Indoor Localization via Maximum Likelihood using Low-Energy Bluetooth iBeacon™Technology

A Major Qualifying Project

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for the

Degree of Bachelor of Science in Electrical and Computer Engineering

By:

Steven Loftin
Sean St.Pierre
Evan Hasenfeld
Wei Chen

Advisor: Professor Kaveh Pahlavan

Co-Advisor: Professor Shamsur R Mazumder

Table of Contents

Executive Summary

Indoor Localization is a fast growing sector in the wireless technology field. Most commercial applications of localization utilize Wifi signals in the 2.4GHz to 5.2GHz range. The goal of our project is to test if Bluetooth Low Energy (BLE) beacons can be used for indoor localization. BLE beacons transmit significantly weaker signals than traditional wifi units. The transmitted power used in our system of -12dBm allows for a max range of about 7 meters, compared to wifi which is able to accurately cover 20 to 35 meters indoors.

With the increased development of BLE sensors, more applications are attempting to utilize them. The sensors being tested in this project are the iBeacon™ Location Beacons. These beacons broadcast a 2.4GHz signal which can be read by most wireless monitoring tools. The goal of this project is to read the received signal strength (“RSS”) of each beacon and from that determine a location in a room. These RSS readings (dBm) can be used with a path-loss model to determine the approximate distance from each beacon. This is done by taking the RSS and passing it through a localization algorithm. For the algorithm to work properly a path-loss model is needed. A path-loss model is a way of describing the fading of a sensor network over a given distance. Once the model is found and the algorithm is implemented the distance can be extracted from the raw RSS data.

With all of the distances found, the sensor network of beacons, gives what it predicts is the distance the user is from each beacon. The algorithm’s job is to take all of these distances and find where the user is in the room. The algorithm is then compared to the Cramer-Rao Lower Bound(CRLB) to see the overall accuracy of the algorithm. Each algorithm has a different method of finding the user’s location, the algorithm primarily focused on in this paper is a newly developed algorithm called the Centroidal Axis algorithm also known as “Maximum Likelihood”. Other algorithms are mentioned but Maximum Likelihood provided the best results for our indoor localization system.

The application side of this project developed into an algorithm testing tool. The application does all of the computations on-board the phone and displays the predicted user location, the error of the predicted to the actual, the sigma (Standard Deviation) and alpha (Gradient Factor) values and finally allows for the locations of the beacons to be changed. This is helpful for testing new algorithms in different locations and with different path-loss models. This application also provides a solid “skeleton” for a user facing application which displays location of the user on the screen inside a given environment.

Abstract

The objective for this project is to further develop location-based algorithms along with a specialized phone application using iBeacon™ technology. This is done in an effort to aid in the tracking of people's distance from specific exhibits within a museum, track the room they're in, and provide context on the exhibits throughout the museum. This information can then be compiled and sent to a cloud-based server.

The project began with wide-ranging data collection in several different environments including classrooms, several museum rooms of differing sizes. This data collected, based on different transmission powers and differing distances, allowed us to obtain several pathloss models and to calculate standard deviation as well as shadow fading of the iBeacon™ devices.

The next part of the project depended upon the creation of a phone application that was capable of detecting the signals from the iBeacon™. To accomplish this, the phone application was developed for an Android Phone with Bluetooth® Low Energy Capabilities. The application could then use the signals surrounding it to triangulate its position, predicting the location of the phone with relative accuracy in the room.

Acknowledgements

During the terms for our MQP projects, we received support from a numbers of individuals. We would, first and foremost, like to give our thanks to both of our advisor, Professor Kaveh Pahlavan, our main advisor, and his Co-Advisor, Professor Shamsur R Mazumder . Professor Pahlavan provided us almost all of the necessary background knowledge included within our project, introducing various methods and answered any technical questions we had. Professor Mazumder gave us many useful suggestions on technical writing, edits on our report, and made our report more professional through countless reviews.

Secondly, we would like to thank the graduate students of Professor Pahlavan; the supportive help of Julang Ying as well as all the members of CWINS Labs for their cooperation during this project, whom held several workshops throughout the terms and provided several algorithms; including his help with location algorithm testing and explanations of theories, which set us on the track to our final product during the project.

Finally, we would like to thank WPI and the Electrical and Computer Engineering department for providing us with the resources, funding, and locations, necessary to allow us to complete this project. Much of this would not be possible without their help in providing guidance, background knowledges, as well as technical supports.

List of Figures

List of Tables

Authorship

Table 1: Authorship

Team Member:	**Sean**	**Evan**	**Steven**	**Wei**
Primary:	Executive Summary, Introduction, Background, System Development (Algorithms), System Development (CRLB), Results, Recommendation, Conclusion and Future direction	Abstract, Acknowledgements, Introduction, Background, System Development (CRLB), Results, Recommendation, Conclusion and Future direction, Appendices	Introduction, Background, System Development (Algorithms), Application Architecture, Results, Recommendation, Conclusion	Acknowledgements Background
Editor:	Application Architecture,	Executive Summary,	Abstract, Introduction, Background, System Development	System Development (CRLB) Introduction

Chapter 1: Introduction

1.1 Introduction

Indoor localization is becoming a prominent focus in the technology industry. Wireless technology is becoming more integral in everyday life, and through movements such as the internet of things, the wireless industry is finding a new way to improve life for their consumers. Indoor localization is something companies have been working towards for some time. The ability to locate a device inside a structure or closed environment where traditional GPS does not function has always been desired for use in activities such as shipment tracking, patient informatics, and navigation.

1.2 Motivations

The motivation behind this project was to test and improve the “Maximum Likelihood” algorithm for use in indoor, Bluetooth-based localization.. Indoor localization is an emerging market and many groups are working on viable implementation methods. While indoor localization has been attempted to some success using Wi-Fi signals[16], we wanted to attempt to locate a position in the room using solely Bluetooth and the maximum likelihood algorithm. Given a phone with Bluetooth Low-Energy capabilities and a setup of four to six iBeacons, our goal is to find the phone’s location within ~1 meter within a short period of time. So to sum up our goals

1. Test and Implement Localization Algorithms.
2. Transfer the “best” algorithm to a hand held device(e.g Cell Phone)
3. Design an application for indoor localization(e.g Business centers, schools, museums).
4. Test if Bluetooth® is viable for localization applications.

1.3 Overview of systems

The goal of this project has evolved over time. Initially it was to create a guided museum tour through one of the local art museums. It has been adapted into an indoor localization algorithm testing tool, and the implementation and testing needed to examine its quality. We believe that three groups may be interested in the research and final deliverables of this project: institutions and research groups, application developers, and companies invested in Bluetooth technologies.

The initial project format from the customer's viewpoint was simple, just download an application and walk through the museum. Each time they approached an exhibit, data would pop up in the application. This data would have included information such as a brief description of the exhibit, other works by the artist, or any other relevant information provided by the museum.

From a technical viewpoint, the technology within the project remained the same. The application will be using a maximum likelihood localization algorithm and a path loss model to determine a receiver's position inside a given room. Knowing a receiver's position can be used in any number of applications, such as tracking or navigation, and thus the use of the gathered information is left open to the end user.

It was decided as a group to move the project towards a more generalized implement-and-test method after numerous difficulties were encountered using a museum tour as the specific implementation. We feel that the information we have gathered over the tests we performed is sufficient to draw conclusions about the algorithms used within their respective environments. Chapter 6, section 2 contains recommendations for future groups or interested entrepreneurs, as the work done here is a baseline for using the maximum likelihood algorithm with Low-Energy Bluetooth.

1.4 Description of Report

The following sections include background, methodology, results and future work which can expand the project. Chapter 2 provides background knowledge on the topics of localization, Bluetooth®, iBeacon™, and market research. Chapter 3 discusses the algorithm and the system created to test it. Chapter 4 contains computational analysis of the algorithm and the Android application created to run it. Chapter 5 discusses the results we obtained and Chapter 6 suggests objectives and goals for future work in similar projects.

Chapter 2: Background

In order to understand the work presented in this document we feel it is important to provide a brief introduction and background of the technology used. Much of the work done is based off of work done by previous research groups and projects, as is standard for scientific work. The following section will provide a short overview of the technology that our research is based upon[21][22].

2.1 Bluetooth®

Bluetooth® Low Energy (BLE) is the focus of this project. Bluetooth® is a wireless technology standard using the 2.4GHz ISM band (industrial, scientific and medical band). It is used for exchanging packets of data over short distances between devices such as computers, mobile phones, and iBeacons™ (Bluetooth ®Technology Website).[3] Bluetooth® was invented by Ericsson, a telecom vendor, and was originally conceived as a wireless alternative to the standard at the time: using data cables to transfer the packages. Currently the Bluetooth® company is controlled by the Bluetooth® Special Interest Group, also known as the Bluetooth® SIG, consisting of companies operating in several fields of business including telecoms, networking, and computing (V., Jan 2011).[2]

The Institute of Electrical and Electronics Engineers, IEEE, is an association of professionals formed in 1963 from the merging of the Institute of Radio Engineers and the American Institute of Electrical Engineers. Today, IEEE’s objectives are the educational and technical advancement of electrical and electronic engineering, telecommunications, computer engineering and allied disciplines. (IEEE, 2017) The IEEE standardized Bluetooth® as IEEE 802.15.1. (How it works | Bluetooth® Technology Website) The Bluetooth® SIG, alongside IEEE, oversees development of the specifications for the current era of Bluetooth® technology. In order for any standard practice to be accepted for the advancement of Bluetooth® as technology, any manufacturer who believes they have an advancement must at a minimum meet the Bluetooth® SIG standards to be able to market it as a Bluetooth® device (Bluetooth® Technology Website). [1]

The Bluetooth® SIG completed the Bluetooth® Core Specification version 4.0, including protocols such as Classic Bluetooth®, Bluetooth® high speed and most important for our purposes, Bluetooth® low energy. Bluetooth® low energy is a subset of version 4.0 Bluetooth® with a new protocol stack for rapid build-up of simple links. As an alternative to the Bluetooth® standard protocols that were introduced in version 1.0 and version 3.0 Bluetooth®, BLE is aimed at very low power applications running off a coin cell. The new chip designs allow two types of implementation: a single-mode implementation, which is farther enhanced compared to past versions, and a dual-mode implementation (Pollicino, J., 2016). [5]

2.2 Localization

Location-based services (LBS) is an increasingly popular technology which has has become an integral part of daily life. It is included in both short-range and long-range networks. Depending on the location of a user, applications with LBSs are able to provide services in various categories such as navigation, mapping, healthcare, even payment. The demand for LBSs is increasing significantly with the expansion of the global portable device market.

The basic components of LBS are a software application (provided by the provider), a communication network (mobile network), a content provider, a positioning device, and the end user’s mobile device. There are several ways to find the location of a mobile client indoors and outdoors. The most popular technology for outdoors is Global Positioning System (GPS). (Liu 2010) [4] During the Vietnam War, the United States Department of Defense launched a series of GPS satellites to support localization during military operations in combat areas. Nowadays, GPS technology is ubiquitous in the civilian market to provide personal navigation services. GPS receivers are designed to determine the locations of boats, planes, or mobile vehicles in open areas such as ocean, sky, and highways. However, the accuracy of GPS positioning is significantly impaired in urban and indoor areas, where received signals can suffer from extensive multipath effects and additional path loss. For those situations, alternative coordinates and visualization techniques may be employed to find the location.

An Indoor Positioning System (IPS) is a system that provides a precise position inside of a closed structure, such as mall, hospital, airport, and university campus. Different from GPS which uses satellites, IPS uses radio waves, magnetic fields, acoustic signals, or other sensory information such as Bluetooth® collected by mobile devices. Several commercial systems can be found in the global market, but still, no standard exists for an IPS system. [10]

IPSes use different technologies, including vision-based (using visual information provided by the camera to predict the distance), wireless-based (receiving signals to infer the distance to known points and get the location of current point), and other methods (acoustic background fingerprint). Among all these solutions, wireless-based localization is the most popular due to its low cost and relatively simple hardware.

The wireless-based localization technologies for IPS can be categorized into three sections: long distance wireless technology, middle distance wireless technology, and short distance technology. For long distance, FM (Frequency Modulation) and GSM/CDMA are common since they are cheap and sustainable. For medium distance, Wifi and ZigBee (IEEE 802.15.4 standard) are the mainstream for wireless localization. Short distances implement Bluetooth®, UWB (Ultra-Wide Band) and RFID (Radio Frequency Identification) as major solutions. Bluetooth® , which contains BLE (Bluetooth® Low Energy) mode since 4.0 standard, is especially common. [9]

2.3 iBeacon™

iBeacon™ is a protocol developed by Apple Inc. in 2013 [6]. It is the name for Apple’s technology standard, which allows Mobile Apps (running on both iOS and Android devices) to listen for signals from iBeacons™ in the physical world and react accordingly. In essence, iBeacon™ technology allows Mobile Apps to understand their position on a micro-local scale, and deliver hyper-contextual content to users based on location. The underlying communication technology is Bluetooth Low Energy [7]. This specific BLE beacon device can be used for many purposes, most importantly in our case, iBeacon™ can be used for indoor positioning system and proximity-based information transfer systems.

Estimote Location Beacons hold a good balance of affordable price and a substantial suite of features including an official API for ranging, large sets of sample code on public websites for development, long battery life up to 5 years, and variable broadcast power for the team to set up for suiting the needs of the project. Estimote Beacons have also already been used in a variety of real-world applications, such as portraits identification in museum and bus service.[8] These features make Estimote Beacons the ideal tools for indoor localization project.

Table 2: iBeacon Location Beacon Technical Specifications

Identification (Hardware revision)	F3.3
MCU	Bluetooth® SoC ARM® Cortex®-M4 32-bit processor with FPU 64 MHz Core speed 512 kB Flash memory 64 kB RAM memory
Radio: 2.4 GHz transceiver	Bluetooth® 4.2 LE standard Range: up to 200 meters (650 feet) Output Power: -20 to +4 dBm in 4 dB steps, “Whisper mode” -40 dBm, "Long range mode" +10 dBm Sensitivity: -96 dBm Frequency range: 2400 MHz to 2483.5 MHz No. of channels: 40 Adjacent channel separation: 2 MHz Modulation: GFSK (FHSS)

	Antenna: PCB Meander, Monopole Antenna Gain: 0 dBi Over-the-air data rate: 1 Mbps (2 Mbps supported)
Sensors	Motion sensor (3-axis) Temperature sensor Ambient Light sensor Magnetometer (3-axis) Pressure sensor EEPROM Memory 1 Mb RTC clock
Additional features	GPIO NFC
Power Supply	4 x CR2477 – 3.0V lithium primary cell battery (replaceable) High efficient Step-Down DC-DC converter
Environmental Specification	Operating Temperature: 0°C to 60°C (32°F to 140°F) Storage Temperature (recommended): 15°C to 30°C (59°F to 86°F) Relative Humidity (operating): 20% to 80% relative humidity Relative Humidity (storage): 10% to 90% relative humidity, non-condensing Splash-proof
Materials	non-flammable enclosure: silicone adhesive layer: double-sided adhesive tape
Size and Weight	Length: 62.7 mm (2.47 inches) Width: 41.2 mm (1.62 inches) Height: 23.6 mm (0.93 inches) Weight: 67g (2.36 ounces)

2.4 Market Research

Market research shows that the iBeacon™ has been implemented in several places. Thus far iBeacons™ have been used for location and NFC (Near Field Communications) applications. Stores have implemented NFC applications to display targeted advertisements to customers based on their in-store location. Industries have implemented Bluetooth® beacons to assist in tracking units in a warehouse or a shipping environment. Companies have used similar devices to survey attendee location “hot spots”. In the medical field, iBeacon™ is used to track Doctor-Patient interaction time, to evaluate if enough care is being provided.

One example of applied iBeacon™ technology is the Near Me feature of Guggenheim app. This feature was introduced on December 11, 2015, for the app used by Solomon R. Guggenheim Museum. [11] By setting up over one hundred Bluetooth® Low Energy iBeacons™ in the Frank Lloyd Wright building, those iBeacons™ were able to transmit signals at ranges that vary from five to fifty feet in order to support the visitor experience. As the visitors using the Near Me app inside the range of an iBeacon™, content associated with that iBeacon™ becomes available. When a visitor opens Near Me, the screen may display information about nearby artworks and exhibitions. [12]

Our product is using iBeacon technology as a testing tool to verify localization algorithms. Once the algorithm can be verified our product also provides the underlying support for applications to implement aspects of localization to their device.

Market Analyst predicts that by the year 2022 the indoor localization market will have a value of 40.99 Billion dollars [17]. The growth in the industry has been extensive over the last few years. Analyst have found that the annual compound growth rate (ACGR) is at a rate of 42% over the forecasted period[17].

Chapter 3: System Development

The system created to test and verify the maximum likelihood algorithm in an environment can be split into two parts: the System Development, where the physical architectures, algorithms, and mathematics are discussed, and the Application Architecture, describing the phone application and software written to run the testing. This chapter will provide an in-depth description of the algorithms used, and analysis of the best hardware setup, and some quantitative data about the hardware of the iBeacons themselves. Discussion of the application and software will follow in Chapter 4.

3.1 Understanding iBeacon™ Characteristics

Before any work can begin regarding algorithms it's important to understand the hardware which is being used and implemented. The beacons being used are classified under version 4.2 Bluetooth® LE Standard. Estimote titled this beacon as the "Location Beacon". This beacon has a Antenna sensitivity of -96dBm, which is important to know when solving for maximum pathloss. These beacons have a transmitted power range from -40 dBm to + 10 dBm which is adjustable in 4 dBm increments. It is also important to note that there is no antenna gain that needs to be accounted for. The advertising interval can also be changed, for this project we decided on advertising in 100ms intervals.

3.2 Algorithms

3.2.1 Introduction to Algorithms: Pathloss and RSSI vs. TOA

Localization Algorithms in a very simple explanation are tools that take input data produced by a sensor network and determines the predicted location of the sending device. There are two popular types of localization algorithms TOA and RSSI. TOA stands for Time Of Arrival, these systems typically used pre-mapped areas and have a “third-party” server to monitor when the data was sent versus when the data is received. The other is RSSI based algorithms which uses the Received Signal Strength to determine the distance of the sensor. For our project we are using RSSI based algorithms because, the packet information sent out by the estimote beacons do not contain time sent or time received.

There are a few different types of RSSI based algorithms, the one we have experimented with and tested the most has been Least Mean Square or LMS. The LMS algorithm works by receiving N number of signals, then attempts to find the distance by passing it through the equation.

$$(x_n - x_0)^2 + (y_n - y_0)^2 = d^2 \qquad \textbf{(3-0)}$$

The x_0 and y_0 in this case would be the sensor location in the environment this system is being implemented, and x_n and y_n are the broadcast locations. The value for distance is obtained by passing the RSSI value through the path loss model. From the pathloss model we are able to extract the distance, but these distances are prone to error. The error is measured in the distance from the initial guess to the point the algorithm found it to be. In some cases the algorithm does not converge, when the algorithm does not converge this means that the estimated point is outside of the intersection of the distances.

