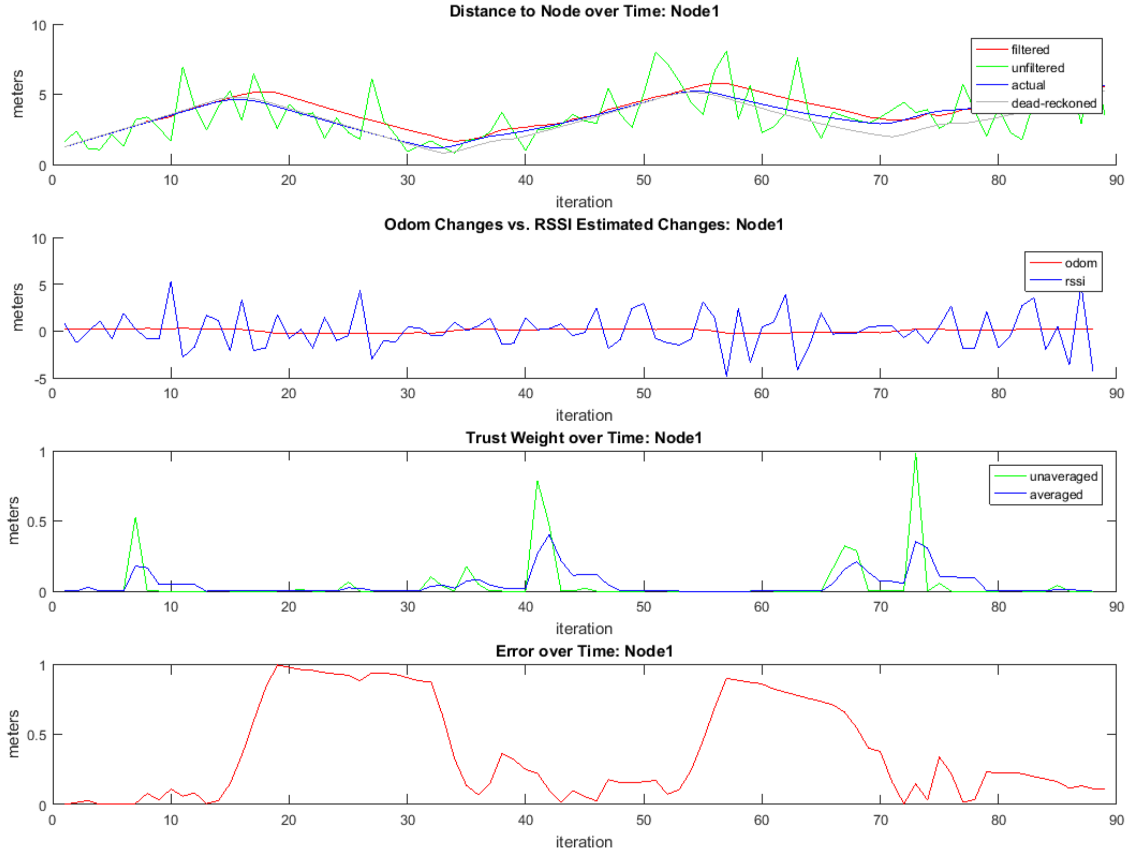


Figure 7.5: Estimated distance, relative displacements, trust weight, and distance estimation error over time for the second landmark.

