POLITECNICO DI MILANO

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING

LAUREA MAGISTRALE DEGREE IN COMPUTER SCIENCE AND ENGINEERING

DIPARTIMENTO DI ELETTRONICA, INFORMAZIONE E BIOINGEGNERIA

ON THE USE OF BEACONS AND UWB TECHNOLOGIES FOR INDOOR NAVIGATION SYSTEMS

Advisor:

Prof. Luciano BARESI

Authors:

Diego ROTA, 841344

Simone MONTALTO, 841359

ACADEMIC YEAR 2016 - 2017

«Success is not final, failure is not fatal: it is the courage to continue that counts.»

W.S.C.

Abstract

The GPS technology allowed to simplify the fruition of the digital map and the navigation in external places. One of the main limits of this technology - due to the shielding of buildings – is that it cannot be used for an indoor environment localization system. In recent years, lots of researches have been carried out about this topic, given the goal of a future development of a system able to localize and provide navigation for people inside buildings. In literature, this problem is known as *Indoor Positioning* and is still an open point with a high number of people working on it. In this regard, Microsoft hosts an annual competition called *Microsoft Indoor Localization Competition*, where companies and universities present their research on the subject.

Moving from this context, we will conduct a study of the best technologies that can solve the problem of Indoor Positioning. The final goal is to design a system which will allow people with disabilities (in particular blind people) to autonomously navigate inside their own home or in a public building without any constraints.

More in depth, we will analyse and compare *Beacon* and *Ultra-Wide-Band*, two available technologies through which to obtain a digital signal for position processing. Next we will examine the problem of signal accuracy and we will study and develop a filtering algorithm to improve the quality of retrieved data.

Finally, we will develop a mobile application (iOS platform) for navigation: it will provide the user's location within the building and it will calculate the optimal path to reach the selected destination, providing indication through voice guidance. As an application designed for people with disabilities, the optimal path calculation algorithm should take into account any obstacles in the environment that will be reported by the navigator if they are on the user's path.

The presented solution must be able to provide an accurate position. Furthermore, the final system will have to be easy to install and configurate within a building and the end-user needs (such as moving as freely as possible in the surrounding space) will also have to be taken into account. The proposed work and techniques will be tested in a real context in order to verify their behaviour.

Sommario

La tecnologia GPS ha permesso di semplificare la fruizione di mappe e la navigazione su strada in ambienti esterni. Uno dei limiti del GPS è l'impossibilità di essere utilizzato con efficacia all'interno di un edificio. Negli ultimi anni si è molto dibattuto sulla possibilità di realizzare un sistema in grado di permettere la localizzazione e la navigazione anche in ambienti interni. Tale problema è noto come *Indoor Positioning* e tutt'ora in letteratura rimane un quesito aperto che in molti stanno cercando di risolvere. A tale proposito, Microsoft organizza annualmente una competizione nota come *Microsoft Indoor Localization Competition*, durante la quale aziende ed università presentano le loro ricerche sull'argomento.

Partendo da questo contesto, effettueremo uno studio delle migliori tecnologie in grado di risolvere il problema dell'Indoor Positioning. L'obiettivo è quello di realizzare un sistema che permetta a persone con disabilità (in particolare per i non vedenti) di poter navigare liberamente all'interno di una abitazione o di un edificio pubblico autonomamente.

Nello specifico, andremo ad analizzare e confrontare *Beacon* ed *Ultra-Wide-Band*, due possibili tecnologie attraverso le quali ricavare un segnale digitale per l'elaborazione della posizione. Successivamente verrà preso in esame il problema dell'accuratezza del segnale ed andremo a studiare e sviluppare un algoritmo di filtraggio per migliorare la qualità dei dati acquisiti. Infine svilupperemo un'applicazione mobile (per iOS) in grado di effettuare la navigazione: segnalerà la posizione dell'utente all'interno dell'edificio, calcolerà il percorso ottimale verso una destinazione selezionata e fornirà indicazioni vocali. Essendo un'applicazione pensata per persone disabili, l'algoritmo di calcolo del percorso ottimale dovrà tenere in considerazione eventuali ostacoli presenti nell'ambiente, i quali verranno segnalati dal navigatore qualora si trovino sul percorso dell'utente.

La soluzione presentata dovrà essere in grado di fornire una posizione accurata e precisa. Inoltre il sistema finale dovrà essere facilmente installabile e configurabile all'interno di un edificio e si dovrà tener conto delle esigenze dell'utente finale, il quale dovrà potersi muovere il più liberamente possibile nello spazio circostante. Il lavoro e le tecniche proposte verranno testate in un contesto reale al fine di verificarne il corretto funzionamento.

Acknowledgements

Vogliamo ringraziare tutti i membri del Dipartimento di Elettronica, Informazione e Bioingegneria del Politecnico di Milano per la loro disponibilità e simpatia dimostrataci in questi lunghi mesi di lavoro. Un particolare ringraziamento va al nostro relatore, Prof. Luciano Baresi, per i suoi suggerimenti e la disponibilità avuta durante la realizzazione di questa tesi.
Un grazie va inoltre a Livia, Sam, Marco e Matteo con i quali abbiamo condiviso infinite giornate di alti e bassi; senza il loro ottimismo ed umorismo sarebbe stata molto più dura.

Desidero ringraziare tutta la mia famiglia, dai più piccoli al più grande, per la vicinanza e l'appoggio incondizionato che mi hanno dato.
Un grazie ai miei coinquilini Michele, Nicolas ed Elisa con cui ho condiviso parte di quest'avventura lontano da casa.
Un ringraziamento ai miei amici e colleghi di lavoro Alberto, Paolo, Evaristo, Davide e Roberto per la disponibilità e talvolta pazienza avuta in questi anni.
Infine, pur essendo consapevole che poche parole non potranno mai essere sufficienti per spiegarlo, un ringraziamento speciale va a Giulio per essere un costante punto di riferimento, a Filippo per essere diventato la mia seconda famiglia e a Martina per non aver mai mollato; questa laurea è un pochino anche loro.

Diego

Desidero innanzitutto ringraziare Carmen, per essermi stata sempre vicina ed avermi sostenuto anche nei momenti più difficili.
Ringrazio la mia famiglia: mamma, papà ed i nonni, che non hanno mai fatto mancare il loro supporto ed il loro affetto, nonostante la distanza che ci separava; Marta, per essere stata sempre disponibile ed avermi fatto sentire a casa quando ne avevo bisogno.
Infine un ringraziamento va a tutte le persone che hanno condiviso con me il percorso universitario.

Simone

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Objective of this thesis

During the past decades technological progress has started to change the quality of human life, signaling the beginning of a new era of communication and information diffusion also based on powerful electronic devices. One of the main challenges is how to make the best use of technologies in order to see a significant social impact of these scientific achievements. In literature, one of the many still opened questions is how to obtain a location based information of an entity inside a multi-dimensional space.
Indoor positioning has emerged as a hot topic that gained gradual interest from both academia and industry. Accurate estimation is necessitated in a variety of location-based services such as healthcare, repository tracking, and security. Additional equipment for location sensing could be used for accurate estimation, but so far they are not widely used in general because those alternatives would require specialization in brands and would be costly (Kul, Özyer, and Tavli, 2014). Among all suggestions in literature including hardware and intense sophisticated computations, a versatile and low-cost location determination technology can be intended as the goal of this branch of research.

Aim of the work of this thesis is, starting from an analysis of the current available commercial implementations, to develop a *proof of concept* of a navigation system for indoor environments with a very high position accuracy, devoted to provide a significant improvement in accessibility, in particular with reference to disable people.

1.2 Thesis outline

The entire work of analysis, research, development and testing presented in this thesis is divided into three main chapters. Each one of them is related to one of the different aspects of the entire system model, which has been built-up to reach the goals previously described.
This model, as represented in Figure 1.1, is conceived in three different layers

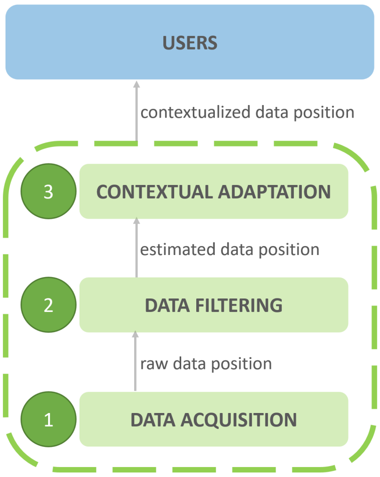

FIGURE 1.1: System model with the 3 different layer

each having their own customized tasks:

- **Layer 1: Data acquisition** - it deals with the management of signals retrieved from the digital instrumentation and the elaboration of a data position coordinates.
- **Layer 2: Data filtering** - it deals with the reprocessing of raw data provided by Layer 1, in order to elaborate a most accurate position after removing signals noises.
- **Layer 3: Contextual adaptation** - it deals with the adaptation of data (coming from Layer 2) to the context of implementation, such as linking position only with a certain sets of allowed points.

In the model just described, the flow of data is mono-directional, starting from Layer 1 up to Layer 3. At the end, the final data position coordinates are made available for users.

Chapter 2 provides an overview about the techniques for indoor positioning and navigation systems, analysing available software methodologies and hardware technologies, both in academic and industrial fields.

Chapter 3 mainly focuses on technologies analysis: it contains an overview of different types of hardware devices and manufacturers, communication protocols and performance tests in order to provide a comparison, with the final purpose of selecting the best technology to use for this work.

Chapter 4 is more theoretical. The first part presents the main steps (from mathematics, physics and automation theory) to create a Kalman filter model, here used to clean noised signals. In the second one, a prototype which integrates hardware technology and filtering algorithm is described.

Chapter 5 provides a guidelines for data contextualization and a description of the main features which characterize the structure of a software navigation system, including a real implementation with focus on map customization (obstacles and path cost).

Chapter 6 describes *Medical Center*, a demo mobile application developed in order to test the functionalities of the system and also proposed as Use Case.

Finally, *Chapter 7* presents the conclusions of this work and an overview of some possible future developments.

Chapter 2

State of the art

Indoor location services have become essentials in people activities like living, working, and studying. For example, tourists have the necessity to move easily inside public but unfamiliar places; in shopping center, users want to find the shop of a specific brand in a short time or they want be able to know where exactly is the object that they are searching for; in hospitals and medical centers, there is the necessity to track in real-time the location of wandering patients for timely treatment; in warehouses, staff need a tracking service that detects the goods and update the inventory in real time; in art exhibitions, viewers could be helped to visit the most interesting paintings in a large museum, obtaining information about the path to follow to reach them and so forth. GPS systems are used for outdoor localization, but this technology can't be used inside a building due to the obstruction of satellite signals, so a lot of studies have been performed in the last years to overcome this problem. Still, nowadays the indoor positioning is still an open question. The advances in mobile and wireless fields have led to a new era for indoor localization. Smartphones and others personal devices like smartwatch have driven widespread development of a variety of systems.
This chapter takes into researches about the techniques for indoor positioning and navigation systems, analysing available software methodologies and hardware technologies, both in academic and industrial fields.

2.1 Positioning technologies

In regard to hardware components, signals and software algorithms used to obtain a (more or less) accurate position, different kinds of techniques will be presented in this section. The work of research is based on *Microsoft Indoor*

Localization Competition[1], the annual reference world conference hosted by Microsoft. A guidelines on this part can also be found in (Jiang Xiao, 2016).

2.1.1 Tag Based Systems

This subsection presents indoor localization using UWB, ultrasonic, infrared, RFID, and wireless sensors. All these approaches require specific hardware to perform the positioning functionality and are summarized as *tag based system*, because users must have a tag device to be localized.

Ultra-Wide-Band

Ultra-Wide-Band or *UWB* is a radio technology originally used for wireless communication and later adapted to perform 2D and 3D indoor positioning. The main feature of UWB signals is that it can easily penetrate through obstacles with low power consumption. This characteristic permits a high positioning accuracy in centimeters scale. Using UWB devices to perform indoor positioning, time-based schemes provide very good accuracy due to the high time resolution (large bandwidth) of UWB signals. The new IEEE 802.15.4a standard has permitted to reduce the cost of UWB chips. Now it is possible to buy a single chip for 20$, making UWB technology competitive with WiFi-based indoor localization, with the advantage of a more accurate precision.

Ultrasonic

The general idea that has led to the use of ultrasonic for indoor positioning is to apply a transceiver to emit and detect ultrasonic signals. Hence, it is possible to compute the distance between a pair of transmitter and receiver given the medium travelling speed and recording the signal travelling time. The emission of ultrasonic wave is directional. This aspect introduces some difficulties in orienting the transceiver precisely. This leads to a decrease in the wave travelling time measurement accuracy, along with high power consumption.

[1] `https://www.microsoft.com/en-us/research/event/microsoft-indoor-localization-competition-ipsn-2017/`

In (Priyantha, Chakraborty, and Balakrishnan, 2000), the difference of propagation speeds has been used and the distance from transmitter via coupled RF and ultrasonic signals estimated. Results have shown a good accuracy (*6cm*), but there is the disadvantage of the inherent narrowband. Despite UWB, the presence of obstacles in the surrounding area may block the ultrasonic wave due to its coverage limitations. So, its uses are affected by technologies limitations.

