Linköping University | Department of Computer and Information Science
Master's thesis, 30 ECTS | Information Technology
2020 | LIU-IDA/LITH-EX-A--20/032--SE

Smartphone Based Indoor Positioning Using Wi-Fi Round Trip Time and IMU Sensors

Smartphone-baserad inomhuspositionering med Wi-Fi Round-Trip Time och IMU-sensorer

Gustav Aaro

Examiner and supervisor: Petru Eles
External supervisor: David Törnqvist

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

硕士学位论文

Dissertation for Master’s Degree

(工程硕士)

(Master of Engineering)

基于 Wi-Fi 往返时间和 IMU 传感器的基于智能手机的室内定位

SMARTPHONE BASED INDOOR POSITIONING USING WI-FI ROUND TRIP TIME AND IMU SENSORS

古斯塔夫 Gustav Aaro

哈爾濱工業大學

Linköping University

2020 年 6 月

国内图书分类号：TP311	学校代码:10213
国际图书分类号：681	密级：公开

工程硕士学位论文
Dissertation for the Master’s Degree in Engineering
(工程硕士)
(Master of Engineering)

基于 Wi-Fi 往返时间和 IMU 传感器的基于智能手机的室内定位

SMARTPHONE BASED INDOOR POSITIONING USING WI-FI ROUND TRIP TIME AND IMU SENSORS

硕士研究生：	古斯塔夫
导师：	臧天仪
副导师：	Petru Eles，臧天仪
实习单位导师：	David Törnqvist，博士
申请学位：	工程硕士
学科：	软件工程
所在单位：	软件学院
答辩日期：	2020 年 6 月
授予学位单位：	哈尔滨工业大学

Classified Index: TP311
U.D.C: 681

Dissertation for the Master’s Degree in Engineering

SMARTPHONE BASED INDOOR POSITIONING USING WI-FI ROUND TRIP TIME AND IMU SENSORS

Candidate：	Gustav Aaro
Supervisor：	Dr. Zang Tianyi
Associate Supervisor:	Petru Eles, Professor
Industrial Supervisor:	David Törnqvist, Dr
Academic Degree Applied for：	Master of Engineering
Speciality：	Software Engineering
Affiliation：	School of Software
Date of Defence：	June, 2020
Degree-Conferring-Institution：	Harbin Institute of Technology

摘　要

虽然全球定位系统（GPS）长期以来一直是对世界上任何地方的实体或个人进行定位的行业标准，但在室内使用时，它的准确性和价值都会大大降低。因此必须使用其他方法来实现室内导航这样的服务。Wi-Fi 协议的一个新标准 IEEE 802.11mc （Wi-Fi RTT）可以根据信号的往返时间（RTT），估计出发射器和接收器之间的距离。本文利用无损卡尔曼滤波器(Unscented Kalman Filter, UKF)，提出了一种基于 Wi-Fi RTT 的智能手机室内定位系统（IPS）。将仅使用基于 RTT 的距离估计作为输入的无损卡尔曼滤波器确立为基准实现。然后本文提出了两种扩展方法来提高定位性能：1)使用智能手机传感器部分惯性测量单元（IMU）作为 UKF 的附加输入的航迹推算算法。2)在非视距条件下，检测和调整距离测量的方法。在办公环境中，在有利的情况（视距条件充足）和次优的情况下（主要是非视距条件），都对实现的智能手机室内定位系统进行了评估。使用这两种扩展方法，在两种情况下都可以达到米级的定位精度，并且 90%的误差都小于 2 米。

关键词：Wi-Fi RTT，室内定位，FTM，手机，传感器融合

Abstract

While GPS long has been an industry standard for localization of an entity or person anywhere in the world, it loses much of its accuracy and value when used indoors. To enable services such as indoor navigation, other methods must be used. A new standard of the Wi-Fi protocol, IEEE 802.11mc (Wi-Fi RTT), enables distance estimation between the transmitter and the receiver based on the Round-Trip Time (RTT) delay of the signal. Using these distance estimations and the known locations of the transmitting Access Points (APs), an estimation of the receiver's location can be determined. In this thesis, a smartphone Wi-Fi RTT based Indoor Positioning System (IPS) is presented using an Unscented Kalman Filter (UKF). The UKF using only RTT based distance estimations as input, is established as a baseline implementation. Two extensions are then presented to improve the positioning performance; 1) a dead reckoning algorithm using smartphone sensors part of the Inertial Measurement Unit (IMU) as an additional input to the UKF, and 2) a method to detect and adjust distance measurements that have been made in Non-Line-of-Sight (NLoS) conditions. The implemented IPS is evaluated in an office environment in both favorable situations (plenty of Line-of-Sight conditions) and sub-optimal situations (dominant NLoS conditions). Using both extensions, meter level accuracy is achieved in both cases as well as a 90th percentile error of less than 2 meters.

Keywords: Wi-Fi RTT; Indoor positioning; FTM; Smartphone; Sensor fusion

Glossary

AoA	Angle of Arrival.
AP	Access Point
FTM	Fine Timing Measurement Protocol.
IMU	Inertial Measurement Unit.
IPS	Indoor positioning system.
LoS	Line-of-sight.
NLoS	Non-line-of-sight.
RSS	Received Signal Strength.
RSSI	Received Signal Strength Indicator.
RTT	Round-Trip Time.
TDoA	Time Difference of Arrival.
ToF	Time of Flight.
Wi-Fi RTT	IEEE 802.11mc protocol.

目　录

摘　要 I

ABSTRACT II

GLOSSARY III

TABLE OF FIGURES VIII

CHAPTER 1 INTRODUCTION 1

1.1 BACKGROUND 1

1.2 THE PURPOSE OF THE PROJECT 3

1.3 RESEARCH QUESTIONS 3

1.4 THE STATUS OF RELATED RESEARCH 3

1.4.1 Fusion of Wi-Fi, smartphone sensors and landmarks using the Kalman filter for indoor localization 4

1.4.2 Smartphone-based indoor positioning using Wi-Fi Fine Timing Measurement protocol 5

1.4.3 SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization 6

1.4.4 Non-line-of-sight identification and mitigation using received signal strength 8

1.5 METHOD 11

1.5.1 Pre-study informal literature review 11

1.5.2 Pre-study experiments 12

1.5.3 Localization performance metrics 12

1.5.4 Ground truth determination 12

1.6 MAIN CONTENT AND ORGANIZATION OF THE THESIS 13

CHAPTER 2 THEORY 14

2.1 NON-LINE-OF-SIGHT PROPAGATION 14

2.2 RANGING TECHNIQUES 15

2.2.1 Received Signal Strength 15

2.2.2 Time difference of arrival 16

2.2.3 Time of flight 17
2.2.4 Wi-Fi RTT 18
2.3 POSITION ESTIMATION TECHNIQUES 20
2.3.1 Trilateration 20
2.3.2 Triangulation 22
2.3.3 Kalman filter 23
2.3.4 Fingerprinting 29
2.4 EMPIRICAL RESEARCH IN SOFTWARE ENGINEERING 29
CHAPTER 3 WI-FI RTT CHARACTERISTICS 32
3.1 STATIONARY MEASUREMENTS 32
3.1.1 LoS measurements 33
3.1.2 NLoS measurements 38
3.2 MOBILE MEASUREMENTS 41
3.2.1 LoS measurements 41
3.2.2 NLoS measurements 43
3.3 CONCLUSIONS 44
CHAPTER 4 SYSTEM REQUIREMENT ANALYSIS 45
4.1 THE GOAL OF THE SYSTEM 45
4.2 THE FUNCTIONAL REQUIREMENTS 45
4.2.1 Logging 45
4.2.2 Wi-Fi RTT 45
4.2.3 Device inertial sensor information 46
4.2.4 NLoS/LoS detection 46
4.2.5 Offline processing of data 46
4.2.6 Position estimation 46
4.2.7 User interface 47
4.3 THE NON-FUNCTIONAL REQUIREMENTS 47
4.4 BRIEF SUMMARY 47
CHAPTER 5 SYSTEM DESIGN 48
5.1 MOBILE APPLICATION DESIGN 48
5.2 POST-PROCESSING APPLICATION DESIGN 50

CHAPTER 6 SYSTEM IMPLEMENTATION AND TESTING 51

6.1 SOFTWARE AND HARDWARE 51
6.1.1 Android Wi-Fi RTT API 51
6.1.2 Smartphone 52
6.1.3 Wi-Fi RTT access points 52
6.2 LOCALIZATION IMPLEMENTATION 54
6.2.1 Baseline implementation 54
6.2.2 First extension: Sensor fusion with dead reckoning 59
6.2.3 Second extension: LoS/NLoS detection 61
6.2.4 Implementation overview 68
6.3 KEY PROGRAM FLOW CHARTS 68
6.3.1 Diagram of localization procedure 69
6.3.2 Ranging process 70
6.4 KEY INTERFACES OF THE SOFTWARE SYSTEM 71
6.5 SYSTEM EVALUATION 72
6.5.1 Measurement calibrations 72
6.5.2 Evaluation environments 72
6.5.3 Evaluation hypotheses 74
6.6 BRIEF SUMMARY 75

CHAPTER 7 RESULTS 76

7.1 PATH A 76
7.2 PATH B 79

CHAPTER 8 DISCUSSION 82

8.1 WI-FI RTT RANGING 82
8.2 SYSTEM EVALUATION 83
8.3 METHOD AND IMPLEMENTATION 85
8.4 ETHICAL ASPECTS OF INDOOR POSITIONING 86

CHAPTER 9 CONCLUSION 88

9.1 FUTURE WORK 89

REFERENCES 91

APPENDIX 96

Appendix A – Measurement calibration 96

STATEMENT OF ORIGINALITY AND LETTER OF AUTHORIZATION.... 97

ACKNOWLEDGEMENTS 98

Table of figures

Figure 2.1 Illustration of the TDOA ranging technique 17
Figure 2.2 Illustration of the FTM ranging procedure using two measurements 19
Figure 2.3 Illustration of trilateration using ideal measurements 21
Figure 2.4 Illustration of 2-D angulation using two angles α and β as well as a distance between two reference points 22
Figure 2.5 Example normal distribution of measurements from two different sensors. 24
Figure 2.6 Illustration of the Kalman filter position estimation of the car example. 25
Figure 2.7 System diagram with signal definitions of a system using a Kalman filter for state estimation. [26] 26
Figure 3.1 Setup for range measurements made in LoS conditions............. 33
Figure 3.2 Ranging measurements in LoS conditions 33
Figure 3.3 Distribution of measurements at each measurement location for AP 4. 34
Figure 3.4 Distribution of measurements at each fixed distance for AP 2... 35
Figure 3.5 Outdoor ranging measurements for AP 4 and a reference line with the expected slope of 1. 36
Figure 3.6 Distribution of outdoor LoS ranging results for AP 4. 37
Figure 3.7 Setup for range measurements made in NLoS conditions. 38
Figure 3.8 Ranging measurements in NLoS for AP 1 and a reference line with the expected slope of 1. 38
Figure 3.9 Ranging measurements in NLoS for AP 1 and a reference line with slope 1. Fixed distances 1 & 2 are excluded. 39
Figure 3.10 Distribution of measurements at each fixed distance in LoS and NLoS. 40
Figure 3.11 The path walked for the mobile LoS measurements. 41
Figure 3.12 Result of the mobile LoS RTT ranging measurements. 42
Figure 3.13 The path walked for the mobile NLoS measurements. 43

Figure 3.14 Result of the mobile NLoS RTT ranging measurements............44
Figure 5.1 Illustration of the publish-subscribe pattern using an event bus. 48
Figure 5.2 Diagram of the android application system design......................49
Figure 5.3 Design of offline application..50
Figure 6.1 Illustration of sample creation for a measurement collection R of size n and a sample size of 4. 62
Figure 6.2 Validation of the Log-distance Path Loss model used 65
Figure 6.3 Calibrated Wi-Fi RTT and RSSI based distance distribution in LoS...66
Figure 6.4 Calibrated Wi-Fi RTT and RSSI based distance distribution in NLoS. 66
Figure 6.5 Implementation overview. 68
Figure 6.6 Overview of the location estimation procedure 69
Figure 6.7 Flow chart of the Wi-Fi RTT ranging process. 70
Figure 6.8 Screenshots of the Android application.......................................71
Figure 6.9 Ground truth and positions of APs for Path A 73
Figure 6.10 Ground truth and positions of APs for Path B 74
Figure 7.1 IPS positioning performance for Path A using different configurations, with a marker placed every 3 meters. Left using baseline implementation and right using baseline together with the dead-reckoning extension...76
Figure 7.2 Cumulative distribution function of the positioning error for all rounds on Path A. 77
Figure 7.3 Mean positioning error ± 1 standard deviation for all walked rounds on Path A. 78
Figure 7.4 IPS positioning performance for Path B using different configurations with a marker placed every 5 meters. Top-left only using the baseline implementation and bottom-right using baseline with both extensions...79
Figure 7.5 Cumulative distribution function of the positioning error for Path B, based on all six rounds. 80
Figure 7.6 Mean positioning error ± 1 standard deviation for all walked rounds on Path B. 81

Chapter 1 Introduction

This chapter introduces the reader to the thesis topic, the aim of the thesis, and the proposed research questions. Next, the status of related research and the research method used is presented.

1.1 Background

GPS has long been considered an industry standard for accurately pinpointing the location of an entity or person anywhere in the world. But due to the gradual loss of received signal strength caused by construction materials such as concrete, the technology loses much of its accuracy and, as a consequence, its applicability when used indoors. To address this issue, several different Indoor Positioning Systems (IPSs) have been developed. A variety of different radio-based technologies such as Wi-Fi and Bluetooth have been used to develop such systems, with various degrees of achieved accuracy [1].

Existing IPSs that use Bluetooth normally consist of several Bluetooth low energy beacons (a device that transmits its identity to nearby portable devices) that are systematically installed in the area where the service is to be available. The position of an entity is then calculated using either a ranging technique in combination with a localization algorithm or by looking up the current signal fingerprint consisting of signal strengths to all beacons within reach in a pre-trained database (fingerprinting) [2]. However, one disadvantage of solutions based on these beacons is that they often require a training phase to yield accurate results. Moreover, they do not bring any additional technical benefits beyond the purpose of indoor localization.

Most of today's existing IPSs that use Wi-Fi technology use the Received Signal Strength Indicator (RSSI) but this measurement is known to have its accuracy limitations [3]. However, in a new standard of the Wi-Fi protocol, IEEE 802.11mc (Wi-Fi RTT), an approximation of the distance between the phone and the access point can be reported based on the Round-Trip Time (RTT) of the signal using the Fine

Timing Measurement (FTM) protocol [4]. With this information and the location of the wireless Access Points (APs), the position of a device supporting the new Wi-Fi protocol can be calculated. By using APs with the new standard to calculate position, it is theoretically possible to create a universal plug and play IPS without the need of a training phase. This technology could in addition to indoor location also bring wireless internet access in the area covered by the service. One of the latest versions of Android (Android P) supports this new Wi-Fi standard and can, therefore, be used to implement an IPS based on Wi-Fi RTT measurements.

Recent work suggests that the new Wi-Fi technology has the potential to achieve meter level accuracy in optimal environments [5]–[7]. However, when the device does not have Line-of-Sight (LoS) to one or more APs, the ranging measurement becomes unreliable due to Non-Line-of-Sight (NLoS) propagation. To mitigate these issues, it is important to be able to detect when a distance measurement has been made under NLoS conditions. By knowing this, the measurement can either be discarded or adjusted according to some model to increase the accuracy of the IPS in such situations. By not exclusively relying on ranging measurements from APs but also utilizing some other source of data, the robustness and accuracy of the IPS could potentially be improved even further through sensor fusion. Modern smartphones have many sensors that are part of the Inertial Measurement Unit (IMU) that could be used for such a purpose, for example, step counter, orientation sensor, and accelerometer.

Accurate indoor positioning can be useful in several different situations. One such popular use case is helping customers navigate within large malls without the need to use indoor maps. By offering a customer application based on an IPS, visitors can be effortlessly guided to their desired destination. However, providing customers inside a mall with indoor navigation does not necessarily require meter level accuracy, in the same way a driver does not need a GPS to know *which lane* on a multi-lane road he is driving on, as long as *which road* is possible to determine. In other situations, high accuracy is essential. For example, consider an indoor navigation service in a crowded office space with a multitude of rooms of different shapes and sizes. Estimating a position with the accuracy of a few meters could, in such case, mean an entirely different room, which for most applications would not be feasible.

1.2 The purpose of the project

The main purpose of the thesis is to investigate if Wi-Fi RTT together with techniques for sensor fusion and NLoS/LoS detection can be used to accurately estimate the indoor position of an Android phone, even in sub-optimal conditions. To achieve this, a test environment consisting of several Wi-Fi RTT access points will be used. A positioning algorithm will be implemented on an Android phone using both the Android Wi-Fi RTT API and sensor data from the device. The algorithm will be evaluated under both LoS and NLoS conditions.

Senion

The work of this thesis was carried out at Senion AB, which is located in Linköping, Sweden. Senion is a global provider of high accuracy systems for indoor positioning. The name Senion is a merge of the concept sensor fusion, which is the technology that their IPS 'Senion IPS' is based upon.

1.3 Research questions

To achieve the purpose of the thesis, the following questions will be investigated:

- How do NLoS conditions affect Wi-Fi RTT ranging measurements?
- Is it possible to accurately determine if a Wi-Fi RTT ranging measurement has been done in LoS or NLoS conditions on an Android smartphone?
- How do NLoS detection and sensor fusion affect the possibility of achieving meter level accuracy for an IPS using Wi-Fi RTT?

1.4 The status of related research

This section aims for the reader to create an understanding of the status of related research and recent publications relevant to the subject of the thesis.

1.4.1 Fusion of Wi-Fi, smartphone sensors and landmarks using the Kalman filter for indoor localization

Chen et al. proposed a sensor fusion framework for combining Wi-Fi, Pedestrian Dead Reckoning (PDR) and landmarks, made for indoor localization [8]. The framework was designed to run on resource-limited smartphones and as such the computational complexity of the methods used was taken into account. Wi-Fi RSS measurements were used for distance estimation by using a path loss model. To mitigate problems with Wi-Fi signal variations, the authors presented a Weighted Loss Path (WPL) model which assigned weights inversely proportionate to the estimated distance for each of the distance estimations obtained through the path loss model. By summarizing each AP's position multiplied with the weight of the distance estimation, the device's estimated position was obtained.

To implement location estimation based on pedestrian dead reckoning, several of the phone's IMU sensors were used. In the proposed algorithm for PDR, the state (X_t, in this case position) at time *t* was calculated as:

$$X_t = X_{t-1} + L_t \begin{bmatrix} sin(\theta_t) \\ cos(\theta t) \end{bmatrix}$$

Equation 1.1

where X_{t-1}, L_t, θ_t is the previous position, step length, and walking direction. As this model only provides relative information, it is dependent on the initial state. For this purpose, landmarks with known positions were leveraged. The algorithm was also restarted each time the device reached a landmark. For step detection, the accelerometer was used. To mitigate noisy sensor data, a smoothing function was applied, after which a simple threshold of the vertical acceleration could be used to detect a step. To approximate the step length of the user, the authors adopted a dynamic approach that considers the step length variation by utilizing a relation between acceleration magnitude and step length originally proposed by Jin et al. [9]. Using data from the accelerometer, the step length L can be calculated by using:

$$L = \beta(a_{max} - a_{min})^{1/4}$$

Equation 1.2

where β is a coefficient that needs to be adjusted for different users, and a_{max}, a_{min} is the highest and lowest acceleration recorded during the step. To estimate walking direction, the Android orientation sensor, which is a combination of magnetometer and accelerometer readings, was used. Gyroscope readings were also used to compensate for electronic interference from the magnetometer by utilizing a Kalman filter to achieve a more robust direction estimation.

Lastly, sensor fusion of the PDR and Wi-Fi data with a Kalman filter was used for position estimation. The Kalman filter was preferred over other possible state estimation techniques such as the particle filter as it is computationally less expensive. The implemented IPS was tested in two different environments. In the first site, the mean localization errors of Wi-Fi WPL and PDR using landmarks were 2.8977m and 1.7547m, respectively, and the proposed fusion model had a mean error of 0.9945m. In the second site, the mean localization errors of Wi-Fi WPL and PDR using landmarks were 3.5189m and 1.7727m respectively and the proposed fusion model had a mean error of 0.8492m.

1.4.2 Smartphone-based indoor positioning using Wi-Fi Fine Timing Measurement protocol

Sami Huilla used the Wi-Fi RTT Android API to implement an IPS using the FTM protocol in his master's thesis, published in 2019 [5]. In this thesis, the accuracy of the technology, as well as the implemented IPS, is evaluated at different environmental conditions. The author mentions that ranging measurement with the technology indicated a need for calibration, as large offsets in distance measurements were observed. The calibration was performed according to the Android Open Source Project Wi-Fi RTT calibration guide [10]. Two APs was set up in an indoor corridor and a robot was driven with constant speed from one AP to the other. The robot completed a round-trip six times to investigate if the phone's orientation influenced the ranging measurements. After the calibration procedure, Huilla concludes that the ranging is mostly accurate within the tolerance stated in the calibration guide (< 2m

90th percentile absolute error). The results did, however, show that the orientation of the phone does influence the ranging accuracy, with a higher accuracy achieved when the top of the phone faced the AP.

Using ranging results from the Android API and multiple FTM responders (APs), two different methods for position estimation were implemented and compared; a Non-linear Least Squares (NLS) algorithm and an Unscented Kalman Filter approach (UKF). The two methods were evaluated in two different environments. One ideal site providing LoS to all APs and one more realistic office site with multiple NLoS conditions. The UKF achieved a mean positioning error of 0.72m and a 90th percentile error of 1.17m on the ideal site. NLS achieved a mean error of 1.01m and a 90th percentile error of 1.89m on the same site. On the second site, UKF achieved a mean error of 4.65m a 7.57m 90th percentile. This was, however, improved by assuming all measurements above 10m were subject to NLoS conditions and using a simple correction formula to mitigate the overestimated distances. With this correction method, the UKF instead achieved a mean error of 2.41m and a 4.49m 90th percentile error in the same site.

1.4.3 SmartPDR: Smartphone-based pedestrian dead reckoning for indoor localization

Kang et al. proposed SmartPDR; a service for indoor localization which does not require any infrastructure, but instead utilizing only inertial sensors of the smartphone [11]. In this paper, the authors argued that a practical IPS is one that should consider the absence of infrastructure or pre-trained databases. The proposed system adopted a pedestrian dead-reckoning approach, utilizing multiple inertial sensors of the smartphone such as accelerometer, magnetometer, and gyroscope. Through the use of these sensors, solutions for step event detection, heading direction estimation, and step length estimation were proposed.

To accomplish step detection, the accelerometer sensor was used. This sensor measures the inertial force acting upon the device in three different axes. Steps taken by the user were detected by reading the inertial force, with a periodical pattern triggering the step detection. The inertial forces acting upon the device along the vertical axis relative to the ground was used as the strongest indication of a step taken. Using raw sensor data to accomplish step detection was however non-trivial. One problem that arose with this approach is that the device orientation affected the measured forces along the different axes. This problem was accounted for by multiplying the acceleration vector a_t for the local coordinate system (LCS) with the rotation matrix of the device, projecting the acceleration to a global coordinate system (GCS) as:

$$a_t^{GCS} = R_t a_t^{LCS}$$

The acceleration was also filtered to remove the influence of gravity on the measurements. This was done by subtracting the gravity contribution which was identified using a high pass filter on the z-axis of a_t^{GCS} and modeling all acceleration along this axis as noise. Using the filtered acceleration measurement, a step was then identified through peaks in the acceleration of the z-axis.

The authors also proposed a method for heading estimation. When holding a smartphone in the hand, the placement is unstable, and the tilt of the local coordinate system axes are normally under continuous change. The tilt of the phone affects the magnetometer reading and was compensated for by once again using the rotation matrix to transform the phone's local coordinate system to the global. Both magnetometer and gyroscope data were then considered to find a good estimate of heading direction, by making sure that both data sources supported new estimations. If a change in heading direction was suggested only by one of the sources, a previous estimate was used until a change was supported by both sensors.

The final method proposed is a technique for step length estimation. To accomplish this task, the authors used an earlier proposed approach that uses accelerometer data to estimate step length. More specifically, the vertical impact a_{pp}^{step}, defined as the difference between the current peak and the previous valley of the step acceleration, was used. According to the earlier proposed method, the step length l_k is linearly related to the fourth square root of the vertical impact as:

$$l_k = \beta \sqrt[4]{a_{pp,t}^{step}} + \gamma$$

The authors, however, considered using the logarithm instead of the fourth square root. Using simulations, they found the estimation error between the two models to be nearly identical in most situations, except for the logarithmic approach performing slightly better for small reference steps and slightly worse as the reference step becomes larger. Therefore, they used a combination of the two approaches as:

$$l_k = \begin{cases} \beta\sqrt[4]{a_{pp,t}^{step}} + \gamma, & \text{for } a_{pp,t}^{step} < \mathrm{a}_{\tau}^{\mathrm{step}} \\ \beta\log\left(a_{pp}^{step}\right) + \gamma, & \text{for } a_{pp,t}^{step} \geq a_{\tau}^{step} \end{cases}$$

for some acceleration threshold a_{τ}^{step} (since a larger acceleration impact indicates a longer step).

Using these three techniques based on data from inertial sensors, the authors proposed the indoor localization system SmartPDR. In the testing environment, SmartPDR achieved an average location error of 1.35m, never exceeding 2m during the whole period of the experiment. Their results show that SmartPDR outperforms dead reckoning approaches only using either gyroscope or magnetometer which had location errors of up to 12m in the same testing environment. In all results, however, the starting position was assumed to be known exactly.

1.4.4 Non-line-of-sight identification and mitigation using received signal strength

Xiao et al. have performed extensive research on how to identify and mitigate non-line-of-sight signals for use in smartphone-based indoor positioning [12]. In this paper, three different algorithms designed to separate LoS and NLoS measurements using the received signal strength are presented. The performance of the algorithms are then compared. The authors explore several different features that can be extracted from samples of RSS measurements collected over a short period. Examples of such features are mean, standard deviation, kurtosis, skewness, Rician K factor, and $\chi 2$ goodness of fit. Mean (μ) and standard deviation (σ_s) are well-known features of probability distributions and are used to derive the other features.

Kurtosis is a measure of the peakedness of the probability distribution and is used with the idea that RSS measurements done in LoS generally have a more centralized distribution than samples collected in NLoS. **Skewness** measures the asymmetry of the probability distribution. NLoS measurements are expected to have a higher degree of asymmetry as different NLoS propagation effects can greatly affect RSS measurements. **Rician K factor** is defined as the ratio between the power in the direct path and the power in other scattered paths [12]. **The goodness of fit (χ2)** is a measure of the distance from the measured received signal strength and the underlying Rician distribution. Rician fading is a stochastic model for radio propagation in multipath conditions which models RSS values as Rice distributed. Compared to scattered signals, a signal in LoS conditions reacts significantly less to the environment which leads to different distributions of RSS and, therefore in theory, also different χ2.

