Indoor RF Positioning System using Collaborative Trilateration Method

Llewellyn Liu Wee Ling

Faculty of Engineering, Computing and Science (FECS)
Swinburne University of Technology
Sarawak Campus, Malaysia

Submitted for the degree of Master of Engineering (Research)

2017
Abstract

In recent years, location aware technology has become a common necessity. There is a growing interest and need for location aware systems such as inventory tracking and logistics, safety and security, navigational and tracking systems and so on. However, the accuracy of the current technology such as Global Positioning System (GPS) is limited to outdoor where there are clear line of sight environments. Aside from the recent advancement of GPS technology, there is still a huge gap in the area of study for indoor location aware systems. This study proposes an indoor localization technique which is a key part of location aware system.

With advancement in the Wi-Fi technology, Wi-Fi clients can now connect with each other without the need of a Wi-Fi Access Point or router. This makes the Wi-Fi technology an attractive candidate for the purpose of indoor localization. This study proposes a design and proof of concept of a low cost and robust indoor positioning algorithm using existing Wi-Fi infrastructure and Wi-Fi-Direct enabled smart phones.
Acknowledgement

This Masters study was not an easy task. It would not have been possible for me to overcome the challenges without the assistance of many individuals. I could not have completed this journey without the continued support from my family, friends and colleagues.

First, I would like to express my gratitude to my main advisor Associate Professor Wallace Wong Shung Hui for taking valuable time out of his busy schedule in order to provide meticulous guidance and feedback through the dissertation process. I also wish to express my sincere thanks to Professor Dennis Wong, my co-advisor, for his insight, consistent support and encouragement.

I am grateful for the motivation provided by my fellow colleagues in Swinburne University of Technology (Sarawak Campus). Their generous sharing and laughter always help to ease my stress level.

My very special thanks to my mother for everything she have done for me. Her support and encouragement which helped me persevere through many difficult times. Also, I would like to express my deepest appreciation to my loving wife, who consistently believed in me and pushes me forward as I was about to give up. Without her, I would never have had the determination to accomplish this journey. To my two princesses, whose laughs and giggles helped me going on day-to-day, they are the constant reminder of why I embark on this journey!

Finally, I thank God for making this possible. Amen!
Declaration

I declare that this thesis contains no material that has been accepted for the award of any other
degree or diploma and to the best of my knowledge contains no material previously published or
written by another person except where due reference is made in the text of this thesis.

[Signature]

Llewellyn Liu Wee Ling

20 June, 2017
# Table of Contents

Abstract .................................................................................................................................................. i

Acknowledgement .................................................................................................................................. ii

Declaration ........................................................................................................................................... iii

Table of Contents ................................................................................................................................... iv

Chapter 1.  Introduction .......................................................................................................................... 1

1.1  Introduction ..................................................................................................................................... 1

1.2  Research Contributions and Objectives .......................................................................................... 4

1.3  Organization of the thesis ................................................................................................................. 6

Chapter 2.  Literature Review and Related Work ...................................................................................... 7

2.1  RF-Based indoor localization methods .............................................................................................. 7

2.1.1  Radio frequency identification (RFID) ......................................................................................... 7

2.1.2  Wi-Fi ............................................................................................................................................. 8

2.2  Localization Methods ....................................................................................................................... 9

2.2.1  Proximity ..................................................................................................................................... 9

2.2.2  Scene Analysis ............................................................................................................................. 15

2.2.3  Triangulation and Trilateration .................................................................................................. 18

2.3  Indoor propagation model ................................................................................................................. 25

2.3.1  Log Path Loss Model .................................................................................................................. 25
List of publications produced as a result of this study... XXI
List of Figures

Figure 2.1: Weighted Screening Method Processflow ................................................................. 13
Figure 2.2: WSM - weighted intersection screening method ....................................................... 14
Figure 2.3: Parameters for intersection of three circles .............................................................. 19
Figure 2.4: (a) No intersection (b) Circle in circle ........................................................................ 20
Figure 2.5: UWB Experiment Setup ............................................................................................ 21
Figure 2.6: AOA simulation results ............................................................................................... 22
Figure 2.7: Cumulative plot of estimation error for Regression and Correlation Based Lateration Method .............................................................................................................. 24
Figure 3.1: Simple Wi-Fi RSSI data collection setup .................................................................... 32
Figure 3.2: Measured RSSI vs Log loss and ITU indoor propagation model ................................. 33
Figure 3.3: Visualization of the filtering process .......................................................................... 36
Figure 3.4: Kalman Filter applied to RSSI used in LANDMARC .................................................. 37
Figure 3.5: Kalman Filter LANDMARC Test bed setup ............................................................... 38
Figure 3.6: Cumulative plot of estimated error LANDMARC vs Kalman Filtered LANDMARC .... 40
Figure 3.7: Location relationship RFID vs Wi-Fi ....................................................................... 43
Figure 3.8: LANDMARC for Wi-Fi model .................................................................................... 45
Figure 3.9: Wi-Fi Router and Repeater placement in Swinburne Research Centre ....................... 46
Figure 3.10: Test Locations in Swinburne Research Centre ......................................................... 47
Figure 3.11: Average Estimation Error of Filtered and Unfiltered LANDMARC for Wi-Fi .......... 48
Figure 3.12: LANDMARC Wi-Fi coverage area highlighted in yellow ......................................... 49
Figure 4.1: One degree margin of error ....................................................................................... 53
Figure 4.2: Point in triangle test .................................................................................................. 54
Figure 4.3: Visualisation of the proposed Trilateration-Triangulation Method ............................... 55
Figure 4.4: Proposed Trilateration-Triangulation Method process flow ........................................ 57
Figure 4.5: Collaborative Trilateration Method Process flow ......................................................... 58
Figure 4.6: Visualization of selected reference points .................................................................... 63
Figure 4.7: Visualization of the trilateration method .................................................................... 64
Figure 5.1: RSSI data collection positions ..................................................................................... 69
Figure 5.2: Simulated Random Clients .......................................................................................... 70
Figure 5.3: Simulation process ...................................................................................................... 73
Figure 5.4: Fix-AP deployments .................................................................................................... 74
Figure 5.5: Probability plot of Estimation Error Collaborative Trilateration Method vs LANDMARC – 25AP ........................................................................................................................................ 76
Figure 5.6: Probability plot of Estimation Error Collaborative Trilateration Method vs LANDMARC – 4AP ........................................................................................................................................ 77
Figure 5.7: Clients color sets ......................................................................................................... 78
Figure 5.8: Estimation error selected clients over 120 iterations .................................................. 79
Figure 5.9: Probability plot of Estimation Error Collaborative Trilateration Method 4AP vs 5AP vs 9AP vs 25AP ........................................................................................................................................ 80
Figure 5.10: Probability plot of Estimation Error Collaborative Trilateration Method 4AP vs 5AP vs 9AP vs 25AP ........................................................................................................................................ 81
List of tables
Table 2-1: Performance vs Accuracy Trade-off...................................................................................... 17
Table 2-2: Distance power loss coefficient $N$ .......................................................................................... 28
Table 2-3: Floor penetration loss factors, $L_f$ (dB) with $n$ being the number of floors ......................... 29
Table 2-4: RF-Based indoor location technologies comparison table ...................................................... 30
Table 2-5: Indoor location methods comparison table............................................................................... 31
Table 2-6: Indoor propagation model comparison table............................................................................. 31
Table 3-1: Kalman Filter Equation............................................................................................................... 34
Table 3-2: Kalman Equations for RFID Experiment.................................................................................... 39
Table 4-1: Quadrant Information table........................................................................................................ 60
Table 4-2: Confidence Rating Table based on Kalman gain (K)................................................................. 66
Chapter 1. Introduction

1.1 Introduction

In these days of mobile computing, social networking, and globalization, there is a growing interest and need for location aware systems such as inventory tracking and logistics, safety and security systems and navigational and tracking systems. This study focuses on an indoor localization technique which is a key part of location aware system. Besides indoor navigational applications, knowing the location of an individual or object could enable useful location-aware service. Services such as highlighting key interesting information in an exhibition or museum as the user navigates around, or providing personalised location-aware advertisement to shoppers as they moved about in the mall. In an unfortunate event where a child wanders off, and got lost in a crowded area, the ability to track and locate indoors would put the parents mind at ease. This is especially true, in large and complex indoor environments. In cases of emergency, the ability to locate the nearest First-Aid kit, Automated External Defibrillator (AED) or find a security officer, would greatly improve the situation for first responders.

However current commercially available location aware systems are mainly based on Global Positioning System (GPS) Technology. Despite the unprecedented advancement of GPS Technology [24], GPS is still limited to outdoor application where there is no obstruction to the satellite’s signal. This causes a huge gap in this area of study for indoor location aware systems, especially in large building complexes and maze like underground cities such as Tenjin Underground City in Japan, Taipei Underground Market in Taiwan and many others. Recent studies on indoor Earth based GPS satellites (LOCATA) [5] shows promising results for indoor
location aware systems. However new and expensive infrastructure would have to be setup in the indoor structures for the system to work. This will be extremely expensive to deploy into every single existing indoor structures.

Radio Frequency (RF) based technologies [17], such as Wi-Fi, Bluetooth and NFC in modern mobile communication devices were not initially designed to provide location aware services. However, with the recent advancement in mobile device technology, which includes an assortment of sensors, transducers and a powerful microprocessor, the RF based technologies in the mobile devices has becomes a viable alternative for this purpose.

Most approaches to RF based positioning systems utilize the radio signal strength index (RSSI), which is a readily measured parameter in all common RF based wireless system. However, RSSI is usually inaccurate and behaves erratically especially in indoor environment. The RSSI can be corrupted by unwanted noise and interference from other devices. As radio waves propagate in an indoor environment, it scatters in unpredictable ways. Absorption, diffraction and reflection can occur when RF waves encounter any opaque surfaces. These can either decrease or increase the RSSI readings at the reception point. This makes the prediction of RSSI in any location of the environment a difficult and complex task.

The IEEE 802.11 standard [1], or more commonly known as Wi-Fi uses license-free frequencies in the 2.4, 3.6, 5 and 60 GHz band. The 2.4 GHz frequency is widely used in today’s mobile devices, which complies with the IEEE 802.11 b/g/n standard protocols. This makes the IEEE 802.11 standard an attractive candidate for the purpose of indoor localization. However every living being and mass bodies will absorb RF waves in this frequency band. Changes in atmospheric condition like temperature and humidity will also affect the propagation of the RF waves and ultimately the RSSI reading.
A conventional Wi-Fi network consists of Wi-Fi Access Points (AP) and Wi-Fi clients. Wi-Fi clients communicate with each other via the Wi-Fi AP. The Wi-Fi AP provides three primary functions, mainly as a physical interface between wired and wireless devices, bridging and routing network traffic and extending network coverage of a wireless network. Generally, the Wi-Fi AP are placed in fix location, and very rarely are they moved, thus making them suitable to be used as point of reference for indoor localization.

With advancement in the Wi-Fi peer-to-peer technology [10], clients can connect with each other without the need of a Wi-Fi Access Point. This is accomplished by Wi-Fi Direct, where the Wi-Fi client can function as a Wi-Fi Client as well as a Wi-Fi AP or Soft-AP. Once connected, the client can exchange location data which are not normally transmitted in the Wi-Fi protocol.

In this thesis, the design and proof of concept of a low cost and robust indoor positioning algorithm using RF is presented.
1.2 Research Contributions and Objectives

The contribution of this thesis is mainly on the design and development of a RF-Based indoor positioning system. In order to achieve this, the following objectives must be accomplished.

1. Accuracy - The location estimation accuracy must be within an acceptable range.
2. Robustness – The system must be able to sustain to a certain degree of accuracy in the event of abnormal measurement or loss of signal.
3. Flexibility and Scalability - The system must be flexible and can be configured in many different ways, and can be easily extended to expand the tracked area.

By relying on existing Wi-Fi infrastructure such as Wi-Fi access points and the user’s own Wi-Fi enabled mobile devices, the cost of implementing the system could be kept at minimal. There should not be a need for specialised equipment or setting up additional infrastructure.

The system could be easily scaled up by adding new Wi-Fi access points outside of the current track area to increase its coverage. The system should also be flexible to accommodate users moving in and out of the tracked area.
As the work explores Wi-Fi positioning methods at conceptual level, the following assumptions were made:

1. All access points (AP) and tracked clients in soft-AP mode have the same transmitting power (Tx dDm) and frequency band.

2. All tracked clients have the same antenna gain, and thus having the same received signal strength index (RSSI).

3. All Fix-AP and Soft-AP have a circular RSSI distribution map within 5 meters from the point of origin, using actual RSSI readings.

4. All tracked clients are stationary throughout the experiments and simulations.

It should be noted that these assumptions are unlikely to exist in the real world, especially transmission power and antenna gain. Different device model and make would have different performance.
1.3 Organization of the thesis

Chapter 1 forms the introduction of this thesis by providing context and presenting the current state of indoor positioning system. The contributions and assumptions of this research are presented and the organization of the thesis is outlined.

Chapter 2 presents the fundamental theory and related works in RF localization and positioning systems. Two commonly used indoor propagation models for radio signal are presented and discussed.

Chapter 3 presents a review of the RFID-based LANDMARC localization approach. The inclusion of a discrete Kalman Filter is proposed to improved accuracy. The idea of using Wi-Fi with a modified LANDMARC method are then presented and the experiment results are discussed.

Chapter 4 focuses on the main contribution of this thesis. It details of the proposed collaborative trilateration method and introduces an alternative trilateration-triangulation approach.

Chapter 5 presents the simulations and experiments results of the proposed collaborative trilateration method. The design of the experiments is presented and the results are discussed.

Chapter 6 summarises the findings of this work. It concludes the work carried out and also discusses the possibilities for future work.
Chapter 2.  Literature Review and Related Work

In this chapter, the fundamental theories and other related works on indoor RF-Based indoor localization systems are reviewed. The general methods used for RF-based indoor localization and the commonly used radio signal indoor RF propagation models are discussed later in the chapter. The benefits and challenges of the different methods are then compared.

2.1  RF-Based indoor localization methods.

This thesis focuses on indoor localization methods based on RFID and Wi-Fi radio frequency technologies.

2.1.1  Radio frequency identification (RFID)

Want, R. (2006) [34] introduces the Radio Frequency IDentification (RFID) technology, which uses electromagnetic waves to identify and track tags attached to objects. These tags contain information regarding the object, and it is stored electronically. Generally, there are two types of RFID tags, passive and active tags. Passive tags draw its power from the RFID reader and active tags have its own power source such as a battery. Active tags can operate at a greater distance from the reader, whereas passive tags need to be within a meter or two from the reader.

RFID was not designed as a solution for indoor localization. However recent studies have made this technology a viable solution for indoor localization by placing tags and readers around the tracked area and on tracked objects.
2.1.2 Wi-Fi

The IEEE 802.11 standard (1997) [1] or Wi-Fi as it is commonly known, is a technology that allows electronic devices to connect to wireless local area network (WLAN). It is the most prevailing wireless network standard used worldwide since its introduction. It is in almost every homes and public places such as cafe, shopping malls and galleries. Many electronic devices these days have built in Wi-Fi compatibility.