Radio-Frequency Identification

Radio-Frequency Identification, better known as *RFID*, has become popular in indoor positioning because it has been used in hospitals for locating patients. Its implementation in this field uses reference tags positioned in a known location to adapt to environmental dynamics to enhance the accuracy of location estimation.

RFID-based 3D positioning approach permits to locate mobile users that are equipped with an RFID reader or tag. In (Seco et al., 2010), 71 tags have been distributed in an area of $1600m^2$ to achieve median accuracy of $1.5m$. To perform this test Gaussian processes have been used to model the RSSI distribution of RFID tags.

Beacon

The impossibility to use GPS signals in closed environments makes road-based navigation systems unreliable indoors. So different systems has been deployed with a beacon-based positioning technology which can estimate the location of an user inside the building. For this purpose, many Bluetooth Low Energy (BLE) beacons are used to cover a building in order to provide an indoor navigation system directly on users mobile devices. This solution permits to avoid the usage of external tags in addition to the smartphone, because beacon signals can be perceived directly by smart devices like smartphone, tablet and smartwatch. This solution permits to obtain a precision of $3m$.

2.1.2 Device-Free Systems

This subsection takes into consideration systems which don't require the use of external devices to keep track of an user or an object in a 2D and 3D environment.

Passive infrared Systems

Passive infrared Systems (*PIR*) are widely adopted to perform indoor positioning. The basis of these technologies relies on detecting changes of the infrared spectrum, when a specified region is monitored, which are caused by human movements. In fact, there's an obvious distinction between body temperature of a person and ambient temperature. Such a difference, that is detectable by infrared systems, makes it possible to develop an infrared-based positioning system based on the thermal radiation caused by people. A lot of systems have been developed with this technology.

In (Hauschildt and Kirchhof, 2010), thermal radiation by thermopile arrays has been measured to determine the position of the object. Experiments have been done in a 4.9x$5.2m$ area and the results can achieve a mean position error that is less than $30cm$ to discriminating and localizing three people even in noisy situations. This technology has two negative aspects: the first is that the cost of covering a large area will be prohibitive; moreover, the radiations from the dynamic environment also brings in challenges for this technology.

UWB Radar

UWB radar can be used for indoor through-wall detection and surveillance. In (Yang, Wang, and Fathy, 2008) an UWB radar system which operates at a frequency of $10GHz$ and permits to detect objects through walls has been suggested. The system is composed of a UWB double-ridge horn antenna as transmitter and a 16-element Vivaldi antenna as the receiver. Exploiting UWB characteristics, through-wall image reconstruction can be performed with high resolution. The high sampling rate is a challenge for this kind of products, so in (Yang and Fathy, 2009) the system has been further improved designing a real time see-through-wall system with customized FPGA firmware. The introduction of a customized FPGA deals with the challenging UWB signal acquisition problem that maintains high equivalent-time sampling resolution. Also, the system is able to motion tracking using a set of analog-to-digital converters, memory and auxiliary circuit.

Ultrasonic

Ultrasonic has been used for security reasons, like fall detection and pilferage detection inside private house or public environment. For example, in (Caicedo and Pandharipande, 2012) ultrasonic has been thought for indoor human detection. The basic idea is to transmit consecutive ultrasonic signals periodically. The echoes caused by user movement will be recorded by the receiver. So, the occurrence of human movement can be tracked by an algorithm with high detection probability. Another example can be found in (Hnat et al., 2012). Ultrasonic transmitters are positioned into houses on each doorway in order to identify users by its height and direction, as well as to record the passing sequence, i.e. the sequence of doorways through which they pass in order to identify their room location. The accuracy is 90%. The problem is that to deploy ultrasonic in large indoor surroundings will be very expensive.

WiFi

The solutions proposed so far have a scalability problem due to high infrastructure cost. Approaches based on WiFi take advantage of the available infrastructures for indoor localization. To exploit *WiFi* for this purpose, it is necessary to perform a process of determining abnormal events. In (Youssef, Mah, and Agrawala, 2007) the possibility to design a reliable device-free passive (DfP) localization system has been discussed. Two algorithms have been suggested for motion detection: moving average (MA) and moving variance (MV). Detection performances of these two algorithms are very high, but there is the problem that they can be used only in controlled environments.

2.1.3 Comparison

The key point of an indoor positioning system is the accuracy of the result. Usually, it is determined while measuring the mean distance error, calculating how close the estimated position (elaborated with a particular technique) is accurate respect to the actual position of user/device. In the last years a lot of researches have been presented, with a measured accuracy which decrease from kilometres to meters; the most recent works have led to an error in the order of centimeters. Nowadays, it can be ensured that the accuracy may vary, depending on technologies, from a minimum of $20cm$ up to a maximum of $100m$.

To solve the problem of indoor positioning, Microsoft has launched in 2014 the *Microsoft Indoor Localization Competition*. It provides a unified venue where researchers from university and industry can compare their technologies and find a complete solution to this problem. During the first edition,

22 solutions have been proposed: at that time, the best one was to use commercial WiFi access point, which achieved an accuracy of $1.6m$. But in fact, this error didn't permit to solve the problem of indoor localization and also the average setting time and calibration time was too high (about 5 hours in order to cover an area of $300m^2$).
Fortunately, during the last 3 years there has been an improvement concerning the accuracy: in the last 2017 Microsoft event, results have become better than in the past. Some of them will be now described, with all the available information concerning environment, devices and techniques. Regrettably, the lack of some details (not provided by original researchers) makes the comparison not so accurate. In the following line are reported a quick analysis of scientific papers.
In (Kuan-Wu Su, 2017) three beacons have been used to cover an area of $35m^2$, positioned on three different walls at an high of $2m$. The accuracy is 80% at the distance less than $1.3m$, and over 90% around $2.5m$.
In (Kota Kikuchi, 2017) have been used BLE-based localization method suitable for the indoor smartphone geolocation service. The experiment reveals that the proposed method can estimate the location with an accuracy of $0.54m$ (50 percentile). No information about environment configuration are provided.
Finally, the 2017 competition winner was the system described in (Ersan Gunes, 2017). The deployed hardware consists of three components: the small and light weighted anchor nodes, (at least) one tag and a gateway which acts as a data collector and transmits the data to the back-end; all devices were conceived by Quantitec and are based on UWB technology. During the test phase, four anchors have been positioned; the distance between them was from $15m$ up to $60m$. The position measurement was performed by calculating the time difference between the anchor nodes and the smart tag as well as through the one way ranging method and internal company algorithms. An accuracy between $5cm$ and $10cm$ has been achieved, depending on the calibration. This very good result is the starting point, in terms of accuracy, for the work described in this thesis.

2.2 Indoor navigation systems

This section is focused on the academic researches in terms of theoretical analysis, proof of concept and test in real scenarios regarding navigation systems, i.e. an electronic system that aids in indoor navigation.

2.2.1 Historical overview

In literature, there are a lot of papers concerning the indoor navigation topics. An important comparison of researches, until 2015, is available as a part of the work of (Alnafessah et al., 2016).
All the papers are divided into three main categories of systems:

- *Building-dependent* indoor navigation systems that utilize *building's infrastructure* (Figure 2.1)
- *Building-dependent* indoor navigation systems that require *dedicated infrastructure* (Figure 2.2)
- *Building-independent* indoor navigation systems (Figure 2.3)

The comparison tables are here reported only for overview purpose; for an in-depth analysis, please refer to the original work. Also references in Figures 2.1, 2.2 and 2.3 are related to the bibliography of the original paper.

Designers-year	Technology	Infrastructure utilized	Interface	Deployment	Testing
Kapic-2003 [33]	WiFi and Bluetooth for positioning and Dijkstra for path planning	Existing WiFi and Bluetooth networks	Voice-based interface and keyboard	Not deployed	Not tested
Nordin and Ali-2009 [34]	Image-based technology	A map of a building as a set of images	Cane with monocamera	Not deployed	Not tested
Guerrero et al.-2012 [35]	WiFi, Bluetooth, and infrared light in an augmented white cane	Existing WiFi and Bluetooth networks	Augmented white cane with voice-based instructions	Prototype deployed	Two tests were done The first was done with five blindfolded volunteers and the results indicated a favorable evaluation, with scores over the minimal acceptance level. The second test was done with two blind people and the rest were blindfolded engineering students
Serrão et al.-2012 [36]	Image-processing algorithms	GIS of a building	Hand-held camera and voice-based instructions	Prototype deployed	The prototype was tested and most planned routes could be followed from the start to the destination; the few problems were due to a lack of landmarks at a certain location
Nakajima and Haruyama-2013 [37]	Visible light communication, Bluetooth, and WiFi	Visible light of a building	Smartphone with an integrated receiver and headphones	Prototype deployed	The system was tested by a number of visually impaired subjects
Priya et al.-2013 [38]	Optical character recognition	Landmarks of a building as images	Hand-held phone with keypad replaced by camera and voice-based instructions	Prototype deployed	Not mentioned
Au et al.-2013 [39]	WiFi	Existing WiFi network and a map of a building	Mobile phone	Prototype deployed	Tested by two subjects
Movea's system-2013 [40]	Phone's existing sensors	Existing WiFi network and a map of a building	Mobile phone	Not mentioned	Not mentioned

FIGURE 2.1: Comparison of building-dependent indoor navigation systems that utilize building's infrastructure, as reported in (Alnafessah et al., 2016)

Designers-year	Technology	Cost	Interface	Deployment	Testing
Kridner-2002 [18]	Infrared	Low	Voice-based and keyboard	Prototype deployed	83.0–99.4% of users are able to reach the destination
Ran et al.-2004 [19]	Combination of GPS and OEM ultrasound positioning system	Medium	Voice-based	Prototype deployed	Tested and achieved about 22 cm accuracy in determining positions
Kulyukin et al.-2004 [20]	RFID	Medium	User communicates with a robot	Prototype deployed	Two tests were conducted: all users in two tests arrived at the destination
Wu et al.-2007 [15]	Different sensors, Dijkstra, and A* search algorithm for path planning	Not determined	Not mentioned	Prototype deployed	Two tests were conducted: their results were 69.79 cm and 54 cm errors in accuracy of determining positions
Riehle et al.-2008 [21]	UWB	High	Voice-based	Prototype deployed	Three tests were conducted but results were not mentioned
Chumkamon et al.-2008 [22]	RFID	Medium	Voice-based	Prototype deployed	Conducted test on simulator
Hedley et al.-2008 [23]	RF	Medium	Dedicated hardware	Prototype deployed	0.49 m error in locating objects indoors
Goyal et al.-2008 [24]	Zigbee-based location engine for positioning and Dijkstra for path planning	Medium	Keypad	Not deployed	Not tested
Ivanov-2010 [25]	RFID	Medium	Voice-based and alphanumeric keypad	Prototype deployed	Test was conducted by blind users in a hospital: all of them arrived at the destination except one
Zöllner and Huber-2011 [26]	Not mentioned	Not determined	Kinect camera mounted on a socket built with Sugru with laptop in backpack and white cane	Not mentioned	Not mentioned
Ganz et al.-2012 [27]	RFID, WiFi, and Bluetooth	Medium	Braille and voice-based	Prototype deployed	Tested with 24 blind and visually impaired people
Rao et al.-2012 [28]	GPS, Zigbee network, and ultrasonic sensor	Medium	Audio signals	Not mentioned	Not mentioned
Firmino and Teófilo-2013 [29]	Bluetooth stations fixed over the tactile floor and a way to connect to the Internet	Low	Mobile phone, audio-based to provide instructions and tactile-based for a user to deal with the system	Prototype deployed	Brief usability test with five visually impaired people
Legge et al.-2013 [30]	Infrared and Bluetooth	Low	Mobile phone with a standard 12-button phone keypad for input (10 digits, # and $*$ symbols)	Prototype deployed	Conducted three experiments and results demonstrate that the technology can be used reliably, but there are still some limitations
Toms-2014 [31]	Used sensors but the technology used is not mentioned	Not determined	Mobile phone	Prototype deployed	Tested the device with two dozen visually impaired people from different age group and got satisfactory responses
Yirka-2014 [32]	Image processing along with WiFi and Infrared	Low	Haptic output	Not mentioned	Not mentioned

FIGURE 2.2: Comparison of building-dependent indoor navigation systems that require dedicated infrastructure, as reported in (Alnafessah et al., 2016)

Designers-year	Technology	Interface	Deployment	Testing
Ram and Sharf-1998 [41]	Ultrasound and pyroelectric	Clasp-style pin affixed to the user's shirt or jacket	Prototype deployed	Not tested
Liu et al.-2008 [42]	Image-based using omnicamera	Omnicamera held by the user	Prototype deployed	Two tests were conducted: guideline detection achieved 96.9% success rate, and junction detection achieved 97.2% success rate
Kulyukin and Kutiyanawala-2010 [43]	Vision techniques	Haptic interface	Prototype deployed	ShopMobile 1 was tested with visually impaired people while whether or not ShopMobile 2 was tested is not mentioned
Folmer-2010 [44]	Dead reckoning	Smartphone	Not mentioned	Not mentioned
Riehle et al.-2013 [45]	Pedestrian dead reckoning	Text-to-speech audio cues associated with map features	Prototype deployed	Not mentioned

FIGURE 2.3: Comparison of building-independent indoor navigation systems, as reported in (Alnafessah et al., 2016)

2.2.2 Latest prototypes

This section will describe the more recent improvements, divided by underlying technology, referred to 2016 and to the first half of 2017.

