

INDOOR TRACKING BY PARTICLE FILTER COMBINING CIR-BASED RANGING AND INERTIAL SENSORS

Master Thesis
University of Bern

presented by

Danilo Burbano Acuña
2015

Head of Research:
Professor Dr. Torsten Braun
Institute of Computer Science

Contents

Contents	i
List of Figures	iii
List of Tables	v
1 Introduction	3
1.1 Motivation	3
1.1.1 Navigation and Pedestrian Dead Reckoning	4
1.1.2 Monte Carlo Localization	4
1.2 WiFi Indoor Localization Issues	5
1.3 Outline	5
2 Related Work	7
2.1 WiFi Indoor Localization	7
2.1.1 Active Localization Systems	8
2.1.2 Passive Localization Systems	8
2.2 Pedestrian Dead Reckoning	9
2.2.1 Sensor Fusion	9
2.2.2 Full Sensor Fusion	10
2.3 WiFi Indoor Tracking	11
2.3.1 Kalman Filter	11
2.3.2 Particle Filter	12
3 Theoretical Background and Underlying Principles	15
3.1 Physical Layer in IEEE 802.11n	15
3.1.1 Physical Layer (PHY) Frame Structure	16
3.1.2 802.11n Packet Format	17
3.1.3 Orthogonal Frequency Division Multiplexing	17
3.1.4 From RSSI to Channel Impulse Response	21
3.2 Inertial Sensors	24
3.2.1 Accelerometer	24
3.2.2 Magnetometer	25
3.3 Monte Carlo Methods	27

3.3.1	Problem Statement	28
3.3.2	Bayes Filtering	29
4	Indoor Tracking by fusing PDR and CSI Information via Particle Filter	33
4.1	Velocity Estimation	33
4.1.1	Speed Component	36
4.1.2	Heading Direction Component	36
4.2	Ranging Algorithm	42
4.3	Data Fusion	44
5	Indoor Tracking System Implementation	47
5.1	Testbed Overview	47
5.2	Hardware Set-up	48
5.2.1	Anchor Nodes	49
5.2.2	Target Node	51
5.2.3	Central Server	51
5.3	Software Set-up	52
5.3.1	Target Node	53
5.3.2	Anchor Nodes	54
5.3.3	Central Server	55
6	Evaluation	57
6.1	Environment Configuration	57
6.1.1	Stationary Target Set-up	59
6.1.2	Mobile Target Set-up	60
6.2	Results Analysis	62
6.2.1	Comparative Analysis	65
6.2.2	Fusion Particle Filter Performance	66
7	Conclusions	69
	Bibliography	71

List of Figures

1.1	Proposed Indoor Tracking System	5
3.1	PHY Frame Structure	17
3.2	802.11 Packet Format - Mixed Format (MF)	18
3.3	802.11 Packet Format - Greenfield (GF)	19
3.4	Digital Implementation of a Baseband OFDM System)	20
3.5	Multipath propagations, received signals, and channel responses	22
3.6	Magnetic field components in Cartesian coordinate system	26
3.7	Earth's magnetic field in Cartesian coordinate system	27
3.8	Colour map depicting the declination angle in degrees	28
4.1	Flow chart for the algorithm used to compute the velocity	34
4.2	Portrait orientation	35
4.3	Landscape orientation	35
4.4	Sensor Coordinate System for Tracking Experiments	35
4.5	Azimuth Angle	41
4.6	Channel Impulse Response Tracking Experiments	42
4.7	NLR model	44
5.1	Network Localization System	48
5.2	Commercial EeeBox PC	49
5.3	Disassembled ePC part one	50
5.4	Disassembled ePC with antennas part two	50
5.5	Intel WiFi Wireless Card 5300	51
5.6	Testbed Software Implementation	52
5.7	IEEE 802.11n packet with payload format	54
6.1	Third floor's blueprint from INF building	58
6.2	Floor plan with four anchor nodes (ANs)	58
6.3	Floor plan with five anchor nodes (ANs)	59
6.4	Floor plan with stationary positions	59
6.5	Floor plan with stationary training positions	60
6.6	Ground truth movement for tracking one	61
6.7	Ground truth movement for tracking two	61

6.8	Ground truth movement for tracking three	62
6.9	Markers for tracking in corridor	63
6.10	Markers for tracking in office two	63
6.11	Markers for tracking in office three	64
6.12	Markers for tracking in server room	64
6.13	Localization Error for Tracking Experiments	65
6.14	Localization Error for Fusion Particle Filter Experiments	67

List of Tables

3.1	IEEE 802.11n PHY Standard	16
3.2	802.11n Packet Format (refer to figure 3.2 and 3.3)[1]	18
3.3	RSSI vs. CIR	23
4.1	Matching Cardinal and Sensor Coordinate Systems for Tracking Experiments	35
6.1	Environmental parameters for the NLR Model for experiment one	60
6.2	Environmental parameters for the NLR Model for experiment two	60
6.3	Ground truth values experiment 1	62
6.4	Ground truth values experiment 2	64
6.5	Localization Errors for Tracking Experiments	65
6.6	Localization Errors for Fusion Particle Filter Experiments	66

Abstract

Wi-Fi indoor localization has been an attractive research area because of the benefits it can bring to the development of a wide variety of pervasive applications for indoor environments. Modern works have shown that Channel Impulse Response (CIR) is a more reliable source of information than Received Signal Strength Indicator (RSSI) for Wi-Fi localization, allowing to build a fine-grained model to correlate distances with transmitted signal power. This thesis proposes a Wi-Fi indoor localization and tracking system making use of a CIR-based ranging model along with particle filter, which is a Bayesian filter that has been successfully implemented in tracking systems. In addition to the ranging measurements, this thesis' contribution is to fuse one additional source of information aimed to increase the accuracy of the indoor tracking system. This additional measurement is the velocity of the mobile target, which is estimated by applying pedestrian dead reckoning (PDR) methods. PDR makes use of the inertial sensors embedded in off-the-shelf mobile devices to derive displacements of a pedestrian user holding a mobile device. Combining this two different observations in a fusion particle filter algorithm, the performance of the indoor tracking system can be increased. Experimental results with three different traces around a floor covering $297m^2$ show that fusion particle filter can achieve a mean error of $1.4m$.

Chapter 1

Introduction

One of the main requirements for indoor tracking is the localization of objects indoors with high accuracy. Unfortunately, the common Global Position System (GPS) technology used in outdoors tracking systems is not a reliable solution because its signals are not able to penetrate in-building materials. One of the best technologies to use instead of GPS for indoor environments is Wi-Fi, because it offers the advantages of open access and low cost properties. Wi-Fi allows many electronic devices to exchange data or connect to the Internet wirelessly using radio waves. The Wi-Fi Alliance defines Wi-Fi devices as any Wireless Local Area Network (WLAN) products that are based on the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards.

Achieving an accurate WiFi-based indoor localization system, would open the gate to the development of various mobile and pervasive applications, such as advertisement promotion in airports or shopping malls, location detection of assets in a warehouse, patient tracking inside the building of a hospital, navigation during emergency rescue and emergency personnel positioning in a disaster area. The primary impediment in wireless indoor localization lies in the rich multipath fading and temporal dynamics indoors. Although ubiquitous, traditional power features like received signal strength indicator (RSSI) fails to provide sufficient distinction and robustness in complex indoor environments, as RSSI is the superimposition of multipath signals with fast changing phases. Therefore, it is required to find a suitable power feature which provides more information regarding the multipath effects. Recent works have shown that channel impulse response (CIR) fits that requirement and can be used to increase the accuracy in indoor environments. In order to build an accurate indoor tracking system, in addition to CIR-based indoor localization, it is required to combine different approaches used for positioning, navigation and tracking systems like pedestrian dead reckoning (PDR) and particle filter to further improve the accuracy. The motivation for choosing these approaches are given in section 1.1. In section 1.2 the problem for WiFi indoor localization is stated along with the proposed solutions to use in this thesis. Finally, section 1.3 gives an overview of this thesis structure.

1.1 Motivation

Motivated by the works done about CIR-based localization (see chapter 2), this thesis implements an indoor tracking system through a CIR-based ranging model. As in this thesis the target

node is mobile, it is required to include some tracking schemes. Thus, following the works done about tracking systems with pedestrian dead reckoning (see chapter 2), this thesis implements a PDR algorithm using off-the-shelf inertial sensors embedded in mobile devices. The main drawback for PDR is the accumulation of errors, which can be tackled using sequential Monte Carlo (SMC) methods. Particle filter is a SMC approach able to fuse different sources of information to address accumulation issues. In addition, this Bayesian filter can deal with noisy measurements and non linear models, which fits the CIR-based ranging model used in this thesis.

1.1.1 Navigation and Pedestrian Dead Reckoning

Navigation is the process of estimating the parameters necessary for describing the current location with respect to some reference frame. These navigation parameters can be classified into two categories. The first category deals with linear displacements and provides information regarding the evolution of position and velocity. The second category deals with angular displacements and describes ones orientation/attitude with respect to a reference frame.

The process of providing pedestrians with guidance information, which in some way can be used for simplifying the task of reaching a particular destination, is called pedestrian navigation. Hence, pedestrian dead reckoning (PDR) is a process originally taken from navigation, where dead reckoning is used to estimate the position, orientation and velocity of a target without external references. PDR exploits the readings of off-the-shelf Inertial Measurement Units (IMUs) embedded in smartphones and has been used along with GPS for map-matching i.e. outdoor tracking. In positioning systems PDR is used to estimate displacement step by step of a pedestrian user by combining step detection, stride length and heading direction estimation.

1.1.2 Monte Carlo Localization

Many real-world data analysis tasks involve estimating unknown quantities from some given observations. In most of these applications, prior knowledge about the phenomenon being modelled is available. This knowledge allows the formulation of Bayesian models, that is prior distributions for the unknown quantities and likelihood functions relating these quantities to the observations.

Often, the observations arrive sequentially in time and one is interested in performing inference on-line. It is therefore necessary to update the posterior distribution (obtained from Bayes' theorem) as data become available. Examples include tracking an aircraft using radar measurements, estimating a digital communications signal using noisy measurements, etc.

Real data can be very complex, typically involving elements of non-Gaussianity, high dimensionality and non-linearity, which conditions usually preclude analytic solutions. According to the field of interest, the problem appears under many different names, including Bayesian filtering, optimal (non-linear) filtering, stochastic filtering and on-line inference and learning.

Sequential Monte Carlo (SMC) methods are a set of simulation-based methods which provide a convenient and attractive approach to computing the posterior distributions. SMC methods are very flexible, easy to implement, parallelisable and applicable in very general settings. The advent of cheap and formidable computational power, in conjunction with some recent developments in applied statistics, engineering and probability, have stimulated many advancements

in this field. There has been a proliferation of scientific papers on SMC methods and their applications. Several closely related algorithms under the names of *bootstrap filters*, *condensation*, *particle filters*, *Monte Carlo filters*, *interacting particle approximations* and *survival of the fittest*, have appear in several research fields.

1.2 WiFi Indoor Localization Issues

The spatial features of emitted wireless signals are the basis of location distinction and determination for wireless indoor localization. The wireless signal power can be easily measured through RSSI, and thus has been adopted in some indoor localization systems (see chapter 2) based on fingerprinting approaches or path loss power models. However, real indoor environments are complex scenarios, where several static and mobile objects are presented, which multiplies multipath fading effects. These issues impact RSSI-based localization and degrade dramatically performance. As the PHY layer counterpart of RSSI, CIR holds potential to achieve accurate indoor localization, because it is able to discriminate multipath characteristics. In 802.11 a/g/n standards, channel response can be partially extracted from off-the-shelf Orthogonal Frequency Division Multiplexing (OFDM) receivers in the format of Channel State Information (CSI), which reveals a set of channel measurements depicting the amplitudes and phases of every subcarrier in the frequency domain.

Figure 1.1 outlines a block diagram of the main components proposed to use for the indoor tracking system implemented in this thesis. The CSI format will be extracted from off-the-shelf Intel WNIC[32] and then converted into CIR aimed to build a fine-grained ranging model. Within the PDR component one of the contributions of this thesis is implemented, the velocity estimation, which is calculated based on the inertial sensors values. The most important contribution of this thesis is the data fusion with particle filter, where velocity from PDR component is combined with the CIR-based ranging in order to achieve a better tracking accuracy.

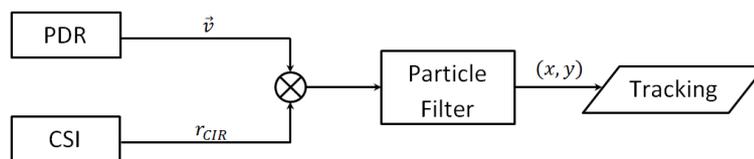


Figure 1.1: Proposed Indoor Tracking System

1.3 Outline

Previous sections introduced the main mechanisms used for the implementation and development of the indoor tracking system. This section outlines the structure of this thesis and the main contents of the coming chapters.

Firstly by introducing the state of the art regarding Wi-Fi indoor localization, the last advances

about a fine-grained power feature is given in chapter 2. Additionally, this chapter summarizes the related work done for tracking systems by applying PDR methods and presents the use of Bayesian filters to increase the tracking accuracy. Following the related work, chapter 3 presents the theoretical foundations and underlying principles upon which the indoor tracking system is built. Starting with IEEE 802.11n PHY layer specifications, then the physics concepts regarding the inertial sensors used in this thesis are given. Finally, Monte Carlo methods introduce the statistics theory background for understanding the particle filter algorithm. Afterwards, chapter 4 introduces the contributions done in this thesis, where the models for estimating the velocity and CIR-based ranging are established. In addition, the details for the combination of the velocity and CIR-based ranging in the particle filter is shown. Chapter 5 outlines the hardware and software components needed for the implementation of the Network-based localization system required for the tracking experiments. Subsequently, chapter 6 reveals the results of the measurements and performance evaluation of the different approaches used in this thesis for all the tracking experiments. Additionally, this chapter presents the environment configuration for performing these tracking experiments. Finally, chapter 7 introduces the analysis done based on the evaluation of the experiments and the relevant findings from the other chapters.

Chapter 2

Related Work

Indoor localization and tracking of wireless devices with high accuracy have been high intensive research works in recent years, because it is the foundation to achieve reliable support location-based services for indoor environments. In contrast to outdoors, where solutions like GPS can be used for localization, indoor environments have low accuracy using such techniques, especially GPS where the signals cannot penetrate in-building materials. Thus, several research work have focused on analysing different solutions like PDR systems along with Wi-Fi components to increase the accuracy of RSSI approaches. Other works use CSI to have a more reliable source of information than RSSI, which leads to build a fine-grained model between distance and wireless signal power, whereas for tracking systems some works focus on fusing PDR models and Wi-Fi information with Bayesian Filters to improve the tracking accuracy of the mobile target.

2.1 WiFi Indoor Localization

Most of research done about WiFi based indoor localization has shown that RSSI is easily affected by multipath effects and Non-Line Of Sight (NLOS) propagation, which leads to significant performance degradation. Consequently, in order to tackle the issue of multipath propagation in indoor environments, some research has focused on analysing OFDM, because this digital multi-carrier modulation method offers benefits for localization and tracking. The channel estimation in OFDM systems can be performed by using pilot symbols. These pilots are inserted into data streams at the transmitter, and removed afterwards at the receiver [12]. CSI is used in OFDM to measure the channel at the subcarrier level, e.g. to support multiple-input multiple-output (MIMO) operation, and has been successfully used [16] to achieve better performance in OFDM systems. Thus, CSI is a fine-grained value from PHY layer, which describes the amplitude and phase of channel contention on each subcarrier in the frequency domain. Here is where CSI has become an interesting subject of research [13],[14],[15], as a reliable metric for indoor localization, because it provides a much richer source of information than RSSI. CIR is a systematic way to categorize channels and has been used in works such as [17],[15],[7] to characterize the individual paths of the communication channel in the time domain as a set of temporal linear filters.

WiFi-based indoor localization can be classified as active localization systems and passive

localization systems:

2.1.1 Active Localization Systems

In active localization systems, targets are required to actively participate in the localization process as in [13] which is a work based on CSI to alleviate multipath effects at the target node, which receives the access point (AP), i.e. anchor nodes, coordinates. In this work the frequency diversity of the subcarriers in OFDM systems is explored, in order to achieve fast and accurate indoor localization of the target node. Signal processing techniques are leveraged in both time and frequency domains to mitigate the multipath effects. Range-based localization is done using a refined indoor propagation model, where the environmental parameters of such model are retrieved by applying a fast training algorithm based on supervised learning, whereas the location determination, i.e. trilateration algorithm, is done by the receiver through aggregation of the CSI values from the physical layer to triangulate the precise position of the target object using a linear least square (LLS) method. Experimental results showed that the accuracy and latency of distance calculation can be significantly enhanced by using CSI, where for instance some positions with serious multipath effects achieve up to 10 times accuracy gain over the corresponding RSSI-based scheme. In average this solution outperforms RSSI around 3 times for the distance determination of a single link, and the median accuracy of this work is 1.2m. It is worth to mention that there was only one testing trace in this work, which was very simple. Thus, it is very likely that the good accuracy has been influenced from this fact.

The main drawback in active localization systems is the intrinsic software and hardware limitation of target nodes. Despite the fact that in some works [13] a good accuracy is achieved, in practice with a high quantity of heterogeneous devices is rather difficult to have all targets with enough computation capabilities to process the algorithms to locate itself. Furthermore, under the assumption that the targets have enough capabilities, there is the issue of the battery, which implies a time constraint for such solution, because the battery can be depleted much faster than in a passive localization solution.

2.1.2 Passive Localization Systems

In passive localization systems, the targets do not need to participate in the localization process as in [17] where Software Defined Radio (SDR) receivers, i.e. anchor nodes, are used to extract CIR at the physical layer. In this work the range-based localization algorithm is done with nonlinear regression (NLR) models, where the measurement parameters are converted into propagation distances based on a relationship with the CSI values as a nonlinear curve fitting problem, which improves the ranging accuracy compared to the commonly used log-distance path loss model (LDPL). The environmental parameters used in the NLR model defined by authors are obtained through algorithms to solve unconstrained optimization problems, in particular the trust region approach is applied in this work. For the location determination a new two-stage trilateration algorithm is proposed by the authors, which is a combination of Weighted Centroid and Constrained Weighted Least Square (WC-CWLS) methods. This combination is

done with the aim of mitigating the influence of ranging (NLOS) errors. Experimental results showed that the algorithm is robust against ranging errors and outperforms the linear least square (LLS) algorithm and Weighted Centroid (WC) algorithm. The mean localization error of the system achieves $2.4m$.