The authors collected RSSI values in the experiment site using hardware which allowed querying AP RSSI values as frequently as 1000 times per second. Using the data sets collected, samples were created and the features described above were extracted. Using these features, three classifiers were developed. Two were based on supervised machine learning and one was based on hypothesis testing . The first algorithm is a *Least Squares Support Vector Machine Classifier (LS-SVMC).* The authors motivate this choice through ease of training and quality of generalization. Next, a *Gaussian Processes Classifier (GPC)* was chosen for its proven capabilities, despite a low computational complexity. This is a quality of interest especially for mobile applications. The last algorithm is based on *Hypothesis Testing Classification (HTC)* using a likelihood ratio test where the two hypotheses are defined as:

$$\begin{aligned} H_l&: \quad \mathrm{h} \leq h_t, \qquad \mathit{LoS\ conditions} \\ H_n&: \quad h > h_t, \qquad \mathit{NLoS\ conditions} \end{aligned}$$

Equation 1.3

for some threshold h_t and

$$h = \frac{p\left(x^{(1)}, \cdots, x^{(M)} \middle| H_l\right)}{p\left(x^{(1)}, \cdots, x^{(M)} \middle| H_n\right)} = \prod_{i=1}^{M} \frac{p\left(x^{(i)} \middle| H_l\right)}{p\left(x^{(i)} \middle| H_n\right)}.$$

Equation 1.4

In Equation 1.4, $p\left(x^{(i)}\middle|H_c\right)$ is the probability distribution function of feature $x^{(i)}$ in condition c. With this approach h_t is set to 1. They remark that the joint distribution of the features would have been optimal but would require calculating convolutions of probability distribution functions which comes with an extraordinarily high computational cost. However, according to empirical tests conducted, only 2.02% of the classifications made by the sub-optimal solution differ from that of the optimal approach which implies that the trade-off between computational costs and analytical accuracy is not very high.

For NLoS mitigation, the method is very similar for the machine learning approaches, but instead of a binary classification problem, it becomes a problem of determining a distance given a sample of RSS measurements. The data sets used for training were collected at specific locations in the test site, where the distance to each AP could be calculated and then be used as training data together with the RSS values at each position. For the hypothesis testing approach, mitigation was instead accomplished by using two different propagation models to estimate the distance from RSS values depending on the determined LoS/NLoS condition.

The classification algorithms were tested in an office environment on two different occasions. One on a weekend when there were no other people in the building (static) and one under more normal, busy circumstances (dynamic). Quite anticipated, all algorithms showed a much higher misclassification error in the dynamic environment. *LS-SVMC* achieved a best (lowest) misclassification error of 0.0648 using only µ and Rician K factor as features. *GPC* achieved a best misclassification error of 0.0599 using μ, Rician K factor, and $\chi 2$ goodness of fit as features. *HTC* had the worst accuracy and achieved a best misclassification error of 0.1568 using μ, Rician K factor, and kurtosis as features. In the dynamic environment, the best misclassification rates achieved were instead 0.1401, 0.1301, and 0.3744 for the three algorithms respectively. Mitigation wise, the machine learning algorithms also outperformed the approach based on hypothesis testing. Both such algorithms were shown to improve distance estimation accuracy to around 0.86m as opposed to over 6.6m using conventional propagation models. In comparison, the hypothesis testing-based mitigation approach could achieve an accuracy of 3.5m.

1.5 Method

This section aims to describe the method used to achieve the purpose of the thesis. The thesis is an empirical study on possible improvements to smartphone-based indoor localization using Wi-Fi RTT. The empirical investigation method used to answer the research questions posed was a controlled experiment. Earlier research has established the possibility of using FTM for indoor localization with good accuracy, but their results have shown that there is still room for performance enhancements in situations when a direct line of sight does not exist between the smartphone and one or multiple FTM responders [5]. Two methods to potentially further improve the robustness and accuracy of Wi-Fi RTT based IPSs in such situations have been identified in previously published work. First, a method for detecting NLoS conditions and adjusting such measurements accordingly, has earlier managed to improve the performance of a Wi-Fi-based IPS [12]. Second, supplying the localization algorithm with motion sensor data through sensor fusion, such as when and in which direction a step is taken, has also been found to have a positive impact on positioning performance [8], [11]. This type of relative positioning technique is called dead reckoning. Both techniques have individually been proven to work well in other indoor localization systems, but how they affect the performance of a Wi-Fi RTT based IPS when used together has at the time of writing, to the best of the author’s knowledge, yet not been investigated.

1.5.1 Pre-study informal literature review

Before any design and implementation, an informal literature review was performed within a wide range of different topics. These topics include existing methods for indoor positioning, the accuracy of Wi-Fi RTT, filtering within control theory, as well as signal theory and LoS/NLoS detection. In addition to literature specific to the topic of indoor localization, literature on research methodology within the field of software engineering was also reviewed. The status of related research shows that the field of indoor positioning has been a popular area of research for many years, but that the Wi-Fi RTT technology is a relatively new topic within the field that still has room for innovation.

1.5.2 Pre-study experiments

To understand the characteristics of Wi-Fi RTT and how the technology behaves in different situations, a more practical pre-study was performed. In this phase, different explorative experiments using the technology were conducted to get a better understanding of the ranging accuracy and the difficulties that arise with NLoS conditions.

1.5.3 Localization performance metrics

Many systems for indoor positioning have been proposed, and with them, different methods for measuring and evaluating relevant performance metrics. To determine the quality of an IPS, several different metrics can be used. Al-Ammar et al. list and describe the most common metrics used to evaluate the performance of an IPS [13]. Some of these include accuracy, availability, coverage area, scalability, cost, and privacy. While many of these metrics are very interesting for commercial systems, only accuracy will be measured and used in the evaluation method of this thesis as the other metrics are irrelevant to the research questions posed. Accuracy (or location error) measures how close the estimated position of the IPS user is compared to the actual position [14]. Therefore, the accuracy of an IPS is the average Euclidean distance between the estimated position and the true position. Liu et al. also argues that precision is an important metric to look at when evaluating an IPS [14]. Precision in the context of indoor positioning is a measurement of the variation of performance (or robustness) and is often presented using the Cumulative Distribution Function (CDF) of the distance errors [14]. Instead of just considering the mean, precision considers the variation in distance errors expressed in the percentile format. Accuracy and precision according to the stated definitions above are the IPS performance metrics considered in this thesis.

1.5.4 Ground truth determination

To determine the positioning error of an IPS along a certain path, a ground truth must be determined, meaning a trace of true positions of the IPS device at each measurement time point. This can be achieved in multiple ways. A common method is to use another

IPS known to have very high accuracy. Huilla used a remote-controlled robot with a known starting position that generated a true path using a lidar sensor [5]. In this work the ground truth was provided by an application developed by Senion. A mobile application was used to record sensor data along predefined paths. In a post-processing step the positions of the RTT measurements were computed with high accuracy by utilizing the motion sensor data and knowledge about the predefined path. This result has been used as ground truth and contains the `RangeResult` objects (see Section 6.1.1 for a detailed description of the properties of this object), together with the two additional properties:

- `logTimeMs1970`: Timestamp of when the ranging result was received in Unix Timestamp format (milliseconds since Jan 01, 1970). This is collected for interpolation purposes and for having an absolute timestamp in addition to the timestamp natively provided by the `RangeResult` object which is relative to the device boot time.
- `position`: Object composed of the local x and y coordinates corresponding to the ground truth at the time of measurement.

1.6 Main content and organization of the thesis

The remainder of the thesis is structured as follows. First, a theoretical background relevant to the field of indoor positioning is presented in Chapter 2. In Chapter 3, the characteristics of Wi-Fi RTT as a ranging technology are investigated and the results from the pre-study experiments are presented. Next, the requirements of the implemented system are stated in Chapter 4. In Chapter 5, the high-level design of the implemented system is presented. Chapter 6 describes the details of the system implementation and the testing procedure. Chapter 7 presents the results of the system evaluation, which are also further discussed in Chapter 8. In chapter 9, the resulting conclusions of the thesis are presented.

Chapter 2 Theory

This chapter aims to make the reader familiar with fundamental concepts and techniques important to the field of indoor positioning.

2.1 Non-line-of-sight propagation

Radio propagation is the manner in which radio waves travel or spread when being transmitted. The aim of this thesis heavily relies on Wi-Fi, which is a high-frequency radio wave technology. When working with radio-based technologies, it is important to understand how such signals propagate in different environments and situations. One special case that is of particular interest to this thesis is how electromagnetic waves propagate when a direct LoS between the transmitter and receiver does not exist.

When a LoS path is not present between transmitter and receiver, diffraction, refraction, and/or multipath reflections are the dominant modes of propagation [15]. The *diffraction* of a signal is the phenomena when an electromagnetic wave bends around a sharp edge, thus enabling coverage in an otherwise shadowed location. *Refraction* is the change of direction of an electromagnetic wave that occurs when it passes through some medium. *Multipath* is the effect when multiple copies of the same wave arrive at the receiver by being reflected off of different objects in the environment, thus taking multiple paths from the transmitter to the receiver. [15]

To determine if a communication system will achieve satisfactory performance in some environment, one can use propagation modelling. However, deterministic indoor propagation modeling is very complex due to a large variance in building materials, furniture, floor layout, etc. It is also subject to change as doors are opened/closed and people moving around in the environment [15]. Multipath propagation can result in both constructive and destructive wave interference, the latter which can lead to fading of the propagating radio wave. Due to this type of environmental signal interference and the complex task of accurately modeling such a propagation, it is important to be aware of the effects it can have on information signals carried by radio waves.

2.2 Ranging techniques

Ranging is the procedure of determining an unknown distance of interest through some measurement technique. In indoor positioning, such a distance is normally the one between a stationary device (anchor) and a mobile device whose position is of interest. While multiple techniques and mediums can be used to achieve such distance measurements, one of the most common is using a radio-based approach. A modern smartphone normally utilizes several different radio-based technologies such as Wi-Fi and Bluetooth daily, which makes ranging techniques based on such signals especially interesting. As such, multiple different techniques based on these popular technologies have been developed.

2.2.1 Received Signal Strength

Received Signal Strength (RSS) is defined as the power present in a received signal and ranging techniques based on this feature exploit the property that the intensity of an emitted signal decreases as the distance from the emission source increases [16]. Since RSS measurements are used in both Wi-Fi and Bluetooth communication, most existing techniques can be used with either of the two. By modeling the path loss of a signal in the environment of interest and using known properties of the antennas involved, RSS can be used to calculate the distance a signal has traveled. In free space, the path loss expressed as a power ratio can be calculated using the equation:

$$L = 20\log(\frac{4\pi R}{\gamma})$$

Equation 2.1

where R is the separation of the two antennas and γ is the wavelength [17]. Using Equation 2.1, the distance can be derived when the transmit and receiver power is known.

However, modeling indoor environments as free space is problematic as walls, doors, furniture, and people can cause significant signal loss and cause severe multipath. In many indoor environments, direct LoS between receiver and transmitter is often a rarity. Accurately modeling propagation on a specific site can be done through advanced software using ray-tracing, but changing so much as the location of a piece of furniture could affect the accuracy of the model [15]. Therefore, site-general models using statistical predictions of path loss are more often used, such as the Log-Distance Path Loss Model:

$$L_{TOTAL} = PL(d_0) + N \cdot log_{10}(d/d_0) + X_S$$

Equation 2.2

where $PL(d_0)$ is the path loss at a reference distance d_0, usually defined as the free-space loss at 1m, N is the path loss distance exponent, d is the distance, and X_S is a Gaussian random variable with 0 mean representing noise [15]. To better fit the environment of interest, both $PL(d_0)$ and N can be determined empirically.

Many other more or less advanced models exist, but due to the complex characteristics of indoor environments from a signal propagation point-of-view discussed above and in Section 2.1, distance measurements based on RSS are difficult to make robust enough. In addition to this, another limitation of Wi-Fi RSS based ranging methods is that many factors in hardware and software design of the Wi-Fi chip in the receiving device have been found to affect the reported signal strength. This means that different devices will achieve a varying degree of accuracy for RSS ranging [18].

2.2.2 Time difference of arrival

Another commonly used ranging technique used for localization uses the Time Difference of Arrival (TDoA) between multiple stationary receivers of a signal transmitted from the device whose position is unknown. Delay in the arrival of the signal corresponds to additional propagation time, which can be translated to a difference in distance from the transmitter, independent on the actual transmission time. For every pair of receivers $R_{i,j}$, a hyperboloid can be calculated corresponding to the possible locations of the transmitter that would cause such a difference in time of arrival, i.e., the range difference between the receivers used in the TDoA measurement and the hyperboloid must be constant. The equation of such a hyperboloid is given by:

$$\begin{aligned} H_{i,j} = {} & \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \\ & - \sqrt{\left(x_j - x\right)^2 + \left(y_j - y\right)^2 + \left(z_j - z\right)^2} \end{aligned}$$

Equation 2.3

where (x_i, y_i, z_i) and (x_j, y_j, z_j) represent the coordinates of the fixed receivers i and j, and (x, y, z) represent the coordinate of the target [14]. For this technique to work, it is required that the clocks of the receivers are accurately synchronized [19].

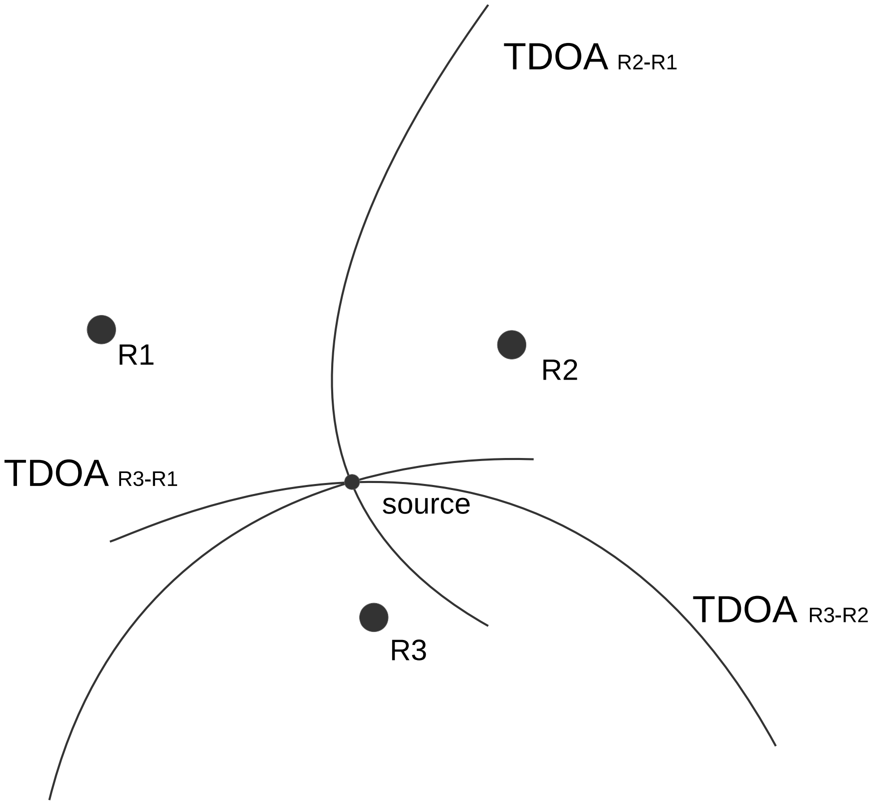


Figure 2.1 Illustration of the TDOA ranging technique

As seen in Figure 2.1, by using at least two TDoA measurements, an intersection between the hyperboloids can be calculated, and the position of the signal source determined.

2.2.3 Time of flight

Many Wi-Fi-based ranging techniques use Time of Flight (ToF) measurements to estimate distances. By clocking the time it takes for a signal to travel between a transmitter and a receiver, the distance between them can be calculated as the signal propagation delay is proportional to the distance traveled [20]. As radio signals travel at the speed of light (c) the distance (d) between a transmitter and receiver can be calculated with a simple formula based on the signal propagation delay (t):

$$d = c \cdot t$$

Equation 2.4

where the signal propagation delay t can be calculated through $t = t_{arrival} - t_{transmit}$ by determining the two timestamps at each of the two devices respectively. A problem that arises with this approach is that the clocks on the receiver and transmitter need to be carefully synchronized to get accurate results of the delay. A clock synchronization error of just 1ns leads to an error of 0.3m. This is hard to accomplish to the level of precision that is required a better approach is to instead measure the round-trip-time-of-flight (RToF) delay. By measuring the time it takes for a signal to be sent plus the time it takes to receive a corresponding acknowledgment from the receiver, the complex task of clock synchronization can be omitted by handling the timing on the same device. The distance between the two devices can then be calculated by instead using the formula

$$d = \frac{RTT - T_{PROC}}{2} c$$

Equation 2.5

where RTT is the measured time it takes for a packet to be sent from one device to the other and then back again, and T_{PROC} is the overhead time spent processing the packet on both devices. Earlier work suggests that processing time uncertainty is usually the main source of error for this method, while multipath propagation also can affect the measurements [21]. Existing ranging techniques based on RToF have different ways of dealing with this, varying from specialized hardware components, different communication flows, and methods of time measurement [20].

2.2.4 Wi-Fi RTT

In 2016, the IEEE 802.11 working group approved amendment 802.11-REVmc2 for the Wi-Fi standard [7], which in this work is referred to as Wi-Fi RTT. The new standard includes an extension of the IEEE 802.11v timing measurement protocol, more specifically, the fine timing measurement protocol (FTM) [4]. FTM enables a pair of Wi-Fi devices to estimate the distance between them by measuring the RTT.

When compared to earlier versions of the timing protocol, FTM has the potential to achieve a higher accuracy as it has stricter requirements for the resolution of timestamps [4].

As explained by Ibrahim et al., the protocol operates as follows [6]:

1. An initiator (the device that initiates the FTM process) initiates the measurement procedure by sending an FTM request to the corresponding AP.
2. If the AP supports the FTM protocol as a responding device (a responder) it will acknowledge the ranging procedure which will then continue.
3. If the AP agrees to continue it will start to send FTM messages and wait for their ACKs.
4. The RTT between the two devices is estimated by the responder, based on the transmission timestamp of the FTM message and the reception of its ACK. The RTT measurement procedure (steps 3 - 4) can be repeated multiple times.

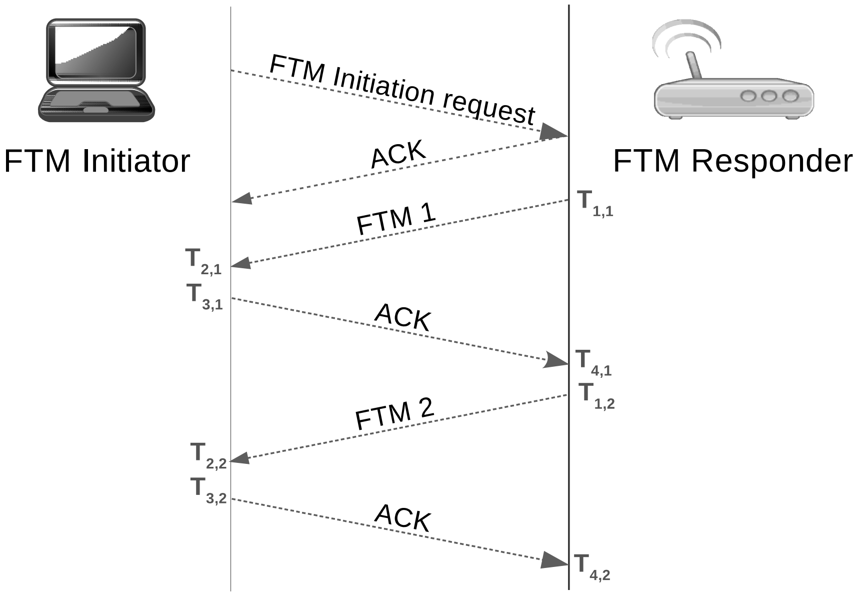


Figure 2.2 Illustration of the FTM ranging procedure using two measurements

By using the timestamps $T_{1,k} - T_{4,k}$ illustrated in Figure 2.2 the RTT can be calculated through the following equation:

$$\mathrm{RTT} = \frac{1}{n}\left(\sum_{k=1}^{n} T_{4,k} - \sum_{k=1}^{n} T_{1,k}\right) - \frac{1}{n}\left(\sum_{k=1}^{n} T_{3,k} - \sum_{k=1}^{n} T_{2,k}\right)$$

Equation 2.6

where n is the total number of FTM – ACK exchanges [6]. As timestamps $\mathrm{T}_{2,\mathrm{k}}, \mathrm{T}_{3,\mathrm{k}}$ are used to measure the processing time of the FTM message on the initiating device, the protocol can be considered a method for ToF measurements. As discussed in Section 2.2.3, many of the problems with methods for measuring ToF arise from the fact that T_{PROC} can be hard to determine. With the approach described above and hardware with high precision clocks, most of these problems can, in theory, be mitigated, and reliable ToF measurements can be achieved. When operating at a bandwidth of 80MHz, Wi-Fi RTT ranging with an Android smartphone is expected to have a 90% CDF error of 2m. If this is not the case, the device should be calibrated. The Android Open Source Project (AOSP) has specified a calibration guide that should be followed until the ranging measurements perform according to the expected tolerance [10].

2.3 Position estimation techniques

Many different techniques for position estimation have been proposed, and in this section four of the most common methods are presented to provide some background to the field of indoor positioning.

2.3.1 Trilateration

Trilateration algorithms use distances measured through some method (for example, one of the ranging techniques described in Section 2.2) to reference points with known locations to determine an estimation of an unknown position [16]. With measured distances to three reference points, a trilateration algorithm can calculate a 2-D position whereas four or more distances can be used to determine a 3D-position.

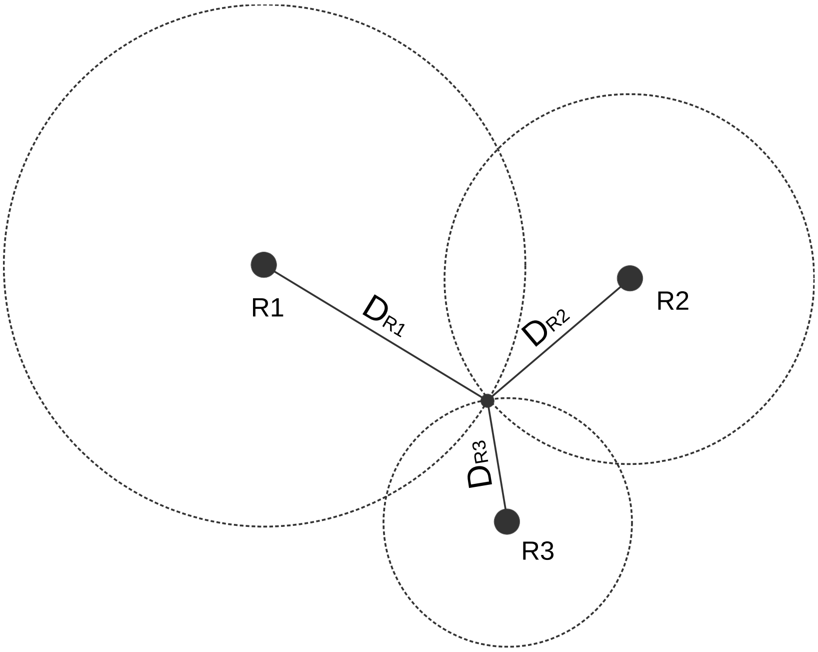


Figure 2.3 Illustration of trilateration using ideal measurements

If the measurements are exact, the circles constructed through the range measurements will intersect in exactly one point, as illustrated in Figure 2.3. The unknown position can in such cases, be calculated by solving a linear equation. Using *N* number of reference points, *N* circles can be constructed through

$$d_i = (x - x_i)^2 + (y - y_i)^2 \qquad ,i = 1,2,3 \ldots N$$

Equation 2.7

with the center point (x_i, y_i) and radius d_i determined from ranging measurements. Given the *N* circles defined in Equation 2.7, the equation for *i* = 1 is subtracted from the others:

$$(x - x_i)^2 + (y - y_i)^2 - (x - x_1)^2 + (y - y_1)^2 = d_i^2 - d_1^2 \qquad ,i = 2,3 \ldots N$$

Equation 2.8

which can be simplified to obtain the linear equation system for finding (x, y):

$$-2x(x_i - x_1) - 2y(y_i - y_1) + x_i^2 - x_1^2 + y_i^2 - y_1^2 = d_i^2 - d_1^2 \qquad ,i = 2,3 \ldots N$$

Equation 2.9

Equation 2.9 can be reformulated into vector-matrix notation on the form $Ax = B$:

$$2\begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \vdots & \vdots \\ x_M - x_1 & y_M - y_1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r_2^2 - d_2^2 - r_1^2 + d_1^2 \\ r_3^2 - d_3^2 - r_1^2 + d_1^2 \\ \vdots \\ r_M^2 - d_M^2 - r_1^2 + d_1^2 \end{bmatrix} \qquad , r_i^2 = x_i^2 + y_i^2$$

Equation 2.10

which for $N \geq 3$ yields a unique solution $x = A^{-1}B$ [22]. However, in practical applications, there is most likely a measurement error involved, and ideal measurements that intersect in exactly one point can never be assumed. In such cases, the method above instead yields the Least-Squares solution.

2.3.2 Triangulation

While very similar to trilateration, triangulation instead uses measured angles to reference points to determine an unknown location. In general, two-dimensional angulation requires two angle measurements and one distance measurement (e.g., the distance between two reference points) [16] as illustrated in Figure 2.4.

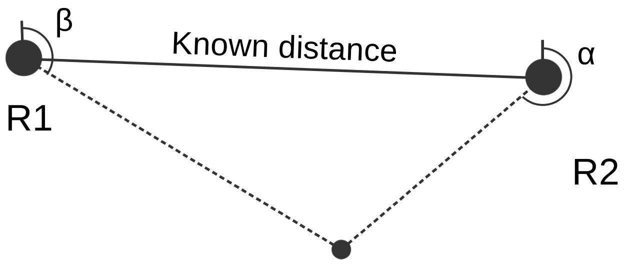


Figure 2.4 Illustration of 2-D angulation using two angles α and β as well as a distance between two reference points

For a three-dimensional location, an azimuth measurement (the angle between a vector projected onto a reference plane and a point of interest on the said plane) is needed in addition to the requirements for a two-dimensional position. With this information, the unknown location can be determined through trigonometric relations. In indoor positioning techniques, the Angle of Arrival (AoA) is usually used for this calculation [13]. The AoA can be determined by devices that, for example, have multiple antennas with known separation. By measuring the TDoA for a signal between the different antennas, the AoA can be determined [16]. However, in indoor locations, the true AoA can be difficult to accurately determine as a result of NLoS propagation effects.

2.3.3 Kalman filter

The Kalman filter is a widely adopted method for optimally estimating the state of a system using noisy measurements. To create a deeper understanding of this method, an example application is first presented that can be used to put the theory into context. Then, the background and mathematical theory behind the method is presented.

Example application

A classic example often used to explain the usefulness of the Kalman filter, is the problem of determining the location of a car driving on a highway [23]. The car has multiple instruments that could be used for this purpose, such as a sensor measuring the acceleration of the vehicle, an odometer that measures the travelled distance, and a GPS receiver that can estimate a position of the car. In this case, the first two instruments provide accurate information with a high frequency, although the information is only relative. That is, the starting position of the car has to be exactly known, in order to determine the current position by only using the acceleration sensor or the odometer. Moreover, the accuracy of the distance information reported by the odometer is affected by tire pressure and could therefore contain errors. Likewise, obtaining a distance from acceleration requires calculating the double integral, which is prone to accumulate small errors over time, causing the distance estimation to drift. The GPS receiver can be used to receive absolute position estimates but is often noisy with a varying uncertainty. When entering a tunnel, for example, the accuracy of GPS is greatly reduced. The frequency at which position estimates can be obtained is also limited.

So, all instruments have benefits and drawbacks, but each could potentially contribute with small pieces of information to yield a good estimate of the current position. This is where the Kalman filter comes in. The Kalman filter considers all pieces of information, together with estimates of how reliable the information is, to yield the best possible estimate of the car position.

