Examensarbete utfört i Kommunikationssystem
vid Tekniska högskolan i Linköping
av
Raja Umair Haider

LiTH-ISY-EX--12/4601--SE
Linköping 2012
Fault Detection in WLAN Location Fingerprinting Systems Using Smartphone Inertial Sensors

Examensarbete utfört i Kommunikationssystem vid Tekniska högskolan i Linköping
av

Raja Umair Haider

LiTH-ISY-EX--12/4601--SE

Handledare: Antionios Pitarokoilis
            ISY, Linköpings universitet
Christos Panayiopoulos
            KIOS Research Center
Christos Laoudias
            KIOS Research Center

Examinator: Eleftherios Karipidis
            ISY, Linköpings universitet

Linköping, 12 September, 2012
Titel

Svensk titel
Fault Detection in WLAN Location Fingerprinting Systems Using Smartphone Inertial Sensors

Författare

Author
Raja Umair Haider

Sammanfattning

Abstract

Indoor positioning is a rapidly growing research area, enabling new innovative location-aware applications and user-oriented services. Location Fingerprinting (LF) is the positioning technique of coupling a physical location with observed radio signal measurements. In the terms of indoor LF using Wireless Local Area Network (WLAN) it refers to the use of network measurements from the WLAN Access Points (APs) to tag known locations. A data set is created containing reference fingerprints for the area of interest and is known as a radio map. A radio map can later be used to find a user’s location in the area of interest.

WLAN infrastructures are vulnerable to many kinds of faults and malicious attacks, including, an attacker jamming the signal from an AP, or an AP becoming unavailable during positioning due to power outage. These faults can be collectively characterized as an AP-failure. In LF positioning systems, AP-failure faults can significantly degrade the performance of a LF system due to the difference between the current fingerprints and radio map created with all APs being available. It is desirable to detect such faulty APs, in order to take actions towards fault-mitigation and restoration, in case of a malicious attack.

In this work, we have developed a fault detection algorithm that uses inertial sensors (i.e., accelerometer, magnetometer) available in smartphones to detect AP-failure faults in LF systems. Inertial Measurement Unit (IMU) has become an integral part of all high-end smartphones. IMU can be used to infer location information on the smartphone. The main idea is to have two parallel position streams, the LF positioning and the IMU positioning, and to compare the mean positioning error between the two. Since IMU positioning is fairly accurate once provided with starting coordinates, we use it to detect abnormal behaviour in LF positioning system, such as highly erroneous estimates signifying an AP-failure fault present in the system. The performance of the proposed detection algorithm is evaluated with several real-life AP-related faults. The proposed algorithm exhibits low probability of false alarms in the detection of faulty APs. The conclusion is that using IMU based positioning is an effective and robust solution in terms of fault detection in LF systems.

Nyckelord

Keywords
AP, LF, KF, IMU, WLAN
Abstract

Indoor positioning is a rapidly growing research area, enabling new innovative location-aware applications and user-oriented services. Location Fingerprinting (LF) is the positioning technique of coupling a physical location with observed radio signal measurements. In the terms of indoor LF using Wireless Local Area Network (WLAN) it refers to the use of network measurements from the WLAN Access Points (APs) to tag known locations. A data set is created containing reference fingerprints for the area of interest and is known as a radio map. A radio map can later be used to find a user’s location in the area of interest.

WLAN infrastructures are vulnerable to many kinds of faults and malicious attacks, including, an attacker jamming the signal from an AP, or an AP becoming unavailable during positioning due to power outage. These faults can be collectively characterized as an AP-failure. In LF positioning systems, AP-failure faults can significantly degrade the performance of a LF system due to the difference between the current fingerprints and radio map created with all APs being available. It is desirable to detect such faulty APs, in order to take actions towards fault-mitigation and restoration, in case of a malicious attack.

In this work, we have developed a fault detection algorithm that uses inertial sensors (i.e., accelerometer, magnetometer) available in smartphones to detect AP-failure faults in LF systems. Inertial Measurement Unit (IMU) has become an integral part of all high-end smartphones. IMU can be used to infer location information on the smartphone. The main idea is to have two parallel position streams, the LF positioning and the IMU positioning, and to compare the mean positioning error between the two. Since IMU positioning is fairly accurate once provided with starting coordinates, we use it to detect abnormal behaviour in LF positioning system, such as highly erroneous estimates signifying an AP-failure fault present in the system. The performance of the proposed detection algorithm is evaluated with several real-life AP-related faults. The proposed algorithm exhibits low probability of false alarms in the detection of faulty APs. The conclusion is that using IMU based positioning is an effective and robust solution in terms of fault detection in LF systems.
Acknowledgments

First of all, I would like to thank Christos Panayiotou for providing me with the opportunity to work at KIOS research center, Cyprus. Also a huge thanks to him for supervising my thesis work and for sharing his professional insight throughout the thesis work. My heartiest thanks to co-supervisor at KIOS Christos Laoudias, who actively encouraged me throughout the thesis process. As this thesis work was my first step into the world of academic research, Christos Laoudias has been a source of constant motivation and inspiration. I am grateful to Christos Laoudias for his patience and understanding while supervising my work. I feel inspired towards continuing research work after working with him. I thank Silouanou Nicolaou for his assistance with the thesis work. A huge thanks to all the colleagues at KIOS research center for their kindness throughout my stay and for taking a keen interest in my thesis work. No doubt it has been a memorable experience working at KIOS.

A special thanks to Eleftherios Karipidis who has been the examiner of the thesis work at Linköping University. It would not be possible to improve the quality of the work without his motivation and support. Antonios Pitarokoilis supervised the thesis work at Linköping University. Thank you Antonios for your great help throughout the thesis work. I am thankful to you for always being accessible for sharing of ideas and for your helpful feedback regarding the thesis work and specifically about report writing. Last but not least, I would like to thank Jonas Callmer for his kind help and concern about the progress of this thesis work.

Umair Haider
Linköping, September 2012
# Contents

1 Introduction

1.1 Positioning Systems .......................................................... 1
1.2 Purpose and Goals ............................................................. 2
1.3 Disposition ................................................................. 2

2 Location Fingerprinting Systems ........................................... 3

2.1 Introduction to Location Fingerprinting ................................. 3
2.2 Theory of Location Fingerprinting ........................................ 4
  2.2.1 Definitions ............................................................. 4
  2.2.2 Components of a LF System ......................................... 4
2.3 Positioning Algorithms ....................................................... 6
2.4 Kalman Filter Smoothing of LF Estimates .............................. 8
  2.4.1 Kalman Filter in Present Work ..................................... 10
2.5 Effects of Faults in LF ....................................................... 11
  2.5.1 Single AP-Failure Faults ........................................... 12
  2.5.2 Example of AP-Failure Fault ................................. 12

3 Inertial Measurement Unit Positioning .................................. 15

3.1 Introduction ............................................................... 15
3.2 Micro Electro Mechanical Systems ..................................... 15
  3.2.1 Accelerometer Sensor ............................................ 16
  3.2.2 Geomagnetic Field Sensor ...................................... 16
  3.2.3 Orientation Sensor ................................................ 16
  3.2.4 Gyroscope Sensor ................................................. 16
3.3 The IMU Positioning Module ............................................. 16
  3.3.1 Data Collection from the Smartphone Sensors .................. 17
  3.3.2 PDR Algorithm ..................................................... 18

4 Fault Detection in Location Fingerprinting Systems .................. 21

4.1 Basics of Fault Detection Algorithm ................................... 21
4.2 The Detection Algorithm ............................................... 22
4.3 Threshold Selection .................................................... 22
  4.3.1 Artificial Fault Injection ....................................... 23
List of Figures

2.1 LF System Operation ........................................... 5
2.2 Feedback Operation of Kalman Filter .......................... 11
2.3 Kalman Filter Smoothing of LF Position Estimates ............... 12
2.4 Single AP-failure Fault [1] ....................................... 13

3.1 Device’s Axes Coordinate System [2] .......................... 17
3.2 Raw Acceleration Signal vs. Band-pass Filtered Signal ........... 19
3.3 IMU-Based Position Estimates ................................... 20

4.1 Block Diagram of Fault-detection Algorithm ...................... 22
4.2 Effects of Single AP-failure Fault in LF Positioning .............. 24

5.1 Experimental Area Floorplan ..................................... 28
5.2 Effects of Single AP-failure Fault in LF Positioning .............. 29
5.3 Fault-free Operation inside Narrow Corridor Space .............. 32
5.4 Multiple AP-failure Fault ......................................... 37
5.5 AP-relocation Fault ............................................... 38
5.6 User Walking Along Unknown Route .............................. 40
List of Tables

2.1 RSS Values from APs [dBm] ........................................... 13

4.1 Artificial Fault Injection .............................................. 23

5.1 Mean Positioning Error for Fault-free and AP Faulty Test Cases ... 29
5.2 Mean Positioning Error with Real AP Faulty Test Cases .............. 30
5.3 Comparison of Mean Positioning Error Values ........................ 30
5.4 Percentage of Correct Detection, Open Space Route with $\gamma = 5$ 31
5.5 Mean Positioning Error for Fault-free Test Cases ..................... 31
5.6 Mean Positioning Error for Fault-free and AP Faulty Test Cases ... 33
5.7 Mean Positioning Error with Real AP Faulty Test Cases ............. 33
5.8 Percentage of Correct Detection for Narrow space route with $\gamma = 2$ 33
5.9 Correct Detection Rate for the Algorithm Proposed in [3] ............ 35
5.10 Number of False Alarms for Each Test Case ........................ 36
5.11 Mean Positioning Error for Multiple AP-failure Fault ............... 36
5.12 Mean Positioning Error for AP-relocation Fault ..................... 38
5.13 Mean Positioning Error for AP$_3$ Faulty ............................ 39
# Abbreviations

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP</td>
<td>Acess Point</td>
</tr>
<tr>
<td>BSI</td>
<td>Base Station Identifier</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>KF</td>
<td>Kalman Filter</td>
</tr>
<tr>
<td>KNN</td>
<td>K-Nearest Neighbour</td>
</tr>
<tr>
<td>LBS</td>
<td>Location Based Services</td>
</tr>
<tr>
<td>LF</td>
<td>Location Fingerprinting</td>
</tr>
<tr>
<td>IMU</td>
<td>Inertial Measurement Unit</td>
</tr>
<tr>
<td>MAC</td>
<td>Media / Medium Access Control</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Squared Error</td>
</tr>
<tr>
<td>NAN</td>
<td>Not A Number</td>
</tr>
<tr>
<td>NN</td>
<td>Nearest Neighbour</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant</td>
</tr>
<tr>
<td>PDR</td>
<td>Pedestrian Dead Reckoning</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identification</td>
</tr>
<tr>
<td>RSS</td>
<td>Received Signal Strength</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This report is written as a partial fulfilment for the degree of Master of Science in Communication Systems at Linköping University. The report describes the master thesis work carried out in the spring of 2012 at KIOS Research center for Intelligent Systems & Networks, Cyprus. The work was done in joint collaboration of Linköping University, Sweden and KIOS Research center, Cyprus.

