# WiFi RTT-based pedestrian navigation and positioning in indoor environments

**Khalil Jibran Raja**

University College London

Civil, Environmental and Geomatic Engineering

Submitted to University College London (UCL) for the award of the degree of Doctor of Philosophy.

# Declaration

I, Khalil Jibran Raja, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

# Abstract

Global Navigation Satellite Systems (GNSS) are the primary tool for navigation and positioning in mobile devices. However, GNSS is unreliable or unusable in many environments, such as large indoor buildings, tunnels, and caves—referred to as GNSS-degraded or GNSS-denied environments. This research focuses on positioning techniques for indoor environments using the WiFi Fine Time Measurement (FTM) also known as WiFi Round Trip Timing (RTT) protocol. This protocol, enabled in 802.11mc-compatible routers and devices, allows for determining the Time of Flight (ToF) of a signal.

The research described in this thesis explores the characteristics of WiFi RTT signals and their use as a positioning solution when combined with techniques such as least squares positioning, filtering, and Simultaneous Localization and Mapping (SLAM). A common assumption in indoor positioning solutions is the prior knowledge of landmark locations; SLAM techniques were investigated to remove this assumption. The filtering methods explored included particle filters, genetic filters, and grid filters. The exploration of SLAM focused on FastSLAM 2.0-based methods, which were extended into posterity SLAM, a form of cooperative SLAM. These methods were further enhanced with RSSI-based outlier detection. This outlier detection method allows the positioning algorithms to account for unreliable signals by identifying inconsistencies between the RSSI of a signal and the RTT measured range of that signal.

The filtering methods achieved sub-2-meter accuracy 97% of the time when the mobile device was stationary. When the device was in motion, it was tracked within 2 meters 81% of the time. For the SLAM algorithm, landmarks were positioned to sub-2-meter accuracy 61% of the time, which improved to 78% when posterity SLAM was incorporated. In summary, this research advances the understanding and application of WiFi RTT for indoor positioning.

# Impact Statement

The research presented in this thesis contributes significantly to the field of indoor navigation and positioning using WiFi RTT. The thesis provides several innovations to the relatively new topic of WiFi RTT positioning, demonstrating sub-two-metre positioning accuracy and in certain cases sub-metre accuracy. In most cases, this level of positioning is sufficient for locating a mobile device to the correct room.

Academically, this study advances WiFi RTT positioning within indoor environments. By using the WiFi RTT protocol available in most modern WiFi access points, it is possible to apply Time of Flight-based positioning algorithms to WiFi signals. This enables positioning algorithms and techniques, previously not widely applicable to indoor environments, to become accessible. The application of these algorithms has the potential to make sub-metre indoor positioning commercially viable in any building worldwide. The implementation of WiFi RTT-based algorithms in this thesis offers practical approaches to indoor positioning challenges. The exploration of WiFi RTT with filtering algorithms marks contributions to indoor positioning methods, while the investigation of SLAM algorithms offers solutions for navigation and positioning in unmapped indoor environments. Additionally, this study demonstrates outlier detection methods to address limitations of WiFi RTT signals, such as NLOS and multipath effects, to improve the accuracy of WiFi RTT-based indoor positioning systems.

Practically, the applications of this research are extensive and diverse. The improved accuracy and reliability of WiFi-based positioning in indoor environments have benefits for pedestrian, emergency, security, military, and visually impaired navigation. This advancement is crucial for location-based services (LBS), which are increasingly integral to operations in extended reality, logistics, targeted advertising, and social networking. Enhanced positioning accuracy can significantly boost these services, improving user experiences and operational efficiency. Improved indoor positioning enables social benefits such as improved emergency caller localisation

and tracking of vulnerable people. Furthermore, the entertainment and gaming industries can utilize these advancements to create more immersive and accurate experiences, leading to innovative applications that seamlessly blend digital and physical worlds in urban settings.

The impact of this research can be amplified through various channels. Research from this paper has been presented at conferences on four separate occasions, at ION GNSS+ 2023, ION GNSS+ 2024, ENC 2024 and Next Gen Nav 2022. These presentations have raised awareness around WiFi RTT and its potential as an indoor positioning solution. Publishing findings in academic journals, such as a paper being published in the Journal of Navigation, will further raise awareness and encourage the adoption of these techniques. Collaboration with industry partners and policymakers can facilitate the translation of these research outcomes into practical applications.

In summary, this research not only contributes to the academic field of navigation and positioning, but also has the potential to transform a wide range of applications. Applications from urban navigation to entertainment are improved, leading to enhanced urban living experiences, more efficient public services, and innovative commercial and cultural applications.

# Acknowledgements

I would first and foremost like to acknowledge my supervisor, Dr Paul Groves. Paul has been a truly formative person in my life, and I feel honoured to have worked with him. When I first started my PhD, I was anxious about the research, but Paul quickly helped alleviate these feelings with his methodical approach and commitment to the field. To Paul, working on WiFi RTT—which at the time seemed crazy to me—was work that needed to be done because it would push the industry into unexplored areas. Looking back, I couldn't agree with him more. Paul's diligence, rigour, deep knowledge, and ability to think independently are qualities I will strive to emulate. That being said, given the quality of his jokes during our meetings, I can confidently say that Paul does not have a future as a comedian.

I would like to thank the UCL CEGE department, which has been my home for eight years. I am also exceedingly grateful to UCL for funding the PhD through a departmental scholarship.

I would also like to acknowledge my research group. Peter, Qiming, and Taylor have been great to chat with and learn from over the past few years, and meeting them has been one of the highlights of my PhD journey.

Last but certainly not least, I owe a great deal of gratitude to my family: my mother, for putting up with my nonsense for far too long, always listening, looking after, and caring for me unconditionally; my father, for being a bastion of support, providing valuable advice, and instilling in me a strong work ethic; my brother, for always being someone I could talk to about anything; my sisters, and the rest of my family. Ultimately, my late grandfather inspired my decision to pursue a PhD. He came to the UK many years ago to acquire his, and though he is not around to see me finish, I remember how happy he was when I started.

# UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **1. For a research manuscript that has already been published:**
   (a) **What is the title of the manuscript?** WiFi-RTT Indoor Positioning Using Particle, Genetic and Grid Filters with RSSI-Based Outlier Detection
   (b) **Please include a link to or doi for the work:** https://doi.org/10.33012/2023.19211
   (c) **Where was the work published?** Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023)
   (d) **Who published the work?** The Institute of Navigation
   (e) **When was the work published?** September 2023
   (f) **List the manuscript's authors in the order they appear on the publication:** Khalil Raja, Paul D. Groves
   (g) **Was the work peer reviewed?** No
   (h) **Have you retained the copyright?** Yes
2. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Khalil Raja conducted most of the research and write-up, Paul D. Groves provided supervision and review of the paper.
3. **In which chapter(s) of your thesis can this material be found?** throughout

**Candidate:** Khalil Raja
**Date:** 15 / 09 / 2024

**Supervisor/Senior Author signature** (where appropriate)**:**
**Date:** 

# UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?** WiFi-RTT SLAM: Pedestrian navigation in unmapped environments using WiFi-RTT and smartphone inertial sensors

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**
   **If 'Yes', please please give a link or doi:**

   (c) **Where is the work intended to be published?** MDPI engineering proceedings

   (d) **List the manuscript's authors in the intended authorship order:** Khalil Raja, Paul D. Groves

   (e) **Stage of publication:** In submission

2. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Khalil Raja conducted most of the research and write-up, Paul D. Groves provided supervision and review of the paper.

3. **In which chapter(s) of your thesis can this material be found?** throughout

**Candidate:** Khalil Raja

**Date:** 15 / 09 / 2024

**Supervisor/Senior Author signature** (where appropriate)**:**

**Date:** [redacted]

# UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?** WiFi-RTT indoor positioning using particle, genetic and grid filters with RSSI-based outlier detection

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**
   **If 'Yes', please please give a link or doi:** No

   (c) **Where is the work intended to be published?** Journal of Navigation

   (d) **List the manuscript's authors in the intended authorship order:** Khalil Raja, Paul D. Groves

   (e) **Stage of publication:** In submission

2. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Khalil Raja conducted most of the research and write-up, Paul D. Groves provided supervision and review of the paper.

3. **In which chapter(s) of your thesis can this material be found?** throughout

**Candidate:** Khalil Raja
**Date:** 15 / 09 / 2024

**Supervisor/Senior Author signature** (where appropriate)**:**
**Date:** 

# UCL Research Paper Declaration Form: referencing the doctoral candidate's own published work(s)

1. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)**:**

   (a) **What is the current title of the manuscript?** WiFi-RTT Posterity SLAM for Pedestrian Navigation in Indoor Environments

   (b) **Has the manuscript been uploaded to a preprint server 'e.g. medRxiv'?**
   **If 'Yes', please please give a link or doi:** No

   (c) **Where is the work intended to be published?** Proceedings of the 37th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2024)

   (d) **List the manuscript's authors in the intended authorship order:** Khalil Raja, Paul D. Groves

   (e) **Stage of publication:** In submission

2. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)**:** Khalil Raja conducted most of the research and write-up, Paul D. Groves provided supervision and review of the paper.

3. **In which chapter(s) of your thesis can this material be found?** throughout

**Candidate:** Khalil Raja

**Date:** 15 / 09 / 2024

**Supervisor/Senior Author signature** (where appropriate)**:**

**Date:** [redacted]

# Contents

**1 Introduction** **19**

1.1 Positioning in Indoor environments . . . . . 19

1.2 WiFi RTT . . . . . 21

1.3 Mitigating NLOS and multipath . . . . . 21

1.4 Prior knowledge of indoor environments . . . . . 22

1.5 Aims of the thesis . . . . . 23

1.6 Research Questions . . . . . 25

1.7 Structure of the thesis and contributions . . . . . 25

**2 Literature Review and Background** **28**

2.1 Overview of Positioning . . . . . 29

2.1.1 Overview of radio positioning technologies . . . . . 30

2.1.2 Overview of Global Navigation Satellite Systems (GNSS) . . . 32

2.1.3 Limitations of GNSS . . . . . 34

2.2 Overview of Indoor Positioning . . . . . 37

2.2.1 WiFi . . . . . 38

2.2.2 Pedestrian Dead Reckoning . . . . . 42

2.2.3 Bluetooth . . . . . 43

2.2.4 Ultrawideband . . . . . 44

2.3 Existing research on WiFi RSSI and WiFi RTT positioning . . . . . 45
2.3.1 WiFi RSSI-based ranging . . . . . 45
2.3.2 WiFi RSSI-based fingerprinting . . . . . 46
2.3.3 WiFi RTT-based ranging . . . . . 47
2.4 Existing research on WiFi-based SLAM for indoor positioning . . . . 52

**3 WiFi RTT Characteristics 56**
3.1 Devices Used . . . . . 57
3.2 Experiment 1 - Instrument Bias . . . . . 57
3.2.1 Introduction . . . . . 57
3.2.2 Experimental Setup . . . . . 58
3.2.3 Results and Discussion . . . . . 60
3.3 Experiment 2 - Multipath effects . . . . . 63
3.3.1 Introduction . . . . . 63
3.3.2 Experimental Setup . . . . . 63
3.3.3 Results and Discussion . . . . . 63
3.4 Experiment 3 - Attenuation and NLOS . . . . . 67
3.4.1 Introduction . . . . . 67
3.4.2 Experimental Setup . . . . . 67
3.4.3 Results and Discussion . . . . . 68
3.5 Experiment 4 - Instrument Orientation . . . . . 72
3.5.1 Introduction . . . . . 72
3.5.2 Experimental Setup . . . . . 73
3.5.3 Results and Discussion . . . . . 73

**4 Positioning algorithms 76**
4.1 Algorithms . . . . . 76
4.1.1 Least squares positioning . . . . . 76
4.1.2 Weighted Least-Squares Multi-Lateration . . . . . 79
4.1.3 Particle Filters . . . . . 80
4.1.4 Genetic Filter . . . . . 86
4.1.5 Grid Filter . . . . . 90

4.1.6 Outlier detection techniques . . . . . 92
4.1.7 Pedestrian Dead Reckoning Model . . . . . 96
4.2 Method and Data collection . . . . . 98
4.2.1 Equipment . . . . . 98
4.2.2 Stationary Data collection - Least Squares algorithms . . . . . 99
4.2.3 Stationary Data collection - Filtering algorithms . . . . . 100
4.2.4 In-Motion Data collection . . . . . 104
4.3 Results and Discussion . . . . . 105
4.3.1 Least squares positioning . . . . . 105
4.3.2 RSSI-Weighted least squares positioning with RSSI-based outlier detection . . . . . 110
4.3.3 Stationary Filtering . . . . . 112
4.3.4 Filtering in-motion . . . . . 119

**5 SLAM** **130**
5.1 Simultaneous Localisation and Mapping . . . . . 131
5.1.1 FastSLAM . . . . . 131
5.1.2 Factor Graph Optimisation SLAM . . . . . 132
5.1.3 Posterity SLAM . . . . . 135
5.2 Experimental Methodology . . . . . 135
5.3 Results and Discussion . . . . . 138
5.3.1 WiFi RTT SLAM . . . . . 138
5.3.2 WiFi RTT Posterity SLAM . . . . . 142

**6 Conclusions** **154**
6.1 WiFi RTT Characteristics . . . . . 154
6.2 WiFi RTT Positioning . . . . . 155
6.3 WiFi RTT SLAM . . . . . 157

**7 Recommendations for future work** **160**

# List of Figures


2.1 Categorisations of positioning and some applications [1] . . . . . 31

2.2 GNSS architecture [1] . . . . . 33

2.3 GNSS trilateration example [1] . . . . . 34

2.4 GNSS multilateration example [1] . . . . . 35

2.5 GNSS error sources [1] . . . . . 36

2.6 GNSS error sources in buildings . . . . . 37

2.7 WiFi FTM RTT in practice [2] . . . . . 42

3.1 Experiment 1 Setup Plan view . . . . . 59

3.2 Experiment 1 mean FTM range error, including linear approximation determined via least squares . . . . . 60

3.3 Experiment 2a setup . . . . . 64

3.4 Experiment 2b setup . . . . . 64

3.5 AP-2 Experiment 2a and 2b (Reflective Surface) mean FTM range error against true range, including linear approximation determined via least squares . . . . . 65

3.6 AP-3 Experiment 2a and 2b (Reflective Surface) mean FTM range error against true range, including linear approximation determined via least squares . . . . . 66

3.7 Scenario 1 with sub-scenarios A and B. The difference between the sub-scenarios is an open and closed door at 1m between the mobile device and AP . . . . . 68
3.8 Scenario 2 with sub-scenarios a and b. The AP and mobile device can be moved in 10cm increments away from each other. The walls between the mobile device and AP each have a width of approximately 10cm. 69
3.9 Scenario 2 with sub-scenarios a (LOS) and b (NLOS) calibrated results, the black line represents the true range (i.e. y=x) . . . . . 72
3.10 Experiment 4 Orientation Test Plan view . . . . . 74
3.11 Experiment 4 mean WiFi RTT at different orientations . . . . . 75

4.1 Generic particle filter and genetic filter process . . . . . 81
4.2 Step-lagged PDR motion model . . . . . 84
4.3 Genetic resampling process . . . . . 89
4.4 Google Pixel 4a in holder, operating position during ranging sessions 99
4.5 Google Nest WiFi Point . . . . . 100
4.6 Experimental layout of APs and device for trial 1. x,y coordinates represent the distance from the origin, z coordinate represents the distance from the floor. Coordinates are represented in millimetres. . 102
4.7 Experimental layout of APs and device for trial 2. x,y coordinates represent the distance from the origin, z coordinate represents the distance from the floor. Coordinates are represented in millimetres. . 103
4.8 Experimental layout of APs and device for trial 3. x,y coordinates represent the distance from the origin, z coordinate represents the distance from the floor. The diagonally striped rectangles are tables that do not block signals, but may reflect them. Coordinates are represented in millimetres. . . . . 103
4.9 Experimental layout of APs and device for filtering trials. x,y coordinates represent the distance from the origin. . . . . 104
4.10 Experimental layout of APs and device for in-motion trials. x,y coordinates represent the distance from the origin. . . . . 106

4.11 the results of the experiments on trial 1 using least squares positioning and outlier detection, with APs spread across multiple rooms . . . . . 109
4.12 the results of the experiments on trial 2 using least squares positioning and outlier detection, with a tighter cluster of APs . . . . . 111
4.13 Trial 1 - RSSI-Weighted least squares positioning with RSSI-based outlier detection positioning result . . . . . 113
4.14 Trial 2 - RSSI-Weighted least squares positioning with RSSI-based outlier detection positioning result . . . . . 114
4.15 Trial 3 - RSSI-Weighted least squares positioning with RSSI-based outlier detection positioning result . . . . . 115
4.16 Environment E particle distribution diagram . . . . . 120
4.17 Environment F particle distribution diagram . . . . . 121
4.18 Trial 1F position per epoch . . . . . 127
4.19 Trial 1R position per epoch . . . . . 127
4.20 Trial 2F position per epoch . . . . . 128
4.21 Trial 2R position per epoch . . . . . 128
4.22 Trial 3F position per epoch . . . . . 129
4.23 Trial 3R position per epoch . . . . . 129

5.1 WiFi RTT FastSLAM algorithm . . . . . 133
5.2 WiFi RTT FastSLAM visual representation . . . . . 134
5.3 SLAM paths for SLAM and Posterity SLAM algorithms . . . . . 137
5.4 Graph showing the performance of the SLAM algorithm in the forward trial . . . . . 139
5.5 Mobile device position error per step for the forward trial . . . . . 139
5.6 Graph showing the performance of the SLAM algorithm in the reverse trial, AP predictions represent the final AP position estimates . . . . 141
5.7 Position Horizontal Error per step for the reverse trial . . . . . 142
5.8 Forward Path posterity SLAM horizontal error per step . . . . . 143
5.9 Reverse Path posterity SLAM horizontal error per step . . . . . 144

5.10 Forward then Reverse AP position error per step. The white cut-off in the centre of the chart represents the switch from the forward trial to the reverse trial. The error bars represent the standard deviation of each landmark particle filter . . . . . 146
5.11 Reverse then Forward AP position error per step. The white cut-off in the centre of the chart represents the switch from the reverse trial to the forward trial. The error bars represent the standard deviation of each landmark particle filter . . . . . 147
5.12 Reverse Trial using Forward Trial Path posterity SLAM position estimate and landmark estimates . . . . . 148
5.13 Short Path regular SLAM position estimate and landmark estimates . 149
5.14 Short using Forward Path posterity SLAM position estimate and landmark estimates . . . . . 149
5.15 Reverse then Short AP position error per step. The white cut-off in the centre of the chart represents the switch from the reverse trial to the short trial. The error bars represent the standard deviation of each landmark particle filter . . . . . 151
5.16 Short then Reverse AP position error per step. The white cut-off in the centre of the chart represents the switch from the short trial to the reverse trial. The error bars represent the standard deviation of each landmark particle filter . . . . . 152

# List of Tables


1.1 Comparison of some Indoor Positioning Technologies . . . . . 24

3.1 Devices used during WiFi RTT Ranging Characteristics experiments 57
3.2 Experiment 1 mean FTM range error and standard deviation . . . . . 59
3.3 Scenario 1a and 1b calibrated results, estimated range standard deviation and RSSI . . . . . 68

4.1 Trial 1 - Access Points and Mobile Device . . . . . 101
4.2 Trial 2 - Access Points and Mobile Device . . . . . 101
4.3 Trial 3 - Access Points and Mobile Device . . . . . 102
4.4 Trial C - Access Points and Mobile Device . . . . . 104
4.5 Trial E and F - Access Points and Mobile Device . . . . . 105
4.6 In motion trial positions . . . . . 107
4.7 Trial 1 - Access Point RSSI and average RTT ranging measurements . 108
4.8 Trial 2 - Access Point RSSI and average RTT ranging measurements . 110
4.9 Positioning solution RMSE for each environment and algorithm configuration . . . . . 113
4.10 Percentage decrease of RMSE against least squares for each environment and algorithm configuration . . . . . 116
4.11 Percentage decrease of RMSE comparing outlier detection against no outlier detection for each algorithm . . . . . 117

4.12 Computation time per epoch for each algorithm alongside mean accuracy improvement over least squares . . . . . 119
4.13 RMSE position error statistics for trials in motion and each algorithm combination . . . . . 122
4.14 Initial Position RMSE for each trial in motion . . . . . 123
4.15 RMSE percentage position error improvement for algorithms and trials in motion . . . . . 125

5.1 Configuration for the SLAM algorithm . . . . . 136
5.2 Statistics for the forward and reverse trials for the SLAM algorithm . 138
5.3 Statistics for the landmark position estimates . . . . . 141
5.4 Statistics for the mobile device position estimates for the Forward and Reverse Trial . . . . . 143
5.5 Statistics for the landmark position estimates for the Forward and Reverse Trial . . . . . 144
5.6 Statistics for the mobile device position estimates for the Short and Forward Trials . . . . . 148
5.7 Statistics for the mobile device position estimates for the Short and Reverse Trials . . . . . 148
5.8 Statistics for the landmark position estimates for the Short and Reverse Trial . . . . . 150
5.9 Statistics for the landmark position estimates for the Short and Forward Trial . . . . . 153

# CHAPTER 1

# Introduction

This chapter will provide an overview of the problem that this thesis aims to address, the aims and objectives and the outline of the thesis along with contributions made in each chapter.

## 1.1 Positioning in Indoor environments

Positioning and Navigation play an integral role in everyday life. Smartphones have almost become a necessity for regular functioning of society, and with this has come a reliance on smartphones for positioning and navigation. Thus, a reliable positioning solution is becoming an ever-growing concern. Smartphones are outfitted with Global Navigation Satellite Systems (GNSS) receivers and a variety of sensors such as Inertial Measurement Units (IMUs), altimeters, mobile network receivers and WiFi receivers. These sensors allow for a smartphone user to be positioned, enabling navigation. GNSS works well in line of sight (LOS) conditions where there is a direct line of sight between the transmitter and the receiver of the signal. However, there are many environments categorised as GNSS degraded and denied environments where GNSS does not provide a reliable positioning solution or does not provide a positioning solution at all. In these environments the GNSS signals

might be blocked, reflected, degraded by multipath effects, received via Non-line-of-sight (NLOS) or received with low power. Some examples include urban canyons, tunnels, jammed environments and the focus of this thesis: indoor environments. Indoor environments are mostly classed as GNSS degraded environments. With GNSS for most indoor environments it is possible to have a good estimate of the building a device is in but not the exact room. With a growing need for accurate location and navigation in many habitable places, i.e. indoor environments, which do not have access to reliable positioning via GNSS, alternatives must be explored.

To improve the performance of indoor positioning, different technologies, methods, and sensors can be used to provide information about a mobile device's position. Indoor positioning in its current state mostly works by using local infrastructure, be this infrastructure dedicated to indoor positioning or infrastructure that has other use-cases but can be used for indoor positioning. Such infrastructure includes WiFi networks, mobile communication base stations, ultra-wideband technology and Bluetooth beacons [3] [4] [5] [6] [7]. Light detection and ranging (LIDAR) [8] and image based solutions can also be used for indoor positioning, but these are typically limited to professional applications such as in factory settings or robotics. Dedicated signals used for positioning would include the WiFi Round Trip Timing (RTT) Protocol [9] (which is a sub-protocol of WiFi), Bluetooth beacons or ultra-wideband beacons. All of these technologies serve to provide information of a mobile device's position, where GNSS is not able to at least position a user in the correct room. The various indoor positioning methods, their advantages, and disadvantages and how they are used will be discussed in Chapter 2.

An indoor positioning system that makes use of multiple indoor positioning technologies as well as other sensors available in a mobile device such as Inertial Measurement Units (IMUs) can produce a more versatile system as the solution is available in environments with different signals available [10].

## 1.2 WiFi RTT

WiFi Fine Time Measurement Round Trip Timing (referred to henceforth as FTM RTT or RTT) was introduced in WiFi protocol update 802.11mc [9]. Google has included FTM RTT in their WiFi aware devices, and included FTM RTT support within Android apps, allowing all FTM RTT enabled Android devices to make use of the protocol [9]. The protocol allows for the round trip time of a WiFi signal between a mobile device and an access point to be calculated and provided to the mobile device. This protocol is specifically designed for positioning. A single access point (AP) can be used to determine proximity to an AP, and 3 or more APs can be used to determine the location of a mobile device using multilateration. This technology has seen limited support in WiFi routers [11] and mobile devices [12] despite having potential to greatly improve current indoor positioning solutions. This is mostly due to the need to change old WiFi infrastructure and/or certain router manufacturers not enabling WiFi RTT by default. In Android 12, the WiFi RTT API provided support for one-sided ranging [13]. This enabled a mobile device to get an RTT measurement from an AP that was 802.11mc or 802.11az compatible but did not have WiFi RTT enabled. This is known as one-sided RTT, as the AP is not cooperative and the measurement received does not account for the turn around time in the AP. It was found that the error of the turnaround is between 2-3m [14]. An indoor positioning system by Horn et al. [14] produced 1-2 metre accuracy for two-sided RTT, but for one-sided RTT the accuracy reduced to 3-4m. One-sided RTT was not used in this thesis.

This thesis will focus on WiFi RTT to understand its limitations and potential to provide sub-metre and reliable indoor positioning and navigation (reliable meaning positioned to sub-two-metre accuracy consistently).

## 1.3 Mitigating NLOS and multipath

RF signals are susceptible to NLOS and multipath [10] [1]. This extends to WiFi RTT. Indoor environments typically contain walls, furniture, people and various

other obstacles. These surfaces can lead to multipath effects where direct, reflected, refracted, diffracted or attenuated signals can reach the receiving antenna via multiple paths. This leads to interference of the signal, which can result in an incorrect range estimate. NLOS signal reception is caused by a signal being obstructed resulting in the strongest signal not being in a straight line, this leads to an overestimate of the range. These factors will reduce the accuracy of the WiFi RTT signals, which in turn reduces the reliability and accuracy of an indoor positioning solution that uses WiFi RTT. The ability to identify and account for NLOS signals or signals that have been interfered with can potentially increase the accuracy and reliability of WiFi RTT. This thesis first explores WiFi RTT's susceptibility to NLOS and multipath effects. This is followed by outlier detection methods that can mitigate the impact of NLOS and multipath effects. These outlier detection models are applied to several WiFi RTT-based positioning algorithms in a series of environments.

## 1.4 Prior knowledge of indoor environments

A common feature of indoor positioning research and to a certain extent real-world systems is the necessity of a survey step or some prior knowledge of the environment. For most techniques, the assumption is some knowledge of the location of the access points or landmarks of an environment. Alternatively, in the case of fingerprinting methods such as RSSI-based fingerprinting, a lengthy survey step is required to collect the fingerprints at each grid point in the given environment [15]. Given the number and diversity of indoor environments, this method is not scalable. In addition, access points can be moved around, and indoor environments have dynamic obstacles which can change the properties of the environment [16]. An approach that is able to reliably position a user to sub-metre accuracy without significant cost of setup is an ideal and scalable approach for indoor positioning. WiFi RTT and Simultaneous localisation and mapping (SLAM) techniques have potential to enable this [17] [18]. SLAM algorithms do not require prior knowledge of an environment as by definition the algorithm is locating the user and landmarks at the same time, in real time. Furthermore, the solution can improve over time with more users as more data can

be collected. This method is likely adopted by large technology companies such as Apple and Google who have access to vast amounts of data (such as GNSS positions, WiFi RTT and RSSI measurements) and a large network of mobile devices.

## 1.5 Aims of the thesis

The overall aim of this thesis is to research and develop positioning algorithms for mobile devices that function indoors and potentially outdoors by using WiFi RTT as the underlying technology.

The target use case for the research presented in this thesis is pedestrian indoor navigation. For this use case, sub-metre accuracy is ideal and sub-two-metre accuracy is acceptable. This level of accuracy will typically place a user in the correct room. For this level of accuracy, there are a number of usable technologies. These are discussed in depth in Section 2. A summary and comparison of these technologies is presented in Table 1.1 to demonstrate the shortcomings and why WiFi RTT was selected. As suggested by the table, WiFi RTT is generally underexplored, but has the potential to provide accurate indoor positioning by applying time-of-flight based positioning techniques to WiFi within the existing widely deployed WiFi network.

The specific objectives are as follows:

- Explore existing research on the effectiveness of WiFi RTT as an indoor positioning solution. Identify any potential gaps in knowledge that would push the frontier of WiFi RTT use. These gaps in knowledge will help inform the direction of the research.

- Investigate WiFi RTT behaviour in NLOS and LOS environments and its signal characteristics in terms of its susceptibility to multipath effects, NLOS signal reception and instrument bias.

- Research positioning algorithms such as Least Squares, weighted least squares, Particle Filters and variations of particle filters such as a genetic filter and a grid filter. These methods were tested in a variety of environments and scenarios to determine whether these multi-epoch algorithms perform better than the basic

| Technology | Typical Accuracy | Pros | Cons |
|---|---|---|---|
| **WiFi RSSI-based ranging** | < 5m | - Widely deployed infrastructure | - Heavily susceptible to NLOS, multipath and other forms of interference |
| **WiFi RSSI fingerprinting** | < 3m | - Widely deployed infrastructure | - Requires surveying the indoor environment |
| **Ultrawideband** | < 30cm | - Highly accurate | - Infrastructure not widely deployed |
| **Bluetooth** | < 3m | - Cheap | - Infrastructure not widely deployed |
| **WiFi RTT** | < 2m | - Potential to be included within widely deployed infrastructure<br>- Time of Flight based | - Susceptible to NLOS, multipath and other forms of interference<br>- Relatively new and underexplored<br>- Instrument calibration issues |

Table 1.1: Comparison of some Indoor Positioning Technologies

single epoch least squares method and can reach a sub-metre mobile positioning accuracy.

- Compare various methods for outlier detection in the context of WiFi RTT-related positioning in order to deduce methods that can be used to mitigate positioning error. Being able to identify anomalous ranging measurements will enable more robust positioning solutions.

- Prior research concerning WiFi RTT has mostly assumed that the location of access points is known. As previously mentioned, this in reality this is difficult to do and a more scalable solution would likely need to predict the location of these access points instead. An objective of this thesis is to explore the application of SLAM algorithms to WiFi RTT positioning to determine the feasibility of SLAM to locate a mobile device to sub-two-metre accuracy and access point positions to sub-two-metre accuracy.

- Explore the feasibility of cooperative SLAM positioning solutions. This solution explores the possibility of using previous SLAM maps of an environment in an

iterative fashion. This innovation essentially revolves around crowdsourcing the information needed to provide a more accurate indoor positioning solution. This allows for indoor maps of environments to be built over-time.

## 1.6 Research Questions

The research questions of this thesis are shown below and are answered at the end of the thesis in the conclusion.

- How susceptible is WiFi RTT to multipath effects, NLOS signal reception and instrument bias?
- How does NLOS reception affect the accuracy of WiFi RTT measurements?
- What accuracy can be achieved when least squares-based positioning algorithms are used for pedestrian WiFi RTT-based positioning and navigation?
- What accuracy can be achieved when filtering-based positioning algorithms such as particle filters, genetic filters and grid filters are used for pedestrian WiFi RTT-based positioning and navigation?
- What accuracy and reliability improvements can be achieved for WiFi RTT-based positioning and navigation by using outlier detection techniques?
- What accuracy can be achieved when SLAM-based techniques are used with WiFi RTT for indoor pedestrian navigation?
- What accuracy can be achieved when SLAM-based techniques are used with WiFI RTT to estimate the position of WiFi RTT access points?

## 1.7 Structure of the thesis and contributions

This thesis consists of seven chapters.

- Chapter 1: Introduction, provides a brief outline into the indoor positioning problem and its importance. This is followed by WiFi RTT and positioning

techniques that can be used to harness this technology for indoor positioning. Then introduces the aims and objectives of the thesis.

- Chapter 2: Literature review and background, provides a review of the limitations of GNSS-based positioning techniques, followed by an exploration of various indoor positioning technologies and techniques. Finally, the chapter provides a detailed analysis of the state-of-the-art in WiFi RTT-based positioning research, as well as research that applied SLAM to WiFi RTT-based positioning systems.

- Chapter 3: WiFi RTT characteristics, aims to uncover insights into the causes of error, magnitude of error and behaviour of WiFi RTT in various situations. This is done with four experiments that aim to expose WiFi RTT to scenarios that may result in range errors. The first experiment bolstered existing research on the existence of instrument bias from the router and/or mobile device, which necessitates the need for calibration of most modern WiFi RTT compatible routers. The second experiment demonstrated that WiFi RTT was susceptible to multipath effects. An experiment specifically designed to identify multipath effects has not been done before this thesis. The third experiment supported evidence of WiFi RTT's susceptibility to multipath effects and NLOS signal reception, and that the relationship between the RSSI and RTT of the signal could be exploited to identify NLOS reception. Finally, this chapter explores the impact of orientation of the access point and the mobile device and whether an orientation-dependent ranging basis exists within the WiFi RTT products tested. To date, no research has explored the relationship of AP to mobile device orientation for WiFi RTT ranging accuracy.

- Chapter 4: Filtering techniques applied to WiFi RTT. This chapter applies various positioning algorithms to WiFi RTT across various experimental environments in both stationary and motion-based scenarios. These algorithms include single-epoch least-squares, weighted least-squares, particle filtering, genetic filtering and grid filtering. In addition, residual-based outlier detection and RSSI-based outlier detection are applied to the algorithms.

Work from this chapter was presented at ION GNSS+ 2023 [19] and is pending review as a journal paper in the Journal of Navigation.

- Chapter 5: WiFi RTT SLAM. This chapter presents and tests a SLAM algorithm applied to WiFi RTT, including RSSI-based outlier detection. Then a posterity SLAM algorithm is presented and tested which demonstrates the potential of cooperative SLAM techniques for WiFi RTT based positioning solutions. Work from this chapter was presented at ENC 2024 and ION GNSS+ 2024.

- Chapter 6: Conclusions, will conclude the result and outcomes of this thesis.

- Chapter 7: Recommendations for future work, will provide recommendations for future work of this research.

# CHAPTER 2

# Literature Review and Background

Navigation and positioning are an important enabler for a large number of activities in the modern day from vehicle navigation [20] and remote surveillance [21] to warehouse robotics positioning [22]. Navigation and positioning is achieved via a variety of methods such as the Global Navigation Satellite System (GNSS), WiFi positioning, inertial sensors and visual methods that take advantage of distinct environmental features. Each method has its own advantages and disadvantages which often depends on the environment. Some methods such as GNSS are highly effective in certain environments (such as large open fields) but ineffective in others (such as tunnels or large multi-storey office buildings). Given the proliferation of smart mobile devices, positioning in any environment is becoming an ever-growing concern. The navigation and positioning space has many problems to solve. This thesis will focus in on the use of mobile devices for navigation and positioning in areas where GNSS is not a viable solution in isolation, this includes most indoor scenarios or other GNSS degraded environments (refer to Section 2.1.3).

The goal of the review is to highlight the advantages and disadvantages of each solution and identify problems that need to be solved and researched further within

the subject area. This thesis focuses on positioning applied to navigation and tracking. Thus, the categories to be explored with respect to positioning are mobile bodies (although static bodies are used to understand the characteristics of signals and for algorithm proof of concepts), real time positioning (although post-processing is used in experiments, the positioning algorithms are aimed to be utilised in real time) and self positioning.

The following review of literature is structured by first reviewing basic positioning concepts and identifying the limitations of current popular positioning methods. Then the main positioning systems for GNSS degraded environments, specifically indoor environments, in use at the time of thesis submission are reviewed. Then the review will focus in on positioning techniques utilising a subset of tools such as WiFi, WiFi RSSI, Bluetooth, Ultrawide band (UWB) and WiFi RTT. The review will then focus primarily on WiFi RTT-based positioning methods alongside outlier detection methods and WiFi RTT Simultaneous Localisation and Mapping (SLAM) research.

## 2.1 Overview of Positioning

Positioning is the determination of the position of a body [1]. This has a broad definition in practise as positioning can be categorised in three ways. This is best represented in Figure 2.1 from [1].

Firstly, a "body" can be classed as being in motion or static. It is important to distinguish between the two classifications as it impacts the solution that must be adopted to position these bodies. For example, if an object is known to be static then it removes the complexity of the velocity variable from calculations as that object's velocity relative to the earth is known, whereas mobile bodies by definition will have unknown velocity so must be accounted for in calculations. As such, it is typically much easier to achieve higher levels of precision when positioning a static object because lots of measurements can be made and the noise can be averaged out. An example of static positioning is construction surveying. Some examples of

mobile positioning include smartphone tracking and vehicular navigation.

Next, positioning can be real-time or post-processed. With real time positioning the position of an object is determined as soon as possible once measurements are recorded, whilst post-processed position solutions are typically determined at any point after a positioning measurements are determined. With post-processed positioning there are more possibilities with what can be done with the measurement data and thus higher accuracy and precision can potentially be achieved which is a large contributor in why it is used for surveying and mapping.

Navigation applications generally use self-positioning whereby the object determines its own position using the positioning system or navigation system available to it. For example, smartphone positioning for navigation when using GNSS is self positioning as it is using that system to determine its own position.

The accuracy of most positioning methods depends on the environment, equipment, and algorithm. For example, WiFi-based positioning would be impractical in an open field away from any broadband network, whilst GNSS performs well in an open field. This means that different positioning solutions will have different value depending on the target environment and ultimately the most versatile solution is that which can utilise as many positioning methods as possible.

### 2.1.1 Overview of radio positioning technologies

Most positioning systems in practical use are radio positioning technologies. They utilise a radio system which functions to broadcast or even transfer information from one terminal to other terminals [10]. Three fundamental techniques that enable positioning via wireless positioning technologies are:

- Time of Flight (ToF) — The distance between a transmitter and a receiver equals the time of flight of the transmitted signal multiplied by the speed of light [10]. If the ToF method follows a Round Trip Timing (RTT) protocol then the time of flight of the transmitted signal multiplied by the speed of light

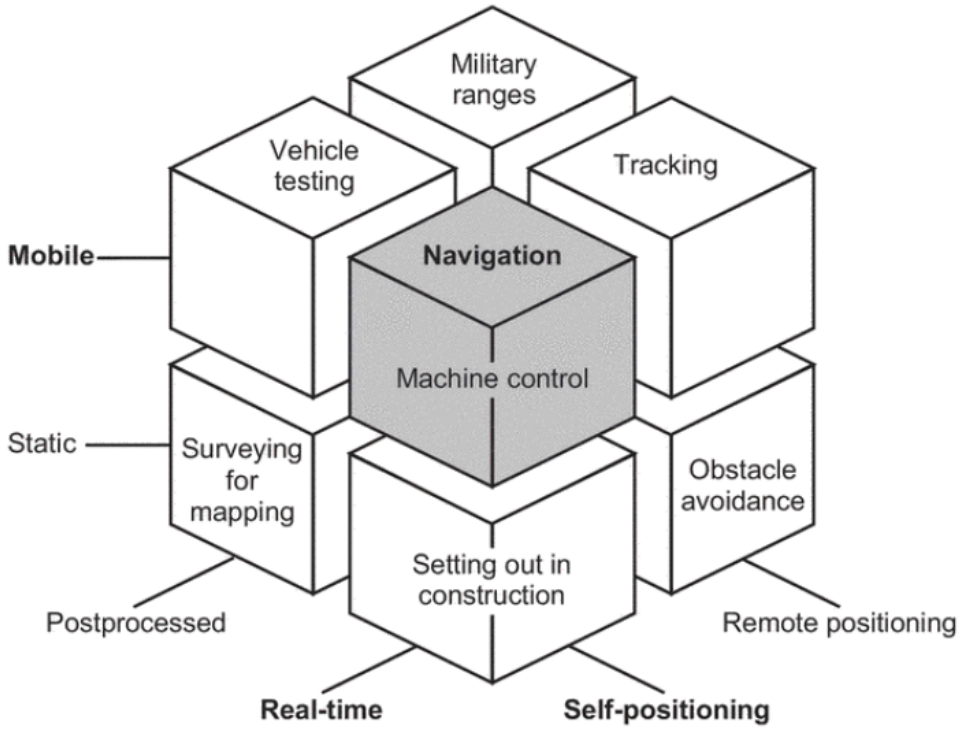


Figure 2.1: Categorisations of positioning and some applications [1]

divided by 2 represents the distance between the transmitter and receiver. There are other types of ToF measurement paradigms. The focus of this thesis will be on RTT. Before continuing it is important to discuss passive ranging which is the paradigm that GNSS follows. In this method the receiver measures the time of arrival of a signal that was transmitted at a known time, the difference between the times is then multiplied by the speed of light in free space to get the distance. This method requires the receiver to be aware of the transmitter's clock as if the clocks are not properly synchronised then the range will be underestimated if the transmitter clock is ahead of the receiver clock and overestimated if the transmitter clock is behind the receiver clock [1]. An RTT protocol allows a receiver to determine its range from an RTT compatible access point (AP). This is the basis of the WiFi Fine Time Measurement (FTM) RTT protocol. If multiple signal sources are introduced it is possible to position a receiver using the various distances. This will be discussed later in Section 2.2.2.

- Received signal strength (RSS) — RSS is the strength of a signal at the receiver. In free space, the intensity of radio waves is inversely proportional to

the square of the distance to the source of the radio waves. In non-free space obstacles distort this relationship. This means that with some calibration it is possible to determine the distance between a radio signal emitter and a receiver if only the signal strength is known. RSSI can be used for WiFi RSSI-based ranging (although WiFi RSSI-based fingerprinting tends to be more accurate) [10] [16], Bluetooth RSSI-based fingerprinting [23], amongst other methods. RSSI fingerprinting with respect to WiFi will be discussed further in section 2.2.1. The common issue with RSS is that due to interference, multipath and path blocking, location accuracy is generally less than what can be achieved by Time of Flight methods and furthermore when RSS is used it must be tailored to the environment in which it will be used due to the need to catalogue AP power variations or fingerprint surveying, resulting in operational expense [10]. Methods of how RSSI is used practically are discussed in Section 2.2.1.

- Angle of Arrival (AoA) also known as Direction of Arrival (DOA) - The wavefront of a signal is perpendicular to the direction of propagation of the wave. Position cannot be determined by one AoA measurement alone, multiple measurements or a combination of AoA and ToF are needed [10]. Whilst not related entirely to AoA, antenna rotation can be used in mobile devices to maximise signal strength (by rotating the entire device). This was noted during the experimental work of this thesis. During experimental work, the mobile device was always oriented to maximise WiFi RSSI and thus provide a more reliable RTT reading.

### 2.1.2 Overview of Global Navigation Satellite Systems (GNSS)

GNSS is a term for a type of navigation system that uses a group of satellites orbiting the earth and transmitting radio signals to provide any receiver access to a 3D positioning solution by passive ranging. GNSS is an example of a self positioning system [1].

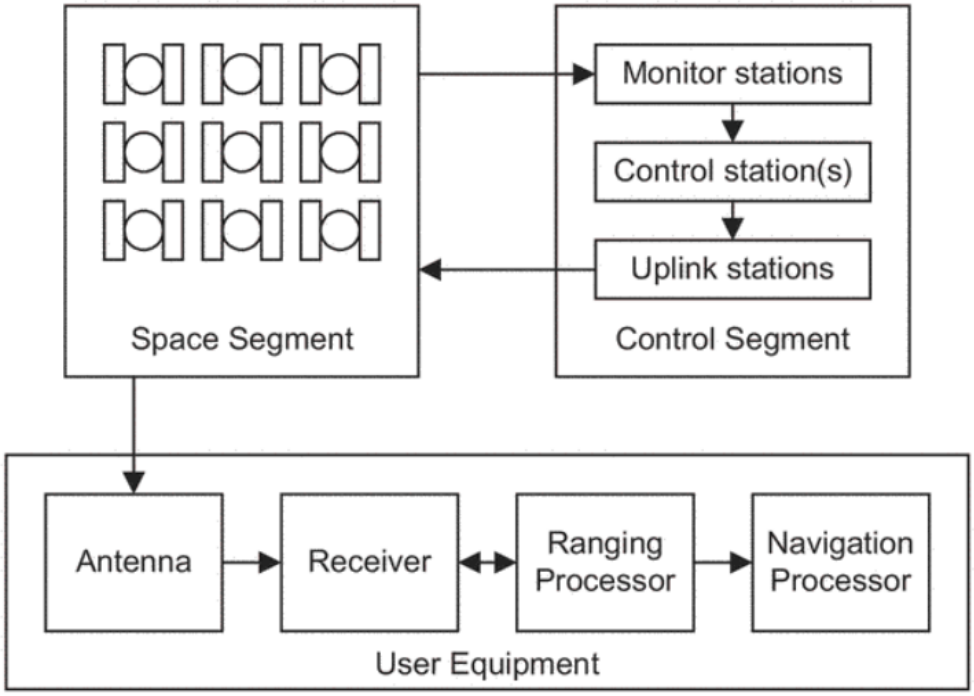


Figure 2.2: GNSS architecture [1]

Figure 2.2 describes the structure of a GNSS. GNSS consists of three components: the space segment, the control segment and the user segment. Each GNSS will have its own independent space and control segments whilst the user segment uses signals from multiple different GNSS constellations. The space segment consists of the satellites which broadcast signals to the control segment and users. GNSS satellites broadcast multiple signals on several frequencies. These comprise ranging codes and for most signals, navigation data messages. Ranging codes contain details on the time at which signals were transmitted whilst the data message includes timing parameters and information about the satellite's orbit. The control segment consists of stations at known locations and is used to determine the satellite orbits and calibrate the satellite clocks. This is also needed for determining whether satellites need to make any manoeuvres to ensure their orbit is correct. Consumer-grade GNSS user equipment used in mobile devices are designed to minimize cost and power consumption and are thus not as accurate as professional grade equipment [1].