There is a variety of existing Wi-Fi based indoor localization and tracking systems. A typical accuracy of these positioning systems is approximately a few meters, with an update frequency of once in few seconds. The main advantage of Wi-Fi based localization systems is its compatibility with existing Wi-Fi infrastructure which reduces the cost of implementing such systems.
2.2 Localization Methods

Modeling radio propagation in the indoor environment is complex. The enclosed power and network lines, moving objects and people in it can cause severe RF multipath, RF absorption, and deflection or reflection of the RF signal.

In order to mitigate the impact of localization errors due to these RF noises, three general methods were considered - proximity, scene analysis and triangulation-trilateration methods. These three methods have unique advantages and disadvantages and it is possible to use combinations of any of these methods for better performance.

2.2.1 Proximity

The proximity based method uses the known location of reference points to collocate the tracked clients. The reference points are usually access points or other forms of transmitting station. The method relies heavily on a dense grid of reference points. The tracked client is considered to be within the proximity of the reference point when the object can be detected by the reference point.

An example of this method for outdoor application is the Cell of Origin (COO) method which is already in use today. The method is often used in conjunction with GPS to provide better precision. It is more commonly known as Assisted-GPS (AGPS) [24] and it is a common feature in all major smart phones brands. Mobile cellular network providers can identify the cell tower that the device is connected to at any given time. It then uses this information to assist the GPS system on the handset to obtain location information for the user.
A. LANDMARC: indoor location sensing using active RFID

Ni, L.M et al. (2004) [28] presented a proximity based indoor location sensing prototype system that uses Radio Frequency Identification (RFID) technology for indoor localization, which they named LANDMARC. The main advantage of LANDMARC is that it improves the overall accuracy of locating tracked clients by utilizing reference tags instead of just RFID reader to account for inaccuracies in RSSI.

The LANDMARC approach can be summarized as follows. Presume that \( n \) number of RF readers, \( m \) number of reference tags and \( u \) number of tracked tags as objects are being tracked. All tags are assumed to be stationary throughout the process. Then the signal strength vector for a tracking tag is defined as:

\[
\vec{S} = (S_1, S_2, \ldots, S_n)
\]

Where \( S_i \) denotes the signal strength of the tracking tag perceived on reader \( i \), and \( i \in (1, n) \). The signal strength vector for the reference tags is defined as:

\[
\vec{\theta} = (\theta_1, \theta_2, \ldots, \theta_n)
\]

Where \( \theta_i \) also denotes the signal strength of the reference tag perceived on reader \( i \), and \( i \in (1, n) \).

The Euclidian distance \( E_j \) for each tracking tag \( p \), and \( p \in (1, u) \) in terms of signal strength can be calculated as:

\[
E_j = \sqrt{\sum_{i=1}^{n}(\theta_i - S_i)^2}
\]
where \( j \in (1, m) \). The Euclidian distance \( (E_j) \) is the distance in terms of signal strength between the tracking tag and reference tag \( r_j \). The value of \( E \) denotes the location relationship between the reference tags and the tracking tag. As an example, the nearer the reference tag is to the tracking tag, the smaller \( E \) value. A tracking tag has its \( E \) vector defined as:

\[
\vec{E} = (E_1, E_2, ..., E_m)
\]

for \( m \) reference tags.

In order to determine the number of reference tags in a reference cell that is to be used to calculate the approximate coordinate of the unknown tag, the \( k \)-nearest neighbor algorithm is applied. On average the authors found that \( k = 4 \) shows the best results in obtaining the most accurate approximate coordinate for each unknown tracking tag.

The unknown tracking tag coordinate \((x, y)\) is obtained by solving:

\[
(x, y) = \sum_{i=1}^{k} w_i (x_i, y_i)
\]

Where \( w_i \) is the weighting factor to the \( i \)-th neighbouring reference tag. The weighting factor is dependent on the Euclidian distance of each reference tag, and is defined as:

\[
w_i = \frac{1}{\sum_{i=1}^{k} \frac{1}{E_i^2}}
\]

The reference tag with the smallest \( E \) value has the largest weight, as the distance between it and the unknown tag is the smallest.
The localization accuracy is measured in terms of the error distance between the approximate coordinate and the actual coordinate:

\[ e = \sqrt{(x - x_0)^2 + (y - y_0)^2} \]

where the tracked tags actual location is given by \((x_0, y_0)\) and the approximated coordinates given by \((x, y)\).

One of the drawbacks to LANDMARC is that the weighting factor is dependent on the Euclidian distance which is calculated based on RSSI readings. The RSSI readings can be affected by RF noises and give inconsistent results. This is why the results have a wide range of errors in position estimation.

Another challenge is that the location information received from the RFID readers needs to be processed on a centralized computational server. The processed location information can then be stored locally or transmitted via a wired or wireless network.

**B. Weighted Proximity Based Location Method (WEP)**

Brida, P et al. (2007) [9] proposed a weighted proximity based location method. Although the method is not as accurate, it is a low cost alternative. The proposed weights are based on the \(N\)-th nearest reference point. The fundamental of this method is that the nearest reference point would be assigned weighting of \(N = 0\) and the farthest reference point would have the weighting of 1.

Each weighted reference point \((x_i, y_i)\) would contribute to the estimated location’s input part, where the estimated location can be written as a total of all the input parts:

\[
(x_{WEP_{est}}, y_{WEP_{est}}) = \left( \frac{\sum_{i=1}^{N} x_i \cdot w_i}{\sum_{i=1}^{N} w_i}, \frac{\sum_{i=1}^{N} y_i \cdot w_i}{\sum_{i=1}^{N} w_i} \right)
\]
The simplicity of the method and ease of implementation makes this method noteworthy. However the weights selection could be improved by using RSSI measurements instead of just $N$th ranking of the reference point.

C. **Weighted Screening Method (WSM)**

Liu, H.H et al. (2014) [26] proposes a Wi-Fi based Weighted Screening Method (WSM) to improve displacement error in trilateration. By using trilateration and proximity, their proposed method identifies possible intersection points that are closer together and screen out points that are further away. The average of the closer intersection points is taken to obtain the tracked client’s position. They also used the radius-based weighted average point in situations of disjointing circles. Figure 2.1 shows the overall process flow of the method.

![Figure 2.1: Weighted Screening Method Processflow](image-url)
The radius-based weights are a ratio of the radius of the circles, as shown in Figure 2.2. Points $I_A$, $I_B$ and $I_C$ are calculated based on the weights ratio of $r_1/r_2$, $r_2/r_3$ and $r_1/r_3$ in situations of disjoint circles.

Figure 2.2: WSM - weighted intersection screening method
2.2.2 Scene Analysis

Another method of localization is through RSSI fingerprinting or scene analysis that matches the RSSI of the tracked client to the RSSI of different locations of the tracked area stored in a database. There are two stages in the RSS-based location fingerprinting technique, an offline stage and online stage.

In the offline stage a site survey is performed in the tracked area. The location coordinates are labelled and the respective RSSI measurements to nearby access points or base stations are collected. The second stage is the online stage, where the measured RSSI of the tracked client is compared with the previously collected RSSI information in the database to estimate the current location of the object.

A. RADAR

The RADAR system, which consists of multiple Wi-Fi access points (AP) and a laptop computer, was one of the earliest known studies of Wi-Fi based indoor localization done by Bahl and Padmanabhan (2000) [4] under Microsoft. They experimented on locating Wi-Fi users by using a method of RSSI ranging. The experiment involves a laptop carried by an individual in an overlapping coverage area. The mean measurement of the RSSI at the APs is used to compare with a database of pre-recorded RSSI fingerprints. The best match between the measured and the pre-recorded RSSI indicates the estimated location of the user target. It was reported in their paper [4] that the method can attain a location accuracy of about 2.94 m with about 50% probability.
B. Ekahau


Ekahau’s real-time location tracking system uses small battery-powered Wi-Fi tags, which are attached to the tracked clients. These tags are essentially active RFID tags. Ekahau provides a complete hardware and software solution which is able to operate over any brand or generation of Wi-Fi network standards to provide an accurate location, on average 1 meter.

The location method Ekahau employs is basically RSSI fingerprinting, however its model can be optimized according to the indoor scenario and thus able to provide better accuracy. This is performed through a calibration process using the Ekahau software. To calibrate an area of 1,111m², Gu, Lo and Niemegeers (2009) [17] mentioned that it may take approximately an hour. However, there is no information about how often the calibration process needs to be carried out. There is also no mention of the sensitivity to environmental change. This is one of the major disadvantages of RSSI fingerprinting based location systems.
C. On Indoor Position Location

Prasithsangaree, P. et al [30] has provided a framework to systematically analyse and estimate indoor location using Wi-Fi. Their framework focuses on addressing key issues such as continuous positioning which burdens the network’s computational power. A built-in self-positioning mechanism was implemented into the tracked client. Therefore, instead of relying on the APs to measure RSSI to the tracked client, it can measure the RSSI from the APs when it requires a position.

On the granularity of the database entries, they studied the grid spacing of 5 feet and 10 feet to evaluate the trade-offs between performance and delay. Table 2-1 below shows the performance vs. accuracy trade-off. Their experiments show that the proposed framework could reduce the time to obtain a fingerprint match by ten times with about 15% trade-off in accuracy.

Table 2-1: Performance vs Accuracy Trade-off

<table>
<thead>
<tr>
<th>Database Granularity</th>
<th>Five Feet</th>
<th>Ten Feet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average Distance Error (x10 ft)</td>
<td>2.17</td>
<td>2.5</td>
</tr>
<tr>
<td>Time to obtain match (seconds)</td>
<td>10.43</td>
<td>1.27</td>
</tr>
</tbody>
</table>
2.2.3 Triangulation and Trilateration

Triangulation and trilateration refers to the method that uses the geometric properties of triangles and distances to estimate the location of the tracked client. Triangulation uses the geometric properties of triangles to estimate by computing angles relative to multiple reference points. Trilateration on the other hand estimates the location of an object by measuring its distances from multiple reference points. Angles is a measured parameter from angle of arrival (AoA) and distances can be calculated from time of arrival (ToA), time difference of arrival (TDoA) or indoor propagation models using RSSI measurements.

A. Traditional Trilateration method

Trilateration is a method of determining locations of points by means of distances, using the geometrical principles of circles, spheres or triangles. The location of the point of interest is the intersection point of the three circles. This can be found by formulating the equations for the three circles and solving the equations for the unknown coordinates \((x, y)\) of the point of interest.
Considering the three right angled triangles $P_1P_4P_5$, $P_2P_4P_5$ and $P_3P_4P_6$ shown in Figure 2.3, the equations below can be defined:

\[ R_1^2 = x^2 + y^2 \quad (1) \]
\[ R_2^2 = b^2 + y^2 = (x - d)^2 + y^2 \quad \therefore b = d - x \quad (2) \]
\[ R_3^2 = (x - i)^2 + (y - j)^2 \quad (3) \]

Subtract equation (1) and (2) to obtain $x$:

\[ R_1^2 - R_2^2 = 2dx - d^2 \]
\[ x = \frac{R_1^2 - R_2^2 + d^2}{2d} \quad (4) \]

Then subtract equation (1) and (3), and swapping in $x$ from (4) $y$ can be obtained:

\[ R_1^2 - R_3^2 = 2ix - i^2 + 2jy - j^2 \]
\[ y = \frac{R_1^2 - R_3^2 + i^2 + j^2}{2j} - \frac{i}{j} \left( \frac{R_1^2 - R_2^2 + d^2}{2d} \right) \quad (5) \]
This traditional method of trilateration proves to be effective in an ideal situation where the estimated distance to all three reference points are accurate.

However, in this study, the estimated distance is calculated based on RSSI reading which can be inaccurate. Imperially, at close proximity to the AP the estimated distance is substantially accurate, but if the client is further away from the AP, the accuracy of the estimated distance worsens.

Also, typically Wi-Fi APs are sparsely distributed to provide better Wi-Fi coverage. It is not uncommon to run into the situation where \( d > R_1 + R_2 \), as shown in Figure 2.4(a) where \( d \) is the distance between two Wi-Fi AP, and \( R_1 \) and \( R_2 \) are the estimated distances calculated from the RSSI between the client to the respective Wi-Fi APs. In this situation there is no solution if the traditional trilateration method is used, as the circles formed do not intersect each other. An alternative trilateration method is proposed in Chapter 4 to address this issue.

![Figure 2.4: (a) No intersection (b) Circle in circle](image)

Figure 2.4(b) shows yet another undesirable situation, when \( d < |R_1 - R_2| \). There is also no solution for this situation because the circles are contained within one another.
B. UWB indoor 3D positioning radar

Zhang C. et al. [43] proposed an indoor localization method based on ultra wideband (UWB) using the time difference of arrival (TDOA) method in conjunction with RSSI. This method provides sub-centimetres accuracy which is excellent for indoor localization. However, this accuracy comes at the cost of additional UWB equipment.

Figure 2.5: UWB Experiment Setup

Figure 2.5 above shows that the experiment was carried out in a small confined space of about 3 meters by 4 meters. One of the walls was made on metal and the receiving stations (Rx) were separated not more than 2 meters apart from each other. The tracked transmission station (Tx) was placed in the centre of the receiving stations.
It was noted in [43] that the distance between the tracked client and reference point can affect the overall accuracy. The nearer the tracked client is to the reference point the higher the accuracy.

C. **WLAN infrastructure for Angle-of-Arrival**

Wong C, et al [39] investigates the potential of wireless local area network (WLAN) technologies specifically Wi-Fi (802.11n) to be used as an indoor localization method using angle of arrival (AOA) estimation. They reported in [39] that the method can achieve a positioning accuracy of 1.7 m with when an extended Kalman filter is applied. Figure 2.6 show the experimental results of the method.

![Figure 2.6: AOA simulation results.](image)
One major challenge to this method is that it requires a direct line of sight (LOS) for an accurate AOA estimation. AOA is not a readily measurable parameter in the Wi-Fi communication protocol. It is usually estimated using the MUltiple Slignal Classification (MUSIC) algorithm [32]. However, in a real world environment there would be walls and other objects creating RF noises. The AOA estimation would then be corrupted from the RF deflection and reflection making it unsuitable for indoor localization.
D. Regression and Correlation Based Lateration Methods

Yang J and Chen Y [40] proposed two methods to improve location accuracy of lateration methods by using regression and correlation based methods. They compared linear and non-linear regression methods to find a suitable propagation model that would fit between RSSI measurements and the distance. They then used the correlation between the RSSI measurements in the local area to estimate an accurate RSSI reading.

![Cumulative plot of estimation error for Regression and Correlation Based Lateration Method](image)

Figure 2.7: Cumulative plot of estimation error for Regression and Correlation Based Lateration Method

Although the result shown in Figure 2.7, that the proposed method can overcome the noisy RSSI signal in the indoor environment, it is insufficient as the average error is still more than 30ft (9.14 meter).
2.3 Indoor propagation model

In this thesis, radio signal indoor propagation model is used to estimate the distance between the transmitter (AP) and receiver (tracked client) from the RSSI measurement data. The estimated distance is later used to estimate the tracked client’s location.