Beacon

In (Ahmetovic et al., 2017) a work of research wanting to provide accurate navigation assistance to people with visual impairments has been suggested. The described approach uses sensor networks of Bluetooth Low Energy (BLE) beacons. This solution permits to minimize the structural modifications of the environment. To prepare the infrastructure, beacons have to be positioned in strategic points and samples of Bluetooth signal need to be collected across the whole environment. To achieve a high level of accuracy, a traditional regression-based localization approach has been utilized, introducing a graph-based localization method using *Pedestrian Dead Reckoning* (*PDR*) and *particle filter*.
The first step of this work has been to collect fingerprint of the RSSI signal of beacons measured by the smartphone (training data). This kind of information enables to localize a smartphone device by comparing the RSSI detected from surrounding beacons with the pre-recorded fingerprints. The probability distribution of the RSSI samples has been modelled by a Gaussian distribution with a mean and a standard deviation based on the mean of the RSSI at location x. Afterwards, a pedestrian motion model has been

used to predicts the user's location by PDR using some smartphone inertial sensors like accelerometers and gyroscopes. They have been used to detect if the user is moving or not and his direction. In particular, the user's motion state has been detected doing the computation of the standard deviation of the magnitude of accelerometer measurements in a short time window. Instead the walking direction is measured using the smartphone orientation. In a real use case, the phone's orientation is approximately parallel to the user's direction. The angle between user's walking direction and the edge is computed using gyroscope data.
To perform the experiment, 24 *Kontakt.io beacons* have been used. The test environment was a 16x$2m$ hallway. Beacons have been positioned at $1m$ of distance each others, alternating the wall, starting $4m$ before and ending $4m$ after the corridor. The results have shown an accuracy error of $0.68m$. To understand how the accuracy achieved with this proposed method would impact the usability while navigating, the system was integrated into a smartphone navigation application and tested into an university campus with the participations of 6 blind people. The environment test was composed of several buildings, with two paths long $230m$ and $390m$ respectively. The results of the users evaluations show that the proposed technique is capable of accurately guiding individuals with visual impairments through a testing navigation field unfamiliar to the users.

UWB

The goal of (Alnafessah et al., 2016) is to develop a system to exploit and integrate current technologies, such as smartphones, to develop a tool to help blind people navigate inside buildings. Their efforts have focused on four main functionalities: mapping, positioning, interface, and navigation. All the system relies in two applications: a smartphone app and a localization server. The localization server is entirely based on Ubisense building representation, which is an academic research package that can be used to track objects using UWB technology. Ubisense system consists of three main components: Ubisense sensors, Ubisense tag and Ubisense location software. The localization system works as follow: Ubisense sensors receive UWB pulses that are generated from tags to find a tag location. The location of the tags is calculated using hyperalgorithms, which use the angle of arrival (AOA) and time difference of arrival (TDOA) of tag signals. The navigation component is mainly responsible for three main functionalities: it has to find the shortest path from the user's current position to the desired destination; it has to convert the path to a set of guidance statements that guide the user; it has to drive the user until he arrives at the desired destination. This component resides in the smartphone application. The shortest path is calculated using Dijkstra algorithm, using a graph representation of the environment. The entire system has been tested in a lab with metal walls and desks and dimensions of 9.557x$9.562m$. In order to evaluate the system, a test with blind user has been performed. This experiment aimed to test the suitability of the system and its interface for blind and visually impaired people since the system is intended for them. The overall accuracy of the localization system is $31.976cm$. In order to test the system's ability to guide users, different positions and destinations have been randomly chosen and the system was asked to guide users through these paths. The system was capable of guiding users to their destinations with an overall satisfaction of 9.3/10.

WiFi with other supporting technologies

Different researches that have involved the combination of multiple technologies have been proposed. (Abu Doush et al., 2017) is an Oxford research which has the goal to develop a navigation system for blind people that allows them to reach a specific object. In particular, this solution has been tested in a library to help people to find the desired book inside it. The developed system utilizes several technologies and integrates them to provide the intended users with the best possible adaptable navigation solution. The proposed system has involved three main steps: localization, destination location and navigation.
The localization step has been done using a combination of various indoor localization technologies. WiFi, Bluetooth and RFID technologies have been integrated, each one of them used according to their corresponding coverage area. For example, WiFi hotspots have been used in order to localize blind users inside a building through fingerprinting technique. Bluetooth receivers, instead, have been used to help users to locate the desired shelf and finally RFID readers and RFID tags have been used at short distance to detect the exact location of the searched book.
The second step involves the research of an object to obtain indications to reach its place. Assuming that the item's ID is given, the system determines where the item is located (on which shelf) and then retrieves the shelf's ID and its location on the map. Once done, the shelf navigation phase can start.
The last step needs to establish step-by-step directions to help users reach the desired item. The navigation phase is further divided into three steps:

- **Navigate users from building entrance to the desired floor:** WiFi technology has been used to determine the current position of the user. To localize people and detect their current location, fingerprinting technique is used. The floor plans have been represented using a graph. All the areas are represented by nodes, linked to each other by edges. Then, a path planning algorithm is used to obtain the best path from the building's entrance to reach the desired floor. In particular, Dijkstra algorithm has been used in the path planning process. The cost function of each path has been modified to represent paths with obstacles. This means that longer paths with less obstacles are considered better than shorter paths with higher number of obstacles.

- **Navigate users inside the floor to reach the desired bookshelf:** Bluetooth technology has been used to determine the desired shelf. Each shelf has to be equipped with a bluetooth transmitter, in order to be identified uniquely. A pairing process is initiated to connect the user's smartphone device with the device on the desired shelf. The user's smartphone pairs only with the shelf-reader device associated with the desired shelf. Once the pairing process is completed, the shelf-reader device broadcasts an alerting message to guide the user to its location.

- **Navigate the user from the shelf's end to the desired item:** The last step involves reading RFID tags in order to find the searched one. The item identification process starts when the blind person scans any of the RFID tags using the shelf-reader. The items are arranged on the shelf in an order that allows the shelf-reader device to determine how close the user is to the desired item, and which direction (both horizontally and vertically) the user should take in order to reach the item.

In order to verify if the proposed solution works, the system has been implemented in one of Yarmouk University Library floors and developed the necessary hardware and software interfaces as a proof of concept. Detailing the solution, six WiFi access points have been placed on the library's first floor, which has a total squared area of $4500m^2$. The localization is done using the signal strength (RSSI) values of each WiFi signal. This technique requires an initial training phase in order to collect the required RSSI values. All the collected data have to be saved into a database to create the required map of the RSSI values. To perform the user location, the current user's RSSI values are measured and compared with the stored map to find the closest matching location. The problem of this solution is that sometimes access points

in public environments have to change their energy levels depending on the number of connections. This make WiFi fingerprinting not work well in a real scenario. During the test phase, the connection to the access points have been made secure with a password to avoid that other people connect to the WiFi.
Each shelf is equipped with a shelf-reader device that contains a controller, a Bluetooth module, an RFID reader, a text-to-speech module and a sound speaker. When the blind person is near the shelf-reader device attached to the shelf that contains the desired item, a Bluetooth connection is established between the user's smartphone and the Bluetooth module on the shelf-reader device. At the end of the pairing process, the user is informed that he is near the correct shelf.
Then can start the last phase. Within a shelf, hundreds of items can be arranged. All the items are equipped with a RFID tags. The RFID reader module scans the tags and reads the tags IDs from a short distance of few centimeters (up to $30cm$). The results of the research have shown that the participants were able to locate objects within $10cm$ of accuracy. Regarding the navigation accuracy, three points have been taken into account. It has been depicted that there is an error that goes from 0.5m up to $1.5m$ during the navigation phase.

In (Dao et al., 2016) has been proposed a system for navigation assistance, designed according to a layered architecture depicted as follow:

- **Physical layer:** it is composed of all the hardware necessary to the system to work. In particular, some environmental cameras for vision-based human detection and tracking, a hand-held smartphone with headset for user-system interfacing and Wi-Fi-based localization, a central server, a number of Wi-Fi access points ensuring wireless communication and at the same time acting as beacons for Wi-Fi-based localization.

- **Logical layer:** this layer provides the Application Program Interface (API) and communication.

- **Functional layer:** it is composed of all software modules, which could be run online or offline, as data acquisition, environment representation, multi-modal user localization, path finding, speech recognition and speech synthesis for user-system communication.

- **Application layer:** the application that provides information about the navigation for blind people.

To start using the navigator, users have to make their request for navigation by speaking into the microphone. The server analyses the request and determine the user's target location on the pre-built map (which is provided in an XML file). Then the multi-modal localization module is activated to continuously estimates the user position. The shortest path from user location to the target one can be determined and navigational indication will be generated and sent to the smartphone and played as synthesized voice to the user. The localization - path-finding - indication process is repeated and the path is recomputed until the user reaches the target location. For the localization purpose, Wi-Fi and camera are the main technologies being used. In order to calculate the user position, the environment has been divided into grid points. At each point, the user appearance probability and precision are calculated with information provided from all available localization technologies. Since each localization technology provides results at different moments, the time past since the last result received from a technology to the current system time can affects the final result. To evaluate the entire system, it has been implemented in Nguyen Dinh Chieu School for Blind Pupils in Hanoi, Vietnam. Ten Wi-Fi access points and four camera have been used to cover one floor composed of nine rooms and a corridor. For the localization test, a person holding a smartphone moves along a path with predefined turning points. Each time it reach one point, the user is asked to press a button to signal the system saving a timestamp which will later be used to interpolate the ground-truth for accuracy evaluation. The system has been tested using only WiFi access points, only the four camera and the composed system. Camera only and multimodal system has returned better results, with an average error on the positioning of $0.89m$. The reliability of 90% is $1.71m$.

2.2.3 Industrial solutions

Here some industrial solutions of indoor navigation systems currently available on the market are presented.

MapsPeople

MapsPeople[2] is a company that helps visitors, customers, travellers or patients to navigate in indoor environments. Its solution is built on Google Maps, which makes the transition from the outdoor world to the indoor environment transparent. All the known functionality and design of Google Maps is brought inside and the user can get directions from any point outside the venue and all the way to a specific product, lecture room, stand or hospital ward. MapsPeople can be integrated into existing applications for mobile and desktop or as a standalone service. It is possible to add specific routing, adding customised points of interest and editing the layout of the map. The system can interface with the most suitable indoor positioning system like WiFi, Bluetooth (BLE) or positioning based on magnetic fields.

Senion

Senion[3] is a comprehensive hardware-software system for indoor positioning of mobile devices. The system is designed to meet the needs of the rapidly growing location-based services market. It uses Beacon technology to provide user position and permits to remove navigational hurdles in hospitals, to enable location-based notifications in shopping malls and to expand accessibility for the visually impaired. Beside the *Beacon Management System*, which permits to manage all the beacons installed in the environment to obtain an accurate position, they produce a service called *WayFinding* that provides a full navigation system customized for the specific environment. It permits to calculate the optimal path to the destination and shows the next two actions to be take in order to arrive at the destination.

Eliko KIO

Eliko KIO[4] is presented as a precise real time location system for tracking any object in 2D or 3D space. It uses the Ultra-Wide-Band technology to enable the micro-position objects through obstructions. KIO also works in non-line-of-sight conditions and both indoors and outdoors. It is suitable for tracking a single object or for building large-scale infrastructures for simultaneous real time positioning of hundreds of objects.

[2] `https://www.mapspeople.com`
[3] `https://senion.com`
[4] `http://www.eliko.ee/products/kio-rtls/`

Infsoft

Infsoft[5] offers indoor navigation and indoor positioning systems for various industries. It uses different technologies to provide the position, according to the final use of the system. In particular, Infsoft uses Bluetooth beacons for indoor positioning or set up the indoor positioning system with Wi-Fi. Furthermore, it realizes high-precision solutions based on Ultra-wideband (UWB). The proposed solutions are studied to work well in environments like airports, train stations, hospitals, offices and industrial areas.

IndoorAtlas

IndoorAtlas[6] uses the Earth's magnetic field to provide Indoor Positioning. By utilizing the built-in magnetic sensor within a smartphone, they are able to detect anomalies in the Earth's magnetic field to enable an hybrid technology to accurately pinpoint and track a person's location indoors. They also provide a cloud-based platform to developers to create venues, manage data and build location-based services within a mobile application that can include advertising, point-of-interest (search), wayfinding and much more. A navigation system is also provided, which permits to search for Point Of Interest, calculate the optimal path and navigate to the desired position.

GoIndoor

GoIndoor[7] offers real time indoor navigation tools and technologies. It uses Beacons as positioning system. Signals received from the beacons are processed by the SDK to compute precise location of the device. Then visitors can use turn-by-turn navigation based on their profiles. For example, route won't include stairs if a person is in a wheelchair; visitors won't see restricted areas.

