# IBEACON LOCALIZATION APPLICATION FOR GARAGES AND PARKING LOTS

A Major Qualifying Project
Submitted to the faculty of
WORCESTER POLYTECHNIC INSTITUTE
in partial fulfillment of the requirements for the
Degree of Bachelor of Science
Electrical and Computer Engineering

Submitted by:
James Atchue, Kornkanok Bunwong
Alexander O'Neil, and Joseph Salice

Advisors:
Professor Kaveh Pahlavan,
Professor Shamsur Mazumder

February 28, 2018

# 1 Abstract

The objective of this project was to develop a system and smart phone application using iBeacon$^{TM}$ technology to determine location and parking availability in parking garages and lots. This task was achieved with the use of localization algorithms and probabilistic models to accurately determine a user's location in comparison to any empty parking spaces and alert the user of the spot's vacancy. The first phase of this process was determining the market space for such technology, as well as a potential customer for the system. Upon completion of this phase, a path-loss model was determined based on the received signal strength from the iBeacon devices inside a test parking garage located at Worcester Polytechnic Institute. The information from this path-loss model helped create some probabilistic models utilizing the Cram*é*r-Rao Lower Bound for accuracy and precision, which were later used to compare with the localization algorithms implemented in the smart phone application.

## 2 Acknowledgments

The team would like to acknowledge several individuals who greatly impacted the success of this project. First, Professor Kaveh Pahlavan, who was the main advisor for this project. Professor Pahlavan's guidance, insight and wisdom was incredibly helpful in first developing path-loss models, and then understanding and implementing the various localization algorithms used throughout this project. Second, Professor Shamsur Mazumder, who graciously assisted Professor Pahlavan in providing guidance and structure to the project, as well working diligently to help improve the quality of our report and progress when Professor Pahlavan was unavailable. Thirdly, Julang Ying, a WPI graduate student, who developed the initial MATLAB codes that were used as a starting point for the final implemented algorithms, as well as providing support and assistance with any questions about coding and the aforementioned algorithms. Without the assistance of these three individuals, this project would not be possible.

Finally, the team would like to thank Worcester Polytechnic Institute and the Electrical and Computer Engineering department for providing funding, resources, and testing environments necessary in the completion of this project.

# 3 Executive Summary

The goal of this project was the development of a location sensing network utilizing Apple's iBeacon$^{TM}$ technology to determine both location and parking availability in parking garages and lots. The primary interface that a potential customer would interact with would be a smart phone application, which depicts any available parking spaces in a given garage or lot. The application contains the majority of the processing power, as it runs localization algorithms in the background while the user interface depicts a layout of the garage. The application communicates with iBeacon devices in front of each spot in the garage. The information coming from the beacon serves two purposes: the readings from the on-board magnetometer, which detects the presence of a metal object, or more specifically a vehicle, as well as allowing the application to determine the proximity to each individual beacon. The application reads the received signal strength from the surrounding beacons, and uses the aforementioned localization algorithms to calculate the relative location of the user, while the magnetometer determines if the spot is available.

The purpose of designing such an application is to help reduce the time and frustration of finding available parking in densely crowded, urban environments. The main scenario that this application could be used for is in an environment similar to General Edward Lawrence Logan International Airport, located in Boston, Massachusetts. The large parking garage at Logan International is infamously difficult to navigate, and is often overcrowded, causing frustration and traffic jams when searching for empty spots. With the use of this application and system architecture, delays and difficulties to find parking could be drastically reduced if not eliminated.

The completion of this project was a multifaceted, complex challenge that required various pieces to build off of one another. The first phase of this process was determining the market space for the overall system, as well as a potential customer. Market analysis determined that the market for parking applications had high desirability with little product or innovation. Upon completion of this initial phase, it was determined that a path-loss model would need to be created to determine the overall accuracy and strength of the supplied iBeacon Estimote devices. This step was critical in developing a path-loss model for analyzing the effectiveness of the specific bluetooth

beacons used, as each location would require a slightly different architecture, based on the wireless conduciveness of the environment. Using the Estimote smart phone application, various readings of the received signal strength from the beacons were taken, both in an outdoor and indoor environment. Based on these readings, a path-loss model was constructed, which compared the received signal strength to the distance, both actual distance and the distance the Estimote application perceived the beacon to be at. From this model, it was evident that the Estimote beacons were highly inaccurate, particularly in the parking garage, where the heavy concrete pitched ceilings created a very poor wireless environment. This model also helped create the baseline parameters from some initial probabilistic models, particularly the Cram$\acute{e}$r-Rao Lower Bound (CRLB) simulations. The calculated signal-to-noise ratio and standard deviation of the received signal strength were used as factors in the CRLB simulations, which gave insight into how effective the beacons would be at not only transmitting around the entire space, but also how accurate the measurements would be, based on the layout of the beacons. The results from the CRLB simulations acted as perfect scenario baselines for the accuracy of the localization algorithm. The first algorithm discussed briefly was trilateration, but this idea was quickly discarded, as trilateration would be highly inaccurate compared to more sophisticated algorithms. The second algorithm explored was Maximum likelihood, which determines which beacons are closest to the user, and estimates the location of the user based on those readings. The third algorithm explored was Least Mean Squared, which is a more powerful version of trilateration, in that it creates several expectation regions for the location of the user, and pinpoints the user where the expectation regions overlap. These two algorithms were both implemented in the smart phone application, and their accuracies were compared with the perfect prediction of the CRLB simulations. The calculations and comparisons demonstrated that the Maximum Likelihood algorithm was slightly more accurate than the Least Mean Squared, though both could be used reliably to achieve an accuracy of error of approximately a meter. This error is acceptable, as the dimensions of a typical parking spot is six meters by three meters, and thus the localization does not necessarily have to be perfect.

Towards the end of the project, some tests were done with different iBeacon$^{TM}$ devices, and it was discovered that the original Estimote iBeacon devices were of low quality, inaccurate, and practically unusable when compared to the new devices. It is recommended that future projects simply begin testing with higher quality devices.

# 4 Authorship/Responsibilities

- James Atchue
  - Authorship: Localization Algorithms, Market Research and Analysis, Path-Loss Models, Localization Algorithms Implementations, Comparison between Cramér-Rao Lower Bound and Localization Algorithms, Proposed Architecture
  - Responsibilities: Smart Phone Application, Localization Algorithm development and implementation
- Kornkanok Bunwong
  - Authorship: Localization Algorithms, Methodology, Path-Loss Models, Localization Algorithms Implementations, Proposed Architecture
  - Responsibilities: Developing guideline for Design and Performance Methodology, Localization Algorithm research and development
- Alexander O'Neil
  - Authorship: Abstract, Executive Summary, Introduction, Foundations of Bluetooth Low Energy, Cramér-Rao Lower Bound, Comparison between Cramér-Rao Lower Bound and Localization Algorithms, Conclusion and Future Recommendations
  - Responsibilities: Cramér-Rao Lower Bound (CRLB) simulations, Final report formatting (LaTex)
- Joseph Salice
  - Authorship: Introduction, Magnetometer Theory, Methodology, Magnetometer Readings and Observations, Conclusion and Future Recommendations, Proposed Architecture
  - Responsibilities: Magnetometer theory and analysis

# Contents

# List of Figures

# List of Tables

# 5 INTRODUCTION

This project was started with the goal of fixing a common problem that most people who commute face on a daily basis: parking availability. The issue that was identified was that when people wanted to park they often would have issues finding an open spot especially in crowded parking garages with multiple levels. With the issue determined the next step was to design a system that could make finding parking spots in a crowded garage quick and easy. To achieve this goal a system was developed with the use of Estimote iBeacons and the built in magnetometer. This system could detect open spots using the magnetometer in the parking garage and navigate the user to the open spot through the use of a smart phone application.

## 5.1 Project Motivation

The motivation for this project was derived from direct observations on the difficulty and frustration of finding parking in busy urban parking lot and garages. In particular, the various parking lots across the Worcester Polytechnic Institute campus generally fill rapidly in the morning, causing commuter students and professors to find alternative street parking, risking scratches or dents to their vehicles, or the potential of having their vehicle towed. This challenge of parking was also evident in a case study of the Logan International Airport garage, which is infamously difficult to navigate and find available parking spaces. From this motivation, the team decided an interesting application for the iBeacon$^{TM}$ technology would be the development of a smart phone application that could both determine whether a parking space was available, and illustrate the vacancy to a user. The creation of such an application could allow drivers to check the availability of the parking lot before entering, as well as determine the fastest route to available spots instead of slowly scanning each space.

## 5.2 Report Outline

This report includes a complete summary of all aspects of this project, including the fundamental background information and technology of the project, the methodology followed for the completion of the project, and the results discovered throughout the entirety of the project. The Background Information section serves as an introduction to Bluetooth Low Energy, Apple's iBeacon$^{TM}$ technology, several localization algorithms, and to magnetometers. The Design and Implementation Methodology section discusses the various stages in research and development of the localization algorithms and smart phone application. The Results and Discussion section depicts all of the results from this project, beginning with the initial path-loss models, and continuing through the Cramér-Rao Lower Bound simulations and localization algorithms, and concluding with a comparison of accuracy between the probabilistic models created by Cramér-Rao Lower Bound and the implemented algorithms. This section also includes readings from the magnetometers and a discussion of the results. The Conclusion and Future Recommendations section summarizes the project in its entirety, and provides recommendations to future project groups that may choose to develop an application similar to the one detailed in the project, or choose to use a similar system architecture. Several appendices are also added to the end of this report. These appendices include the MATLAB code used to generate the path-loss model diagrams, as well as the code to generate the various Cramér-Rao Lower Bound simulations. Additional appendices include the C.# and MATLAB code for the construction of the localization algorithms in the smart phone application.

# 6 Background Information

This project focuses on three main aspects, each of which are key components to the overall success of the smart phone application and therefore the project. The first component is the use of Bluetooth Low Energy (BLE) technology, a wireless network connection which is used to exchange tiny and static radio signals within short distances. The BLE standard for this project is Apple's iBeacon $^{\text{TM}}$ technology. The physical foundation of this project is the use of Estimote iBeacon $^{\text{TM}}$ devices to create the wireless network. The second component is the localization algorithms and comparative probabilistic models created by Cram$\acute{e}$r-Rao Lower Bound simulations. The localization algorithms are used to determine a users' relative distance from a iBeacon $^{\text{TM}}$ device, and calculate the users location based on the response from several devices simultaneously. The accuracy of the algorithms were compared against the ideal errors illustrated by the Cram$\acute{e}$r-Rao Lower Bound simulations. The third and final component of this project are the internal magnetometers embedded in the Estimote devices, which can be calibrated to determine the presence of a large metallic object in front of the device. The magnetometers are crucial in determining the availability of the parking space in a garage. The final section of this background section analysis the market space for the proposed system architecture and smart phone application.

## 6.1 Foundations of Bluetooth Low Energy Technology

Bluetooth Low Energy (BLE) is a wireless network connection used to send static radio signals within local areas, up to approximately 200 meters, depending on the device. BLE was first introduced by Bluetooth Special Interest Group (SIG) in 2009 [1]. BLE was later re-branded as Bluetooth Smart, though the terms are interchangeable. The goal of creating BLE was to create a standard that would allow devices to run for long spans of time with low energy consumption. Depending on the device, a BLE device's battery can last two-three years.

Both BLE and classic Bluetooth utilize the 2.4 GHz ISM band, though they are not compatible. Both technologies use frequency hopping spread spectrum to spread their RF energy. Classic Bluetooth uses seventy-nine 1 MHz-wide channels, while BLE uses forty 2 MHz-wide channels[2]. This spectrum can be seen below in Figure 1, which depicts not only the forty 2 MHz-wide channels, but differentiates three "advertising" channels, which are used by a device to send advertising packets with information about the device so other BLE devices can connect [3][4]. These three channels are selected in the lower, middle, and upper regions to avoid interference that may interfere with multiple channels at once.

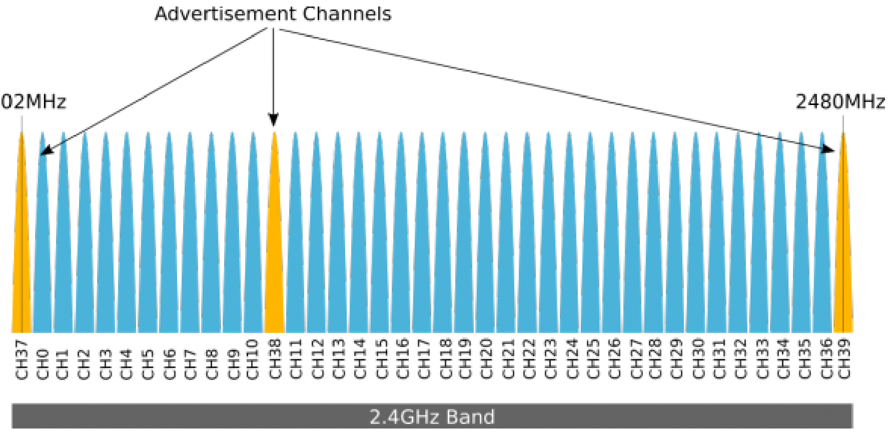


**Figure 1:** BLE's utilization on 2.4 GHz band, using forty 2 MHz-wide channels. [Argenox Technologies]

### 6.1.1 iBeacon$^{\text{TM}}$ Technology

iBeacon$^{\text{TM}}$ is a version of BLE created by Apple to provide local area geolocation, and was intended for indoor environments. This technology was first implemented through Apple's operating system for iPhones and iPads, iOS, and has been available since iOS 7. A device with iBeacon technology can be used to establish a region around an object, allowing any iOS device to determine a proximity estimation to said device, as well as determine if the iOS device has entered or left the created region[5]. It is important to note that iBeacon is omni-directional, meaning a transmitter device will broadcast its information, but does not receive signals. The transmitting device broadcasts its information using three-part identifier packets which contain a Universal Unique Identifier (a 16-byte string), a 2-byte Major number, and a 2-byte Minor number[5]. Table 1 illustrates an example of how these values might be used for an east coast retail store. The UUID is shared by all three locations, while each store is assigned a unique Major value, allowing an iOS device to identify which store it is in. Inside each store, different departments are assigned a unique Minor value.

| Store Location | | Boston | New York | Worcester |
|---|---|---|---|---|
| UUID | | 23DF590C-0DAB-767E-DEA6-2AC2FD4211AC | | |
| Major | | 1 | 2 | 3 |
| Minor | Automotive | 10 | 10 | 10 |
| | Toys | 20 | 20 | 20 |
| | Shoes | 30 | 30 | 30 |

**Table 1:** An example of iBeacon packet identification, depicting the UUID, Major, and Minor values.

### 6.1.2 Estimote Beacons and Virtual Beacons

For this project, a combination of Estimote Proximity Beacons and Virtual Beacons were be analyzed, as they incorporate BLE through iBeacon$^{TM}$ for localization. The Estimote Proximity Beacon is developed by Estimote Inc., and can be configured via an Estimote mobile application. Utilizing the Estimote application, the transmitted power can be altered from 4 dBm to -30 dBm. The beacons themselves are approximately 55mm by 38mm by 18mm, and are equipped with built in motion, light, and temperature sensors[6]. Figure 2 illustrates several Estimote Proximity Beacons, as well as the main interface of the Estimote mobile application.