Propagation model is a characterization of radio wave propagation as a function of frequency, distance and other conditions based on empirical mathematical formulation. A model is established with the goal of generalizing the way radio waves are propagated from one place to another. Such models typically predict the path loss along a link or the effective coverage area of a transmitter.

There are two models which are commonly used - the Log Path Loss model and the ITU indoor propagation model.

2.3.1 Log Path Loss Model

Rappaport, T.S. explained in [31] that the log-distance path loss model is based on an average path loss. This model predicts the path loss of a RF signal in an indoor environment over distance. The model is formally expressed as:

\[
PL = P_{TX_{dBm}} - P_{RX_{dBm}} = PL_0 + 10\gamma \log_{10} \frac{d}{d_0} + X_g
\]
The parameters of the equation are as follows:

- **$PL$** is the total path loss in Decibel (dB)
- **$P_{Tx_{dBm}}$** is the transmitted power in Decibel-miliwatts (dBm), sometimes referred to as Transmit Signal Strength Index (TSSI)
- **$P_{Rx_{dBm}}$** is the received power in Decibel-miliwatts (dBm), sometimes referred to as Received Signal Strength Index (RSSI)
- **$PL_0$** is the total path loss at the reference distance
- **$d$** is the distance of the path between the transmitter and receiver
- **$d_0$** is the reference distance, usually at 1km
- **$\gamma$** is the path loss exponent coefficient
- **$X_g$** is the Gaussian random variable with 0 mean

The distance $d$ can then be calculated from the RSSI reading as:

$$
d = d_0 \cdot 10^{-\frac{P_{Tx_{dBm}} - P_{Rx_{dBm}} - PL_0 - X_g}{10 \gamma}}
$$

This model is not suitable to be used to estimate distance as it does not take into account the indoor parameters such as walls and floors. This model is meant for long distance link budget calculation, and it is not practical to be applied to an indoor environment as the reference distance needed in this model is 1 km.
2.3.2 ITU Indoor Propagation Model

The International Telecommunication Union indoor propagation model [20] is a radio propagation model that estimates the path loss inside a building defined by any form of obstructions. This model approximates the total path loss of an indoor RF signal between devices. The model assumes that the base station and the device are located in the same building or indoor environment. The general model can be express as:

\[ PL = 20 \log_{10} f + N \log_{10} d + L_f(n) - 28 \]

The parameters of the equation are as follows:

- \( PL \) is the total path loss in Decibel (dB)
- \( f \) is the transmission frequency in megahertz (MHz)
- \( N \) is the distance power loss coefficient
- \( d \) is the distance of the path between the transmitter and receiver
- \( L_f \) is the floor penetration loss factor in Decibel (dB)
- \( n \) number of floors between the base station and the portable device (\( n \geq 1 \))
Table 2-2: Distance power loss coefficient $N$

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Residential</th>
<th>Office</th>
<th>Commercial</th>
</tr>
</thead>
<tbody>
<tr>
<td>900 MHz</td>
<td>–</td>
<td>33</td>
<td>20</td>
</tr>
<tr>
<td>1.2–1.3 GHz</td>
<td>–</td>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td>1.8–2 GHz</td>
<td>28</td>
<td>30</td>
<td>22</td>
</tr>
<tr>
<td>2.4 GHz</td>
<td>28</td>
<td>30</td>
<td>–</td>
</tr>
<tr>
<td>3.5 GHz</td>
<td>–</td>
<td>27</td>
<td>–</td>
</tr>
<tr>
<td>4 GHz</td>
<td>–</td>
<td>28</td>
<td>22</td>
</tr>
<tr>
<td>5.2 GHz</td>
<td>30 (apartment)</td>
<td>31</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>28 (house)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8 GHz</td>
<td>–</td>
<td>24</td>
<td>–</td>
</tr>
<tr>
<td>60 GHz</td>
<td>–</td>
<td>22</td>
<td>17</td>
</tr>
<tr>
<td>70 GHz</td>
<td>–</td>
<td>22</td>
<td>–</td>
</tr>
</tbody>
</table>

Table 2-2 above shows the recommended distance power loss coefficient $N$ for ITU model. In this thesis, coefficient for $N$ of 30 is used as RSSI data was collected using a Wi-Fi setup, which operates in the 2.4 GHz frequency range. The setup was in an indoor office environment. The floor penetration loss factor of 14 is used for the operating frequency of 2.4 GHz in an office environment, based on recommendation in Table 2-3.
Table 2-3: Floor penetration loss factors, $L_f$ (dB) with $n$ being the number of floors penetrated ($n \geq 1$)

<table>
<thead>
<tr>
<th>Frequency</th>
<th>Residential</th>
<th>Office</th>
<th>Commercial</th>
</tr>
</thead>
<tbody>
<tr>
<td>900 MHz</td>
<td>–</td>
<td>9 (1 floor)</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td></td>
<td>19 (2 floors)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>24 (3 floors)</td>
<td></td>
</tr>
<tr>
<td>1.8-2 GHz</td>
<td>4 $n$</td>
<td>15 + 4 ($n - 1$)</td>
<td>6 + 3 ($n - 1$)</td>
</tr>
<tr>
<td>2.4 GHz</td>
<td>$10^{(1)}$ (apartment)</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 (house)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5 GHz</td>
<td></td>
<td>18 (1 floor)</td>
<td>26 (2 floors)</td>
</tr>
<tr>
<td>5.2 GHz</td>
<td>$13^{(1)}$ (apartment)</td>
<td>16 (1 floor)</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>7$^{(2)}$ (house)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.8 GHz</td>
<td></td>
<td>22 (1 floor)</td>
<td>28 (2 floors)</td>
</tr>
</tbody>
</table>

(1) Per concrete wall.  
(2) Wooden mortar.

The distance $d$ can then be calculated from the RSSI reading as:

$$d = 10^{\frac{(PL - 20 \log f - P_f(n) + 28)}{N}}$$

This model is more comprehensive as it takes into account the transmission frequency, other indoor parameters such as walls and floors. It also does not require a reference distance measurement.

This study finds that there is a strong correlation empirically with the RSSI data collected and the ITU modeled RSSI by using the recommended constant value. This will be further discussed in Chapter 3.
2.4 Summary

Table 2-4 compares the benefits and challenges of RFID and Wi-Fi based localization methods.

Table 2-4: RF-Based indoor location technologies comparison table

<table>
<thead>
<tr>
<th>Technology</th>
<th>Benefits</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>RFID</td>
<td>• Small and light weight</td>
<td>• Centralised system. RFID tags do not have a processor to compute location information. The computation is done on a centralised computer.</td>
</tr>
<tr>
<td></td>
<td>• Cheap</td>
<td>• Low user base and infrastructure. Mainly used in large warehouse and specialised application such as animal tracking.</td>
</tr>
<tr>
<td></td>
<td>• Low power consumption</td>
<td></td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>• High user base</td>
<td>• Expensive. Its expensive to setup multiple redundant Wi-Fi AP for indoor localization, which does not increase the network coverage area.</td>
</tr>
<tr>
<td></td>
<td>• Used in conjunction with a processor to process its data.</td>
<td>• High power consumption. Having Wi-Fi turned on constantly on mobile devices will drain the battery capacity at an increasing rate.</td>
</tr>
<tr>
<td></td>
<td>• Infrastructure required are well in established in most location.</td>
<td></td>
</tr>
</tbody>
</table>

The RF based indoor positioning methods can be generally divided into 3 types: proximity, scene analysis and, trilateration. The benefits and challenges of the methods are listed in Table 2-5.
### Table 2-5: Indoor location methods comparison table

<table>
<thead>
<tr>
<th>Method</th>
<th>Benefits</th>
<th>Challenges</th>
</tr>
</thead>
</table>
| Proximity     | • Flexible and scalable  
• Robust - Can function even when a reference point go offline. | • Accuracy depends on density of reference point.  
• High density of reference points could increase cost |
| Scene Analysis| • High accuracy  
• Does not require additional infrastructure | • Not flexible and scalable.  
Requires fingerprinting data collection and data training.  
• Not robust. When an AP go offline the system will fail to function with high accuracy. |
| Trilateration | • Flexible and scalable can be applied to any technology  
• Does not require additional infrastructure to work, keeping cost low. | • Not robust. Cannot function when a reference point goes offline, accuracy will be affected.  
• Accuracy depends on distance to reference point and line of sight. |

Currently there are two commonly used indoor radio propagation models - Log Path Loss model and the ITU indoor propagation model. Table 2-6 compares both models.

### Table 2-6: Indoor propagation model comparison table

<table>
<thead>
<tr>
<th>Model</th>
<th>Benefits</th>
<th>Challenges</th>
</tr>
</thead>
</table>
| Log-Loss    | • Simple to calculate                      | • Transmission frequency not considered  
• Indoor environment not considered  
• Reference distance 1km |
| ITU         | • Comprehensive  
• Includes Transmission Frequency  
• Includes Indoor environment | • No reference distance  
• Complex calculation |
Chapter 3. Radio technology as an Indoor Positioning System

This chapter discusses the use of RFID and Wi-Fi technology for indoor positioning. The inclusion of the discrete Kalman Filter into RFID-based LANDMARC technique to filter out the noisy RSSI measurements is demonstrated. The use of Wi-Fi based LANDMARC is then proposed.

3.1 Indoor Propagation Model

A simple test was carried out to collect raw RSSI data using Wi-Fi. A hundred readings of RSSI data was collected at every interval of 1 meter, up to 10 meters away from the AP and in different orientation in a typical office setting as shown in Figure 3.1.

Figure 3.1: Simple Wi-Fi RSSI data collection setup
The average data was then plotted against the modeled RSSI from both the Log-loss and ITU models, as shown in Figure 3.2.

![Measured RSSI vs Log Loss and ITU Model](image)

**Figure 3.2: Measured RSSI vs Log loss and ITU indoor propagation model.**

The results show that the ITU model can match the average RSSI measurements up to 5 meters as shown in Figure 3.2. However, beyond 5 meters the raw RSSI reading starts to fluctuate and becomes unusable. Based on this test, the ITU indoor propagation model is selected for this study.

### 3.2 Challenges with LANDMARC and the Discrete Kalman Filter

As mentioned in Chapter 2, one of the major challenges faced by LANDMARC is the erratic estimated location based on raw RSSI readings. Raw RSSI readings are susceptible to RF noises and therefore produce inconsistent results [28]. Much work have been carried out to address this issue [19] [21] [41]. Studies [18] [25] [29] have shown that the effects of RF noise on the RSSI reading can be reduced by using Kalman filter.
The Kalman filter was introduced by Kalman (1960) [22] as a recursive solution to the discrete data linear filtering problem. It is a computationally efficient set of mathematical equations to estimate the state of a process, in a way that minimizes the mean of the squared error. The filter supports estimations of past, present, and future states; even without knowledge of the precise nature of the modeled system. This is ideal for this study, as it is extremely difficult to model an indoor location system.

Two set of equations forms the Kalman filter. The time update equations and the measurement update equations. The time update equations are responsible for projecting forward in time the current state and the error covariance to obtain the a priori estimates for the next time step. The measurement update equations are responsible for providing feedback by incorporating the latest measurement into the previous estimate to obtain an improved current estimate. The time update equations can also be thought of as predictor equations, while the measurement update equations can be thought of as corrector equations. Table 3-1 below shows the mathematical equations of the Kalman filter.

<table>
<thead>
<tr>
<th>Time Update Equations</th>
<th>Measurement Update Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Project the state ahead</td>
<td>1. Compute the Kalman gain</td>
</tr>
<tr>
<td>( \hat{x}<em>k = A\hat{x}</em>{k-1} + Bu_{k-1} )</td>
<td>( K_k = \frac{P_k^-H^T}{HP_k^-H^T + R} )</td>
</tr>
<tr>
<td>2. Project the error covariance ahead</td>
<td>2. Update estimate with measurement</td>
</tr>
<tr>
<td>( P_k^- = AP_{k-1}A^T + Q )</td>
<td>( \hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) )</td>
</tr>
<tr>
<td>3. Update the error covariance</td>
<td>3. Update the error covariance</td>
</tr>
<tr>
<td>( P_k = (I - K_kH)P_k^- )</td>
<td></td>
</tr>
</tbody>
</table>
An in-depth introduction to the Kalman filter was further elaborated by Bishop and Welch (2001) [7]. It was described in their paper that the parameters $A, B$ and $H$ of the equations are numerical constant. The control parameter is denoted by $u_k$. The process noise covariance is denoted by $Q$ and measurement noise covariance is denoted by $R$. Both the process noise covariance and measurement noise covariance are assumed to be constant. The a previous estimate error covariance is denoted by $P_{k-1}$ and the current estimate error covariance is denoted by $P_k$. The Kalman Gain is denoted by $K_k$ and the measured value is $z_k$.

The filter starts by providing an initial estimate for $\hat{x}_{k-1}$ and $P_{k-1}$ to the time update equations. The projected state $\hat{x}_k$ and the estimated error covariance $P_k$ are obtained from the initial estimate values of $\hat{x}_{k-1}$ and $P_{k-1}$. During the measurement update, the Kalman Gain $K_k$ is calculated from the previous estimated error covariance and the measurement noise covariance. The initial estimate is then updated with measurement $z_k$, to obtain an improved current estimate and finally the estimated error covariance $P_k$ is updated to complete the first iteration of the filter. The updated estimate measurement and estimated error covariance is then used in the next set of time update equations and measurement update equations for a new iteration.
Figure 3.3: Visualization of the filtering process

The system only needs to remember the conditions of the previous estimate and the current estimate. This makes it computationally viable to be implemented into devices with limited storage space such as in an active RFID tag. For localization purpose, the control parameter in the time update equations can be extended to include human walking model into it in order to better estimate tracked person.

The filter also provides flexibility in tuning the process noise and measurement noise covariance, to increase or decrease the filter’s responsiveness towards raw measurement data. For example if the tracked person is stationary, then the filter can be tuned to be less responsive so the next estimate could be more accurate. On the other hand if the tracked person is moving, then the filter could be tuned to be more responsive thus making it more accurate in that situation.
3.3 Proposed Implementation of the Kalman Filter onto LANDMARC

Is this work, Kalman filter is applied to the estimated RSSI value instead of the actual measured RSSI value to estimate the tracking tag’s location. As the filter reaches a stable state the RSSI data would be less influenced by RF Noise, and thus gives better accuracy [18] [25] [29]. Figure 3.4 below shows how the filter is added to the LANDMARC algorithm.

![Diagram showing Kalman Filter applied to RSSI used in LANDMARC](image)

Figure 3.4: Kalman Filter applied to RSSI used in LANDMARC

3.3.1 Experiment Setup

One of the Engineering Lab in Swinburne Sarawak Campus is selected as the test area. The lab is a typical open office layout with little obstacles. The Texas Instrument CC2530 active RFID tags were used. A simple setup of four RFID readers was deployed onto a 2m x 2m square test bed. The test bed is placed at the height of 1m from the floor. The reference tags comprise of 9 active RFID tags which are placed in a 0.5m x 0.5m grid at the centre of the test bed. Another 5 active RFID tags
were placed randomly and serve as tracked tags. The positioning of this setup is as illustrated in Figure 3.5 below.