GNSS positioning works by passive ranging in three spatial dimensions and one temporal dimension. One range measurement from a satellite will give a sphere of

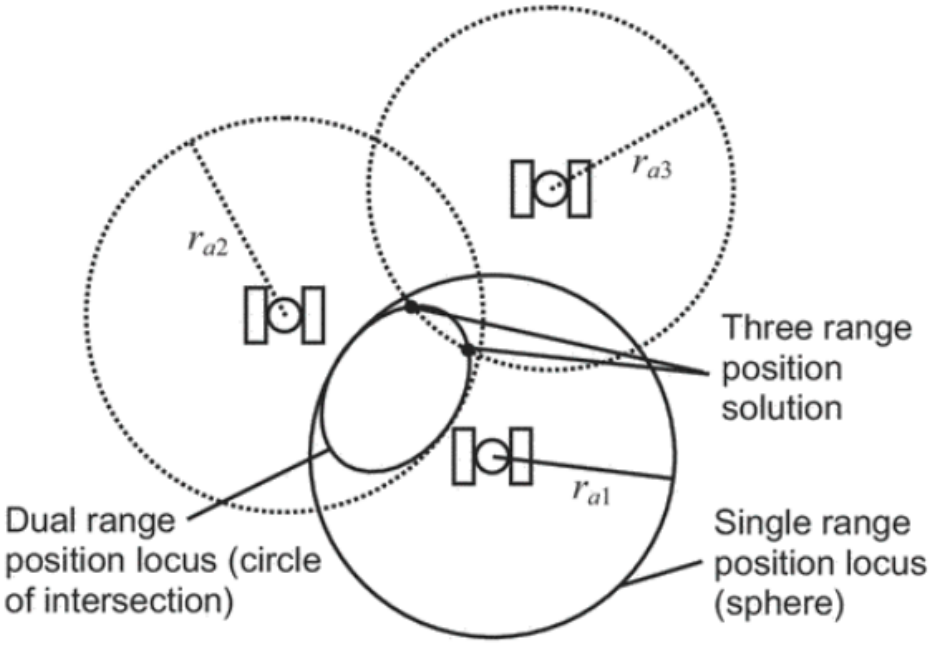


Figure 2.3: GNSS trilateration example [1]

possible positions around the satellite. Two range measurements will give a circle of intersection which represents the locus of the user position. A third satellite will give two possible positions of the device. In most scenarios one will be far more likely than the other as the other may be in a location where it is highly unlikely for the user to be. However, if both solutions are just as likely, then another ranging measurement is required. The case of three satellites is shown diagrammatically in Figure 2.3 [1]. Then in Figure 2.4 the case of adding another satellite to confirm the position (multilateration) is shown.

### 2.1.3 Limitations of GNSS

Figure 2.5 shows some GNSS error sources. Navigation solution errors arise from the differences between the true and broadcast ephemeris (set of data that contains position, velocity, and clock information for navigation constellations in the Global Navigation Satellite System (GNSS) and satellite clock errors. Refraction of signals caused by the ionosphere and troposphere result in pseudo-ranges larger than they would be in free space [1].

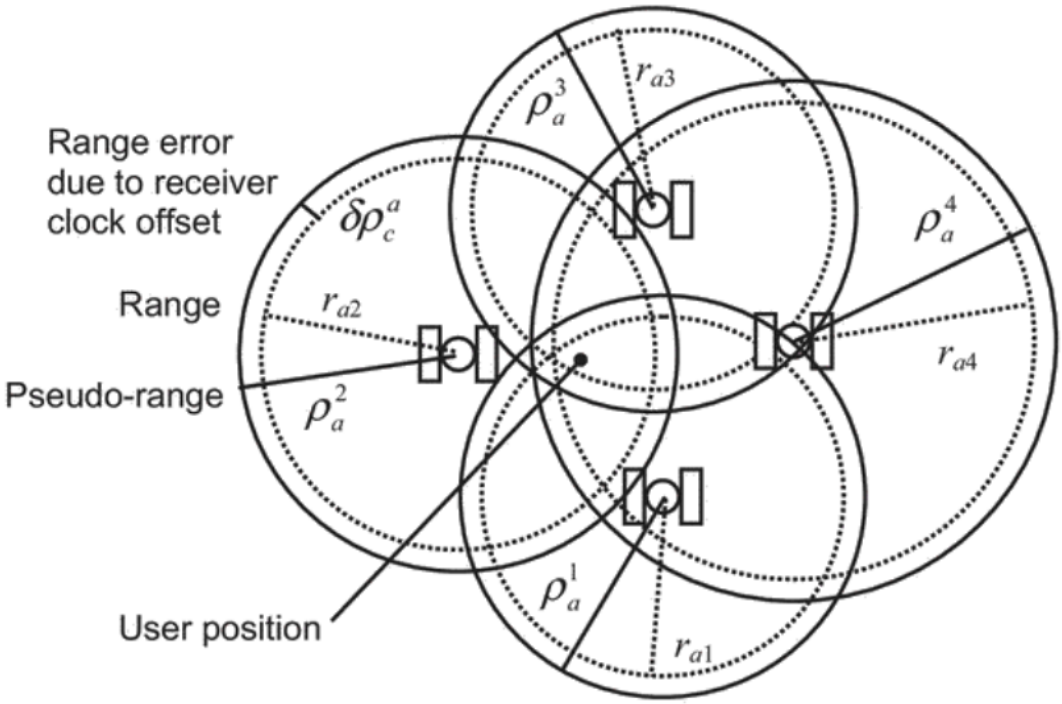


Figure 2.4: GNSS multilateration example [1]

Receiver measurement error can be caused by radio interference (RF) and multipath interference. Radio interference can degrade GNSS performance by disrupting a direct GNSS signal to a receiver, this can be caused by communication signals, strong signals in neighbouring and harmonic frequencies and even deliberate GNSS signal jamming. RF interference is particularly potent as GNSS signals are very weak when compared to most terrestrial radio signals. Multipath interference occurs when a signal reaches a receiver by more than one path as a result of GNSS signals being reflected by buildings, vehicles or the ground. This introduces another source of interference and is prevalent in urban environments as signals have more surfaces to reflect from due to a high concentration of buildings.

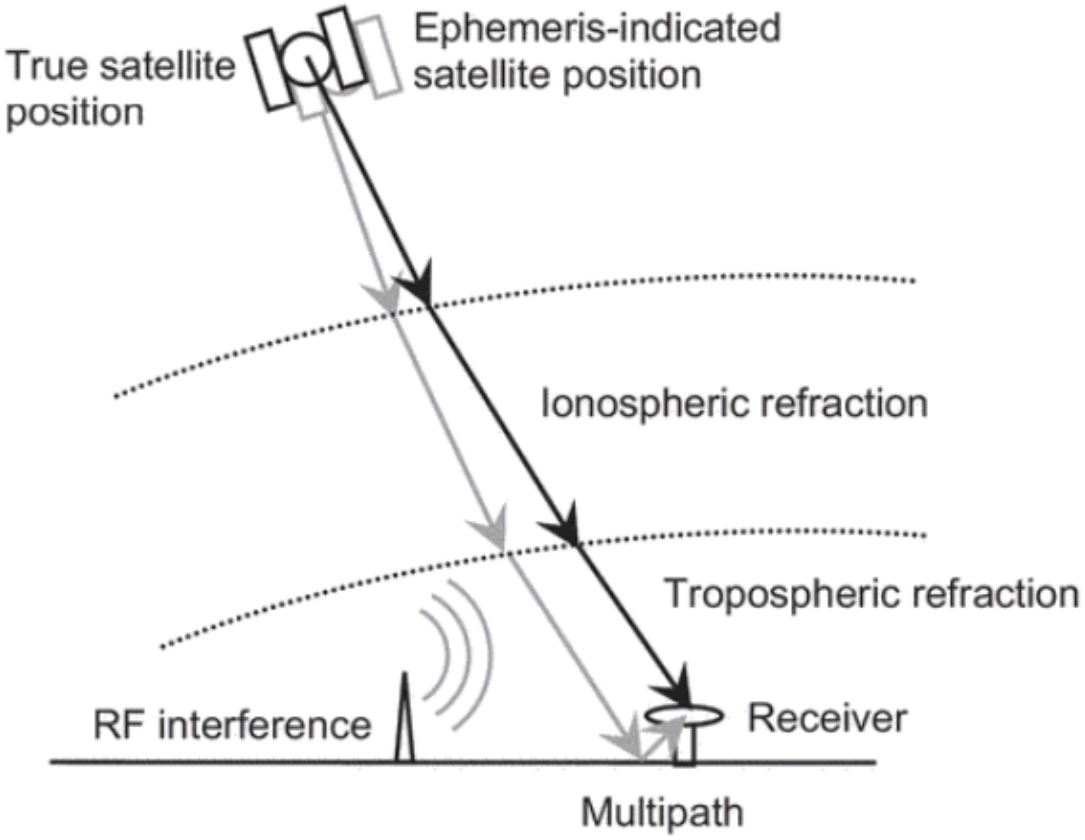


Figure 2.5: GNSS error sources [1]

Figure 2.6 focuses on the error sources in buildings, an example of a GNSS degraded environment. As discussed previously, multipath effects will result in the same signal being received via multiple paths by the receiver meaning that the signal received from different paths will overlap. GNSS transmitted signals can also be interfered with by attenuation from buildings. Attenuation will result in the signal being weaker as a result of the building materials. This will reduce the strength of the signal, thus the pseudo-range measurement will be inaccurate, and multipath effects are enhanced on some signals and diminished on others.

A receiver is also subject to non-line of sight (NLOS) signal reception. NLOS signal reception is another error source that involves the strongest signal received by a receiver being a signal that has reflected off of surfaces either outside or inside a building. This results in the signal propagation time between the transmitter and receiver to be longer than the propagation time of an unobstructed signal; this will

result in a positive pseudo-range error. Finally, in GNSS-degraded environments it is likely that some GNSS signals simply do not reach the receiver, this is a result of signals being absorbed and reflected by buildings. This further reduces the accuracy and reliability of the system as fewer received signals result in a position solution with fewer points of reference (satellites). For example, if there were 5 satellites transmitting whilst the user was indoors (in reality, typically, 20-30 are receivable outdoors), if at least 2 signals are reflected/absorbed or otherwise blocked then the solution cannot achieve a full position fix. If 1 signal is reflected/absorbed then there will be 2 equally possible positions, even though in most circumstances one will be more likely than the other (as one will most likely be inside the earth). These limitations demonstrate the need for a more reliable positioning solution for mobile positioning and navigation that can be used when GNSS is not reliable. The use of specific indoor positioning technologies will be discussed in the following section.

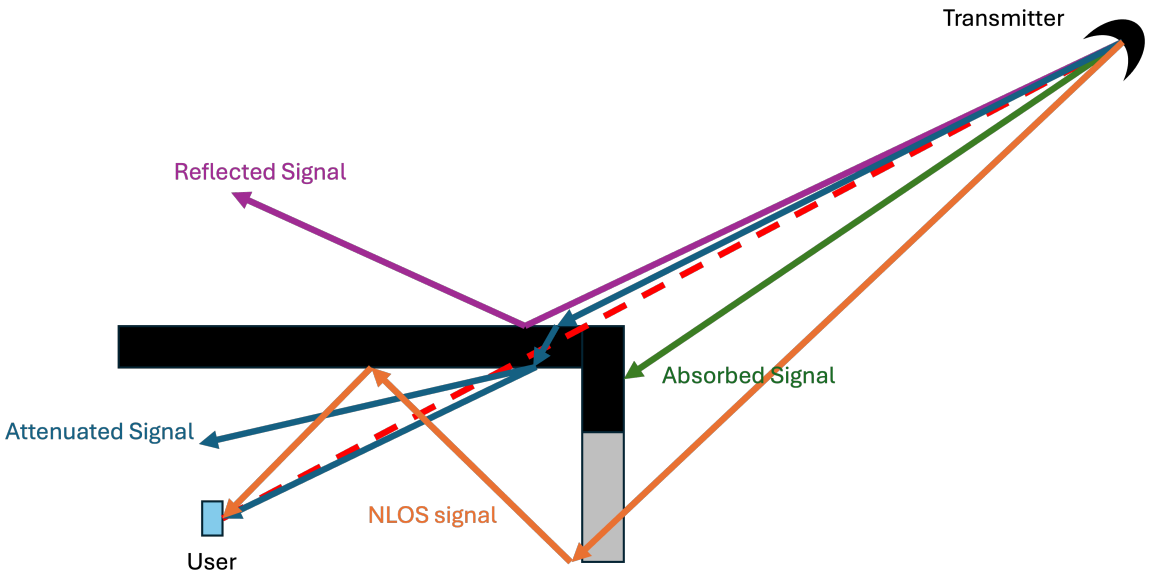


Figure 2.6: GNSS error sources in buildings

## 2.2 Overview of Indoor Positioning

An Indoor positioning system (IPS) is a system which can deduce the position of a device in buildings. However, with this term it is important to appreciate that the

technologies described can also be used for outdoor positioning and actually a ubiquitous positioning system that works both indoors and outdoors is more useful. Alternative methods to GNSS are required in order to improve the accuracy and performance of positioning and navigation indoors. Indoor environments are varied and dynamic so IPSs must be able to account for unpredictable environments. For example, people moving within an environment, doors being open or closed, moving furniture. The most researched indoor positioning systems will be explored within this section, these include WiFi RSSI-based positioning, WiFi Fine Time Measurement (FTM) RTT, Bluetooth, Ultrawideband, Visual Methods and Pedestrian Dead Reckoning.

### 2.2.1 WiFi

Wireless local area networks (WLAN) or WiFi networks are deployed in most indoor environments as they are the most popular infrastructure to connect devices in a building to the internet. WiFi signals are a form of radio signals and their primary function is to provide internet connectivity to devices that are on the WiFi network. As RF signals, they can also be used to determine the distance between an access point (AP) and an RF signal receiver such as a mobile device, this can currently be done via WiFi RSSI and WiFi FTM RTT. With three or more APs it is possible to use the signals from each of these APs to form a 2D positioning solution of a device (and four or more for a 3D positioning solution). This is typically done through multilateration, a technique that involves using the measured distance from at least three fixed points to determine the position of an object (using simultaneous equations), Kalman filtering and particle filtering. [10]. Another technique that can be used with WiFi signals is WiFi fingerprinting [24] [16] [25]. A common characteristic of WiFi, Bluetooth and UWB is that they operate using low power generally up to 100m [10]. Furthermore, some environments will have the IPS in that environment optimised for positioning such as in warehouse environments where the primary purpose of the solution is positioning. Whereas in most office buildings where WiFi may be the primary source of positioning, the network will not be optimised for positioning as the WiFi infrastructure is likely optimised for

connectivity. These will be discussed in the following subsections.

**WiFi RSSI**

WiFi RSSI represents the signal strength of the WiFi signals received by a device on the network. RSSI-based positioning techniques are separated into two approaches, fingerprinting and RF propagation loss model approaches. RF propagation loss models are the less performant of the two models as they rely on a relationship between RSSI and distance which then feeds into a multi-lateration algorithm to determine the positioning solution. This is not an ideal approach as RSSI can be heavily influenced by multipath interference and NLOS signal reception as a result of obstacles and surfaces. There is no agreed upon standard for conversion between RSSI and distance, and even then this is determined on a site-by-site basis in research or the accuracy limitations are accepted by the positioning solution [24]. Generally, the more accurate method of positioning that uses WiFi RSSI is fingerprinting.

WiFi fingerprinting takes place in two phases, the training/survey phase followed by a positioning phase. To summarise the process, the survey phase involves creating a database of detected WiFi signals at specific reference points (RPs), i.e. known locations in an environment. Each data point in the database will contain the RP coordinates and the corresponding RSSI of all WiFi signals detected at that RP. Therefore, each RP is characterised by the combination of RSSI measurements received at that RP. Once in a database these can be queried. [16], [26], [10], [24]. The positioning phase involves a device sampling the RSSI signals where it is currently located and comparing this result to a database on a server or to the device. The algorithm compares the received RSSI measurements from the mobile device with the fingerprints in the database collected from the survey phase. The server will return the most probable position of the mobile device based on the result of the pattern matching algorithm used. This means that a positioning fix can be determined without knowing the location of the APs nor the distance or angle between the APs and the device [26].

There are various pattern matching algorithms for comparing received vectors and surveyed fingerprints. Most commonly, variations of the K-nearest neighbour algorithm are used [26] [25] [16] but statistical Bayesian classification methods [27] [10] and convolutional neural networks (CNNs) [28] have also been used for RSSI fingerprint pattern matching.

K-nearest neighbours (KNN) [29] is a pattern matching algorithm that determines the K most similar instances of the measurement to the database reference points. Based on this, the target's position can be estimated to be at the location where the instances most closely match. Similarity is typically determined by a distance between the measurement and a reference point typically using Euclidean distance [30]. A great example can be seen in [10] from page 160-161. A process of kNN applied to RSSI fingerprinting is summarised as follows:

1. The RSSI vector at a given grid point is sent to the server that contains the fingerprint database. This is shown in Equation 2.1. Where P is the RSSI vector and $A_n$ represents RSSI in dBmW at the nth access point.

$$\boldsymbol{P} = (A_1, A_2 ... A_n)^T \tag{2.1}$$

2. Equation 2.2 [10] is then applied to the normalised received RSSI vector and the database to get the signal strength Euclidean distance vector differences, $\boldsymbol{D}$, between them.

$$\boldsymbol{D_n} = |\boldsymbol{S_T} - \boldsymbol{S_n}| \tag{2.2}$$

In Equation 2.2, $\boldsymbol{S_T}$ represents the received RSSI vector, the target signal strength read at each AP, $\boldsymbol{S_n}$ is a database signal strength vector with the same component layout as $\boldsymbol{S_T}$ and $n$ is the index of the reference point.

3. The smallest D values up to K (the number of reference points to include in the

nearest neighbour determination), are found, and a position is determined from the corresponding reference point by averaging the coordinates of the selected reference point (this average can be unweighted or can be weighted according to D).

Fingerprinting is dependent on establishing a reliable database of fingerprints which requires upfront effort and expense to construct. Furthermore, due to dynamic environments it is also possible for the fingerprint database to be less reliable as the fingerprints of a location may change after the time that the data was taken [1]. In [16] the fingerprint database of an underground train station was collected, but it was found that trains entering and exiting the station could alter the expected fingerprint of a specific reference point, indicating that dynamic environments can disrupt the reliability of a fingerprint map. This could be solved by surveying a location at multiple different times and then storing the means and standard deviations of the RSSI measurements in the database.

A way of improving the performance of fingerprinting for positioning is by integrating more signal measurements at reference points in the database. For example, using Bluetooth signals and WiFi FTM RTT ranging measurements [31] [32].

**WiFi FTM RTT**

WiFi Fine Time Measurement (FTM) Round Trip Timing (RTT) (referred to in this thesis as FTM RTT, WiFi FTM RTT or WiFi RTT) is a feature included in the IEEE 802.11mc WiFi protocol in 2016 [33]. FTM RTT involves measuring the time taken for a signal to be transmitted and received from a mobile device to an AP and then the AP to the mobile device. RTT does not require clock synchronisation as the time taken to send and receive an RF packet is measured instead [34].

This is represented visually in Figure 2.7. Since radio signals travel at the speed of light if you multiply the time taken by the speed of light and divide this value by 2 you can get the distance between the device and the AP. With multiple ranges multi-lateration can be used to determine the position of the device. Google

suggests that with about 4 APs 1-2m accuracy can be achieved for most buildings [2]. A ranging request consists of 8 bursts of the FTM requests, then a mean and standard deviation of these 8 bursts is given to the user for the ranging result. It has been demonstrated that WiFi RTT is susceptible to multipath effects and NLOS reception [34] [35]. Thus, applying mitigation techniques from the GNSS community poses a great opportunity for WiFi RTT research and application. WiFi RTT is an interesting research area, research on the technology is limited, and it also has high potential as a core aspect of indoor positioning systems. This is because of widespread WiFi access points in indoor environment and higher accuracy when compared to WiFi RSSI RF loss propagation models [36] (these are compared as the same amount of set-up is required for both methods).

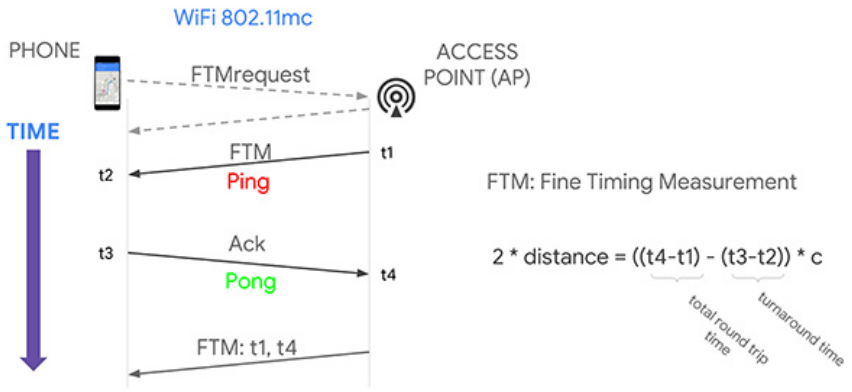


Figure 2.7: WiFi FTM RTT in practice [2]

### 2.2.2 Pedestrian Dead Reckoning

Pedestrian navigation is a complex application of navigation technology as it must work in varied environments, both indoor and outdoor. At the same time devices must be mobile, so will be low-cost, lightweight and low-power. Inertial Measurement Units (IMUs) are devices that contain accelerometers and gyroscopes, most will contain three accelerometers and three gyroscopes each limited to one sensitive axis. An accelerometer measures specific force, essentially non-gravitational acceleration and not the total acceleration of the IMU housing the accelerometer. A gyroscope measures angular rate of the IMU with respect to inertial space. With current technology smaller IMUs found in smartphones (or other pedestrian devices)

will have low inertial navigation performance [1].

Pedestrian Dead Reckoning (PDR) involves step detection [1]. Using an IMU, more specifically the accelerometers in an IMU and optionally, the gyroscopes, it is possible to identify steps made by a pedestrian holding a device. In this thesis, mobile devices are assumed to house the IMU as opposed to externally mounted IMUs [1]. PDR is made up of three phases: step detection, step length estimation and navigation-solution update. Put simply, steps can be detected by identifying points where the specific force output by the accelerometers cross acceleration due to gravity or from peaks in the accelerometer signals [1]. The step length determination process has many variables that would result in an assumed step length to be invalid, such as terrain, whether the pedestrian is running, the pedestrian's size, slope and obstacles. This problem can be solved by using context identification and then selecting the appropriate step length model [1]. PDR is rarely used in isolation as a positioning solution as the error in distance travelled grows as you travel further. Thus, PDR is used in combination with other solutions such as WiFi RTT, WiFi RSSI fingerprinting and GNSS.

### 2.2.3 Bluetooth

Bluetooth is a type of Wireless Personal Access network (WPAN) that uses radio frequency signals to transmit packets of data between Bluetooth compatible devices. Bluetooth positioning systems typically include a set of Bluetooth beacons distributed throughout an environment and a Bluetooth compatible mobile device. Some Bluetooth methods adopt a proximity approach where if a device connects with a Bluetooth beacon then it is assumed that the device is within 10m. Then as more beacons connect or disconnect from the Bluetooth compatible mobile device it is possible to determine the position of the mobile device by determining the overlapping constant radius circles [10] [37] [38]. Thus, in this method it is assumed that the location of the Bluetooth beacons is known. Additionally, the average time to determine a location estimation off of five Bluetooth beacons was 19.2 seconds. This is long compared to the time taken for a user walking at an average walking

speed to pass through a 20m diameter circle (the range of a single Bluetooth beacon) [10].

The more up-to-date Bluetooth positioning technology involves Bluetooth Low Energy (BLE), a feature of Bluetooth 4.0. This method follows a similar paradigm to WiFi RSSI, whereby the two primary positioning methods are RF propagation loss and fingerprinting. BLE improves Bluetooth based positioning as the beacons used require less power and will last longer in an environment and enquiry from the target to the transmitter are significantly faster between the beacon and receiver [1]. Furthermore, BLEs will provide the RSSI of the signal [1]. However, a common problem with Bluetooth beacons is that they require prior set-up and thus have setup and maintenance costs. This means they are unlikely to be used in environments where positioning is not an important requirement.

### 2.2.4 Ultrawideband

Ultrawideband (UWB) communication systems use bandwidths larger than most cellular and WLAN wireless networks, so absolute bandwidths of at least 500MHz [1]. The wide bandwidth translates to higher resolution timing of pulse arrivals, so multipath effects are easier to distinguish. This is because the correct signal (at 1GHz for example) and multipath components with a differential path delay greater than a certain threshold can be separated [1]. Furthermore, UWB NLOS reception is less severe as it is possible to distinguish between an LOS estimate and NLOS estimate as the variance of the TOA is greater for the NLOS estimate [10]. UWB signals typically provide ranges by following the ToF paradigm and thus can use similar methods to particle filtering. Generally, UWB range measurements can achieve cm level accuracy in LOS conditions [10]. However, the power of the signals are restricted by regulated power constraints and are thus limited to short range environments [10]. Moreover, UWB access points suffer from the same issue as Bluetooth beacons whereby devices must be actively maintained, and a cluster must be set up.

## 2.3 Existing research on WiFi RSSI and WiFi RTT positioning

### 2.3.1 WiFi RSSI-based ranging

As discussed previously, WiFi RSSI-based positioning follows two key paradigms, RF loss propagation ranging or fingerprinting. This section will go over RF loss propagation. An early paper by Kotanen et al. [39] explored WiFi-RSSI positioning using an RF loss propagation model and an extended Kalman filter. A Kalman filter is an algorithm that takes a series of measurements observed over time such as signal strength, ranges, IMU data as well as noise and bias then produce an estimate on these unknown variables depending on how those data points have changed [40]. The measured error of this system was 2.6m on average. Kotanen et al. deduced that this was inaccurate for mobile pedestrian positioning as in most cases it would not position a mobile device within the correct room but could be suitable for indoor proximity detection or determining the correct floor that a device was on in a building. The errors were caused by several factors including noise, multipath propagation, interference and obstacles, factors for which the RF loss propagation model used could not take into account. Furthermore, the coordinates of the APs, the TX power levels of the APs and other information that affects the propagation model must be determined prior to positioning in order to achieve a reliable positioning solution. This phase increases the operational expense of this method.

Bose et al. [41] conducted experiments on using RF loss propagation from WiFi routers for positioning. They first constructed a model of the correlation of RSSI to distance and found that the signal strength value shows an exponential decrease with respect to distance. Applying the experimentally deduced trend line as the RF loss propagation model to the various APs with trilateration found that in LOS conditions the average positioning error was 2.3m whilst in NLOS conditions the average positioning error was 2.9m. Again, two issues with this are the required knowledge of the AP locations and the RSSI behaviours of each AP.

### 2.3.2 WiFi RSSI-based fingerprinting

A common problem within fingerprinting is that a changing environment impacts the performance of an RSSI fingerprint database. This is backed up by Bensky [10], Zixiang Ma et al. [42], Ma et al. [16], and Yiu et al. [43].

Yiu et al. [43] conducted experiments on WiFi RSSI fingerprinting using K-nearest neighbours. Roughly, the RMS localisation accuracy achieved by using 4-NN was between 5.4m and 8.3m for different length scenarios with the largest RMS localisation being on the longest path. Chengqi Ma et al. [16] researched the benefits of applying a filtered solution alongside a dead reckoning system. These two additions are both designs that can improve all WiFi based positioning solutions. Therefore, they are discussed here to represent their improvement in comparison to WiFi RSSI fingerprinting. These experiments were conducted in the London Underground and found that both trains and people affected the accuracy of the positioning solution. The experiments compared a KNN matching algorithm to a Kalman Filter fingerprinting implementation combined with PDR to provide a velocity update during the correction stage of the Kalman filter. During peak congestion times when there are more people and more trains there was an average positioning error of 2.9m whereas in low congestion times the error was 1.7m. It was also observed that during the off-peak times that the maximum average positioning error discrepancy between the two methods was 0.47m whilst for peak times the maximum discrepancy between average positioning errors of the methods were 2.29m, both in favour of the Kalman Filter and PDR method. This indicates that the Kalman filter algorithm and integration of PDR (which is less dependent on the environment) was a more performant positioning method. Dong et al. [44] developed a multimodal fingerprinting model that used WiFi RSSI fingerprints but identified that most models do not account for the geometric relations between WiFi APs and the location of the fingerprint. Having this information allows for a more granular understanding of the features of the fingerprint. The algorithm was able to provide a median positioning error of 2.1m. This thesis also suggested that future work could involve fusing WiFi RTT data into the fingerprint map.

Dai et al. [45] implemented a method to autonomously collect WiFi RSSI fingerprints using a robot to roam an environment. It used visual methods and IMUs to construct a map and then collected fingerprints throughout the experimental environment. This method saved 68% time on average when compared to manual RSSI fingerprint collection. Due to fingerprint maps changing it is likely that surveyed maps would need to be changed in the long term, meaning this specific solution is potentially not cost-effective. An issue with fingerprinting is the data collection aspect of the method during the positioning phase. This is an issue for fingerprinting and any sort of surveying based method. Therefore, methods which can reduce or even remove the surveying phase present an opportunity to make indoor positioning solutions easier to deploy.

### 2.3.3 WiFi RTT-based ranging

WiFi RTT ranging was popularised by Diggelen et al. in an article called "How to achieve 1-meter accuracy in Android" [2] where it was announced that the WiFi RTT protocol would be publicly enabled in all subsequent Google WiFi-aware Devices. A demonstration of WiFi RTT was carried out using multi-lateration as the positioning algorithm. A total of eight bursts of FTM packets are transferred between the mobile device and an AP. This allows the system to calculate range statistics such as mean and variance which enables a better accuracy than if one burst alone was used. It was noted in this article that WiFi RTT could have range calibration offsets up to 0.5m [2] and also suffered from multipath effects which will both increase the error in a positioning solution.

Garcia-Fernandez et al. [46] researched a solution to help identify and account for instrument bias present in WiFi Access Points by providing a database of the AP location and bias to other users once one user had interacted with an AP. The existence of instrument bias has also been noted by Diggelen et al. [2], Ma et al. [35] and Horn et al. [47]. The research in [46] involved carrying out a test on 5 Access point in an outdoor environment (open sky without obstruction between the mobile

devices and the access points) in order to test in an environment where multipath effects are reduced as much as possible. The results showed an instrument bias between 1 and 1.5m. When a second device used the database containing the AP locations and the AP bias after the first device had gathered this information it was found that the horizontal positioning error was within 10cm of the manually calibrated method. However, the vertical and height errors were greater by 1.2m and 3m on average. The vertical and height error position dimensions were larger because of Dilution-Of-Precision (DOP) which causes an error amplification when the geometry of the receivers and transmitters are not diverse [1] (i.e. all observations aligned). A paper by Dong et al. [48] explored error on WiFi RTT devices. The paper investigated the error of WiFi RTT. It was found that in all scenarios a hardware-dependent bias was present. With obstructions between the mobile device and AP it was found that obstructions made of metal caused a ranging error between 2m to 2.5m greater than a glass or wooden obstruction. These biases did not change depending on whether the range or position varied. The standard deviation of the range estimate increased by roughly 6 times when both the range and position varied when compared to no range variation with position variation and almost 30 times compared to no range or position variation.

Another source of error that was highlighted by Jurdi et al. [49] is that the body of a pedestrian blocking a WiFi RTT signal can result in as much as an 8m increase in ranging error with larger fluctuations. This demonstrates the susceptibility of WiFi RTT to NLOS signal reception error. This error was also identified by Mohsen et al. [50] who used the effect of body blockage to identify the presence of a human in an environment regardless of if they were carrying a device or not. This is because they identified that a human blocking a signal would result in an outlier ranging error that would not normally be present, thus indicating the presence of a moving object such as a pedestrian.

Ma et al. [35] explored a variety of positioning algorithms that used only WiFi RTT. This included recursive least squares (RLS), weighted concentric circle generation

(WCCG), clustering based trilateration (CbT) as well as standard iterated least squares (using multilateration). RLS is a version of sequential least squares that finds coefficients to minimise a weighted least square cost function at each step. WCCG is a method which provides a solution to the case where there is no horizontal intersect between the ranging circles of two APs in multilateration. Instead, by applying probability weighting to a set of concentric circles around the AP it increases the chances of finding an intersection point and also provides a weighting for the reliability of that intersection point, providing a form of outlier detection. The method can be thought of increasing the thickness of a concentric circle and applying a probability distribution as a gradient on the thickness. CbT is an algorithm that searches for high densities of AP ranging circle intersection points and assigns a higher probability to that location for being the position of the device. It can be used for identifying anomalous points as those points will be excluded from the clusters if they do not meet the neighbouring point requirements. All variations of the algorithm outperformed basic least squares. CbT and WCCG with instrument bias not removed performed very well, as there was only a 10cm difference between the same method with instrument bias removed. A shortfall of this model is that knowledge of the AP locations is required.

Guo et al. [51] focused on a method that incorporated both RSSI ranging and RTT data into a positioning solution. The algorithm used was a Kalman Filter with RSSI path loss ranging and RTT data used as inputs to the filter. For static positioning it was found that the model started with a mean positioning error of 3.41m for RSSI based fingerprinting. RTT ranging on its own had a mean positioning error of 2.042m whilst the ranging error for a model that integrated both RSS and RTT into a Kalman Filter had a ranging error of 1.435m. The inclusion of RSSI path loss ranging into the Kalman filter had the effect of identifying outliers as if there were a large disparity between the RSSI path loss range and the RTT range then the measurement would be removed from the solution. The inclusion of RSSI data gave the Kalman filter more information for the measurement update phase. Furthermore, it demonstrated how both metrics together can improve the overall positioning solution, as WiFi RTT-based trilateration produced a positioning error

of 2.04m. As with most RTT-based positioning and RSSI loss propagation based positioning, the locations of APs were required knowledge.

Dong et al. [52] developed a technique to identify NLOS signals in real time. The paper used machine learning algorithms and resulted in outlier detection of over 96%. The machine learning algorithms explored were random forest, least square support vector machine and a deep neural network approach. The experiment involved smartphones at several stationary locations with 3 APs. Two test sites were used, an office site for training and testing and then a student accommodation environment for validation. The machine learning algorithms were essentially intending to determine whether a given signal was from an NLOS or LOS signal. The models were trained with different combinations of datasets of RSSI and RTT range data. It was found that only using RTT range data or only using RSSI could not provide good performance for NLOS/LOS signal identification. This paper found that the features extracted from the RTT range data are more helpful for NLOS identification compared to RSSI. It was also found that the deep neural network model had the best detection accuracy but had the highest computation complexity and thus took the longest time to train. Overall, this paper takes a different approach to NLOS detection than other papers which focus on RSSI and found that RTT range-based features might be better for outlier detection for RTT based positioning.

Han et al. [53] implemented an algorithm that integrated PDR with WiFi RTT to create a novel positioning solution. The model harnesses PDR to conduct trajectory alignment and step length estimation to test the validity of positioning updates that are suggested by WiFi RTT multilateration. The accelerometers of the mobile's IMU can be used to determine distance travelled by counting the number of steps and multiplying this by a step length. The gyroscope can be used to recognise turning direction. By combining these methods it is possible to obtain a user's trajectory. Han et al. suggested that by combining PDR and WiFi RTT into a Kalman Filter it is possible to use both the WiFi RTT measurements and IMU

sensor data to reach a position estimate. It was found that while walking in a straight line in LOS environments the proposed method of WiFi RTT positioning combined with an Extended Kalman Filter (EKF) had a positioning error of 1.359m.

Another research paper that used PDR in combination with WiFi RTT (referred to as FTM ranging in the paper) by Sun et al. [54] integrated PDR (specifically step detection and length estimation) with WiFi RTT in an extended Kalman Filter. The model in this paper also used RSSI for outlier detection by determining the standard deviation of the RSSI and ranging data in one second. If the standard deviation was above a certain threshold then the data point was removed from consideration for the positioning solution. This method achieved an RMSE of 1.1m. This is better than the EKF alone which achieved an RMSE of 2.74m.

Si et al. [55] proposes a model using weighted least squares for RTT based positioning and feeds the difference between a predicted RSSI and the measured RSSI, the RSSI variance, the distance measurement and the RSSI measurement into a Naive Bayes Classifier. The predicted RSSI was acquired by collecting RSSI data in the environment at various distances from an AP and then fitting a double exponential model to this data such that this model gave the best possible prediction for the relationship between RSSI and the true distance. The Naive Bayes Classifier then classifies the signals into NLOS or LOS signals. Then a probability is applied to these classifications which is used for the weight matrix during weighted least squares. This makes sense as it uses the RSSI measurements for an RTT signal as a proxy for reliability of the RTT measurement which will be able to distinguish between NLOS and LOS signals. This solution outperformed traditional least squares positioning with no outlier detection by 29.1% in terms of mean position error and is also practically a good solution as RSSI data is provided with all RTT measurements.

Cao et al. [56] developed a solution that focused on the identification of NLOS signals and a LOS ranging calibration model to improve LOS range accuracy. They

were able to achieve a mean absolute error of 1.082m. The NLOS identification model followed an RSSI-based outlier detection model that use a path loss model calibrated to the environment being tested in.

The recent research on WiFi RTT points to the fact that, as expected, WiFi RTT based positioning performs significantly better when other sensors and WiFi RSSI are introduced. This is because these signals help with identifying outliers from the WiFi RTT ranging results. Since these sensors and signals are present when WiFi RTT is operational, it makes sense that any research going forward should focus on fusing the various sensors and signals. It is worth noting, the work in this thesis was done in parallel to some of the research discussed above. Specifically, work in this thesis explored RSSI-based outlier detection around the same time as other researchers.

## 2.4 Existing research on WiFi-based SLAM for indoor positioning

Simultaneous Localisation and Mapping (SLAM), first discovered by John Leonard and Hugh Durrant-Whyte [18], is a procedural method to build a map of an unknown environment while at the same time navigating that environment using the map. SLAM uses sensors within the device being positioned, such as IMUs, visual data, signal data etc. This section will focus on the application of SLAM to mobile devices where WiFi is a key component of creating the map of the WiFi AP locations in an environment. SLAM is a solution to the problem of requiring previous knowledge of the environment in order to carry out positioning. SLAM can enable mobile devices to position themselves in any building where the signals that it uses to construct a map of the environment are present.

A paper by Faragher et al. [57] demonstrated WiFi SLAM. A positioning technique for an indoor environment where an initial GPS position fix was used then WiFi signals and IMU data was used in combination with SLAM. The IMU in the device

was used for step detection (accelerometer) and as a compass (magnetometer). Using an assumed step length it is possible to roughly calculate the distance moved and heading of a device. However, in this experiment the step length for each particle was randomly assigned within a range to account for different step lengths and noise was applied at each epoch to account for changes in minor changes in step length. Then various other signals were polled every second including WiFi RSSI, GNSS and cellular measurements. As with most SLAM systems, a particle filter was used for navigation and mapping. A particle filter, based on sequential Monte Carlo can be used to estimate the position of a mobile device. The algorithm takes a number of particles to represent a distribution of likely states. Whenever the device moves the algorithm predicts the new state based on the movement (deliberately adding noise during the process) and then compares this prediction with the measured state to determine how well they correlate. The weighted average of all particles should be a good estimate of the actual position of the device. For a more thorough description of the particle filter refer to Section 4.1.3. In this paper, the particle filter is initialised using a GPS position, its associated uncertainty, the average step length and compass bias. As the device moves from the starting point the position solution from the IMU becomes less accurate. This is because error in the IMU measurement will affect the predicted state which will affect the predicted state of the next IMU measurement and so on. Essentially, over time the error in predicted state increases if left uncorrected.

However, the position solution corrects itself when the device passes a location it has better certainty on, either from a more reliable GNSS signal or because the device has already been at that position (thus the WiFi signal signatures at that location match). Loop closure (or a device returning to a location it has previously visited) is a key part of SLAM, it is the primary way to remove the drift accumulated from the IMU errors that accumulate over time. By returning to a previously travelled reference point it is possible to reset the position of a device and also adjust the rest of the map to account for the drift error. For this method, over a 15-minute walking period the final position error was 4 metres compared to an 86m error for an

uncorrected particle filter solution. Given that the SLAM solution has no prior knowledge of the environment, this is a good result, especially since the final position error is not reflective of the position error throughout the positioning phase which was lower but not measured in the experiment.

A paper by Ferris et al. [58] also explored WiFi SLAM. The results of this paper yielded an average position error of 3.97m with a standard deviation of 0.59m. This model however used a Gaussian Process Latent Variable Model, a technique used for mapping high-dimensional data (signal strength information for all WiFi APs in the environment in this case) to a low-dimensional latent space (a two-dimensional latent space in this case [xy coordinates]) [59]. A paper by Liu et al. [60] focused on WiFi SLAM that integrated visual methods through Google's Tango (now known as ARCore [61]). This tool enables mobile devices to combine visual input from the Tango's camera which had better optical sensors than an ordinary phone camera with IMU data to track the movement of the device. This paper used a particle filter as the positioning engine and ran two experiments. The first experiment used WiFi RSSI and step-detection-based PDR and the second experiment used WiFi RSSI and Tango-based PDR using visual inertial odometry. The Tango-based-PDR required the user to hold the phone upright such that the phone's camera could view the environment to conduct visual inertial odometry. The Tango-based-PDR and WiFi SLAM yielded a average positioning error of 0.6m which is better when compared to step-detection-PDR and WiFi SLAM which yielded an average positioning error of 4.76m. The results of the Tango/visual-based model are impressive and has its benefits for specific mobile positioning tasks. However, in most cases of mobile pedestrian navigation it is unlikely that a user will be holding their phone upright with the camera on, especially considering the power requirements.

WiFi RTT SLAM has not been explored in much depth at the time of writing. WiFi RTT SLAM was explored by Gentner and Avram [62] in 2021. This positioning solution used WiFi RTT signals and IMU data (gyroscope and accelerometer)

processed using a particle filter. As a mobile device moves around an environment it is possible to narrow down the location of each AP to a specific area, thus enabling map creation. The WiFi RTT SLAM positioning algorithm used in this paper yielded an average positioning error under 1m. Gentner and Avram did not explore any outlier detection methods and more importantly did not use a pedestrian dead reckoning model for estimating the movement of the pedestrian, instead using a set of fixed markers with a known position in the environment. This is an improvement on WiFi SLAM when compared to WiFi RSSI SLAM and a good positioning performance. This paper is a great proof of concept for WiFi RTT SLAM and more research on this technique to explore alternative filter methods, outlier detection models and more sensor fusions would advance the field significantly.

# CHAPTER 3

# WiFi RTT Characteristics

This chapter describes the tests undertaken to understand the characteristics of WiFi RTT and how the protocol behaves in certain conditions and environments. Each test aimed to identify a characteristic of WiFi RTT and these results have been used to guide the best methods for data collection, device calibration, outlier detection and the positioning solution. The tests undertaken include:

Experiment 1: Basic WiFi RTT AP to mobile device range test between 0 - 5m at 0.5m intervals. The objective of this experiment was to identify the presence of any instrument bias in the AP or mobile device.

Experiment 2: Incremental WiFi RTT AP to mobile device range tests between 1 - 1.5m at 0.02m intervals comparing results with the devices on a reflective surface against the devices on a dull surface. The objective of this experiment was to identify the impact of multipath effects for WiFi RTT.

Experiment 3: Various WiFi RTT AP to mobile device ranging non line of sight and multipath scenarios, although multipath scenarios will be present in Experiments 1 and 2, the aim of this experiment is to enhance multipath effects through the setup of the experiment. The objective of this experiment was to understand the effects of building attenuation and non-line of sight signal reception on WiFi RTT.

Table 3.1: Devices used during WiFi RTT Ranging Characteristics experiments

| Access Point/Device | Description | MAC address |
|---|---|---|
| AP-1 | Google Nest WiFi Router | cc:f4:11:4b:fb:1b |
| AP-2 | Google Nest WiFi Point | b0:e4:d5:1a:85:5d |
| AP-3 | Google Nest WiFi Point | b0:e4:d5:01:eb:5d |
| Mobile Device | Google Pixel 4a | |

Experiment 4: WiFi RTT AP ranging error based on orientation of the mobile device compared to the AP. The objective of this experiment was to determine if there were ranging errors depending on the orientation of the access point relative to the mobile device.

This chapter will first describe the devices used. Then each experiment will be discussed including an introduction, experimental setup and a discussion of the results.

