SAPIENZA
UNIVERSITÀ DI ROMA

FACOLTÀ DI INGEGNERIA
DELL'INFORMAZIONE, INFORMATICA E
STATISTICA

Corso di Laurea Magistrale in
Ingegneria delle Comunicazioni

TOWARDS ROBUST INDOOR LOCALISATION BY COMBINATION OF MEASUREMENTS

Relatore
Chiar.ma Prof.ssa
Maria-Gabriella Di Benedetto

Laureando
Giulio Giaconi
Matricola 1225338

Correlatore
Ing. Pietro Obino
Telecom Italia S.p.A.

ANNO ACCADEMICO 2012/2013

To my mother,

who has always encouraged me to do my best

and supported me in every occasion.

Acknowledgements

First of all, I would like to thank my supervisors Prof. Maria-Gabriella Di Benedetto and Ing. Pietro Obino for their constant help, patience, trust and support in writing this thesis.

Prof. Di Benedetto, in particular, offered me last summer the opportunity to attend the Cost Action IC0902 Summer School on Cognitive Wireless Communications at King's College of London and the possibility to visit the Imperial College of London, within the framework of the "Excellence Track". These experiences represented for me a unique way of growing from an academic, professional and human point of view. In addition, I would like to thank the Cost Action IC0902 for the Grant and the support they gave me for my stay in London.

A special thank also to Ing. Pietro Obino and many others in Telecom Italia, who made me feel at home and helped me for every problem without a second thought.

At Imperial College I met Dr. Deniz Gündüz, whom I want to thank for his careful supervision during my Short Term Scientific Mission. I really appreciated a lot my time and work there, especially thanks to Dr. Gündüz, who has always been extremely kind and helpful with me.

I am especially grateful to my family for all the unconditional support and guide I have been given since the very beginning. If I have achieved my goal, it is above all thanks to them.

Thank you to all my friends and all those at the university who shared with me troubles and worries during these five years. A warm thank to Nicola, who has always been very supportive and friendly with me.

Last but not least, thank you, Piera, for having always been by my side.

Rome, October 22, 2013

Giulio Giaconi

Abstract

High accuracy wireless indoor localisation is nowadays a very popular research area due to the almost infinite possible applications. Technologies which are already broadly deployed are very attractive for implementing localisation services, because they have already an existing infrastructure and could therefore reduce costs and time for installation and deployment of such services. For this reason, the technology considered in this work is the IEEE 802.11 WLAN. However, by using only Wi-Fi power measurements, it is not possible to obtain a very accurate system. Therefore, since this algorithm is mainly conceived for modern mobile devices, which are typically embedded with many sensors, also information from them is considered. The purpose of this thesis is thus to define an indoor localisation system which is capable of exploiting different information sources to improve the localisation estimation. Information from Wi-Fi Received Signal Strength Indicator (RSSI) and inertial sensors like accelerometer, gyroscope and magnetometer is considered and integrated by a particle filter. First, a neural network is used to match the RSSI fingerprints received during the online phase with those ones collected in the fingerprinting database during the offline phase, thus giving a first estimate of the position. Secondly, a step detection algorithm is implemented to properly estimate users' movements by exploiting information from inertial sensors and returning a second estimate. Finally, a particle filter integrates the two estimates to obtain the final position estimation. In addition, another extremely promising technology is considered, the Visible Light Communication (VLC). Thanks to their properties, it is universally recognized that LED lamps will replace the traditional ones in the future and this will enable to take advantage of them even for communication and localisation purposes. As part of the Thesis, I have carried out an internship at Telecom Italia S.p.A. and a Short-Term Scientific Mission at Imperial College of Science, Technology and Medicine of London, United Kingdom. Moreover, a prototype of this algorithm will be developed by Telecom Italia S.p.A.

Following an introduction on indoor localisation in Chapter 1, Chapter 2 shows practically that RSSI methods are inaccurate for localisation purposes, by using an indoor wireless coverage simulation software. Chapter 3 and 4 focus on the indoor localisation algorithm, while Chapter 5 gives an overview on VLC indoor localisation. Finally, in Chapter 6 the conclusions and some possible future improvements are drawn.

La localizzazione indoor costituisce oggigiorno un'area di ricerca molto importante a causa della miriade di possibili applicazioni che ne potrebbero derivare. Le tecnologie che risultano essere già ampiamente diffuse sono da preferirsi per l'implementazione di sistemi di localizzazione, dal momento che è possibile usarne l'infrastruttura già esistente, riducendo pertanto costi e tempi di installazione dei servizi. Questo è il motivo per cui il sistema sviluppato in questo lavoro si basa principalmente sullo standard IEEE 802.11 WLAN. Tuttavia, avvalendosi solamente di misure di potenza del segnale Wi-Fi non è possibile ottenere un sistema molto preciso. Pertanto, dato che questo algoritmo è pensato principalmente per moderni dispositivi mobili, i quali dispongono di numerosi sensori integrati, anche l'informazione da essi ricavata è considerata. L'obiettivo di questa tesi è perciò quello di definire un sistema per la localizzazione indoor che sia in grado di avvalersi di diverse fonti di informazione per migliorare la stima della posizione. Le informazioni relative alla potenza ricevuta del segnale Wi-Fi (RSSI) e le informazioni ricavate dai sensori inerziali come accelerometro, giroscopio e magnetometro sono raccolte e integrate mediante un filtro particellare. Innanzitutto, una rete neurale è utilizzata per confrontare i fingerprints RSSI ricevuti durante la fase online con quelli salvati all'interno del fingerprinting database durante la fase offline, ottenendo una prima stima della posizione. In seguito, un algoritmo di step detection è implementato per stimare il movimento degli utenti, grazie all'uso dell'informazione proveniente dai sensori inerziali, producendo pertanto una seconda stima. Infine, un filtro particellare integra le due informazioni per ottenere la stima finale della posizione. Inoltre viene analizzata anche un'altra tecnologia molto promettente, la Visible Light Communication (VLC). Grazie alle loro proprietà, è universalmente riconosciuto che le lampade LED rimpiazzeranno in futuro quelle tradizionali e si può pertanto pensare di sfruttare anche questa tecnologia per fini di comunicazione e localizzazione. Nell'ambito della tesi, ho svolto un tirocinio presso Telecom Italia S.p.A. e un periodo di ricerca presso l'Imperial College of Science, Technology and Medicine di Londra, Regno Unito nell'ambito

della Cost Action IC0902. Inoltre, un prototipo di questo algoritmo verrà sviluppato da Telecom Italia S.p.A.

Dopo una breve introduzione e un inquadramento del problema della localizzazione indoor nel Capitolo 1, il Capitolo 2 mostra, grazie all'uso di un software per la simulazione della copertura Wi-Fi indoor, che i metodi basati sul segnale di potenza Wi-FI sono imprecisi per i fini di localizzazione. I Capitoli 3 e 4 si concentrano sull'algoritmo per la localizzazione indoor, mentre il Capitolo 5 fornisce una panoramica sulla localizzazione mediante VLC. Le conclusioni e i possibili sviluppi futuri sono infine presentati nel Capitolo 6.

List of Figures

List of Tables

Contents

Chapter 1

Introduction to Indoor Localisation

Localisation is the definition of an object's position according a specific coordinate system. High-accuracy indoor localisation is a necessary requirement for a number of location-based applications including logistic, security services, tracking, communication and military systems. Moreover, the advent of ubiquitous and simple localisation techniques will open to new applications in the future, like fraud detection, museum touring, augmented reality.

The Global Positioning System (GPS) is the most used and popular navigation system but its efficiency is highly limited in indoor environments since it relies on electromagnetic signal propagation. The satellites transmit ranging signals on two frequencies, 1575.42 MHz and 1227.6 MHz. At these frequencies, microwave signals attenuate and scatter while passing through ceilings and walls and also multipath occurs. For these reasons GPS cannot be used for positioning in indoor environments.

Therefore in the last years several indoor localisation strategies based on different technologies have been developed, including IEEE 802.11 Wireless Local Area Network (WLAN), IEEE 802.15.1 Bluetooth (BT), Infrared (IR), Radio-Frequency Identification (RFID), Ultra-Wide Band (UWB), Universal Mobile

Telecommunications System (UMTS) (25, 34, 35, 38, 45). However, differently from outdoor localisation, indoor positioning is extremely challenging due to the complex propagation environment and unpredictable time varying conditions, such as the random presence of people, equipment or furniture, which cannot be predicted and taken into consideration by the models. This is the reason why the proposed methods are generally not so much accurate and research is still ongoing.

In this introductory chapter it is explained how the most important current indoor localisation systems work, outlining advantages and disadvantages of each one.

1.1 Localisation

The purpose of localisation is to find someone's or something's position. The first step in a localisation system is to define the coordinate system, also called *frame*, to which all the measurements are referred. An item is then associated to a set of three coordinates, x, y and z which represents its position in space without ambiguity.

1.1.1 Coordinate Systems

There are three main three-dimensional coordinate systems or frames: the world frame, the local frame or the body frame.

1.1.1.1 World Frame

This system represents the absolute position in the world. This coordinate system is important because it is absolute and therefore it can be used to represent the position of every object in the world without ambiguity. To give an example of such a system, the following definition is reported (57):

- x_w is defined as the vector product $y_w \times z_w$ (it is tangential to the ground at the device's current location and roughly points east).;
- y_w is tangential to the ground at the device's current location and points towards the magnetic North Pole;
- z_w points towards the sky and is perpendicular to the ground.

The following Figure 1.1 represents the previous definition.

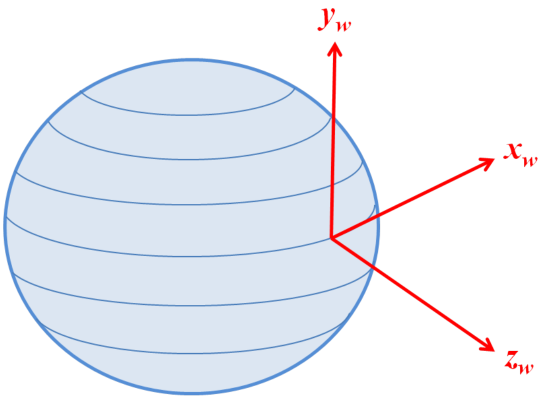


Figure 1.1: World frame defining x, y and z axes.

In the context of an indoor localisation system, the world frame would be quite large, useless and maybe even difficult to manage with. Thus the coordinate system considered in this work is the local frame.

1.1.1.2 Local Frame

There is a more convenient reference system to adopt in the limited context of indoor localisation: the local frame. In this system, the frame is fixed with respect to the building or the map in which navigation is considered. This usually does not correspond to the global frame and in general there will be a certain angle offset θ between the two systems. To understand why it is easier to manage with such a system, let us consider a building with walls perpendicular to each other, and which directions are 30 degrees offset from global frame axes. A map of the building might result much more clear to a user if the walls are drawn perfectly horizontally and vertically instead of pointing in their actual world frame direction. The following definitions apply:

- x_l: an axis in the same plane as x_w, offset by an angle θ.
- y_l: an axis in the same plane as y_w, offset by an angle θ.
- z_l: an axis perpendicular to the ground, equal to z_w.

Figure 1.2 represents the local frame with respect to a map.

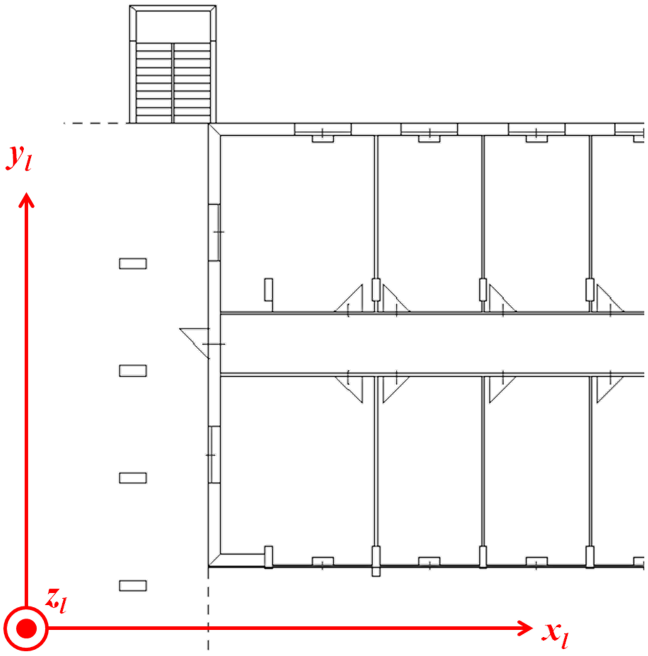


Figure 1.2: Local frame defining x, y and z axes.

1.1.1.3 Body Frame

The body frame is the system referred to the handheld device. The measurements of the sensors are all referred to this reference system, which is fixed to the mobile device and therefore changes continuously in orientation with respect to the world frame or to the local frame. The axes are defined in the Android Sensor API as follows (63):

- x_b: horizontal axis from the left to the right to the device.
- y_b: vertical axis from the bottom to the top of the device.
- z_b: axis from the backside to the front side of the device.

This frame is used by:

- the acceleration sensor,
- the gravity sensor,
- the gyroscope,
- the linear acceleration sensor,
- the geomagnetic field sensor.

This frame is shown in Figure 1.3.

As explained later, one of the most important issues in the proposed algorithm is how to convert measurements from this frame to the local one. The conversion can be described in terms of the rotation needed to align the local frame with the body frame. Rotations can be described for examples by Euler angles, which are a set of three angles that specify the rotations performed around three pre-defined axes. Euler angles are simple to understand, but have several limitations. A better but much more complicated system for representing rotations is the usage of quaternions. These concepts will be explained better in Chapter 3.

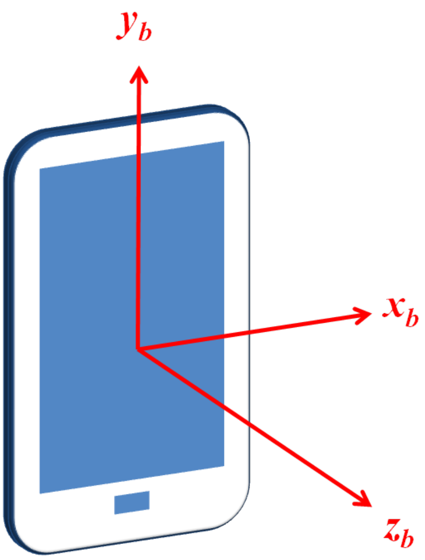


Figure 1.3: Body frame defining x, y and z axes.

1.2 Anchor Free and Anchor Based Scenarios

First of all two main schemes, the *anchor-based* method and the *anchor-free* method have to be taken into account. The first one considers the presence of the so-called *anchors*, which are network nodes whose position is known a priori. The other nodes in the network use anchors' position in order to locate themselves. In an anchor-free approach instead, nobody knows its coordinates and therefore the elements only use relative coordinates. An anchor-free system is extremely easy to develop and to implement because there are no conditions in the nodes' disposal and they can be deployed in random positions. However, the position provided is relative and it is valid only inside the network. The anchor-based system, on the contrary, provides absolute coordinates and therefore nodes' position can be used outside the system itself for other applications. However, this system implies a careful planning of anchors' positions and the nodes have to be sufficiently close to the anchors to be located. In the literature both of the approaches have been widely developed.

1.3 Localisation Phases

As reported in (7), a localisation process is made up of three main tasks:

1. ***ranging***,
2. ***positioning***,
3. ***refinement***.

First, one has to define a method to estimate the distance: this phase is called *ranging*. Once the distances are properly estimated, it is possible to infer one's position through appropriate positioning algorithms in the *positioning* phase. Finally, the position is refined recursively through the *refinement* phase.

Range free approaches, which are independent of the Euclidean distance, have been developed too. In these approaches the distance is not directly estimated and therefore the position of the node is approximated by other means. For example, in the *centroid method* node's position is approximated from the nearest anchors' positions, in the *DV-Hop* and *Amorphius* methods the distance is expressed as number of hops between different nodes, in the *Point in Triangulation - PiT* and *Approximate Point in Triangulation - APiT* methods, node's position is assumed to be the barycentre of the triangle formed by the three nearest nodes.

1.3.1 Ranging

Supposing the transmitting signals are known, the ranging methods are based on the analysis of the received signal which is modified somehow as an effect of the distance it has travelled. Different ranging methods have been developed in the literature and the most important are:

- time-based ranging
  - Time of Arrival (ToA),
  - Time Difference of Arrival (TDoA)
- direction-based ranging
  - Angle of Arrival (AoA)
- Received Signal Strength Indicator (RSSI)
- carrier phase

1.3.1.1 Time-Based Ranging

The distance estimation is done by measuring the signal propagation time. The hypothesis is that the signal propagation velocity is equal to the light velocity ($c \approx 3 \cdot 10^8 m/s$). These methods suffer from the presence of multipath effect, which unfortunately affects a lot indoor environments. Another practical issue in the deployment of such systems is the need of a precise clock synchronization of the different nodes.

1.3.1.2 Time of Arrival (ToA)

The propagation time is measured as the difference between receiving and transmitting time. From the propagation time T_f, the distance d is estimated through Equation 1.1:

$$T_f = \frac{d}{c} \tag{1.1}$$

where c is the light propagation velocity. There are two ways of evaluating distance: the *One-Way TOA* and the *Two-Way TOA*.

1. **One-Way TOA Ranging**. In the one-way TOA the first node (node A) transmits its signal at time t_1. The transmitted signal arrives at node B at time t_2. It is important that the nodes are perfectly synchronized regarding the oscillators which define the system clock. An even small time error can cause huge distance errors happen because of the extremely high transmitting velocity. In the first method, T_f is then equal to:

$$T_f = t_2 - t_1 \tag{1.2}$$

As a result, the receiving node has to know the instant at which the signal has been sent from the receiver and both transmitter and receiver have to be synchronized. The operating scenario is represented in Figure 1.4

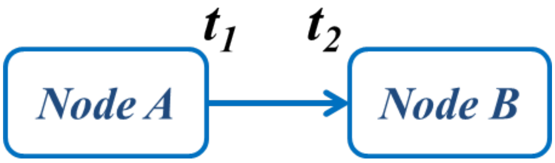


Figure 1.4: Time of Arrival one-way (ToA).

2. **Two-Way TOA Ranging**. In the second method the system measures the round trip time. Node A transmits its signal at time t_1 and the signal arrives at node B at time t_2. Then, node B processes the message in fixed and known time T_d and answers node A, sending a message at time t_3. Node A receives the message at time t_4 and therefore it is able to evaluate the distance through the relation 1.3

$$T_f = \frac{(t_2 - t_1) + (t_4 - t_3)}{2} \tag{1.3}$$

This method is applied in every situation where the synchronization of transmitter and receiver is not possible. However relative clock drifts still affects the performances. The following Figure 1.5 shows this approach.

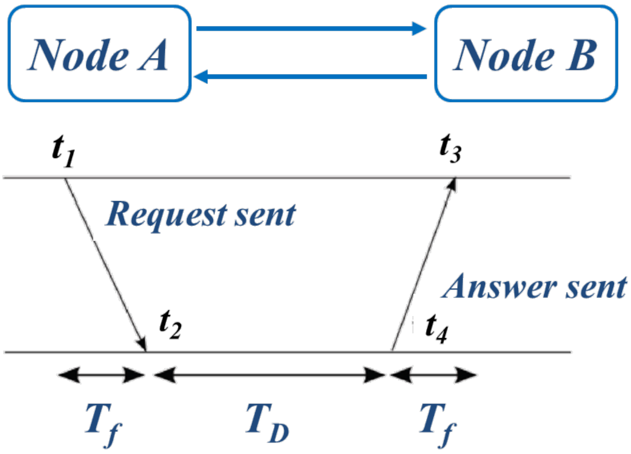


Figure 1.5: Time of Arrival two-way (ToA).

1.3.1.3 Time Difference of Arrival (TDoA)

In this method, node A transmits two different pulses, a radio frequency signal and a ultrasound pulse. Since the ultrasound pulse is slower, the two signals have different velocities and the receiver infers its distance from the transmitter considering the time differences and the different propagation velocities. Figure 1.6 gives an example of this approach.

1.3.1.4 Direction Ranging: Angle of Arrival (AoA)

In this method, the direction origin of the signal is considered. This approach uses directional antennas to estimate the angle of arrival of the received signals. Some disadvantages are interference and obstacles. Figure 1.7 represents this method.

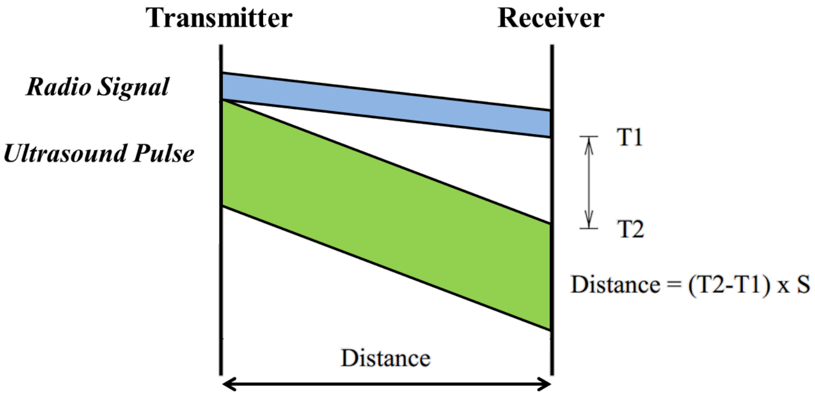


Figure 1.6: Time Difference of Arrival (TDoA).

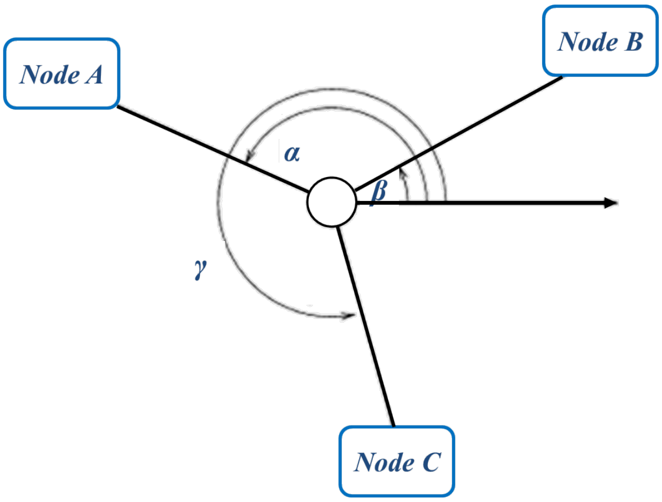


Figure 1.7: Angle of Arrival (AoA).

1.3.1.5 Received Signal Strength Indicator (RSSI)

This approach is one of the most used for WLAN localisation, since it does not require any additional hardware and it exploits information that is easy to recover. In general, it is cheaper and easier to implement a RSSI approach rather than time based methods. During propagation, a signal attenuates and therefore it is possible to compute approximately the distance based on the signal attenuation, once the transmitting power is known. The received power in dBm is given by the Received Signal Strength Indicator (RSSI). Ideally, the relation between the signal attenuation and the distance covered by the signal is given by the Friis Equation 1.4:

$$P_R = \frac{P_T G_T G_R \lambda^2}{(4\pi)^2 R^n} \tag{1.4}$$

where P_R and P_T are the received and transmitted power respectively, $\lambda = c/f$ is the signal wavelength, R is the distance between transmitter and receiver expressed in meters and n the signal propagation constant, which depends on the environment ($n = 2$ in free space scenarios). The received power can then be expressed in dBm through Equation 1.5:

$$P_{dBm} = 10 \log_{10}(10^3 P_{Watt}) \tag{1.5}$$

Therefore, combining Equations 1.4 and 1.5, one can obtain the distance d from transmitter to receiver from the inverse relation. The above equations refers to an ideal scenario, where only free space propagation losses are taken into account. Unfortunately, this is not the case in real conditions, where multipath, fading, noise and signal distortion occur. Other models have been developed to account for some of these aspects. There are basically two families of methods for the electromagnetic signal propagation estimation: the *deterministic* and the *statistical* or *empirical* methods (8). The deterministic methods evaluate the electromagnetic field in every single point of space and they are very accurate: this is the reason why they are very used in the environmental impact evaluation. However, they need a precise knowledge of the environment with its electrical properties, they are suitable just for the considered scenario and they are generally very difficult to develop. The most important methods of this family are:

- Method of Moments (MoM);
- Finite-Difference Time-Domain (FDTD);
- Ray Tracing.

The statistical methods, on the contrary, are very simple methods and they need only a general description of the environment. As a consequence, they produce much less accurate results but they can be applied to different scenarios. They can be divided in indoor and outdoor models. Indoor models evaluate the *path loss*, defined as 1.6:

$$PL_{dB} = 10\log_{10}\frac{P_T}{P_R} = 10\log_{10}\frac{R^2}{G_T G_R}\left(\frac{4\pi}{\lambda}\right)^2 \tag{1.6}$$

where the second relation derives from Equation 1.4 with $n = 2$. Considering a reference distance R_0, Equation 1.6 is multiplied by R_0, obtaining:

$$PL_{dB} = 10\log_{10}\frac{(4\pi)^2 R_0^2}{G_T G_R \lambda^2} + 10\log_{10}\frac{R^2}{R_0^2} \tag{1.7}$$

A generic model uses a parameter γ to express the relation between path loss and distance. This is clear in the following equation:

$$PL_{dB} = PL(R_0) + 10\gamma\log_{10}\frac{R}{R_0} \tag{1.8}$$

where $PL(R_0)$ is the value of PL at a reference point, chosen to be close to the transmitter. Therefore, the purpose of indoor empirical models is to determine the γ coefficient. In addition, also the wall and floor attenuation is considered, adding other two terms to the previous relation:

$$PL_{dB} = PL(R_0) + 10\gamma \log_{10} \frac{R}{R_0} + \sum_{q=1}^{Q} FAF(q) + \sum_{p=1}^{P} WAF(p) \tag{1.9}$$

where $FAF(q)$ and $WAF(p)$ are the Floor Attenuation Factor and Wall Attenuation Factor respectively.

1.3.1.6 Carrier phase

This technique is used also in the GPS system. The satellite transmitter sends the same signals with the same propagation velocity. The receiver knows in advance these signals and measures the number of carrier cycles (therefore it considers the phase difference) which occur during the signal propagation time. Then the transmitter-receiver distance estimation is obtained with an error within a wavelength fraction. The accuracy of such systems is extremely high (order of centimetres). Obviously, they are based on a satellite transmission and therefore this method cannot be used in indoor scenarios.

1.3.2 Positioning

After determining the position, nodes can determine their position. Different algorithms have been developed for this purpose:

- Minimum Maximum (Mini-Max),
- Triangulation,
- Trilateration,
- Multilateration.

1.3.2.1 Minimum Maximum (Mini-Max)

This is a range-based and anchor-based method and it exploits RSSI values. It is very simple and brings to estimates that are very close to the multilateration method. From the RSSI value of a node i, one obtain the distance d_i. Then four straight lines at distance d_i from every anchor are drawn, obtaining a square of side $2d_i$ with the centre represented by node i. This algorithm is applied at least three times to three different nodes to find the intersection between the squares. The centre of the intersection, which is again a square, is the estimated position. The smaller the square is, the higher is the accuracy.

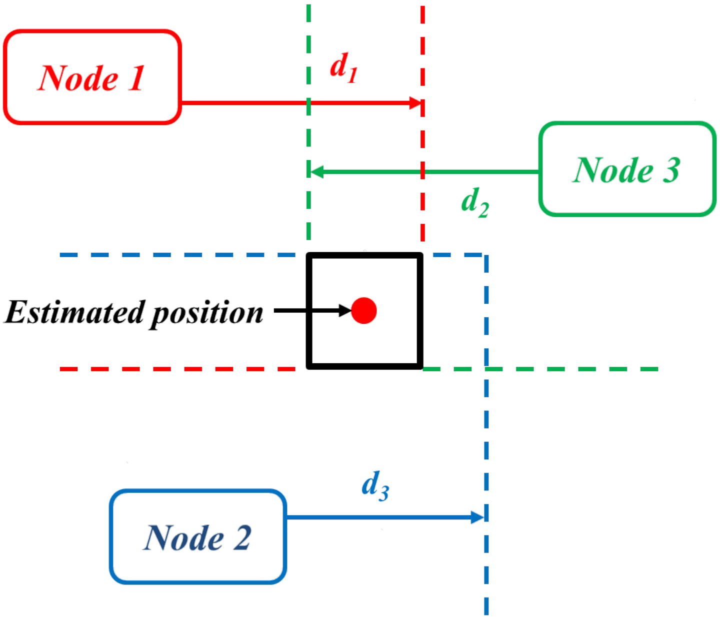


Figure 1.8: Mini-Max.