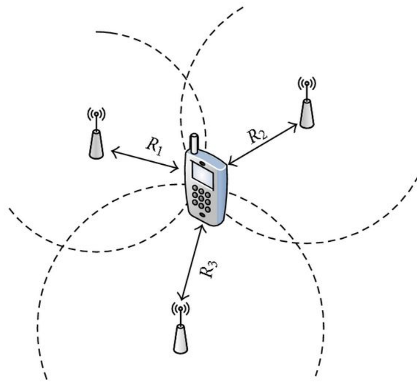

Figure 1: LMS Concept Diagram [14]

The Maximum Likelihood algorithm, the algorithm used in this project, is an RSSI based localization algorithm. This algorithm works by using a scoring method combined with a pre-mapped room. The room that the system is being implemented is broken down into a grid of points. Each point then receives a "score", a score is given to the point in the room from each beacon. The point in the room(reference point) receives a signal from every Beacon, if the signal received at the reference point is within a certain value the point receives a 1 from that beacon. The highest score depends on the number of beacons. The tolerance is determined by finding F(σ)for a determine accuracy, if RSSI(measured) is within of RSSI(calculated) then the signal scores a point at that reference point. Once all of the reference points have received scores from all of the beacons, the centroid is found of all the highest scoring reference points is found and assumed to be the location of the user.

3.2.2 Path Loss Equation

Calculating the distance of a receiver from a signal source can be done with a path loss equation, defined below in equation (3-1). Based on the distance 'd', maximum allowable path loss 'L_P', and measured pathloss at a distance of one meter L_0, the constant α (gradient factor) can be determined, allowing for the calculation of any distance knowing only the received signal strength.[16]

$$L_P = L_0 + 10\alpha \, log_{10}(d) \quad for \; d < d_{BP} \tag{3-1}$$

$$L_P = L_0 + [10\alpha_1 \, log_{10}(d) + 10\alpha_2 \, log_{10}(d/d_0)] \quad for \; d > d_{BP} \tag{3-2}$$

Equation (3-2) is the equation to use when the distance of the system exceeds the given breakpoint distance (d_{BP}) of the system. In equations (3-2) d_0 is the d_{BP} and the d value is the new distance. In order to solve for maximum distance before breakpoint use the following equation[16].

$$d_{[m]} = 10^{\frac{(RSSI - L_0)}{10\alpha}} \tag{3-3}$$

3.2.3 Implementation: Maximum Likelihood[MLE]

When implementing the Maximum Likelihood Algorithm[MLE] there are a few variables that need to be considered. The first is the number of beacons being used in the system. This is important because the amount of beacons determine the size of the "score matrix' which will be discussed shortly. Another variable to consider is the target accuracy denoted by the variable T_P. Equation (3-1) is the complementary error function formula, $\boldsymbol{\gamma}$ denotes the desired percent accuracy, $\boldsymbol{\sigma}$ denotes the standard deviation and F($\boldsymbol{\sigma}$) denotes shadow fading.[16]

$$1-\gamma = 0.5erfc(F(\sigma)/\sigma\sqrt{2}) \tag{3-4}$$

This formula is then solved for F($\boldsymbol{\sigma}$). The solved equation is then compared to the absolute value of the difference of RSSI measured(dBm) versus RSSI theoretical(dBm). Equation (3-5) is the logic used in the scoring of each point.[16]

$$(\left| RSSI_{Measured(dBm)} - RSSI_{Theoretical(dBm)} \right| < \sigma\sqrt{2} * erfc^{-1}(2-(2 * T_P))) \tag{3-5}$$

Equation (3-5) is important because this tells us what the acceptable bounds of RSSI are in order to have a "good" score. Once the room has been divided into a granularity that is suitable for the application, each point in the room is assigned a score matrix. The score matrix is a [1 x $N_{\text{Beacons Used}}$] matrix, if the difference of $RSSI_{\text{Measured}}$ and $RSSI_{\text{Theoretical}}$(found utilizing the pathloss model) is less than F(σ) that beacon receives a score of 1 at that point. This process is done for each beacon, then repeated for each point in the room. Once all the scores have been assigned each points score matrix is summed up and the resultant number is that points “total score”. Below is a simple example on how each section of the room is assigned a score.

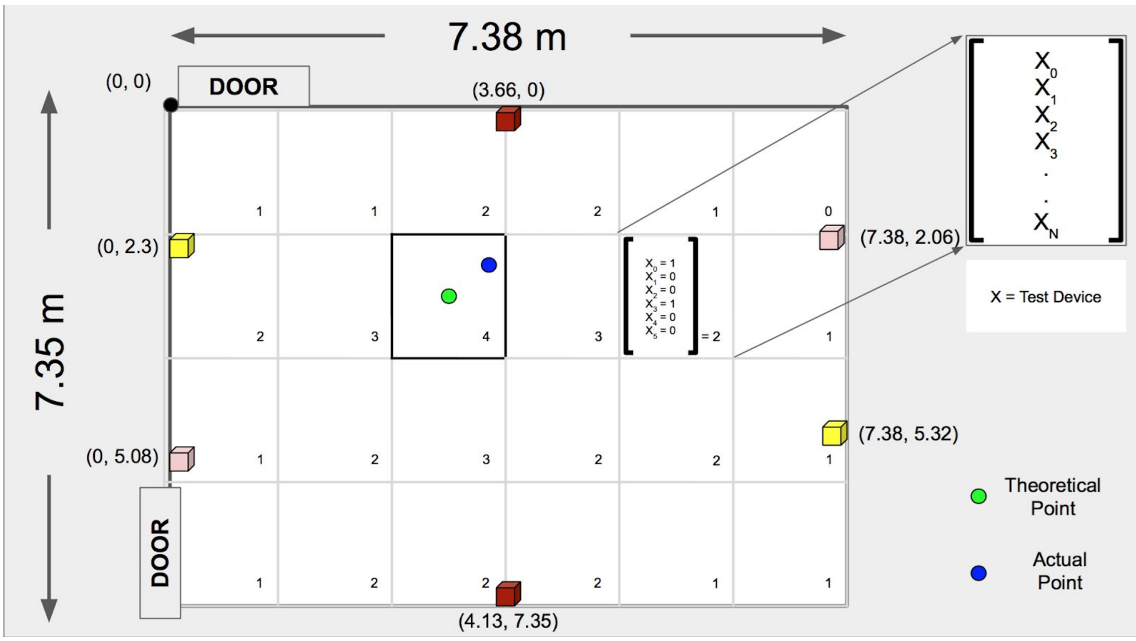

Figure 2 : Maximum Likelihood graphical representation.

In order to find the location that the algorithm predicts the highest scoring points are observed and a polygon is formed. Then the x and y coordinates are extracted using the following equations:

$$x_{Coordinate} = \left[\frac{x_1 + x_2 + x_3 \ldots\ldots + x_n}{2}\right] \qquad \textbf{(3-6)}$$

$$y_{Coordinate} = \left[\frac{y_1 + y_2 + y_3 \ldots\ldots + y_n}{2}\right] \qquad \textbf{(3-7)}$$

The result of these two equations are the location that the algorithm places the user at. From this data the error of the algorithm can be calculated using the following equation:

$$error\ distance = \sqrt{(x_{actual} - x_{calculated})^2 + (y_{actual} - y_{calculated})^2} \qquad \textbf{(3-8)}$$

In the equation above x $_{actual}$ and y $_{actual}$ are the coordinates of the user in the room, as where x $_{calculated}$ and y $_{calculated}$ are where the algorithm thinks the user is.

The average error in distance is a good indicator on the overall accuracy of the algorithm.

3.2.4 Implementation: Least Mean Square [LMS]

The Least Mean Square algorithm calculates the theoretical x and y values by taking references points(RP) and logging their x and y coordinates in the following fashion. [16]:

$$F = [f_1(x,y), f_2(x,y), f_3(x,y) \ldots\ldots f_n(x,y)]^T \tag{3-9}$$

Once this data has been logged the Jacobian matrix of F which is noted a **J** is form. The Jacobian is constructed in the following fashion[16]:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial f_1(x,y)}{\partial x} & \frac{\partial f_1(x,y)}{\partial y} \\ \cdots & \cdots \\ \frac{\partial f_N(x,y)}{\partial x} & \frac{\partial f_N(x,y)}{\partial y} \end{bmatrix} \tag{3-10}$$

The equation then can be started from an arbitrary point, this point is denoted by the following equation[16]:

$$l(n) = [x(n),\ y(n)] \tag{3-11}$$

This point can then be iterated by the following form:

$$l(n+1) = l(n) + E_n \tag{3-12}$$

Where:

$$E_n = -(J^T J)^{-1} J^T F \tag{3-13}$$

The E_n denotes the error in the solution or how much the algorithm needs to correct in order to move closer to where the theoretical location is. Equations (3-9) is where to location of all the Beacons or RP's would be entered. This algorithm works by starting at an arbitrary point then solving systems of quadratic equations to slowly converge in an area where all of the beacon's coverage radii intersect. Below is a graphical representation of the approximation process[16].

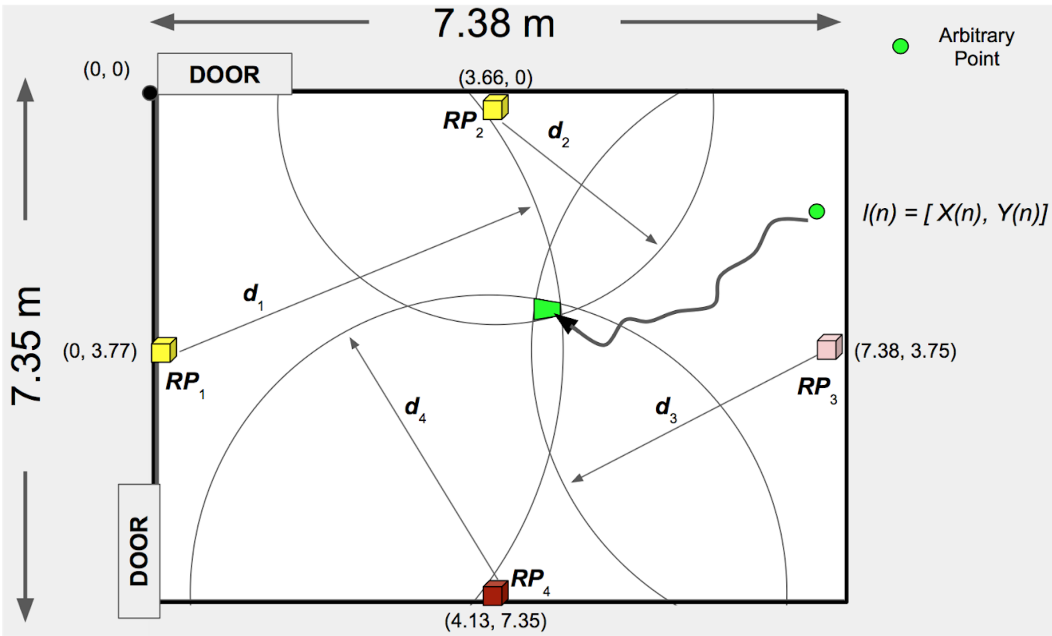

Figure 3 : Least-Mean Square Graphical Representation

Unlike the Maximum Likelihood Algorithm, Least Mean Square won't always work as show in Section 5.2.1. For LMS to work the arbitrary point must converge to an area where all of the RP's coverages intersect at that given RSSI value. It is possible that not all of the circles will intersect and the algorithm will never converge. Non-convergence is a limiting factor to the LMS algorithm and controlling this is key to proper implementation.

Since the calculations are ultimately done on a hand held device runtime of the algorithm is something to be strongly considered. Later in Chapter 4 and Chapter 5, we discuss why we selected the MLE algorithm over the LMS algorithm and how runtime efficiency was calculated and the role it played in the project.

3.3 Cramer-Rao Lower Bound

3.3.1 CRLB Background Knowledge

In our localization system, we need to implement an algorithm to compare the accuracy of various alternatives for localization. Cramer-Rao Lower Bound (CRLB) is able to measure the spread of the error associated with a location estimate, for comparing the precision of location estimations by alternative approaches for localization. The smaller the variance, the smaller is the chance that the error in location estimate is large. CRLB provides ideal values of error so that we can compare them with the collected data under the algorithms we used to see which algorithm provides better results.

3.3.2 Application of CRLB on RSSI localization

For single observation, which is noted by an O, corrupted by zero mean Gaussian Noise, the observation power, which is noted by an P_{r_i}, is [16]:

$$O = P_{r_i} = P_0 - 10\alpha log(r) + X \tag{3-14}$$

The probability distribution function of the observation is:

$$f(O/r) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(O-P(r))^2}{2\sigma^2}}$$

By applying to Fisher Matrix, it turns to [16]:

$$F = -E[\frac{\partial^2 ln\, f(O/r)}{\partial r^2}] = E[\frac{\partial ln\, f(O/r)}{\partial r}]^2 = \frac{(10)^2\alpha^2}{(ln\ 10)^2\sigma^2 r^2} \tag{3-15}$$

And the CRLB will be:

$$F^{-1} = \frac{(ln\ 10)^2}{(10)^2}\frac{\sigma^2}{\alpha^2}r^2 \Rightarrow \sigma_p = \frac{ln\ 10}{10}\frac{\sigma}{\alpha}r \qquad \textbf{(3-16)}$$

As α stands for path loss variable, σ stands for variance and r for distance. The equation reveals that the spread of error goes one positive ratio with distance. [16]

For multiple observations (Access Points) the observation power, is:

$$O = P_r = P_o - 10\alpha log(r_i) + X \quad i = 1, 2, 3, 4, \ldots N \qquad \textbf{(3-17)}$$

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \qquad \textbf{(3-18)}$$

By differentiating it:

$$dP_i(x, y) = \frac{-10\alpha_i}{ln10}\left(\frac{x-x_i}{r_i^2}dx - \frac{y-y_i}{r_i^2}dy\right) \qquad \textbf{(3-19)}$$

In vector form, the relation between dP and dr would be:

$$dP = Hdr \Rightarrow dr = (H^T H)^{-1} H^T dP \qquad \textbf{(3-20)}$$

Where:

$$dP = \begin{bmatrix} dP_1 \\ dP_2 \\ \cdot \\ \cdot \\ dP_N \end{bmatrix};\ dr = \begin{bmatrix} dx \\ dy \end{bmatrix};\ H = -\frac{10}{ln10} I_x[\alpha_1 \ldots \ldots \alpha_N] \begin{bmatrix} \frac{x-x_1}{r_1^2} & \frac{y-y_1}{r_1^2} \\ \vdots & \vdots \\ \frac{x-x_N}{r_N^2} & \frac{y-y_N}{r_N^2} \end{bmatrix} \qquad \textbf{(3-21)}$$

Assuming each access point is corrupted by independent zero mean Gaussian Noise, we get [16]:

$$E\{|dP|^2\} = cov(dP_i, dP_j) = \sigma_p{}^2(i = j) \parallel 0\ (i \neq j) \quad i = 1, 2, 3, 4, \dots N \qquad \textbf{(3-22)}$$

$$E\{|dr|^2\} = cov(dr) = \sigma_0{}^2(H^\tau H)^{-1} \Rightarrow CRLB = Trace[\sigma_0{}^2(H^\tau H)^{-1}] = \sigma_x{}^2 + \sigma_y{}^2 = \sigma_r{}^2 \qquad \textbf{(3-23)}$$

$$F = E\{|dr|^2\}^{-1} = \frac{H^\tau H}{\sigma_p{}^2} \qquad \textbf{(3-24)}$$

By applying the equation in Matlab we can plot spectrum analyzer and use the graph to identify the effect of location of access points, their value of variance for the spreading of errors inside the selected space. Usually, the access points in the middle holds the least variance of error, the points attaches the side have worse results and those corner ones are the worst. Also, as the distance between those access point increases, the variance of the error went up and vice versa. Additionally, as more access points are added, the area that holds less variance of error expanded, those areas with measurement issues such as corners and sides are improved.

Chapter 4: Application Architecture

In order to test the viability of the maximum likelihood algorithm in the field, an Android application was developed. The application took readings in real-time, ran the maximum likelihood algorithm with that data, and displayed results to the screen, such as error, position, and individual beacon readings.

The application was written for an Android device using a mix of Java for the software and XML for the user interface design and settings. The application may be rewritten for an iOS device, as it contains only simple mathematics and calls to the bluetooth API that most phones have, but the project timeline did not support making a cross-platform implementation. The pseudo-code for the maximum likelihood algorithm can be found later, in section 4.3, and the full application code may be found in the public repository located at the URL found in Appendix I.

4.1 Application Flow

The testing application runs in a fairly linear fashion. There are few control statements, allowing the tool to continuously collect data and publish it to the screen. Below, in Figure 4, is a flow chart of the application from the moment it is opened on a user's phone.

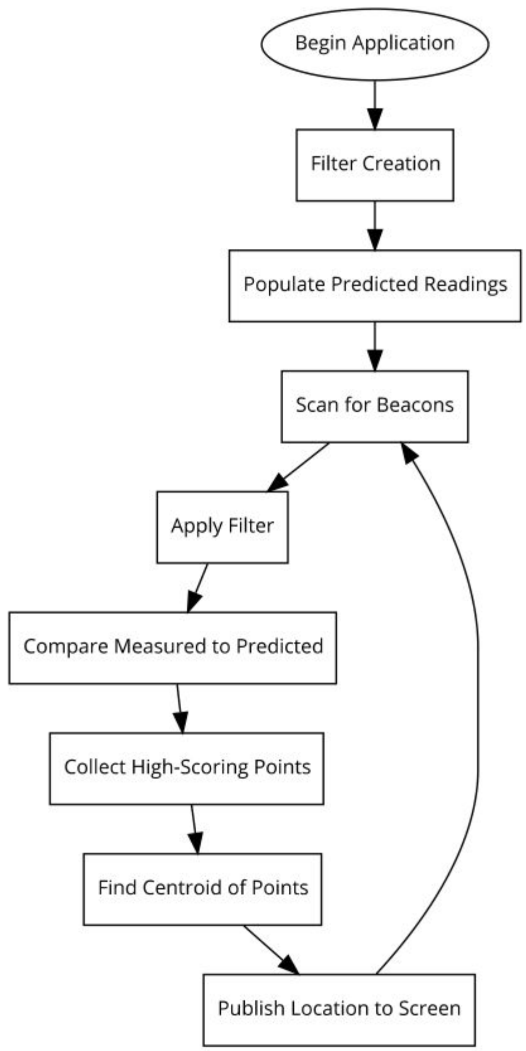

Figure 4: Test Application Flow Chart

The application first creates a filter to be used with the bluetooth readings it takes. This filter is used in order to isolate iBeacon™ signals by removing any packets that do not have the correct manufacturer ID. This ID is a number consisting of the first four bytes of the packet transmitted, and is different for each manufacturer of bluetooth technology (a full list of these IDs can be found through bluetooth.com). This filter is easily changed by simply changing the few bytes in the code, allowing for this application to be adapted to pick up any signals the user wishes.