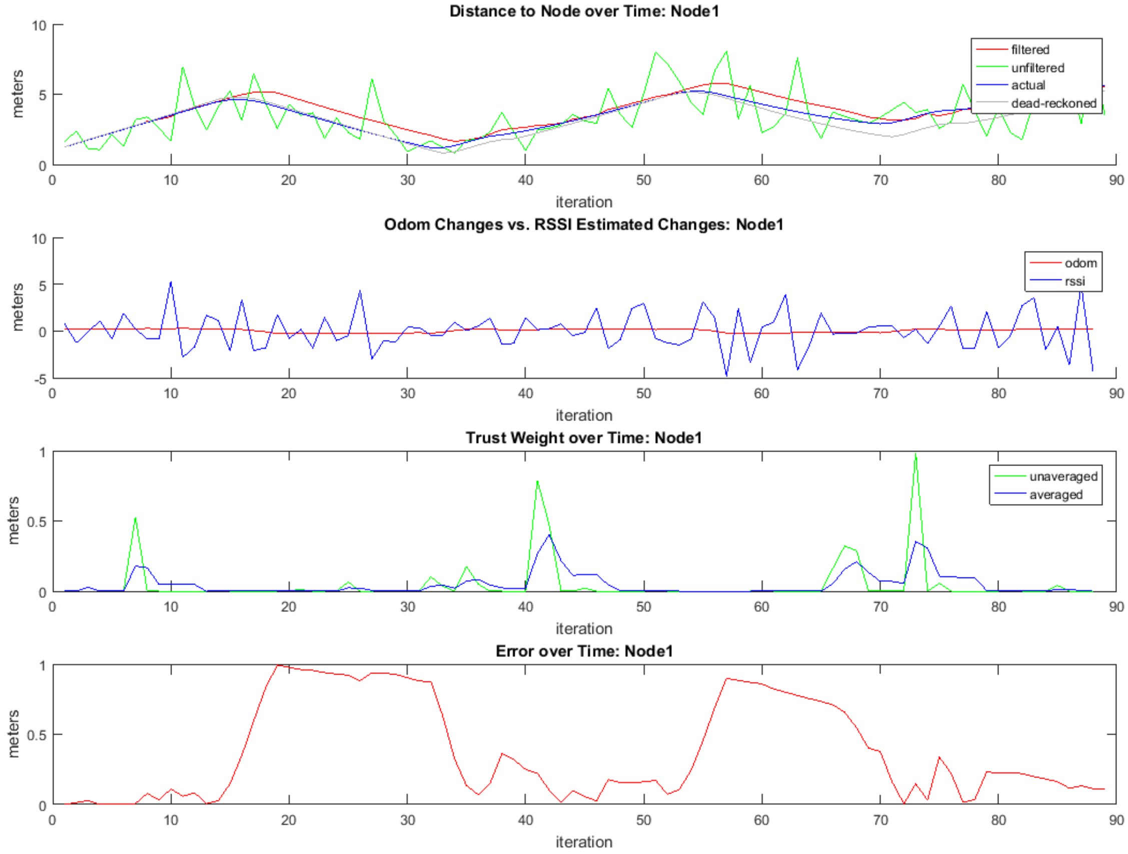


Figure 7.6: Estimated distance, relative displacements, trust weight, and distance estimation error over time for the third landmark.