An interesting work [14] proposes a passive crowdsourcing CSI-based indoor localization scheme focusing on estimating the moving distance of the mobile target purely based on 802.11n CSI. Authors implemented an architecture where the APs record the RSS values and the CSI values of the received signal from the target node and subsequently those APs send the data to a localization server. Based on the sequence of the CSI values taken during a time window, the server estimates the moving distance of the client in this time window. The RSS values are used to build a matching with the fingerprint RSS values stored in the fingerprint database, which itself is populated using crowdsourcing techniques. The estimated moving distance, together with the estimated geodesic distance of different fingerprint locations in the map are used to further improve the quality of the matching, and thus, the accuracy of the localization. Authors estimated the moving speed by CSI through wireless radio propagation models, where the speed estimation is transform to a frequency estimation problem following the basic physics laws of wave propagation, whereas the wireless signal propagation in indoor environment was modelled as Rician fading channel. Evaluation results indicate that this solution provides accurate localization with error $2m$ at 80% at very complex indoor environment with minimal overhead.

An advantage of passive localization systems over active localization systems, is that whenever a software modification is needed, active localization systems would require an update in every single target node, whereas for passive localization systems the update could be done just in the anchor nodes or in a central server where the localization algorithms are running. Therefore, in passive location systems the deployment and support are easier than in active localization systems.

2.2 Pedestrian Dead Reckoning

There are several works that have adopted Pedestrian Dead Reckoning (PDR) as an important component for tracking, where the displacement of the pedestrian user is estimated through inertial sensors such as accelerometers, magnetometers and gyroscopes. Based on the techniques used in those works, it would be possible to classify PDR solutions as Sensor Fusion and Full Sensor Fusion. The former refers to works where the orientation of the smartphone is assumed to be known and the latter refers to works where the orientation of the smartphone is estimated.

2.2.1 Sensor Fusion

These works made use of the readings from accelerometers and gyroscopes to estimate the displacement of a pedestrian user under the assumption that the way a pedestrian user is holding the smartphone is constant. In [10], [11] the heading direction estimation of the user is based on the measures from the gyroscope, whereas the displacement of the user is estimated only

with accelerometer values. Authors in [11] proposed an algorithm known as Heuristic Drift Elimination (HDE), which is dynamic feedback drift elimination technique used to tackle the problem of error accumulation, however is not a practical solution because it requires specialized sensors in the foot of the pedestrian. Thus, authors in [10] extended HDE to include a turn insufficient effect component, which is built under the premise that the swing of mobile phones will cause readings of gyroscopes lower than real angular velocity of person movement during turning or moving on curved paths.

The main drawback for using these approaches is that although gyroscopes are relatively immune to environmental disturbance, the accumulation of errors can increase without a bound even after some drift reduction mechanisms, specially when the walking takes a considerable time.

2.2.2 Full Sensor Fusion

These works made use of the readings from accelerometers, magnetometers and gyroscopes to estimate the displacement of a pedestrian user, and in addition these works also derive the orientation of the smartphone while it is carried during walking. In [6] accelerometer and magnetometer are used to estimate the device orientation. Then, walking and running model is built based on accelerometer measures, whereas principal component analysis (PCA) of the horizontal acceleration is used to estimate the moving direction. Other works [7] make use of the gyroscope to distinguish between random device's orientation changes and physical turns of the pedestrian user. In this work the distance estimation is made with the measures of the accelerometer and the moving direction is estimated with a combination of the measures from the gyroscope and magnetometer. Some works [8] merged the methods detailed above to estimate the device orientation, and use the accelerometer not only for step detection but also to determine heading direction of the user by combining accelerometer readings with the readings from the magnetometer and gyroscope.

Unfortunately, inertial sensors from smartphones are very noisy because of the low cost micro electromechanical systems (MEMS) embedded in those devices. Thus, PDR based on smartphones have some drawbacks such as accumulation of errors and noise. For instance, when using the gyroscope an integration over time of the angular velocity is needed in order to estimate the angle, which produces high cumulative errors.

For estimating the angle with magnetometer, one has to take into account that this sensor is very sensitive to interference from other magnetic fields, making it really difficult to find the earth's magnetic field, especially for indoor environments, where it is possible to find a lot of different materials and electrical devices that can generate additional magnetic fields, which disturb the magnetometer readings.

There are several mechanisms used to reduce the noise and cumulative errors from the raw data taken from smartphone's inertial sensors such that we can get more reliable data. For instance the data from the accelerometer can be processed with a low pass filter, which is enough to implement an algorithm for step detection and distance estimation.

In order to reduce the noise in the magnetometer there are some mechanisms [28] where the

accelerometer is used to compensate the magnetometer readings whereas hard-iron effects are identified to subtract magnetic fields within the vicinity of the magnetometer leading to more accurate data. For solutions where the gyroscope is used, the magnetometer can be used to mitigate gyroscope errors [22].

2.3 WiFi Indoor Tracking

Most of the positioning algorithms for indoor localization do not consider mobility (see section 2.1), which normally works well for static targets, but for mobile targets, the localization is required to be done continuously, where filtering techniques can help to smooth the trajectory and reduce the estimation error. Some works have adopted PDR tracking schemes (see section 2.2). However, in order to tackle the intrinsic dead reckoning errors that still persist even after applying some noise and drift reduction techniques, some suitable mechanisms have to be taken into account to further improve the accuracy. Several works [6], [7], [8] make use of Bayesian filters (e.g. Kalman Filter, Particle Filter), because such filters allow easily modelling a dynamic system from sensor measurements. Several works have shown that by combining PDR with Bayesian filters the accuracy of the model is considerably increased.

2.3.1 Kalman Filter

Kalman filter recursively estimates the state of a process using a model of time propagation and observations containing some inaccuracy, alternating between the prediction phase and the update phase. A Kalman filter is commonly used in outdoors tracking applications, implementing PDR/GPS integrated methods based on Kalman filter [6], where the system conducts position correction accompanying the GPS measurements based on a Kalman filter framework and a map node matching. Kalman filter has also been used to minimize the effect of noise in some indoor tracking solutions [7], where the distance estimation error has been reduced and PDR predictions are adjusted after detecting physical turns using the smartphone's gyroscope. In [7] Kalman filter uses the past distance and relative speed estimates to correct the errors in the current estimates. This filtering generally performs well, except in cases when the user is close to AP and takes a turn. Authors in [7] make corrections to the Kalman filter's through estimation of the the new relative speed by considering the user's relative speed right before the turn and the orientation change of the user from her gyroscope. Thus, whenever the user takes a physical turn, Kalman filter's relative speed estimate is corrected to account for the abrupt change. This solution achieves a median error of $2.3m$.

Kalman filter performs comparable to particle filter when only RSS measurements are considered. However, when integrating additional information to the particle filter thanks to its availability to fuse data, the particle filter is in average 30% better than Kalman filter [18]. Furthermore, the main drawback of Kalman Filter is that its main assumption, i.e. the linear model, is hardly fulfilled in real life. The Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF) are proposed to solve the non-linear estimation problem by linearizing all the non-linear

models. However, they are only reliable for systems that are almost linear. Distributed information like the map information is impossible to be integrated for tracking by EKF or UKF. Thus, as an alternative to Kalman filter and its derivatives, particle filters are attracting more and more attention recently in research works about indoor tracking.

2.3.2 Particle Filter

Particle filters are able to estimate the parameters of a system (e.g. user location) using Bayesian principles. Unlike a Kalman filter, particle filters are easily adapted to handle the presence of obstacles like walls, non-linearities and non-Gaussian noise models. This is particularly important when such a wide variety of sensing types are being combined. A particle filter will usually be more computationally expensive than a Kalman filter, but they can still work with reasonable speed on a smartphone processor. In contrast to Kalman filter, the performance of a particle filter can be scaled with available computation power by varying the number of particles that are tracked.

One can see in works such as [19], where a tracking system is implemented by exploiting particle filters to combine dead reckoning, RSS-based readings and knowledge of floor plans together. Authors in [19] implemented a PDR component in a smartphone that outputs a human motion vector model, which is then used as input for the particle filter component, the WiFi component records RSS values periodically from all available APs in the floor as a RSS vector. In addition authors in [19] exploit RSS readings through comparative analysis of the relationship between the RSS values during the motion of a pedestrian, concluding in three observations:

- **Turn Verification:** It is used to handle unconscious human behaviours that cause great change on the readings of direction sensors, for instance hand trembling while a pedestrian user is walking. RSS vectors between continuous steps are examined to distinguish pedestrian turning from hand trembling.
- **Room Distinguishing:** It is used to distinguish which room the pedestrians enter when two doors are close to each other. When a pedestrian user is entering a room, there is a clear tendency in the change of RSS vectors.
- **Entrance Discovery:** It is used to try to discover a possible path when the estimated position remains almost the same while particles keep dying for a number of steps.

The aim of adopting particle filters [19] is to represent and control uncertainty of PDR, leveraging the constraints imposed by floor plan and the indication of the Wi-Fi component. The particle filter component redistributes every particle according to the motion vector in propagation phase. Then, the correcting phase first corrects the weight of each particle according to the floor plan and calculates the weighted center of the particles. Upon the geometric relationship between the new center and last tracking position, the particle filter component invokes Turn Verification, Room Distinguishing and Entrance Discovery of Wi-Fi component to further correct the particle weights. The resampling phase follows and outputs the center of weighted particles as the current estimated location of the pedestrian. Authors in [19] used an active localization system based on three major components: PDR, WiFi and Particle Filter,

where the particle filter component is installed in off-the-shelf smartphones. Thus, the number of particles is limited, although the system achieves a good localization error of 0.71 m in a college building covering $1362m^2$.

Another work [8] proposes an application for robust WiFi indoor tracking by combining complementary localization approaches for dead reckoning and WiFi signal strength strategies. The complementary and redundant characteristics of the two approaches allow the system to operate robustly even in environments where one or more individual sensors maybe disrupted. The system makes use of RSSI for position estimation. In addition, a fingerprinting technique is used to establish a relation between RSSI and position. Furthermore, once a calibration database of the environment has been generated, it can then be used across different smartphones without the need for re-calibration. Authors make use of WiFi radio combined with PDR for use in the system. This approach combines multiple complimentary localization systems including dead reckoning, WiFi, and GSM using a particle filter for robust localization over multiple floors of an indoor building. A walking motion model combined with a heading estimator provides a pre-filtered dead reckoning sensor estimate to the particle filter. The combined sensor data is fused and filtered using a particle filter which results in a smooth and continuous position estimation state. At runtime WiFi signal strength fingerprinting is used to initialize the system and it provides a rough global location estimate. The user's movement is also tracked at high frequency using dead reckoning. Authors in [8] introduce a particle filter to combine these different sources of information. To minimize the computational requirements of the solution, authors focus on keeping the dimensionality of the particle filter as low as possible. For this reason, authors do not track heading within the filter, but estimate only the linear position. The dead reckoning is performed in a pre-processing step, and all the particles in the filter are periodically updated based on a model of the variance of the dead reckoning estimate. Experimental results are analysed using an online system and an offline system. In the former the algorithms ran in the smartphone, whereas in the latter the smartphone just collected the data, which is afterwards processed by the algorithms in a laptop. The average mean error of the system is $3m$.

The fusion availability of particle filter is one of the main advantages of this Bayesian filter, where RSS measurements, acceleration and map information can be used all together to achieve a better accuracy than Kalman filters[18]. This work uses the acceleration from MEMS to derive a walking distance which is used to built a fusion model of RSS measurements and accelerometer with particle filter. Furthermore, authors made use of the map information to update the weights of the particles, which reduces the uncertainty of walking trajectory.

These works have shown that particle filter is very useful for indoor tracking because it offers two main advantages. First, it is able to handle non-linear models such as range-based localization for CSI [17]. Second, particle filter allows data fusion, which means that different source of information like RSSI, CSI, PDR, environment map information [17],[8] can be used to estimate the position of a target with high accuracy.

Chapter 3

Theoretical Background and Underlying Principles

There are different approaches used for indoor tracking (see chapter 2) and depending on the specific approach, some technical specifications and theoretical foundations are required to fully understand the way to implement the correspondent mechanism and procedures for localization and tracking of wireless devices. The contents of this chapter was written on selecting literature describing the IEEE 802.11n standard [1],[2], OFDM modulation method[3], CIR power features[4], inertial sensors for navigation [21], [30], [22], [23] and Monte Carlo methods with particle filter [25]. The contents of the upcoming sections strongly relies on the mentioned sources.

This chapter is organized as follows: An overview of the physical layer specifications for IEEE 802.11 is first given in section 3.1. This section also establishes the theoretical basics for using the power features from CIR. In section 2.2 the main concepts regarding the inertial sensors used in the PDR algorithm implemented (see chapter 4) in this thesis is introduced. Finally, in section 3.3 the theoretical background about sequential Monte Carlo methods with an emphasis in a generic particle filter algorithm is presented.

3.1 Physical Layer in IEEE 802.11n

IEEE 802.11 is a group of medium access control (MAC) and physical layer (PHY) specifications for the implementation of reliable Wireless Local Area Network (WLAN) communication. The 802.11 family is a set of over-the-air modulation techniques that share the same basic protocol (see table 3.1). These standards provide the foundations for several wireless network products using the Wi-Fi brand and also defines the segment of the radio frequency spectrum used by 802.11 per country.

The 802.11n amendment [2] includes many improvements for WLAN range, reliability, and throughput. Beginning at the PHY layer, where advanced signal processing and modulation techniques have been added to exploit multiple antennas and wider channels. In the subsequent layer, MAC protocol extensions make more efficient use of available bandwidth. Combining these High Throughput (HT) enhancements, it is possible to boost data rates up to 600 Mbps more than a ten-fold improvement over 54 Mbps 802.11a/g.

Table 3.1: IEEE 802.11n PHY Standard

Release Date	2009
Standard	802.11n
Frequency Band	2.4 GHz, 5 GHz
Bandwidth	20 MHz, 40 MHz
Modulation	OFDM
Advanced Antenna Technologies	MIMO, up to 4 spatial streams
Maximum Data Rate	600 Mbits/s

802.11n operates on both the 2.4 GHz and the 5 GHz(optional) bands. IEEE 802.11n was built on previous 802.11 standards by adding MIMO and 40 MHz channels to the PHY layer, and frame aggregation to the MAC layer.

In addition to 802.11n enhancements, there is also the capacity to receive and/or transmit simultaneously through multiple antennas. 802.11n defines many $M \times N$ antenna configurations, ranging from 1×1 to 4×4 . MIMO uses multiple antennas to consistently resolve more information than possible using a single antenna. One method to provide this is through Spatial Division Multiplexing, which spatially multiplexes multiple independent data streams, transferred simultaneously within one spectral channel of bandwidth. MIMO is able to increase considerably data throughput as the number of resolved spatial data streams is increased. Each spatial stream requires a discrete antenna at both the transmitter and the receiver.

The minimum number of antennas in use on both sides of the link defines the number of simultaneous data streams. However, the individual radios often reduce the number of spatial streams that may carry unique data. Equation 3.1 helps to identify the capability of a given radio

$$M \times N = Z \quad (3.1)$$

where M and N , are the maximum number of transmit and receive antennas that can be used by the radio respectively. Whereas Z , is the maximum number of data spatial streams the radio can use. For example, a radio that can transmit on two antennas and receive on three, but can only send or receive two data streams would be $2 \times 3 : 2$.

In addition, 802.11n incorporates an optional feature, which is the 40 MHz channels. Previous 802.11 products use channels that are in general 20 MHz wide. In contrast 802.11n products have the option to use 20 or 40 MHz wide channels, assuming the AP has 40 MHz capability as well. By operating with a bandwidth of 40 MHz, it is possible to provide channels twice the PHY data rate available over a single 20 MHz channel. The wider bandwidth can be enabled in either the 2.4 GHz or the 5 GHz mode, but must not interfere with any other 802.11 or non-802.11 (such as Bluetooth) system using the same frequencies.

3.1.1 Physical Layer (PHY) Frame Structure

The 802.11 Physical Layer uses burst transmissions or packets. Each packet contains a *preamble*, *header* and *payload data* as illustrated in figure 3.1. The receiver uses the *preamble* to estimate channel characteristics for equalization, it is also used to obtain time and frequency



Figure 3.1: PHY Frame Structure

synchronization. The *preamble* is simply a bit sequence that receivers watch for to lock onto the rest of the transmission. The next bits are used for packet configuration like format and data rates, such information is provided by the *header*. Finally, the last bits contains the *payload data*, which is the user's data being transported.

The 802.11 standards define three *frame* types: Management Frames, Control Frames and Data Frames. Each frame consists of a MAC header, payload and frame check sequence (FCS). Some frames may not have the payload. The first two bytes of the MAC header form a frame control field specifying the form and function of the frame.

3.1.2 802.11n Packet Format

802.11n standards defines two operating modes: Greenfield (HT) and Mixed (Non-HT) as detailed in table 3.2. Greenfield is able to work only with HT systems, i.e. where no legacy systems exist. HT systems are not able to switch between Greenfield and Mixed, they will only use one or the other.

An 802.11n AP using Non-HT mode sends all frames in the old 802.11a/g format so that legacy stations can understand them. In such scenario, an AP must use 20 MHz channels and none of the new HT features described in this section. All products must support this mode to ensure backward compatibility, but an 802.11n AP using Non-HT delivers no better performance than 802.11a/g.

The mandatory HT Mixed mode will be the most common 802.11n AP operating mode. In this mode, HT enhancements can be used simultaneously with HT Protection mechanisms that permit communication with legacy stations. HT Mixed mode provides backwards compatibility, but 802.11n devices pay significant throughput penalties as compared to Greenfield mode.

3.1.3 Orthogonal Frequency Division Multiplexing

Orthogonal Frequency Division Multiplexing (OFDM) is widely applied in wireless communications systems due to its high rate transmission capability with high bandwidth efficiency and its robustness with regard to multipath fading and delay. It has been used in digital audio broadcasting (DAB) systems, digital video broadcasting (DVB) systems, digital subscriber line (DSL) standards, and wireless LAN standards such as the American IEEE standard 802.11 (WiFi) and its European equivalent HIPRLAN/2. It has also been proposed for wireless broadband access standards such as IEEE standard 802.16 (WiMAX) and as the core technique for the fourth-generation (4G) wireless mobile communications.