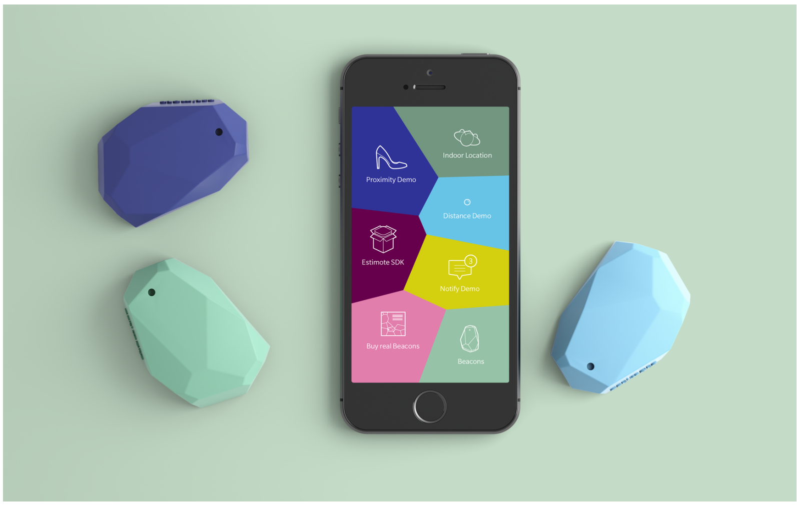
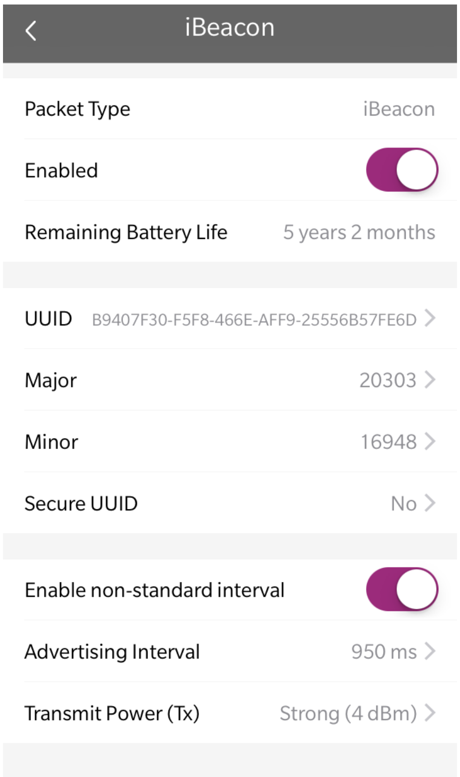


**Figure 2:** Three Estimote Proximity Beacons and the main user interface of the Estimote mobile application. [Estimote]

All of the Estimote Proximity Beacons have a default UUID, while the Major and Minor codes are randomized. The UUID B9407F30-F5F8-466E-AFF9-25556B57FE6D default can be changed by the user. Figure 3 depicts an example of the default UUID, as well as randomized Major and Minor codes, as seen on the Estimote mobile application. Note that the UUID could be changed from this screen, and the transmit power can also be altered from this screen.



iBeacon

| | |
|---|---|
| Packet Type | iBeacon |
| Enabled | ● |
| Remaining Battery Life | 5 years 2 months |
| UUID | B9407F30-F5F8-466E-AFF9-25556B57FE6D |
| Major | 20303 |
| Minor | 16948 |
| Secure UUID | No |
| Enable non-standard interval | ● |
| Advertising Interval | 950 ms |
| Transmit Power (Tx) | Strong (4 dBm) |

**Figure 3:** Estimote mobile application screen depicting the UUID, Major, and Minor codes, as well as the transmit power. [Estimote]

## 6.2 Magnetometer Theory

One of the built in sensors in the Estimote iBeacon$^{\text{TM}}$ device is a magnetometer, an instrument that can measure magnetism or the change of a magnetic field at a particular location. When a chunk of magnetic material is introduced to the sensors location a change in the magnetic field can be detected by the magnetometer. However this will only happen if a magnetic metal is introduced into the sensor's field of view. This is the key difference between metal detectors which can detect all metals compared to magnetometers which can only detect a special group of metals "magnetic metals" such as iron, nickel, cobalt or their alloys like steel. Magnetometers work by measuring the magnetic flux density at the point in space where the sensor is located. Materials that are magnetic can distort the magnetic flux that is flowing around them, a distortion that magnetometers can detect. An example of a magnetic material disrupting the magnetic flux can be seen below in Figure 4.

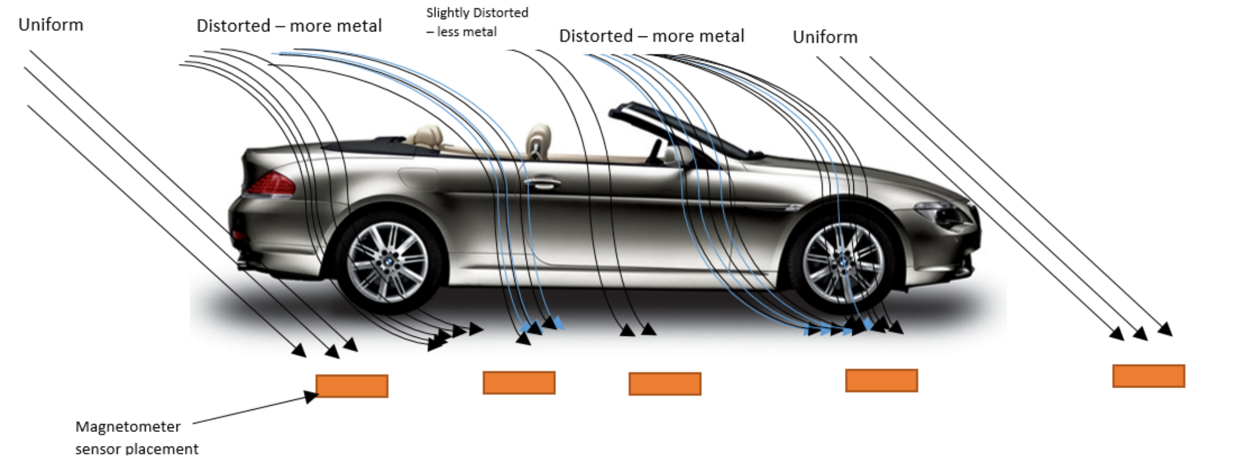

Uniform
Distorted – more metal
Slightly Distorted – less metal
Distorted – more metal
Uniform
Magnetometer sensor placement

**Figure 4:** Magnetic field disruption caused by a vehicle, with magnetometers in place to record the magnetic field. [NXP Semiconductors]

In reference to the previous figure before the car approaches the sensor, the magnetometer would experience a uniform field due to no interference from magnetic materials. However, as the vehicle passes over the sensor it would distort the earth's magnetic field and create regions of decreased flux and increased flux line concentration. This distortion can be measured and graphed as seen in below in Figure 5.

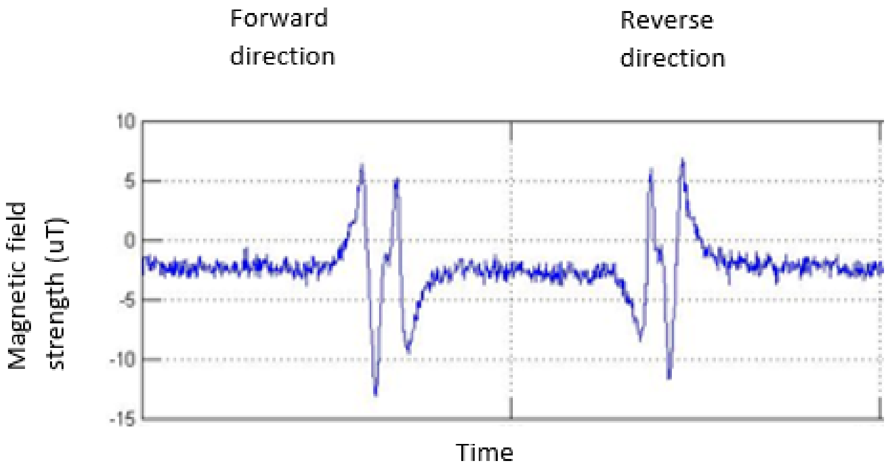


**Figure 5:** Magnetic Field Strength with respect to time

There are multiple types of magnetometers with the simplest example being a compass that only measures the direction of an ambient magnetic field. Other magnetometers can be used to detect the direction, strength and relative change of a magnetic field at a particular location. In our testing case the magnetometer was used as a metal detector, magnetometers can detect large objects such as cars up to around 10 meters. The two basic types of magnetometers can be classified as vector magnetometers and scalar magnetometers. Vector magnetometers measure the vector components of a magnetic field as the name suggests while scalar magnetometers measure the magnitude of the vector magnetic field.

The performance and strength of magnetometers can be measured through their technical specifications which include; Sample Rate, Bandwidth, Resolution, Absolute error, and Thermal stability. The sample rate of a magnetometer is the amount of measurements taken every second, this is one of the most important specifications because a high sample rate allows you to measure quick changes in the magnetic field. The bandwidth characterizes how well a magnetometer can measure quick changes in the magnetic field. The resolution is the smallest change in the magnetic field that the magnetometer can measure which is measured in teslas. Absolute error is difference between the measured magnetic field and the actual value. The thermal stability of a magnetometer is how its measurements and accuracy are affected by variations of the temperature surrounding the sensor. Combining all of these technical specifications together can determine the capabilities of the given magnetometer.

## 6.3 Cram$\acute{e}$r-Rao Lower Bound

A crucial tool in comparing the accuracy of the localization algorithms was through comparing the ranges of accuracy with the contour of location error standard deviation created by a mathematical approximation called Cramer-Rao Lower Bound (CRLB). This approximation allows for a contour mapping of location error based on a set of predetermined access points. The purpose of these models is to determine the most effective locations for the iBeacon crystals, such that not only the entire space was mapped by the beacons, but that the localization accuracy has a low error rate.

## 6.4 Localization Algorithms

In order to achieve a successful application for localization, a localization algorithm is required to calculate location. There are numerous algorithms that can be utilized for localization, each with some advantages and disadvantages. In the scope of this project, three localization algorithms were analyzed and compared, with a heavy focus on accuracy. These three algorithms include trilateration, least mean squared, and maximum likelihood.

### 6.4.1 Trilateration

The initial algorithm that was analyzed is a fairly simple algorithm called trilateration. This algorithm is very similar to LMS but it uses spheres to determine distances and locations in three dimension and uses circles to determine distances and locations in one and two dimensions configuration[7].

Trilateration calculations use distance measurements to determine the three dimensional coordinates of unknown positions[8]. This algorithm is very similar to LMS but it uses spheres to determine distances and locations in three dimension and uses circles to determine distances and locations in one and two dimensions configuration. This is done by creating a radius around different access points and determining where the circle radii intersect. A basic example of trilateration can be seen in figure 6.

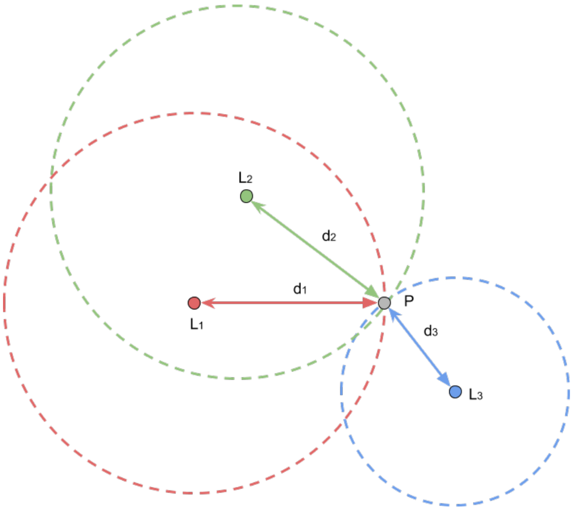


**Figure 6:** Example of trilateration. Each point has a radius, and the Intersection point of the radii is the location.

### 6.4.2 Least Square Method

This algorithm estimates the location of an access point (AP) in x and y coordinates as the minimizer of the sum of error functions:

$$E = \sum_{i=1}^{n} f i^2 \tag{1}$$

Where $(x_i, y_i)$ is the location of of the $i^{th}$ base station and $f_i$ is the error function of each base station. Note that the choice of $f_i$ can be varied. In this case, there are two options for error functions to be considered:

$$f_{1i}(x, y) = \sqrt{(x_i - x^2) + (y_i - y^2)} - d_i^2 \tag{2}$$

$$f_{2i}(x, y) = (x_i - x^2) + (y_i - y^2) - d_i^2 \tag{3}$$

These two options for error function will be describe more in depth in the Methodology section. In order to solve the minimization problem, an iterative approach is required. Thus, Newton's method and Gauss-Newton's method are compared [9][10]. Newton's method is an iterative method for finding the roots of a differentiable function. Gauss-Newton's method is a particular case of Newton's method and it is used to solve nonlinear least squares to minimize least square residuals.

### 6.4.3 Maximum Likelihood

Maximum likelihood algorithm is similar to and based on LMS and trilateration algorithms but with much more accuracy. This algorithm provides a method of estimating distances, locations, and parameters of a statistical model based on observations and estimations. The algorithm makes an estimation of parameter $\theta$ based on the input value of $\hat{\theta}$ that would provide a maximum likelihood function. Maximum Likelihood uses bounds to create zones of possible maxima that would allow the function to converge on a single point[11]. Using the derived path-loss model (discussed in Section 7.1) would give:

$$d(p_r) = 10^{\frac{p_r - (p_0 + x)}{10\alpha}} \tag{4}$$

Based on the above equation and using a computed distance from a single base station an inner and outer radius would be determined by:

$$r_{inner} = d\left(p_{avg} + 2\sigma\right) \tag{5}$$

$$r_{outer} = d\left(p_{avg} - 2\sigma\right) \tag{6}$$

Using two or more base-station distances would allow the algorithm to have a densely populated region between the inner and outer radii from each function, thus providing an estimate localization. To increase search space in non-line of sight conditions the parameter $\sigma$ can be changed to create larger overlapping regions based on the path-loss model[10] . An example of maximum likelihood can be seen in Figure 7.

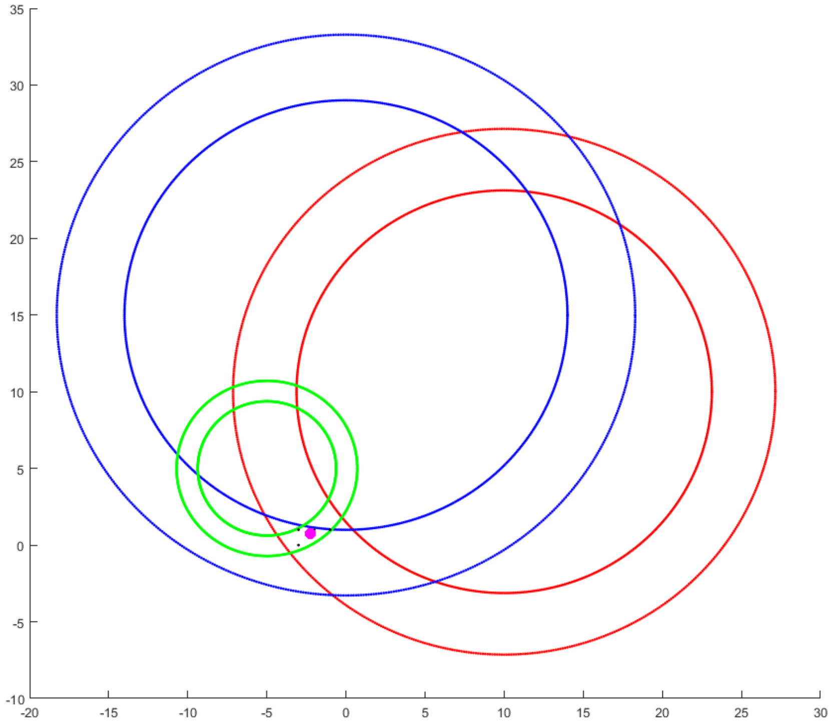


**Figure 7:** Example of results from maximum likelihood algorithm

This process can be accomplished by determining the parameters that would maximize the probability of making the observations or guesses given the parameters. Out of the three algorithms, this one provides maximum accuracy for estimating distances, locations, and parameters.

## 6.5 Market Research and Analysis

The project's objectives have solutions that cover many entities including airports, concessions, universities, malls, stadiums, and municipal, all of which have parking lots or garages. However, for the scope of this project, one entity in particular was chosen to model the objectives and solutions. Massport, The Massachusetts Port Authority, maintains and owns various parking areas throughout the state of Massachusetts mainly for use in public facilities; these locations consist of Worcester Regional Airport, Hanscom Field, Flynn Cruiseport Boston, Conley Terminal, and Boston Logan [12]. Most of these locations have "dummy" parking which does not have metrics on the amount of spaces available or smart pay systems. At Worcester Airport for example, parking is charged by a legacy gate ticketing system. The processes of parking looks like the following:

- Enter parking lot, receive ticket at entrance
- Find open parking spot
- Pay for parking inside at box
- Place receipt into ticketing machine upon exit

This parking architecture introduces many issues such as congestion, unknown parking availability, and confusion of payment. Especially at peak rush hour times congestion can become an issue at both exiting the parking lot as well as the ticket payment boxes. To relieve the bottlenecks of parking, many systems can be automated thus eliminating a lot of the existing parking infrastructure.