As shown in Figure 1.9, the estimated position by Mini-Max is close to the position computed through Lateration method, which is represented by the intersection of the three circumferences. For every anchor i, the related squares is obtained by adding and subtracting the estimated distance from the anchor

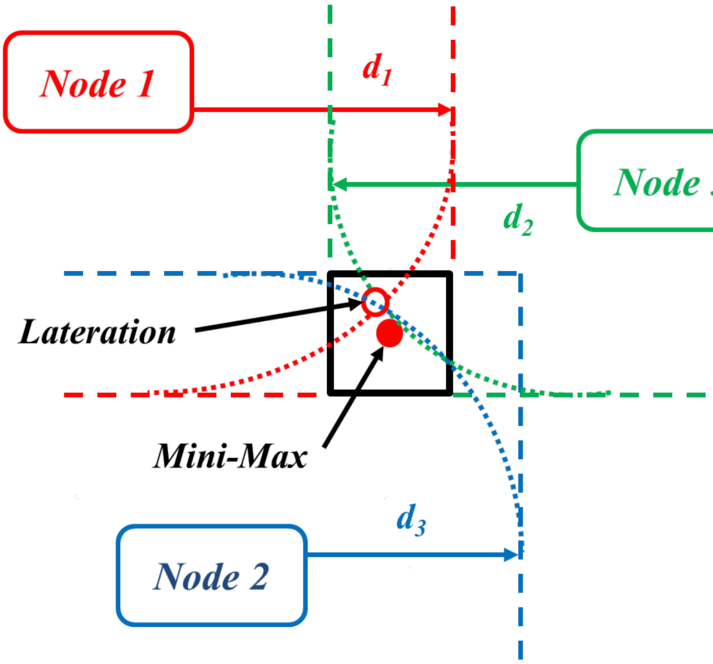


Figure 1.9: Mini-Max and Lateration.

position (x_i, y_i):

$$[x_i - d_i, y_i - d_a] \times [x_a + d_a, y_a + d_a] \tag{1.10}$$

The intersection of the squares is computed by taking the maximum of all coordinate minimums and the minimum of all maximums (22):

$$[\max(x_i - d_i), \max(y_i - d_i)] \times [\min(x_i + d_i), \min(y_i + d_i)] \tag{1.11}$$

From Equation 1.11, it is clear the origin of the method's name. The final position is set to the average of both corner coordinates.

1.3.2.2 Triangulation

It is an angle based and anchor based method. It uses trigonometric rules to estimate the distance, based on the angles that a point in the space forms with the nodes, which have known coordinates. It is generally used in Non Line-Of-Sight (NLOS) conditions, when the target node and the anchors cannot see each other. Referring for example to Figure 1.10, the distance AB is known since A and B are anchor nodes.

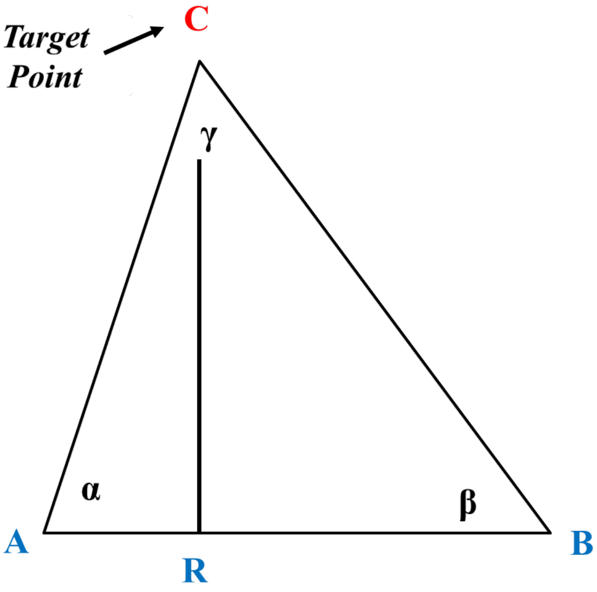


Figure 1.10: Triangulation.

The objective is to estimate the position of node C, supposing to know the angles α, β, γ. From the law of sines, one can simply obtain:

$$AC = \frac{AB \sin \beta}{sin\gamma}, \qquad BC = \frac{AB \sin \alpha}{sin\gamma} \tag{1.12}$$

$$RC = BC \sin \beta = AC \sin \alpha \tag{1.13}$$

From RC one can obtain the coordinates of point C.

1.3.2.3 Trilateration

This method is similar to the Mini-Max approach, but it is more complex and efficient. The information obtained from the algorithm is the set of points (locus) at the same distance d from the anchor nodes. The set of point forms a circumference in the 2-D trilateration and a sphere in the 3D trilateration, while the estimated distances are the radii. Since the target node is at the circles' intersection, in 2-D trilateration the distances from three different nodes are necessary normally to obtain the position without ambiguity. Similarly, in 3-D trilateration distances from four anchors are needed to obtain a unambiguous point. Considering for example the 2-D trilateration, Figure 1.11 represents the described scenario.

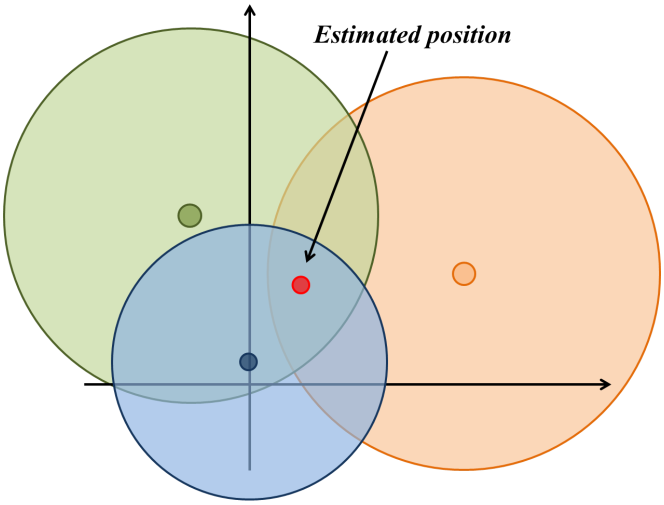


Figure 1.11: Trilateration.

The circumference equation of centre x_i, y_i and radius r_i is $(x-x_i)^2+(y-y_i)^2 =$

r_i^2, therefore the node's estimated coordinates x, y have to satisfy Equation 1.14

$$\begin{aligned}(x-x_1)^2+(y-y_1)^2&=r_1^2\\(x-x_2)^2+(y-y_2)^2&=r_2^2\\(x-x_3)^2+(y-y_3)^2&=r_3^2\end{aligned}\tag{1.14}$$

The algorithm first considers only two circles. If no points are found as intersection of the circumferences, the radii are increased proportionally. If there is only one intersection point, then the algorithm ends. However, if there are two intersection points, the point at smaller distance from the third anchor is to be considered. The algorithm is iterated for the three anchors and then one evaluates the average of the found coordinates. Often the intersections are not only represented by single points but by small areas, which represent the margin of error of this approach.

1.3.2.4 Multilateration

It represents the general case of trilateration, when all the n anchors in the node's coverage area are considered. The system of equations becomes 1.15:

$$\begin{aligned}&(x-x_1)^2+(y-y_1)^2=r_1^2\\&(x-x_2)^2+(y-y_2)^2=r_2^2\\&\vdots\\&(x-x_n)^2+(y-y_n)^2=r_n^2\end{aligned}\tag{1.15}$$

1.3.3 Refinement

This phase refines the position obtained in the localisation phase, by taking into account information which was not available before, like the ranges between neighbouring nodes and anchors.

Savarese *et al.* (23) propose an iterative refinement procedure, which considers all the distances between neighbouring nodes. Nodes update their positions in a small number of steps and at the beginning of each step, a node broadcasts its position estimate in the network, while receiving the position estimates of the other nodes. Then it performs again the second task (Lateration) to determine the new position. When, after a certain number of steps, the position update remains stable, the refinement procedure stops and returns the final position of the node. In this method it is very important to efficiently prevent errors, which can be quickly propagated through the network.

Savvides *et al.* (26) describe an iterative multilateration technique. In a first phase, a node is located through a process of quadratic error minimization. Then, a relation for every neighbouring node is added to the node itself. Every relation comes with a confidence weight, which express how good the neighbouring node position estimation is. So a new system is created and by solving it, a new position is obtained, with a better confidence weight. As long as there are sufficient position estimation updates, the procedure is iterated by sending messages continuously, thus overloading the network and increasing power consumption, latency periods and protocol's complexity.

1.3.4 Already Developed Approaches

1.3.4.1 The Active Badge Location System

The Active Badge Location System (27) determines the location of an individual thanks to an infrared tag in the form of an 'Active Badge' that emits a unique code for approximately a tenth of a second every 15 seconds. The receiver can estimate user's position thanks to a proximity system. The disadvantage of this system is the high infrastructure cost and the fact that infrared systems suffer from interference caused by heating and light sources.

1.3.4.2 The Active Bat

Active Bat (28) is a low-power location system, with an accuracy of the order of centimetres. It is based on the principle of trilateration and relies on multiple ultrasonic receivers embedded in the ceiling, measuring time-of-flight (TDoA system).

1.3.4.3 RADAR

RADAR (29) is a RSSI-based localisation system. It records and processes signal strength information at multiple base stations, by combining empirical measurements and the propagation channel model of the considered indoor environment. The disadvantages of this system are a high cost of the infrastructure deployment and a low accuracy.

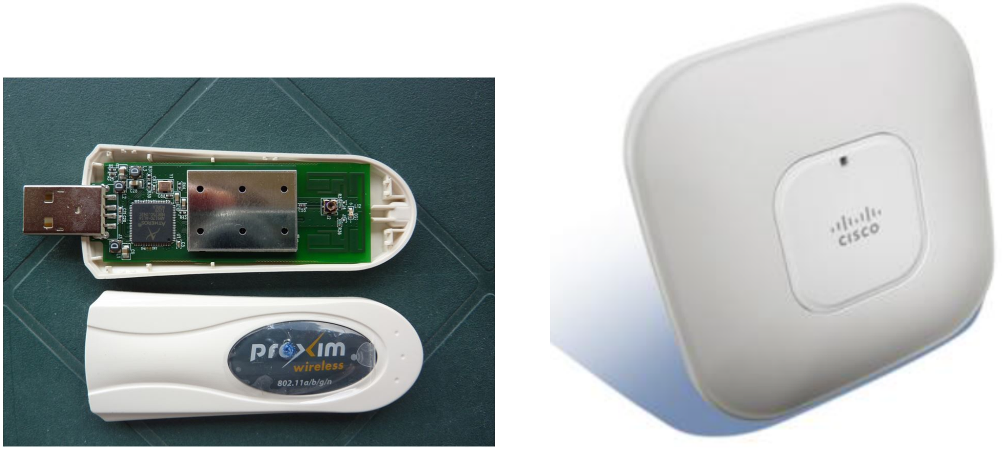
Chapter 2

Indoor Wireless Coverage and Propagation Assessment

In this chapter WLAN and LTE indoor wireless coverage have been evaluated by means of a dedicated software. The indoor environment considered is represented by one of the Telecom Italia premises, which is an example of a typical office environment. This task is composed of two phases:

- a *survey* to collect real WLAN signals and evaluate directly on the field the indoor wireless coverage;
- a *simulation* to simulate the propagation in the environment through specific software and evaluate how simulation and real measurements match.

As shown later, the simulation and the survey data do not match in areas which are quite distant from the AP. Thus, it is impossible to predict accurately how RSSI values for localisation purposes will be on specific points of an indoor environment. It is therefore clear that localisation methods based on RSSI values must be supported by other measurements.



(a) Proxim ORiNOCO® USB Adapter. **(b)** Cisco® Aironet 1140 Series AP.

Figure 2.1: The antennas considered.

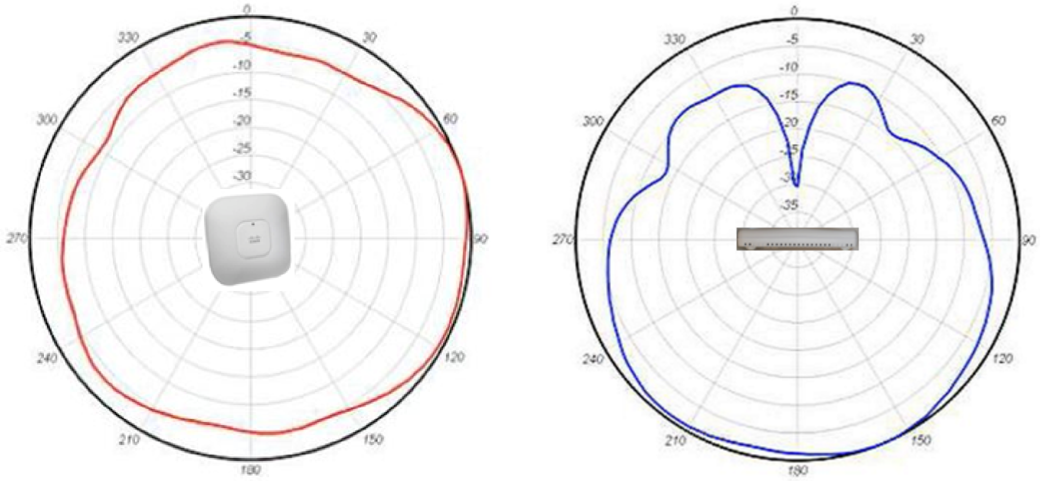
2.1 The Survey

The survey consists in the measurement of the RSSI fingerprints values, collected for example with a laptop. Since the considered building is basically composed of the same structure repeated several times, the indoor wireless coverage assessment considers just one area.

2.1.1 Receiving and Transmitting Antennas

The transmitting antenna, a Cisco® Aironet 1142n Access Point, is one of those already deployed in the environment. The receiving antenna, which is connected to the laptop for the collection of measurements, is the Proxim Wireless ORiNOCO® 802.11 a/b/g/n USB Adapter 2.1.

This antenna is mounted on the rear of the laptop screen in different orientations in order to consider all the possible polarizations. The survey is conducted by putting the laptop on a wooden tray and moving it slowly through the building's hallways.



(a) Azimuth plane.

(b) Elevation plane.

Figure 2.2: Azimuth and elevation patterns with respect to the AP.

2.1.1.1 Azimuth and Elevation Diagram

Before starting the survey, it is very important to understand how the azimuth and elevation radiation diagrams are oriented with respect to the external case of the Access Point.

In general, it can be said that the azimuth plane is as if you are looking top down through an AP and it is the normal antenna model type to select in the AirMagnet® software. Differently, elevation diagram is as if you are looking at the AP side on. Figure 2.2 shows this distinction.

Known that, it is clear why in the datasheet it is suggested to mount the antenna on the ceiling and not on the wall. The reason is that this antenna has an omnidirectional azimuth radiation diagram, that it to say that the antenna emits power in the same way for every angle in the plane which is parallel to the AP basis. Differently, the elevation diagram is concentrated top ahead, perpendicularly with respect to the AP basis. If such an AP is mounted on a wall, the room directly behind the AP will not receive a good amount of power, first of all because of the elevation diagram, but also because of the presence of a metal plate at the basis of the AP. Moreover, this situation increases potentially the interference between different floors. If this AP is mounted on a ceiling instead, one obtains a better coverage on the rooms on the floor and also have less interference among different floors.

It is interesting to note that the APs in the Telecom Italia premises are currently installed on walls instead of ceiling, thus being placed in a wrong way. However, this installation was preferred for simplicity and because on the ceiling there were already a lot of existing equipments.

2.1.1.2 Electromagnetic Field Polarization

Thanks to different measurements, it has been possible to understand in which plane the receiving antenna receives a higher field intensity. This knowledge has been later confirmed also by looking inside the transmitting antenna. The confirmation is the fact that the plane found through measurements is parallel to the plane of the transmitting antenna, where the electromagnetic field oscillates. So, once the theory has been confirmed by the measurements, the environment survey could start.

2.1.2 Software

The employed software is the AirMagnet® Survey. Measures are conducted in *passive survey* mode, with different antenna orientations. *“A passive survey is a quick and easy survey method that gathers all available 802.11 information in the wireless environment. This allows the user to view any traffic present, be it from a neighboring network or simply from ambient noise levels”* (54).

2.1.3 The Procedure

The first step has been the import of the map in .DWG extension. The map is loaded in the software and a new project is created.

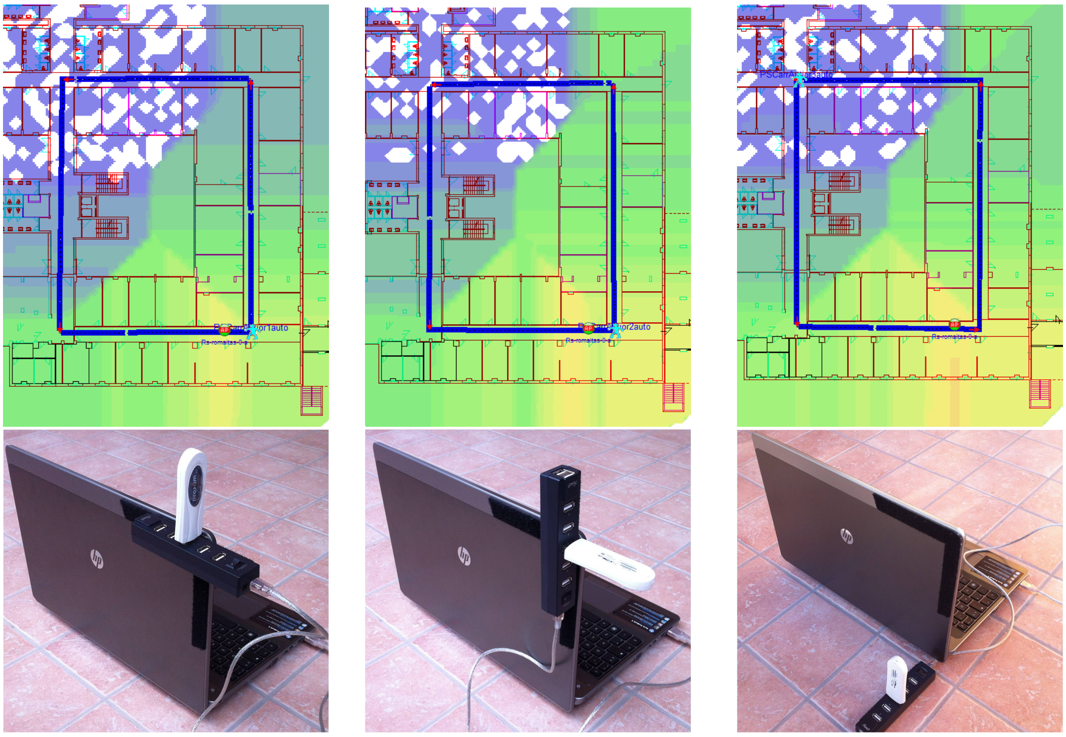
Once the software is running, one has to define the initial position, start the survey, and walk through the path. To check if there are any differences, walking has been done both clockwise and anticlockwise on the same path. Even different walking velocities have been adopted. If the *auto sampling through path* option is chosen in the configuration window, AirMagnet® Survey will automatically mark data sampling points on the site map, which are represented by the blue dots along the survey path. While walking, it is only necessary to mark the changes of direction on the map.

2.1.4 Survey Results

In Figure 2.3 three screenshots are shown. They show the received power on a path, with the specific position of the antenna shown below. They are all referred to a single AP, which has been identified by the software with its BSSID (Basic Service Set Identification), that is its MAC address. White spots are regions without coverage, while the signal level increases gradually from blue/green to yellow/orange. As already explained in Section 2.1.1.2, due to the polarization of the field emitted by the transmitting antenna, the received power levels in the first picture are considerably lower than those in the other two pictures.

2.2 The Simulation

The aim of this chapter is to look for a correspondence between the Survey and the Planner tools of AirMagnet® software. This is necessary if one wants to really know if the software works properly and if there is a good match between



(a) First position. **(b)** Second position. **(c)** Third position.

Figure 2.3: Wi-Fi power received considering three different positions of the receiving antenna.

simulation and real data.

Before starting the simulation with the planner tool, one first has to define the structural elements on the map, creating walls and windows with the correct attenuation levels. Therefore there are two options: either using the predefined attenuation levels proposed by the software, or defining new materials with personalized attenuation levels. This second way has been followed in order to obtain more precise simulations.

2.2.1 Experimental Setup

The target is to find the attenuation levels of the materials present in the indoor environment, in terms of dB. These values will be used in the simulation tool to verify if the simulation matches with the actual signal collection performed during the survey phase.

For this purpose, a Rohde & Schwarz® SMBV100A Vector Signal Generator and a Rohde & Schwarz® FSW Signal and Spectrum Analyzer have been used. The antennas connected to the instruments have a vertical polarization and a gain of 5dBi (55). Figure 2.4 shows the antenna considered.

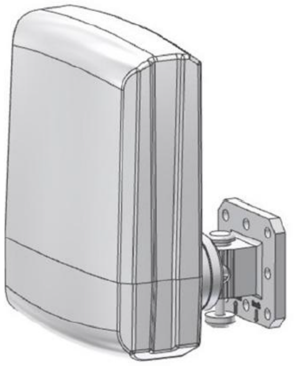


Figure 2.4: Antenna used for assessing the attenuation levels of material.

At the receiver side it is important to focus only on the direct path from transmitter to receiver, excluding other secondary paths, caused by reflections. This is the reason why an aluminium shield has been added to the antenna and mounted over a paper structure. Such a modification produces an antenna similar to a horn one. This antenna is mounted on a plastic pole and can be adjusted in height, in order to be at the same height of the transmitting antenna. Figure 2.5 shows this modification.

The measurement setup scheme is shown in Figure 2.6, while Figure 2.7 shows the practical implementation of the system.

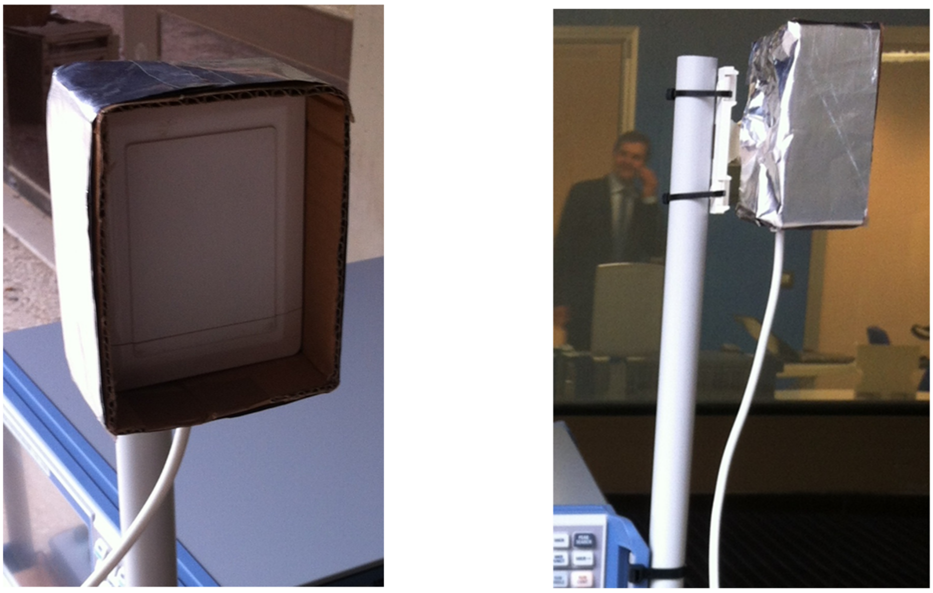


Figure 2.5: Receiving antenna modified.

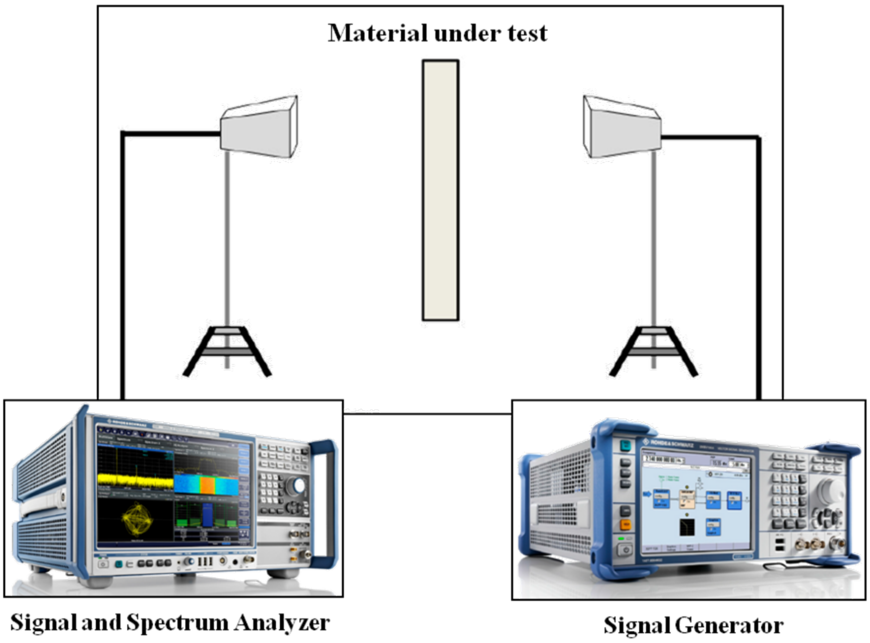
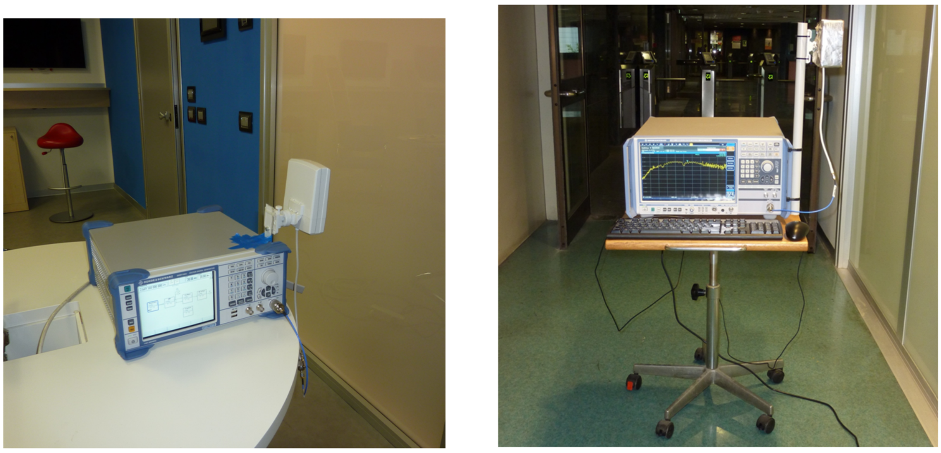


Figure 2.6: Measurement setup.

The signal generator is used to test the material at two different frequencies:



(a) Signal Generator.

(b) Signal Analyzer.

Figure 2.7: Experimental setup.

the Wi-Fi 2400 channel 1 and the LTE 2600.

2.2.1.1 Wi-Fi 2.4 GHZ Spectrum

There are four different frequency ranges in which Wi-Fi operates: 2.4 GHz, 3.6 GHz, 4.9 GHz, and 5 GHz bands. In this work the 2.4 GHz band is considered. Wi-Fi 802.11 b/g/n standards divide this band into 14 channels of 22 MHz each one 2.8. The channels are partially overlapping, so there is a strong interference among adjacent channels. The two channel groups 1, 6, 11 and 2, 7, 12 are non overlapping and they are used together in environments with many coexisting wireless networks. The AP considered transmits in the first channel, centred at 2.412 GHz and this is the reason why in the following the focus is on the first channel.

In addition to the centre frequency of 2.412 GHz , also 2.42 GHz and 2.404 GHz frequencies have been considered (at a distance of 8 MHz from the channel centre), to verify if the material attenuation varies with the frequency.

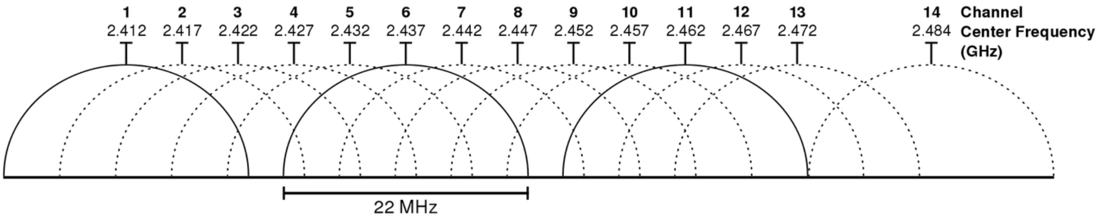

1 2 3 4 5 6 7 8 9 10 11 12 13 14 Channel
2.412 2.417 2.422 2.427 2.432 2.437 2.442 2.447 2.452 2.457 2.462 2.467 2.472 2.484 Center Frequency (GHz)
22 MHz

Figure 2.8: Wi-Fi channels (56).

2.2.1.2 LTE 2.6 GHZ Spectrum

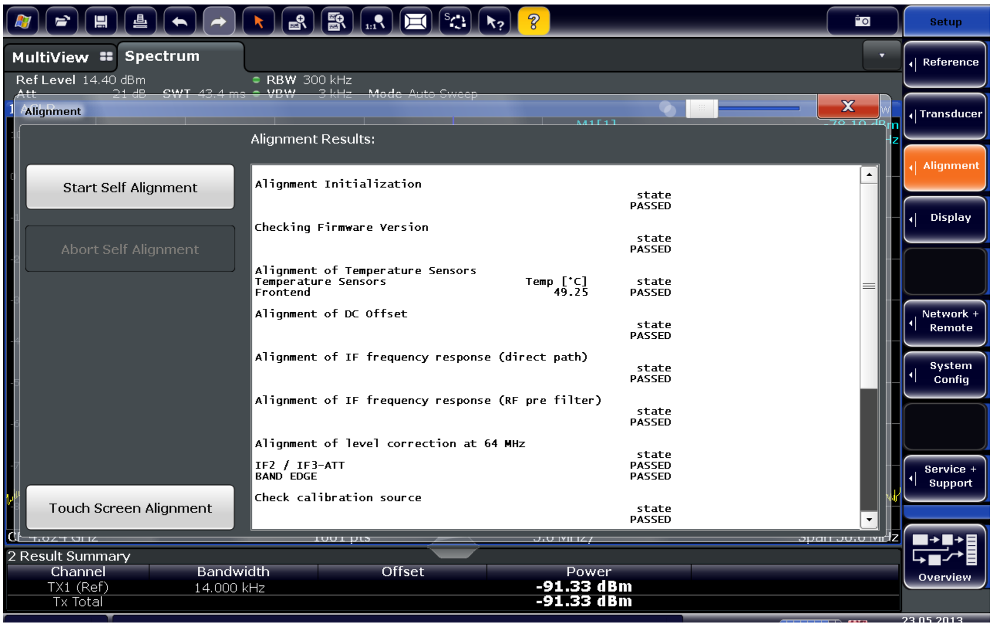
LTE 2.6 GHz has been chosen because transmissions in this frequency range have been detected with the analyzer. Again, not only a centre frequency of 2.66242 GHz is considered, but also two edge frequencies at distance 6.74 MHz from it, and precisely 2.66916 GHz and 2.65568 GHz.