The basic idea underlying OFDM systems is the division of the available frequency spectrum into several subcarriers. To obtain a high spectral efficiency, the frequency responses of the

Table 3.2: 802.11n Packet Format (refer to figure 3.2 and 3.3)[1]

<p>Preamble Mixed Mode</p>	<p>HT Legacy L-STF, L-LTF, L-SIG are backward compatible to a/g systems. L-SIG contains RATE and LENGTH values that inform Legacy systems how long to hold off next Tx attempt</p> <p>HT Mixed Mode HT-SIG (2 symbols): Indicates MCS (Modulation and Coding Scheme), Length, and other HT-specific parameters HT-STF (1 symbol), HT-LTF (≥ 1 symbol): Allow sync and channel estimation on HT bandwidth (more subcarriers than L-LTF). Additional HT-LTF symbols are included for MIMO modes to <i>sound</i> the multiple channels (paths)</p>
<p>Preamble Greenfield Mode</p>	<p>L-STF, L-LTF, L-SIG are dropped, HT-STF and HT-LTF replace L-STF/LTF Otherwise similar to MF Preamble</p> <p>Fields:</p> <ul style="list-style-type: none"> ● RATE (4 bits): Indicates Data FEC coding and modulation (8 combinations), aka MCS ● LENGTH (12 bits): Number of octets (bytes) carried in Payload ● PARITY (1 bit): Even parity-check on RATE+LENGTH data ● TAIL (7 bits): Used for SIGNAL symbol FEC decoding
<p>Payload</p>	<p>56 (20 MHz) or 114 (40 MHz) subcarriers Data subcarriers use BPSK, QPSK, 16QAM, or 64QAM modulation. Same in all symbols Pilots subcarriers (BPSK only) are used to track frequency/phase and amplitude variations over the burst Optional Short Guard Interval can be used if multipath environment allows</p>

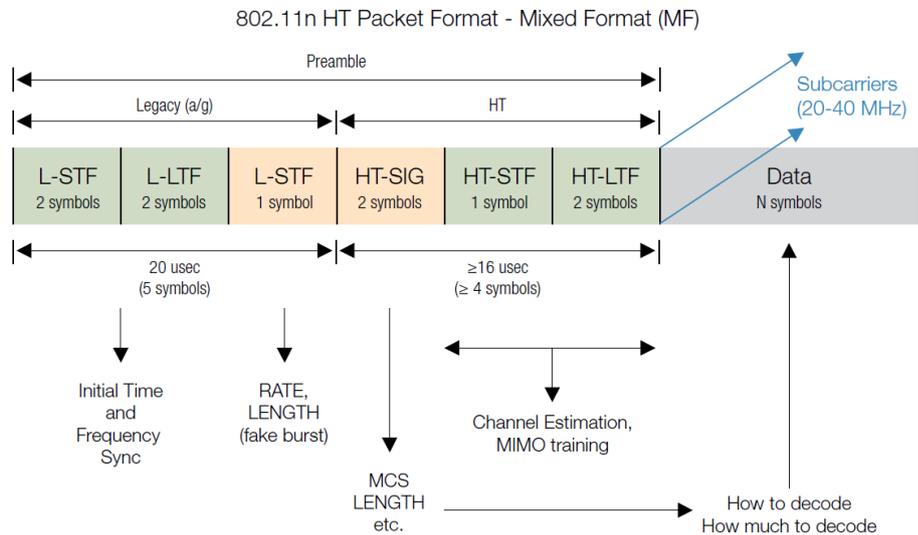


Figure 3.2: 802.11 Packet Format - Mixed Format (MF)

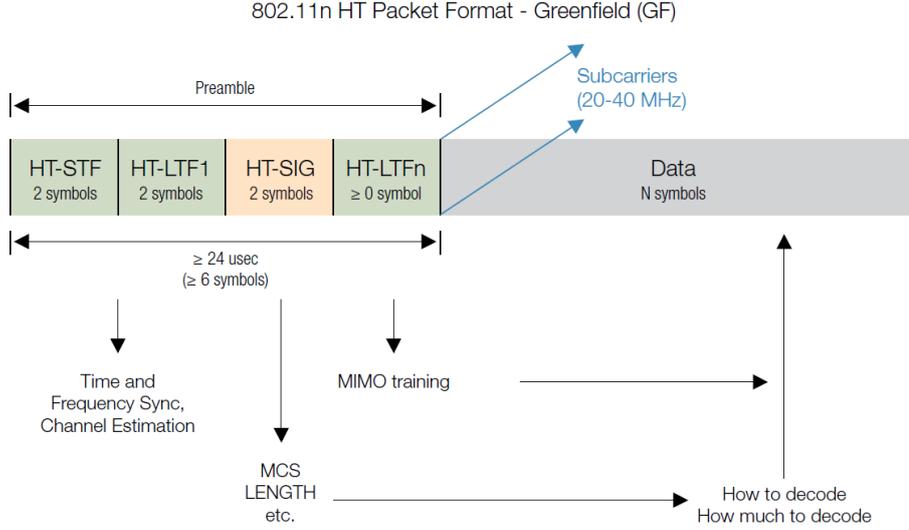


Figure 3.3: 802.11 Packet Format - Greenfield (GF)

subcarriers are overlapping and orthogonal, hence the name OFDM. This orthogonality can be completely maintained with a small price in a loss in SNR, even though the signal passes through a time dispersive fading channel, by introducing a cyclic prefix (CP). A block diagram of a baseband OFDM system is shown in Figure 3.4. The binary information is first grouped, coded, and mapped according to the modulation in a *signal mapper*. After the guard band is inserted, an N-point inverse discrete-time Fourier transform ($IDFT_N$) block transforms the data sequence into time domain (note that N is typically 256 or larger). Following the IDFT block, a cyclic extension of time length T_G , chosen to be larger than the expected delay spread, is inserted to avoid intersymbol and intercarrier interferences. The D/A converter contains low-pass filters with bandwidth $1/T_S$, where T_S is the sampling interval. The channel is modelled as an impulse response $g(t)$ followed by the complex additive white Gaussian noise (AWGN) $n(t)$, where α_m is a complex values and $0 \leq \tau_m T_S \leq T_G$.

$$g(t) = \sum_{m=1}^M \alpha_m \delta(t - \tau_m T_S) \quad (3.2)$$

At the receiver, after passing through the analog-to-digital converter (ADC) and removing the CP, the DFT_N is used to transform the data back to frequency domain. Lastly, the binary information data is obtained back after the demodulation and channel decoding.

Let $\bar{X} = [X_k]^T$ and $\bar{Y} = [Y_k]^T$ ($k = 0, \dots, N - 1$) denote the input of IDFT block at the transmitter and the output data of DFT block at the receiver, respectively. Let $\bar{g} = [g_n]^T$ and $\bar{n} = [n_n]^T$ ($n = 0, \dots, N - 1$) denote the sampled channel impulse response (CIR) and AWGN,

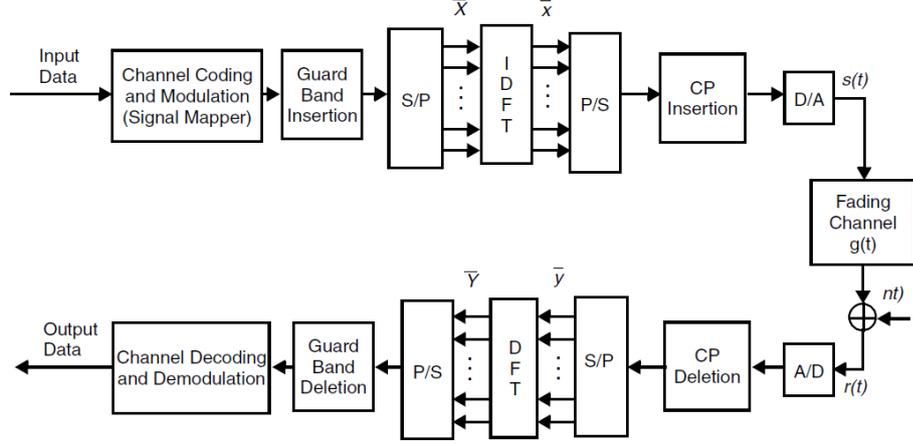


Figure 3.4: Digital Implementation of a Baseband OFDM System [3]

respectively. Define the input matrix $\underline{X} = \text{diag}(\bar{X})$ and DFT matrix,

$$\underline{F} = \begin{pmatrix} W_N^{00} & \dots & W_N^{0(N-1)} \\ \vdots & \ddots & \vdots \\ W_N^{(N-1)0} & \dots & W_N^{(N-1)(N-1)} \end{pmatrix} \quad (3.3)$$

where $W_N^{i,k} = (1/\sqrt{N})^{-j2\Pi(ik/N)}$. Also define $\bar{H} = DFT_N(\bar{g}) = \underline{F}\bar{g}$, and $\bar{N} = \underline{F}\bar{n}$.

Under the assumption that the interferences are completely eliminated [3], equation 3.4 can be derived:

$$\bar{Y} = DFT_N(IDFT_N(\bar{X}) \otimes \bar{g} + \bar{n}) = \underline{X}\bar{F}\bar{g} + \bar{N} = \underline{X}\bar{H} + \bar{N} \quad (3.4)$$

where $\bar{H} = DFT_N(\bar{g})$ is the channel state information (CSI) and $\bar{N} = \underline{F}\bar{n}$ is the noise in the frequency domain. This equation demonstrates that OFDM is equivalent to transmission of data over a set of parallel channels.

As a result, the fading channel of the OFDM system can be viewed as a 2D lattice in a time-frequency plane, which is sampled at pilot positions and the channel characteristics between pilots are estimated by interpolation. The art in designing channel estimators is to solve this problem with a good trade-off between complexity and performance.

Channel estimation is indispensable in an OFDM system in order to achieve high data rates as well as reliable communication. Two basic 1D channel estimation mechanisms in OFDM systems that are commonly used are block-type pilot and comb-type pilot.

3.1.4 From RSSI to Channel Impulse Response

RSSI is a MAC layer component that measures the transmitted signal power by characterizing the attenuation of radio signals during propagation, making it one of the most prevalent power features, which is accessible in wireless techniques ranging from UWB, ZigBee, and WiFi to cellular networks. However, RSSI in complex indoor environments suffers from temporal fluctuations, becoming it a fickle and coarse-grained feature. In addition, common indoor environments increase multipath effects, which complicates the wireless propagations and deflects RSSI-based ranging. Therefore, a better way to characterize and model small scale multipath effects is required to achieve a more accurate power-based ranging.

Originated from wireless channel sounding, several recent works have analysed the PHY layer and leveraged the finer-grained power feature, channel response, to discriminate multipath characteristics. This section aims to provide the theoretical background basics for this new feature.

RSSI Variation Due to Multipath Shadowing

A transmitted signal in a common indoor environment is propagated to the receiver through multiple paths. Where, every single path contributes to a differently delayed, attenuated, and phase shifted signal. Consequently, the combination of numerous alias versions of the original signal compose the received signal. The complex baseband signal voltage measured at the receiver at a specific time, therefore, is denoted as:

$$V = \sum_{i=1}^N \|V_i\| e^{-j\theta_i} \quad (3.5)$$

where V_i and θ_i are the amplitude and phase of the i th multipath component (note that the signal modulation schemes are implicitly considered), and N is the total number of components. RSSI is then simply the received power in decibels (dB):

$$RSSI = 10 \log_2 (\|V_i\|^2) \quad (3.6)$$

As a superposition of multipath components, RSSI not only varies over distance on the order of the signal wavelength but also fluctuates over time even at a static link. A slight change in certain multipath components may add up to significant constructive or destructive relative phases of the delayed signals, which, as a consequence, lead to considerable fluctuations in RSSI. In fact, the variations of RSSI even at an immobile receiver in 1 minute can be as large as 5 dB in a typical laboratory environment. Other empirical studies have also shown that RSSI readings vary at both small (seconds) and large (hours) granularities and can be as high as 7 dB in a typical student cubicle[4].

Characterizing Multipath Propagation

RSSI is not able to acquire the multiple effects, which is its main drawback. In order to completely characterize the particular paths, the wireless propagation channel is modelled as a tem-

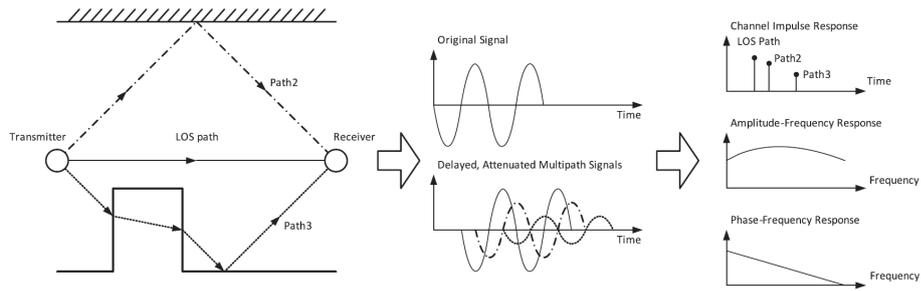


Figure 3.5: Multipath propagations, received signals, and channel responses[4]

poral linear filter, known as Channel Impulse Response (CIR). Under the time-invariant assumption, CIR $h(\tau)$ is denoted as:

$$h(\tau) = \sum_{i=1}^N a_i e^{-j\theta_i} \delta(\tau - \tau_i) \quad (3.7)$$

where a_i, θ_i , and τ_i are the amplitude, phase, and time delay of the i th path, respectively. N is the total number of multipath and $\delta(\tau)$ is the Dirac delta function. Each impulse represents a delayed multipath component, multiplied by the corresponding amplitude and phase.

In the frequency domain, the constructive and destructive phases also cause frequency selective fading, which is characterized as the Channel Frequency Response (CFR). CFR consists of amplitude-frequency response and phase-frequency response. Figure 3.5 shows a multipath propagating condition, the transmitted signals and the received signals, and illustrative channel responses. Given infinite bandwidth, CIR is equivalent to CFR. And CFR is the Fourier transform of CIR.[4]

The small scale multipath effect can be described by CIR and CFR, both are widely used for channel measurement. It is important to note that CIR and CFR are power measurements with respect to complex amplitude, while another pair of parameters in terms of signal power is Power Delay Profile (PDP) and Power Spectrum Density (PSD).

Summarizing, channel response is to RSSI what a rainbow is to a sunbeam, where components of different wavelengths are separated. Channel response possesses finergrained frequency resolution and equivalently higher time resolution to distinguish multipath components, yet at the cost of slight modification of firmware or hardware on off-the-shelf platforms, just as the prism used to disperse the sunlight [4]. A concise comparison of channel response and RSSI is given in table 3.3. The underlying time resolution of CIR is much higher, hence CIR-based localization differs from RSSI. On the other hand, a modified wireless adapter can be used to obtain raw received signal power at the PHY sampling rate.

Table 3.3: RSSI vs. CIR

Category	Layering	Resolution	Stability	Accessibility
RSSI	MAC	Time: Packet level Frequency: N/A	Low	Handy access
CIR	PHY	Time: Multipath cluster Frequency: Subcarrier level	High for CFR as a whole structure	WiFi NIC

Deriving CIR

In the time domain, the received signal $r(t)$ is the temporal convolution of transmitted signal $s(t)$ and channel impulse response $h(t)$:

$$r(t) = s(t) \otimes h(t) \quad (3.8)$$

Accordingly, the received signal spectrum $R(f)$ is simply the multiplication of the transmitted signal spectrum $S(f)$ and the channel frequency response $H(f)$ in the frequency domain:

$$R(f) = S(f) \times H(f). \quad (3.9)$$

As demonstrated in equations 3.8 and 3.9, CIR can be derived from the deconvolution of received and transmitted signals, while CFR is the ratio of the received and the transmitted spectrums. Since the calculation of convolution is non trivial, a practical method to derive CIR is to convert temporal convolution into multiplication in the frequency domain, followed by an inverse Fourier transform. In case of a flat transmission power spectrum, CIR is approximated by:

$$h(t) = \frac{1}{P_s} \mathfrak{F}^{-1} \{ S^*(f) R(f) \} \quad (3.10)$$

where \mathfrak{F}^{-1} denotes the inverse Fourier transform. $R(f)$ is the Fourier transform of the received signal $r(t)$, that is, its spectrum. $S^*(f)$ is the complex conjugate of the Fourier transform of the transmitted signal $s(t)$. P_s approximates the transmitted signal power, which, under the flat transmission assumption, is nearly a constant within the band of interest.

Precisely measuring and modeling the wireless channel often involves dedicated infrastructures such as Vector Network Analyzer (VNA) or Software Defined Radio (SDR). On the other hand, although the derivation of CIR/CFR is modulation independent, it might be more convenient to implement the process on commercial devices with particular modulation schemes. For instance, if OFDM is adopted, such as in IEEE 802.11a/g/n, the receivers are then readily capable of calculating CFR/CIR, since the amplitudes and phases on each subcarrier provide a sampled version of the signal spectrum, while FFT/IFFT operations are integrated in OFDM receivers.

Recent advances in the wireless community have taken this one step further. Leveraging the off-the-shelf Intel 5300 WNIC and a modified driver, a group of sampled versions of CFRs within

the WiFi bandwidth are revealed to upper layers in the format of Channel State Information (CSI) [5]. Each CSI depicts the amplitude and phase of a subcarrier:

$$H(f_k) = \|H(f_k)\| e^{j \sin \angle(H)} \quad (3.11)$$

$H(f_k)$ is the CSI at the subcarrier with central frequency of f_k , and $\angle H$ denotes its phase. Hence a group of CSIs $H(f_k)$, ($k = 1, \dots, K$), reveals K sampled CFRs at the granularity of subcarrier level.

In fact, this sample version of CFR has been employed in recent adaptive wireless communication systems to improve reliability and throughput, as well as for precise indoor localization on off-the-shelf platforms. Therefore, this thesis makes use of this Intel 5300 WNIC (see chapter 5) adapter to obtain this sample version of CFR, e.g. CSI, and transform it into CIR aimed on building a fine-grained ranging-based model (see chapter 4) for indoor localization.

3.2 Inertial Sensors

As seen in chapter 2, there are several works that have adopted pedestrian dead reckoning (PDR) as an important component for tracking, where the displacement of the pedestrian user is estimated through inertial sensors such as accelerometers, magnetometers and gyroscopes.

Inertial sensors are sensors based on inertia. These range from micro electromechanical systems (MEMS) inertial sensors, measuring only a few square mm , up to ring laser gyroscopes which are extremely accurate but expensive and can measure $50cm$ in diameter. Emerging MEMS technology makes low-cost and small size inertial sensors, which have successfully been integrated into mobile devices. In this thesis the accelerometer and magnetometer sensors will be used, to calculate the velocity of a pedestrian user holding a smartphone.

In the context of pedestrian navigation where cost, size and power consumption dictate sensor selection, these sensors are of lowest accuracy and reliability. The mechanisms used to produce reliable solutions are described in chapter 2 and the ones used in particular in this thesis are described in chapter 4

In the literature inertial sensors are also known as inertial measurement units (IMUs), which strictly speaking refers to the raw data values of the inertial sensors. In this thesis the names IMUs and MEMS will be used interchangeably.

3.2.1 Accelerometer

MEMS-based accelerometers are devices that measure the proper acceleration. In relativity theory, proper acceleration is the physical acceleration experienced by an object. The acceleration is measurable by sensors. These sensors are part of the sensing cluster of ubiquitous technologies. Sensing technologies make use of physical parameters from the environment, such as temperature, pressure, force and light. An accelerometer measures weight per unit of mass, a quantity also known as specific force, or g-force. Measuring g-forces allows users to for instance interact with products by means of gesture recognition.

Accelerometers are one of the simplest but also most applicable micro electromechanical

devices. They are widely used in cost sensitive, low power, motion and tilt-sensing applications like mobile devices, gaming systems, disk drive protection, image stabilization, sports and health devices. The best known applications are the Wii remote of Nintendo, Android's smartphones and Apple's iPhones. The most common technologies that enable the use of accelerometers in those devices, rely on capacitors and gas chambers.