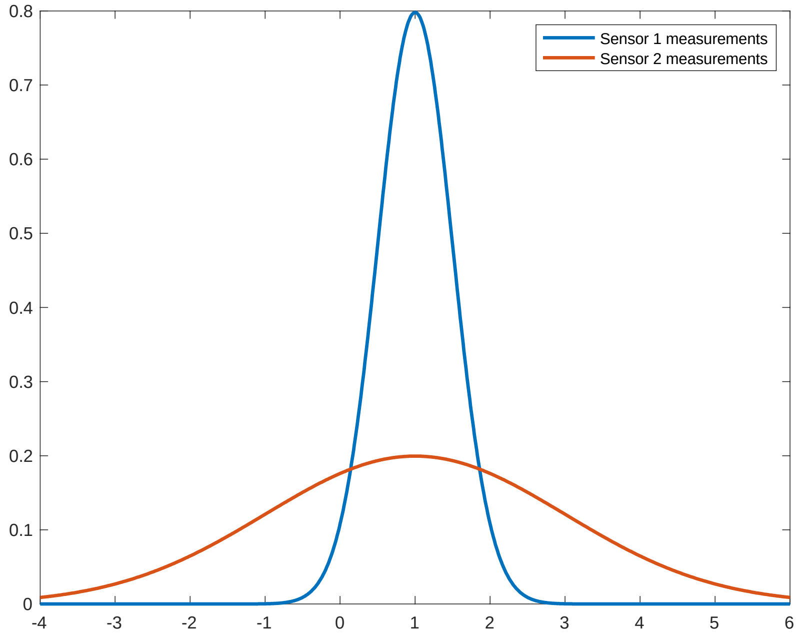


Figure 2.5 Example normal distribution of measurements from two different sensors.

Sensor measurements often contain noise, and one way of modelling this noise is by making multiple measurements in the same setup and look at the variance of the measurement distribution. Consider the two arbitrary sensors in Figure 2.5 as an example and let 1 be the correct value that both sensors should ideally measure. Although both sensors have the correct mean measured value of 1, the output of Sensor 1 would be trusted to a higher degree than Sensor 2 as it’s variance is much smaller. The two sensors could, for example, represent the GPS receiver when operating inside a tunnel (Sensor 2) and outside the tunnel (Sensor 1).

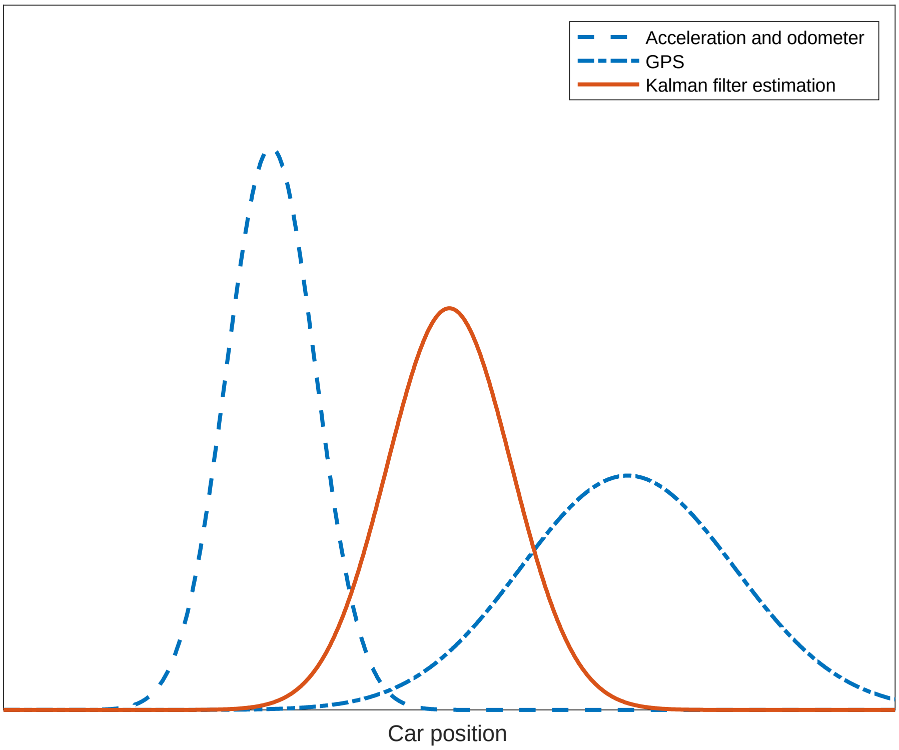


Figure 2.6 Illustration of the Kalman filter position estimation of the car example.

The Kalman filter considers the modeled variance of different sources of information and fuses the measurements together using the uncertainty in the calculations. In Figure 2.6 above, the sources of information have different uncertainty and different position estimations but are both incorporated in the final Kalman filter estimation. The Kalman filter could also be used as an alternative to trilateration. The filter would, in this case, take a set of distance estimations to reference points as measurements, and estimate the position that would lead to such measurements.

Background and mathematical theory

The Kalman filter is named after Rudolph E. Kalman who published a paper introducing the method in 1960 [24]. Since then, the method has become widely popular for use in various applications. Many such applications have been developed within the field of navigation and localization including the navigation system on the Apollo spacecraft and user location estimation using hand-held GPS receivers [25]. The main idea of the Kalman filter is to continuously update the mean and covariance of a state estimation (in the example above, the position of a car) through frequent measurements passed through a linear filter so that the covariance of the estimation itself is minimized [24]. Measurements in a real-world system are more than likely to contain noise and if this noise is gaussian with known covariance, the Kalman filter has been proven to provide optimal state estimation [24]. Since the first publication 1960, many different variations and extensions of the Kalman filter have been proposed and with them as many different mathematical notations. In this thesis, notations from the book *Adaptive Filtering and Change Detection* by Fredrik Gustafsson will be used [26].

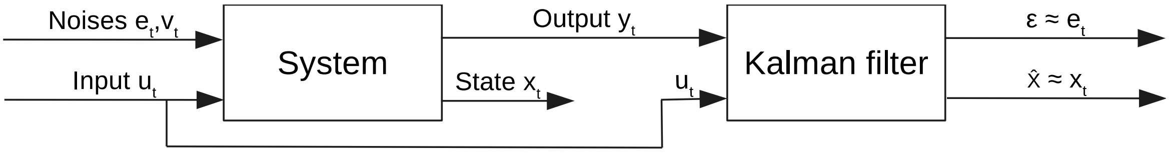


Figure 2.7 System diagram with signal definitions of a system using a Kalman filter for state estimation. [26]

A general state-space model for the Kalman filter using the signal definitions of the diagram in Figure 2.7 can be expressed as:

$$x_{t+1} = \mathrm{A_t x_t + B_{u,t} u_t + B_{v,t} v_t}$$
$$y_t = C_t x_t + e_t$$

Equation 2.11

where x_t is the state and y_t is the output of the system [26]. In Equation 2.11, A, B, and C are known matrices representing different features of the system modeling. A represents the state transition model, which takes a previous state and predicts a new state. The B matrix represents the control-input model, which is applied to the input signal u to transform the input to a state prediction contribution. In the example presented earlier, the state transition model and control-input model, would most likely use the previous position estimate of the car together with odometer and acceleration readings to predict the new position. The C matrix represents the observation model, which maps the true state x to a vector of measurements. For the noise signals, only their approximated covariances are known. In the example above, the observation model would be responsible for transforming a position to a GPS reading.

The Kalman filter can be implemented as a recursive function consisting of two parts; time update and measurement update. Given the previous estimations, the predicted state estimate $\hat{x}_t$ and its predicted covariance P_t (uncertainty) are calculated in the time update phase through:

$$\hat{x}_{t|t-1} = A_t\hat{x}_{t-1|t-1} + B_{u,t}u_t$$
$$P_{t|t-1} = A_tP_{t-1|t-1}A_t^T + B_{v,t}Q_tB_{v,t}^T$$

Equation 2.12

where Q is the covariance matrix for the process noise v. In the measurement update phase, measurements of y are used to correct the estimation. In this step, three quantities are calculated and used [26]:

$$\mathrm{\epsilon_t = y_t - C_t}\hat{x}_{t|t-1}$$

Equation 2.13

$$S_t = C_tP_{t|t-1}C_t^T + R_t$$

Equation 2.14

$$K_t = P_{t|t-1}C_t^T\left(C_tP_{t|t-1}C_t^T + R_t\right)^{-1} = P_{t|t-1}C_t^TS_t^{-1}$$

Equation 2.15

Equation 2.13 is known as the innovation and is defined as the difference between the observed measurement at time t and the predicted measurement. Equation 2.14 is the covariance matrix of the innovation, where R is the covariance matrix of the measurements. Equation 2.15 is the Kalman gain. The Kalman gain is based on the current covariance estimates and can be seen as a parameter weighting the most recent measurements to the state estimate. A low Kalman gain indicates that measurements are noisy and the current state estimate is incorporated more into the new estimate, whereas a high Kalman gain indicates that the measurements should be more trusted [25]. With these three quantities and the predictions made in the first phase, the measurement update phase of the algorithm is then carried out as:

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t\epsilon_t$$

$$P_{t|t} = P_{t|t-1} - K_t S_t K_t^T$$

Equation 2.16

The state estimate $\hat{x}_{t|t}$ has been proven to be optimal if the covariance matrices Q and R are used.

The main drawback of the original Kalman filter is its limitation to linear models [24]. This means that if either the state transition model, the control-input model, or the observation model is non-linear, the Kalman filter is not applicable. For such system models, many extensions of the Kalman filter have been proposed. A typical non-linear discrete state-space model for the Kalman filter is:

$$x_{t+1} = f(x_t) + v_t$$
$$y_t = h(x_t) + e_t$$

Equation 2.17

where f and h are the nonlinear state transition and measurement functions. One of the most common non-linear Kalman filters extensions is the Extended Kalman Filter (EKF). The main idea of the EKF is to linearize the non-linear functions h and f in Equation 2.17 with first-order Taylor-expansions by substituting the linear transformation matrices used in the original Kalman filter with the Jacobian matrices of h and f, respectively. The Jacobian of a multivariate function is a row matrix with all first-order derivatives of the function, which is also the transpose of the function's gradient. This type of linear approximation works well for quasi-linear transformations when the Jacobian matrix calculations are relatively non-expensive [27]. While these two conditions are fulfilled in many different applications, there are situations when the EKF performs poorly [24].

When the models are highly non-linear, the linearization used in EKF might lead to significant errors. In such situations, the Unscented Kalman Filter (UKF) might be a better choice. It is based on the idea that a probability distribution is easier to estimate than a nonlinear function [28]. This is done by letting a set of carefully chosen, so-called, sigma points pass through the nonlinear transformations to capture the effect of the model nonlinearities on the means and covariances during the filtration [24]. This way, the characteristics of the resulting Gaussian distributions are captured without linearization. There are multiple methods for selecting sigma points for the unscented transform, but one increasingly popular is a method proposed by Rudolph Van der Merwe in 2004, using three parameters α, β, κ to control how the sigma points are distributed and weighted [29]. Here, α controls the size of the sigma point distribution, β is a non-negative weighting term which can be used to incorporate knowledge of the distribution, and κ is a binary parameter that, when set to ≥ 0, guarantees positive semi-definiteness of the covariance matrix [30]. However, while UKF often performs better for highly non-linear problems, it also brings a higher computational cost as all sigma points calculations can be rather expensive [27].

2.3.4 Fingerprinting

A popular technique for indoor positioning that does not require any ranging measurement is fingerprinting. Instead of determining different distances or angles and estimating a location through trilateration/triangulation, location fingerprinting matches the fingerprint of some characteristics of a signal such as RSS to readings at known locations [13]. Fingerprinting methods usually build a database, which maps positions to a set of signal characteristics used as fingerprints during an initial training, or offline, phase [22]. During the online phase, the measured fingerprint is compared to the entries in the database of known fingerprints and the location corresponding to the best matching fingerprint is then determined as the device's position. A popular method is also to determine the *k*-nearest fingerprint neighbors and then estimate an average position based on those entries [22]. When compared to ranging using RSS as discussed in Section 2.2.1, fingerprinting using RSS measurements often yield more robust results as the NLoS propagation confusing the distance estimations generally is a less significant source of error for fingerprinting. As long as the signal characteristics are somewhat consistent in the area that is to be covered, multipath and attenuation effects should not affect the performance of an IPS using fingerprinting as much.

2.4 Empirical research in software engineering

When conducting empirical research, multiple different methods and research paradigms can be applied. There are two kinds of study paradigms that have different approaches to empirical studies; *exploratory* and *explanatory* research [31].

Exploratory research studies objects in their normal setting and observations made are the base for any findings of the study. *Explanatory* research is instead focused on quantifying a relationship or comparing different groups to identify a cause-effect relationship. This type of study is usually associated with quantitative research as studies often are conducted through controlled experiments to yield quantitative data [31]. For quantitative research (i.e. research whose result can be quantified for comparison and statistical analysis), there are mainly two methodologies normally used; *case study* and *experiment* [32].

A *Case study* is performed to study a single entity or phenomenon in its real-life context and typically may be hard to clearly distinguish from its environment [31]. The level of control is often lower than in experiments and is considered an observational study rather than a controlled study [33]. *Experiments* are highly controlled studies where one or more variables are systematically manipulated, and all other variables are fixed. The effect of the manipulations made is measured and used to draw conclusions [32]. A precondition to any controlled experiment is a clear hypothesis drawn from related theory to which the result can be compared to. The design of the experiment should be decided with the hypothesis in focus, such as which variables to include in the experiment and how they can be measured [33].

C. Wohlin et al. describe the general process when designing and conducting a controlled experiment [34]. The first part of this process is *scoping*. In this step, the hypothesis should be clear and with this in focus, the objectives and goals of the experiment must be clearly defined. It is important to consider *what* should be studied, for *what* purpose, and in *which* context. The next step is the *planning* phase. In this phase, the context of the experiment is detailly defined, variables and their possible values determined, and the experiments are designed. In the planning phase, it is also important to consider the validity of the expected results by reflecting on the generality of the findings and how the values of context-specific variables might affect the result.

After the scoping and planning of the experiment has been completed, the *operation* can begin. This consists of three steps; preparation, execution, and data validation. In the preparation step any preparations that must be taken to yield valid data are taken, such as setting up the environment and creating data collection forms. After the execution, it is important to try to verify that the collected data is correct and properly reflects the design of the experimentation.

After the experiments have been conducted and their data collected, it is time for *analysis and interpretation*. An important first step here is to try to understand the data collected by visualization which will help to provide an informal interpretation. Data set reduction is thereafter performed if deemed necessary because of, for example, redundant data. When the data has been post-processed, a hypothesis test is performed through an appropriate method based on the type of data and pre-defined metrics. The last step of the experiment process is to document the results to make sure that the results are taken care of in an appropriate way, depending on the purpose and context of the experiment.

In this thesis, an explanatory approach was taken, and controlled experiments were conducted to answer the research questions posed. For a detailed description of the experiments conducted, see Chapter 3 and Section 6.5.

Chapter 3 Wi-Fi RTT Characteristics

Wi-Fi RTT is the nickname for a new standard of the Wi-Fi protocol, IEEE 802.11mc, that includes the FTM protocol (see Section 2.2.4). To understand the characteristics of Wi-Fi RTT distance measurements and how the technology behaves in different situations, explorative experiments using the technology were conducted as part of a practical pre-study. In this chapter, the results of these experiments are presented. Four FTM-enabled access points were used, from here on referred to as AP 1-4. The hardware specification of these APs and software used to collect the data are specified in Section 6.1. A total of four experiments were conducted with different sight conditions (LoS or NLoS) and motion types (stationary or mobile):

Motion type Sight condition	Stationary	Mobile
LoS	Experiment 1	Experiment 3
NLoS	Experiment 2	Experiment 4

Table 3.1. Overview of the conducted pre-study experiments.

3.1 Stationary measurements

To explore how well the reported Wi-Fi RTT distance corresponds to the true distance, multiple stationary measurements were conducted with known separation. The measurement procedure was carried out similar to the AOSP Wi-Fi RTT calibration guide [10]. An access point was mounted 20cm above the floor and the phone was mounted on a movable mount at the same height. The top of the phone was facing the AP during the entire procedure. The AP and device were then separated with a known distance (a ground truth). Starting at 1m, the separation was gradually increased with steps of 1m up until a separation of 10m was reached. 100 measurements were made at each separation. The experiment was conducted in both LoS and NLoS conditions and was performed one time for each AP.

3.1.1 LoS measurements

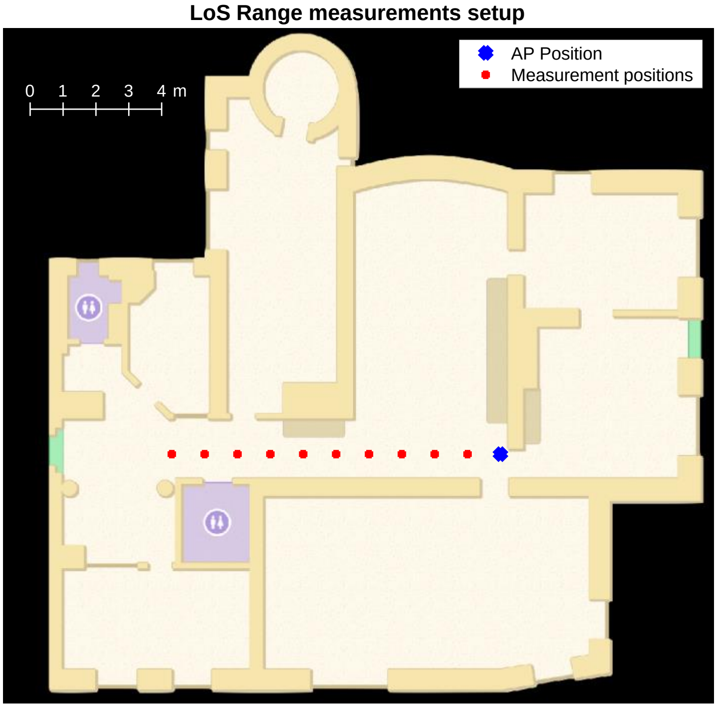


Figure 3.1 Setup for range measurements made in LoS conditions

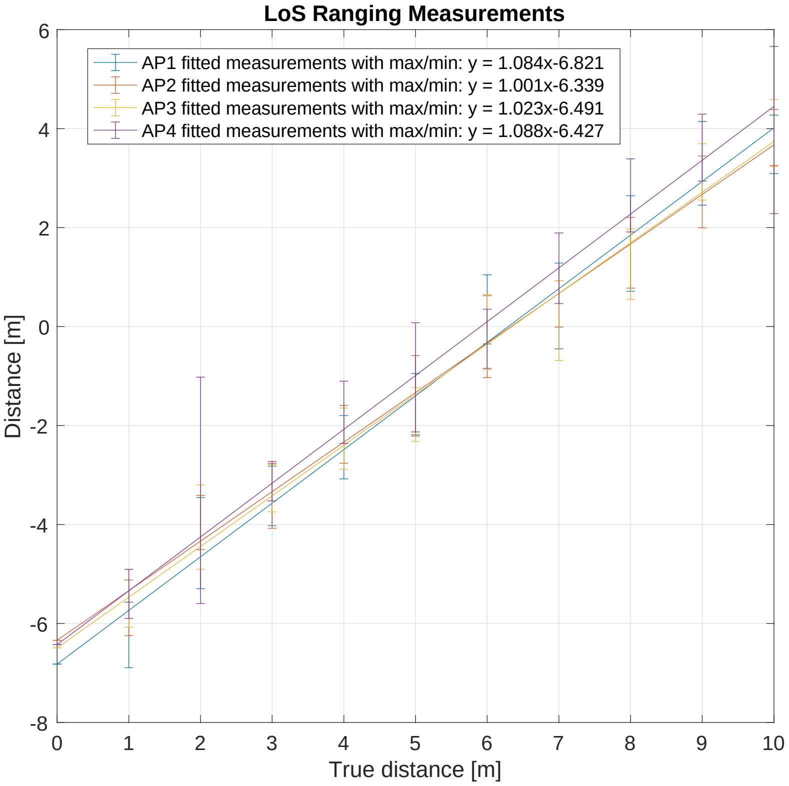


Figure 3.2 Ranging measurements in LoS conditions

The first experiment was conducted in an unobstructed indoor corridor with the measurements distributed as illustrated in Figure 3.1. The resulting fitted lines of the measurements (see Figure 3.2) show that ranging measurements have a constant negative error of ~6-7m. Although the result differs somewhat between the access points, there are no alarmingly large differences between the results. This indicates that there is no need for individual calibration values for each access point. A control-measurement with one of the APs was also performed using the same procedure. The result of this was within 5% of the first measurement for the same AP showing that the result is somewhat reproducible. One thing that can be noted is that the slope of all fitted lines is greater than 1 which is the expected value. If there only was a constant error (a bias) always present in all measurements, the slope would be 1. This does not seem to be the case however, and this could be explained by multipath or other propagation effects which leads to a higher discrepancy in the measured distance at some fixed distances.

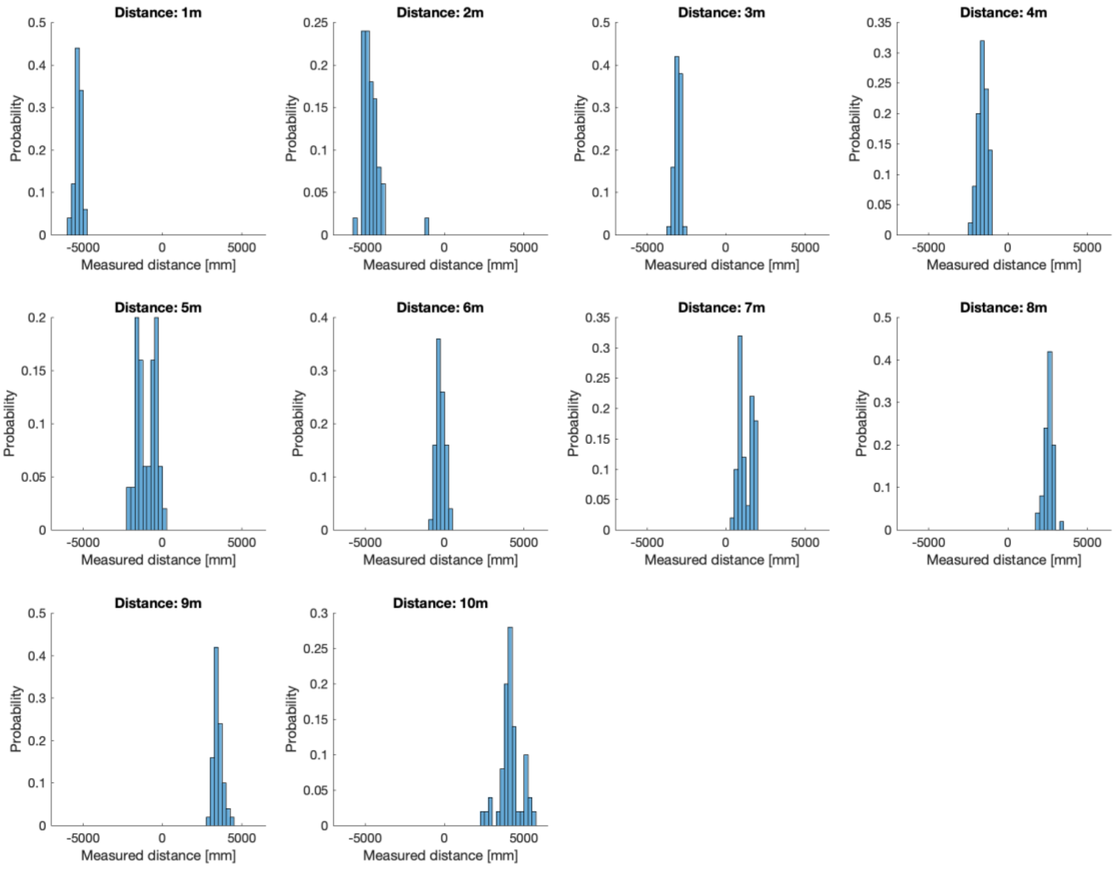


Figure 3.3 Distribution of measurements at each measurement location for AP 4.

For AP 4, which had the steepest slope, there are multiple such locations. For example, as can be seen in Figure 3.3 the distribution has two peaks at distances 5m and 7m, indicating two dominant paths which cause a large variation in the measurements. At 10m, there is one larger and one smaller peak, and the variation is significantly bigger than for most other measurement locations. Multipath propagation will lead to an increased measured distance. Also, the likelihood of multipath propagation increases with distance. Therefore, the slope will likely be shifted higher than 1 if multipath propagation effects are present.

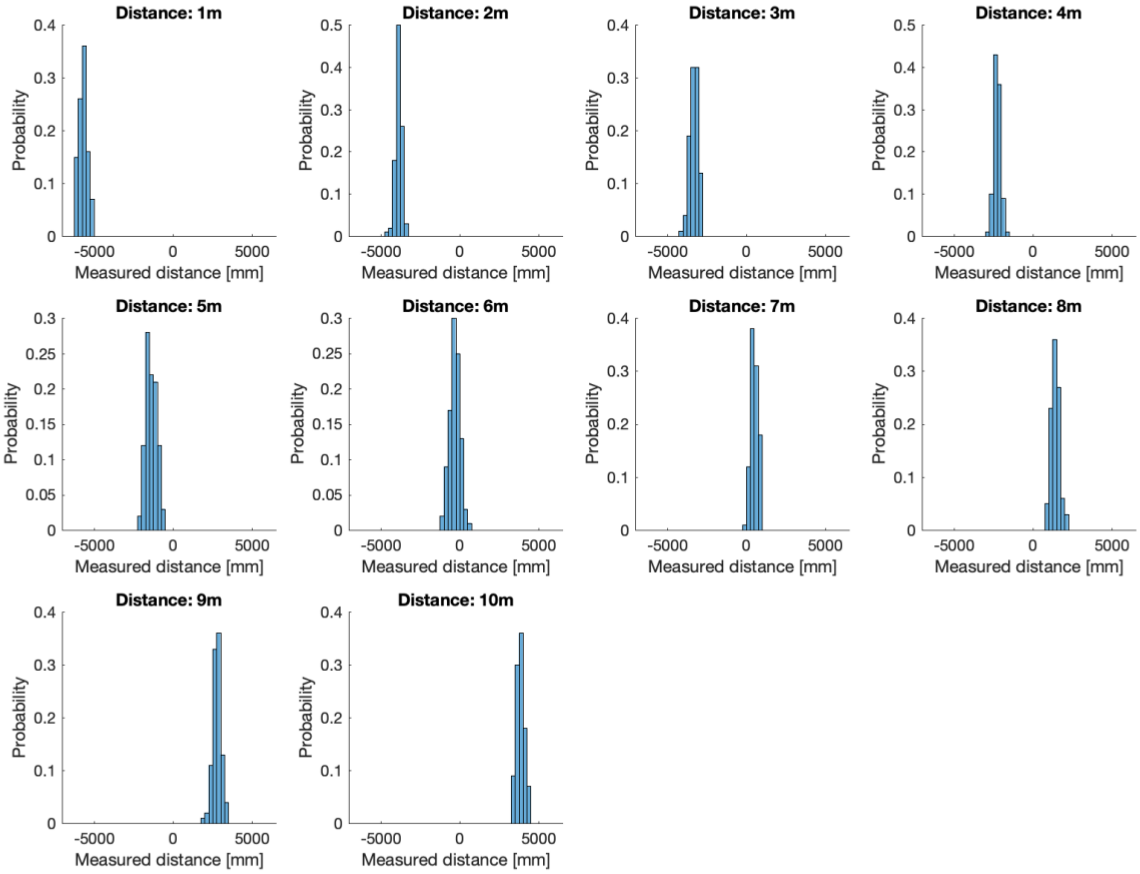


Figure 3.4 Distribution of measurements at each fixed distance for AP 2.

When looking at AP 2, which had the slope closest to 1 (1.001) in Figure 3.2, the distribution of the measurements looks much more concentrated. There are no obvious double peaks, and the variance is small and quite constant when compared to AP 4. This supports the theory of propagation effects causing the steeper slope. This is, however, also troublesome, as the measurement procedure was identical to that of the other APs and was performed in the same static environment with only minutes in between. As some small differences in the measurement positions are likely to have occurred between measurements, these could very well be the reason for the differences in the measurement distributions. This is not a desirable behavior as it means small displacements of the smartphone can lead to large measurement differences.