2.2.2 Calibration of Instruments

First of all, the calibration of the instruments has been performed. The spectrum analyzer performs it automatically, by selecting the relative function, as shown in Figure 2.9.

About the signal generator, its output has been connected to the spectrum analyzer and the resulting measure have been read. If the transmitted power is much different from the received power, it means that there are problems. Figure 2.10 shows the process, while Table 2.1 reports the received power during the calibration.

As Table 2.1 shows, both Wi-Fi 2400 and LTE 2600 have been considered. The power emitted by the signal generator is set to 10 dBm in both cases and also the output of the second harmonic is considered. In both cases, the difference between input and output power is approximately 0.5 dBm, while the power received at the second harmonic frequency is negligible. The calibration phase is



Ref Level 14.40 dBm
Att 21 dB SWT 43.4 ms
RBW 300 kHz
VBW 3 kHz Mode Auto Sweep

Alignment Results:

```
Alignment Initialization
                                                          state
                                                          PASSED

Checking Firmware Version
                                                          state
                                                          PASSED

Alignment of Temperature Sensors
Temperature Sensors                            Temp [°C]   state
Frontend                                           49.25   PASSED

Alignment of DC Offset
                                                          state
                                                          PASSED

Alignment of IF frequency response (direct path)
                                                          state
                                                          PASSED

Alignment of IF frequency response (RF pre filter)
                                                          state
                                                          PASSED

Alignment of level correction at 64 MHz
                                                          state
IF2 / IF3-ATT                                             PASSED
BAND EDGE                                                 PASSED

Check calibration source
                                                          state
                                                          PASSED
```

2 Result Summary

Channel	Bandwidth	Offset	Power
TX1 (Ref)	14.000 kHz		-91.33 dBm
Tx Total			-91.33 dBm

Figure 2.9: Spectrum analyzer autocalibration.

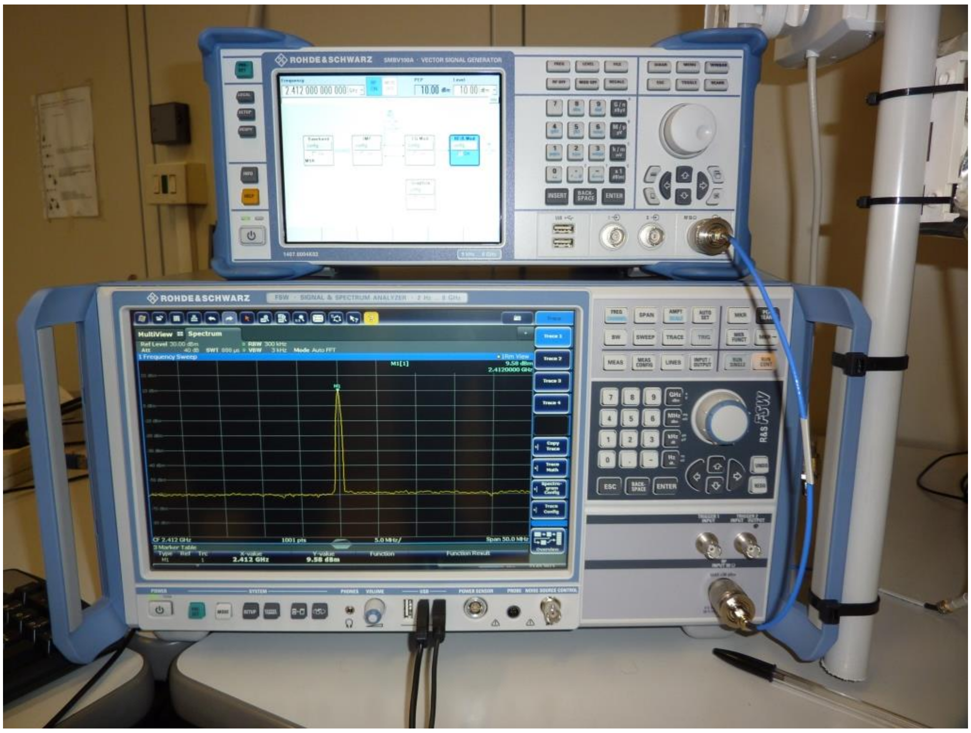


Figure 2.10: Signal generator calibration.

Instrument considered	Power (dBm)	Frequency (GHz)	Case
Signal generator	10	2.412	Wi-Fi 2.4
Spectrum analyzer	9.58	2.412	Wi-Fi 2.4
Spectrum analyzer	−24.95	4.824	Wi-Fi 2.4
Signal generator	10	2.662	LTE 2.6
Spectrum analyzer	9.40	2.662	LTE 2.6
Spectrum analyzer	−28.93	5.325	LTE 2.6

Table 2.1: Calibration of the signal generator: results.

thus considered to be concluded.

2.2.3 Reference Values

To obtain the attenuation coefficients of the different materials, measurements are compared with a reference scenario. Thus, the attenuation coefficients are computed by subtracting the received power in every situation with that one of the reference scenario. The reference scenario is the situation in which the transmitting and receiving antennas are 1 meter far from each other, in LOS condition. The distance of 1 meter has been chosen because it satisfies all the *far field* conditions both considering the Wi-Fi 2.4 GHz and the LTE 2.6 GHZ. This is important, because only in the far field condition, concepts like radiation pattern, gain, efficiency and in general all the antenna parameters are defined.

The conditions and the values referred to the frequency $f = 2.4$ GHz are shown by Equation 2.1.

$$\begin{aligned} & r \gg D \rightarrow r \geq 10D \rightarrow r \geq 1m \\ & r \gg \tfrac{\lambda}{2pi} \rightarrow r \geq 10\tfrac{\lambda}{2pi} \rightarrow r \geq 0.19m \\ & r \geq \tfrac{2D^2}{\lambda} \rightarrow r \geq 0.16m \end{aligned} \tag{2.1}$$

where D is the diameter of the transmitting antenna and $\lambda = c/f = 0.125$ m is the signal wavelength, given the light velocity $c = 3\exp 8$ and the signal frequency f. The resulting condition is $r \geq 1m$ and thus, the distance of 1 meter is adequate for measuring the Wi-Fi 2.4 GHz signal.

The same conditions are also satisfied at the frequency $f = 2.6$ GHz, as shown by Equation 2.2

$$\begin{aligned} & r \gg D \rightarrow r \geq 10D \rightarrow r \geq 1m \\ & r \gg \tfrac{\lambda}{2pi} \rightarrow r \geq 10\tfrac{\lambda}{2pi} \rightarrow r \geq 0.18m \\ & r \geq \tfrac{2D^2}{\lambda} \rightarrow r \geq 0.17m \end{aligned} \tag{2.2}$$

where $\lambda = c/f = 0.115$ m. Again, the distance of 1 meter is satisfactory.

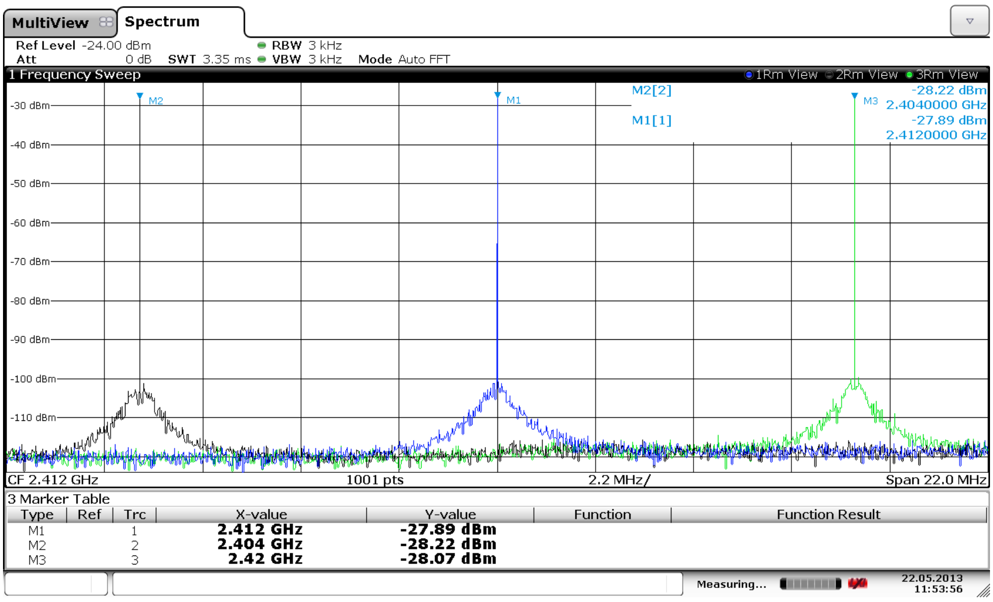
2.2.4 Signal Analyzer Screenshots

Received power measurements have been carried out for all the materials present in the environment and for both the Wi-Fi and LTE cases. Table 2.2 shows the frequency values considered for collecting the power received. Considering Wi-Fi, the transmitting power is set to 10 dBm, while for the LTE measurements the transmitting power is set to 20 dBm, due to the strong interferences with already deployed LTE transmitters.

In the following Figures 2.11, 2.12, 2.13, 2.14, some screenshots of the spectrum analyzer are shown . In every picture there are three peaks, which corresponds to the three frequencies considered, the central one and two more at the boundaries of the signal range (3 dB cut-off frequencies). The two border frequencies have been computed by measuring real Wi-Fi transmission and looking at the frequencies where the signal power dropped by approximately 3 dB.

Case	Frequency (GHz)
Wi-Fi 2.4	2.404
Wi-Fi 2.4	2.412
Wi-Fi 2.4	2.42
LTE 2.6	2.66568
LTE 2.6	2.66242
LTE 2.6	2.66916

Table 2.2: Frequencies considered for the material measurements.



Ref Level -24.00 dBm
Att 0 dB SWT 3.35 ms RBW 3 kHz VBW 3 kHz Mode Auto FFT

3 Marker Table

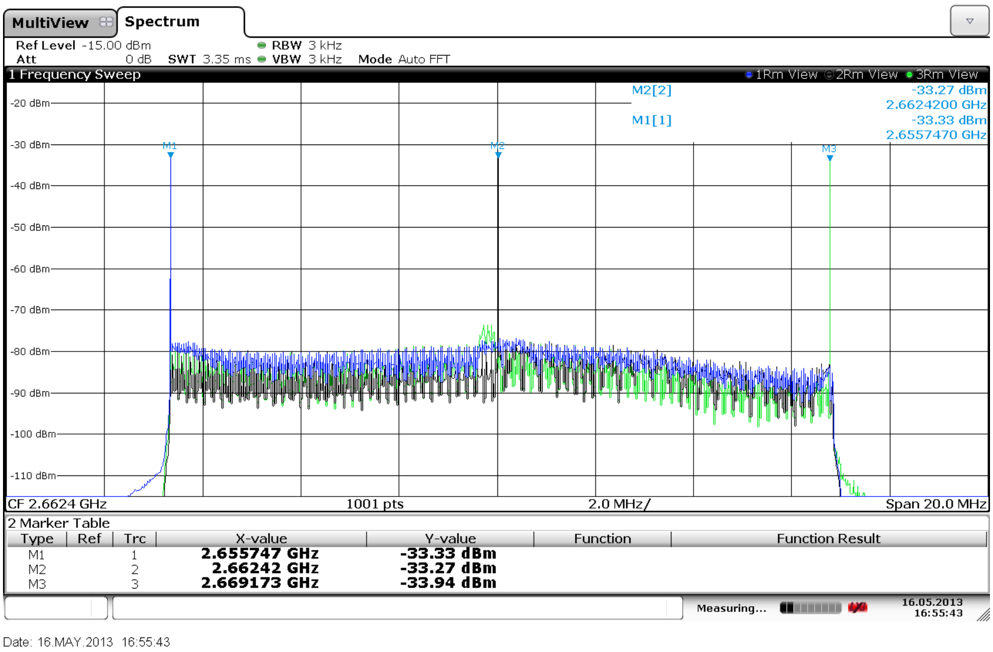
Type	Ref	Trc	X-value	Y-value	Function	Function Result
M1		1	2.412 GHz	-27.89 dBm		
M2		2	2.404 GHz	-28.22 dBm		
M3		3	2.42 GHz	-28.07 dBm		

Date: 22.MAY.2013 11:53:55

Figure 2.11: Wi-Fi Power received - reference scenario.

By analyzing the measurements in Wi-Fi frequency range, the following considerations apply.

- Most of internal walls basically attenuate signals of 3 dB. However, the walls have metallic structures (every 1.5 metres approximately) which rise a lot the attenuation factor of the wall, till about 10 dB. This is an important



Ref Level -15.00 dBm ● RBW 3 kHz
Att 0 dB SWT 3.35 ms ● VBW 3 kHz Mode Auto FFT

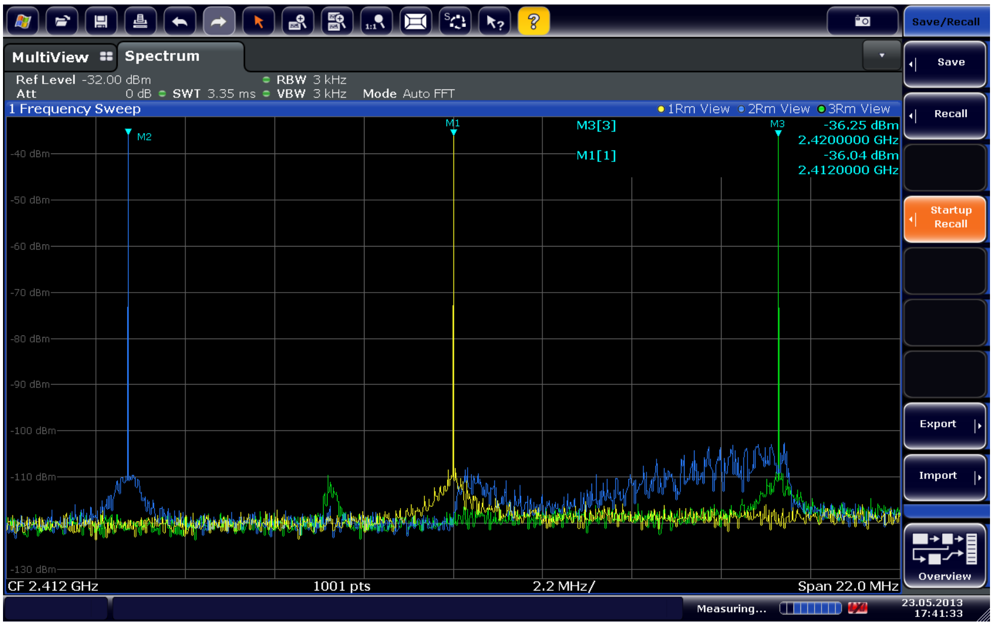
M2[2] -33.27 dBm 2.6624200 GHz
M1[1] -33.33 dBm 2.6557470 GHz

2 Marker Table

Type	Ref	Trc	X-value	Y-value	Function	Function Result
M1		1	2.655747 GHz	-33.33 dBm		
M2		2	2.66242 GHz	-33.27 dBm		
M3		3	2.669173 GHz	-33.94 dBm		

Date: 16.MAY.2013 16:55:43

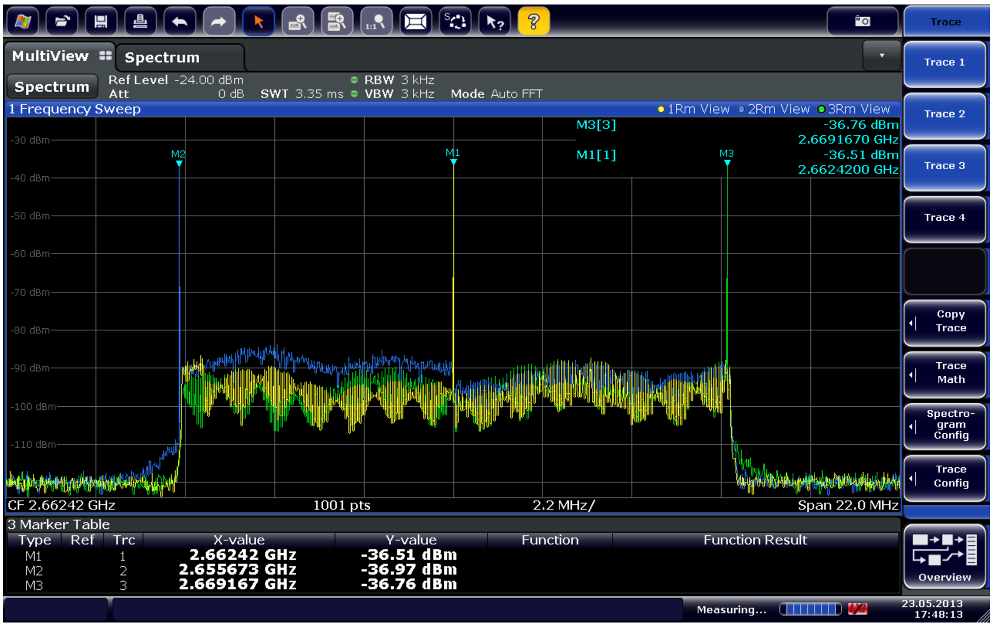
Figure 2.12: LTE Power received - reference scenario.



Ref Level -32.00 dBm ● RBW 3 kHz
Att 0 dB ● SWT 3.35 ms ● VBW 3 kHz Mode Auto FFT

M3[3] -36.25 dBm 2.4200000 GHz
M1[1] -36.04 dBm 2.4120000 GHz

Figure 2.13: Wi-Fi Power received - concrete wall.



Ref Level -24.00 dBm
Att 0 dB
SWT 3.35 ms
RBW 3 kHz
VBW 3 kHz
Mode Auto FFT

M3[3] -36.76 dBm 2.6691670 GHz
M1[1] -36.51 dBm 2.6624200 GHz

CF 2.66242 GHz
1001 pts
2.2 MHz/
Span 22.0 MHz

3 Marker Table

Type	Ref	Trc	X-value	Y-value	Function	Function Result
M1		1	2.66242 GHz	-36.51 dBm		
M2		2	2.655673 GHz	-36.97 dBm		
M3		3	2.669167 GHz	-36.76 dBm		

23.05.2013
17:48:13

Figure 2.14: LTE Power received - concrete wall.

fact to stress, because it shows that the actual attenuation factor of every single section of a wall really depends on many factors and can be very different from adjacent sections. However, for the sake of simplicity, 3 dB has been considered for every internal wall.

- The attenuation of internal windows is approximately 1 dB.
- The attenuation of other walls are 15 dB, 6 and 8 dB.

2.2.5 Simulation Results with Only One AP Considered

Due to time constraints, only the Wi-Fi case has been fully investigated. The values found above have been used in the AirMagnet® Planner tool. Then, the simulation has been launched to obtain the simulated coverage of the AP in the target area, as shown in Figure 2.15.

Simulation results and survey data are quite similar only in the surroundings

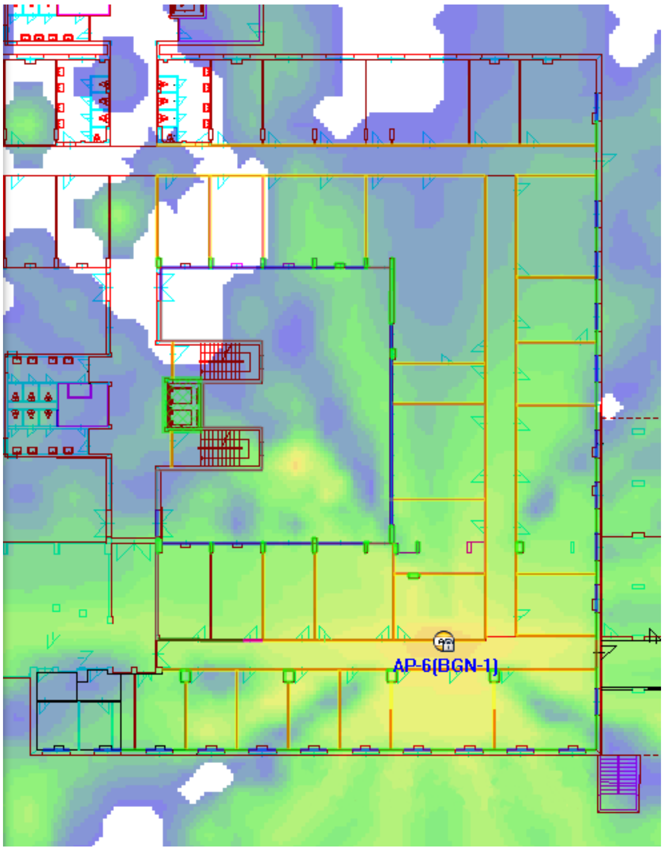


Figure 2.15: Simulation results.

of the AP and only considering the survey related to the receiver which is parallel to the polarization plane of the transmitting antenna. However, they do not match if considering areas which are farther from the AP.

Thus, this is a practical demonstration of the imprecision of localisation methods based on RSSI fingerprints, given the difficulty in predicting precise RSSI values in a specific point, also considering the required level of accuracy for localisation purposes. Thus, one has the need to improve the localisation results by considering other measurements, as explained in the following Chapters 3 and 4.

Chapter 3

The Theoretical Foundation of the System

Most of indoor positioning methods investigated in academia and industry are based on IEEE 802.11 Wireless Local Area Networks (WLAN) standard. This technology has been chosen for localisation purposes because it is widespread and it is especially deployed in indoor environments. In these locations, time or direction-based ranging methods are not suitable since they require Line-of-Sight paths and special expensive hardware. For this reason, in this work WLAN RSSI method is preferred to the others. In addition, thanks to the proliferation of mobile devices with many embedded sensors, the so-called “smartphones”, a positioning scheme known as *Pedestrian Dead Reckoning* has become more and more relevant.

Differently from Chapter 1, which gives a State-of-the-Art on localisation techniques, in this chapter the focus is on the methods chosen for the proposed system and the theory behind them is explained. The following Chapter 4 completes the system description, by illustrating how the different parts of the algorithm are implemented in practice and providing at the same time some relevant results.

3.1 An Overview of the Proposed System

The proposed indoor localisation system is conceived mainly for mobile devices like smartphones. This system is based on the fusion of multiple information sources via a particle filter. The considered information sources are:

- RSSI WLAN values collected from nearby access points;
- measurements from sensors like magnetometer, accelerometer, gyroscope and gravimeter;
- the map of the indoor environment with information about positions of walls and doors.

A very important aspect, especially for the Inertial Navigation System as it will be clearer later, is how to infer the user's *initial position.* This problem can be solved for example by obtaining last known GPS position from the mobile device. As an alternative, the user himself could locate himself, for example by scanning a QR code, RFID tag or a NFC tag and thus obtaining the information about his current position.

First of all, the system can be divided in two different phases: the offline and the online phase. The *offline phase* is the initial phase of system deployment, during which *RSSI fingerprints* at known reference points are collected and stored in the fingerprinting database on the server. This phase is necessary in order to build a "radio map" for the target environment, which is essential for the following algorithm. The *online phase* occurs every time a user asks for the localisation service, for example while entering the building where the localisation service is implemented. While moving inside the building, the user collects new fingerprints, which are compared to those saved in the database, with the help of a best matching algorithm or through a neural network. Instead of using a deterministic or a probabilistic best matching approach, a *neural network* has been implemented for this mapping operation, because of the advantages such approach provides.

Figure 3.1 gives an overview of the online phase of the proposed system.

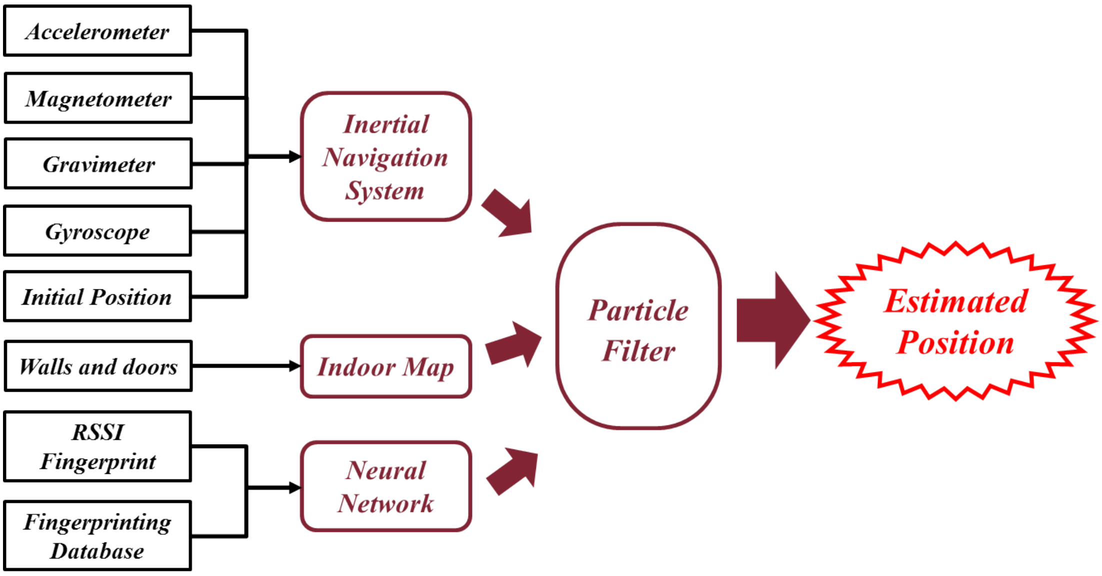


Figure 3.1: The online phase of the system.

RSSI values are not the only source of information considered. Thanks to the sensors embedded in modern smartphones, a *pedestrian dead reckoning* algorithm is also implemented. This algorithm processes values from sensors like accelerometer, magnetometer, gravimeter and gyroscope in order to update the user's position at every time step. In order for this algorithm to properly work, it is necessary to obtain the user's initial position. This can be accomplished by using a *GPS* signal (for example while entering the building), a *QR code*, a *Radio Frequency IDentification* tag, a *Near Field Communication tag* or *cellular positioning methods*.

To merge the position estimated by the neural network with the one estimated by the dead reckoning approach, a *particle filter* is considered. The particle filter considers a system with a state and a measurement equation. Noisy measurements are represented by the position estimated by the neural network while the motion model is represented by the sensors' integration equations. The *map* of the indoor environment has to be considered, because it gives necessary information about the positions of walls and doors and thus it is extremely useful in order to assess the weights of each particle in the filter.

3.2 The Fingerprinting Approach

As the name suggests, this method considers the collection of RSSI values, the so-called *fingerprints*, at some known positions in the target environment. The collected fingerprints are necessary to create the relation between RSSI values and actual positions on the map. As explained before, this method implies two distinct phases: an *offline* training phase and an *online* position determination phase. Figure 3.2 gives an overview of this system.

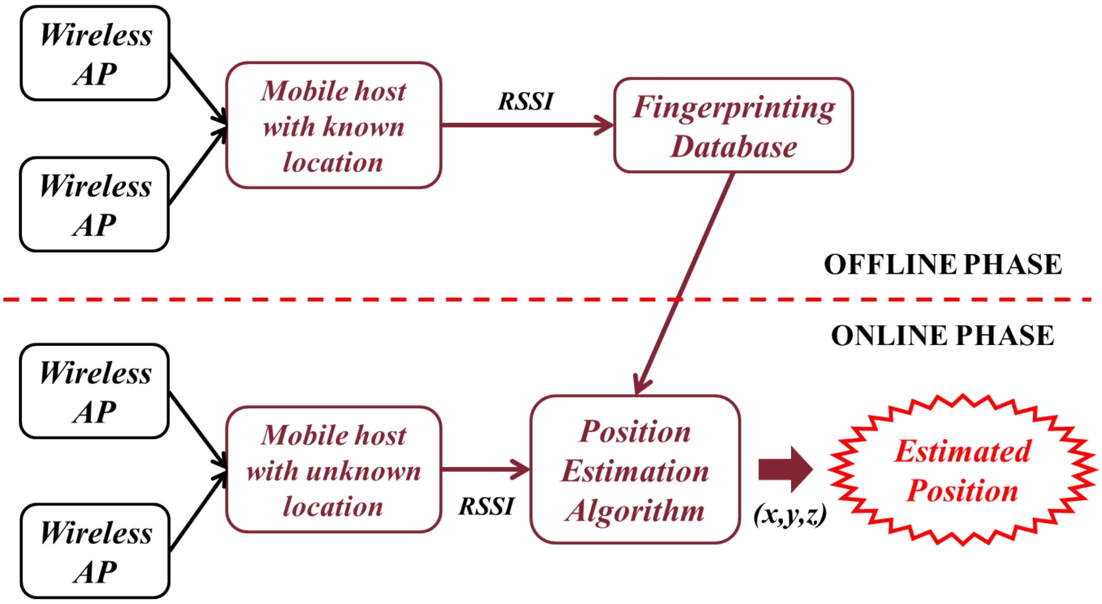


Figure 3.2: An overview of RSSI fingerprinting.