[5] `https://www.infsoft.com`
[6] `http://www.indooratlas.com`
[7] `https://www.goindoor.co`

Chapter 3

Data Acquisition

This chapter will deal with Layer 1, which has been presented in Section 1.2. It will describe the work of analysis done to choose which technology is the best to perform the scope of indoor positioning. Two main technologies will be analysed: *Beacon* and *Ultra-Wide-Band*.

3.1 Test environments description

The following sections describe the multiple tests that have been carried out in order to perform the analysis of the various positioning systems. As known, to obtain a reasonable comparison a controlled environment needs to be set up. Hence, the third floor of DEIB building (Politecnico di Milano) was chosen for such purpose. In particular, two different environments were identified:

- A standard DEIB office room (we'll refer to it as *TE1*)
- A part of DEIB floor, composed of multiple office rooms, hallway, bathroom and laboratories (we'll refer to it as *TE2*)

These choices aim to simulate a wide range of possible real configurations and it's therefore important to focus on the different elements composing the two environments. For this reason, an accurate description is given in the next sections.

3.1.1 Test Environment 1 (TE1)

TE1 is represented in Figure 3.1. It is a 4.17x$4.65m$ office room, with a surface of about $20m^2$. The perimeter walls are made of different materials: three of these are made of plasterboards and divide this room from the other internal spaces of the department, the last one is built in reinforced concrete and corresponds to the external wall of the entire building. The latter one is also equipped with four windows. The room is furnished with a circular wooden table on one side of the office and another wooden desktop with chairs on the opposite side. Two wooden office closets of different sizes complete the furnishing. As you can see in Figure 3.1, the room is further equipped with a TV, air conditioning and a whiteboard placed near the door. No other major elements that could potentially create line-of-sight problems during the tests are in the room.

3.1.2 Test Environment 2 (TE2)

TE2 is represented in Figure 3.2. It is a part of the third floor of DEIB (Department of Electronic, Information and Bioengineering of Politecnico di Milano) with a surface of about $370m^2$ (21.75x$17m$). This one too is made of multiple office rooms, that can be more or less shaped as the one explained in TE1. A bathroom and an utility room are also there. A major part of the surface is occupied by a complex set of internal emergency stairs. In its center a narrow space hosts a laboratory and two little archives.
All the interior walls are made of plasterboards, with the exception of one wall pertaining to the stairs which is in reinforced concrete. The central room features many windows along the entire length of both walls. Metal closets are also provided in some of the rooms and hallways. The entire TE2 environment is subjected to multiple electromagnetic waves, due to the high presence of electronic devices and networks.

FIGURE 3.1: TE1 - A standard DEIB office room

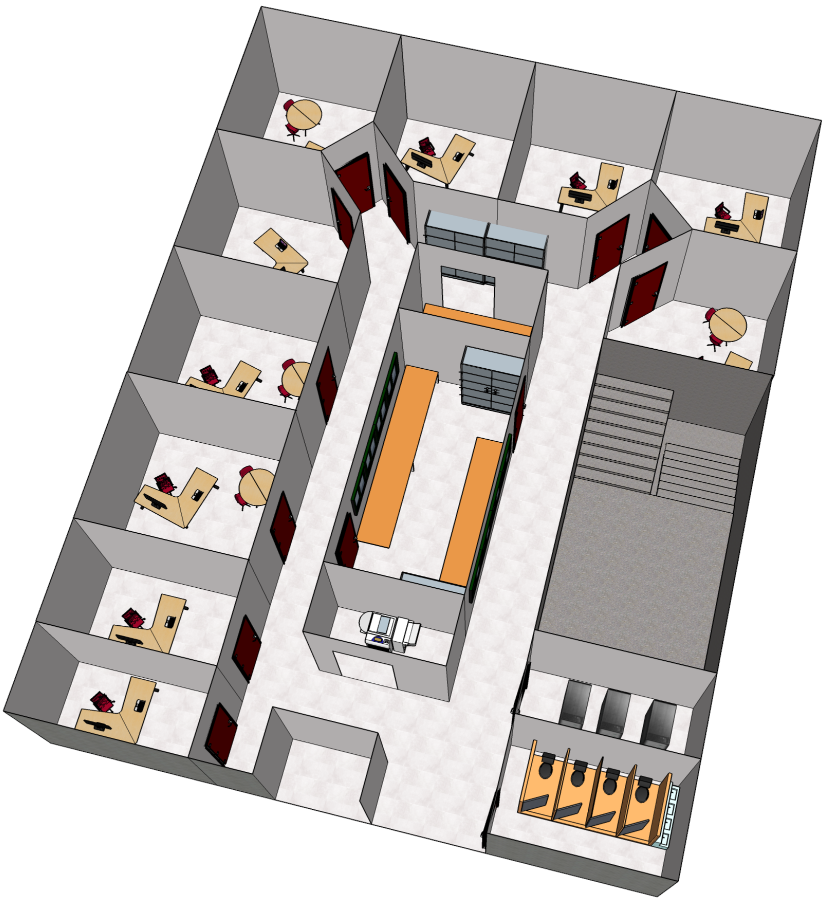

FIGURE 3.2: TE2 - A part of DEIB third floor

3.2 Beacon

Over the last few years, *beacon technology* has been used by a lot of companies. The word beacon identifies a little electronic device that transmit radio signals in broadcast over *Bluetooth 4.0*, also known as *Bluetooth Low Energy* or *BLE*. It works as follow: the radio signal can be intercepted by mobile devices like smartphones and tablets that are in a range of less than $100m$ (this range can vary depending on different parameters). The devices that intercept the signal can perform some actions based on the information that are sent from the beacon. To intercept the radio signals, devices must be bluetooth enabled. Beacons send a predefined packet that is chosen during the configuration phase. It can contain links, strings or other information which devices near the beacon can read and, therefore, react to them. Beacons can send radio signals but cannot receive any information from near devices. This one-way operation mode does not require pairing between beacon and devices that want to read the signal and so allow to use the technology in a wide range of applications.

Beacons can take the advantages of BLE technology, but at the same time inherit the disadvantages. Bluetooth is a standard which operates in a range of frequencies that goe from 2.4 to $2.4835GHz$. These frequencies are also used by other electronic devices, so, as mentioned in (Granby, 2012), it is quite easy to have interferences between them. Possible objects which might conflict with Bluetooth devices are, for example, routers that use $2.4GHz$ frequency, microwave ovens, office lighting but also water and the human body. All of these elements are easily found in an environment, so it is important to keep this problem in mind when it comes to dealing with beacons in a real environment.

The first implementation of beacon was by Apple in 2013, who introduced the *iBeacon* protocol based on the concept of beacon. Apple doesn't build a real beacon hardware, but only the protocol that can be integrated to communicate with iOS devices like iPhone, iPad or Apple Watch. Beacons are mainly powered by batteries which can guarantee a duration from a matter of months up to 2-3 years, depending on the frequency and the power of the radio signal. It is also common to use USB power adapter or solar panels to feed them.

Today's market offer lots of devices - produced by more than one company – which are compatible with different beacon protocols. The underlying technology is the same among all these products. All of them use a micro-controller with a BLE chip (a particular version of Bluetooth chip that send less data in order to consume less energy). The only thing that can change is the protocol that the company decide to integrate in their own devices.

3.2.1 Beacon protocols

One of the main characteristics of BLE is the ability to work in two different modes: **connected** and **advertising**. The former uses a *one-to-one* connection to transfer data between two devices. The latter, instead, broadcast data to many devices (this is a one-to-many connection). Beacon based on BLE uses advertising mode to communicate with many devices: there are several protocols which define the packet structure that is sent from the beacon in this modality. Some of these protocols are open source and freely usable, others are closed and a fee could be required. Nowadays, according to (ARMmbed, 2015), there are four main famous protocols: *iBeacon*, *AltBeacon*, *URIBeacon* and *Eddystone*.

iBeacon

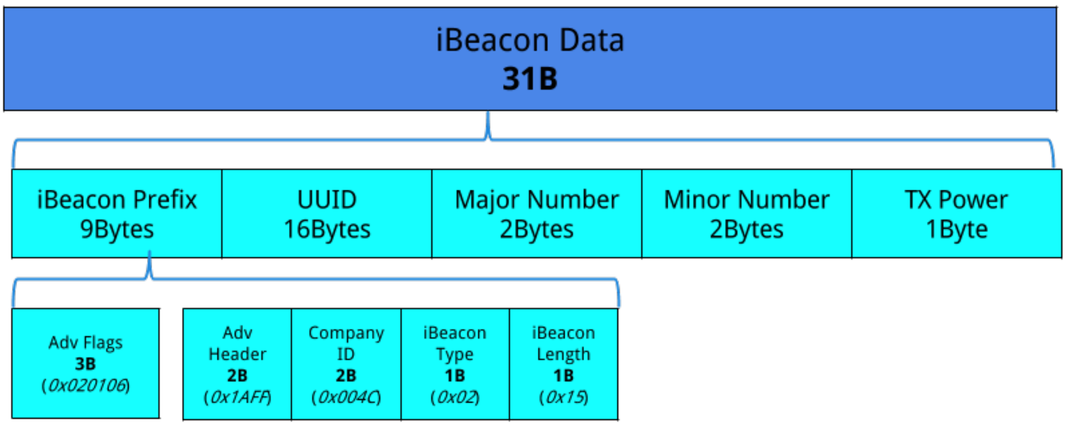

FIGURE 3.3: iBeacon protocol packet structure

Apple's iBeacon was the first protocol to be released. It was introduced in 2013 and is integrated in the Apple SDK under **CLLocationManager** library. It is a closed protocol but their packets can be read also by Android devices. Generally, the length of an iBeacon packet is 31 Bytes, divided as follow:

- 9 Bytes for *prefix*.
- 16 Bytes for *UUID*.

- 2 Bytes for *Major number*.
- 2 Bytes for *Minor number*.
- 1 Byte for *transmission Power*.

UUID uniquely identifies each beacon. It is also possible grouping beacons: the membership group of the beacon can be retrieved using the *Major number*, which is the same among beacons of the same group; one beacon in a group can be identified using the *Minor number*. Finally, the transmission Power indicates the signal strength one meter from the device; this parameter must be calibrated by the user or the manufacturer before the use.

The prefix field can be further divided into five parts:

- 3 Bytes for *advertising flag*.
- 2 Bytes for *advertiser header*.
- 2 Bytes for *company ID*.
- 1 Byte for *iBeacon type*.
- 1 Byte for *iBeacon length*.

AltBeacon

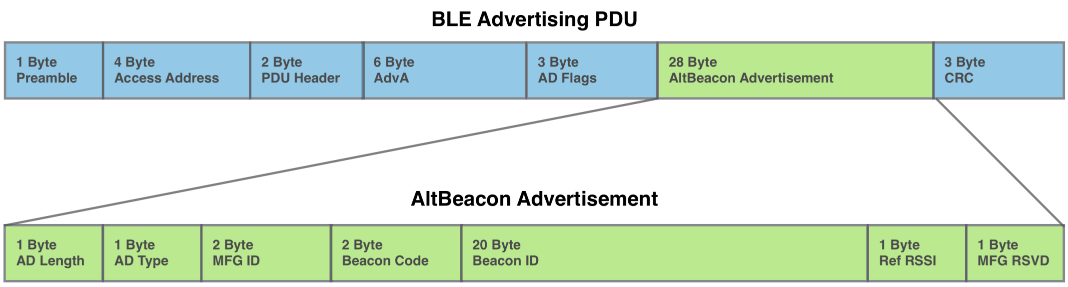

FIGURE 3.4: AltBeacon protocol packet structure

AltBeacon is an open source protocol provided by Radius Networks and realized in response to the iBeacon closed protocol. For this reason, its packet covers the same functionality as the Apple solution. The advantage of AltBeacon is that it can send more user data inside a single packet, which has here a total length of 28 Bytes. Up to 25 Bytes of them can be used for user information, against the 20 Bytes of iBeacon. AltBeacon packet is composed as follow:

- 1 Byte for *preamble*.
- 4 Bytes for *access address*.
- 2 Bytes for *PDU header*.
- 6 Bytes for *AdvA*.
- 3 Bytes for *AD flags*.
- 28 Bytes for *AltBeacon advertisement*.
- 3 Bytes for *CRC*.

URIBeacon

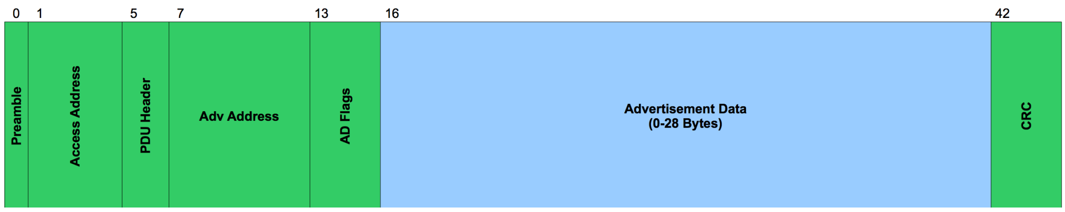

FIGURE 3.5: URIBeacon protocol packet structure

URIBeacon is a project started by Google. This protocol is similar to iBeacon, but adds a field in the payload to include an URL. For this reason, the device who receives URIBeacon packets can reacts directly to it by reading the URL. This feature makes the protocol similar to a QR code that works over BLE.