## 3.1 Devices Used

The devices outlined in Table 3.1 have been used for the experiments in this thesis. These devices are all produced by Google, a supporter of WiFi RTT. These devices have been used as they are WiFi RTT compatible and are cost-effective. Henceforth, the access points will be referred to by their AP designations (AP-1, AP-2, AP-3). The MAC (Media Access Control) address of each AP is important for distinguishing the different devices. The router, AP-1 is a device connected directly to the broadband line and a power outlet, whilst the points only connect to a power outlet and are reliant on the router (AP-1) being online in order to function.

## 3.2 Experiment 1 - Instrument Bias

### 3.2.1 Introduction

This experiment aims to determine whether a fixed range offset exists between actual distance measurements and FTM range measurements for 3 different Wi-Fi

FTM/RTT capable Access Points, and if so, what this range offset is.

After setting up three Wi-Fi RTT (Round Trip Timing) capable Google Nest Access Points (1 Google Nest Router and 2 Google Nest Points), it was observed that a mobile device's Fine Time Measurement (FTM) ranges would have a constant negative bias when utilising the Wi-Fi RTT capabilities of the access point. For example, a mobile device one metre away from the Access Point (AP) would return a negative range, indicating that the FTM range was being underestimated. This required further investigation to determine the necessity of calibrating the ranging measurements to improve the accuracy of the system. This experiment did not test different mobile devices.

### 3.2.2 Experimental Setup

The experimental layout is shown in Figure 3.1. A mobile device and an AP were placed with no obstacles between them and no walls within 3m. Ideally there should be no walls, but this is not possible given constraints of the experimental environment. Both devices were at the same height. These conditions exist to ensure that the device and AP have an unobstructed line of sight and also so that reflections off of nearby surfaces can be mitigated. The mobile was moved from the AP in 0.5m increments from 0 to 5m. At each increment the FTM RTT readings from the mobile device over a 30-second period were logged. The logging included the timestamp of the reading, the measured range, the measured range standard deviation, the number of successful FTM bursts, RSSI, AP name and channel frequency (MHz). The reference range between the AP and the mobile device was measured using a laser measurer (a measuring tape would also suffice). For all readings the channel frequency was 5180MHz, this was the default presented by the WiFi RTT logging mobile app used. This process was repeated for three APs, all in the same location for their respective measurement period. This experiment was repeated 5 times for each AP.

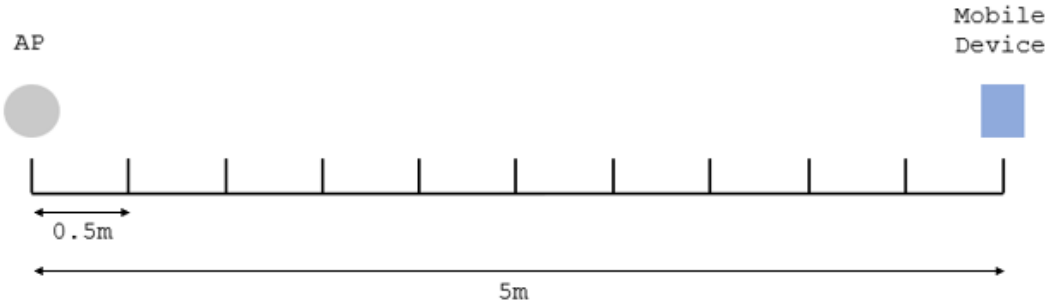


Figure 3.1: Experiment 1 Setup Plan view

Table 3.2: Experiment 1 mean FTM range error and standard deviation

| Reference Distance (m) | AP-1 mean error (m) | AP-2 mean error (m) | AP-3 mean error (m) | AP-1 standard deviation (m) | AP-2 standard deviation (m) | AP-3 standard deviation (m) |
|---|---|---|---|---|---|---|
| **0.00** | -0.52 | -2.69 | -2.60 | 0.14 | 0.30 | 0.15 |
| **0.39** | -0.51 | -2.35 | -2.45 | 0.22 | 0.32 | 0.55 |
| **0.89** | -0.41 | -2.52 | -2.35 | 0.38 | 0.44 | 0.45 |
| **1.39** | -0.56 | -2.47 | -2.29 | 0.19 | 1.24 | 0.90 |
| **1.89** | -0.55 | -2.91 | -2.14 | 0.33 | 0.36 | 1.06 |
| **2.39** | -0.32 | -2.21 | -2.44 | 0.14 | 0.17 | 0.77 |
| **2.89** | -0.34 | -2.67 | -2.46 | 0.54 | 0.38 | 0.15 |
| **3.39** | -0.09 | -2.62 | -1.99 | 0.17 | 0.55 | 1.04 |
| **3.89** | 0.06 | -2.27 | -2.07 | 0.27 | 0.56 | 0.90 |
| **4.39** | -0.22 | -2.54 | -2.86 | 0.40 | 0.17 | 0.20 |
| **4.89** | 0.11 | -2.02 | -2.21 | 0.14 | 0.70 | 0.65 |
| | -0.30 | -2.48 | -2.35 | 0.26 | 0.47 | 0.62 |

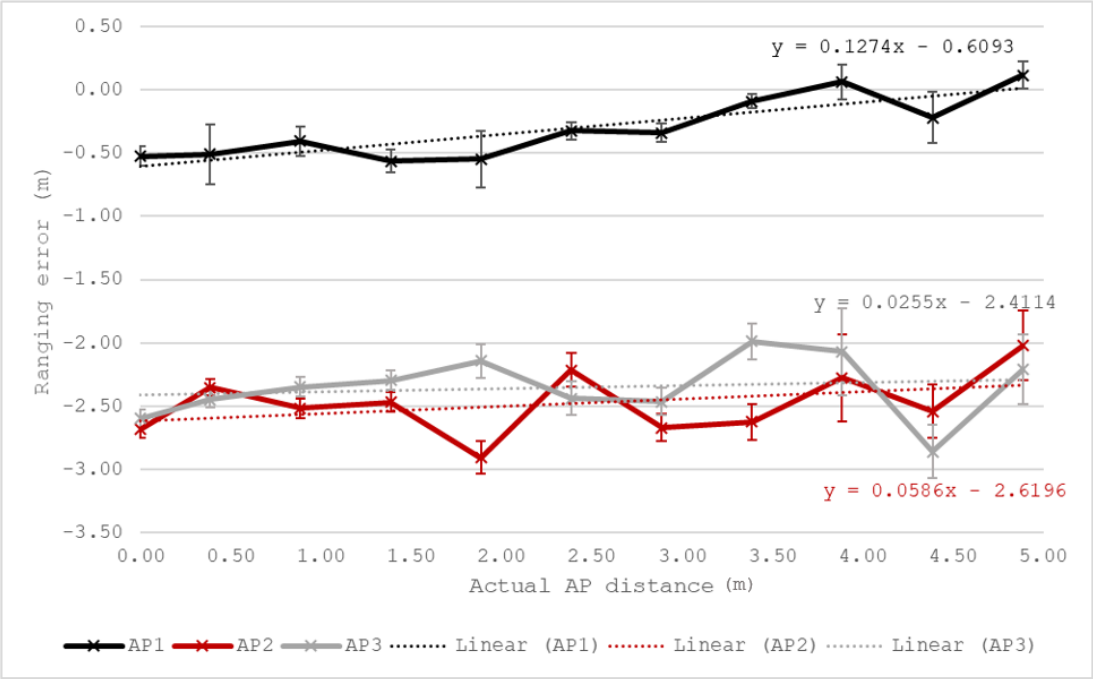


Figure 3.2: Experiment 1 mean FTM range error, including linear approximation determined via least squares

### 3.2.3 Results and Discussion

Table 3.2 shows the mean measurement range error from the FTM RTT measurement between the mobile device and the AP, as well as the standard deviation of the measurements over the five repetitions. Figure 3.2 shows the FTM range error, with the standard deviation represented as error bars.

As can be seen in Figure 3.2, the FTM range error for almost all measured distances is negative. This negative offset is more severe in AP-2 and AP-3, the Google WiFi Points, with an average offset of -2.48m and -2.35m respectively. Whereas in AP-1, the Google WiFi Router the range offset is less severe at an average of -0.3m. Due to the fact that the ranging offset is consistently negative, it is unlikely that the error is a result of noise, NLOS reception and multipath effects. Noise leads to more random errors which contrasts the consistent offset of the results. NLOS errors are always positive whilst the offsets seen here are negative, and multipath effects should vary as the devices are moved. The equations for the linear line of best fit for each AP are also shown. These are determined via linear regression, as described in Equation 3.1 and Figure 3.2.

$$y = Ax + B \tag{3.1}$$

$$A = \frac{(\Sigma y)(\Sigma x^2) - (\Sigma x)(\Sigma xy)}{n(\Sigma x^2) - (\Sigma x)^2}$$

$$B = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{n(\Sigma x^2) - (\Sigma x)^2} \tag{3.2}$$

The gradients (A) for the ranging error over distance AP-2 and AP-3 are 0.0586 and 0.0255 respectively. The RMS of these values are 0.9827 and 0.9789 respectively due to the small magnitude of these gradients and their high RMS it is appropriate to put this down to statistical noise, thus indicating that there is not an observable scaling factor involved in the ranging error for AP-2 and AP-3. The gradient associated with AP-1 however is 0.127. This is larger than the gradients of AP-2 and AP-3 and indicates that there is a factor causing the ranging error to reduce as the distance between the AP and mobile device increases. However, this is unlikely to be related to the offset itself and is more indicative of multipath effects which vary as the mobile or AP move.

For AP-2 and AP-3, the y intercepts (b) are $-2.61$m and $-2.41$m respectively. The ranging offset for AP-2 and AP-3 falls between $-2.4$m and $-2.6$m. This indicates that for Nest points, in order to produce accurate FTM range results, the measurements must be calibrated to account for negative ranging errors of this magnitude. This ranging offset is not always negative. The y intercept for AP-1 is smaller in magnitude at $-0.6$m, meaning that the range offset is less severe, hence why the average range error is $-0.3$m. Accounting for this offset and the standard deviation of the measurement, AP-1 (Google Nest router) should be calibrated between $-0.3$ and $-0.6$m. For subsequent experiments this offset was determined at the start of the experiments.

An important note is that since the errors between the Google router and Google points are so different, it is likely that the source of error originates from device manufacturing in software in the mobile device or the software/hardware in the APs as opposed to any errors that may result via signal propagation especially since the APs were tested under identical conditions. However, understanding the source of this offset is outside the scope of this experiment. It could be deduced that there is a fault in the devices, but this could only be confirmed by understanding the device bias specification set by the manufacturer, which is not publicly available. Dong et al. [48] discovered that the bias does differ from device to device, both for the APs and the mobile devices.

Furthermore, whilst for the 2 Nest points (AP-2 and AP-3) the offset is relatively consistent, it is possible that this does not apply to other Nest Points for a multitude of reasons: the points tested could have been from the same batch and the error could vary from batch to batch, there could exist a bias in the mobile device, this can be tested by using an alternative mobile device that is WiFi RTT compatible. Over time the instrument bias does change, the bias was noticed to coincide with software updates of the access points.

Ultimately, for the sake of the following experiments, this range error will be removed from the results, henceforth referred to as “calibrated results”, however, the ranging error will be monitored throughout testing to determine if these values need to be adjusted. Before every experiment the ranging offset was determined and calibrated into the ranging measurements by measuring between 0m to 2m at 0.5m increments. Want et al. [2], Ma et al. [35], Dong et al. [48] and Horn et al. [47] amongst others also noted instrument bias within the range of 1.2m to 3m.

## 3.3 Experiment 2 - Multipath effects

### 3.3.1 Introduction

These experiments aimed to understand how WiFi RTT range behaves on a more granular level. In essence, the mobile device was moved from the APs in much smaller increments of 2cm. This is important to understand as it can give potential insights into how signals should be modelled and whether certain patterns can be identified that would reveal information on the error sources or predictable features of RTT. Experiment 2a and 2b had very similar experimental setups, however Experiment 2b had all devices on a reflective surface. The purpose was to determine the impact of multipath effects by utilising the reflective surface to increase the strength of signals reflected by the surface. The experimental setup differences are described in Experimental Setup.

### 3.3.2 Experimental Setup

Experiment 2a was laid out as shown in Figure 3.3. The AP was placed on a large table, the mobile device was placed 1m away from it in a standing position and moved in 0.02m increments up to 1.5m, this was carried out for two APs (AP-2 and AP-3) three times each. At each increment, WiFi RTT readings were collected over a 30-second period at 500ms intervals. Experiment 2b was set up as shown in Figure 3.4. The experimental process was identical with the only difference being that below the mobile device and AP a reflective surface should be placed. In this experiment aluminium foil was used to produce a reflective surface. The distance between the mobile and the APs was measured using a laser distance measurer.

### 3.3.3 Results and Discussion

The calibrated results of experiments 2a and 2b are shown in Figure 3.5, which shows the calibrated results for AP-2 and Figure 3.6 which shows the calibrated results for AP-3. The graphs show the reference range against the calibrated estimated range error, with error bars representing the standard deviation of that specific reading. AP-

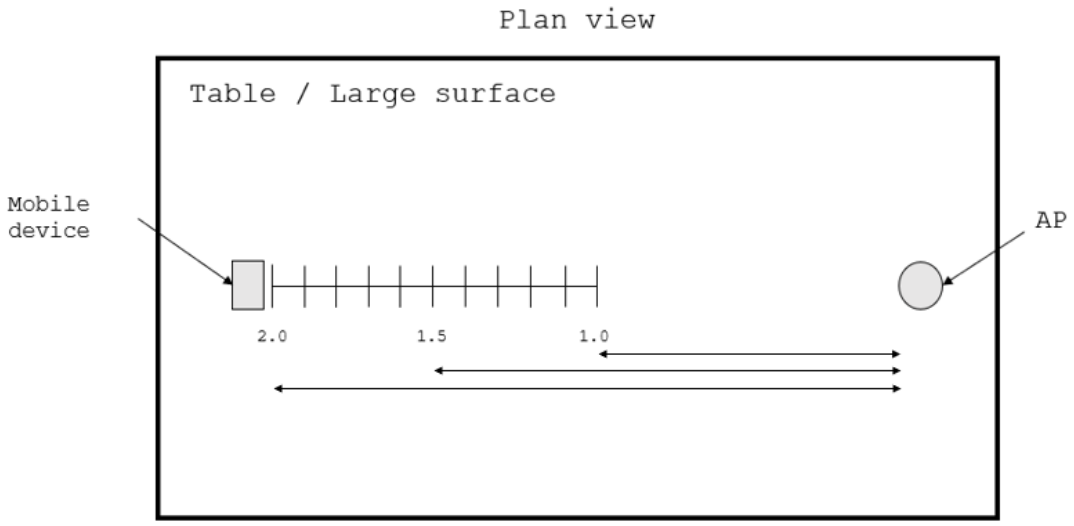


Figure 3.3: Experiment 2a setup

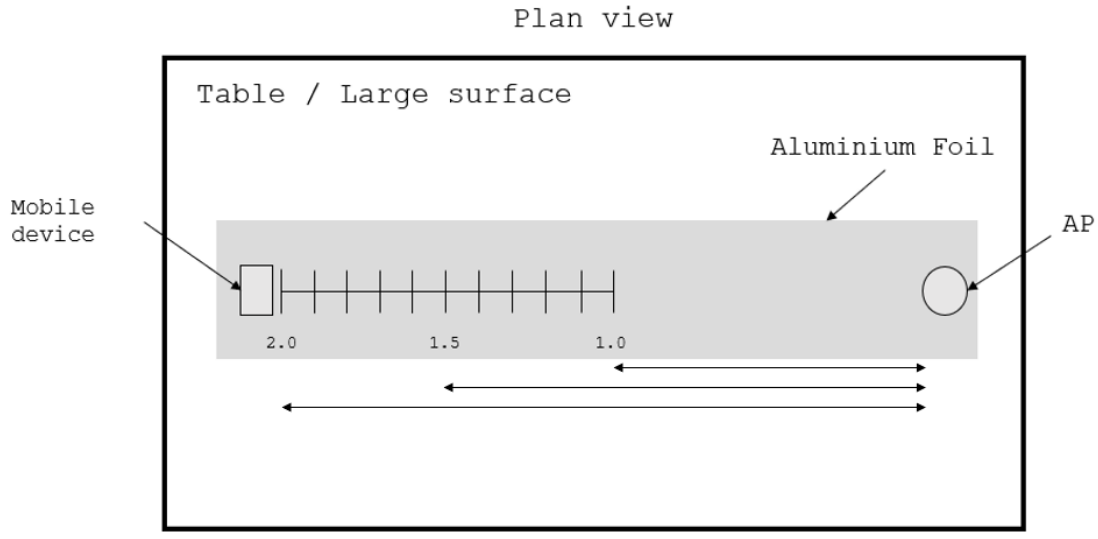


Figure 3.4: Experiment 2b setup

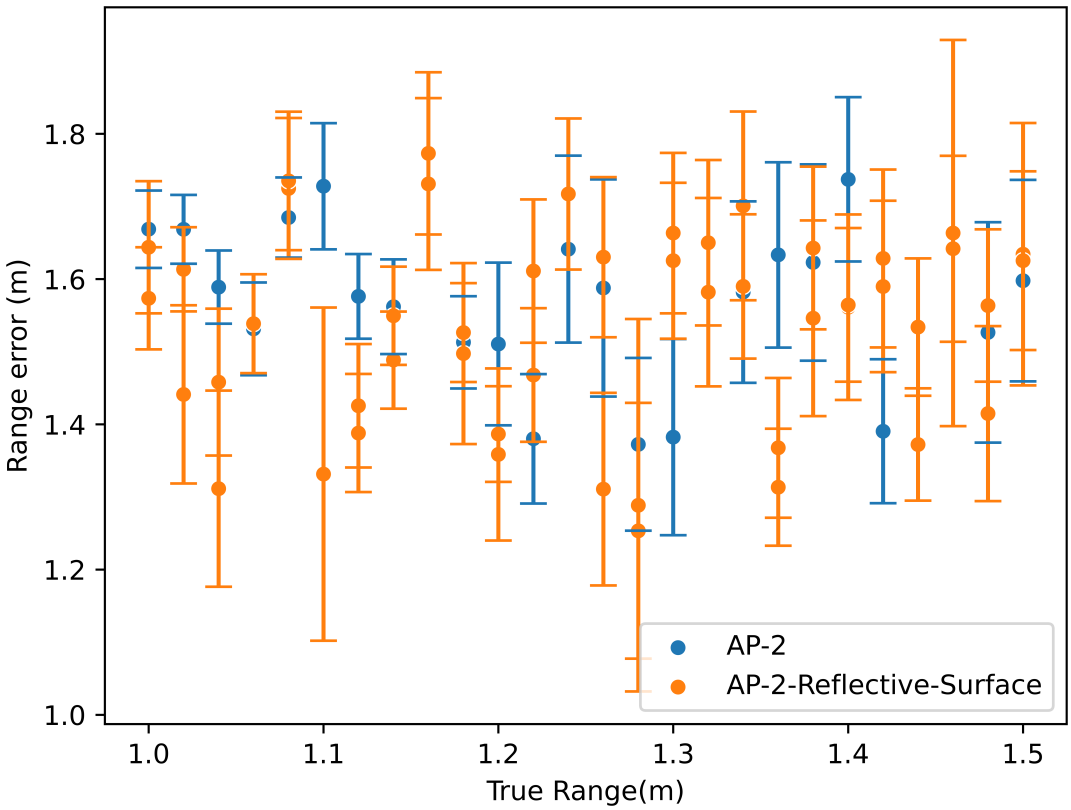


Figure 3.5: AP-2 Experiment 2a and 2b (Reflective Surface) mean FTM range error against true range, including linear approximation determined via least squares

1 was not used due to limitations in the experimental environment. The results exhibit noise, and it is unclear whether there is an underlying trend. There is a potential sinusoidal or period based trend shown in both APs, but this relationship is not consistent between the two APs. However, this is difficult to conclude or disprove given the data collected. It is immediately clear that the standard deviations and differences of the ranges from experiment 2b (reflective surface) are larger than experiment 2a. This indicates that the aluminium foil is introducing further interference in the ranges. This could be due to enhanced multipath effects as the signals are reflected off the surface with greater strength, thus equally strong signals are reaching the devices and therefore the devices interpret signals as equally possible range results. This would explain why a larger standard deviation and spread in the results may arise. Since the true reading is being obfuscated by the non-true reflected readings, however, for this to be the case then the ranging errors should be larger as a result of the longer path. To date, no experiment such as this has been conducted on WiFi RTT.

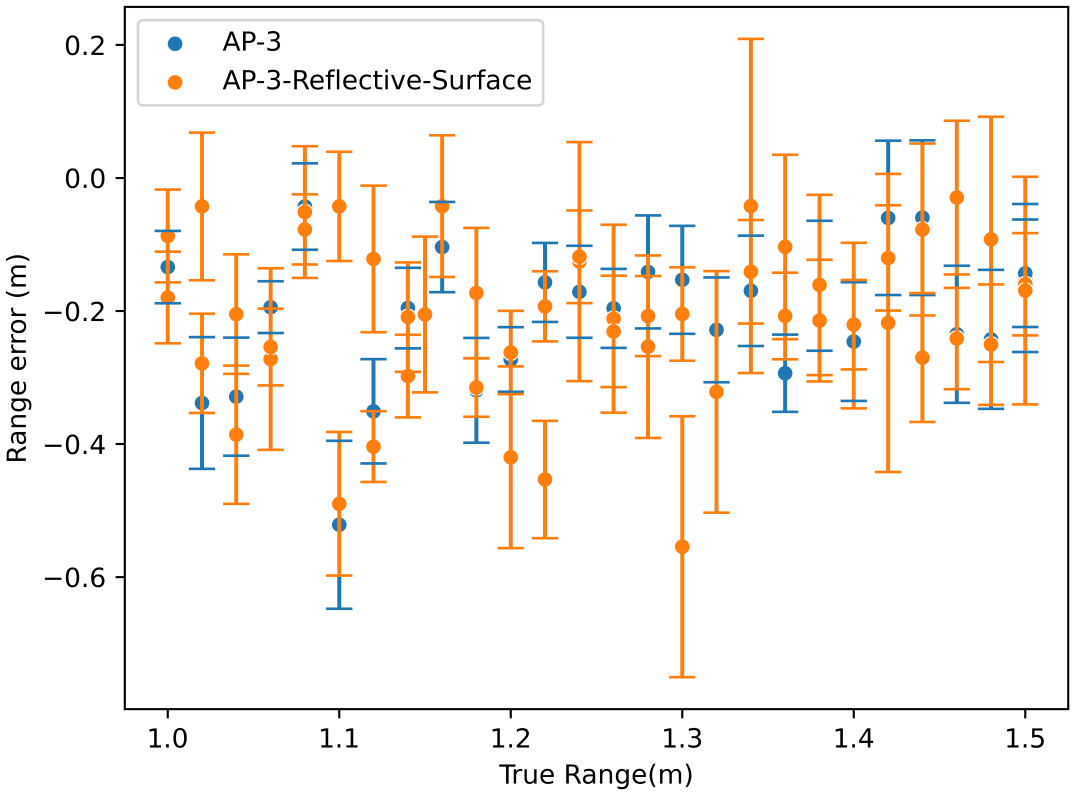


Figure 3.6: AP-3 Experiment 2a and 2b (Reflective Surface) mean FTM range error against true range, including linear approximation determined via least squares

## 3.4 Experiment 3 - Attenuation and NLOS

### 3.4.1 Introduction

In a non-experimental environment, it is highly unlikely that a mobile device will have a clear or unobstructed line of sight to an AP at all times. The previous experiments have all investigated the behaviour of WiFi RTT given a direct line of sight between the AP and the mobile device. The following experiments investigated WiFi RTT's behaviour in scenarios that force NLOS signal transmission between the mobile device and AP.

### 3.4.2 Experimental Setup

For this experiment there are 2 scenarios, each split into 2 sub scenarios. The aim of each scenario is to place the mobile device and AP in a different situation to investigate various impacts of obstructed transmission. Each subscenario of a scenario represents a LOS condition (where the mobile and AP have a clear line of sight) and a NLOS condition (where there is an obstruction between the mobile and AP).

The experimental layouts of each scenario are shown in Figure 3.7 and Figure 3.8. In Scenario 1, a door is placed between the mobile device and AP in the NLOS condition, this experiment aims to test the performance of WiFi RTT when completely obstructed. In Scenario 2, the mobile device and AP are obstructed with a varying distance, but there is an opening to one side allowing for a reflected signal between the mobile device and AP. As seen in each figure (Figures 3.7 and 3.8) they are split into sub scenarios 'a' and 'b'. Scenario a represents the LOS condition and b represents the NLOS condition. Once the AP and Mobile device were set up as specified, measurements were logged at the specified ranges. Readings were collected over 30 seconds at 500ms intervals. RSSI readings were also collected, but these readings are already included in the Google WiFi RTT Scan App's data collection [63]. AP biases identified in experiment 1 were calibrated into the results.

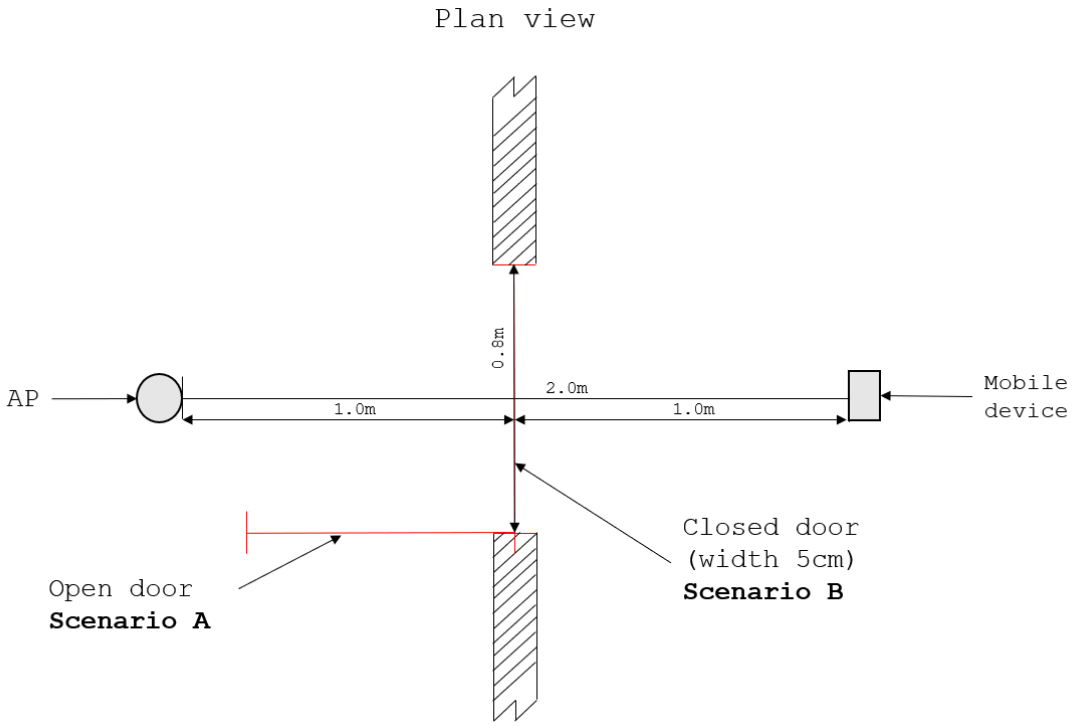


Figure 3.7: Scenario 1 with sub-scenarios A and B. The difference between the sub-scenarios is an open and closed door at 1m between the mobile device and AP

### 3.4.3 Results and Discussion

The calibrated results for scenario 1 are shown in Table 3.3. Overall the ranging error is less than 10cm with the LOS scenario (sc1-a) showing marginally higher accuracy and a marginally stronger RSSI than the NLOS scenario results. The error caused by NLOS here is on average 0.045m. It is important to note that there are no unobstructed lines of sight between the access point and mobile device. In the

Table 3.3: Scenario 1a and 1b calibrated results, estimated range standard deviation and RSSI

| subscenario-trial | Reference Range (m) | Average Est. Range (m) | SD (m) | Average RSSI (dBmW) |
|---|---|---|---|---|
| sc1-a-trial1 | 2 | 2.0445 | 0.0635 | -48.7 |
| sc1-a-trial2 | 2 | 2.0333 | 0.0679 | -48.7 |
| sc1-a-trial3 | 2 | 2.0242 | 0.0505 | -48.4 |
| sc1-b-trial1 | 2 | 2.0738 | 0.0810 | -49.9 |
| sc1-b-trial2 | 2 | 2.0872 | 0.0691 | -48.9 |
| sc1-b-trial3 | 2 | 2.0758 | 0.0764 | -49.9 |

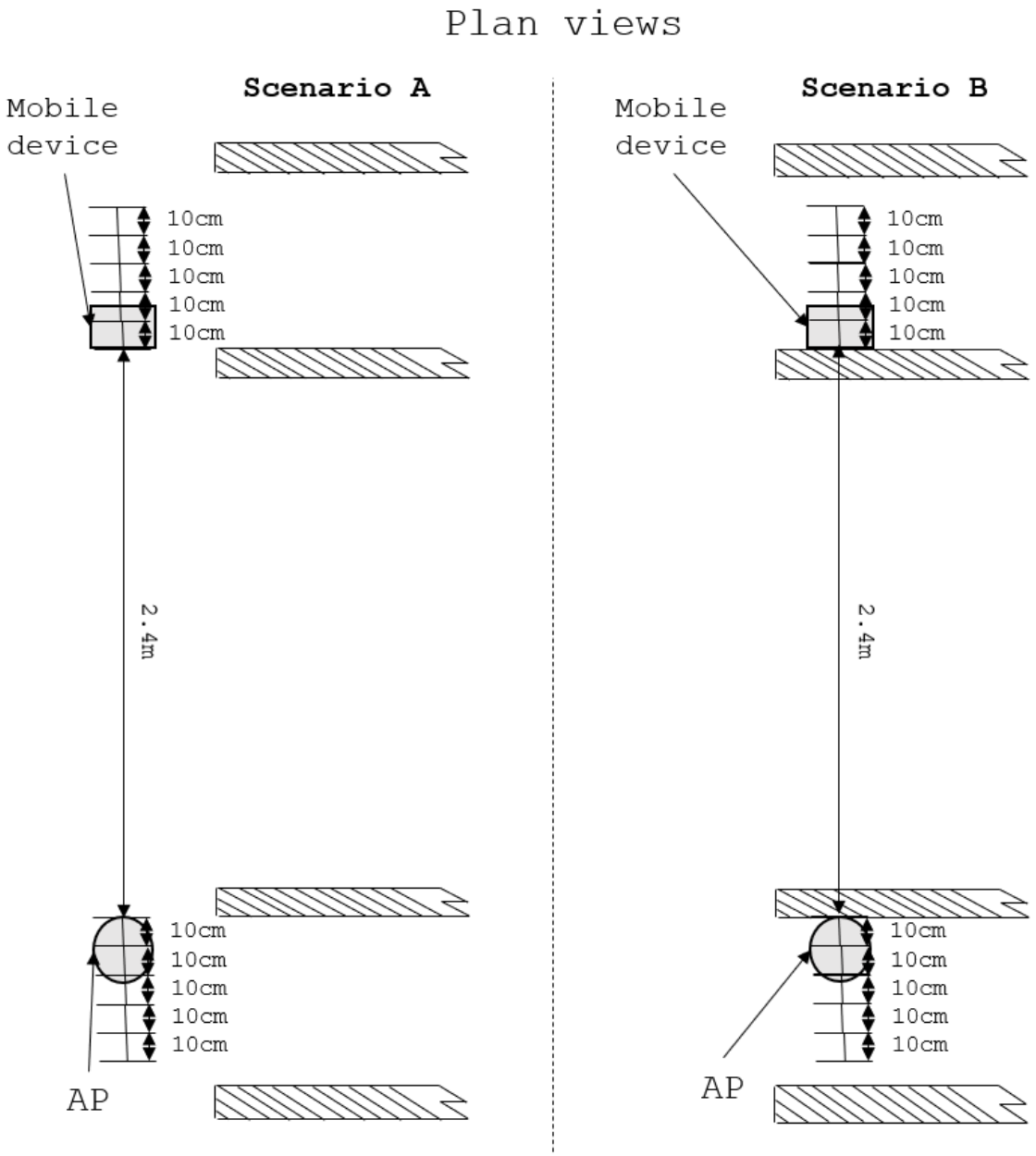


Figure 3.8: Scenario 2 with sub-scenarios a and b. The AP and mobile device can be moved in 10cm increments away from each other. The walls between the mobile device and AP each have a width of approximately 10cm.

NLOS scenario the door is closed and the room is closed off, thus the signals must pass through the door or walls to reach the receivers. The source of signal obstruction is most likely the permittivity of the wall. This indicates that in this experimental environment, doors have a minuscule impact on the RTT ranging measurement. It is unlikely to be NLOS reception caused by the signals reflecting off walls and other surfaces or multipath effects. It is possible that the strongest signal will have had to pass through the door in scenario B to reach each receiver. This explains the slightly higher range estimate as well as the decrease in the signal strength. The door that obstructs the signal path in scenario 1 is made of wood. Research conducted by Suherman et al [64] and Mohammed et al [65], investigated the impact of building materials on WiFi received signal strength. Overall the research found that wood caused a minor reduction to the RSSI when compared to free space. Suherman et al. found that wood reduced the signal strength by an average of 0.73dBm whilst concrete reduced the signal strength by 1.53dBm on average. This is inline with the experimental results achieved in these experiments, as the RSSI strength decrease caused by the wooden door in Scenario 1 was substantially lower than the 2 brick walls of Scenario 2.

The calibrated results for scenario 2 are shown in Figure 3.9. By comparing scenario 2a to 2b, the results shows that the LOS conditions produce a more accurate ranging solution with a smaller standard deviation than the NLOS conditions. Ranging under NLOS conditions produces a ranging solution with an average ranging error of 0.81m and a standard deviation of 0.17m (standard deviation based on three measurements at the same distance), these are less accurate when compared to the LOS conditions results which produce a ranging error of $-0.21$m and a standard deviation of 0.08m. In Figure 3.9, RSSI has also been plotted for each data point. The RSSI results show that under LOS conditions with a more reliable ranging solution, the RSSI is stronger whereas under NLOS conditions the RSSI is significantly weaker. The difference between the two scenarios is an average of 18.8 dBm, with a standard deviation of 1.37dBm in NLOS conditions and 0.56dBm in LOS conditions (standard deviation based on three measurements at the

same distance). This suggests that it is possible to identify reflected NLOS signals by a large decrease in the RSSI when compared to the RTT reading received from that AP or other signals in an experimental environment, although in the latter case this could be as a result of a greater distance between the AP and mobile device. This observation of reduced RSSI resulting in greater ranging errors between 0.4 - 2m indicates the potential of RSSI as an indicator of ranging error.

One of the most important results of this experiment is the correlation between low RSSI and NLOS conditions. This trend is likely due to two factors: attenuation at the point of reflection or RTT signals reflecting off surfaces in the environment and thus the signal strength being lower than a direct signal would have been.

Dong et al. [48] conducted experiments to understand WiFi RTT error characteristics, in these experiments, in addition to comparing NLOS and LOS conditions they also explored changes in accuracy when the user was moving and wood, glass and metal as the obstruction. The paper found that NLOS conditions could cause as much as a 2.8m ranging error when the obstruction was made of metal and both the mobile device and AP were stationary, demonstrating the impact of NLOS effects. When the range and position were varied there was up to a 2.9m ranging error in NLOS conditions. Additionally, there was a 3.3m standard deviation for the WiFi RTT ranges for this experiment, indicating that the combination of motion and metal obstructions can significantly impact the accuracy of WiFi RTT. This level of error could place a user in an incorrect room. The paper identified that obstructions weakened the average RSSI of the RTT signal, consistent with the findings in this thesis. Furthermore, a metal obstruction provided more signal interference than wood or glass. These results are as expected, as a more conductive obstacle will typically yield more signal interference.

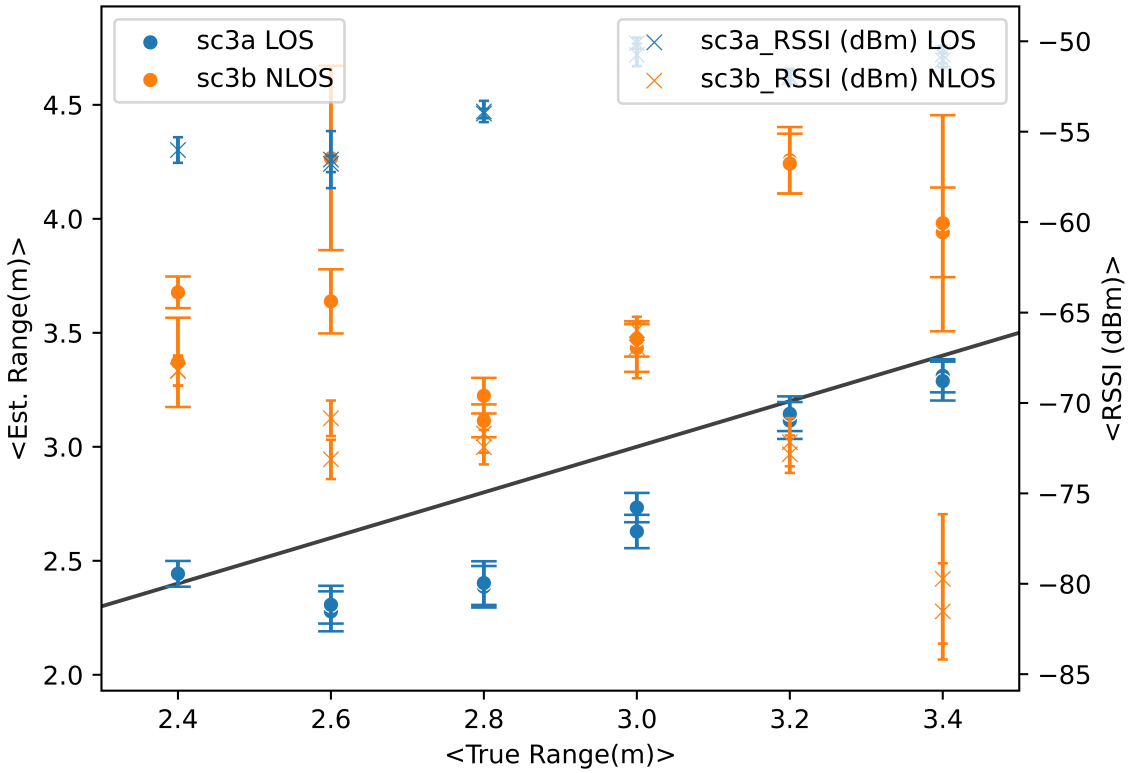


Figure 3.9: Scenario 2 with sub-scenarios a (LOS) and b (NLOS) calibrated results, the black line represents the true range (i.e. y=x)

## 3.5 Experiment 4 - Instrument Orientation

### 3.5.1 Introduction

This experiment aims to determine whether the orientation of a mobile device relative to a AP impacts WiFi RTT ranging error. During data collection, significant ranging errors were noticed when the mobile device and access point were at certain orientations. By orientation, what is meant is shown in Figure 3.10.

This requires further investigation to determine the necessity of calibrating the ranging measurements based on relative orientation of the devices to improve the accuracy of the system. This experiment will aim to provide clarification on this observation. This experiment will not explore testing different mobile devices.

### 3.5.2 Experimental Setup

The experimental layout is shown in Figure 3.10. A mobile device and an AP (AP-1) were placed with no obstacles between and no walls within 3m. Both devices were at the same height. These conditions exist to ensure that the device and AP have an unobstructed line of sight and also so that reflections off of nearby surfaces can be mitigated. The ranges were measured at a 1m and a 2m distance. At each position, the FTM RTT readings were logged from the mobile device over a 30-second period. The reference range between the AP and the mobile device was measured using a measuring tape and the angle between the mobile device and the front of the AP was measured with a protractor. For all readings the channel frequency was 5180MHz, this was the default presented by the WiFi RTT logging mobile app used. This experiment was then repeated 5 times for each AP.

### 3.5.3 Results and Discussion

As can be seen in Figure 3.11 the WiFi RTT ranging error does vary depending on the relative orientation of the mobile device to the access point. This variation also changed depending on the distance between the mobile device and the access point. For the 1m trial, the measured range varied between 0.61m to 1.59m whereas for the 2m trial, the range varied between 0.5m and 5.05m. At 0 degrees both range measurements were within 2.5cm of the true range. However, for the 1m trial at 180 and 315 degrees, the range error was greatest with range errors of 0.59m and 0.53m respectively. Then for the 2m trial at 90, 180 and 315 degrees the range error was greatest with range errors of 1.45m, 2.39m and 3.05m respectively. The greatest outliers for both ranges occurred at 180 and 315 degrees. At 45 and 135 degrees, there were large negative ranging errors of -0.39m and -0.36m for the 1m trial respectively and $-1.5$m and $-0.86$m for the 2m trial respectively.

These errors could be attributed to multipath effects. There are a few pieces of evidence to support this. Firstly, there are both negative and positive ranging errors these are likely a result of constructive and destructive interference of the signals.

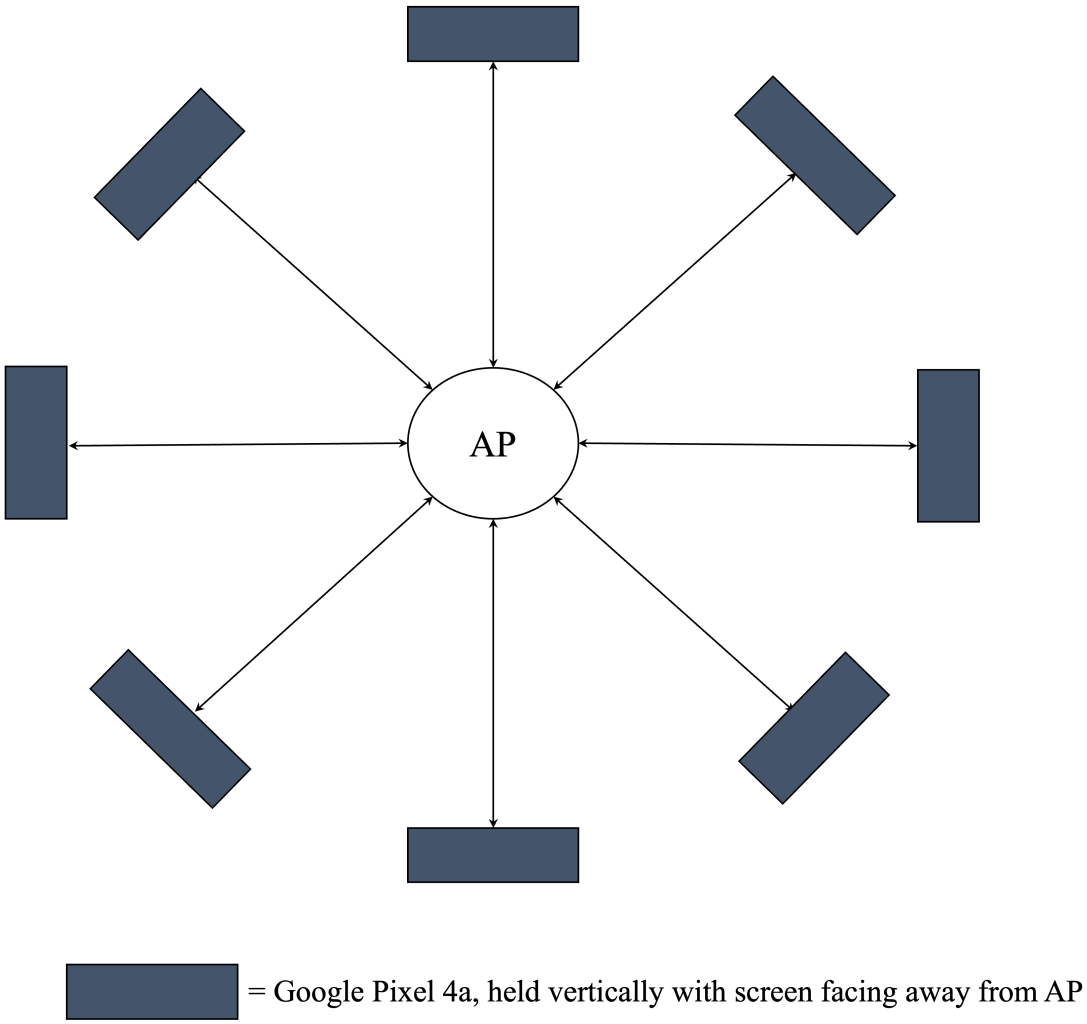


Figure 3.10: Experiment 4 Orientation Test Plan view

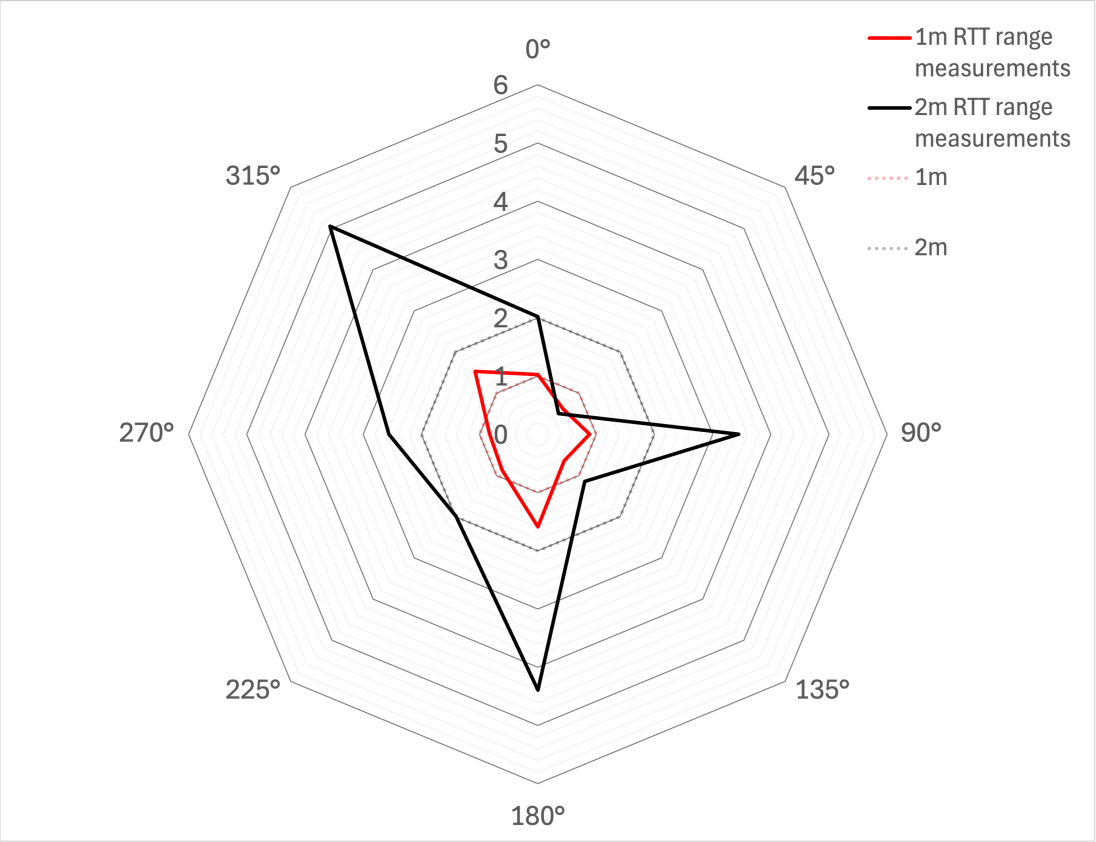


Figure 3.11: Experiment 4 mean WiFi RTT at different orientations

For the 2m trials, a correct signal had an RSSI of approximately −66dBmW (0 and 225 degrees) whereas the outlier signals at 45, 180 and 315 degrees had an RSSI of −72dBmW, −71dBmW and −75dBmW respectively. This indicates some sort of signal interference as there are no physical obstacles to attenuate or cause NLOS signals. There could be attenuation or reflection effects present in the hardware that would cause ranging error, but these would not cause negative ranging errors.