This method is relatively easy because it does not consider the electrical or geometrical properties of the environment, since it is based only on the received signal power. A disadvantage is the duration of the offline phase and the fact that this approach defines beforehand the performance of the system (the more the reference points are, the more precise the method will be). There are, however, solutions to the first issue. For example, Kothari *et al.* (24) use a robot to help them with the RSSI offline collection.

The most important matching algorithms can be basically divided into deterministic and probabilistic approaches.

- The *deterministic approaches* are quite simple but not very accurate. The *Nearest Neighbour in Signal Space (NNSS)* algorithm associates to the currently measured RSSI vector the collected vector at minimum distance according to a specific metric. The general form is expressed by Equation 3.1:

$$D = \frac{1}{N}\left(\sum_{i=1}^{N}\frac{1}{\omega_i}|s_i - \rho_i|^p\right)^{\frac{1}{p}} \tag{3.1}$$

  where:

  - N is the total number of APs;
  - ρ_i and s_i are the current offline RSSI value and the online fingerprint value respectively (related to the i-th access point);
  - ω_i is the weight assigned to the i-th AP measure.

  The most used metric is the Euclidean distance, where $p = 2$ and $\omega_i =$

$1, \forall i = 1 : N$. In this case, Equation 3.1 can be rewritten as Equation 3.2.

$$D = \sqrt{\sum_{i=1}^{N} (s_i - {\rho_i}^2)} \tag{3.2}$$

An improvement of this algorithm is the *k-Nearest Neighbour in Signal Space (kNNSS)*, where the average distance from k nearest neighbours in signal space is considered. In other words, first the k sample locations whose RSSI signatures are the closest in Euclidean distance to the currently collected RSSI vector are found. Then, the estimator returns the estimated location as an average of the coordinates of these top k locations.

This approach has been used for example by Tsai *et al.* (21) in their implementation of a RF-fingerprint-based location system at the Taipei World Trade Center (TWTC), Taiwan. The performances reached however are not excellent, since the 50-percentile localisation error is about 6.2 m.

- The *probabilistic approaches* uses Bayes Theorem and the definition of conditional probability. They achieve generally better performances, requiring however higher computational sources, since they need a higher number of RSSI samples taken per position, with consequences on their training time and cost (25). First, one has to know the probability density which better estimates RSSI offline values, also implementing an indoor radio propagation channel model. For every position it is possible to estimate the conditional probability density function by creating an histogram with the values recorded by the APs for every point. These values will be weighted in relation to the histogram associated to the APs in every position. Otherwise it is possible to consider that the RSS values of an AP for a certain position are Gaussian distributed.

3.3 Artificial Neural Networks

In this work a neural network is chosen to map RSSI values into actual positions. Artificial neural networks (ANN) are intelligent *multi-input multi-output* systems which have the capability of adaptation and learning (30). They are inspired by biological neural networks and they are characterized by massive parallel computing of simple processing units, interconnected among each other. These units are called *neurons* and they are organized in layers. The first layer is the input layer, where input data get into the network and the last layer is the output layer, which returns the output of the network. In the middle there are hidden nodes which process properly input data to return outputs. An ANN can be represented as in Figure 3.3.

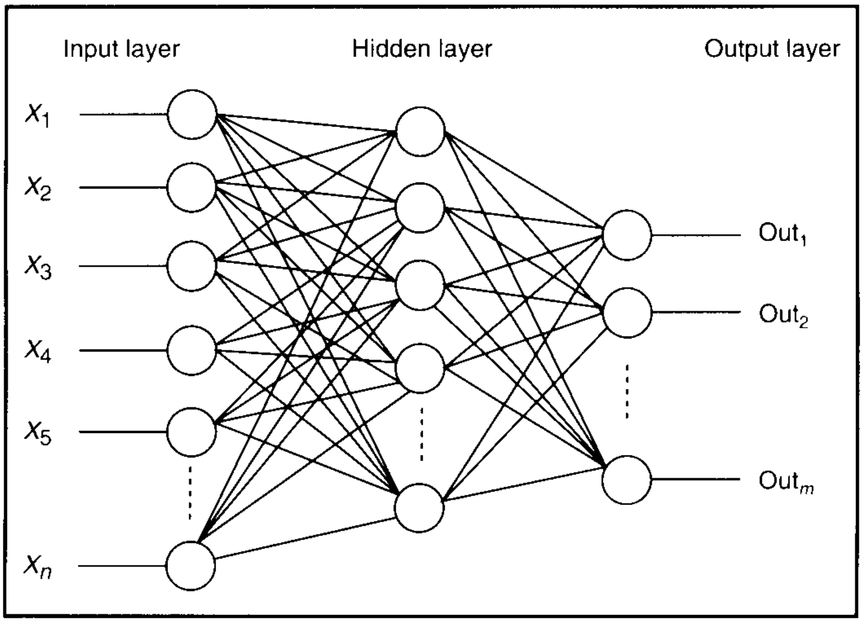


Figure 3.3: The structure of a neural network.

An ANN resembles the brain in two aspects:

- knowledge is acquired by the network from its environment through a *learn-*

ing process;

- interneuron connection strengths, known as *synaptic weights*, are used to store the acquired knowledge.

These systems are particularly suitable in *multi-sensor data fusion* since they are able to merge and process different kinds of information from different information sources. Here, this approach is very useful for the *pattern recognition*, which, in this case, is the automatic classification of an input fingerprint among those recorded in the database. In other words, neural networks are powerful tools for mapping the online collected fingerprints with those saved during the offline phase. This approach has already been pursued previously by many authors like in (9, 31, 32, 50). In (9), for example, a fingerprinting approach and the best matching algorithm is replaced by using a neural network. A feed-forward back-propagation algorithm has been applied and RSSI values are pre-processed in order to increase the system accuracy.

3.3.1 ANN Taxonomy

Neurons are intelligent autonomous units able to adapt themselves by acquiring information from outside or from other nodes. There are two main types of neural networks:

- *feed-forward networks* where the connection among neurons is one-way;
- *feedback* or *recurrent networks* where the connection is bidirectional with the possibility of feedbacks.

3.3.2 Learning Algorithms

At the beginning, the ANN has to learn the relation between inputs and outputs, in order to understand what the system *decision boundaries* are. These functions are normally nonlinear so that the problem of classification becomes a problem of function approximation. Instead of following a set of rules specified by humans, ANNs learn underlying rules (like input-output relationships) from a given collection of representative examples. This is one of the major advantages (64). The procedure used to perform the learning process is the *learning algorithm*, by which the synaptic weights are modified to attain a desired design objective. Actually, there are three main learning algorithms:

- *supervised learning* or *learning with a teacher* uses training samples, with a target output for every input. The desired target for every fingerprint in input is the reference location. In this system, a supervised learning algorithm is considered.
    - *reinforcement learning* is a variant of supervised learning in which the network provides only a critique on the precision of network outputs, not the correct answers themselves.
- *unsupervised learning* or *learning without a teacher* does not require target data associated with inputs. It explores correlations between patterns in the data and organizes patterns into categories from these correlations.
- *hybrid learning* combines the first two techniques since a part of the weights is usually determined through supervised learning, while the other is obtained through unsupervised learning.

3.3.2.1 Training, Testing and Validation Sets

After the training, the neural network performances are determined by another set of samples, collected at the same time of the training samples: the *testing set.* Last but not least, one has to consider the *overfitting* problem, that occurs when the network is too much adapted to training data. Possible reasons for this issue to occur are:

- the chosen model is redundant and it adapts itself too much to the training data;
- the training sample is too small compared to the free parameters of the system;
- the number of features is too high compared to the problem.

The problem with an overfitted network is that the resulting system is adapted also to the particularities of the training data and in particular to its noise, thus having low generalization capabilities. The *cross-validation technique* avoids this problem by dividing the training set in two subsets. The first one is the *training set* itself which is generally the 70% of the complete training set, while the remaining one is the *validation set*. When overfitting occurs, the error on the training set continues to reduce, while the error on other data, for example the validation set, gets bigger. This behaviour is used to measure the error: the learning is stopped at the minimum error on the validation set. Figure 3.4 represents the situation applied to a neural network implemented in MATLAB®. The learning stopping point is at the point of minimum error on the validation data set (green line).

3.4 Inertial Navigation System: Step-Based Dead Reckoning

As explained before, the inertial navigation system is a fundamental part of the system, implemented to increase the accuracy of the position estimation.

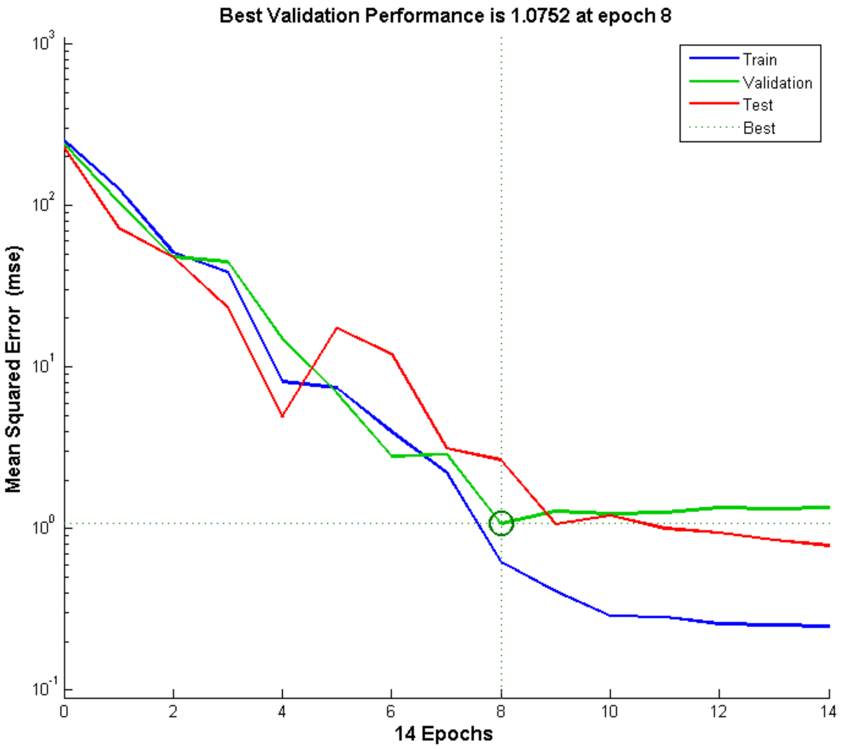


Figure 3.4: Example of *cross-validation stop-learning criteria*. If the error on validation data increases while the error on training data decreases, *overfitting* occurs.

The lost cost MEMs (Micro Electro-Mechanical sensors) integrated in smartphones cannot be very expensive and thus they are inevitably quite inaccurate. Therefore, it is important to consider the possible sources of errors which could propagate indefinitely through the algorithm, if not regularly corrected. This correction could be made for example by using RFID tags (18), map matching (18) or landmark recognition (20). Actually, due to the inaccuracy of the sensors, it is not possible to determine a user's position by a simple time integration of the acceleration values recorded. Therefore it is necessary to develop other approaches. One of the most important methods is the *step detection algorithm*, which first detects the steps done by a person and after, it estimates the length of the steps and movement direction.

Basically, this technique estimates user's location on the basis of a previously estimated or known position and detecting how the user is moving by combination of different measurements. This process is known as *sensor fusion* since it properly combines information from different sensors sources, like accelerometer, magnetometer and gyroscope. First of all, it is important to define an initial location, which could be determined by using last known GPS position as in (17) or through RFID tags (18) or cellular phone positioning (19).

Many authors describe inertial navigation systems, as in (1), (2), (4) and (3). In (10) a transformed signal space is considered where the basis are the most important APs, selected through a Principal Component Analysis (PCA) process. They propose a probabilistic approach, which treats RSSI values as random variables and reference locations are modelled with a Gaussian distribution. Michaelides *et al.* (16) construct a likelihood matrix for WSN based on sensors' observations. The considered indoor area is divided into a grid and if the received power at a node is beyond (under) a threshold, it adds $+1$ (-1) to the cells that are inside its Region of Convergence (ROC). The device is located in the cell with the greater value.

3.4.1 Deployed Sensors

The measurements used in the algorithm derive from various sensors.

3.4.1.1 Accelerometer and Gravimeter

The accelerometer measures the device acceleration in m/s^2 with respect to the three axes of the body frame. The gravity is always present in the measurements and, if the orientation of the device with respect to the local frame coordinate system is known, also the direction of gravity can be estimated. Once known the direction of gravity, one can subtract it from the accelerometer values to obtain the real acceleration of the device. Potentially, by integrating the device acceleration, the velocity is obtained and by integrating the velocity the change in position is estimated. However, this is not possible in reality because of the low precision of the sensors and therefore it is necessary to think about other methods to estimate the position of a person, like for example step detection methods. Through specific algorithms, by knowing the steps done, one can obtain also the distance covered. These aspects will be treated more in detail in the following sections.

A gravimeter is a type of accelerometer which detects local gravitational field.

3.4.1.2 Gyroscope

The gyroscope measures the angular velocity around the three axes of the phone. The measurements are expressed in degrees or radians per second, and indicate how the device is rotating. Integrating the gyroscope output provides the total change in angle during the integration time. Hence, if the starting orientation is known, the current orientation can be estimated by observing the gyroscope output. For example, for an Android device, rotation is positive in the anticlockwise direction, like defined in the Android Developer Guide (57). Gyroscopes suffer from bias and integration errors which introduce drift in signal. If not corrected, these errors can propagate indefinitely.

3.4.1.3 Magnetometer

The magnetic sensor measures the strength and the direction of the magnetic field, with respect to the usual three axis of the body frame. The measurements are expressed in micro-Tesla. They are absolute values and thus errors do not propagate as in the gyroscopes. However, in indoor environments magnetometers suffer typically from many magnetic interferences. Sources of interference could be for example the ferromagnetic material used in the building construction, electric wires inside walls and ceilings, exterior equipment such as whiteboards and computers. All these elements create a magnetic field which interferes with the geomagnetic field.

3.4.2 Orientation Projection

The values recorded by the sensors are all referred to the body frame. However, the mobile device can have any arbitrary and variable orientation and it is therefore necessary to move from the body frame to the local frame, so that the measurements can be really useful in determining the user's position in the target environment. Therefore, the first step in the dead reckoning algorithm is the orientation projection, that is the projection of the measurements from the body frame to the local frame. For this purpose, Euler angles are used.

3.4.2.1 Euler Angles

The arbitrary orientation of a body in the space can be obtained by means of the so-called *Euler angles*, which have been introduced by the Swiss mathematician Leonhard Euler. This set of three angular rotations, namely *roll*, *pitch* and *yaw*, represents the rotations of the body with respect to the main axes, x, y and z respectively. The rotation is defined to be positive in clockwise direction. Figure 3.5 shows the angles.

By knowing the gravity acceleration, given by the gravimeter sensor, it is possible to find the Euler angles by the following relations 3.3 3.4 3.5 (5):

$$roll = \arctan\left(\frac{g_y}{g_z}\right) \tag{3.3}$$

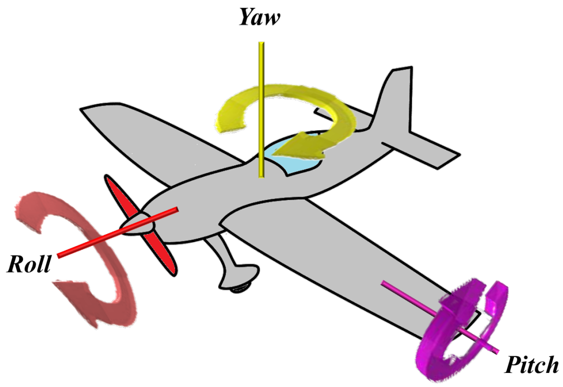


Figure 3.5: Euler angles *roll*, *pitch* and *yaw*.

$$pitch = \arctan\left(\frac{g_x}{g_z}\right) \tag{3.4}$$

$$yaw = \arctan\left(\frac{g_y}{g_x}\right) \tag{3.5}$$

where g_x, g_y and g_z represent the gravity values recorded by the gravimeter with respect to the body frame axes. These gravity values are filtered by a moving average in order to reduce the noise.

3.4.2.2 Gymbal Lock Singularity

The Gymbal Lock is the loss of one degree of freedom which occurs when two axes become parallel. Therefore the 3-D space degenerates to a 2-D space and the rotation of both of the two parallel axes gives the same result, with no difference between them. Thus, *roll* and *pitch* rotate around the same vertical axis. This situation occurs when *pitch* is equal to 90, that is to say when the object is looking up or down (5). An example is shown in Figure 3.6.

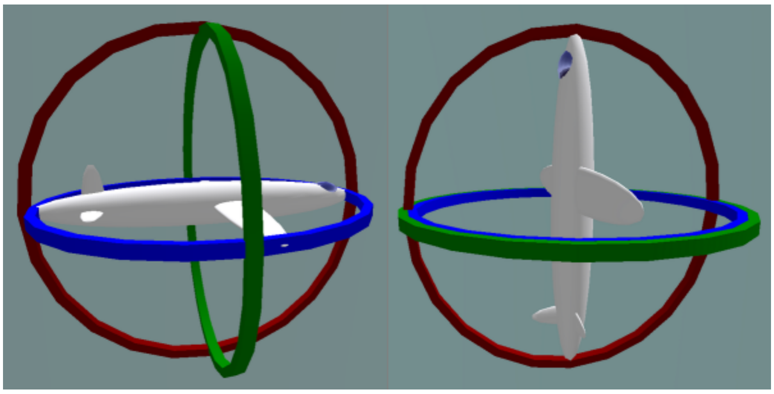


Figure 3.6: Gymbal lock singularity (58).

3.4.2.3 Rotation Matrix

Given a vector referred to the body frame, it is possible to transform it with respect to the local frame by multiplying it with the *rotation matrix* (2). A rotation matrix describes the orientation of one coordinate system with respect to another (6). One of its most important properties is its orthogonality. This property is fundamental because it assures that if two vectors are perpendicular in one frame, they are perpendicular also in every other frame. Moreover, the length of a vector has to be the same in every frame of reference and this is guaranteed by the fact that columns and rows are unit vectors, that is to say they have magnitude, which equals 1. Once known the Euler angles, the rotation matrix can be written as:

$$R = \begin{bmatrix} c(r)c(y) + s(r)s(p)s(y) & c(p)s(y) & -s(r)c(y) + c(r)s(p)s(y) \\ -c(r)s(y) + s(r)s(p)c(y) & c(p)c(y) & s(r)s(y) + c(r)s(p)c(y) \\ s(r)c(p) & -s(p) & c(r)c(p) \end{bmatrix}$$

where r, p and y indicate *roll*, *pitch* and *yaw* respectively while c and s indicate cos and sin respectively.

Actually, to represent the rotations of a body in space it could be possible to use also the *quaternions*, which have the advantages of requiring only 4 values, while rotation matrices have 9 elements. Moreover, quaternions avoid the gymbal lock problem. However, dealing with quaternions is quite complicated, while rotations matrices are more intuitive.

3.4.2.4 Some Adjustments

- The first problem to solve is the gymbal lock. The solution is represented by the method proposed in (2), which has been implemented in the current work. All the rotations around the vertical axis are assigned arbitrarily to *yaw*, while setting *roll* equal to zero. Therefore the values of *pitch* (equals to 90) and *roll* (equals to 0) are known and the only unknown quantity is *yaw*. Matrix R can be rewritten as:

$$R = \begin{bmatrix} c(y) & 0 & s(y) \\ -s(y) & 0 & c(y) \\ 0 & -1 & 0 \end{bmatrix}$$

  Now it is possible to compute *yaw* as:

$$yaw = \arctan\left(\frac{-R(2,1)}{R(1,1)}\right) \tag{3.6}$$

- Other issues are caused by numerical errors, namely *integration* and *quantization errors*. These errors cause two main issues in the rotation matrix. The first one is the loss of orthogonality of the matrix R while the second is the fact that columns and rows of the matrix are no more normalized.

To avoid these issues, the method proposed by Ayub *et al.* in (2) is implemented again. This technique basically consists in a correction of the elements of the matrix in order to satisfy the two conditions of orthogonality and normalization. Matrix R is rewritten as:

$$R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$$

and the vectors R_1 and R_2 are defined as:

$$R_1 = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} r_4 \\ r_5 \\ r_6 \end{bmatrix}$$

If the orthogonality condition is satisfied, the dot product of the R_1 andR_2 vectors must be zero. If not, this product is representative of how much the R_1 and R_2 rows are rotating towards each other. The error is given by Equation 3.7:

$$Error = R_1^T \cdot R_2 \tag{3.7}$$

The idea is to distribute equally this error in the two vectors R_1 and R_2 in order to make them perfectly orthogonal, obtaining the vectors R_{1orth} and

R_{2orth}, as shown in Equations 3.8 and 3.9

$$R_{1orth} = R_1 - \frac{Error}{2} R_2 \tag{3.8}$$

$$R_{2orth} = R_2 - \frac{Error}{2} R_1 \tag{3.9}$$

Finally, the R_3 row of the matrix is set to be orthogonal both to R_1 and to R_2, by computing the cross product of R_{1orth} and R_{2orth}:

$$R_{3orth} = R_{1orth} \times R_{2orth} \tag{3.10}$$

The last step is the normalization of the rows vectors. The method proposed in (2) is based on Taylor's expansion, obtaining:

$$R_{1norm} = \frac{1}{2}(3 - R_{1orth} \cdot R_{1orth}) R_{1orth} \tag{3.11}$$

$$R_{2norm} = \frac{1}{2}(3 - R_{2orth} \cdot R_{2orth}) R_{2orth} \tag{3.12}$$

$$R_{3norm} = \frac{1}{2}(3 - R_{3orth} \cdot R_{3orth}) R_{3orth} \tag{3.13}$$

3.4.2.5 Gyroscope Update

After a first step in which no previous sensors values are present, the rotation matrix is updated also by using gyroscope measurements. Gyroscope measurements are important since their integration over time brings to the current change in angle. The variations of the angles estimated with the gyroscope measurements are given by the following Equations 3.14, 3.15 and 3.16.

$$d\theta_x = gyro_x \delta t \tag{3.14}$$

$$d\theta_y = gyro_y \delta t \tag{3.15}$$

$$d\theta_z = gyro_z \delta t \tag{3.16}$$

where δt is the time step and $gyro_x$, $gyro_y$ and $gyro_z$ are the measurements from gyroscope with respect to the local frame axes. The update of the rotation matrix R is thus given by (6):

$$R_{gyro} = R \begin{bmatrix} 1 & -d\theta_z & d\theta_y \\ d\theta_z & 1 & -d\theta_x \\ -d\theta_y & d\theta_x & 1 \end{bmatrix}$$

3.4.2.6 Orientation Estimation

To estimate the orientation, the magnetometer values are considered. First, the magnetic values are multiplied by the rotation matrix R in order to obtain the magnetic field values with respect to the local frame. The magnetic field has in general three components on the three local frame axis. Then, the components parallel to x and y axis are considered and the orientation *heading* is then estimated by Equation 3.17:

$$heading = \arctan\left(\frac{m_y}{m_x}\right) \tag{3.17}$$

where m_x and m_y are the components of the magnetometer projected along the x and y axes.

3.4.3 Movement Threshold

Before starting the step detection algorithm, a threshold has to be applied in order to distinguish between movements related to the walking process and other movements. First, the summation of all the elements of the three acceleration vectors (related to the three local frame axes) are computed. Then, the following Equation 3.18 is computed:

$$if\left(\sqrt{sum_x^2 + sum_y^2 + sum_z^2} \geq movementThreshold\right) \tag{3.18}$$

where sum_x, sum_y and sum_z are the summations of the acceleration vectors parallel to axis x, y and z respectively and $movementThreshold$ is a value set to distinguish between the two kinds of movement. Such a condition allows to take into account the accelerations along the three axes considering more samples at the same time, that is a certain time window. The adoption of this time window allows possible involuntary or noisy movements to compensate with each other. In other words, for example if a noisy vibration at time t causes a positive acceleration along axis x, the same casual vibration will cause a negative acceleration along the same axis at a time not too far away from the first, for example at time $t+1$ or $t+2$. Thus, the summation of the elements of the vector related to the x axis will be approximately equal to zero, since the two values have opposite signs.

3.4.4 Step Detection

A step detection algorithm has the aim to understand the distance travelled. There are two main step detection algorithms, zero-crossing and peak detection method. The first method is implemented for example in (11), while the second method is used in (1) and (3). Both of them are based on the analysis of the component of the acceleration which is parallel to the local frame z axis.

3.4.4.1 The Walking Behaviour

When a person walks, the z axis acceleration has a sin-like pattern, with regular peaks caused by the legs' movement. As clearly explained in (12), the walking phase is composed of two phases, a swing phase and a heel-touch-down phase.

In the swing phase the leg is moved from behind the gravity centre of human body to the front of it and, as a result, the foot is accelerated during this phase. In the heel-touch-down phase, the foot, which is now in front of the body, hits the ground. The ground repulses the foot because of the action-reaction law and accelerates it. Thus, the foot is accelerated in both of the phases, but the duration of the second phase is much shorter than the first one. In the same work (12), the authors show a real measured horizontal acceleration signal where it is clear how strong the acceleration is in the heel-touch-down phase.

3.4.4.2 Zero-Crossing and Peak Detection Methods

The zero-crossing method detects the steps by counting the crossings of the zero level, so that each crossing is a step. However, the presence of noise leads to false step detection, which can be eventually rejected by setting proper thresholds. The peak detection method detects steps by detecting the peaks of the vertical acceleration, which are generated by the impact of the foot with the ground during the heel-touch-down phase. It has been decided to implement the peak detection method because zero-crossing method is not reliable when noise is present. The reason behind is that the setting of thresholds to eliminate false detections in the zero-crossing method can be very difficult to implement in practice. The peak detection method instead does not suffer particularly in terms of performance if some noise is added to the signal.

3.4.4.3 Relative Threshold Scheme

For the practical implementation of the peak detection method, a relative threshold scheme similar to (1) is implemented. It adopts two different thresholds: the *time threshold* δ_T and the *amplitude threshold* δ_A.

A new step is detected when a valid local maximum and a valid local minimum are detected in sequence. A *local maximum* is valid if it occurs at least δ_T ms after the most recent valid local minimum. In addition, the value of the local maximum has to exceed that of the most recent local minimum by at least δ_A. Similarly, a local minimum is valid if it occurs at least δ_T ms after the most recent valid local maximum, and the value of the local minimum is lower than that of the most recent local maximum by at least the amplitude threshold value.

The problem is how to choose the value of the two thresholds. This depends actually on the characteristics of the movement of every person and on its walking speed. As a rule, representing the movement as a sinusoid, the time threshold between two peaks (one maximum and one minimum) has to be less than half of the period of the sinusoid, while the amplitude threshold has to be a little lower than the sinusoid amplitude.

Some mean values can be used, like for example 150 milliseconds of time threshold, which is the case in which a person takes two steps per seconds, that is to say 4 or 5 peaks. It corresponds to a normal walking speed. The amplitude threshold instead has to be estimated through calibrations.

3.4.5 Step Length Estimation

After detecting a step, it is important to detect how long a step is in order to compute the actual distance covered. In literature, there are several methods for estimating the length of a step.

First of all, there are two main families of step length estimation methods: the *static methods* and the *dynamic methods*. The static methods suppose that every detected step has the same length, which can be determined through Equation 3.19.

$$stepLength = height \times k \tag{3.19}$$

where k is equal to 0.415 for men and 0.413 for women (3) and *height* is the height of the person being tracked.

The dynamic methods, instead, suppose that every step has a different length, which can be determined through various approaches, like the Weinberg (13), the Scarlet (15) or the Kim (12) methods. In the present work, a controlled environment in which every step has the same length of 1 metre is considered. Thus none of these approaches has been actually implemented. However, these methods, and especially the Weinberg approach, are not difficult to implement in a software.

3.4.5.1 The Weinberg Approach

This approach has been proposed by Weinberg in (13) and it has been verified in (14). According to this method, the k^{th} step length is given by Equation 3.20:

$$stepLength = K\sqrt[4]{a_k^{vmax} - a_k^{vmin}} \tag{3.20}$$

where a_k^{vmax} and a_k^{vmin} are respectively the maximum and the minimum value of the projected vertical acceleration during the k^{th} step. The constant K is dependent on each pedestrian and can be determined through calibrations.

3.5 Particle Filter

The information provided by the RSSI collection and the Inertial Navigation System is integrated in a particle filter. Particle filtering is a general Monte Carlo (sampling) method for estimating the state of a system at a certain time k. What is known are only the noisy measurements recorded at each time step up to and including time k (59) (62) and the past states. In particular, a particle filter has been chosen because, differently from Kalman filters, it is suitable for working with non-linear processes and functions. In a general discrete-time state-space model, the state of a system and the measurements evolve according to the following equations:

$$x_k = f_k\left(x_{k-1}, v_{k-1}\right) \tag{3.21}$$

$$z_k = h_k\left(x_k, n_k\right) \tag{3.22}$$

where x_k represents the state of the system at time k, v_{k-1} represents the noise related to the state, f_k is a possibly non-linear and time-dependent function describing the evolution of the state. The state vector x_k is supposed to be hidden or unobservable so that information about the current state of the system is obtained only through noisy measurements of it, represented by the vector z_k. h_k is a possibly non-linear and time-dependent function describing the measurement process and n_k represents the noise on the measurements. In particular, Equation

3.21 is called *state equation* and Equation 3.22 is called *measurement equation* The noisy measurements up to time k are denoted by $z_{1:k}$.