![Figure 3.5: Kalman Filter LANDMARC Test bed setup](image)

The 4 RFID readers are placed at location (0, 0), (2, 0), (0, 2) and (2, 0), where the X and Y are measured in terms of meters (m). Placement of reference tags are defined as follows:

\[
Reference\ Tag\ Location = \{(x, y)|0.5m \leq x \leq 1.5m, 0.5m \leq y \leq 1.5m\}
\]

The reference tags are separated with a distance of 0.5m in both X and Y directions. The tracked tags are placed randomly within the boundaries of the reference tags. All tags are stationary throughout the experiment. The location estimation error \( e \) was calculated and the RSSI readings were recorded every 5 minutes. The results of both the original LANDMARC and the proposed LANDMARC with Kalman filter were compared.
3.3.2 Kalman Filter Parameters

A simplistic model is applied in this experiment, \( \hat{x}_k^- \) represents the estimated \( x \) and \( y \) coordinates of the tracked tags. Since all the tracked tags are stationary throughout the experiment, the control parameter \( u_{k-1} \) and process error covariance \( Q \) is eliminated. Constant \( A \) also have the value of 1 since the measurement value should remain constant as the tags are stationary. It is known that the measurement value is a composite of the state value and some noise, therefore the value of constant \( H \) is 1. The measurement noise covariance \( R \) is arbitrarily chosen as 0.1. This value controls the sensitivity of the filter towards the measured value \( z_k \). Table 3-2 shows the equations used in this experiment.

Table 3-2: Kalman Equations for RFID Experiment

<table>
<thead>
<tr>
<th>Time Update Equations</th>
<th>Measurement Update Equations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Project the state ahead</td>
<td>1. Compute the Kalman gain</td>
</tr>
<tr>
<td>( \hat{x}<em>k^- = \hat{x}</em>{k-1} )</td>
<td>( K_k = \frac{P_k^-}{P_k^- + 0.1} )</td>
</tr>
<tr>
<td>2. Project the error covariance ahead</td>
<td>2. Update estimate with measurement</td>
</tr>
<tr>
<td>( P_k^- = P_{k-1} )</td>
<td>( \hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-) )</td>
</tr>
<tr>
<td></td>
<td>3. Update the error covariance</td>
</tr>
<tr>
<td></td>
<td>( P_k = (1 - K_k)P_k^- )</td>
</tr>
</tbody>
</table>
3.3.3 Experiment Results

Figure 3.6 below shows the probability-cumulative error plot of the LANDMARC only method versus the LANDMARC with Kalman filtered. The results show that the accuracy range of the LANDMARC only method is between 0.05m to 0.5m, compared to the accuracy range of LANDMARC with Kalman filter between 0.19m to 0.32m. The accuracy variation also improves from 0.45m to 0.13m.

Figure 3.6: Cumulative plot of estimated error LANDMARC vs Kalman Filtered LANDMARC
3.3.4 Summary

The experiment shows that Kalman filtering the raw RSSI readings could reduce the impact of RF noises. It provides a more stable RSSI reading for localization computation. This idea could be applied to other RF technologies.

RFID tag is used to identify and store information about the object it is being attached to. Although it possible to use RFID for indoor localization, it requires a major infrastructure investment. LANDMARC for example, requires a large number of reference tags to be deployed in the tracked area. Other methods that leverage on existing RF infrastructure would be needed if investment into RFID infrastructure is not possible.

3.4 Wi-Fi Recent Advancement (Wi-Fi Peer to Peer)

In the early IEEE 802.11 standard, ad hoc direct device-to-device connectivity was possible. The device either acts as a Wi-Fi client or a Wi-Fi Access Point. However, this capability was never widely used. It also faces several challenges such as the lack of efficient power saving support, extended quality of service (QoS) capabilities [36] and security mechanisms [37].

3.4.1 Wi-Fi Direct

In a traditionally Wi-Fi network, a Wi-Fi client discovers and connects to Wireless Local Area Network (WLAN) through Wi-Fi Access Point (AP). Once connected, it can connect to the internet, communicate or share files with other devices connected on the same network domain. New to Wi-Fi is the Wi-Fi Direct technology which was recently developed by the Wi-Fi Alliance as standard for direct device-to-device connectivity without requiring the presence of an Access Point [35].
In an overview by Camps-Mur, Garcia-Saavedra and Serrano [10], Wi-Fi Direct was designed upon the successful Wi-Fi “infrastructure” mode. It enables the Wi-Fi devices to dynamically acts as a Wi-Fi client and a Soft-AP at the same time. This is achieved by using different frequencies or time-sharing a channel through virtualization techniques. In a Wi-Fi Direct connection, devices negotiate with each other which device to take up AP functionalities. This enables even legacy Wi-Fi devices to seamlessly connect to Wi-Fi Direct devices. Wi-Fi Direct is therefore benefited from all the enhanced QoS, power saving, and security mechanisms developed for the Wi-Fi “infrastructure” mode over the past years.

### 3.4.2 Android 4.3 Wi-Fi Scan Feature

In an attempt to improve location accuracy, Google’s android devices are equipped with a feature that uses Wi-Fi to assist in determining the device location. This is done by scanning for Wi-Fi access points and cross checking with Google’s Wi-Fi location data. However, Wi-Fi is often turned off when not connected to a network to conserve battery.

Google has added a new feature to the devices Wi-Fi settings in Android 4.3 API (Android Developer. 2013) [3]. This new feature is called “Wi-Fi scan-only mode”. This mode enables the device to scan for access points without connecting to it. This can be done even when the Wi-Fi is turned off. This feature helps in improving the overall location accuracy as well as reducing battery usage.

### 3.5 LANDMARC Wi-Fi

With the findings from the earlier Kalman filtered LANDMARC experiments and coupled with the recent advancement in Wi-Fi technology, a Wi-Fi based system seems to be a probable alternative to RFID. Smart devices these days are equipped with Wi-Fi capability. They are also equipped with
powerful CPU, which is capable of offloading computation task from the centralized server. There is no need for a centralize computation server. Nevertheless, the differences in the inner workings of Wi-Fi and RFID require a version of LANDMARC for Wi-Fi to be developed.

RFID Readers needs to transmit the information read from the tags back to a centralized base station or computer for processing. This differs from Wi-Fi as Wi-Fi APs do not transmit information back to a centralized computer. Wi-Fi clients are also equipped with a CPU which can perform the computation locally, unlike RFID tags.

In LANDMARC, the RSSI measurements of the reference tags and the tracked tags to the readers are compared at the base station to determine their location relationship. The location relationship between reference tag and tracking tag are defined by the Euclidian distance in RSSI between them. Since RFID tags do not interact with each other, this method is used to determine their location relationship.

![Diagram showing the location relationship between RFID and Wi-Fi](image)

Figure 3.7: Location relationship RFID vs Wi-Fi
By means of Wi-Fi Direct, Wi-Fi Direct enabled devices can communicate with each other directly without the need of a Wi-Fi AP. Even in traditional Wi-Fi infrastructure mode, clients can establish connection with each other via ad-hoc connection where one client serves as an AP and the other a client. The RSSI between the devices can be directly defined as the location relationship between them. The estimated distance between them can be calculated using the ITU indoor propagation model. Figure 3.7 above visualizes the location relationship differences between RFID and Wi-Fi.

The proposed Wi-Fi version of LANDMARC in this thesis can be modelled by replacing the Euclidian distance in RSSI with the estimated distance from the ITU indoor propagation model. The estimated distance is computed from the measured RSSI using the ITU indoor propagation model. In a study by Chrysikos, Georgopoulos, Birkos and Kotsopoulos (2009) [12] it is stated that the ITU model provides a reliable prediction for single floor modelling. This study also found this to fit well with the RSSI distance measured in the indoor propagation model experiments carried out earlier, and thus it is used as a basis for distance estimation.

Referring to Chapter 2.3.2, the estimated Distance $d$ can then be calculated as below.

$$d = 10^{\frac{L - 20 \log f - P_f(n) + 28}{N}}$$

In [31], it is stated that the total path loss $L$, can also be express as $L = P_{TX} - P_{RX}$ where $P_{TX}$ is the transmission power in dBm and $P_{RX}$ is the RSSI reading in dBm. The transmission power can be calculated from the manufacturer’s datasheet using the transmission output power and antenna gain specifications. Assumptions were made earlier in this study that all clients have the same antenna gain, there for the antenna loss is negligible.
After the estimated distance is calculated, the algorithm proceeds with the LANDMARC’s k-nearest neighbour algorithm to determine the k nearest neighbour. Similar to LANDMARC, 4 of the nearest reference points is selected. In this case, 4 Wi-Fi Fix-APs would be selected as reference point.

Next the weighting factor is computed from the 4 reference point, by replacing the Euclidian distance $E$ in the LANDMARC’s version with the estimated distance $d$. The weighting factor $w_i$ can be defined as:

$$ w_i = \frac{1}{d_i^2} $$

$$ \sum_{i=1}^{k} \frac{1}{d_i^2} $$

Finally, the estimated location is computed in a similar way as LANDMARC. The estimated location can be computed by solving:

$$(x, y) = \sum_{i=1}^{k} w_i(x_i, y_i)$$

Figure 3.8: LANDMARC for Wi-Fi model

Figure 3.8 shows the overall LANDMARC Wi-Fi model. The experimental setup is discussed in the next section.
3.5.1 LANDMARC Wi-Fi Experiments

In order to validate the LANDMARC Wi-Fi model, a series of experiments were carried out in the Swinburne Sarawak Research Centre. This location is an open office area with some obstacles such as furniture, glass, small offices, room partitions and carpeted flooring.

The experiment consists of a single smartphone (Google-Nexus 4), a wireless router (TPLink-TLWR740N) and three wireless repeaters (TPLink-TLWA701N). The wireless router and repeaters were placed at fixed locations throughout the centre. Figure 3.9 below shows where the router and repeaters are placed. There is an existing ‘Swinwifi’ wireless router (Cisco AIR-SAP2602EZK9) in the research centre which is also used in the experiments.

![Figure 3.9: Wi-Fi Router and Repeater placement in Swinburne Research Centre](image)

The routers and wireless repeaters are used as landmarks for the experiments, as in the original LANDMARC’s reference tags. The smartphone is programmed with an application which scans all
Wi-Fi Fix-APs and collect RSSI data from those APs. The Kalman filter is also programmed into the application to filter the RSSI readings. LANDMARC Wi-Fi algorithm is then performed to obtain an estimated location. The estimated location is compared with actual location to determine the performance of the method. The experiment is carried out with both raw RSSI and filtered RSSI. The raw RSSI, filtered RSSI, and estimated locations are stored on a file in the smartphone and can be extracted for further analysis.

Figure 3.10: Test Locations in Swinburne Research Centre

Figure 3.10 shows the test locations where the experiments are carried out. The selection of these test locations is done randomly. The APs’ positions are placed randomly instead of a grid to simulate a real world environment.
3.5.2 LANDMARC Wi-Fi Results Analysis

The graph on Figure 3.11 below shows the average estimation error of LANDMARC Wi-Fi using both Kalman filtered and unfiltered RSSI data. The location estimated for test locations 1 to 6 and 10 shows inaccurate estimation for both filtered and unfiltered with an error of over 5 meters. Test locations 7 to 9 shows a better accuracy at around 1 meter of estimation error, and site 11 shows average accuracy at an average estimation error of slightly above 3 meters. Test location 12 is located behind a concrete wall to a nearest AP. The unfiltered LANDMARC Wi-Fi shows poor accuracy with an average estimation error at over 9 meters. However for the filtered LANDMARC for Wi-Fi shows average accuracy with an average estimation error of around 2 meters. This shows that the Kalman filter is working to filter out the RF noises which affect the RSSI readings. The error covariance $P_k$ and the measurement noise covariance $R$ of the Kalman filter converges the estimated measurement to it’s true value over a few iterations that cannot be achieved by simple averaging.

![Average Estimation Error Filtered vs Unfiltered LANDMARC](image)

**Figure 3.11: Average Estimation Error of Filtered and Unfiltered LANDMARC for Wi-Fi**
From this experiment, it was noted that both filtered and unfiltered LANDMARC cannot function outside the perimeter of the reference points. In the case of LANDMARC Wi-Fi, it is the Fix-APs. Figure 3.12 shows the coverage area of the reference points in yellow. Test locations outside of the area would result in a wrong estimated location.

![Diagram of LANDMARC Wi-Fi coverage area](image)

**Figure 3.12**: LANDMARC Wi-Fi coverage area highlighted in yellow

The estimated location is calculated based on the total of the weighted reference point coordinates. \((x, y) = \sum_{i=1}^{k} w_i(x_i, y_i)\). Therefore the estimated location would never be outside the perimeter formed by the selected reference points, since the weights \(w_i = \frac{1}{d_i^2} / \sum_{i=1}^{k} \frac{1}{d_i^2}\) are a fraction instead of a whole number.

On the other hand, location 8 and 11 which fall inside the perimeter give acceptable results. It was found that location 7 and 9 also give significantly accurate results as they are very close to the APs. Location 12 also gives a favourable result for the filtered version of LANDMARC Wi-Fi.
From the results of the experiments, LANDMARC Wi-Fi can be a viable solution for indoor localization. It can be said that if the tracked client is near an AP, such as in test location 8, good accuracy can be achieved. However, if the tracked client is far from any AP, such as in location 11 then acceptable accuracy cannot be achieved.

It was found that if the tracked client is within 1 or 2 meters from the AP and it is outside of the perimeter, such as test location 7, 9 and 12, its estimated location is pulled nearer to the AP, and thus giving an acceptable accuracy. This however is not ideal as it does not reflect the actual distance of the tracked client from the AP, as predicted by the ITU indoor propagation model.

The actual distance and the predicted distance from the ITU model are very similar when the client is near to the AP. But due to the weighting factor of LANDMARC Wi-Fi, the estimated location is pulled closer to the closest AP. Earlier in this chapter, it was noted that the correlation between the actual distance and the ITU model distance is very strong up to 5 meters. Beyond 5 meters the RSSI reading becomes erratic even when filtered. Therefore, in this study it can be concluded that the usable range for localization purpose is only up to 5 meters.
3.5.3 LANDMARC Wi-Fi Conclusion

Although LANDMARC Wi-Fi can be a viable solution, in a real world there is only one AP which provides internet service to all the clients in the area. It is not practical to setup multiple APs in the same area, as it would be a waste of resources. An alternative is to use the readily smartphones in everyone’s pocket as a Soft-AP if its location is known when it is near to an AP.

The estimated distance and the ITU modelled distance are very different due to the way that LANDMARC works. It does not reflect the actual distance as predicted by the ITU model.