Many universities and companies are spending resources on bringing parking solutions into the twenty first century with IoT (Internet of Things) applications. These solutions are looking to provide both parking availability metrics and payments systems. The overall goal of the research is to replace legacy parking infrastructure, such as ticketing, with more advanced "smart" parking architecture. Some of the solutions found during the research of this project include license plate tracking, vehicle transponders, phone applications, pre-registration techniques, state sanctioned billing, among many others. Needless to say, the area of parking research is growing as the need for smarter parking systems grows. One specific example is Beijing where parking garages can hold upwards of thousands of vehicles; these congestion issues are outlined in a report from Beijing Technical University students where they report that parking need is growing exponentially while solutions to maintain this parking are non-existent[13]. In 2014, Beijing had “5 million vehicles in Beijing with only 2 million parking lots,” which further shows the need for parking and smarter systems to control the parking[13]. As parking lots grow in their number of spots, congestion and related parking issues will grow as well.

Some parking garages have started to include systems that would make “semi-smart” infrastructures that output information about the parking statistics of the garage. One company innovating in this field is AllTrafficSolutions, which has implemented their ParkTrak systems at the Natick Mall in Massachusetts, seen in Figure 2.5. ParkTrak is a devices that uses lidar and laser to detect vehicles.

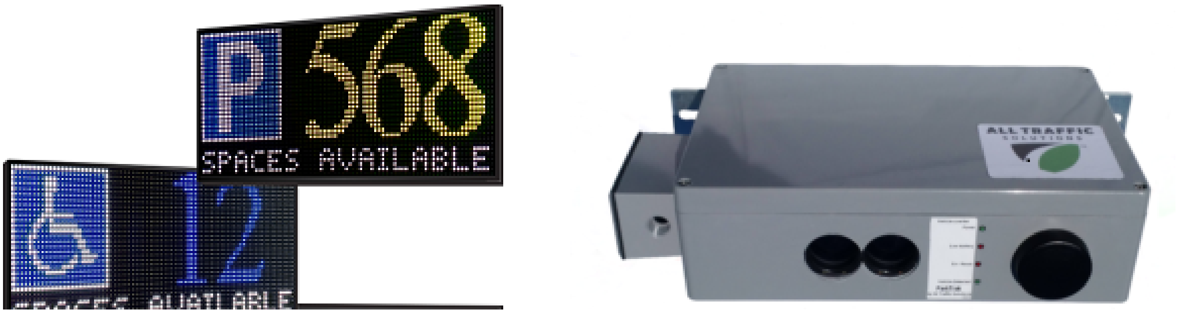


**Figure 8:** ParkTrak device and LED Signage

Their system counts the number of vehicles entering and exiting the garage and is able to display an available parking sign that integrates the remaining parking available. The architecture of this solutions is only “semi-smart” because it cannot determine where the open parking spots are located, only that the garage has a certain amount of vehicles parked. ParkTrak also does not address the ticketing and payment for parking inside the garage, this is left to legacy systems. AllTrafficSolutions claims that their ParkTrak has been implemented into over 5000 commercial and municipal parking garages[14]. The price tag for the system, LED Signs, and implementation costs around $50,000 depending upon the size and amount of entrances to a garage[15]. This was the associated cost that Miami Beach, Florida had incurred to implement the system into their municipal parking lot.

A more encompassing system is OmniPark, which has solutions for enforcement, payment, and tracking of vehicles within its parking area[16]. While their system has significantly more cost and implementation requirements, it has almost all features that a “smart” enabled parking garage would have; their product is referred to as a “fully integrated parking management solution.” Users pay for parking in a pay-by-plate model where bills are mailed based on vehicle registration or paid online for a cheaper fee. The downside to this system is that enforcement is still done manually where an attendant would be required to travel the lot and record license plates and call for vehicle towing when a violator has parked in the lot. OmniPark does not release pricing on their systems, but their charing model does including both implementation and monthly recurring costs.

There are many solutions being researched today that are enabling parking lots to become more involved and smarter with the end users. In the case of Beijing it will help offset the city’s 5+ million vehicle parking requirement and in other implementations will reduce parking congestion and improve overall user experience. The field of research around parking is a new and growing field that will be innovating for decades to come, especially as self driving vehicles are slowly finding their way to market.

# 7 Design and Implementation Methodology

When members of WPI community such as students, faculty, and staff commute to campus and try to park their cars at a parking garage, they often do not find an available parking spot right away and have to keep driving around in the parking garage to find a parking spot that is available. This process might take seconds or even up to an hour. One possible solutions to use iBeacons to create a wireless network that can detect open spaces. iBeacon$^{TM}$ devices can be used to do localization and help people to find available parking spaces without losing much time and energy, and therefore to localize or detect where are cars located and can also be an assistance to find available parking spaces. The mission of this project is to use wireless technologies to make applications for parking garages to guide cars to open parking spots and to approximate the total time a car has been park at parking spot in order to replace the legacy ticketing system and ease the congestion. In order to achieve the mission of this project, the objectives of this project are as following:

1. Determine and develop a path-loss model based on IEEE standards and Estimote iBeacon$^{TM}$ devices
2. Apply the derivation of the path-loss model to the location algorithms and determine accuracy using comparisons to Cram*é*r-Rao Lower Bound
3. Develop magnetometer baseline readings and determine bounds to detect all sizes of vehicles
4. Implement a smart phone application for users to gather data from iBeacon devices

## 7.1 Development of Path-Loss Models

A major portion of the project is dependent upon path loss models, as they are a basis for trilateration and localization. Path loss models in general determine the reduction in power density as radio waves travel through space. By using the general Friis transmission equation, and applying a determinant path loss model an approximate distance can be calculated between a wireless transmitter and receiver. Applying these approximated distances to localization algorithms, a beacon's general location can be determined; this is the basis of trilateration, which is a method of determining a two dimensional location using received signal strength measurements.

The team's initial goal was to calculate path loss models for existing iBeacon$^{TM}$ localization applications, specifically that of Estimote. In doing so, an application specific path loss model can be formulated for the project. One of the many goals of the project is to be able to determine a beacon's general location within a parking lot or parking garage. To understand the difference in the parking environments path loss models will be evaluated for both structure and outside variations of parking lots. The team took measurements of both hardware beacons, Estimote Crystal, and virtual beacons, an Android application broadcasting as a beacon. The overall architecture of the project depends on the results from these measurements as the accuracy of each device will better determine parking locations.

The inputs for the Path-loss model are determined by collecting measurements of received signal strength (RSSI) in dBm at different distances between a smart phone and an iBeacon$^{TM}$ device. The measurements are collected at the WPI parking garage below the rooftop field. A measurement tape was used to measure the parking stall in order to get the exact dimension. One parking space is 9 ft. wide and 18 ft. long or 2.7 m wide and 5.5 m long, then the Estimote application was used to obtain the RSSI reading at a every parking space away from the iBeacons. In other words, a smart phone with Estimote application is at the line of the first parking stall and that would indicate a distance of zero meter between the smart phone and the iBeacon. The smart phone and the iBeacon$^{TM}$ device was then moved one parking spot away from each other, then the RSSI at the location is obtained. The process was repeated until the RSSI can no longer be read. The collected data was used to develop a Path-loss model based upon the Estimote application[17].

A statistical path loss model is characterized by Equation 7, which is the derived received signal strength from the Friis Transmission Equation.

$$P_r = P_0 - 10\alpha log(d) + X \tag{7}$$

where power received is in dBm. To convert between mW received and dBm, the following Equation 8 was used.

$$p\,[dBm] = 10log_{10}\frac{x}{1mW}\,[mW] \tag{8}$$

In the equation, alpha ($\alpha$) denotes the distance power gradient, distance (d) represents the actual distance between the bluetooth receiver and the iBeacon, and lastly $\sigma_{SF}$ is the standard deviation of the shadow fading. Using this path loss equation the team will be able to calculate approximate distances using an alternative form in Equation 9.

$$d(P_r) = 10^{\frac{P_r-(P_0+X)}{10\alpha}} \tag{9}$$

## 7.2 Comparison between Cram$\acute{e}$r-Rao Lower Bound Simulations and Implemented Algorithms

The key focus of the comparison between the accuracy of the implemented algorithms is how close they compare to the probabilistic models created by the Cram$\acute{e}$r-Rao Lower Bound simulations. This portion of the project is crucial in understanding how effective the algorithms are, and having the ability to tweek them as necessary. The first phase of this process was to develop the CRLB models based on various proposed architectures. Then, the algorithms were developed and compared with results from the CRLB simulations various times to attempt to improve their accuracy.

## 7.3 Magnetometer Magnetic Field Analysis

One of the architecture topologies involved with the project will consist of iBeacon$^{TM}$ devices utilizing magnetometers to detect cars in parking spaces throughout a proposed parking garage. With Estimote iBeacons in every spot in a proposed parking garage open spots can be tracked through the use of the built in magnetometer which can then feed data to a map on a smart phone application which can direct the driver to any open spots. An example of this proposed architecture with Estimote iBeacons and smart phone application can be seen in Figure 9 below.

In order to set up this system each iBeacon would have to be calibrated when placed to have a baseline for the magnetic field. Determining the baseline is important because any on site interference can interfere with the detection of cars.This system would work by having magnetometer telemetry from the iBeacon sent to a server on site that would then relay the information to a display on a customer's smart phone.

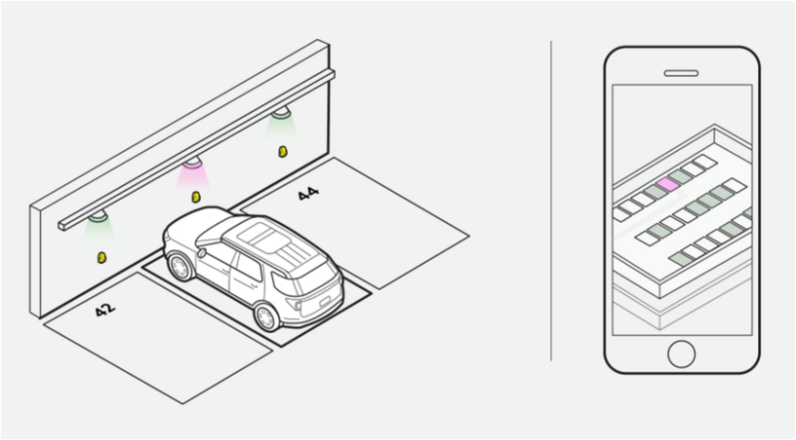


**Figure 9:** Example communication between Estimote Magnetometer and a smart phone application [Estimote]

Figure 10 demonstrates a simple equation that will be tested in this section. On the left side of the equation we see a ferrous metal and the earth's magnetic field uninterrupted. On the right side of the equation we see the ferrous metal introduced into the field which causes disruptions in the earth's magnetic field. In the situation that will be tested the ferrous metal acts as the vehicle and the magnetometer from the Estimote iBeacon$^{TM}$ device will measure the magnetic field which can be seen as the red lines in Figure 10.

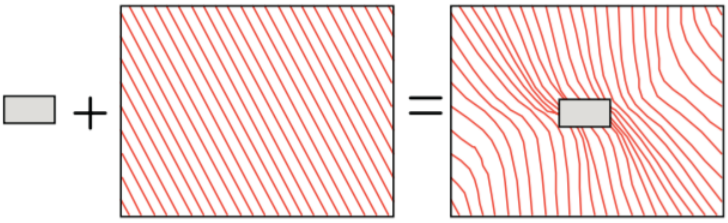


**Figure 10:** Ferrous Metal in a magnetic field

To collect data about magnetometers and how they interact with the goal of vehicle detection, a test would needed to be setup involving a car and one of the built-in Estimote iBeacons$^{TM}$ device magnetometers. This test was done by first deploying and setting an Estimote iBeacon$^{TM}$ devices and taking a baseline reading for the earth's magnetic field at that location. Figure 11 illustrates the setup can be seen below with the vehicle and magnetometer.

With the above setup in place, a vehicle can be driven over the magnetometer while

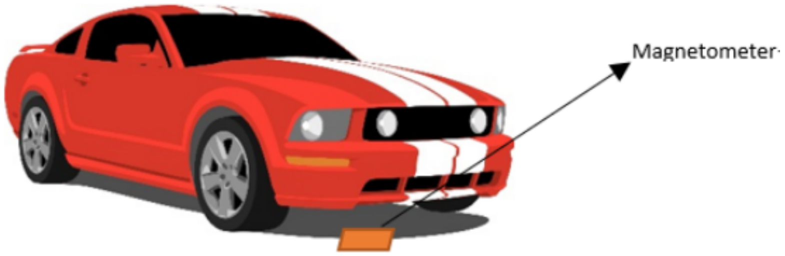


**Figure 11**

taking readings and the resulting data can give insight on how vehicles interact with the earth's magnetic field. For more data about different cars multiple test can be done with vehicles of varying size such as buses, trucks and vans.

## 7.4 Application Development and Proposed Architecture

The overall architecture of the team's project is incorporated into a cell phone application. This app is responsible for collecting and displaying information to the user; it calculates distances using the algorithms derived during the course of the project and can accurately locate itself within a parking garage. Furthermore the app collects data from the iBeacon transmissions which contains the magnetometer data.

After collecting and calculating data, the phone application transmits it's data array to a web server which can broadcast all data to phones even the ones out of range of certain beacons. This comes into use when a particular phone is on one side of the garage and therefore out of range of beacons on the opposing side. Using this network of information the entire garage's status can be transmitted to each phone. The web server takes data from each phone and calculates various metrics that help to determine the garage's overall status.

The structure of the app is an MVVM (Model View View-Model) design pattern where updates in the system thread will not block the UI (User Interface), this is done because the cost of calculating the algorithms continuously strains the application's main thread. By using MVVM the application can accurately calculate and display data simultaneously without obstructing the user's available information. The RSSI from the iBeacons is obtained in lower level Bluetooth frameworks that the phone provides through an API. What is received by the application is a value in dBm which is passed to the path loss model equation that determines an approximate distance.

The application opens in scan mode, where it's continuously scanning for BLE (Bluetooth Low Energy) devices. Using a list of known devices, those that were installed in the garage, it can rapidly filter and log the incoming RSSI values from the phone's API. While the application has three of many recently updated RSSI values it can run the algorithm routine which determines the user's location within the garage. The three-of-many terminology here describes that the devices needs up to date information from at least three of the many iBeacons that are installed in the garage. The phone applications database contains values of RSSI values and time stamps to be able to identify and use the most recent data.

Due to multipath fading, noise, and interference from other signals, wireless signals can be corrupted or distorted, causing the RSSI values received by the smart phone to "jump." This jump is due to a single iBeacon packets being received at different times and from different angles/locations. In order to deal with the fluctuations of power strength levels an averaging algorithm is required to smooth the data and obtain a more constant result.

Lastly is the collection of data from the iBeacons pertaining to telemetry. The team is only using the magnetometer telemetry from the devices which is contained within the ServiceData bytes of the advertising packet. This data is parsed using byte matching where the packet is identified and split into sections based on the specifications provided by Estimote for location data within the byte stream.

# 8 Results and Discussion

The following sections and subsections include the results and findings of the project. Data results include path-loss modeling, Cramér-Rao Lower Bound, localization algorithms and magnetometer findings. The path-loss model section determines a suitable path-loss model for the projects selected testing environment. The following section on Cramér-Rao Lower Bound includes the findings of how the localization algorithms fared in terms of accuracy. The final section goes over how the magnetometer was used for vehicle detection in a parking garage.

## 8.1 Path-Loss Models

The first step before creating a path loss model was determining an appropriate testing environment. The team chose three parking environments to test for the determination of path loss models. The Worcester Polytechnic Parking Garage and the Higgins House parking lots were selected along with the library parking lot.The Higgins House and Library lots depict outside parking environments as there are very few obstructions for radio waves, meanwhile the parking garage represents an inside parking environment

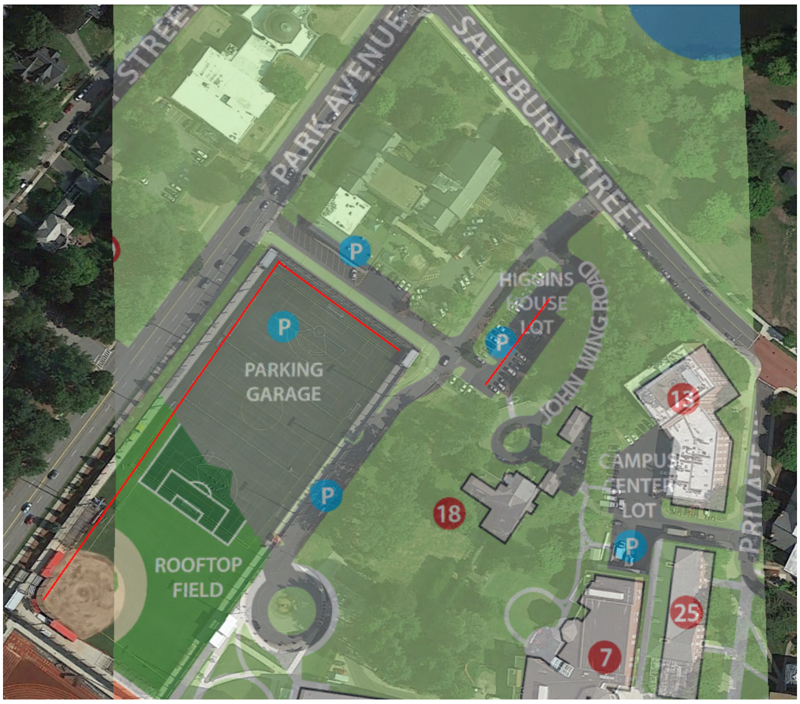


**Figure 12:** Parking Garage and Higgins House Lot

because there are concrete structures and other confounding variables that would affect the propagation of radio waves. Figure 12 shows the parking garage and Higgins House lots overlayed on Google Maps with approximated maximum measurement distances, while Figure 13 shows the same for the library lot.