Before scanning is performed, the application populates its matrix of predicted RSSI readings. Methods for population of this array and the following mathematics can be found in section 4.3. This step only needs to be run once, after the room is set up and the iBeacon™ locations are known. Changing anything about the physical implementation of the room requires that the setup be redefined for the application.

The application scans for around 500 *mS* of each second. After taking readings of each Bluetooth® signal it can see, it filters out the signals that do not match the correct manufacturer code. Each beacon in the room uses a different minor ID, which is a tag used to identify individual beacons. The read RSSI values are associated with their minor IDs, in order to begin the maximum likelihood algorithm. This algorithm is discussed in-depth in section 4.1, so the inner-working will no be discussed here.

After the maximum likelihood algorithm determines the predicted point within the room it displays the coordinates to the screen. This streamlined process of taking readings and getting a predicted location every second means that any setup of a room wishing to use maximum likelihood can be easily tested. Error can be found by simply taking the distance between the predicted and actual point, and beacons can be moved and constants changed in order to determine the most successful implementation.

4.2 Time and Space Efficiency

In order to maintain the real-time element of updating a user on their location within a room, the work done between readings must not exceed the frequency of updates minus the time taken (in milliseconds) to take the readings.

$$MLE\ Runtime_{[s]} \leq 1_{[s]} / f_{poll} - t_{poll[s]} \quad \textbf{(4-1)}$$

By default this means that the maximum likelihood algorithm must complete and return a predicted point in half a second or less. The time efficiency of the algorithm can be defined as:

$$Time\ Efficiency = O(mn\tfrac{p}{q}) \quad \textbf{(4-2)}$$

where *m* and *n* are the dimensions of the room in meters, *p* is the number of beacons that readings are taken from, and *q* is the smallest unit of measurement. The application uses *q=0.01*, giving 1 centimeter of granularity to the calculations. In the case of the room we most commonly tested in, this means that the algorithm would run its most common instruction (comparing read and predicted RSSI values) about 325 million times. The comparison itself is not time intensive, however, so this can be run very quickly.

The space efficiency of the algorithm is roughly equivalent to its time efficiency. As the size of the room increases, the granularity increases, or the number of beacons increases, the time and space taken increase multiplicatively.

Although the algorithm requires that calculation and comparison of read RSSI values be performed a very large number of times, any phone with a decent processor should be able to handle the task. Older phones or slow computers may have difficulty running the algorithm in a real-time application, such as localization while moving. The phone that ran the algorithm on it used a 1.8GHz processor and put all of the work on one core, where if desired much of it could be done through multiple cores due to the independence of the information used in the calculations.

4.3 Generalized Pseudocode

Below is some pseudocode to assist with visualizing how the algorithm works. Previous section have gone into detail about what is done, and should be relied on for a more in-depth explanation. The code below is to demonstrate a generalized form of the algorithm for use in programming.

```
smallest_measurement = 0.01 // 1 centimeter
predicted_readings = [x_dimension by y_dimension by number of beacons]
scores = [x_dimension by y_dimension]
for each point in the room
      for each beacon
            predicted_readings(beacon) = Predict(location)

get readings from beacons
for each point in the room
      for each beacon
            compare read value to predicted value
            if within tolerance
                  increment score total for that point in the room

add all x location of all highest scoring points, divide by # of them
add all y location of all highest scoring points, divide by # of them

return the found x,y
```

4.4 Libraries and Environments

The application was created with use of external libraries and development environments which should be mentioned, as the code cannot be run without them. Where replacements are known they are mentioned. Furthermore, this section cannot cover any of the necessary accommodations for the transferring of the application to iOS devices. While no Android-specific features are exploited, it cannot be certain that the application will function the same on a different operating system.

The AltBeacon library[18], which is an alternative to the standard Estimote library, provides the API to access the data read from the Bluetooth® signals, such as RSSI, IDs, and other data they broadcast. This library was chosen based on the extensive examples given on their website, not due to shortcoming of Estimote's library. It is certain that the same results could be achieved with a different library, however AltBeacon was chosen early in the design process and we found no reason to switch.

The second portion of the app which is necessary to its architecture is Gradle and the Android Studio environment[18]. The structural files and scripts used in building are an integral part of the application, and needed in order for the code to compile into an APK. For this reason the code for the application cannot be fully posted in the document, however a link to the public repository on GitHub can be found in Appendix .

Chapter 5: Results

5.1 Preliminary Results

In order to begin working with the iBeacon™ technology which the project is based on, initial data about the behavior of the iBeacons™ was taken. While the algorithm that the AltBeacon library uses for calculating distance may work for general usage, we wanted to find an improved algorithm which would work better for indoor localization. This data was taken with the intent of developing a model of the iBeacon™ that would be accurate for our project, rather than assuming the parameters specified by the iBeacon's™ manufacturer.

As soon as we began taking readings it was instantly noticed that the readings are quite sporadic and can fluctuate values of up to (+-) 20 dBm. So in order to counteract this we enacted a smoothing technique for the raw received data. The smoothing technique is to convert the received signal power into milliwatts and then average the milli-watt power then convert back to dBm. This is done using the following equations:

$$P_{[dBm]} = 10^{\frac{P_{r[dBm]}}{10}} \quad \textbf{(5-1)}$$

Then the readings are averaged using a simple averaging algorithm below:

$$P_{avg[mW]} = \frac{P_{r0[mW]} + P_{r\,1[mW]} + P_{r\,2[mW]} \text{........} + P_{r\,n[mW]}}{Number\ of\ Readings\ in\ Set} \quad \textbf{(5-2)}$$

The the average received power in milliwatts (P $_{avg}$(mw)) is converted back to dBm using the following equation:

$$P_{[dBm]} = 10Log_{10}[\frac{P_{avg[mW]}}{1mW}] \quad \textbf{(5-3)}$$

Below is an example of how the smoothing technique was used and its results:

Table 3: Data Smoothing Example of Converted Data

Power in dBm		Power in mW	
Beacon 1	Beacon 2	Beacon 1	Beacon 2
-76	-86	2.5118e-8	2.5118e-9
-75	-79	3.1622e-8	1.2589e-8
-76	-80	2.5118e-8	1.0000e-8
-67	-78	1.9952e-7	1.5848e-8
-63	-79	5.0118e-7	1.2589e-8
-68	-76	1.5848e-7	2.5118e-8
-70	-80	1.0000-7	1.000e-8
-69	-81	1.258e-7	7.9432e-9
-70	-73	1.000e-7	5.0118e-8
-79	-84	1.258e-8	3.9810e-9
Average(non converted)		Log Average (After dBm to mW to dBm conversion)	
-71.3(dbm)	-79.6(dBm)	-73.6539(dBm)	-80.9931(dBm)

5.1.1 Estimote's Path-Loss Approximation

The first steps taken towards developing our own model of the iBeacon™ were to take readings based on the official Estimote application for iOS and Android. To collect distance versus signal strength data, an iBeacon™ was placed at a distance from two phones. One phone allowed us to observe the received signal strength through the use of the "BLE Scanner app", and the other was used to read the calculated distance through the "Estimote app". The beacons were set to broadcast at -4dBm in 100ms intervals. The Estimote app stated that a -4dBm broadcast strength would give accurate results to around a 10-meter range, which we determined to be sufficient range for indoor localization.

Table 4: Initial iBeacon Reading Comparisons

RSS(dBm)	App assumed D (m)	Measured D (m)
-38	>1	1
-45	~1	1.5
-49	~1	1.5
-54	~2	2
-55	~2	2.5
-62	~3	3
-70	~3	3.5
-89	~4	4

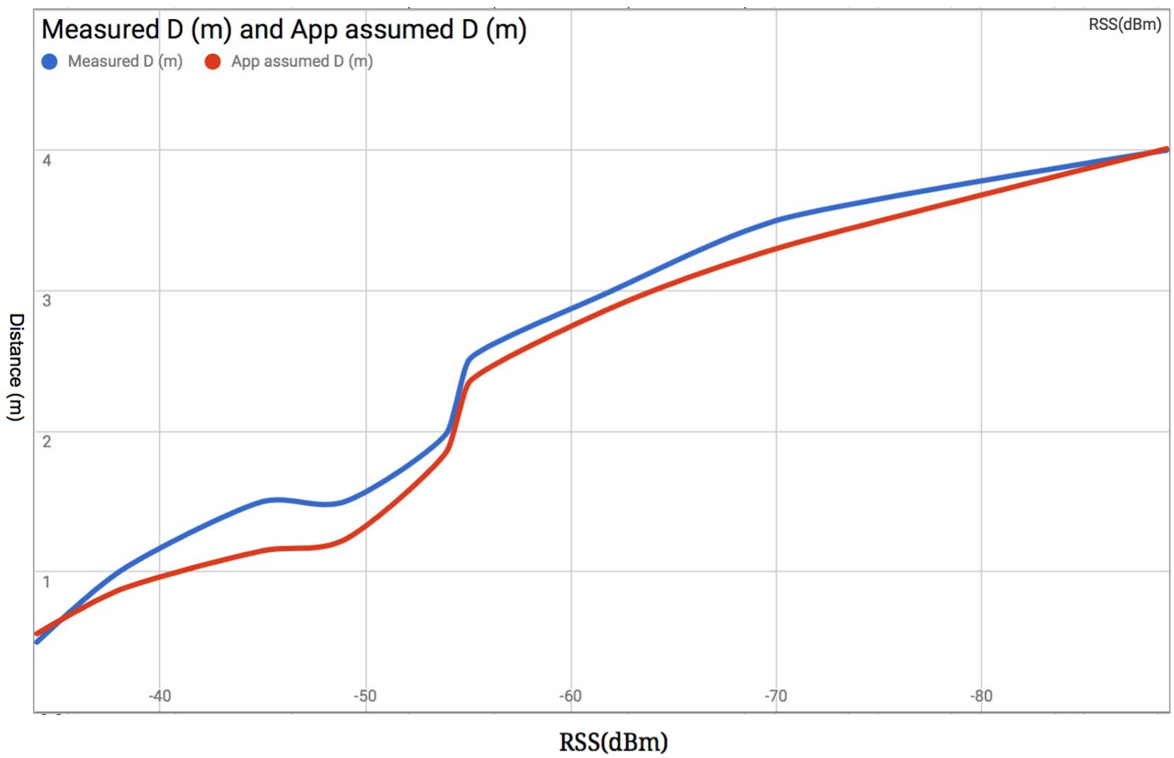

Figure 5: Initial iBeacon Reading Comparisons

After searching through Estimote's iBeacon library it was determined that the application does not use a standard path-loss model to determine distance from a beacon. Instead, they use equation (5-4)[18]

$$D[m] = 0.89976 * (\frac{Measured\ RSSI_{[dBm]}}{L_{0[dB]}})^{7.7095} + 0.111 \qquad \textbf{(5-4)}$$

Regardless of the model that Estimote uses, the next phase was to determine our own path-loss model for the iBeacons.

5.1.2 Measured Path-Loss Model

While reading calculated distances from the iBeacon™ allowed us to analyze the default iBeacon™ parameters, calculating our own path loss model required readings independent of the Estimote library. Data points were collected in a similar fashion to the previous section, with the iBeacon™ signal being measured by the BLE Scanner app at various distances. Each distance was measured with a tape measure, however, to ensure physical accuracy.

Later on we discovered that the iBeacons we had been using were changing based on their remaining battery life. Six new iBeacons were purchased to continue testing, with the intention that they would all be at full power and give more consistent results. After measuring and calculating their alphas, first-meter losses, and sigmas, we determined that setting the broadcasting power to -12dBm was suitable for our needs.

The figure below was created by using a MatLab which attempts to fit an equation following the format of $y = var_1 - 10 * var_2 * log_{10}(x)$ (the code and a brief discussion are available in 4.3). The script's results are shown below, giving an α of 2.42 and an L_0 = -63.79dB.

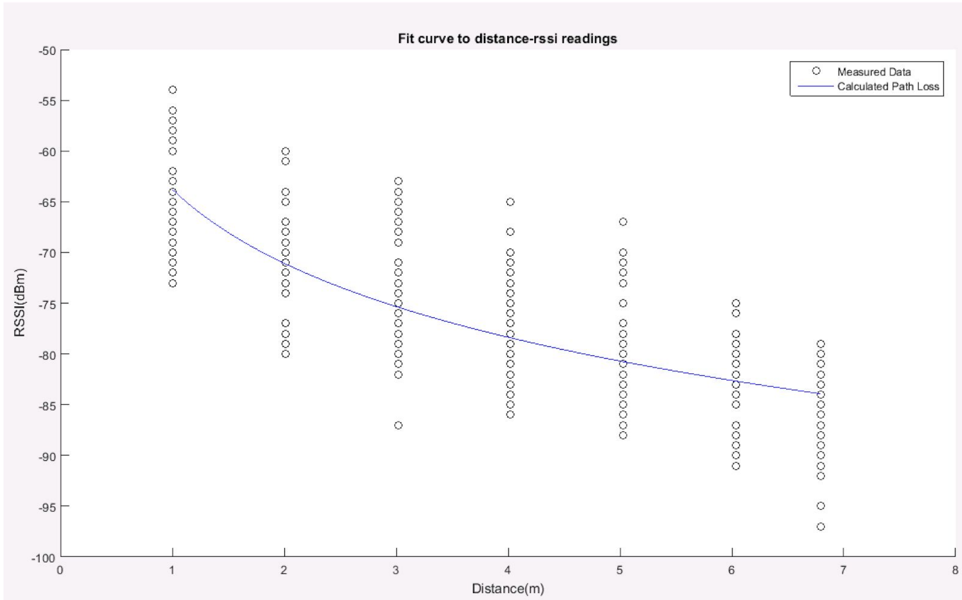

Figure 6: Best-Fit Curve from RSS vs Distance

Using the determined α and L_0 we obtain the full path-loss model of

$$L_P = -63.79 - 24.2 * log_{10}(d_{[m]})$$

Solved for d, we obtain the following formula to determine distance from RSSI:

$$d_{[m]} = 10^{\frac{L_P + 63.79}{-24.2}} \quad \textbf{(5-5)}$$

5.1.3 CRLB Implementation for Ideal Architecture

Throughout the project many beacon implementations were tested, with various number of beacons and in a variety of rooms. So in order to avoid guessing CRLB was used as a preliminary testing tool, allowing us to quickly test as many designs as we could create. Below are a few examples of implementation designs we felt modeled the designs we were looking for and showed the ranges of localization errors received in each formation.

The first CRLB heat map we simulated, seen below, is labeled as CRLB for a 6 Beacon setup: Hexagon Ceiling setup, was our first formation of beacons we set up. As seen in the color bar, we received a heavy spread of 1.1 to 1.8

all-throughout the room. In this setup, as seen by the pins we to receive conceivable lows of 1.077 in pockets of the room and roughly 1.3-1.5 around the beacon placement areas. As seen in many designs like this one, there is a consistent fading seen around the corners of the room. In this room, in the absolute corners we can see a localization minimum error of roughly 2.13.

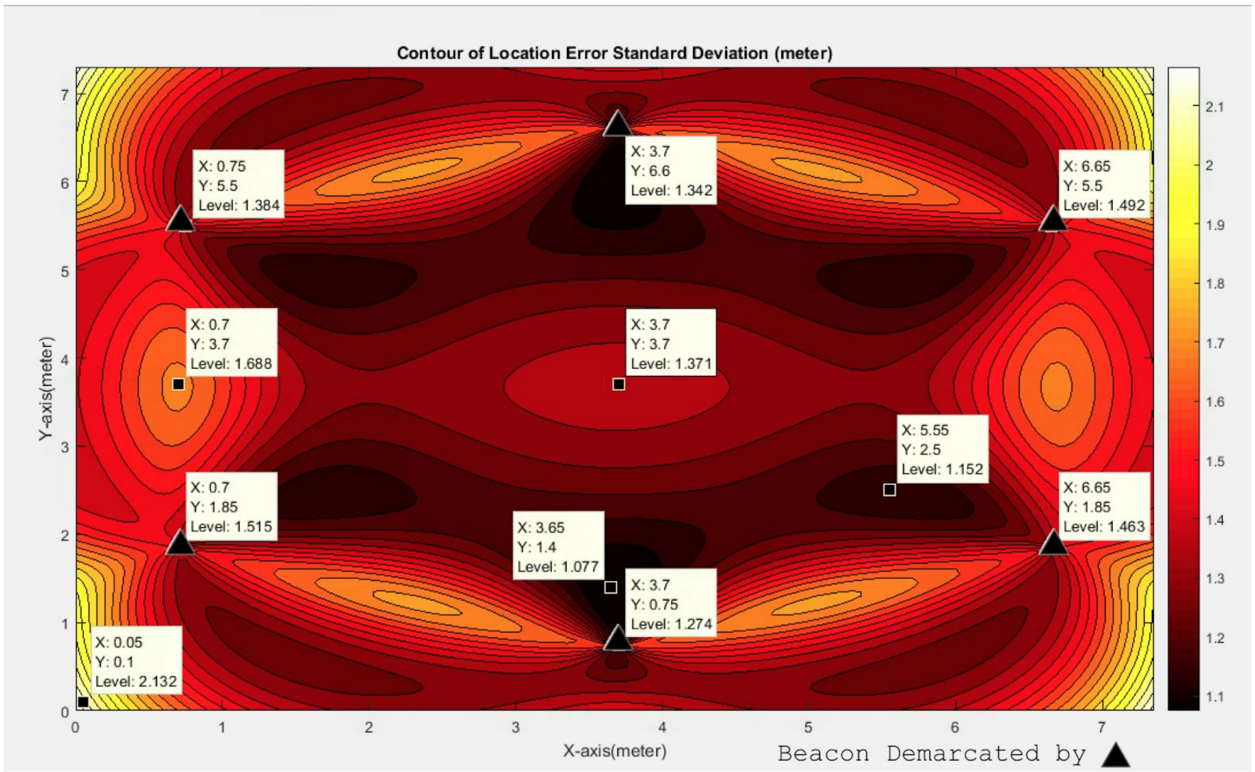

Figure 7: CRLB for a 6 Beacon setup: Hexagon Ceiling setup

The next CRLB heat map we simulated, seen below, is labeled as CRLB for a 6 Beacon setup: Hexagon Wall setup, was our next formation of beacons we set up. In this implementation we set up, similarly to the last one, in a hexagonal pattern around the room; but unlike the last one we moved these beacons out of the middle of the room, of the ceiling, and placed them on the center of the walls, between the floor and ceiling. As seen in the color bar below, we were able to received another heavy spread of 1.3 to 1.7 all-throughout the center of the room in a sort of 6-pointed star shape. In this setup, as seen by the pins we to receive conceivable lows of 1.334 in pockets of the room and roughly 1.3-1.5 around the beacon placement areas. As seen in many designs like this one, there is a consistent fading seen around the corners of the room. In this room, unlike the previous one, putting the beacons on the walls we were able to push the fading seen commonly in the corners to between the beacons out of the corners. In the areas between the beacons we saw a localization error approximation of 2.094. In the absolute corners we can see a localization minimum error of roughly 1.774, better then the previous implementation.