The total length of the packet is 31 Bytes, but is limited to 28 Bytes if advertising mode is used. To encode the URI 19 Bytes are consumed; prefixes (*www*, *http://*, *https://*, etc.) and suffix (*.com*, *.org*, etc.) are already encoded and may be ignored. This choice allows to save overhead space, but has the counter effect of limiting the URI types that can be sent.

Eddystone

Eddystone is another protocol developed by Google. It supports different kinds of packets. The first one, called *Eddystone-UID*, is very similar to Apple's iBeacon. The others are useful if there is the necessity to send additional information: it is possible to use *Eddystone-TLM* or *Eddystone-URL* packets,

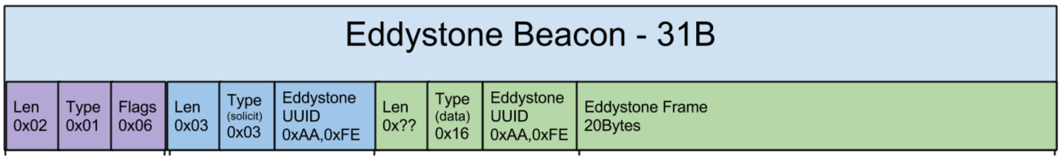

FIGURE 3.6: Eddystone protocol packet structure

which allow to integrate respectively information as battery voltage, temperature, beacon uptime, packets sent since last time or URL (like URIBeacon). Eddystone is an open source protocol and is fully compatible both with iOS and Android.

3.2.2 Possibles uses of beacons

Beacons have been used in recent years for different purposes. Mainly, they have been installed in shops to send advertisement messages: people who have the seller's application installed on their smartphone can intercept the packets sent from beacon and visualize the advertisement directly on their own display. Another usage can be found in the Major League Baseball Stadium (USA): beacons are used to perform the check-in when a spectator enters the stadium. Researchers are also trying to use beacons to produce indoor positioning systems, based on the power of the signal (RSSI) perceived by an electronic device. Another interesting use can be found in the healthcare sector: some beacon devices are used to track patients' movements inside their own homes. Beacons are also used in the business field to help employees to schedule the use of a conference room: a beacon positioned in a conference room can notify others employees (through an application) if that room is empty or not.

3.2.3 Manufacturers of beacon

The number of beacon manufacturers has been increasing over the last few years. According to (Shea, 2016), nowadays there are three main companies involved in the construction of this type of hardware: *Estimote*, *Kontakt.io* and *Radius Networks*.

FIGURE 3.7: From up-left: Estimote beacons, Kontakt.io beacons, Radius Networks beacons

Estimote

Estimote[1] is a company founded in 2012, specializing in the making of beacon devices. Their catalogue offers a wide variety of beacons, each one fitted to a specific functionality. All the devices have the same core: an *ARM Cortex* processor flanked by a *BLE card*. Depending on the model chosen, each beacon can integrate different sensors. For example motion, temperature, ambient light, magnetometer and pressure sensors are available to the developers in the Location Beacon version. The operating range is also different, from a minimum of $7m$ to a maximum of $200m$. The integrated battery can offer up to 5 years of use, but devices that are fed through power adapter are also available. Estimote beacons support both iBeacon and Eddystone protocols, so they are fully compatible with iOS, Android and Windows Phone devices.

Kontakt.io

Kontakt.io[2] is a company founded in 2013 specializing in the making of products for the Internet of Things. As Estimote, also beacon devices produced by Kontakt.io are compatible both with iBeacon and Eddystone protocols. The main difference with the former manufacturer is the software used to

[1]Estimote, `http://estimote.com`
[2]Kontakt.io, `https://kontakt.io`

support the devices: they allow not only to send data from beacon to the listening devices but also to collect data and send them to a server in order to execute any kind of jobs.

Radius Networks

Radius Networks[3], founded in 2011, produces not only hardware, but also the open source protocol AltBeacon. They make economic devices, with a price that goes from $10\$$ up to a maximum of $51\$$. This leads to the production of devices with a lower operating range (up to $50m$) and less battery duration (up to 285 days of life) than direct competitors like Estimote. Radius Networks supports not only its own protocol (developed in response to closed protocols) but also Apple's iBeacon and Google's Eddystone. It was also the first company to be certificated as official supporter of iBeacon protocol in 2013.

3.2.4 Optimal environment configuration

Before using beacons for the indoor localization, different phases are needed to calibrate and discover the best configuration parameters, such as *transmission power* and *height of the beacons*.

For the purpose of this thesis three Estimote Beacon have been used. In particular, the Location Beacon model - that offers a Bluetooth range up to $200m$ and a battery life of 5 years – has been chosen. The first step was the selection of a beacon protocol. As mentioned before, Estimote suggests two different solutions: iBeacon and Eddystone. The former has been configured on the three beacons through the cloud web interface. Since a mobile application was required to exchange packets and iBeacon fully works only on iOS devices, an iPhone app was developed. This application reads each packet sent from the beacons and analyzes it. iPhone 6S with iOS 10 was used to perform tests.

Since neither beacons or iOS SDK can estimate the distance between each beacon and the receiver, a calculation is required. For this reason the RSSI value has been used. This is the general idea: initially it is needed to measure the transmission power at a known distance (during tests a distance of 1 meter has been chosen); this measure will be referenced in the pseudocode as txPower. Knowing this value, it was possible to evaluate the distance

[3]Radius Networks, `https://www.radiusnetworks.com`

between beacon and device using the RSSI, that is a value always available inside iBeacon packets. The final algorithm is expressed in Listing 3.1.

LISTING 3.1: RSSI estimated distance

```
def calculateDistance(txPower, rssi):
  if (rssi == 0):
    return -1.0

  ratio = rssi*1.0/txPower
  if (ratio < 1.0):
    return pow(ratio,10)
  else:
    distance = (0.89976)*pow(ratio,7.7095) + 0.111
    return distance
```

Constants values used in the algorithm have been retrieved from (Radius-Network, 2014). One of the key settings for this algorithm is to determine the best height where to place each beacon. Therefore some tests have been carried out to find it.

Tests to identify best height

To obtain the best height on which position beacons, the same beacon was placed in a free-of-noises room at different heights. For each one, the same iPhone was positioned at seven different distance points and the position, calculated through RSSI information, was calculated and memorized for an interval of 4 minutes for each point. In all the tests, Broadcasting Power was setted to $+4dBm$. This is the list of the seven tests with the related real distance of the iOS device from the beacon:

- Test 1: 0.00m
- Test 2: 0.20m
- Test 3: 0.50m
- Test 4: 1.00m
- Test 5: 1.20m
- Test 6: 1.50m
- Test 7: 2.00m

All the seven tests have been done at four distinct heights, listed below:

- Height 1: 0.00m
- Height 2: 0.49m
- Height 3: 1.80m
- Height 4: 2.00m

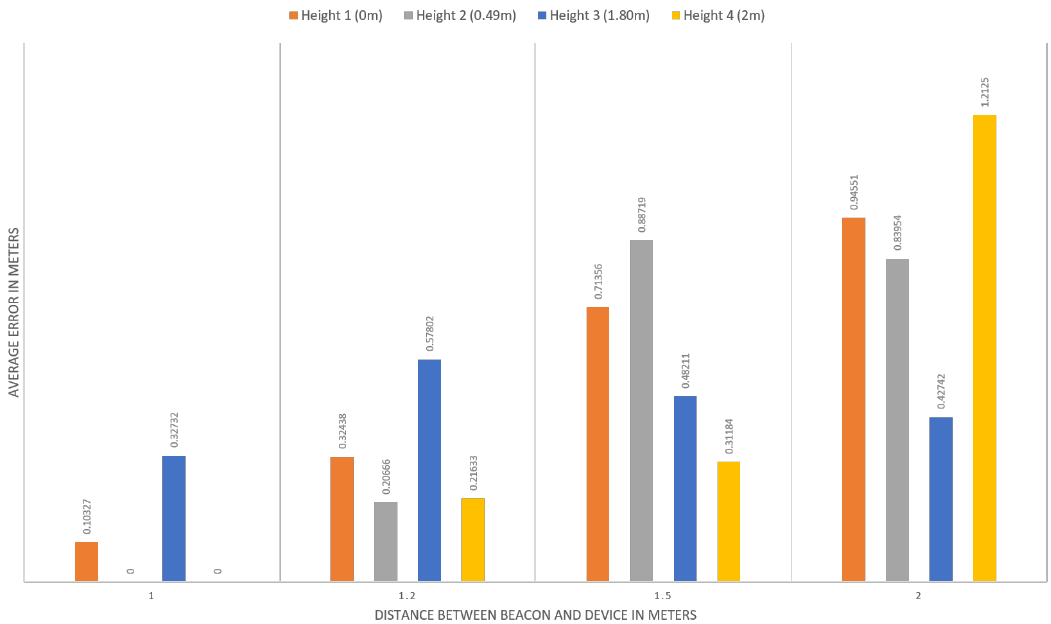

FIGURE 3.8: Average Error of height tests with beacons

Since it was a static test, during the 4 minutes of sampling both beacon and smartphone were never moved from their positions. At the end of the sampling phase, all the measurements were elaborated in order to obtain an average error related to the real and the estimated distance. The results are represented in Figure 3.8 (notice that tests 1,2 and 3 are not reported due to their very low error).

During tests from 1 to 4, all the four heights have reported an average error between $0.10m$ and $0.40m$. In tests 4 and 5, height 1, 2 and 4 have a better estimated distance, with an average error up to $0.32m$. In tests 6 and 7, the average error increases in all three cases 1,2 and 4, a better result has been obtained in height 3, with an error under the threshold of $0.50m$. In a real case usage, people will be at a distance greater than $1.5m$ from each beacon, so it is reasonable to choose height 3 as the best one to position each beacon, that is the height with a lower average error in test 6 and 7. The next step will be to choose the best Broadcasting Power of each beacon.

Tests to identify best broadcasting power

In order to choose the best broadcasting power, tests are performed with one beacon positioned at a height of $1.80m$. This height was measured as the best on which to set the devices, because in a static test and in a controlled environment, it gives a lower average error on the middle-long distance (see previous section).
Broadcasting power of Estimote beacon can be configured at three different values through the web cloud interface:

- Power 1: -16dBm
- Power 2: -8dBm
- Power 3: +4dBm

For each power, different tests have been performed (always in a static case) measuring the average error between the real distance and the estimated one. Tested distances are in a range from $0m$ up to $3m$. The sampling time was 4 minutes for each test. In this phase, beacon and iOS device (which perceive, calculate and memorize each estimated distance) are in a controlled environment, which means that there aren't people or others significant obstacles which can interfere with bluetooth signal. To ensure this, smartphone is also located on a rubberised tripod.

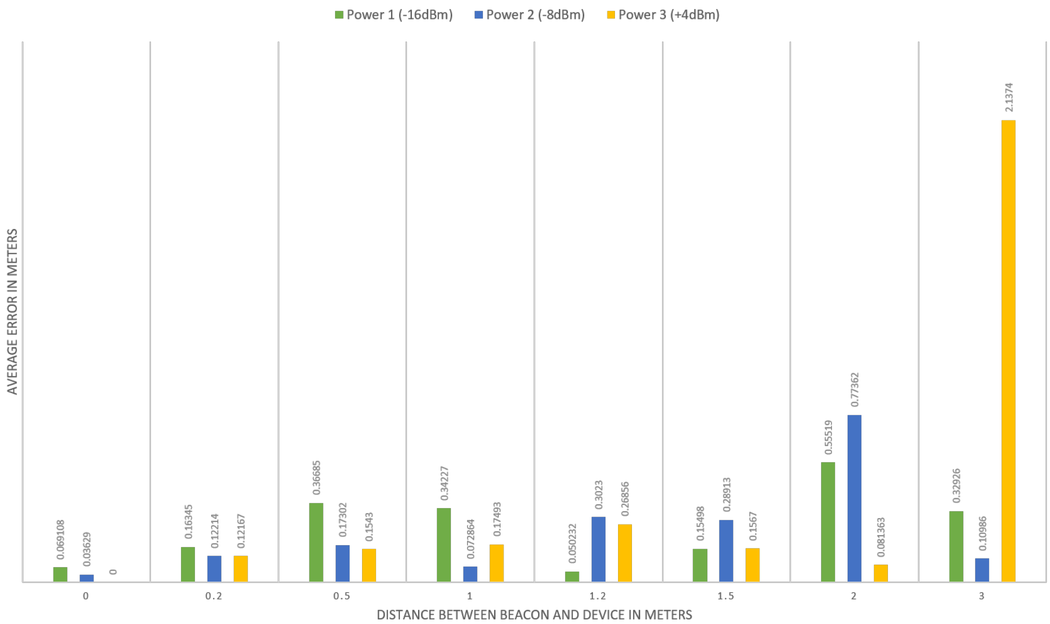

FIGURE 3.9: Average Error of power tests with beacons

Data analysis has reported that Power 2 has a lower average error on small, medium and long distances. Until $2m$, Power 3 return good results, with an average error under $0.5m$. The problem with this setting is that when the device is positioned at a distance greater than $2m$, the estimated distance becomes casual, with an average error that is more than $2m$. For this reason Power 3 was discarded. Comparing Power 1 and Power 2 is possible to detect a similar behaviour. Indeed, from $0m$ up to $1.2m$, Power 2 behaves slightly better than Power 1 but both powers returns an average error under the threshold of $0.5m$. As depicted in Figure 2.7, from $1.2m$ up to a distance of $2m$, the average error is similar both for Power 1 and Power 2 and it is less than $1m$. The average of all measurements has highlighted that in general Power 2 has a better accuracy than Power 1. The analysis of this results has led to choose Power 2 as best broadcasting power with beacon positioned at $1.8m$ of height. For the next phase, each beacon will be positioned at $1.80m$ of height, with a broadcasting power of $-8dBm$.