3.5.1 The Analytical Foundation

Analytically speaking, the purpose of a filter is to find the conditional probability $p(x_k|z_{1:k})$, given the set of observations $z_{1:k}$. As stated in (59), this can be accomplished by two different recursive steps: the prediction and the update step. In the *prediction step*, $p(x_k|z_{1:k-1})$ is computed from the filtering distribution $p(x_{k-1}|z_{1:k-1})$ at time $k-1$:

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{1:k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{3.23}$$

where $p(x_{k-1}|z_{1:k-1})$ is assumed to be known due to recursion and $p(x_k|x_{k-1})$ is given by Equation 3.21. The distribution $p(x_k|z_{1:k-1})$ can be thought to be the probability of x_k before receiving the most recent measurement z_k.

In the *update step*, this probability is updated with the new measurement z_k using Bayes rule, obtaining:

$$p(x_k|z_{1:k}) \propto p(z_k|x_k)p(x_k|z_{1:k-1}) \tag{3.24}$$

3.5.2 The Kalman Filter

If at time $k-1$, the probability $p(x_{k-1}|z_{1:k-1})$ is Gaussian, then it can be proved that at the next time step the probability $p(x_k|z_{1:k})$ is Gaussian, provided that the two following conditions are satisfied:

1. the state and the measurement evolution functions, f_k and h_k, must be linear;
2. the noise on the state and the measurements, v_k and n_k, must be Gaussian.

In this case, the state and the measurement equations simply become:

$$x_k = F_k x_{k-1} + v_{k-1} \tag{3.25}$$

$$z_k = H_k x_k + n_k \tag{3.26}$$

where v_{k-1} and n_k are Gaussian random variables and F_k and H_k are matrices which express the evolution of the state and of the measurement, which are assumed to be known. This system, found with the two previous conditions, brings to the well-known Kalman filter algorithm (59).

3.5.3 The Particles

In general it is not possible to solve analytically the prediction and the update steps (Equations 3.23 and 3.24). Thus, one has to think about approximate methods such us the Monte Carlo sampling, which are based on simulation rather than on analytical solutions. The basic concept is to approximate the posterior distribution at time $k-1$, $p(x_{0:k-1}|z_{1:k-1})$ with a weighted set of samples $\left(x^i_{0:k-1}, \omega^i_{k-1}\right)^N_{i=1}$, the so-called *particles.* These particles are updated at each time step to get an approximation of the posterior distribution $p(x_{0:k}|z_{1:k})$.

To reach this purpose, the idea is to use samples extracted from a proposal distribution $q(x)$ to approximate the target distribution $p(x)$. This is necessary since the target distribution is not known in general. Then, one has to compensate for the difference between the proposal and the target distributions, by weighting each particle x^i with a coefficient ω_i, which satisfies the relation:

$$\omega_i \propto \frac{p(x^i)}{q(x^i)} \tag{3.27}$$

The weights are normalized to sum to one. After every time step, only the particles with higher weights are considered and are carried on in the algorithm for the next steps. The particles with lower weights instead will not be considered.

The final position estimation at every step is computed as the mean of the resampled particles.

3.5.4 Information About the Map

In particular, the particle filter has as input also information about the positions of walls and doors. This is important because if the algorithm detects that a particle is crossing a wall, it assigns to it a weight, which is equal to zero. Thus, in the following step the particle is no more considered, so that only the particles which actually do not crosses walls are taken into account.

It is important also not to reduce the number of particles at each iteration. Therefore, the more important particles are considered more than once, until the vector of the particles is full, as it was at the beginning of the algorithm.

Chapter 4

The Implementation of the Algorithm

While the previous Chapter 3 gives a theoretical basis of the various elements of the system, the current chapter explains how these elements are actually implemented and interconnected with each other.

4.1 Validation Scenarios

The first step after the creation of an algorithm is its validation. For this reason, some validation scenarios have to be considered in the next sections. These are simple controlled indoor environments, where it is possible to evaluate the algorithm performances because everything is known a-priori. In particular, it is possible to evaluate the differences between nominal values and estimated ones, in order to try to minimize the errors committed.

The first scenario is represented in Figure 4.1. It is a square room of 4 meters side with various walls inside. These walls are made up of plasterboard with a signal attenuation of 3 dB. This scenario is considered only for the validation of the neural network. A grid is represented to better read the RSSI values at the different coordinates.

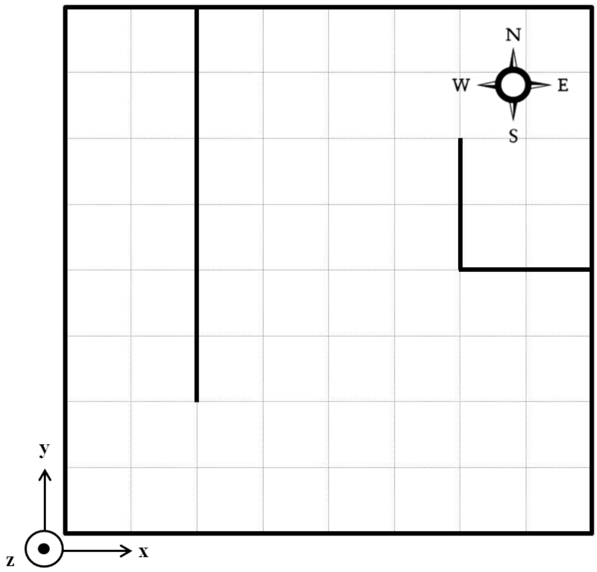


Figure 4.1: The first validation scenario, used for validating the neural network.

The second scenario, represented in Figure 4.2, is more complicated since it is used for the validation of both the dead reckoning algorithm and the particle filter. It is a square room of 24 meters side with many plasterboard walls inside.

In both scenarios, the magnetic north is considered to be parallel to the y direction, pointing up. The local frame coordinate system is also represented in the figures.

4.2 Neural Network Implementation

A simple *feed-forward* neural network is implemented. It is composed of three layers. The first layer is the *input layer* and is made up of three neurons which represent the three inputs: the RSSI fingerprint recorded at a specific location and the x and y coordinates of this location (the target position). It is not

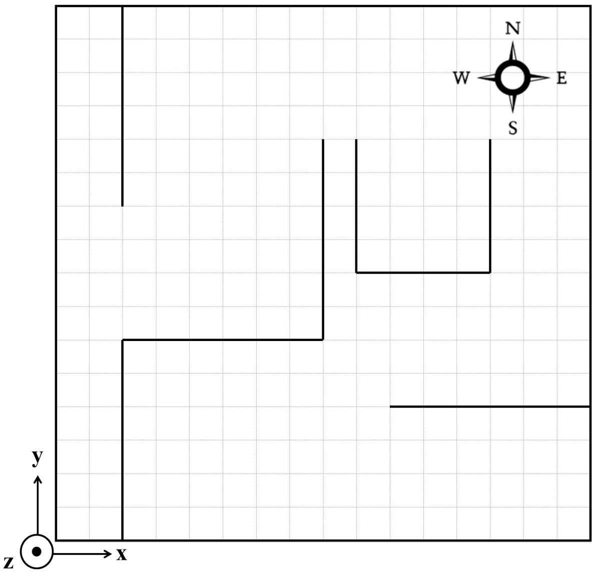


Figure 4.2: The second validation scenario, used for validating the inertial navigation system and the particle filter.

necessary to specify the BSSID vector, since, for the sake of simplicity, the RSSI fingerprint is considered to be composed of elements, which are already in order with respect to the access points. In other words, the first element of the RSSI vector is referred to the first AP, the second element is referred to second AP and so on. The *hidden layer* is composed by 10 neurons. This value has been obtained in an empirical way, after comparing the algorithm's performance with different numbers of neurons. The *output layer* is composed by the values returned by the algorithm, that is to say the estimated position in terms of x and y coordinates. In Figure 4.3 the network is shown as represented by MATLAB® software.

There are three access points at different positions but at the same height ($2.8m$). The access points radiation diagram is the same of Chapter 2. The heat map with the different levels of power as estimated by AirMagnet® software in the first scenario, are represented in Figure 4.4.

Two different sets of reference points are considered and compared: in the first

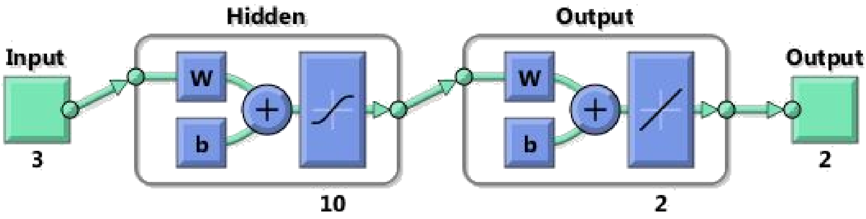


Figure 4.3: The neural network implemented.

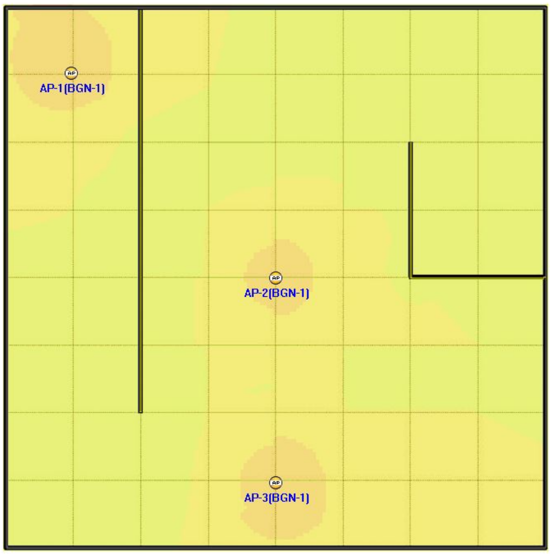


Figure 4.4: Distribution of power emitted by APs.

set there are 21 reference points while in the second one there are 34 reference points. The different sets of reference points and their distribution in the scenario are shown in Figures 4.5 and 4.6.

Figures 4.7 - 4.13 represent the results obtained from 21 reference points and 500 simulations, while Figures 4.14 - 4.20 show the results obtained from 34 reference points and 500 simulations. In both cases, the same 4 points are used to test the performances of the system.

Figures 4.7 and 4.14 show the histograms of the Root Mean Squared Error (RMSE) relative to all the 4 test points, considering separately the x and y

coordinates. Root of Mean Squared Error is used to show the discrepancy between real and estimated values.

Figures 4.8 - 4.11 and 4.15 - 4.18 represent the histograms of the errors consid-

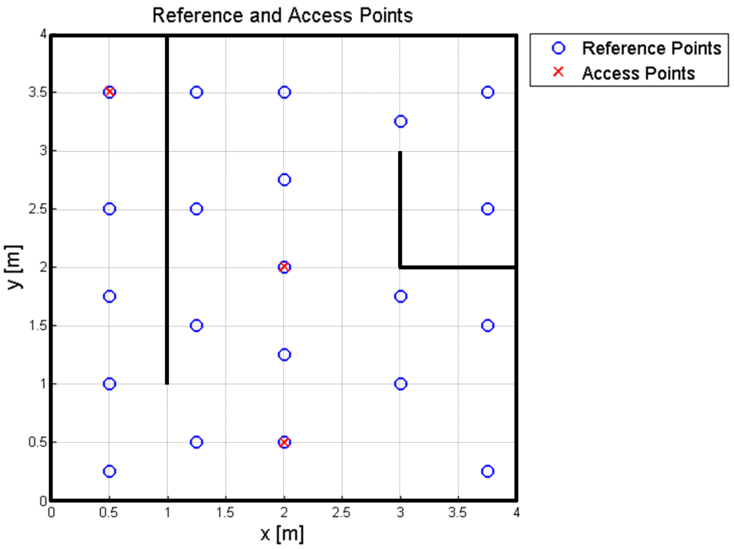


Figure 4.5: Scenario with 21 reference points.

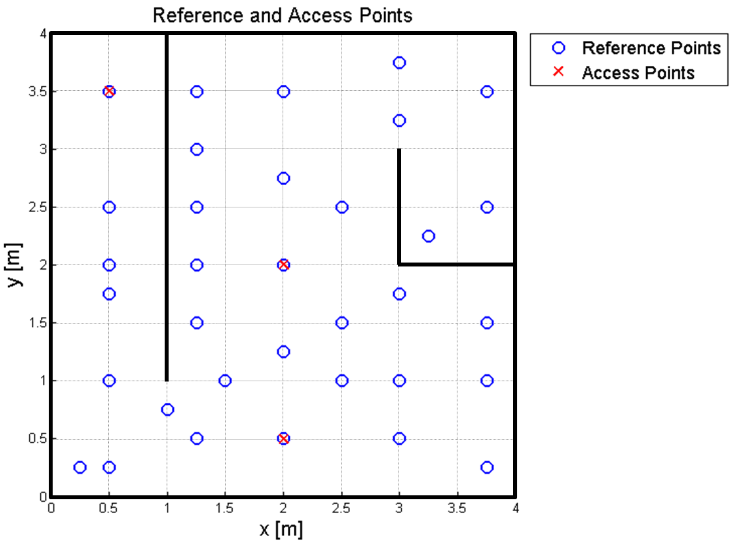


Figure 4.6: Scenario with 34 reference points.

ering the 4 points separately. They are drawn to check the system performances for every single point.

Figures 4.12 and 4.19 represent all the points estimated by the neural network. Considering that 500 simulations have been done, there are 500 estimated points for every single test point. The black squares represent the real positions of the test points, known a priori, while the black circles represent the final estimated points as a mean of the 500 estimated points from every single simulation. In addition, 2D Gaussians centred at the round points have been drawn to graphically give an idea of the scattering of the single estimates.

Figures 4.13 and 4.20 represent 3D Gaussians centred at the final estimated points. The higher the Gaussian are, the less the points are scattered around the finale estimate.

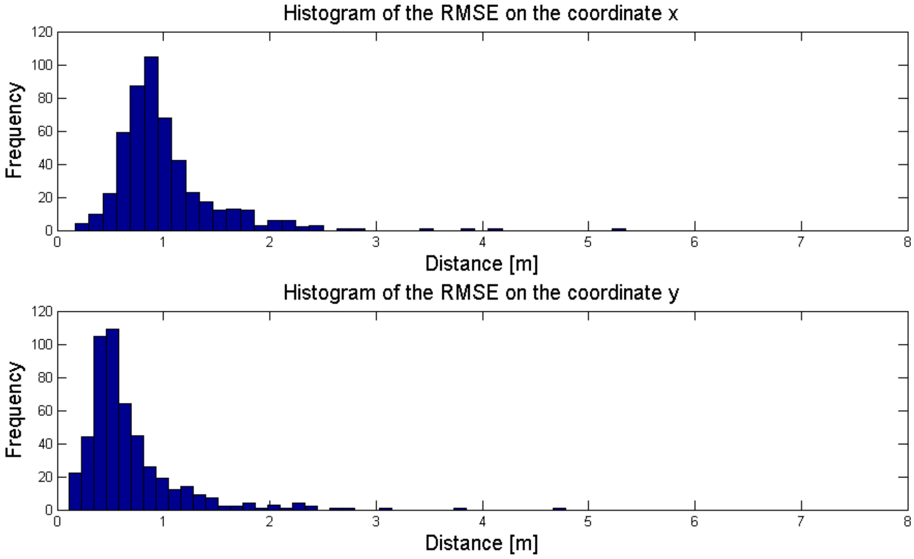


Figure 4.7: Histograms of RMSE on x and y coordinates (21 ref. points).

The results confirm that by increasing the number of reference points, it is possible to obtain better performances. This is particularly evident comparing Figures 4.12 and 4.19. In the figure related to 21 referents points the 2-D Gaus-

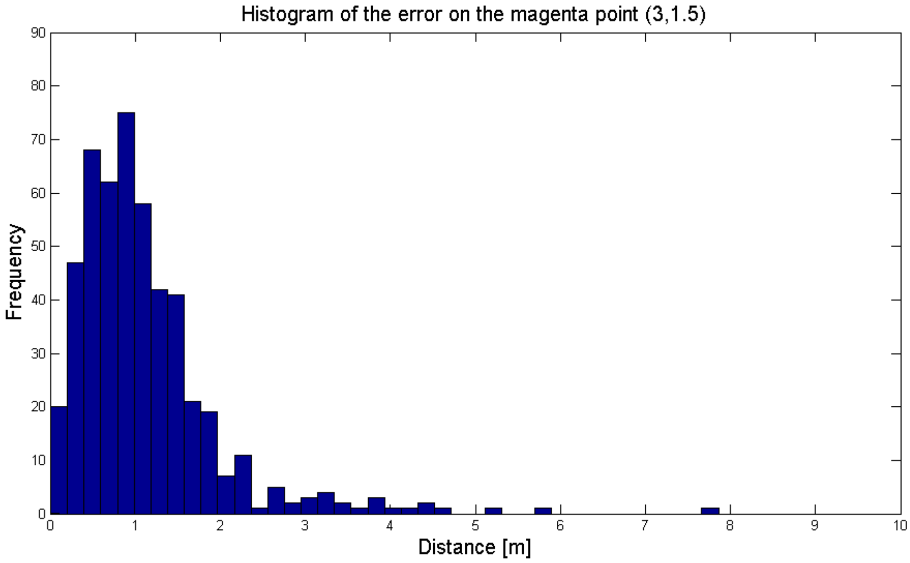


Figure 4.8: Histogram of the errors on the magenta point (21 ref. points).

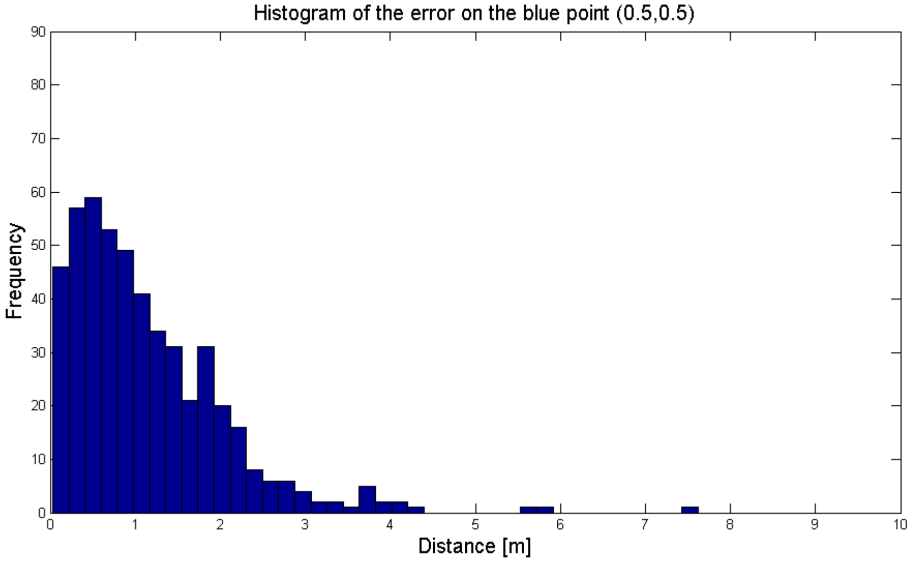


Figure 4.9: Histogram of the errors on the blue point (21 ref. points).

sians are wider than in the case of 34 reference points, thus revealing that the uncertainty in the determination of the position is higher. The same conclusion can be made by comparing the 3-D Gaussians (Figures 4.13 and 4.20). In the 34

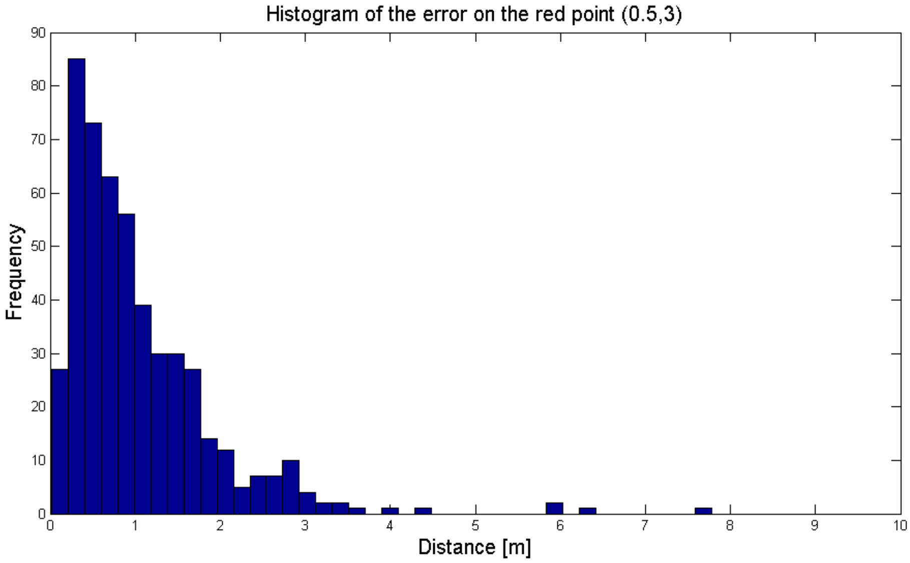


Figure 4.10: Histogram of the errors on the red point (21 ref. points).

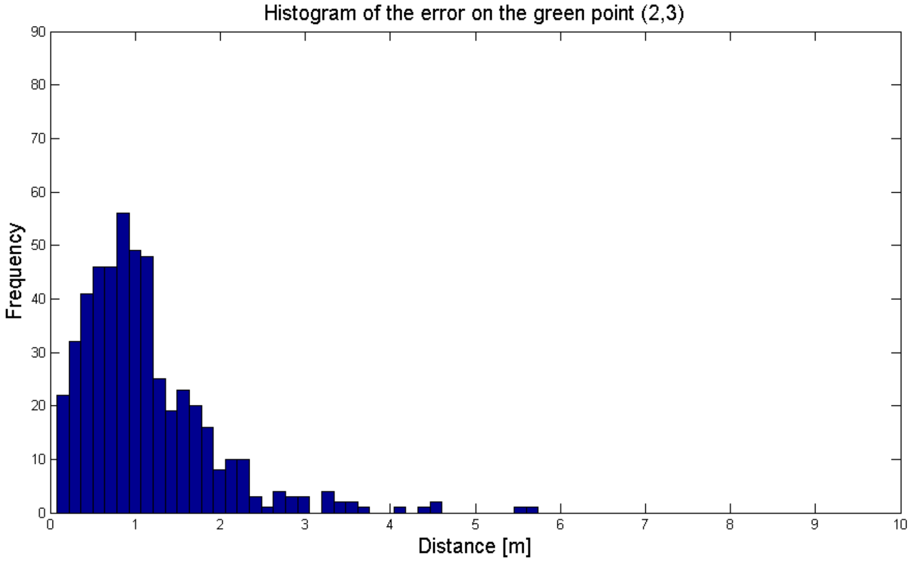


Figure 4.11: Histogram of the errors on the green point (21 ref. points).

reference points case, the 3D Gaussians are much higher, therefore indicating a greater concentration of the points around the estimated point.

In particular, Table 4.1 shows the performance results obtained. In this table

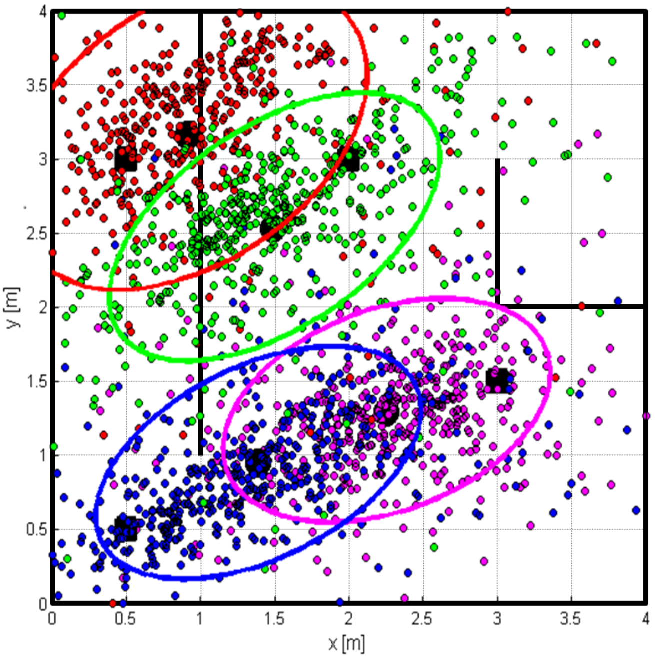


Figure 4.12: All the estimated points (21 ref. points).

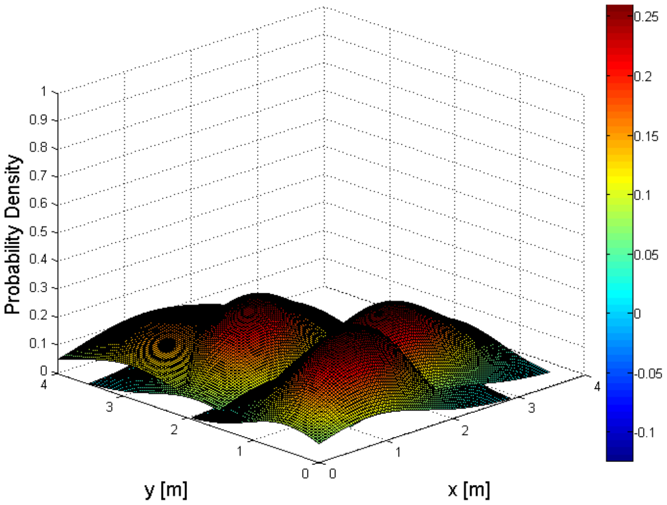


Figure 4.13: 3-D Gaussians (21 ref. points).

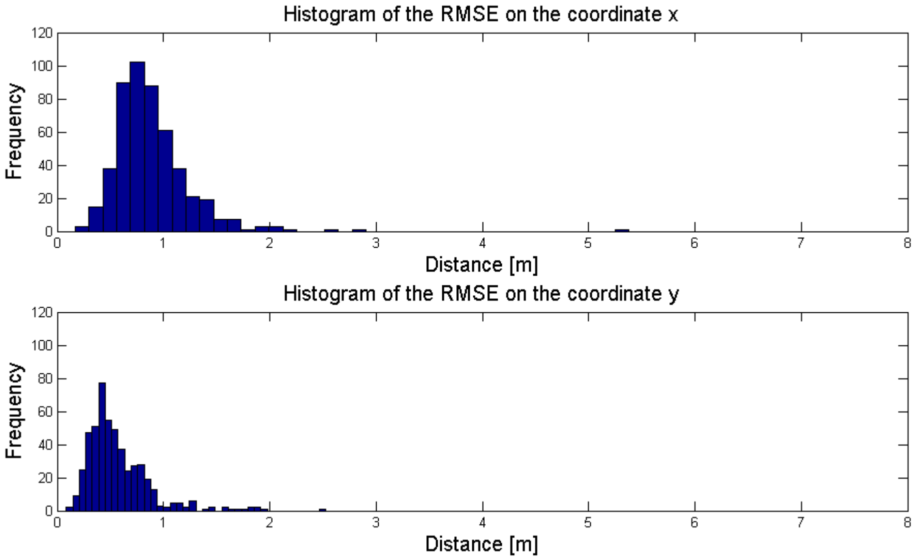


Figure 4.14: Histograms of RMSE on x and y coordinates (34 ref. points).

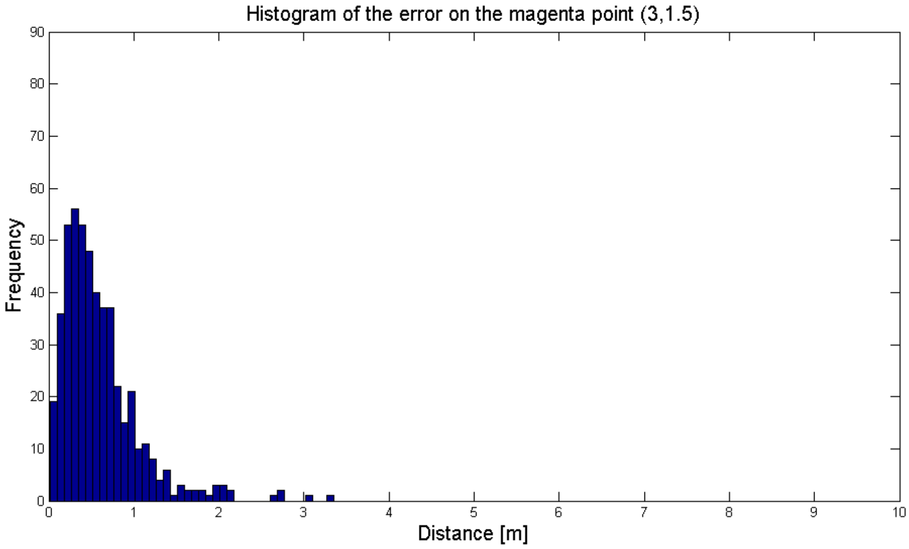


Figure 4.15: Histogram of the errors on the magenta point (34 ref. points).

the numeric values are given in meters.