To rule out the option of specific APs causing measurement deviations, the same measurement procedure was also performed outside in an open field where unwanted multipath propagation effects should be minimal.

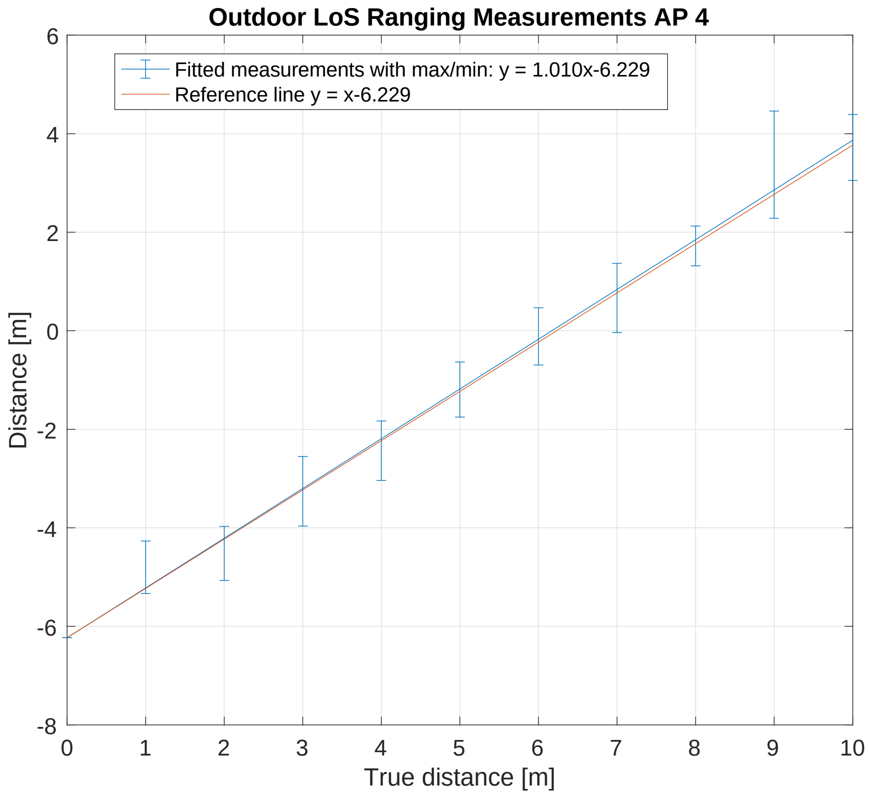


Figure 3.5 Outdoor ranging measurements for AP 4 and a reference line with the expected slope of 1.

As can be seen in Figure 3.5, the results in this environment were much better for AP 4, with a slope of only 1.01.

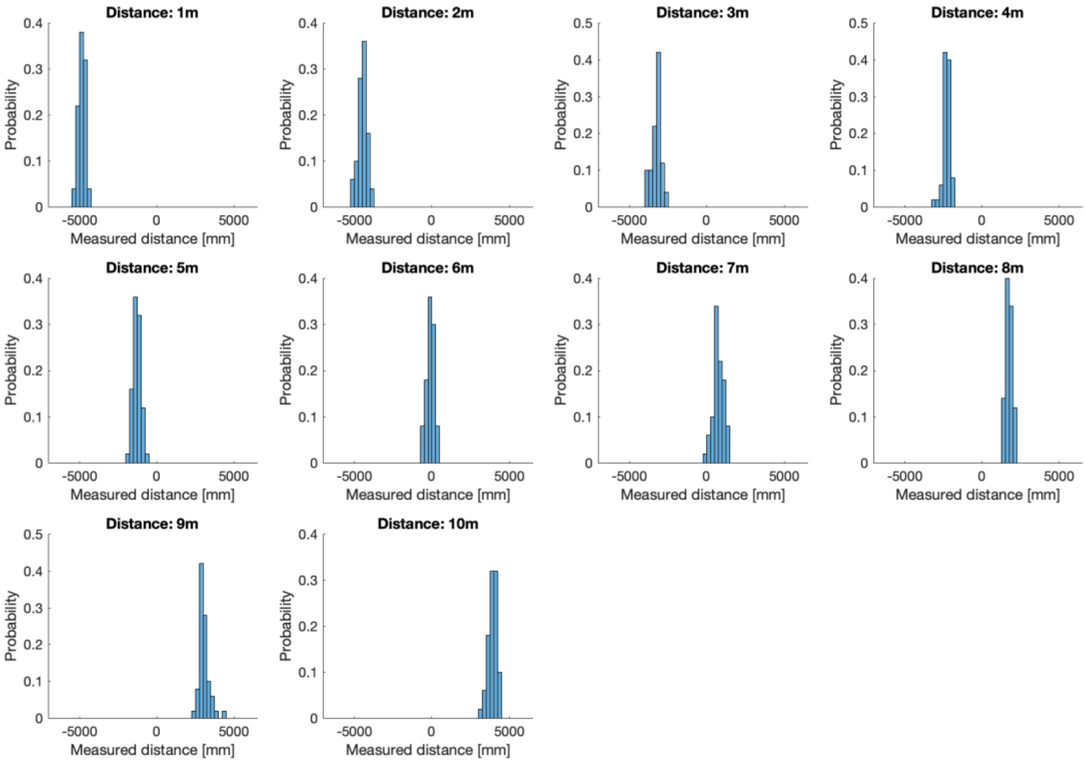


Figure 3.6 Distribution of outdoor LoS ranging results for AP 4.

The measurement distributions were also much better in this environment, with very small variance, also supporting the theory of unwanted propagation effects causing overestimated distance measurements. This experiment also indicates that the increased slope previously measured for some access points is most likely a result of the test environment, and not the access point themselves.

3.1.2 NLoS measurements

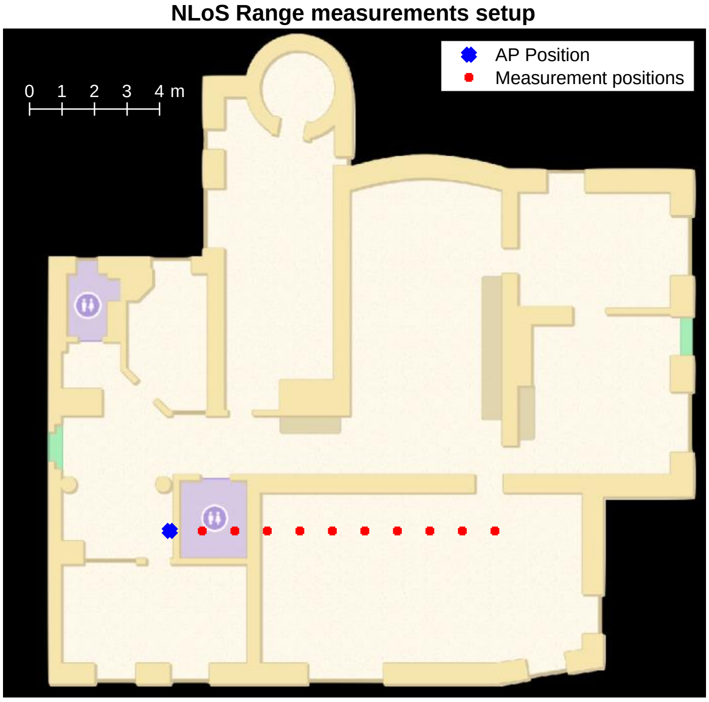


Figure 3.7 Setup for range measurements made in NLoS conditions.

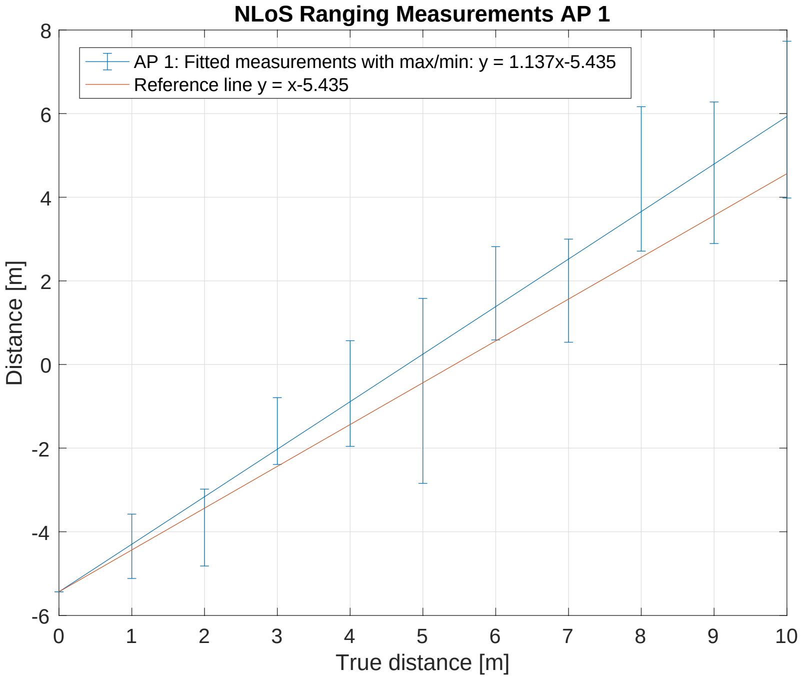


Figure 3.8 Ranging measurements in NLoS for AP 1 and a reference line with the expected slope of 1.

To understand how the RTT ranging behaves when a LoS path is not present, a similar experiment was conducted using the NLoS setup illustrated in Figure 3.7. As the results in LoS showed no major differences between the four APs, this measurement was performed using only one AP. The results show that the fitted line has a larger offset on the y-axis when compared to measurements made in LoS conditions which is expected since the NLoS conditions likely will lead to the radio waves having to travel a longer distance. However, in Figure 3.8, two measurement locations stand out from the rest. It was found that the wall between the access point and the first two measurement locations was a thin drywall, whereas the second wall was a thick concrete wall. The result indicates that the radio waves could propagate through the drywall at a higher degree than the concrete wall. As a clear majority of the walls in the experiment environment are made of concrete, the drywall was not very representative of the general NLoS condition. When excluding the first two measurement locations from the results, the result of the linear regression becomes quite different.

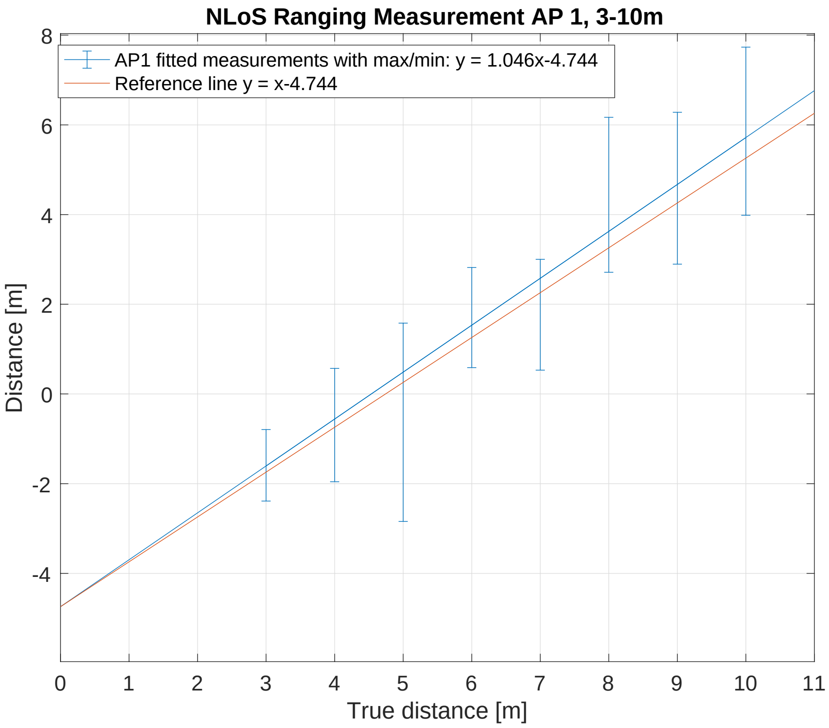


Figure 3.9 Ranging measurements in NLoS for AP 1 and a reference line with slope 1. Fixed distances 1 & 2 are excluded.

As seen in Figure 3.9, the slope is much closer to 1 and the offset on the y-axis becomes even larger when excluding the first two measurement locations.

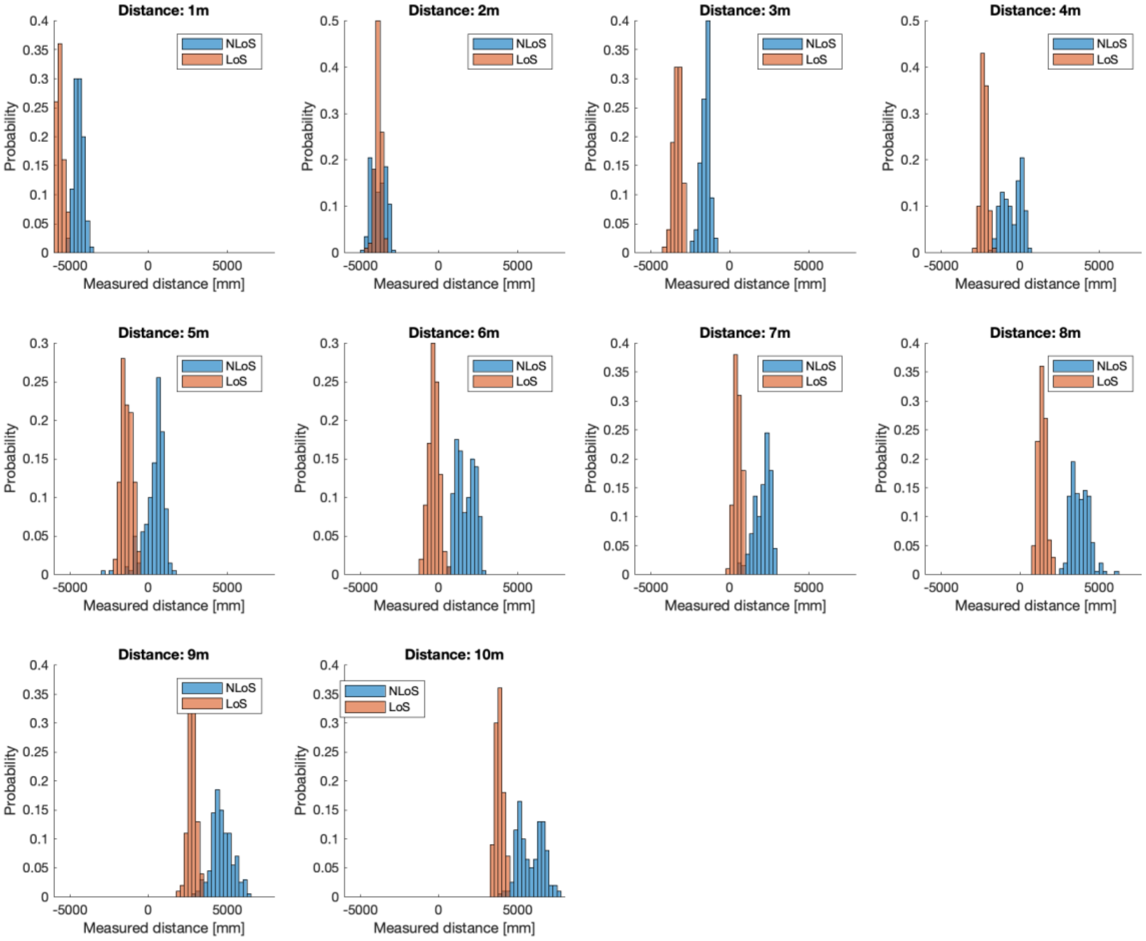


Figure 3.10 Distribution of measurements at each fixed distance in LoS and NLoS.

As seen in Figure 3.10, the distribution of the NLoS measurements at each measurement position has a bigger variance when compared to the same fixed distance in LoS and the occurrences of multiple peaks seem to be more common. The difference in mean measured distance does, however, seem to be a rather consistent ~2m, except for the first two measurement locations. This is also approximately the difference in offset on the y-axis when comparing the fitted lines in Figure 3.2 (LoS ranging measurements) and Figure 3.9 (NLoS ranging measurements).

3.2 Mobile measurements

Although the stationary measurements help to answer some questions about the ranging accuracy of Wi-Fi RTT, it does not represent a typical use-case for ranging which is often performed in mobile situations. To investigate if movement has any impact on the ranging performance, another test was conducted. In this test the mobile sensor logging application described in Section 1.5.4 was used to record ranging measurements every 0.4 seconds while following a certain path. The true distance at each measurement was then calculated at each ranging timepoint which was compared against the ranging result.

3.2.1 LoS measurements

The first test was conducted in the same indoor corridor used in Section 3.1.1, with constant LoS conditions.

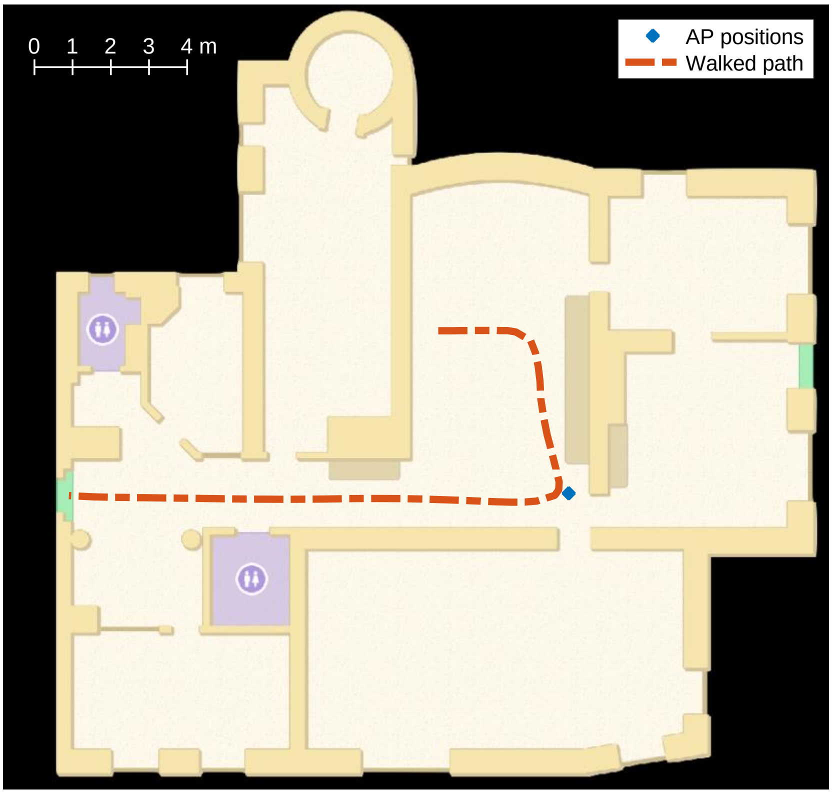


Figure 3.11 The path walked for the mobile LoS measurements.

As seen in Figure 3.11, one AP was placed in one end of the corridor. The path was walked in both directions while keeping the top of the phone aligned with the walking direction. This was done to represent a more typical use-case where different smartphone antenna directions relative to the AP are present and a body potentially is blocking the otherwise LoS path. The path was walked a total of eight times, one round-trip for each AP.

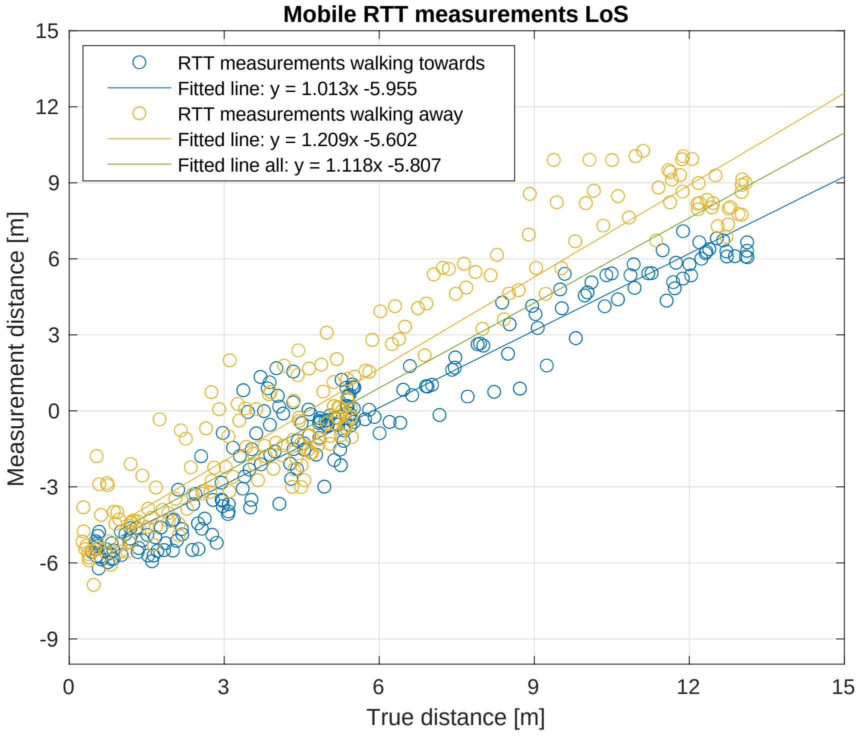


Figure 3.12 Result of the mobile LoS RTT ranging measurements.

In Figure 3.12, the result of this experiment is shown. When walking towards the AP, the fitted line is similar to the results gotten in the stationary measurements, but the offset is somewhat bigger. However, when walking away from the AP, the slope is instead close to 1.2, and the offset on the y-axis is even bigger. The reason why the slope differs this much is hard to determine, but the body blocking the signal when walking away from the AP may cause NLoS propagation effects which have a bigger effect at longer distances.

3.2.2 NLoS measurements

A similar experiment was also conducted in NLoS conditions. Here, the APs were placed in a room behind thick concrete walls with constant NLoS conditions along the path.

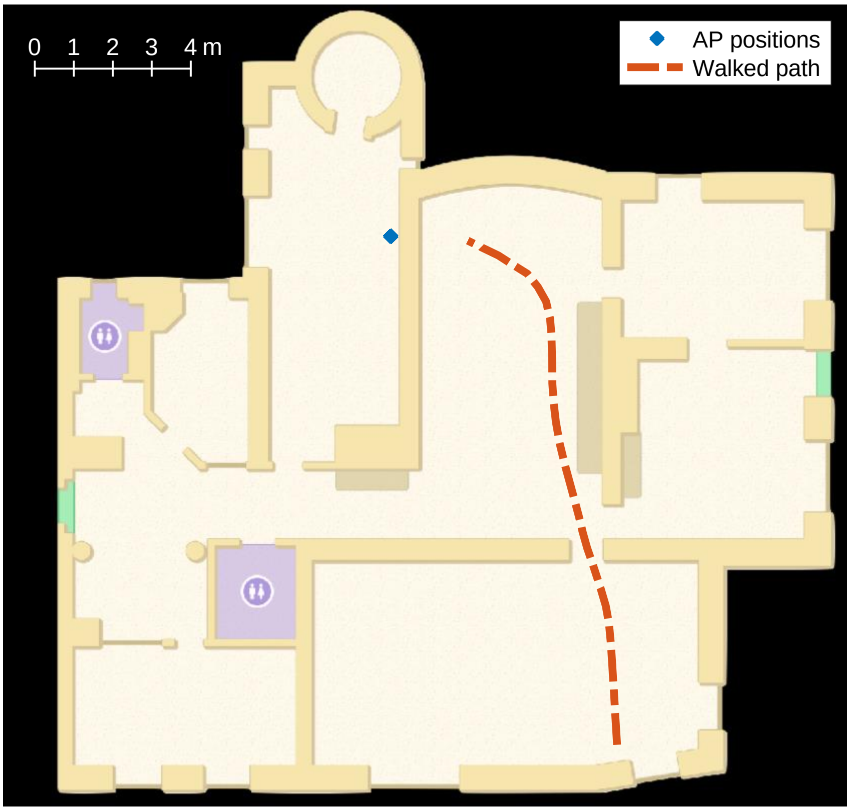


Figure 3.13 The path walked for the mobile NLoS measurements.

As seen in Figure 3.13, one part of the path had two concrete walls blocking the line of sight and the rest of the path only one. The path was walked in both directions for all APs.

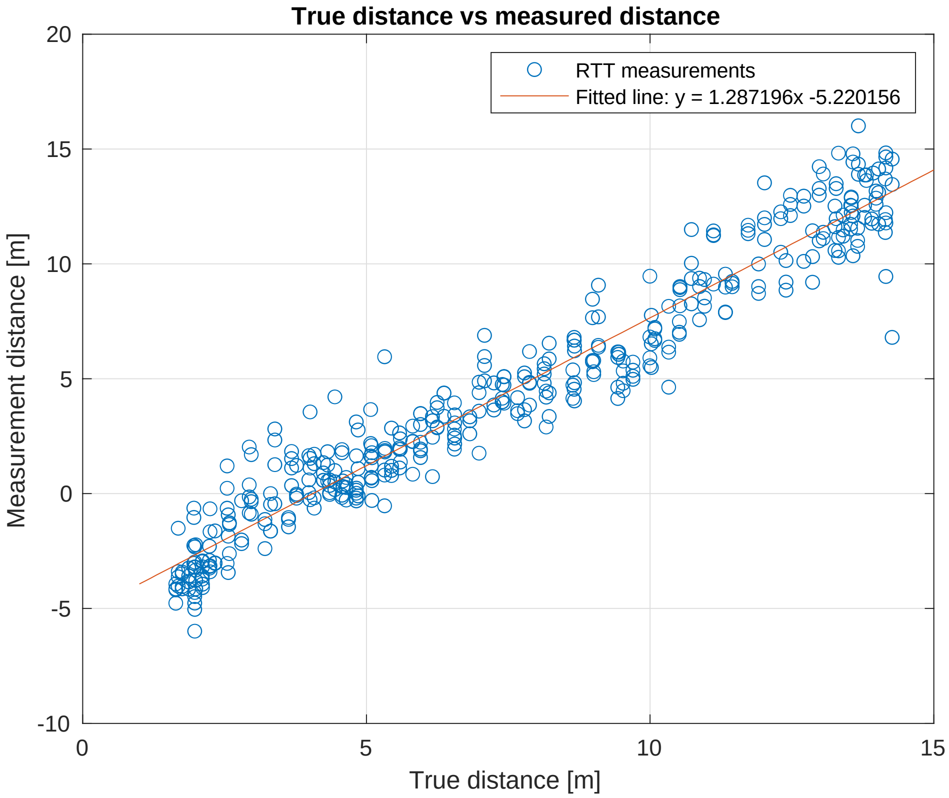


Figure 3.14 Result of the mobile NLoS RTT ranging measurements.

The result of the measurements is presented in Figure 3.14. As found earlier when walking away from an AP in LoS, the slope is substantially larger than 1 but with an even larger offset on the y-axis. This indicates that the body blocking the line of sight when walking away from an AP is the reason for the increased slope in that situation in the same way it appears to affect measurements made in "total" NLoS conditions.

3.3 Conclusions

In this practical pre-study of Wi-Fi RTT as a ranging technique, it was found that the hardware and software combination used resulted in a constant negative ranging offset. This is later addressed (see Section 6.1.3.1), as negative distances are not physically defined. It was also confirmed that measurements made when the receiver was in NLoS had a larger variance, as well a larger mean distance. A larger distance was also found to be reported in situations when the smartphone was moving. These characteristics are all considered when implementing the Wi-Fi RTT-based IPS in Chapter 6 .

Chapter 4 System Requirement Analysis

4.1 The goal of the system

The goal of the system is to provide an estimate of a smartphone device’s location. To accomplish the goal, the system should utilize Wi-Fi RTT technology as well as additional sources of information to provide the best possible estimate for the position of the device. The system should be developed in two parts; an online mobile application that can provide position estimates in real-time and record any information used for this purpose, as well as an offline application that can calculate the device positions through post-processing of recorded data.