3.2.5 Trilateration technique

As explained in (Cellar, 2014), *Trilateration* is a technique that permits to obtain the position of an object in the space. Initially, the distance between the receiver and the transmitter must be measured: this can be estimated, when beacons and iBeacon protocol are used, through the RSSI info. The estimated distance allows to draw a virtual circle around the transmitter (where it is the ray of this circle). The circumference area represents all the possible points in which the object could be located. Adding another transmitter, it is possible to draw a second circle that intersects the first one in two points. These could be the possible positions of the object to be localized. With a third transmitter, this ambiguity disappears. So, to estimate the position in a two-dimensional space, three transmitter and one receiver are needed (see Figure 3.10). If there is the necessity to position an object in a three-dimensional space, it is not possible to use three transmitters only. In fact the problem is that, with each estimated distance, a sphere is drawn and the interception of three spheres is a circle. As a consequence, a fourth transmitter must be used to eliminate the ambiguity.

Following are examples of formulas used to estimate abscissa and ordinate of one object on a two-dimensional Cartesian space, with three transmitter and one receiver:

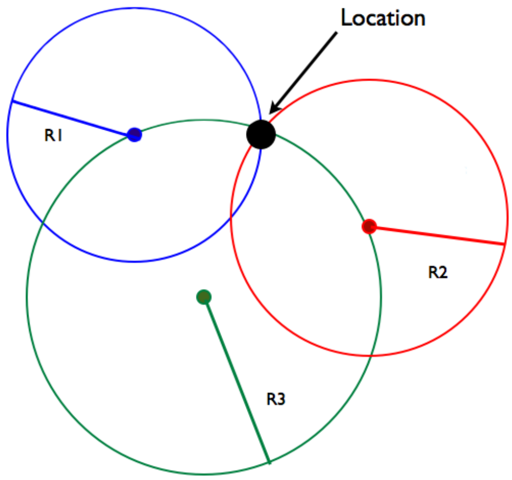

FIGURE 3.10: Trilateration technique

$$x = \frac{(d_1^2 - d_2^2 - i_1^2 - j_1^2 + i_2^2 + j_2^2)(j_3 - j_2) - (d_2^2 - d_3^2 - i_2^2 - j_2^2 + i_3^2 + j_3^2)(j_2 - j_1)}{2((i_2 - i_1)(j_2 - j_3) - (i_3 - i_2)(j_2 - j_1))} \tag{3.1}$$

$$y = \frac{(d_1^2 - d_2^2 - i_1^2 - j_1^2 + i_2^2 + j_2^2) - 2x(i_2 - i_1)}{2(j_2 - j_1)} \tag{3.2}$$

where:

- d_i represents distance between transmitter i and receiver;
- i_l represents abscissa position of transmitter l;
- j_m represents ordinate position of transmitter m.

3.2.6 Static trilateration test for indoor positioning

For indoor positioning using trilateration technique, TE1 has been chosen as test environment. The target of the test was to verify if it was possible to use beacons (with the configuration identified during the previous analysis) and trilateration technique to estimate the position of a person in a room. For this experiment only three beacons were available, so there wasn't the possibility to execute a test in a three-dimensional space. To avoid any kind of errors, both beacons and smartphone were located on the same plane. In particular, the chosen plane was the TE1 floor. The first step was to execute static test in a controlled environment. On the room floor, a right-angled triangle has been drawn, having both the two catetis and hypotenuse of $2m$. Inside the triangle 10 squares have been drawn, each of them with a side of $0.50m$ (half of the squares that were on the hypotenuse were out of the triangle, but these parts have also been included during the test). An area of $0.25m^2$ has been chosen because, ideally, it can contain entirely one person. The three beacons have been located at the three vertices of the triangle. In this phase, 10 tests have been carried out and each one of them has involved a different square drawn on the floor. For each test, the smartphone was positioned exactly in the centre of the square and the packets sent from beacons were processed in order to obtain an estimated position of the device in the space. The sampling time was 1 minute for each test. During the execution, there weren't people in the room, in order to avoid any kind of disturbance. At the end of each test, it has been calculated the Least Square Error (*LSE*) between the real position and every single measurement. Then the average of these values has been elaborated in order to obtain an average error between the real and the estimated position. In the Figure 3.11, the triangle with square subdivision is shown.

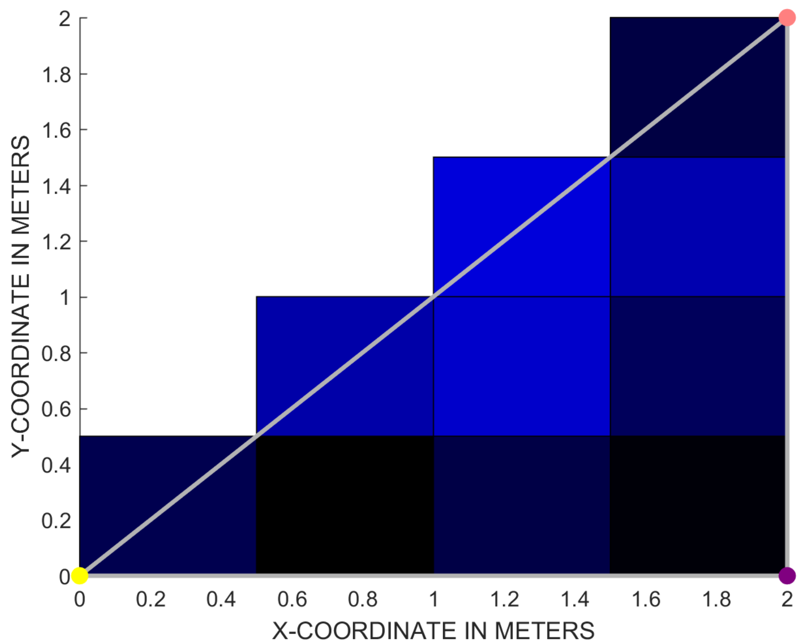

FIGURE 3.11: Trilateration test results with beacons

Each square is coloured with a different blue tint, where a lighter blue represents a low error. On the contrary, a darker blue tint represents a higher error. The tests have highlighted that there is a higher error when the smartphone is located in one of the three corners of the triangle. With this configuration, the device is at a distance of $0.25m$ from the nearest beacon, but at the same time it is at $1.75m$ from the others two beacons. For example, when the smartphone is located in the upper corner, the average LSE between the real position and the estimated one is $0.85m$. For the left-bottom and the right-bottom corner the LSE is similar, with the former value of $1.28m$ and the latter one of $1.13m$. Smartphone positioned in the middle of the triangle, instead, returns a LSE of $0.33m$. It hence become evident how difficult it is to obtain an accurate estimated position when there is more than $1.5m$ of distance between receiver and beacons. This thesis is confirmed by the value of average error calculated when the device is located in a corner of the triangle, where the receiver is positioned from two transmitters at a distance higher than $1.45m$.

3.2.7 Conclusions

The goal of this section was to evaluate the performance of beacons when they are used to estimate the position of an object in a two-dimensional space. Initially the goal was to found the best settings of the environment. First tests were necessaries to define the height on which to position each beacon. Some experiments have highlighted the need to set them at a height of $1.80m$, that is the average height of a person. Once this was found, the next step was to set up the broadcasting power of beacons. Estimote beacons permit to configure three different values: $-16dBm$ (weak), $-8dBm$ (normal), $+4dBm$ (high). The best broadcasting power (at the height of $1.80m$ found in the previous step) results to be $-8dBm$.

Once the configuration phase ended, the last step was to use trilateration technique to estimate the position of a person in a room. In a controlled environment with a small number of interferences, the LSE of real position with reference to the estimated one was too high. The problem is that when the receiver is positioned at a distance more than $1.45m$ from the transmitter, distances are not so accurate. Also, small disturbances lead to an increment of the average error.

These data allow to conclude that it is not possible to use beacons in a real environment to estimate the position, because the receiver, most of the time, is far more than $1.45m$ from the transmitter and it is not feasible to control disturbances (introduced by a lot of factors as people, walls and windows) that create conflict with bluetooth frequencies.

3.3 Ultra-Wide-Band

One of the problems of radio signals is their propensity to disturb themselves when operate on the same frequency, because there is a high probability that data sent over the channel could collide. The solution is to use different frequencies to avoid such disorders. *Ultra-Wide-Band* (*UWB*) technology tries to resolve this kind of problem. As described in (Rouse, 2008), the solution adopted is to send a large amount of data over a wide spectrum of frequencies. This permits to avoid to operate on the same frequencies of others near devices, eliminating interferences and noises. Frequency spectrum is regularized in order to avoid that different technologies could operate on the same frequency. Generally, to each technology is permitted to use only a limited spectrum: UWB could operate on a wide range of $500MHz$. This is allowed because it operates at a very low power, so each single pulse emitted could be perceived only at little distances. Used power is less than $0.5mW$, which allows to reach a distance of $100m$. Operating with this range, there are less possibilities that UWB signal can collide with others radio signals. All this characteristics allows not only to send a large amount of data, but also to carry signals through doors, walls and other obstacles, that generally tend to reflect radio signals which operates in a more limited range of frequencies and at a higher power. Using a very low power, the pulse percepts by the receiver will likely result to be below the noise level. To avoid this problem, a training phase is needed: transmitter and receiver must be coordinated in order to be able to send and receive pulses with a high accuracy. During training phase, some pulses (tipically between 128 and 1024 that represents a single bit of information) are sent. The receivers accumulate these pulses and learn how to recognize UWB signal in order to avoid treating them as noises pulses. This new technology can be compared with bluetooth: both operates at short-distances, but UWB has the advantages of being less susceptible to disturbances.

3.3.1 Ultra-Wide-Band signal propagation

One of the fundamental question that arises is about the UWB through-the material propagation capabilities. In fact, there are multiple type of materials commonly encountered in building environments (as in test environment TE1 and TE2 described previously). Concerning that, this section briefly reports experimental results and textual conclusions of a research which had the objective to examine propagation through walls made of typical building materials and thereby acquire ultra-wide-band characterization of various materials. For an in-depth analysis of the research, refer to the original work by (Safaai-Jazi et al., 2002).

Electromagnetic characterization of materials and walls commonly encountered in indoor environments was undertaken with the aim of assessing their impacts on UWB propagation. For this scope, ten different wall materials were selected for UWB characterization. Among them, plasterboards, glass, wallboard, styrofoam, cloth office partition, wooden sample door, wood, structure wood, brick, concrete block, and reinforced concrete column. Table 3.1 lists the selected samples and their dimensions. Thickness data are taken as an average of 6 to 8 repeated measurements for best accuracy; they are reported as the first number in the dimensions column (highlighted in bold).

TABLE 3.1: Experiments results of signal propagation through different materials @5GHz

Material	**Dimensions (cm)**	ε_r	**Loss (dB)**
Drywall	**1.16992** x 121.8 x 196.9	2.44	0.45
Cloth Partition	**5.9309** x 140.7 x 153.1	1.23	2.55
Structure Wood	**2.06781** x 121.5 x 197.8	2.11	1.35
Sample door	**4.44754** x 90.70 x 211.8	2.08	2.0
Ply Wood	**1.52146** x 121.9 x 197.51	2.49	1.75
Glass	**0.235661** x 1.44 x 111.76	6.40	1.25
Styrofoam	**9.90702** x 121.8 x 197.7	1.11	0.10
Bricks (single)	**8.71474** x 5.82676 x 19.8	4.22	6.45
Concrete Block	**19.45** x 39.7 x 19.5	2.22	13.60
Reinforced Concrete	**60.96** x 121.92 x ...	-	-

In the third column is reported the *relative permittivity* ε_r (also commonly known as *dielectric constant*) of the material. The relative permittivity of a material is its (absolute) permittivity expressed as a ratio relative to the permittivity of vacuum. It is a material property that affects the Coulomb force between two point charges in the material. Relative permittivity is the factor by which the electric field between the charges is decreased relative to vacuum.

The fourth column contains the amount of signal loss (in *dB*) detected during experiments. As it can be seen, this quantity may increase up to 150 times to the minimum one (approximately equal to 0 when there is free-space). The reinforced concrete wall (last row) resulted in a very small amount of received power. Due to this fact, no further processing could be done and signal loss could not be measured (or, more generically, it can be intended as an impossible transmission).