1.1 Positioning Systems

Global Positioning System (GPS) provides accurate navigation and positioning services in outdoor environments. Location Based Services (LBS) help to improve safety, logistics, focused advertising, travelling, security and other applications [4]. LBS have attracted a lot of research in the last decade, and the user-oriented applications revolving around LBS are growing.

GPS signals are weak and do not provide efficient positioning inside buildings and in urban canyon environments. However, since people spend most of their time in indoor environments, such as office buildings, hospitals, educational campuses, shopping malls etc., it is desired to have reliable positioning services indoors. Indoor positioning systems can also be used for asset tracking in hospitals and in large warehouses. Another promising solution based on indoor positioning is the availability of in-building guidance in hotels and conference centres. To solve the problem of indoor positioning, solutions range from all sort of technological backgrounds, using cell phone towers for localization to using Radio Frequency Identification (RFID) tags for close range positioning. One popular approach is to exploit the growing Wireless Local Area Network (WLAN) infrastructure, which is wide spread in urban environments and can be used for providing positioning information.

Location Fingerprinting (LF) is the positioning technique of coupling a physical location with some observed radio signal measurements. In terms of indoor LF using WLAN, it refers to the use of network measurements from WLAN Access Points (APs) to tag known locations, later this data can be used to find a user’s location in the area of interest. Fingerprinting has been the most successful method
to address the indoor positioning problem as it exhibits high positioning accuracy when compared to other systems.

Since LF using WLAN is highly critical of the underlying WLAN network of APs, any fault in these APs can greatly degrade the performance of LF positioning. The power outage to an AP, or jamming its radio signal is characterized as an AP-failure fault. In this work, we have developed an algorithm that uses inertial sensors (i.e., accelerometer, gyroscope and digital compass) present in high-end smartphones to detect AP-failure faults in a LF system.

1.2 Purpose and Goals

The purpose of this thesis is to study the effect of AP-failure faults in LF positioning systems. Such faults in APs can severely degrade the performance of a LF positioning system, therefore we aim at developing efficient algorithms to successfully detect the presence of such faulty APs using inertial sensors present in smartphones. This is expected to improve the Quality of Service (QoS) of the LF systems in general.

The evaluation of the algorithms developed in presence of real-life AP faults is also part of this thesis work. In order to establish solid scientific grounds, the thesis work also includes comparison of our fault detection solution with related work found in literature.

1.3 Disposition

The rest of the report is arranged in the following manner:

**Chapter 2** describes the basic principles and operation of Location Fingerprinting (LF) systems.

**Chapter 3** is about inertial sensors present in high-end smartphones and describes their use to provide a positioning solution.

**Chapter 4** describes the fault detection algorithm, which uses inertial sensors to detect AP-failure faults in LF systems.

**Chapter 5** presents the experimental results of fault detection algorithm developed in this work, along with its application and evaluation with real-life AP-failure faults.

**Chapter 6** presents the conclusions of the thesis work along with some possible future work.
Chapter 2

Location Fingerprinting Systems

This chapter provides the background information and the notation used for LF systems.

2.1 Introduction to Location Fingerprinting

WLAN LF is a positioning technique that uses the WLAN APs and provide location estimates of wireless devices and smartphones. One important advantage of LF over other indoor-positioning systems is the lack of need for any new hardware installation in the desired area of interest. The ability of LF to usefully exploit the IEEE 802.11 network, also known as WLAN makes it cost effective and more likely to be used for indoor positioning. Additionally almost every smartphone and laptop devices includes built-in WLAN adapters, this enables the collection of RSS data from WLAN APs without any other specialized equipment. The massive availability of WLAN APs and smartphones motivates the use of WLAN LF as an attractive solution to the indoor positioning problem.

The word Fingerprinting reflects that each physical location in the observation area has a unique identifier: usually it is comprised of Received Signal Strength (RSS) and Base Station Identifier (BSI), which is the MAC address of the wireless card of WLAN AP. The uniqueness of the fingerprints is based on the fact that WLAN radio signals attenuate during propagation in indoor environments, thus resulting in different RSS values at different locations inside a building. The RSS value is measured in dBm.

Here the physical location is the x and y co-ordinates provided by the floor plan of the building. Using a pre-recorded database of fingerprints, the location of a device inside the area of interest can be inferred by matching the current fingerprint of the device within the database. The area of interest is defined as the area inside the building for which reference fingerprints had been collected a priori. There are several variants of the general system, differing in the method used to
decide on the best match. A detailed overview of the indoor positioning algorithms describing the process of choosing the best match is presented in Section 2.3. In the following, different components of a LF system, and positioning algorithms used in LF are presented.

2.2 Theory of Location Fingerprinting

We start out by presenting the basic concepts and components of a LF system.

2.2.1 Definitions

LF systems use a set of predefined reference locations \( \mathcal{L} = \{\ell_i = (x_i, y_i), i = 1, \ldots, M\} \) to collect RSS values from \( N \) APs deployed in the area of interest. \( N \) is the number of WLAN APs used at the time of collecting the reference fingerprints. This can be any number of APs, typically a higher density of APs provides better positioning accuracy. However studies have shown that the improvement in positioning accuracy reaches a saturation point regardless of adding more APs in the area of interest. The positioning accuracy is the difference between the estimated location of the device and the actual location of the device, it is measured in meters. \( M \) is the total number of reference locations inside the area of interest.

A reference fingerprint \( r_i = [r_{i1}, \ldots, r_{iN}]^T \) associated with location \( \ell_i \), is a vector of RSS samples and \( r_{ij} \) denotes the RSS value related to the \( j \)-th AP. Usually, \( r_i \) is averaged over multiple fingerprints collected at \( \ell_i \) to alleviate the effect of noise in RSS measurements and outlier values. During LF positioning, the reference data is exploited to obtain a location estimate \( \hat{\ell} = (\hat{x}_{LF}, \hat{y}_{LF}) \) given a new fingerprint \( s = [s_1, \ldots, s_n]^T \) measured at the unknown location \( \ell = (x, y) \).

2.2.2 Components of a LF System

A LF system has typically 4 components. The radio map, the WLAN nodes also known as APs, the server, and the wireless terminal, which can be a laptop computer, smartphone, etc. Fig. 2.1 illustrates the system level operation of a LF positioning system and it also illustrates the different system components. These LF system components will be described in the following.

Radio Map

The popular approach in many LF systems is to record fingerprint data at predefined reference locations, \( \mathcal{L} \), in area of interest of the LF system. This collection of fingerprint data is also known as the offline phase of the LF system. The database of the reference fingerprints collected is known as the radio map. Each fingerprint in the radio map is associated with a pre-determined physical location. It is assumed that the WLAN AP coverage is present in the whole area of interest. A smartphone, running a logging application, can be used to collect RSS fingerprints to create a radio map.
So essentially, the radio map contains a database of RSS values and MAC-addresses of the APs at each reference location. Each fingerprint in the radio map is linked to a physical location in form of $x$ and $y$ co-ordinates of the floor plan image. Ideally the reference fingerprints should be uniformly collected over the whole area of interest. However, the grid density is typically non uniform due to building walls, furniture and other instalments inside the area of interest that limit the positions where measurements could be performed. The resolution of reference locations plays an important role in location estimation process. Intuitively a sparsely collected radio map will give location estimates with low positioning accuracy. On the other hand, it is a time consuming and laborious task to collect reference fingerprints at every location inside the area of interest. One rule of thumb is to collect the radio map with grid density of 1 or 2 fingerprints per square meter.

**The WLAN Setup**

The WLAN APs are an intergral component of the LF indoor positioning systems. LF uses RSS and the BSI of radio signals from the APs installed in the area of interest. This information is already present in all WLAN radio beacons, and LF exploits this information at no extra cost. Location inference is done based on the best match between the current RSS values and the ones a priori stored in the radio map. This process is explained in detail in Section 2.3.

**Server**

Depending upon the scale of the LF positioning system, the radio map could be a large data set. This requires the need for a central server, where this map can be kept and also intelligently updated with new location fingerprints, while removing obsolete fingerprints. Most LF systems use a pre-recorded radio map,
however research has been done in developing other methods that can alleviate the
tedious task of map building. Crowd sourcing is one approach in which the users
of the system gradually build radio map. This collaborating approach relieves the
hectic task of creating the radio map a priori, alongside the added advantage of
gradual upgrade. On the other hand, the crowd sourcing approach comes with
its own drawbacks, most importantly the relation between the number of actively
contributing users versus the time required to have a reliable map. Since
the user’s are contributing on a voluntary basis, it can be a slow process of reaching
a suitable density of reference fingerprints. In addition, crowd sourcing assumes
trustworthy users. However, this cannot be guaranteed at all times. A user might
tag a wrong location either by mistake or on purpose. The latter implies that
crowd sourcing can be exploited for malicious attacks against the LF system,
degrading drastically its performance. A malicious attack in this respect could
be deliberate wrong position tagging to affect the performance of LF positioning.
Such wrong position tagging creates fingerprints in the radio map which do not
correspond to the correct location. The presence of wrong reference fingerprints
can severely degrade the positioning accuracy of the LF system. However, the
popular approach in many LF systems is to have pre-recorded radio map. This
radio map file is downloaded to the client device upon request.

Terminal Positioning Device

The wireless client devices can range from smartphones to laptop computers.
Nowadays the growing acceptance of smartphones as the primary hand-held com-
unication device makes them more likely to be used for LF positioning applica-
tions. The requirements for successfully running the indoor positioning solution are
to have built-in IEEE 802.11 chip also known as WLAN adapter, sufficient mem-
ory and processing power. Currently, almost all high-end smartphones, tablets
and laptops are shipped with these basic requirements. Finally the wireless client
needs to install a positioning software application.