4.2 The functional requirements

This section describes the functional requirements of the system.

4.2.1 Logging

- The online application shall be able to log and save to the storage of the device, the raw results of Wi-Fi RTT requests made .
- The online application shall be able to log and save to the storage of the device, any additional sources of information such as sensor data used in the position estimate .
- The online application shall be able to log and save to the storage of the device, position estimates made .
- All log entries shall include the type of data, timestamp, and additional data type-specific information.

4.2.2 Wi-Fi RTT

- The online application shall be able to detect nearby access points with Wi-Fi RTT capabilities.
- The online application shall be able to send ranging requests to discovered Wi-Fi RTT enabled access points and handle the result.

4.2.3 Device inertial sensor information

- The online application shall be able to detect when the user takes a step.
- The online application shall be able to listen for, and if necessary, process data from inertial sensors of the device.
- The online application should, at all times, be able to give an estimate of the current compass bearing of the device.

4.2.4 NLoS/LoS detection

- The system shall be able to detect if a Wi-Fi RTT measurement has been under NLoS conditions or if a LoS exists.

4.2.5 Offline processing of data

- The offline processing application shall be able to parse recorded logs of raw measurement data collected by the mobile application to calculate estimates of the device’s positions offline.
- The offline processing application shall be able to visualize any resulting position.
- The offline processing application shall be able to calculate accuracy and precision for logs that contain ground truth information.

4.2.6 Position estimation

- The system shall use an established method for state estimation that can handle multiple data sources, such as the Kalman filter.
- The system shall be able to use ranging results from multiple Wi-Fi RTT ranging requests to provide a position estimate of the smartphone device running the system.
- The system shall be able to use data from device sensors in the position estimation technique.
- The system shall be able to use NLoS/LoS detection information about a W-Fi RTT measurement when estimating a position.
- The offline and online applications should use the same method for position estimation.

4.2.7 User interface

- The user shall be able to see the current bearing and position estimation on an indoor map in the online application.
- The user shall be able to start and stop logging of information in the online application.

4.3 The non-functional requirements

- The mobile application shall be able to provide position estimation updates at a frequency of at least 2Hz, which can be considered real-time.
- The online and offline applications should be written in such a way that they can share code for common tasks, such as position estimation.
- The online application should be developed in the form of a native Android application.
- The method for NLoS/LoS detection should have a misclassification error of <50% in the tested environment.

4.4 Brief summary

The goal of the system is to provide indoor positioning for a smartphone device. The position estimate should use three different sources of information: ranging information from Wi-Fi RTT access points, step detection and bearing estimation from the device's inertial sensors, and NLoS/LoS detection information about a W-Fi RTT measurement.

Chapter 5 System Design

In this chapter, the system design of the different modules is presented. First, the design of the mobile application is presented. Next, the design of the offline application for post-processing of recorded logs is presented.

5.1 Mobile application design

The mobile application was developed in the form of a native Android application. The application had to be able to collect and handle many different sources of data asynchronously. To avoid high coupling between components, an event-driven architecture was used based on the publish-subscribe pattern. This pattern allows for multiple subscribers and publishers of events. An event can have multiple subscribers and a subscriber can be subscribed to multiple events. The event-based design was implemented using a library called RxKotlin[1], which provides several helpful components for composing asynchronous and event-driven applications.

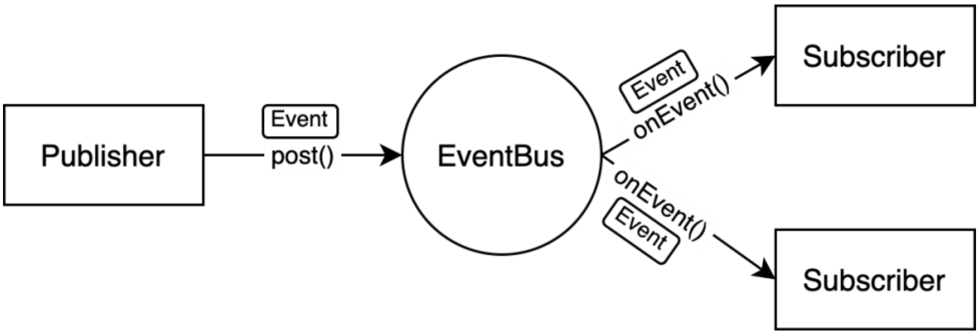


Figure 5.1 Illustration of the publish-subscribe pattern using an event bus.

As illustrated in Figure 5.1, the pattern was implemented using an event bus. The EventBus class contains two static methods, *post* and *subscribe*. The *subscribe* method takes a type as input and converts it into an observable object for that type, which is notified whenever an object of that type is posted. The *post* method simply posts an object of any type.

[1] https://github.com/ReactiveX/RxKotlin

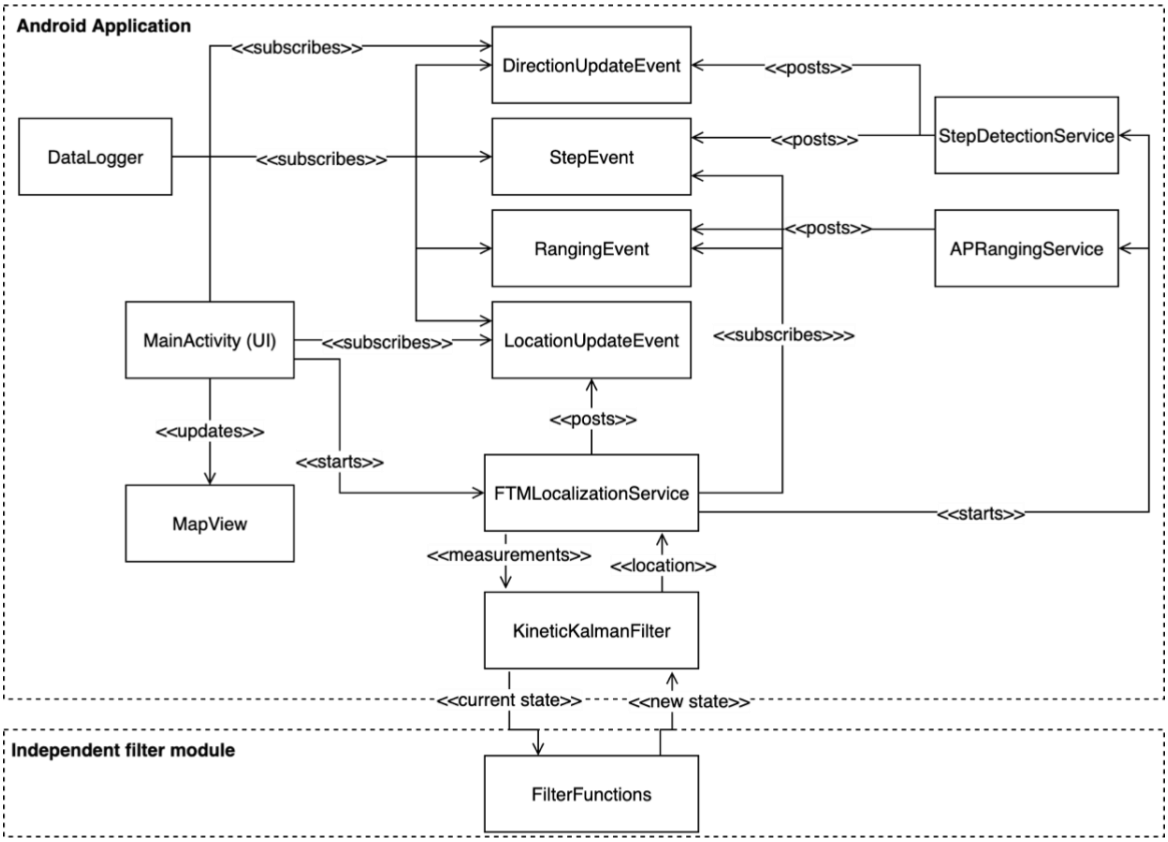


Figure 5.2 Diagram of the android application system design.

In Figure 5.2, the design of the Android application is presented. Android services were used to listen for data and emit corresponding events. These are components running in the background that can perform long-running operations that do not require a user interface. The step detection service listens for inertial sensor information to track the current bearing and detect when a step is taken by the user. The AP ranging service keeps track of nearby Wi-Fi RTT access points and continuously sends ranging requests to these. The FTM localization service listens for both step and ranging events and sends the data to a Kalman filter instance to update the current location estimate. When a new position estimate is available, a location update event is posted which updates the UI. As data logging was a requirement for the mobile application, the DataLogger class subscribes to all events posted and saves these to a file in JSON[2] format.

[2] JavaScript object notation

5.2 Post-processing application design

The main purpose of the offline application is to post-process and visualize logs for the development and evaluation of the implemented IPS. Common mathematical functions and filter logic were written in a pure Kotlin library, which could be used both by the mobile application and for offline log processing.

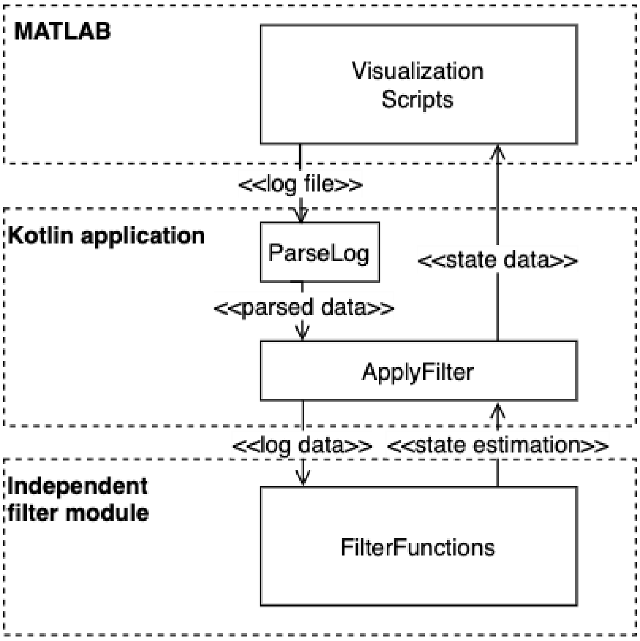


Figure 5.3 Design of offline application

For offline log visualization, MATLAB was used. As Kotlin can be compiled to Java and MATLAB is built upon the JVM, MATLAB can call functions of the Kotlin library to process the data which can then be plotted by the MATLAB engine. For linear algebra and matrix operations in Kotlin, a well-established library called Koma was used[3].

[3] http://koma.kyonifer.com/

Chapter 6 System Implementation and Testing

This chapter presents the details of the implemented IPS and the testing procedure conducted. First, the software and hardware used in the IPS is presented. Then, the details of the localization procedure is described. Next, a few important diagrams of the system implementation is illustrated. Last, the system evaluation procedure is presented.

6.1 Software and hardware

This section describes the hardware and software used for the implementation and testing of the system.

6.1.1 Android Wi-Fi RTT API

To gather FTM ranging information, the Android Wi-Fi RTT API was used. Since Android version 9 (nicknamed Pie), released in August 2018, the open-source mobile operating system owned by Google supports Wi-Fi RTT and ranging using the FTM protocol [10]. Android devices that support Wi-Fi RTT can use Wi-Fi scanning to find nearby APs and then determine if the APs returned from the scan can act as FTM responders through the property `is80211mcResponder` of the obtained `ScanResult` objects. Once available FTM-capable APs have been mapped out, a ranging request can be constructed using the `ScanResult` objects previously obtained from the Wi-Fi scan. Along with a list of such objects a callback object with two functions, `onRangingResults` and `onRangingFailure`, is passed to the `wifiRttManager.startRanging()` function to start the ranging process. When the process has been completed, a list of `RangingResult` is returned to one of the functions in the callback object. `RangingResult` object contains the following information about the FTM measurement [35]:

- `rssi`: Received Signal Strength Indicator (RSSI) to the FTM responder.
- `macAddress`: Mac address of the FTM responder that performed the RTT measurement.
- `distanceMm`: Distance from the FTM responder in mm.
- `distanceStdDevMm`: The standard deviation of the measurements performed.

- `numAttemptedMeasurements`: The total number of attempted measurements.
- `numSuccessfulMeasurements`: The number of successful measurements.
- `rangingTimestampMillis`: The timestamp of the ranging as milliseconds since boot provided by `SystemClock.elapsedRealTime()`.
- `status`: The status of the performed measurement, 0 indicating the measurement was successful and any other value indicates a failure.

6.1.2 Smartphone

The smartphone used for all testing was a Google Pixel 3a running Android 10 provided by Senion. The software of Google Pixel phones is unaltered versions of the Android operating system directly from Google without any third-party software tweaks. This ensures that the Wi-Fi RTT API used for ranging is the AOSP native implementation and has not been modified.

6.1.3 Wi-Fi RTT access points

While support for Wi-Fi RTT has started to become available in some consumer router hardware it was not a common feature at the time of writing. The hardware used as access points in this thesis were four Compulab Wi-Fi Indoor Location Devices (WILD). WILD has the same hardware as the company's miniature PC Fitlet2 designed for IoT applications and has the following specification [36]:

- CPU: Quad-core Celeron J3455
- RAM: 4 GB DDR3L
- WiFi: Intel 8260AC 802.11ac + Bluetooth 4.2
- Storage: M.2 SATA SSD 32 GB

The devices come pre-installed with a customized Debian GNU/Linux distribution called WILD. The devices were using firmware version 0.7.2 of the WILD software which comes with an Intel AC8260 WIFI driver with support for FTM responder mode.

The FTM properties of the WILD devices can be configured using the `/etc/hostpd.conf` file and was using the default configuration of the firmware. This was set up to use the 5.0 GHz Wi-Fi band and 80Mhz channel bandwidth for the

FTM responder mode. This configuration can be replicated by setting the following values in the configuration file [37]:

```
# Set channel bandwidth to 80MHz
ieee80211ac=1
vht_oper_chwidth=1
# Set Wi-Fi band to 5GHz
hw_mode=a
ieee80211n=1
ieee80211ac=1
```

Earlier research has found that the Wi-Fi Intel 8260AC chip has tended to report ranging results with a negative offset of about ~6m when used for FTM ranging [5]–[7]. According to Guo et al., this offset could be a result of differences in the network-card hardware and firmware [7]. When conducting the experiments on Wi-Fi RTT ranging characteristics presented in Chapter 3, the symptom of constant negative range offsets was confirmed with the hardware and software used in this thesis as well and a calibration had to be performed.

6.1.3.1 Calibration method

A ranging measurement from a single Wi-Fi RTT AP can be modelled as:

$$y = k_0 d + m_0 + noise$$

Equation 6.1

where y is the measured distance, d is the true distance, and k_0 and m_0 are tunable model parameters. In the pre-study experiments presented in Chapter 3 the tunable parameters were found to differ in LoS and NLoS situations. In LoS conditions, k_0 was usually found to be close to 1 and m_0 was around -6. In NLoS, the experiments indicated greater values for both k_0 and m_0. Using the empirically derived parameters, a ranging measurement was calibrated according to:

$$d_{calibrated} = \frac{\mathrm{y} - \mathrm{m}_0}{\mathrm{k}_0}$$

Equation 6.2

This correction method is not based on any physical model but rather derived from empirical data showing that such a method would on average, for a set of ranging measurements, yield a fitted line with slope 1.0 and offset 0 on the y-axis. For verification of this calibration method on empirical data, see Appendix A – Measurement calibration.

6.2 Localization implementation

This section describes the details of the implemented solution for localization. Three different modules of the implementation will be presented. First, a fundamental baseline implementation is presented. Then, two different extensions to the baseline implementation are described. First, a method for sensor fusion with dead reckoning. Then, a method for detecting signal NLoS/LoS conditions. The mobile application was developed in Kotlin which is the preferred language for Android app development.

6.2.1 Baseline implementation

Huilla concluded that the Unscented Kalman Filter (UKF), when compared to the other implemented algorithm which used a non-linear least square method, yielded the highest positioning accuracy [5]. The algorithms did not use any other sources of data than the measured distances to APs. As this thesis aims to investigate the improvement possibilities of a Wi-Fi RTT-based IPS, the plain UKF approach is used as a baseline implementation. As such, the baseline implementation estimates the device’s position using only collected RTT ranging measurement and a UKF.

The implemented UKF uses the three-parameter sigma selection approach described in Section 2.3.3. The three parameters were set to the same values as Huilla used in his thesis [5]:

```
alpha = 0.8
```

```
kappa = -1
beta = 2
```

For the state model, a stationary 2-D position model was chosen. This means that the state vector was defined as:

$$x = [\,x \quad y\,]^T$$

While Huilla used a constant velocity state model including velocities vx and vy in the state vector, this was found to increase the position lag in localization. The reason for this is that this model assumes constant velocities which is rarely the case in reality. Such model might work well in environments with large distances, where the user can be assumed to keep a rather constant velocity. However, in office environments like the testing environment, this model is not very realistic and leads to a lag in the state prediction phase after a sudden change in direction or velocity. The initial state is set to the position of the closest AP, determined by the first set of measurements.

The state transition function f responsible for predicting the next state given the current state was defined as a lambda function only returning the current state:

```
val fstate = { state: Matrix<Double> -> state }
```

The reason for the simplicity of this function is that the best prediction that can be made given a state that only includes the current position estimate, is that the position will remain the same the next time step. This means that the state transition function is linear in this case. The P matrix is the covariance matrix of the state and was initialized as an identity matrix with the size of the number of variables in the state:

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The measurement function h was the only non-linear function used in the UKF. This function is responsible for mapping a state to several measurements, in this case, the distances to all APs. For this purpose, a matrix containing the positions of all APs providing distance measurements in time step k was constructed:

$$apPositions_k = \begin{bmatrix} AP_1x & AP_1y \\ \vdots & \vdots \\ AP_nx & AP_ny \end{bmatrix}$$

for a measurement vector of size n. With this matrix, the measurement function calculates the Euclidean distance between the current state and all the APs that provided measurements in time step k as:

```
val hmeas =
{ state: Matrix<Double>, apPositions: Matrix<Double> ->
        Matrix(apPositions.numRows(), 1) { row, _ ->
           sqrt( (state[0] - apPositions[row, 0]).pow(2) +
                 (state[1] - apPositions[row, 1]).pow(2) )
        }
}
```

The covariance matrices of the process and measurement, Q and R, was defined as diagonal matrices with the estimated variances of both quantities, respectively:

$$Q = \begin{bmatrix} \sigma_Q^2 & 0 \\ 0 & \sigma_Q^2 \end{bmatrix}, \qquad R = \begin{bmatrix} \sigma_R^2 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_R^2 \end{bmatrix}$$

where the size of R is dependent on the size of the measurement vector. The values of σ_Q and σ_R were determined experimentally by examining how the values of the innovation and its covariance defined in Equation 2.13 and Equation 2.14, respectively, related to each other during sample measurements. Since S_t is the covariance of the innovation, the standard deviation can be obtained by computing the square root of S_t. The values of σ_Q and σ_R should ideally be chosen so that roughly 68% (1 standard deviation) of the innovations are within the standard deviation.

With the defined variables above, the UKF update function was applied periodically at a frequency of 3.33 Hz, and whenever a `RangingEvent` was posted:

```
fun performTimeUpdate(currentState: Matrix<Double>,
                  P: Matrix<Double>, dt:Double):KFResult {
    val Q =
       getQStdForDt(dt).pow(2) * eye(currentState.numRows())
    val timeUpdateResult =
       FilterFunctions.ukfTimeUpdate(currentState,P,fstate,Q)
    return KFResult(timeUpdateResult.x, timeUpdateResult.P)
}
```

As the uncertainty of a Kalman filter grows with time, the process standard deviation needs to be adjusted if the time between updates is not constant. As the time between updates does vary in this case, the helper function `getQStdForDt()` scales σ_Q linearly according to the time since the last update.

Whenever a `RangingEvent` was posted, a measurement update using the ranging results was also performed using calibrated distances:

```
fun performMeasurementUpdate(currentState: Matrix<Double>, P:
Matrix<Double>, dt: Double, measurements:
List<CorrectedRangingResult>): KFResult {

    val fullApPosMatrix = getApPosMatrix(measurements)
    val Q = getQStdForDt(dt).pow(2) *
        eye(currentState.numRows())
    var intermediateState = currentState;
    var intermediateCovariance = P;
    measurements.forEachIndexed { index, element ->
        val z = Matrix(1,1){ _,_ ->
            element.correctedDistanceMeter
        }
        val R = FilterConfig.DEVIATION_MEAS.pow(2) *
                eye(z.numRows())
        val apPosMatrix = fullApPosMatrix.getRow(index)
        val  measurementUpdateResult =
            FilterFunctions.ukfMeasurementUpdate(
                intermediateState,
                intermediateCovariance,
                fstate,z,createHmeas(apPosMatrix),Q,R)

        intermediateState = measurementUpdateResult.x
        intermediateCovariance = measurementUpdateResult.P
    }
    return KFResult(intermediateState,intermediateCovariance)
}
```

The measurements were passed to the filter one-by-one. This means multiple measurement updates were performed for each set of measurements, and sigma points were generated and passed through the unscented transform between each measurement update. As the baseline implementation uses no other source of data than distance measurements, the input vector *u* and its corresponding input-control function were not used.

6.2.2 First extension: Sensor fusion with dead reckoning

The first potential improvement technique to extend the baseline implementation is fusing the Wi-Fi RTT measurements with inertial sensor data used for detecting steps and walking direction. Estimating a position with the use of such data and a known starting position is called dead reckoning. Modern smartphones have several inertial sensors that could be used to get information about steps taken and walking direction, for example, the magnetometer, gyroscope, and accelerometer [8]. These are the sensors used in this extension.

Estimating device heading using the above-mentioned sensors is not a trivial task, however. Kang et al. successfully implemented and combined three different inertial sensor-based techniques to accomplish a method for accurate pedestrian dead-reckoning indoor localization [11]. Through complex post-processing of raw sensor data, they were able to achieve impressive performance. However, the extent and complexity of their work are considerable. Moreover, as the dead reckoning implementation in this work will act as an extension to other data sources, the accuracy of this extension alone does not have as high requirements on performance. Therefore, a more straight-forward method was chosen. The Android operating system provides API access to several different sensors which are either hardware-based or software-based [38]. Hardware-based sensors provide raw data from the hardware sensors such as gyroscope and accelerometer readings. Software-based sensors are synthetic sensors combining filtered and post-processed data from several hardware sensors, hiding some of the complexity involved in using data from multiple sensors to accomplish certain tasks [39].

Such a software-based sensor was used for device heading estimation, namely the rotation vector sensor (`Sensor.TYPE_ROTATION_VECTOR`). This sensor uses sensor fusion to combine accelerometer, magnetometer, and gyroscope data. The obtained vector can be converted to a rotation matrix through `SensorManager.getRotationMatrixFromVector()`, which in turn can be converted into a device orientation by using `SensorManager.getOrientation()`. Through the orientation information obtained using this method, the device azimuth (angle to the magnetic north) used for heading estimation is extracted through:

```
val azimuth = orientation[0]
```

For step detection, another synthetic sensor was used; the step detector sensor (`Sensor.TYPE_STEP_DETECTOR`). This sensor is triggered each time a user takes a step and is based on accelerometer data [38].

To make the baseline localization implementation more robust, the idea with this extension is to supply the UKF filter with the dead reckoning data collected as described above. Such data can be modeled as an input signal u to the system. With this signal, the full state space-model in Equation 2.11 is implemented so that the input signal directly affects the state estimation as follows:

$$x_{t+1} = \mathrm{A_t x_t + B_{u,t} u_t + B_{v,t} v_t}$$

The input signal with dead reckoning data was modeled as $u_t = L[sin\theta_t \quad cos\theta_t]^T$, where L is a constant step length, and θ_t is the walking direction estimation at time *t*. It should be noted that the accuracy of this model is dependent on the step length constant and for an IPS exclusively based on a dead reckoning algorithm, it would most likely not be accurate enough. However, as this implementation was used together with other sources of positioning information, it was considered sufficient. As the state x_t is defined as a row vector of the 2-D position, the corresponding input-control matrix was chosen as $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ which implies that the contribution in each axis is simply added to the current state. In the implemented UKF this was handled by the f function, which for this reason was modified as:

```
val fstate= { state: Matrix<Double>, input:Matrix<Double>? ->
    if(input != null){
        state + B * input
    } else {
        state
```

```
     }
 }
```

With this modification of the state transition function, the input vector was then added to the state each time a StepEvent was posted by modifying the time update function of the Kalman filter to include an input vector as:

```
fun performTimeUpdate(currentState: Matrix<Double>,
                     P: Matrix<Double>, dt:Double,
                     inputVector: Matrix<Double>?):KFResult {
     val Q =
     getQStdForDt(dt).pow(2)*eye(currentState.numRows());

     val timeUpdateResult =
     FilterFunctions.ukfTimeUpdate(currentState,P,fstate,inputVe
     ctor,Q)

    return KFResult(timeUpdateResult.x, timeUpdateResult.P)
}
```

6.2.3 Second extension: LoS/NLoS detection

Detecting measurement conditions and adjusting measurements accordingly has previously been shown able to improve the performance of Wi-Fi RTT based IPSs [12]. Different methods for detection of sight condition have been proposed and to investigate which works well for Wi-Fi RTT, two different methods for detection were implemented and tested. For this purpose, a dataset of 2000 Wi-Fi RTT measurements was collected at the testing site, equally divided between LoS and NLoS conditions. During the collection of the dataset, the distance between AP and smartphone was widely varied for both conditions. For the method requiring training, 80% of the dataset was used for training, and 20% for evaluation. Below, the two different methods considered are described and discussed.

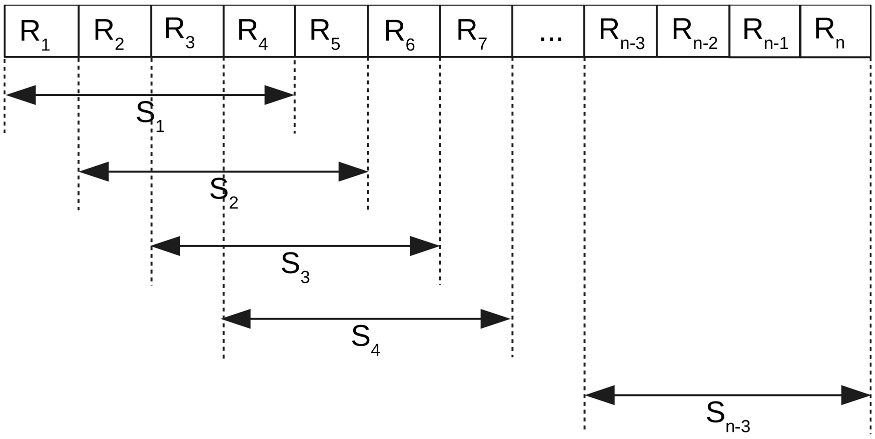


Figure 6.1 Illustration of sample creation for a measurement collection R of size n and a sample size of 4.

Hypothesis testing (HT): Firstly, a method similar to Xiao et al.'s approach using hypothesis testing was implemented [12]. In their data collection process, Xiao et al. used hardware that could sample RSSI samples at a frequency of up to 1000Hz. This is far higher than the highest possible sampling frequency for an Android app. The way an Android application can measure RSSI values for nearby APs is normally by performing a Wi-Fi scan, but since Android 9, the operating system has limited the number of Wi-Fi scans an application can initiate to 4 times per 2 minutes [40]. Nevertheless, the Wi-Fi RTT API returns an RSSI measurement with every ranging result, and instead limiting the highest possible RSSI sampling frequency to the highest stable ranging frequency. This was found to be 2.5Hz, as any higher frequency often resulted in large periods without any successful rangings.