The improvement with the introduction of more reference points are approximately of the order of 10 centimetres.

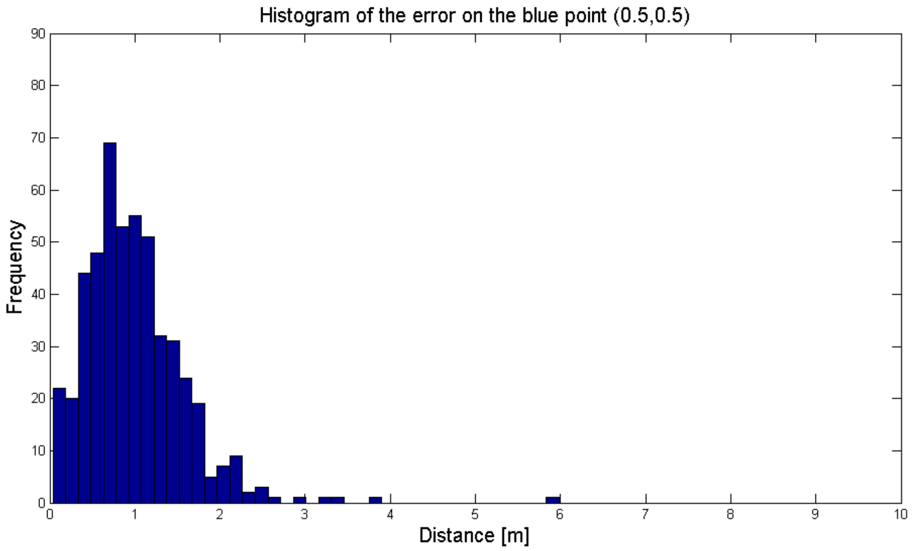


Figure 4.16: Histogram of the errors on the blue point (34 ref. points).

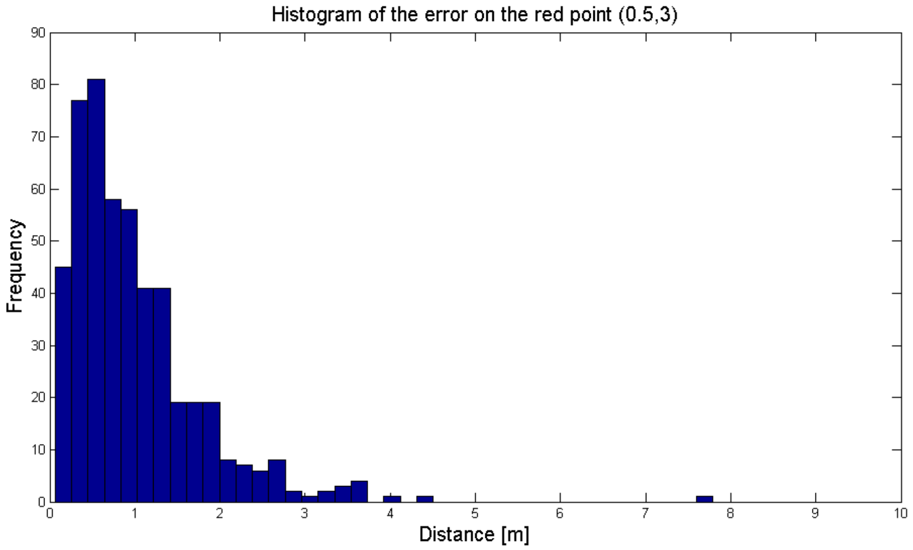


Figure 4.17: Histogram of the errors on the red point (34 ref. points).

It has to be noted the fact that, even increasing the number of reference points from 21 to 34 the real blue point is even outside the 2-D Gaussian border which is centred to the relative estimated point. The reason behind is the fact that

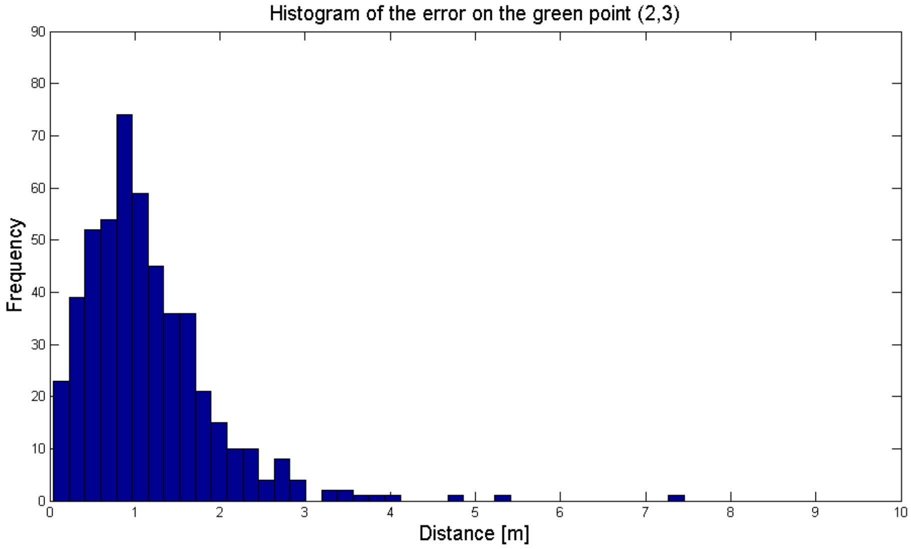


Figure 4.18: Histogram of the errors on the green point (34 ref. points).

the blue point is on the left border of a region of points having the same RSSI values and thus it is not a problem caused by the neural network implementation. Therefore, this incongruity shows the limits of RSSI fingerprinting methods, which are typically not very accurate.

4.3 Step Detection Algorithm Implementation

To validate the step detection algorithm and to check how it behaves in different situations, some walking behaviours have been set. As already stated before, the scenario considered is the second one.

The initial position has been set to the coordinate $(1.5, 0)$, while the walking path has been divided into three different parts. From the considerations of Chapter 3, sin-like acceleration variations have been set along the local frame z axis. In addition, through real measurements, it has been found that the same sin-like variations, even if of a lower magnitude, are registered also in the body

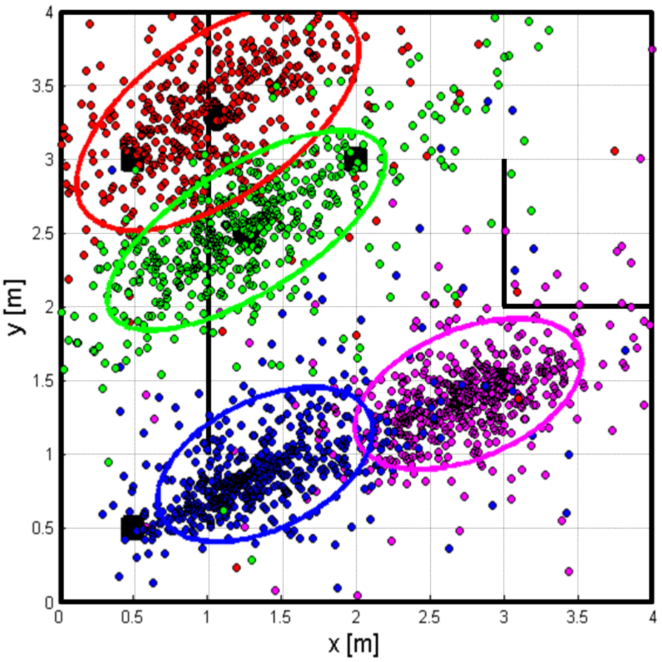


Figure 4.19: All the estimated points (34 ref. points).

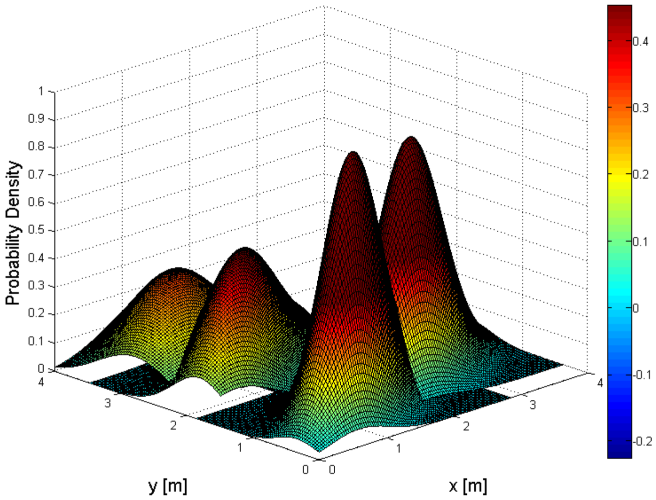


Figure 4.20: 3-D Gaussians (34 ref. points).

Quantity Investigated	**21p [m]**	**34p [m]**	**Improvement [m]**
RMSE on x coordinate	1.039	0.888	0.151
RMSE on y coordinate	0.691	0.568	0.122
Mean error on magenta point x	0.927	0.491	0.435
Mean error on magenta point y	0.464	0.292	0.171
Mean error on blue point x	0.981	0.910	0.071
Mean error on blue point y	0.546	0.454	0.092
Mean error on red point x	0.802	0.739	0.063
Mean error on red point y	0.639	0.522	0.117
Mean error on green point x	0.878	0.877	0.001
Mean error on green point y	0.646	0.601	0.044

Table 4.1: System performances on the basis of the two sets of reference points.

frame axis which is parallel to the direction of movement. These considerations are applied in the creation of walking models for the validation of the algorithm.

In the first part, the user walks towards north (along local frame axis y) at a very slow constant velocity. The length of the step is set to 1 metre. The mobile phone is placed horizontally so that the body frame axes x and y are parallel to the respective local frame axes, while body frame z axis points towards the sky. Figure 4.21 represents the situation, where in blue is indicated the local frame and in red the body frame.

The distance covered is 9 metres at a velocity of 0.318 m/s, thus requiring 29 seconds approximately. The accelerometer components along the body frame axes have been set to:

- 0, along the x axis,
- $2\sin(2t) + 2$, along the y axis, that is the direction along which the person

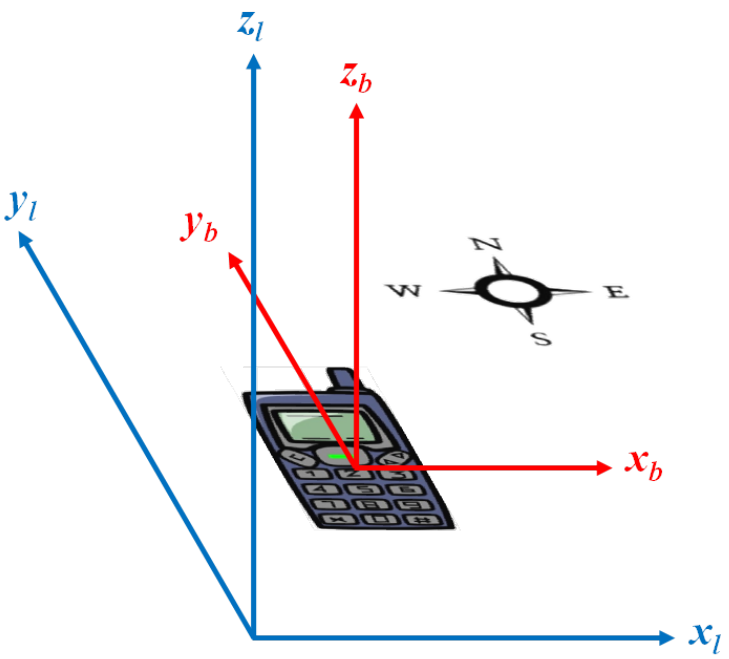


Figure 4.21: The smartphone orientation in the first part of the walking route.

is moving,

- $(5\sin(2t)) - 9.81$, along the z axis.

The gravimeter components have been set to:

- 0, along the x and y axes,
- -9.81, along the z axis.

The magnetometer values have been recorded by placing a smartphone device in the same orientation. These values are:

- 0.1, along the x axis,
- 23, along the y axis,
- -29, along the z axis.

All the gyroscope components have been set to 0.

In the second part, the user walks towards north-east, that is with an inclination of 45 degrees with respect to x axis. The mobile phone is rotated 90 degrees anticlockwise around z axis and it is further inclined of 45 degrees with respect to the ground plane. This represents the typical situation in which a person holds the smartphone perpendicularly with respect to the direction of movement. Figure 4.22 represents the situation.

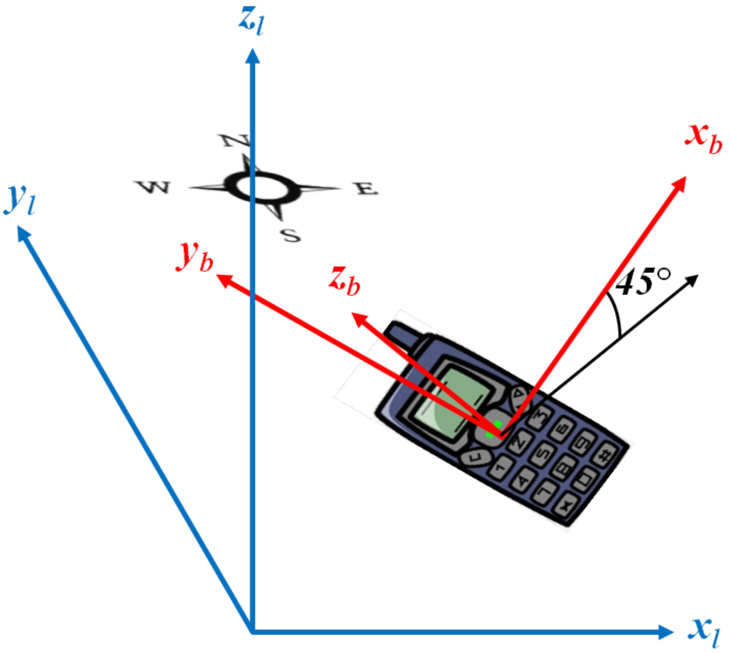


Figure 4.22: The smartphone orientation in the second part of the walking route.

To better show the actual body frame axes orientation, Figure 4.23 gives a lateral view of the axes, being parallel to north-east direction.

The distance covered is approximately 16 metres at a velocity of 1.432 m/s, thus requiring 12 seconds approximately. The length of each step is again set to 1 metre. The accelerometer components are now set to:

- $(3\sin(8.8t) + 3 - 9.81 + \sin(8.8t))/\sqrt{2}$, along the x and z axes,

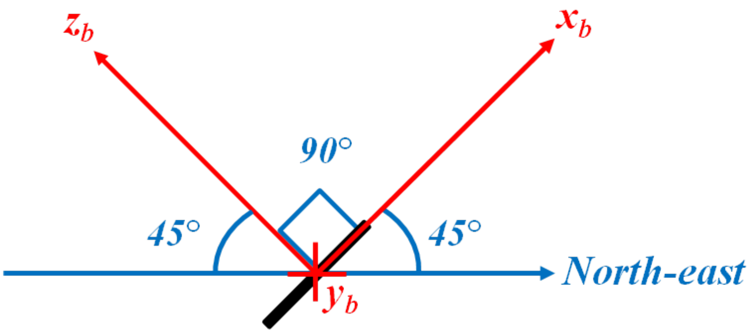


Figure 4.23: Lateral view of the axes, along north-east direction.

- 0, along the y axis,

The gravimeter components have been set to:

- $-9.81/\sqrt{2}$, along the x and z axes,
- 0, along the y axis.

The magnetometer components have been set to:

- -36, along the x axis,
- -2.522, along the y axis,
- -55, along the z axis.

All the gyroscope components have been set to 0.

In the third part the user walks towards south, being placed vertically, as if it was in a pocket. Body frame x axis is parallel to $-y$ local frame axis, body frame y axis points towards the sky while body frame z axis is parallel to $-x$ local frame axis. Figure 4.24 represents the situation.

The distance covered is equal to 12 metres at a velocity of 2 m/s, thus requiring 6 seconds. The length of each step is set to 1 metre. The accelerometer components are now set to:

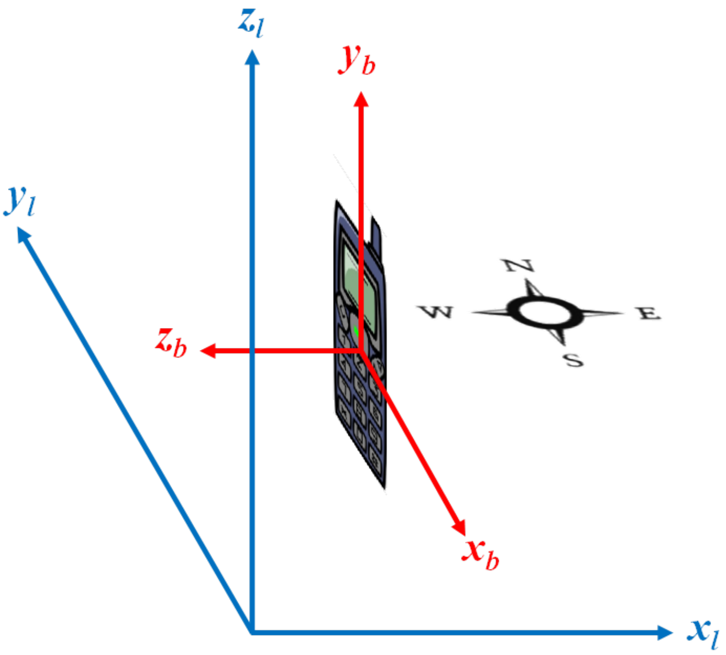


Figure 4.24: The smartphone orientation in the third part of the walking route.

- $\sin(4\pi t) + 1$, along the x axis,
- $3\sin(4\pi t) - 9.81$, along the y axis,
- 0, along the z axis.

The gravimeter components have been set to:

- 0, along the x and z axes,
- -9.81, along the y axis.

The magnetometer components are:

- -21, along the x axis,
- 43, along the y axis,
- 0.09, along the z axis.

Again, all the gyroscope components have been set to 0.

Once set the parameters of the walking model, the sensor's algorithm has been validated. Figure 4.25 shows the theoretical path (in red) compared to the one as estimated by the sensors (the blue dots). The blue dots actually specify the positions where a step is estimated, while the red crosses represent the theoretic positions of steps.

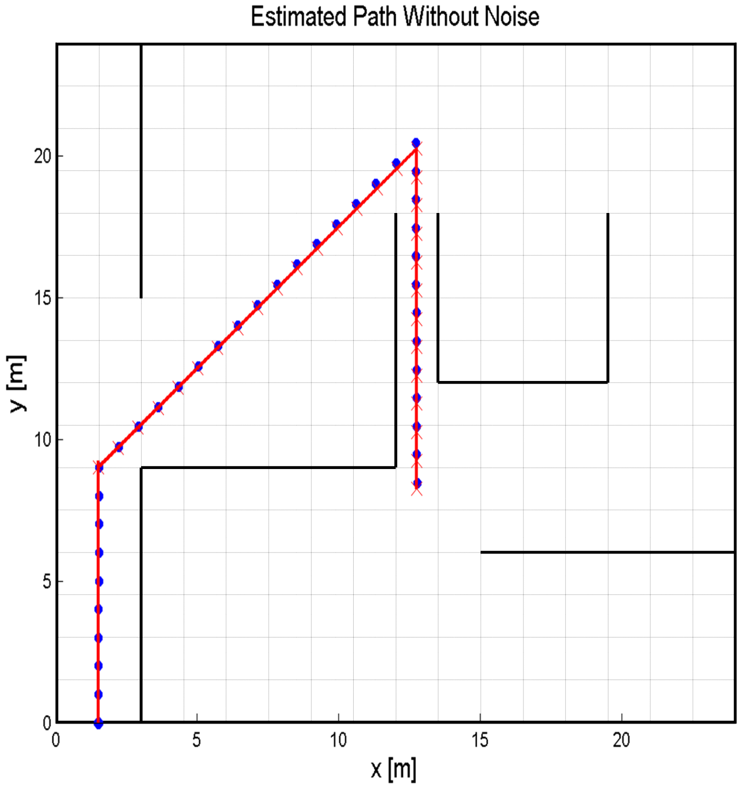


Figure 4.25: Estimated path without noise.

In addition, Gaussian noise on the measurements of sensors has been added to verify what the system performances would be. This is represented in Figure 4.26. Error is added to the step length estimation and to all the measurements from sensors, with the only exception of gravimeter, which is particularly sensitive to noise. The MATLAB® function used is *randn*. The figure shows the results from

200 simulations.

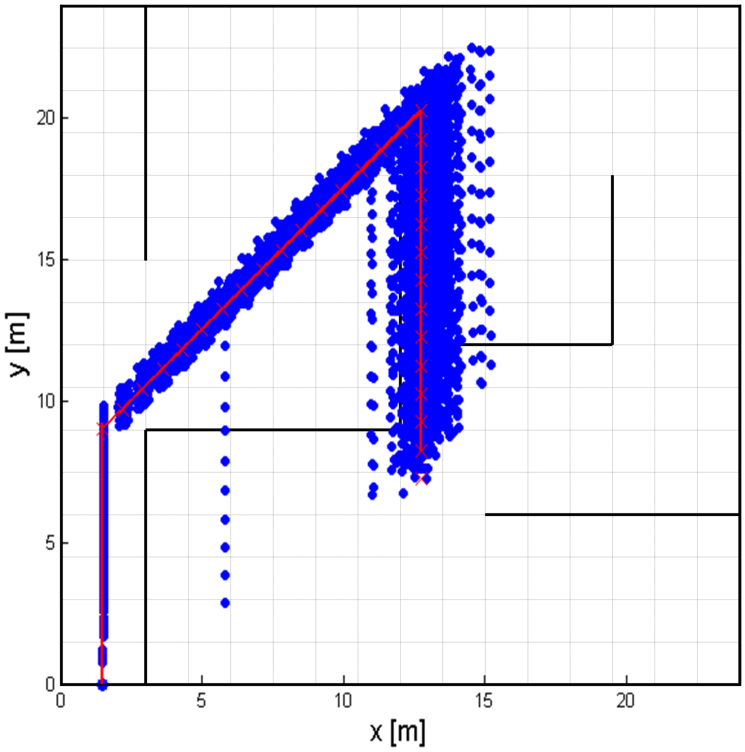


Figure 4.26: Estimated path with noise.

As it can be seen the sensors algorithm is very accurate without adding noise, but it can be quite inaccurate when adding noise. Table 4.2 shows the performances of the sensors algorithm when Gaussian noise is applied. RMSE stands for Root Mean Square Error.

Quantity Considered	**Error [m]**
Mean error on the final position estimation	2.021117
Mean error on coord. x, second part of the route	0.2143320
Mean error on coord. y, second part of the route	0.4036245
RMSE on coord. x, second part of the route	0.2519698
RMSE on coord. y, second part of the route	0.4535147
Mean error on coord. x, third part of the route	1.734861
Mean error on coord. y, third part of the route	2.154782
RMSE on coord. x, third part of the route	2.200567
RMSE on coord. y, third part of the route	2.504078

Table 4.2: Sensors algorithm performances in the presence of Gaussian noise.

4.4 Particle Filter Implementation

4.4.1 Particle Filter Initialization and Update

As stated in the previous Chapter 3, the state of the system at every time step is given only by a noisy measurement of it. In this case, this measurement is given by RSSI while the unobservable state is given by the motion model, as recorded by the sensors. Thus, the particle filter's main purpose is to fuse the two information in the best way.

First of all, the set of N particles is initialised from the user's initial position and they all have the same weight, that is the same importance.

Then, at every time in which a step is detected by the step detection algorithm, the new positions are updated on the basis of the motion characteristics, as recorded by the sensors. The trajectory and the distance covered by every particles is different from another one because a different Gaussian noise realisation is added to every particle. In particular, the heading and the distance covered by every particle i are estimated respectively by the following MATLAB® equations:

$$heading^i = heading + randn(1) \tag{4.1}$$

$$stepLength^i = stepLength + randn(1) \tag{4.2}$$

where *heading* and *stepLength* are estimated through the previous algorithms. Thus, the new positions $(xPup^i, yPup^i)$ of the i-th particle is given by:

$$xPup^i = xP^i + stepLength^i \cos\left(heading^i\right) \tag{4.3}$$

$$yPup^i = xP^i + stepLength^i \sin\left(heading^i\right) \tag{4.4}$$

where (xP^i, yP^i) represent the positions of the particle at the previous step.

On the same time, the sensors algorithm return the position $(Xrssi, Yrssi)$ as estimated by the RSSI. This position represents the noisy measurements of the system:

$$xZup^i = Xrssi + randn(1) \tag{4.5}$$

$$yZup^i = Yrssi + randn(1) \tag{4.6}$$

where $(xZup^i, yZup^i)$ is the new position of the particle.

4.4.2 Map Filtering and Weights Update

After the new positions of the particles are computed, it is necessary to understand if every particle moving from its previous position to its last position has passed through a wall or not. For this purpose, a matrix composed by the coefficients of the lines which include the path and the wall is created. Then, the Rouché-Capelli Theorem is applied to this matrix to find if there is an intersection point. If this is the case, the solution is found by applying Cramer's rule. Thus, any intersection between the lines which include the wall and the path segments are found. Since in reality the aim is to verify if there are intersections between segments and not lines, one has to check also if the intersection is in the region of interest, that is if the intersection is within a rectangle area whose limits are given by the wall's and the path's coordinates.

This algorithm is applied for every particle and for every wall. If the intersection is actually in the region of interest, the weight of the particle is set equal to zero. Otherwise, if the particle does not pass through any wall, its weight is updated by the following relation:

$$\omega^i = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(xPup^i - xZup^i)^2 + (yPup^i - yZup^i)^2}{2\sigma^2}\right) \tag{4.7}$$

Then, a new set of particles is created, by selecting those ones with the greatest weights. Finally, the state is estimated as the mean of the vector of the resampled particles.

4.4.3 Simulation Results

4.4.3.1 Details on the Particle Filter Operation

Figure 4.27 shows details of the particles' update process considering only 5 particles. The blue points are the updates of the previous particles at every step, while the red points are the particles which are actually selected at every step after the resampling process. At the following step, only these red points are considered for the creation of new particles. The blue lines connect every particle with its update. The filter has to prevent that the particles cross walls.

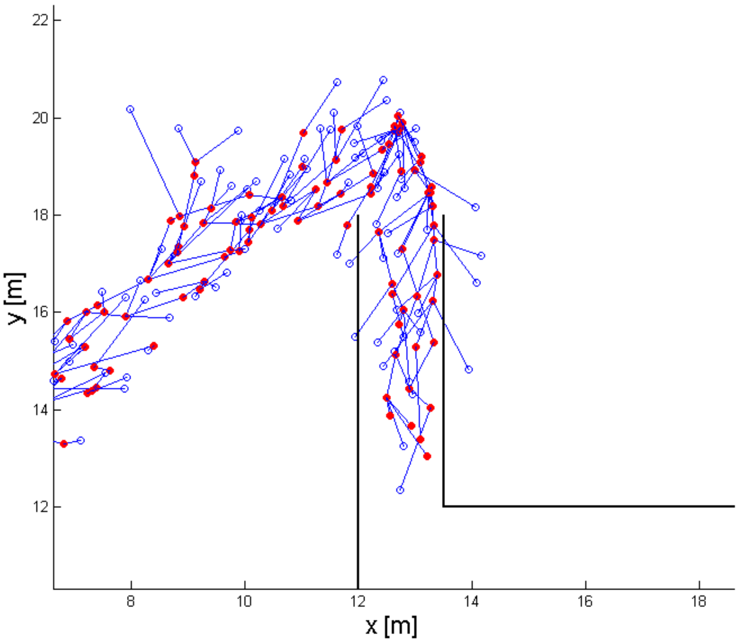


Figure 4.27: Details of the particle update process.

Figure 4.28 shows how the filter prevent particles from crossing walls. 50 particles have been considered and no noise is added. A new wall is added to interrupt the theoretical path. As it can be seen, the particles are not able to follow the theoretical path anymore. Figure 4.29 shows the positions as estimated by the neural network (black triangles), the sensors algorithm (the blue dots) and the particle filter (the green diamonds).

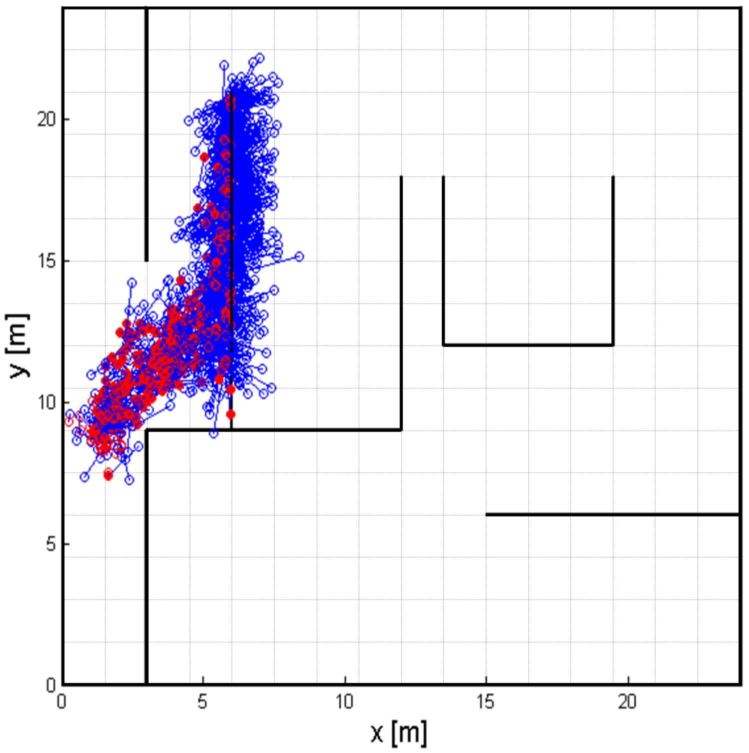


Figure 4.28: The particle filter outputs when adding a new wall.