This study has used an Android HTC Desire® smartphone. The KIOS posi-
tioning solution used in this research is an application developed in Java program-
ing language for the Android devices.

2.3 Positioning Algorithms

LF systems differ mainly on the use of the positioning algorithm. Popular ap-
proaches use either a deterministic Euclidean algorithm, or a probabilistic algo-


Here, we will present the deterministic K-Nearest Neighbour (KNN) algorithm
and the probabilistic Minimum Mean Squared Error (MMSE) algorithm, which are the two popular algorithms used to estimate \( \hat{\ell}_{LF} \).
2.3 Positioning Algorithms

K-Nearest Neighbour Algorithm

The idea of Nearest Neighbour (NN) \[5\] search is to find the reference fingerprint \( r_i = [r_{i1}, ..., r_{iN}]^T \) associated with location \( \ell_i \), which minimizes the Euclidean distance between the observed fingerprint during positioning \( s \) and the reference fingerprints \( r_i \). The Euclidean distance \( D_i \) is calculated as shown in (2.1).

\[
D_i = \sqrt{\sum_{j=1}^{N} (r_{ij} - s_j)^2} \quad \forall \ i = 1, \ldots, M \tag{2.1}
\]

In (2.1) \( N \) refers to number of APs associated with location \( \ell_i \). The total number of reference fingerprints present in the RSS radio map are denoted by \( M \). The complexity of the NN algorithm increases when \( N \) increases. The complexity is directly proportional to the size of the radio map, which is given by \( O(NM) \).

The location prediction is done based on minimizing the distance measure \( D_i \). A reference location \( \ell_i \) is returned as the predicted location if it minimizes \( D_i \) as shown here, \( \hat{\ell}(s) = \arg \min_{\ell_i} D_i \).

In the K Nearest Neighbours (KNN) method \[5\] location is estimated as the mean of \( K \) reference locations with the shortest Euclidean distances between the observed fingerprint \( s \) and the reference fingerprints \( r_i \). The choice of an appropriate value of \( K \) can be experimentally selected using a smaller set of test fingerprints. It is intuitively seen that there can be more than one neighbours which are fairly close in terms of the Euclidean distance measure, and, thus better accuracy can be achieved in terms of positioning error. Studies such as \[5\], \[7\] have shown that using \( K = 3 \) and \( K = 4 \) yields better accuracy when compared to the basic NN search. However, using much larger values of \( K \), could essentially result in the degradation of the positioning accuracy. This is based on the fact that some of reference locations could be too far from the actual location \( \ell \). Moreover using a larger number of nearest neighbours results in additional computational complexity at the price of a fractional improvement in positioning accuracy. The KNN algorithm has a small computational time and can be executed in real time for positioning.

Minimum Mean Squared Error Algorithm

The probabilistic methods treat the unknown location \( \ell = (x, y) \) as a random variable. Using the conditional probabilities \( p(\ell_i|s), i = 1, \ldots, M \), the unknown location \( \ell \) can be estimated given the measured fingerprint \( s \).

The probabilistic Minimum Mean Square Error (MMSE) method \[6\] uses the conditional probabilities \( p(\ell_i|s) \) and determines location as \( \hat{\ell}(s) = \sum_{i=1}^{M} \ell_i p(\ell_i|s) \). \( M \) is the total number of reference locations inside the area of interest. Applying Bayes’ rule we have

\[
p(\ell_i|s) = \frac{p(s|\ell_i)p(\ell_i)}{p(s)} = \frac{p(s|\ell_i)p(\ell_i)}{\sum_{i=1}^{M} p(s|\ell_i)p(\ell_i)}.
\tag{2.2}
\]
The prior \( p(\ell_i) \) can be assumed to be uniformly distributed for all reference locations in the area of interest. This assumption is motivated by the fact that all reference locations are equally likely to be the actual location of the user. This assumption is valid since the algorithm has no prior knowledge of the location of the user. In addition, using (2.2) the problem reduces to estimating \( p(s|\ell_i) \) and assuming that RSS measurements from neighbouring APs are independent we get
\[
p(s|\ell_i) = \prod_{j=1}^{N} p(s_j|\ell_i).
\]
\( N \) is the total number of APs used to record reference fingerprints.

One way to calculate \( p(s_j|\ell_i) \) from the reference data is to use the probability density function (pdf) of the RSS values collected at location \( \ell_i \) from the \( j \)-th AP. In our implementation of MMSE algorithm, we have used a Gaussian Kernel Density Estimator (KDE) as proposed in [6]. This approach is known as the Kernel approach. The conditional probability is thus calculated using (2.3)
\[
p(s_j|\ell_i) = \frac{1}{N} \sum_{i=1}^{N} K(s_j; r_{ij}).
\]
The Gaussian kernel is expressed in (2.4)
\[
K(s_j; r_{ij}) = \frac{1}{\sqrt{2\pi\sigma}} \exp \left( -\frac{(s_j - r_{ij})^2}{2\sigma^2} \right)
\]
where \( \sigma \) is the kernel width. Selection of an appropriate value of \( \sigma \) is a design problem and it can determined in the offline phase using additionally collected validation fingerprints. In our case, we have used a kernel width of \( \sigma = 6 \).

The positioning error in a LF system is the distance between the estimated location, \( \hat{\ell}_{LF} \) and true location \( \ell \) given by (2.5),
\[
\epsilon = ||\hat{\ell}_{LF} - \ell||^2 = (\hat{x}_{LF} - x)^2 + (\hat{y}_{LF} - y)^2.
\]
The LF position estimates, \( \hat{\ell}_{LF} = (\hat{x}_{LF}, \hat{y}_{LF}) \), are typically oscillating on the floor plan image and do not reflect the motion of the user indoors. These LF position estimates should be filtered in order to remove some erroneous estimates along with better modelling the motion of the user. This can be achieved by using Kalman Filter (KF) smoothing. In the next section, we describe the smoothing of LF estimates through using a KF.

### 2.4 Kalman Filter Smoothing of LF Estimates

The KF is used extensively to estimate the unknown state of a process. KF comprises of a set of mathematical equations that recursively predict a quantity ahead in time, and, correct the prediction value by incorporating new noisy measurements. The wide applicability of KF into problems of state estimation are in part due to its recursive structure, and the fact that the exact model of the process to be estimated is not required to be known in order to apply Kalman filtering. A classic text describing the theory and operation of Kalman filtering can be found in [8].
2.4 Kalman Filter Smoothing of LF Estimates

KF can also be used as a smoothing operator. KF uses a motion model to effectively reflect the motion of an object. In our work, we smooth out the LF position estimates, also removing some outlier estimates which are present even in the fault-free operation of a LF system.

Basics of Kalman Filtering

The KF estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the KF fall into two groups: time update equations and measurement update equations. The simple form of KF time update and measurement update equations are presented in (2.6) and (2.7) respectively.

\[ a_k = \Phi a_{k-1} + \Gamma w_k \]  
\[ b_k = Ma_k + u_k \]  

Where \( k \) represents the time instant \( t_k \). In our case, \( a_k \) is initialized with the LF position estimates and the velocity of the user. For the first iteration of the KF algorithm, \( a_1(t_k) \) and \( a_2(t_k) \) are initialized by the \( x_{LF} \) and \( y_{LF} \) respectively. Similarly, \( b_1(t_k) \) and \( b_2(t_k) \) also contain the initial location of the user in terms of \( x \) and \( y \) co-ordinates. \( v_1(t_k) \) and \( v_2(t_k) \) contain the velocity information of the user along both the \( x \) and \( y \) axes. These are initialized with a zero value assuming that the user is stationary in the beginning.

\[ a_k = \begin{bmatrix} a_1(t_k) \\ a_2(t_k) \\ v_1(t_k) \\ v_2(t_k) \end{bmatrix} \]  
\[ b_k = \begin{bmatrix} b_1(t_k) \\ b_2(t_k) \end{bmatrix} \]  
\[ \Gamma = \begin{bmatrix} \Delta t^2 / 2 & 0 \\ 0 & \Delta t^2 / 2 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \]  
\[ w_k = \begin{bmatrix} w_1(t_k) \\ w_2(t_k) \end{bmatrix} \]  
\[ u_k = \begin{bmatrix} u_1(t_k) \\ u_2(t_k) \end{bmatrix} \]  
\[ \Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \]
\( M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \) \hspace{1cm} (2.14)

The time interval between two consecutive position estimates is given by \( \Delta t \). \( w_k \) is the noise process. It is a Gaussian distributed random vector with zero mean and the covariance matrix \( Q \)

\[ Q = \begin{bmatrix} \sigma_Q^2 & 0 \\ 0 & \sigma_Q^2 \end{bmatrix}. \hspace{1cm} (2.15) \]

The parameter \( \sigma_Q^2 \) can be estimated by the study of the mobility conditions in the area of interest.

The measurement noise is represented by \( u_k \), which is also a Gaussian distributed random vector with zero mean and covariance matrix \( R \)

\[ R = \begin{bmatrix} \sigma_R^2 & 0 \\ 0 & \sigma_R^2 \end{bmatrix}. \hspace{1cm} (2.16) \]

The variance of the LF positioning error in both the \( x \) and \( y \) coordinates is represented by the parameter \( \sigma_R^2 \). This can be set based on the mean positioning error of the LF positioning algorithm in use. For detailed information on the working of the KF, see [8].

The time update equations are used to calculate a priori estimates for the next time step. The current state and error covariance estimates are projected forward in time by using the time update equations. The measurement update equations are used to calculate an improved a posteriori estimate by correcting the a priori estimate [8].

Fig. 2.2 shows the feedback operation of the KF. The time update, and measurement update equations can take different mathematical forms, depending on the particular use of the filter. We have used, our own slightly changed form of the time and measurement update equations to better suit our purpose in LF positioning. In the following, a description of the time update and measurement update equations used in our work is presented.