The implemented classifier divides the collected dataset into smaller samples through a sliding window principle (see Figure 6.1) and extracts statistical features from each of these. In the training phase, distributions of the features were calculated. As previously discussed in Section 1.4.4, a sample is then classified using the distributions as NLoS/LoS using a likelihood ratio test where the two hypotheses are defined as:

$$\begin{aligned} H_l&: \quad \mathrm{h} \geq h_t, \qquad LoS\ conditions \\ H_n&: \quad h < h_t, \qquad NLoS\ conditions \end{aligned}$$

Equation 6.3

for some threshold h_t and

$$h=\frac{p\left(x^{(1)},\cdots,x^{(M)}\middle|H_l\right)}{p\left(x^{(1)},\cdots,x^{(M)}\middle|H_n\right)}=\prod_{i=1}^{M}\frac{p\left(x^{(i)}\middle|H_l\right)}{p\left(x^{(i)}\middle|H_n\right)},$$

Equation 6.4

where $p\left(x^{(i)}\middle|H_c\right)$ is the probability density function (PDF) of feature $x^{(i)}$ in condition c. With this likelihood ratio test, a threshold $h_t=1$ was used. The features used were the RSSI sample kurtosis, skewness, mean, standard deviation, and the Rician K factor, all modeled to follow normal distributions. The general PDF of a normal distribution is

$$p(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Equation 6.5

where σ is the standard deviation and μ is the mean of the distribution. The idea behind this method is that the sampled features hypothetically have different distributions in the different situations and the PDF was used to determine which of the two distributions was the most likely given a sample.

Kurtosis is a measure of the “peakedness” of a probability distribution and was included with the idea that RSSI measurements made in LoS conditions generally are expected to be more centralized. The formula for kurtosis of a sample is:

$$kurtosis=\frac{\sum_{i=1}^{N}(Y_i-\bar{Y})^4/N}{s^4}$$

where $\bar{Y}$ is the sample mean, s is the standard deviation and N is the sample size.

Skewness is a measure of the asymmetry of a probability distribution and the skewness of a Gaussian distribution is 0. This feature was included as that the distributions of RSSI measurements are expected to be more asymmetrical in NLoS conditions. The formula for skewness of a sample is:

$$skewness=\frac{\sum_{i=1}^{N}(Y_i-\bar{Y})^3/N}{s^3}$$

where $\bar{Y}$ is the sample mean, s is the standard deviation and N is the sample size.

A Rician distribution is defined using two shape parameters, v, and σ, and the signal envelope of a radio link with LoS conditions is expected to follow such a distribution.

The Rician K factor is a ratio of the power received via the LoS path to the power contribution via NLoS paths and is defined as $v^2/2\sigma^2$ [41]. The parameters v and σ, and therefore the Rician K factor, was numerically estimated using a fixpoint iteration method proposed by C. Koay et al. [42]. In addition to the statistical RSSI features, a Wi-Fi RTT specific parameter, RTT distance standard deviation, was included as a feature. The motivation behind this is that ranging measurements made in environments with severe multipath tended to have a higher variance when looking at the Wi-Fi RTT ranging characteristics results presented in Chapter 3, and therefore also standard deviation.

Every combination of features was evaluated at different sample sizes, and the best performing combination of feature set and sample size used the mean RSSI, kurtosis, and skewness as features, with a sample size of 3. For the training data partition used for evaluation, this configuration achieved a total miss classification rate of 3.6%.

Path Loss Consensus (PLC): The first classifier requires a rather expensive and time-consuming training phase. Models requiring training in some way also face the risk of being over-fitted during training, reducing the generalization of the method. Furthermore, it can be difficult to truly understand the decisions of the classifiers. As a result of these drawbacks, a second method was implemented. The idea behind this method is to utilize a model for free-space path loss (FSPL) to get a free space distance approximation based on the RSSI and compare this to the RTT based distance. For this purpose, the Log-Distance Path Loss Model (see Equation 2.2) was used. RSSI is a measurement of path loss in decibel and as such, can be expressed as:

$$\text{RSSI} = PL(d_0) + N \cdot log_{10}(d/d_0) + X_S$$

Equation 6.6

By using a reference distance $d_0 = 1m$ and merging constants, this expression can be written as:

$$\text{RSSI} = \text{C}_1 + \text{C}_2 \, \text{lo}g_{10}(d)$$

Equation 6.7

To find appropriate values for $C_1 \, and \, C_2$ in Equation 6.7 that best fit the testing environment, empirical RSSI data collected in Section 3.1.1 was used. Using the collected data and the model in Equation 6.7, C_1 *and* C_2 were determined through linear regression.

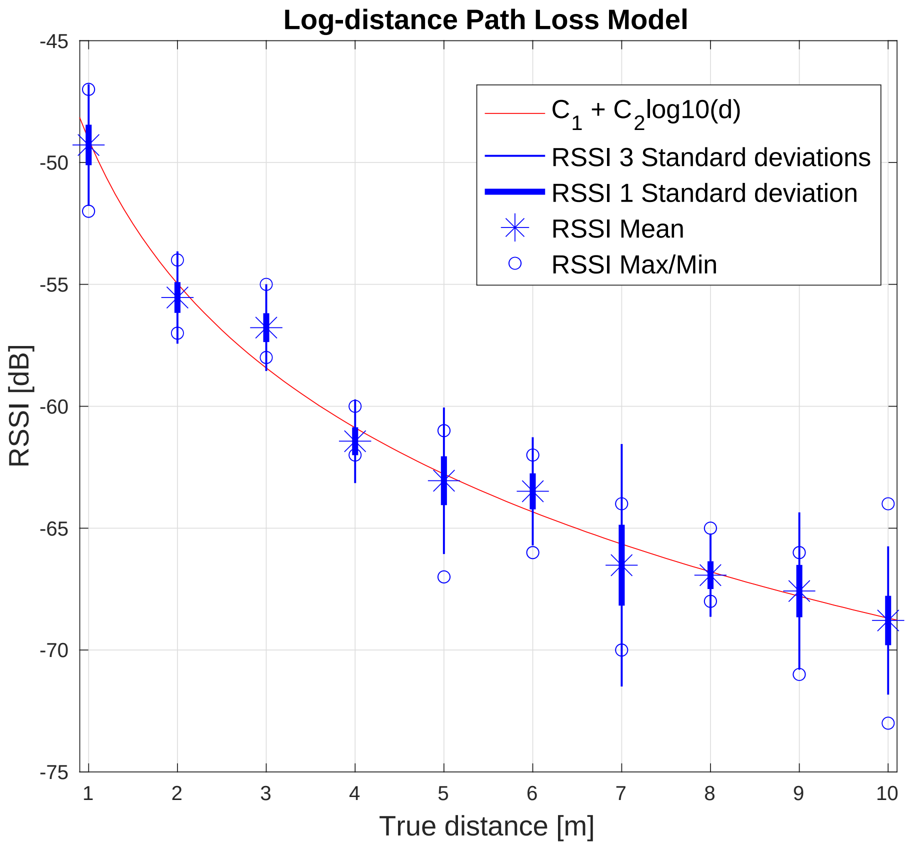


Figure 6.2 Validation of the Log-distance Path Loss model used

This procedure resulted in $C_1 = -49.062$ *and* $C_2 = -19.628$, and the resulting model is illustrated in Figure 6.2. Given that the path loss model estimates a distance based on path loss in free space, it will likely overestimate the distance in NLoS situations where the measured RSSI is likely to be lower than in LoS situations at the same distance.

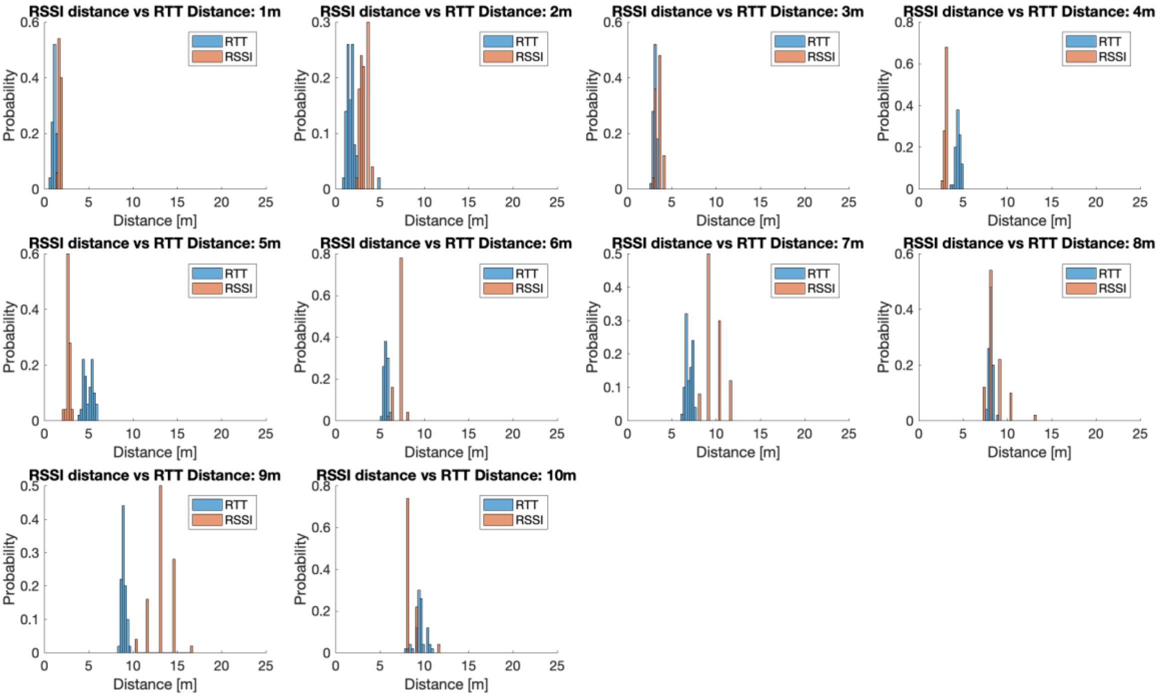


Figure 6.3 Calibrated Wi-Fi RTT and RSSI based distance distribution in LoS.

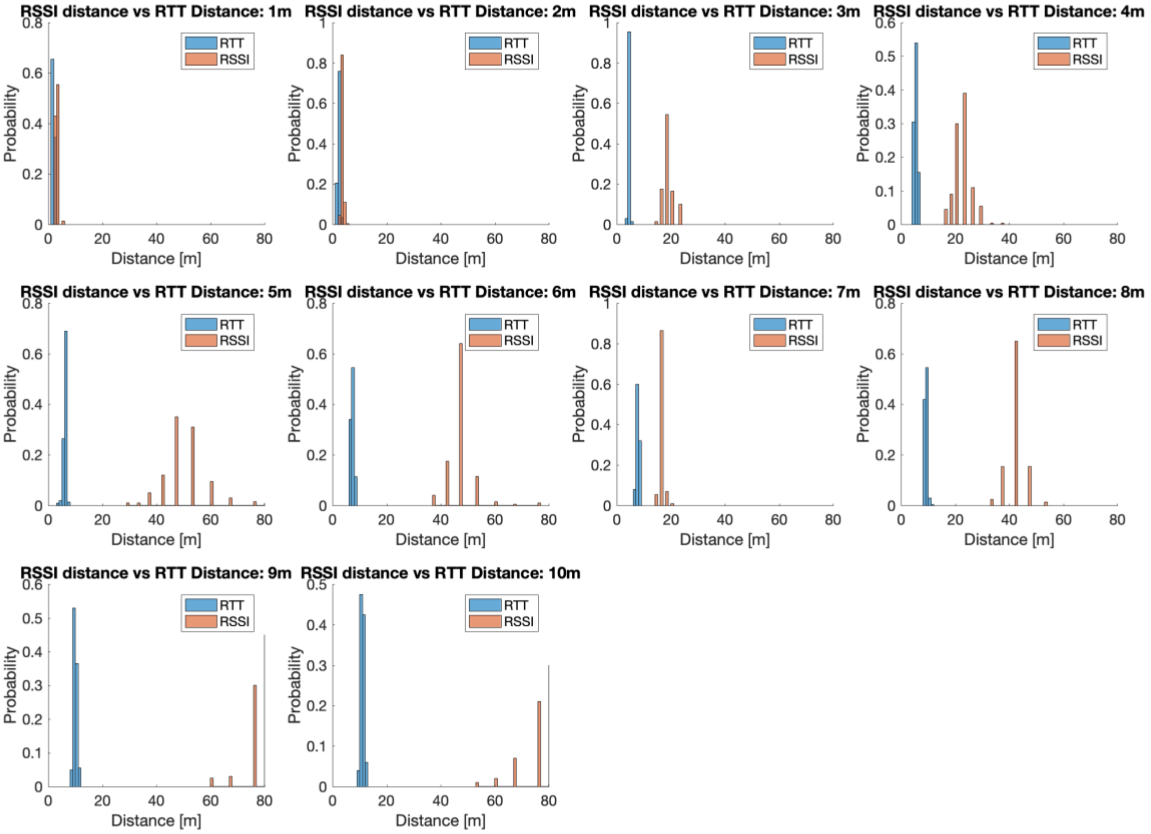


Figure 6.4 Calibrated Wi-Fi RTT and RSSI based distance distribution in NLoS.

Using the data presented and discussed in Section 3.1 Stationary, Figure 6.3 and Figure 6.4 were constructed. As can be seen in Figure 6.3, the difference between the calibrated RTT distance and the RSSI based difference is rather small for all distances. However, in NLoS (Figure 6.4) this no longer holds as the RSSI based distance severely overestimates the distance (with exceptions of the measurements at distance 1m and 2m, for reasons discussed in Section 3.1.2). Based on this behaviour of the two different distance estimation techniques, this method calculates the difference between the RSSI based distance and the RTT estimate, and given that the difference is low enough according to some threshold, a consensus has been reached and the signal is classified as LoS. As can be seen in Figure 6.3, the RSSI and Wi-Fi RTT based distances are fairly closely grouped in LoS, but the RSSI based distances have a larger variance at lager separations. Different methods to calculate a dynamic threshold based on physical models using either RTT distance or RSSI were tested, but for the empirical evaluation data, the best threshold was found to be a constant distance value. With the distance difference threshold set to 5m, this method achieved a total miss classification rate of 8.1% for the evaluation data.

While both methods achieved promising results for the evaluation data, another data set was also used for evaluation. As HT uses raw RSSI values as a feature, one cannot know to which degree a measurement is classed as NLoS solely based on a low RSSI value. As the received signal strength naturally decreases with distance, there might be situations when a signal is classified as NLoS, but in reality, it was only a low RSSI value as a result of a large distance between the AP and receiver. To test these situations, another dataset collected in a long corridor with constant LoS condition to an AP was used. The distance between the AP and receiver was gradually increased up to 10m with increments of 1m and multiple measurements were collected at each fixed distance. For this dataset, using the same previously best-performing features and a sample size of 3, HT achieved a miss classification rate of 58.6%, dramatically lowering its performance. To improve the classifier in these situations, one might want to consider removing mean RSSI as a feature. Meanwhile, PLC showed an increase in performance for this dataset with a miss classification rate of only 0.4%.

Since a method that requires training is not very desirable in the first place for the reasons mentioned above, and PLC can achieve roughly the same or better performance in some situations, it is chosen as the method for detection of signal state.

6.2.4 Implementation overview

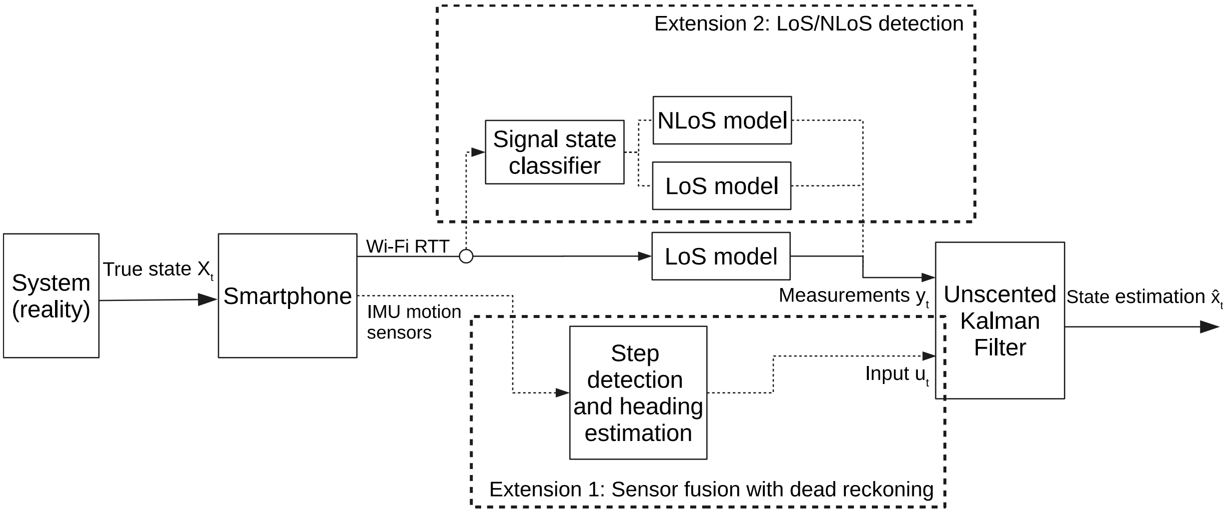


Figure 6.5 Implementation overview.

To summarize the localization implementation, and overview of the two extensions and how they are applied is illustrated in Figure 6.5. Figure 6.5 Implementation overview.

6.3 Key program flow charts

In this section, key diagrams of the implementation are presented.

6.3.1 Diagram of localization procedure

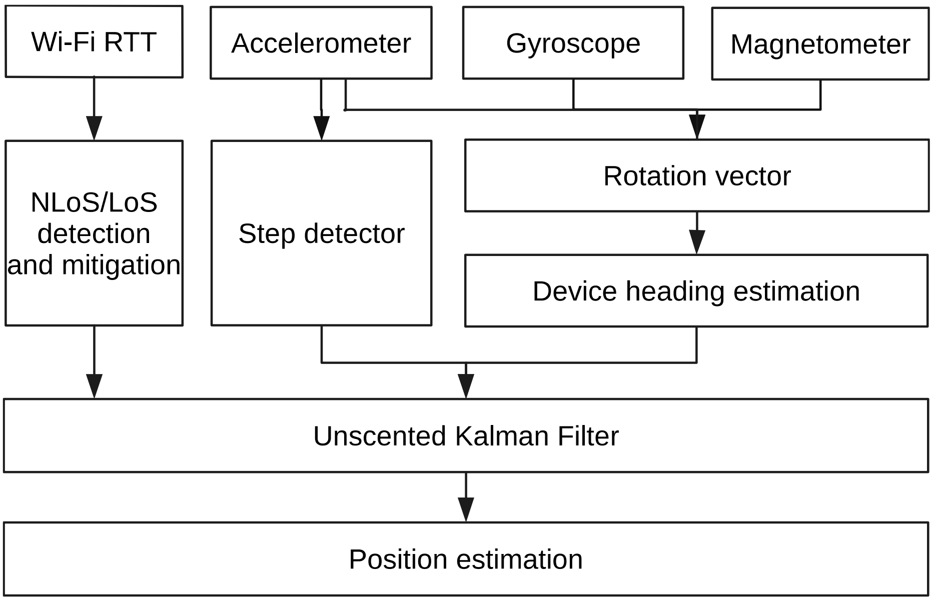


Figure 6.6 Overview of the location estimation procedure

In Figure 6.6, an overview of the localization procedure with the different extensions is presented. Wi-Fi RTT measurements are passed to the filter in the *measurement update*, while the dead-reckoning data is passed in the *time update* phase as an input signal.

6.3.2 Ranging process

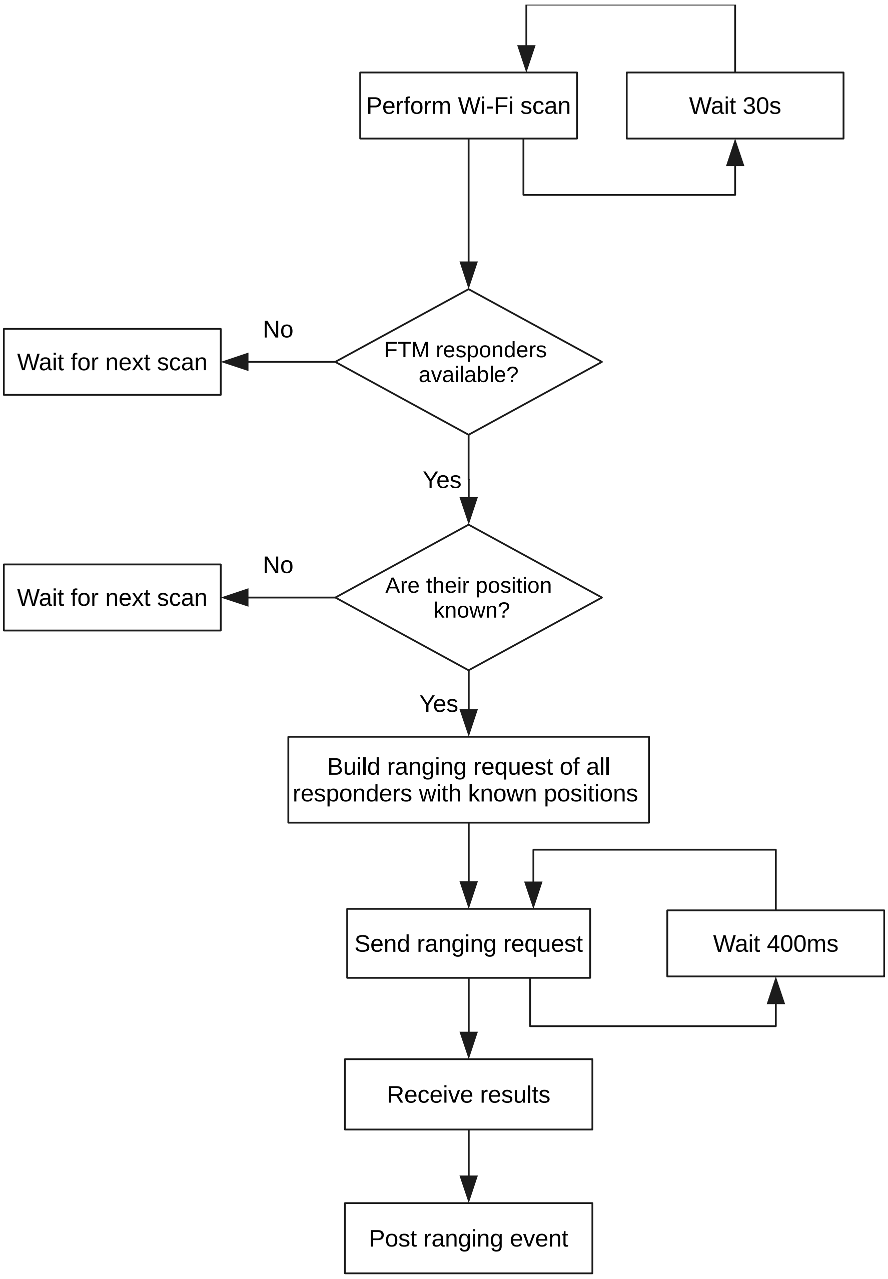
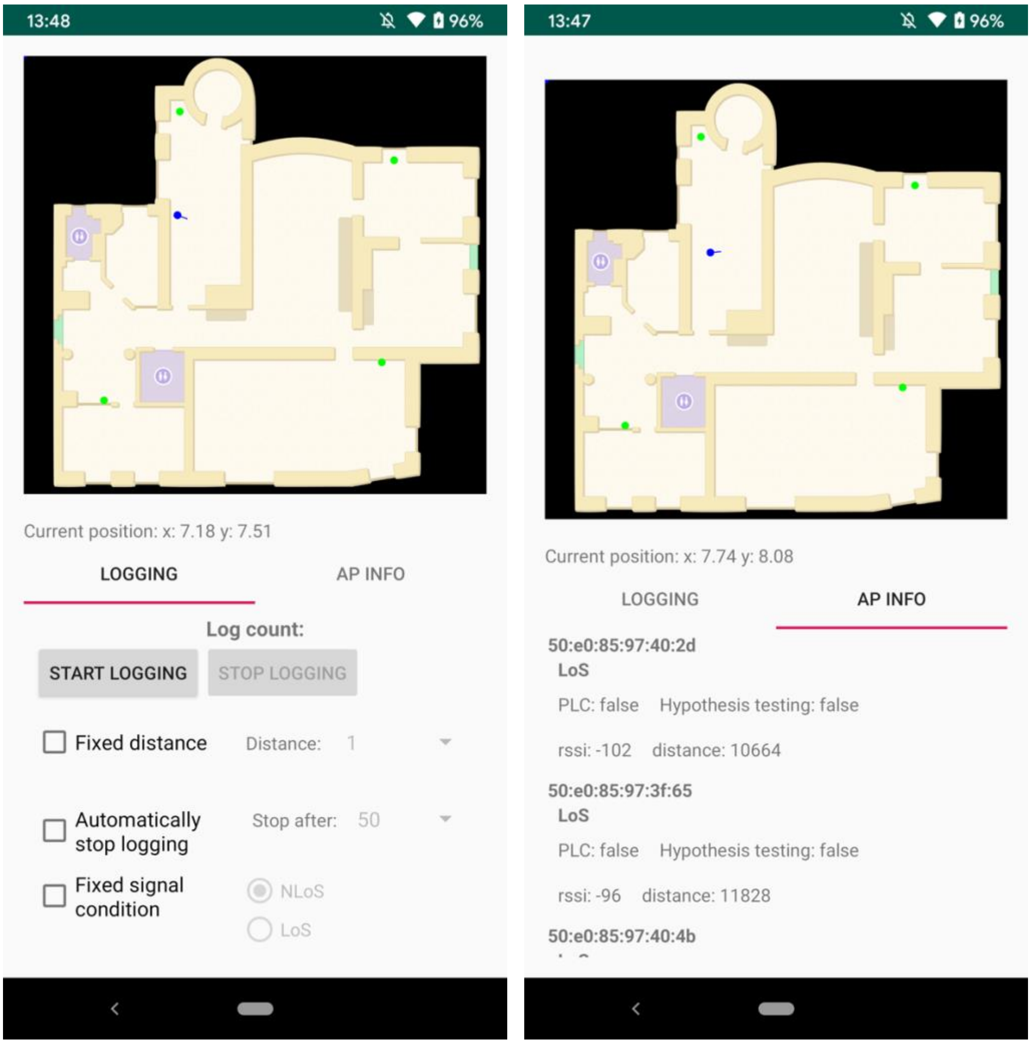


Figure 6.7 Flow chart of the Wi-Fi RTT ranging process.

In Figure 6.7, a flow chart of the ranging process is presented. As no information about the positions of APs is transmitted through the Wi-Fi RTT API, this information is configured locally in the application. If an unknown FTM responder is found in the scanning process, it is simply ignored. All APs that are recognized through the configuration are included in the ranging request.

6.4 Key interfaces of the software system



Current position: x: 7.18 y: 7.51

Log count:

☐ Fixed distance Distance: 1

☐ Automatically stop logging Stop after: 50

☐ Fixed signal condition ● NLoS ○ LoS

Current position: x: 7.74 y: 8.08

50:e0:85:97:40:2d
LoS
PLC: false Hypothesis testing: false
rssi: -102 distance: 10664

50:e0:85:97:3f:65
LoS
PLC: false Hypothesis testing: false
rssi: -96 distance: 11828

50:e0:85:97:40:4b

Figure 6.8 Screenshots of the Android application.

The implemented Android application presented in Figure 6.8 was developed both as a tool for aiding development and as a proof of concept for the IPS. For the latter, a map view showing the real-time position and bearing estimations was implemented. As stated in the requirements in Chapter 4, the mobile application should be able to log any data used for the position estimate. This was the first implemented feature and was mainly used during the initial experiments presented in Chapter 3. With the controls shown in the left image above, the RTT data logged could be labeled with helpful information about the ranging conditions to aid the processing of the data.