To gain more insight into the effects of various walls on the propagation of UWB pulses, the free-space and through signals for three of the measured materials are presented in Figures 3.12, 3.13 and 3.14. These three samples have been chosen because they characterize quite well TE1 and TE2.

Figure 3.12 represents the propagation through drywall; as it can be noted there's no significant difference between free-space and through material amplitude. On the contrary, Figures 3.13 and 3.14 report propagation through bricks and concrete block respectively. Here it is evident how the amplitude becomes biased.

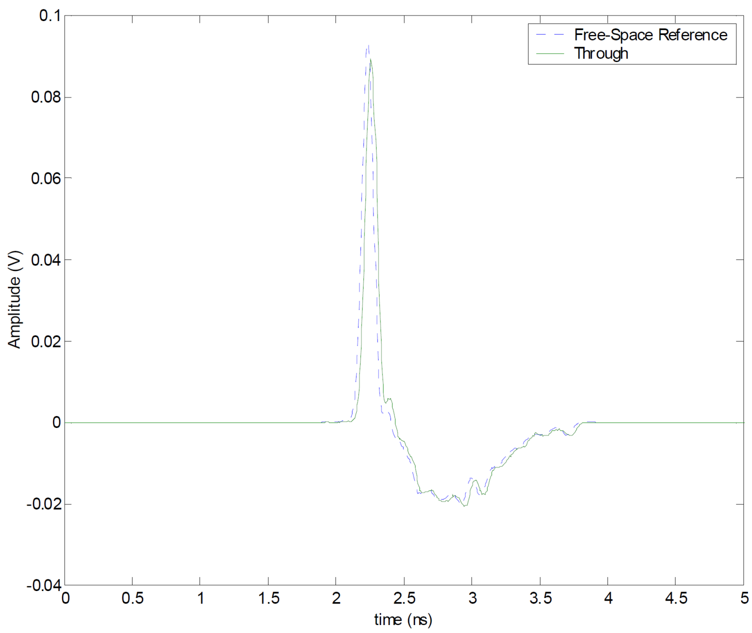

FIGURE 3.12: Drywall 'free-space' and 'through' propagation

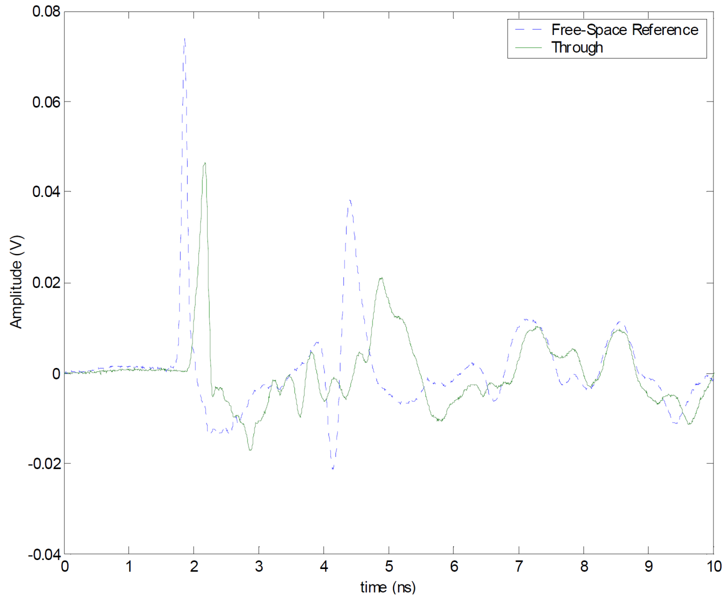

FIGURE 3.13: Bricks ‘free-space’ and ‘through’ propagation

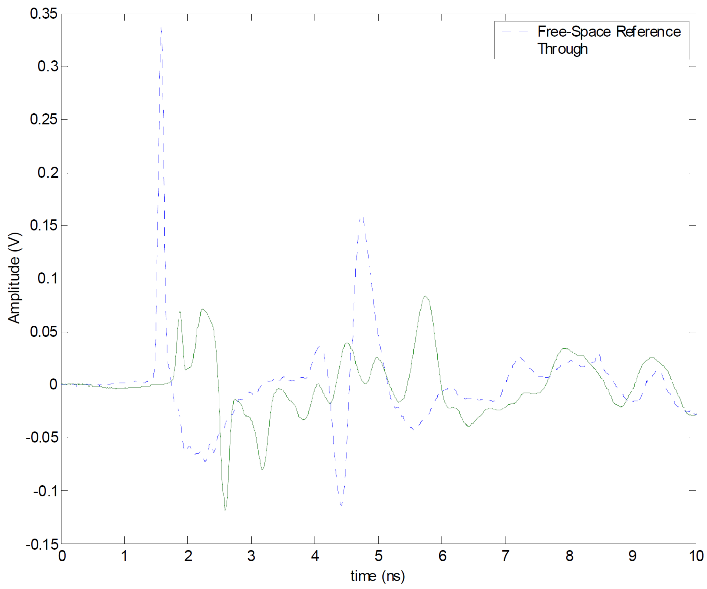

FIGURE 3.14: Concrete block ‘free-space’ and ‘through’ propagation

3.3.2 Possible uses of Ultra-Wide-Band

UWB features described previously permit to use this technology in a wide range of short-distance applications. In some circumstances it can replace bluetooth technology. For example, it is used to transfer photos or media files from a digital camera to a printer, in order to print them without using a PC. The feature to pass through objects has also permitted more advanced uses: for example, UWB technology is used for precision radar imaging, useful to locate objects through walls. The wide spectrum of frequencies, that permits to avoid interferences, has also lead to some studies in order to use UWB for indoor positioning. UWB was also proposed to be used in Personal Area Networks and in particular in IEEE 802.15.3a draft PAN standard. Still, high production costs and slow progresses over the years are the main causes of a slow growth of this technology in consumer market.

3.3.3 Pozyx

Nowadays there are few companies that produce UWB devices. One of these is Pozyx[4], a startup founded in 2015 after a crowdfunding campaign regarding their first products.
A Pozyx system is made of a *tag* and one or more *anchors*. Each of these elements integrate an UWB antenna; furthermore, some sensors are integrated in tag device, like accelerometer, gyroscope, magnetometer and altimeter. The core of a Pozyx tag is a microcontroller unit, which interacts with all the sensors and performs some kind of operations such as ranging, positioning and calibration. The board could operate in two different ways: it could be mounted on an Arduino (through the integrated pin) or it could be connected to an external device (like a Raspberry Pi or a PC) using an USB serial interface. In the first case it is possible to use Arduino library to communicate with Pozyx tag, otherwise a Python library is available for the serial connection. In both cases, the communication is managed by tag microcontroller in real-time.

3.3.4 Distance estimation with Pozyx

Distance measurement between UWB transmitter and receiver is performed differently in regard to beacons. In the latter case, transmission power was used to estimate the distance. The problem of this solution was that the signal

[4]Pozyx, `https://www.pozyx.io/Documentation`

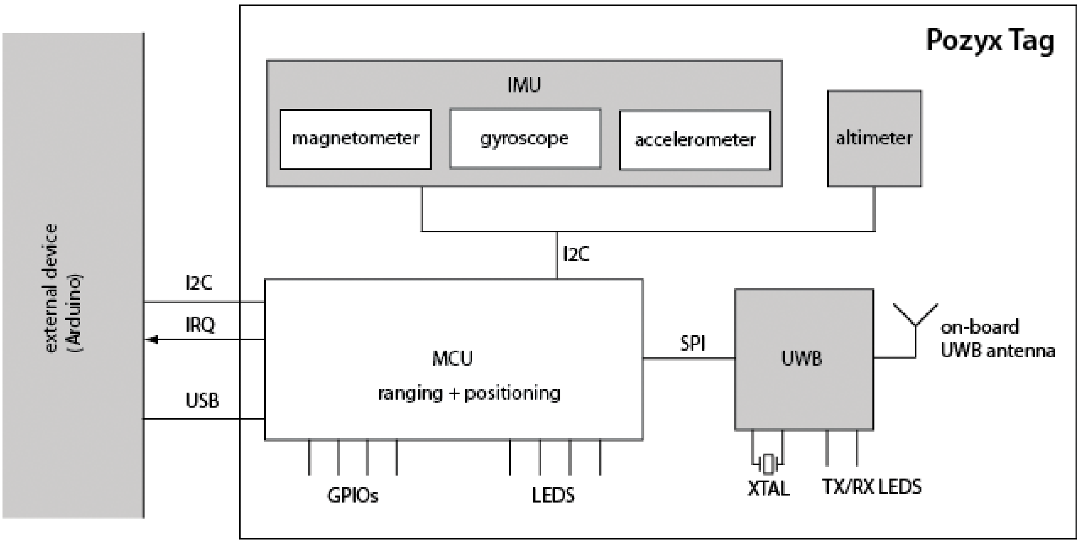

FIGURE 3.15: Pozyx tag device datasheet

(then the transmission power) was subject to noises and reflections, which made it impossible to obtain an accurate result. Pozyx has implemented a *two-way ranging mode* to perform distance estimation, that consists in sending a message back and forth. Using this technique, a radio wave is sent from one device to another and it is measured how long it takes. It is known that radio waves travel at the speed of light, so it is easy to divide the velocity by the estimated time of flight, thus obtaining the distance between the two devices. The problem of this solution is that radio waves move fast, so it is necessary to accurately estimate the timing of the signal pulse. The solution can be found in *Heisenberg's uncertainty principle*:

It is impossible to know, at the same time, both the timing and the frequency of a signal.

Generally, the range of frequencies (or bandwidth) of a signal is known and it is possible to obtain the pulse timing through this formula:

$$\Delta f \Delta t \geq \frac{1}{4\pi} \tag{3.3}$$

So, to obtain a small timing, it is required to have a wide range of frequencies. Pozyx implementation of UWB uses a bandwidth of $500MHz$ which permits to obtain a pulse timing of 0.16ns. This result permits to distinguish several reflections of the signal. Hence, it remains possible to do accurate ranging even in places with a lot of reflectors, has it happened in indoor environments. What has been said so far explains why it is not possible to use

Wi-Fi signal to obtain an accurate precision in a positioning system: Wi-Fi bandwidth is $20MHz$ and that permits to obtain a timing of 4ns. Using this value in association with the speed of light, a too long impulse that blurs with signal refractions is obtained (in practice, it can no longer distinguish the real signal from the signals bouncing onto objects).

3.3.5 Position estimation with Pozyx

To obtain an estimated position it is needed to perform measurements and to have some reference points to be compared. In a system composed by Pozyx devices, anchors are the reference points, meanwhile measurements are the distance estimations between tag and each anchor. This configuration permits to use different algorithms in order to obtain a position on a two-dimensional or three-dimensional space using respectively three or four anchors paired to a tag. The most common algorithm, that has been also used for beacons experiments, is the trilateration one. It uses the basic geometry concepts of drawing a circle around each anchor with a ray given by the distance between anchor and tag. The intersection of the three circles (in a two-dimensional space) permits to estimate the position of an object. For more details, the reader is invited to read the previous sections. One of the problems of this algorithm is that measurements are not so accurate. Circles could intercept in more than one point, so it is difficult to estimate an accurate position. The basic idea (also used in Pozyx) to have a better estimation is to found exactly one point that is nearest to all the three circumferences. The starting point to find it is the distance between the user and anchor i, that is denoted by symbol d_i:

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{3.4}$$

with x and y who identifies user position ($\vec{p}$), x_i and y_i who identifies anchor i position ($\vec{p_i}$, that is known in advance) and with i that goes from 1 to N (N is the number of anchors). Solving the square of left and right side of equation, (3.4) become:

$$d_i^2 = (x - x_i)^2 + (y - y_i)^2 \tag{3.5}$$

$$d_i^2 = x^2 + x_i^2 - 2xx_i + y^2 + y_i^2 - 2yy_i \tag{3.6}$$

The problem with (3.6) is that it's non-linear in x_2 and y_2. So, it is needed to eliminate these two terms by subtracting d_N^2 from d_i^2, obtaining $N-1$ equations:

$$d_i^2 - d_N^2 = -2x(x_i - x_N) + x_i^2 - x_N^2 - 2y(y_i - y_N) + y_i^2 - y_N^2 \tag{3.7}$$

These operations permit to obtain linear equations in the unknown coordinates x and y. It is possible to write the entire equations in a matrix form:

$$b = A \begin{bmatrix} x \\ y \end{bmatrix} \tag{3.8}$$

with:

$$b = \begin{bmatrix} d_1^2 - x_1^2 - y_1^2 - d_N^2 + x_N^2 + y_N^2 \\ d_2^2 - x_2^2 - y_2^2 - d_N^2 + x_N^2 + y_N^2 \\ \vdots \\ d_{N-1}^2 - x_{N-1}^2 - y_{N-1}^2 - d_N^2 + x_N^2 + y_N^2 \end{bmatrix} \tag{3.9}$$

and:

$$A = -2 \begin{bmatrix} x_1 - x_N & y_1 - y_N \\ x_2 - x_N & y_2 - y_N \\ \vdots & \vdots \\ x_{N-1} - x_N & y_{N-1} - y_N \end{bmatrix} \tag{3.10}$$

In a two-dimensional space with only three anchors, it is therefore possible to find user position by resolving only this equation (3.11):

$$\vec{p} = A^{-1} b \tag{3.11}$$

The algorithm just described is called *Linear Least Squares algorithm*. It is *linear* because it is needed to linearize the equations and *least square* because inverting the A matrix will result in a minimization of the squared error of all equations.