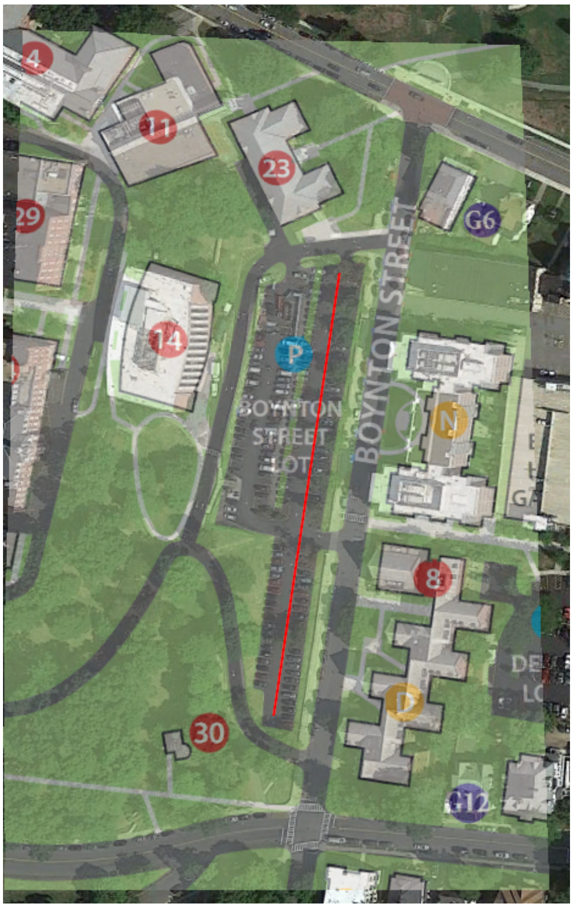


**Figure 13:** Library lot

The maximum distance measurements depicted by the red lines on each figure represent the maximum distance measurements achievable from each lot. For the Parking Garage the maximum distance is 183 meters, Higgins House Lot is 49 meters, and lastly the Boyton Street Lot is 176 meters. These parking lots were chosen for their space and difference in environments, the Parking Garage representing a congested space with obstructions, the Boynton Lot as a moving environment, and the Higgins House lot as a mostly static environment with very little car movement. The team researched standards on parking spaces and regulations, it was determined that there is no specific standard on spacing for parking lots but states do release guidelines on how far apart spacing should be for various vehicles. The state of Massachusetts parking bylaws outline:

> “For the purposes of this bylaw, minimum parking space width shall be measured perpendicular to the center line of the parking space. For standard cars the minimum parking space width shall be nine (9) feet.”

After measuring the WPI parking lots, all three were found to be within these specifications of a nine foot parking space, visualized in Figure 14. However not all parking lots having a standard parking space which will make localization metrics a little bit more difficult as there are no specific “bounds” that would determine if a vehicle was in a spot or not. This might be an argument for further use of the magnetometer which would distinguish the specific spots that were open. Using this info the team could take both approximate locations provided by the application and exact measures calculated by counting the spaces.

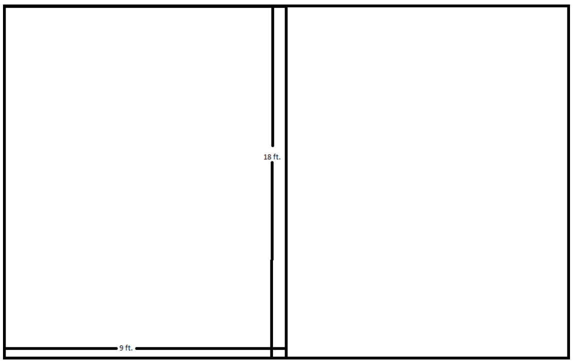


**Figure 14:** Standard parking space size, of eighteen feet by nine feet

Using the collected and calculated data the team came to a few conclusions about iBeacon$^{TM}$ accuracy as well as the applied path loss models from the generic Beacon Scanning and Estimote applications. The data was compared to actual distance vs application calculated difference which resulted in a perfect difference calculation. In general the virtual iBeacon broadcaster was more accurate with a percent difference of 28.71% for the garage and 24.25% difference for the outside parking lot. The Estimote iBeacon crystal had difference percentages of 33.28% and 27.01% for the garage and outside lots respectively. Based on the calculated power gradients the group came to the conclusion that the Beacon Scanning app was using a distance power gradient between 2 and 3 for its path loss model. The alphas calculated were 2.45, 2.77, 2.68, and 2.65

with an average alpha of 2.63. While the IEEE802.11 specification calls for an alpha of 2, the results collected could be the result of constant moving interference as well as a custom power gradient using in the library of the Beacon Scanning application which is developed by a third party team.

Comparing the outside lot and inside garage environments it is apparent that the garage introduces more multipath obscurities and has a direct effect on the accuracy of determining distance. The WPI parking garage has bellow like structures on the ceiling, including concrete poles and tight rows between parking rows. This garage is a good example of a challenging environment in which to determine distances based on RSS values. The Estimote iBeacon crystal had a different of 6.26% in percent error between the garage and lot while the virtual iBeacon had a difference of 4.46%. This shows that there's a consistent difference in accuracy between the two environments.

The first meter path loss was calculated using Estimote's specifications for transmitter gain, and the distance power gradient from the collected data can be averaged to around 2.5 (rounding down from 2.68 of the actual mean calculated). The various measurements between the different testing environments and beacons can be seen both tabulated and graphically in Tables 2-5, as well as in Figures 15-18. The tables include the RSSI values and distances used to determine the value of alpha and shadow fading, as well as the percent difference in the distance calculation. The graphs accompanying each table is a visualization of the path loss model for each case, relating the RSSI to the distance, in log-log scale.

<table>
<tr><th colspan="5">Parking Garage<br>Estimote iBeacon Crystal</th></tr>
<tr><th>RSS<br>(dBm)</th><th>Measured<br>Distance<br>(m)</th><th>Calculated<br>Distance<br>(m)</th><th>Actual<br>Distance<br>(ft)</th><th>Percent<br>Difference<br>(%)</th></tr>
<tr><td>-62</td><td>0.6</td><td>0.91</td><td>3</td><td>-34.40</td></tr>
<tr><td>-70</td><td>2.5</td><td>1.83</td><td>6</td><td>36.67</td></tr>
<tr><td>-73</td><td>3.69</td><td>2.74</td><td>9</td><td>34.48</td></tr>
<tr><td>-90</td><td>7.41</td><td>5.49</td><td>18</td><td>35.03</td></tr>
<tr><td>-84</td><td>10.05</td><td>8.23</td><td>27</td><td>22.09</td></tr>
<tr><td>-94</td><td>14.23</td><td>16.46</td><td>54</td><td>-13.02</td></tr>
<tr><td>-101</td><td>18.7</td><td>19.21</td><td>63</td><td>-2.64</td></tr>
<tr><td colspan="5">alpha = 2.45<br>Standard Deviation of Shadow Fading = 7.33</td></tr>
</table>

**Table 2:** Path loss measurement table for the Estimote Beacon in a parking garage environment.

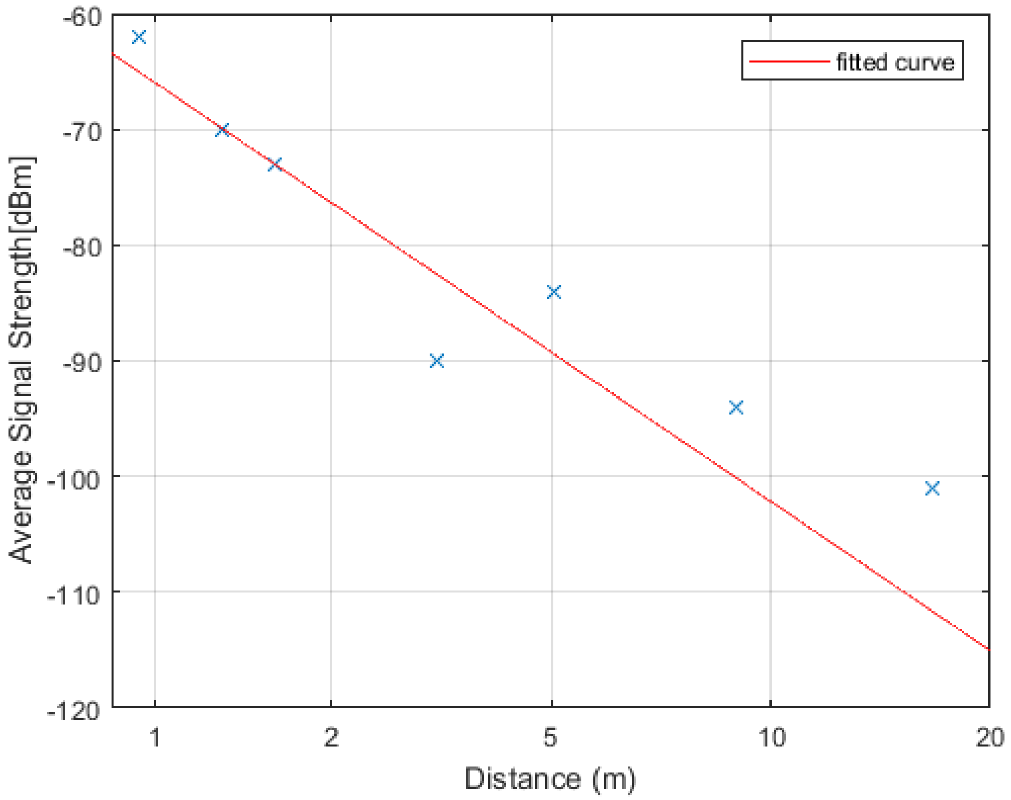


**Figure 15:** Path Loss Model for the Estimote Beacon in a parking garage environment.

| Parking Garage<br>Estimote Virtual Beacon | | | | |
|---|---|---|---|---|
| RSS (dBm) | Measured Distance (m) | Calculated Distance (m) | Actual Distance (ft) | Percent Difference (%) |
| -61 | 0.8 | 0.91 | 3 | -12.5 |
| -65 | 2.85 | 2.74 | 9 | 3.87 |
| -68 | 5.1 | 5.49 | 18 | -7.07 |
| -93 | 15 | 8.23 | 27 | 82.22 |
| -98 | 17.64 | 16.46 | 45 | 7.15 |
| -101 | 23.71 | 19.21 | 63 | 23.44 |
| alpha = 2.77<br>Standard Deviation of Shadow Fading = 7.39 | | | | |

**Table 3:** Path loss measurement table for the Estimote Virtual Beacon in a parking garage environment.

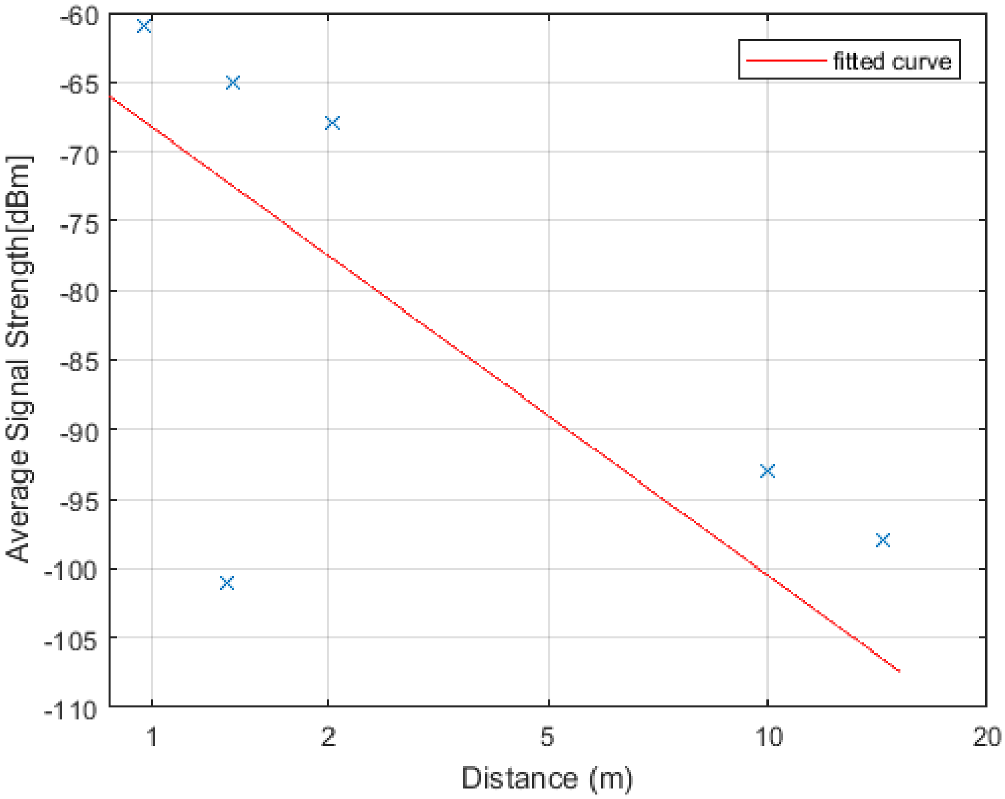


**Figure 16:** Path Loss Model for the Estimote Virtual Beacon in a parking garage environment.

| Parking Garage<br>Estimote Virtual Beacon | | | | |
|---|---|---|---|---|
| RSS (dBm) | Measured Distance (m) | Calculated Distance (m) | Actual Distance (ft) | Percent Difference (%) |
| -65 | 1.25 | 0.91 | 3 | 36.67 |
| -79 | 4.6 | 2.74 | 9 | 67.64 |
| -81 | 6.11 | 5.49 | 18 | 11.34 |
| -87 | 9.13 | 8.23 | 27 | 10.91 |
| -92 | 16.68 | 16.46 | 45 | 1.32 |
| -98 | 25.77 | 19.21 | 63 | 34.17 |
| alpha = 2.77<br>Standard Deviation of Shadow Fading = 7.39 | | | | |