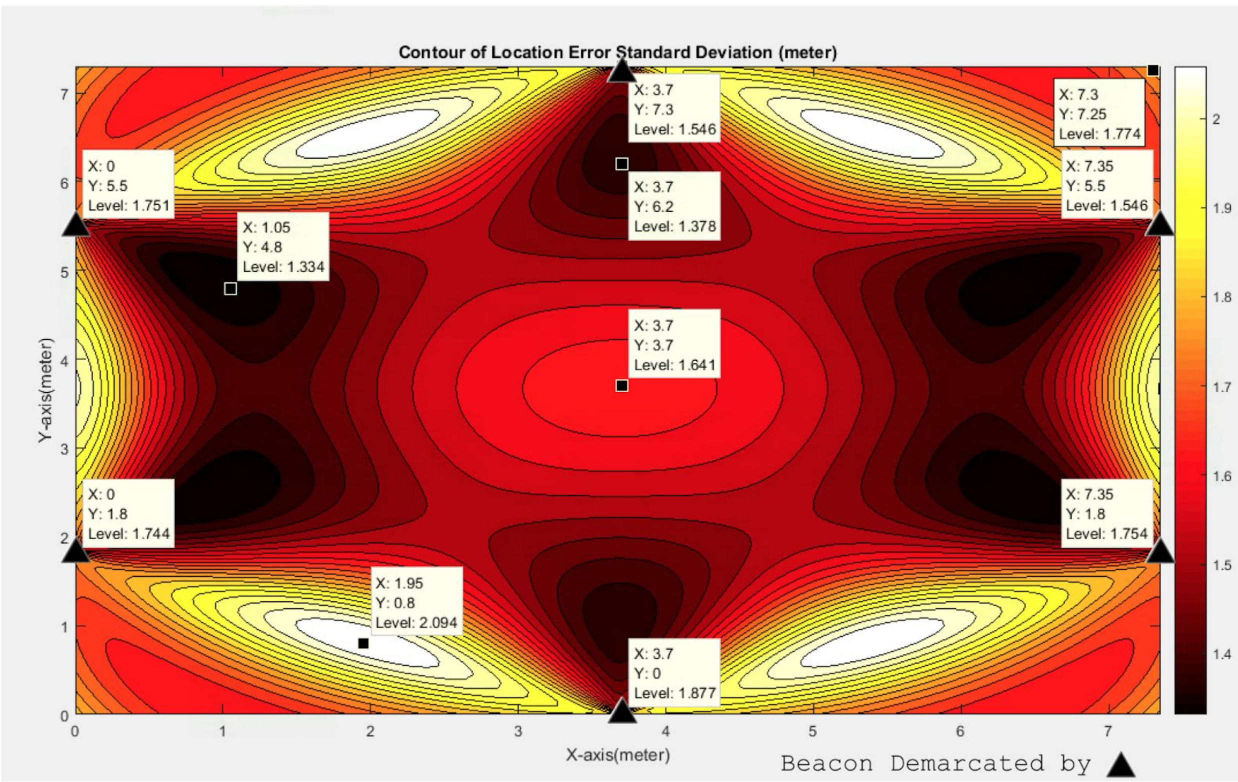

Figure 8: CRLB for a 6 Beacon setup: Hexagon Wall setup

The final CRLB heat map we simulated, seen below, is labeled as CRLB for a 6 Beacon setup: Two line setup, was our next formation of beacons we set up. In this implementation we set up, unlike the last ones, we were able to see a mirrored pattern along the the center vertical line in the room. As seen in the color bar below, we were able to received a heavy spread of 0.8 to 1.6 throughout the center of the room, stretching to the beacons. In this setup, as seen by the pins we to receive conceivable lows of 0.866 in large pockets of the room and roughly

1.0-1.3 around the beacon placed in the left of the room and 1.5-1.6 seen over the right beacons in the room areas. As seen in many designs like this one, there is a consistent fading seen around the corners of the room. In this room, unlike the previous one, putting the beacons in straight lines away from the corners, the error in the corners was drawn larger increasing their fading from the beacons to the corners. In the areas between the beacons we saw a localization error approximation of 1.532. In the absolute corners we can see a localization minimum error of roughly 3.37, worse then all previous implementation.

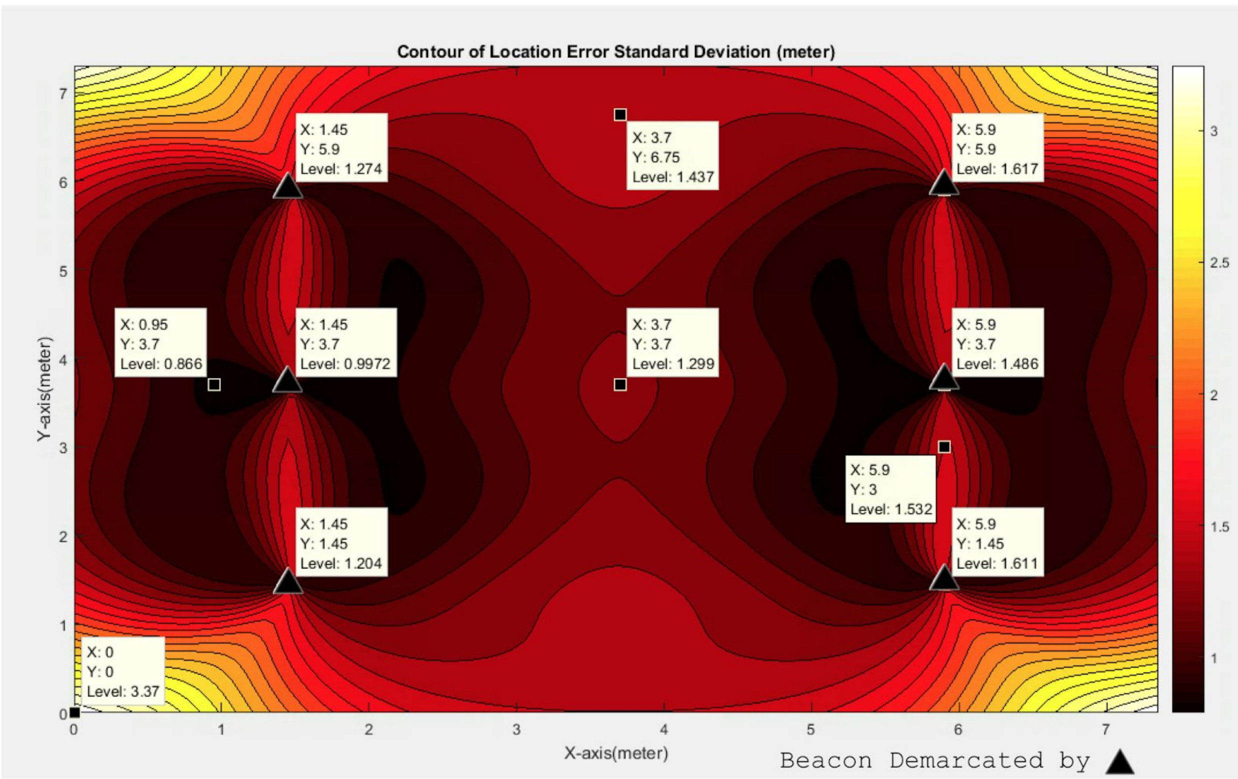

Figure 9: CRLB for a 6 Beacon setup: Two line setup

Table 5: CRLB Beacon Layout Analysis

	Localization Distance Error Range (meters)	Best Locations
Hexagonal Ceiling Setup	1.05 - 2.15	Near Beacons and in Corners
Hexagonal Wall Setup	1.3 - 2.05	Near Beacons and Center of room
Two Line Setup	0.8 - 3.25	Center of Room and Middle of each Wall

Using more iBeacons may lead to higher accuracy of results. While CRLB simulation does not show the physical error of an implementation, it provides a good baseline for setups. As seen in section 5.2.2 even though our measured error rate was higher than the CRLB, it followed a similar pattern based on location. Below are four images of possible setups using eight iBeacons instead of the six that we used. Note that this increases the cost of implementation, and most likely has a diminishing return due to iBeacon signals clashing.

Below is one of our assumed 8 beacon setups, called CRLB for a 8 Beacon setup: Corners and Middle Walls setup. In this setup we put all beacons in corners and in the center of each wall. This setup allowed for even distribution of localization error throughout the room, on the scale of 1.1 - 1.5, everywhere except the areas between the beacons on the wall. These areas showed localization error reaching only around 1.808.

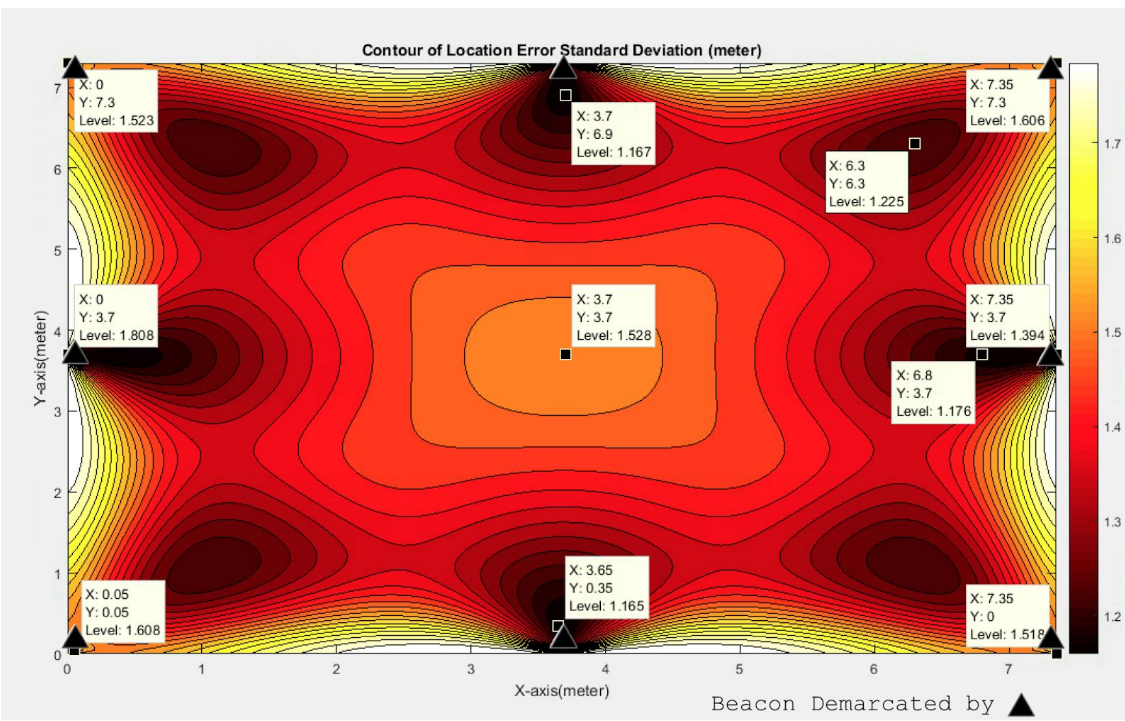

Figure 10: CRLB for a 8 Beacon setup: Corners and Middle Walls setup

Below is another one of our assumed 8 beacon setups, called CRLB for a 8 Beacon setup: Diamond setup. In this setup we put all beacons in the center of each wall and in the middle of those beacons on the wall. This setup allowed for even distribution of localization error throughout the room, on the scale of .7 - 1.3, everywhere except, like in setups similar to this, the corners of the room. These corners exhibited decent slow descent into localization error reaching only around 1.915.

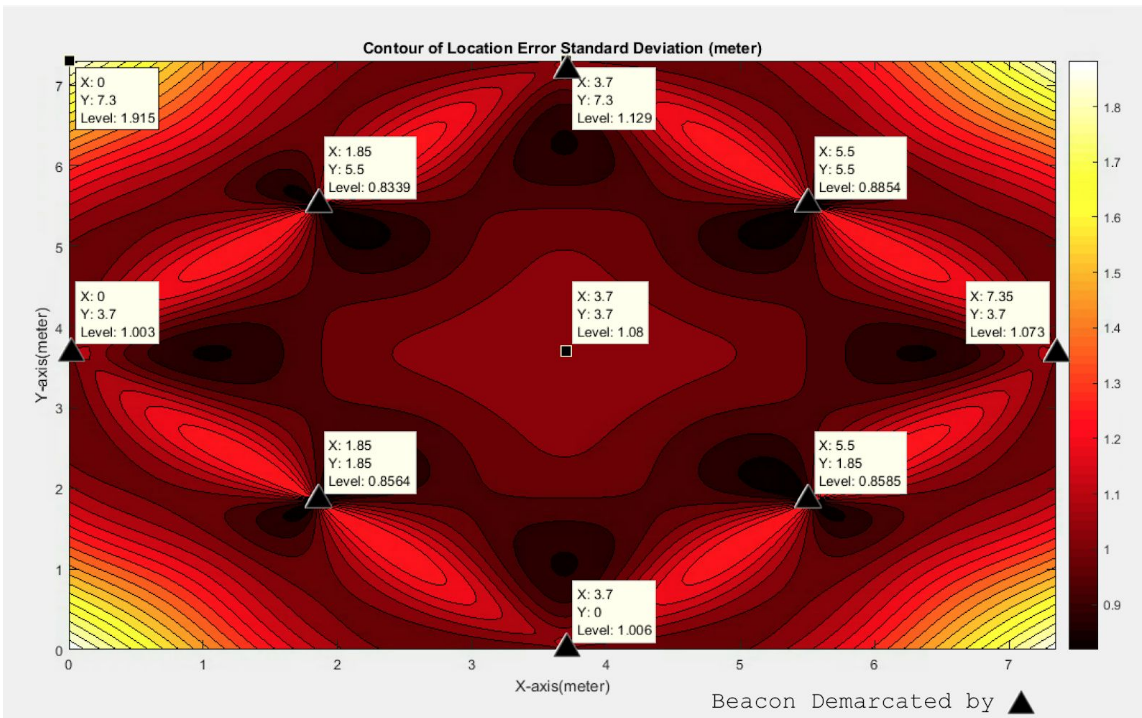

Figure 11: CRLB for a 8 Beacon setup: Diamond setup

Below is a third example of our assumed 8 beacon setups, called CRLB for a 8 Beacon setup: Outside Box - Inside Box setup. In this setup we put all beacons in the corners, and created a smaller box inside the room. This setup allowed for even distribution of localization error throughout the room, on the scale of 1.0 - 1.7, everywhere except on the diagonals between the outside box beacons and the inner box beacons. These diagonals exhibited localization error showing only around 2.65.

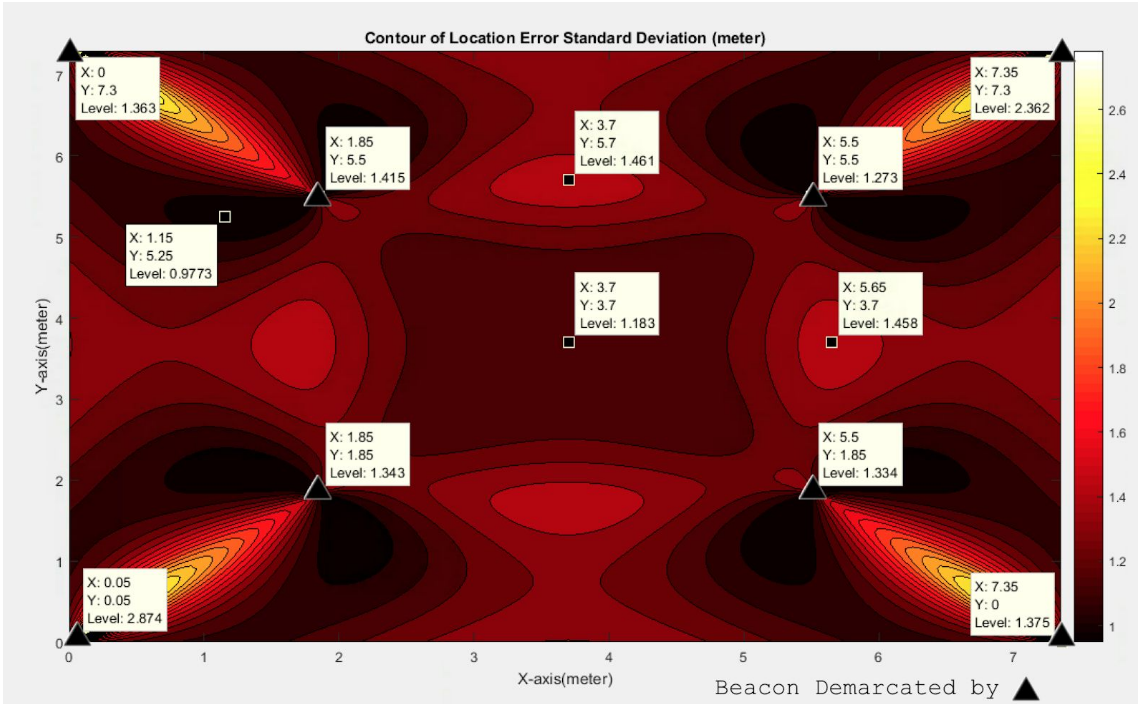

Figure 12: CRLB for a 8 Beacon setup: Outside Box - Inside Box setup

Below is our fourth and final example of an assumed 8 beacon setup, called CRLB for a 8 Beacon setup: Two line setup. In this setup we put all beacons in the two lines of four beacons lining the middle of each side of the room. This setup allowed for even distribution of localization error throughout a center band of the room, on the scale of 0.5 - 1.6, everywhere except, like in our previous two line setup, in the corners of the room. These corners exhibited rather fast descent into localization error reaching only around 2.786.