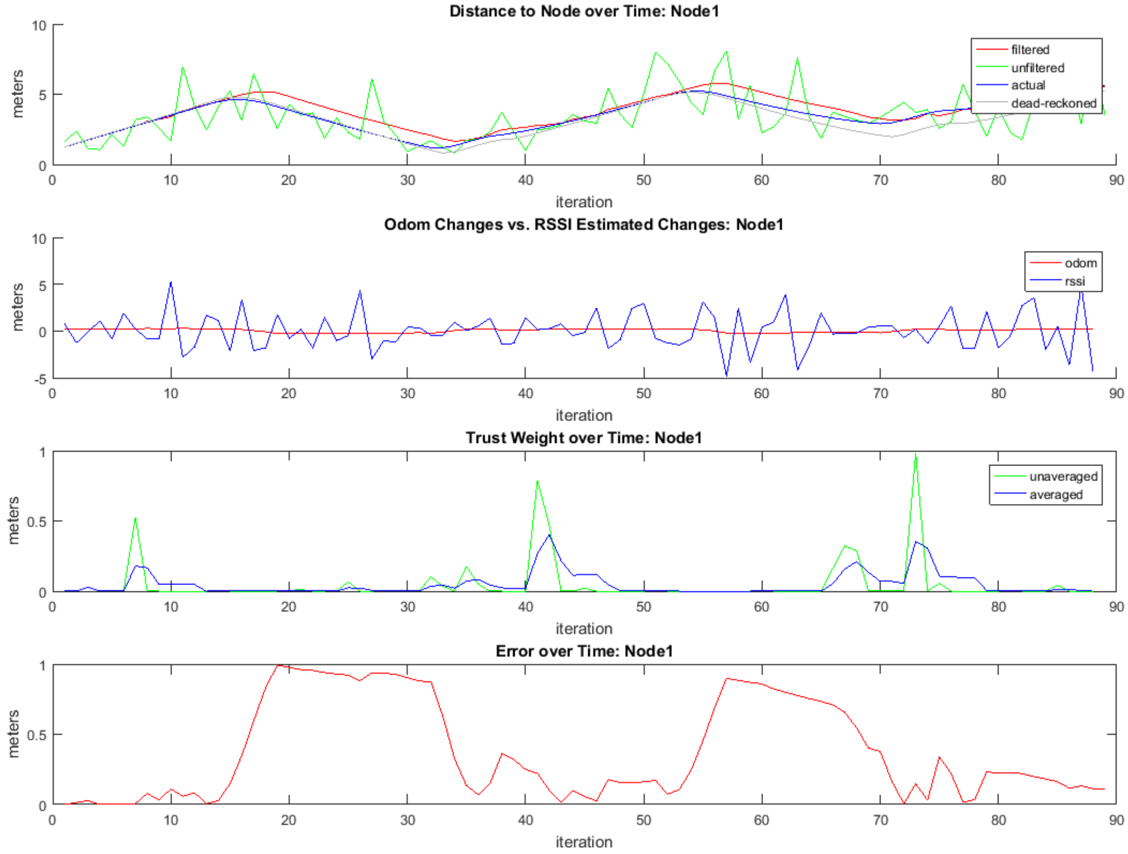


Figure 7.7: Estimated distance, relative displacements, trust weight, and distance estimation error over time for the fourth landmark.

CHAPTER 8: CONCLUSION

The work presented here has led to the conclusions discussed in the Section 8.1 and inspired further research objectives discussed in Section 8.2.

8.1 Conclusion

Path loss models and localization techniques have been explored, with a focus on techniques applicable to common transmitter module hardware. For the applications discussed here, an interest in methods which utilize signal strength decay distance estimation using a single, omnidirectional antenna is shown as RSSI is made available to the user on most transmission hardware. A solution involving statistical properties of RSSI when operating within an indoor environment undergoing multipath fading was desirable; however, models which describe multipath are designed for long range, cellular communications or assume various properties about the signal scattering within the environment. The Ricean model assumes when a line of sight signal is present, there is a dominating component that is found when compared to all other paths, which are Gaussian distributed. The Rayleigh distribution assumes no dominate component is present, modeling the system with a modified Gaussian. These models present two problems for use in RSSI ranging. These models represent signal envelope amplitude, not average power for the entire signal. While some methods have modified these distributions to do so, they have not achieved high resolution results, with errors up to four meters. The larger issue is the Gaussian assumption. Because multipath interference is so dependent on the geometry of the environment, the effects on the signal can be incredibly biased depending on the paths taken. However, no practical statistical model can account for environment geometry and impedance of materials. For this reason, an approach which utilized the odometry available from the mobile robot to detect multipath through changes in RSSI which far exceed the expected values determined from changes in distance.

To develop localization algorithms for functioning in highly reflective indoor environments, a simulation method had to be defined. No simulator well modeled multipath interference indoors without excessive computation, which generates extremely long run times for a robotics localization algorithm. To solve this problem, a Markov chain was trained to generate multipath attenuated RSSI data from collected data gathered from the actual environments of interest. The chain would generate RSSI over distance plots which can be used as a lookup table during simulation for finding RSSI values a certain distance away from a node. This method proved useful in generating realistic data for testing algorithms without tedious computations involving path simulation.