### 2.4.1 Kalman Filter in Present Work

The input to the KF in our case, are the LF position estimates: \( \hat{\mathbf{r}}_{LF} = (\hat{x}_{LF}, \hat{y}_{LF}) \). The KF uses a motion model, inputting one LF estimate and predicting it ahead in time. KF achieves better modelling of motion of the user, since it has the feedback control, which corrects highly erroneous estimates. The specific KF recursive time update and measurement update equations used in our work are presented in the following. The 1-step ahead prediction is represented by an overhead bar over the quantity.

\[ \mathbf{a}_k = \Phi \mathbf{a}_{k-1} \hspace{1cm} (2.17) \]

In (2.17), the previous estimate is updated by the update matrix \( \Phi \).
2.5 Effects of Faults in LF

In this section, we investigate the operation of LF systems in the presence of faults, specifically effects of WLAN AP-failure faults are studied in detail.

Figure 2.2. Feedback Operation of Kalman Filter

\[
\hat{P}_k = \Phi P_{k-1} \Phi^T + \Gamma Q \Gamma^T
\]  
(2.18)

\[
K_k = \hat{P}_k M^T (M \hat{P}_k M^T + R)^{-1}
\]  
(2.19)

\[
a_k = \alpha_k + K_k (b_k - M \alpha_k)
\]  
(2.20)

\[
P_k = (I - KM) \hat{P}_k
\]  
(2.21)

Where \(I\) is the identity matrix. \(K_k\) is the KF gain and is calculated using (2.19) for every iteration of the KF algorithm. \(P\) is the error covariance matrix. The error covariance matrix \(P\) is initialized by the mean positioning error of the LF estimates. In our case, we have used \(\sigma_R^2 = 3\) meters, which is the mean positioning error of our LF system. Moreover, the time interval between the consecutive LF estimates, \(\Delta t = 1\) second.

The output from the KF, \(\hat{\ell}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF})\) is smooth when compared to the raw LF estimates.

Fig. 2.3 shows the effect of KF smoothing over the \(\hat{\ell}_{LF} = (\hat{x}_{LF}, \hat{y}_{LF})\) estimates. The Kalman filtered estimates are smooth and adequately reflect the motion of the user.
2.5.1 Single AP-Failure Faults

WLAN infrastructures are vulnerable to many kinds of faults and malicious attacks, including, an attacker jamming the signal from an AP, or an AP becoming unavailable during positioning due to power outage. These faults can be effectively characterized as an AP-failure. In LF positioning systems, AP-failure faults can significantly degrade the performance of a LF system due to the difference between the current fingerprints and radio map created with all APs being available.

In the following, the effect of a single AP-failure fault on LF positioning estimates is explained with help of an example.

2.5.2 Example of AP-Failure Fault

Assume that we have collected RSS measurements from 5 APs at 6 distinct locations $\ell_i$, $i = 1, \ldots, 6$ and the fingerprints in the radio map are shown in Table 2.1. In Fig. 2.4, assume that in the fault-free case the fingerprint observed at the unknown location $\ell$ during positioning (is close to $\ell_3$) is $s = [-69, -33, -56, -77, -31]$ in dBm. By using the Euclidean metric of (2.1) we obtain the ordering $\{\ell_3, \ell_2, \ell_4, \ell_5, \ell_1, \ell_6\}$ for the candidate locations with respect to increasing RSS distance. Thus, the standard 1NN method [5] would correctly determine $\ell_3$ as the user location.

Assuming that AP$_2$ has failed and is not detected at the unknown location, the user would observe the corrupt fingerprint $s' = [-69, \text{NaN}, -56, -77, -31]$ in dBm. In this case using (2.1) to calculate the RSS distances would result in the wrong ordering $\{\ell_5, \ell_4, \ell_2, \ell_6, \ell_1, \ell_3\}$, thus introducing high error in the estimated
user location. (The value $-95 \text{ dBm}$ is used to handle the missing RSS value with respect to the radio map.)

Figure 2.4. Single AP-failure Fault [1]

<table>
<thead>
<tr>
<th>Location</th>
<th>$AP_1$</th>
<th>$AP_2$</th>
<th>$AP_3$</th>
<th>$AP_4$</th>
<th>$AP_5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-30</td>
<td>-70</td>
<td>-80</td>
<td>-58</td>
<td>-36</td>
</tr>
<tr>
<td>2</td>
<td>-49</td>
<td>-49</td>
<td>-65</td>
<td>-65</td>
<td>-65</td>
</tr>
<tr>
<td>3</td>
<td>-70</td>
<td>-30</td>
<td>-58</td>
<td>-80</td>
<td>-36</td>
</tr>
<tr>
<td>4</td>
<td>-80</td>
<td>-58</td>
<td>-30</td>
<td>-70</td>
<td>-36</td>
</tr>
<tr>
<td>5</td>
<td>-65</td>
<td>-65</td>
<td>-49</td>
<td>-49</td>
<td>-25</td>
</tr>
<tr>
<td>6</td>
<td>-58</td>
<td>-80</td>
<td>-70</td>
<td>-30</td>
<td>-36</td>
</tr>
</tbody>
</table>

Table 2.1. RSS Values from APs [dBm]
Chapter 3

Inertial Measurement Unit Positioning

This chapter will briefly describe the inertial sensors, the sensor based positioning algorithm and the implementation used in this work.

3.1 Introduction

Since the proliferation of Micro-Electro-Mechanical-Systems (MEMS) into most smartphones, Inertial Measurement Unit (IMU) based positioning services provided on the smartphones are growing. MEMS IMU chips are devices that can sense movement, rotation and gravitational forces, through several built-in sensors, such as, accelerometers and gyroscopes. IMU positioning on the smartphone is a good supplement to other absolute positioning solutions, such as LF and Global Positioning System (GPS). However, IMU positioning lacks the knowledge of self-obtainable start location, thus a starting location coordinates, \( \ell_0 \) should be provided through other means. This could be either a manual entry of position coordinates, or could be acquired from a GPS signal.

3.2 Micro Electro Mechanical Systems

MEMS accelerometers, magnetometers and gyroscopes are soon going to be essential in all smartphones. This high trend in MEMS IMU motivates the development of new positioning and location based services for the end user. In this work, we have used accelerometer and magnetometer MEMS on a HTC Desire© smartphone to derive position and heading information. A brief introduction of the inertial sensors of the smartphone used in this work follows in order to understand the concept of IMU positioning on smartphones.
3.2.1 Accelerometer Sensor

Almost all high-end smartphones available are equipped with a tri-axial accelerometer. This MEMS accelerometer sensor can sense acceleration values in the range of ±2g, where g is the constant for Earth’s gravitational pull and equals to 9.8 m/s². This sensor measures the acceleration force with added effect of the force of gravity. Acceleration values are sensed on three axis (X, Y and Z) of the device. Consider the device layed flat in front of the user, then the Y-axis of the device is tangential to the ground and points towards the current line of sight, X-axis points towards the right, and Z-axis pointing outwards from the screen, perpendicular to the ground. The axes’s orientation is illustrated in Fig. 3.1.

3.2.2 Geomagnetic Field Sensor

The Geomagnetic field sensor provides magnetic field strength at the three coordinate axes. The raw sensor values are measured in micro Tesla (µT). The purpose of this sensor is to provide information about the device’s direction of movement, also known as heading, however geomagnetic field sensor is seldom used alone. Most position inferring applications use this sensor in conjunction with the accelerometer sensor to acquire device’s direction of movement, in form of an angle measured between the magnetic North pole and the device’s Y-axis.

3.2.3 Orientation Sensor

Apart from the hardware sensors available on the device, there are also software-based inertial sensors which derive their data from the hardware sensors. Orientation sensor is one example of a software sensor provided by the Android sensor framework. This sensor provides information about the orientation of the device with respect to the magnetic North pole. This value is in degrees and is called the Azimuth angle, i.e., the angle between the device’s Y-axis and the magnetic North pole. Orientation sensor derives its data from the accelerometer and magnetic field sensors.

3.2.4 Gyroscope Sensor

Gyroscope sensor senses the rotation of the device about its own axis. These rotation values are calculated for all three axis. Gyroscopes are relatively new addition among the MEMS sensors on the smartphones. Coupled with the Geomagnetic field sensor the gyroscope can provide significantly higher accuracy in determining device’s current direction.

3.3 The IMU Positioning Module

The IMU positioning employed in this work is based on the concept of Pedestrian Dead Reckoning (PDR). PDR exploits the kinematics of human movement and can
be defined as the process of estimating the present position by projecting travelled distance and azimuth from a known starting point [9].

3.3.1 Data Collection from the Smartphone Sensors

In this study, an Android HTC Desire® smartphone has been used for the purpose of sensor data collection. A custom application was developed in Java programming language to collect data from the phone’s sensors. Previous studies, such as [10] shows that sampling frequency in the range 16-20 Hz is sufficient to deduce human walking pattern from an acceleration signal. The accelerometer sensor was sampled at approximately 40 Hz, which is well above the minimal required for PDR. For estimation of device’s direction of movement, the orientation sensor was sampled at about 10 Hz. Sensor data was collected with the HTC Desire® smartphone while walking and later post-processed in Matlab® to apply the PDR algorithm.
3.3.2 PDR Algorithm

The PDR positioning algorithm is based on [9]. In this research, a simpler form of the algorithm is used, one that fulfils our purpose at the expense of minimum resources. The applied PDR approach can be divided into two phases, step detection and determination of the step heading. The PDR algorithm detects and counts the number of steps taken, and project each next step with the azimuth angle for that step. Azimuth angle is defined in 3.2.3 For experimental purposes, the step length is held constant at 0.75 meter. Further details on non-constant step length estimation, can be found in [9].

The major indication of a step taken is featured as peak in the vertical accelerometer axis of the phone, relative to the ground. Since device orientation may be tilted, this high peak can be registered in all the three axes. Therefore, the total acceleration signal, \( A \), is calculated as the Root Mean Squared (RMS) of the individual acceleration of \( x, y, \) and \( z \) accelerometer axis, minus the constant gravitational component of \( g = 9.8 m/s^2 \). This is given by (3.1),

\[
A = \sqrt{a_x^2 + a_y^2 + a_z^2} - g. \tag{3.1}
\]

In (3.1) the gravitational component of \( g = 9.8 m/s^2 \) is subtracted from the total acceleration force due to the fact that the accelerometer sensor measures the acceleration with the added force of gravity. Since, we require only the acceleration caused due to the motion of the user, therefore the gravitational component present in the measurements needs to be subtracted. The total acceleration has noise and irregularities, which are filtered by a band-pass filter.