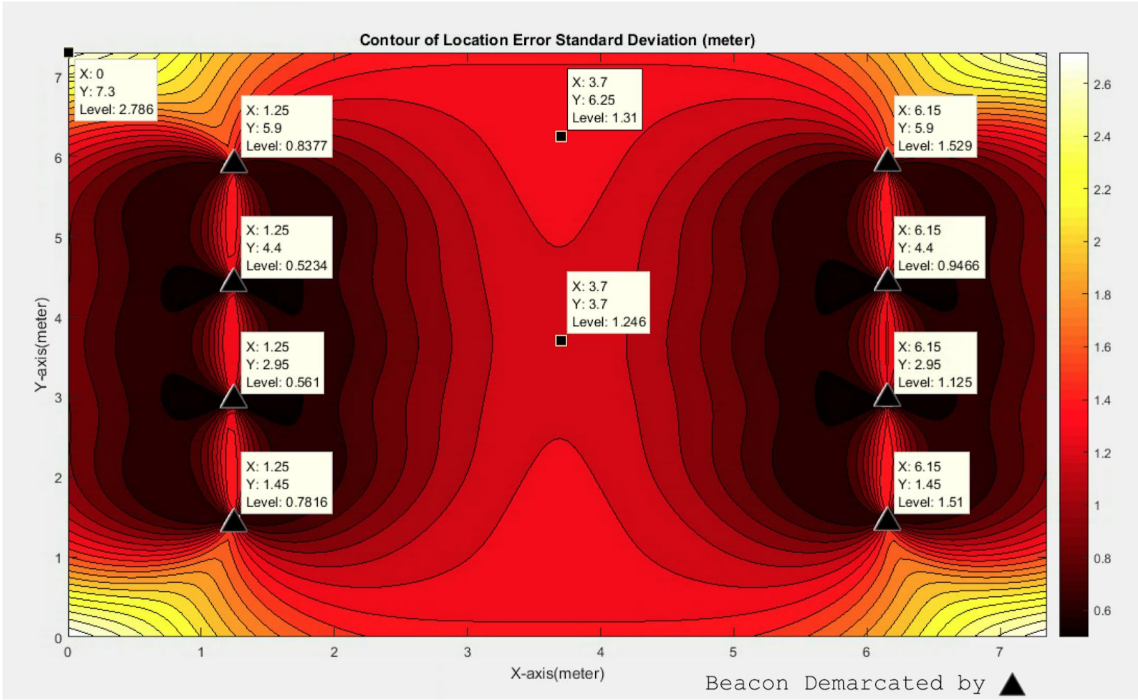

Figure 13: CRLB for a 8 Beacon setup: Two line setup

5.2 Algorithm Results and Development

5.2.1 Least Mean Square

The Least Mean Square algorithm only had successful convergence with our four beacon implementation. Below is the output of the LMS MatLab code (which can be found in Appendix C) provided to us, this result is a zoomed in view on the area(s) that the algorithm was trying to converge to. This is a 6 beacon set up, with the following beacon layout:

Table 6 Beacon Layout LMS test scenario

Beacon:	[X-Coordinate, Y-Coordinate]
Lemon(mID: 1)	[1.475 , 1.836]
Lemon(mID: 11)	[1.475, 5.509]
Candy(mID: 2)	[5.900, 1.836]
Candy(mID: 22)	[5.900, 5.509]
BeetRoot(mID: 3)	[3.688, 1.469]
BeetRoot(mID:33)	[3.688, 5.876]

The Matlab result was(code can be found in Appendix C):

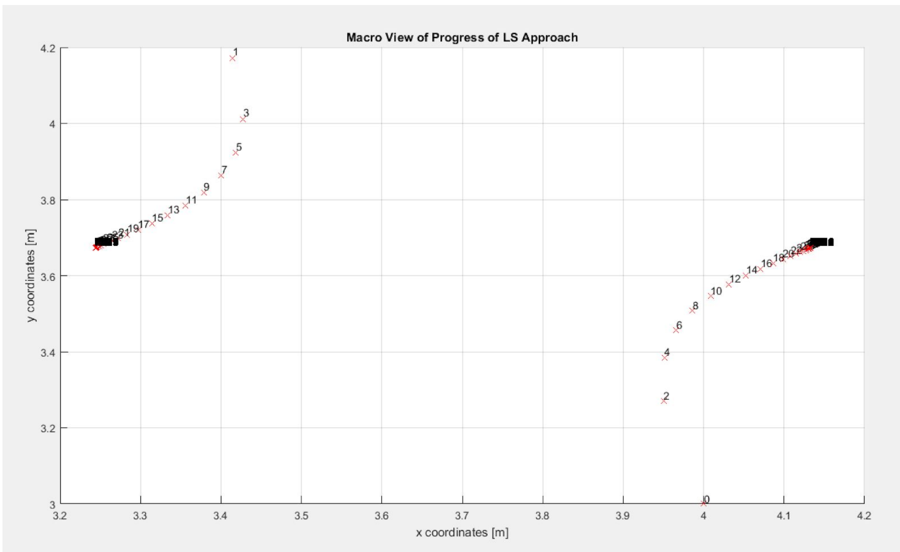

Figure 14: Non-Convergence Results of LMS

It is apparent that LMS did not converge here, demonstrated by it being unable to choose between the two points. What is interesting to note that one of the calculated convergence points is fairly close to the initial guess point. Unfortunately there is no way for the algorithm to know which of the two is correct and will run indefinitely. For this reason the group decided not to use the LMS algorithm: convergence is not guaranteed. The group also preferred implementations using a higher number of beacons, which negatively impact the probability of convergence of the LMS algorithm.

5.2.2 Maximum Likelihood

Once the hardware parameters and location constants such as α and σ were determined, and the ideal beacon setups found through CRLB, iBeacons were placed and data points taken. These data were put through the developed MLE to obtain a guess point, and then compared to the actual measured location of the receiver.

Below is a visual representation of MLE scoring using a six iBeacons setup. The iBeacons were placed in a hexagonal pattern on the ceiling of the room, 1.5 meters above the receiver (this height difference is accounted for in determining distances). The red point represents the calculated location, and the blue point is where the receiver actually was in the room. There is an error of roughly 1.5 meters between the two points.

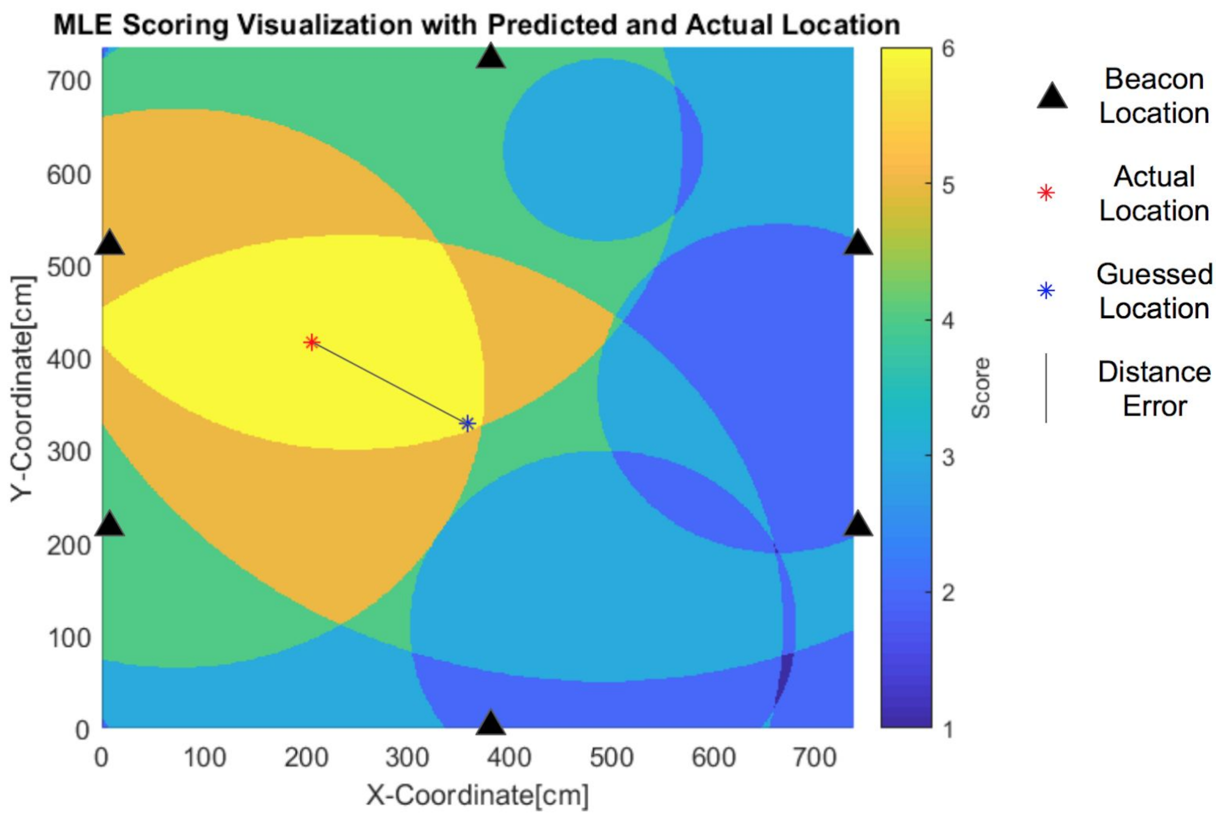

Figure 15: MLE Score Image of Low Error

An error of 1.5 represents one of the better locations and implementations we tested at. With the same iBeacon locations much higher error rates could be seen when standing closer to the corners of the room. Below is another scoring image, demonstrating this.

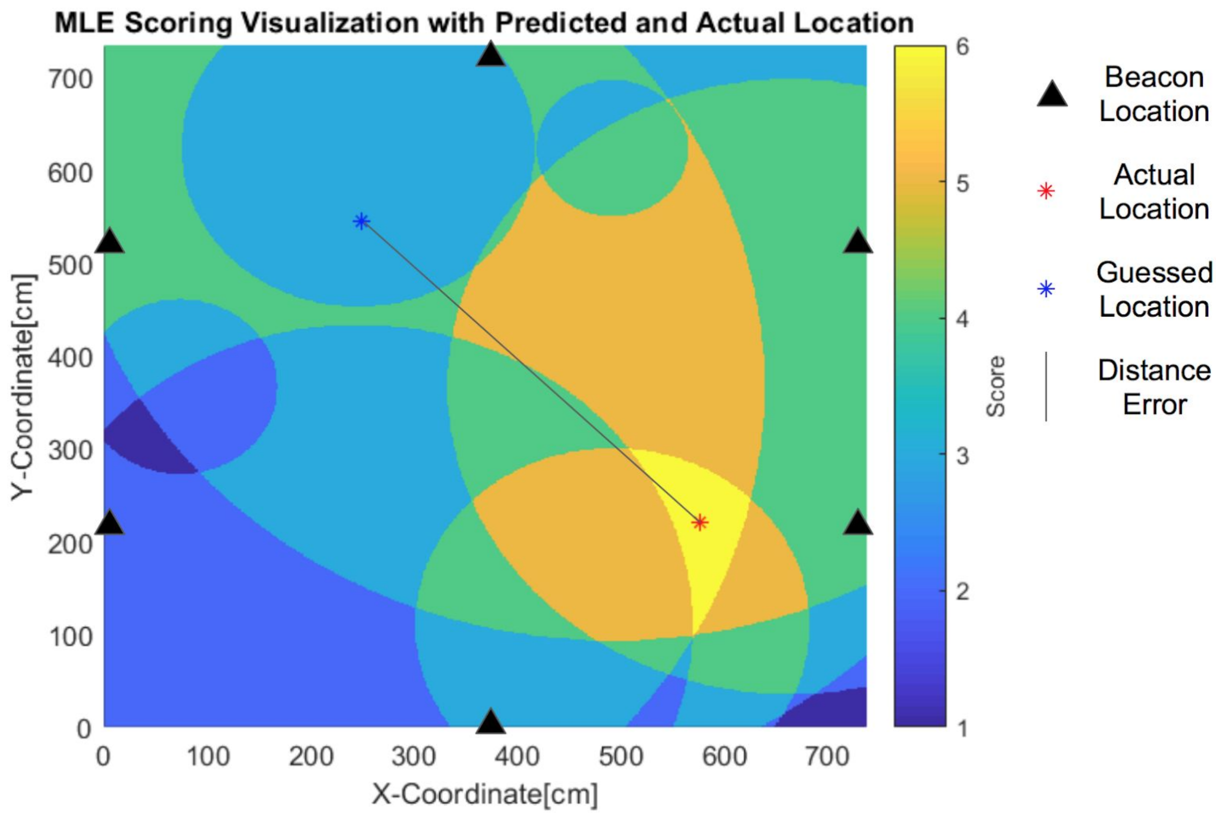

Figure 16: MLE Score Image of High Error

The error between these two points is roughly 4.8 meters. The algorithm correctly locates the center of the high-scoring centroid, but the scoring itself leads the algorithm to believe that the receiver is in a completely different location.

An error of 4.8 meters within a 7.38 by 7.35 meter room indicates that for most points in the room, it could appear as if the receiver was anywhere else in the room. Generally it was found that the error measured matched curve of the

CRLB error found for the implementations. The error itself however was greatly magnified.

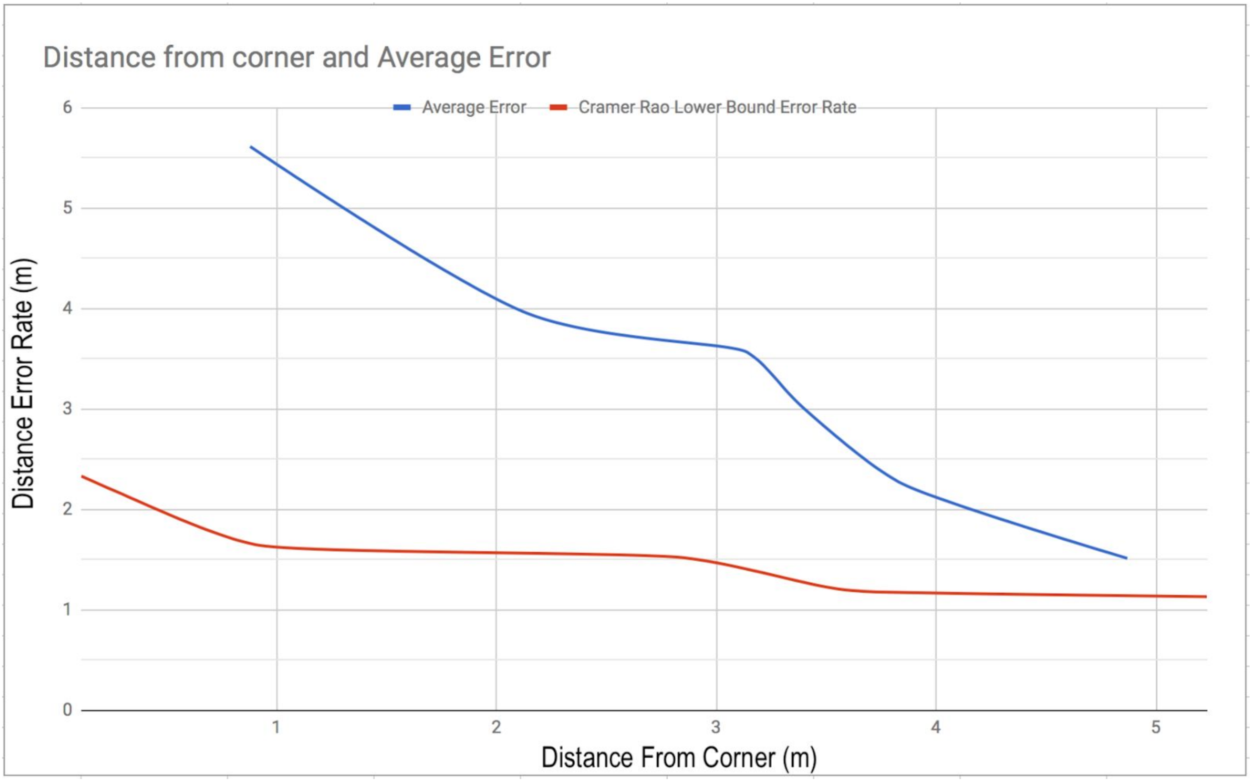

Figure 17: Measured Error vs CRLB over Distance

The graph above displays the error experienced by the system as the user moves towards the center of the room. As scene from above our actual error was very high towards the corners, we expeciend an average error in the corners of 5.5 meters. As readings were taken approaching the center of the room(denoted as 5m from coner) reading approached the CRLB steadily at our lowest error for this setup was 1.5 meters.

The data taken was processed with the path-loss constants found in section 5.1 of $\alpha = 2.42, \quad \sigma = 5.34, \quad L_0 =- 63.79dB$. The coverage γ , discussed in section 3.2.3, was set to 95%. Attempt at a higher accuracy involved changing the γ value to be more restrictive on what points scored and calculating a different $L_0 =- 64.51dB$ & $\alpha = 2.45$ through a slightly different best-fit line, but these changes did not improve the accuracy of the results. Below is the MLE result with the changed variables run on the same set of data as Figure 18 (the first score image).

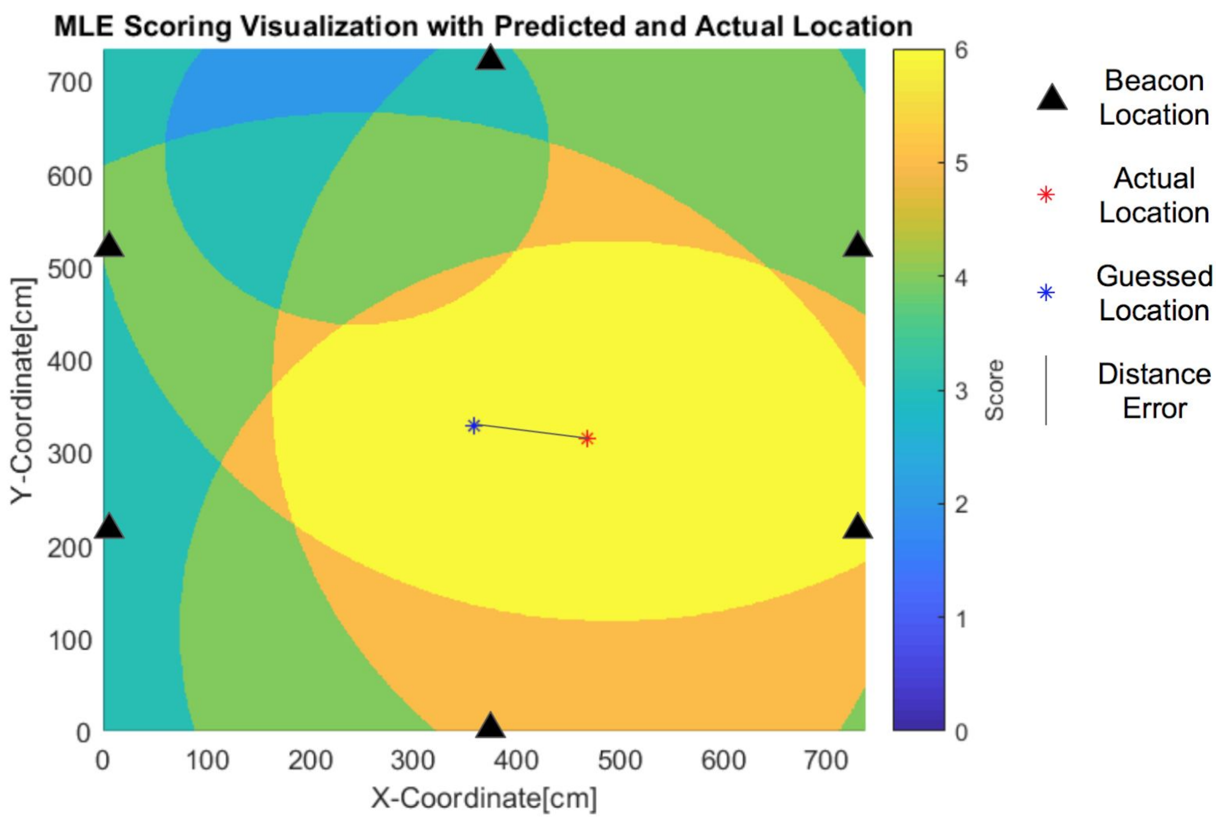

Figure 18: MLE Score Image with Adjusted Parameters

While the predicted location is technically closer to the real location, the size of the high-scoring region indicates lower accuracy of the parameters. Using the original set allows us to create smaller regions of high-scoring points, which in turn increases the accuracy of the predicted point.