# CHAPTER 4

# Positioning algorithms

This chapter will describe a number of positioning algorithms that use WiFi RTT and the experiments conducted to determine their positioning reliability and accuracy. The algorithms explored include least squares, weighted least squares, particle filters, genetic filters and grid filters. In addition, residual-based outlier detection is tested and an RSSI-based outlier detection model is introduced. Section 4.1 describes the algorithms, Section 4.2 describes the experimental methodology and Section 4.3 discusses the results of the experiments.

## 4.1 Algorithms

### 4.1.1 Least squares positioning

#### Unweighted Least Squares Multi-lateration

Least squares positioning is a method that determines the position of an object by minimising the sum of the squared differences between the measured distances from reference points and the predicted distance between a candidate position and the reference points. Least Squares has been applied to WiFi FTM RTT, as discussed in Ma et al. [35]. There are two versions of the algorithm used for this experiment,

one that ignores the z (vertical) value (i.e. a 2D positioning solution) and one that includes the z value in the calculation of the position. It is worth noting that, although the signal geometry is insufficient for accurate vertical positioning, the z position and axis is included in the following equations as it could help improve the accuracy of a 2D solution, and deriving the 2D least squares solution from this equation is trivial. For this experiment, it is assumed that the mobile device knows the location of all APs. Estimating the AP position from the ranging results is out of the scope of this experiment.

The distance between each device and an AP can be calculated using Pythagoras' theorem. Equation 4.1 shows the start of the least squares process derived from the distance between each AP and the device. In Equation 1 $x$, $y$, $z$ represents the coordinates of the mobile device. $X_n$, $Y_n$, $Z_n$ represents the coordinates of the nth AP and $d_n$ represents the distance between the nth AP and the mobile device.

$$\begin{bmatrix} (X_1 - x)^2 & (Y_1 - y)^2 & (Z_1 - z)^2 \\ (X_2 - x)^2 & (Y_2 - y)^2 & (Z_2 - z)^2 \\ \vdots & \vdots & \vdots \\ (X_n - x)^2 & (Y_n - y)^2 & (Z_n - z)^2 \end{bmatrix} = \begin{bmatrix} d_1^2 \\ d_2^2 \\ \vdots \\ d_n^2 \end{bmatrix} \tag{4.1}$$

To get the intersection point of these ranges, and thus the position of the object, subtract the last equation in the matrix from the rest of the equations, as shown in Equation 4.2. This is a technique also used by Ma et al. [35]. If there are less than 3 APs then this equation cannot work, and therefore it may be beneficial to switch to a 2D version of Equation 4.2, shown in Equation 4.3. This involves omitting the z coordinate, assuming that all devices lie on the same plane. However, it is worth noting that whilst 3 APs is the lower limit, it is best to have a few more APs in order to allow measurements to be removed if measurements from those APs are invalidated due to outlier detection. For this research, 6 APs were used.

$$\begin{bmatrix} (X_n - X_1)\,x & (Y_n - Y_1)\,y & (Z_n - Z_1)\,z \\ (X_n - X_2)\,x & (Y_n - Y_2)\,y & (Z_n - Z_2)\,z \\ \vdots & \vdots & \vdots \\ (X_n - X_{n-1})\,x & (Y_n - Y_{n-1})\,y & (Z_n - Z_{n-1})\,z \end{bmatrix} = \begin{bmatrix} \left(d_1^2 - d_n^2 + X_n^2 + Y_n^2 + Z_n^2 - X_1^2 - Y_1^2 - Z_1^2\right)/2 \\ \left(d_2^2 - d_n^2 + X_n^2 + Y_n^2 + Z_n^2 - X_2^2 - Y_2^2 - Z_2^2\right)/2 \\ \vdots \\ \left(d_{n-1}^2 - d_n^2 + X_n^2 + Y_n^2 + Z_n^2 - X_{n-1}^2 - Y_{n-1}^2 - Z_{n-1}^2\right)/2 \end{bmatrix} \tag{4.2}$$

$$\begin{bmatrix} (X_n - X_1)\,x & (Y_n - Y_1)\,y \\ (X_n - X_2)\,x & (Y_n - Y_2)\,y \\ \vdots & \vdots \\ (X_n - X_{n-1})\,x & (Y_n - Y_{n-1})\,y \end{bmatrix} = \begin{bmatrix} \left(d_1^2 - d_n^2 + X_n^2 + Y_n^2 - X_1^2 - Y_1^2\right)/2 \\ \left(d_2^2 - d_n^2 + X_n^2 + Y_n^2 - X_2^2 - Y_2^2\right)/2 \\ \vdots \\ \left(d_{n-1}^2 - d_n^2 + X_n^2 + Y_n^2 - X_{n-1}^2 - Y_{n-1}^2\right)/2 \end{bmatrix} \tag{4.3}$$

The least squares equation is shown in Equation 4.4.

$$\mathbf{P} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{4.4}$$

where (based on Equation 4.2),

$$\mathbf{P} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{4.5}$$

$$\mathbf{A} = \begin{bmatrix} X_n - X_1 & Y_n - Y_1 & Z_n - Z_1 \\ X_n - X_2 & Y_n - Y_2 & Z_n - Z_2 \\ \vdots & \vdots & \vdots \\ X_n - X_{n-1} & Y_n - Y_{n-1} & Z_n - Z_{n-1} \end{bmatrix} \tag{4.6}$$

$$\mathbf{P} = \begin{bmatrix} \left(d_1^2 - d_n^2 + X_n^2 + Y_n^2 + Z_n^2 - X_1^2 - Y_1^2 - Z_1^2\right)/2 \\ \left(d_2^2 - d_n^2 + X_n^2 + Y_n^2 + Z_n^2 - X_2^2 - Y_2^2 - Z_2^2\right)/2 \\ \vdots \\ \left(d_{n-1}^2 - d_n^2 + X_n^2 + Y_n^2 + Z_n^2 - X_{n-1}^2 - Y_{n-1}^2 - Z_{n-1}^2\right)/2 \end{bmatrix} \tag{4.7}$$

The position of the mobile device **P** will be calculated at each epoch.

### 4.1.2 Weighted Least-Squares Multi-Lateration

The FTM RTT readings produced by the device provide a standard deviation of the 8 burst ranging packets, this represents the standard deviation of that specific ranging measurement (this also highlights the fact that the outputted ranging measurement is a mean of the ranging measurements of the 8 burst ranging packet). By using weighted least squares it is possible to account for the varying precision of each FTM RTT measurement. The algorithm mostly stays the same except for the addition of the measurement error covariance matrix in the least squares calculations. This is shown in Equations 4.8 and 4.9.

$$\mathbf{P} = (\mathbf{A}^T\mathbf{C}_k^{-1}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{C}_k^{-1}\mathbf{b} \tag{4.8}$$

where,

$$\mathbf{C}_k = \begin{bmatrix} \sigma_{k1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{k2}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_{k(n-1)}^2 \end{bmatrix} \tag{4.9}$$

k represents the epoch of the matrix, $\sigma$ represents the standard deviation of the RTT signals of the AP at that epoch. The measurement error covariance matrix assumes that the measurement errors are independent for different APs for each epoch

and therefore the covariances are 0, and only the variances of each measurement are accounted for, as shown with the main diagonal.

### 4.1.3 Particle Filters

Bayesian filtering is a probabilistic approach to estimate the state of a system that evolves over time. By combining prior knowledge with new information from measurements that provide information on how the system has changed. As new data becomes available the probability distribution of the system's state is updated using Bayes' theorem [66]. In the context of positioning and navigation the state would most likely be the coordinates, velocities etc. of an object to be tracked. Bayesian filters include Kalman filters and particle filters amongst others. Particle filtering, is based on the sequential Monte Carlo method which involves simulating a large number of particles and assigning a position within the state space and weight to each particle. The position and weight distribution is then used to form a probability distribution of the final position, providing a weighted average of the particles to get the estimated position. As an example, if all particles are clumped around a specific place, this means the particle filter has determined there is a high probability that the object being tracked is in that location.

Particle Filters are a promising method for WiFi RTT positioning as they are better suited than standard Kalman filters for highly nonlinear problems [67] [68]. WiFi RTT is a non-linear problem as the RTT measurements are not linear functions of the mobile device position and are non-Gaussian. The latter is mainly a result of NLOS reception errors and multipath effects.

The process for the particle filter follows the diagram shown in Figure 4.1. Broadly, a particle filter begins with an initialisation step where the particle states are initialised, this can be done in several ways such as a uniform distribution in the state space or a Gaussian distribution around an initial state. Following this, there is a prediction step which involves moving the particles in the state space based on some inputs. The next step is the update step which involves weighting the particles

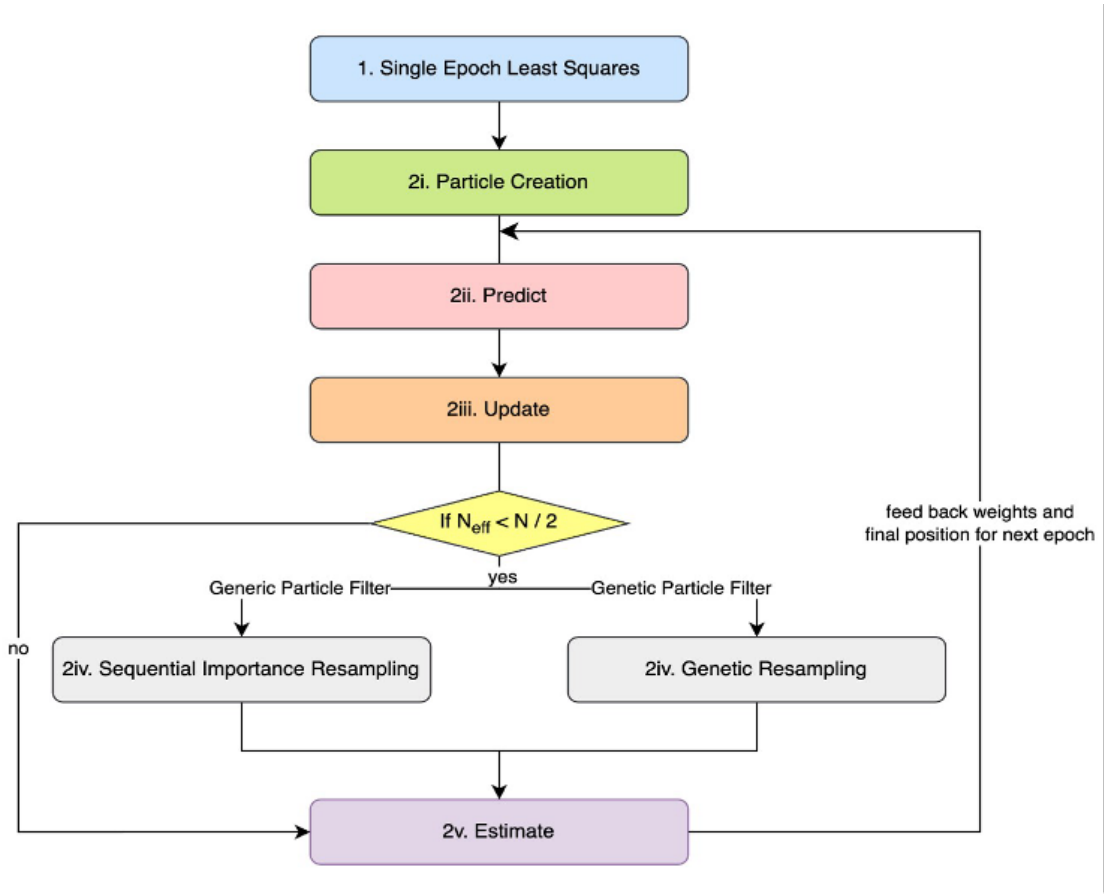


Figure 4.1: Generic particle filter and genetic filter process

based on some measurements, the particles which most closely align with the measurement are weighted higher. This process can be repeated indefinitely so long as there is new information. There are other steps such as resampling and estimation which will be explained later, but the above steps give a broad overview of how a particle filter works.

An initial position estimate is determined based on a single-epoch least-squares algorithm.

Initialisation – $N_p$ particles are created using the initial position estimate as the mean. The particles are then randomly distributed around the initial position estimate to account for uncertainty. The standard deviation of the state distribution matches the uncertainty of the initial position estimate. The initial distribution of the particles follows a Gaussian distribution.

Prediction – For positioning in motion, a pedestrian dead reckoning model described in Section 4.1.7 is used. The positions of the particles are adjusted using:

$$Q^j_{k+1} = Q^j_k \cos(\varphi^j_{k+1}\ ) r^j_{k+1} \tag{4.10}$$

$$R^j_{k+1} = R^j_k\ sin(\varphi^j_{k+1}\ ) r^j_{k+1} \tag{4.11}$$

Where Q are the x coordinates of the particles and R are the y coordinates of the particles, according to the coordinate frame of the environment, which can vary for indoor environments, k represents the epoch of the particles and j represents the particle index. Where the displacement, $r_k$, and orientation $\varphi_k$, are retrieved as described below and defined in Equation 4.12 and 4.13 respectively. As steps are discrete events, the movement of the device appears to be discrete from the perspective of the filter as the prediction step will update the device's position at each step. This is an issue as WiFi measurements can be recorded mid-step, which are used to determine the weights of particles. Thus, only updating the particle states once per step will not give a precise representation of the device' state. To resolve this, step-lagged smoothing is used. This method allows the filter to account for the movement of the mobile device during a step by using the time taken between a step along with the step length to determine a velocity of the step. When a new epoch of WiFi RTT ranges are received, the distance travelled is determined using the step velocity and the time taken between the WiFi RTT epochs, allowing the particles access to a more precise representation of the mobile device's expected position during a step. The step length and orientations are retrieved as described in Section 4.1.7 and then system noise is applied to these values, as described as follows for the displacement and heading respectively.

$$r^j_{k+1} = \frac{N(q_{k+1}, \sigma_q)}{n_e} \tag{4.12}$$

$$\psi_{k+1}^{j} = N(\psi_{k+1}, \sigma_{\psi}) \tag{4.13}$$

Where, $r_k$, represents the displacement of the device for an epoch, q represents the step length, $\sigma_q$, represents the assumed standard deviation of the step length error, $n_e$ represents the number of filter epochs that have occurred between a full step, N indicates that each element is sampled from a Gaussian distribution. In Equation 4.13, $\psi_k$ represents the heading of the mobile device, $\psi_{k+1}^{j}$, represents the heading of the device with Gaussian noise incorporated and $\sigma_{\psi}$ represents the assumed standard deviation of the heading error.

The full motion model algorithm is shown in Figure 4.2 and occurs during the prediction step of all filters.

Update – The weights of each particle are updating by calculating how closely the measured ranges to each landmark match the particles distance to the landmarks. The model used for weighting is described below:

$$w_{k+1}^{j} = w_{k}^{j} \prod_{i=1}^{n} \frac{1}{\sigma_k^i \sqrt{2}\pi} e^{-\frac{1}{2}\left(\frac{z_k^i - d_k^{j,i}}{\sigma_k^i}\right)^2} \tag{4.14}$$

The update step for computing weightings is the step for determining how strongly a particle state matches the state suggested from measurements. The Euclidean distance, $d_k^{j,i}$, between each particle, $j$, and the landmarks is computed using Pythagoras' theorem, where $i$ is the AP landmark being measured from, $n$ is the number of APs and $k$ represents the epoch. This distance is then treated as the mean in a Normal distribution alongside a standard deviation. This standard deviation is modified using RSSI-based outlier detection which is described in Section 4.1.6. Once the normal distribution is determined, the PDF of the distribution of $z_k^i$, the measurement obtained of the distance between the AP, $i$, and the mobile device is calculated using Pythagoras' theorem at epoch $k$. This gives the particle weight for that landmark. The weights for all landmarks for each epoch are

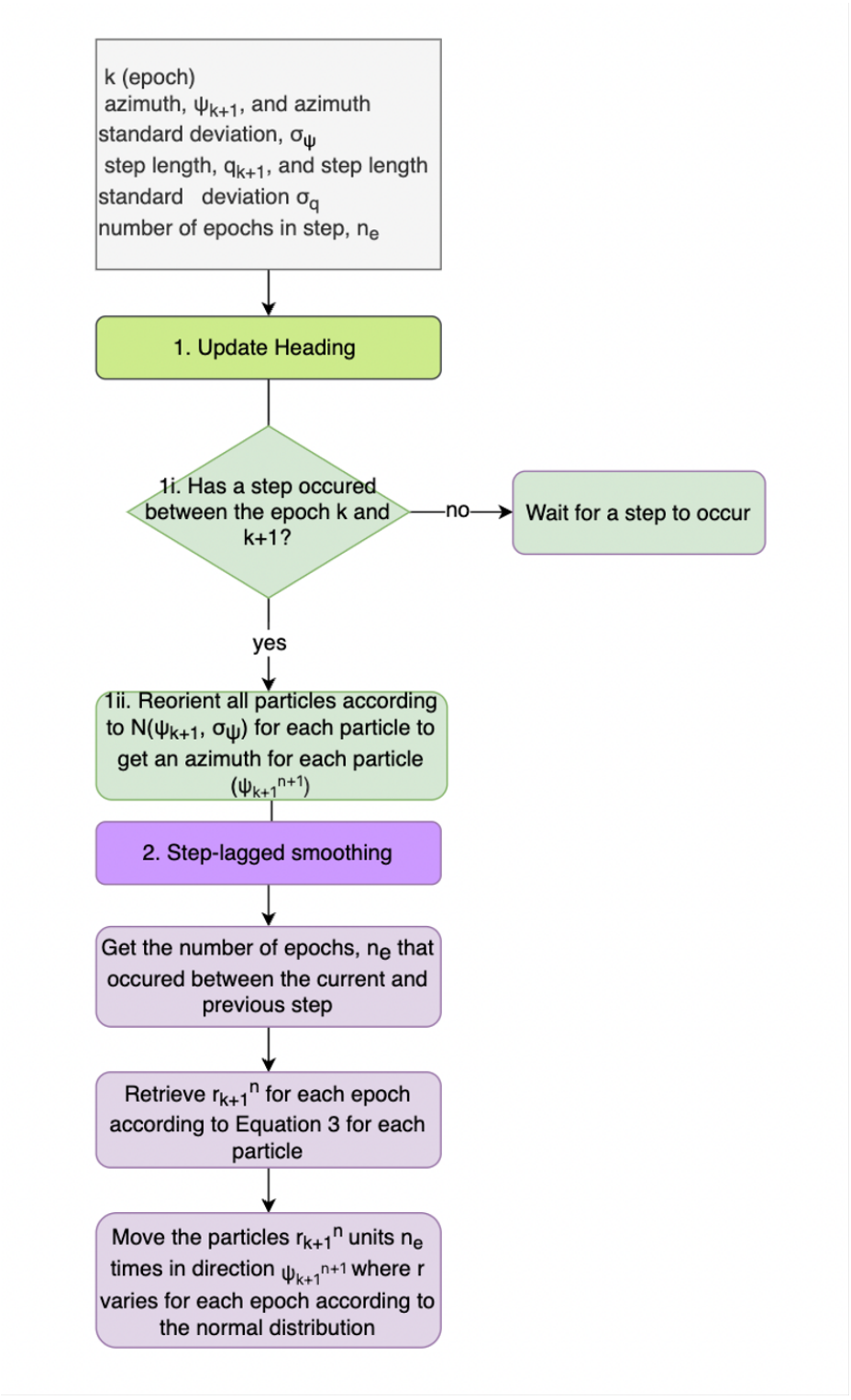


Figure 4.2: Step-lagged PDR motion model

then multiplied together and the previous weight of that particle, $w_k^j$, to give a final weight for that particle, $w_{k+1}^j$, this process is repeated for all particles. Following this, the weights are normalised using L1 normalisation.

Resampling – If Equation 4.15 evaluates to be true then Sequential Importance Resampling (SIR) is carried out, in this process the higher weighted particles are retrieved and replace lower weighted particles whilst resetting all weights. The effective sample size, $N_{eff}$, is determined as shown in Equation 4.16 so long as Equation 4.15 evaluated as true.

$$N_{eff} < \frac{N_p}{2} \tag{4.15}$$

$$N_{eff} = \frac{1}{\sum_{n=1}^{N_p}(w_k^j)^2} \tag{4.16}$$

Where $N_p$ is the number of particles and $w_k^j$ is the normalised weight of particle $j$ at epoch $k$. In most particle filters, SIR is used [67], where resampling is triggered because the number of effective particles is too low. SIR essentially replicates higher weighted particles and deletes lower weighted particles. This is done to ensure more efficient computation or use of particles. Over time the number of particles with extremely small weights will increase, meaning a large amount of unnecessary computation as only a small number of particles with higher weighting contribute to the positioning solution. This is known as particle degeneracy. As the algorithm progresses there will be particles that have very low weight and contribute nothing to the solution and are thus wasted computation. The resampling step essentially removes these low weighted particles in favour of the higher weighted particles such that more particles are being used to identify the position of the mobile device.

Estimate – compute the mean and standard deviation of the states using the particle weights according to the following:

$$\bar{q}_w^k = \frac{\sum_{j=1}^{N_p} w_k^j Q_k^j}{\sum_{j=1}^{N_p} w_k^j} \tag{4.17}$$

$$\bar{r}_w^k = \frac{\sum_{j=1}^{N_p} w_k^j R_k^j}{\sum_{j=1}^{N_p} w_k^j} \tag{4.18}$$

Where $\bar{q}_w^k$ is the weighted average of the x coordinate of the position estimate at epoch k and $\bar{r}_w^k$ is the weighted average of the y coordinate of the position estimate at epoch k. The Prediction, Update, Resampling and Estimation steps are repeated for each epoch.

### 4.1.4 Genetic Filter

#### Genetic Algorithm

A genetic algorithm [69] [70] is an optimisation method that simulates the biological evolution process, typically conforming to natural selection. Similar to a particle filter, there are multiple discrete candidate solutions that represent a possible state of the system. The process of a genetic algorithm requires representing every candidate solution as a vector of real numbers which represent characteristics of the possible state of the system. This is known as real-value encoding. There is also binary encoding, but that will not be explored in this thesis.

There are generally three operations within a genetic algorithm: selection, crossover and mutation. The selection operation is responsible for classifying the candidate solutions into good solutions and bad solutions or alternatively for ranking the solutions. This is typically done by assigning a ”fitness value” which is used to compare the candidate solutions against each other. The fitness value represents how strong the candidate solution is relative to an optimal solution. The candidate solutions are then passed onto the mating pool with their ranking according to the fitness value for further processing. This process is akin to ”survival of the fittest”

where candidates are ranked based on their strength. These candidate solutions are then passed onto the crossover operation.

The crossover operation is akin to mating. In this process two candidate solutions (typically "good" candidate solutions) are selected and information is exchanged between them to create two new candidate solutions. This exchange can take various forms, the operation used in this thesis is an arithmetic crossover which is explained in the following section. Finally, as with nature, a mutation is applied to the offspring solutions. This is done to ensure population diversity and to ensure the genetic algorithm does not get trapped into local optimal solutions.

**Genetic Resampling**

The genetic filter offers an alternative to SIR at the resampling step; instead, the resampling follows a genetic algorithm. The genetic filter is otherwise identical to the particle filter as can be seen in Figure 4.1. The intention of using a genetic algorithm for the resampling step is to provide an alternative and potentially more effective way of mitigating particle degeneracy and increasing the diversity of the particles during resampling than SIR, such that fewer particles could be used for the same level of accuracy. Mitigating particle degeneracy is important as it reduces the chance of the particles of a particle filter focusing on false positives too quickly. Essentially, the genetic filter enables the particles more opportunity to explore the search space to identify regions which could result in a higher weighting. A genetic filter has been applied to an Ultrawideband (UWB) positioning solution in [71].

The resampling step of the genetic filter is shown in Figure 4.3. It is composed of 3 steps: classification, crossover and mutation. No prior-processing of the particles is necessary as the particles of the particle filter are already an appropriate data structure for a genetic algorithm as each particle is real-value encoded by default with their combination of position, orientation and velocity states. At the classification step, the algorithm gets a set of strongly weighted and weakly weighted particles. Classification is the process of sorting the particles into mating pools to be used in the crossover step. The particles are sorted into ascending order based on

weight, which is the fitness value of this algorithm. Then the particles are split into a strong particle set and a weak particle set. The particle sets are determined with Equation 4.19.

$$nint(p) \ \leq \ N_{eff} < nint(p+1) \tag{4.19}$$

Where nint(p) represents the nearest integer to p. p represents the integer that is used as the index to split the ordered particle set, where the set with index 0 to p is the higher weighted set and the set with starting index p+1 to the total number of particles, $N_p$, is the lower weighted set. $N_{eff}$ is calculated using Equation 4.16. For the crossover step, each strongly weighted particle undergoes an arithmetic crossover with another strongly weighted particle to produce two offspring particles which will replace the two parent particles, then all weakly weighted particles undergo an arithmetic crossover with a strongly weighted parent particle and the new particles replace the original particles. Equation 4.20 and 4.21, describe the arithmetic crossover conducted on the higher weighted set, the two offspring particles, *j* and *j+1* are then used for the next epoch. The total number of particles remains the same. $\omega_k^{off}$ represents the states of the offspring particles, $\omega_k^h$ represents the states of the higher weighted particles.

$$\begin{aligned} \omega_k^{off,j} &= \ \alpha_1 \omega_k^{h,j} + (1-\alpha_1)\omega_k^{h,j+1} \\ \omega_k^{off,j+1} &= \ \alpha_2 \omega_k^{h,j+1} + (1-\alpha_2)\omega_k^{h,j} \end{aligned} \quad where \quad j \leq p \tag{4.20}$$

$$\begin{aligned} \alpha_1 &= \frac{w_k^{h,j}}{(w_k^{h,j} + \ w_k^{h,j+1})} \\ \alpha_2 &= \frac{w_k^{h,j+1}}{(w_k^{h,j} + \ w_k^{h,\ j+1})} \end{aligned} \quad where \quad j \leq p \tag{4.21}$$

Equation 4.22 describes the arithmetic crossover conducted for the lower weighted particle set. In this version of the crossover, instead of taking two parents from the same set, the highest weighted particle from the higher-weighted set is crossed

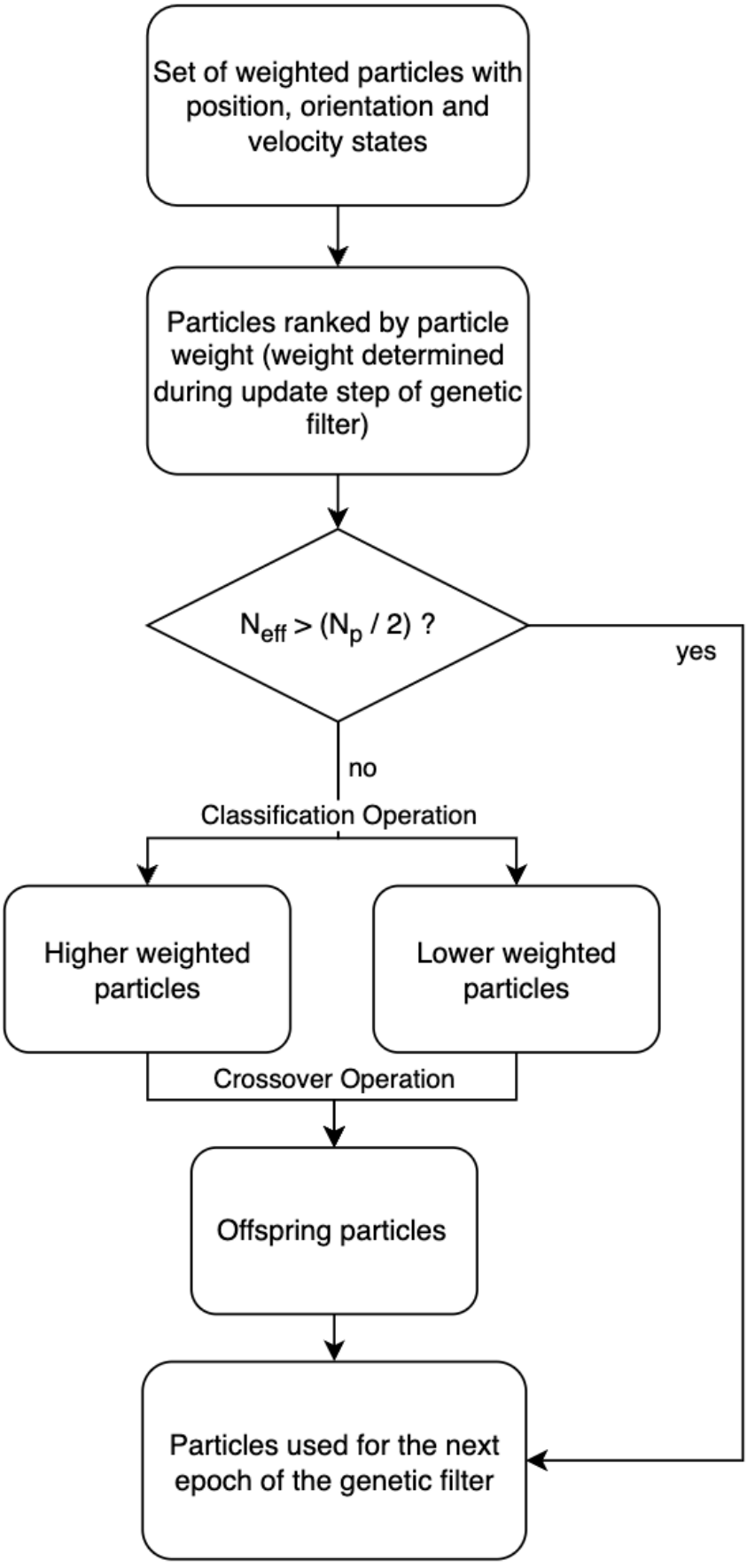


Figure 4.3: Genetic resampling process

over with a random particle from the lower-weighted particle set. The higher weighted parent particle and the new offspring particle are then passed onto the next generation. $\omega_k^l$ represents the states of the lower-weighted particles. Where $\beta$ is a random number between 0 and $(N_p - N_{eff})/N_p$.

$$\omega_k^{off,j} = \beta\omega_k^{l,j} + (1-\beta)\,\omega_k^{h,0} \quad where \quad p \leq j+1 \leq Np \tag{4.22}$$

For this algorithm, a mutation step was not implemented. This is because during the prediction step of the particle filter noise is applied to all the particles following a Gaussian distribution which achieves a similar objective as mutation. The noise added during the prediction step could be deemed as more appropriate than mutation, as the particles are mutated in accordance with an observed change in the system state.

### 4.1.5 Grid Filter

The Grid filter differs from the genetic filter and particle filter in the initialisation, prediction and resampling steps. Instead of using particles, the search space is split into grid squares. Each grid square intersection represents a candidate position of the mobile device and fundamentally, instead of particles moving around and each particle's weight being updated, weights are moved throughout the grid to represent the posterior distribution of the device's state. The process follows the following steps:

Initialisation - The search area is initialised as a square grid of a given size $\kappa$ (in mm) centred around the initial position estimate, this is then split into 2D grid squares with grid spacing, $\eta$ (in mm), to form an array of dimensions $\kappa/\eta$. Following this, the grid squares are weighted according to their distance to the centre of the grid (the initial position estimate) according to a Gaussian distribution with a standard deviation that matches the sensor noise. For this thesis $\eta$, was set as 50mm and $\kappa$ was initialised as 2m.

Prediction - For positioning in motion, the pedestrian dead reckoning model

described in Section 4.1.7 is used to retrieve the displacement and heading values. However, the position displacement differs from that of the particle filters in order to ensure that the coordinates remain in a grid formation. The displacement of all grid coordinates is determined as described by:

$$Q^j_{k+1} = Q^j_k \; + \; cos\,(\bar{\varphi}_k) \;\; \bar{r}_k \tag{4.23}$$

$$R^j_{k+1} = R^j_k \; + \; sin(\bar{\varphi}_k) \; \bar{r}_k \tag{4.24}$$

where the mean values of the heading, $\bar{\varphi}_k$ and displacement, $\bar{r}_k$ are used to displace the grid. However, this does not effectively account for sensor noise. Thus, a separate process must be followed in order to incorporate sensor noise into the grid coordinates and weights. This is described as follows. First, a new set of grid coordinates "c grid" are generated that are equal to the grid coordinates of the main grid. A multivariate Gaussian distribution is generated from the heading uncertainty and step length uncertainty. From this distribution, a new set of weights are generated for the "c grid" by calculating the probability density function of the distribution for the distances between each coordinate and the mean of the coordinates. With the weights of the "c grid" and the weights of the grid coordinates from the previous epoch the weights are convolved using direct convolution, to form a new set of weights for the grid coordinates that incorporate the sensor noise of the PDR.

Weighting – each grid point is weighted according to its distance from all the landmarks. The model used for the weighting is the same as the particle filter and genetic filter, described in Equation 4.14. This weighting represents the likelihood that the device is at that grid point. Following this, the weights are normalised using L1 normalisation. Using the weights of each grid point, the weighted average position is determined according to Equation 4.17 and 4.18. This represents the estimated position of the mobile device at that epoch. Repeat the Prediction,

Weighting and Estimation steps for each epoch.

### 4.1.6 Outlier detection techniques

In any positioning solution that uses signals, an important consideration is the reliability of the signals, since unreliable signals can ultimately result in a positioning solution with an error greater than what is acceptable for the use case. In the context of mobile indoor positioning this value would be anything above 2m as this is a big enough ranging error to place a mobile device in the incorrect room in most circumstances. Thus, within WiFi RTT based positioning in most scenarios, some ranging signals will be inaccurate and unreliable, therefore, excluding these signals from the positioning solution should theoretically improve the positioning accuracy. Two outlier detection models will be explored in this thesis: RSSI Threshold-based outlier detection and Residual-based outlier detection.

#### Residual-based outlier detection

Residual-based outlier detection is described in the following equations. It is a method used to identify how well a solution fits the measurements, this identifies gross outliers and excludes them from the final solution. The residuals have been normalised here as this is generally a more reliable indicator [1].
Firstly, the residual vectors must be calculated, as shown in Equation 4.25.

$$\boldsymbol{v} = \left(\boldsymbol{A}(\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{T} - \boldsymbol{I})\boldsymbol{b} \tag{4.25}$$

Next, the residual covariance matrix should be calculated, as shown in Equation 4.26.

$$\boldsymbol{C}_{v} = \left(\boldsymbol{I} - \boldsymbol{A}(\boldsymbol{A}^{\boldsymbol{T}}\boldsymbol{A}\right)^{-1}\boldsymbol{A}^{T})\sigma^{2} \tag{4.26}$$

Finally, the normalised residuals should be calculated and these should be compared with a threshold. In Equation 4.27 if the condition is true for the residual then measurement $\varsigma$ is an outlier and should be removed from the process such that the solution can be redetermined without the gross outlier. It is likely that removing the single greatest outlier may result in the other residuals being reduced when the

positioning solution is redetermined.

$$|v_j| < \sqrt{C_{v_{\varsigma\varsigma}}}T \tag{4.27}$$

**RSSI-based outlier detection**

As discovered in Section 3, there is a correlation between the RSS of a signal and the RTT range when a signal is LOS. Signals that do not conform to this correlation are more likely to have originated from NLOS reception for a given RTT range as a result of obstacles affecting the signal strength or the signals reflecting off walls. Since RSSI is available on all RTT readings it is possible to use this as a metric for the identification of outlier results. These outlier results can then be excluded from further processing in a positioning algorithm. The algorithm detects NLOS signals and severe multipath interference by finding and accounting for inconsistencies between the RTT range measurement and the RSSI of the signal. Guo et al [36] and Sun et al [54] have also used RSSI for outlier detection of RTT signals, however, these algorithms were optimised for the environments by conducting surveys of the RSSI path loss model and fingerprinting.

The RSSI-based outlier detection algorithm of this thesis is discussed for application to weighted least squares and particle filtering separately as the implementations are slightly different.

**RSSI-based Weighted Least Squares with RSSI-based path loss derived outlier detection**

This method takes in the strengths of the other outlier detection and positioning methods to form a solution that allows the RSSI-threshold to be set automatically, correctly distinguishes between LOS and NLOS signals and weights signals according to their closeness to a predicted RSSI value. The algorithm is defined as follows: For each epoch:

1. Use Equation 4.28 and 4.29 to determine a RSSI-threshold for an AP, these equations were derived from [10] but modified slightly to account for its use as threshold detection by increasing the constant value. Repeat this for all APs. This

gives the expected RSSI signal for the mean measured distance, which is the expected RSSI if the signal was unobstructed (i.e. LOS), therefore if the RSSI is weaker than this prediction it could indicate that the signal is obstructed and thus could provide an incorrect range estimate.

RSSI-threshold determination:

$$R_{i,k}^{threshold} = -(51.4dbmW + 20\log_{10}(\bar{d}))\ where\ \ d_{min} < \bar{d} < 8m \tag{4.28}$$

$$R_{i,k}^{threshold} = -(65.5dbmW + 33log_{10}(\bar{d})/8)\ \ where\ \bar{d} > 8m \tag{4.29}$$

For each access point, i, the expected RSSI is computed given, $\bar{d}$, which represents the mean measured RTT distance between an AP and the mobile device across 2 seconds worth of data (or approximately 20 RTT measurements if sampling at 100ms), giving the RSSI threshold, $R_{i,k}^{threshold}$. k represents the epoch and $d_{min}$ is a minimum distance, in this case 4m, below which RSSI-based outlier detection is not used.

2. If the RTT estimated range for an AP at an epoch is less than 4m then select that AP for that epoch, this is done because it was deduced during experimental analysis that for ranges below 4m, RSSI-based outlier detection resulted in a higher rate of false positives than above 4m.

3. If the RTT estimated range for an AP at an epoch is more than 4m then check if the RSSI of that signal is stronger than the RSSI-threshold. If it is then select that AP.

4. Check if the number of selected APs in an epoch is greater than 3, if it is not then increment the RSSI-thresholds of all APs for that epoch by 1dBm and repeat step 3 until there are at least 3 APs stored.

Steps 2–4 are described in Algorithm 1

**Algorithm 1** RSSI threshold based outlier detection

**Data:** n WiFi RTT raw datapoints ***RTT***, ***RSSI*** and RSSIThreshold
**Result:** $n_a$ WiFi RTT datapoints held in matrix ***RTT***

```
1  n_a = 0
2  RTT = [ ]
3  while n_a < 3 do
4      for i = 0; i < n; i = i + 1 do
5          if RTT_i < 4m then
6              if RSSI_i ≥ RSSIThreshold then
7                  n_a = n_a + 1
8                  RTT.append(RTT_i)
9              end
10         end
11     end
12     if n_a < 3 then
13         RSSIThreshold = RSSIThreshold - 1
14     end
15 end
16 return RTT
```

5. Compute the weighted least-squares position of all stored APs (that passed the above checks) using the processes described in Section 4.1.1 and 4.1.2. For the error covariance matrix for each AP, use the difference between the RSSI threshold and RSSI for that AP at that epoch for the sigma value in Equation 4.1.2 as described in Equation 4.30. This is done because the difference between the two values will provide an indication if the signal is not LOS and has thus been obstructed. This can be used as a metric for reliability as it provides an indication of whether the signal is direct or not.

$$\sigma_{ki} = R_{ki} - R_i^{threshold} \tag{4.30}$$

Where $R_{ki}$ is the RSSI value for epoch, $k$, and AP, $i$, and $R_i^{threshold}$ is the RSSI threshold for AP, $i$.

**RSSI-based outlier detection for filtering**

The RSSI-threshold is calculated using Equation 4.28 and 4.29.

During the filtering process, following the prediction step but before the update step, each AP is compared against the measured RSSI threshold of that AP. If the measured RSSI is lower than the threshold then this particle is treated with lower confidence and the absolute difference between the measured RSSI and threshold is taken, $\delta_{rssi}$. For each epoch these differences are normalised using L1 normalisation to produce an RSSI-based weighting factor, $\epsilon_k^i$, for AP, i and the epoch, k.

$\epsilon_k^i$ are then used during the update step as a multiple of the standard deviation of the measured distances, this increases the measurement noise of the measurements with an RSSI below the threshold with the magnitude of the difference creating greater measurement noise. The standard deviation is computed in the following equation:

$$\sigma_k^i = \sigma \times (1 + \epsilon_k^i) \tag{4.31}$$

This can then be used to obtain the standard deviation, $\sigma_k^i$, used in Equation 4.14 for all filters. Where $\sigma$ is the sensor standard error of the WiFi RTT range.

### 4.1.7 Pedestrian Dead Reckoning Model

A motion model must be integrated into the prediction step of the filters in order to properly reflect the expected position of a particle. Pedestrian dead reckoning is an approach that takes advantage of step detection and uses a human step to indicate the movement of the mobile device. The step detection is carried out using the Google Android API [9] due to its convenience and as this thesis does not intend on exploring step detection models. This detects a "high variation in acceleration" [9] from the mobile device's accelerometer. Following step detection, a step length estimation is required in order to determine the distance travelled by a mobile device when a step occurs. The step length varies for many reasons. It can vary due to different pedestrians having different walking gaits and pedestrians walking at different speeds. The model used in this thesis is based on the Mikov step length estimation equation [72].

$$q = max(\alpha t_{step} \sqrt[4]{(f_{z,max} - f_{z,min})}, q_{max}) \tag{4.32}$$

Where q is the step length, $\alpha$ is a constant parameter that is calibrated per pedestrian, $t_{step}$ is the time taken for a given step, $f_{z,max}$ is the maximum specific force along the z axis during the step and $f_{z,min}$ is the minimum specific force along the z axis during the step. Where the z axis points towards the outside of the front face of the screen. This model was selected because it provided an improvement [72] over the commonly used Weinberg model [73] but does not require significant calibration such that this model would be difficult to reproduce. The model was calibrated by walking along 2 paths of known length with different step sizes and selecting the value of the constant that minimised the overall distance measurement error. An upper limit of $q_{max}$ = 1100mm was imposed to mitigate the impact of gross outliers in the mobile device's IMU measurements.