The choice of the band-pass filter is motivated by the fact that we need to remove the high frequency noise present in the sensor measurements and also the low frequency peaks caused due to the shaking of the device. Since the shaking motion of the device also registers a peak in the total acceleration \( A \), a band-pass filter effectively removes these irregularities and noise. The cut-off frequencies are \( f_{c1} = 0.75 \) Hz and \( f_{c2} = 2.75 \) Hz, as advised in [9]. The bandwidth of the bandpass filter is 2 Hz. The raw and filtered acceleration signals are shown in Fig. 3.2.

Each peak in the filtered acceleration signal shows a step taken. A peak-detection algorithm is used to count the number of steps taken. Each step detected by the PDR algorithm is projected along with the direction of movement, this directional information is obtained from the orientation sensor present on the smartphone.

The orientation sensor provides azimuth angle, which is measured in degrees. Azimuth angle is measured as the angle between the device’s Y-axis (when Y-axis is pointing towards the direction of user) and the magnetic North pole. Since the orientation sensor is highly noise prone to outside disturbances, such as metal objects, electrical power lines, to add reliability to the sensor readings, these values are passed through a linear averaging filter as proposed in [9].

In the PDR algorithm, each step detected by the peak detection algorithm is projected ahead by the measured azimuth angle. The azimuth angle is represented by \( \theta \). The trigonometric sin and cos functions are used for the projection of the azimuth angle using the static step length, \( S = 0.75 \) meter. For the projection
in the rectangular coordinates, the azimuth angle measured from the orientation sensor is later converted into radians. Both the $x$ and $y$ position coordinates in $\hat{\ell}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$ are projected using (3.2) and (3.3) respectively. Here it is important to know that the error in the sensor measurement of the azimuth angle is translated into the prediction error of the position coordinates, $(\hat{x}_{IMU}, \hat{y}_{IMU})$

\begin{align*}
x_t &= x_{t-1} + S \sin(\theta_t) \quad (3.2) \\
y_t &= y_{t-1} + S \cos(\theta_t) \quad (3.3)
\end{align*}

Finally, the output of the IMU positioning algorithm is in the form of position estimates, $\hat{\ell}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$

Fig. 3.3 shows the IMU estimates, $\hat{\ell}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$ for a test walk pattern.

From Fig. 3.3 it can be seen that the IMU estimates are fairly accurate when the user is walking in a straight path. The IMU position estimates (depicted with green stars) coincide with the actual locations (shown as red dots), and only at the end of the route the $\hat{\ell}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$ start drifting. However, walking along a straight path for a few seconds guarantees accurate IMU estimates that are adequate for our fault detection purposes.

In order to do fault detection in LF positioning system, it requires to have a secondary source of position estimates, which can be used to detect wrong LF position estimates. The IMU based positioning solution implemented in this work
can be used on commercial smartphones, exhibits fairly accurate position estimates: thus enabling it’s use in fault detection in LF systems. This concept will be explained in detail in the following chapter.
Chapter 4

Fault Detection in Location Fingerprinting Systems

This chapter describes the fault detection algorithm developed in this study.

4.1 Basics of Fault Detection Algorithm

The positioning accuracy of LF systems can be significantly degraded by AP-failure faults due to the difference between the current fingerprints and radio map created with all APs being available. It is highly desired to detect such faulty APs, in order to take actions towards fault mitigation and restoring the APs, in case of a malicious attack.

We have developed a fault detection algorithm that uses the inertial sensors available in off-the-shelf smartphones to detect AP-failure faults in LF systems. The main idea is to have two parallel position streams, the \( \hat{\mathbf{l}}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF}) \) and the \( \hat{\mathbf{l}}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU}) \) streams, and to compute the mean positioning error, \( \bar{\epsilon} \) between the two. These two position estimate streams are collected while the user (security/maintenance personnel) walks along a predefined route. A route inside area of interest is selected and fingerprint data is collected both for fault-free and AP-faulty operation of the algorithm. The same route is used so that the effect of an AP-failure fault can be easily studied, keeping all other factors the same. The fault detection is carried out by studying the \( \bar{\epsilon} \) values. The condition that the user walks along a known route is to facilitate the evaluation of the algorithm. In Section 5.2 it is shown that the algorithm can successfully detect faults while the user walks along an unknown route.

Since IMU positioning is fairly accurate once provided with starting coordinates, we can use it to detect abnormal behaviour in LF positioning system, such as highly erroneous estimates signifying an AP-failure fault present in the system.
4.2 The Detection Algorithm

The detection algorithm takes as input the two streams $\hat{\mathbf{l}}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF})$, $\hat{\mathbf{l}}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$ and outputs a decision about whether the LF system contains a faulty AP or not. Fig. 4.1 illustrates the system level operation of the fault detection algorithm.

![Figure 4.1. Block Diagram of Fault-detection Algorithm](image)

The AP-failure faults are detected by comparing the mean positioning error over the whole path, $\bar{\epsilon}$, defined in (4.1),

$$\bar{\epsilon} = \frac{1}{T} \sum_{t=1}^{T} \epsilon(t), \quad \epsilon(t) = \|\hat{\mathbf{l}}_{KF}(t) - \hat{\mathbf{l}}_{IMU}(t)\|$$  \hspace{1cm} (4.1)

where we have a series of location estimates calculated at time $t : t = 1, \ldots, T$.

If $\bar{\epsilon} \geq \gamma$, it signifies the detection of a AP-failure fault present in the system, where $\gamma$ is an appropriately selected threshold.

4.3 Threshold Selection

Considering the large WLAN setups in certain indoor environments, such as shopping malls and university campuses, it is not practical to turn off APs manually and physically create AP-failure faults in order to select the detection threshold $\gamma$.

For the fault detection algorithm to be applicable an appropriate threshold selection process is required, without the need to create any AP failure faults in the LF system. To solve this problem, our fault detection algorithm works with a threshold value that can be selected by studying fault free walking patterns in LF positioning. The basic idea is to collect position estimates, $\hat{\mathbf{l}}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF})$ while the user walks on pre-defined route in the area of interest, assuming all APs are available for LF positioning: fault-free case. This data is later post-processed...
by artificially injecting faults to study the effects of an AP-failure fault in the system. The next section describes the concept in detail.

### 4.3.1 Artificial Fault Injection

In Section 2.5, we described the effect of an AP-failure fault on LF positioning. During positioning if an AP which is turned off, it’s RSS values in the location fingerprint will be replaced by a Not a Number (NaN) value. This signifies that this particular AP is missing during positioning, thus inducing a large error in the position estimate $\hat{\ell}_{LF}$.

This introduces the concept of artificial fault injection in the positioning data collected when all APs were available for positioning. In artificial fault injection, we manipulate the collected fingerprints through the use of a MATLAB script, replacing the RSS values corresponding to an AP with NaN values, thus simulating the effect of being unavailable during positioning. The mean positioning error, $\bar{\epsilon}$ is calculated using (4.1) after artificial fault injection.

This approach saves the laborious task to collect new data for studying the AP-failure fault in the system, alongside making the fault detection algorithm more practical to be used in large scale WLAN setups.

In the following, we demonstrate the effects of single AP-failure fault injection through an example. Positioning data was collected in the fault-free case while the user is walking along a pre-defined route. This data is later post-processed as described above. Table 4.1 shows indicative RSS values from 4 APs collected at 6 successive locations. The RSS column corresponding to the faulty AP is replaced by NaN values. In the positioning algorithm, these NaN values are replaced by a very low RSS value of -95 dBm. This lower RSS value introduces an error in LF location estimation.

<table>
<thead>
<tr>
<th>Location</th>
<th>$AP_1$</th>
<th>$AP_2$</th>
<th>$AP_3$</th>
<th>$AP_4$</th>
<th>Artificial Fault Injection in $AP_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-30</td>
<td>-70</td>
<td>-80</td>
<td>-58</td>
<td>NaN</td>
</tr>
<tr>
<td>2</td>
<td>-49</td>
<td>-49</td>
<td>-65</td>
<td>-65</td>
<td>NaN</td>
</tr>
<tr>
<td>3</td>
<td>-70</td>
<td>-30</td>
<td>-58</td>
<td>-80</td>
<td>NaN</td>
</tr>
<tr>
<td>4</td>
<td>-80</td>
<td>-58</td>
<td>-30</td>
<td>-70</td>
<td>NaN</td>
</tr>
<tr>
<td>5</td>
<td>-65</td>
<td>-65</td>
<td>-49</td>
<td>-49</td>
<td>NaN</td>
</tr>
<tr>
<td>6</td>
<td>-58</td>
<td>-80</td>
<td>-70</td>
<td>-30</td>
<td>NaN</td>
</tr>
</tbody>
</table>

**Table 4.1. Artificial Fault Injection**

The fault-free and with an AP faulty operation of the LF system is shown in Fig. 4.2. The faulty AP is marked with red color. The effect of a single AP being faulty creates wrong position estimates. The mean positioning error $\bar{\epsilon}$ is calculated for both the fault-free fingerprint data and the artificially created fingerprint data containing a single AP failure fault. The effect of a faulty AP is demonstrated by a significant increase in the mean positioning
error. Thus, the detection of faulty AP is possible by calculating $\bar{\epsilon}$ and checking if it is above a pre-defined threshold value.

The selection of the threshold value requires the calculation of mean positioning error both for fault-free and faulty LF operation. The difference between the two numbers can be exploited to select an appropriate threshold. Hence, a threshold value can be selected if it fulfils the conditions in (4.2),

$$\bar{\epsilon}_{ff} < \gamma < \bar{\epsilon}_{faulty}. \quad (4.2)$$

For example, in the fault-free operation of the LF system we get $\bar{\epsilon}_{ff} = 3.9$ meters. Now using the same fault-free fingerprints and artificially creating fault in AP$_1$, we get $\bar{\epsilon}_{faulty} = 6.7$ meters. Here $\gamma = 5$ meters can be selected as the threshold value to detect AP failure faults. In order to select a suitable threshold value, the mean positioning error should be studied for several APs being faulty in the system. In this way, the threshold selected would be more reliable.