**Table 4:** Path loss measurement table for the Estimote Beacon in an open parking lot environment.

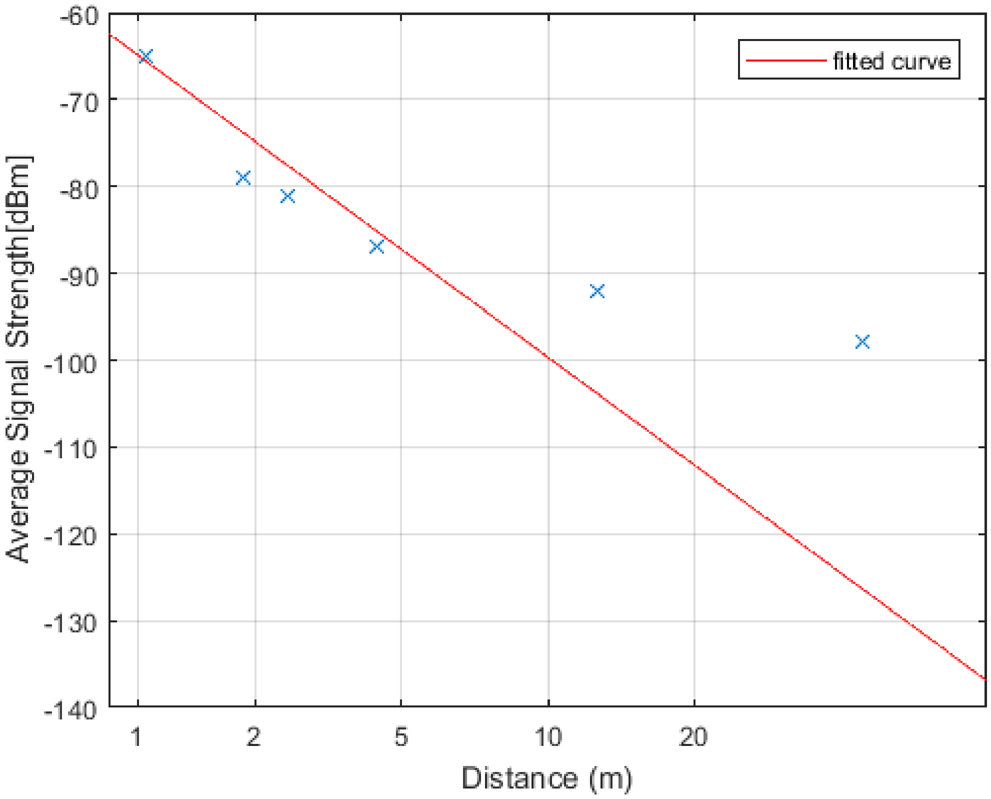


**Figure 17:** Path Loss Model for the Estimote Beacon in an open parking lot environment.

| Parking Garage<br>Estimote Virtual Beacon | | | | |
|---|---|---|---|---|
| RSS (dBm) | Measured Distance (m) | Calculated Distance (m) | Actual Distance (ft) | Percent Difference (%) |
| -63 | 0.85 | 0.91 | 3 | -7.07 |
| -69 | 3.09 | 2.74 | 9 | 16.61 |
| -83 | 5.99 | 5.49 | 18 | 9.15 |
| -93 | 15.44 | 8.23 | 27 | 87.57 |
| -97 | 19.3 | 16.46 | 45 | 17.23 |
| -99 | 24.2 | 19.21 | 63 | 25.99 |
| alpha = 2.77<br>Standard Deviation of Shadow Fading = 7.39 | | | | |

**Table 5:** Path Loss Model for the Estimote Virtual Beacon in an open parking lot environment.

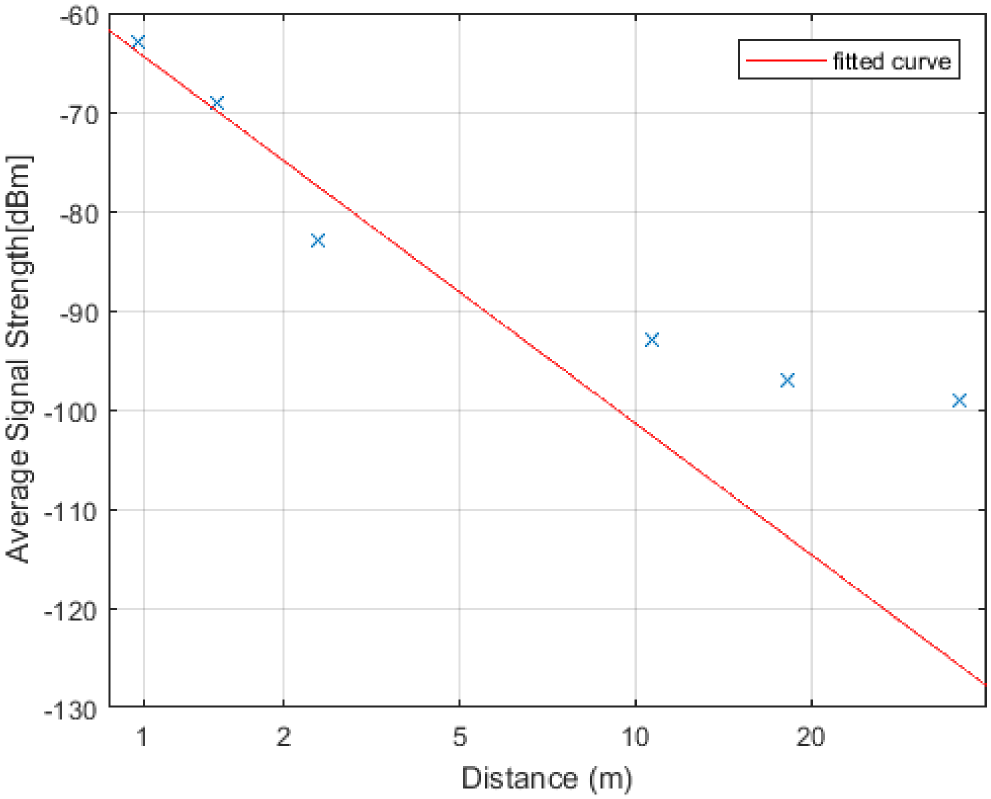


**Figure 18:** Path Loss Model for the Estimote Virtual Beacon in an open parking lot environment.

## 8.2 Cram$\acute{e}$r-Rao Lower Bound Approximations and Algorithm Implementations

### 8.2.1 Cram$\acute{e}$r-Rao Lower Bound Simulations

As briefly discussed previously, a crucial tool in comparing the accuracy of the localization algorithms was through comparing the ranges of accuracy with the contour of location error standard deviation created by a mathematical approximation called Cram$\acute{e}$r-Rao Lower Bound (CRLB). This approximation allows for a contour mapping of location error based on a set of predetermined access points. The purpose of these models was to determine the most effective locations for the iBeacon crystals, such that not only the entire space was mapped by the beacons, but that the localization accuracy had a low error rate.

The primary purpose of CRLB is determining the lowest value of variance of an estimator. This ranging approximation can be calculated using the inverse Fisher Information Matrix. The first step of this calculation is to recall the path-loss model equation:

$$Obs = P_r = P_0 - 10\alpha log(d) + X \tag{10}$$

Where “Obs” is the observed received signal strength reading for a single beacon, and equals the received power, Pr. The probability distribution of the observation Obs was taken given some distance d, resulting in:

$$f(Obs|d) = (\frac{1}{\sqrt{2\pi\sigma}})e^{\frac{(P_r - P_0 + 10\alpha log(d))^2}{2\sigma^2}} \tag{11}$$

The probability distribution function of the observation point was used to find to find the Fisher matrix, which is shown in Equation 12.

$$-E\left[\frac{\delta^2 ln(f(Obs|d))}{\delta d^2}\right] = E\left[\frac{\delta ln(f(Obs|d))}{\delta d}\right]^2 = \left[\frac{10\alpha}{ln(10)\sigma d}\right]^2 \tag{12}$$

As mentioned previously, CRLB is equivalent to the inverse of the Fisher equation, and is also equivalent to the variance estimation, 2. This is shown in Equation 13.

$$CRLB = \left[\frac{ln(10)\sigma d}{10\alpha}\right]^2 = \sigma^2 \tag{13}$$

The use of CRLB can be applied to localization in a two-dimensional space, based on the location of the reference points ($x_i$, $y_i$) and the access points (x,y). These points are used to estimate location based on the received power, dP, as seen in Equation 14.

$$dP_i = -\frac{10\alpha_i}{ln(10)\alpha}(\frac{x-x_i}{r_i^2}dx + \frac{y-y_i}{r_i^2}dy)\cdots i = 1,2...,N \tag{14}$$

The covariance of the location estimate was used due to the shadow fading being a zero Gaussian random variable as shown in Equation 15.

$$cov(dP_i, dP_j) = \begin{cases} \sigma^2, i = j \\ 0, i \neq j \end{cases} \cdots i,j = 1,2...,N \tag{15}$$

Which yields:

$$\sigma_r = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{16}$$

Utilizing these mathematics, a MATLAB script was created to draw a heat-map of the distance error approximations. The MATLAB code for these scripts can be found in Appendix B. These models were created using a thirty meter by thirty meter parking lot design, where the test dimensions were selected to incorporate a total of twenty parking spaces, each eighteen feet by nine feet, as well as a strip of road for traffic to drive down in between the two sets of spaces. The models also used a shadow fading value of three, as that value corresponds to the shadow fading value calculated from the original path loss models. The first simulation was creating with the beacons placed directly in the center of each parking space, creating a network of twenty access points. The heat-map of this architecture can be seen below in Figure 19, with the highest accuracy of half a meter occurring close to the access points, and the lowest accuracy of 3 meters occurring in the driving lane.

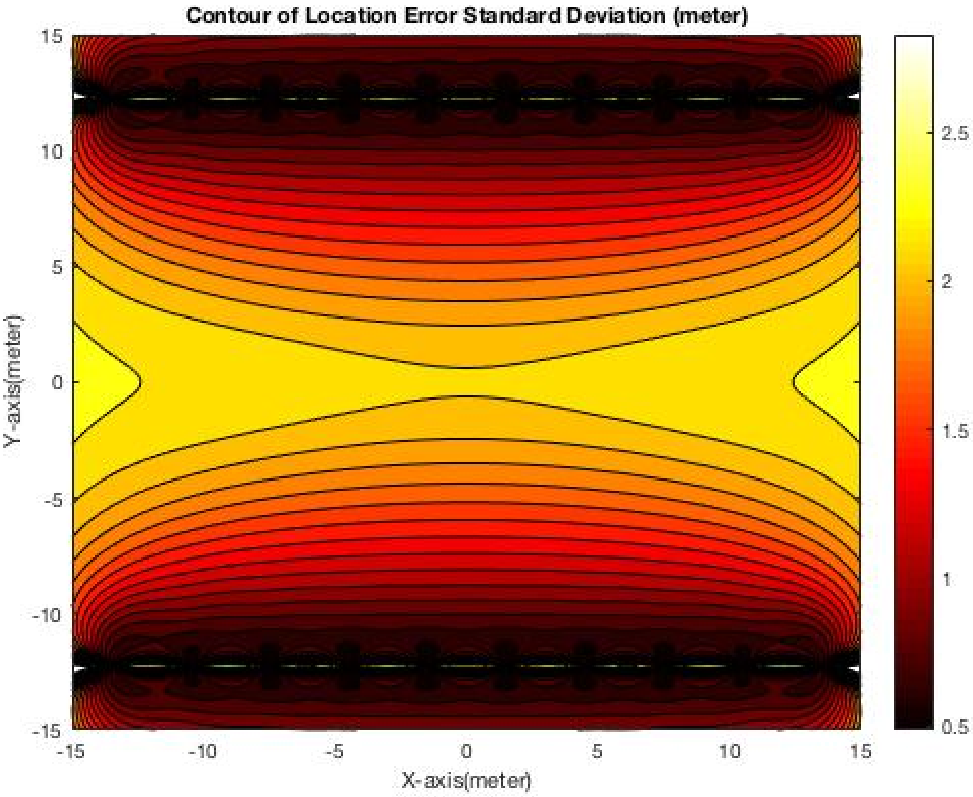


**Figure 19:** CRLB heat-map for every space architecture. In this configuration, a beacon is placed in the center of each parking space. Note that the scale of accuracy ranges from 0.5 meters to 3 meters, with a majority of the error in the 1-2.5 meter error range.

Another possible beacon architecture was explored to examine the effect of using fewer beacons in an attempt to reduce potential purchasing costs. This implementation, shown in Figure 20, deploys a beacon in every other space. When compared to the first implementation shown in Figure 19, it is evident that this deployment suffers a severe drop in accuracy, with the a majority of the accuracy occurring in the range of 2-3.5 meters, even approaching closer to the beacons. This accuracy is unacceptable, as the previous implementation had a maximum error of 3 meters, which only occurred at the extremities of the layout: between the beacons (due to the pseudo-omnidirectional nature of the antenna on the iBeacon$^{TM}$ devices, the beacons only transmit in hemispheres, and do not transmit parallel to the other beacons).

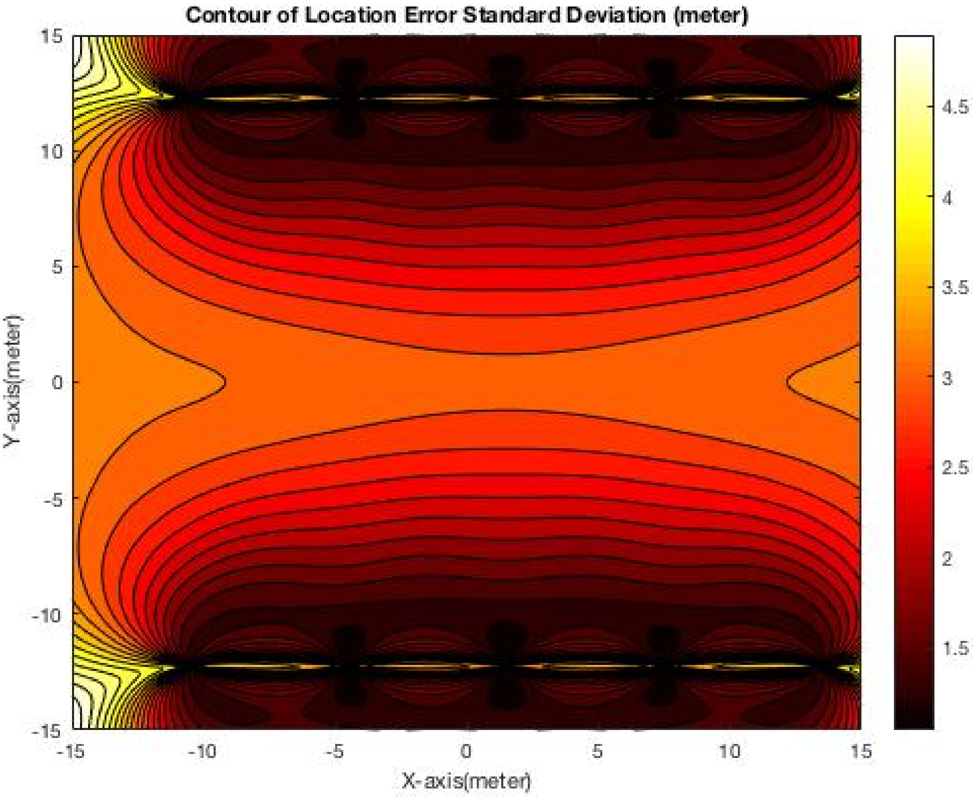


**Figure 20:** CRLB heat-map for every other space architecture. In this configuration, a beacon is placed in the center of every other parking space. Note that the scale of accuracy ranges from 0.5 meters to 5 meters, with a majority of the error in the 2-3.5 meter error range.

The final beacon configuration examined was an implementation that took into account the use of the iBeacon$^{TM}$ devices as magnetometers as well. In the configuration shown in Figure 21, the devices were placed at the back of every spot, as that location is ideal for the magnetometer to sense if a vehicle is present, as discussed in the Magnetometer Results and Discussion section. Similar to the first configuration in Figure 19, the accuracy of this model ranges from 0.5 meters to 3 meters, and is primarily in the range of 1-2.5 meters, which is acceptable for this application, as a vehicle is typically at least 2 meters long and wide.