The direction of travel was assumed to match the heading output by the Android 'orientation sensor' [9], which calculates 3D orientation using the device's magnetometers, accelerometers and gyroscopes. This was transformed to the coordinate frame of the indoor map. The initial heading error is smaller than subsequent heading error as these are subject to random hand-orientation changes from the pedestrian during walking.

The standard deviation of the step length assuming white noise was calibrated as follows. A pedestrian carrying the mobile device walks 20 steps in a straight line 10 times, the real distance travelled is measured and the estimated distance travelled is calculated using the PDR model, the standard deviation of the difference is computed and this is divided by the number of steps to give the standard deviation of a step.

## 4.2 Method and Data collection

The algorithms were tested in both static and motion-based scenarios. The method and experimental setup used for data collection are outlined in the following sections; once collected, the positioning solution was determined in post-processing.

### 4.2.1 Equipment

The mobile device used was a Google Pixel 4a. This is shown in Figure 4.4. The Google Pixel 4a is WiFi RTT compatible and readily available. The data collected by the mobile device included WiFi RTT data collected at 100ms intervals, WiFi RSSI data collected at 100ms intervals, on-board accelerometer, gyroscope, magnetometer and orientation data at sub 20ms intervals and the mobile-provided step counter was used for step identification. No extra equipment was used to enhance the sensor quality of the mobile device as this does not represent a realistic scenario for a pedestrian and the intention is to test the algorithm's effectiveness with low-cost sensors. A custom mobile app was developed that allowed for collecting all data simultaneously. The orientation of the mobile device for the stationary experiments is shown in Figure 4.4. For the experiments in motion the mobile device was held in the hand with the back of the phone parallel to and facing the ground.

For the stationary data collection for the least squares algorithm, one Google Nest WiFi Router and two Google Nest WiFi Points were used twice to produce 6 total access points for a given experiment. For the stationary data collection for the filtering algorithms and the in-motion data collection, four Google Nest WiFi routers and two Google Nest WiFi Points were used for the data collection. These devices are WiFi RTT compatible and relatively cheap. Each router is less than $100. A Google Nest WiFi router is shown in Figure 4.5. It is important to note that for all of the APs, a negative ranging bias existed. This varied for each AP; AP-3 and AP-4 had a small bias (0.6m) whilst AP-1, AP-2, AP-5, AP-6 had similar biases larger than AP-3 and AP-4 (1.5m). These biases were deducted from the

Figure 4.4: Google Pixel 4a in holder, operating position during ranging sessions

WiFi RTT ranges for all data analysis.

During the data collection a secondary mobile device was used to video record the trials in-motion in order to provide better ground truth data for the movement of the pedestrian. Timestamps could be used to match the location of the pedestrian at a given time in their movement to the ground truth. Finally, all distances in the environments were measured using a laser distance measure or a tape measure. As each environment had their own coordinate reference system, the APs and mobile device positions could be determined in the context of the environment's coordinate reference system.

### 4.2.2 Stationary Data collection - Least Squares algorithms

The APs and mobile device (standing, refer to Figure 4.4) were laid out as shown in Figure 4.6 for Experiment 1, Figure 4.7 for Experiment 2 and Figure 4.8 for Experiment 3. Note that a coordinate system was used based on the experimental environment, with a convenient position set as the origin, as shown by the axes in Figure 4.6, Figure 4.7 and Figure 4.8. This system will vary from environment to environment. The coordinates and devices for each experiment are shown in Table

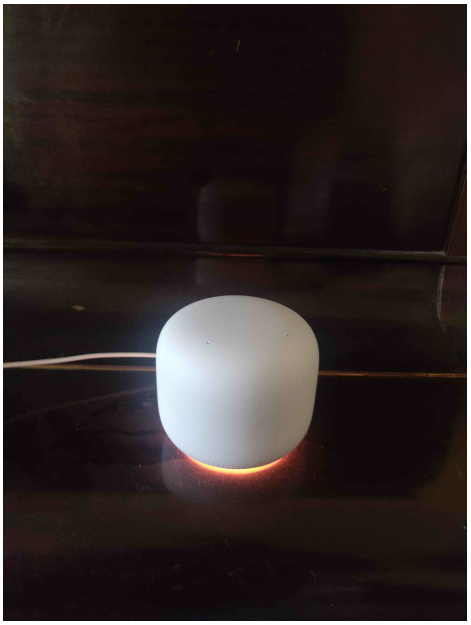

Figure 4.5: Google Nest WiFi Point

4.1, 4.2 and 4.3. All walls of the experimental environment had widths of approximately 10cm, excluding the exterior walls which were approximately 30cm. The location of the APs were mostly governed by the locations of power sockets in the experimental environment. In this experiment, a WiFI RTT scan was initiated that ranged and logged the resulting FTM RTT measurements every 500ms for one minute, resulting in approximately 120 readings per AP per session.

### 4.2.3 Stationary Data collection - Filtering algorithms

This section describes the stationary data collection for the particle filters. The environments were selected to test the algorithms across a diverse range of environments that aimed to include LOS and NLOS signal reception as well as multipath effects. Six environments were used. The diagrams of the environments are shown in Figure 4.9. In this experiment a scan was initiated that ranged and logged the resulting FTM RTT measurements every 100ms for one minute, resulting in approximately 600 readings per AP per session. The location of access points and the mobile device are as follows.

Trial A - Same as Table 4.1

Trial B - Same as Table 4.2

Table 4.1: Trial 1 - Access Points and Mobile Device

| Access Point/Device | Description | Coordinates (x,y,z) (mm) | Actual distance from AP to device (mm) |
|---|---|---|---|
| AP-1 | Google Nest Point | (2500, 8900, 760) | 2737 |
| AP-2 | Google Nest Point | (2500, 6350, 760) | 3245 |
| AP-3 | Google Nest Router | (7325, 3600, 850) | 5106 |
| AP-4 | Google Nest Router | (8550, 3250, 600) | 6027 |
| AP-5 | Google Nest Point | (9460, 6375, 750) | 4689 |
| AP-6 | Google Nest Point | (7000, 9750, 1290) | 2441 |
| Mobile Device | Google Pixel 4a | (5150, 8220, 850) | 0 |

Table 4.2: Trial 2 - Access Points and Mobile Device

| Access Point/Device | Description | Coordinates (x,y,z) (mm) | Actual distance from AP to device (mm) |
|---|---|---|---|
| AP-1 | Google Nest Point | (8200, 3300, 500) | 1639 |
| AP-2 | Google Nest Point | (7420, 5150, 650) | 3221 |
| AP-3 | Google Nest Router | (8150, 1900, 650) | 1972 |
| AP-4 | Google Nest Router | (10350, 1650, 650) | 1439 |
| AP-5 | Google Nest Point | (11500, 2650, 850) | 4743 |
| AP-6 | Google Nest Point | (9300, 5100, 500) | 2183 |
| Mobile Device | Google Pixel 4a | (9800, 2980, 650) | 0 |

Table 4.3: Trial 3 - Access Points and Mobile Device

| Access Point/Device | Description | Coordinates (x,y,z) (mm) | Actual distance from AP to device (mm) |
|---|---|---|---|
| AP-1 | Google Nest Router | (1100, 1300, 750) | 2724 |
| AP-2 | Google Nest Router | (250, 3750, 750) | 2912 |
| AP-3 | Google Nest Router | (200, 6000, 750) | 4066 |
| AP-4 | Google Nest Router | (6200, 6050, 750) | 4245 |
| AP-5 | Google Nest Router | (6200, 2450, 750) | 3178 |
| AP-6 | Google Nest Router | (4700, 100, 750) | 3444 |
| Mobile Device | Google Pixel 4a | (3100, 3150, 750) | 0 |

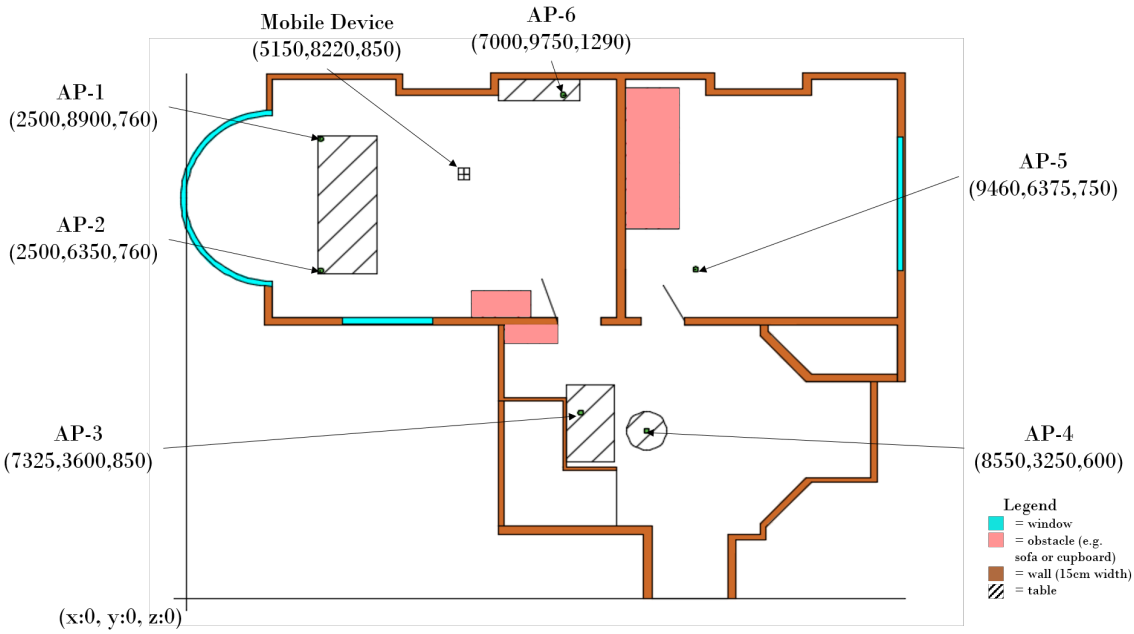


Figure 4.6: Experimental layout of APs and device for trial 1. x,y coordinates represent the distance from the origin, z coordinate represents the distance from the floor. Coordinates are represented in millimetres.

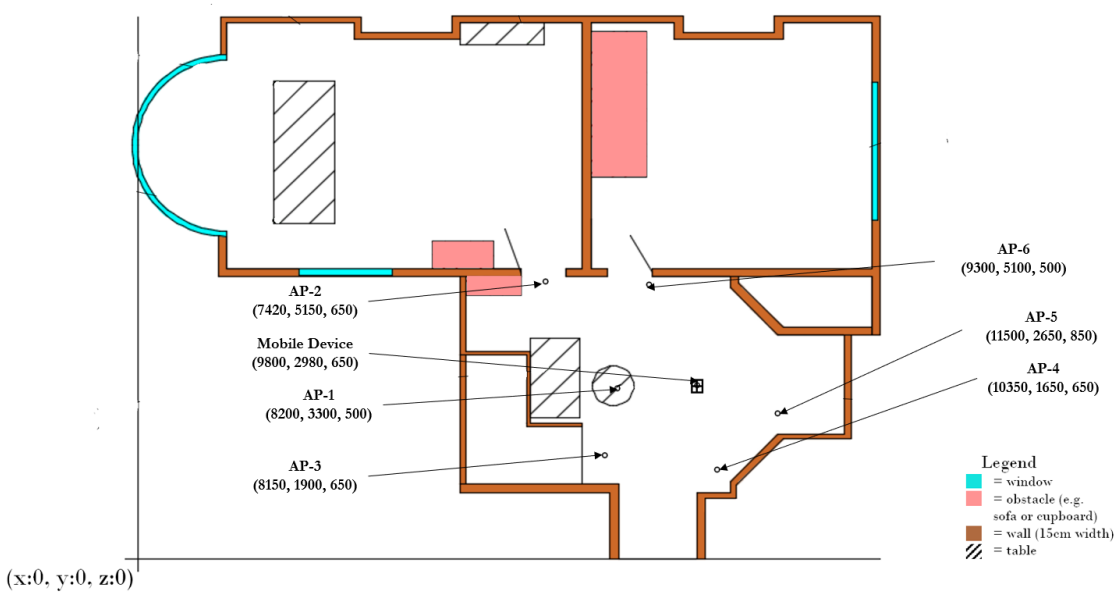


Figure 4.7: Experimental layout of APs and device for trial 2. x,y coordinates represent the distance from the origin, z coordinate represents the distance from the floor. Coordinates are represented in millimetres.

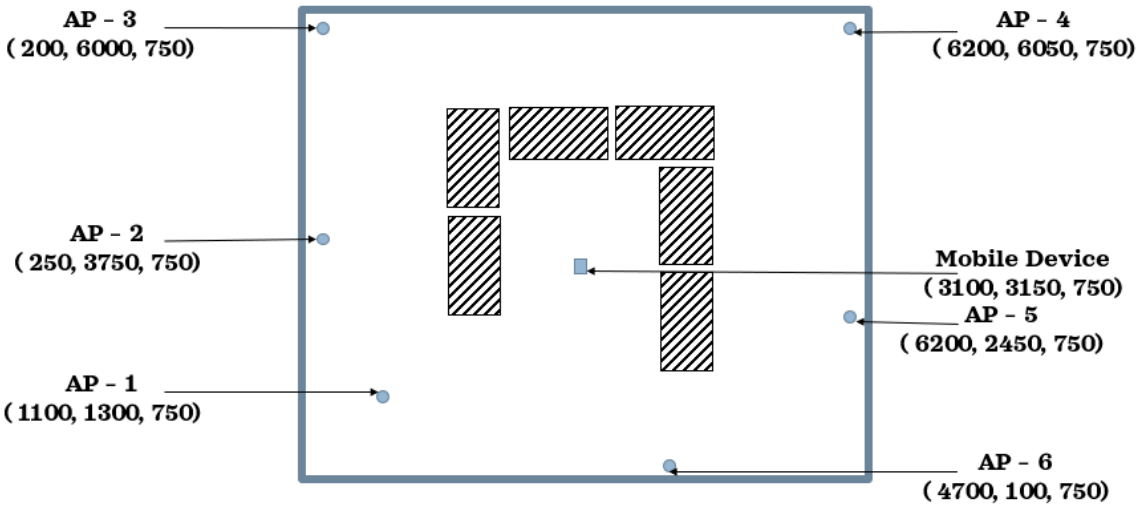


Figure 4.8: Experimental layout of APs and device for trial 3. x,y coordinates represent the distance from the origin, z coordinate represents the distance from the floor. The diagonally striped rectangles are tables that do not block signals, but may reflect them. Coordinates are represented in millimetres.

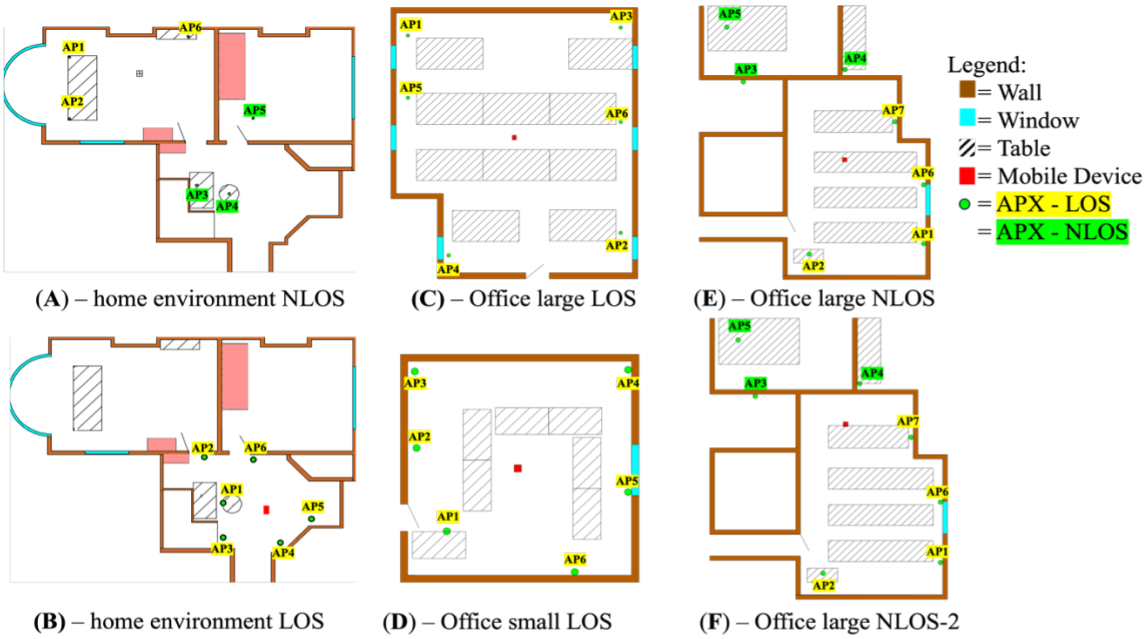


Figure 4.9: Experimental layout of APs and device for filtering trials. x,y coordinates represent the distance from the origin.

Trial C - Table 4.4

Trial D - Same as Table 4.3

Trial E and F - Table 4.5

### 4.2.4 In-Motion Data collection

The experiments to test the algorithms in motion involved moving through different routes in a single environment. The environment was irregular in shape and had

Table 4.4: Trial C - Access Points and Mobile Device

| Access Point/Device | Description | Coordinates (x,y,z) (mm) |
|---|---|---|
| AP-1 | Google Nest Router | (50, 10000, 500) |
| AP-2 | Google Nest Router | (10080, 1270, 500) |
| AP-3 | Google Nest Router | (10080, 10350, 500) |
| AP-4 | Google Nest Router | (1720, 270, 500) |
| AP-5 | Google Nest Router | (10080, 6200, 500) |
| AP-6 | Google Nest Router | (50, 7250, 500) |
| Mobile Device | Google Pixel 4a | (5000, 5500, 500) |

Table 4.5: Trial E and F - Access Points and Mobile Device

| Access Point/Device | Description | Coordinates (x,y,z) (mm) |
|---|---|---|
| AP-1 | Google Nest Router | (11200, 1500, 500) |
| AP-2 | Google Nest Router | (6000, 1000, 500) |
| AP-3 | Google Nest Router | (3000, 9300, 500) |
| AP-4 | Google Nest Router | (7500, 9450, 500) |
| AP-5 | Google Nest Router | (2250, 11800, 500) |
| AP-6 | Google Nest Router | (11200, 4300, 500) |
| Trial E Mobile Device | Google Pixel 4a | (7600, 5500, 500) |
| Trial F Mobile Device | Google Pixel 4a | (7000, 7900, 500) |

many walls, both interior and exterior, to create a diverse range of multipath and NLOS scenarios during the routes walked by the pedestrian. The algorithms were tested across 3 different routes, varying in complexity and NLOS conditions; these are shown in Figure 4.10. Each route was repeated in both directions, so in total 6 scenarios are collected. In Figure 4.10 the "forward" route starting point is shown with the purple rectangle and the "reverse" route starting point is shown with the red rectangle. The trials involved a pedestrian holding the smartphone and walking on top of fixed step markers placed on the ground along the route of the trial. The markers were approximately 670mm apart. The location of each marker was measured against the reference origin to generate the step marker's ground truth coordinates. In order to align the ground truth data with the measured data, the trials were filmed. The timestamps of the steps from the video were used to determine the expected position of a device at a given time, allowing the algorithms to be compared to the ground truth at each step during data analysis.

## 4.3 Results and Discussion

### 4.3.1 Least squares positioning

Figure 4.11 shows the results of the positioning experiment for Trial 1, using least squares positioning as the positioning solution. The results also show the two outlier

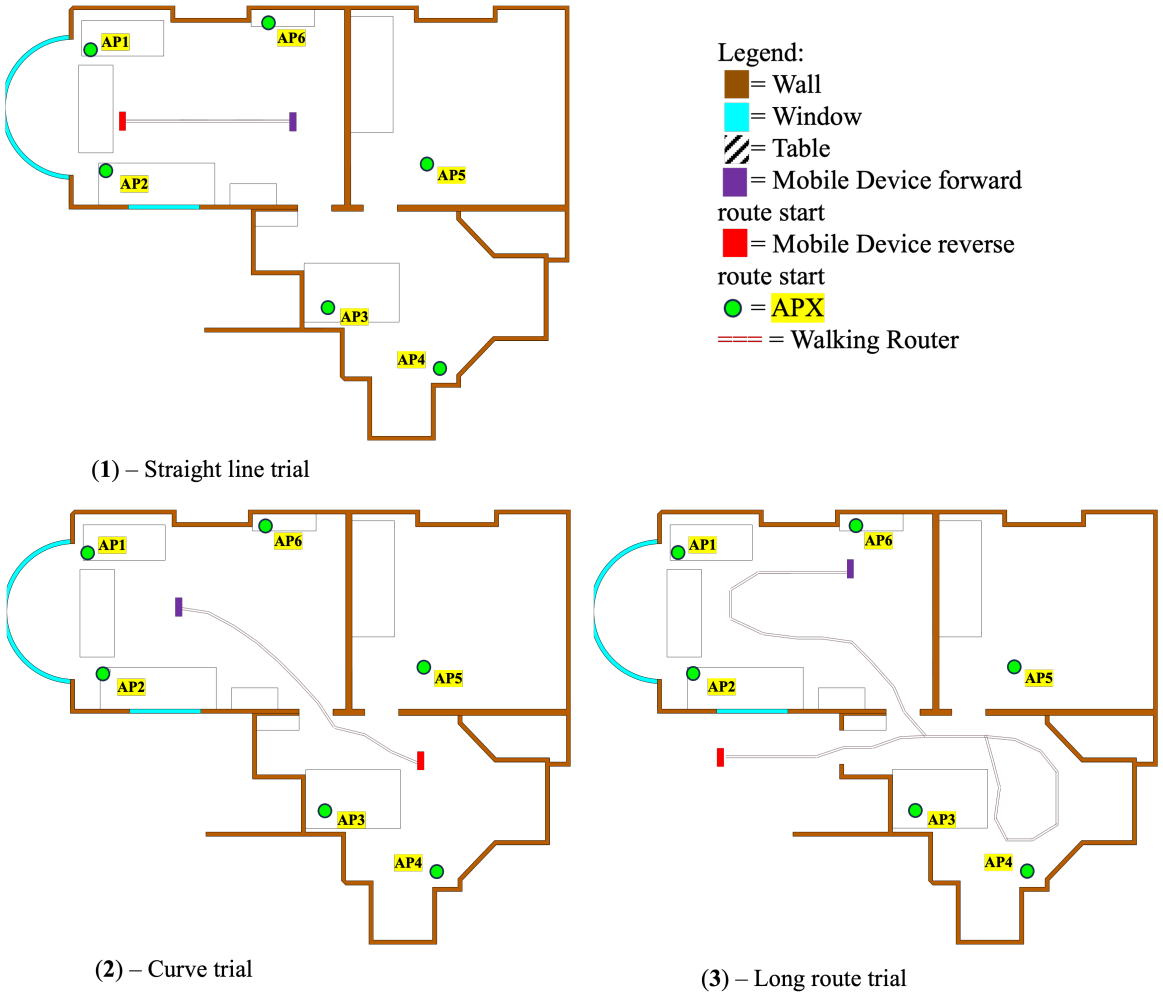


Figure 4.10: Experimental layout of APs and device for in-motion trials. x,y coordinates represent the distance from the origin.

Table 4.6: In motion trial positions

| Access Point/Device | Description | Coordinates (x,y,z) (mm) |
|---|---|---|
| AP-1 | Google Nest Router | (2050, 9150, 500) |
| AP-2 | Google Nest Router | (2500, 6450, 500) |
| AP-3 | Google Nest Router | (7650, 3150, 500) |
| AP-4 | Google Nest Router | (10250, 1700, 500) |
| AP-5 | Google Nest Router | (9950, 6450, 500) |
| AP-6 | Google Nest Router | (6300, 9850, 500) |
| Trial 1 Mobile Device Forward | Google Pixel 4a | (6800, 7450, 500) |
| Trial 1 Mobile Device Reverse | Google Pixel 4a | (2780, 7450, 500) |
| Trial 2 Mobile Device Forward | Google Pixel 4a | (4100, 7850, 500) |
| Trial 2 Mobile Device Reverse | Google Pixel 4a | (9700, 4200, 500) |
| Trial 3 Mobile Device Forward | Google Pixel 4a | (6000, 8650, 500) |
| Trial 3 Mobile Device Reverse | Google Pixel 4a | (3150, 4450, 500) |

detection algorithm position results.

In Figure 4.11, the calibrated least squares positions are tightly clustered at approximately 2m away from the true position of the mobile device. The solutions from each epoch are consistent with each other. Once RSSI-based outlier detection is introduced, the positioning solution drastically improves, placing the position within 20cm of the mobile device with consistent reliability and accuracy for this trial. This can be quantified via the Root Mean Square Error (RMSE) which was 2.03m and 0.23m for Least squares without RSSI-based outlier detection and Least Squares with RSSI-based outlier detection, respectively. The residual-based outlier detection does not perform well and does not seem to provide a positioning solution that is more accurate than the raw least-squares results.

The RSSI readings and average ranges for each AP are shown in Table 4.7. As can be seen, a lower RSSI correlates with a larger range discrepancy of approximately 1.9m. This explains the large position inaccuracy seen with the raw results and once

Table 4.7: Trial 1 - Access Point RSSI and average RTT ranging measurements

| Access Point | Mean RSSI (dBm) | True Range (m) | Mean Measured Range (m) | Ranging Error (m) |
|---|---|---|---|---|
| AP-1 | -58 | 2.73 | 3.12 | 0.39 |
| AP-2 | -59 | 3.25 | 2.91 | -0.34 |
| AP-3 | -75 | 5.11 | 7.00 | 1.89 |
| AP-4 | -71 | 6.03 | 7.92 | 1.89 |
| AP-5 | -61 | 4.69 | 4.85 | 0.16 |
| AP-6 | -56 | 2.44 | 1.82 | -0.62 |

the larger discrepancies are removed, the positioning accuracy performs well as shown by the large decrease in RMSE described above, hence providing further evidence that RSSI can be a strong indicator for WiFi RTT measurement accuracy and reliability.

Furthermore, this demonstrates issues with the least-squares positioning method without outlier detection, as an incorrect measurement can affect the entire positioning solution and thus this positioning solution can produce inaccurate final positions depending on the inaccuracy of that specific reading. A database of RSSI path loss models could allow for thresholds to be tailored to each specific environment. This could allow environment-specific outlier detection on all RTT readings, which would present the possibility to identify inaccurate RTT range in most environments, environments with highly reflective surfaces or environments with large amounts of high density obstacles will be a problem for any RSSI-based outlier detection technique. However, this suffers from a common issue of indoor positioning solutions where there is a reliance on costly surveying of the environment before indoor positioning can be conducted. As a result, the RSSI-based outlier detection model will retain a generic model in order to avoid environment over-optimisation.

Figure 4.12 shows the results of the positioning experiment for Trial 2, using least squares positioning as the positioning solution. The results also show the positioning

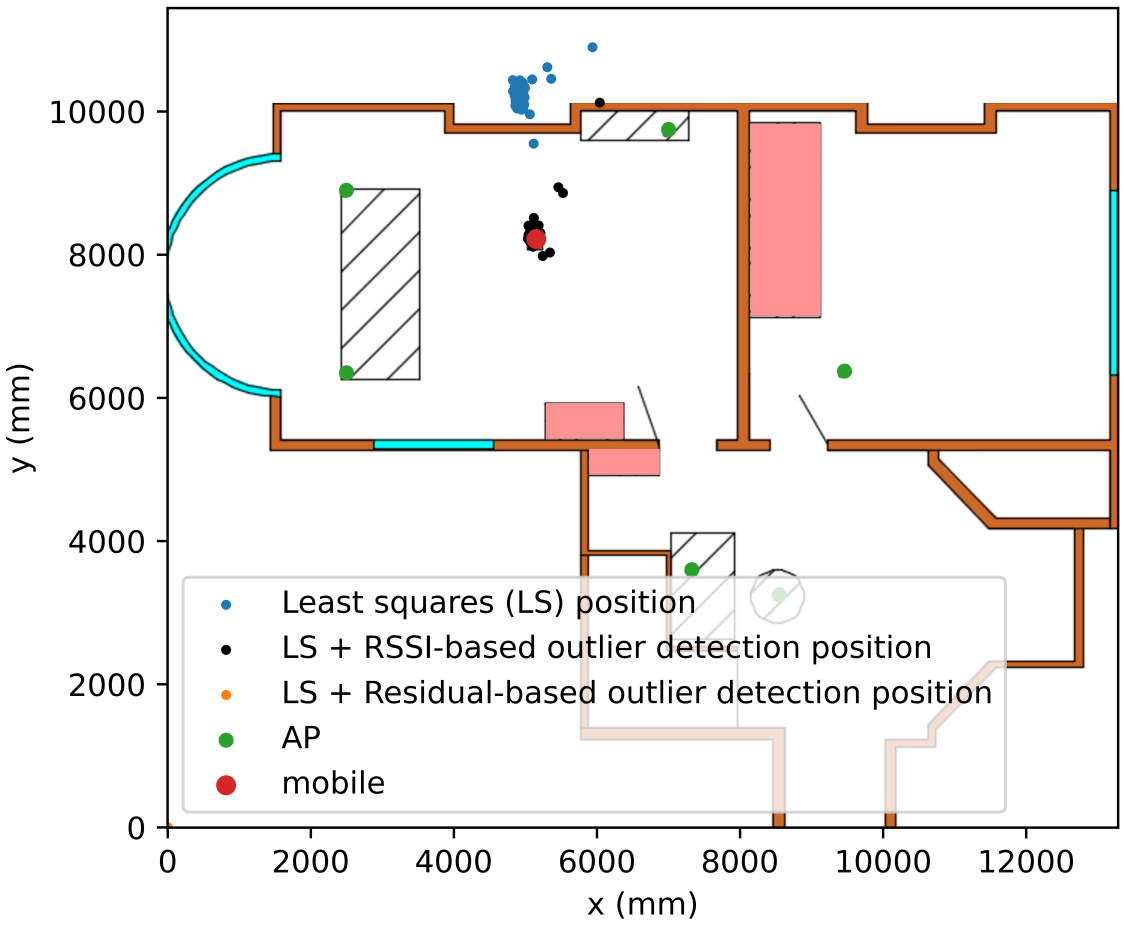


Figure 4.11: the results of the experiments on trial 1 using least squares positioning and outlier detection, with APs spread across multiple rooms

Table 4.8: Trial 2 - Access Point RSSI and average RTT ranging measurements

| Access Point | Mean RSSI (dBm) | True Range (m) | Mean Measured Range (m) |
|---|---|---|---|
| AP-1 | -58 | 1.64 | 0.87 |
| AP-2 | -56 | 3.22 | 3.81 |
| AP-3 | -57 | 1.97 | 2.06 |
| AP-4 | -53 | 1.44 | 0.78 |
| AP-5 | -58 | 1.74 | 1.26 |
| AP-6 | -58 | 2.18 | 2.34 |

solution when the two outlier detection algorithms (RSSI-based outlier detection and residual-based outlier detection) are applied to the least squares method.

In trial 2 the APs had unobstructed lines of sight to the mobile device. As seen in Figure 4.12, the positioning solutions for the least squares with no outlier detection, least squares with RSSI-based outlier detection and least squares with residual-based outlier detection perform similarly with an RMSE of 487mm, 667mm and 480mm, respectively. This actually suggests that RSSI-based outlier detection reduced the accuracy of the positioning solution when compared to no outlier detection and residual-based outlier detection provided a minor improvement. Table 4.8 shows the mean RSSI values and mean measured ranges, generally there are no major discrepancies in the ranges. The RSSI values of Trial 2 have a range of 5dBmW compared to a range of 19dBmW in Trial 1, this suggests that the RF signals in Trial 2 were roughly similar in strength.

### 4.3.2 RSSI-Weighted least squares positioning with RSSI-based outlier detection

For Trial 1, The positioning solution using RSSI-weighted least squares positioning with RSSI-based outlier detection showed an RMSE of 239mm compared to 2025mm for least-squares using all measurements. This is visualised in Figure 4.13, suggesting that in this trial this model works well for automatically selecting an acceptable RSSI threshold value. In Trial 2 the algorithm produces an RMSE of

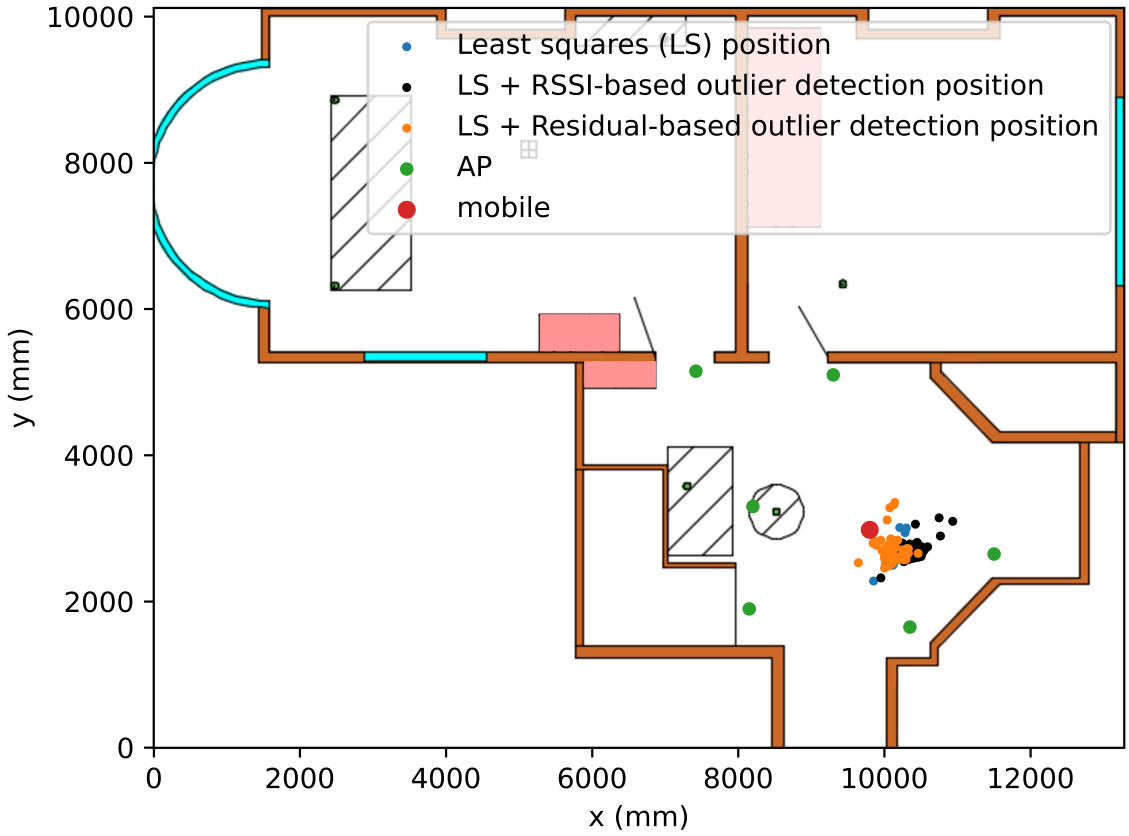


Figure 4.12: the results of the experiments on trial 2 using least squares positioning and outlier detection, with a tighter cluster of APs

458mm, meaning that the algorithm performs better than least-squares with no outlier detection, even when all signals are unobstructed which had an RMSE of 487mm. This demonstrates the benefit of not removing signals that have a measured RTT range below 4m, this is because these signals have a higher chance of being reliable. The results of this trial are visualised in Figure 4.14.

In Trial 3, this positioning solution shows further improvements. The RMSE for least squares with no outlier detection, manually configured RSSI-based outlier detection and the weighted least squares algorithm were 499mm, 493mm and 418mm respectively. This is visualised in Figure 4.15.

The RSSI-based weighted least squares algorithm performs best on all trials. This is because in shorter range environments where manually configured RSSI threshold-based outlier detection was used, signals may be incorrectly removed. This doesn't occur in the RSSI-weighted least squares positioning algorithm as smaller ranges are kept regardless of whether they are above or below their respective RSSI thresholds. The algorithm also performs well in obstructed line of sight environments as the RSSI-threshold detection that uses an RF loss propagation model can predict when a signal has not come from a direct line of sight.

### 4.3.3 Stationary Filtering

The resulting measurements were then processed by all algorithms with and without RSSI-based outlier detection to determine a final position solution. First, all algorithms will be analysed against what WiFi RTT has been advertised to be capable of, then each algorithm will be compared against one another, then the algorithms will be compared with and without outlier detection. Finally, the computational efficiency of the algorithms will be explored.

The root-mean-square error (RMSE) of the positioning solution for each environment is shown in Table 4.9. 27 out of 42 or 64.3% of tests produced sub-metre accuracy, 41 out of 42 or 97.6% trials had an RMSE below 2 metres,

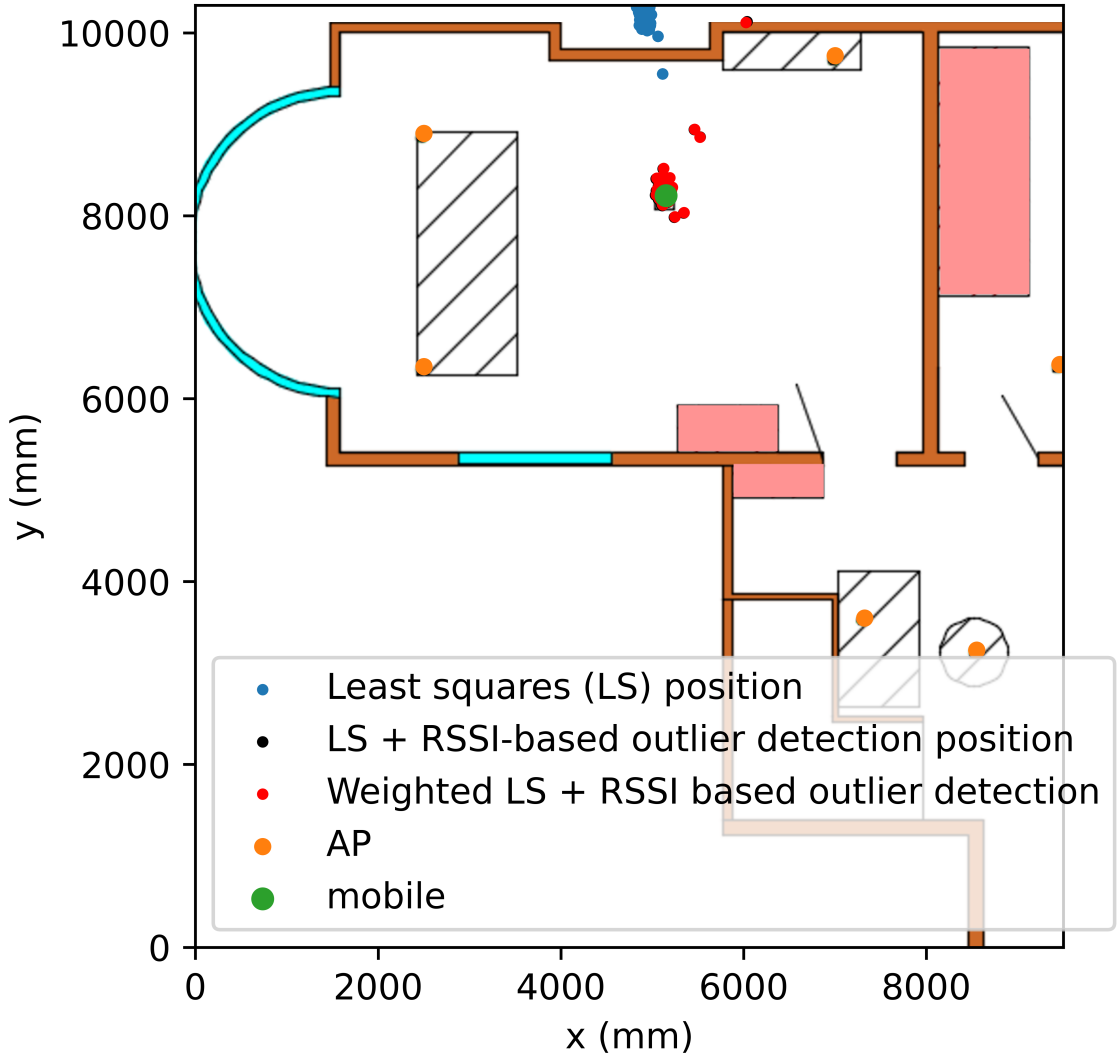


Figure 4.13: Trial 1 - RSSI-Weighted least squares positioning with RSSI-based outlier detection positioning result

| Environment | A (NLOS) | B (LOS) | C (LOS) | D (LOS) | E (NLOS) | F (NLOS) |
|---|---|---|---|---|---|---|
| Algorithm | RMSE (m) | | | | | |
| Single Epoch Least Squares | 2.19 | 1.25 | 0.93 | 0.54 | 1.26 | 1.17 |
| Particle Filter | 1.09 | 0.41 | 0.40 | 0.46 | 1.24 | 1.20 |
| Particle Filter + outlier detection | 0.71 | 0.46 | 0.35 | 0.54 | 1.15 | 0.75 |
| Genetic Filter | 1.16 | 0.58 | 0.38 | 0.43 | 1.12 | 0.88 |
| Genetic Filter + outlier detection | 0.85 | 0.51 | 0.35 | 0.54 | 0.62 | 0.56 |
| Grid Filter | 1.09 | 0.28 | 0.52 | 0.44 | 1.19 | 1.17 |
| Grid Filter + outlier detection | 0.45 | 0.34 | 0.48 | 0.46 | 1.10 | 1.07 |

Table 4.9: Positioning solution RMSE for each environment and algorithm configuration

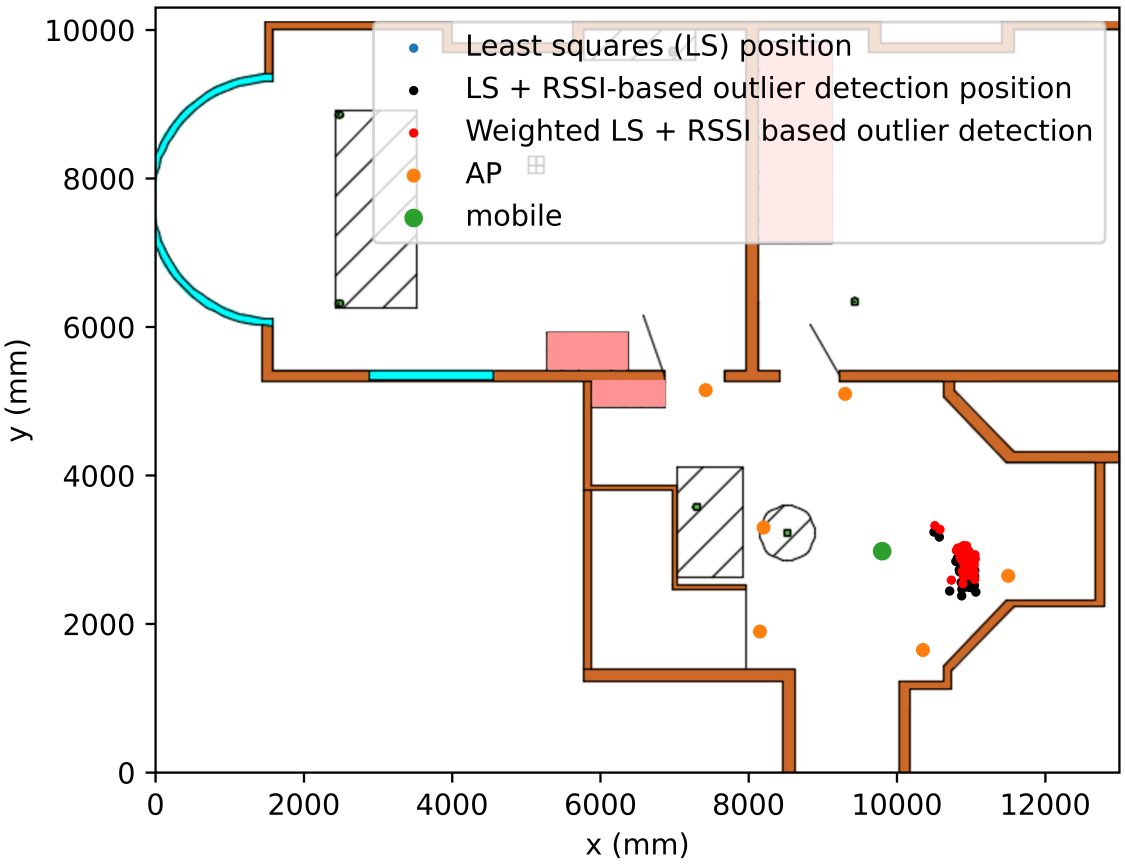


Figure 4.14: Trial 2 - RSSI-Weighted least squares positioning with RSSI-based outlier detection positioning result

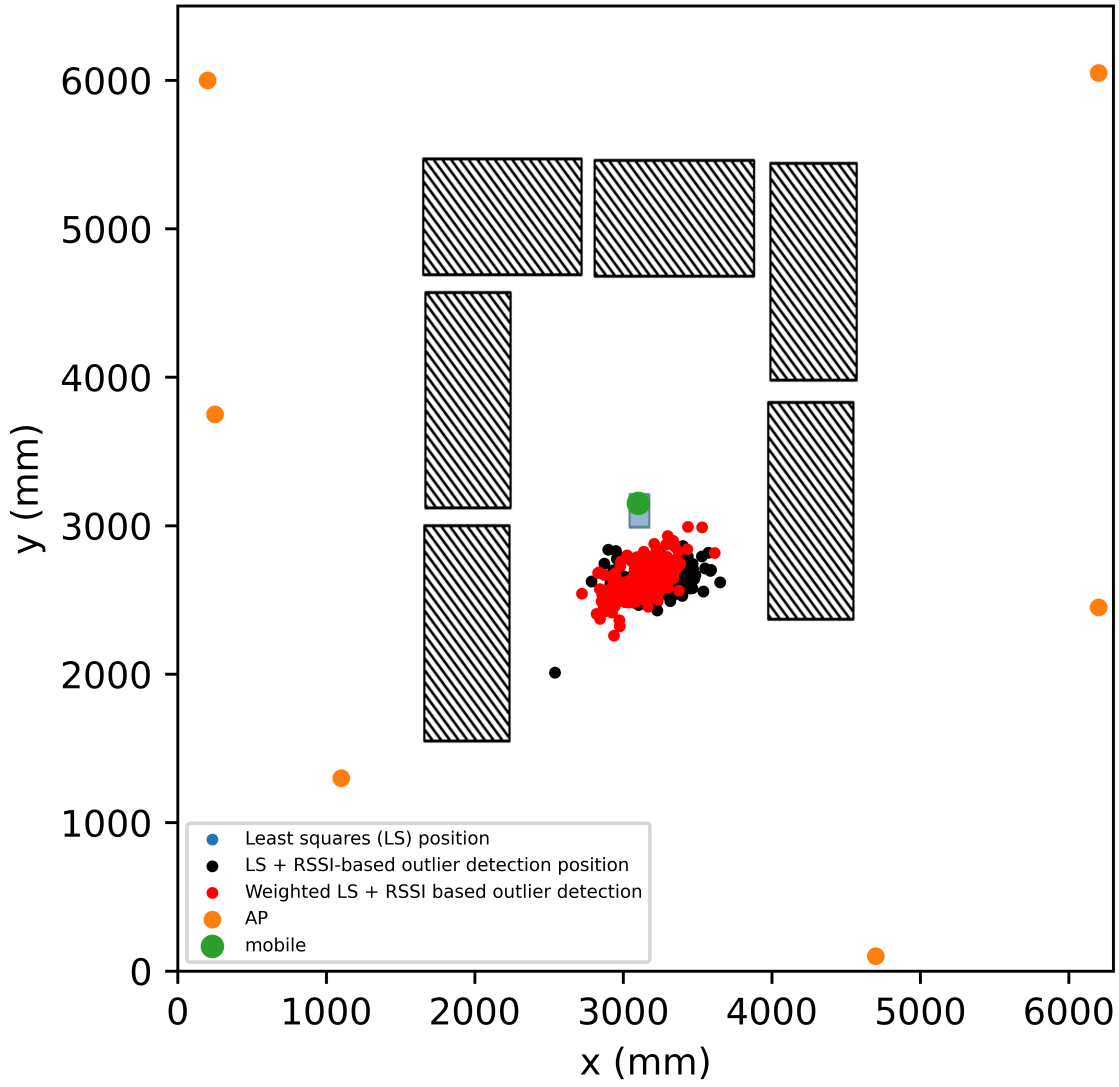


Figure 4.15: Trial 3 - RSSI-Weighted least squares positioning with RSSI-based outlier detection positioning result

| Environment | A (NLOS) | B (LOS) | C (LOS) | D (LOS) | E (NLOS) | F (NLOS) | Mean Percentage Improvement |
|---|---|---|---|---|---|---|---|
| Algorithm | RMSE (%) | | | | | | |
| Particle Filter | 50% | 67% | 57% | 16% | 2% | -3% | 32% |
| Particle Filter + outlier detection | 67% | 63% | 62% | 1% | 9% | 36% | 40% |
| Genetic Filter | 47% | 54% | 59% | 21% | 11% | 25% | 36% |
| Genetic Filter + outlier detection | 61% | 59% | 62% | 0% | 51% | 52% | 48% |
| Grid Filter | 50% | 78% | 45% | 19% | 6% | 0% | 33% |
| Grid Filter + outlier detection | 80% | 73% | 49% | 16% | 13% | 9% | 40% |
| Mean Percentage Improvement | 59% | 66% | 56% | 12% | 15% | 20% | |

Table 4.10: Percentage decrease of RMSE against least squares for each environment and algorithm configuration

indicating there is a strong argument for WiFi RTT being able to produce sub-metre accuracy for positioning as claimed by Google [2] as long as the AP biases are calibrated. Environments B, C and D where all APs had a LOS to the mobile device produced sub-metre accuracy for 20 out of 21 trials, indicating that in optimal conditions (with calibration), WiFi RTT could provide a reasonable positioning solution for most pedestrian navigation use cases. In environments A, E and F where there were NLOS signals present, only 7 out of 21 trials achieved sub-metre accuracy. This is to be expected as NLOS and multipath effects vary substantially from environment to environment. As most indoor pedestrian navigation and tracking use cases will likely involve NLOS signals, this needs to be improved. The filters and outlier detection provide an improvement. In the case of the genetic filter's improvement over the particle filter, this could be attributed to poor mitigation of particle degeneracy as the genetic filter has the most advanced particle degeneracy mitigations.