The selection of $\gamma$ implies a trade off between false alarms in the fault-free case and high detection rate when faults are present. In addition, it can be intuitively delivered that selecting a smaller value of threshold could result in some false alarms of AP failure. On the other hand, setting the threshold to a higher value could lead to some AP failure faults left undetected. Hence, threshold selection is left as a design choice to the personnel responsible for the LF system operation.

The threshold selection process comprises of studying a number of artificially created faulty positioning scenarios. In case of large scale implementation of LF systems, it does not require to create artificial faults for all the APs present in the system. The effect of AP-failure faults can be studied for some of the APs present in the system. Since $\bar{\epsilon}$ is a stochastic variable, it is better to study a suitable number of artificially created APs, in order to incorporate all possible effects of faults.
4.3 Threshold Selection

Guidelines to Select $\gamma$

The guidelines to choose an appropriate threshold $\gamma$ are listed as follows:

1. Identify a route in the area of interest, assuming fault-free conditions, collect LF and IMU positioning estimates for this route.

2. Calculate the mean positioning error in position estimates over the whole route as described in (4.1).

3. Create multiple data sets based on the fault-free data files, each containing a different AP being faulty through artificial fault injection as described in Section 4.3.1.

4. Repeat step 2 for the faulty AP cases, by calculating the mean positioning error for the faulty cases.

5. A suitable threshold $\gamma$ should be selected, such that it fulfils the conditions in (4.2).

A route should be selected such that most APs in the area of interest have strong RSS values in that route. The idea is to identify the missing RSS values from faulty APs, however if the APs are far away from the route their RSS values will be very weak and will not play an important role in LF positioning. The route selected also affects the threshold value $\gamma$. For instance, different routes indoors may have different level of positioning accuracy for the LF system. The mean positioning error $\bar{\epsilon}$ calculated in the fault-free operation is a base line value for the threshold. The routes which are significantly different from one another would require selection of different threshold value. In the next chapter the relation between routes and threshold value is further investigated in Section 5.1 and Section 5.1.
Chapter 5

Experimental Results

This chapter presents the performance evaluation of the fault detection algorithm using experimental data.

5.1 Experimental Setup

We start out by first describing our indoor test environment used to collect location fingerprints.

The experiments to collect LF fingerprints were carried out at the premises of the KIOS Research Center for Intelligent Systems & Networks, University of Cyprus. The KIOS Research Center occupies the second floor of a two storey building. This is a typical office environment and the total area is 28.12 x 19.90 $m^2$. The floor has an open plan interior design and consists of a foyer and several cubically shaped workstations. A radio map has been created a priori. A total of 105 locations were used, with 15 fingerprints at each reference location. The reference fingerprints are collected at a sampling rate of 1 fingerprint per second. Through collecting more fingerprints at each location and using the averaged value removes some of the noise present due to the temporal variations in radio signals from WLAN APs.

The reference fingerprints in the radio map were collected using a HP iPAQ© Personal Digital Assistant (PDA), using a dedicated application that records the RSS values from the surrounding APs. The same device has been used in LF positioning to eliminate the cross-platform errors introduced due to different WLAN sensors chips built in the devices.

The WLAN infrastructure inside our experimentation area consisted of four APs. The RSS fingerprints from these four APs have been used to create the radio map file. Fig. 5.1 shows the floorplan, the reference fingerprint locations and the position of the APs inside the area of interest. Additionally it is important to understand that the performance evaluation of the algorithm is carried out when the user walks along the same route as for fault-free and AP-failure cases. This is to verify the performance of the IMU system. However, the main contribution of the thesis work is to use only the IMU positioning stream for fault detection along
any unknown route inside the area of interest. This is later described in detail in Section 5.2.

Figure 5.1. Experimental Area Floorplan

We have used the probabilistic MMSE Kernel algorithm to estimate the unknown location $\hat{\ell}_{LF}$. We have identified one walking route in the open space inside KIOS premises. This route was chosen because all four APs have a strong radio signal in this route. Here on, this walking route will be labelled as the open space route. Fig 5.2a illustrates the user walking along the open space route.

The threshold $\gamma$ is selected by studying both the fault-free and with a single AP-failure operation of the LF positioning system. In the following, the experimental results for these tests are presented. Fig. 5.2 illustrates the positioning performance when a different AP is faulty (shown with red color).

Threshold Selection

We carried out five test walks along a predefined path in the open space inside KIOS premises. The RSS fingerprints were collected which yielded the LF position estimates: $\hat{\ell}_{LF} = (\hat{x}_{LF}, \hat{y}_{LF})$. The LF position estimates over the whole path were later filtered by the use of KF, resulting in $\hat{\ell}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF})$ position estimates. The $\hat{\ell}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$ estimates were also calculated by walking along the same route.

The mean positioning error between the two positioning streams, $\bar{\epsilon}$ is calculated for all five test cases according to (4.1). The artificial fault injection method was used to manipulate the collected fingerprints to respectively simulate the effect of
5.1 Experimental Setup

Figure 5.2. Effects of Single AP-failure Fault in LF Positioning.

AP₁, AP₂ and AP₃ being faulty during LF positioning. Similarly (4.1) is used to calculate \( \bar{\epsilon} \) for all the cases. Table 5.1 contains the results.

<table>
<thead>
<tr>
<th>Case</th>
<th>Fault-free</th>
<th>AP₁</th>
<th>AP₂</th>
<th>AP₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>3.9</td>
<td>6.7</td>
<td>6.8</td>
<td>7.4</td>
</tr>
<tr>
<td>Test 2</td>
<td>4.6</td>
<td>5.8</td>
<td>7.8</td>
<td>6.6</td>
</tr>
<tr>
<td>Test 3</td>
<td>3.9</td>
<td>6.0</td>
<td>6.7</td>
<td>7.0</td>
</tr>
<tr>
<td>Test 4</td>
<td>4.2</td>
<td>5.5</td>
<td>7.9</td>
<td>5.9</td>
</tr>
<tr>
<td>Test 5</td>
<td>4.1</td>
<td>5.5</td>
<td>7.2</td>
<td>6.3</td>
</tr>
<tr>
<td>Mean ( \bar{\epsilon} )</td>
<td>4.1</td>
<td>5.9</td>
<td>7.2</td>
<td>6.6</td>
</tr>
</tbody>
</table>

Table 5.1. Mean Positioning Error for Fault-free and AP Faulty Test Cases

The average value of \( \bar{\epsilon} \) for all five test cases is used when selecting the threshold. As shown in Table 5.1, there is a significant increase in \( \bar{\epsilon} \) when an AP was
Experimental Results

unavailable during positioning. A threshold value, $\gamma = 5$ meters is selected from the results shown in Table 5.1. This value is selected as it fulfils the condition in (4.2).

In this section, the performance of fault detection algorithm is evaluated while the user walks along the open space route. For the open space route the threshold is set to $\gamma = 5$ meters. The open space route presents the typical indoor office environment. This threshold value is selected following the process of threshold selection presented in Section 4.3. For each AP being faulty, LF fingerprints have been collected via five test walks along the open space route. This data was collected in real fault scenarios, where $AP_1$, $AP_2$ and $AP_3$ were turned off manually. The important thing here is that now the fault detection algorithm will be evaluated in presence of real AP-failure faults in the system.

We will evaluate the performance of the AP-failure fault detection algorithm in the scenario of a user walking in the open space route.

### Percentage of Correct Fault Detection

The mean positioning error $\bar{\epsilon}$ in meters for each of the five test cases is shown in Table 5.2.

<table>
<thead>
<tr>
<th>Case</th>
<th>$AP_1$</th>
<th>$AP_2$</th>
<th>$AP_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>5.9</td>
<td>8.0</td>
<td>6.3</td>
</tr>
<tr>
<td>Test 2</td>
<td>5.0</td>
<td>8.7</td>
<td>6.4</td>
</tr>
<tr>
<td>Test 3</td>
<td>4.9</td>
<td>7.9</td>
<td>4.7</td>
</tr>
<tr>
<td>Test 4</td>
<td>5.9</td>
<td>9.1</td>
<td>5.6</td>
</tr>
<tr>
<td>Test 5</td>
<td>5.1</td>
<td>9.0</td>
<td>6.2</td>
</tr>
</tbody>
</table>

Table 5.2. Mean Positioning Error with Real AP Faulty Test Cases

The results in Table 5.2 are acquired with real life AP failure faults. The threshold $\gamma$ is selected through studying artificially created AP faults, as described in Section 4.3.1. Table 5.1 shows the mean positioning error after artificial fault injection. Here, it would be interesting to statistically compare the two results. Table 5.3 shows the mean values for five test cases of artificial fault injection and tests with real AP-failure fault. $AP_2$ exhibits similar mean values for $\bar{\epsilon}$ for the two cases of artificial fault injection and real AP fault. However, these numbers differ in case of $AP_1$ and $AP_3$.

<table>
<thead>
<tr>
<th>Mean $\bar{\epsilon}$</th>
<th>$AP_1$</th>
<th>$AP_2$</th>
<th>$AP_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial Fault Injection</td>
<td>5.9</td>
<td>7.2</td>
<td>6.6</td>
</tr>
<tr>
<td>Real AP-failure Fault</td>
<td>5.3</td>
<td>7.1</td>
<td>5.8</td>
</tr>
</tbody>
</table>

Table 5.3. Comparison of Mean Positioning Error Values
5.1 Experimental Setup

Table 5.4 shows the performance of the fault detection algorithm in terms of percentage of correct fault detection, for each of three AP being faulty respectively.