6.5 System evaluation

This section describes the procedure for system evaluation that was carried out. An experiment was carried out to investigate how the performance of a Wi-Fi RTT based IPS using UKF is affected by the extensions to the baseline implementation described above.

6.5.1 Measurement calibrations

As earlier discussed, the Wi-Fi RTT measurements required adjustment before they could be used for localization. For this purpose data from Chapter 3 was used. As a normal use case for the IPS involves a user moving around in the indoor environment, the results from the mobile ranging experiments presented in Section 3.2 were used for calibration purposes.

Condition	Correction slope (k_0)	Correction offset (m_0)
LoS	1.118	5.807
NLoS	1.287	5.220

Table 6.1. Calibration parameters used.

The slope and offset of the fitted lines presented in Table 6.1 were used as calibration parameters and measurements were adjusted according to Equation 6.2. When LoS/NLoS detection was enabled in the evaluation, measurements determined to have been made in NLoS were adjusted with the NLoS parameters and vice versa. When signal state detection was not enabled, the LoS parameters were always used.

6.5.2 Evaluation environments

This section describes the characteristics of the testing environments. The experiment was performed for two paths in the office environment at Senion. The two paths have different LoS characteristics; one with mostly favorable/LoS conditions to all APs, and one path where LoS conditions were rare.

6.5.2.1 Path A – Constant LoS conditions

The first testing path was set up in an open kitchen area of the office so that a constant LoS would exist for all APs. The APs were set up around the open area, as can be seen in Figure 6.9. While being open, the area was furnished with tables and chairs.

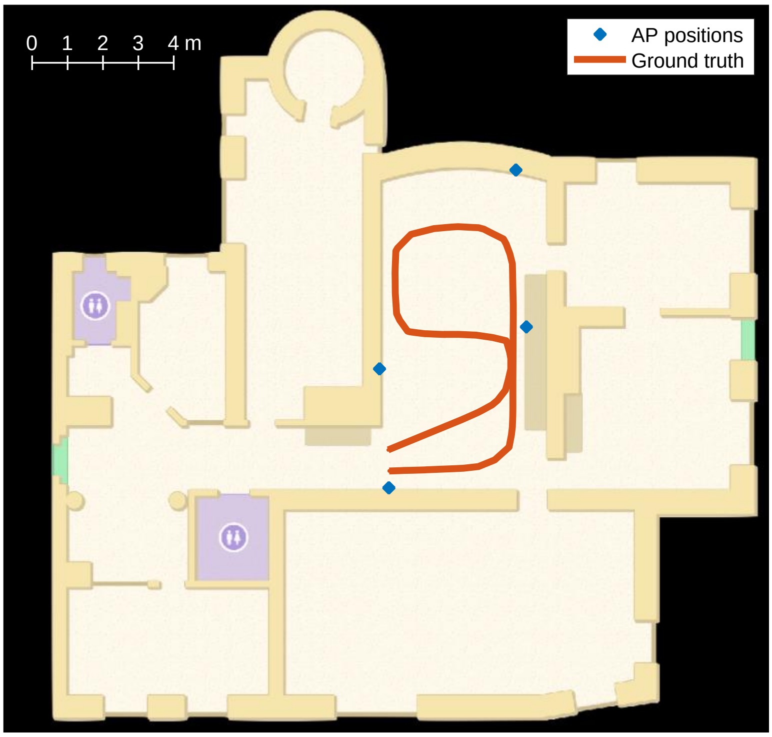


Figure 6.9 Ground truth and positions of APs for Path A

Path A was also chosen to include some tight curves in a rather small area, to see how well the implemented IPS would track such small changes of movement.

6.5.2.2 Path B – Mostly NLoS conditions

The second testing path was constructed to test the IPS in more challenging situations. As earlier research had indicated that ranging in NLoS conditions led to a substantial decrease in positioning performance when only relying on Wi-Fi RTT measurements, this path was constructed to test the implemented IPS performance in such situations.

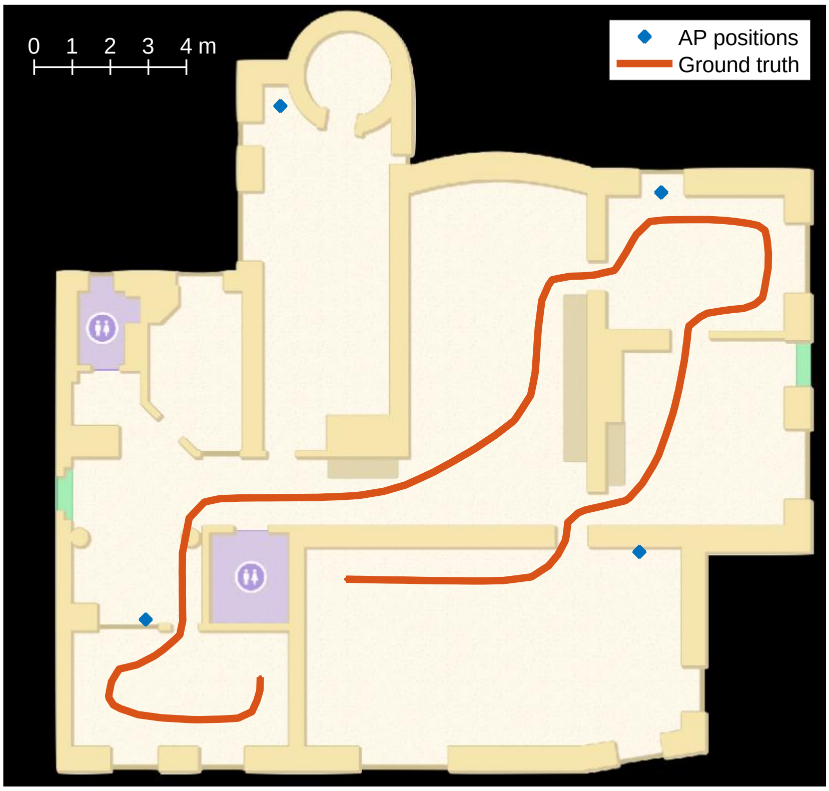


Figure 6.10 Ground truth and positions of APs for Path B

As can be seen in Figure 6.10, there is no location along the path where a direct LoS exists to more than one AP at a time. In the figure above, the thicker walls are made of concrete, while the thinner walls are either glass or drywall. The path passes through multiple office environments with desks, chairs, and monitors.

6.5.3 Evaluation hypotheses

The main problem with an IPS purely based on Wi-Fi RTT measurements is that the RTT delay could be heavily impacted by different indoor propagation effects such as multipath. A signal link can have a significantly increased RTT delay if it bounces multiple times, as the traveled path becomes longer. The effect should however only lead to over-estimations of the distance, which makes it possible to adjust with basic methods if the LoS/NLoS condition is known. Therefore, accurate detection of such conditions should be able to increase the performance when NLoS conditions are dominant. Another method to increase the performance in such situations is through including another source of data in the localization algorithm. Dead reckoning algorithms have previously been used for such purposes with good results and should be able to improve performance in this context as well, given that the implementation used for this purpose is robust enough. To facilitate the interpretation and discussion of the evaluation results, three hypotheses are established:

H1: The baseline implementation can achieve meter level accuracy in favorable conditions.
H2: An IPS using both of the two extension implementations will have higher accuracy than the baseline implementation in situations with dominant NLoS conditions.
H3: Each extension individually will show some indication of performance increase over the baseline implementation.

As Huilla’s IPS managed to achieve meter level accuracy in favorable conditions [5], **H1** is included to make sure that the baseline implementation in this work has comparable performance in similar situations. **H2** and **H3** are included to help answer the third research question, which examines how dead reckoning and techniques for LoS/NLoS can be used to improve localization performance of a Wi-Fi RTT based IPS.

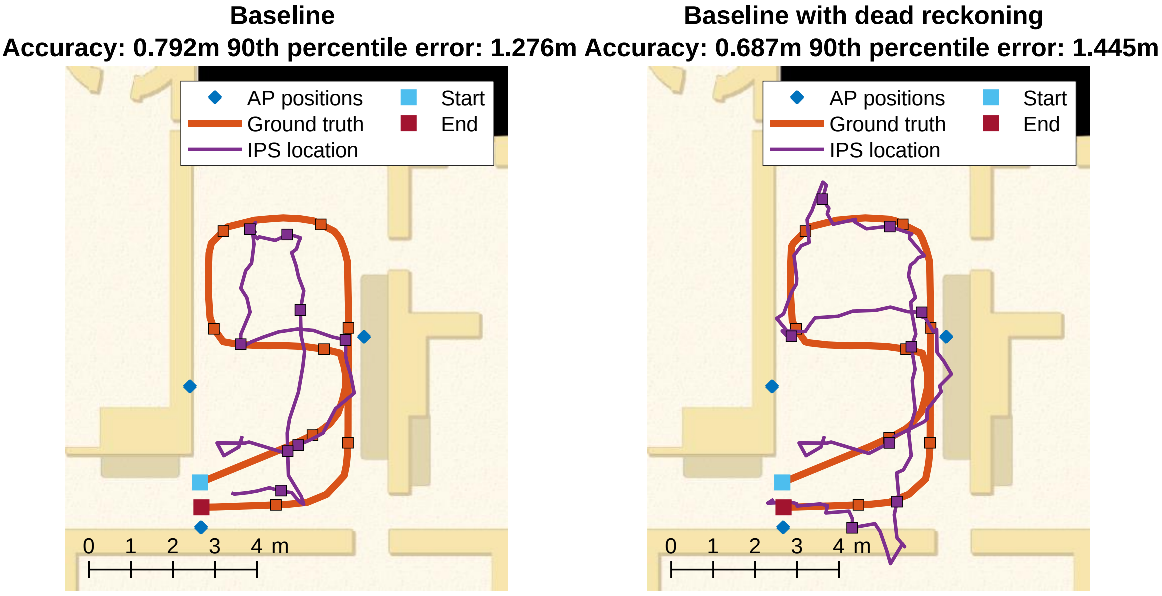
6.6 Brief summary

The implemented positioning system can be divided into three separate modules; a baseline implementation, and two extensions to this. The baseline implementation is a UKF using RTT ranging measurements only. The first extension to the baseline implementation samples device sensor data to estimate a walking direction and detect when the user has taken a step. The second extension detects measurements that have been made in NLoS conditions and mitigate these according to a model. The implemented IPS was tested in two different situations; one where LoS are present to all APs throughout the test and one where LoS conditions are rare.

Chapter 7 Results

In this chapter, the result of the system evaluation is presented. When evaluating the positioning performance of the implemented IPS, the same filter parameters were used for the two testing paths. These were tuned to yield as good results as possible for both of the paths. The values of σ_Q (standard deviation of the process) and σ_R (standard deviation of the measurements) were set to 0.3 and 2.0, respectively. The step length of the dead reckoning model was set to 0.55m. The initial covariance (uncertainty) was set to 40 to allow the filter to quickly pivot from the initial state (the position of the closest AP) to an actual location estimate. To generate the ground truth, the procedure described in Section 1.5.4 was used. The data collected was post-processed through the offline application using different combinations of the two filter extension implementations to investigate their impact on the IPS localization performance in different situations.

7.1 Path A



Baseline
Accuracy: 0.792m 90th percentile error: 1.276m

Baseline with dead reckoning
Accuracy: 0.687m 90th percentile error: 1.445m

Figure 7.1 IPS positioning performance for Path A using different configurations, with a marker placed every 3 meters. Left using baseline implementation and right using baseline together with the dead-reckoning extension.

Path A was walked a total of six times, and a typical round is illustrated in Figure 7.1. As all APs were LoS during the whole path, the NLoS detection was excluded. Using the filter configuration stated above, the baseline implementation achieved an accuracy of 0.79m and a 90th percentile error of 1.28m in the round presented above. The biggest issue seems to be the loop where the baseline is quite constricted. When also applying dead reckoning, the performance was increased in terms of accuracy but decreased in terms of precision, instead achieving an accuracy of 0.69m and a 90th percentile error of 1.45m. The tracking of the loop is improved using dead reckoning as the estimation is more expanded. However, as can be seen in Figure 7.1, the IPS location estimate becomes more overextended in some situations using this configuration, sometimes overshooting in turns that are dominantly present in the path. This seems to be the reason why the 90th percentile error was increased when using dead reckoning.

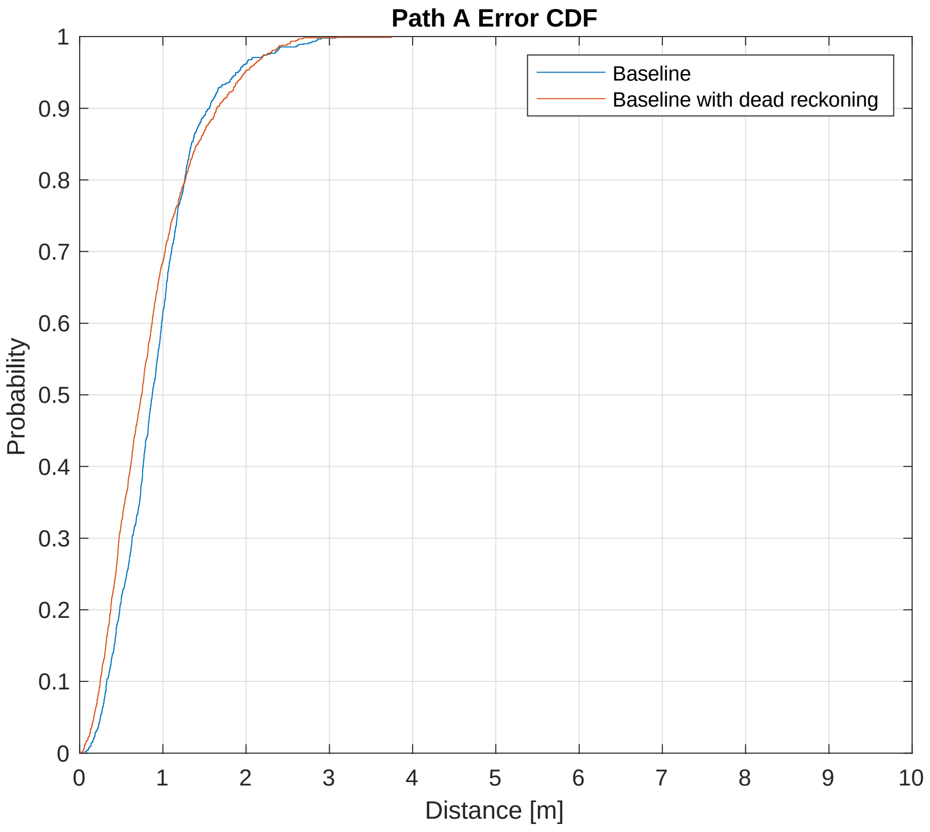


Figure 7.2 Cumulative distribution function of the positioning error for all rounds on Path A.

In Figure 7.2, the CDF of the positioning error for all six rounds is presented. It seems that dead reckoning improved the overall median accuracy for the IPS. Nonetheless, the symptom of a slightly inferior 90th percentile error when using dead reckoning was also present in a majority of the rounds. Despite this, however, both configurations managed to achieve a total 90th percentile error <2m which is within the Wi-Fi RTT ranging accuracy tolerance recommended by AOSP [10]. Both configurations also achieved a total mean positioning error of less than 1m.

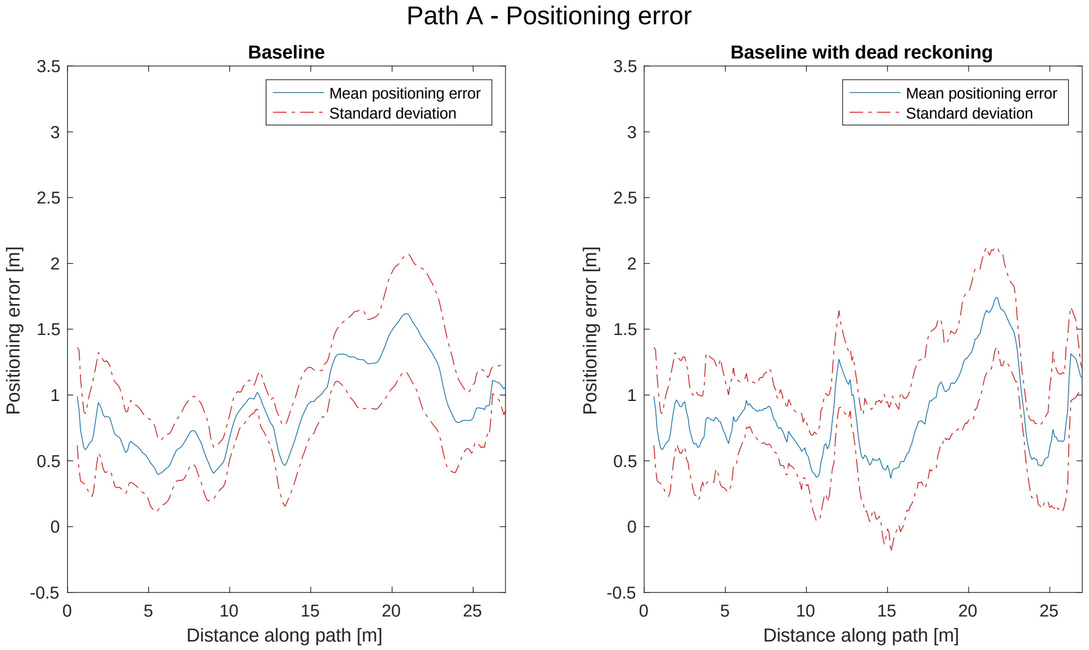
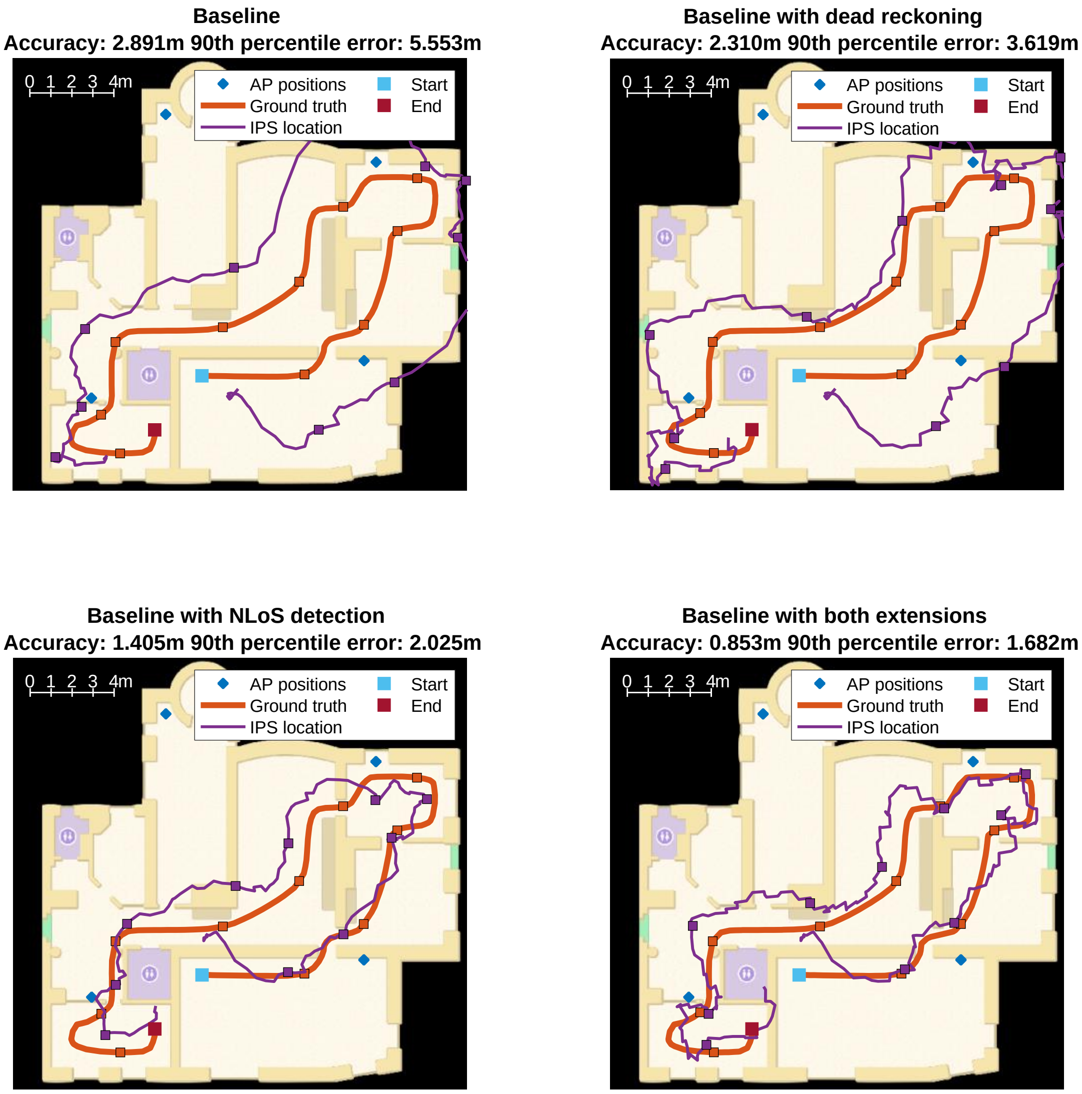


Figure 7.3 Mean positioning error ±1 standard deviation for all walked rounds on Path A.

In Figure 7.3, the positioning error along the path for all rounds on Path A is presented. While generally being fairly tightly grouped with a small standard deviation, the more difficult areas of the path become apparent. Approximately at 20m (along the straight part of the path on the right side), the baseline configurations show the largest mean positioning error and standard deviation indicating an area with large uncertainty. The tracking of this subpath was rather flawed for the baseline, often being shifted ~1m to the left. While the tracing of this particular part of the path seems to be improved when using dead reckoning, this configuration instead struggles a little bit later in the curve where it often surpassed the ground truth. The largest standard deviation was, however, found at approximately 15m along the path which indicates that the top right corner was sometimes also difficult for the dead reckoning implementation to track properly.

7.2 Path B



Baseline
Accuracy: 2.891m 90th percentile error: 5.553m

Baseline with dead reckoning
Accuracy: 2.310m 90th percentile error: 3.619m

Baseline with NLoS detection
Accuracy: 1.405m 90th percentile error: 2.025m

Baseline with both extensions
Accuracy: 0.853m 90th percentile error: 1.682m

Figure 7.4 IPS positioning performance for Path B using different configurations with a marker placed every 5 meters. Top-left only using the baseline implementation and bottom-right using baseline with both extensions.

Path B was, just like Path A, walked a total of six times and a typical round is illustrated in Figure 7.4. Using the same filter configuration as before, it becomes apparent that the performance of the baseline implementation is no longer satisfactory, only achieving an accuracy of 2.89m and a 90th percentile error of 5.55m. The best configuration this round reached a best accuracy of 0.85m and 90th percentile error of 1.68m and was achieved using both NLoS detection and dead reckoning. Here it is clear that both extensions to the baseline implementation are needed. The extension having the largest impact on the performance appears to be the NLoS detection and mitigation, more than doubling the accuracy when compared to the baseline. For this path, however, dead reckoning also had a positive impact on the performance in contrast to Path A, where the impact was negative in terms of the 90th percentile error. The most apparent difference between the plain baseline and the use of dead reckoning seems to be an improved tracking of curves when using dead reckoning.

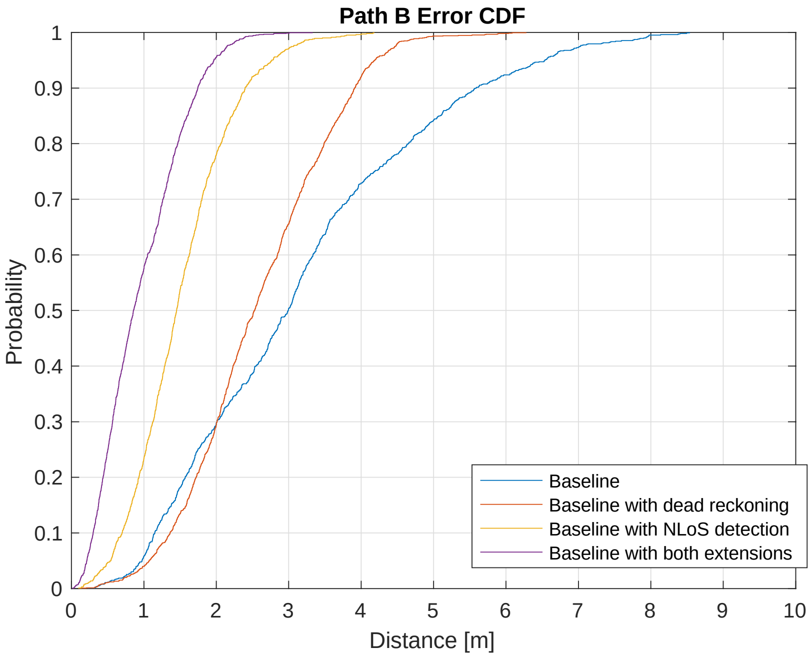


Figure 7.5 Cumulative distribution function of the positioning error for Path B, based on all six rounds.

Figure 7.5 presents the CDF of the positioning error for all six rounds on Path B. Here, the impact of the two extensions becomes even more apparent, all indicating a performance improvement over the baseline implementation. However, only the configuration using both extensions managed to achieve a total mean error of less than 1m and 90th percentile error of less than 2m.

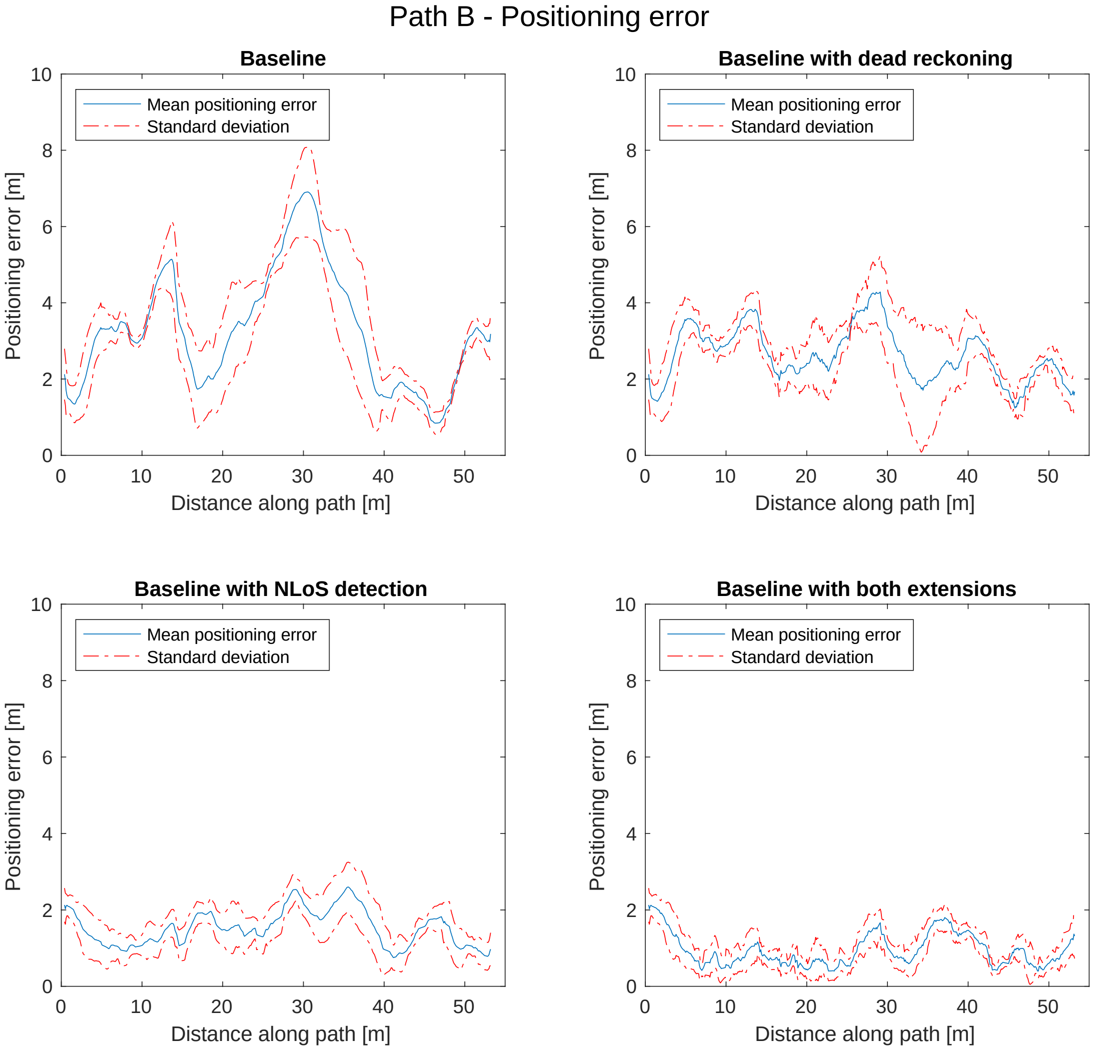


Figure 7.6 Mean positioning error ±1 standard deviation for all walked rounds on Path B.