Table 4.10 shows the percentage improvement of each algorithm combination against the baseline least squares algorithm. In all environments except environment F, all of the filters improved the positioning accuracy over least square. However, in environment F, the particle filter and grid filter without outlier detection performs

| | A (NLOS) | B (LOS) | C (LOS) | D (LOS) | E (NLOS) | F (NLOS) |
|---|---|---|---|---|---|---|
| Particle Filter | 0% | 0% | 0% | 0% | 0% | 0% |
| Particle Filter + outlier detection | 35% | -12% | 12% | -18% | 7% | 38% |
| Genetic Filter | | | | | | |
| Genetic Filter + outlier detection | 27% | 11% | 8% | -26% | 45% | 36% |
| Grid Filter | | | | | | |
| Grid Filter + outlier detection | 59% | -20% | 8% | -4% | 8% | 9% |
| Mean percentage improvement | 40% | -7% | 9% | -16% | 20% | 28% |

Table 4.11: Percentage decrease of RMSE comparing outlier detection against no outlier detection for each algorithm

worse than least squares. The Grid Filter with outlier detection produced the highest improvement overall of 80% in Environment A. The algorithms provided the greatest mean improvement in Environment B; this is the simplest environment with the smallest distances between the APs and mobile devices and no NLOS signals and would be expected to have the highest accuracy given there are fewer error sources when compared to more complex environments like E and F.

The best performing algorithm on average across all environments was the genetic filter with and without outlier detection with respectively 48% and 36% mean percentage improvement over single epoch least squares. This could be attributed to better handling of particle degeneracy as well as RSSI-based outlier detection, which provides a bonus 12% over the algorithm with no outlier detection. Furthermore, the only case where the algorithms did not improve accuracy was with the particle filter without outlier detection in environment F.

Table 4.11 focuses specifically on the improvement that RSSI-based outlier detection provided for each algorithm in each environment. For all the environments where NLOS signals were present, RSSI-based outlier detection provided an improvement on average. This is because the RSSI-based outlier detection model's purpose is to de-weight NLOS signals by identifying inconsistencies with the received RSSI and measured range as signals that are not direct will have weaker RSSIs due to signal

reflection, building attenuation and multipath effects. All results produced by the genetic filter with outlier detection and 5 out of 6 of the results produced by the particle filter with outlier detection resulted in sub-metre accuracy. This is because the RSSI-based outlier detection is identifying and de-weighting NLOS signals successfully, thus prioritising stronger and more reliable RTT signals. In environment F, the mean improvement was 28% with the outlier detection providing a 38% improvement for the particle filter and a 36% improvement for the genetic filter. For environments B, C and D, RSSI-based outlier detection was less effective and in 5 out of 9 cases provided worse performance than no outlier detection. This is because the outlier detection model is best placed for identifying outliers from NLOS signals whereas in situations where LOS signals it is possible that the model will remove signals that have reliable ranges but have had reduced RSSI for other reasons such as noise or multipath effects. The model caused a 16% reduction in performance accuracy in environment D, suggesting that the algorithm may need to be better refined for environments with LOS signals. However, these environments are less common in real world use cases like pedestrian navigation and thus are not as important as getting a better method for dealing with NLOS signal reception error. Additionally, any outlier detection algorithm will have false negatives if not properly tuned to a specific scenario.

Comparing the stationary results to another paper, Guo et al. [36] showed that a positioning model using a Kalman filter and WiFi RTT achieved a mean positioning error of 2.042m. When the algorithm incorporated RSSI data into a form of outlier detection, the mean positioning error improved to 1.435m. When compared to RSSI fingerprinting, which achieved 3.41m on the same data in that paper, it is clear that WiFi RTT can provide an improvement over current techniques.

The computational efficiency of all algorithms (ran with outlier detection) was compared and the results are shown in Table 4.12, showing the computation time per epoch and the mean accuracy improvement. It is worth noting that the code is currently not optimised and is written in Python and ran on a MacBook Pro 2021

| Algorithm | Computation Time per epoch (ms) | Mean Accuracy Improvement |
|---|---|---|
| Least Squares | 0.47 | 0% |
| Particle Filter (400 particles) | 3.39 | 40% |
| Genetic Filter (400 particles) | 5.52 | 48% |
| Grid Filter (400 grid intersects) | 3.21 | 40% |

Table 4.12: Computation time per epoch for each algorithm alongside mean accuracy improvement over least squares

(Apple M1 Pro) so these computation times could be reduced significantly. The computational efficiency is as expected; the genetic filter has more computation than the other algorithms and as expected has the longest total processing time. Positioning performance would seem to correlate with processing load.

### 4.3.4 Filtering in-motion

In this section, the algorithms will be analysed for all scenarios where the mobile device was in-motion. Each scenario going forward and in reverse will be analysed separately. Table 4.13 summarises the results of the trials such as the minimum positioning solution RMSE at any step in the trial, the maximum positioning solution RMSE at any step in the trial, the average positioning solution RMSE of all steps in the trial and the final positioning estimate RMSE of a trial.

As shown in Table 4.13, the least squares solution positioning accuracy is poorer than the other algorithms. The average positioning RMSE for least squares was 2.79m when taking an average across all trials. For all trials except Trial 2R, least squares was the worst performing algorithm based on the average RMSE across all steps. Furthermore, for all trials except Trial 2R and 3R, the final position RMSE was highest for least squares. Observing the least squares position estimates in the maps shown in Figures 4.18, 4.19, 4.20, 4.21, 4.22 and 4.23, it is clear that irrespective of the results in Table 4.13, the position estimates do not correctly follow the pedestrian's path and are visibly inaccurate, placing the pedestrian in an incorrect room on most occasions, demonstrating the importance of the PDR model

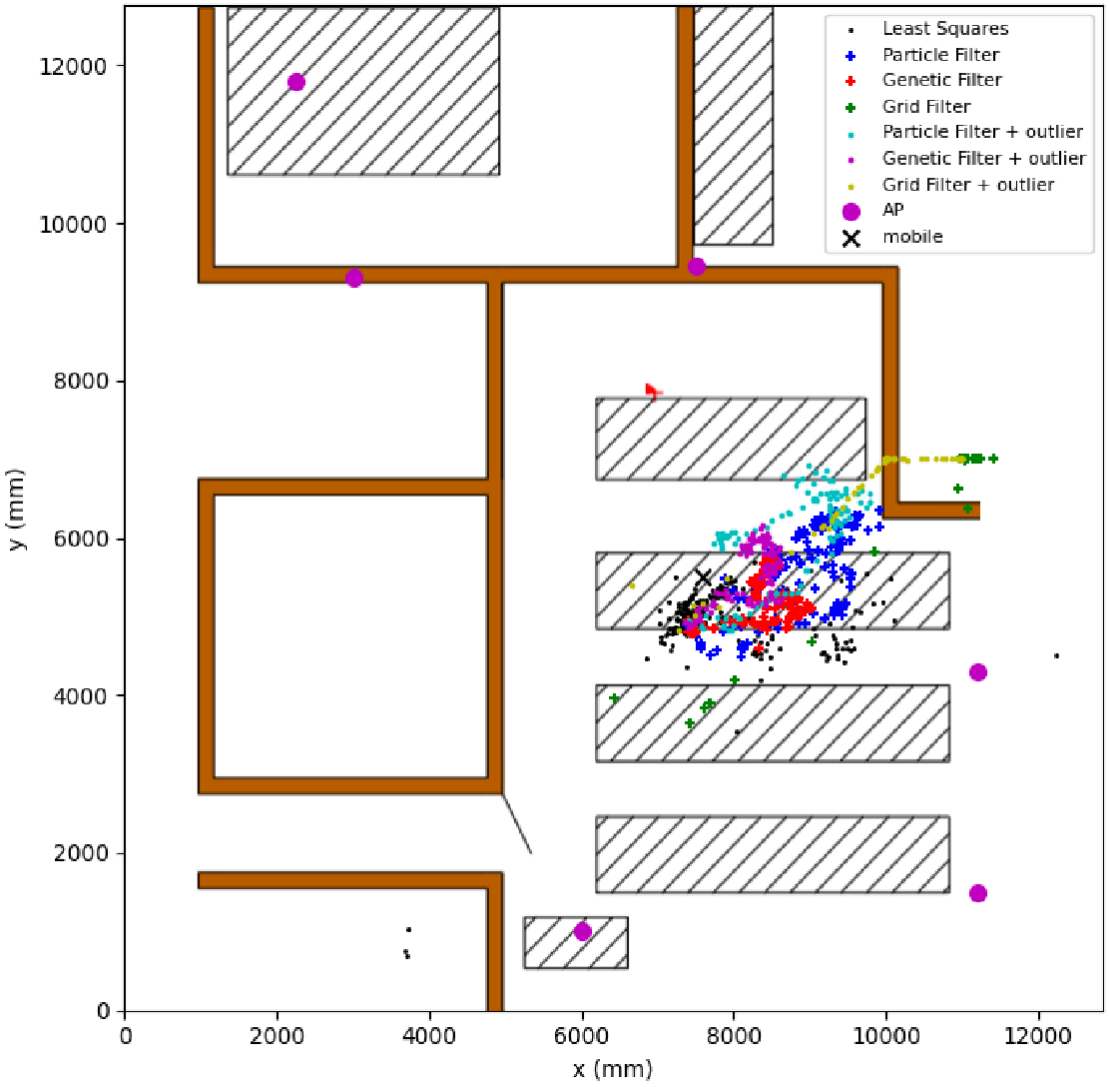


Figure 4.16: Environment E particle distribution diagram

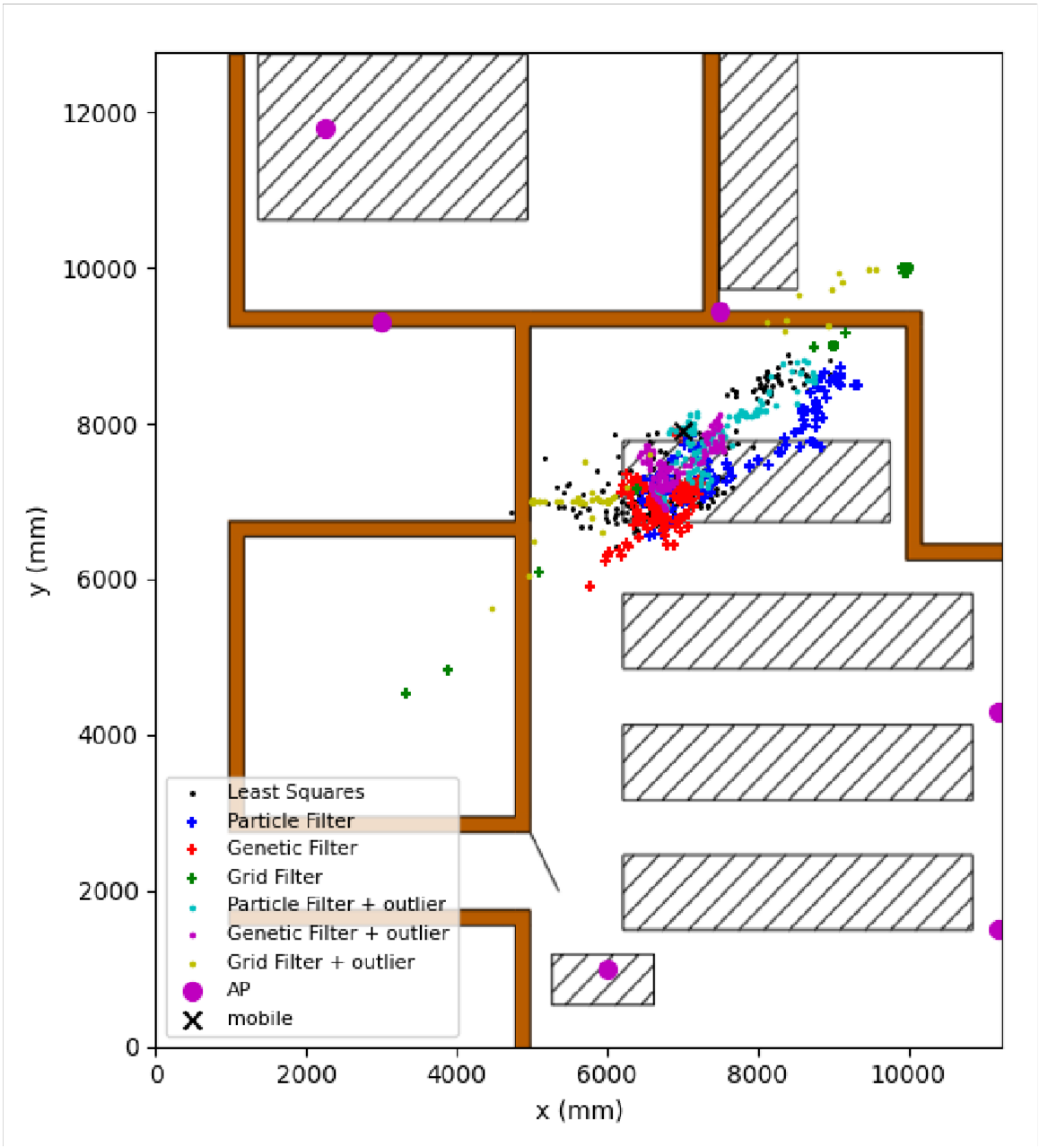


Figure 4.17: Environment F particle distribution diagram

| Environment | Statistic over all steps for Trial | Least Squares | Particle Filter | Particle Filter + RSSI | Genetic Filter | Genetic Filter + RSSI | Grid Filter | Grid Filter + RSSI |
|---|---|---|---|---|---|---|---|---|
| Trial 1F position RMSE (m) | Minimum | 0.58 | 0.36 | 0.44 | 0.41 | 0.38 | 0.41 | 0.24 |
| | Maximum | 5.25 | 0.79 | 0.58 | 0.64 | 0.78 | 1.16 | 1.08 |
| | Average | 3.33 | 0.65 | 0.50 | 0.50 | 0.51 | 0.85 | 0.75 |
| | Final Position | 3.99 | 0.47 | 0.48 | 0.57 | 0.69 | 0.41 | 0.25 |
| Trial 1R position RMSE (m) | Minimum | 2.47 | 2.34 | 2.04 | 2.25 | 2.09 | 2.31 | 2.37 |
| | Maximum | 4.73 | 3.11 | 3.11 | 3.11 | 3.11 | 3.11 | 3.11 |
| | Average | 3.37 | 2.53 | 2.25 | 2.44 | 2.32 | 2.55 | 2.66 |
| | Final Position | 3.37 | 2.25 | 2.55 | 2.68 | 2.46 | 2.66 | 2.76 |
| Trial 2F position RMSE(m) | Minimum | 0.97 | 0.57 | 0.27 | 0.83 | 0.42 | 0.56 | 0.57 |
| | Maximum | 8.49 | 1.64 | 1.62 | 1.62 | 1.62 | 1.76 | 1.64 |
| | Average | 3.03 | 1.19 | 0.90 | 1.28 | 0.97 | 1.29 | 1.19 |
| | Final Position | 8.49 | 0.78 | 0.30 | 1.15 | 0.67 | 0.56 | 0.55 |
| Trial 2R position RMSE (m) | Minimum | 0.22 | 1.06 | 0.96 | 1.12 | 0.96 | 1.74 | 1.70 |
| | Maximum | 3.01 | 3.01 | 3.01 | 3.01 | 3.01 | 3.01 | 3.01 |
| | Average | 1.76 | 1.43 | 1.41 | 1.48 | 1.50 | 2.32 | 2.30 |
| | Final Position | 2.41 | 1.78 | 1.86 | 1.89 | 2.10 | 2.42 | 2.41 |
| Trial 3F position RMSE (m) | Minimum | 0.58 | 0.28 | 0.26 | 0.10 | 0.15 | 0.52 | 0.36 |
| | Maximum | 9.81 | 3.13 | 3.01 | 3.39 | 3.23 | 4.22 | 4.22 |
| | Average | 2.74 | 1.11 | 1.02 | 1.16 | 1.05 | 1.86 | 1.77 |
| | Final Position | 4.00 | 1.00 | 0.78 | 1.14 | 0.87 | 2.51 | 2.37 |
| Trial 3R position RMSE (m) | Minimum | 0.11 | 1.05 | 0.89 | 0.80 | 0.88 | 0.15 | 0.23 |
| | Maximum | 8.33 | 2.76 | 2.58 | 2.63 | 2.64 | 2.03 | 1.90 |
| | Average | 2.48 | 2.06 | 1.75 | 1.93 | 1.69 | 1.28 | 1.27 |
| | Final Position | 1.43 | 1.51 | 1.15 | 1.77 | 1.30 | 0.77 | 0.76 |
| Average of Average across all Trials position RMSE (m) | | 2.79 | 1.49 | 1.31 | 1.47 | 1.34 | 1.69 | 1.66 |

Table 4.13: RMSE position error statistics for trials in motion and each algorithm combination

| | Initial Position RMSE (m) |
|---|---|
| Trial 1F | 0.58 |
| Trial 1R | 3.11 |
| Trial 2F | 1.62 |
| Trial 2R | 3.01 |
| Trial 3F | 1.44 |
| Trial 3R | 1.85 |

Table 4.14: Initial Position RMSE for each trial in motion

in the positioning algorithm. The following analysis will exclude least squares due to the reasons described above.

In 8 out of 36 trials with filtered solutions, the average position RMSE over all steps of a given trial was sub-metre and in 29 out of 36 trials the average position RMSE was below two metres. In virtually all trials, the particle filter and genetic filter performed similarly to each other and the grid filter performed worse on average in comparison to the other algorithms when looking at average RMSE. All algorithms in Trial 1R had a positioning RMSE greater than 2m. For Trial 1R, as can be seen in Figure 4.19, the path of all filters are translated around 2.5m to the north of the actual path. This positioning error is worse than Trial 1F for all algorithms. The poor accuracy of the algorithms in comparison could be a result of poor initialisation which could be a result of fewer LOS APs. During the earlier part of the trial. This discrepancy is potentially caused by the geometry of the Access Points, as shown in Figure 4.19, in combination with the pedestrian's path. In this trial the pedestrian blocks AP1 and AP2 with their body. This results in attenuation and since the range is lower than 4m the outlier detection algorithm will not de-weight the range estimate and without outlier detection the algorithms will treat the ranges as reliable, even though they are NLOS due to the pedestrian's body blocking the signals. This threshold could be calibrated and varied to account for these scenarios, but this has not been explored in this thesis.

As shown in Table 4.14, the initial position RMSE was 3.11m, which is

approximately 5.4 times the initial position RMSE of Trial 1F. This will drastically reduce the accuracy of the results from all algorithms. The grid filter in Trial 2R also had an average position RMSE greater than 2m. In Figure 4.21, this is evident as, compared to the other algorithms, the path of the pedestrian is estimated at a different location. In Table 4.14, it can be seen that the initial position RMSE was greater than 3m, contributing to the overall positioning inaccuracy of the trial. For all algorithms, the larger positioning error at the final steps of Trial 2R is due to a poor orientation value at step 7. This poor orientation value could be caused by many things such as the low cost sensors exacerbating movements of the device. This poor orientation value resulted in an incorrect trajectory of the pedestrian. Poor orientation readings are to be expected and also impacted the performance of the algorithms in 3F and 3R. This demonstrates that improving the orientation sensor, orientation processing model or applying outlier detection to the orientation could improve the overall positioning accuracy of the algorithms.

In 33 out of 36 trials, the final position estimate was more accurate than the initial position estimate. This indicates that not only were the algorithms successful at maintaining the path of the pedestrian, they also improved the accuracy of the position estimate compared to the initial position estimate. With outlier detection included, the greatest absolute improvement was made by the particle filter in Trial 2F with an improvement of 1.32m and the greatest improvement without outlier detection was made by the particle filter in Trial 2R with an improvement of 1.23m. The greatest percentage improvement between the initial and final position estimates without outlier detection was made by the grid filter in Trial 2F with an improvement of 287%.

Trials 3F and 3R were the longest trials with a combination of different manoeuvres by the pedestrian. These trials consisted of 35 steps with the pedestrian also walking outside of the building. Figure 4.22 shows a sample run on the map when the device is moving through the forward scenario (Trial 3F). Figure 4.23 shows a sample run on the map when the device is moving through the reverse scenario

| | | Particle Filter + RSSI | Genetic Filter + RSSI | Grid Filter + RSSI |
|---|---|---|---|---|
| Trial 1F | Average Position RMSE percentage improvement of RSSI-based outlier detection against no outlier detection | 24% | -1% | 11% |
| Trial 1R | | 11% | 5% | -4% |
| Trial 2F | | 24% | 25% | 8% |
| Trial 2R | | 1% | -1% | 1% |
| Trial 3F | | 8% | 9% | 5% |
| Trial 3R | | 15% | 13% | 0% |
| Average | | 14% | 8% | 3% |

Table 4.15: RMSE percentage position error improvement for algorithms and trials in motion

(Trial 3R), beginning outside of the building. In the case of Trial 3F, the grid filter final position RMSE is almost double that of the other filters with an RMSE of 2.51m and 2.37m for the grid filter without and with RSSI-based outlier detection, respectively, and from Figure 4.22, the path is also visibly incorrect, unlike the other filters. However, in Trial 3R it appears that the grid filter better guides the PDR model through the path and most accurately represents the path of the user. The final position estimate for the grid filter in this trial was sub-metre both with and without outlier detection.

Table 4.15 shows the RMSE percentage improvement of the average positioning error for each trial for the algorithms with and without RSSI-based outlier detection. As can be seen in 15 out of 18 trials, the RSSI-based outlier detection improved the positioning solution. Any outlier detection will have a false positive rate, so these results are quite promising as the false positive rate does not result in a decrease in accuracy to the overall positioning solution. The largest improvement was with the genetic filter in Trial 2F where the outlier detection provided an improvement of 25%. The outlier detection provided the greatest improvement for the particle filter with an average improvement of 14%, whilst the outlier detection provided the least improvement to the grid filter with an average of 4%.

Using an EKF, Han et al. [53] was able to achieve a positioning error of 1.359m. Another paper by Sun et al. [54] achieved an RMSE of 1.1m when a form of

RSSI-based outlier detection was used. This improved over an EKF alone, which achieved an RMSE of 2.74m. Cao et al. [74] who also used a form of RSSI-based outlier detection achieved an accuracy of 1.082m. A caveat to these comparisons is that they were performed on different data, making the comparisons less meaningful. The range of results here combined with the findings in this thesis for filtering algorithms suggests that WiFi RTT when combined with outlier detection can reliably achieve sub-two-metre accuracy.

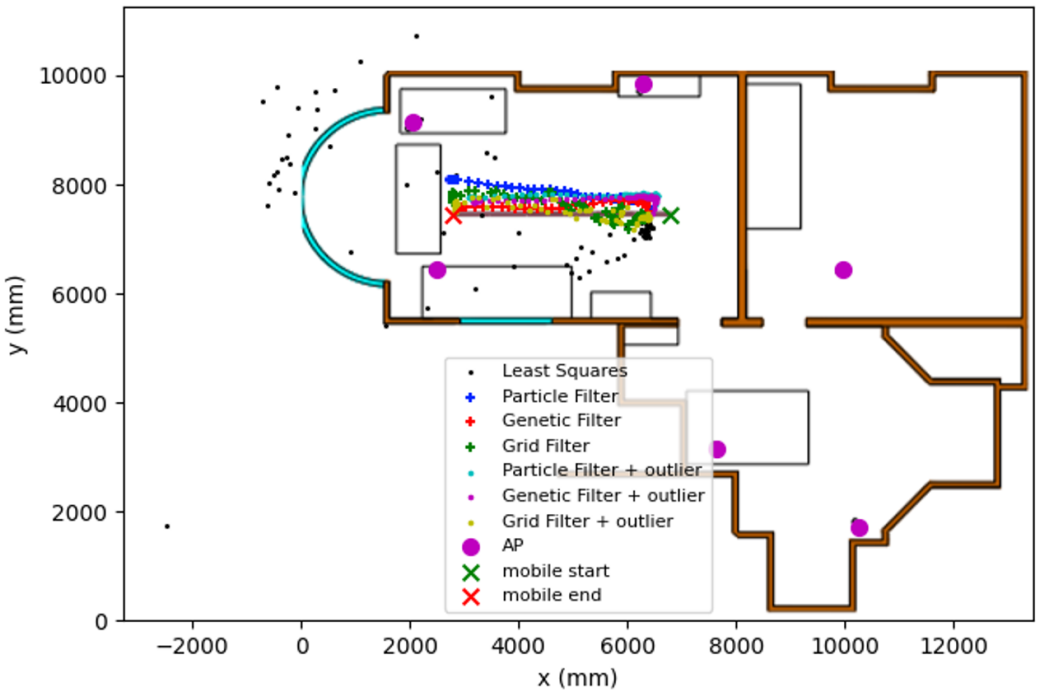


Figure 4.18: Trial 1F position per epoch

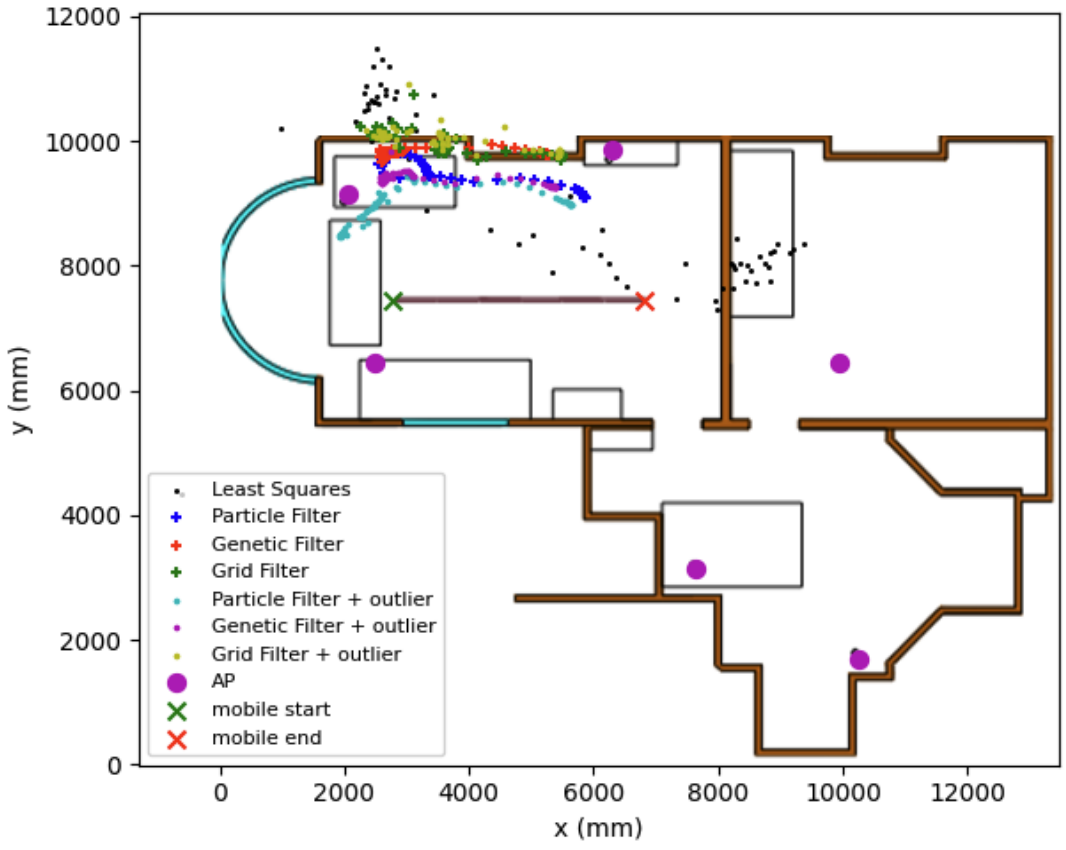


Figure 4.19: Trial 1R position per epoch

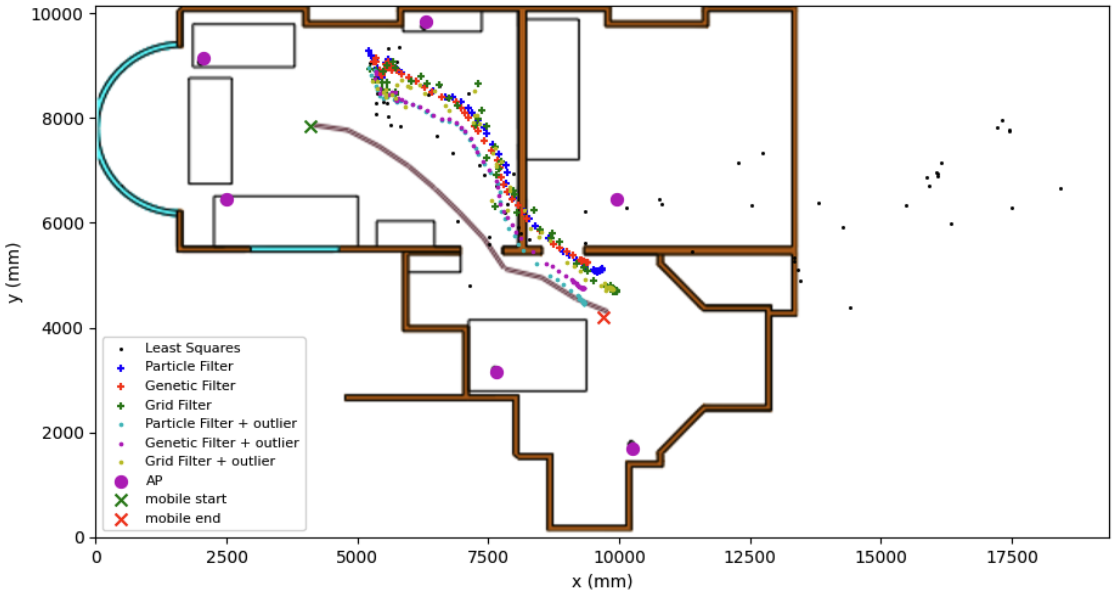


Figure 4.20: Trial 2F position per epoch

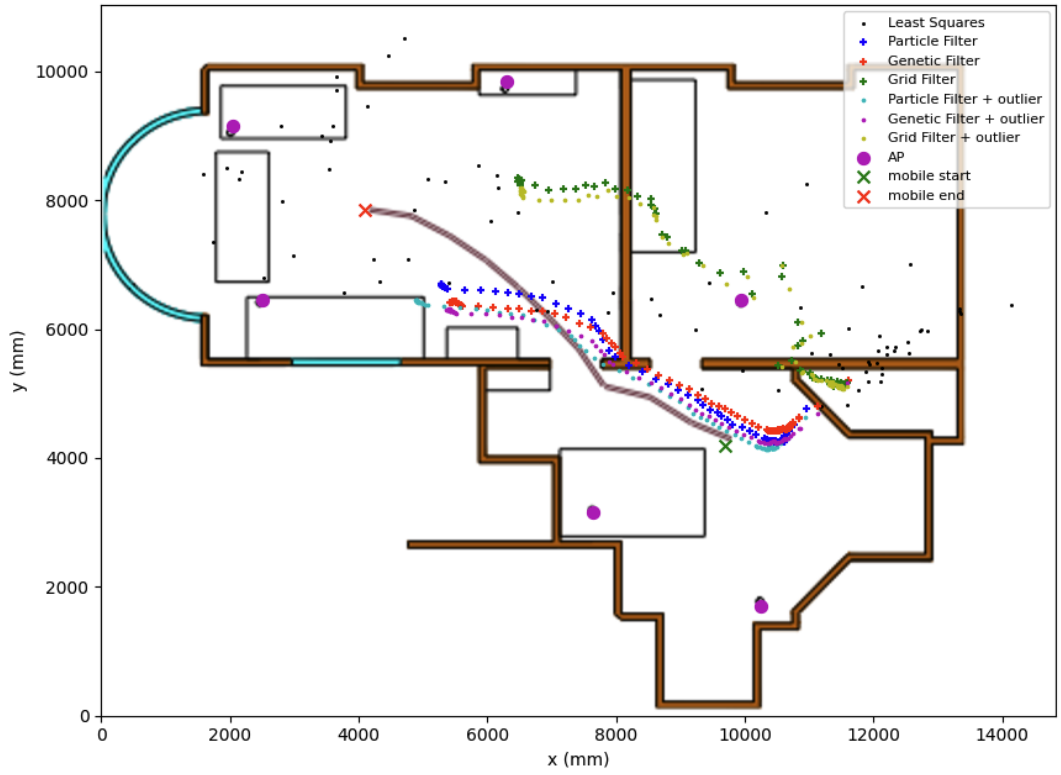


Figure 4.21: Trial 2R position per epoch

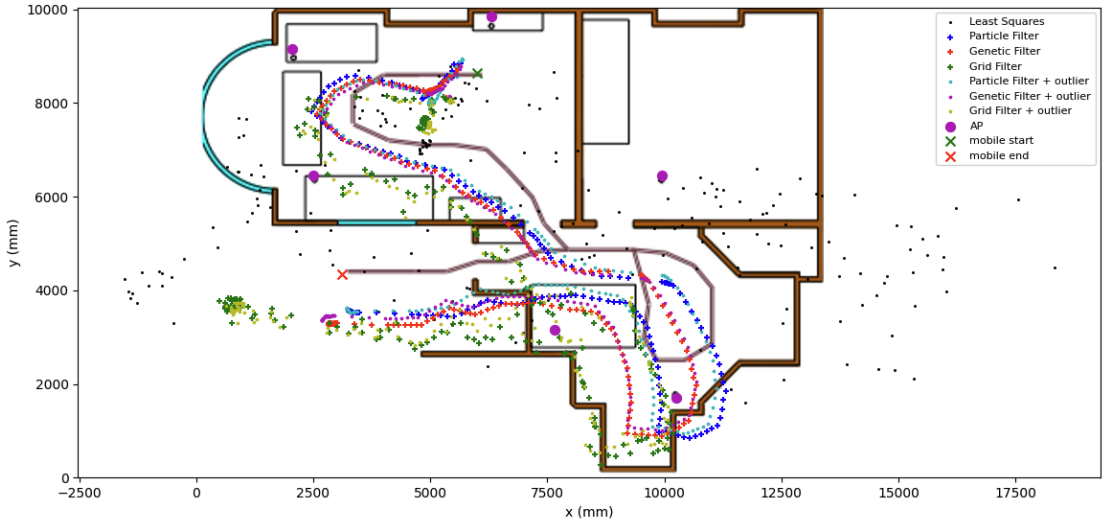


Figure 4.22: Trial 3F position per epoch

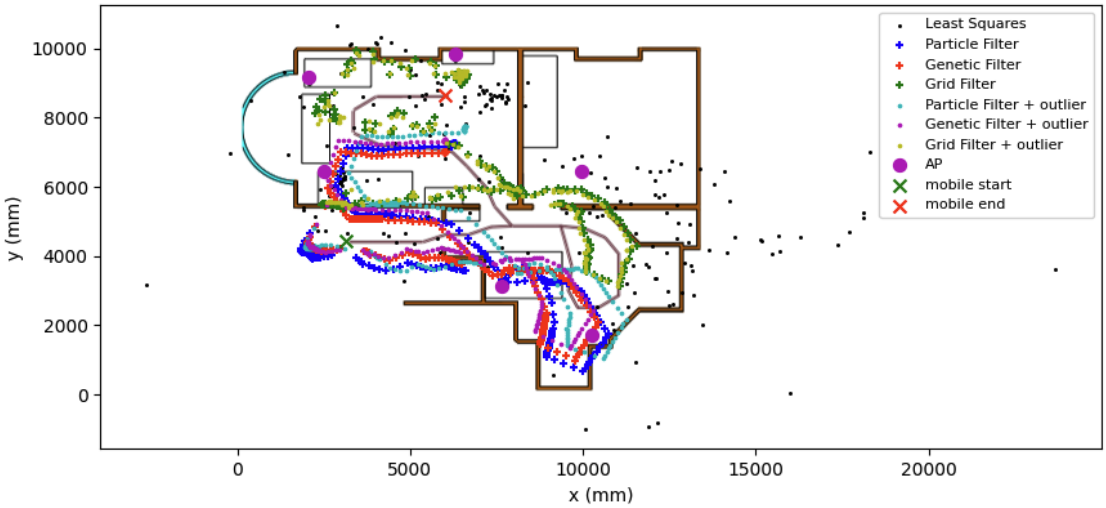


Figure 4.23: Trial 3R position per epoch

# CHAPTER 5

# SLAM

As mentioned in previous chapters of this thesis, an assumption of the majority of research concerning WiFi RTT and RSSI-fingerprinting with the notable exception of [62] is prior knowledge of the location of access points or the presence of a fingerprint map. The gathering of this information requires a lengthy and potentially costly survey step. Large technology companies such as Google, Apple, and Huawei are able to construct these RSSI maps and estimate the location of access points with the benefit of a large network of mobile devices and data [2]. With WiFi RTT there is an opportunity to improve on these models and also provide a more accessible way to estimate the location of access points and construct WiFi RTT and even RSSI fingerprint maps (if you were to walk to all parts of the grid of the environment) of indoor environments faster. This opportunity is best taken of advantage of by using simultaneous localisation and mapping (SLAM) techniques. These techniques allow for a mobile device moving through an environment to position itself as well as the location of landmarks simultaneously, essentially enabling indoor positioning with no prior knowledge of an environment. This chapter will explore WiFi RTT SLAM which will then extend to Posterity WiFi RTT SLAM which is an unexplored technique for WiFi RTT based positioning whereby SLAM maps of an environment can be re-used

for future datasets, hence the term “posterity”. These algorithms were presented at the ENC 2024 conference and the ION GNSS+ 2024 conference.

## 5.1 Simultaneous Localisation and Mapping

There are several types of SLAM algorithms, these include factor graph optimisation SLAM algorithms (GraphSLAM) and FastSLAM.

### 5.1.1 FastSLAM

Fast SLAM 2.0 [75] [76] conceptually follows a similar flow to a Particle filter. However, in addition to the particle filter of the mobile device state estimates; each landmark has their own particle filter state with their own state estimates. This means that the estimated position of the landmarks are actually variable, and the location estimate is determined as the mobile device moves through the environment. During the earlier epochs of the SLAM model, the algorithm is more akin to Odometry (the use of data from motion sensors to estimate change in position over time). Due to the uncertainty around the landmark locations, the algorithm relies more heavily on the motion sensors in the earlier epochs of the algorithm. Furthermore, by using particles, the mobile device and landmark states can scale more efficiently whilst dealing with a more complex non-linear error distributions and AP geometries. This is a general benefit of Monte-Carlo based filters. This means that FastSLAM is well suited for positioning during operation as the number of particles can be altered to accommodate for computational requirements. Admittedly, this will have an effect on positioning accuracy. The algorithm used in this paper follows the process shown in Figure 5.1. The particle filter follows the model described in Section 4.1.3.

1. The mobile device position and heading are assumed to be known. The landmark / access point coordinates are initialised based on a uniform distribution in the environment, where each access point will have its own particle filter. This initialisation occurs when each access point is observed.

2. The PDR model is applied during the prediction step. In this step the mobile device particles are moved according to the PDR model with noise distributed on a Gaussian distribution. The PDR follows the algorithm described in Section 4.1.7.

3. For the access point update step the particles of each access point are weighted against the distance between the particle and the estimated mean coordinates of the mobile device following equation 4.14.

4. The particles of the mobile device are weighted against their distance from each estimated access point position, the estimate is based on the mean coordinates of each access point particle filter.

5. The particles for both the mobile device and access points go through Sequential Importance Resampling (SIR) if the particle degeneracy limit is exceeded.