Accelerometers in Android platform measures the acceleration applied to the device, including the force of gravity. Conceptually, an acceleration sensor determines the acceleration that is applied to a device (A_d) by measuring the forces that are applied to the sensor itself (F_s) using the following relationship:

$$A_d = - \sum \frac{F_s}{mass} \quad (3.12)$$

However, the force of gravity is always influencing the measured acceleration according to the following relationship:

$$A_d = -g - \sum \frac{F}{mass} \quad (3.13)$$

For this reason, when the device is sitting on a table (and not accelerating), the accelerometer reads a magnitude of $g = 9.81m/s^2$. Similarly, when the device is in free fall and therefore rapidly accelerating towards the ground at $9.81m/s^2$, its accelerometer reads a magnitude of $g = 0m/s^2$. Therefore, to measure the real acceleration of the device, the contribution of the force of gravity must be removed from the accelerometer data. This can be achieved by applying a high-pass filter. Conversely, a low-pass filter can be used to isolate the force of gravity

3.2.2 Magnetometer

A MEMS-based magnetic field sensor is a small-scale micro electromechanical device for detecting and measuring magnetic fields. With the advancements in sensor technology, the Earth's magnetic field can now be measured with the help of a sensor commonly known as the magnetometer. Many of these operate by detecting effects of the Lorentz force: a change in voltage or resonant frequency may be measured electronically, or a mechanical displacement may be measured optically. Compensation for temperature effects is necessary. Such instruments have medical and biomedical applications.

It has been known for centuries that there exists a magnetic field everywhere on the surface of the earth. The earth's magnetic field is a naturally occurring planetary phenomenon, which can be modelled as a dipole and follows the basic laws of magnetic fields summarized and corrected by Maxwell [22]. In order to derive a relationship between the magnetic field and a given observation point p with respect to a frame similar to the one used for navigation systems, e.g. PDR, the governing equations for magnetic field intensity need to be transformed into the

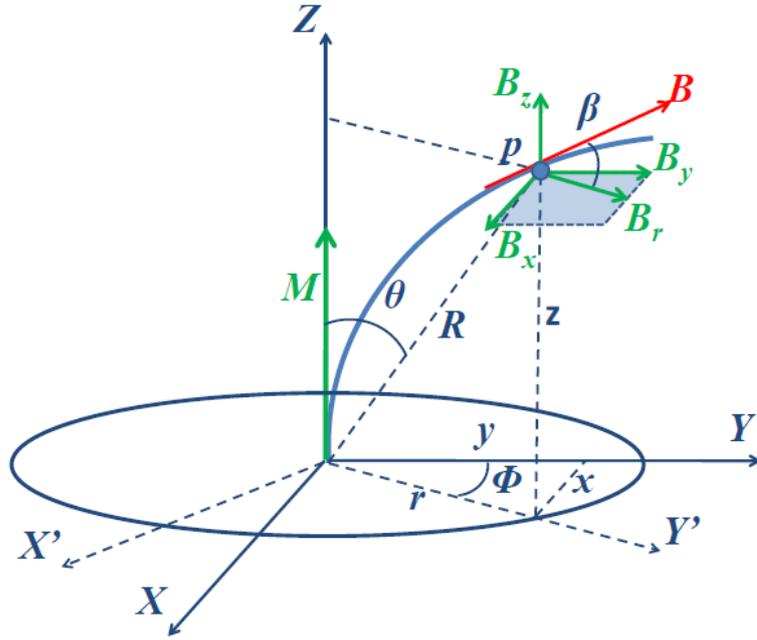


Figure 3.6: Magnetic field components in Cartesian coordinate system[23]

Cartesian coordinate system as:

$$\begin{pmatrix} B_x \\ B_y \\ B_z \end{pmatrix} = \begin{pmatrix} B_r \sin \phi \\ B_r \cos \phi \\ B_{z_cyl} \end{pmatrix} \quad (3.14)$$

where B_{z_cyl} is the z-axis component of the magnetic field in the cylindrical coordinate system [23]. These components in a Cartesian coordinate system are further elaborated in figure 3.6

The dipole \vec{B} is a three dimensional vector originating at the positive pole of the dipole, the magnetic south and ends at the magnetic north pole. In a Cartesian coordinate system (i, j, k) , the earth's magnetic field is given by:

$$\vec{B} = B_x \vec{i} + B_y \vec{j} + B_z \vec{k} \quad (3.15)$$

The x-axis is oriented along the geographical meridian and is positive in the North direction. The y-axis is aligned with the geographic parallel with East in the positive direction and z-axis is directed downwards. The magnetic field vector is further elaborated in figure 3.7. Here \vec{H} is the horizontal field component. The angle between the True North and \vec{H} is called the declination angle D whereas the angle between the magnetic field \vec{B} and horizontal plane is called the inclination angle I .

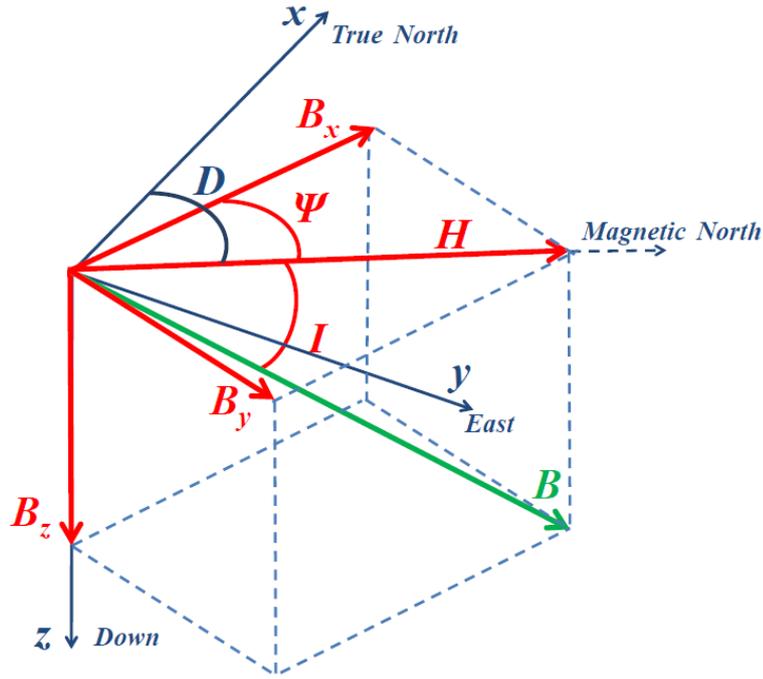


Figure 3.7: Earth's magnetic field in Cartesian coordinate system[23]

The horizontal field magnitude is given by:

$$H = (B_x^2 + B_y^2)^{1/2} \quad (3.16)$$

From figure 3.7 and by using trigonometric identities, it is possible to derive the relationships for inclination angle as:

$$I = \arctan \left(\frac{B_z}{(B_x^2 + B_y^2)^{1/2}} \right) \quad (3.17)$$

Due to non symmetric and time varying nature of the Earth's outer core, the Earth's magnetic field changes temporally as well as spatially. For instance, figure 3.8 illustrates the variation in the declination angle over Canada for a particular day.

3.3 Monte Carlo Methods

Sequential Monte Carlo (SMC) integration methods address the problem of the evaluation of complex high-dimensional integrals (formulated in the coming subsection). These methods are based on a set of simulations aimed to computing the posterior distributions and have the great advantage of not being subject to any linearity or Gaussianity constraints on the model, and

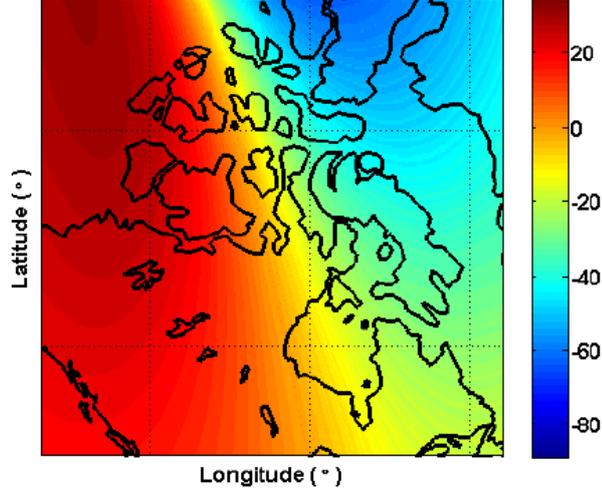


Figure 3.8: Colour map depicting the declination angle in degrees[23]

they also have appealing convergence properties.

The probability notations used in this chapter are: \mathbf{x}_t to denote both the random variable and its realisation. Consequently, the continuous probability distributions will be denoted using $p(d\mathbf{x}_t)$ and discrete distributions using $p(\mathbf{x}_t)$.

3.3.1 Problem Statement

For the sake of simplicity, the signals are modelled as Markovian, non-linear, non-Gaussian state-space models, though SMC can be applied in a more general setting. The unobserved signal (hidden states) $\{\mathbf{x}_t; t \in \mathbb{N}\}$, $\mathbf{x}_t \in \mathcal{X}$, is modelled as a *Markov* process of initial distribution $p(\mathbf{x}_0)$ and transition equation $p(\mathbf{x}_t|\mathbf{x}_{t-1})$. The observations $\{\mathbf{y}_t; t \in \mathbb{N}^*\}$, $\mathbf{y}_t \in \mathcal{Y}$, are assumed to be conditionally independent given the process $\{\mathbf{x}; t \in \mathbb{N}\}$ and of marginal distribution $p(\mathbf{y}_t|\mathbf{x}_t)$. The signal and the observations up to time t are denoted by $\mathbf{x}_{0:t} \triangleq \{\mathbf{x}_0, \dots, \mathbf{x}_t\}$ and $\mathbf{y}_{1:t} \triangleq \{\mathbf{y}_0, \dots, \mathbf{y}_t\}$, respectively.

The aim is to estimate recursively in time the *posterior distribution* $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, its associated features (including the marginal distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$, known as the *filtering distribution*, and the expectations.

$$I(f_t) = \mathbb{E}_{p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})}[f_t(\mathbf{x}_{0:t})] \triangleq \int f_t(\mathbf{x}_{0:t})p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})d\mathbf{x}_{0:t} \quad (3.18)$$

for some function of interest $f_t : \mathcal{X}^{(t+1)} \rightarrow \mathbb{R}^{n_{f_t}}$ integrable with respect to $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$. Examples of appropriate functions include the conditional mean, in which case $f_t(\mathbf{x}_{0:t}) = \mathbf{x}_{0:t}$, or the conditional covariance of \mathbf{x}_t where $f_t(\mathbf{x}_{0:t}) = \mathbf{x}_t\mathbf{x}_t^T - \mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}[\mathbf{x}_t]\mathbb{E}_{p(\mathbf{x}_t|\mathbf{y}_{1:t})}^T[\mathbf{x}_t]$.

At any time t , the posterior distribution is given by *Bayes's theorem*

$$p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})}{\int p(\mathbf{y}_{1:t}|\mathbf{x}_{0:t})p(\mathbf{x}_{0:t})d\mathbf{x}_{0:t}} \quad (3.19)$$

It is possible to obtain straightforwardly a recursive formula for this joint distribution $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$,

$$p(\mathbf{x}_{0:t+1}|\mathbf{y}_{1:t+1}) = p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) \frac{p(\mathbf{y}_{t+1}|\mathbf{x}_{t+1})p(\mathbf{x}_{t+1}|\mathbf{x}_t)}{p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t})} \quad (3.20)$$

The marginal distribution $p(\mathbf{x}_t|\mathbf{y}_{1:t})$ also satisfies the following recursion.

$$\textit{Prediction} : p(\mathbf{x}_t|\mathbf{y}_{1:t-1}) = \int p(\mathbf{x}|\mathbf{x}_{t-1})p(\mathbf{x}_{t-1}|\mathbf{y}_{1:t-1})d\mathbf{x}_{t-1} \quad (3.21)$$

$$\textit{Updating} : p(\mathbf{x}_t|\mathbf{y}_{1:t}) = \frac{p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})}{\int p(\mathbf{y}_t|\mathbf{x}_t)p(\mathbf{x}_t|\mathbf{y}_{1:t-1})d\mathbf{x}_t} \quad (3.22)$$

These expressions and recursions are deceptively simple because one cannot typically compute the normalizing constant $p(\mathbf{y}_{1:t})$, the marginals of the posterior $p(\mathbf{x}_{0:t}|\mathbf{y}_{1:t})$, in particular $p(\mathbf{x}_t|\mathbf{y}_t)$, and $I(f_t)$ since they require the evaluation of complex high-dimensional integrals.

When one has a large number of samples drawn from the required posterior distributions, it is not difficult to approximate the intractable integrals appearing in equations 3.19, 3.21 and 3.22. It is however, seldom possible to obtain samples from these distributions directly. One therefore has to resort to alternative MC methods, such as importance sampling (SIS) method. Unfortunately, it has been shown that SIS is guaranteed to fail as t increases, because the distribution of the importance weights $\tilde{w}^{(i)}$ becomes more and more skewed. Practically, after a few time steps, only one particle has non-zero importance weight. The algorithm, consequently, fails to represent the posterior distributions of interest adequately. This problem can be surmounted by including an additional selection step, which is explained in the coming section.

3.3.2 Bayes Filtering

Bayes filters address the problem of estimating the state \mathbf{x} of a dynamic system from sensors measurements. With the set of state and evidence variables for a given problem decided on, the next step is to specify how the world evolves (the transition model) and how the evidence variables get their values (the sensor model). The transition model specifies the probability distribution over the latest state variables, given the previous values. The mechanism used to calculate this probability is the first-order *Markov* process, in which the current state depends only on the previous state and not on any earlier states.

For example in mobile robot localization the dynamic system is a mobile robot and its environment, the state is the robot's pose therein (often specified by a position in a two-dimensional Cartesian space and the robot's heading direction θ), and measurements may include range measurements, camera images, and odometry readings. Bayes filters assume that the environments

An important particular case arises when the prior distribution is adopted as importance distribution in equation 3.24

$$\pi(\mathbf{x}_{0:t}|\mathbf{y}_{1:t}) = p(\mathbf{x}_{0:t}) = p(\mathbf{x}_0) \prod_{k=1}^t p(\mathbf{x}_k|\mathbf{x}_{k-1}) \quad (3.25)$$

In this case, the importance weights satisfy $w_t^{(i)} \propto \tilde{w}_{t-1}^{(i)} p(\mathbf{y}_t|\mathbf{x}_t^{(i)})$.

Note that in equation 3.23, $\tilde{w}_{t-1}^{(i)}$ does not appear because the propagated particles $\mathbf{x}_{0:t-1}^{(i)}$ have uniform weights after the resampling step at time $t - 1$. Additionally, it is not required to store the paths of the particles from time 0 to time t if we are only interested in estimating $p(\mathbf{x}_t|\mathbf{y}_{1:t})$.

The bootstrap particle filter has several attractive properties. Firstly, it is very quick and easy to implement. Secondly, it is to large extent modular. That is, when changing the problem one need only change the expressions for the importance distribution and the importance weights in the code. Thirdly, it can be straightforwardly implemented on a parallel computer. Finally, the resampling step is a black box routine that only requires as inputs the importance weights and indices (both being one-dimensional quantities). This enables one to easily carry out sequential inference for very complex models.

Chapter 4

Indoor Tracking by fusing PDR and CSI Information via Particle Filter

In the previous chapter, the underlying principles required to implement the tracking algorithms were introduced. The models defined in works about PDR methods presented in chapter 2 along with the concepts mentioned in chapter 3, constitute the main sources of information for the implementation of the velocity estimation algorithm presented in section 4.1 are summarized in the block diagram depicted in figure 4.1. Section 4.2 introduces the models used to convert the power feature from CIR into propagation distances, which allows to compute the distance between each anchor node (AN) and the current position. Based on the velocity and ranging estimations, section 4.3 introduces the integration of this two measurements in the observation model used for particle filter.

The main contribution of this thesis is the data fusion between ranging obtained from CSI and the velocity acquired from PDR to improve the tracking accuracy. The mechanisms and techniques used to accomplish this tasks are described hereinafter.

4.1 Velocity Estimation

Before describing the mechanisms used to compute the velocity, it is important to make a distinction between speed and velocity. Speed describes only how fast an object is moving, whereas velocity gives both how fast and in what direction the object is moving [26]. For instance, if a motorcycle is announced to travel at 100 km/h, its speed has been specified, while on the contrary if the same motorcycle is announced to move at 100 km/h to the north, its velocity has now been specified.

As described in last chapter, there are several mechanisms used to build a motion model based on PDR, which leads to derive the pedestrian's displacement. This thesis will go one step further to calculate the velocity of the pedestrian during his or her displacement based on this model, where the accelerometer and magnetometer from off-the-shelf smartphones are the only one source of information. In contrast to all the approaches presented in chapter 2, this thesis introduces a method to compute the velocity of a pedestrian user based on the timestamp recordings from the sensors readings. Before the overall function of the algorithm depicted in

figure 4.1 is explained, it is necessary to indicate an important aspect regarding the values taken from the inertial sensors.

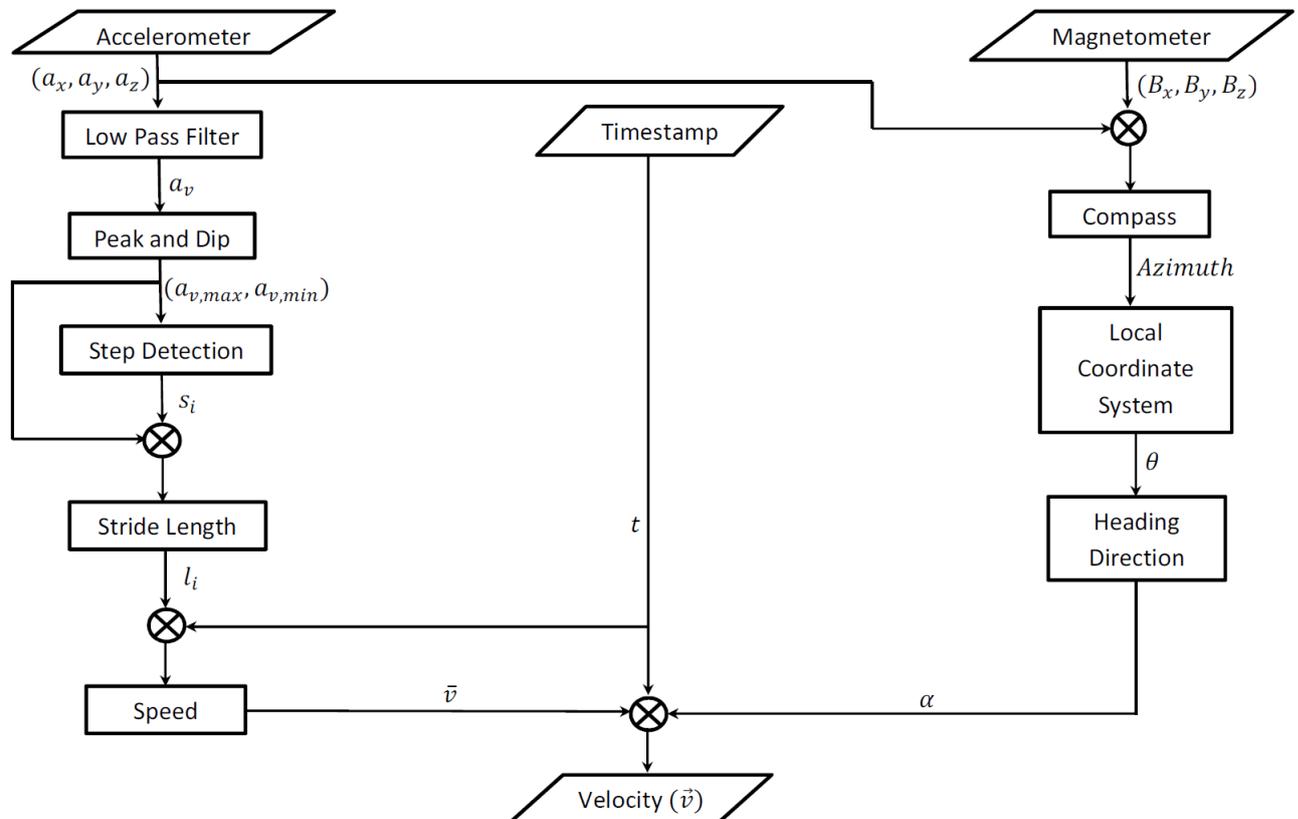


Figure 4.1: Flow chart for the algorithm used to compute the velocity

Most of the off-the-shelf smartphones receive an array of size $[1 \times 3]$, with values in m/s^2 for the accelerometer and in μT for the magnetometer. When the raw measurements from the sensors are taken, because some inertial sensors, e.g. accelerometer and magnetometer, read values from the three dimensions. As a result, every time a reading is taken from a sensor, it will have measurements for the three axes relative to the smartphone's orientation, i.e. the sensor coordinate system must be taken into account whenever a sensor reading is taken. This is particularly important for works where PDR systems are implemented, because it is required to know the vertical acceleration for the step detection, as well as for building a compass, where the direction is really important.