<table>
<thead>
<tr>
<th>Faulty AP</th>
<th>AP1</th>
<th>AP2</th>
<th>AP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Detection (%)</td>
<td>80</td>
<td>100</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 5.4. Percentage of Correct Detection, Open Space Route with $\gamma = 5$

Percentage of False Alarms Under Fault-free Conditions

A fault detection algorithm is desirable to have a low probability of false alarms, i.e., ideally no or only rare detections should occur in the fault-free case. We collected five data files, walking along a route inside KIOS premises, under fault-free conditions. These files are then used to evaluate the probability of false alarms of the algorithm with the pre-selected threshold value of $\gamma = 5$ meters. In our evaluation data set of five test walks collected under fault-free conditions, the detection algorithm performed with 100% accuracy, resulting in no false alarms. The mean positioning error values for the 5 test cases is shown in Table 5.5. It shows that the threshold value of $\gamma = 5$ was indeed not exceeded.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean Positioning Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>3.1</td>
</tr>
<tr>
<td>Test 2</td>
<td>4.1</td>
</tr>
<tr>
<td>Test 3</td>
<td>3.9</td>
</tr>
<tr>
<td>Test 4</td>
<td>3.9</td>
</tr>
<tr>
<td>Test 5</td>
<td>4.2</td>
</tr>
</tbody>
</table>

Table 5.5. Mean Positioning Error for Fault-free Test Cases

We conclude that the fault detection algorithm works satisfactorily as shown by the results in Table 5.4 and Table 5.5. Typically the probability of false alarms is compared with the probability of missing a fault detection. Table 5.4 shows that $AP_1$ and $AP_3$ have 20% probability of missing a AP-failure fault. This implies that a lower value of threshold $\gamma$ could lead to detecting these 20% no detections at the expense of some false alarms. The selection of threshold value is always selected in the light of this trade-off.

In order to study the effect of differing indoor environments, the fault detection algorithm is also evaluated inside a narrow corridor space inside KIOS premises. Due to the presence of a thick wall and the fact that some APs are situated away from the narrow corridor, it presents a unique environment to study the performance of the detection algorithm. The RSS values from the APs located away from the narrow corridor space attenuate significantly inside the narrow corridor. These APs do not play an important role in LF positioning when the
user walks inside the narrow corridor. One AP is located inside the corridor space, this AP exhibits very strong RSS values inside the corridor, thus is important in LF positioning. Fig. 5.3 illustrates the $\hat{l}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF})$ position estimates and the $\hat{l}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$ estimates, when the user is walking along a route in the narrow corridor space. Here, this route will be known as the narrow corridor route. The procedure to evaluate the performance of the detection algorithm is the same as described before for the open space route in Section 5.1. LF positioning data was collected via five tests while the user walks along the narrow corridor route. This data was collected under fault-free operation of the LF system.

The mean positioning error $\bar{\epsilon}$ between the $\hat{l}_{KF} = (\hat{x}_{KF}, \hat{y}_{KF})$ position estimates, and the $\hat{l}_{IMU} = (\hat{x}_{IMU}, \hat{y}_{IMU})$ estimates was calculated. Later on, artificial fault injection method was used to simulate the effect of AP failure in AP$_1$, AP$_2$ and AP$_3$. The mean positioning error for fault-free and faulty AP cases is shown in Table 5.6. The mean positioning error shown in Table 5.6 is calculated through artificial fault injection methods described in Section 4.3.1. Based on the results shown in Table 5.6 we selected a threshold value of $\gamma = 2$ meters. This threshold value is selected such that it fulfils the condition in (4.2). In Table 5.6 it is evident that the positioning error is the largest when AP$_1$ is faulty. This is due to the fact that AP$_1$ has the strongest RSS values inside the narrow corridor space and thus its failure causes the largest positioning error. Additionally, it can be observed in Table 5.6 that failure of AP$_2$ does not result in a large positioning error. For instance, test 1 results in positioning error of 0.7 meters which is less that the positioning error in the fault-free case. This
5.1 Experimental Setup

<table>
<thead>
<tr>
<th>Case</th>
<th>Fault-free</th>
<th>$AP_1$</th>
<th>$AP_2$</th>
<th>$AP_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>1.2</td>
<td>6.9</td>
<td>0.7</td>
<td>1.7</td>
</tr>
<tr>
<td>Test 2</td>
<td>1.5</td>
<td>10.7</td>
<td>1.7</td>
<td>1.9</td>
</tr>
<tr>
<td>Test 3</td>
<td>1.1</td>
<td>12.2</td>
<td>1.0</td>
<td>1.7</td>
</tr>
<tr>
<td>Test 4</td>
<td>0.8</td>
<td>6.9</td>
<td>6.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Test 5</td>
<td>1.5</td>
<td>11.6</td>
<td>1.5</td>
<td>1.7</td>
</tr>
<tr>
<td>Mean $\bar{\epsilon}$</td>
<td>1.2</td>
<td>9.6</td>
<td>2.1</td>
<td>1.8</td>
</tr>
</tbody>
</table>

**Table 5.6.** Mean Positioning Error for Fault-free and AP Faulty Test Cases

Since $AP_1$ is located inside the narrow corridor space, it exhibits high RSS values in the reference fingerprints corresponding to the areas inside the corridor. The failure of $AP_1$ creates a large error in mean positioning error because the RSS values in the positioning fingerprints are replaced by a very low RSS value. This creates a large Euclidean difference between the reference fingerprints and the positioning fingerprints corresponding to $AP_1$. This leads to estimation of wrong position estimates by the LF positioning algorithm, thus creating a huge mean positioning error. Failure in $AP_1$ is detected with 100% success rate. However,

<table>
<thead>
<tr>
<th>Case</th>
<th>$AP_1$</th>
<th>$AP_2$</th>
<th>$AP_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>6.8</td>
<td>2.3</td>
<td>2.1</td>
</tr>
<tr>
<td>Test 2</td>
<td>5.3</td>
<td>1.1</td>
<td>1.6</td>
</tr>
<tr>
<td>Test 3</td>
<td>4.7</td>
<td>1.0</td>
<td>1.7</td>
</tr>
<tr>
<td>Test 4</td>
<td>5.0</td>
<td>0.4</td>
<td>1.8</td>
</tr>
<tr>
<td>Test 5</td>
<td>5.7</td>
<td>1.0</td>
<td>1.6</td>
</tr>
</tbody>
</table>

**Table 5.7.** Mean Positioning Error with Real AP Faulty Test Cases

The percentage of correct detection based on the mean positioning error values is shown in Table 5.8

<table>
<thead>
<tr>
<th>Faulty $AP$</th>
<th>Correct Detection (%)</th>
<th>$AP_1$</th>
<th>$AP_2$</th>
<th>$AP_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Detection (%)</td>
<td>100</td>
<td>20</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

**Table 5.8.** Percentage of Correct Detection for Narrow space route with $\gamma = 2$
Experimental Results

$AP_2$ and $AP_3$ are located away from the corridor space and their radio signal is heavily attenuated inside the corridor because of the wall. The RSS values from $AP_2$ and $AP_3$ are a small number under fault-free operation of the LF system. Thus replacing these RSS values by a -95 dBm does not create a large Euclidean difference between the reference fingerprints and the positioning fingerprints. This failure of $AP_2$ and $AP_3$ does not create huge positioning error between the KF and LF positioning streams. This is evident by the low percentage of successful detection of AP failure as shown in Table 5.8.

5.1.1 Comparison of Fault Detection Algorithm with Related Work

The fault detection algorithm developed in this work is compared with a related work [3] on fault detection in LF positioning systems. However, their solution is not based on using inertial sensors present on the smartphone.

In [3], the authors detect the presence of an AP-failure fault by comparing subsequent LF position estimates. The main idea is that the positioning error between current and previous LF estimates can signify a possible AP failure fault. The user is assumed to be walking indoors at a steady pace and LF estimates are sampled every 1.5 seconds. This implies that the user can not travel a large distance in such short time interval. When the subsequent LF estimates differ more than a pre-selected threshold value, it could be due to a AP-failure error in the LF system. The threshold value proposed in [3] is 2.13 meter. If the positioning error between two subsequent LF estimates is greater than the threshold value, the system enters the integrity monitoring approach. The integrity monitoring approach is based on removing the effect of faulty AP in LF positioning. However, we are only interested in fault detection part proposed in [3]. More information on how integrity monitoring approach works can be found in [3].

Here we will compare the fault detection system proposed in [3] with our detection algorithm. The comparison will be done in terms of correct detection rate and the number of false alarms generated by the algorithms. The positioning fingerprints for this evaluation were collected while the user is walking inside KIOS premises. Five test walks were carried out under fault free conditions, each containing 28 LF estimates. Here we would like to point out that these LF estimates were sampled at 1 second time intervals. Since in our experimental setup the LF estimates are sampled at a rate of 1 second, the detection threshold of 2.13 meters proposed in [3] is lowered in order to adapt to their algorithm. As the sampling rate of collecting LF estimates is reduced to 1 second rather than 1.5 seconds proposed by [3]. The detection threshold is also lowered by factor of 3, thus lowering the threshold proposed by [3] to 1.42 meters. Another important difference to be known is that our fault detection algorithm detects an AP-failure fault over the whole route. Whereas, the approach presented in [3] detects faults on a per-sample basis.

The algorithm proposed in [3] performs a per-sample fault detection whereas our algorithm performs detection over the whole route. In order to have a fair comparison, their per-sample detection is converted to a per-route detection with
5.1 Experimental Setup

For example, out of the 28 LF estimates, if their algorithm detects more than 14 instants of positioning error exceeding the threshold, it will be counted as one detection over the whole route. Firstly we compare the two algorithms in terms of successful detection percentage while \( AP_1 \), \( AP_2 \) and \( AP_3 \) are respectively turned off during LF positioning. The threshold value of \( \gamma = 5 \) meters is used in our detection algorithm.

Table 5.9 shows the results for the 5 test cases. The overall percentage is also listed.

<table>
<thead>
<tr>
<th>Case</th>
<th>( AP_1 )</th>
<th>( AP_2 )</th>
<th>( AP_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>Detected</td>
<td>Detected</td>
<td>Undetected</td>
</tr>
<tr>
<td>Test 2</td>
<td>Detected</td>
<td>Undetected</td>
<td>Detected</td>
</tr>
<tr>
<td>Test 3</td>
<td>Detected</td>
<td>Undetected</td>
<td>Undetected</td>
</tr>
<tr>
<td>Test 4</td>
<td>Detected</td>
<td>Undetected</td>
<td>Detected</td>
</tr>
<tr>
<td>Test 5</td>
<td>Undetected</td>
<td>Undetected</td>
<td>Undetected</td>
</tr>
<tr>
<td>Correct Detection (%)</td>
<td>80</td>
<td>20</td>
<td>40</td>
</tr>
</tbody>
</table>

Table 5.9. Correct Detection Rate for the Algorithm Proposed in [3]

The results presented in Table 5.9 can be compared with the correct detection rates for our algorithm presented in Table 5.4. This comparison is performed when the user is walking along the open space route presented in Section 5.1. In case of \( AP_1 \) being faulty both our algorithm and the algorithm proposed in detect with 80 % success rate. However, our solution provides better success rate in case of \( AP_2 \) and \( AP_3 \) being faulty.