3.6 Summary

This study finds that the ITU indoor propagation model fits well with the RSSI distance measured. The ITU indoor propagation model will be used as a basis for distance estimation in the next chapters. It is also found that by filtering raw RSSI readings with Kalman filter, the effects of RF noise can be reduced. The proposed LANDMARC Wi-Fi algorithm is not suitable to be used with Wi-Fi as it requires multiple APs to be setup in the same area for acceptable accuracy.

In the next chapter, a new collaborative trilateration method which leverages on close proximity clients is proposed.
Chapter 4. Proposed Wi-Fi Based Collaborative Trilateration Method

This chapter focuses on the details of a proposed collaborative trilateration method. The general mechanism of the trilateration method, which incorporates the findings discussed in Chapter 3, is discussed. The proposed alternative method of trilateration-triangulation is then explained. Finally, details of the proposed collaborative trilateration method for indoor localization will be discussed. Portions of this chapter had been published. [27]

4.1 Proposed Alternative Trilateration-Triangulation Method

Chapter 3 shows a strong correlation between the ITU models estimated distance to the actual distance between 1 to 5 meters. This study proposes an alternative trilateration-triangulation method which would have the tracked client anchored to the closest reference point. Therefore the position of the tracked client is located around the closest reference point with a radius between 1 to 5 meters. This specific range is used because of the strong correlation between the ITU models estimated distance and the actual distance. In order to triangulate the location, three reference points are required in this method. The selection of these reference points will be explained later in this chapter.
The tracked client’s estimated coordinate \((x_n, y_n)\) is:

\[
x_n = X_1 + R1 \cos(\theta) \quad \therefore 0 \leq \theta \leq 2\pi \tag{1}
\]

\[
y_n = Y_1 + R1 \sin(\theta) \quad \therefore 0 \leq \theta \leq 2\pi \tag{2}
\]

where \(n\) is the unique identifier of the set coordinates, \(R1\) is the estimated distance between the tracked client and the closest reference point, and \((X_1, Y_1)\) are the coordinates of the closest reference point. Theta \(\theta\) may range from 0 to \(2\pi\) radians or from 0 to 359 degrees.

The first part of this method determines all possible coordinates for the tracked client around the circumference of the closest reference point. In order to reduce the list of potential location for the tracked client, equation 1 and 2 can be solved by incrementing theta \(\theta\) by \(\frac{\pi}{180^\circ}\) or 1 degree. The variance between each set of coordinate, is 8.73 cm for every increment of 1 degree at the maximum radius of 5 meters, this is an acceptable margin of error. Figure 4.1 below, visualizes the parameters of the margin of error between two sets of coordinates. At the end of this step there would be 360 sets of coordinates, with the radius of the estimated distance \(R1\) from the closest reference point.

\[
\begin{align*}
(5, 0) & \quad \theta = 1^\circ \\
5m & \quad 5m \\
8.73\, \text{cm} & \quad (4.999, 0.087)
\end{align*}
\]

Figure 4.1: One degree margin of error
The next step involves determining the coordinates which are inside the perimeter formed by the three reference points. Figure 4.2 below shows the triangle formed by P₁, P₂ and P₃. If P₄ is the tracked client, then P₄' is the mirror of P₄ on the other side of the vector \( \overrightarrow{P₁P₂} \).

![Figure 4.2: Point in triangle test](image)

In order for a set of coordinate to be within the triangle \( P₁P₂P₃ \), it must meet all the following criteria:

- right of the vector \( \overrightarrow{P₁P₂} \) formed by P₁ and P₂
- above the vector \( \overrightarrow{P₁P₃} \) formed by P₁ and P₃
- left of the vector \( \overrightarrow{P₂P₃} \) formed by P₂ and P₃

The resulting cross product of vector \( \overrightarrow{P₁P₂} \) and vector \( \overrightarrow{P₁P₄} \), and the resulting cross product of vector \( \overrightarrow{P₁P₂} \) and vector \( \overrightarrow{P₁P₄'} \) will be in opposing direction because they are in opposing sides of vector \( \overrightarrow{P₁P₂} \). This information can be used to determine the side of which the point of interest falls in. However because the triangle can be oriented in multiple ways in 3D-Space, there is no other fix value to compare with except the third point of the triangle, which is on the correct side of the vectors. Thus the resulting cross product of vector \( \overrightarrow{P₁P₂} \) and vector \( \overrightarrow{P₁P₄} \), and the resulting cross product of vector \( \overrightarrow{P₁P₂} \) and vector \( \overrightarrow{P₁P₃} \) will be in same direction, if the dot product of these two
cross products are greater than or equal to zero. Otherwise, if the dot product is less than zero then they are not in the same direction, hence it will be outside of the triangle. If the cross products are in the same direction, then this test would need to be repeated for the other two vectors in the triangle. \( P_4 \) is set to be in the triangle, if its cross product with vector \( \overrightarrow{P_1P_2} \) is in the same direction as \( P_3 \), and its cross product with vector \( \overrightarrow{P_1P_3} \) in the same direction as \( P_2 \), and also with vector \( \overrightarrow{P_2P_3} \) in the same direction as \( P_1 \).

At the end of this step, only a fraction of the 360 sets of coordinates will be within the triangle formed, which is shown in the green outline in Figure 4.3. The yellow triangle is the active region formed by the chosen 3 reference points.

Figure 4.3: Visualisation of the proposed Trilateration-Triangulation Method

In an ideal situation, the estimated distances calculated from RSSI, \( R^2 \) and \( R^3 \) from the second \((X_2, Y_2)\) and third \((X_3, Y_3)\) reference point, should be similar to the Euclidean distance calculated from the set of coordinates which are in the triangle. However, due to RF noise which causes a
variation in the measured radio signal strength, there exist differences between the estimated distance and the calculated Euclidean distance.

The last step in this method is to determine which set of coordinate will produce the least distance variation error between the estimated distance calculated from RSSI and the Euclidean distance calculated from the sets of coordinates. These are captured as:

\[ E_2 = |R2 - \sqrt{(x_n - X_2)^2 + (y_n - Y_2)^2}| \]  
\[ E_3 = |R3 - \sqrt{(x_n - X_3)^2 + (y_n - Y_3)^2}| \]

\( E_2 \) and \( E_3 \) are the distance variation error to the second and third reference.

The total distance variation error \( E \) for a set of coordinates is given by:

\[ E = q_2 E_2 + q_3 E_3 \]

where \( q_2 \) and \( q_3 \) is the weighting factor calculated from the estimated distance from the RSSI, which is defined as:

\[ q_i = \frac{1}{R_i^2} \left( \frac{1}{R_2^2} + \frac{1}{R_3^2} \right)^{-1} \]

The weighting factor serves as a mechanism to minimize the impact from the third and furthest reference point to the total distance variation error. This would improve the overall accuracy of the estimated coordinate of the tracked client.

The estimated coordinate of the tracked client is then given by the set of coordinate which meets all the following criteria.

1. Is within the triangle formed by the 3 reference points
2. Has the least total distance variation error
Figure 4.4: Proposed Trilateration-Triangulation Method process flow

Figure 4.4 show the overall process flow of this method. This method of trilateration-triangulation ensures that there is a solution even when the three circles formed by the reference point do not intersect or are contained within one another, as the solution is anchored to the first reference point which fulfils all the selection criteria.

A noticeable challenge to this method is that, only clients who are within the 5 meters radius of a Fix-AP can be located. In order to extend this accuracy range this study proposed an indoor localisation method which uses this proposed alternative trilateration-triangulation method and Wi-Fi Direct enabled devices which can serve as Soft-AP.
4.2 Proposed Collaborative Trilateration Method

By applying the method proposed in section 4.1, this study proposed a collaborative trilateration method for indoor localization. The iterative process of the Kalman filter can be applied to the estimated location to improve on the overall accuracy of the estimated location. The proposed method can be broken down into three general processes, ‘distance estimation’, ‘reference point selection’ and ‘trilateration-triangulation and filtering’. The iterative process will continue until a high confidence level on the estimated location is reached. The confidence level is derived from a parameter of the Kalman filter, which will be explained later in chapter 4.2.3. Figure 4.5 shows the process flow of the iterative process.

![Figure 4.5: Collaborative Trilateration Method Process flow](image)
At any point of time, when the client has a high confidence level on its estimated location, it is then qualified to be a Soft-AP in the coverage area. This can be achieved with either Wi-Fi Direct technology or with traditional ad-hoc Wi-Fi peer to peer. The Soft-AP can then serves as a reference for others to estimate their own location. This can help improve the accuracy of the estimated location for clients who are further than 5 meters from any Fix-AP. The coverage area is defined as the area formed by the Fix-APs as mentioned in chapter 3. Clients outside of this area cannot be accurately located. Therefore it is assumed that all tracked clients are within the coverage area.

4.2.1 Distance Estimation
The estimated distance is used as a measure between the tracked client to all Fix-AP and known Soft-AP within the coverage area. This distance is calculated from RSSI using the ITU indoor propagation model as previously explained in chapter 3. After the estimated distance are calculated, they are then sorted from the nearest to the furthest.

4.2.2 Reference Point Selection
In the real world, Fix-APs are not placed in a grid like manner or close to one another, they are spread out strategically to maximise Wi-Fi network coverage. This is not ideal for indoor localization, as the RSSI readings fluctuate greatly at distance greater than 5 meters from the source. The reference point selection process is a critical step in the overall method, as are used as alignment points for the trilateration-triangulation process.

In the initial state where the tracked client does not have an estimated coordinate, three of the nearest Fix-APs are selected to be used in the proposed alternative trilateration-triangulation method to obtain an initial estimate. This initial estimate is used to determine the quadrant information which will be used in the reference point selection criteria for later iterations.
A. Quadrant Information

Quadrant information can be calculated based on the coordinates of the reference points and the estimated coordinate of the tracked client. Taking the track client’s coordinate \((x_0, y_0)\) as origin, and the \(n\)-th reference point coordinate \((X_n, Y_n)\), the quadrant information can be calculated by solving:

\[
\Delta X = X_n - x_0 \quad \text{(7)}
\]
\[
\Delta Y = Y_n - y_0 \quad \text{(8)}
\]

The reference point is set to be in a specific quadrant based on the results of delta \(X\), \(\Delta X\) and delta \(Y\), \(\Delta Y\) from equation (7) and (8). The table below shows the summary of the quadrant information.

<table>
<thead>
<tr>
<th>(\Delta X)</th>
<th>(\Delta Y)</th>
<th>Quadrant</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 0</td>
<td>≥ 0</td>
<td>1</td>
</tr>
<tr>
<td>≤ 0</td>
<td>&gt; 0</td>
<td>2</td>
</tr>
<tr>
<td>&lt; 0</td>
<td>≤ 0</td>
<td>3</td>
</tr>
<tr>
<td>≥ 0</td>
<td>&lt; 0</td>
<td>4</td>
</tr>
</tbody>
</table>

After the initial estimate and all the quadrant information is obtained, the next step in the method is to select the reference points. These can be either Fix-AP or Soft-AP. The quadrant information is used as a criterion for reference point selection. This is to ensure that the points selected will form a triangle and the tracked client is within this triangle.
B. First Reference Point

The first reference point \((X_1, Y_1)\) is the key to the overall accuracy of the proposed method, as the estimated coordinate will be anchored to this point. The highest priority is to select a Fix-AP as reference. Therefore the nearest Fix-AP with the estimated distance, \(d_1\) of 5 meters or less is to be selected.

In the event when the distance to the nearest Fix-AP is more than 5 meters, then the nearest known Soft-AP which has an estimated distance of 5 meters or less and in a different quadrant from the nearest Fix-AP will be used as the first reference point instead. The nearest Fix-AP will then be selected as the second reference point \((X_2, Y_2)\), the algorithm would then proceed to select the third reference point.

If both the above mention criteria is not meet, then the estimated distance to the nearest Fix-AP or known Soft-AP is more than 5 meters. In this case the nearest Fix-AP is selected as the first reference point, regardless of the distance. In this situation, the tracked client will have a low confidence level rating, on its location as the estimated location will not be filtered.
C. Second Reference Point

Recent advancement of Wi-Fi and Wi-Fi Direct as discussed in chapter 3 can be used on Wi-Fi enabled mobile devices such as smart phones, tablets or personal Wi-Fi modules. These devices can serve as Soft-AP if they are within 5 meters of a Fix-AP and their location is known. In this study, it is assumed that these mobile devices are stationary after their locations are confidently known.

This would extend the accuracy range of the Fix-AP by using the Soft-AP as leverage. Therefore the nearest known Soft-AP which is in a different quadrant from the first reference point is selected as the second reference point \((X_2, Y_2)\).

In the event where there is no known Soft-AP near to the tracked client, then the next nearest Fix-AP which is also in a different quadrant from the first reference point will be selected as the second reference point. The estimated distance to the second reference point is given by \(d_2\).
D. Third Reference Point

The criteria used to select the third reference point \((X_3, Y_3)\) is the same criteria as the second reference point, which is the next nearest known Soft-AP. Additionally the selected reference point has to be in a different quadrant from the first and second reference points.

Similarly, in the event that there is no known Soft-AP, then the third nearest Fix-AP is selected as the reference point. The estimated distance to the third reference point is then given by \(d_3\).

![Figure 4.6: Visualization of selected reference points](image)

The reference points are selected in this manner is so that the estimated coordinate of the tracked client is located within the triangle formed by the selected reference points, and anchored to the first reference point as shown in Figure 4.6 above.
4.2.3 Trilateration-Triangulation and Filtering

Once the three reference points are selected, the estimated coordinate would then be calculated using the proposed alternative trilateration-triangulation method. Figure 4.7 below shows the parameters of the proposed alternative trilateration-triangulation method with the selected reference points.

![Diagram of trilateration method](image)

**Figure 4.7: Visualization of the trilateration method.**

In order to determine the viability of the estimated coordinate, an algorithm to determine the confidence level of estimation is applied. This confidence level algorithm is based on the Kalman filter. However not all estimated coordinates need to be filtered as the filtering process can be computationally taxing on the mobile devices.
Empirically, an estimated coordinated is deemed to be accurate if it was located within 5 meters from a Fix-AP or known Soft-AP. Coordinates derived from least favourable reference points set, where the first reference point is greater than 5 meters away, can be ignored as the estimated coordinate is likely to be inaccurate. Doing so will ensure that computing resources will not be wasted.

The Kalman filter is only applied to the estimated coordinate if the estimated distance to the first reference point, \(d_1\) is 5 meters or less. In subsequent iteration, the filter is only applied if the same first reference point was selected again. Otherwise the estimated position will not be filtered.