5.3 Finalized Software Deliverables

In the process of testing implementation and optimizing the MLE a number of software deliverables were created. These include things such as MatLab scripts for finding best-fit lines, Java implementations of the algorithm, and an Android application used for testing the localization algorithm in the room of interest. The code for most of these can be found in Appendix E.

5.3.1 Android Application

The finalized Android application is able to predict a user's location within a room defined by the dimensions, beacon locations, and room path-loss constants. It displays the iBeacons currently being seen by the phone, and can be filtered to only see beacons of a specific major ID. The constants can also be changed from within the application in order to quickly test a different implementation within the same room. The application could theoretically be used with non-iBeacon Bluetooth signals, as long as the same part of their packet is dedicated to obtaining the signal strength broadcasted. The finer points of the AltBeacon library, such as filters and meshing, cannot be changed from within the app, and must be changed in the APK code itself through an environment such as Android Studio

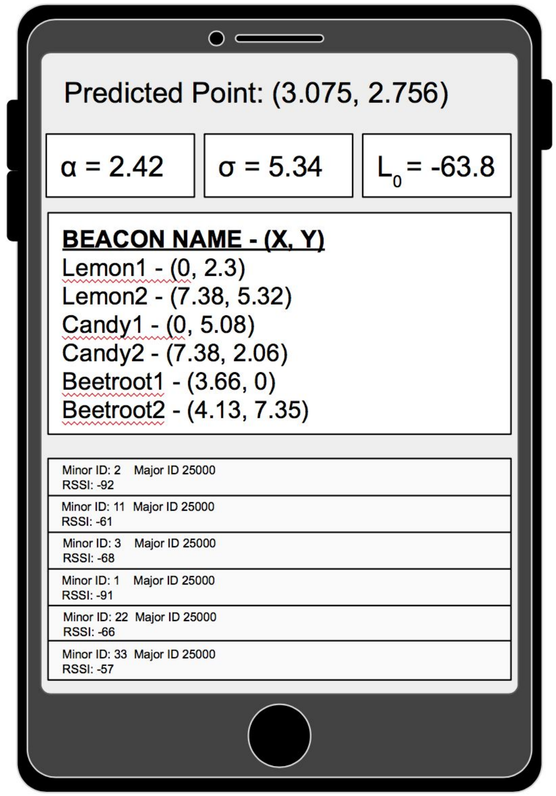

Figure 19: Android Application Diagram

The following few section will provide a brief overview of the code written while completing this project. MatLab scripts, generalized algorithms, and other code used for data gathering or analysis can be found here, in Appendix E, or online, on the project team's Git.

5.3.4 Maximum Likelihood Algorithm Code

The maximum likelihood algorithm was written in both Java and MatLab for varying purposes in this project. It was initially written as a standalone implementation which took user inputs and predicted the location in the room.

The further developed MatLab script was created to take a full set of readings from a beacon setup and return an array full of predicted locations. All of the beacon locations, path-loss model constant, and room specifications can be modified to fit whatever architecture is desired. Note that they do calculate distances and measurement granularity differently, so care should be exercised when running them. Both sets of code can be found in Appendix E, and can be mostly copied and pasted into their respective environments and run.

Note that this script runs when given an *nxm* matrix consisting of *n* beacons and *m* readings for each beacon. This script was written for MatLab R2017b, and may function differently on other versions of MatLab.

A similar script was written in Java. It was written in Java due to the ease of moving the algorithm to our mobile platform afterword for use in real-world measurements. All of the variables and constants, including the signal readings for each beacon. The MatLab version is recommended, however, as the Java version can only take one set of RSSI readings at a time. This should be used for either proof-of-concept, or for adaptation to other mediums.

Chapter 6: Conclusion and Future Directions

6.1 Overall Conclusions

After our testing and data analysis, we conclude that using four to six Low-Energy Bluetooth iBeacons with the Maximum Likelihood algorithm is not sufficient to accurately localize a receiver within a room. Errors of up to 4.8 meters within a 7.4 meter square room signify that the implementations we tried would not be reliable for indoor localization.

After working with the algorithms Least Mean Square and Maximum Likelihood, our view on what was an acceptable tolerance adapted. Initially we were hoping to have most of our error under a meter: we quickly learned that this was difficult to do consistently with the hardware we were tasked to use. This encouraged adaptations to made to the algorithms, such as data smoothing or changing target accuracy. The largest source of error is most likely the characteristics of the environment which was used for testing and the hardware. Throughout the project path-loss constants had to be recalculated due to changes in beacon battery. Once the constants were recalculated the results appeared to be consistent with what was initially found, but changed hardware is always a possible source of error.

Efficiently utilizing Cramer Rao Lower Bound we were able to run many simulations based on differing iBeacon implementations. Looking at these results above in 5.1.3 CRLB Implementation for Ideal Architecture and below in 6.2 Future Directions we were able to use the simulations of heat maps to find several beacon placement formations. This benefited us as a useful tool in order to find how the placement of the beacons will have affected each others signal propagation.

At the end of the project we did successfully test and implement localization algorithms, and used them on handheld devices. Although our user application wasn't as developed as we hoped. What was produced is a great testing tool and also a wonderful foundation for app developers to use if they desire an element of localization to their product.

6.2 Future Directions

Due to the numerous variables of environment, architecture, hardware variances, and algorithm diversity, there are a number of steps that could be taken to improve upon our methods and the results obtained.

First we will suggest a look into types of beacons. The iBeacons were a great tool to use in this project, but with the varied types of beacons in the market with a range of features, styles, and signal strengths, it would behoove of any group moving forward with this research to look into all the possible options presented to them. Further it would be necessary to see the user interface associated with the beacons themselves. When beginning this project there were 3 apps being used by our group to control and view the received signal information for the beacons. Eventually two of the apps combined into one and improved our user experience with a combination of features that complement each other nicely. Even with this development there was much to be desired from the Edistone and estimote applications used to manage the beacons. The most important part of an app we look for is a responsive application which is easy to use and can respond to the beacons.

An idea that was proposed but never tested was utilizing the Maximum Likelihood Algorithm for cases where Least Mean Square has convergence issues. As seen in section 5.2.1 one of the theorized convergence point was very accurate, MLE would be able to confirm that the convergence point on the other side of the room is wrong and to count only the on which aligns closest to MLE's calculated point. As far as the infinite iteration issues, a cycle limit can be set. It was observed that typically the algorithm would converge for our setting in fewer than 50 iterations. One limitation for our beacon implementations was we were limited in terms of “freedom” in regards to the Z-Axis. One of our ideas was to layer the algorithm, so in essence set at granularity on the Z-Axis and then have levels of score matrix planes. Then connect the plane to the plane above it and score the cube created between the two score planes.

The second area that could be improved upon is the path-loss model. Rather than having one model for the system of iBeacons, where the α, L_0, & σ are determined from a large set of readings, it is possible to develop a specific path-loss model for each iBeacon. This would lead to a higher spatial and temporal complexity for whatever medium runs the algorithm, as lookups would need to be performed for each signal read, but would most likely produce improved results. Since computational time of MLE was never an issue during testing, the increase would most likely not be a factor.

Another area of consideration in improving these methods is to use an LMS algorithm along side the MLE algorithm. Both of these algorithms have strengths that may reduce the impact of the other’s weaknesses. LMS has a possibility of not converging to a point, where it reduces to two points and then never settles. MLE may be able to produce a general location which can then be compared to the two points LMS found, and decide on the point that matches both. It appears as though LMS may have a lower error distance, which would make up for MLE having problems with determining the correct location.

Bibliography

[1] How it works | Bluetooth Technology Website. (n.d.). Retrieved October 7, 2017, from <https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works>

[2] V. (2011, January 11). High Tech Traveler. Retrieved October 07, 2017, from <https://web.archive.org/web/20110111075514/http://www.hoovers.com/business-information/--pageid__13751--/global-hoov-index.xhtml>

[3] IEEE. (2017). Retrieved October 07, 2017, from <http://www.ieee.org/index.html?WT.mc_id=head_bm>

[4] Y. Liu and Z. Yang, Location, Localization, and Localizability. Location-awareness Technology for Wireless Networks, Springer, 2010.

[5] Pollicino, J. (2016, July 14). Bluetooth SIG unveils Smart Marks, explains v4.0 compatibility with unnecessary complexity. Retrieved October 07, 2017, from <https://www.engadget.com/2011/10/25/bluetooth-sig-unveils-smart-marks-explains-v4-0-compatibility-w/>

[6] "iOS: Understanding iBeacon". Apple Inc. February 2015.

[7] http://www.ibeacon.com/what-is-ibeacon-a-guide-to-beacons/ What is iBeacon, a Guide to Beacons

[8] https://estimote.com/products/

[9] Curran, Kevin; Furey, Eoghan; Lunney, Tom; Santos, Jose; Woods, Derek; McCaughey, Aiden (2011). "An Evaluation of Indoor Location Determination Technologies". *Journal of Location Based Services*. **5** (2): 61–78. doi:10.1080/17489725.2011.562927

[10] http://www.cse.wustl.edu/~jain/cse574-14/ftp/indoor.pdf Survey of Wireless Based Indoor Localization

[11] http://www.ibeacontrends.com/beacons-at-the-guggenheim-museum-in-new-

york/

[12] Guggenheim App Adds Feature to Highlight Artworks Near Users. Retrieved October 7, 2017, from <https://www.guggenheim.org/news/guggenheim-app-adds-feature-to-highlight-artworks-near-users >

[13] https://developer.android.com/training/data-storage/room/index.html

[14] https://www.researchgate.net/figure/Representation-of-ToA-localization-technique_fig9_258389120 A Review of Antennas for Indoor Positioning Systems

[15] Zhouchi Li, Yang Yang, Kaveh Pahlavan, "Using iBeacon for newborns localization in hospitals", Medical Information and Communication Technology (ISMICT) 2016 10th International Symposium on, pp. 1-5, 2016, ISSN 2326-8301

[16] Pahlavan, K. and Krishnamurthy, P. (2013). *Principles of wireless access and localization*. 2nd ed.

[17] Marketsandmarkets.com. (2018). *Indoor Location Market worth 40.99 Billion USD by 2022*. [online] Available at: https://www.marketsandmarkets.com/PressReleases/indoor-location.asp [Accessed 25 Feb. 2018].

[18] https://gist.github.com/JoostKiens/d834d8acd3a6c78324c9

[19] Z. Li, “Sensor Behavior Modeling and Algorithm Design for Intelligent Presence Detection in Nursery Rooms using iBeacon,” thesis, 2016.

[20] Y. Sun, O. Tunde-Lukan, and T. Weng, “iBeacon Applications and Hybrid Wi-Fi Localization,” thesis, 2017.

[21] S. Redetzke, A. Vanner, and R. Otieno, “Smart Room Attendance Monitoring and Location Tracking with iBeacon Technology,” thesis, 2017.

[22] Q. Alhumoud, C. Connor, and D. Goodrich, “Using iBeacon for Navigation and Proximity Awareness in Smart Buildings, ” thesis, 2017.

Appendix A: Initial iBeacon™ Readings

RSSI(-dbm)	Distance(m)	RSSI(-dbm)	Distance(m)	RSSI(-dbm)	Distance(m)
-77	2.04216	-86	3.81	-88	5.4528
-73	2.04216	-86	3.81	-90	5.4528
-75	2.04216	-89	3.81	-80	5.4528
-73	2.04216	-91	3.81	-83	5.4528
-73	2.04216	-87	3.81	-83	5.4528
-74	2.04216	-89	3.81	-85	5.4528
-76	2.04216	-87	3.81	-94	5.4528
-81	2.04216	-89	3.81	-93	5.4528
-80	2.77368	-89	3.81	-89	5.4528
-75	2.77368	-89	3.81	-93	5.4528
-73	2.77368	-86	3.81	-87	5.4528
-76	2.77368	-87	3.81	-86	5.4528
-74	2.77368	-80	4.29	-83	5.4864
-77	3.16992	-84	4.29	-81	5.4864
-72	3.16992	-84	4.29	-89	5.4864
-75	3.16992	-87	4.29	-90	5.4864
-75	3.16992	-85	4.29	-89	5.4864
-83	3.16992	-87	4.29	-90	5.4864
-81	3.16992	-82	4.29	-88	5.4864
-84	3.71856	-86	4.29	-92	5.4864
-84	3.71856	-84	4.29	-89	5.4864
-88	3.71856	-87	4.29	--86	5.4864
-86	3.71856	-84	4.29	-82	5.4864
-77	3.71856	-87	4.572	-90	5.4864
-82	3.71856	-85	4.572		
-83	3.71856	-79	4.572		
-85	3.71856	-82	4.572		
-79	3.74904	-80	4.572		

-82	3.74904	-84	4.572		
-75	3.74904	-89	4.572		
-81	3.74904	-85	5.1816		
-76	3.74904	-87	5.1816		
-80	3.74904	-90	5.1816		
-80	3.74904	-90	5.1816		
-80	3.74904	-88	5.1816		
-82	3.74904	-89	5.1816		
-85	3.74904	-91	5.1816		
-84	3.74904	-95	5.1816		
-85	3.74904				
-82	3.74904				
-82	3.74904				
-74	3.74904				
-76	3.74904				
-86	3.74904				
-81	3.74904				
-81	3.74904				

Appendix B: Information Collection 2

RSSI(dBm)	App assumed D (meter)
-73	1
-83	1
-78	1
-83	1
-81	1
-75	1
-80	1

-78	1
-85	1
-79	1
-74	1
-79	1
-77	1
-70	1
-85	1
-87	1
-71	1
-74	1
-73	1
-81	1
-82	1
-80	1
-72	1
-82	1
-83	2
-86	2
-83	2
-79	2
-75	2
-85	2
-77	2
-76	2
-84	2
-86	2
-84	2
-82	2

Appendix C: Finalized MatLab and Java Scripts

Android Application Code Repository

https://github.com/Ploob/Pahlavan_Museum_MQP_2017

MatLab Log10 Best-Fit

```
function bestfit(minimum_x, minimum_y)
       ydata= -1 * []
       xdata= []

    fun = @(x, xdata)x(1)-10*x(2).*log10(xdata);
    x0 = [minimum_x, minimum_y];
    x = lsqcurvefit(fun, x0, xdata, ydata);
    fprintf("RSSI at 1 meter is %f\n", x(1));
    fprintf("Alpha calculated to be %f\n", x(2));
    hold on
    distances = linspace(xdata(1), xdata(end));
    plot(xdata, ydata, 'ko', 'DisplayName', 'Measured Data');
    plot(distances, fun(x, distances), 'b-', 'DisplayName', 'Best-Fit Line');
    plot(xdata, x(1) - 10 * x(2) * log10(xdata), 'r-', 'DisplayName', 'Calculated Path Loss');
    legend('show');
    title('Fit curve to distance-rssi readings');
    hold off;
```

MatLab Maximum Likelihood with Input Data Table Support

```
function locations = maxlikelihood(dataGrid)
       % USER VARIABLES ____________________________________________________________
       x_dim_m = 7.38;
       y_dim_m = 7.35;
       ceilingToAntenna = 1.5;
        alpha = 5.08;
       firstMeter = -48.31;
       sigma = 5.5321;
       targetAccuracy = 0.98;
       smallestMeasurement = 0.01; % cm accuracy for room

       actualPosition = [2.7 6.45];
       numBeacons = 6;
       beacon1 = [5.18, 0];
       beacon2 = [2.19, 0];
       beacon3 = [0, 3.66];
       beacon11 = [2.06, 7.35];
       beacon22 = [5.32, 7.35];
       beacon33 = [7.38, 4.13];
       beaconList = [beacon1, beacon2, beacon3, beacon11, beacon22, beacon33];
       %readingInput = [-77.90650628    -79.7028457    -77.87372943    -67.63557415
-78.94657752    -75.94412956];
```

```
% END OF USER VARIABLES ___________________________________________________

dbTolerance = sigma*sqrt(2)*erfcinv(2 - 2*targetAccuracy);
x_dim = x_dim_m / smallestMeasurement;
y_dim = y_dim_m / smallestMeasurement;

%readingList = readingInput * -1;
sz = size(dataGrid);
numRows = sz(1);
locations = zeros(numRows, 2);
% Loop per row of data in dataGrid
for dataGridRow = 1:numRows

readingList = -1 * dataGrid(dataGridRow,:);
predictedReadings = zeros(y_dim,x_dim,numBeacons);
%scores = zeros(x_dim, y_dim);
scores = zeros(y_dim, x_dim);

% Fill the table of predicted readings
for i = 1:x_dim
for j = 1:y_dim
        for k = 1:numBeacons
        predictedReadings(j,i,k) = firstMeter - 10 * alpha *
log10(sqrt((i*smallestMeasurement - beaconList(2*k-1))^2+(j*smallestMeasurement -
beaconList(2*k))^2+ceilingToAntenna^2));
        %predictedReadings(i,j,k) = firstMeter - 10 * alpha *
log10(sqrt((i*smallestMeasurement - beaconList(2*k-1))^2+(j*smallestMeasurement -
beaconList(2*k))^2+ceilingToAntenna^2));
        end
end
end

% Fill the table of scores
for i = 1:x_dim
        for j = 1:y_dim
        tot = 0;
        for k = 1:numBeacons
                if abs(predictedReadings(j,i,k) - readingList(k)) < dbTolerance

                tot = tot + 1;
                end
        end
        %scores(i,j) = tot;
        scores(j,i) = tot;
        %scores2(i,j) = tot * 255 / numBeacons;
        end
end
% Identify high scoring points and find centroid
x_tot = 0;
y_tot = 0;
highScore = 0;
totPts = 0;
for i = 1:x_dim
        for j = 1:y_dim
        %if scores(i,j) > highScore
        if scores(j,i) > highScore
                highScore = scores(j,i);
                x_tot = i;
                y_tot = j;
                totPts = 1;
        elseif scores(j,i) == highScore
                x_tot = x_tot + i;
                y_tot = y_tot + j;
                totPts = totPts + 1;
        end
        end
```

```
        end
        % Calculate the centroid in meters from origin
        predictedX = x_tot / totPts * smallestMeasurement;
        predictedY = y_tot / totPts * smallestMeasurement;

%       fprintf("Predicted location: %f, %f\n", predictedX, predictedY);
        locations(dataGridRow,1) = predictedX;
        locations(dataGridRow,2) = predictedY;
        End
```