To account for multipath interference, the trust filter was created. The trust filter is a weighting system, which estimated distance to each node from comparing a predicted distance determined from the robot's motion and the measured distance from RSSI values. If the change in predicted distance closely matched the measured distance, then a high trust factor is assigned and the distance estimation closely follows the measured value. If the measured value changes rapidly compared to the predicted value, then the trust factor is low, relying on the predicted values for the distance estimation. This method proved reliable in simulation, yielding an approximate 1 meter resolution in most cases. This method proved more reliable than the method presented by Menegatti, which did not incorporate the changes in distance with respect to landmark. However, similarly to Menegatti's method, this method tends to be highly reliant on the accuracy of the odometry sensors. The predicted distance which is used to determine the trust factor may only be valid if the odometry is fairly reliable over the distance increment in which measurements are taken. If the odometry measurement is inaccurate, but also matches with the measured distance from a multipath-attenuated distance measurement, the trust filter will incorporate an erroneous measurement into the distance estimate. Additionally, if the environment produces multipath interference at all distances, the trust factor will strictly rely on odometry for distance estimates, as the trust factor will be so low the weight of the RSSI-distance measurement will be negligible. As a result, the EKF will be using the current values of the state vector for distance estimation, causing no new information to be introduced to the filter, and the robot's estimated position will simply drift off inaccurately following the odometry measurements.

8.2 Future Work

Further research on this topic includes utilizing a hardware-based solution for multipath interference. Ultra wideband solutions have proven to provide a much more robust distance estimation due to the ability to filter out multipath by examining how each frequency is attenuated for the same transmission. Despite the additional costs associated with utilizing uncommon modulator hardware, the precision allows focus to be directed towards solving other problems in range-only SLAM applications. Once an ultra wideband ranging solution is in place, research efforts will be directed toward adapting other SLAM algorithms for range-only applications, such as FastSLAM, which due to its robustness for poor initial state information may provide solutions which remove the need for an initialization stage for unlocalized wireless landmarks. Another aspect of wireless beacon localization is inferring orientation over time. Direct signal strength measurements do not provide any information regarding orientation, but aggregating distance estimates over time may provide insight into heading information of the mobile robot.

REFERENCES

[1] S. Sain, “Modelling and Characterization of Wireless Channels in Harsh Environments,” *Thesis, Mälardalen University*, no. 1154, pp. 1–36, 2011.

[2] B. Sklar, *Digital Communications: Fundamentals and Applications.* New York : Prentice Hall, 1988.

[3] “Omni antenna vs. directional antenna - cisco.” `http://www.cisco.com/c/en/us/support/docs/wireless-mobility/wireless-lan-wlan/82068-omni-vs-direct.html`. (Accessed on 03/28/2017).

[4] “Inverse-square law - wikipedia.” `https://en.wikipedia.org/wiki/Inverse-square_law`.

[5] “Two-ray ground-reflection model - wikipedia.” `https://en.wikipedia.org/wiki/Two-ray_ground-reflection_model`. (Accessed on 03/28/2017).

[6] H. Farahani, *ZigBee Wireless Network and Transceivers.* 2008.

[7] S. Thrun, “Particle Filters in Robotics,” *Proceedings of Uncertainty in AI*, vol. 1, pp. 511–518, 2002.

[8] G. Kantor and S. Singh, “Preliminary results in range-only localization and mapping,” *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 2, no. May, pp. 1818–1823, 2002.

[9] S. Riisgaard and M. R. Blas, “SLAM for Dummies,” *A Tutorial Approach to Simultaneous Localization and Mapping*, vol. 22, no. June, pp. 1–127, 2004.

[10] C. Steffes and S. Rau, “Multipath detection in TDOA localization scenarios,” *2012 Workshop on Sensor Data Fusion: Trends, Solutions, Applications, SDF 2012*, no. September, pp. 88–92, 2012.

[11] By and M. Viswanathan, “Simulation of Digital Communication Systems using Matlab,” p. 292, 2013.

[12] I. Poole, “Electromagnetic waves reflection, refraction, and diffraction :: Radio-electronics.com.” `http://www.radio-electronics.com/info/propagation/em_waves/electromagnetic-reflection-refraction-diffraction.php`. (Accessed on 03/28/2017).

[13] E. S. S. Suthersan, B. Raton, and C. R. C. Press, “Sklar, B. "Rayleigh Fading Channels",” in *Mobile Communication Handbook*, pp. 1–40, 1999.

[14] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication.* Cambridge University Press., 2006.

[15] D. Puccinelli and M. Haenggi, "Multipath fading in wireless sensor networks," *Proceeding of the 2006 international conference on Communications and mobile computing - IWCMC '06*, p. 1039, 2006.