4.4.3.2 Simulations Without Noise

In this paragraph, simulations without noise on the measurements from sensors are shown. Since in real situations there is always some noise, these simulations are actually not very significant. Anyway, they give an idea about the system performances and the fact that the particle filter gives actually better estimates than using only RSSI measurements.

Figure 4.30 shows the update process of the particles in the absence of Gaussian noise. 200 particles and only one simulation are considered. Figure 4.31

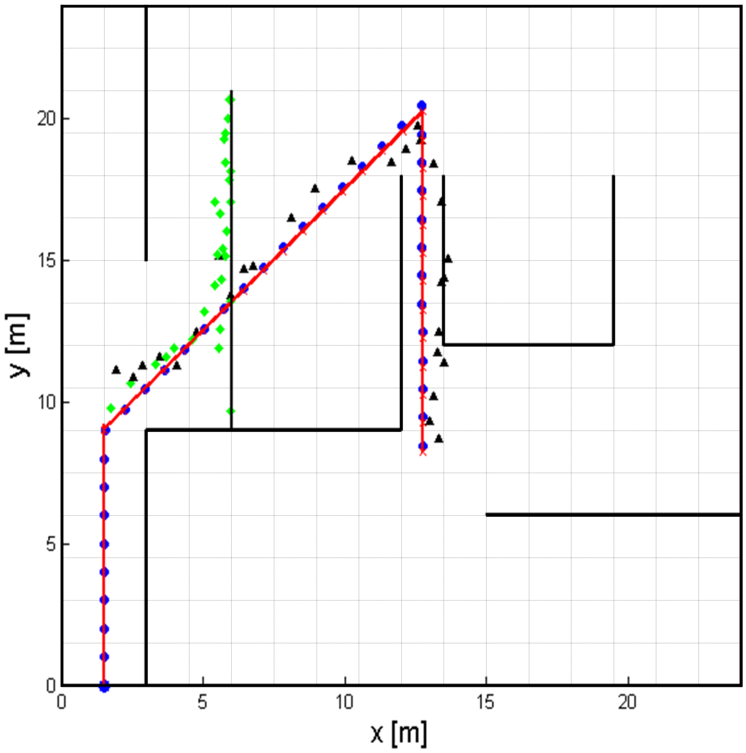


Figure 4.29: Positions estimated by the various methods.

shows the estimated positions.

Table 4.3 shows the performances obtained, where INS stands for Inertial Navigation System .

Figure 4.32 represents the case of 200 particles and 50 simulations. No noise is considered. Table 4.4 compares the performances of the methods.

4.4.3.3 Simulations With Noise

Figure 4.33 shows the simulation results in the presence of Gaussian noise both on the measurements from sensors and on the RSSI values. 200 particles and 50 simulations are considered.

The improvements are shown in Table 4.5. It is important to note that, in

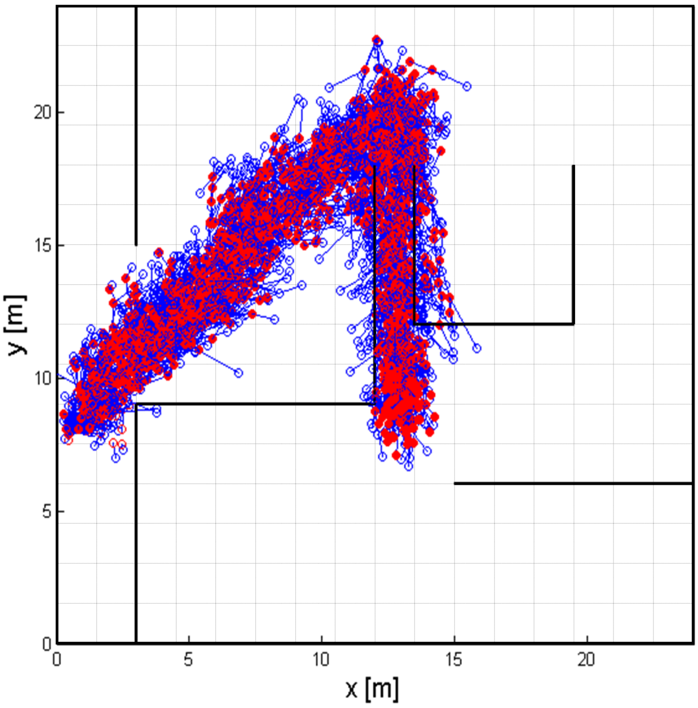


Figure 4.30: The update process of the particles (no noise).

order to focus on the relevant values, two estimations, one with the highest error and one with the lowest error, have been removed from the computation of the mean errors.

Figures 4.34, 4.35, 4.36, 4.37 represent graphically the errors committed.

Figures 4.34 and 4.35 represent the absolute errors in metres on the coordinates x and y respectively. Thus, the values reported are evaluated as means of absolute errors, which are computed as the absolute value of the differences between the estimated and the real values. It is important to note an increase of the error in all the measurements from step 16. This happens because at step 16, there is a strong discontinuity in the measurements from sensors, which cause problems with the estimations.

Figures 4.36 and 4.37 represent the percent errors in metres on the coordinates

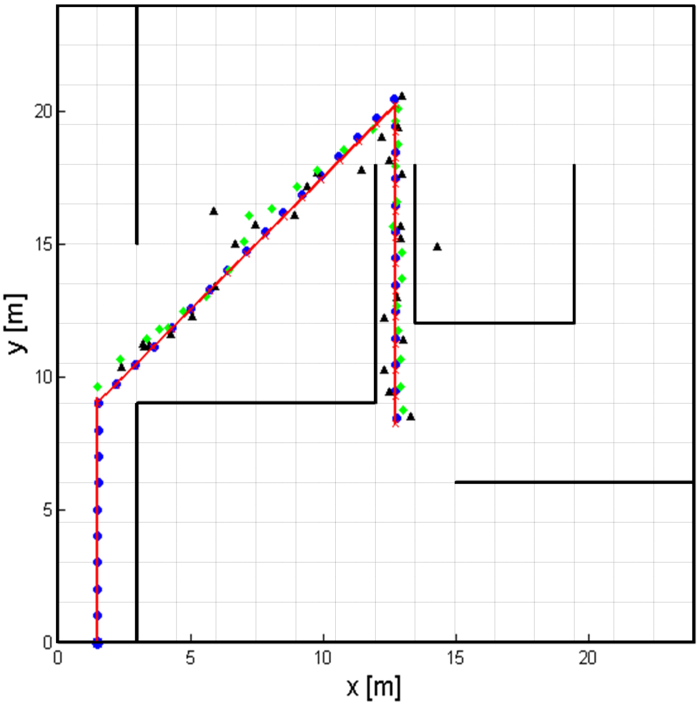


Figure 4.31: Results of 1 simulation with 200 particles (no noise).

x and y respectively. These are evaluated as absolute values of the differences between the estimated and the real values, divided by the real values and multiplied by 100. The errors represent thus percentages, so, for every step, one has the difference between the approximate and the exact values as a percentage of the exact value itself. This helps to see how close the estimate is to a real value, without giving a numeric value but a significant percentage reference.

In both cases, the errors committed by the particle filter are smaller than those committed by taking into account only RSSI values. Thus, it is proved that the proposed system improves the performances of a RSSI-based localisation system.

Quantity Investigated	INS [m]	RSSI [m]	Filter [m]
Mean error on the final position	0.203	0.541	0.444
Mean error on x, 2^{nd} part of the route	0.019	0.682	0.529
Mean error on y, 2^{nd} part of the route	0.107	0.764	0.595
RMSE on x, 2^{nd} part of the route	0.022	0.899	0.630
RMSE on y, 2^{nd} part of the route	0.122	1.028	0.693
Mean error on x, 3^{rd} part of the route	0.014	0.453	0.157
Mean error on y, 3^{rd} part of the route	0.202	0.301	0.375
RMSE on x, 3^{rd} part of the route	0.018	0.514	0.201
RMSE on y, 3^{rd} part of the route	0.202	0.448	0.520

Table 4.3: Performance comparison among the different methods (1 simulation, 200 particles, no noise).

4.4.4 Floating-Point Arithmetic Issue

While testing the algorithm, at the beginning, few particles were able to overstep walls even if they could not do it in theory. Many simulations have been done to find the solution of this problem, which was not caused by the particle filter algorithm. In Figure 4.38 it is possible to see some of the 100 particles used in the simulation crossing walls, even if it should not be possible.

As shown in this figure, the algorithm sometimes is not able to verify that the intersection of a wall with a particle is or not inside the region of interest. Just to give an example, it seemed that in some occasions, for MATLAB® the identity $6 = 6$ was not true anymore. This problem is due to the floating-point arithmetic used by any software and thus also by MATLAB®. By using floating-point arithmetic, software have to approximate every real number with a finite number of digits. MATLAB®, in particular, has 16 significant digits to represent any number and thus it cannot represent all numbers exactly, because it has to

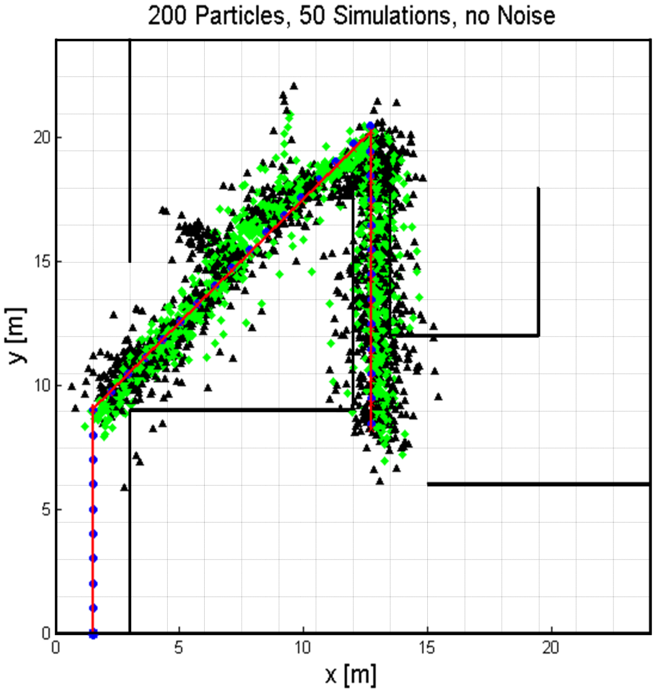


Figure 4.32: Results of 50 simulations with 200 particles (no noise).

approximate them. The solution to avoid this issue is basically not to check if a number is equal to another, but, for example, to check if the absolute value of their difference is smaller than a very small quantity, like for example $1\exp(-8)$.

Quantity Investigated	INS [m]	RSSI [m]	Filter [m]
Mean error on the final position	0.240	0.797	0.695
Mean error on x, 2^{nd} part of the route	0.019	0.832	0.748
Mean error on y, 2^{nd} part of the route	0.107	0.668	0.656
RMSE on x, 2^{nd} part of the route	0.022	1.067	0.870
RMSE on y, 2^{nd} part of the route	0.122	0.818	0.735
Mean error on x, 3^{rd} part of the route	0.015	0.585	0.376
Mean error on y, 3^{rd} part of the route	0.202	0.543	0.462
RMSE on x, 3^{rd} part of the route	0.018	0.706	0.424
RMSE on y, 3^{rd} part of the route	0.202	0.667	0.538

Table 4.4: Performance comparison among the different methods (50 simulation, 200 particles, no noise).

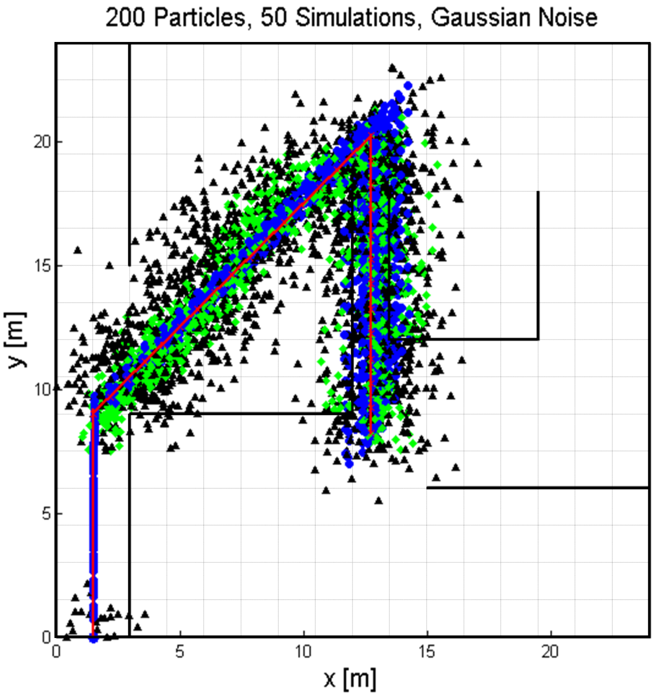


Figure 4.33: Results of 50 simulations with 200 particles (with noise).

Quantity Investigated	INS [m]	RSSI [m]	Filter [m]
Mean error on the final position	1.223	1.714	1.081
Mean error on x, 2^{nd} part of the route	0.164	1.247	0.890
Mean error on y, 2^{nd} part of the route	0.298	1.168	0.763
RMSE on x, 2^{nd} part of the route	0.194	1.537	1.039
RMSE on y, 2^{nd} part of the route	0.330	1.401	0.897
Mean error on x, 3^{rd} part of the route	1.053	1.542	1.203
Mean error on y, 3^{rd} part of the route	1.423	1.567	1.259
RMSE on x, 3^{rd} part of the route	1.354	2.020	1.598
RMSE on y, 3^{rd} part of the route	1.614	1.950	1.562

Table 4.5: Performance comparison among the different methods (50 simulations, 200 particles, noise present).

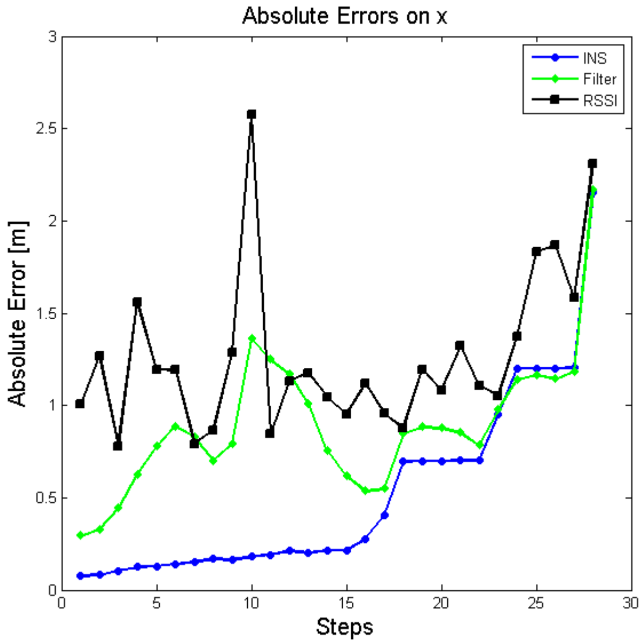


Figure 4.34: Absolute error on x (50 simulations, 200 particles, Gaussian noise).

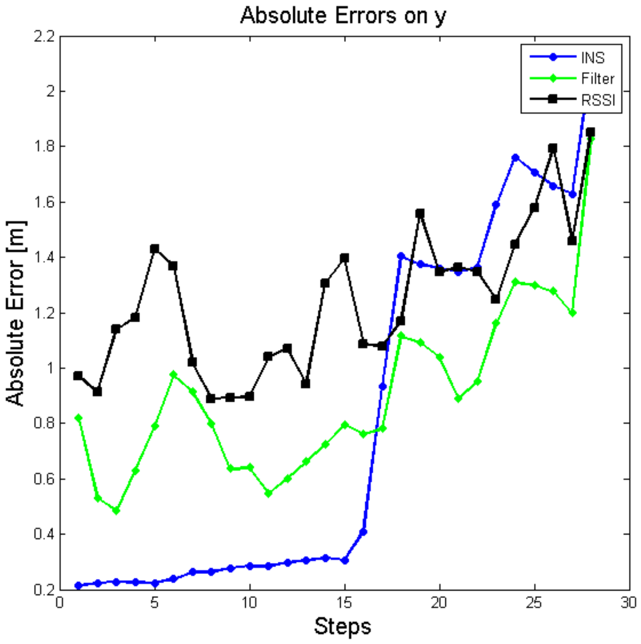


Figure 4.35: Absolute error on y (50 simulations, 200 particles, Gaussian noise).

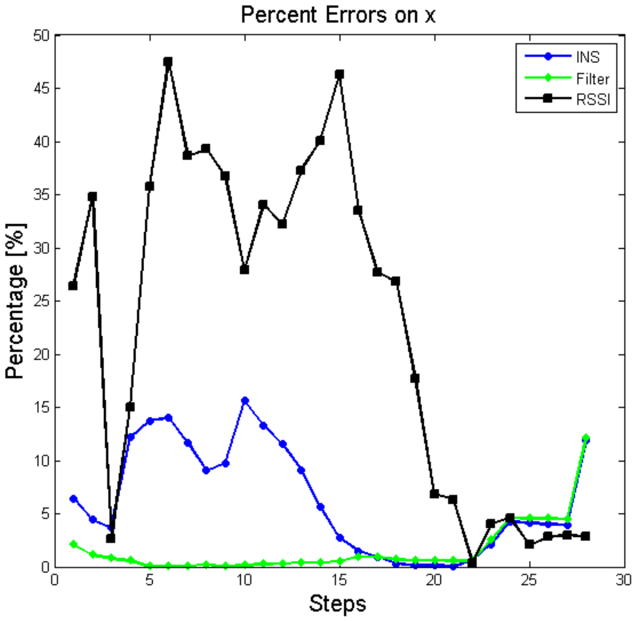


Figure 4.36: Percent errors on x (50 simulations, 200 particles, Gaussian noise).

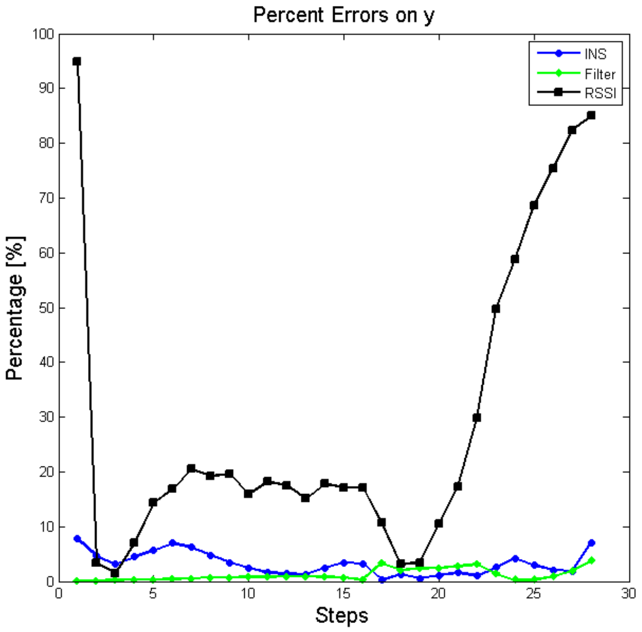


Figure 4.37: Percent errors on y (50 simulations, 200 particles, Gaussian noise).

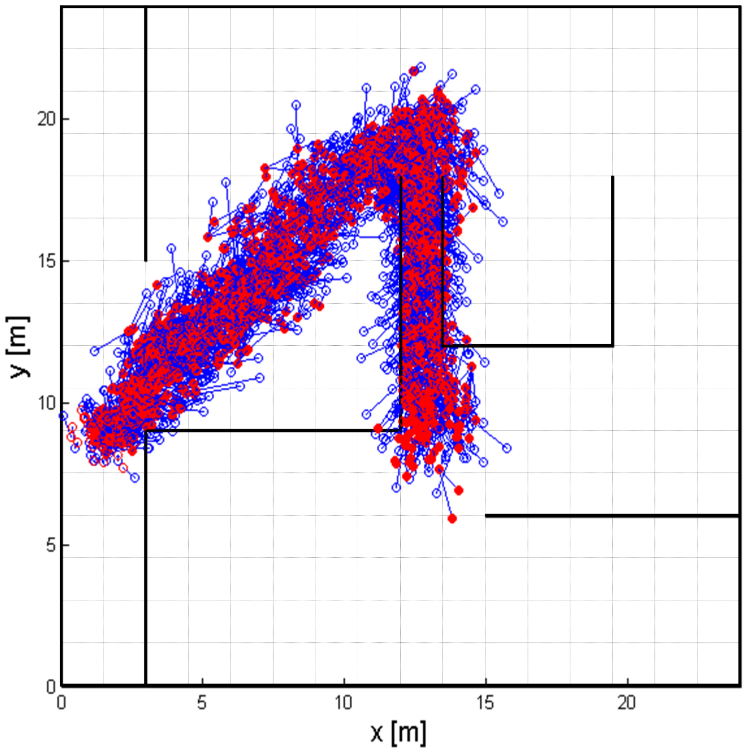


Figure 4.38: Some particles are able to cross walls, due to floating-point arithmetic issues.

Chapter 5

Indoor Localisation Through Visible Light Communications

The contents of this chapter have been written during the Short-Term Scientific Mission at the Imperial College of Science, Technology and Medicine, London, in the period June-July 2013, within the framework of COST Action IC0902 and under the supervision of Dr. Deniz Gündüz.

Visible Light Communication (VLC) is the technology which uses light that is visible to humans for communication purposes. In recent years, illumination provided by light emitting diodes (LEDs) has become more important because of its advantages compared to the traditional lightning. These are in particular low power consumption, extremely long lifetime, robustness, environmental friendliness, low voltage operation, flicker free, smooth dimming, reduced maintenance, good colour-rendering property, smaller size and cooler operation (36, 41). This is the reason why it is clear that LEDs will be the future desired lightning sources. The contemporary use of LEDs for illumination, localisation and communication purposes explain the growing interest in this field. Some advantages deriving from the use of this technology for localisation are (53):

- visible light is free and so there are not fees or royalties to be paid since there are no regulations such as the radio communication systems.
- VLC LEDs will be used everywhere, which makes visible light LEDs ideal for ubiquitous data transmission.
- since there is no interference with electronic devices, visible light can be used in hospitals and other areas like airplanes without problems.
- VLC is harmless for human health, since it does not emit electromagnetic radiation.

The primary aim of this chapter is to propose a VLC indoor joint illumination and localisation system. The proposed localisation method is based on the collection of power measurements from transmitting LEDs (RSS). It is an anchor-based method, where the anchors are the LEDs installed typically on the ceiling of a room. The anchors' position is known a priori and they form a regular (normally rectangular) grid pattern. This system can fully cooperate with the localisation system proposed in the previous chapters and it can be a substitute in case of low WLAN signal coverage.

5.1 State-of-the-Art

In the last years many works have proposed different localisation systems based on VLC. In (37), Cossu *et al.* propose a localisation and communication system based on RGB LEDs which adopts adaptive OFDM model. Through experiments, they have demonstrated that their system is fully compatible with the requirements of line-of-sight VLC system. They guarantee a data-rate between 300 and 400 Mb/s within an illumination angle of 90°, at a distance of 90 cm, while the angular resolution is smaller than 1°. Each reference node transmits a unique identifier, represented by a single tone located at frequencies lower than 700 KHz so that there is no overlap with the data-rate signal. In (40), Jung *et al.* use the time difference of arrival (TDOA) localisation algorithm and the performances are evaluated by computer simulation. Since LEDs can modulate signals while illuminating, each LED has a unique assigned frequency address. Several band-pass filters at the receiver side are able to detect each frequency. The location accuracy is less than 1 cm in the space of 5m x 5 m x 3 m. Despite the excellent results provided, one of the most important drawbacks of TDOA method is the requirement of perfect synchronisation between transmitters and receivers. In (47), the authors propose a fingerprinting localisation method through infrared LED devices, which estimates the distance through the information on impulse responses received by the various transmitters. Zhou *et al.* (60) propose a method which recovers the channel characteristics and solves the Lambertian equation group analytically. The LEDs transmit on the basis of time division multiplexing (TDM), so that in every time slot just one transmitter can send its ID encoded into three 16-bit RZ codes. Other authors propose hybrid methods, which exploit other technologies to provide better results. In (39), Jia merges a range-free VLC based positioning method with optical receive device with a conventional received signal strength indication (RSSI) based positioning method.

5.2 Operating Principle

5.2.1 System Model

The indoor area is divided into regions covered by only one single LED. Actually, more LEDs for illumination purposes can be used in the same area but only one of them is considered to be used for localisation, so that each one has its own coverage area, without interfering with other LEDs. Figure 5.1, represents the system model. The LEDs are disposed in a geometric layout.

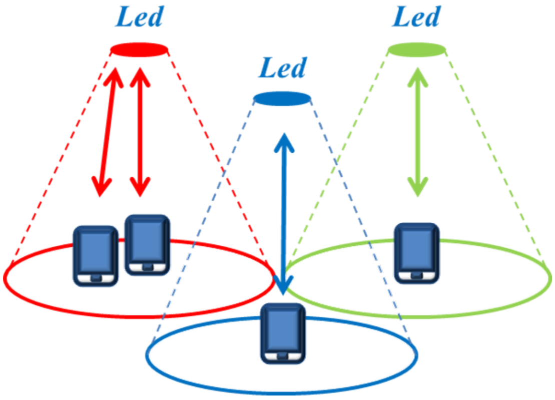


Figure 5.1: Example of LED installation.

Radiation from LEDs follow a Lambertian pattern, so that the impulse response from LED_i to the mobile terminal is:

$$h_i(t) = \begin{cases} \frac{n+1}{2\pi} \frac{cos^n(\phi_i) cos(\theta_i) A_R}{R_i^2} \delta(t - \frac{R_i}{c}) & \text{if } \theta \leq FoV \\ 0 & \text{if } \theta > FoV \end{cases} \tag{5.1}$$

where:

- n is the mode number of the radiation lobe, which states the source directionality,
- ϕ_i is the angle between the vector which represents the orientation of the source and the vector which points from source to receiver,
- θ_i is the angle between the vector which represents the orientation of the receiver and the vector which points from receiver to source,
- A_R is the receiver area,
- FoV is the field-of-view of the receiver,

- R_i is the distance between the source and receiver,
- c is the speed of light.

LEDs are considered to be of first-order Lambertian pattern ($n = 1$), which is the most common case while the receiving angle is always smaller than FoV. Moreover, the difference of impulse delay is ignored since it of nanosecond scale while the system works on really small distances (60). The gains of the optical filter and optical concentrator are assumed to be 1 for simplicity. To better understand the model, the reference system is illustrated in Figure 5.2.

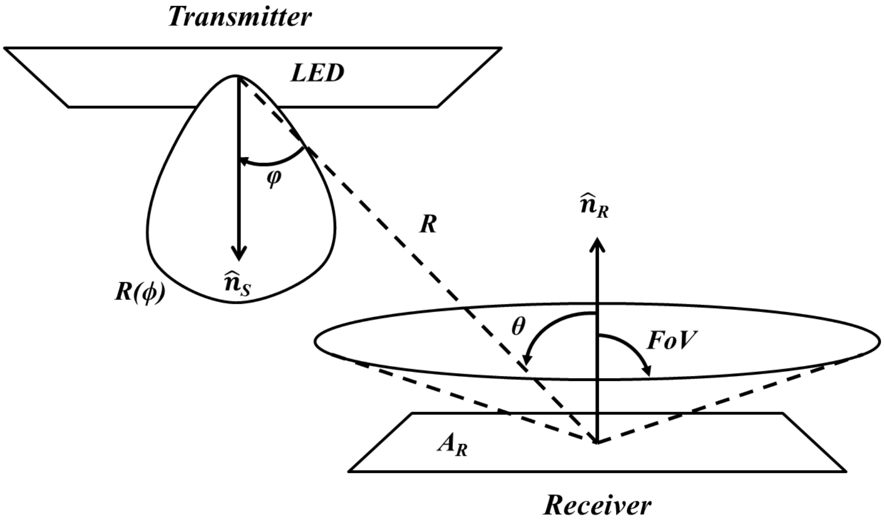


Figure 5.2: Geometry of source and detector, without reflectors.

Received energy is calculated through the signal integration at the receiver side. For every transmission only the highest peak, corresponding to the LOS path, is considered. In (42), the authors show that, considering a Lambertian pattern for the reflections and in the case of visible light communications, the peak is generally much higher than those caused by the multipath. Therefore, in first analysis it is possible to limit the integration time to the time necessary to acquire only the first peak of the signal.

5.2.2 LED Dimming and Flicker

First of all, it is important to distinguish between LED dimming and flickering. LED flickering is the modulation of light output (luminous flux) and all commercial electric light sources running on AC power modulate light output (44). It can be harmful, causing even important health effects like neurological problems, epileptic seizure headaches, fatigue, blurred vision, eyestrain, apparent slowing or stopping of motion, reduced visual task performance and distraction.