6. Finally, the position estimate of the mobile device and all access points is calculated using the weighted average of the particle positions.
Figure 5.2 shows a visual representation of the FastSLAM system. As the mobile device denoted by the "x" traverses the environment, the landmark particle filters are estimated, as shown by the coloured dots.

### 5.1.2 Factor Graph Optimisation SLAM

Factor Graph Optimisation SLAM or GraphSLAM [77] [78] is a popular form of SLAM where the problem is solved by using graphs consisting of nodes and edges. The node corresponds to the mobile device's states or landmark states, a different node is produced at each epoch. Every edge between any two nodes correspond to the constraints between the nodes. For example, the edge between mobile device state nodes would represent a step, whereas the edge between a mobile device state node and a landmark state node would represent a WiFi RTT measurement. The overall objective of GraphSLAM is to find a node configuration that minimises the

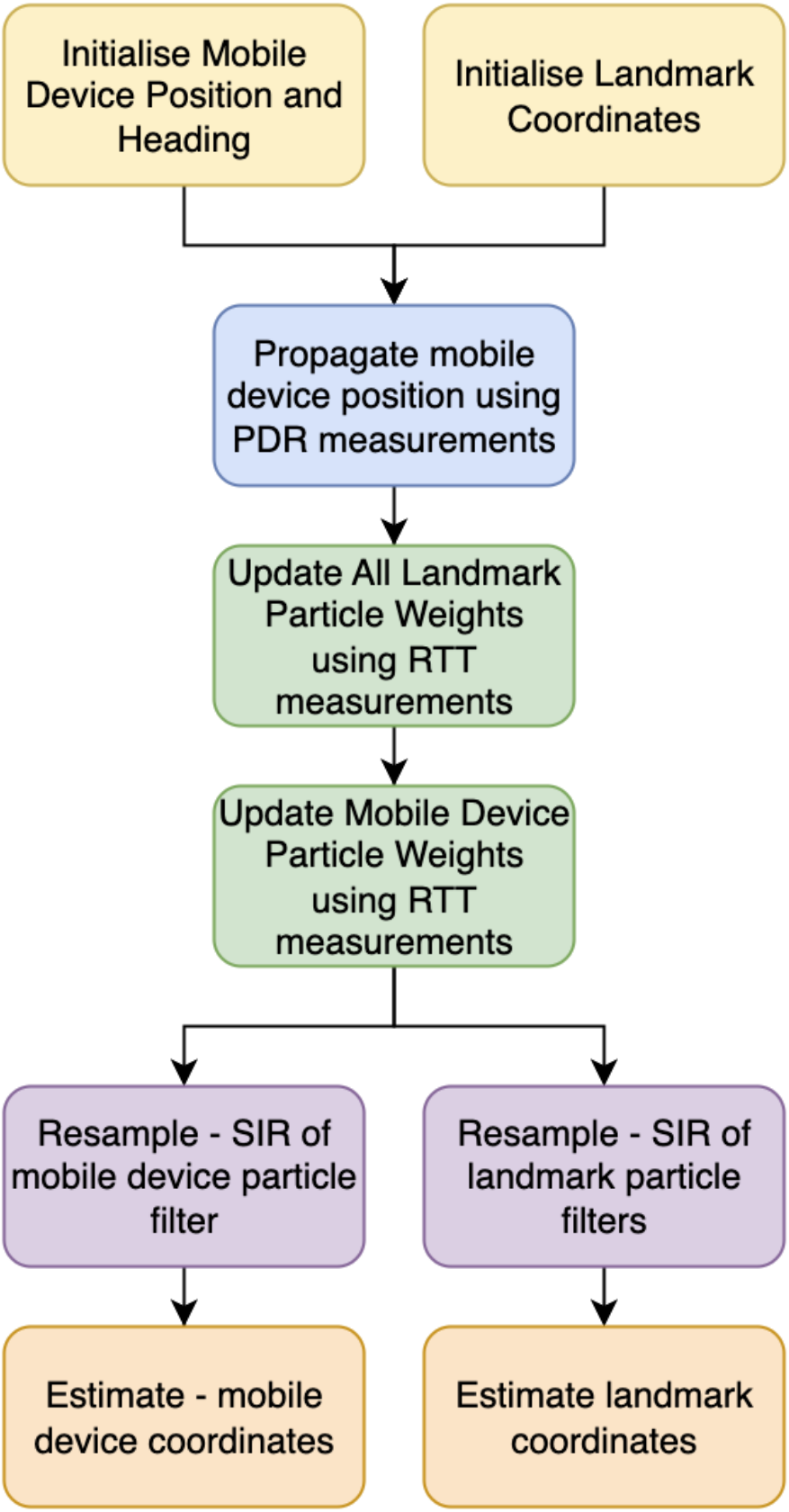


Figure 5.1: WiFi RTT FastSLAM algorithm

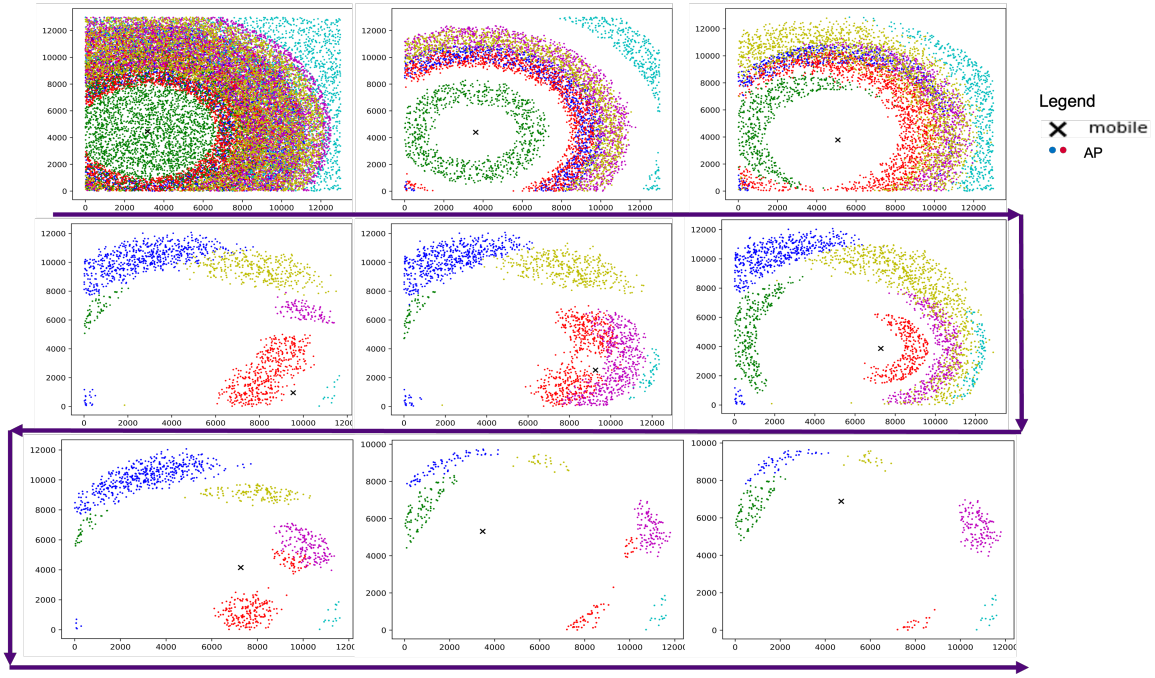


Figure 5.2: WiFi RTT FastSLAM visual representation

errors introduced by the constraints. The algorithm is split into two parts, graph construction and graph optimisation. The graph construction involves moving the mobile device through the environment and storing the node states and the edge constraints. The graph optimisation step which aims to find a node configuration that minimises the errors of each edge, this can use a Gauss-Newton optimisation algorithm [79] since the optimisation is essentially a least-squares problem.

A notable benefit of GraphSLAM is that loop closures can be used to improve the map as constraints (edges) can be generated between the nodes. Loop closure is when a device returns to a point it has previously observed. This means the map can be modified to optimise the edges that appeared due to the loop closure, meaning a more reliable SLAM map. FastSLAM does not benefit from this feature as it does not directly keep track of previous mobile device positions.

GraphSLAM is more computationally intense, making it less optimal for mobile devices with limited computational power. Furthermore, as the primary objective is navigation, the algorithm should be better at positioning the mobile device in real time without the need for a loop closure. For these reasons the SLAM algorithm

that will be used will be FastSLAM 2.0 as this is better suited for WiFi RTT error sources. The ability to alter the number of particles to account for different computational requirements and selectively update landmarks when they are observed allows the algorithm to be more flexible for available computational power.

### 5.1.3 Posterity SLAM

The posterity SLAM algorithm is similar to the regular FastSLAM 2.0 algorithm with one key difference. During the initialisation step of the landmark particles instead of using a uniform distribution throughout the environment; each landmark's particles are initialised with a Gaussian distribution around the final landmark coordinates of the previous SLAM trial. The standard deviation of each access point particle filter of the previous trial is taken to initialise the standard deviation of the Gaussian distribution of the access point particle filters for the new trial. This essentially allows each SLAM trial to benefit from previous SLAM trials improving the overall positioning solution over time whilst being aware of potential uncertainty in those estimates. Posterity SLAM is a form of cooperative SLAM as it takes advantage of multiple cooperative mobile devices to map a given environment. Allowing multiple mobile devices to construct a shared map simultaneously using their own paths, cooperative SLAM, is scope for future work.

## 5.2 Experimental Methodology

The equipment and layout used was the same as in Table 4.6. The methodology for testing the SLAM algorithm involved moving through a route both forward and in reverse in an environment. These routes were designed to enable NLOS signal reception to occur. The routes and environment explored are shown in Figure 5.3. The trials involved a pedestrian holding the smartphone and walking on top of fixed step markers placed on the ground along the route of the trial. The markers were approximately 670mm apart, the location of each marker was measured against the reference points in the environment to generate the step marker's ground truth coordinates. In order to align the ground truth data with the measured data, the

| | |
|---|---|
| Number of Particles in the mobile device particle filter | 50000 |
| Number of Particles in each landmark particle filter | 10000 |
| Initial Position Error | 0.1m |
| Initial Heading Error | 0.03 radians |
| Mikov constant | 0.6 |
| Sensor Heading Error | 0.3 radians |
| Step Length Error | 0.1m |
| Landmark Measurement Error | 2m |
| Mobile Device Measurement Error | 1m |

Table 5.1: Configuration for the SLAM algorithm

trials were filmed. The timestamps of the steps from the video were used to determine the expected position of a device at a given time, allowing the analysis to have an accurate comparison point at each step.

The standard deviation of the step length assuming white noise, used to determine the noise applied during the prediction step was calibrated as follows. A pedestrian carrying the mobile device walks 20 steps in a straight line 10 times, the real distance travelled is measured and the estimated distance travelled is calculated using the PDR model, the standard deviation of the difference is computed and this is divided by the number of steps to give the standard deviation of a step. All data was collected simultaneously using a custom mobile app. RTT and RSSI measurements were received at 100ms intervals and IMU data was collected at approximately 20ms intervals. RSSI-based outlier detection is implemented as described in Section 4.1.6 ”RSSI-based outlier detection for filtering”. The locations of the access points were also calculated and these were used to test the effectiveness of the SLAM algorithm landmark predictions.

For posterity SLAM, the same data as the SLAM algorithm was used. For the second SLAM trial the landmark position estimates of another trial were used to provide an initial position estimate of the landmark positions. For this set of experiments the parameters were set up as shown in Table 5.1.

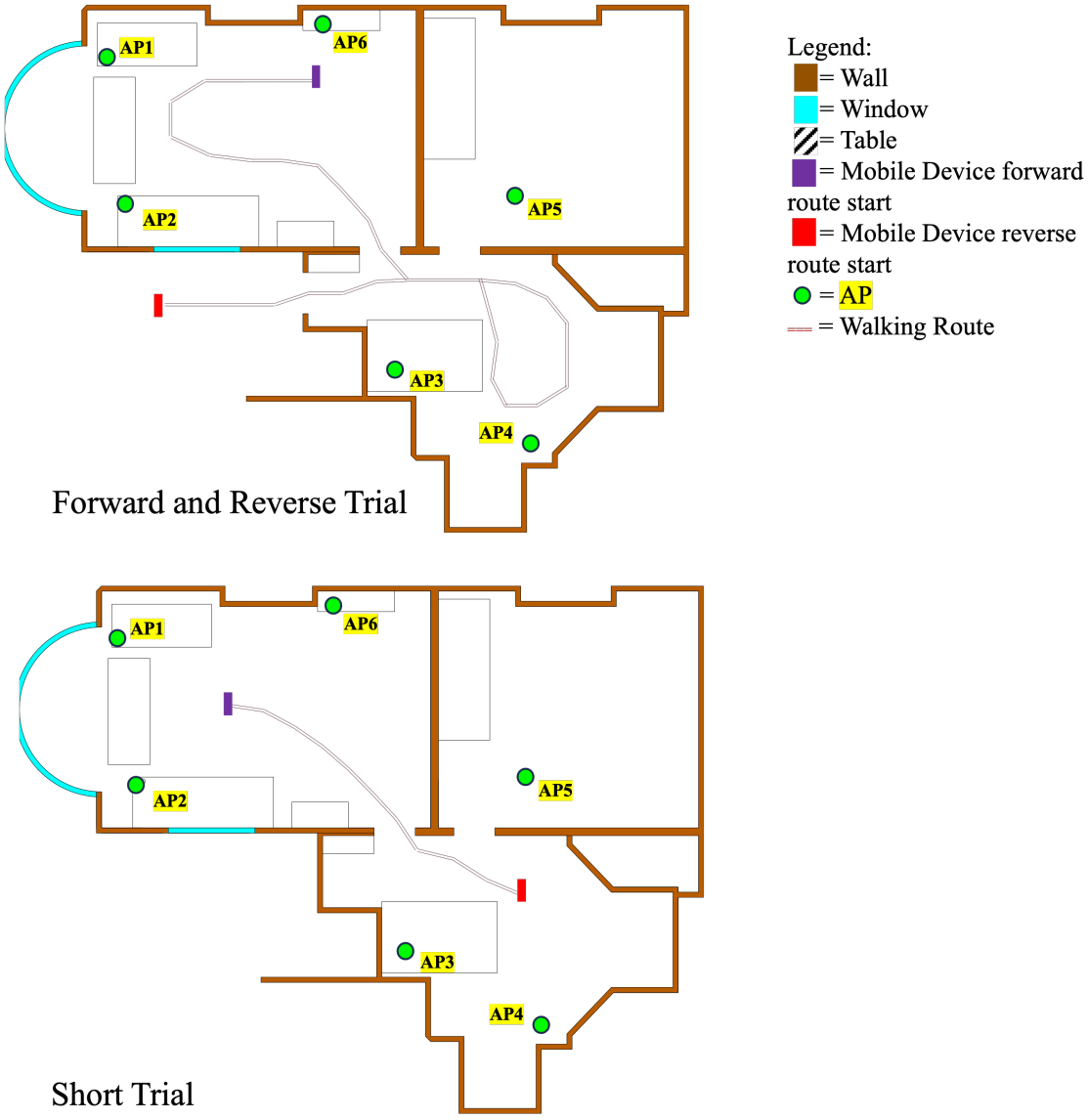


Figure 5.3: SLAM paths for SLAM and Posterity SLAM algorithms

| Trial | Forward | Reverse |
|---|---|---|
| **Maximum Horizontal Error (m)** | 3.2 | 2.73 |
| **Mean Horizontal Error (m)** | 1.09 | 1.91 |
| **Standard Deviation (m)** | 0.69 | 0.59 |
| **Median Horizontal Error (m)** | 1.00 | 2.03 |
| **Final Horizontal Error (m)** | 1.01 | 1.7 |

Table 5.2: Statistics for the forward and reverse trials for the SLAM algorithm

## 5.3 Results and Discussion

This section will first evaluate the accuracy of the SLAM algorithm, including the mobile device position estimates and the landmark position estimates. The posterity SLAM algorithm will then be explored, looking at the horizontal error of the mobile device position estimates and the landmark position estimates when using the landmark position estimates from the first trial to initialise the landmark positions for the second trial.

### 5.3.1 WiFi RTT SLAM

In Table 5.2, the statistics of the performance of the algorithm in the forward and reverse trials can be seen. Overall the results are promising with the mean, median, and final position horizontal error for the forward trial being about a metre. The maximum horizontal error and standard deviation for the trial are 3.2m and 0.69m respectively. In Figure 5.4 and Figure 5.5 during the loop section of the trial in the bottom right of the environment, the position horizontal error increases by more than double. The algorithm then recovers the positioning solution to sub-metre accuracy, ending the trial at 1.01m.

In the reverse scenario, where the mobile device begins outside and ends inside, the results are slightly worse. Firstly, in Figure 5.6, whilst the rough shape of the predicted path of the pedestrian matches the true path, the entire path is translated down the y-axis by around 2m. The final horizontal position error is 1.7m with an

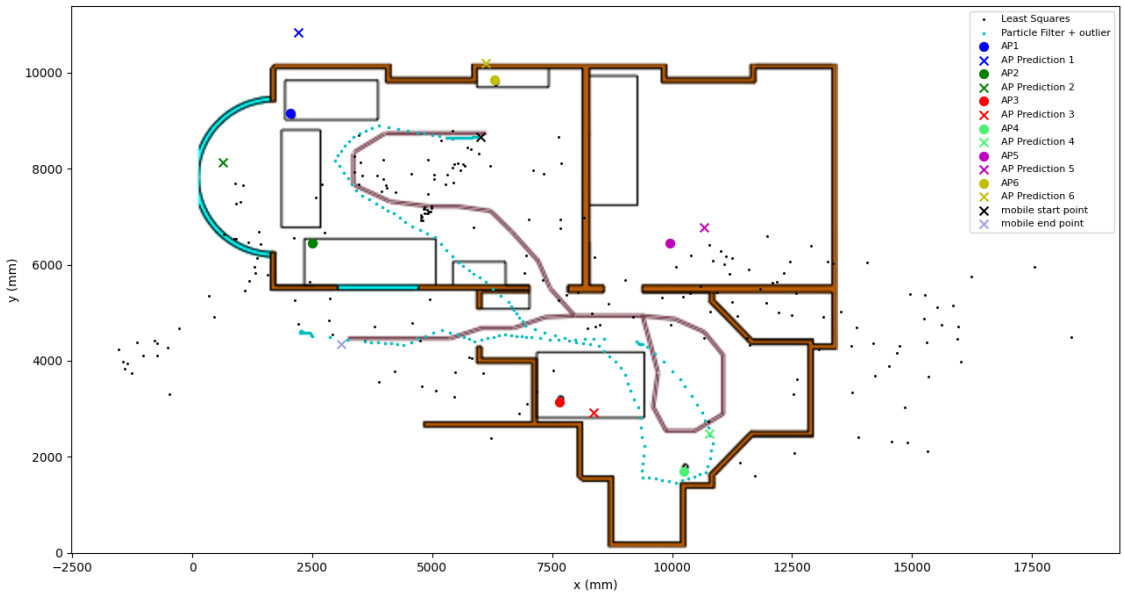


Figure 5.4: Graph showing the performance of the SLAM algorithm in the forward trial

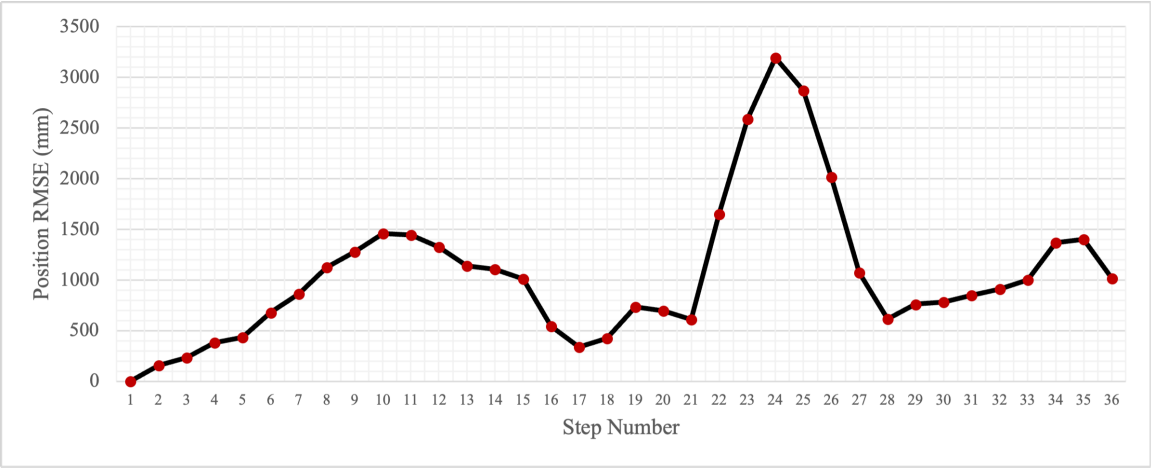


Figure 5.5: Mobile device position error per step for the forward trial

average and median horizontal error of 1.91m and 2.03m respectively. This is nearly a metre worse than the forward trial. In the reverse scenario the mobile device has no line of sight with any AP at the start of the trial, whereas in the forward trial the mobile device has a direct line of sight with 3 APs at the start of the trial. The lack of a line of sight signal will drastically reduce the reliability of the signals and thus the overall positioning solution will be degraded with fewer line of sight APs. The NLOS signal reception will also have an impact on the predictions of the landmark coordinates as in the earlier stages of the algorithm, the distance between each mobile device and the AP will be less reliable. Therefore, the starting layout of APs and the number of LOS signals plays a major role in the accuracy of the positioning solution.

When we compare these results with previous WiFi SLAM research, the results are promising. In Gentner and Avram's paper [62] they achieved an average positioning error under 1m, which is better than what was achieved in this thesis. However, it is very important to note that that paper did not incorporate a pedestrian dead reckoning model and assumed knowledge of the user's location along known markers on the user's path. Ferris et al. [58] and Liu et al. [60] achieved an average position error of 3.79m and 4.76m, respectively, when using WiFi RSSI and a pedestrian dead reckoning model alone. In Liu et al. [60], when they integrated visual inertial odometry within the PDR model, they were able to achieve sub-metre accuracy. This indicates that WiFi RTT is a promising solution on its own, as its performance was better than WiFi RSSI and has potential to improve even further when integrated with more advanced sensors such as visual sensors.

The final estimated landmark coordinates shown in Table 5.3 are important to analyse as they will demonstrate the overall effectiveness of the SLAM algorithm. At the end of the forward trial, AP6 reached a horizontal error of just 0.16m and the maximum horizontal error of 2.31m was seen on AP2. This can be visualised in Figures 5.4 and 5.5. This demonstrates that the algorithm was able to identify the location of 4 landmarks with sub-metre accuracy during this trial. In the forward

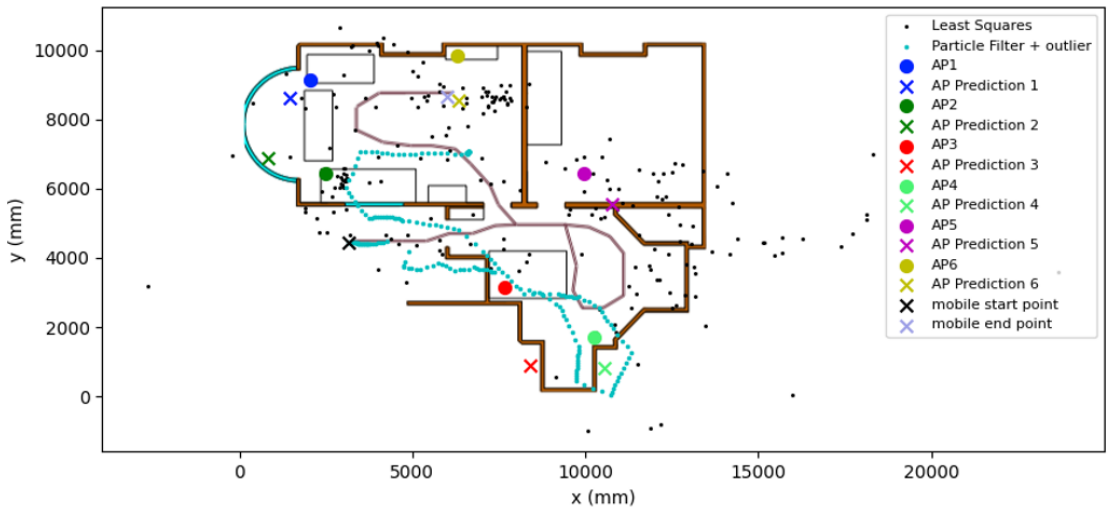


Figure 5.6: Graph showing the performance of the SLAM algorithm in the reverse trial, AP predictions represent the final AP position estimates

| Landmark | Forward | Reverse |
|---|---|---|
| AP1 Final Position Horizontal Error (m) | 1.59 | 0.78 |
| AP2 Final Position Horizontal Error (m) | 2.31 | 1.73 |
| AP3 Final Position Horizontal Error (m) | 0.58 | 2.4 |
| AP4 Final Position Horizontal Error (m) | 0.59 | 0.94 |
| AP5 Final Position Horizontal Error (m) | 0.61 | 1.21 |
| AP6 Final Position Horizontal Error (m) | 0.16 | 1.32 |

Table 5.3: Statistics for the landmark position estimates

trial, as can be seen in Figure 5.4, the path of the device roughly matches the shape of the true path of the pedestrian. For the reverse trial final landmark horizontal position error, the average AP final position horizontal error is 1.4m, with the lowest position horizontal error being 0.78m and the highest position horizontal error being 2.4m. Interestingly, in the forward scenario, AP3 had a horizontal error of 0.58m and the same AP had a horizontal error of 2.4m in the reverse. This can be visualised in Figures 5.6 and 5.7.

The SLAM algorithm is able to determine the location of the landmarks to a degree of accuracy below 2m 83% of the time and sub-metre 50% of the time. It was observed that when the starting position of the mobile device had access to line of sight signals

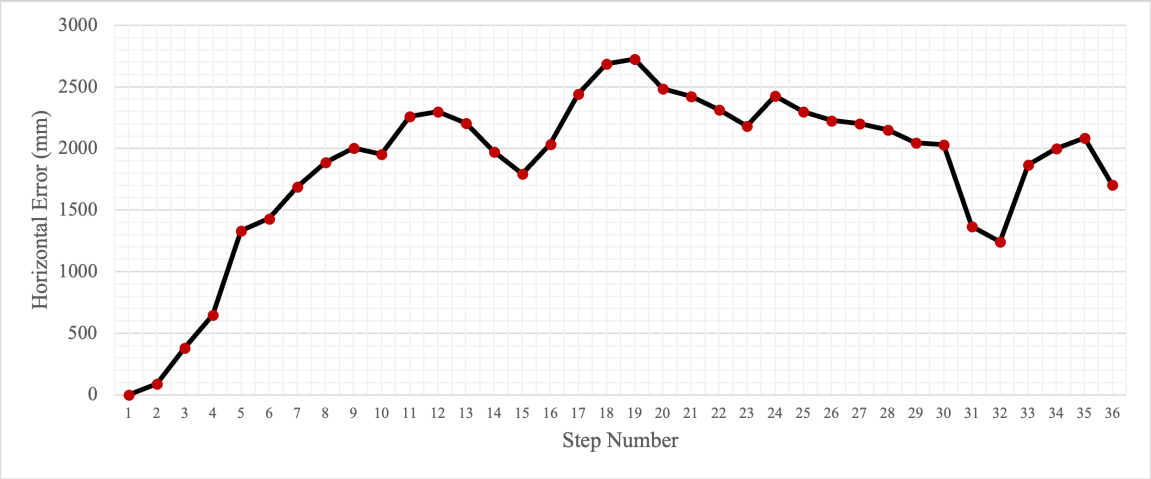


Figure 5.7: Position Horizontal Error per step for the reverse trial

the location of the landmarks could be determined to sub-metre accuracy 67% of the time. With NLOS signals, the accuracy declined, which is expected as it impacts both the location estimates of the mobile device and the access points.

### 5.3.2 WiFi RTT Posterity SLAM

This section will explore the use of SLAM-determined landmark coordinates to initialise the landmark coordinates for new SLAM trials. All combinations of the forward, reverse and short trials are shown, these include: forward using reverse, reverse using forward, forward using short, short using forward, reverse using short and short using reverse. For example, with forward using reverse, the trial is the forward trial using the landmark position estimates from the reverse trial.

The positioning metrics of the forward using reverse and reverse using forward trials can be seen in Table 5.4 alongside the trials on their own. Posterity SLAM improved the final horizontal error of the mobile device from 1.55m for the reverse trial to 0.8m from 1.55m for the reverse using forward trial, providing sub-metre accuracy to the positioning solution. The mean horizontal error improved from 1.78m to 1.14m, whilst the median improved from 1.96m to 1.05m. However, in the case of forward using reverse whilst the final horizontal error was lower by 0.04m the maximum horizontal error increased by 0.62m and the mean horizontal error increased to 1.06m from 0.92m. The change in horizontal error per step for the forward path

| | Forward Trial | Reverse Trial | Reverse Using Forward | Forward Using Reverse |
|---|---|---|---|---|
| Maximum Horizontal Error (m) | 2.76 | 2.45 | 2.47 | 3.38 |
| Mean Horizontal Error (m) | 0.92 | 1.78 | 1.14 | 1.06 |
| Standard Deviation (m) | 0.58 | 0.56 | 0.57 | 0.86 |
| Median Horizontal Error (m) | 0.87 | 1.96 | 1.05 | 1.06 |
| Final Horizontal Error (m) | 0.89 | 1.55 | 0.80 | 0.85 |

Table 5.4: Statistics for the mobile device position estimates for the Forward and Reverse Trial


Figure 5.8: Forward Path posterity SLAM horizontal error per step

trials can be seen in Figure 5.8. The forward using reverse trial begins well, but its position estimates degrade during the loop section of the path. The change in horizontal error per step for the reverse trials can be seen in Figure 5.9. Posterity SLAM demonstrates that the mobile device can be more consistently accurately positioned than regular SLAM.

The final horizontal position error for all APs for the forward-reverse trials are shown in Table 5.5. The final horizontal position error of the landmarks for the reverse using forward trial improved for 4 out of 6 access points when compared to the reverse trial. All landmarks were positioned to sub two metre accuracy. The

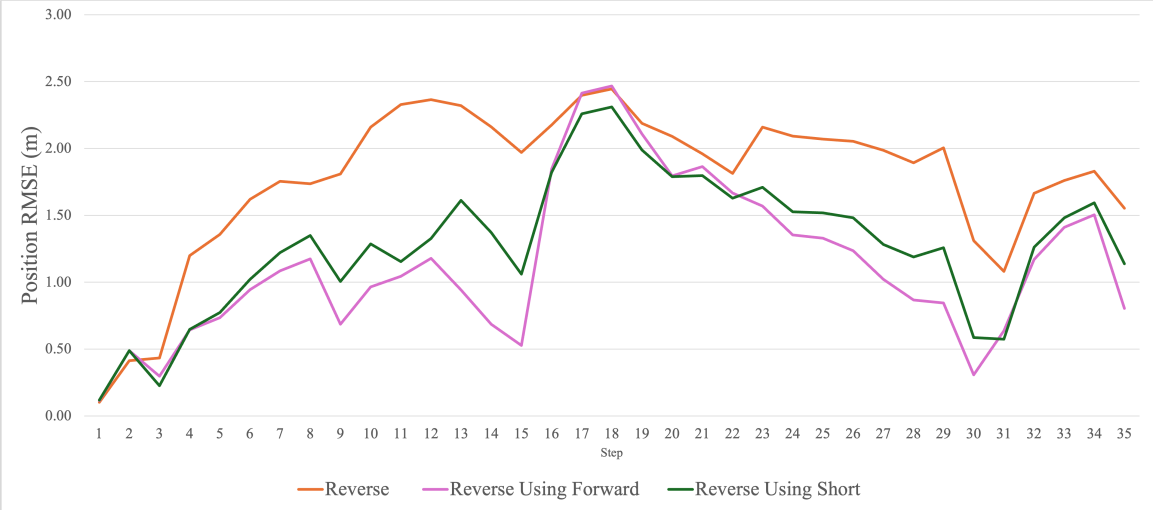


Figure 5.9: Reverse Path posterity SLAM horizontal error per step

| | Forward Trial | Reverse Trial | Forward Using Reverse | Reverse Using Forward |
|---|---|---|---|---|
| AP1 Final Horizontal Position Error (m) | 1.43 | 1.44 | 1.11 | 0.55 |
| AP2 Final Horizontal Position Error (m) | 2.43 | 1.83 | 1.51 | 2.0 |
| AP3 Final Horizontal Position Error (m) | 0.84 | 2.69 | 2.11 | 1.60 |
| AP4 Final Horizontal Position Error (m) | 0.79 | 0.97 | 0.83 | 1.1 |
| AP5 Final Horizontal Position Error (m) | 0.76 | 0.95 | 1.06 | 0.75 |
| AP6 Final Horizontal Position Error (m) | 0.57 | 1.3 | 0.93 | 0.25 |

Table 5.5: Statistics for the landmark position estimates for the Forward and Reverse Trial

most substantial improvement was in AP3 which improved from 2.69m to 1.6m. Comparing the AP horizontal errors of the reverse using forward trial to the forward trial on their own shows that 4 out of 6 AP positions improved. Comparing the forward and reverse using forward data, the best improvement was 0.88m on AP1. The largest decrease in accuracy was on AP3 with a decrease of 0.76m. However, this is in contrast to a 1.09m improvement when comparing the reverse using forward to the reverse trial alone. This is somewhat undesirable. An ideal outcome would be able to take the best landmark position estimates of both datasets.

In the forward using reverse data also shown in Table 5.4, the landmark horizontal error only improved for 2 APs when compared to the forward trial alone. The greatest improvement being 0.92m on AP2 and the greatest decrease in accuracy being 1.27m on AP3. Comparing the forward using reverse trial to the reverse trial alone showed that posterity SLAM improved the horizontal error for 5 out of 6 APs. The mean horizontal error across all APs were 1.06m, 1.26m, 1.14m and 1.53m for the reverse using forward, forward using reverse, forward and reverse trials respectively. This demonstrates that posterity SLAM can improve the average AP horizontal error. This can be seen in Figures 5.10 and 5.11 which shows the AP position error per epoch for the reverse using forward trial and the forward using reverse trial respectively. Posterity SLAM allows the second trial to have a better starting point for the landmark estimates, enabling a better set of landmark position estimates. There is an opportunity for the most accurate landmark estimates to be picked over time as more trials occur, thus improving the overall positioning solution. This opportunity has not been explored in this thesis and represents scope for future work.

Finally, Figure 5.6 shows the path and landmark estimates using regular SLAM for the reverse path and Figure 5.12 shows the reverse using forward and landmark estimates using posterity SLAM. As can be seen the landmark position estimates have benefited from the landmark estimates of the forward trial and this has in turn made the estimated pedestrian path more closely follow the ground truth.

The Short Trial dataset was used to demonstrate the poor performance of the algorithm on shorter paths with fewer turns, more specifically poor performance for the AP positions. The reason that a shorter path would have poorer performance for determining landmark positions is that the algorithms have less time to allow the landmark particle filters to converge to a strong landmark position estimate, so essentially the algorithm relies heavily on Odometry for the positioning solution. Furthermore, it is possible that observing landmarks from a range of different directions also has an impact on the landmark positioning accuracy. Theoretically,

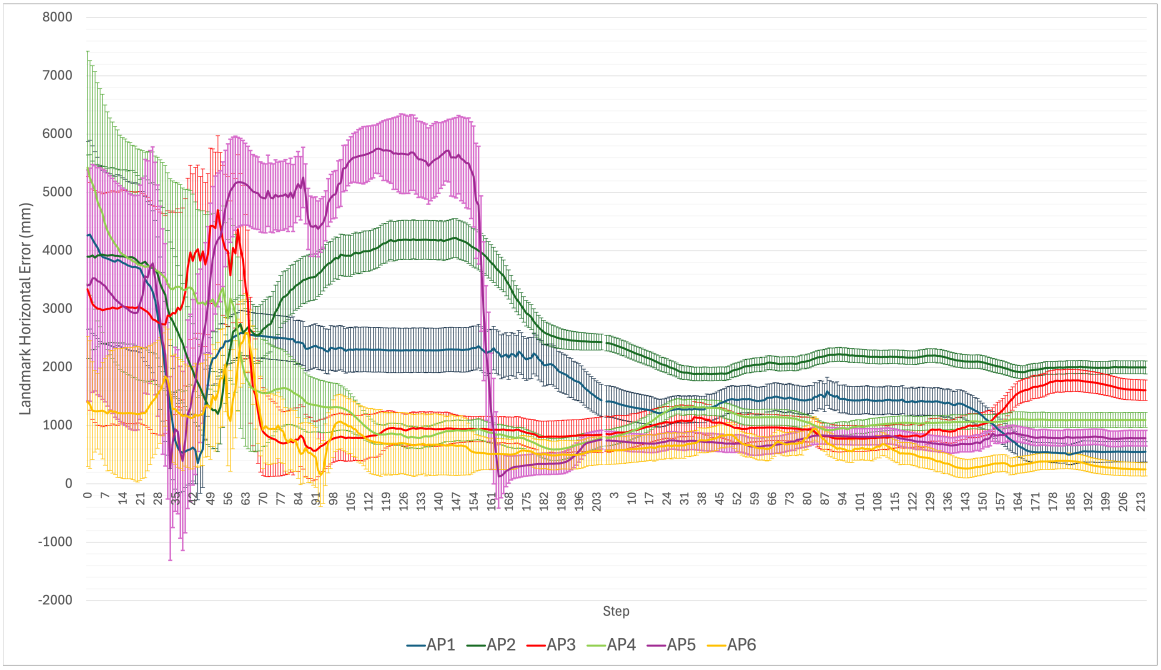


Figure 5.10: Forward then Reverse AP position error per step. The white cut-off in the centre of the chart represents the switch from the forward trial to the reverse trial. The error bars represent the standard deviation of each landmark particle filter

this scenario is where posterity SLAM can provide a significant improvement to the positioning solution as more reliable landmark position estimates can be used.

The mobile device positioning metrics for the short and forward trials are shown in Table 5.6 and the metrics for the short and reverse trials are shown in Table 5.7. The Short trial final positioning error was 1.42m. Using posterity SLAM, this improved to 0.82m when the forward trial landmark position estimates were used and 0.96m when the reverse trial landmark position estimates were used. The improvement can be seen by comparing Figure 5.13 which shows the short path without posterity SLAM and Figure 5.14 which shows the short path when the landmark position estimates of the forward trial are used to initialise the short SLAM trial.