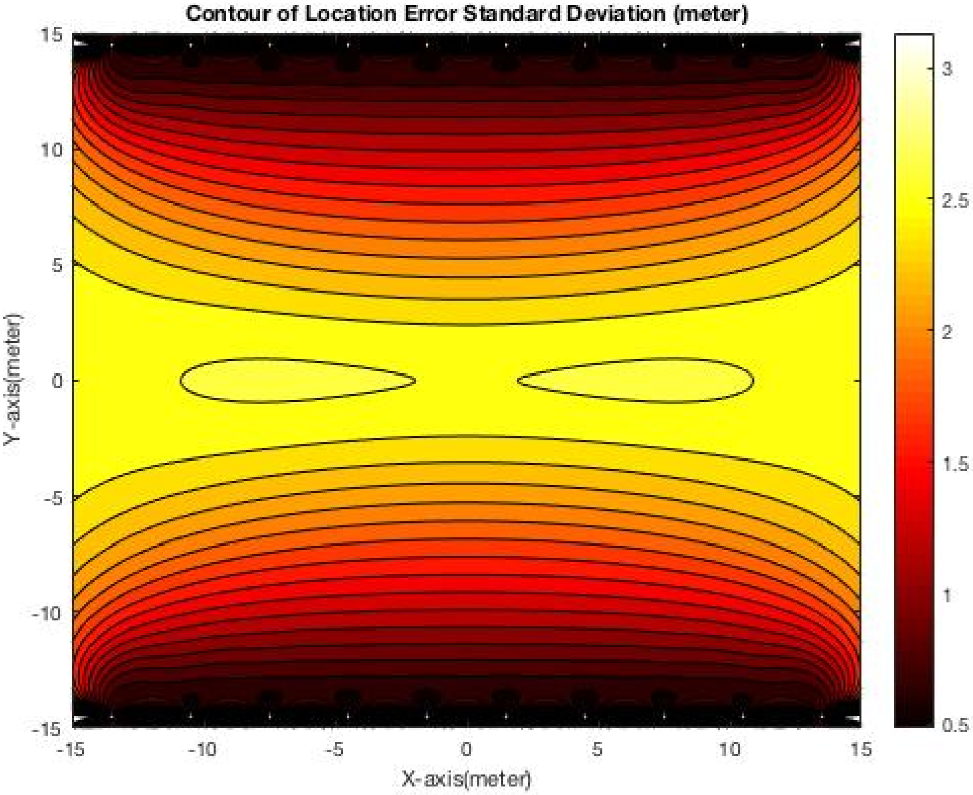


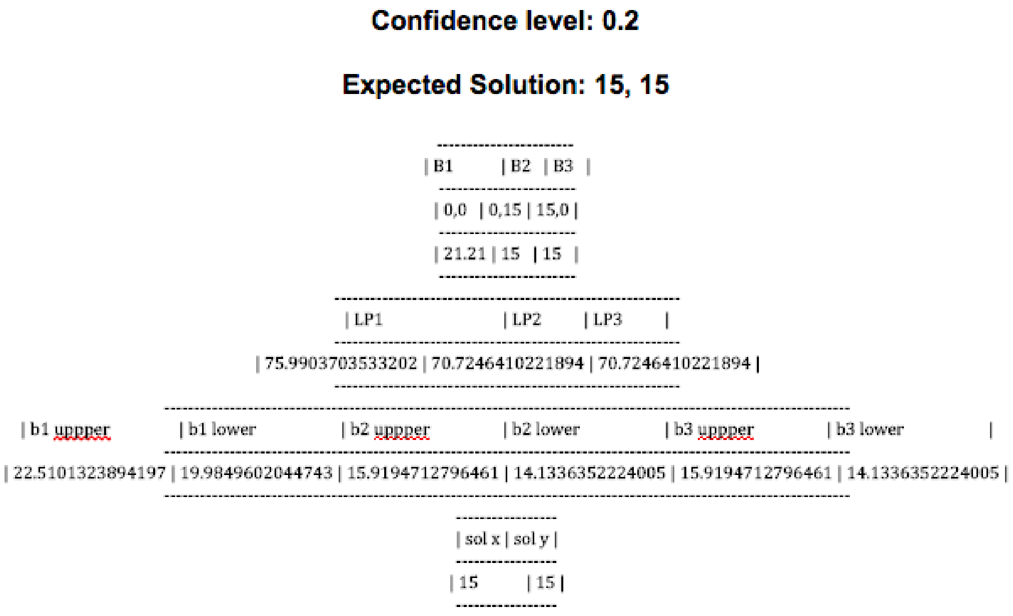
**Figure 21:** CRLB heat-map for every other space, at the back of each space architecture. In this configuration, a beacon is placed at the back of every parking space, on the wall in front of the space. Note that the scale of accuracy ranges from 0.5 meters to 3 meters, with a majority of the error in the 1-2.5 meter error range.

### 8.2.2 Localization Algorithms Implementations

The two algorithms that have been implemented as of thus far are least means squared and maximum likelihood. During B term, the team was provided with basic implementations of both algorithms in MATLAB code. The bulk of implementation was spent re-coding and analyzing the algorithms into something that could be functional on a mobile device. Appendix C demonstrates the Least Means Squared algorithm and the maximum Likelihood algorithm. Both implementations have worked successfully using randomized values provided to it; below is a set of test cases based on problem 15.2 from Professor Kaveh Pahlavan's book Principles of Wireless Access and Localization. The test case parameters are listed in the tables below, while the solution from the groups implementation of Maximum Likelihood is listed below each table.

| Access Point Coordinates (m) | Access Point Distance to Reference (m) |
|---|---|
| (0,0) | 15 |
| (0,15) | 16 |
| (-5,5) | 5 |

**Table 6:** Localization Test Case 1

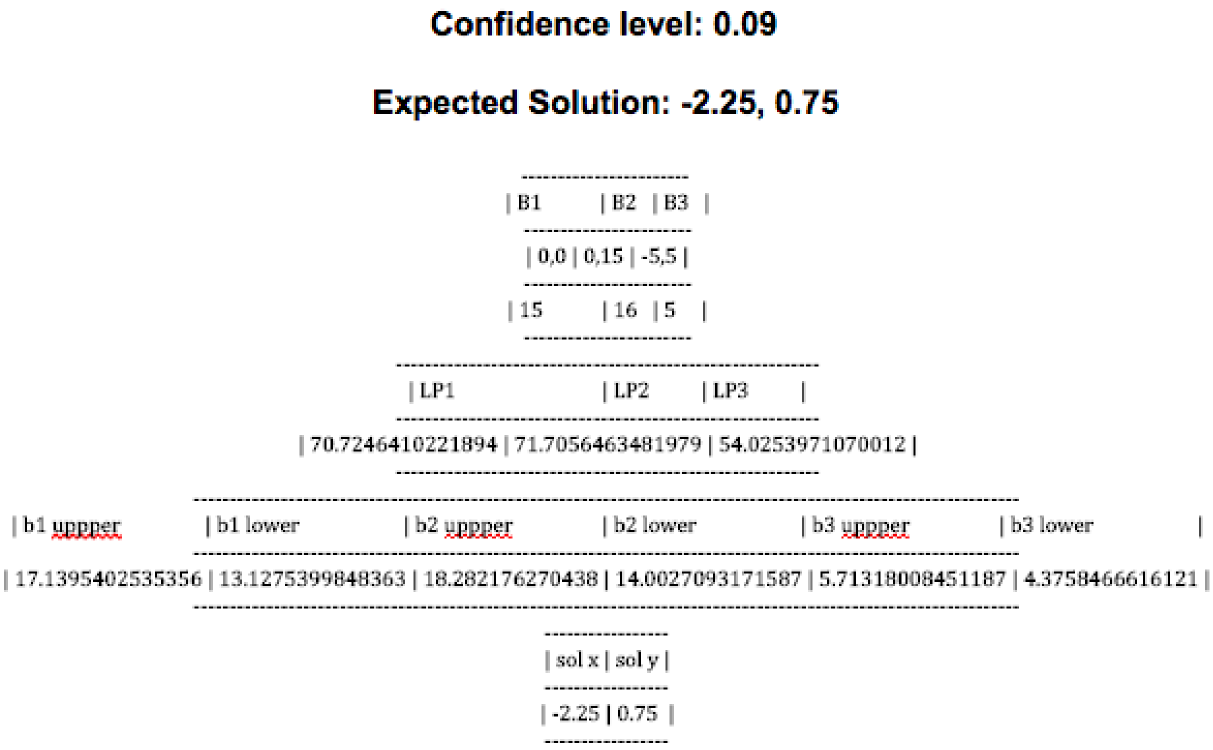


```
Confidence level: 0.2

Expected Solution: 15, 15

----------------------
| B1        | B2  | B3  |
----------------------
| 0,0  | 0,15 | 15,0 |
----------------------
| 21.21 | 15  | 15  |
----------------------

-----------------------------------------------------
| LP1                | LP2       | LP3        |
-----------------------------------------------------
| 75.9903703533202 | 70.7246410221894 | 70.7246410221894 |
-----------------------------------------------------

----------------------------------------------------------------------------------------------------
| b1 uppper        | b1 lower            | b2 uppper           | b2 lower            | b3 uppper           | b3 lower            |
----------------------------------------------------------------------------------------------------
| 22.5101323894197 | 19.9849602044743 | 15.9194712796461 | 14.1336352224005 | 15.9194712796461 | 14.1336352224005 |
----------------------------------------------------------------------------------------------------

----------------
| sol x | sol y |
----------------
| 15      | 15 |
----------------
```

**Figure 22:** Localization Test Case 1 Results

| Access Point Coordinates (m) | Access Point Distance to Reference (m) |
|---|---|
| (0,0) | 21.21 |
| (0,15) | 15 |
| (15,0) | 15 |

**Table 7:** Localization Test Case 2

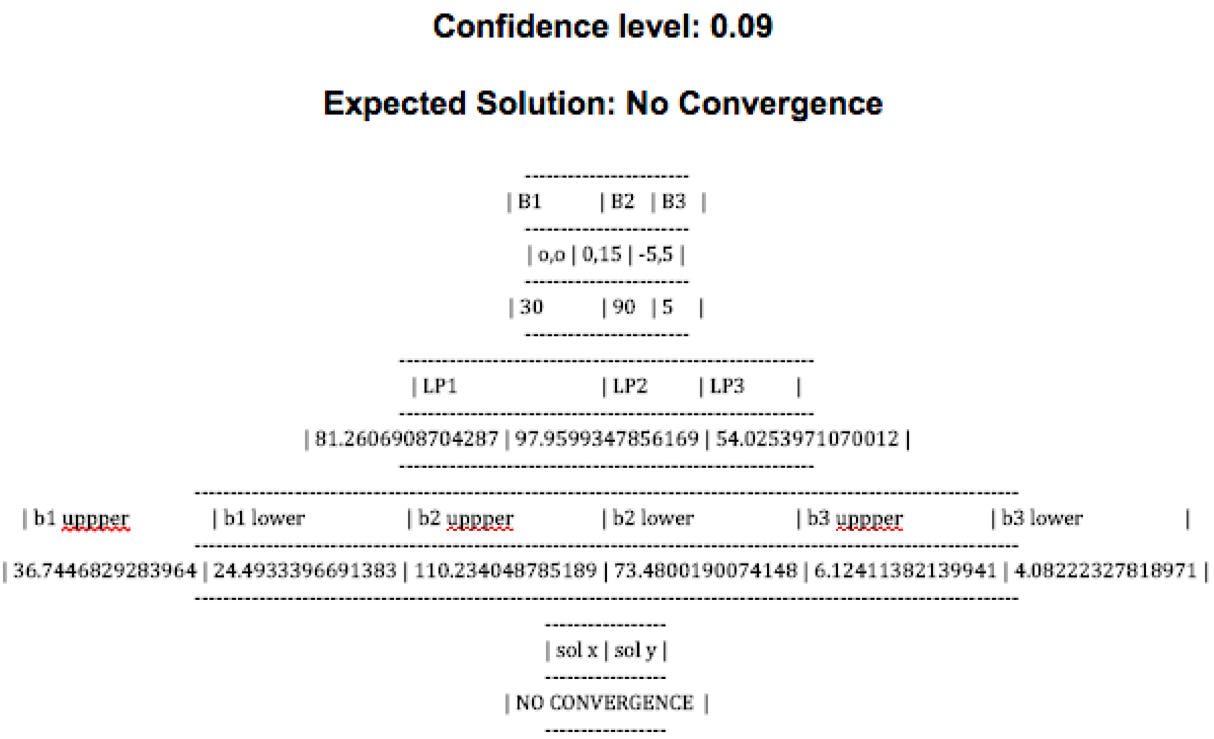


```
                          Confidence level: 0.09

                       Expected Solution: -2.25, 0.75

                            ----------------------
                           | B1      | B2  | B3  |
                            ----------------------
                            | 0,0 | 0,15 | -5,5 |
                            ----------------------
                           | 15      | 16  | 5   |
                            ----------------------
                 ---------------------------------------------------------
                 | LP1                  | LP2        | LP3        |
                 ---------------------------------------------------------
             | 70.7246410221894 | 71.7056463481979 | 54.0253971070012 |
                 ---------------------------------------------------------
         ------------------------------------------------------------------------------------------------------------
| b1 uppper       | b1 lower         | b2 uppper        | b2 lower         | b3 uppper        | b3 lower        |
         ------------------------------------------------------------------------------------------------------------
| 17.1395402535356 | 13.1275399848363 | 18.282176270438 | 14.0027093171587 | 5.71318008451187 | 4.3758466616121 |
         ------------------------------------------------------------------------------------------------------------
                              ----------------
                              | sol x | sol y |
                              ----------------
                              | -2.25 | 0.75  |
                              ----------------
```

**Figure 23:** Localization Test Case 2 Results

| Access Point Coordinates (m) | Access Point Distance to Reference (m) |
|---|---|
| (0,0) | 30 |
| (0,15) | 90 |
| (-5,5) | 5 |

**Table 8:** Localization Test Case 3



**Confidence level: 0.09**

**Expected Solution: No Convergence**

```
                        ----------------------
                        | B1       | B2  | B3  |
                        ----------------------
                        | o,o | 0,15 | -5,5 |
                        ----------------------
                        | 30       | 90  | 5   |
                        ----------------------

          ------------------------------------------------------
           | LP1                  | LP2        | LP3        |
          ------------------------------------------------------
       | 81.2606908704287 | 97.9599347856169 | 54.0253971070012 |
          ------------------------------------------------------

        -------------------------------------------------------------------------------------------------------------
| b1 uppper        | b1 lower          | b2 uppper         | b2 lower          | b3 uppper         | b3 lower          |
        -------------------------------------------------------------------------------------------------------------
| 36.7446829283964 | 24.4933396691383 | 110.234048785189 | 73.4800190074148 | 6.12411382139941 | 4.08222327818971 |
        -------------------------------------------------------------------------------------------------------------

                        ----------------
                        | sol x | sol y |
                        ----------------
                      | NO CONVERGENCE  |
                        ----------------
```

**Figure 24:** Localization Test Case 3 Results

Doing localization in real-time can be inaccurate due to the sporadic signal strength values (RSSI) received from the iBeacon devices. RSSI values can range because of multi path fading and the general radiation of RF signals, and thus must be "smoothed" to be used for accurate calculations. The team came up with an averaging slide algorithm that would help keep consistent RSSI values for the use of calculating distance from path loss. The most recent 10 values of RSSI are kept in an array; when a new value is received all previous values are shifted right by one place and the oldest value is dropped. The newest value is placed at the front of the array and the average of the 10 values it taken and provided to the path loss model for further calculation.

Putting the two pieces together achieved an accurate and fast result. Doing time lapse testing on the algorithm, averaging, and scanning resulted in the following table of benchmarks. Table 9 demonstrates that calculating the algorithm in real-time on a user's device is both possible and fast. It displays concurrent iterations of simultaneous localization calculations, and the timer per iteration until complete.

| Concurrent Iterations | Time per Iteration (ms) |
|---|---|
| 1 | 10 |
| 10 | 13 |
| 100 | 38 |
| 1000 | 67 |

**Table 9:** Concurrent iterations of simultaneous localization calculations, and the timer per iteration until complete.

### 8.2.3 Comparison between Algorithms and CRLB simulations

The team decided to do data collection with two different iBeacon devices from two different manufacturers. Using two different beacons would give a larger dataset based on variables that were not considered during the project such as antenna radiation pattern, spread, and actual vs claim device transmit power. The first type of beacon used was Estimote's which had configurable transmit settings. The packets from the Estimote beacons were set to transmit packets at a 100ms interval and at a power level of 4dBm ( 2.5mW). The second contender was Varatio's beacon which likewise had configurable settings and transmit intervals. The settings for this test were set to 100ms transmit time with an 8 dBm ( 6.2mW) transmit power. The different power level would help to determine if the transmitted power had an effect on the accuracy of the algorithms. Below are two figures from the test compared to the respective beacon's CRLB data. Figure 25 is Estimote's location beacon, while Figure 26 was Varatio's beacon. Both tests evaluated the Maximum likelihood and Least Means Squares algorithms with the sliding averaging smoothing technique.