In the first iteration, the tracked client’s initial estimated location is feed to the time update equations of the Kalman Filter (9). An initial value of ‘1’ is selected for the estimated error covariance \(P\), constant \(A, B\) and \(H\). All tracked clients are assumed to be stationary throughout the entire process. Therefore the control parameter \(u\) and process noise \(Q\) in (10) is assumed to be zero.

\[
\hat{x}_k^- = A\hat{x}_{k-1}^- + Bu_{k-1} \tag{9}
\]

\[
P_k^- = P_{k-1}^- + Q \tag{10}
\]

In subsequent iterations, the estimated location is fed into measurement update equations of the Kalman filter. The Kalman gain (11) is calculated and the estimated location (12) and estimate error covariance (13) are updated. The entire process repeats itself and updates equations (9) and (10) for the next iteration.

\[
K_k = P_k^-H^T(HP_k^-H^T + R)^{-1} \tag{11}
\]

\[
\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{12}
\]

\[
P_k = (I-K_kH)P_k^- \tag{13}
\]
The estimated error covariance $P$ is being minimized by the Kalman Gain ($K$) when the residue $(z_k - H\hat{x}_k)$ from equation (12) is in complete agreement or zero. Hence when the previous estimated and new estimated location is in agreement, it can be concluded that the system has reach a stable state.

In order to rate the confidence level on the accuracy of the estimated position, the Kalman gain ($K$) from equation (11) is used as a measure of confidence, as it is derived from the estimated error covariance. Table 4-2 shows the confidence level on the accuracy of the estimated coordinate of the tracked client based on Kalman gain.

<table>
<thead>
<tr>
<th>Kalman Gain ($K$)</th>
<th>Confidence Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K \leq 0.050$</td>
<td>High</td>
</tr>
<tr>
<td>$0.050 &lt; K \leq 0.075$</td>
<td>Medium</td>
</tr>
<tr>
<td>$K \geq 0.100$</td>
<td>Low</td>
</tr>
</tbody>
</table>

Tracked clients with a high level of confidence are deemed to be accurately located and their location known. They will be qualified to be used as by other clients as Soft-AP assuming that they are stationary. Once the known client moves they will no longer be qualified to be used as Soft-AP.
4.3 Summary

In this chapter, a collaborative trilateration method for indoor localization is proposed. The method leverages on the current Wi-Fi infrastructure and recent advancements in Wi-Fi Direct for this purpose. The Kalman filter is used to minimise the effects of RF noises, and as a measure of confidence in the estimated coordinates.

The next chapter will discuss on the simulations of the proposed method for indoor localization. It will also discuss on the design of experiments and the results obtained.
Chapter 5. Simulations & Discussions

In this chapter, the simulations using real data carried out using the proposed collaborative trilateration method for indoor localization will be discussed. The limitations and the design of the simulation will be explained and discussed. Next the findings and results of the simulations and the effects of the parameters in the simulations are explained. Portions of this chapter had been published in [27].

5.1 The Set Up

In order to validate the proposed iterative trilateration-triangulation method for indoor localization, simulation using real RSSI data collected from wireless routers, repeaters and smartphones was carried out. The university’s Multi-Purpose Hall was selected as test site. RSSI readings were recorded using a smart phone at different locations around the router and repeaters in the hall.
Figure 5.1: RSSI data collection positions.

The data were collected at a distance of 1 to 10 meters from the router and repeaters, with an interval of 1 meter between each set of readings. At each location, a set of 100 RSSI readings were collected, similar to experiments carried out in chapter 3. However, in order to collect data from every direction around the router and repeaters, 8 positions were identified. They are north, south, east, west, northeast, northwest, southeast, and southwest. Figure 5.1 shows the location where the readings were taken. A total of 800 data from every direction for each interval were collected.
A similar process was repeated for the smartphone, which was set in tethering mode. RSSI readings were recorded from a laptop which was carried around the different locations around the smartphone. This RSSI data is to simulate the Soft-APs RSSI data.

5.1.1 Design of Simulations

The experiments were design to study the effects of Fix-AP and Soft-AP density, Fix-AP layout as well as the Kalman filter’s responses to measurements. The LANDMARC Wi-Fi was also simulated for comparison.

In the design of experiments, 20 random tracked clients were simulated to be deployed in a 400 m² square area. This is so that, sufficient density of Soft-AP is reached in the simulated area. The tracked clients were introduced 5 at a time after 30 iterations. This is to allow the clients to obtain stable location estimation and to observe the effects of Soft-AP density on the accuracy of the proposed method.

![Figure 5.2: Simulated Random Clients](image-url)
Figure 5.2 shows the distribution of the 20 random tracked clients which were used in the simulations. Clients 1 to 5 were introduced at the beginning of the simulation. After 30 iterations, clients 5 to 10 were then introduced. Clients 11 to 15 were introduced 30 iterations after them. Finally clients 16 to 20 were introduced 30 iterations later. The simulation will continue for 30 more iteration for clients 16 to 20 to obtain a stable estimation. This is to simulate the effects of client density within the track area. In the beginning of the simulation the density is low, and as the simulation progresses known soft-AP are slowly being introduced into the system.

Once the location of the tracked client has been successfully estimated and a high confidence level is achieved, it is then upgraded to a Soft-AP. Newly introduced clients or clients who have not been located would then be able to use these Soft-AP as reference points in their own localization process. This would simulate the intended real world application where tracked clients are constantly changing and if a client were to be stationary for a period of time, it would then be confidently located and can serve as a Soft-AP for others.
5.1.2 Simulations

The simulations start by setting up the Fix-AP positions. The positions of the Fix-APs are known to the system. Five random clients are then added to the area. The actual coordinates of these clients are \((x_{n0}, y_{n0})\) where \(n\) is the identifier of each client. The distance between each client to the Fix-AP is calculated. This distance is then rounded and used as an index to lookup for a random RSSI reading from the RSSI database. This is to simulate the natural behaviour of the raw RSSI measurement under different conditions.

It was observed from the raw RSSI data collected, that the RSSI value beyond 10 meters from Fix-APs and Soft-APs began to behave erratically and becomes unpredictable. Hence in the simulations, lookup index beyond 10 are considered to be too far from the Fix-AP or Soft-AP and assigned a RSSI reading of -99 dBm.

Based on the RSSI data, the iterative process then begins with the selection of reference points, and computing the estimated position \((x_n, y_n)\) of each tracked client. The Kalman filtering process is applied if the conditions stated in chapter 4 are met. In the following iteration, the RSSI value is refreshed with a new random RSSI value from the RSSI database. This is also to simulate the behaviour of the RSSI reading from raw RSSI measurement. Figure 5.3 below shows the overall simulation process.
A series of simulations were conducted in to validate the proposed collaborative trilateration method. Four sets of deployment of Fix-AP were simulated to study the effects of Fix-AP deployment on the accuracy of the method. Figure 5.4 shows the different deployment used in the simulations.
a. 4 Fix-AP at 4 corners of coverage area
b. 5 Fix-AP at 4 corners and centre of coverage area.
c. 9 Fix-AP spread out in 10 m by 10 m grid.
d. 25 Fix-AP spread out in 5 m by 5 m grid

Figure 5.4: Fix-AP deployments
In the first simulation setup four Fix-APs were placed at the corners in the 20m by 20m coverage area. In a design guideline by Cisco (2014) [14], it is stated that it would be best to maintain a spacing of above 8.53 m or 28 feet between Fix-AP. This is an ideal deployment for signal coverage and Wi-Fi location based services. The second setup consists of 5 Fix-APs, similar to the first setup with an additional Fix-AP at the centre of the coverage area. This is to study the effect of the centre Fix-AP.

The third and fourth setup consists of 9 and 25 Fix-APs which were deployed in a 10m x 10m grid and a 5m x 5m grid. These setups are ideal for localization but are too extensive for signal coverage. It is suggested in a troubleshooting guideline by Cisco (2010) [13] that too many Fix-AP in the same area can cause radio congestion and interference. This will also reduce the throughput of the network. The third and fourth setup is for comparison with other methods which relies heavily on dense landmark deployment like LANDMARC.
5.2 Simulation Results and Analysis

5.2.1 Comparison with LANDMARC

The results show that the proposed method has a slight improvement in accuracy over the LANDMARC Wi-Fi which depends heavily on known reference points. Figure 5.5 below shows the probability plot of the estimation error for the proposed Collaborative Trilateration Method versus LANDMARC Wi-Fi using the 25 Fix-APs grid setup which is ideal for localization. Ideally the plots should have a narrow estimation error range and small estimation error. LANDMARC can achieve an average estimation error of around 1.24 meters, as shown in the black plot. On the other hand, the proposed method can achieve an average estimation error of around 0.89 meters. This is a performance gain of roughly 25%.

Figure 5.5: Probability plot of Estimation Error Collaborative Trilateration Method vs LANDMARC – 25AP.
On the other hand, with the 4 Fix-AP at the corner setup which is ideal for signal coverage and location based services, the Collaborative Trilateration Method performs significantly better than LANDMARC, as shown in Figure 5.6. The average estimation error for LANDMARC is about 5.61 meters, whereas for the proposed method it is about 2.85 meters. This is a performance gain of more than 49%. This shows that the proposed method is effective even though there is significantly less known reference point. This shows that the proposed method which relies on the known Soft-AP to extend the effective range of Fix-AP is working as theorised.

![Figure 5.6: Probability plot of Estimation Error Collaborative Trilateration Method vs LANDMARC – 4AP.](image)

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>StDev</th>
<th>N</th>
<th>AD</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>LANDMARC Wi-Fi</td>
<td>5.612</td>
<td>0.3507</td>
<td>120</td>
<td>5.869</td>
<td>&lt;0.005</td>
</tr>
<tr>
<td>Collaborative Trilateration</td>
<td>2.848</td>
<td>0.6957</td>
<td>120</td>
<td>9.636</td>
<td>&lt;0.005</td>
</tr>
</tbody>
</table>

N – Sample Size
AD – Anderson Darling Test
P - pValue
5.2.2 Effects of Soft-AP Density

The density of Soft-AP is a key variable in the proposed method, as it can improve or deteriorate the accuracy of the estimated location. This is because the overall system relies heavily on the initial accuracy of the Soft-AP estimated location. If the estimated location of the Soft-AP is erroneous, it would deteriorate the estimation accuracy of other unknown clients near to it. On the other hand, if the estimated location of the known Soft-AP is acceptably accurate, then it would improve the accuracy of other unknown clients which are near to it.

![Figure 5.7: Clients colour sets](image)

Figure 5.7 shows the colour sets of each set of clients. Clients 1 to 5 are blue, clients 6 to 10 are red, clients 11 to 15 are green and finally clients 16 to 20 are brown.
Figure 5.8 below shows the estimation error performance of selected tracked clients with the ‘4 Fix-AP setup’ at the corner. In the first 30 iterations the estimated location of clients 1 to 5 cannot be determined, as shown by the blue shaded plots. This is because they do not have sufficient reference points which are nearby to be used to determine their location.

![Figure 5.8: Estimation error selected clients over 120 iterations](image)

After clients 6 to 10 were introduced, and their location deemed accurately located, they serve as Soft-AP for clients 1 to 5. Clients 1 to 5 would then use the newly known Soft-AP as reference points to estimate their location. The graph in Figure 5.8 shows that client 3’s estimation error begins to improve after clients 6 to 10 were introduced as shown by the red shaded plots. The accuracy of the estimated location of client 1 to 5 improves over subsequent iterations.

Overall, as clients 11 to 15 and 16 to 20 were introduced into the scene, shown by the green and brown shaded plots, the general accuracy of the estimated locations of all clients can be seen as improving. However, at the individual client level, some estimations may deteriorate.
5.2.3 Effects of Fix-AP Density

RF noise is a major limiting factor of using RF technology to locate and track clients. In order to overcome this factor, the density of the Fix-AP in the tracked area can be increased to create a setup which is ideal for localization.

The density and the deployment of the Fix-AP greatly affect the accuracy of the proposed Collaborative Trilateration Method. In the simulation, it was found that the accuracy of the method improves significantly as the density of the Fix-AP was increased. Figure 5.9 below shows the probability plot of the estimated location error for the different Fix-AP deployment. However, increasing Fix-AP will incur additional cost without increasing signal coverage in the tracked area. In fact too many Fix-AP in the same area can cause radio congestion and interference [13].

![Probability Plot of 4AP, 5AP, 9AP, 25AP](image)

Figure 5.9: Probability plot of Estimation Error Collaborative Trilateration Method 4AP vs 5AP vs 9AP vs 25AP
The average estimation error ranges from 2.4 meters with the 4 Fix-APs setup to about 0.9 meter with 25 Fix-APs setup. A similar trend is observed when the measurement error covariance of the Kalman filter is tuned to slow down the response of the filter as shown in Figure 5.10 below.

![Probability Plot of 4AP, 5AP, 9AP, 25AP](image)

**Figure 5.10:** Probability plot of Estimation Error Collaborative Trilateration Method 4AP vs 5AP vs 9AP vs 25AP

The proposed collaborative trilateration method works better when the filter is tuned to have a slower response. It is less susceptible to RF noises, as shown in Figure 5.10. The plots do not overlap each other as in Figure 5.9, and a clear improvement can be observed as the Fix-AP deployment is increased.
5.3 Summary

The simulations have shown that the density of Soft-AP is critical to the proposed method. At low Soft-AP density, the location of the tracked client can only be determined if it is within certain distance, depending on the correlation of the RSSI estimated distance to the Fix-AP. Generally, high Fix-AP density is good for the overall accuracy. However, it is not practical as the implementation cost is high and it would not improve on the network coverage. On the other hand, this can be compensated with high Soft-AP density, for example in a crowded space with many tracked clients or human around. With sufficient Soft-AP, the proposed method can achieve good accuracy even with low Fix-AP density.
Chapter 6. Conclusion and Recommendations

The principal purpose of this study was to design and develop a RF-Based indoor positioning system which:

a. is accurate
b. leverages on existing infrastructure to reduce cost
c. is robust and adaptable to the indoor environment
d. flexible and scalable

6.1 High Accuracy

The proposed Collaborative Trilateration Method was able to produce an acceptable accuracy. The simulations with the lowest Fix-AP density shows that the proposed system can achieve an accuracy of roughly 2.5 meters. The idea of using Wi-Fi Direct and Soft-AP as additional reference points proves to be a major contributor in improving the estimation accuracy. This is an acceptable accuracy as Wing, M.G. (2011) noted that the highest consumer grade GPS has an average error of 1.5 meters under clear skies [38].

6.2 Flexible and Scalable

The system is more robust when there is a higher density of tracked clients or Soft-AP. However with low density of Soft-AP, the system is not robust, as only the estimated location of clients who
are very near to a Fix-AP will be accurate. Additional sensors such as MEMS would be needed in this scenario to improve robustness of the system.

The tracked indoor area can be easily extended by adding another Fix-AP into it and updating the community shared Fix-AP database with its exact location information.

6.3 Recommendations for Future Works

The following recommendations are offered as possible ways to improve this study.

1. Although simulated using actual RSSI data, live experiments should be carried out using multiple Fix-AP and user devices the determine the performance and viability of the proposed Collaborative Trilateration Method with moving tracked clients; and

2. A walking model can be implemented into the system to track the movement of the client within the indoor environment. This can be done using the control parameter of the Kalman filter. Further integration with microelectromechanical systems (MEMS) sensors can be applied to accurately determine the location of the user while walking.
Reference


[14] Cisco. 2014. Wi-Fi Location-Based Services 4.1 Design Guide. [ONLINE].
Available at:


[27] Liu, L. and Wong, W., 2015, June. Indoor positioning through an iterative method in dense Wi-Fi-direct networks with Wi-Fi direct devices. In Consumer Electronics-Taiwan (ICCE-TW), 2015 IEEE International Conference on (pp. 308-309). IEEE.