MatLab Least Mean Square for Single Set Support

```
clc;clear all;close all;
%% This Matlab code solve Problem 15.2 in textbook
known_references = [10,10;0,15;-5,5];
initial_guess = [5,2];
distances = [15,10,5];

if size(known_references,2) ~= 2
    error('location of known reference points should be entered as Nx2 matrix');
end

figure(1);
hold on
grid on
i=1;
temp_location(i,:) = initial_guess ;
temp_error = 0 ;

for j = 1 : size(known_references,1)
    temp_error = temp_error + abs((known_references(j,1) - temp_location(i,1))^2 +
(known_references(j,2) - temp_location(i,2))^2 - distances(j)^2) ;
end

estimated_error = temp_error ;
plot(temp_location(i,1),temp_location(i,2),'rx') ; % plot
text(temp_location(i,1), temp_location(i,2)*(1 + 0.005) , num2str(0));
disp(['The initial location estimation is:
',num2str([temp_location(i,1),temp_location(i,2)])]);
% new_matrix = [ ];
while norm(estimated_error) > 1e-2 %iterative process for LS algorithm

    for j = 1 : size(known_references,1)  %Jacobian has been calculated in advance
        jacobian_matrix(j,:) = -2*(known_references(j,:) - temp_location(i,:)) ;  %partial
derivative is i.e. -2(x_1-x)
        f(j) = (known_references(j,1) - temp_location(i,1))^2 + (known_references(j,2) -
temp_location(i,2))^2 - distances(j)^2 ;
    end

    estimated_error = -inv(jacobian_matrix' * jacobian_matrix) * (jacobian_matrix') * f' ;
%update the U and E

    temp_location(i+1,:) = temp_location(i,:) + estimated_error' ;

    plot(temp_location(i+1,1),temp_location(i+1,2),'rx') ; % plot
```

```
    text(temp_location(i+1,1), temp_location(i+1,2)*(1 + 0.005) , num2str(i));

    i = i + 1;
lx=num2str(temp_location(i,1));ly=num2str(temp_location(i,2));err=sqrt(estimated_error(1)^2+es
timated_error(2)^2);
disp(['The ',num2str(i-1), 'th estimated location is:  ','[',lx,',',ly,']',' with an error of
', num2str(err)]);
end
axis([1.1*min(temp_location(:,1)), 1.1*max(temp_location(:,1)), 0.9*min(temp_location(:,2)),
1.1*max(temp_location(:,2))]);
title('Progress of LS Approach')
xlabel('x [m]');
ylabel('y [m]');
```

Java Maximum Likelihood for Single Set

```
public class AbstractedAlgorithm {
/*
 * Constants to set based on implementation, path loss, room variables, etc.
 */
    static double x_dim_m = 7.38; // Keep in meters, cm as smallest unit
    static double y_dim_m = 7.35; // Keep in meters, cm as smallest unit
    static double alpha = 2.45;
    static double firstMeter = -64.51;
    static double sigma = 4;
    static double targetAccuracy = 0.9;
    static int unitPerMeter = 100; // Don't touch unless you know what you're doing

    // Beacon locations, measured in meters and cm
    static Point beacon1 = new Point(2.46, 1.1); // 1
    static Point beacon2 = new Point(2.46, 6.24); // 2
    static Point beacon3 = new Point(0.74, 3.67); // 3
    static Point beacon11 = new Point(4.92, 1.1); // 11
    static Point beacon22 = new Point(4.92, 6.24); // 22
    static Point beacon33 = new Point(6.64, 3.67); // 33
    static Point[] beaconList = {beacon1, beacon2, beacon3, beacon11, beacon22, beacon33};
    static double[] readingList = {-71,    -77,    -67,    -81,    -74,    -77};

    static double dbTolerance;
    static int x_dim;
    static int y_dim;
    static double[][][] predictedReadings;
    static int[][] scores;
    static int numBeacons;
    public static void main(String[] args) {
        dbTolerance = dbRange(targetAccuracy);
        System.out.println("dB tolerance set to " + dbTolerance);
        x_dim = (int)(x_dim_m * 100);
        y_dim = (int)(y_dim_m * 100);
        numBeacons = beaconList.length;

        predictedReadings = new double[x_dim][y_dim][beaconList.length]; // Array of
predictions
        for(int j=0; j<y_dim; j++) { // Fill the prediction array
               for(int i=0; i<x_dim; i++) {
                      for(int k=0; k<numBeacons; k++) {
                             predictedReadings[i][j][k] = predictRssi(beaconList[k], new
Point(i,j));
```

```
                    }
                }
            }

            scores = new int[x_dim][y_dim]; // Array of scores based on read values
            int tot;
            for(int j=0; j<y_dim; j++) { // Fill the score array
                    for(int i=0; i<x_dim; i++) {
                            tot = 0;
                            for(int k=0; k<numBeacons; k++) {
                                    if(abs(predictedReadings[i][j][k] - readingList[k]) <=
dbTolerance) {
                                            tot++;

                                    }
                            }
                            scores[i][j] = tot;
                    }
            }

            // Collect list of good points
            ArrayList<Point> goodPoints = new ArrayList<Point>();
            int bestScore = 0;
            for(int j=0; j<y_dim; j++) {
                    for(int i=0; i<x_dim; i++) {
                            if(scores[i][j] > bestScore) {
                                    goodPoints.clear();
                                    goodPoints.add(new Point(i,j));
                                    bestScore = scores[i][j];
                            }else if(scores[i][j] == bestScore) {
                                    goodPoints.add(new Point(i,j));
                            }
                    }
            }

            for(int i=0; i<goodPoints.size(); i++) {
                    System.out.println("High score of " + bestScore + " found at " +
goodPoints.get(i).x + ", " + goodPoints.get(i).y);
            }

            // Find centroid
            int x_tot = 0;
            int y_tot = 0;
            for(int i=0; i<goodPoints.size(); i++) {
                    x_tot += goodPoints.get(i).x;
                    y_tot += goodPoints.get(i).y;
            }
            System.out.println("There are " + goodPoints.size() + " goodPoints");
            System.out.println("xtot = " + x_tot + "ytot = " + y_tot);
            System.out.println("Average: " + x_tot/goodPoints.size() + ", " +
y_tot/goodPoints.size());
            System.out.println("X: " + (float)x_tot/goodPoints.size()/unitPerMeter + ", Y: " +
(float)y_tot/goodPoints.size()/unitPerMeter);

            System.out.println("Run complete");
        }

        // db readings are allowed to be within this range, +/- in order to score
        // PAGE 53, HIS BOOK
        public static double dbRange(double accuracy) {
            return sigma * sqrt(2) * Erf.erfcInv(2*(1-accuracy));
        }

        public static double predictRssi(Point beacon, Point standingPoint) {
            //double mDistance = sqrt(pow(beacon.x - standingPoint.x/unitPerMeter, 2) +
pow(beacon.y - standingPoint.y/unitPerMeter, 2));
```

```
        double mDistance = sqrt(pow(beacon.x - standingPoint.x/unitPerMeter, 2) + pow(beacon.y
- standingPoint.y/unitPerMeter, 2) + pow(1.5,2));

        //System.out.println(mDistance);
        double rssi = firstMeter - 10 * alpha * log10(mDistance);
        return rssi;
    }

}
```

MatLab CRLB Code

```
close all;clear all;clc;warning off;
APx(1)=-5;APy(1)=-5;
APx(2)=-5;APy(2)=5;
APx(3)=0;APy(3)=0;
APx(4)=5;APy(4)=-5;
APx(5)=5,APy(5)=5;
SD=2.5; % Standard Deviation of Shadow Fading
NUM=5; % Number of Access Points
% Locations of Receivers
pace=0.1;
mx=-5:pace:5;
my=-5:pace:5;
nxy=length(mx);
for yi=1:nxy
    for xi=1:nxy
        for i1=1:NUM
            alpha=2.6;
            r(i1,xi,yi)=sqrt((mx(xi)-APx(i1))^2+(my(yi)-APy(i1))^2); % Distance Between
Transmitter and Receiver
            H1(i1,xi,yi)=-10*alpha/log(10)*(mx(xi)-APx(i1))/(r(i1,xi,yi))^2; % First Column of
H Matrix
            H2(i1,xi,yi)=-10*alpha/log(10)*(mx(yi)-APy(i1))/(r(i1,xi,yi))^2; % Second Column
of H Matrix
        end
        H(:,:,xi,yi)=[H1(:,xi,yi),H2(:,xi,yi)];
        Covv(:,:,xi,yi)=SD^2*((H(:,:,xi,yi))'*H(:,:,xi,yi))^(-1); % Covariance Matrix of Error
Estimate
        SDr(xi,yi)=sqrt(Covv(1,1,xi,yi)+Covv(2,2,xi,yi)); % Standard Deviat3ion of Location
Error
    end
end
SDr=SDr';
figure(1)
contourf(mx,my,SDr,20);
xlabel('X-axis(meter)');
ylabel('Y-axis(meter)');
title('Contour of Location Error Standard Deviation (meter)');
Colorbar;
```

Appendix D: 10/29/17 Museum Readings

Distance	Room 1 LOS				
3.81	-76				
6.12648	-84				
8.62584	-86		Room 1 LOS	Room 1	ALL Rooms
11.39952	-85	Mean shadow fading:	-46.5122	-47.9521	-47.7686
9.144	-88	σ shadow fading:	2.2543	2.3655	3.4156
Distance	Room 1 Non-LOS				
3.87096	-82				
6.4008	-85		Room 1 Non-LOS		
7.22376	-87	Mean shadow fading:	-49.7519		
12.4968	-92	σ shadow fading:	0.6091		
Distance	Room 2 LOS				
6.096	-80				
8.41248	-83		Room 2 LOS	Room 2	
9.47928	-85	Mean shadow fading:	-45.4799	-46.889	
13.1	-90	σ shadow fading:	1.5363	3.0633	
Distance	Room 2 Non-LOS				
13.1064	-93				
6.12648	-88		Room 2 Non-LOS		
3.87096	-76	Mean shadow fading:	-48.298		
6.27888	-82	σ shadow fading:	3.7737		
Distance	Room 3 LOS				
2.99	-75				

6.85	-87		Room 3 LOS	Room 3	
3.93	-72	Mean shadow fading:	-45.4701	-48.5379	
3.55	-77	σ shadow fading:	4.1705	4.9538	
Distance	Room 3 Non-LOS				
2.99	-80		Room 3 Non-LOS		
6.85	-90	Mean shadow fading:	-52.6283		
3.93	-86	σ shadow fading:	1.9002		

Appendix E: 2/4/18 -8DBM Readings

Distance From Beacon (m)					
Beacon 1 = 3.59664		Beacon 2 = 3.9624		Beacon 3 = 2.60604	
Power (dB)			Power(mw)		
Beacon 1	Beacon 2	Beacon 3	Beacon 1	Beacon 2	Beacon 3
75	80	72	3.16E-08	1.00E-08	6.31E-08
73	79	72	5.01E-08	1.26E-08	6.31E-08
84	98	69	3.98E-09	1.58E-10	1.26E-07
90	79	72	1.00E-09	1.26E-08	6.31E-08
89	87	70	1.26E-09	2.00E-09	1.00E-07
86	88	70	2.51E-09	1.58E-09	1.00E-07
81	87	69	7.94E-09	2.00E-09	1.26E-07
82	86	69	6.31E-09	2.51E-09	1.26E-07
82	85	70	6.31E-09	3.16E-09	1.00E-07
83	85	69	5.01E-09	3.16E-09	1.26E-07
83	85	72	5.01E-09	3.16E-09	6.31E-08
80	84	72	1.00E-08	3.98E-09	6.31E-08
84	86	73	3.98E-09	2.51E-09	5.01E-08

83	80	73	5.01E-09	1.00E-08	5.01E-08
79	81	72	1.26E-08	7.94E-09	6.31E-08
89	80	70	1.26E-09	1.00E-08	1.00E-07
92	80	71	6.31E-10	1.00E-08	7.94E-08
89	79	71	1.26E-09	1.26E-08	7.94E-08
88	85	74	1.58E-09	3.16E-09	3.98E-08
87	89	71	2.00E-09	1.26E-09	7.94E-08
83	87	71	5.01E-09	2.00E-09	7.94E-08
83	89	71	5.01E-09	1.26E-09	7.94E-08
83	94	71	5.01E-09	3.98E-10	7.94E-08
81	98	73	7.94E-09	1.58E-10	5.01E-08
81	93	74	7.94E-09	5.01E-10	3.98E-08
81	95	72	7.94E-09	3.16E-10	6.31E-08
80	82	73	1.00E-08	6.31E-09	5.01E-08
80	91	73	1.00E-08	7.94E-10	5.01E-08
81	79	74	7.94E-09	1.26E-08	3.98E-08
85	85	74	3.16E-09	3.16E-09	3.98E-08
84	86	74	3.98E-09	2.51E-09	3.98E-08
84	86	60	3.98E-09	2.51E-09	1.00E-06
84	84	71	3.98E-09	3.98E-09	7.94E-08
85	84	72	3.16E-09	3.98E-09	6.31E-08
82	94	71	6.31E-09	3.98E-10	7.94E-08
82	91	72	6.31E-09	7.94E-10	6.31E-08
81	90	72	7.94E-09	1.00E-09	6.31E-08
Average(non converted)			Log Average (After dBm to mW to dBm conversion)		
Beacon 1	Beacon 2	Beacon 3	Beacon 1	Beacon 2	Beacon 3
-83.21621622(dB)	-86.24324324(dB)	-71.32432432(dB)	-81.44907236(dB)	-83.72247258(dB)	-70.09536669(dB)

Appendix F: 2/10/18 -8DBM Readings

Distance From Origin (m)	
X = 2.5146	Y = 5.45592

Power (dB)						Power(mw)					
80	74	73	71	82	75	1.00E-08	3.98E-08	5.01E-08	7.94E-08	6.31E-09	3.16E-08
85	77	72	70	80	74	3.16E-09	2.00E-08	6.31E-08	1.00E-07	1.00E-08	3.98E-08
87	77	72	70	72	75	2.00E-09	2.00E-08	6.31E-08	1.00E-07	6.31E-08	3.16E-08
82	78	84	69	73	70	6.31E-09	1.58E-08	3.98E-09	1.26E-07	5.01E-08	1.00E-07
84	79	82	67	74	69	3.98E-09	1.26E-08	6.31E-09	2.00E-07	3.98E-08	1.26E-07
83	82	80	64	76	69	5.01E-09	6.31E-09	1.00E-08	3.98E-07	2.51E-08	1.26E-07
82	86	80	66	77	69	6.31E-09	2.51E-09	1.00E-08	2.51E-07	2.00E-08	1.26E-07
88	83	81	66	76	69	1.58E-09	5.01E-09	7.94E-09	2.51E-07	2.51E-08	1.26E-07
83	81	79	65	77	75	5.01E-09	7.94E-09	1.26E-08	3.16E-07	2.00E-08	3.16E-08
87	83	77	66	76	74	2.00E-09	5.01E-09	2.00E-08	2.51E-07	2.51E-08	3.98E-08
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-84.1	-80	-78	-67.4	-76.3	-71.9	-83.4331	-78.6985	-76.0715	-66.8345	-75.4577	-71.0898

Distance From Origin (m)											
X = 3.59664						Y = 3.29184					
Power (dB)						Power(mw)					
78	78	79	81	69	72	1.58E-08	1.58E-08	1.26E-08	7.94E-09	1.26E-07	6.31E-08
80	85	76	72	75	72	1.00E-08	3.16E-09	2.51E-08	6.31E-08	3.16E-08	6.31E-08
72	84	84	72	73	81	6.31E-08	3.98E-09	3.98E-09	6.31E-08	5.01E-08	7.94E-09
73	86	81	73	75	71	5.01E-08	2.51E-09	7.94E-09	5.01E-08	3.16E-08	7.94E-08
72	86	76	72	72	75	6.31E-08	2.51E-09	2.51E-08	6.31E-08	6.31E-08	3.16E-08
72	85	78	68	71	84	6.31E-08	3.16E-09	1.58E-08	1.58E-07	7.94E-08	3.98E-09
77	83	77	69	78	74	2.00E-08	5.01E-09	2.00E-08	1.26E-07	1.58E-08	3.98E-08
76	84	76	70	79	72	2.51E-08	3.98E-09	2.51E-08	1.00E-07	1.26E-08	6.31E-08
75	86	78	73	77	72	3.16E-08	2.51E-09	1.58E-08	5.01E-08	2.00E-08	6.31E-08
76	79	74	79	76	73	2.51E-08	1.26E-08	3.98E-08	1.26E-08	2.51E-08	5.01E-08
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-75.1	-83.6	-77.9	-72.9	-74.5	-74.6	-74.3525	-82.5749	-77.1821	-71.5837	-73.4171	-73.3227

Distance From Origin (m)											
X = 5.74548						Y = 5.95884					
Power (dB)						Power(mw)					
75	74	76	64	80	82	3.16E-08	3.98E-08	2.51E-08	3.98E-07	1.00E-08	6.31E-09
77	75	73	65	79	81	2.00E-08	3.16E-08	5.01E-08	3.16E-07	1.26E-08	7.94E-09
73	71	74	65	76	82	5.01E-08	7.94E-08	3.98E-08	3.16E-07	2.51E-08	6.31E-09
80	68	77	67	76	80	1.00E-08	1.58E-07	2.00E-08	2.00E-07	2.51E-08	1.00E-08
81	68	73	65	80	85	7.94E-09	1.58E-07	5.01E-08	3.16E-07	1.00E-08	3.16E-09
77	67	75	64	81	79	2.00E-08	2.00E-07	3.16E-08	3.98E-07	7.94E-09	1.26E-08
77	66	74	63	82	75	2.00E-08	2.51E-07	3.98E-08	5.01E-07	6.31E-09	3.16E-08
80	67	74	70	91	77	1.00E-08	2.00E-07	3.98E-08	1.00E-07	7.94E-10	2.00E-08
77	73	72	71	85	80	2.00E-08	5.01E-08	6.31E-08	7.94E-08	3.16E-09	1.00E-08
78	73	78	70	87	75	1.58E-08	5.01E-08	1.58E-08	1.00E-07	2.00E-09	3.16E-08
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-77.5	-70.2	-74.6	-66.4	-81.7	-79.6	-76.8752	-69.1424	-74.2561	-65.6463	-79.8703	-78.5539

Distance From Origin (m)											
X = 6.73608						Y = 4.02336					
Power (dB)						Power(mw)					
76	74	78	83	72	81	2.51E-08	3.98E-08	1.58E-08	5.01E-09	6.31E-08	7.94E-09
79	76	72	83	72	77	1.26E-08	2.51E-08	6.31E-08	5.01E-09	6.31E-08	2.00E-08
71	77	67	81	74	77	7.94E-08	2.00E-08	2.00E-07	7.94E-09	3.98E-08	2.00E-08
73	69	69	74	77	78	5.01E-08	1.26E-07	1.26E-07	3.98E-08	2.00E-08	1.58E-08
73	68	66	81	81	84	5.01E-08	1.58E-07	2.51E-07	7.94E-09	7.94E-09	3.98E-09
72	68	75	83	87	79	6.31E-08	1.58E-07	3.16E-08	5.01E-09	2.00E-09	1.26E-08
73	69	72	82	80	80	5.01E-08	1.26E-07	6.31E-08	6.31E-09	1.00E-08	1.00E-08
76	72	76	79	81	79	2.51E-08	6.31E-08	2.51E-08	1.26E-08	7.94E-09	1.26E-08
72	73	72	76	79	77	6.31E-08	5.01E-08	6.31E-08	2.51E-08	1.26E-08	2.00E-08
73	73	70	89	79	83	5.01E-08	5.01E-08	1.00E-07	1.26E-09	1.26E-08	5.01E-09
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6