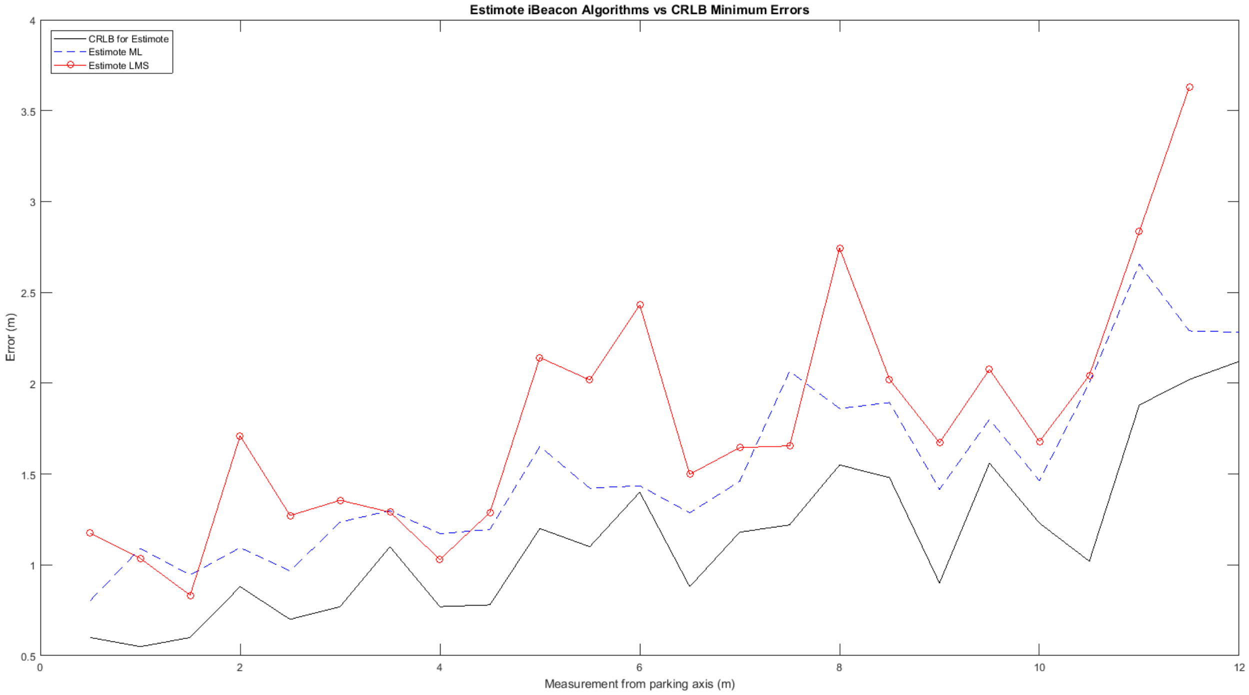


**Figure 25:** Comparison between the CRLB, ML, and LMS for Estimote beacon. Note that the ML more accurately maps to the CRLB simulation than the LMS.

To support the data collection a root mean square deviation was built to compare the variance between the algorithms and performance over different sample spaces of iterations. Figure 27 shows the Estimote location beacon's RMSE sample points vs error.

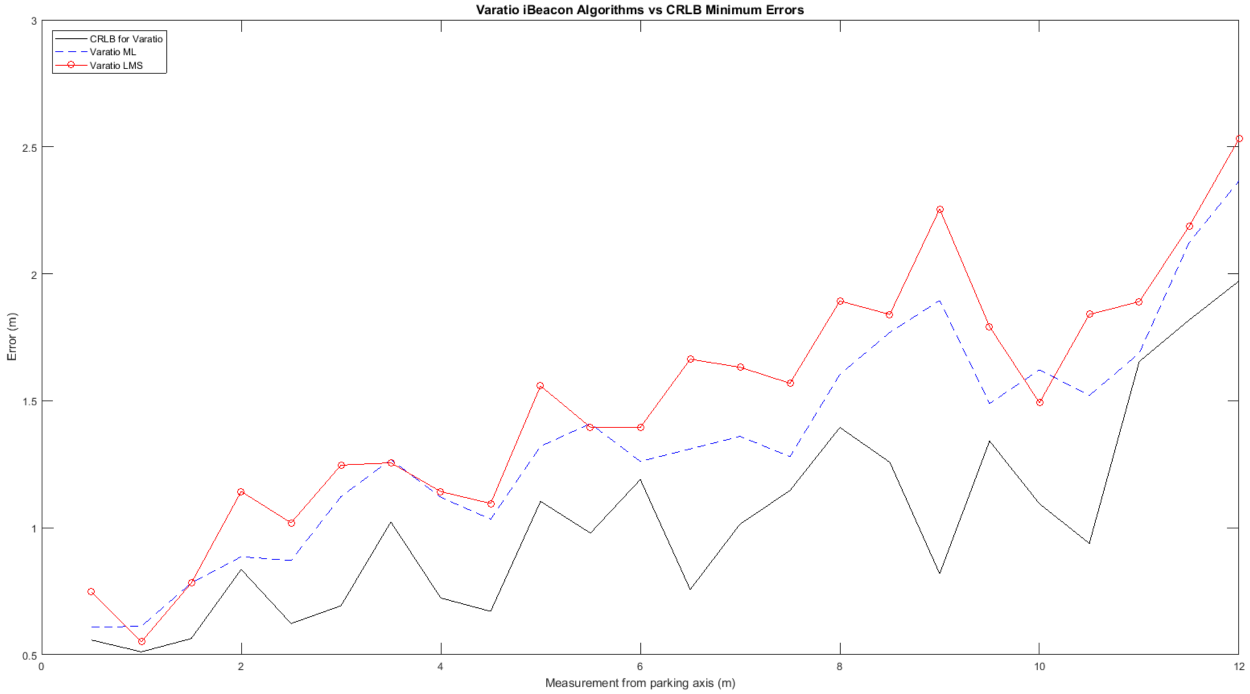


**Figure 26:** Comparison between the CRLB, ML, and LMS for Varatio beacon. Note that the accuracy of the exact same algorithms was higher on the higher quality beacons.

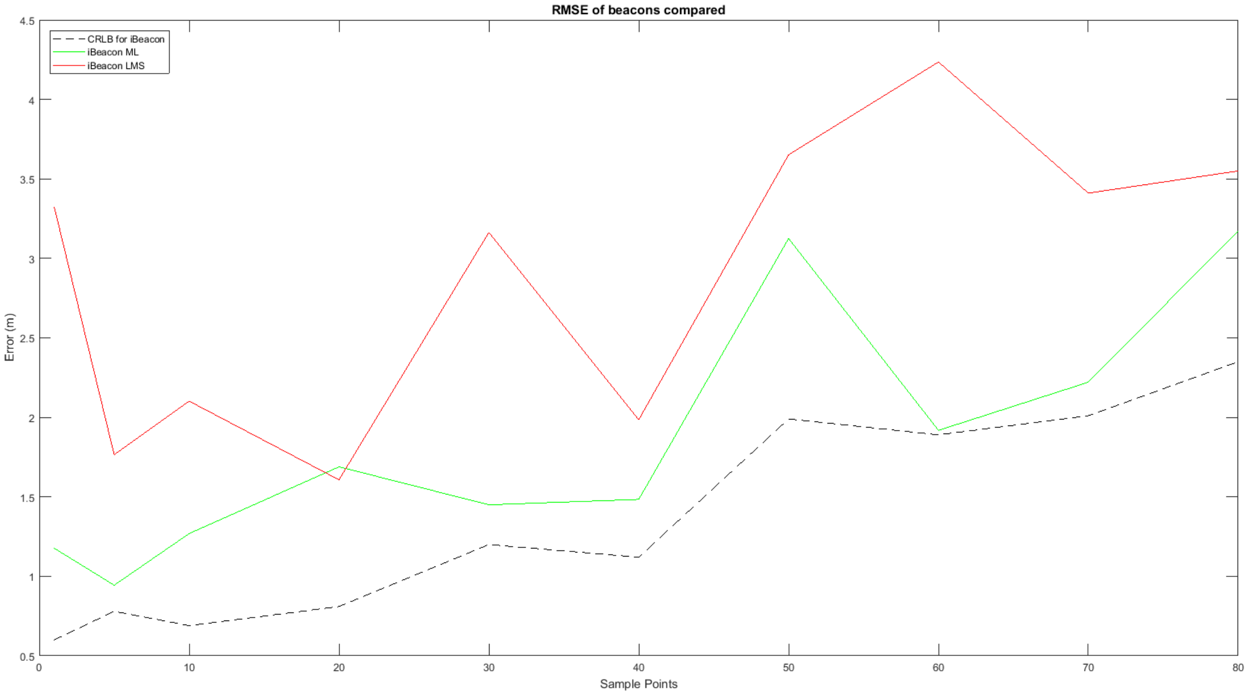


**Figure 27:** Estimote beacon RSME sample points vs. error.

## 8.3 Magnetometer Readings and Observations

The goal of implementing a magnetometer in the project was to be able to use it for vehicle detection. Vehicle detection was accomplished through the built in magnetometer of the estimote iBeacon. The built in magnetometer can take readings of the earth's magnetic field at a deployed location after determining a baseline for the set location. Once a baseline for the earth's magnetic field at a set location is determined the data collection can begin. Collecting data involves driving a car in the range of the sensor and collecting readings on the X,Y, and Z axis. The magnetic disturbances caused by the vehicle can be used to classify different types of vehicles like cars, trucks, vans, buses and more.

Testing for the project began with an average 4 door sedan in an area with a determined baseline for the earth's magnetic field being around -7.5$\mu$T (micro-teslas). With the car the driving past the sensor a dip can be seen down to -30$\mu$T with the reading returning back to the baseline after the car has past. From this data it can be extrapolated that a change of more than 20$\mu$T can signal that a car has entered the range of the magnetometer. For larger vehicles such as trucks, buses and vans a larger change can be expected in the earth's magnetic field because of the larger amount of ferrous metal. The graph seen below demonstrates the collection of data from the magnetometer. The graph below in Figure 28 only shows one instant of testing with a single vehicle to further explore how vehicles interact with the earth's magnetic field more types of vehicles can be tested such as vans, trucks and buses. These different vehicles would all have different magnetic signatures and distort the magnetic field differently leading to a different graph for every type of different vehicle.

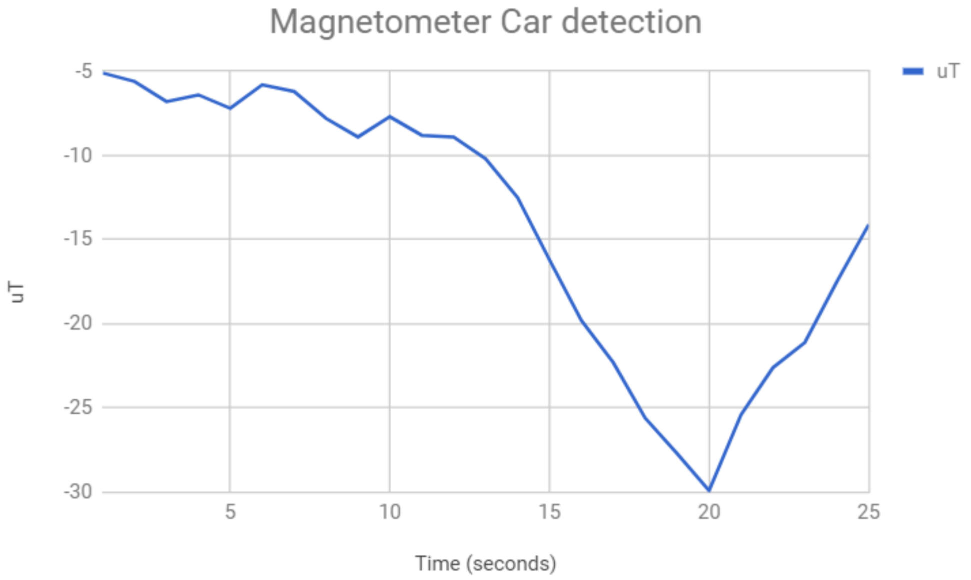


**Figure 28:** Magnetometer magnet field detection. The large increase in magnetic field is caused by approaching a large metallic object, such as a vehicle.

Figure 29 demonstrates the testing done with the addition of tracking the vehicle's direction and how that affects the magnetic field. It demonstrates that when reversing the direction the car moves past the sensor it also reverses the magnetic field in the chart. This can be helpful in identifying cars by their magnetic signature.

To conclude, testing for magnetometers found that fluctuations more than 20$\mu$T in the earth's magnetic field signaled that a car had entered the range of the magnetometer. To further test and validate this data more testing will need to be done with a wide variety of vehicles including but not limited to buses, trucks, vans and electric cars such as Tesla's.

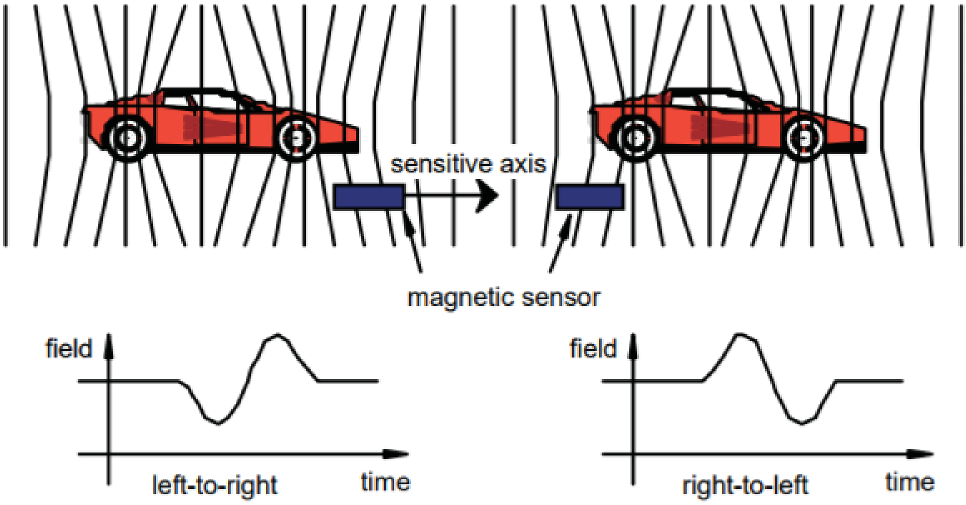


**Figure 29:** Observing magnetic fields in opposite directions.

# 9 PROPOSED ARCHITECTURE

The project architecture will consist of three separate topologies that will supplement each other in order to achieve the project objectives. Trilateration is the first architecture that will be implemented and mainly deals with the ticketing portion of the project; The second topology is made up of magnetometer capable iBeacon$^{TM}$ devices which will handle the function of determining open spaces. Lastly a phone application will be the result of the end user experience and will tie together the information collected from trilateration and the magnetometer sensors. Users will have a phone application that receives data from the iBeacon$^{TM}$ devices and will transmit the data to a web application where it will track the user as it enters and exits the parking area as well as the open spots. When a user of the application is looking for a spot, the magnetometer sensor system will be able to determine the spots already taken and relay that information back to the user.

The complete system architecture will consist of the following:

1. End user phone application that collects/displays information and broadcasts collected data to a web server

2. Magnetometers in parking spots

3. Processing Server that collects the information and hosts Web Application

# 10 Conclusion and Future Recommendations

The goal of this project was to create a system that could not only detect cars in parking spots but also direct a user via a smart phone application to any open spots in a crowded parking garage. The overarching architecture of this project was divided into several interconnected, interwoven sections, each of which were dependent on the previous section. The first section focused on analyzing the wireless capabilities of the Estimote iBeacon$^{TM}$ devices by creating various path-loss models for two different testing environments, an open parking lot and a roofed parking garage. The results were compared with ideal path-loss models created by the use of a virtual beacon from the Estimote smart phone application. The second section was creating a layout for the iBeacon$^{TM}$ devices to not only cover the entire test space, but to receive an an acceptable accuracy of approximately a meter. Several different layouts were examined using the Cram$\acute{e}$r-Rao Lower Bound, which generated heat maps to distinguish which regions had the highest ideal accuracy, as well as a value of that accuracy. Once a configuration had been agreed on, the implementation and testing of the localization algorithms began, resulting in the implementation of the Least Mean Squared and Maximum Likelihood algorithms. These algorithms were then compared with the ideal cases presented by the Cram$\acute{e}$r-Rao Lower Bound simulations, and adjusted based on their precision. The final crucial component of this project was the addition of a method to detect the vacancy of each parking space through the use of an on-board magnetometer in each Estimote iBeacon$^{TM}$ device. Overall, this project was a successful proof of concept for the application of parking space monitoring, as the algorithms performed as expected, and where relatively accurate when compared with the Cram$\acute{e}$r-Rao Lower Bound simulations.

To further this project in the future more steps could be taken with the research of magnetometers and application development. For example, more vehicles of a wider variety could be tested to see the effect different vehicle types has on the magnetic field. An interesting investigation would be to explore the variation between a small motorcycle compared to a large SUV, and determine a proper magnetometer threshold based on the received readings. In terms of the application development, a full fledged application could be developed with an intuitive user interface to streamline the process of finding open spots in a parking garage, including the construction of a basic navigation feature to direct a user how to reach a parking space. Unfortunately, this high level of an application may be outside of the scope of this project, as well as outside the possibilities of the Estimote iBeacon$^{TM}$ devices. One final steps could also include measuring the radiation patterns from the antennas to understand what orientation best fits these beacons.

A key point of advice for any project groups that may follow this group is to invest is higher quality devices that the Estimote iBeacon$^{TM}$ devices. The team initially struggled with the Estimote devices, as their low cost is a byproduct of their low accuracy and overall usefulness. A major part of localization using iBeacon devices is selecting the right hardware. Also, iBeacon devices with a stronger transmit power level perform consistently better and likewise with transmit interval where a faster transmit interval far out-performed the slower intervals.