As an example to illustrate the natural orientation of a device, figure 4.2 shows the default position for Android smartphones, which is defined as portrait. This orientation has the sensor coordinate system matching the widely known Cartesian coordinate system, whereas when the orientation is landscape, as illustrated in figure 4.3, the sensor coordinate system does not match the Cartesian coordinate system.

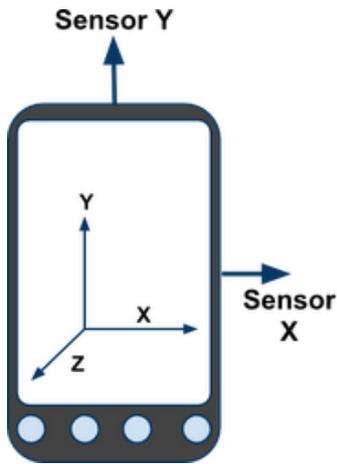


Figure 4.2: Portrait orientation

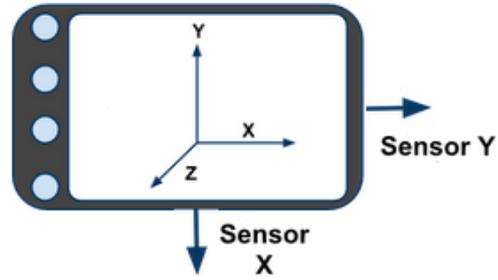


Figure 4.3: Landscape orientation

Table 4.1: Matching Cardinal and Sensor Coordinate Systems for Tracking Experiments

	Cartesian Coordinate System	Sensor Coordinate System
Axis	+x	+x
Axis	-x	-x
Axis	+y	+z
Axis	-y	-z
Axis	+z	-y
Axis	-z	+y

As the main contribution of this thesis is the data fusion, estimating the orientation of the smartphone is out of the scope of this work. Thus, in this thesis the tracking experiments (see chapter 6) assumed that the pedestrian user will always walk holding the smartphone flat as illustrated in figure 4.4. Therefore, the Cartesian coordinate system will match the sensor coordinate system following the manner described in table 4.1.

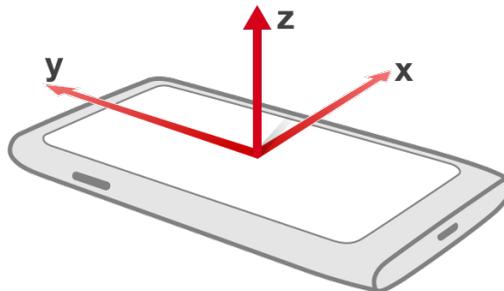


Figure 4.4: Sensor Coordinate System for Tracking Experiments

4.1.1 Speed Component

This component was mainly built based on [6], which was used for the step detection and stride length calculation. The first step to compute the speed is to take the raw values from the accelerometer, as shown in figure 4.1. Then, the sequence of accelerometer values is smoothed through a low pass filter using equation 4.1 and depending on the orientation of the smartphone, the vertical acceleration is taken from one of the three axes (a_x, a_y, a_z). As described in table 4.1, for this thesis the vertical acceleration matches axis -z, thus $a_v = -a_z$. Furthermore, the peak and dip from a_v were detected as steps through a time window, where a dip threshold was introduced to avoid false detections resulting from shallow dips.

$$a_{v,i} = (1 - \alpha)a_{v,i} + \alpha(a_{v,i-1}) (\alpha \geq 0.9) \quad (4.1)$$

After each step is detected, equation 4.2 is used to estimate the stride length, which was proposed in [9] and has been widely used in many PDR systems. l is the stride length, $a_{v,max} - a_{v,min}$ is the peak-to-dip value from a_v on each time window, and K is a coefficient calibrated for individuals.

$$l = K(a_{v,max} - a_{v,min})^{1/4} \quad (4.2)$$

Finally, the speed estimation in this thesis is done through a straightforward model, that just consisted of computing the mean of the speed calculated for each stride. This average speed is determined using the widely known velocity formula from physics laws of motion. Thus, by applying equation 4.3 the average speed was estimated for each tracking experiment, this speed is assumed to be constant.

$$\bar{v} = \frac{1}{N} \sum_{i=1}^N \left(\frac{l_i}{\Delta t_i} \right) \quad (4.3)$$

N is the total number of steps, l_i and Δt_i are the straight length and the time interval at the i th step respectively. The timestamp information as shown in 4.1 is used to compute the time interval between each successive step.

4.1.2 Heading Direction Component

The heading direction component was built using a software-based sensor, which is known in Android platform as orientation sensor, this sensor derives its data from the accelerometer and magnetometer. The mathematics behind this software-based sensor which allows to build an electronic compass[28] are presented herein. The compass heading is a function of all three accelerometer and magnetometer readings as depicted in figure 4.1. Considering that in all the tracking experiments the smartphone will remain flat as illustrated in 4.4, then the compass heading could be computed from the arctangent of the ratio of the two horizontal magnetic field components.

A positive yaw angle ψ is defined to be a clockwise rotation about the negative z-axis. Similarly, a positive pitch angle θ and positive roll angle ϕ are defined as clockwise rotations

about the positive x and positive y-axes respectively. The accelerometer readings provides pitch and roll angle information which is used to correct the magnetometer data.

Any orientation of the smartphone can be modelled as resulting from rotations in yaw, pitch and roll applied to a starting position with the phone flat and pointing northwards. The accelerometer, G_r , and magnetometer, B_r , readings in this starting reference position are shown in equations 4.4 and 4.5 respectively.

$$G_r = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (4.4)$$

$$B_r = B \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} \quad (4.5)$$

The acceleration due to gravity is $g = 9.81m/s^2$. B is the geomagnetic field strength which varies over the earth's surface from a minimum of $22 \mu T$ over South America to a maximum of $67 \mu T$ south of Australia. δ is the angle of inclination of the geomagnetic field measured downwards from horizontal and varies over the earth's surface from -90° at the south magnetic pole, through zero near the equator to $+90^\circ$ at the north magnetic pole.

The smartphone accelerometer, G_p , and magnetometer, B_p , readings measured after the three rotations $R_z(\psi)$, then $R_y(\theta)$ and finally $R_x(\phi)$ are described by the equations:

$$G_p = R_x(\phi)R_y(\theta)R_z(\psi)G_r = R_x(\phi)R_y(\theta)R_z(\psi) \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (4.6)$$

$$B_p = R_x(\phi)R_y(\theta)R_z(\psi)B_r = R_x(\phi)R_y(\theta)R_z(\psi)B \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} \quad (4.7)$$

The three rotation matrices referred to in equations 4.6 and 4.7 are:

$$R_x(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \quad (4.8)$$

$$R_y(\theta) = \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \quad (4.9)$$

$$R_z(\psi) = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.10)$$

Equation 4.6 assumes that the smartphone is not undergoing any linear acceleration and that the accelerometer signal G_p is a function of gravity and the smartphone orientation only. An electronic compass will give erroneous readings if it is subject to any linear acceleration.

The magnetometer readings must be corrected for Hard-Iron and Soft-Iron effects to reduce noise generated from nearby magnetic fields which disturbs the magnetometer readings. Equation 4.7 ignores any stray magnetic fields from Hard and Soft-Iron effects. The standard way of modelling the Hard-Iron effect is as an additive magnetic vector, V , which rotates with the sensor coordinate system and is therefore independent of smartphone orientation. Equation 4.7 then becomes:

$$B_p = R_x(\phi)R_y(\theta)R_z(\psi)B \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} + V = R_x(\phi)R_y(\theta)R_z(\psi)B \begin{pmatrix} \cos \delta \\ 0 \\ \sin \delta \end{pmatrix} + \begin{pmatrix} V_x \\ V_y \\ V_z \end{pmatrix} \quad (4.11)$$

V_x , V_y and V_z , are the components of the Hard-Iron vector.

The orthogonal components of the horizontal magnetic field (see chapter 5) are used for estimating the heading. In order to calculate the yaw angle ψ , where ψ is computed relative to magnetic north, a tilt-compensated algorithm is used. This algorithm first calculates the roll and pitch angles ϕ and θ from the accelerometer reading by pre-multiplying 4.6 by the inverse roll and pitch rotation matrices giving:

$$R_y(-\theta)R_x(-\phi)G_p = R_y(-\theta)R_x(-\phi) \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = R_z(\psi) \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (4.12)$$

where the vector $\begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix}$ contains the three components of gravity measured by the accelerometer.

Expanding equation 4.12 gives:

$$\begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (4.13)$$

$$\Rightarrow \begin{pmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \begin{pmatrix} G_{px} \\ G_{py} \\ G_{pz} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \quad (4.14)$$

The y component of equation 4.14 defines the roll angle ϕ as :

$$G_{py} \cos \phi - G_{pz} \sin \phi = 0 \quad (4.15)$$

$$\Rightarrow \tan(\phi) = \left(\frac{G_{py}}{G_{pz}} \right) \quad (4.16)$$

The x component of equation 4.14 defines the pitch angle θ as:

$$G_{px} \cos \theta + G_{py} \sin \theta \sin \phi + G_{pz} \sin \theta \cos \theta = 0 \quad (4.17)$$

$$\Rightarrow \tan(\theta) = \left(\frac{-G_{px}}{G_{py} \sin \phi + G_{pz} \cos \phi} \right) \quad (4.18)$$

With the angles ϕ and θ known from the accelerometer, the magnetometer reading can be de-rotated to correct for the phone orientation using equation 4.11:

$$R_z(\psi) \begin{pmatrix} B \cos \delta \\ 0 \\ B \sin \delta \end{pmatrix} = \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} B \cos \delta \\ 0 \\ B \sin \delta \end{pmatrix} = R_y(-\theta) R_x(-\phi) (B_p - V) \quad (4.19)$$

$$\Rightarrow \begin{pmatrix} \cos \psi B \cos \delta \\ -\sin \psi B \cos \delta \\ B \sin \delta \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} B_{px} - V_x \\ B_{py} - V_y \\ B_{pz} - V_z \end{pmatrix} \quad (4.20)$$

$$= \begin{pmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{pmatrix} \begin{pmatrix} B_{px} - V_x \\ B_{py} - V_y \\ B_{pz} - V_z \end{pmatrix} \quad (4.21)$$

$$\begin{pmatrix} (B_{px} - V_x) \cos \theta + (B_{py} - V_y) \sin \theta \sin \phi + (B_{pz} - V_z) \sin \theta \cos \phi \\ (B_{py} - V_y) \cos \phi - (B_{pz} - V_z) \sin \phi \\ -(B_{px} - V_x) \sin \theta + (B_{py} - V_y) \cos \theta \sin \phi + (B_{pz} - V_z) \cos \theta \cos \phi \end{pmatrix} = \begin{pmatrix} B_{fx} \\ B_{fy} \\ B_{fz} \end{pmatrix} \quad (4.22)$$

The vector $\begin{pmatrix} B_{fx} \\ B_{fy} \\ B_{fz} \end{pmatrix}$ represents the components of the magnetometer sensor after correcting for the Hard-Iron offset and after de-rotating to the flat plane where $\theta = \phi = 0$.

The x and y components of equation 4.22 give:

$$\cos \psi B \cos \delta = B_{fx} \quad (4.23)$$

$$\sin \psi B \cos \delta = B_{fy} \quad (4.24)$$

$$\Rightarrow \tan(\psi) = \left(\frac{-B_{fy}}{B_{fx}} \right) = \left(\frac{(B_{pz} - V_z) \sin \phi - (B_{py} - V_y) \cos \phi}{(B_{px} - V_x) \cos \theta + (B_{py} - V_y) \sin \theta \sin \phi + (B_{pz} - V_z) \sin \theta \cos \phi} \right) \quad (4.25)$$

Equation 4.25 allows solution for the yaw angle ψ .

Since equations 4.16, 4.18 and 4.25 have an infinite number of solutions at multiples of 360° , it is standard convention to restrict the solutions for roll, pitch and yaw to the range -180° to 180° . A further constraint is imposed on the pitch angle to limit it to the range -90° to 90° . This ensures only one unique solution exists for the compass, pitch and roll angles for any phone orientation.

Equation 4.25 assumes knowledge of the Hard-Iron offset V , which is a fixed magnetic offset adding to the true magnetometer sensor output. The Hard-Iron offset is the sum of any intrinsic zero field offset within the magnetometer sensor itself plus permanent magnetic fields within the Printed Circuit Board (PCB) generated by magnetized ferromagnetic materials. It is quite normal for the Hard-Iron offset to greatly exceed the geomagnetic field. It is common practice for magnetometer sensors to be supplied without zero field offset calibration since the standard Hard-Iron estimation algorithms will compute the sum of both the magnetometer sensor zero field offset and the PCB Hard-Iron offset.

In the absence of any Hard-Iron effects, the locus of the magnetometer output under arbitrary smartphone orientation changes lies on the surface of a sphere in the space of B_{px} , B_{py} and B_{pz} with a radius equal to the magnitude of the geomagnetic field B . In the presence of Hard-Iron effects, the locus of the magnetic measurements is simply displaced by the Hard-Iron vector V so that the origin of the sphere is equal to the Hard-Iron offset V_x , V_y and V_z . The Hard-Iron offset V can then be computed by fitting the magnetometer measurements to the equation:

$$(B_p - V)^T (B_p - V) = B^2 \quad (4.26)$$

The mathematics and algorithms for computing V using equation 4.26 can be found in [29].

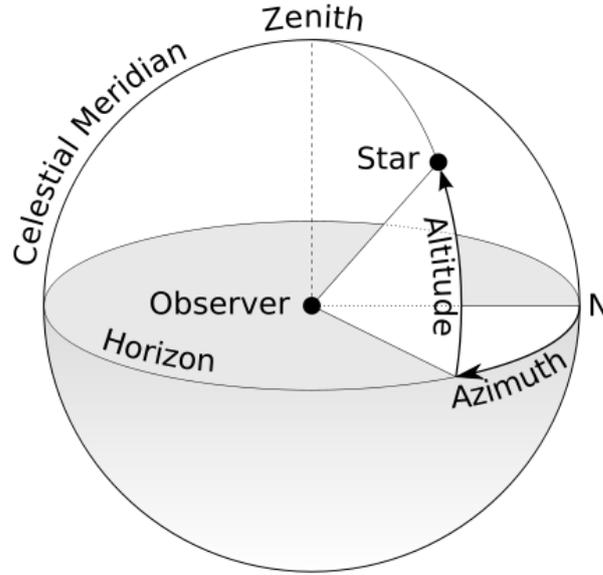


Figure 4.5: Azimuth Angle

Azimuth Computation

An *Azimuth* is an angular measurement in a spherical coordinate system. The *Azimuth* is the angle formed between a reference direction (North) and a line from the observer to a point of interest projected on the same plane as the reference direction orthogonal to the *Zenith* as depicted in figure 4.5.

The final step to build a compass is to calculate the *Azimuth*. Heading is estimated using the orthogonal components of the horizontal magnetic field (see chapter 3). Thus the local level must be known in order to find the horizontal field component of the measured Earth's magnetic field. In order to estimate the heading with respect to true North instead of the magnetic North, the declination angle D also needs to be predicted using one of the Earth's magnetic field models [31],[22]. After resolving the magnetic field to the local level and estimating the declination angle, a simple trigonometric relationship derived from equation 4.25 is used for estimating the heading from the measured Earth's magnetic field:

$$Azimuth = \arctan\left(\frac{-B_{fy}}{B_{fx}}\right) \pm D \quad (4.27)$$

Once the compass has been built, the final step is to use the *Azimuth* measurements and rotate those values around the horizontal axis to calibrate the heading direction with the local coordinate system, as shown in equation

$$\theta = \begin{cases} 90 - Azimuth & \text{if quadrant is I} \\ 450 - Azimuth & \text{else} \end{cases} \quad (4.28)$$

Finally, the speed and heading direction components are used together to derive the velocity, using equation 4.29

$$\vec{v} = \begin{pmatrix} \bar{v} \cdot \sin \theta \\ \bar{v} \cdot \cos \theta \end{pmatrix} \quad (4.29)$$

\bar{v} is the speed from equation 4.3 and θ is the heading direction from equation 4.28.

4.2 Ranging Algorithm

Ranging refers to determining the distance between the AN and a given static position. The range-based localization model used in this thesis is based on [17]. To define the relationship between the propagation distances and the estimated power, first it is required to acquire the fine-grained information from CIR (see chapter 3), which is accomplished with the configuration described in chapter 5. Figure 4.6 displays an example of the amplitudes of the measured CIR in the tracking experiments presented in chapter 6.