Appendix A: MathLab Source Codes

%IMA_GUI M-file for IMA_GUI.fig
%   IMA_GUI, GUI for Simulation.
%
% Begin initialization code - DO NOT EDIT

function varargout = IMA_GUI(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @IMA_GUI_OpeningFcn, ...
    ' gui_OutputFcn', @IMA_GUI_OutputFcn, ...
    ' gui/LayoutFcn', [], ...
    ' gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before IMA_GUI is made visible.
function IMA_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   unrecognized PropertyName/PropertyValue pairs from the
%            command line (see VARARGIN)

% Choose default command line output for IMA_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes IMA_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = IMA_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
Pool = round(csvread('Pool2.csv'));
assignin('base','Pool',Pool)

% --- Executes on selection change in SetupAP.
function SetupAP_Callback(hObject, eventdata, handles)
% hObject handle to SetupAP (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns SetupAP contents as cell array
% contents{get(hObject,'Value')} returns selected item from SetupAP

%select AP Setup
APSetup_selected = get(hObject,'Value');

axes(handles.area)

[grab_pts,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4] = SetupAP( APSetup_selected );
% Clear Data when AP setup Change
Act_pts = [];
Est_pts = [];
Est_pts_old = [];
Rate = [];
Ref_pts = [];

% Assign Variables
assignin('base','AP',grab_pts)
assignin('base','Pool',Pool)
assignin('base','PoolQ1',PoolQ1)
assignin('base','PoolQ2',PoolQ2)
assignin('base','PoolQ3',PoolQ3)
assignin('base','PoolQ4',PoolQ4)
assignin('base','Rate',Rate)

% Clear Pts if AP changes.
assignin('base','Act_pts',Act_pts)
assignin('base','Est_pts',Est_pts)
assignin('base','Est_pts_old',Est_pts_old)
assignin('base','Ref_pts',Ref_pts)

%Plot AP points
scatter(grab_pts(:,2), grab_pts(:,3),100, 'square','fill','blue')
axis([-1 21 -1 21], 'square')
% --- Executes during object creation, after setting all properties.
function SetupAP_CreateFcn(hObject, eventdata, handles)
    % hObject    handle to SetupAP (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    empty - handles not created until after all CreateFcns called

    % Hint: popupmenu controls usually have a white background on Windows.
    % See ISPC and COMPUTER.
    if ispc && isequal(get(hObject,'BackgroundColor'),
        get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
    end

% --- Executes on button press in Add5Rdm.
function Add5Rdm_Callback(hObject, eventdata, handles)
    % hObject    handle to Add5Rdm (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
    axes(handles.area)

    %Add 5 tracked points
    %import pool and current
    Pool = evalin('base','Pool');
    PoolQ1 = evalin('base','PoolQ1');
    PoolQ2 = evalin('base','PoolQ2');
    PoolQ3 = evalin('base','PoolQ3');
    PoolQ4 = evalin('base','PoolQ4');
    Act_pts = evalin('base','Act_pts');
    Est_pts = evalin('base','Est_pts');
    Rate = evalin('base','Rate');
    AP = evalin('base','AP');

    %Generate random number
    temp_idx = rand_int(1,size(Pool),5,1,1,0);

    % Add 5
    [Act_pts,Est_pts,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Rate] =
       Add5pts(temp_idx,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Act_pts,Est_pts,Rate);

    %Plot Actual Points
    scatter(AP(:,2),AP(:,3),100,'square','fill','blue')
    axis([-1 21 -1 21],'square')

    labels = num2str((1:size(Act_pts,1))','%d');
    text(Act_pts(:,1), Act_pts(:,2), labels, 'horizontal','left',
        'vertical','bottom')
    hold on
    scatter(Act_pts(:,2), Act_pts(:,3),100,'^','fill','black')
    hold off
%write back to base.
assignin('base','Pool',Pool)
assignin('base','PoolQ1',PoolQ1)
assignin('base','PoolQ2',PoolQ2)
assignin('base','PoolQ3',PoolQ3)
assignin('base','PoolQ4',PoolQ4)
assignin('base','Act_pts',Act_pts)
assignin('base','Est_pts',Est_pts)
assignin('base','Rate',Rate)

TableData = [Act_pts Est_pts(:,2) Est_pts(:,3)];

set(handles.ActTable1,'data',TableData,'ColumnName',{'ID', 'X', 'Y','EstX','EstY'});

% --- Executes on button press in Add5Q1.
function Add5Q1_Callback(hObject, eventdata, handles)
% hObject    handle to Add5Q1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.area)
%import pool and current
Pool = evalin('base','Pool');
PoolQ1 = evalin('base','PoolQ1');
PoolQ2 = evalin('base','PoolQ2');
PoolQ3 = evalin('base','PoolQ3');
PoolQ4 = evalin('base','PoolQ4');
Act_pts = evalin('base','Act_pts');
Est_pts = evalin('base','Est_pts');
Rate = evalin('base','Rate');
AP = evalin('base','AP');

%gen random number
% temp_idx = [134;181;286;282;238]; %4AP
temp_idx = [134;181;285;281;237]; %5AP
% temp_idx = [133;180;282;278;234]; %9AP
% temp_idx = [126;173;273;269;225]; %25AP
% Add 5
[Act_pts,Est_pts,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Rate] = Add5pts(temp_idx,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Act_pts,Est_pts,Rate);

%Plot Actual Points
scatter(AP(:,2), AP(:,3),100, 'square','fill','blue')
axis([-1 21 -1 21],'square')

labels = num2str((1:size(Act_pts,1))','%d');
text(Act_pts(:,2), Act_pts(:,3), labels, 'horizontal','left', 'vertical','bottom')
hold on
scatter(Act_pts(:,2), Act_pts(:,3),100, '^','fill','black')
hold off
%write back to base.
assignin('base','Pool',Pool)
assignin('base','PoolQ1',PoolQ1)
assignin('base','PoolQ2',PoolQ2)
assignin('base','PoolQ3',PoolQ3)
assignin('base','PoolQ4',PoolQ4)
assignin('base','Act_pts',Act_pts)
assignin('base','Est_pts',Est_pts)
assignin('base','Rate',Rate)

TableData = [Act_pts Est_pts(:,2) Est_pts(:,3)];

set(handles.ActTable1,'data',TableData,'ColumnName',{'ID','X','Y','EstX','EstY'});

% --- Executes on button press in Add5Q2.
function Add5Q2_Callback(hObject, eventdata, handles)
% hObject handle to Add5Q2 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
axes(handles.area)
%import pool and current
Pool = evalin('base','Pool');
PoolQ1 = evalin('base','PoolQ1');
PoolQ2 = evalin('base','PoolQ2');
PoolQ3 = evalin('base','PoolQ3');
PoolQ4 = evalin('base','PoolQ4');
Act_pts = evalin('base','Act_pts');
Est_pts = evalin('base','Est_pts');
Rate = evalin('base','Rate');
AP = evalin('base','AP');

%gen random number
% temp_idx = [59;75;65;354;368]; %4AP
temp_idx = [59;75;65;353;367]; %5AP
% temp_idx = [58;74;64;350;364]; %9AP
% temp_idx = [56;72;62;336;350]; %25AP
% Add 5
[Act_pts,Est_pts,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Rate] = Add5pts(temp_idx,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Act_pts,Est_pts,Rate);

%Plot Actual Points
scatter(AP(:,2), AP(:,3),100, 'square','fill','blue')
axis([-1 21 -1 21],'square')

labels = num2str((1:size(Act_pts,1)),'%d');
text(Act_pts(:,2), Act_pts(:,3), labels, 'horizontal','left', 'vertical','bottom')
hold on
scatter(Act_pts(:,2), Act_pts(:,3),100, '^','fill','black')
hold off

%write back to base.
assignin('base','Pool',Pool)
assignin('base','PoolQ1',PoolQ1)
assignin('base','PoolQ2',PoolQ2)
assignin('base','PoolQ3',PoolQ3)
assignin('base','PoolQ4',PoolQ4)
assignin('base','Act_pts',Act_pts)
assignin('base','Est_pts',Est_pts)
assignin('base','Rate',Rate)

TableData = [Act_pts Est_pts(:,2) Est_pts(:,3)];

set(handles.ActTable1,'data',TableData,'ColumnName',{'ID', 'X', 'Y','EstX','EstY'});

% --- Executes on button press in Add5Q3.
function Add5Q3_Callback(hObject, eventdata, handles)
% hObject    handle to Add5Q3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.area)

Pool = evalin('base','Pool');
PoolQ1 = evalin('base','PoolQ1');
PoolQ2 = evalin('base','PoolQ2');
PoolQ3 = evalin('base','PoolQ3');
PoolQ4 = evalin('base','PoolQ4');
Act_pts = evalin('base','Act_pts');
Est_pts = evalin('base','Est_pts');
Rate = evalin('base','Rate');
AP = evalin('base','AP');

%gen random number
% temp_idx = [145;376;139;287;402]; %4AP
temp_idx = [145;375;139;286;401]; %5AP
% temp_idx = [144;372;138;283;398]; %9AP
% temp_idx = [137;358;131;274;384]; %25AP
% Add 5
[Act_pts,Est_pts,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Rate] = Add5pts(temp_idx,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Act_pts,Est_pts,Rate);

%Plot Actual Points
scatter(AP(:,2), AP(:,3),100, 'square','fill','blue')
axis([-1 21 -1 21], 'square')

labels = num2str((1:size(Act_pts,1)),'%d');
text(Act_pts(:,2), Act_pts(:,3), labels, 'horizontal','left', 'vertical','bottom')
hold on
scatter(Act_pts(:,2), Act_pts(:,3),100, '^','fill','black')
hold off

%write back to base.
assignin('base','Pool',Pool)
assignin('base','PoolQ1',PoolQ1)
assignin('base','PoolQ2',PoolQ2)
assignin('base','PoolQ3',PoolQ3)
assignin('base','PoolQ4',PoolQ4)
assignin('base','Act_pts',Act_pts)
assignin('base','Est_pts',Est_pts)
assignin('base','Rate',Rate)

TableData = [Act_pts Est_pts(:,2) Est_pts(:,3)];

set(handles.ActTable1,'data',TableData,'ColumnName',{'ID', 'X', 'Y','EstX','EstY'});

% --- Executes on button press in Add5Q4.
function Add5Q4_Callback(hObject, eventdata, handles)
% hObject    handle to Add5Q4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.area)
%import pool and current
Pool = evalin('base','Pool');
PoolQ1 = evalin('base','PoolQ1');
PoolQ2 = evalin('base','PoolQ2');
PoolQ3 = evalin('base','PoolQ3');
PoolQ4 = evalin('base','PoolQ4');
Act_pts = evalin('base','Act_pts');
Est_pts = evalin('base','Est_pts');
Rate = evalin('base','Rate');
AP = evalin('base','AP');

%gen random number
% temp_idx = [280;51;67;239;289]; %4AP
   temp_idx = [279;51;67;238;288]; %5AP
% temp_idx = [276;50;66;235;285]; %9AP
% temp_idx = [267;48;64;226;276]; %25AP
% Add 5
[Act_pts,Est_pts,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Rate] = Add5pts(temp_idx,Pool,PoolQ1,PoolQ2,PoolQ3,PoolQ4,Act_pts,Est_pts,Rate);

%Plot Actual Points
scatter(AP(:,2), AP(:,3),100, 'square','fill','blue')
axis([-1 21 -1 21],'square')

labels = num2str((1:size(Act_pts,1))','%d');
text(Act_pts(:,2), Act_pts(:,3), labels, 'horizontal','left', 'vertical','bottom')
hold on
scatter(Act_pts(:,2), Act_pts(:,3),100, '^','fill','black')
hold off

%write back to base.
assignin('base','Pool',Pool)
assignin('base','PoolQ1',PoolQ1)
assignin('base','PoolQ2',PoolQ2)
assignin('base','PoolQ3',PoolQ3)
assignin('base','PoolQ4',PoolQ4)
assignin('base','Act_pts',Act_pts)
assignin('base','Est_pts',Est_pts)
assignin('base','Rate',Rate)

TableData = [Act_pts Est_pts(:,2) Est_pts(:,3)];

set(handles.ActTable1,'data',TableData,'ColumnName',{'ID', 'X', 'Y', 'EstX', 'EstY'});

% --- Executes on button press in LANDMARC.
function LANDMARC_Callback(hObject, eventdata, handles)
    Act_pts = evalin('base','Act_pts');
    Est_pts = evalin('base','Est_pts');
    Est_pts_old = evalin('base','Est_pts_old');
    AP = evalin('base','AP');
    Rate = evalin('base','Rate');

% --- Executes on button press in Loop.
function Loop_Callback(hObject, eventdata, handles)
    Act_pts = evalin('base','Act_pts');
    Est_pts = evalin('base','Est_pts');
    AP = evalin('base','AP');
    Rate = evalin('base','Rate');
    Ref_pts= evalin('base','Ref_pts');

    %Refresh RSSI and Get ITU Dist
    [P2P_ITUDist, P2AP_ITUDist] = GrabRSSI( Act_pts,AP);
    assignin('base','P2P_ITUDist',P2P_ITUDist);
    assignin('base','P2AP_ITUDist',P2AP_ITUDist);
    assignin('base','Ref_pts_old',Ref_pts); % Backup old Ref_pts

    %Select 3 ref pts
    [Ref_pts,Rate] = SelectRefPts(P2P_ITUDist, P2AP_ITUDist,AP,Est_pts,Act_pts,Rate,Ref_pts);
assignin('base','Ref(pts)',Ref(pts));
assignin('base','Rate',Rate);
assignin('base','Est_pts_old',Est_pts); % Backup old Est_pts

Est_pts_old = evalin('base','Est_pts_old'); % Read Old Values back

% Trilateration
[Est_pts] = Trilateration( Est_pts, Ref_pts );
% Kalman Filter
[Est_pts] = RFKalmanFilter( Est_pts, Est_pts_old, Ref_pts );

% Get Color rating
[Confidence] = Rating( Est_pts, Rate );

% LANDMARC Benchmark
[LANDMARC_est] = LANDMARC_Benchmark (P2AP_ITUDist, AP, Act_pts);

% Plot Actual Points
scatter(AP(:,2), AP(:,3),100, 'square', 'fill', 'blue')
axis([-1 21 -1 21], 'square')
labels = num2str((1:size(Act_pts,1))', '%d');
text(Act_pts(:,2), Act_pts(:,3), labels, 'horizontal', 'left', 'vertical', 'bottom')
hold on
scatter(Act_pts(:,2), Act_pts(:,3),100, '^', 'fill', 'black')