Additional MQP References can be found in: [18][19]

# REFERENCES

[1] J. Tillison, “Bluetooth Low Energy (BLE) – A Short History of the BLE standard and GATT,” *eeNews Embedded*, 2016.

[2] M. Siekkinen, M. Hiienkari, and J. Nurminen, “How Low Energy is Bluetooth Low Energy?” *2012 IEEE Wireless Communications and Networking Conference Workshops (WCNCW) IEEE*, pp. pp.232–237, 2012.

[3] H. Liu, H. Darabi, P. Banerjee, and J. Liu, “Survey of Wireless Indoor Positioning Techniques and Systems,” *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. vol. 37, no. 6, p. pp. 1067–1080, 2007.

[4] Argenox.com, “Introduction To Bluetooth Low Energy (BLE) V4.0.”

[5] A. Inc., “Getting Started with iBeacon,” vol. 1, 2017.

[6] L. Matsuk, “Technical specification of Estimote Beacons and Stickers.” *Estimote Community Portal.*

[7] J. Yim, “A Review of Indoor Positioning Techniques ,” *Advanced Science and Technology Letter*, vol. Vol. 112(Architecture and Civil Engineering 2015), pp. 46–49, 2015.

[8] W. Murphy and W. Hereman, “Determination of a Position in Three Dimensions Using Trilateration and Approximate Distances ,” 1999.

[9] Q. Alhumoud, C. Connor, and D. Goodrich, “Using iBeacon for Navigation and Proximity Awareness in Smart Buildings,” *Worcester Polytechnic Institute.*

[10] K. Pahlavan and P. Krishnamurthy, “Principles of Wireless Networks,” 2002.

[11] M. Shchekotov, “Indoor localization methods based on Wi-Fi lateration and signal strength data collection,” *2015 17th Conference of Open Innovations Association (FRUCT)*, 2015.

[12] Massport.com, “About Massport. ,” 2017.

[13] W. W.Yan-ling, W.Xin and Z. Ming-chun, “ Current Situation and Analysis of Parking Problem in Beijing.” vol. 1st edition, 2016.

[14] A. T. Solutions., “ParkTrak Parking Counting System | All Traffic Solutions. ,” 2017.

[15] C. A. Miami Beach Commission, “ Miami Beach Commission,” *2012*.

[16] OmniPark, “OmniPark - About.” 2017.

[17] J. Xiong, Q. Qin, and K. Zeng, “A Distance Measurement Wireless Localization Correction Algorithm Based on RSSI,” *2014 Seventh International Symposium on Computational Intelligence and Design*.

[18] A. V. Savannah Redetzke and R. Otieno, “Smart room attendance monitoring and location tracking with ibeacon technology.”

[19] C. C. Qusai Alhumoud and D. Goodrich, “Using ibeacon for navigation and proximity awareness in smart buildings.”

# Appendix A: MATLAB Code for Path-Loss Modeling

```
clc;clear all;close all;

d=[0.85;3.09;5.99;15.44;19.3;24.2];
Pr=[-63;-69;-83;-93;-97;-99];

d_dB=10*log10(d);
F1=fit(d_dB,Pr,'poly1');

plot(F1, d,Pr, 'x' );
grid on
xlabel('Distance (m)');
ylabel('Received Signal Strength[dBm]');

disp(F1);
disp('Mean value of shadow fading is:');
disp(mean(Pr+20+2*d_dB));
disp('Standard Deviation of shadow fading is:');
disp(std(Pr+40+2*d_dB));
```

# APPENDIX B: MATLAB CODE FOR CRAM*é*R-RAO LOWER BOUND SIMULATIONS

```
MATLAB CODE FOR CRLB SIMULATIONS WITH BEACON IN EVERY
   SPACE
clc; clear all; close all;
APx(1) = 1.5; APy(1) = 12.25; APx(2) = 4.5; APy(2) =
   12.25; APx(3) = 7.5; APy(3) = 12.25;
APx(4) = 10.5; APy(4) = 12.25; APx(5) = 13.5; APy(5) =
   12.25;
APx(6) = -1.5; APy(6) = 12.25; APx(7) = -4.5; APy(7) =
   12.25; APx(8) = -7.5; APy(8) = 12.25;
APx(9) = -10.5; APy(9) = 12.25; APx(10) = -13.5; APy(10)
   = 12.25;
APx(11) = 1.5; APy(11) = -12.25; APx(12) = 4.5; APy(12) =
    -12.25; APx(13) = 7.5; APy(13) = -12.25;
APx(14) = 10.5; APy(14) = -12.25; APx(15) = 13.5; APy(15)
    = -12.25;
APx(16) = -1.5; APy(16) = -12.25; APx(17) = -4.5; APy(17)
    = -12.25; APx(18) = -7.5; APy(18) = -12.25;
APx(19) = -10.5; APy(19) = -12.25; APx(20) = -13.5; APy
   (20) = -12.25;
SD = 3;
NUM = 20;
mx = -15:0.1:15;
my = -15:0.1:15;
nxy = length(mx);
for yi = 1:nxy
for xi = 1:nxy

for i1 = 1:NUM
alpha(i1) = 2.5;
r(i1,xi,yi) = sqrt((mx(xi)-APx(i1))^2+(my(yi)- APy(i1))
    ^2);
H1(i1,xi,yi) = -10*alpha(i1)/(log(10))*(mx(xi)- APx(i1))/
    r(i1,xi,yi)^2;
H2(i1,xi,yi) = -10*alpha(i1)/(log(10))*(my(yi)- APy(i1))/
    r(i1,xi,yi)^2;
end
H(:,:,xi,yi) = [H1(:,xi,yi) H2(:,xi,yi)];
Covv(:,:,xi,yi) = SD^2*((H(:,:,xi,yi)'*H(:,:,xi,yi))^(-1)
    );
SDr(xi,yi) = sqrt(Covv(1,1,xi,yi)+Covv(2,2,xi,yi));
end
end
SDr = SDr';
contourf(mx,my,SDr,20);
colormap(hot);
colorbar;
xlabel('X-axis(meter)');
ylabel('Y-axis(meter)');
title('Contour of Location Error Standard Deviation (
    meter)');
```

```
MATLAB CODE FOR CRLB SIMULATIONS WITH BEACON IN EVERY
   OTHER SPACE
clc; clear all; close all;
APx(1) = 1.5; APy(1) = 12.25; APx(2) = 7.5; APy(2) =
   12.25;
APx(3) = 13.5; APy(3) = 12.25; APx(4) = -4.5; APy(4) =
   12.25;
APx(5) = -10.5; APy(5) = 12.25; APx(6) = 1.5; APy(6) =
   -12.25;
APx(7) = 7.5; APy(7) = -12.25; APx(8) = 13.5; APy(8) =
   -12.25;
APx(9) = -4.5; APy(9) = -12.25; APx(10) = -10.5; APy(10)
   = -12.25;
SD = 3;
NUM = 10;
mx = -15:0.1:15;
my = -15:0.1:15;
nxy = length(mx);
for yi = 1:nxy
for xi = 1:nxy

for i1 = 1:NUM
alpha(i1) = 2.5;
r(i1,xi,yi) = sqrt((mx(xi)-APx(i1))^2+(my(yi)- APy(i1))
   ^2);
H1(i1,xi,yi) = -10*alpha(i1)/(log(10))*(mx(xi)- APx(i1))/
   r(i1,xi,yi)^2;
H2(i1,xi,yi) = -10*alpha(i1)/(log(10))*(my(yi)- APy(i1))/
   r(i1,xi,yi)^2;
end
H(:,:,xi,yi) = [H1(:,xi,yi) H2(:,xi,yi)];
Covv(:,:,xi,yi) = SD^2*((H(:,:,xi,yi)'*H(:,:,xi,yi))^(-1)
   );
SDr(xi,yi) = sqrt(Covv(1,1,xi,yi)+Covv(2,2,xi,yi));
```

```
end
end
SDr = SDr';
contourf(mx,my,SDr,20);
colormap(hot);
colorbar;
xlabel('X-axis(meter)');
ylabel('Y-axis(meter)');
title('Contour of Location Error Standard Deviation (
   meter)');
```

```
MATLAB CODE FOR CRLB SIMULATIONS WITH BEACON AT BACK OF
   EVERY SPACE
clc; clear all; close all;
APx(1) = 1.5; APy(1) = 14.5; APx(2) = 4.5; APy(2) = 14.5;
    APx(3) = 7.5; APy(3) = 14.5;
APx(4) = 10.5; APy(4) = 14.5; APx(5) = 13.5; APy(5) =
   14.5;
APx(6) = - 1.5; APy(6) = 14.5; APx(7) = - 4.5; APy(7) =
   14.5; APx(8) = - 7.5; APy(8) = 14.5;
APx(9) = - 10.5; APy(9) = 14.5; APx(10) = - 13.5; APy(10)
    = 14.5;
APx(11) = 1.5; APy(11) = - 14.5; APx(12) = 4.5; APy(12) =
    - 14.5; APx(13) = 7.5; APy(13) = - 14.5;
APx(14) = 10.5; APy(14) = - 14.5; APx(15) = 13.5; APy(15)
    = - 14.5;
APx(16) = - 1.5; APy(16) = - 14.5; APx(17) = - 4.5; APy
   (17) = - 14.5; APx(18) = - 7.5; APy(18) = - 14.5;
APx(19) = - 10.5; APy(19) = - 14.5; APx(20) = - 13.5; APy
   (20) = - 14.5;
SD = 3;
NUM = 20;
mx = - 15:0.1:15;
my = - 15:0.1:15;
nxy = length(mx);
for yi = 1:nxy
    for xi = 1:nxy

        for i1 = 1:NUM
            alpha(i1) = 2.5;
            r(i1, xi, yi) = sqrt((mx(xi) - APx(i1)) ^ 2 +
                (my(yi) - APy(i1)) ^ 2);
            H1(i1, xi, yi) = - 10 * alpha(i1) / (log(10))
                * (mx(xi) - APx(i1)) / r(i1, xi, yi) ^ 2;
            H2(i1, xi, yi) = - 10 * alpha(i1) / (log(10))
```

```
                * (my(yi) - APy(i1)) / r(i1, xi, yi) ^ 2;
        end
        H(:, :, xi, yi) = [H1(:, xi, yi) H2(:, xi, yi)];
        Covv(:, :, xi, yi) = SD ^ 2 * ((H(:, :, xi, yi)'*
           H(:,:,xi,yi))^(-1));
        SDr(xi, yi) = sqrt(Covv(1, 1, xi, yi) + Covv(2,
           2, xi, yi));
    end
end
SDr = SDr';
contourf(mx, my, SDr, 20);
colormap(hot);
colorbar;
xlabel('X-axis(meter)');
ylabel('Y-axis(meter)');
title('Contour of Location Error Standard Deviation (
   meter)');
```

## APPENDIX C: MATLAB CODE FOR LOCALIZATION ALGORITHMS

```
clc;clear all;close all;
CertainRange=0.2;
sigma=8;dbp=5;
Pt=13;
fc=2.4e9;
c=3e8;
L0=-20*log10(c/4/pi/fc);
known_references = [10,10;0,15;-5,5];
distances = [15,16,5];
FadeMargin=sigma*sqrt(2)*erfcinv(1+CertainRange);
Lp=L0+20*log10(dbp)+35*log10(distances./dbp);
Bound=[dbp.*10.^((Lp+FadeMargin-L0-20*log10(dbp))./35);
   dbp*10.^((Lp-FadeMargin-L0-20*log10(dbp))./35)];
pace=1;
x=-100:pace:100;y=-100:pace:100;
l1=length(x);
k=0;
for i=1:l1
    for j=1:l1
        circle1 = (x(i)-known_references(1,1))^2+(y(j)-
           known_references(1,2))^2 >= (Bound(1,1))^2;
        circle2 = (x(i)-known_references(1,1))^2+(y(j)-
           known_references(1,2))^2 <= (Bound(2,1))^2;
        circle3 = (x(i)-known_references(2,1))^2+(y(j)-
           known_references(2,2))^2 >= (Bound(1,2))^2;
        circle4 = (x(i)-known_references(2,1))^2+(y(j)-
           known_references(2,2))^2 <= (Bound(2,2))^2;
        circle5 = (x(i)-known_references(3,1))^2+(y(j)-
           known_references(3,2))^2 >= (Bound(1,3))^2;
        circle6 = (x(i)-known_references(3,1))^2+(y(j)-
           known_references(3,2))^2 <= (Bound(2,3))^2;
        if circle1 && circle2 && circle3 && circle4 &&
           circle5 && circle6
            k=k+1;
            sol(k,1)=x(i);
            sol(k,2)=y(j);
        end
    end
end
figure(1)
angle=0:0.5:360;
hold on
plot(Bound(1,1).*cosd(angle)+known_references(1,1),Bound
   (1,1).*sind(angle)+known_references(1,2),'r.')
plot(Bound(2,1).*cosd(angle)+known_references(1,1),Bound
   (2,1).*sind(angle)+known_references(1,2),'r.')
plot(Bound(1,2).*cosd(angle)+known_references(2,1),Bound
   (1,2).*sind(angle)+known_references(2,2),'b.')
plot(Bound(2,2).*cosd(angle)+known_references(2,1),Bound
   (2,2).*sind(angle)+known_references(2,2),'b.')
plot(Bound(1,3).*cosd(angle)+known_references(3,1),Bound
   (1,3).*sind(angle)+known_references(3,2),'g.')
plot(Bound(2,3).*cosd(angle)+known_references(3,1),Bound
   (2,3).*sind(angle)+known_references(3,2),'g.')
for i=1:k
    plot(sol(i,1),sol(i,2),'k.');
end
ex=mean(sol(:,1));ey=mean(sol(:,2));
plot(ex,ey,'m.','MarkerSize',30);
hold off
disp(['The estimated location is:  ','[',num2str(ex),',',
   num2str(ey),']']);
```

# APPENDIX D: C# CODE FOR LOCALIZATION ALGORITHMS

```
public static void maxLH(List < Beacon > beacons, double certainty) {


 List < double[] > bounds = new List < double[] > ();
 //List of array size 2 for bounds calculation
 List < double > Lp = new List < double > ();
 //Lp for each distance bounds
 List < int > x_sol = new List < int > ();
 List < int > y_sol = new List < int > ();
 double solution_x = 0, solution_y = 0;
 var x = Enumerable.Range(-100, 200).ToList();
 var y = Enumerable.Range(-100, 200).ToList();
 var CertainRange = certainty; //Certainty Range

 //Use 802.11 channel model C to find bound for certainty range
 int sigma = 8; //Shadow Fading
 double dbp = 5; //??
 int Pt = 13; //Transmit Power
 double fc = 2.4e9; //Frequency
 double c = 3e8; //Speed of Light


 var L0 = -20 * Math.Log10(c / 4 / Math.PI / fc);
 double FadeMargin = sigma * Math.Sqrt(2) * SpecialFunctions.ErfcInv(1 + Certain


 foreach(var b in beacons) {
  int i = 0;

  Lp.Add(L0 + 20 * Math.Log10(dbp) + 35 * Math.Log10(b.Distance / dbp));

  i++;
```

```
}

int t = 0;
foreach(var l in Lp) {

 double[] temp = new double[] {
  dbp * Math.Pow(10, ((Lp[t] + FadeMargin - L0 - 20 * Math.Log10(dbp)) / 35)),
   dbp * Math.Pow(10, ((Lp[t] - FadeMargin - L0 - 20 * Math.Log10(dbp)) / 35))
 };

 bounds.Add(temp);

 t++;
}

int k = 0;
for (int i = 0; i < x.Count(); i++) {
 for (int j = 0; j < y.Count(); j++) {
  bool cir1 = TestRange((Math.Pow(x[i] - beacons[0].Location[0], 2) + System.M
  bool cir2 = TestRange((Math.Pow(x[i] - beacons[1].Location[0], 2) + System.M
  bool cir3 = TestRange((Math.Pow(x[i] - beacons[2].Location[0], 2) + System.M
  if (cir1 && cir2 && cir3) {
   k = k + 1;
   x_sol.Add(x[i]);
   y_sol.Add(y[j]);
  }
 }
}

try {
 solution_x = x_sol.Average();
 solution_y = y_sol.Average();

} catch {
    Console.WriteLine("\n ERROR NO CONVERGENCE");
  }
}

public static bool TestRange(double numberToCheck, double bottom, double top) {
  return (numberToCheck >= bottom && numberToCheck <= top);
}
```