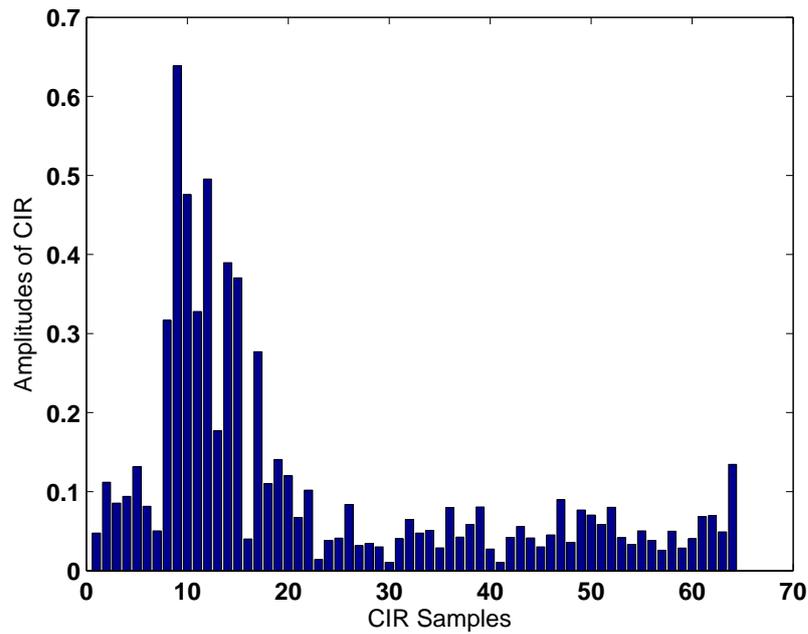


Figure 4.6: Channel Impulse Response Tracking Experiments

It is possible to clearly identify a strongest path between the CIR samples[17] depicted in figure 4.6, which is used in this thesis as the estimated power e.g. RSS. This path is chosen

in an attempt to mitigate the influence of multipath propagation, as the signal from the line of sight (LOS) propagation is the one with the strongest power, whereas the others should be from multipath propagation. Thus, equation 4.30 defines the final estimated power. This equation allows to obtain the strongest power even if no LOS exists, by selecting between the weak non line of sight (NLOS) paths, the one with the strongest power.

$$RSS = 10 \cdot \log_{10}[\max(|h(\tau)|)^2] \quad (4.30)$$

$|h(\tau)|$ indicates the amplitudes of CIR over 64 samples.

The relationship between RSS values and propagation distances is modelled as a nonlinear curve fitting problem[17]. Equation 4.31 characterizes the non linear regression model (NLR) used to compute the ranging values for each AN.

$$\hat{d}_i = \alpha_i \cdot e^{\beta_i \cdot RSS_i} \quad (4.31)$$

\hat{d}_i is the distance between the target node and i th AN, RSS_i is the RSS values obtained at the i th AN, α_i and β_i are two unknown environmental parameters in the model that need to be obtained empirically. In this thesis, specific (α, β) pairs are set for each AN from some initial static measurements and those specific values are reported in chapter 6.

Given K training positions (see chapter 6) in the initial static measurements, (d_{ij}, RSS_{ij}) are collected at the j th training position from the i th AN. In this thesis the nonlinear least square criterion was applied, in which the sum of squared residuals should be minimized as described in equation 4.32

$$\arg \min_{(\alpha_i, \beta_i)} \sum_{j=1}^K (\alpha_i \cdot e^{\beta_i \cdot RSS_{ij}} - d_{ij})^2 \quad (4.32)$$

This equation requires minimizing an objective function, therefore, it can be defined as an unconstrained optimization problem. There have been extensive studies and developments of algorithms for solving unconstrained optimization problems. In this thesis iterative methods that converge to a solution for optimization are used. Iterative methods for optimization can be classified in two categories: line search methods and trust region methods [24]. In this thesis the latter [20] is applied, because it is robust and has strong global convergence properties.

Figures from ?? to 4.7 illustrates the NLR model to fit the RSS measurements and the ground truth distances for each AN from the stationary points as described in chapter 6.

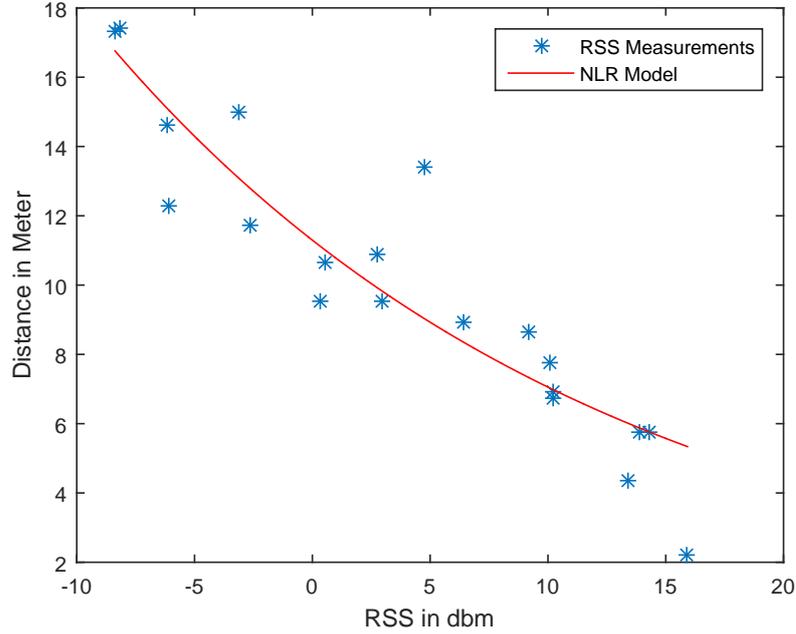


Figure 4.7: NLR model

4.3 Data Fusion

This section introduces the integration of the velocity from PDR and ranging from CIR described in the last two sections above. This thesis adopted particle filter to achieve such task, because one of the most relevant characteristics of particle filter is data fusion. The main contribution of this thesis is to fuse the velocity from PDR with ranging from CIR to improve the tracking accuracy.

For this thesis the transition model is described in equation 4.35, whereas the observation models are described in equations 4.36 and 4.37.

Particle filter is initialized with a set of weighted particles:

$$X_t = \langle p_k^i, w_k^i \rangle, i = 1, \dots, N \quad (4.33)$$

$$p_k^i = \begin{pmatrix} x_k^i \\ y_k^i \\ \hat{x}_k^i \\ \hat{y}_k^i \end{pmatrix} \quad (4.34)$$

p_k^i is the estimated position and velocity with weight w_t^i of the i th particle at step k . N is the

number of particles. The new set p_k^i is iteratively generated from p_{k-1}^i using the transition model below:

$$\begin{pmatrix} x_k^i \\ y_k^i \\ \hat{x}_k^i \\ \hat{y}_k^i \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1}^i \\ y_{k-1}^i \\ \hat{x}_{k-1}^i \\ \hat{y}_{k-1}^i \end{pmatrix} + \begin{pmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{pmatrix} \mathcal{N} \quad (4.35)$$

Δt is the time interval between each iteration, and \mathcal{N} represents the system noise which is zero-mean and Gaussian.

Two estimations are used in particle filter to derive the position: ranging and fusion. The former considers just equation 4.36, which is the ranging observation from CIR, and the latter considers equations 4.36 and 4.37, which is the velocity observation from PDR in addition to ranging.

$$r_k^i = d_k^{i,j} - d_k^{CSI,j} \quad (4.36)$$

r_k^i is the ranging observation for the i th particle at step k , $d_k^{i,j}$ is the distance between the i th particle and the j th AN, and $d_k^{CSI,j}$ is the estimated distance from CIR (see equation 4.31) and the j th AN

$$v_k^i = \begin{pmatrix} \hat{x}_k^i \\ \hat{y}_k^i \end{pmatrix} - \begin{pmatrix} \hat{x}_k^{PDR} \\ \hat{y}_k^{PDR} \end{pmatrix} \quad (4.37)$$

v_k^i is the velocity observation for the i th particle at step k , \hat{x}_k^i is the i th velocity from particle filter in the x-axis, \hat{x}_k^{PDR} is the estimated velocity from PDR (see equation 4.29 in the x-axis, \hat{y}_k^i is i th velocity from particle filter in the y-axis and \hat{y}_k^{PDR} is the estimated velocity from PDR in the y-axis.

Once the observation models have been defined, the particle weights can be updated, as described in equation 4.38. The weights are computed under the assumption that the position estimated by CIR and the velocity estimated by PDR are Gaussian distributed around the true values.

$$w_k^i = \begin{cases} w_{k-1}^i \cdot \prod_{j=1}^{N_{ANs}} P(r_k^i | p_k^i) & \text{if observation is ranging} \\ w_{k-1}^i \cdot \prod_{j=1}^{N_{ANs}} (P(r_k^i | p_k^i)) \cdot P(v_{x,k}^i | p_k^i) \cdot P(v_{y,k}^i | p_k^i) & \text{if observation is ranging and } \vec{v} \end{cases} \quad (4.38)$$

In this thesis, one additional functionality will be implemented to be analysed in chapter 6, which consists in fusing the CIR output values from three different antennas. Thus, equation 4.38 becomes:

$$w_k^i = \begin{cases} w_{k-1}^i \cdot \prod_{j=1}^{N_{ANs}} \cdot (\prod_{n=1}^3 P(r_k^i | p_k^i)) & \text{if observation is ranging} \\ w_{k-1}^i \cdot \prod_{j=1}^{N_{ANs}} \cdot (\prod_{n=1}^3 P(r_k^i | p_k^i)) \cdot P(v_{x,k}^i | p_k^i) \cdot P(v_{y,k}^i | p_k^i) & \text{if observation is ranging and } \vec{v} \end{cases} \quad (4.39)$$

Chapter 5

Indoor Tracking System Implementation

Chapters 3 and 4 conveyed the general knowledge of the approaches required to implement an indoor tracking system. In order to develop the algorithms based on those concepts, first it is required to establish a testing environment upon which the programs for localization and tracking can be executed whether in an on-line or off-line fashion. This chapter introduces in section 5.1 all the configurations and modifications at hardware and software level for an appropriate and effective operation for this testbed environment.

5.1 Testbed Overview

In order to be able to take all the measurements required to evaluate the experiments, a testbed environment is required. The set-up of this testbed comprises both hardware and software specific configurations for reading the CSI information from Wireless Network Interface Controllers (WNICs). Therefore, a network localization system is designed to accomplish this task. Figure 5.1 outlines in detail the main components of the testbed required for its operation, with anchor nodes (ANs), a central server and a single mobile node. In this localization system the target is a mobile node, which broadcasts wireless packets including the measurements from the inertial sensors. The CSI information will be extracted at the receivers' side, i.e. ANs, meanwhile all the data will be sent to a central server where the programs required for localization and tracking will be executed at the end by an off-line process.

For the communication, the ANs and the server will be connected through a wired Ethernet network, whereas the mobile node and the anchor nodes communication with each other is done in a wireless context.

Regarding the configurations for gathering the CSI information from WNIC, a tool that records detailed measurements of the wireless channel along with received 802.11 packet traces is used [32], which is known as CSI Tool. Running on a commodity 802.11n WNIC, this CSI tool requires some configurations both at the sender and receivers sides to generate CSI information effectively. For this thesis the *monitor mode* option is used and the main requirements for receiver and sender nodes are detailed below:

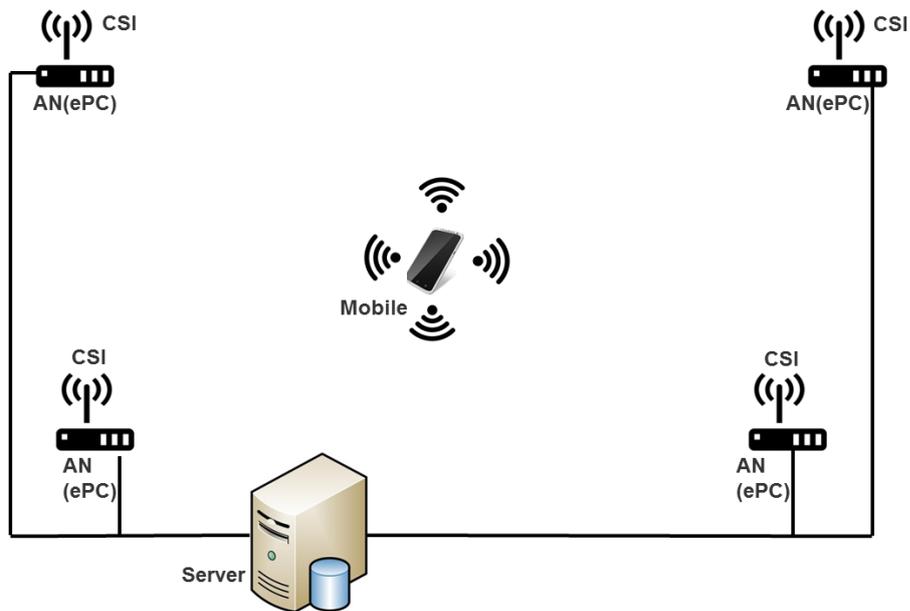


Figure 5.1: Network-based Localization System

Receiver:

1. The receiver's logical configuration (e.g., channel type is HT, and channel width) and its physical configuration (e.g., correct number of antennas connected, all antennas work in the configured band, all antennas are enabled in the firmware) support the transmitted rate.
2. The receiver decodes the packet correctly, with no errors.

Sender:

1. The packet is transmitted using an 802.11n HT rate.
2. The packet is addressed both to and from a hardcoded, fixed address 00:16:ea:12:34:56.
3. The packets must be sent using packet injection, i.e. transmit packets in monitor mode.

The specific configurations and operations for each one of these devices are presented hereinafter.

5.2 Hardware Set-up

It is required to have three kinds of different machines for these experiments: One for the anchor node, another one for the central server and finally one for the target node. The hardware components employed are the following:



Figure 5.2: Commercial EeeBox PC

5.2.1 Anchor Nodes

The machines used as anchor nodes are commercially known as EeeBox PC, shown in Figure 5.2. In this thesis those devices will be mentioned hereafter as ePCs. The ePC's technical specifications are the following:

- **OS:** Linux Ubuntu 10.04.4 LTS (Lucid Lynx)
- **Processor:** Intel Atom D410/1.66 GHz
- **WiFi card:** 802.11b/g/n
- **Memory:** RAM 1GB
- **Storage:** 250GB

The ePC comes by default with WiFi interfaces that are not able to extract CSI information. Thus, it was replaced (see Figures 5.3, 5.4 and 5.5) by Intel WiFi Wireless Card (IWL) 5300 802.11n MIMO with 3 antennas radios, which provides 802.11n channel state information in a format that reports the channel matrices for 30 subcarrier groups, which is about one group for every 2 subcarriers at 20 MHz or one in 4 at 40 MHz [32].



Figure 5.3: Disassembled ePC with antennas (part one)



Figure 5.4: Disassembled ePC with antennas (part two)



Figure 5.5: Intel WiFi Wireless Card 5300

5.2.2 Target Node

The machine used as target node is a mobile device. In the experiments a Samsung Galaxy S6 Edge was used with the following technical specifications:

- **OS:** Android 5.0 (Lollipop)
- **Processor:** Quad 2.1GHz + Quad 1.5GHz
- **WiFi card:** 802.11 a/g/n/ac (2.4/5GHz), HT80 MIMO(2 × 2) 620Mbps, Dual band
- **Memory:** RAM 3GB
- **Storage:** 32GB
- **Inertial Sensors:** Accelerometer, Light, Gyroscope, Proximity, Magnetometer, Barometer, Fingerprint, Hall, HRM

At the time of the experiments, Samsung Galaxy S6 Edge was between the devices with the most advanced inertial sensors for smartphones, which was a requirement for a suitable operation of the network system and above all to acquire the information of the inertial sensors as fast as possible. The inertial sensors used for this measurements were the Accelerometer working with a frequency of $220Hz$ and the Magnetometer working with a frequency of $100Hz$.

5.2.3 Central Server

The machine that will act as a central server obviously requires to have more capabilities than the other nodes, because of the data that will be store there and the programs to process the localization and tracking algorithms. This server has the following technical specifications:

- **OS:** Linux Ubuntu 12.04.5 LTS (Precise Pangolin)
- **Processor:** Intel Core i7 2.40GHz
- **Ethernet card:** Realtek PCIe GBE

- **Memory:** RAM 16GB
- **Storage:** 1TB

5.3 Software Set-up

Each one of the machines involved in the network localization system, required specific software configurations to operate appropriately. For including the measurements from the inertial sensors in the wireless packets, the payload of such packets was used to accomplish that task, which requires additional modifications. Figure 5.6 summarizes the software architecture for the whole Testbed.

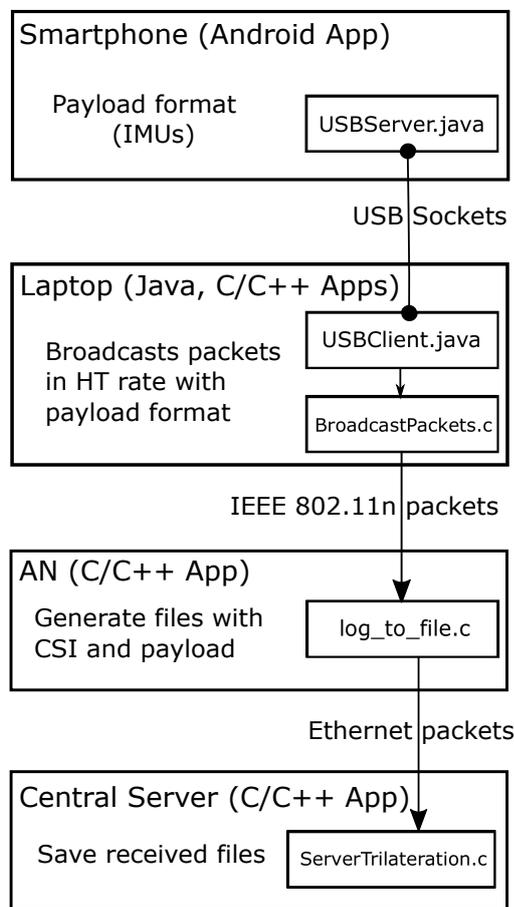


Figure 5.6: Testbed Software Implementation

The target node is composed of a smartphone and laptop connected through USB. In the smartphone an Android application builds the values of the payload by concatenating in a string

format the readings from inertial sensors, i.e. IMUs, and the timestamp. Then, the laptop using a Java application receives this string value in a USB socket, forthwith a C/C++ program ensembles it in the payload of an IEEE 802.11n packet (see figure 5.7), and periodically broadcasts packets in HT rate.

Afterwards, the ANs upon reception of the packets, generate the CSI information and save each packet in a binary file using a C/C++ program. To assure uniqueness, this file is named by taking part of the timestamp from the payload and the AN's number. For instance following the example from figure 5.7, and assuming it was received in AN4, then this packet would generate a file named 1072688414EP004. Once the file is generated, each AN sends it towards the central server over the Ethernet network.

Finally, the central server implements a C/C++ listener program which receives all the Ethernet packets and save these binary files in a directory for each AN. This allows the central server to run the localization and tracking programs locally.

The specific framework implementation for each actuator presented in figure 5.6 is described below:

5.3.1 Target Node

As described at the beginning of the chapter, the mobile target required some particular configurations for sending packets in *monitor mode* such that receivers can extract the correspondent CSI information. Additionally to these CSI Tool requirements, the target node also requires to take into account two modifications: First the environment upon the configuration will be made is Android, and second the packets that are sent must include in the payload the values from the inertial sensors, such that receivers can extract the values from accelerometer and magnetometer in addition to the CSI information.