The AP final horizontal position error for the short and reverse trial combinations can be seen in Table 5.8. The short trial landmark position errors are on average 2.78m with a maximum horizontal error of 5.57m for AP2. 1 out of 6 APs were

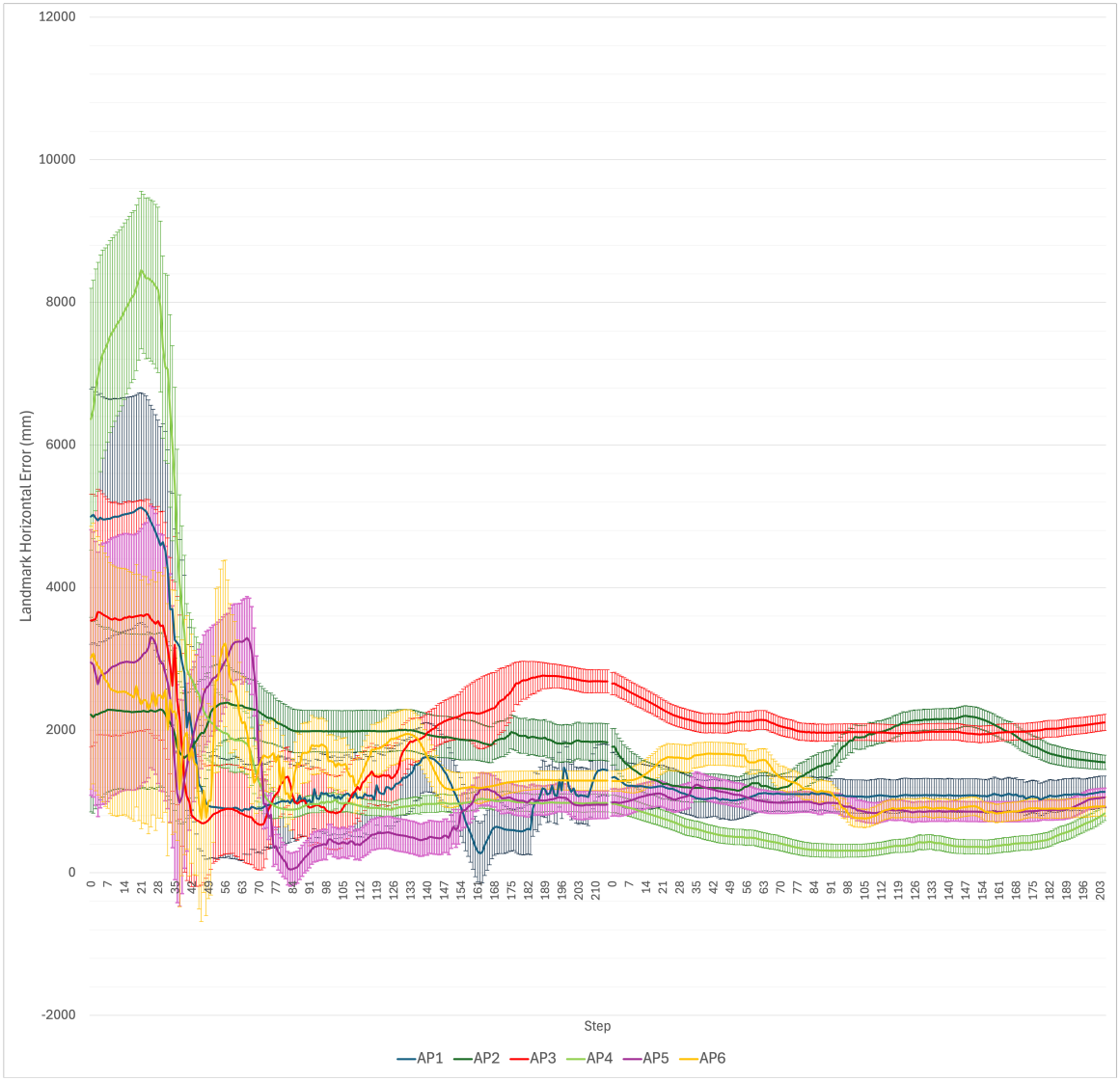


Figure 5.11: Reverse then Forward AP position error per step. The white cut-off in the centre of the chart represents the switch from the reverse trial to the forward trial. The error bars represent the standard deviation of each landmark particle filter

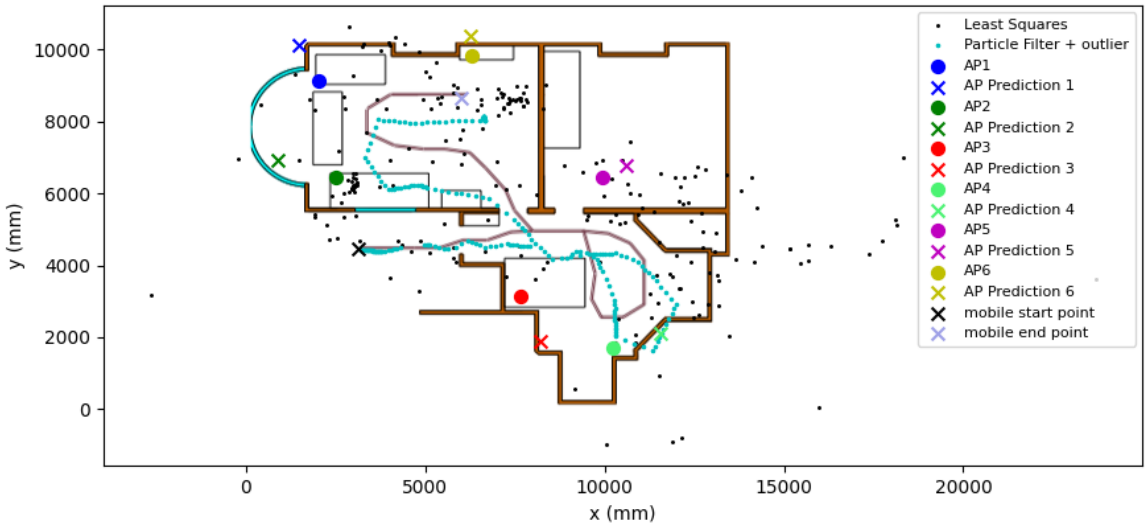


Figure 5.12: Reverse Trial using Forward Trial Path posterity SLAM position estimate and landmark estimates

| | Forward Trial | Short Trial | Short Using Forward | Forward Using Short |
|---|---|---|---|---|
| Maximum Horizontal Error (m) | 2.76 | 1.42 | 0.84 | 2.77 |
| Mean Horizontal Error (m) | 0.92 | 0.42 | 0.40 | 0.87 |
| Standard Deviation (m) | 0.58 | 0.41 | 0.26 | 0.60 |
| Median Horizontal Error (m) | 0.87 | 0.24 | 0.37 | 0.75 |
| Final Horizontal Error (m) | 0.89 | 1.42 | 0.84 | 0.77 |

Table 5.6: Statistics for the mobile device position estimates for the Short and Forward Trials

| | Reverse Trial | Short Trial | Short Using Reverse | Reverse Using Short |
|---|---|---|---|---|
| Maximum Horizontal Error (m) | 2.45 | 1.42 | 0.96 | 2.31 |
| Mean Horizontal Error (m) | 1.78 | 0.42 | 0.49 | 1.28 |
| Standard Deviation (m) | 0.56 | 0.41 | 0.33 | 0.51 |
| Median Horizontal Error (m) | 1.96 | 0.24 | 0.43 | 1.29 |
| Final Horizontal Error (m) | 1.55 | 1.42 | 0.96 | 1.14 |

Table 5.7: Statistics for the mobile device position estimates for the Short and Reverse Trials

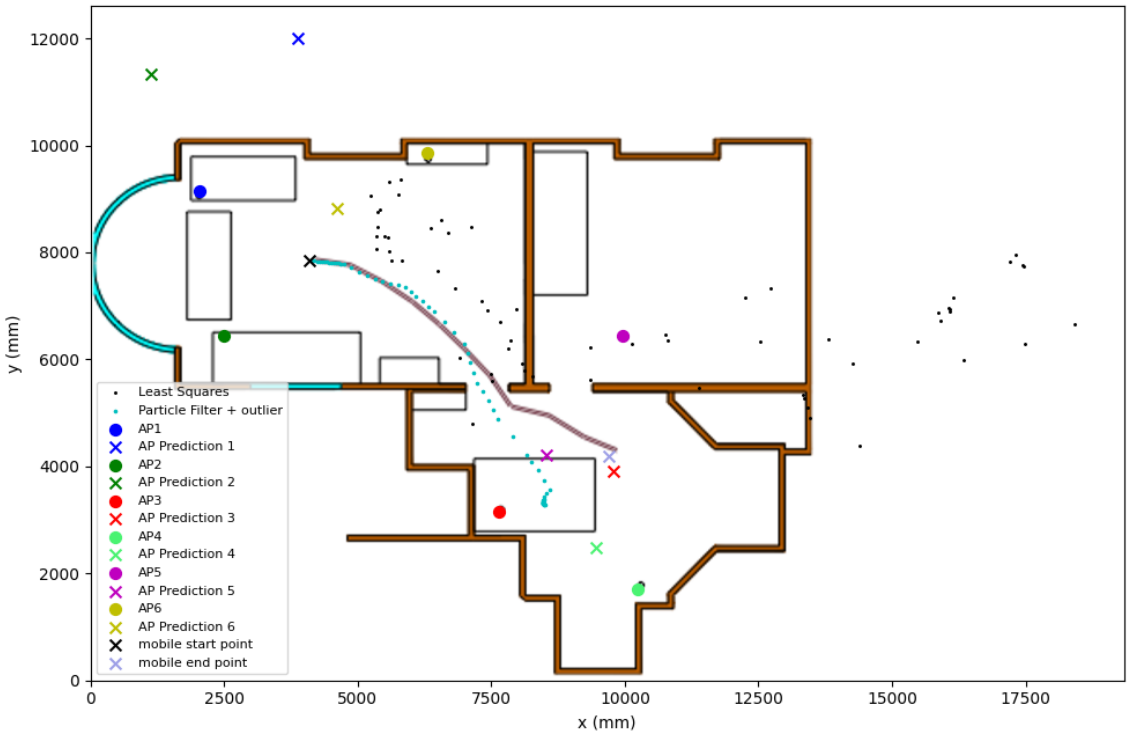


Figure 5.13: Short Path regular SLAM position estimate and landmark estimates

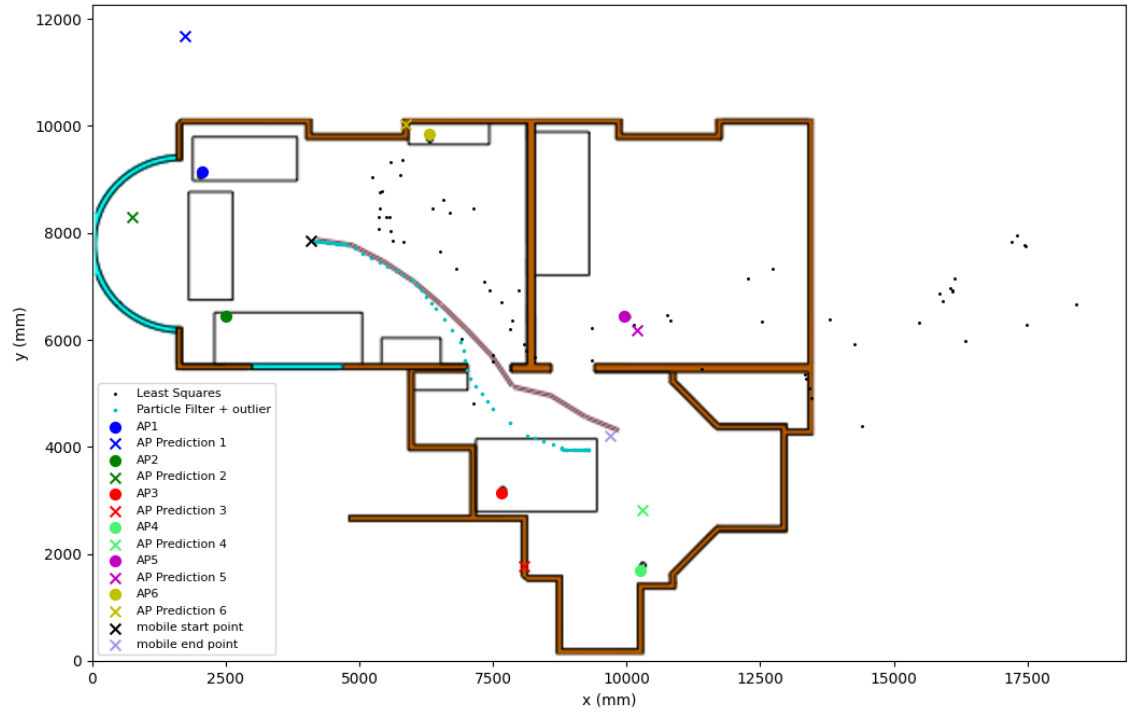


Figure 5.14: Short using Forward Path posterity SLAM position estimate and landmark estimates

| | Reverse Trial | Short Trial | Short Using Reverse | Reverse Using Short |
|---|---|---|---|---|
| AP1 Final Horizontal Position Error (m) | 1.22 | 2.78 | 1.57 | 0.68 |
| AP2 Final Horizontal Position Error (m) | 2.15 | 5.57 | 1.51 | 1.84 |
| AP3 Final Horizontal Position Error (m) | 2.43 | 2.36 | 1.88 | 1.98 |
| AP4 Final Horizontal Position Error (m) | 0.34 | 1.29 | 0.17 | 0.80 |
| AP5 Final Horizontal Position Error (m) | 0.91 | 2.66 | 0.60 | 0.66 |
| AP6 Final Horizontal Position Error (m) | 1.18 | 2.14 | 0.52 | 0.62 |

Table 5.8: Statistics for the landmark position estimates for the Short and Reverse Trial

positioned to sub two metre accuracy. When posterity SLAM using the reverse trial is incorporated, the errors improve to 4 out of 6 being sub-metre with all 6 AP positions improving. In the case of the short using reverse trial, in 5 out of 6 trials the landmark position accuracy improved compared to the reverse trial alone. Additionally, the landmark position estimates improved for 4 out of 6 trials for the reverse trial when the short landmark estimates were used when compared to the reverse trial alone. Despite the poor performance of the short trial, it was still useful in providing a more accurate set of landmark position errors for the reverse trial. The final mobile device horizontal error for the reverse using short trial improved to 1.14m from 1.55m for the reverse trial. This trend of improvement in AP position accuracy per step can be seen in Figures 5.15 and 5.16 which show the short using reverse and reverse using short trials respectively.

The trend of results for the landmark position estimates is mostly similar for the short and forward trial combinations shown in Table 5.9. The short trial landmark position error improved for 6 out of 6 APs when the forward trial landmarks were used for initialisation. However, when the short trial landmark estimates were used for the forward trial, 5 out of 6 AP position errors increased when compared to the forward trial alone.

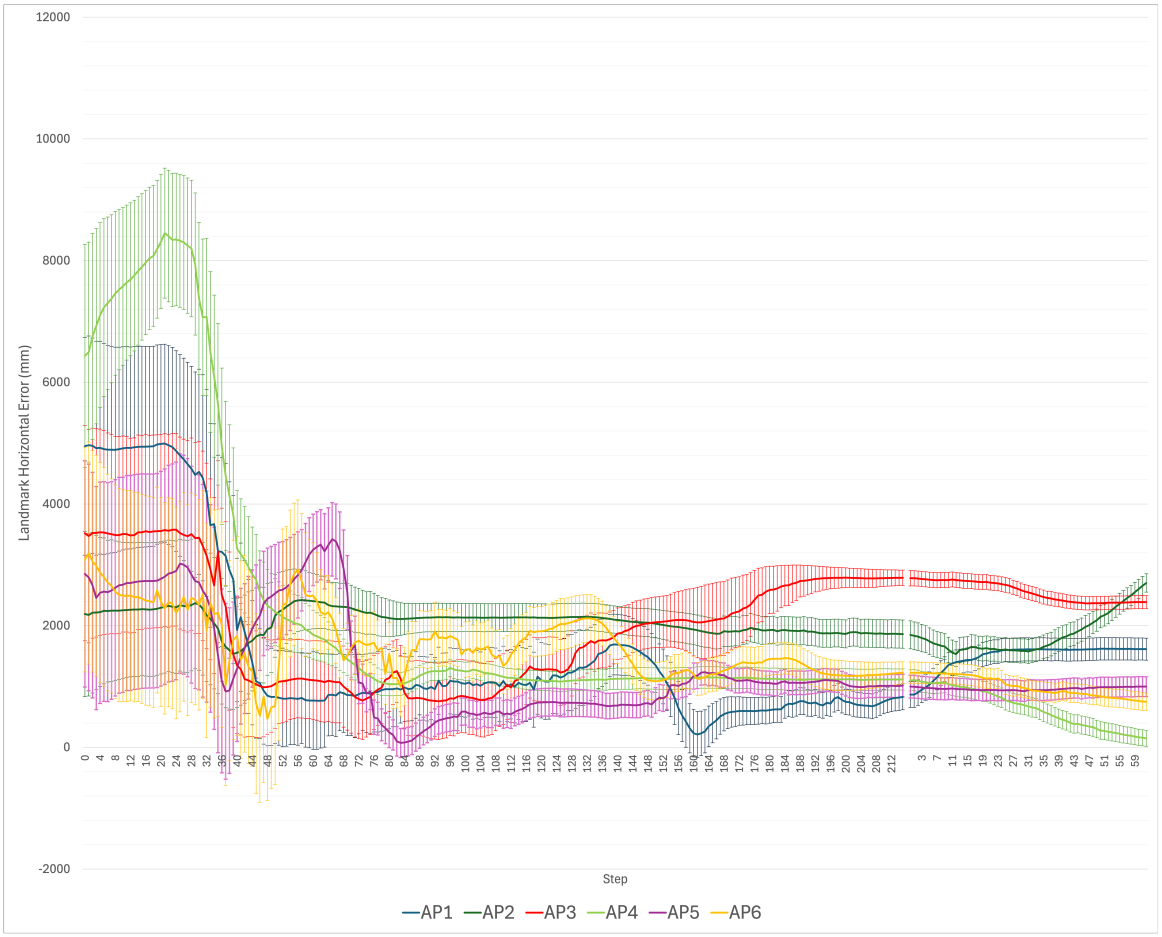


Figure 5.15: Reverse then Short AP position error per step. The white cut-off in the centre of the chart represents the switch from the reverse trial to the short trial. The error bars represent the standard deviation of each landmark particle filter

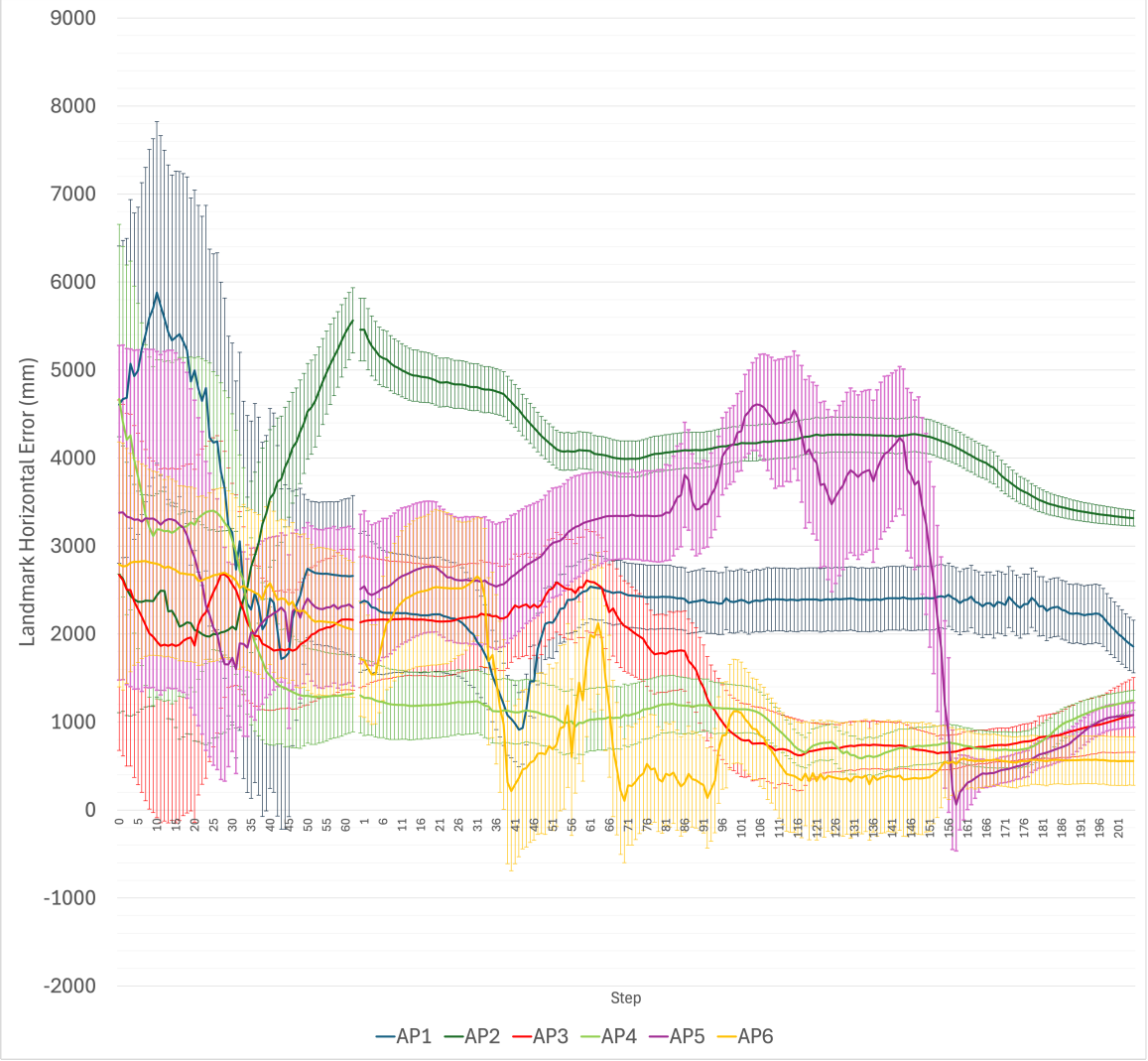


Figure 5.16: Short then Reverse AP position error per step. The white cut-off in the centre of the chart represents the switch from the short trial to the reverse trial. The error bars represent the standard deviation of each landmark particle filter

| | Forward Trial | Short Trial | Short Using Forward | Forward Using Short |
|---|---|---|---|---|
| AP1 Final Horizontal Position Error (m) | 1.69 | 2.78 | 2.40 | 1.33 |
| AP2 Final Horizontal Position Error (m) | 2.51 | 5.57 | 1.95 | 2.30 |
| AP3 Final Horizontal Position Error (m) | 0.74 | 2.36 | 1.37 | 1.14 |
| AP4 Final Horizontal Position Error (m) | 0.94 | 1.29 | 0.83 | 1.13 |
| AP5 Final Horizontal Position Error (m) | 0.78 | 2.66 | 0.43 | 0.95 |
| AP6 Final Horizontal Position Error (m) | 0.39 | 2.14 | 0.75 | 0.47 |

Table 5.9: Statistics for the landmark position estimates for the Short and Forward Trial

Posterity SLAM benefits the overall positioning solution as data from trials can be shared meaning that over time the system can have a better estimate of the landmark positions due to more data. This specific version of the algorithm presents a positioning solution that has sub two metre accuracy without the need for a dedicated survey step, as previous SLAM paths essentially conduct the survey steps automatically. Furthermore, it has been shown that landmark position estimate accuracy is normally lower for shorter paths. This impacts the positioning solution as the particle filter is not able to weight paths correctly because the individual particle weights are less reliable. Posterity SLAM offers a solution to this problem as the landmark estimates of longer trials can be used to improve the positioning solution for shorter paths.

# CHAPTER 6

# Conclusions

This thesis explored indoor positioning using WiFi RTT based positioning solutions. Review of the literature highlighted a number of gaps in knowledge, these included the exploration of filtering techniques, WiFi RTT SLAM, WiFi RTT cooperative SLAM and systems that integrated WiFi RTT alongside other radio signals. Outlier detection methods were also under researched. A common issue within WiFi RTT and other WiFi-based positioning techniques were the requirements of a survey phase where certain data of an environment would be collected before positioning could actually be done, for example the position of APs or fingerprints. A technique that removes the need for any prior knowledge of an environment and any survey phases is SLAM. This is a technique that is very promising in the context of WiFi RTT positioning. Using SLAM to conduct the survey step in order to provide an estimate of AP positions for subsequent SLAM was also explored. This technique is known as Posterity SLAM. There were a number of conclusions which are discussed below.

## 6.1 WiFi RTT Characteristics

The answers to the research questions *How susceptible is WiFi RTT to multipath effects, NLOS signal reception and instrument bias?* and *How does NLOS reception*

*affect the accuracy of WiFi RTT measurements?* are provided below.

- Instrument Bias — It was confirmed that a device-dependent instrument bias existed in the APs and/or mobile device. During initial experimentation, the bias was between -2.4m and -2.6m for two Google Nest WiFi Points and -0.6m for a Google Nest WiFi router when a Google Pixel 4a smartphone is used. This bias changed over time, the changes coinciding with software updates to the mobile device or AP.

- Multipath interference — It was demonstrated that WiFi RTT is susceptible to multipath interference. In the presence of forced multipath conditions, the error of WiFi RTT increased.

- NLOS signal reception error — Experiments that forced NLOS conditions demonstrated that WiFi RTT is susceptible to attenuation and NLOS reception. The WiFi RTT signal was attenuated by a wooden door, resulting in a range error of roughly 0.045m. It was found that NLOS conditions which resulted in the WiFi RTT signal being reflected before reaching the mobile device led to ranging errors of more than a metre in indoor environments.

- The NLOS signal reception experiment showed that the RSSI of the WiFi RTT signal could be used to identify potential NLOS conditions if the WiFi RTT measured range did not correlate with the RSSI path propagation of the signal. This presented an opportunity for RSSI-based outlier detection.

## 6.2 WiFi RTT Positioning

The answers to the research questions *What accuracy can be achieved when least squares-based positioning algorithms are used for pedestrian WiFi RTT-based positioning and navigation?*, *What accuracy can be achieved when filtering-based positioning algorithms such as particle filters, genetic filters and grid filters are used for pedestrian WiFi RTT-based positioning and navigation?* and *What accuracy and reliability improvements can be achieved for WiFi RTT- based positioning and navigation by using outlier detection techniques?* are provided below.

- Filtering techniques applied to WiFi RTT in stationary scenarios — a particle filter with sequential importance resampling, a genetic filter and a grid filter, were tested in a variety of environments. It is possible to achieve sub-metre accuracy in the tested environments, as 64.3% of trials produced sub-metre accuracy. Sub-two-metre accuracy is consistently achievable in the tested environments, as 97.6% of trials produced sub-two-metre accuracy. The results were compared to single-epoch least squares, and it was found that the algorithms including outlier detection improved the positioning accuracy on average across all environments by 40%, 48% and 40% for the particle filter, genetic filter and grid filter respectively. The best performing algorithm on average across all environments was the genetic filter. This could be attributed to better handling of particle degeneracy. The grid filter performed better than the other filters in the simplest environments with only LOS signals. WiFi RTT can provide a viable solution for sub-two-metre indoor positioning accuracy in most environments, provided that RSSI-based outlier detection is used in NLOS environments. In LOS conditions, WiFi RTT is able to offer sub-metre accuracy. The computational efficiency of the algorithms were also compared, and it was shown that the genetic filter takes around 60% longer per epoch than the particle and grid filter for 400 particles.

- Filtering techniques applied to WiFi RTT in motion-based scenarios — In motion-based scenarios, single-epoch least squares is completely ineffective. All filtering algorithms combined with PDR both with and without RSSI-based outlier detection demonstrated that sub-two-metre positioning accuracy is achievable when combined with WiFi RTT in the residential environment used. In 33 out of 36 trials, the final position estimate was more accurate than the initial position estimate. In motion-based scenarios, the best performing filter varied per trial, with no clear standout best. WiFi RTT can be fused with sensors available in a mobile device to produce an integrated navigation solution that can achieve sub-metre accuracy for a moving pedestrian. The performance of the algorithm at the start of a trial is impacted by the initial position estimate. The initial position estimates for certain trials were

impacted by NLOS error at the start of the trials caused by walls, obstacles, and pedestrian body-blocking. Furthermore, poor orientation readings and to a certain extent, poor step length estimation can result in a degraded path estimation due to over-reliance on the PDR at the prediction step.

- RSSI-based outlier detection in stationary scenarios — An RSSI-based outlier detection algorithm was developed and was applied to all positioning algorithms. In the stationary scenarios the outlier detection provided a 10.3%, 16.8% and a 10% mean improvement in the positioning accuracy across all environments for the particle filter, genetic filter and grid filter respectively. RSSI-based outlier detection improved the positioning accuracy of the algorithms for 13 out of 18 of the stationary trials. However, the 5 trials which resulted in a reduced positioning accuracy came from environments with only LOS conditions. RSSI-based outlier detection is likely to provide false positives in these kinds of environments, but so long as there is an overall improvement in the positioning solution in a diverse range of environment, these false positives are acceptable. Furthermore, it is rare to have only LOS conditions in a practical environment.

- RSSI-based outlier detection in motion-based scenarios — The RSSI-based outlier detection algorithm was applied to all positioning algorithms. RSSI-based outlier detection was found to improve the positioning accuracy of the algorithms for 15 out of 18 of the in-motion trials. This improvement was between 0 and 25%. The RSSI-based outlier detection improved accuracy, but to a lesser degree than in the static experiments.

## 6.3 WiFi RTT SLAM

The answers to the research questions *What accuracy can be achieved when SLAM-based techniques are used with WiFi RTT for indoor pedestrian navigation?* and *What accuracy can be achieved when SLAM-based techniques are used with WiFI RTT to estimate the position of WiFi RTT access points?* are provided below.

- WiFi RTT SLAM — a FastSLAM algorithm was applied to WiFi RTT positioning in order to explore WiFi RTT's potential as a positioning solution where prior knowledge of the environment is not a requirement (except for initial position and initial heading which can be readily available due to GNSS readings when entering a building and the mobile device's orientation sensors). The algorithm was able to provide a final horizontal error that was sub-two-metre and a mean horizontal error across all steps below sub-two-metre. The algorithm was able to determine the location of the landmarks to sub-two-metre 83% of the time and sub-metre 50% of the time. Performance was found to be better when more direct line of sight signals were available at the starting position. With NLOS signals the accuracy was worse, which is expected as it impacts both the location estimates of the mobile device and the access points. These results are promising as they represent that a mobile device can be position to sub-two-metre accuracy without prior knowledge of the indoor environment.

- WiFi RTT Posterity SLAM — the FastSLAM algorithm was then improved by taking advantage of previous SLAM maps to initialise the landmark state of the landmark particle filters of subsequent trials. This is referred to as Posterity SLAM and is a type of cooperative SLAM. Posterity SLAM benefits the overall positioning solution as data from trials can be shared, meaning that over time the system can have a better estimate of the landmark positions due to more data. This specific version of the algorithm presents a positioning solution that has sub two metre accuracy for the mobile device without the need for a dedicated survey step, as previous SLAM paths essentially conduct the survey steps automatically. The number of improved landmark estimates due to posterity SLAM was greater than or equal to 50% for all but one trial. Posterity SLAM achieved sub-two-metre landmark position accuracy 78% of the time, improving on regular SLAM, which achieved sub-two-metre accuracy 61% of the time. The landmark position accuracy was sub-metre 42% of the time for Posterity SLAM and 28% of the time for regular SLAM. A limitation of this algorithm is the potential error that can be caused by using poor AP

position estimates. However, it would be expected that with more data in the same environment, this error would reduce over time.

# CHAPTER 7

# Recommendations for future work

This chapter will provide recommendations for future work that this thesis did not explore or only partially explored.

- Experimenting on multiple mobile devices and APs — overall, all algorithms would benefit from being tested with different WiFi routers and mobile devices. The experiments in this paper were limited to Google Nest products and the Google Pixel 4a in this regard. Testing on more devices would allow for a better understanding of instrument bias. Newer products may also have better sensors, receivers and transmitters, meaning the algorithms could benefit from the more up-to-date equipment. Conversely, cheaper equipment may lead to worse performance, making the algorithm designs more critical.
- Experimenting in more environments and scenarios — overall, all algorithms would benefit from being tested in more environments. For example, environments with thicker walls, pedestrians moving around, APs at different heights, shopping centres, airports, hotels etc. Motion-based scenarios could also benefit from having more steps and longer trials.
- Automatic Instrument bias calibration — it was observed that a constant instrument bias existed for the WiFi RTT ranges. For the duration of the

research, this bias was manually calibrated out of the WiFi RTT ranges. Research should be conducted on whether this bias is consistent for each specific WiFi RTT product or if the bias varies across the same product line depending on the batch. If the biases are consistent across different WiFi product lines, then it is possible that these biases could be added to a database which could be used to automatically calibrate the instrument bias. SLAM methods could also be used to calibrate these AP biases.

- Automatic orientation-dependent bias calibration and understanding of changing mobile device orientation instead of AP orientation. The research conducted in this paper on the orientation-dependent bias was inconclusive. More research is needed to understand the nature of this error and if possible calibrate for it. Calibrating for this error should be relatively straightforward, as with a position estimate and a possible location for the landmark, it is possible to determine the orientation of the signal with respect to the AP and the orientation of the signal with respect to the mobile device.

- Multipath and NLOS — the specific error pattern and nature of multipath interference on WiFi RTT was not explored in-depth. A better analysis of multipath in WiFi RTT could provide insights into further methods of outlier detection. Identifying signals that have succumbed to multipath would serve to improve the overall positioning solution.

- Improved Pedestrian dead reckoning algorithm — this thesis did not optimise the pedestrian dead reckoning model, better models could certainly be used. All algorithms could benefit from using a superior pedestrian dead reckoning algorithm. This would allow the positioning algorithm to be less susceptible to erroneous orientation or step length estimates.

- RTT fingerprinting — WiFi RTT could be used for fingerprinting in a similar fashion to WiFi RSSI. The use of WiFi RTT in fingerprint maps could potentially allow for a better indoor positioning solution when using fingerprinting. With more data, it is likely that fingerprint maps would

improve, representing another use case that could give RTT better mainstream adoption.

- Computational characteristics of the algorithms — the algorithms as they are currently written are noticeably slow and not optimised for real-time use. Improving the speed of the algorithms to allow them to be used in real time would be useful future work.

- WiFi RTT GraphSLAM — Factor graph optimisation is another SLAM method that is popular. This thesis did not apply GraphSLAM to WiFi RTT SLAM. The application of GraphSLAM to WiFi RTT would be interesting and could outperform FastSLAM. The application of more state-of-the-art SLAM algorithms would also be useful future work.

- WiFi RTT Posterity SLAM with explicit focus on selecting more accurate access point estimates. The posterity SLAM algorithms had an issue where while the accuracy of the landmark position estimates would improve for some APs, for other APs the landmark position accuracy would reduce. This is potentially problematic as over-time these errors could grow. This problem could be solved with more data, better tuning of the algorithms and better outlier detection. More SLAM maps will be constructed as more users traverse an environment. These SLAM maps can be compared to identify outliers in landmark position estimates. With every SLAM trial, a set of new SLAM maps can be constructed based on the posterity of a combination of other SLAM maps. With every new SLAM map, the number of landmark position estimates that can be constructed will grow. Overtime the system will have a better estimate of the landmark positions as with more trials the APs will be ranged from more angles making the landmark position estimates more reliable.

- WiFi RTT Cooperative SLAM — Multiple devices traversing multiple paths can be used simultaneously to construct SLAM maps. This could be particularly practical in large venues where there will be many mobile devices. Multiple mobile devices contributing to the construction of a single SLAM map at the same time would allow for more geometries to each AP to be determined. By

having more geometries to each AP, it would be expected that the AP position estimates would be better. This improvement in AP position estimate would allow for a better indoor positioning system in a shorter amount.

- One-sided WiFi RTT was not explored in this thesis. The use of one-sided WiFi RTT would allow access to more potential range measurements in many environments. However, these range measurements are less accurate than two-sided WiFi RTT. Using one-sided WiFi RTT measurements would be useful for determining how the algorithms perform in more practical scenarios where WiFi RTT is not supported by default. This is because the only way to access a range measurement from these APs is by using one-sided WiFi RTT. A combination of two-sided WiFi RTT and one-sided WiFi RTT would also be interesting to explore.

- In 2023, WiFi Protocol 802.11az launched [80]. This new protocol, known also as "Next Generation Positioning" could improve WiFi RTT to consistent sub-metre accuracy. Following this, WiFI Protocol 802.11bk is under development and aims to improve WiFi RTT even further. This upgrade will allow WiFi RTT to use 320MHz waveforms, which will improve the precision of WiFi RTT ranging [81]. The APs used in this thesis were only 802.11mc compatible, and it is unclear when 802.11az will roll out to production use. 802.11bk is still under active development. The algorithms would benefit from being trialled on these upgrades to WiFi RTT.

# Bibliography


[1] Groves PD. Principles of GNSS, inertial, and multisensor integrated navigation systems. 2nd ed. Boston/London: Artech house; 2013.

[2] Van Diggelen F. How to achieve 1-meter accuracy in Android; 2018. Available from: `https://www.gpsworld.com/how-to-achieve-1-meter-accuracy-in-android/`.

[3] Dardari D, Closas P, Djurić PM. Indoor Tracking: Theory, Methods, and Technologies. IEEE Transactions on Vehicular Technology. 2015;64:1263-78. Available from: `https://api.semanticscholar.org/CorpusID:23341942`.

[4] Liu H, Darabi H, Banerjee P, Liu J. Survey of Wireless Indoor Positioning Techniques and Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2007;37(6):1067-80.

[5] Gu Y, Lo A, Niemegeers I. A Survey of Indoor Positioning Systems for Wireless Personal Networks. Communications Surveys Tutorials, IEEE. 2009 01;11:13 32.

[6] Nuaimi K, Kamel H. A survey of indoor positioning systems and algorithms; 2011. p. 185 190.

[7] Mainetti L, Patrono L, Sergi I. A survey on indoor positioning systems. 2014 22nd International Conference on Software, Telecommunications and

Computer Networks (SoftCOM). 2014:111-20. Available from: https://api.semanticscholar.org/CorpusID:18335231.

[8] lidar 101: an introduction to lidar technology, data, and applications coastal remote sensing program 2012;. Available from: https://coast.noaa.gov/data/digitalcoast/pdf/lidar-101.pdf.

[9] Google. Wi-Fi location: ranging with RTT; 2022. Available from: https://developer.android.com/guide/topics/connectivity/wifi-rtt.

[10] Bensky A. Wireless positioning technologies and applications. 2nd ed. Boston/London: Artech House; 2008.

[11] WiFI RTT supported access points;. Available from: https://developer.android.com/develop/connectivity/wifi/wifi-rtt#supported-aps.

[12] WiFI RTT supported smartphones;. Available from: https://developer.android.com/develop/connectivity/wifi/wifi-rtt#supported-phones.

[13] RangingRequestBuilder;. Available from: https://developer.android.com/reference/android/net/wifi/rtt/RangingRequest.Builder#addNon80211mcCapableAccessPoint(android.net.wifi.ScanResult).

[14] Horn BKP. Indoor Localization Using Uncooperative Wi-Fi Access Points. Sensors. 2022;22(8). Available from: https://www.mdpi.com/1424-8220/22/8/3091.

[15] Chatzimichail A, Tsanousa A, Meditskos G, Vrochidis S, Kompatsiaris I. In: RSSI Fingerprinting Techniques for Indoor Localization Datasets; 2020. p. 468-79.

[16] C M, C W. Subway station real-time indoor positioning system for cell phones. IEEE. 2017.

[17] Faragher R, Sarno C, Newman M. Opportunistic radio SLAM for indoor navigation using smartphone sensors. 2012 04:120-8.

[18] Leonard JJ, Durrant-Whyte HF. Simultaneous map building and localization for an autonomous mobile robot. In: Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91; 1991. p. 1442-7 vol.3.

[19] Raja KJ, Groves PD. WiFi-RTT Indoor Positioning Using Particle, Genetic and Grid Filters with RSSI-Based Outlier Detection. Proceedings of the 36th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2023). 2023.

[20] Abbott E, Powell D. Land-vehicle navigation using GPS. Proceedings of the IEEE. 1999;87(1):145-62.

[21] Räty TD. Survey on Contemporary Remote Surveillance Systems for Public Safety. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews). 2010;40(5):493-515.

[22] Faisal M, Hedjar R, Al Sulaiman M, Al-Mutib K. Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. International Journal of Advanced Robotic Systems. 2013;10(1):37.

[23] Subedi S, Pyun JY. Practical Fingerprinting Localization for Indoor Positioning System by Using Beacons. 2017.

[24] Kul G, Ozyer T, Tavli B. IEEE 802.11WLAN Based Real Time Indoor Positioning:Literature Survey and Experimental Investigations. 2014.

[25] Dong L, Baoxian Z, Cheng L. A Feature-Scaling-Based k-Nearest Neighbor Algorithm for Indoor Positioning Systems. 2016.

[26] He S, Chan SHG. Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons. 2016.

[27] Sun S, Chen N, Tian T. An Innovative Indoor Location Algorithm Based on Supervised Learning and WIFI Fingerprint Classification. Lecture Notes in Electrical Engineering, vol 473 Springer. 2018.

[28] Song X, Fan X, Xiang C, Ye Q, Liu L, Wang Z, et al. A Novel Convolutional Neural Network Based Indoor Localization Framework With WiFi Fingerprinting. IEEE Access. 2019;7:110698-709.

[29] Altman NS. An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. The American Statistician. 1992;46(3):175-85. Available from: `http://www.jstor.org/stable/2685209`.

[30] Subramanian D. A Simple Introduction to K-Nearest Neighbors Algorithm; 2019. Available from: `https://towardsdatascience.com/a-simple-introduction-to-k-nearest-neighbors-algorithm-b3519ed98e`.

[31] Faragher R, Harle R. Location fingerprinting with bluetooth low energy beacons. IEEE journal on Selected Areas in Communications. 2015;33(11):2418-28.

[32] Martin-Escalona I, Zola E. Improving fingerprint-based positioning by using IEEE 802.11 mc FTM/RTT observables. Sensors. 2022;23(1):267.

[33] IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std 80211-2016 (Revision of IEEE Std 80211-2012). 2016:1-3534.

[34] Horn B. Doubling the Accuracy of Indoor Positioning: Frequency Diversity. Sensors. 2020 03;20:1489.

[35] Ma C. Wi-Fi RTT Ranging Performance Characterization and Positioning System Design. UCL (University College London); 2020.

[36] Guo G, Chen R, Ye F, Peng X, Liu Z, Pan Y. Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS Ranging Approach. IEEE Access. 2019;7:176767-81.

[37] Faragher R, Harle R. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In: Proceedings of the 27th International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GNSS+'14); 2014. p. 201-10.

[38] Hossain AM, Soh WS. A comprehensive study of bluetooth signal parameters for localization. In: Personal, Indoor and Mobile Radio Communications, 2007. PIMRC 2007. IEEE 18th International Symposium on. IEEE; 2007. p. 1-5.

[39] Kotanen A, Hannikainen M, Leppakoski H, Hamalainen TD. Positioning with IEEE 802.11b wireless LAN. In: 14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003.. vol. 3; 2003. p. 2218-22 vol.3.

[40] Julier SJ, Uhlmann JK. Unscented filtering and nonlinear estimation. Proceedings of the IEEE. 2004;92(3):401-22.

[41] Bose A, Foh CH. A practical path loss model for indoor WiFi positioning enhancement. In: 2007 6th International Conference on Information, Communications Signal Processing; 2007. p. 1-5.

[42] Ma Z, Wu B, Poslad S. A WiFi RSSI ranking fingerprint positioning system and its application to indoor activities of daily living recognition. International Journal of Distributed Sensor Networks. 2019;15(4):1550147719837916.

[43] Yiu S, Dashti M, Claussen H, Perez-Cruz F. Wireless RSSI fingerprinting localization. Signal Processing. 2017;131:235-44.

[44] Dong Y, Arslan T, Yang Y. A Multimodal Graph Fingerprinting Method for Indoor Positioning Systems. In: 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN); 2023. p. 1-6.

[45] Dai S, He L, Zhang X. Autonomous WiFi fingerprinting for indoor localization. In: 2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS). IEEE; 2020. p. 141-50.

[46] Garcia-Fernandez M, Hoyas-Ester I, Lopez-Cruces A, Siutkowska M, Banqué-Casanovas X. Accuracy in WiFi Access Point Position Estimation Using Round Trip Time. Sensors. 2021;21(11):3828.

[47] Horn BK. Observation model for indoor positioning. Sensors. 2020;20(14):4027.

[48] Dong Y, Shi D, Arslan T, Yang Y. Error Investigation on Wi-Fi RTT in Commercial Consumer Devices. Algorithms. 2022;15(12). Available from: https://www.mdpi.com/1999-4893/15/12/464.

[49] Jurdi R, Chen H, Zhu Y, Loong Ng B, Dawar N, Zhang C, et al. WhereArtThou: A WiFi-RTT-Based Indoor Positioning System. IEEE Access. 2024;12:41084-101.

[50] Mohsen M, Rizk H, Yamaguchi H, Youssef M. LocFree: WiFi RTT-based Device-Free Indoor Localization System. In: Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Spatial Big Data and AI for Industrial Applications. GeoIndustry '23. New York, NY, USA: Association for Computing Machinery; 2023. p. 32–40. Available from: https://doi.org/10.1145/3615888.3627813.

[51] Guo G, Chen R, Ye F, Peng X, Liu Z, Pan Y. Indoor smartphone localization: A hybrid WiFi RTT-RSS ranging approach. IEEE Access. 2019;7:176767-81.

[52] Dong Y, Arslan T, Yang Y. Real-Time NLOS/LOS Identification for Smartphone-Based Indoor Positioning Systems Using WiFi RTT and RSS. IEEE Sensors Journal. 2022;22(6):5199-209.

[53] Han K, Yu SM, Kim SL, Ko SW. Exploiting User Mobility for WiFi RTT Positioning: A Geometric Approach. IEEE Internet of Things Journal. 2021.

[54] Sun M, Wang Y, Xu S, Qi H, Hu X. Indoor Positioning Tightly Coupled Wi-Fi FTM Ranging and PDR Based on the Extended Kalman Filter for Smartphones. IEEE Access. 2020;8:49671-84.

[55] Si M, Wang Y, Seow CK, Cao H, Liu H, Huang L. An Adaptive Weighted Wi-Fi FTM-Based Positioning Method in an NLOS Environment. IEEE Sensors Journal. 2022;22(1):472-80.

[56] Cao H, Wang Y, Bi J, Zhang Y, Yao G, Feng Y, et al. LOS compensation and trusted NLOS recognition assisted WiFi RTT indoor positioning algorithm. Expert Systems with Applications. 2024;243:122867.

[57] Faragher R, Harle R. SmartSLAM-an efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing. In: Proceedings of

the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013); 2013. p. 1006-19.

[58] Ferris B, Fox D, Lawrence ND. Wifi-slam using gaussian process latent variable models. In: IJCAI. vol. 7; 2007. p. 2480-5.

[59] Higuchi T. Gaussian process latent variable models for visualisation of high dimensional data. In: Journal of Statistical Computation and Simulation 59.1; 1997. p. 1-27.

[60] Liu R, Marakkalage SH, Padmal M, Shaganan T, Yuen C, Guan YL, et al. Collaborative SLAM based on WiFi fingerprint similarity and motion information. IEEE Internet of Things Journal. 2019;7(3):1826-40.

[61] ARCORE;. Available from: https://developers.google.com/ar.

[62] Gentner C, Avram D. WiFi-RTT-SLAM: Simultaneously Estimating the Positions of Mobile Devices and WiFi-RTT Access Points. In: Proceedings of the 34th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2021); 2021. p. 3142-8.

[63] Google Wifi RTT scan app; 2016. Available from: https://play.google.com/store/apps/details?id=com.google.android.apps.location.rtt.wifirttscan&hl=en_US&pli=1.

[64] Suherman NM, Sagala RS, Prayitno H. Wifi-friendly building, enabling wifi signal indoor: an initial study. 2018.

[65] Mohammed RA, Al-Nakkash AH, Salim ONM. A Comprehensive Study of the Environmental Effects on WiFi Received Signal Strength: Lab Scenario. 2020.

[66] Bayes's theorem. Oxford University Press; 2008. Available from: https://www.oxfordreference.com/view/10.1093/acref/9780199541454.001.0001/acref-9780199541454-e-139.

[67] Arulampalam M, Maskell S, Gordon N, Clapp T. A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking. IEEE Trans Signal Process. 2002;50:174-88.

[68] Gentner C, Ulmschneider M, Kuehner I, Dammann A. WiFi-RTT Indoor Positioning. In: 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS), Portland, Oregon; 2020. p. 1029-35.

[69] Park S, Hwang J, Rou K, Kim E. A new particle filter inspired by biological evolution: Genetic filter. World Academy of Science, Engineering and Technology. 2007;33:83-7.

[70] Lawrence ND. Monte Carlo filter using the genetic algorithm operators. In: Nips. vol. 2. Citeseer; 2003. p. 5.

[71] Zhou N, Lau L, Bai R, Moore T. A Genetic Optimization Resampling Based Particle Filtering Algorithm for Indoor Target Tracking. 2021.

[72] Mikov A, Moschevikin A, Fedorov A, Sikora A. A localization system using inertial measurement units from wireless commercial hand-held devices,. International Conference on Indoor Positioning and Indoor Navigation. 2013:1-7.

[73] Weinberg H. Using the adxl202 in pedometer and personal navigation applications,. 2002. Available from: `https://api.semanticscholar.org/CorpusID:55560101`.

[74] Hongji C. Indoor Positioning Method Using WiFi RTT Based on LOS Identification and Range Calibration. UCL (University College London); 2020.

[75] Montemerlo M, Thrun S, Roller D, Wegbreit B. FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence. IJCAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 2003. p. 1151–1156.

[76] Montemerlo M, Thrun S, Koller D, Wegbreit B. FastSLAM: a factored solution to the simultaneous localization and mapping problem. In: Eighteenth National Conference on Artificial Intelligence. USA: American Association for Artificial Intelligence; 2002. p. 593–598.

[77] Thrun S, Montemerlo M. The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures. The International Journal of Robotics Research. 2006;25:403 429. Available from: https://api.semanticscholar.org/CorpusID:16610913.

[78] Grisetti G, Kümmerle R, Stachniss C, Burgard W. A tutorial on graph-based SLAM. IEEE Transactions on Intelligent Transportation Systems Magazine. 2010 12;2:31-43.

[79] Floater MS. Lecture 13: Non-linear least squares and the Gauss-Newton method. 2018. Available from: https://www.uio.no/studier/emner/matnat/math/MAT3110/h19/undervisningsmateriale/lecture13.pdf.

[80] Segev J, Want R. Newly released IEEE 802.11az standard improving Wi-Fi location accuracy is set to unleash a new wave of innovation; 2024. Available from: https://standards.ieee.org/beyond-standards/newly-released-ieee-802-11az-standard-improving-wi-fi-location-accuracy-is-set-to-unleash-a-new-wave-of-innovation/.

[81] Segev J, Want R. IEEE P802.11 - 320 MHZ POSITIONING TASK GROUP; 2024. Available from: https://www.ieee802.org/11/Reports/tgbk_update.htm.