In Figure 7.6, the positioning error along the path for all rounds on Path B are presented. Once again, it is evident that the two extensions have a large positive impact on the performance and the mean positioning error is reduced when one or both of them are used. When comparing the two top plots, it seems that the dead reckoning extension had the largest impact at approximately 30m. This is around the open kitchen area that Path A was set up in. When looking at Figure 7.4, the curve in this area was not tracked particularly well by the baseline and the dead reckoning algorithm made a big difference. Neither of the two bottom plots show any large spikes in the mean positioning error and generally show a very small variance between rounds.

Chapter 8 Discussion

In this chapter, the result of the thesis is discussed. First, the characteristics and difficulties of Wi-Fi RTT ranging are considered. Next, the result of the system evaluation is analyzed and briefly compared to results achieved in related works. Then, a few methodologic aspects and areas of improvement are reflected upon. Last, a few ethical aspects of indoor positioning are highlighted which should be taken into consideration when implementing similar systems.

8.1 Wi-Fi RTT ranging

An important part of this thesis was to map out the accuracy and behavior of the Wi-Fi RTT technology. In an attempt to understand the capabilities and shortcomings of Wi-Fi RTT as a ranging technology, several ranging experiments in different situations were conducted (see Chapter 3). One of the most important take-aways from these experiments is that, for the combination of hardware and software used for this thesis, a constant negative ranging error is always present in all ranging measurements. This issue was confirmed with two different phones of different models. Since negative distances are not physically interpretable, the reported distances need to be adjusted through some model to be used for any localization purpose. Another thing that became apparent is that indoor environments can be challenging for the technology and that small changes in position or orientation can lead to large differences in the reported distances. Even when conducting ranging measurements in LoS conditions, it proved difficult to reproduce the results of the conducted experiments. On some occasions the distribution of ranging results showed multiple peaks for some measurement locations, indicating multipath effects, whereas they would not be present at the same location at other times. These inconsistencies are very difficult, if not impossible, to model which makes an accurate method for correction of ranging measurements a truly challenging task.

In NLoS the distribution of ranging results was found to have a bigger variance and more often showing multiple peaks, indicating a stronger presence of multipath effects. It was, however, clear that distance measurements made in NLoS had a larger offset in general, which is an expected effect of NLoS propagation. Another important finding was that movement of the device appears to affect ranging results. When moving, the range offset appeared to be increased in both LoS and NLoS situations. Even when moving towards an AP in LoS, there was an increase in offset when in theory such a situation should lead to a marginally shorter RTT and therefore also estimated distance.

The behavior of ranging measurements in the different situations shows that there are some difficulties and peculiarities that need to be considered when using the technology for indoor localization, especially in situations when a direct LoS is not present. It should, however, be noted that the results of these experiments are not necessarily representative of the technology as a whole, as they are most likely dependent on the testing environment and the combination of hardware and software.

8.2 System evaluation

In Section 6.5.3, three hypotheses were established regarding the localization performance of the implemented IPS. The results presented in Chapter 7 confirm the first hypothesis: although not with a great margin, the baseline implementation *was* able to achieve meter level accuracy in situations with favorable conditions. In the open environment with only a body sometimes blocking LoS conditions, the baseline implementation which only used Wi-Fi RTT range measurements, achieved a mean positioning error of 0.79m and a 90th percentile error of 1.28m. The biggest issue in this environment seemed to be a too constricted estimation that often cut corners early. When also using dead reckoning this problem was reduced and the accuracy was improved by ~0.1m, instead reaching 0.69m. The 90th percentile error, however, was somewhat reduced using dead reckoning, instead achieving a precision of 1.45m. This was most likely a result of the dead reckoning algorithm overshooting corners. This could be a symptom of the step length used in the dead reckoning model being bigger than the actual step length in certain situations. In paths with many short sub-paths and turns, the step length is likely smaller when making turns and using a model with a constant step length could therefore result in overshooting. Path A is made up of multiple short distances and the overshooting errors are here likely to become more significant. The area used for Path A was the largest open space available in the office but is still relatively small (approximately 3x6 meters). It is possible that a larger open space with wider turns and longer straight sub-paths would have given another result.

The second hypothesis established, stated that the configuration using both extensions would outperform the baseline in situations where NLoS conditions are dominant. This was confirmed when looking at the results from Path B indicating an improvement by a factor of more than three for both accuracy and 90th percentile error over the baseline when using both extensions. Although both extensions improved the performance in path B, the NLoS detection extension had the biggest positive impact on the performance. This also confirms that Wi-Fi RTT ranging in NLoS conditions can be problematic and have a large effect on the localization performance if not mitigated.

The third and final hypothesis stated that both extensions would show some indication of performance improvement over the baseline. This was very clearly confirmed by the results from path B when looking at the positioning error CDF. In Path A, the dead reckoning extension improved the accuracy but increased the 90th percentile error. The 90th percentile error was increased by almost 0.18m, indicating that the dead reckoning extension can lead to larger jumps in position estimations. This is especially a problem in smaller environments where the step length is likely smaller.

In an open area similar to that of Path A, the UKF based Wi-Fi RTT IPS implemented by Huilla in [5] managed to achieve an accuracy and 90th percentile error of 0.71/1.16m which both are better than the 0.79/1.28m achieved by the baseline implementation in Path A. Dead reckoning in the same environment slightly improved the accuracy, at the cost of a larger 90th percentile error. In an office environment with mostly NLoS conditions to the APs similar to Path B, Huilla's IPS achieved a mean error of 2.41m and a 4.49m 90th percentile error by assuming all measurements above 10m were made in NLoS and using a correction formula to mitigate these distances. The results achieved for Path B of 0.85m/1.68m using both extensions are therefore an improvement, achieving meter level accuracy and a 90th percentile error of less than 2m. However, it should be noted that the results were achieved on two different sites and the numbers might not be directly comparable. Differences in material and thickness of walls can lead to different types of NLoS propagation effects and both of the sites used for system evaluation in Huilla's work were larger than the ones used in this thesis. The sizeable improvement of using both extensions over just the baseline implementation in Path B does, however, show great potential for sites with dominant NLoS conditions.

8.3 Method and implementation

The ranging data reported through Wi-Fi RTT is noisy and poses several difficulties when used for indoor positioning. Previous research investigating the technology for indoor positioning has indicated that using a filter to process the data can be considered a necessity to achieve reasonably good results. In this thesis, the unscented Kalman filter was used for this purpose as it is an established algorithm that in theory should handle non-linearity of the models well. One of the core concepts of Kalman filters, however, is that the noise in both process and measurements are assumed to be Gaussian distributed. If this is not the case, the filter will not yield optimal estimations and other filter solutions might be better suited. Distributions of Wi-Fi RTT measurements made over time in a stationary position was in this thesis on multiple occasions found to contain multiple peaks and other unsymmetrical features which are not present in Gaussian distributions. This indicates that the ranging error of the technology might not be Gaussian distributed, at least not in all situations. This behavior is likely just a symptom of indoor propagation effects, but since these are often present another filtering method could be better suited. An example of such a filter is the Particle filter, which uses a set of particles to represent a distribution and has no constraints on either linearity of models or noise distribution.

Another characteristic of the Wi-Fi RTT ranging measurements found, was the constantly present negative offset in the reported distances. Because of this, all ranging measurements used for localization purposes, regardless of line of sight condition, had to be adjusted through some method. The calibration parameters were derived from empirical data gathered in the same setting as was used for the IPS validation. In this thesis, measurements were classified to have been made in either LoS or NLoS, and parameters from a simple linear regression were used as a basis for this calibration in each of the two cases. This method of “correcting” range measurements has no support from any physical model but is purely based on the empirical data collected in the environment. This proved to work rather satisfactory for the tested environment but requires a training phase in which empirical ranging data is collected. This introduces a risk of over-fitting the system to the environment. The ranging data collected can be assumed to differ between sites and even between different parts of certain sites. Therefore, the method might not work as well in other environments. This decreases the generalizability of the method and a more general correction method would have been preferred.

The dead reckoning implementation used in this thesis is based on the synthetic rotation vector sensor available through the Android API. While this is based on fused data from multiple sensors which is heavily filtered, one of the sources used is the magnetometer sensor. This sensor is known to be inaccurate and heavily affected by the environment such as the presence of nearby electrical fields. While the rotation vector was found to be a lot more stable than using the magnetometer sensor directly, it is still influenced by these factors. In the tested office environment, the rotation vector yielded a heading estimation accurate enough to improve the position estimation, but this might not be the case in all sites. For this reason, a heading estimation method not relying on the magnetometer sensor would be more generalizable to other sites.

Furthermore, the ground truth procedure presented in Section 1.5.4 was used throughout the thesis, including the results. The method uses software developed by Senion and the ground truth was generated by walking a predefined path while continuously collecting timestamped sensor data. Although the ground truth data used in the system evaluation is regarded to be of high accuracy, no system is perfect, and it would have been interesting to know the characteristics of the ground truth errors. It is possible that other methods for ground truth determination would have given a slightly different result.

8.4 Ethical aspects of indoor positioning

Wi-Fi RTT has been developed and made available in consumer products for the sole purpose of enabling high accuracy localization in situations where other technologies such as GPS fails to do so. While the type of positioning system the technology enables is intended to favor the end-user, it also brings concerns about data privacy. The increased possibility of tracking a user's location may come at the cost of the user's privacy if the data is not carefully handled. An IPS at a workplace could for example be used by managers to track employees in a way that creates a feeling of constantly being monitored. There are, however, scenarios and environments where tracking of personnel is necessary for safety reasons. In risky work environments, for example, a tracking system of employees could mean the difference between life and death in a rescue operation. It is also possible to handle sensitive location data in an ethical manner. Using encryption or anonymized user location data can be part of such a policy, but it ultimately comes down to who the data is shared with, for what purpose and what information is disclosed.

In some commercial IPSs the system shares positioning data with other users or services and depending on what the user has knowledge of and/or has agreed to, this could be considered unethical handling of sensitive information. In this thesis, however, an independent IPS was implemented that requires no other components than ranging-capable access points and a smartphone. As the smartphone calculates its own position, the location data is only available locally on the phone and is not shared with any other service. This type of IPS can be considered ethical as only the user has access to, and can benefit from, the location data.

Chapter 9 Conclusion

This thesis has investigated how Wi-Fi RTT together with sensor fusion and NLoS/LoS detection can be used to determine the indoor position of an Android smartphone. A test configuration consisting of several Wi-Fi RTT access points was set up in an office environment and a positioning algorithm based on an Unscented Kalman Filter was implemented. The implemented IPS was evaluated in both favorable conditions (plenty of LoS situations) and sub-optimal conditions (dominant NLoS situations). From the results of the system evaluation, it can be concluded that meter level accuracy is possible to achieve in both the environments, by using both NLoS/LoS detection and dead reckoning. A 90th percentile error of less than two meters was also achieved in both situations, demonstrating the robustness of the IPS. The novel method used for detection and mitigation of NLoS conditions proved to be the most important feature in this attempt, more than doubling the achieved accuracy on the path with dominant NLoS conditions. While dead reckoning also improved the performance of the IPS, it is clear that multipath and other propagation effects leading to inaccurate ranging measurements are the biggest problem for Wi-Fi RTT as a ranging technique used for indoor positioning purposes.

In the beginning of this thesis, three research questions were posed to guide the work and help to accomplish the research purpose. Through the work that has been conducted, the research question can now be answered.

How do NLoS conditions affect Wi-Fi RTT ranging measurements?
In general, NLoS conditions displayed indications of the Wi-Fi signal being affected by multipath propagation. For ranging measurements conducted in NLoS situations, this appeared to have two effects; 1), a longer mean distance was reported, most likely as a result of the signal having to travel a longer path, and 2), distributions of distance measurements conducted in a stationary setup showed an increased variance. These are both effects that needs to be considered when using the technology for indoor positioning.

Is it possible to accurately determine if a Wi-Fi RTT ranging measurement has been made in LoS or NLoS conditions on an Android smartphone?

In this work, two different methods for LoS/NLoS detection was considered. The first method, HT, depended on a training phase, in which normal distributions of several different statistical features derived from range measurements were produced in the two different sight conditions. After the training phase, these distributions were used in a hypothesis test to classify samples of range measurements. This method achieved a very low misclassification rate at short distances, but performed poorly at long distances in LoS.

The second method, PLC, did not require a training phase, as it instead compared the Wi-Fi RTT based distance estimation with a distance estimation derived from the measurement's RSSI value. Given that the difference was big enough according to a set threshold, the measurement was classified as NLoS. This method had a misclassification rate of less than 10% in both of the tested situations. This suggests that it *is* possible to determine the sight conditions of a ranging measurement with good accuracy, at least in the tested environment.

How do NLoS detection and sensor fusion affect the possibility of achieving meter level accuracy for an IPS using Wi-Fi RTT?
As expected, the implemented IPS achieved meter level accuracy in an open area environment without using any of these techniques. In the tested environment where LoS conditions were rare, meter level accuracy was only achieved using both the dead reckoning algorithm as well as the method for LoS/NLoS detection. Therefore, when used together with a Wi-Fi RTT based IPS, these techniques can enable meter level accuracy even in challenging situations.

9.1 Future work

To further improve the performance and robustness of the Wi-Fi RTT based IPSs, it is important to continue to investigate the noise characteristics and behavior in different situations. As mentioned earlier, there are other filter solutions such as the Particle filter which might be better suited for Wi-Fi RTT as some data indicates that the noise may not be Gaussian distributed. If a model of the measurement noise is found, it could also be used to improve the measurement correction procedure and NLoS mitigation method presented in this thesis. Techniques such as machine learning are also interesting candidates to further improve the detection and mitigation of NLoS conditions.

References

[1] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen, “A Realistic Evaluation and Comparison of Indoor Location Technologies: Experiences and Lessons Learned,” in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, 2015, pp. 178–189, doi: 10.1145/2737095.2737726.

[2] Y. Wang, X. Yang, Y. Zhao, Y. Liu, and L. Cuthbert, “Bluetooth positioning using RSSI and triangulation methods,” in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, 2013, pp. 837–842, doi: 10.1109/CCNC.2013.6488558.

[3] C. Chen, Y. Chen, H. Lai, Y. Han, and K. J. R. Liu, “High accuracy indoor localization: A WiFi-based approach,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 6245–6249, doi: 10.1109/ICASSP.2016.7472878.

[4] E. Au, “The Latest Progress on IEEE 802.11 mc and IEEE 802.11 ai [Standards],” *IEEE Veh. Technol. Mag.*, vol. 11, no. 3, pp. 19–21, 2016.

[5] S. Huilla, “Smartphone-based Indoor Positioning Using Wi-Fi Fine Timing Measurement Protocol,” University of Turku, Finland, 2019.

[6] A. Mohamed Ibrahim *et al.*, “Verification : Accuracy Evaluation of WiFi Fine Time Measurements on an Open Platform.,” *Mob. Comput. Netw.*, p. 417, 2018.

[7] G. Guo, R. Chen, F. Ye, X. Peng, Z. Liu, and Y. Pan, “Indoor Smartphone Localization: A Hybrid WiFi RTT-RSS Ranging Approach,” *IEEE Access*, vol. 7, pp. 176767–176781, 2019, doi: 10.1109/ACCESS.2019.2957753.

[8] Z. Chen, H. Zou, H. Jiang, Q. Zhu, Y. C. Soh, and L. Xie, “Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization,” *Sensors*, vol. 15, no. 1, pp. 715–732, 2015.

[9] Y. Jin, Hong-Song Toh, W. Soh, and Wai-Choong Wong, "A robust dead-reckoning pedestrian tracking system with low cost sensors," in *2011 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2011, pp. 222–230, doi: 10.1109/PERCOM.2011.5767590.

[10] Android Open Source Project, "Wi-Fi RTT (IEEE 802.11mc)." [Online]. Available: https://source.android.com/devices/tech/connect/wifi-rtt. [Accessed: 03-Feb-2020].

[11] W. Kang and Y. Han, "SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization," *IEEE Sens. J.*, vol. 15, no. 5, pp. 2906–2916, May 2015, doi: 10.1109/JSEN.2014.2382568.

[12] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-Line-of-Sight Identification and Mitigation Using Received Signal Strength," *IEEE Trans. Wirel. Commun.*, vol. 14, no. 3, pp. 1689–1702, Mar. 2015, doi: 10.1109/TWC.2014.2372341.

[13] M. A. Al-Ammar *et al.*, "Comparative Survey of Indoor Positioning Technologies, Techniques, and Algorithms," in *2014 International Conference on Cyberworlds*, 2014, pp. 245–252, doi: 10.1109/CW.2014.41.

[14] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007, doi: 10.1109/TSMCC.2007.905750.

[15] J. S. Seybold, *Introduction to RF propagation.* Wiley, 2005.

[16] J. Hightower and G. Borriello, "Location sensing techniques," *IEEE Comput.*, vol. 34, no. 8, pp. 57–66, 2001.

[17] "IEEE Standard for Definitions of Terms for Antennas," *IEEE Std 145-2013 (Revision IEEE Std 145-1993)*, pp. 1–50, Mar. 2014, doi: 10.1109/IEEESTD.2014.6758443.

[18] G. Lui, T. Gallagher, B. Li, A. G. Dempster, and C. Rizos, "Differences in RSSI readings made by different Wi-Fi chipsets: A limitation of WLAN localization," in *2011 International Conference on Localization and*

GNSS (ICL-GNSS), 2011, pp. 53–57, doi: 10.1109/ICL-GNSS.2011.5955283.

[19] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal Process. Mag.*, vol. 22, no. 4, pp. 41–53, Jul. 2005, doi: 10.1109/MSP.2005.1458284.

[20] L. Schauer, F. Dorfmeister, and M. Maier, "Potentials and limitations of WIFI-positioning using Time-of-Flight.," *Int. Conf. Indoor Position. Indoor Navig. Indoor Position. Indoor Navig. (IPIN), 2013 Int. Conf.*, pp. 1–9, 2013.

[21] S. B. Wibowo, M. Klepal, and D. Pesch, "Time of flight ranging using off-the-self ieee802. 11 wifi tags," in *Proceedings of the International Conference on Positioning and Context-Awareness (PoCA'09)*, 2009.

[22] S. Sand, A. Dammann, and C. Mensing, "Position Estimation," in *Positioning in Wireless Communications Systems*, Wiley, 2013, pp. 69–99.

[23] Mathworks, "Understanding Kalman Filters: Why Use Kalman Filters?," 2017. [Online]. Available: https://se.mathworks.com/videos/understanding-kalman-filters-part-1-why-use-kalman-filters--1485813028675.html. [Accessed: 04-May-2020].

[24] G. Hendeby, "Performance and Implementation Aspects of Nonlinear Filtering," Institutionen för systemteknik, 2008.

[25] P. Zarchan and H. Musoff, *Fundamentals of Kalman filtering : a practical approach.* American Institute of Aeronautics and Astronautics, Incorporated, 2015.

[26] F. Gustafsson, *Adaptive filtering and change detection :.* Wiley, 2001.

[27] J. J. LaViola, "A comparison of unscented and extended Kalman filtering for estimating quaternion motion," in *Proceedings of the 2003 American Control Conference, 2003.*, 2003, vol. 3, pp. 2435–2440 vol.3.

[28] A. K. Singh and B. C. Pal, "Chapter 4 - Decentralized Dynamic

Estimation Using PMUs,” in *Dynamic Estimation and Control of Power Systems*, A. K. Singh and B. C. Pal, Eds. Academic Press, 2019, pp. 61–91.

[29] A. S. Paul and E. A. Wan, “RSSI-Based Indoor Localization and Tracking Using Sigma-Point Kalman Smoothers,” *IEEE J. Sel. Top. Signal Process.*, vol. 3, no. 5, pp. 860–873, Oct. 2009, doi: 10.1109/JSTSP.2009.2032309.

[30] R. Van Der Merwe and others, “Sigma-point Kalman filters for probabilistic inference in dynamic state-space models,” OGI School of Science & Engineering at OHSU, 2004.

[31] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, “Empirical Strategies,” in *Experimentation in Software Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 9–36.

[32] C. Wohlin, M. Höst, and K. Henningsson, “Empirical Research Methods in Web and Software Engineering,” in *Web Engineering*, E. Mendes and N. Mosley, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 409–430.

[33] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, “Selecting Empirical Methods for Software Engineering Research,” in *Guide to Advanced Empirical Software Engineering*, F. Shull, J. Singer, and D. I. K. Sjøberg, Eds. London: Springer London, 2008, pp. 285–311.

[34] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, “Experiment Process,” in *Experimentation in Software Engineering*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 73–81.

[35] Android Developers, “RangingResult.” [Online]. Available: https://developer.android.com/reference/android/net/wifi/rtt/RangingResult. [Accessed: 03-Feb-2020].

[36] Fanless IoT Technology by Compulab, “WiFi Indoor Location Device.” [Online]. Available: https://fit-iot.com/web/product/wild/. [Accessed: 18-Feb-2020].

[37] Fanless IoT Technology by Compulab, “WILD software.” [Online].

Available: https://fit-pc.com/wiki/index.php?title=WiFi_Indoor_Location_Device_(WILD)_software#Change_channel_bandwidth. [Accessed: 18-Feb-2020].

[38] Android Developers, "Motion sensors." [Online]. Available: https://developer.android.com/guide/topics/sensors/sensors_motion. [Accessed: 18-Mar-2020].

[39] G. Milette and A. Stroud, *Professional Android Sensor Programming*. Somerset, UNITED STATES: John Wiley & Sons, Incorporated, 2012.

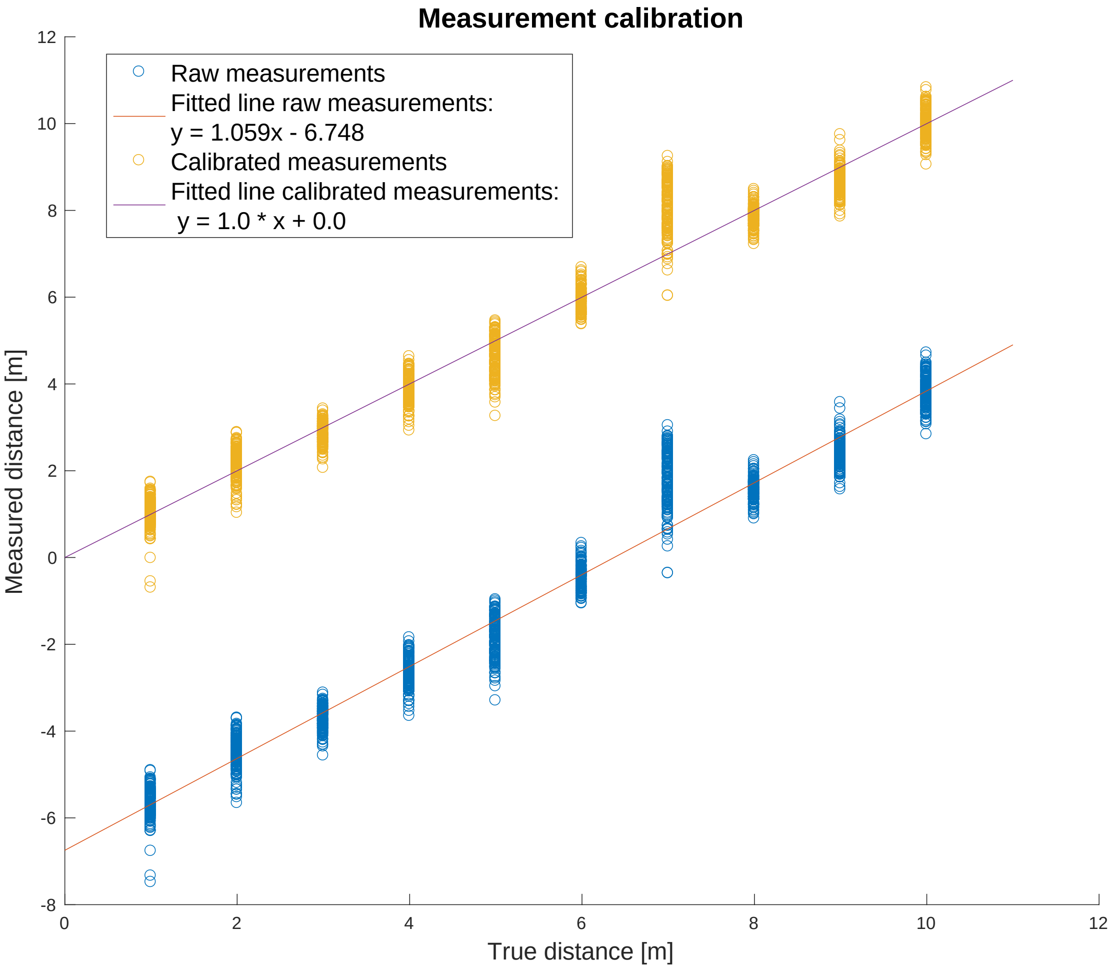
[40] Android Developers, "Wi-Fi scanning overview." [Online]. Available: https://developer.android.com/guide/topics/connectivity/wifi-scan. [Accessed: 12-Mar-2020].

[41] A. Abdi, C. Tepedelenlioglu, M. Kaveh, and G. Giannakis, "On the estimation of the K parameter for the Rice fading distribution," *IEEE Commun. Lett.*, vol. 5, no. 3, pp. 92–94, 2001.

[42] C. G. Koay and P. J. Basser, "Analytically exact correction scheme for signal extraction from noisy magnitude MR signals," *J. Magn. Reson.*, vol. 179, no. 2, pp. 317–322, 2006.

Appendix

Appendix A – Measurement calibration



Statement of Originality and Letter of Authorization

学位论文原创性声明

Statement of Originality

本人郑重声明：此处所提交的学位论文《中文题目 English Title》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：　　　　　　　　　　日期：　　　年　　月　　日

学位论文使用权限

Letter of Authorization

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：　　　　　　　　　　日期：　　　年　　月　　日

导师签名：　　　　　　　　　　日期：　　　年　　月　　日

Acknowledgements

I would like to thank everyone who supported me during this work and made this thesis possible. First and foremost, I would like to express my gratitude to my supervisors at Senion, David Törnqvist and Per Skoglar, for always taking their time to answer my questions and providing me with continuous feedback. I would also like to thank Petru Eles, who has been both my supervisor and examiner during the project, for always being helpful and supportive. Finally, I would like to take the opportunity to thank all my friends for their support, and especially Jesper Holmström, for contributing with valuable comments and proofreading.