The first step to accomplish these configurations is to fulfil the requirements described in section 5.1 for CSI Tool on the sender side. CSI Tool has been successfully tested in Linux environments, where the main components to satisfy the requirements for sending the packets as CSI Tool demands is to activate *monitor mode* in the wireless interface and use LORCON (Loss of Radio Connectivity), which is an open source C/C++ library for injecting 802.11 frames.

Under Android environment LORCON library installation can be managed through NDK (Native Development Kit), where a cross-compiler can be used to compile the required C/C++ libraries, such that, the programs for packet injection can be interpreted and run by Android. On the other hand activating *monitor mode* in Android is a challenging task, because most of the mobile device's WNICs (including vendors like Apple, Samsung, Motorola, Sony, Nokia, LG, Asus and HTC) do not implement *monitor mode* [27]. Thus, a modification of the firmware and driver of the adapter is required to enable it. Moreover, to activate HT rate transmission for injection also requires modifications for the drivers as was done for the Intel 5300 WNIC.

Under the constraints aforementioned and considering that the aim of this thesis is to evaluate the performance of a WiFi indoor tracking scenario, a faster alternative solution was adopted to enable the target node to fulfil CSI Tool requirements and workaround with the payload. As depicted in figure 5.6, this solution consists of connecting through USB the smartphone to a laptop, such that, the laptop is used as a bridge to include the inertial sensors data in the payload of the packets, which in turn are broadcast in *monitor mode* and with HT rate. The

configurations and components implemented to achieve this are detailed below:

1. Installation of Intel 5300 WNIC and CSI Tool in a laptop under Linux Ubuntu 12.04.
2. Configuration of LORCON libraries for packet injection.
3. Android application running in the smartphone, which plays the role of server for the USB communication and sends the inertial sensor values to the laptop.
4. Java application running in the laptop, which plays the role of client for the USB communication and receives the data from the inertial sensors to execute a C/C++ program that sends the wireless packets.
5. C/C++ program running on the laptop, which takes the values from the inertial sensors and include those in the payload before broadcasting the wireless packets.

The Android application takes the measurements from the accelerometer, compass and the time between each sensor event, which is a timestamp in nanoseconds. The timestamp has two purposes: First it is used to uniquely identify each packet, and second to enable the computation of time intervals and total times for the tracking experiments (see chapter 6)

The values that are sent into the payload are depicted in figure 5.7. Accelerometer readings are raw values, whereas the magnetometer readings are post processed through the software-based orientation sensor, which outputs the *azimuth* values, i.e. electronic compass (see chapter 4). Considering that the orientation sensor requires accelerometer and magnetometer values, then the inertial sensors reading frequency is defined for the lowest one, in this case the magnetometer, which means $100Hz$.

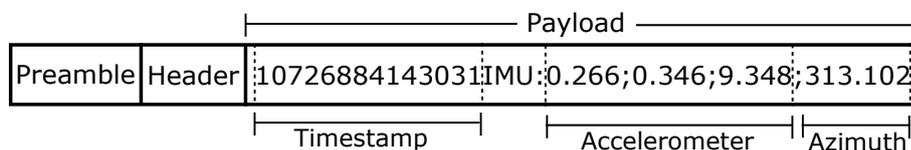


Figure 5.7: IEEE 802.11n packet with payload format

5.3.2 Anchor Nodes

Once the ePCs have the IWL 3000 WNIC, those will be able to generate the CSI information upon reception of the wireless packets sent by the target node. This is possible because of a custom modified firmware and open source Linux drivers built on the IWL 5300 WNIC, which is known as CSI Tool [32] as described at the beginning of this chapter.

In order to generate the CSI information and further send these data to the central server over the Ethernet network, the following software configurations were done in the ANs:

1. Install CSI Tool as described in [34]. A correct configuration of this point, fulfils the requirements described in section 5.1 for CSI Tool on the receiver side.

2. Modifications in *log_to_file* C/C++ program:

- Generation of files for each packet received, following the format: TTTTTTTTTT-TEP00N. TTTTTTTTTT is the timestamp value from the smartphone and N is the ePC number. The aim of this format is to have uniqueness in the files to avoid overwriting.
- Ethernet sockets enable sending the generated files with the CSI information and payload to the central server.

The packet transmission is around 100 packets per second. In the tracking measurements (see chapter 6), 40 packet per second on average were received.

5.3.3 Central Server

The central server plays two roles, first it is the repository of information, where all the data is stored, and it is also the machine where the Matlab programs implementing the algorithms for localization and tracking are run in an off-line fashion. CSI Tool required a modification in the Matlab program *read_bf_file* and the C/C++ program *read_bfee* to enable payload readings along with the CSI information. Following these modifications the components created in Matlab are the following:

- **Ranging:** This component is addressed to train the environmental parameters (α, β) of the NLR model using the trust region algorithm (see chapter 6).
- **Tracking:** This component has the core algorithms for localization and tracking which includes PDR, ranging particle filter and fusion particle filter (see chapter 6).
- **Evaluation:** This component has the algorithms required to generate the CDFs, based on the ground truth positions from the tracking experiments and the estimated ones from the localization algorithms (see chapter 6).

Chapter 6

Evaluation

In previous chapters the theoretical background (see chapter 3) and models specifications (see chapter 4) used to implement the network-based localization system (see chapter 5) required to evaluate the approaches developed in this thesis have been described. This chapter introduces the results of the measurements for one environment, but with two different set-up configurations. Results are analysed and evaluated to show the performance of the different approaches studied in this thesis.

Before presenting the evaluation results, section 6.1 introduces the environment upon which the network-based localization system was built, and how the experiments were undertaken. In section 6.2 the result of the experiments are presented, where the performance is evaluated by analysing the cumulative distribution function (CDF) of the algorithms for the whole experiments.

6.1 Environment Configuration

The experiments were conducted at the *Institute of Computer Science (INF)* of the *University of Bern*, where part of the third floor with an area of 297 m^2 was chosen to deploy the network localization system (see chapter 5). The floor's blueprint depicted in Figure 6.1 labels the main sections where the experiments took place.

The choice of this floor in particular was a purely pragmatic decision because most of the offices located in such floor by the time the experiments took place were free, such that the measurements could be conducted without disturbing other ones. Additionally, in this floor there are offices of different sizes, which facilitates the diversification of walking trajectories for further evaluation. As it is illustrated in Figure 6.1 the rooms where the tracking experiments took place are seminar room, server room, office one, office two, office three, office four and corridor.

The experiments were organized in two different sessions: the first one deploying four anchor nodes (ANs) and the second one deploying five ANs as is illustrated in Figures 6.2 and 6.3 respectively. In the former, anchor node one (AN1) was deployed in the seminar room,

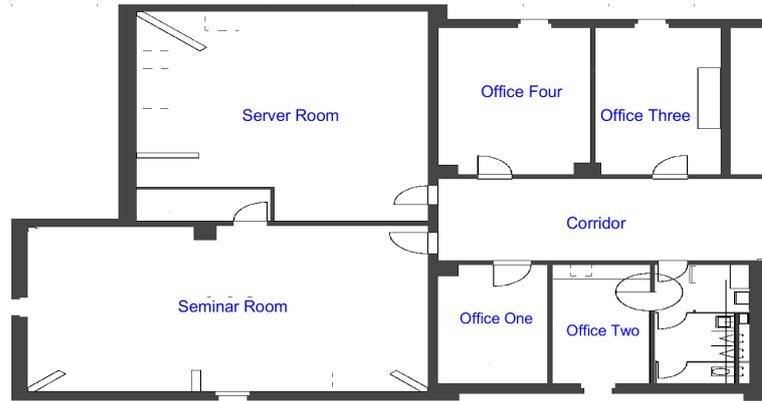


Figure 6.1: Third floor's blueprint from INF building

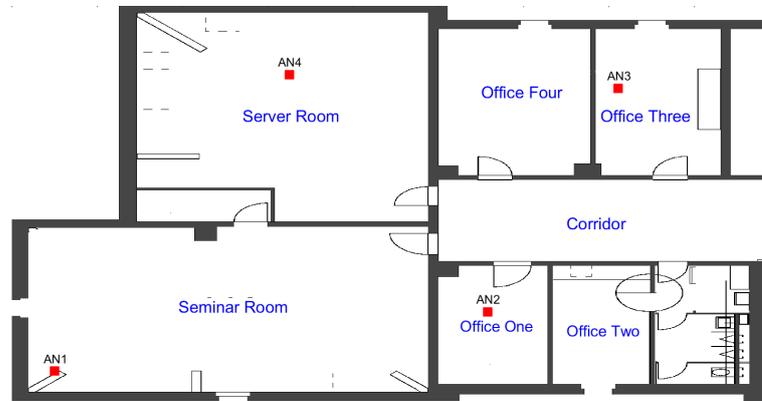


Figure 6.2: Floor plan with four anchor nodes (ANs) for experiment 1

anchor node two (AN2) in office one, anchor node three (AN3) in office three and anchor node four (AN4) in the server room, for the latter AN1 was deployed in seminar room, AN2 in office one, AN3 in office two, AN4 in office three and anchor node five (AN5) in the server room. Considering that these sets of experiments will be referred later on, the environment with four anchor nodes will be defined as *experiment 1*, whereas the environment with five anchor nodes will be defined as *experiment 2*.

In *experiment 1*, the distribution of the anchor nodes was made in an attempt of covering the entire area, such that all the ANs were able to received the wireless packets. This specific deployment was required because ranging with less than four ANs for localization is not advisable. For *experiment 2*, it was possible to deploy the ANs more near to the corners than *experiment 1*, because at least four ANs will be able to receive the wireless packets which does not have a big impact in the ranging estimation.

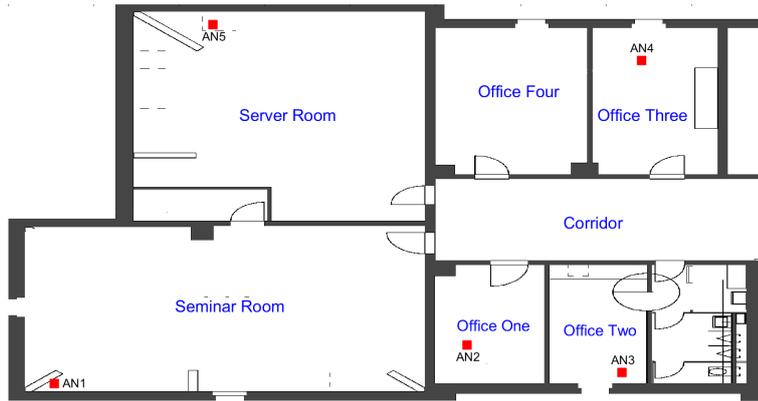


Figure 6.3: Floor plan with five anchor nodes (ANs) for experiment 2

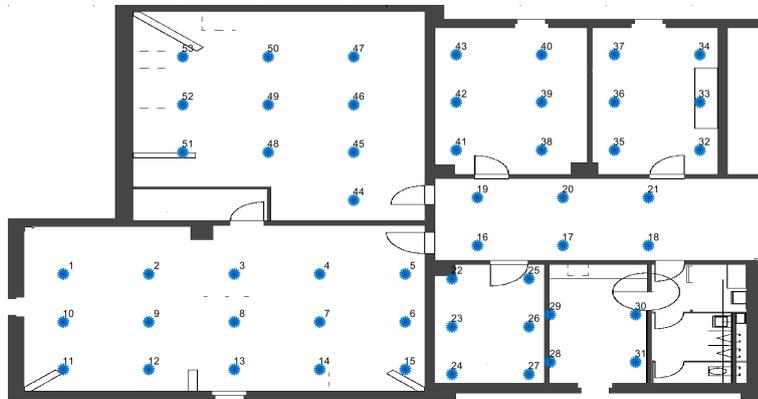


Figure 6.4: Floor plan with 53 stationary positions

6.1.1 Stationary Target Set-up

The first step for the evaluation of the experiments is to take some initial measurements aimed to train the environmental parameters α and β required for the non-linear regression (NLR) model (see chapter 4). For both experiments, 53 positions spread over the whole interesting area were conducted as shown in figure 6.4. For collecting this information a laptop was used, which was stationary at each of the 53 positions and was configured to send 300 packets, such that each AN could have a reasonable number of packets for the training.

Accurate ranging is a preliminary step for accurate range-based localization. In both experiments 20 positions uniformly distributed over the map were chosen from the 53 available (see figure 6.4) for the training as is illustrated in figure 6.5. Based on the trust region algorithm for the NLR model and initial measurements, α and β were obtained as shown in tables 6.1 and 6.2.

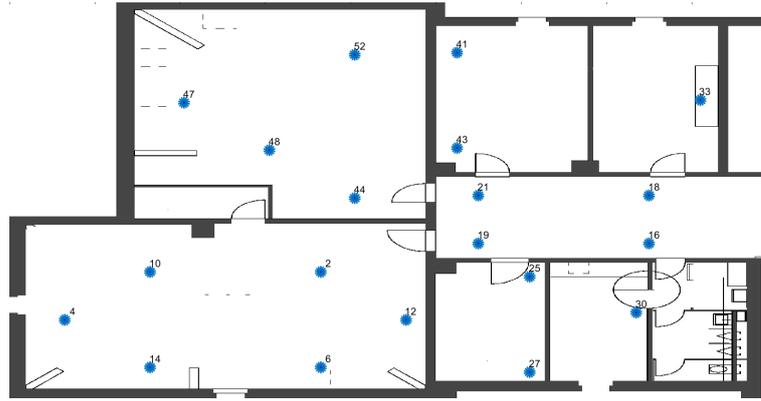


Figure 6.5: Floor plan with 20 stationary training positions

Table 6.1: Environmental parameters for the NLR Model for experiment one

	AN1	AN2	AN3	AN4
α	13.22	8.302	9.959	10.26
β	-0.03819	-0.04356	-0.04672	-0.0434

Table 6.2: Environmental parameters for the NLR Model for experiment two

	AN1	AN2	AN3	AN4	AN5
α	14.08	8.619	11.47	11.3	13.16
β	-0.04799	-0.05237	-0.04786	-0.04708	-0.0398

6.1.2 Mobile Target Set-up

Once the static measurements were ready, three tracking experiments were undertaken to analyse the performance of the system for the mobile target, in which a smartphone was held by a person following the traces illustrated in figures 6.6, 6.7 and 6.8. There, it can be seen that the starting point for all the tracking experiments was in the seminar room. The track that ended in office three will be defined as *tracking 1*, the track ended in office two will be defined as *tracking 2* and the track ended in the server room will be defined as *tracking 3*.

There were some adjustments made into the environment during the tracking experiments for further calculation of the physical variables associated to each tracking experiment, which consists of markers (see figures 6.9, 6.10, 6.11 and 6.12) to identify each step. In addition, an application that emits constant ticks was used in another smartphone placed in the pocket, where the main idea was to give a step, every time a beep was heard in an attempt of walking in a constant speed. Additionally, the application that is running on the mobile node to broadcast wireless packets also registers the timestamp from the beginning to the end of the experiments, allowing to have the time taken to complete each track. All these configurations were done

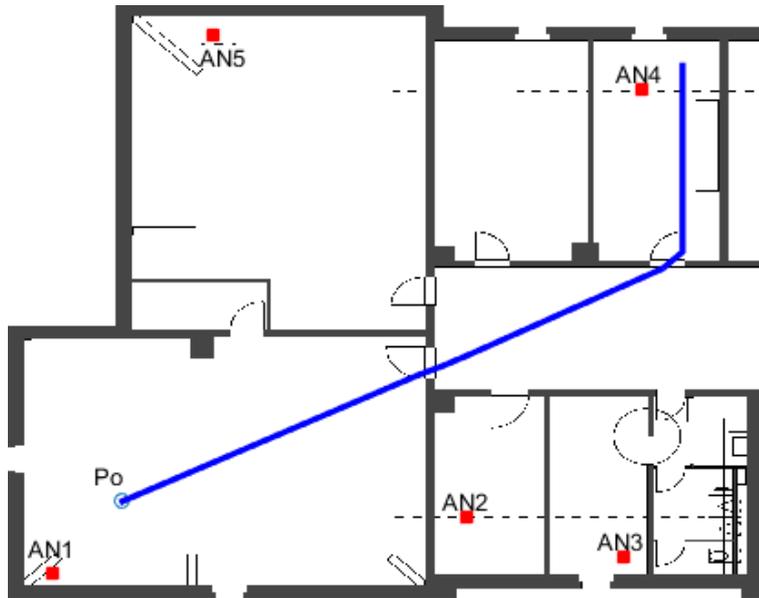


Figure 6.6: Ground truth movement for tracking 1

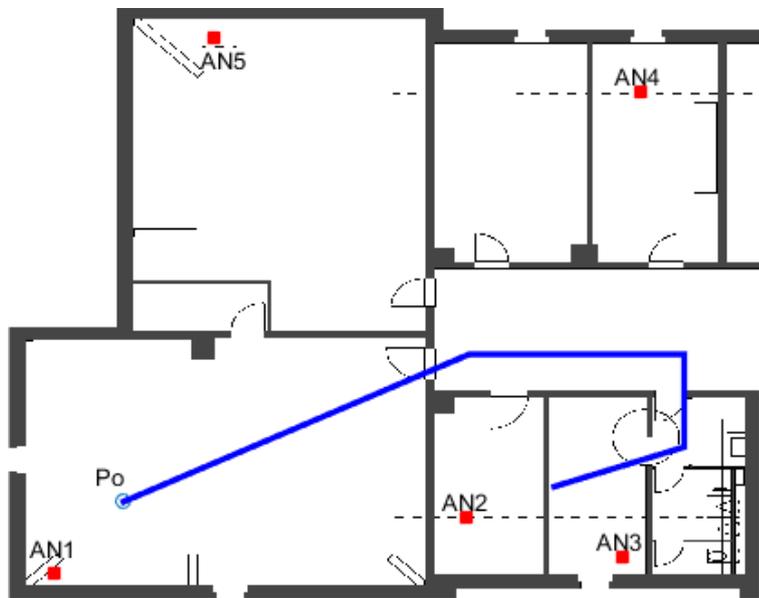


Figure 6.7: Ground truth movement for tracking 2

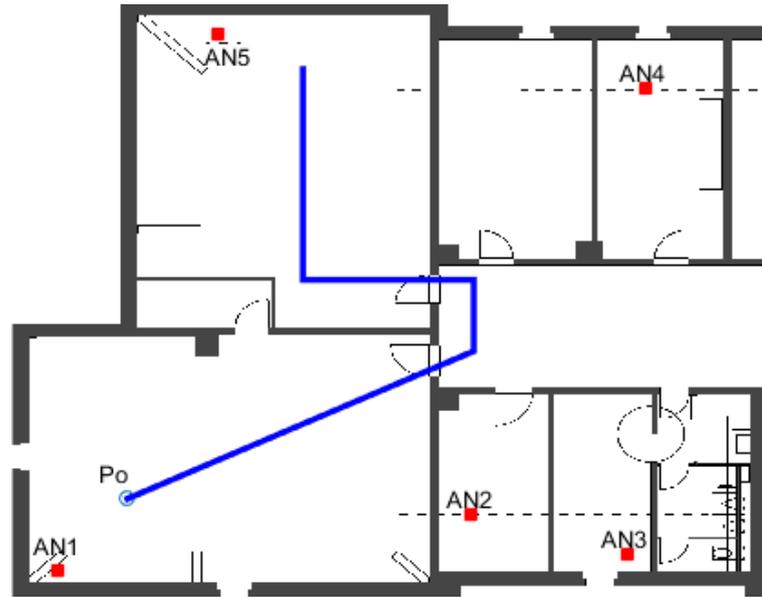


Figure 6.8: Ground truth movement for tracking 3

Table 6.3: Ground truth values experiment 1

Experiment 1			
Trace	Time (s)	Distance (m)	Speed (m/s)
Tracking one	19.618	18.2	0.928
Tracking two	19.618	18.9	0.944
Tracking three	19.292	18.2	0.943

focused on obtaining the more approximate ground truth values for time, distance, angles and velocity as well.