Now, we compare the performance of our detection algorithm with their solution in terms of number of false alarms. The results of the performance comparison is shown in Table 5.10. The two detection threshold values differ significantly. The significant difference between the two threshold value is due to the fact that the threshold value of 1.42 meters is adapted from the detection algorithm presented in [3]. Additionally, [3] uses only the LF estimates for fault detection which is completely different from our detection approach of using the IMU positioning stream for fault detection. The threshold value of \( \gamma = 5 \) meters is selected following the guidelines presented in Section 4.3. Thus the two threshold values compared in Table 5.10 are justified to be different.

As shown in Table 5.10, the fault detection algorithm proposed in [3] delivers high number of false alarms. This is due to the fact that LF estimates are sporadic in nature even under fault free conditions. These fluctuations are due to temporal changes in the RSS fingerprints collected during positioning. The same fingerprint data was used to evaluate the performance of our algorithm. As shown in Table 5.10 our algorithm does not cause any false alarms. When compared with 60 % false alarms caused by the algorithm proposed in [3], this is a major advantage of our solution. This motivates the use of redundant positioning stream, which in our case is the IMU positioning stream. The mean positioning error between the Kalman filtered position estimates and the IMU estimates is an efficient way to
detect AP-failure faults in LF systems.

Evaluation in Presence of Multiple AP-failure Faults

In this section, the performance of our fault detection algorithm will be evaluated in presence of multiple AP-failure faults in LF system. Multiple AP-failures are expected to result in a even larger value of mean positioning error, $\bar{\epsilon}$. Fingerprint data was collected for five test walks while AP 1 and AP 3 were turned off simultaneously. The same detection threshold of $\gamma = 5$ meters is used for these evaluation tests.

The detection algorithm detects multiple AP-failure with 100% accuracy. This is also intuitively understood, since multiple AP failures are expected to create large differences between the radio map and the positioning fingerprints. This is further demonstrated by the large mean positioning error between the KF and IMU streams. Table 5.11 shows the mean positioning error for the five test cases. The unit of measurement for $\bar{\epsilon}$ is meters.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean Positioning Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>9.3</td>
</tr>
<tr>
<td>Test 2</td>
<td>10.1</td>
</tr>
<tr>
<td>Test 3</td>
<td>11.0</td>
</tr>
<tr>
<td>Test 4</td>
<td>11.2</td>
</tr>
<tr>
<td>Test 5</td>
<td>10.2</td>
</tr>
</tbody>
</table>

Table 5.11. Mean Positioning Error for Multiple AP-failure Fault

Fig. 5.4 illustrates the operation of the LF system in presence of multiple APs being faulty. In this particular scenario, $AP_1$ and $AP_3$ are faulty. The position estimates in case of multiple AP-failure are pushed back in the wrong direction, as depicted in the Fig. 5.4.
Evaluation in Presence of AP-relocation Fault

Another type of AP related fault is caused when the physical position of an AP is changed after collecting the radio map. This causes a difference in RSS values in the positioning fingerprints. This fault is known as an AP-relocation fault. Here, we will evaluate the performance of the detection algorithm in terms of detecting such AP-relocation faults. Fig. 5.3 illustrates the operation of LF system under AP-relocation fault.

Fingerprint data is collected while $AP_1$ inside KIOS premises is moved to a new location. Five fingerprint data collection tests are carried out. The objective is to investigate if the threshold selected for AP-failure faults can also be used for detecting AP-relocation faults. The threshold value of $\gamma = 5$ meters originally selected after artificial fault injection is investigated. Table 5.12 shows the mean positioning error $\bar{\epsilon}$ after $AP_1$ has been relocated. It is clear that AP-relocation creates a significant increase in the $\bar{\epsilon}$ values. The positioning error under fault free operation is also presented for comparison.

The operation of the LF positioning while $AP_1$ is relocated is illustrated in Fig. 5.5. It is clear that the performance of the system is severely affected by relocating $AP_1$ to a new location.

Based on the mean positioning error values presented in Table 5.12 the algorithm detects AP-relocation faults with 80% correct detection rate. This high success rate leads to the conclusion that the same threshold value can be used to detect both AP-failure and AP-relocation faults. This fact makes evident the effectiveness of the detection algorithm.
5.2 Evaluation Along an Unknown Route

Up till now, the detection algorithm was evaluated under the condition that the user walks along a known route. This route has to be the same as for fault-free operation of the LF system. This greatly limits the use of the detection algorithm and undermines the IMU positioning solution. We detect the faults based on the mean positioning error between the LF estimates and the IMU estimates. The mean positioning error is calculated over the whole route. The actual route of the user is not required to be known, since our fault detection algorithm detects faults based on the mean positioning error of the LF and IMU positioning streams.

A fault detection algorithm that can be used anywhere inside the area of interest to successfully detect AP-failure faults is desired.

The performance of the detection algorithm will now be evaluated when the
5.2 Evaluation Along an Unknown Route

User walks along a unknown route. The motivation is to check if the AP-failure faults can be detected along any route inside the area of interest. The only limiting condition is that the user walks in a straight line. This is due to the limitations of the IMU based positioning, which is accurate when the user walks in a straight line. Fingerprint data was collected while $AP_3$ was manually turned off, thus causing AP-failure fault in the LF position estimates. The important point here is that the user walks along a completely unknown route inside the area of interest. The objective is to detect if the detection algorithm can detect AP-failure fault present in the system. The user also tracks itself by the IMU positioning. The mean positioning error $\bar{\epsilon}$ between the two streams will signify the presence of an AP-failure fault.

Considering the test data set comprising of five test cases, each with the user collecting LF fingerprints while walking along an unknown route. This test was evaluated for performance in terms of successful detection. Table 5.13 shows the mean positioning error values for the 5 test cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>Mean Positioning Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test 1</td>
<td>7.3</td>
</tr>
<tr>
<td>Test 2</td>
<td>8.7</td>
</tr>
<tr>
<td>Test 3</td>
<td>8.1</td>
</tr>
<tr>
<td>Test 4</td>
<td>8.2</td>
</tr>
<tr>
<td>Test 5</td>
<td>7.4</td>
</tr>
</tbody>
</table>

Table 5.13. Mean Positioning Error for $AP_3$ Faulty

As shown in Table 5.13 $\bar{\epsilon}$ values for all the test cases are above the threshold value of $\gamma = 5$ meters. The algorithm detects AP-failure with 100% accuracy. This removes the limitation that we had set while deriving the algorithm. The limitation that the user should walk along the same route for both fault-free and faulty operation of the LF system. This limitation was set forth in order to ensure that same system characteristics prevail for all the test cases. The main idea was to keep all other factors same and check how much the mean positioning error varies as a result of a AP-failure fault. In this way, we freeze the effect of AP-failure fault and can ensure that the increase in the mean positioning value is only due to the AP-failure fault. However, in a practical implementation scenario this greatly limits the applicability of the algorithm. Also, the question arises that if the user is walking along the same path, why do we need the IMU positioning to detect the AP-failure? This is a valid question. The IMU positioning stream is required when the user is walking along an unknown route and the only way to calculate the mean positioning error is to have a redundant positioning source. This in our case is the IMU based positioning that derives the positioning information from inertial sensors on the smartphone. The fact that our algorithm can detect AP-failure faults successfully when the user is walking along a unknown route highlights the applicability of the algorithm. In terms of implementation, it is clear that our algorithm can detect AP-failure faults anywhere in the area of interest. Hence,
Experimental Results

the use of the algorithm is not limited to walking along the same route at all times. The user can walk and detect possible AP related faults along any route inside the area of interest. This test also leads to the conclusion that the threshold selected for one route is also effective for a completely different route.

Fig. 5.6 illustrates the operation of the LF system while the user is walking along an unknown path.

![Figure 5.6. User Walking Along Unknown Route](image-url)
Chapter 6

Conclusions and Future Work

This chapter describes the conclusions of this thesis work alongside future work and possible enhancements.

6.1 Discussion and Conclusions

In this section, we discuss and conclude the performance of the fault detection algorithm developed in this work. As shown by the performance evaluation in Chapter 5 the detection algorithm can successfully detect several types of AP based faults in a LF system. It is a threshold based detection algorithm with a simple procedure of selecting an appropriate threshold value as described in Section 4.3.

The performance of the fault detection algorithm has been evaluated under several kinds of AP related faults. Experimental results show that AP-failure and AP-relocation faults can significantly affect the performance of a LF system. The AP-failure and AP-relocation faults are detected with high success rate as shown in Table 5.4 and Table 5.12 respectively.

The performance of a fault detection algorithm also depends upon the number of false alarms generated by the algorithm. Our fault detection algorithm performs exceptionally well with no false alarms for the 5 test data sets as shown in Table 5.5.

The algorithm developed in this work provides a robust solution towards effective fault detection in LF systems. One advantage is that it does not require any dedicated hardware and can be used on the off-the-shelf smartphones.

6.2 Future Work

This section briefly describes the possible future enhancements of the current work.
Development of an Application

At present, the fault detection algorithm is applied offline through the use of Matlab© computing language. A possible future work can be to develop an application for Android smartphones that can perform online fault detection using our algorithm.

Fault Identification

Fig. 5.2 illustrates the effect of AP-failure faults for several APs in a LF system. Every AP results in different pattern of faulty location estimates. This can be exploited by using a pattern recognition approach to identify exactly which AP is faulty. This can be a very useful application for LF systems using large scale WLAN setups. Identifying exactly which AP is faulty can help in taking timely measures towards restoring that particular AP.

Enhancement of PDR Algorithm

IMU based positioning solution developed in this work uses a basic PDR approach. The IMU position estimates are accurate once the user is walking along a straight path. The sensory information derived from the smartphone inertial sensors (i.e., accelerometer, magnetometer) is highly noise prone. Additionally, the magnetometer sensor is affected by outside noise sources such as electrical cables and metal reinforcements inside the buildings. This causes the sensor readings to drift away after a few seconds, this is known as the drifting effect. It is needed to develop advanced signal processing algorithms to alleviate the effect of the noise in magnetometer’s sensory readings. This can lead to improving the IMU based positioning.
Bibliography