LED dimming is the variation of a LED light brightness for the convenience of illumination for working or studying. In addition, dimming of LED light can also save a lot of energy, when just a little light is needed. Dimming is necessary in RGB LED lamps to change the light colour, because the colour is controlled by adjusting the relative intensities of primary LEDs in the lamp. Traditionally there are two possible ways of dimming LEDs: the Amplitude Modulation (AM) and the Pulse Width Modulation (PWM). In the pulse width modulation scheme, the diode is turned on and off with fixed frequency and variable duty cycle and therefore the brightness of a LED is adjusted by varying the duty cycle without changing the LED current. Amplitude modulation uses the variable DC current to directly dim the LED. However, at lower current concentrations, LEDâĂŹs efficacy tends to increase and therefore the AM dimming is not linear. This is the reason why PWM dimming is typically used by the industry. Recent developments (49) show that combining these methods and driving the diodes with different current shapes can have significant difference on spectral and power characteristics of the diode. Hybrid PWM/AM uses variable peak current and variable duty cycle to control the average forward current. The different LED dimming schemes are depicted in Figure 5.3.

It is thus clear that also the dimming has to be taken into account in designing the LED positioning system, since dimming results in a lower emission of energy.

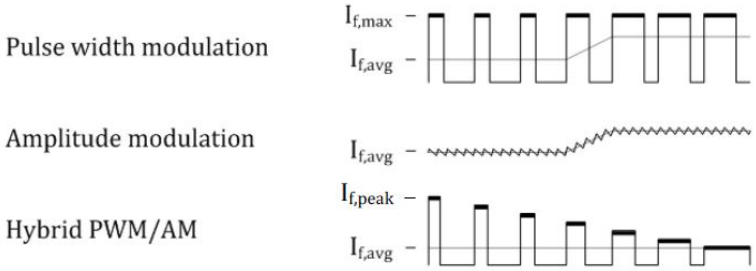


Figure 5.3: LED dimming schemes. $I_{f,max}$ is the maximum forward current of a LED.

As it will be clearer later, lowering the level of transmitted energy implies a greater variance and therefore a greater level of uncertainty about the channel characteristics. This higher uncertainty about the channel characteristics reflects on a greater error on the distance estimation and if the accuracy is too low, the system is not any more able to guarantee the performances required.

5.2.3 System Requirements

Insufficient refresh rates cause flicker effect and uncomfortable perception to human eyes. Therefore pulses sent by LEDs have to be at high frequencies in order to be perceived as a continuous luminous flux by human eyes. As stated by the Ferry-Porter theory and confirmed by people's experience, the refresh rate of a LED display should not be less than 240Hz (46). Here for the sake of simplicity, a limit frequency of 200 Hz is considered.

The aim is to find the minimum update rate of localisation information in order to efficiently track the movement of a person. So, supposing a maximum velocity of 7 km/h (1.944 m/s) and a position update of at least every 10 cm, one obtains:

$$T_f = \frac{0.1}{1.944} \simeq 0.05 \quad s \tag{5.2}$$

Therefore, in order to have the desired tracking properties, one has to update the location information at least every $T_f = 50ms$. Moreover, one has to take into account also the condition on the minimum dimming frequency, obtaining the minimum number a LED has to be switched on and off in the T_f period, as computed in Equation 5.3:

$$N \geq 0.05 * 200 = 10 \quad times \tag{5.3}$$

Equation 5.3 states that in every T_f time period, if the LED is not always on, it has to flicker for at least 10 times, in order not to cause problems for human eyes. This means that if the LED has to transmit just a little amount of power, it cannot transmit it in just one time remaining switched off for the remaining time, but it has to properly distribute it in the time period in order to make the light appear continuous to human eyes.

5.2.4 Problem Formulation and Analysis

1-D situation is considered. In this simple situation the angles, the transmitter height and the distance are all related.

The signal is transmitted by the VLC transmitter LED, it is sent over a channel and it is corrupted by white Gaussian noise before arriving to the mobile terminal, that needs to be located. The purpose of the receiver is to estimate the channel characteristic parameter H in the presence of additive noise. From the Lambertian Equation 5.1 it is possible to estimate the distance from the transmitter to the receiver. The received waveform can be expressed in a general form as (52):

$$r(t) = s(t,h) + w(t), \quad 0 \leq t \leq T, \tag{5.4}$$

where T is the integration time necessary to compute the received energy, $w(t)$ is a sample noise function from a white Gaussian noise process with spectral height $N_0/2$, $s(t)$ is a known input signal and h is the parameter that expresses the quantity that needs to be estimated. From h, it is possible to compute the distance of the mobile terminal from the transmitter. h is a random variable, whose a priori density is assumed to be known, so that it is possible to use a Bayesian estimation procedure. The probability density of h is $p_H(h)$ and the objective is to find the Maximum A Posteriori (MAP) estimate.

The transmitted signal $s(t)$ is known and can be expressed as $\sqrt{E}s(t)$ where E represents the signal energy and $s(t)$ has unity energy, that is to say it satisfies the relation:

$$\int_0^T s^2(t)\,dt = 1 \tag{5.5}$$

Moreover, for linear signalling, Equation 5.4 can be rewritten as:

$$r(t) = h\sqrt{E}s(t) + w(t), \quad 0 \leq t \leq T \tag{5.6}$$

Before estimating, it is important to reduce first the statistical data by means of a sufficient statistics, which still contains all the relevant information about the unknown parameter. In this case a sufficient statistic is r_1, given by Equation

5.7:

$$r_1 = \int_0^T s(t)r(t)\,dt. \tag{5.7}$$

This expression actually represents the correlation receiver, that is a system that correlates the input $r(t)$ with a stored replica of the transmitted signal $s(t)$. The output r_1 is still a Gaussian random variable and the following Equation 5.8 apply:

$$\begin{aligned} E(r_1|h) &= h\sqrt{E} \\ Var(r_1|h) &= N_0/2. \end{aligned} \tag{5.8}$$

The likelihood function is the probability of having certain observations (the received signal), given the parameter that needs to be estimated (the channel characteristic). In this case, the logarithm of the likelihood function, often called the *log likelihood function* is:

$$l(h) = -\frac{1}{2}\frac{(R_1 - h\sqrt{E})^2}{N_0/2} \tag{5.9}$$

The MAP estimate is the value of h where the expression:

$$l(h) = -\frac{1}{2}\frac{(R_1 - h\sqrt{E})^2}{N_0/2} + \ln p_H(h) \tag{5.10}$$

is a maximum.

Now the probability $p_H(h)$ has to be found. For this objective, one has to remind the physical structure of the system. Considering the coverage area of a single LED in 1-D, a uniform distribution of the mobile terminal inside the coverage region is assumed (and there is no reason to do differently). The coverage area in 1-D is a line which represents the basis of a triangle, while the upper vertex is represented by the LED. Denoting with b the distance of the mobile terminal from the centre of the triangle and L half of the basis of the triangle, it is possible to write:

$$\begin{aligned} b &\sim u[0, L] \\ p_B(b) &= \frac{1}{L}, \quad 0 \leq b \leq L \end{aligned} \tag{5.11}$$

The next step is to compute the distribution of a new random variable d, which is the distance from the transmitter to the mobile terminal. t is the height of the triangle. The limits of this new random variable are t and $\sqrt{L^2 + t^2}$. Moreover, the new random variable is obtained from the previous by the following operation:

$$d = g(b) = \sqrt{b^2 + t^2} \tag{5.12}$$

To solve this problem, the density function method is used (33, 43, 51). First, the inverse function $g^{-1} = \gamma$ is found:

$$g^{-1}(d) = \gamma(d) = b = \sqrt{x^2 - t^2} \tag{5.13}$$

Then, the derivative of γ with respect to d is taken:

$$\gamma'(d) = \frac{d}{\sqrt{d^2 - t^2}} \tag{5.14}$$

The probability density of d is then:

$$p_D(d) = p_B(\gamma(d))|\gamma^{'}(d)| = \frac{1}{L}\left|\frac{d}{\sqrt{x^2 - t^2}}\right| \tag{5.15}$$

Now, the probability density of H, $p_H(h)$ has to be found. The relationship between the channel characteristic h and the distance derives from the Lambertian pattern Equation 5.1:

$$H = f(d) = \frac{\alpha}{d^2} \tag{5.16}$$

where α is a constant. To obtain the $p_H(h)$, the density function method is applied again. The inverse function $f^{-1} = \delta$ is found:

$$f^{-1}(h) = \delta(h) = d = \sqrt{\frac{\alpha}{h}} \tag{5.17}$$

The derivative of δ with respect to h is:

$$\delta'(h) = -\frac{\alpha}{2h^2}\sqrt{\frac{h}{\alpha}} \tag{5.18}$$

The probability density of h is then:

$$p_H(h) = p_D(g(h))|\delta'(h)| = -\frac{1}{L}\left|\frac{\sqrt{\frac{\alpha}{h}}}{\sqrt{\frac{\alpha}{h}} - t^2}\frac{\alpha}{2h^2}\sqrt{\frac{h}{\alpha}}\right| \tag{5.19}$$

Now it is possible to resolve the MAP estimation. Equation 5.10 is rewritten on the basis of Equation 5.19:

$$l(h) = -\frac{1}{2}\frac{(R_1 - h\sqrt{E})^2}{N_0/2} + \ln\left(-\frac{\alpha}{2Lh^2(\sqrt{\frac{\alpha}{h}} - t^2)}\right) \tag{5.20}$$

To obtain the MAP density, the expression of $l(h)$ is differentiated, obtaining:

$$\frac{\partial l(h)}{\partial h} = \frac{R_1 - h\sqrt{E}}{N_0/2}\sqrt{E} + \frac{\alpha 4Lh(\sqrt{\frac{\alpha}{h}} - t^2) - Lh^2\sqrt{\alpha}(h)^{-3/2}}{[2Lh^2(\sqrt{\frac{\alpha}{h}} - t^2)]^2} \tag{5.21}$$

Then, the result is set to zero. The result is the value of h which satisfies the MAP condition. The normalized error variance for MAP is:

$$\frac{E[h_\epsilon^2]}{\sigma_h^2} = \sigma_{h_\epsilon,n}^2 = \frac{\sigma_{h_\epsilon}^2}{\sigma_h^2} = \left(1 + \frac{2\sigma_{h_\epsilon}^2 E}{N_0}\right)^{-1} \tag{5.22}$$

The quantity $\sigma_{h_\epsilon}^2 E$ is the expected value of the received energy. As it can be seen, the accuracy in terms of h gets worse, i.e. the normalized variance increases, as the received energy decreases. It means that, since the distance d is related to the channel characteristic h from Equation 5.16, if the error in the channel characteristic determination increases, also the error in the distance determination increases as a consequence.

As a final step, one could produce graphics of the distance accuracy with respect to the energy transmitted by the LED in order to define a threshold above which the system has to operate, i.e. the minimum error variance in the distance estimation. This is important because a LED has a dimming capability. Therefore, in this way it is possible to define the minimum transmitted energy to reach certain performances for localisation purposes.

5.2.5 Other Relevant Aspects

In 1-D positioning systems it is necessary to have at least two distance measurements from two different LEDs to know exactly the mobile terminal's location (in 2-D at least three transmitters should be used). However, one of the initial hypotheses was that just one LED covers every area. This makes the analysis simpler, as it is not necessary to deal with interference among multiple LED transmissions. However this represents a problem in terms of accuracy for the localisation service, because if one has just one distance from a LED, the point could be located both at one side (e.g. left) and at the other (e.g. right) with respect to the LED itself. A solution to this problem could be the use of memory. If the mobile terminal records few last positions and few last areas which it has passed through, it could be able to determine its exact location between the two possible positions.

One problem arises from the interference between the LEDs involved in the localisation and those aimed only at illumination. This problem is caused by the fact that the receiver is not able to distinguish between the power emitted for localisation and the one emitted for illumination. This problem could be solved by the use of two different frequency bands (using for example also the infrared light band), one for illumination and the other for localisation. In this way it is possible to separate the two luminous fluxes, one coming from the single localisation LED and the other originating from the multiple illumination LEDs. A similar approach is proposed by (37). In this case, however, every single LED transmits a unique identifier, represented by a single frequency tone and the system becomes consequently very complicated.

Also the transmission of a unique identifier by each LED is an important issue. It is necessary that every mobile terminal is able not only to locate itself with respect to a certain LED, but also to understand which the closest LED is.

To address this issue a solution could be the use of a modulation applied to the transmitted signal as proposed by Wang *et al.* (48), instead of using different frequencies.

5.3 Possible Future Extensions

The physical coverage area is divided in subunits which contain more LEDs. The time is slotted and in every slot dedicated to localisation services, only one transmitter can send its information. So, the receiver always knows the transmitter. Every frame is divided in slot and for the position determination every transmitter has its own slot to transmit its power.

Therefore a problem is to find the right number of LEDs involved in the positioning to reach a certain accuracy. It has already been shown that in the 1-D case, just two LEDs are necessary to locate a user. However, if dimming occurs, the number of LED should be incremented to obtain the desired accuracy in terms of localisation error. What would be important to obtain is the relation between the minimum number of LEDs required for a certain localisation accuracy and the power emitted. By knowing this relation, one can increase the number of LEDs involved in the localisation while decreasing the power emitted by each, thus leaving accuracy the same.

The authors in (61) analytically determine the optimal power allocation between more transmitters that minimizes the squared position error bound with a fixed total power amount. They develop both an optimal and a near-optimal strategy (with a lower computational cost) in order to allocate power to different anchors for localisation purposes. In addition, they show that their approach is much better than a uniform allocation strategy.

Chapter 6

Conclusions, Improvements and Future Implementations

As demonstrated in Chapter 4, the proposed system actually improves the localisation estimation, compared to a system which makes use of only RSSI values. Telecom Italia S.p.A. is currently implementing the described algorithm on a smartphone. The data acquisition phase has already been accomplished.

That being stated, there are still many considerations about possible future implementations and improvements that can enhance further the performances of the system from many points of view.

Shortening of the offline phase. One of the greatest issues of this system is the duration of the offline phase in the fingerprinting algorithm. A possible way to address this issue could be the acquisition of the different fingerprints at different times and not necessarily all in the first initialisation phase. This means that the reference RSSI values can be collected and added to the database even during the online phase, reducing a lot the work to be done during the offline phase. The application of this idea would reduce the strong difference between the offline and the online phases. Obviously, this solution is possible only if RSSI values are not the only source of information used by the system.

In particular, it is possible to collect RSSI values in areas which are close to the entrances of the building, so that the first movements are well recorded by the RSSI algorithm. Then, thanks to dead reckoning algorithm, the system could continue to track movements with a certain accuracy even if no RSSI fingerprints have been collected nearby. Of course, using only the inertial navigation system, errors can propagate and increase through the various steps. Thus, it would be necessary to periodically correct these errors, for example by giving directly to the users the possibility to locate themselves by scanning a QR code, RFID tag or a NFC tag, installed in particular points in the environment.

Then, when the user exits from the indoor environment, the smartphone sends to the central server the collected RSSI measurements with the respective estimated positions in order to be saved in the fingerprinting database located in the server, for a later use.

Implementation of the neural network on mobile devices. If the previous method was applied, it would be a possibility for the implementation of the algorithm directly on the user's smartphone, instead of realising it on a central server. Thus, the scenario would be the following. When entering the building, the user downloads the neural network parameters from the server. Therefore, every user would have its own trained neural network on his smartphone. Then, the neural network works in parallel with the inertial navigation system algorithm. The two estimates, produced by the two parallel algorithms would be then processed by the particle filter, which returns the final estimation.

Such a method would avoid a massive communication between a central server and the single users, because the algorithm would work directly on the user's mobile device. Thus, this approach would reduce the need of creating new network infrastructure and would be based mainly on the user's smartphone, thus saving time and costs for the deployment. In addition it is important considering privacy concerns, since the information about the user's position would remain at the mobile device, without the need to send it on the network.

Landmark recognition. A system for *landmark recognition* could allow to recognise particular places like elevator or stairs only on the basis of particular RSSI values or measurements from inertial sensors. In particular, such systems would be really important in places where the Wi-Fi signal coverage is low.

Subspaces and cascade of artificial neural networks. It can be possible to divide very large indoor environments like office buildings or shopping malls into many subspaces, characterised by different behaviours in terms of Wi-Fi signal coverage. First of all, an ANN is used to determine the probability for a measured RSSI fingerprint to belong to one of the subspaces. Then, another ANN, which has been trained only with data from the specific subspace, is used to infer user's position within the considered subspace. With such approach, the localisation problem would be split in two different phases, thus having the possibility to solve the two problems independently. This approach has been proposed in (31).

LTE, UMTS and GSM Received Power. As well as Wi-Fi RSSI values, it could be possible to consider also received power values of LTE, UMTS or GSM technologies.

The MATLAB® code of the algorithm is available on request to the author of the thesis.

Bibliography

[1] J. Yunye J, T. Hong-Song, S. Wee-Seng and W. Wai-Choong, "A robust dead-reckoning pedestrian tracking system with low cost sensors," *Pervasive Computing and Communications (PerCom), 2011 IEEE International Conference on*, pp. 222-230, 21-25 March 2011. 50, 60, 61

[2] S. Ayub, A. Bahraminasab and B. Honary, "A Sensor Fusion Method for Smart phone Orientation Estimation," *13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, Liverpool, 2012. 50, 54, 55, 56, 57

[3] A. R. Pratama, Widyawan and R. Hidayat, "Smartphone-based Pedestrian Dead Reckoning as an indoor positioning system," *System Engineering and Technology (ICSET), 2012 International Conference on*, pp. 1,6, 11-12 Sept. 2012. 50, 60, 62

[4] U. Steinhoff and B. Schiele, "Dead reckoning from the pocket - An experimental study," *Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on*, pp. 162-170, March 29 2010-April 2 2010. 50

[5] D. Comotti and M. Ermidoro, "Progetto di Microelettronica - Sviluppo di algoritmi per la stima dell'orientamento di un sensore inerziale," 29 March 2011. 52, 53

[6] W. Premerlani and P. Bizard, “Direction Cosine Matrix IMU: Theory”. 54, 58

[7] L. Pappalardo, “Localizzazione: Problema, Tecniche, Algoritmi,” `http://didawiki.cli.di.unipi.it/lib/exe/fetch.php/rhs/localizzazione.pdf`, 2011, retrieved July 19, 2013. 7

[8] M. Cavagnaro, “Impatto ambientale dei campi elettromagnetici,” 2012. 12

[9] L. Gogolak, S. Pletl and D. Kukolj, “Indoor fingerprint localization in WSN environment based on neural network,” *Intelligent Systems and Informatics (SISY), 2011 IEEE 9th International Symposium on*, pp.293-296, 8-10 Sept. 2011. 46

[10] F. Shih-Hau, L. Pochiang and L. Tsung-Nan, “Indoor localization by a novel probabilistic approach,” *Signal Processing Advances in Wireless Communications, 2007 IEEE 8th Workshop on* , pp.1,4, 17-20 June 2007. 50

[11] S. Hyuck Shin, C. Gook Par and S. Choi, “New Map-Matching Algorithm Using Virtual Track for Pedestrian Dead Reckoning,” *ETRI Journal*, vol. 32, no. 6, pp. 891-900, Dec. 2010. 60

[12] J. Kim, H. Jang, D. Hwang and C. Park, “A Step, Stride and Heading Determination for the Pedestrian Navigation System," *Positioning*, Vol. 1 No. 8, 2004. 60, 62

[13] H. Weinberg, “Using the ADX1202 in pedometer and personal navigation applications,” 2002. 62

[14] D. Alvarez, R. C. Gonzalez, A. Lopez, and J. C. Alvarez, “Comparison of step length estimators from wearable accelerometer devices,” *Proc. IEEE EMBS*, Aug. 2006, pp. 5964-5967. 62

[15] J. Scarlet, "Enhancing the Performance of Pedometers Using a Single Accelerometer," *Analog Devices AN-900 Application Note*, 2005. 62

[16] M.P. Michaelides and C. G. Panayiotou, "SNAP: Fault Tolerant Event Location Estimation in Sensor Networks Using Binary Data," *Computers, IEEE Transactions on* vol. 58, no. 9, pp. 1185-1197, 2009. 50

[17] T. Höllerer, D. Hallaway, N. Tinna and S. Feiner, "Steps Toward Accommodating Variable Position Tracking Accuracy in a Mobile Augmented Reality System," *In Proc. AIMS'01*, pp. 31-37, 2001. 50

[18] S. Kaluwahandi and Y. Tadokoro, "Portable traveling support system using image processing for the visually impaired," *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol.1, pp. 337-340, 2001. 49, 50

[19] G. Retscher, "Pedestrian navigation systems and location-based services,"*3G Mobile Communication Technologies, 2004. 3G 2004. Fifth IEE International Conference on*, pp. 359-363, 2004. 50

[20] H Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, Romit R. , "No need to war-drive: unsupervised indoor localization," *Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys '12)*, pp. 197-210, 2012. 49

[21] S.-H. Tsai and S.-Y. Lau and P. Huang, "WSN-based real-time indoor location system at the Taipei World Trade Center: Implementation, deployment, measurement, and experience,' *Sensors, 2012 IEEE*, pp. 1-4, 2012. 44

[22] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: a quantitative comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499-518, November 2003. 16

[23] C. Savarese, J. M. Rabaey, and K. Langendoen, "Robust Positioning Algorithms for Distributed Ad-Hoc Wireless Sensor Networks," *Proceedings of the General Track of the annual conference on USENIX Annual Technical Conference (ATEC '02)*, pp 317-327, 2012. 20

[24] N. Kothari, B. Kannan, and M. B. Dias, "Robust Indoor Localization on a Commercial Smart-Phone," *tech. report CMU-RI-TR-11-27*, Robotics Institute, Carnegie Mellon University, August 2011. 43

[25] B. Dawes, K.-W. Chin, "A comparison of deterministic and probabilistic methods for indoor localization," *Journal of Systems and Software*, vol. 84, Issue 3, Pages 442-451, March 2011. 2, 44

[26] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in Ad-Hoc networks of sensors," *Proceedings of the 7th annual international conference on Mobile computing and networking (MobiCom '01)*, pp 166-179, 2001. 20

[27] R. Want, A. Hopper, V. Falcão and Jonathan Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.*, vol. 10, no.1, pp 91-102, January 1992. 20

[28] A. Harter, A. Hopper, P. Steggles, A. Ward and P. Webster, "The anatomy of a context-aware application," *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (MobiCom '99)*, pp 59-68, 1999. 21

[29] P. Bahl, V. N. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, pp. 775-784, 2000. 21

[30] A. Uncini, *Algoritmi adattativi per circuiti intelligenti: appunti del corso di circuiti e algoritmi per l'elaborazione dei segnali'*, 2010. 45

[31] M. Borenovic and A. Neskovic, "ANN based models for positioning in indoor WLAN environments," *Telecommunications Forum (TELFOR), 2011 19th*, pp. 305-312, 22-24 November 2011. 46, 119

[32] A. Shareef, Y. Zhu and M. Musavi, "Localization using neural networks in wireless sensor networks," *Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (MOBILWARE '08)*, ng), article 4 , pp. 1-7, 2008. 46

[33] A. Hallam, "Transformation of Random Variables," *Econ 671: Econometrics I Course*, `http://www2.econ.iastate.edu/classes/econ671/hallam/documents/Transformations.pdf`, Iowa State University, 1999, Retrieved July 05, 2013. 112

[34] U. Birkel and M. Weber, "Indoor localization with UMTS compared to WLAN," *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pages 1-6, 13-15 November 2012. 2

[35] M. Bouet and A.L. dos Santos, "RFID tags: Positioning principles and localization techniques," *Wireless Days, 2008. WD '08. 1st IFIP*, pages 1-5, 24-27 November 2008. 2

[36] B. Cook, "New developments and future trends in high-efficiency lighting," *Engineering Science and Education Journal*, vol. 9, no. 5, pp. 207-218, October 2000. 102

[37] G. Cossu, M. Presi, R. Corsini, P. Choudhury, A. M. Khalid, and E. Ciaramella, "A visible light localization aided optical wireless system," *GLOBE-*

COM Workshops (GC Wkshps), 2011 IEEE, pages 802-807, 5-9 December 2011. 103, 115

[38] N. Fallah, I. Apostolopoulos, K. Bekris and E. Folmer, "Indoor human navigation systems: A survey," *Interacting with Computers*, vol. 25, no. 1, pp. 21-33, 2013. 2

[39] Z. Jia, "A visible light communication based hybrid positioning method for wireless sensor networks," *Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on*, pp 1367-1370, 2012. 104

[40] S.-Y. Jung, S. Hann, and C.-S. Park, "TDOA-based optical wireless indoor localization using LED ceiling lamps," *Consumer Electronics, IEEE Transactions on*, vol. 57, no. 4, pp. 1592-1597, November 2011. 104

[41] T. Komine and M. Nakagawa, "Fundamental analysis for visible-light communication system using LED lights," *Consumer Electronics, IEEE Transactions on*, vol. 50, no. 1, pp. 100-107, February 2004. 102

[42] K. Lee, H. Park, and J. R. Barry, "Indoor channel characteristics for visible light communications," *Communications Letters, IEEE*, vol. 15, no. 2, pp. 217-219, February 2011. 106

[43] G. Scarano, *Segnali, Processi aleatori, Stima*, Casa Editrice Università La Sapienza, 2009. 112

[44] M. Poplawski, "What You Need to Know about LED Flicker and Dimming," `http://apps1.eere.energy.gov/buildings/publications/pdfs/ssl/poplawski_dimming_lightfair2012.pdf`, 2012, Retrieved July, 2013. 107

[45] C.-H. Lim, Y. Wan, B.-P. Ng and C.-M.S. See, "A real-time indoor wifi localization system utilizing smart antennas," *Consumer Electronics, IEEE Transactions on*, vol. 53, no. 2, pp. 618-622, May 2007. 2

[46] W. Bin and Z. Pu, "Algorithm of Dispersed PWM and Dynamic Refresh Mode for LED Display," *Control, Automation and Systems Engineering (CASE), International Conference on*, pp. 1-3, 30-31 July 2011. 108

[47] A.M. Vegni and M. Biagi, "An indoor localization algorithm in a small-cell LED-based lighting system," *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, pp. 1-7, 13-15 November 2012. 104

[48] Z. Wang, W.-D. Zhong, C. Yu, J. Chen, C.P. Shin, Francois, and W. Chen, "Performance of dimming control scheme in visible light communication system," *Optics Express*, vol. 20, no. 17, pp. 18861-18868, 2012. 116

[49] K. H. Loo, Lun Wai-Keung, Tan Siew-Chong, Y. M. Lai and C. K. Tse, "On the driving techniques for high-brightness LEDs," *Energy Conversion Congress and Exposition (ECCE), IEEE*, pp. 2059-2064, 20-24 September 2009. 107

[50] C. Laoudias, C.G. Panayiotou and P. Kemppi, "On the RBF-based positioning using WLAN signal strength fingerprints," *Positioning Navigation and Communication (WPNC), 2010 7th Workshop on*, pp. 93,98, 11-12 March 2010. 46

[51] Robert B. Ash., "Lecture 1: transformation of random variables," *Lectures on Statistics*, `http://www.math.uiuc.edu/~r-ash/Stat/StatLec1-5.pdf`, University of Illinois, 1999. 112

[52] H. L. Van Trees, "Detection, Estimation, and Modulation Theory," *John Wiley and Sons, Inc.*, 2001. 110

[53] D. Shin, D. K. Jung, Y. J. Oh, T. Bae, H.-C. Kwon, C. Cho, J. Son, D. O'Brien, T.-G. Kang, E. T. Won, and T. Matsumura, "Visible light communication: Tutorial," 2008, http://www.ieee802.org/802_tutorials/2008-03/15-08-0114-02-0000-VLC_Tutorial_MCO_Samsung-VLCC-Oxford_2008-03-17.pdf, Retrieved July 03, 2013. 102

[54] AirMagnet®, "AirMagnet® Survey/Planner: User Guide," 2011. 25

[55] "Dual Band ANT. Spec (RFW-SW-AN8205D-WG)," 20 May 2013. 28

[56] R. Flickenger, C. E. Aichele, C. Fonda, J. Forster, I. Howard, T. Krag and M. Zennaro, "Wireless Networking in the Developing World," 2006. ix, 31

[57] Google Inc., "Android Motion Sensors," http://developer.android.com/guide/topics/sensors/sensors_motion.html, Retrieved July 04, 2013. 3, 51

[58] http://en.wikipedia.org/wiki/File:Gimbal_lock.png, http://en.wikipedia.org/wiki/File:No_gimbal_lock.png, Retrieved October 10, 2013. ix, 54

[59] E. Orhan, "Bayesian Inference: Particle Filtering," http://www.bcs.rochester.edu/people/robbie/jacobslab/cheat_sheet/particle-filtering.pdf, Retrieved August 18, 2013, August 2012. 63, 64, 65

[60] Z. Zhou, M. Kavehrad, and P. Deng, "Indoor positioning algorithm using light-emitting diode visible light communications," *Optical Engineering*, vol. 51, no. 8, 2012. 104, 106

[61] W. Dai, Y. Shen and M.Z. Win, "On the minimum number of active anchors

for optimal localization," *Global Communications Conference (GLOBECOM), IEEE*, pp. 4951-4956, 2012. 116

[62] G. Kitagawa, "Monte Carlo filter and smoother for non-Gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, 1996. 63

[63] Google Inc., "Sensors Overview," `http://developer.android.com/guide/topics/sensors/sensors_overview.html`, Retrieved July 04, 2013. 5

[64] A. K. Jain, Jianchang Mao and K. M. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31-44, 1996. 47