The markers for *experiment 1* were placed on the floor each 0.7 m, whereas the markers in *experiment 2* were placed each 0.5 m, the results of the calculations are assessed in table 6.3 and 6.4. In practice, the effect of these markers distribution gave each experiment a different walking speed, where the person holding the mobile node in *experiment 1* was forced to walk using a fast style, whereas for *experiment 2* the person was able to move in a more paced form.

6.2 Results Analysis

For the evaluation of the tracking experiments, the localization errors for the respective positions that follows each trajectory are calculated. The CDFs for each round of experiments, as well as tables that summarizes the mean, standard deviation and maximal values of the localization errors are presented. The terminology used for each measurement is described below:



Figure 6.9: Markers for tracking in corridor



Figure 6.10: Markers for tracking in office
two



Figure 6.11: Markers for tracking in office three

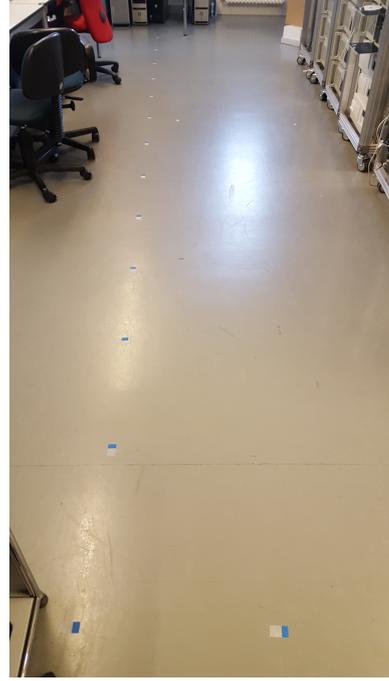


Figure 6.12: Markers for tracking in server room

Table 6.4: Ground truth values experiment 2

Experiment 2			
Trace	Time (s)	Distance (m)	Speed (m/s)
Tracking one	30.874	20	0.648
Tracking two	30.874	20	0.653
Tracking three	32.428	21	0.648

- **PDR:** Measurements for localization estimation based only on PDR component.
- **PF-CSI:** Measurements for localization estimation with particle filter using one ranging observation, which was computed through CSI by averaging the power from the 3 antennas.
- **PF-Fusion:** Measurements for localization estimation with particle filter fusing one ranging observation from CSI and the velocity observation from PDR.
- **PF-Fusion 3 Antennas:** Measurements for localization estimation with particle filter fusing three ranging observations from CSI and the velocity observation from PDR.

Considering that this set of measurements will be referred later on, the acronym PF-CSI will be used as ranging particle filter, whereas the acronym PF-Fusion will be used as fusion particle filter.

6.2.1 Comparative Analysis

As depicted in figure 6.13, the results from *experiment 1* and *experiment 2*, have some different performances, in addition the use of one more AN in *experiment 2* should reflect some improvement tendency. Thus, it is worth to make an analysis focus on the comparison between them. Table 6.5 summarizes the relevant values for each of the algorithms implemented considering the use of ANs in the particle filter approaches for deeper analysis.

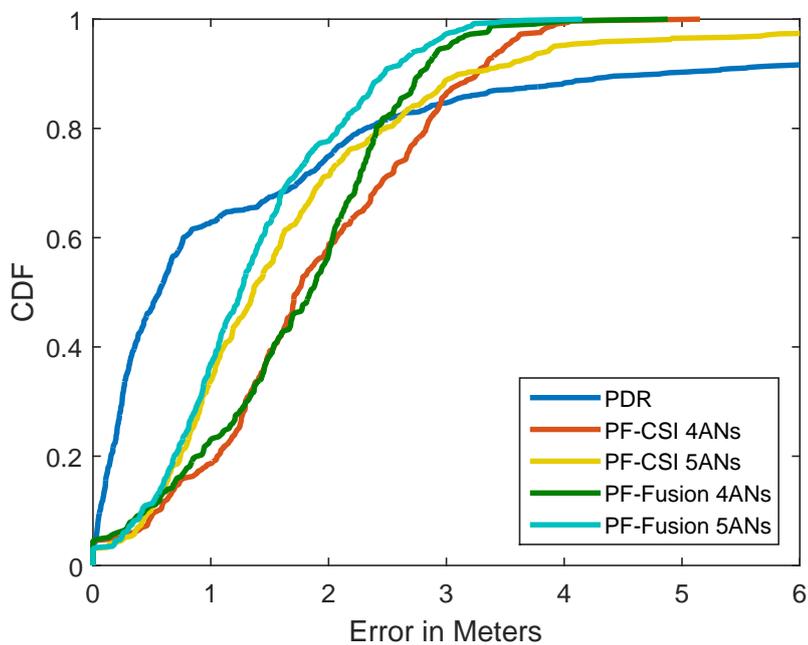


Figure 6.13: Localization Error for Tracking Experiments

Table 6.5: Localization Errors for Tracking Experiments

Approach	Mean	Standard Deviation	Maximum Error
PDR	1.7m	2.7m	12.2m
PF-CSI 4ANs	1.9m	1m	5.2m
PF-CSI 5ANs	1.7m	1.5m	11.6m
PF-Fusion 4ANs	1.7m	0.9m	4.9m
PF-Fusion 5ANs	1.4m	0.8m	4.2m

Analysing the results from both experiments, PDR has in average very good results for at least 60% of the whole trajectories. One would expect under those results that PDR outperforms particle filter, but the statistical values reveal that this does not necessary hold, because particle filter approaches in average are more accurate and stable than PDR. The main reason for such results is that PDR has considerable higher maximum errors than particle filter approaches as shown in table 6.5.

Additionally, these results demonstrate that fusion particle filter is an approach that can exploit the advantages from ranging particle filter and PDR as well. The experiment with 4ANs shows that there are certain traces where ranging particle filter could be used alone, but in other traces it would be advisable to fuse with some additional information, like the velocity from PDR. Hence, in order to have stable and accurate results, using fusion particle filter would be the best option. It is worth to mention that particle filter can be used to address the problem of error accumulation for PDR by fusing additional observations.

In general, both session experiments reflect the expected results. First fusion particle filter is the most accurate and stable approach, where fusion particle filter (5ANs) with a mean error of $1.4m$ outperforms fusion particle filter (4ANs). These measurements reflect it that a good distribution and training of the ANs have an impact in the tracking accuracy.

Results from *experiment 1* and *experiment 2* reveals that PDR is not a reliable solution for tracking systems. Therefore, it is highly advisable to use PDR along with particle filter to obtain the best performance as well as a reliable solution.

6.2.2 Fusion Particle Filter Performance

The general performance evaluation considering a CDF plot 6.14 for the fusion particle filter approaches including the algorithm with 3 antennas is presented in detail in table 6.6. It is possible to identify that fusion particle filter outperforms fusion particle filter with 3 antennas in *experiment 1* and *experiment 2* as well. This reveals that by adding the ranging information from each individual antenna does not make a considerable impact on the performance.

The reason for these results is that some antennas with very bad measurements influenced negatively the tracking accuracy. In contrast, fusion particle filter by averaging the values from the antennas is able to mitigate the bad measurements. Therefore, using the average information of the antennas is enough to achieve a good performance for fusion particle filter. This implies that using diversity in the observation for particle filter is advisable to achieve better accuracy.

Table 6.6: Localization Errors for Fusion Particle Filter Experiments

Approach	Mean	Standard Deviation	Maximum Error
PF-Fusion 4ANs	$1.7m$	$0.9m$	$4.9m$
PF-Fusion 4ANs 3 Antennas	$2m$	$1.1m$	$5.6m$
PF-Fusion 5ANs	$1.4m$	$0.8m$	$4.2m$
PF-Fusion 5ANs 3 Antennas	$1.7m$	$0.8m$	$5.9m$

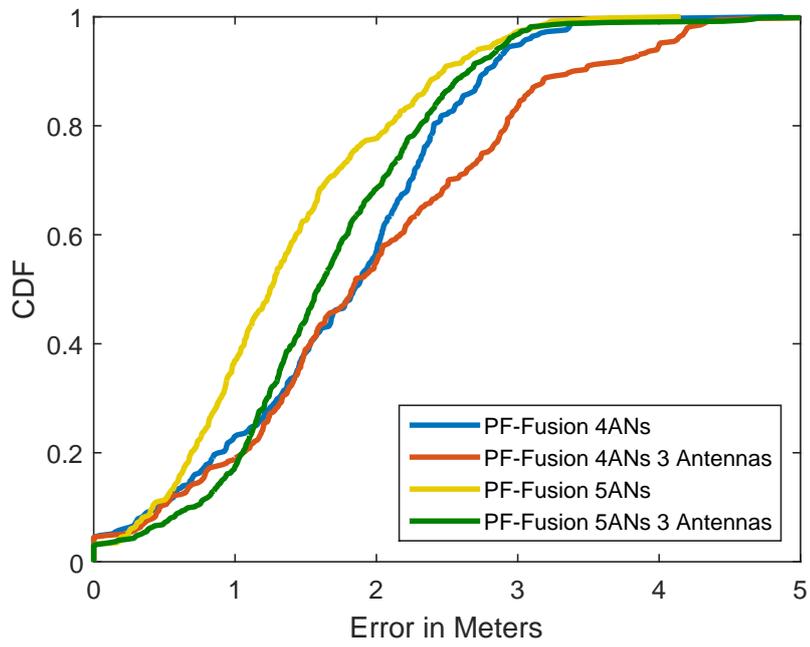


Figure 6.14: Localization Error for Fusion Particle Filter Experiments

Chapter 7

Conclusions

Due to the lack of GPS signals for indoor environments, different approaches for indoor positioning and tracking has attracted lots of research interest. Radio-based positioning techniques and pedestrian dead reckoning are two typical methods to track mobile targets. However, they have different advantages and disadvantages. For PDR, it can estimate the moving velocity but the tracking accuracy is prone to accumulated errors. Radio-based positioning techniques is robust to accumulated error but lack of velocity estimation. Therefore, in this thesis, particle filter is adopted to fuse the radio and inertial sensors information to provide a tracking method with high accuracy and reliability.

To estimate the motion model of the mobile target, the accelerometer is adopted to estimate the moving speed and a compass to estimate the moving orientation. To achieve high accuracy of velocity estimation, a low pass filter is applied on the raw data. In addition, it is important to take into account mechanisms to identify noise and drift reduction of the raw data.

This thesis proposed a particle filter approach in which ranging information from CIR and velocity information from inertial sensors are two observation inputs. These two informations are fused by multiplying their likelihoods together. To achieve the high accuracy of ranging, some enhanced ranging methods are adopted[17].

Based on the proposed particle filter, a set of experiments were conducted to evaluate its performance. The results of the experiments are in line with the theory and this thesis proposal, where particle filter combining CIR-based ranging and the velocity from PDR outperforms the other approaches. Upon these results, it is possible to make the following conclusions:

- Diverse data fusion is the key for particle filter, because it has been shown in chapter 6 that fusion particle filter outperforms PDR and CSI-based positioning. Particle filter does not appear to improve the result as in the case of fusing the power information from three antennas.
- The experimental results have shown that the proposed fusion particle filter algorithm, achieves a reasonable result for an indoor tracking system with a mean error of $1.4m$.
- PDR is not a reliable approach for tracking systems, because it is prone to accumulated

errors. Therefore, it should be used along with a Bayesian filter to achieve a more stable solution.

- CIR-based positioning lacks of movement information, i.e., velocity, which makes its performance low with respect to PDR and fusion particle filter.
- The distribution of ANs also impacts that accuracy. By using 5 ANs correctly distributed across the environment, the CIR-based positioning performance is slightly increased.

Bibliography

- [1] Primer, Wi-Fi: Overview of the 802.11 Physical Layer and Transmitter Measurements, Available in www.tektronix.com/wifi
- [2] 802.11n-2009 - IEEE Standard for Information technology- Local and metropolitan area networks- Specific requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput, Available in <https://standards.ieee.org/findstds/standard/802.11n-2009.html>
- [3] Y. Shen and E. Martinez, "Channel Estimation in OFDM Systems", in Freescale Semiconductor, Rev. 0, 1/2006
- [4] Z. Yang, Z. Zhou, Y. Liu, "From RSSI to CSI: Indoor Localization via Channel Response", in ACM Computing Surveys, Vol. 46, No. 2, Article 25, Publication date: November 2013.
- [5] D. Halperin, W. Huy, A. Shethz, and D. Wetherall, "Tool Release: Gathering 802.11n Traces with Channel State Information", in ACM SIGCOMM Computer Communication Review, Volume 41, Number 1, January 2011
- [6] N. Kakiuchi and S. Kamijo, "Pedestrian Dead Reckoning for Mobile Phones through Walking and Running Mode Recognition", in Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, October 6-9, 2013.
- [7] A. Mariakakis, S. Sen, J. Lee and K. Kim, "SAIL: Single Access Point-Based Indoor Localization", in MobiSys'14, June 16-19, 2014, Bretton Woods, New Hampshire, USA.
- [8] N. Kothari, B. Kannan and M. B. Dias, "Robust Indoor Localization on a Commercial Smart-Phone", The Robotics Institute, Carnegie-Mellon University, August 2011.
- [9] H. Weinberg, "Using the ADXL202 in Pedometer and Personal Navigation Applications", Analog Devices, Inc AN-602 Application Note, 2002.
- [10] F. Hong, H. Chu, L. Wang, Y. Feng and Z. Guo, "Pocket Mattering: Indoor Pedestrian Tracking with Commercial Smartphone", in International Conference on Indoor Positioning and Indoor Navigation, 13-15th November 2012

- [11] J. Borenstein and L. Ojeda, "Heuristic Drift Elimination for Personnel Tracking Systems", in *The Journal of Navigation* 2010.
- [12] A. A. Ali, V. D. Nguyen, K. Kyamakya and A.S. Omar, "Estimation of the Channel-Impulse-Response Length for Adaptive OFDM Systems Based on Information Theoretic Criteria", in University of Magdeburg, International University Bremen, University of Klagenfurt
- [13] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "FILA: Fine-grained Indoor Localization", in *INFOCOM, 2012 Proceedings IEEE*
- [14] Z. Jiang, J. Zhao, X. Liy, W. Xi, K. Zhao, S. Tangz, J. Han, "Communicating Is Crowdsourcing: Wi-Fi Indoor Localization with CSI-based Speed Estimation", in School of Electronic and Information Engineering, China, Department of Computer Science, Chicago, Department of Computer and Information Science, Philadelphia, July 2013.
- [15] Z. Zhou, Z. Yang, C. Wu, W. Sun, and Y. Liu, "Lifi: Line-of-sight identification with wifi," in *INFOCOM, 2014 Proceedings IEEE*, April 2014
- [16] D. Halperin, W. Hu, A. Shethy, and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements", in *SIGCOMM'10*, September 2010, New Delhi, India.
- [17] Z. Li, T. Braun, D. C. Dimitrova, "A Passive WiFi Source Localization System based on Fine-grained Power-based Trilateration", in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2015.
- [18] H. Wang, H. Lenz, A. Szabo, J. Bamberger, U. D. Hanebeck, "WLAN-Based Pedestrian Tracking Using Particle Filters and Low-Cost MEMS Sensors", in *Corporate Technology, Information and Communications, CT IC 4*, Germany and *Intelligent Sensor-Actuator-Systems Laboratory*, Institute of Computer Science and Engineering, Universitat Karlsruhe (TH), Germany, December 9, 2007
- [19] F. Hong, Y. Zhang, Z. Zhang, M. Wei, Y. Feng and Z. Guo, "WaP: Indoor Localization and Tracking Using WiFi-Assisted Particle Filter", *39th Annual IEEE Conference on Local Computer Networks, LCN 2014*, Edmonton, Canada.
- [20] R. B. S. Richard H. Byrd and G. A. Shultz, "A trust region algorithm for nonlinearly constrained optimization", in *SIAM Journal on Numerical Analysis*, vol. 24, 1987, pp. 1152-1170.
- [21] M. Andrejasic, I. Poberaj "MEMS Accelerometers", University of Ljubljana, Faculty for mathematics and physics, Marec 2008.
- [22] M. H. Afzal, V. Renaudin and G. Lachapelle, "Use of Earth's Magnetic Field for Mitigating Gyroscope Errors Regardless of Magnetic Perturbation", in *Sensors 2011*, ISSN 1424-8220, www.mdpi.com/journal/sensors

- [23] M. H. Afzal, "Use of Earth's Magnetic Field for Pedestrian Navigation", Department of Geomatics Engineering, in UCGE Reports, July 2011
- [24] Y. Yuan, "A review of trust region algorithms for optimization", State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences.
- [25] A. Doucet, N. Freitas, N. Gordon, "Statistics for Engineering and Information Science", Sequential Monte Carlo Methods in Practice.
- [26] Wilson, Edwin Bidwell (1901). Vector analysis: a text-book for the use of students of mathematics and physics, founded upon the lectures of J. Willard Gibbs. p. 125. This is the likely origin of the speed/velocity terminology in vector physics.
- [27] A. Blanco M. Eissler, "One firmware to monitor 'em all.", in Core Security Technologies, August 2012.
- [28] T. Ozyagcilar, "Implementing a Tilt-Compensated eCompass using Accelerometer and Magnetometer Sensors", in Freescale Semiconductor, Rev. 3, 01/2012.
- [29] T. Ozyagcilar, "Calibrating an eCompass in the Presence of Hard and Soft-Iron Interference", in Freescale Semiconductor, Rev. 3, 04/2013.
- [30] MEMS-based accelerometers [Online]. Available: http://www.wikid.eu/index.php/MEMS-based_accelerometers
- [31] Magnetic reference field models [Online]. Available : http://www.geomag.nrcan.gc.ca/mag_fld/magref-en.php
- [32] Linux 802.11n CSI Tool [Online]. Available: <http://dhalperi.github.io/linux-80211n-csitool/>
- [33] FAQ CSI Tool [Online]. Available: <http://dhalperi.github.io/linux-80211n-csitool/faq.html>
- [34] CSI Tool Installation [Online]. Available: http://dhalperi.github.io/linux-80211n-csitool/old_installation.html