-73.8	-71.9	-71.7	-81.1	-78.2	-79.5	-73.2890	-70.8779	-70.2757	-79.3551	-76.2157	-78.9340

Distance From Origin (m)											
X = 3.64236						Y = 1.20396					
Power (dB)						Power(mw)					
65	76	76	72	78	68	3.16E-07	2.51E-08	2.51E-08	6.31E-08	1.58E-08	1.58E-07
65	74	75	75	73	75	3.16E-07	3.98E-08	3.16E-08	3.16E-08	5.01E-08	3.16E-08
66	79	68	74	74	80	2.51E-07	1.26E-08	1.58E-07	3.98E-08	3.98E-08	1.00E-08
69	75	67	75	65	76	1.26E-07	3.16E-08	2.00E-07	3.16E-08	3.16E-07	2.51E-08
65	81	66	74	65	80	3.16E-07	7.94E-09	2.51E-07	3.98E-08	3.16E-07	1.00E-08
73	98	68	76	65	82	5.01E-08	1.58E-10	1.58E-07	2.51E-08	3.16E-07	6.31E-09
74	88	67	80	66	83	3.98E-08	1.58E-09	2.00E-07	1.00E-08	2.51E-07	5.01E-09
65	85	67	83	67	83	3.16E-07	3.16E-09	2.00E-07	5.01E-09	2.00E-07	5.01E-09
73	84	67	81	67	84	5.01E-08	3.98E-09	2.00E-07	7.94E-09	2.00E-07	3.98E-09
71	85	68	78	80	88	7.94E-08	3.16E-09	1.58E-07	1.58E-08	1.00E-08	1.58E-09
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-68.6	-82.5	-68.9	-76.8	-70	-79.9	-67.3014	-78.8896	-68.0093	-75.6882	-67.6581	-75.8985

Distance From Origin (m)											
X = 0.6096						Y = 6.7056					
Power (dB)						Power(mw)					
77	73	77	80	83	76	2.00E-08	5.01E-08	2.00E-08	1.00E-08	5.01E-09	2.51E-08
83	75	75	70	82	77	5.01E-09	3.16E-08	3.16E-08	1.00E-07	6.31E-09	2.00E-08
74	73	75	69	74	72	3.98E-08	5.01E-08	3.16E-08	1.26E-07	3.98E-08	6.31E-08
75	72	92	67	74	73	3.16E-08	6.31E-08	6.31E-10	2.00E-07	3.98E-08	5.01E-08
85	78	76	67	75	71	3.16E-09	1.58E-08	2.51E-08	2.00E-07	3.16E-08	7.94E-08
86	79	93	77	77	69	2.51E-09	1.26E-08	5.01E-10	2.00E-08	2.00E-08	1.26E-07
87	74	88	77	72	73	2.00E-09	3.98E-08	1.58E-09	2.00E-08	6.31E-08	5.01E-08
87	72	91	78	73	74	2.00E-09	6.31E-08	7.94E-10	1.58E-08	5.01E-08	3.98E-08
81	73	86	78	76	75	7.94E-09	5.01E-08	2.51E-09	1.58E-08	2.51E-08	3.16E-08
81	74	85	79	74	74	7.94E-09	3.98E-08	3.16E-09	1.26E-08	3.98E-08	3.98E-08
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					

Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-81.6	-74.3	-83.8	-74.2	-76	-73.4	-79.1382	-73.8067	-79.2995	-71.4319	-74.9395	-72.7986

Appendix G: 2/17/18 Max Likelihood Readings

Distance From Origin (m)											
X = 4.32816						Y = 3.62712					
Power (dB)						Power(mw)					
85	86	77	71	68	79	3.16E-09	2.51E-09	2.00E-08	7.94E-08	1.58E-07	1.26E-08
84	79	78	74	67	81	3.98E-09	1.26E-08	1.58E-08	3.98E-08	2.00E-07	7.94E-09
84	79	77	74	72	81	3.98E-09	1.26E-08	2.00E-08	3.98E-08	6.31E-08	7.94E-09
80	78	81	75	71	82	1.00E-08	1.58E-08	7.94E-09	3.16E-08	7.94E-08	6.31E-09
78	79	80	73	72	75	1.58E-08	1.26E-08	1.00E-08	5.01E-08	6.31E-08	3.16E-08
78	78	80	74	70	76	1.58E-08	1.58E-08	1.00E-08	3.98E-08	1.00E-07	2.51E-08
77	80	80	76	71	77	2.00E-08	1.00E-08	1.00E-08	2.51E-08	7.94E-08	2.00E-08
76	80	81	77	68	78	2.51E-08	1.00E-08	7.94E-09	2.00E-08	1.58E-07	1.58E-08
73	81	80	76	72	76	5.01E-08	7.94E-09	1.00E-08	2.51E-08	6.31E-08	2.51E-08
73	84	81	71	67	77	5.01E-08	3.98E-09	7.94E-09	7.94E-08	2.00E-07	2.00E-08
74	82	84	70	66	75	3.98E-08	6.31E-09	3.98E-09	1.00E-07	2.51E-07	3.16E-08
74	79	82	70	67	74	3.98E-08	1.26E-08	6.31E-09	1.00E-07	2.00E-07	3.98E-08
84	77	81	72	68	77	3.98E-09	2.00E-08	7.94E-09	6.31E-08	1.58E-07	2.00E-08
84	79	80	75	71	79	3.98E-09	1.26E-08	1.00E-08	3.16E-08	7.94E-08	1.26E-08
84	80	80	74	70	81	3.98E-09	1.00E-08	1.00E-08	3.98E-08	1.00E-07	7.94E-09
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-78.8	-80.4	-79.5	-74.1	-69.8	-78.2	-77.0305	-79.8338	-79.2233	-73.6630	-69.3398	-77.6346

Distance From Origin (m)											
X = 4.4						Y = 0.75					
Power (dB)						Power(mw)					
71	79	81	82	76	76	7.94E-08	1.26E-08	7.94E-09	6.31E-09	2.51E-08	2.51E-08

75	78	82	76	80	75	3.16E-08	1.58E-08	6.31E-09	2.51E-08	1.00E-08	3.16E-08
72	78	81	77	81	76	6.31E-08	1.58E-08	7.94E-09	2.00E-08	7.94E-09	2.51E-08
76	77	82	78	77	67	2.51E-08	2.00E-08	6.31E-09	1.58E-08	2.00E-08	2.00E-07
77	77	81	77	76	67	2.00E-08	2.00E-08	7.94E-09	2.00E-08	2.51E-08	2.00E-07
75	78	82	78	81	68	3.16E-08	1.58E-08	6.31E-09	1.58E-08	7.94E-09	1.58E-07
76	77	86	79	89	70	2.51E-08	2.00E-08	2.51E-09	1.26E-08	1.26E-09	1.00E-07
76	77	85	81	76	69	2.51E-08	2.00E-08	3.16E-09	7.94E-09	2.51E-08	1.26E-07
75	78	80	80	81	70	3.16E-08	1.58E-08	1.00E-08	1.00E-08	7.94E-09	1.00E-07
75	81	79	79	80	75	3.16E-08	7.94E-09	1.26E-08	1.26E-08	1.00E-08	3.16E-08
81	80	80	77	81	67	7.94E-09	1.00E-08	1.00E-08	2.00E-08	7.94E-09	2.00E-07
85	80	82	78	80	68	3.16E-09	1.00E-08	6.31E-09	1.58E-08	1.00E-08	1.58E-07
81	81	81	76	86	67	7.94E-09	7.94E-09	7.94E-09	2.51E-08	2.51E-09	2.00E-07
80	82	80	78	76	67	1.00E-08	6.31E-09	1.00E-08	1.58E-08	2.51E-08	2.00E-07
78	81	79	77	79	69	1.58E-08	7.94E-09	1.26E-08	2.00E-08	1.26E-08	1.26E-07
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-74.8	-78	-81.9	-78.7	-79.7	-71.3	-74.3851	-77.8585	-81.4861	-78.3519	-78.5264	-70.0134

Distance From Origin (m)											
X = 2.7						Y = 6.45					
Power (dB)						Power(mw)					
76	81	81	73	85	75	2.51E-08	7.94E-09	7.94E-09	5.01E-08	3.16E-09	3.16E-08
78	83	82	74	82	81	1.58E-08	5.01E-09	6.31E-09	3.98E-08	6.31E-09	7.94E-09
77	86	82	75	76	82	2.00E-08	2.51E-09	6.31E-09	3.16E-08	2.51E-08	6.31E-09
78	87	79	76	77	81	1.58E-08	2.00E-09	1.26E-08	2.51E-08	2.00E-08	7.94E-09
78	87	77	65	89	81	1.58E-08	2.00E-09	2.00E-08	3.16E-07	1.26E-09	7.94E-09
76	88	77	66	76	86	2.51E-08	1.58E-09	2.00E-08	2.51E-07	2.51E-08	2.51E-09
84	77	79	65	86	73	3.98E-09	2.00E-08	1.26E-08	3.16E-07	2.51E-09	5.01E-08
86	78	79	65	82	73	2.51E-09	1.58E-08	1.26E-08	3.16E-07	6.31E-09	5.01E-08
79	76	74	66	76	74	1.26E-08	2.51E-08	3.98E-08	2.51E-07	2.51E-08	3.98E-08
76	76	76	69	79	73	2.51E-08	2.51E-08	2.51E-08	1.26E-07	1.26E-08	5.01E-08
80	78	76	71	82	74	1.00E-08	1.58E-08	2.51E-08	7.94E-08	6.31E-09	3.98E-08
82	80	78	70	82	74	6.31E-09	1.00E-08	1.58E-08	1.00E-07	6.31E-09	3.98E-08

80	84	79	70	79	75	1.00E-08	3.98E-09	1.26E-08	1.00E-07	1.26E-08	3.16E-08
79	83	78	71	80	80	1.26E-08	5.01E-09	1.58E-08	7.94E-08	1.00E-08	1.00E-08
79	80	76	68	78	78	1.26E-08	1.00E-08	2.51E-08	1.58E-07	1.58E-08	1.58E-08
Average(non converted)						Log Average (After dBm to mW to dBm conversion)					
Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6	Beacon 1	Beacon 2	Beacon 3	Beacon 4	Beacon 5	Beacon 6
-78.8	-81.9	-78.6	-69.4	-80.8	-77.9	-77.9065	-79.7028	-77.8737	-67.6356	-78.9466	-75.9441

Appendix H: 2/18/18 Readings

Beacon 11													
Power (dBm)							Power(mw)						
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
65	68	71	75	78	84	79	3.16E-07	1.58E-07	7.94E-08	3.16E-08	1.58E-08	3.98E-09	1.26E-08
67	67	72	74	80	85	84	2.00E-07	2.00E-07	6.31E-08	3.98E-08	1.00E-08	3.16E-09	3.98E-09
64	68	74	84	81	75	83	3.98E-07	1.58E-07	3.98E-08	3.98E-09	7.94E-09	3.16E-08	5.01E-09
65	67	75	80	86	78	83	3.16E-07	2.00E-07	3.16E-08	1.00E-08	2.51E-09	1.58E-08	5.01E-09
65	70	73	77	85	82	80	3.16E-07	1.00E-07	5.01E-08	2.00E-08	3.16E-09	6.31E-09	1.00E-08
67	72	74	79	86	81	83	2.00E-07	6.31E-08	3.98E-08	1.26E-08	2.51E-09	7.94E-09	5.01E-09
65	74	73	70	72	76	81	3.16E-07	3.98E-08	5.01E-08	1.00E-07	6.31E-08	2.51E-08	7.94E-09
65	69	75	71	75	78	79	3.16E-07	1.26E-07	3.16E-08	7.94E-08	3.16E-08	1.58E-08	1.26E-08
68	71	77	81	77	79	80	1.58E-07	7.94E-08	2.00E-08	7.94E-09	2.00E-08	1.26E-08	1.00E-08
67	67	71	83	80	80	84	2.00E-07	2.00E-07	7.94E-08	5.01E-09	1.00E-08	1.00E-08	3.98E-09
68	66	72	78	77	81	82	1.58E-07	2.51E-07	6.31E-08	1.58E-08	2.00E-08	7.94E-09	6.31E-09
Average(non converted)							Log Average (After dBm to mW to dBm conversion)						
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
-66	-69	-73.364	-77.455	-79.727	-79.909	-81.636	-65.798	-68.441	-73.025	-75.279	-77.705	-78.941	-81.253
Beacon 2													
							Power(mw)						

Power (dB)													
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
67	72	74	76	83	85	88	2.00E-07	6.31E-08	3.98E-08	2.51E-08	5.01E-09	3.16E-09	1.58E-09
66	78	77	80	82	88	90	2.51E-07	1.58E-08	2.00E-08	1.00E-08	6.31E-09	1.58E-09	1.00E-09
66	70	79	78	88	90	87	2.51E-07	1.00E-07	1.26E-08	1.58E-08	1.58E-09	1.00E-09	2.00E-09
67	71	74	77	81	83	92	2.00E-07	7.94E-08	3.98E-08	2.00E-08	7.94E-09	5.01E-09	6.31E-10
64	77	79	81	84	91	90	3.98E-07	2.00E-08	1.26E-08	7.94E-09	3.98E-09	7.94E-10	1.00E-09
71	80	82	86	82	79	95	7.94E-08	1.00E-08	6.31E-09	2.51E-09	6.31E-09	1.26E-08	3.16E-10
70	77	87	83	85	88	87	1.00E-07	2.00E-08	2.00E-09	5.01E-09	3.16E-09	1.58E-09	2.00E-09
66	74	81	84	80	90	83	2.51E-07	3.98E-08	7.94E-09	3.98E-09	1.00E-08	1.00E-09	5.01E-09
72	80	78	83	85	88	91	6.31E-08	1.00E-08	1.58E-08	5.01E-09	3.16E-09	1.58E-09	7.94E-10
66	72	76	85	82	81	90	2.51E-07	6.31E-08	2.51E-08	3.16E-09	6.31E-09	7.94E-09	1.00E-09
67	71	79	84	87	89	88	2.00E-07	7.94E-08	1.26E-08	3.98E-09	2.00E-09	1.26E-09	1.58E-09
Average(non converted)							Log Average (After dBm to mW to dBm conversion)						
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
-67.455	-74.727	-78.727	-81.545	-83.545	-86.545	-89.182	-66.904	-73.419	-77.523	-80.306	-82.950	-84.672	-88.132
Beacon 3													
Power (dB)							Power(mw)						
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
62	77	73	86	87	81	84	6.31E-07	2.00E-08	5.01E-08	2.51E-09	2.00E-09	7.94E-09	3.98E-09
73	65	74	80	81	87	80	5.01E-08	3.16E-07	3.98E-08	1.00E-08	7.94E-09	2.00E-09	1.00E-08
63	74	75	82	82	87	86	5.01E-07	3.98E-08	3.16E-08	6.31E-09	6.31E-09	2.00E-09	2.51E-09
72	77	76	78	83	82	83	6.31E-08	2.00E-08	2.51E-08	1.58E-08	5.01E-09	6.31E-09	5.01E-09
69	79	79	76	81	79	85	1.26E-07	1.26E-08	1.26E-08	2.51E-08	7.94E-09	1.26E-08	3.16E-09
73	79	81	79	80	83	88	5.01E-08	1.26E-08	7.94E-09	1.26E-08	1.00E-08	5.01E-09	1.58E-09
71	72	78	76	81	84	80	7.94E-08	6.31E-08	1.58E-08	2.51E-08	7.94E-09	3.98E-09	1.00E-08
70	73	78	73	79	88	82	1.00E-07	5.01E-08	1.58E-08	5.01E-08	1.26E-08	1.58E-09	6.31E-09
72	71	80	73	86	82	90	6.31E-08	7.94E-08	1.00E-08	5.01E-08	2.51E-09	6.31E-09	1.00E-09
66	80	78	80	79	80	87	2.51E-07	1.00E-08	1.58E-08	1.00E-08	1.26E-08	1.00E-08	2.00E-09
73	79	78	81	77	79	89	5.01E-08	1.26E-08	1.58E-08	7.94E-09	2.00E-08	1.26E-08	1.26E-09

Average(non converted)							Log Average (After dBm to mW to dBm conversion)						
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
-69.455	-75.091	-77.273	-78.545	-81.455	-82.909	-84.909	-67.480	-72.377	-76.601	-77.076	-80.646	-81.944	-83.710
Beacon 1													
Power (dB)							Power(mw)						
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
54	69	66	70	75	88	90	3.98E-06	1.26E-07	2.51E-07	1.00E-07	3.16E-08	1.58E-09	1.00E-09
56	65	67	71	75	82	87	2.51E-06	3.16E-07	2.00E-07	7.94E-08	3.16E-08	6.31E-09	2.00E-09
59	64	63	70	77	85	89	1.26E-06	3.98E-07	5.01E-07	1.00E-07	2.00E-08	3.16E-09	1.26E-09
60	71	72	65	67	84	86	1.00E-06	7.94E-08	6.31E-08	3.16E-07	2.00E-07	3.98E-09	2.51E-09
57	60	69	68	73	85	95	2.00E-06	1.00E-06	1.26E-07	1.58E-07	5.01E-08	3.16E-09	3.16E-10
57	61	68	75	78	82	92	2.00E-06	7.94E-07	1.58E-07	3.16E-08	1.58E-08	6.31E-09	6.31E-10
56	65	63	68	73	88	84	2.51E-06	3.16E-07	5.01E-07	1.58E-07	5.01E-08	1.58E-09	3.98E-09
58	67	67	65	71	81	86	1.58E-06	2.00E-07	2.00E-07	3.16E-07	7.94E-08	7.94E-09	2.51E-09
59	69	64	77	70	89	85	1.26E-06	1.26E-07	3.98E-07	2.00E-08	1.00E-07	1.26E-09	3.16E-09
59	61	65	76	72	81	87	1.26E-06	7.94E-07	3.16E-07	2.51E-08	6.31E-08	7.94E-09	2.00E-09
58	61	69	72	78	80	97	1.58E-06	7.94E-07	1.26E-07	6.31E-08	1.58E-08	1.00E-08	2.00E-10
Average(non converted)							Log Average (After dBm to mW to dBm conversion)						
1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter	1 meter	2 meter	3 meter	4 meter	5 meter	6 meter	7 meter
-57.545	-64.818	-66.636	-70.636	-73.545	-84.091	-88.909	-57.204	-63.473	-65.880	-69.051	-72.237	-83.152	-87.500