[16] "dbm - wikipedia." `https://en.wikipedia.org/wiki/DBm`. (Accessed on 03/28/2017).

[17] H. Friis, "A Note on a Simple Transmission Formula," *Proceedings of the IRE*, vol. 34, no. 5, pp. 254–256, 1946.

[18] J. S. Seybold, *Introduction to RF Propagation.* John Wiley and Sons, 2005.

[19] S. Lanzisera, D. T. Lin, and K. S. J. Pister, "RF Time of Flight Ranging for Wireless Sensor Network Localization," *2006 International Workshop on Intelligent Solutions in Embedded Systems*, pp. 1–12, 2006.

[20] M. Popa, J. Ansari, J. Riihijarvi, and P. Mahonen, "Combining Cricket System and Inertial Navigation for Indoor Human Tracking," *2008 IEEE Wireless Communications and Networking Conference*, pp. 3063–3068, 2008.

[21] V. Honkavirta, T. Perälä, S. Ali-Löytty, and R. Piché, "A comparative survey of WLAN location fingerprinting methods," *Proceedings - 6th Workshop on Positioning, Navigation and Communication, WPNC 2009*, vol. 2009, pp. 243–251, 2009.

[22] A. Awad, T. Frunzke, and F. Dressler, "Adaptive distance estimation and localization in WSN using RSSI measures," *Proceedings - 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, DSD 2007*, no. Dsd, pp. 471–478, 2007.

[23] S. Shue and J. M. Conrad, "A survey of robotic applications in wireless sensor networks," *Southeastcon, 2013 Proceedings of IEEE*, pp. 1–5, 2013.

[24] W. Murphy and W. Hereman, *Determination of a position in three dimensions using trilateration and approximate distances.* PhD thesis, 1995.

[25] M. Tanikawara, K. Ohba, Y. Kubo, and S. Sugimoto, "The rice distribution and ML positioning using received signal powers in sensor networks," *International Journal of Innovative Computing, Information and Control*, vol. 8, no. 3 B, pp. 2359–2374, 2012.

[26] S. Hara, D. Zhao, K. Yanagihara, J. Taketsugu, K. Fukui, S. Fukunaga, and K. Kitayama, "Propagation characteristics of IEEE 802.15.4 radio signal and their application for location estimation," *Vehicular Technology Conference, 2005. VTC 2005-Spring. 2005 IEEE 61st*, vol. 1, no. c, pp. 97–101, 2005.

[27] N. A. Dieng, M. Charbit, C. Chaudet, L. Toutain, and T. Ben Meriem, "A multipath data exclusion model for RSSI-based indoor localization," *Wireless Personal Multimedia Communications (WPMC), 2012 15th International Symposium on*, pp. 336–340, 2012.

[28] J. O'Hollaren and D. Shell, "Incremental Multi-Robot Deployment for Line-of-Sight Chains Using Only Radio Signal Strength," *ICRA 2010 Multi-Robot Autonomy Workshop*, 2010.

[29] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics.* Cambridge, Massachusettes: MIT Press, 2006.

[30] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based SLAM with Rao-Blackwellized particle filters by adaptive proposals and selective resampling," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, pp. 2432–2437, 2005.

[31] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," *Proc. of 8th National Conference on Artificial Intelligence/14th Conference on Innovative Applications of Artificial Intelligence*, vol. 68, no. 2, pp. 593–598, 2002.

[32] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "The SLAM problem: A survey," *Frontiers in Artificial Intelligence and Applications*, vol. 184, no. 1, pp. 363–371, 2008.

[33] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, vol. 82, no. 1, p. 35, 1960.

[34] R. Faragher, "Understanding the basis of the kalman filter via a simple and intuitive derivation [lecture notes]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 128–132, 2012.

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

[36] A. Torres-González, J. R. Martinez-De Dios, and A. Ollero, "Efficient robot-sensor network distributed SEIF range-only SLAM," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1319–1326, 2014.

[37] C. M. Bishop, *Pattern Recognition and Machin Learning.* Springer-Verlag New York, Inc., 2006.