X = Est_pts(:,2);
Y = Est_pts(:,3);
labels = num2str((1:size(Est_pts,1))', '%d');
text(X, Y, labels, 'horizontal', 'left', 'vertical', 'bottom')
scatter(X, Y,100,Confidence, 'filled', 'v')
hold off

TableData = [Act_pts Est_pts(:,2) Est_pts(:,3)];
set(handles.ActTable1,'data', TableData, 'ColumnName', {'ID', 'X', 'Y', 'EstX', 'EstY'});

% Error Calculation
myError = sqrt(((Est_pts(:,2) - Act_pts(:,2)).^2 + (Est_pts(:,3) - Act_pts(:,3)).^2));
LANDMARC_Error = sqrt(((LANDMARC_est(:,2) - Act_pts(:,2)).^2 + (LANDMARC_est(:,3) - Act_pts(:,3)).^2));
set(handles.ErrorTable, 'data', myError, 'ColumnName', {'Error'});

assignin('base','myError', myError);
assignin('base','LANDMARC_Error', LANDMARC_Error);
myError = myError';
LANDMARC_Error = LANDMARC_Error';
dlmwrite('ErrorSimulation.csv',myError,'-append');
dlmwrite('LANDMARC_Error.csv',LANDMARC_Error,'-append');
assignin ('base','Est_pts',Est_pts);

% --- Executes on button press in Iterate10.
function Iterate10_Callback(hObject, eventdata, handles)
% hObject    handle to Iterate10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

for loop =1:30
    Loop_Callback(hObject, eventdata, handles);
end

% --- End of IMA_GUI.m File ---
%Function GrabRSSI
%   Lookup for Random RSSI reading in RSSI database

function [ P2P_ITUDist, P2AP_ITUDist ] = GrabRSSI( Act_pts, AP )

%Calculate Lookuptable Distance
P2PDist = round(pdist2(Act_pts(:,[2,3]),Act_pts(:,[2,3])));
P2APDist =round(pdist2(AP(:,[2,3]),Act_pts(:,[2,3])));

% RSSI Lookup Table
filename = 'LookupTable.csv';
LookupTable = round(csvread(filename));
% Assing Value for distance more than 10m
TooFar = -99;

%Lookup Client to AP RSSI
for col=1:size(Act_pts,1)
    for row=1:size(AP,1)
        if (row==1)
            RSSI = vlookup(LookupTable,P2APDist(row,col),randi([2 62]));
            if isnan(RSSI)
                RSSI = TooFar;
            end
        else
            RSSIx = vlookup(LookupTable,P2APDist(row,col),randi([2 62]));
            if isnan(RSSIx)
                RSSIx = TooFar;
            end
            RSSI = [RSSI ; RSSIx];
        end
        if (col ==1)
            P2AP_RSSI = RSSI;
        else
            P2AP_RSSI = [P2AP_RSSI ; RSSI];
        end
        clear RSSI
        clear RSSIx
    end
end
%Lookup Client to Client RSSI
for col=1:size(Act_pts,1)
    for row=1:size(Act_pts,1)
        if (row==1)
            RSSI = vlookup(LookupTable,P2PDist(row,col),randi([2 62]));
            if isnan(RSSI)
                RSSI = TooFar;
            end
        else
            RSSIx = vlookup(LookupTable,P2PDist(row,col),randi([2 62]));
            if isnan(RSSI)
                RSSIx = TooFar;
            end
            RSSI = [RSSI ; RSSIx];
        end
    end
    if (col ==1)
        P2P_RSSI = RSSI;
    else
        P2P_RSSI = [P2P_RSSI RSSI];
    end
    clear RSSI
    clear RSSIx
end
P2P_ITUDist = ITUmodel(P2P_RSSI);
P2AP_ITUDist = ITUmodel(P2AP_RSSI);
end
%--- End of GrabRSSI.m File ---%
function [ Ref_pts, Rate ] = SelectRefPts(P2P_ITUDist, P2AP_ITUDist, AP, Est_pts, Act_pts, Rate, Ref_pts)

if isempty(Ref_pts)
    Ref_pts_old = [];
else
    Ref_pts_old = Ref_pts;
end
Ref_pts = [];
Rate_tmp = Rate;

for n=1:size(Act_pts,1)
    % Setting up
    % Distance APX APY
    APs = [P2AP_ITUDist(:,n) AP(:,2) AP(:,3)];
    P2Ps = [P2P_ITUDist(:,n) Est_pts(:,2) Est_pts(:,3) Est_pts(:,6)];
    SubX = P2Ps(n,2); % Remove own X from list
    SubY = P2Ps(n,3); % Remove own Y from list
    P2Ps(n,:) = [];
    % Clear 0 dist to self
    % Add Quadrant Data to APs
    if not(isnan(SubX))
        tempQuad = [];
        for i = 1:size(APs,1)
            dx = APs(i,2) - SubX;
            dy = APs(i,3) - SubY;
            if (i==1)
                Quad = Sector(dx,dy);
            else
                tempQuad = Sector(dx,dy);
            end
            Quad = [Quad ; tempQuad];
        end
    else
        Quad = nan(size(APs,1),1);
    end

    APs = [APs Quad];
    APs = sortrows(APs,[1]);

    % Add Quadrant Data to P2Ps
    if not(isnan(SubX))
        tempQuad = [];
        for i = 1:size(P2Ps,1)
            dx = P2Ps(i,2) - SubX;
            dy = P2Ps(i,3) - SubY;
            if (i==1)
                Quad = Sector(dx,dy);
            else
                tempQuad = Sector(dx,dy);
            end
            Quad = [Quad ; tempQuad];
        end
    else
        Quad = nan(size(P2Ps,1),1);
    end

    Ref_pts = [Ref_pts, Quad]
end

end
else
    tempQuad = Sector(dx,dy);
end
Quad = [Quad ; tempQuad];
end
else
    Quad = nan(size(P2Ps,1),1);
end
P2Ps = [P2Ps Quad];
P2Ps = sortrows(P2Ps,[1]);

% Clear Selection flags and Selection
AP1Sel = false;
AP2Sel = false;
AP3Sel = false;

AP1 = [];
AP2 = [];
AP3 = [];

% Reference Point 1
if (APs(1,1) <= 5) % if Fix AP less than 5m
    AP1 = APs(1,:);
    APs(1,:) = []; % Delete selected APoint
    AP1Sel = true;
    Rate_tmp(n) = Rate_tmp(n) - 1;
    if Rate_tmp(n)==0
        Rate_tmp(n)=1;
    end
elseif (APs(1,1) > 5) % Fix AP > 5m
    i=1;
    while not(AP1Sel)
        if not(and(isnan(P2Ps(i,2)),isnan(P2Ps(i,3)))) % Look for known SoftAP
            if (P2Ps(i,4)<=0.05) % check for known softAP "Green"
                if (P2Ps(i,1)<=5) % check softAP less than 5 m
                    AP1 = P2Ps(i,:);
                    P2Ps(i,:) = []; % Delete selected Point
                    AP1Sel = true;
                    j=1;
                    while not (AP2Sel)
                        if and(not(isnan(AP1(4))),not(isnan(APs(j,4)))) % Check make sure is not NaN
                            if (AP1(4)~= APs(j,4)) % Check Diff quadrant
                                AP2 = APs(j,:);
                                APs(j,:) = []; % delete selected
                            AP2Sel = true;
Rate_tmp(n) = Rate_tmp(n)- 1;
    if Rate_tmp(n)==0
        Rate_tmp(n)=1;
    end
end
j = j+1;

if j > size(APs,1)
    if AP2Sel == false
        AP2 = APs(1,:);
        APs(1,:)=[]; % delete selected points
        AP2Sel = true;
        Rate_tmp(n) = Rate_tmp(n)- 1;
        if Rate_tmp(n)==0
            Rate_tmp(n)=1;
        end
    end
end
end
end
end

i=i+1;

if i> size(P2Ps,1)
    AP1 = APs(1,:);
    APs(1,:) = []; %delete selected points
    AP1Sel = true;
end
end
end

%% Reference Point 2

i = 1;
%look for nearest Soft AP in diff quat from Ref 1
while not (AP2Sel)
    if not(and(isnan(P2Ps(i,2)),isnan(P2Ps(i,3)))) % Look for known SoftAP
        if (P2Ps(i,4)<=0.05) % check for known softAP "Green"
            if and(not(isnan(AP1(4))),not(isnan(P2Ps(i,4)))) % Check make sure is Quad is not NaN
                if (P2Ps(i,1)<=5)
                    if (AP1(4)== P2Ps(i,4)) % Check Diff quadrant
                        AP2 = P2Ps(i,:);
                        P2Ps(i,:)=[]; %delete selected point
                        AP2Sel = true;
                end
            end
        end
    end
end
end
end
i = i+1;
if i > size(P2Ps,1)
  if AP2Sel == false
    AP2 = APs(1,:);
    APs(1,:)=[]; %delete selected points
    AP2Sel = true;
  end
end

%% Reference Point 3
i = 1;
%look for nearest Soft AP in diff quad from Ref 1 and 2
while not (AP3Sel)
  if not(and(isnan(P2Ps(i,2)),isnan(P2Ps(i,3)))) % Look for known SoftAP
    if (P2Ps(i,4)<=0.05) % check for known softAP "Green"
      if not(isnan(AP2(4)))
        if and(not(isnan(AP1(4))),not(isnan(P2Ps(i,4)))) % Check make sure is Quad is not NaN
          if (P2Ps(i,1)<=5)
            if (AP2(4) ~= P2Ps(i,4))
              if (AP1(4) ~= P2Ps(i,4)) % Check Diff
                AP3 = P2Ps(i,:);
                P2Ps(i,:) = []; %delete selected points
                AP3Sel = true;
              else
                AP3Sel = true;
              end
            end
          end
        end
      end
    end
  else
    i = i+1;
    if i > size(P2Ps,1)
      if AP3Sel == false
        AP3 = APs(1,:);
        APs(1,:)=[]; % delete selected points
        AP3Sel = true;
      end
    end
  end
end

Ref_pts = [Ref_pts; AP1(1) AP1(2) AP1(3) AP2(1) AP2(2) AP2(3) AP3(1) AP3(2) AP3(3)]; % save ref Pts selected
end
Rate = Rate_tmp;
end
%--- End of SelectRefPts.m---%
% Trilateration
%   Execute the Triangulation-Trilateration process if confidence level
%   is low.
function [ Est_pts ] = Trilateration( Est_pts, Ref_pts )

Est_pts_tmp = []; 
for i=1:size(Est_pts,1) 
   if (Est_pts(i,6)>= 0.0495) % Continue Triangulation-Trilateration process for unknown pts.  
      [X,Y] = APAnchor( Ref_pts(i,1), Ref_pts(i,2), Ref_pts(i,3), Ref_pts(i,4), Ref_pts(i,5), Ref_pts(i,6), Ref_pts(i,7), Ref_pts(i,8), Ref_pts(i,9));  
      else 
         X = Est_pts(i,2); 
         Y = Est_pts(i,3); 
   end 
   Est_pts_tmp = [Est_pts_tmp; Est_pts(i,1) X Y Est_pts(i,4) Est_pts(i,5) Est_pts(i,6) Est_pts(i,7)]; 
end
Est_pts = Est_pts_tmp;
end
%% End of Trilateration.m File---%
% Triangulation-Trilateration Method
% function [X,Y] = APAnchor (R1,X1,Y1,R2,X2,Y2,R3,X3,Y3)

t=[0:359]';

% Anchor the estimated location to the nearest AP and Generate 360 estimated pts
x = X1 + R1*cosd(t);
y = Y1 + R1*sind(t);

% Create polygon from the 3 reference point
xv = [X1;X2;X3];
yv = [Y1;Y2;Y3];

% Check if points are in polygon
in = inpolygon(x,y,xv,yv);

% Calculate Weights for estimation error calculation
E1 = 1/(R2^2);
E2 = 1/(R3^2);
E = E1+E2;

% Calculate estimation error from RSSI and Euclidean distance
dx1 = x - X2;
dy1 = y - Y2;
d1 =abs(R2 - sqrt((dx1.^2+dy1.^2)));

dx2 = x - X3;
dy2 = y - Y3;
d2 =abs(R3 - sqrt((dx2.^2+dy2.^2)));

% Apply weights to estimation error.
d = abs( (E1/E)*d1 + (E2/E)*d2);
% Consolidate data
coor = [x y in d];

% Sort to select coordinates in polygon which has the least estimation error
coor = sortrows(coor,[-3 4]);

if (coor(1,3)==0)
    X = NaN;
    Y = NaN;
else
    X = coor(1,1);
    Y = coor(1,2);
end
% End of file

% Selective Kalman Filter
% Filter only if 1st ref points is within 5m
function [ Est_pts ] = RFKalmanFilter( Est_pts,Est_pts_old,Ref_pts )

Est_pts_tmp = []; for i=1:size(Est_pts,1)

    if isnan(Est_pts_old(i,2))
        Est_pts_old(i,2) = Est_pts(i,2);
        Est_pts_old(i,3) = Est_pts(i,3);
    end
    % Read previous data
    zx = Est_pts(i,2);
    px = Est_pts(i,4);
    zy = Est_pts (i,3);
    py = Est_pts(i,5);
    kx = Est_pts(i,6);
    ky = Est_pts(i,7);
    x = Est_pts_old (i,2);
    y = Est_pts_old (i,3);
    % Apply filter if 1st Ref <5m
    if (Ref_pts(i)<= 5)
        [X,pX,Y,pY,Kx,Ky] = kalmanxy(x,zx,px,y,zy,py);
        %Else no filter
    else
        if and(isnan(zx),isnan(zy))
            X = x;
            Y = y;
        else
            X = zx;
            Y = zy;
        end
        pX = px;
        pY = py;
        Kx = kx;
        Ky = ky;
    end

    Est_pts_tmp = [Est_pts_tmp;Est_pts(i,1) X Y pX pY Kx Ky];
end
Est_pts = Est_pts_tmp;
end
% End of File
function [X,pX,Y,pY,Kx,Ky] = kalmanxy(x,zx,px,y,zy,py)

% output
% X and Y is the estimated X and Y coor
% pX and pY are the error corvariance

% % Input
% x and y are previous estimate
% zx and zy are measurement
% px and py are previous error corvariance

R = 0.1; % measurement noise assumption

%fix nan bug
if and(isnan(zx),isnan(zy))
    zx = x;
    zy = y;
end

%%% Measurement Update %%%
% Calculate Kalman Gain
Kx = px/(px + R);
Ky = py/(py + R);

%estimate new XY
X = x + (Kx*(zx-x));
Y = y + (Ky*(zy-y));

%update error corvariance
pX = (1-Kx)* px;
pY = (1-Ky)* py;

%%% Time Update to be added for future %%%
% X = x %+ control
% Y = y %+ control
%
% pX = px
% pY = pY

% End of file
List of publications produced as a result of this study


II. Liu, L. and Wong, W., 2015, June. Indoor positioning through an iterative method in dense Wi-Fi-direct networks with Wi-Fi direct devices. In Consumer Electronics-Taiwan (ICCE-TW), 2015 IEEE International Conference on (pp. 308-309). IEEE.