3.3.6 Optimal environment configuration

All the following tests have been performed using a Pozyx system composed by 4 anchors and 1 tag. As done with beacons devices, also Pozyx require to perform some tests to reach the optimal configuration of the environment.

3.3.7 Tests to identify best height

Pozyx documentation suggests to set the four anchors at least at human height, with the opportunity to position them on ceiling. A set of tests have been performed to confirm this thesis. Two heights have been chosen as test height:

- Height 1: 0.00m
- Height 2: 1.59m

For each of these, eight tests have been performed where tag device has been located in a different distance from the anchor:

- Test 1: 0.00m
- Test 2: 0.20m
- Test 3: 0.50m
- Test 4: 1.00m
- Test 5: 1.20m
- Test 6: 1.50m
- Test 7: 2.00m
- Test 8: 3.00m

Tag and one of the anchors were statically positioned in TE2 without any kind of noises. Data recorded in a sampling period of 2 minutes was used to establish the average error between real and estimated distance. The latter one was calculated using the two-way ranging described in the previous sections. The results are reported in Figure 3.16.

Height 2 has reported the best results. This confirms Pozyx thesis, who suggests to collocate the anchors (at least) at human height. The average error is about $5cm$ in tests where anchor was positioned at Height 2, with a peak of $12cm$ when tag is situated at a distance of $1m$. Instead at Height 1 the mean of average error is $15cm$, with peaks of more than $30cm$.

These tests have permitted to conclude that, with anchor located at high height (as suggested by manufacturer) and in a controlled environment with statically positioned devices, it is possible to obtain an error less than $10cm$ also at long distances. Next step will be to try to estimate the position of an object.

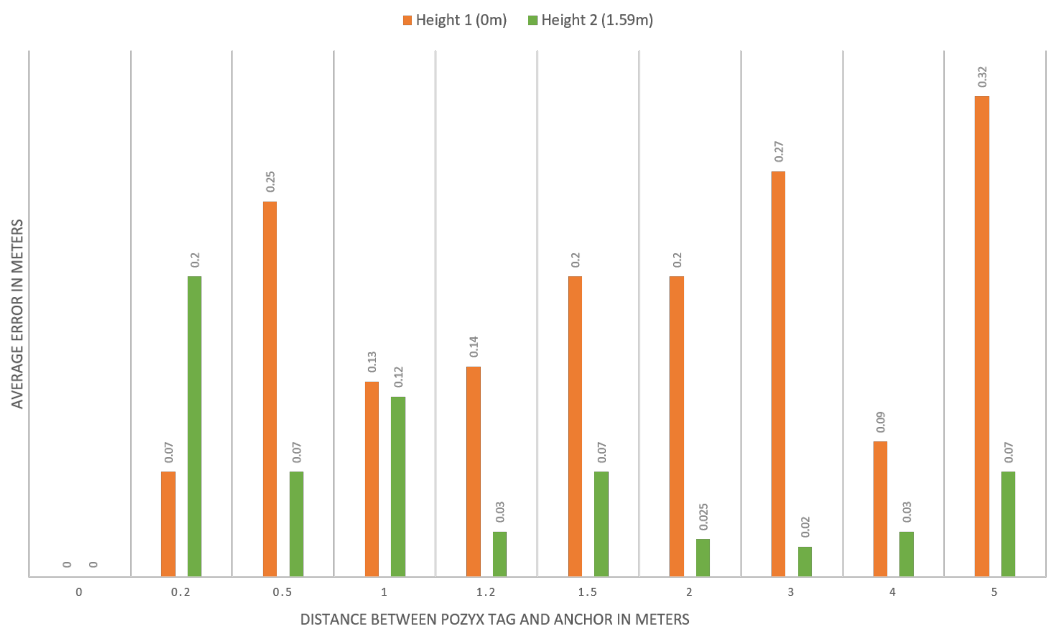

FIGURE 3.16: Average Error of height tests with Pozyx

3.3.8 Static positioning test for UWB indoor positioning

The objective of these experiments was to verify if Pozyx technology could be used for indoor positioning and if a low average error (as found in previous section) is maintained also in positioning. The environment chosen to perform this tests was TE1. As done with beacons trilateration tests, a right angled triangle with both catetis and hypotenuse of $2m$ has been drawn. It was sectioned in 10 squares, each one with a side of $0.50m$. Four anchors have been positioned around the triangle, maintaining a minimum height of $1.59m$ for each one. Then, 10 experiments have been performed. For each test, Pozyx tag was positioned in the middle of the current square and the estimated position was measured for a sampling time of $4min$. At the end of each tests, LSE between real and estimated position was elaborated and then the average error between all the LSE was calculated. In these static tests, estimated positions result to be very accurate. The resulted average error is below the threshold of $5cm$ for all the tests, with a best result of $4.2cm$. These data confirm the good results obtained and allow to perform further tests in a dynamical mode inside a more big environment.

3.3.9 Dynamic positioning test for UWB indoor positioning

In order to verify if Pozyx devices could be used to track user movements both in a little and in a big environment, another set of experiments has been carried out. For this purpose, both TE1 and TE2 have been used as test environments. This has allowed not only to perform tests in areas of different size, but also to verify if walls, windows and other elements which lead to loss of a direct line-of-sight between tags and anchors could generate problems.

Little environment with direct line-of-sight

First test has been performed in TE1, a small environment with no obstacles that can interfere with line-of-sight. All the four anchors have been used and were positioned on different walls near the corners of the room at various heights (but still holding the minimum required height found in the previous tests). An user, with a Pozyx tag in his hands, walks along the room. Some

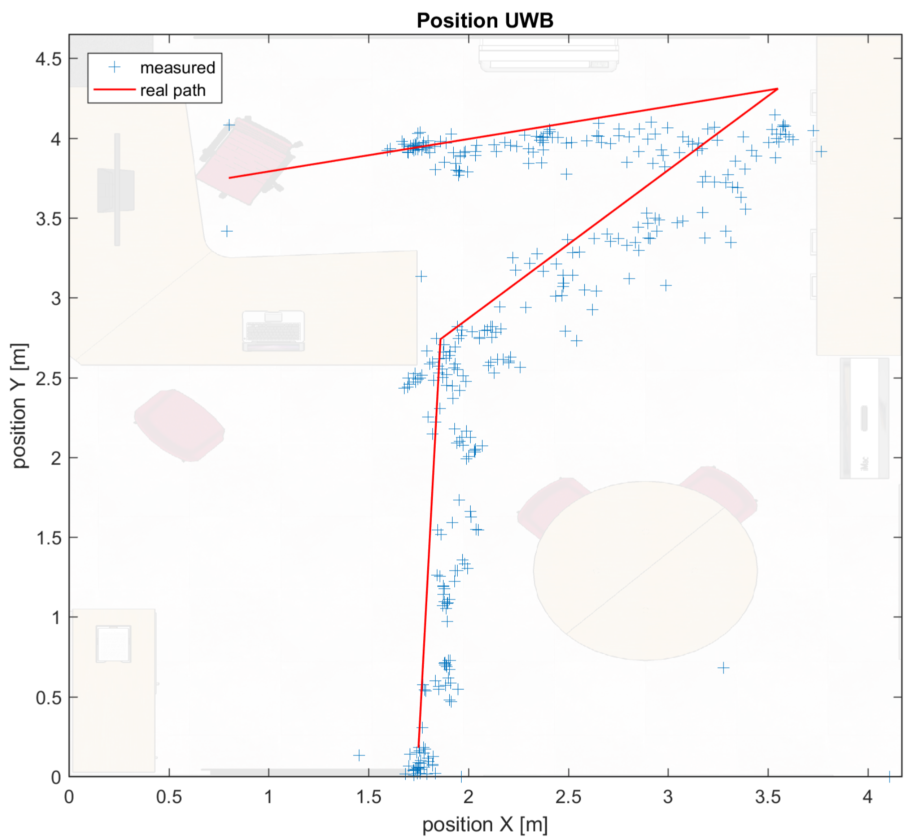

FIGURE 3.17: UWB dynamic positioning in TE1

tracks were drawn on the floor and the estimated position (related to time unit) has been stored to verify the correlation between real and estimated track of the user. As Pozyx tag was kept in hands, it is important to remember that a human error might occur (tag could not precisely follow tracks drawn on the floor but its position could vary of some centimeters).

All the tracks were followed precisely by the positioning algorithm; for example, as you can see in Figure 3.17, there is an error less than $10cm$ and the track is followed congruently. There are few outliers which should be cut off by a filtering algorithm, but in general the average error is good and follows the guidelines declared by Pozyx company.

Big environment with no direct line-of-sight

Starting from the good results of the previous section, it would be useful to know how the system works in a bigger environment. In this phase, it was necessary to use an environment with obstacles of different kinds: for this reason TE2 has been chosen. This stage was divided into two tests: the first one has involved half part of TE2, in order to see if it was possible to operate in a large environment. Given the results of this, the entire surface of TE2 has been used for the second test. In particular, these are the covered areas for both tests:

- Test 1: $150m^2$
- Test 2: $370m^2$

Test 1 results

The peculiarity of this test is that anchors have not been positioned along the perimeter of the environment. Instead, the chance to position them in strategic points of the environment in order to obtain a better coverage of the entire surface has been exploited. In particular, anchors have been positioned in four different places, with each pair of devices separated at most by one or two plasterboard walls or by one windows. As consequence, Pozyx tag was not in a direct line-of-sight with more than one anchor at a time.

Before performing a dynamic test, a static one was necessary to establish the average error between the real and the estimated position: in an environment of $150m^2$, error increases of an order of magnitude, with values less than $50cm$. This result was confirmed during the dynamic tests. As an example, in Figure 3.18 you can notice that the estimated position follows the real track of the user in the environment. So, even with no line-of-sight and in

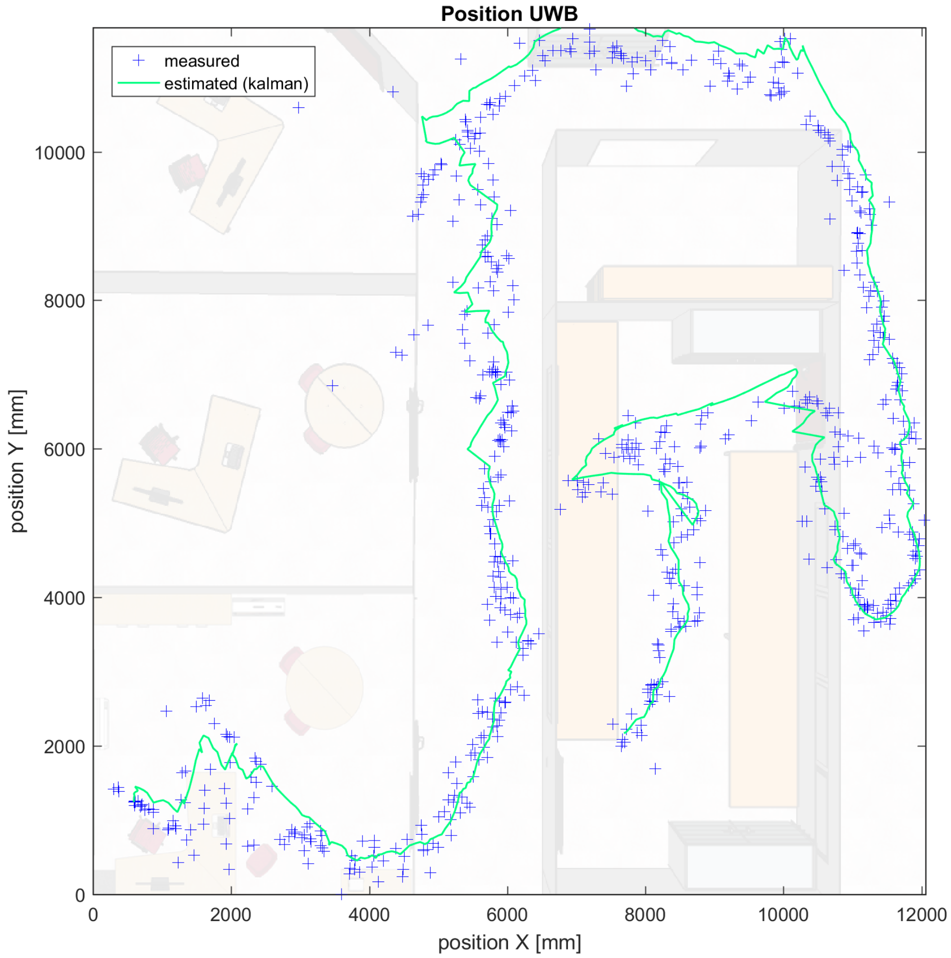

FIGURE 3.18: Test 1 results of UWB dynamic positioning in TE2

a bigger room, results permit to have an accurate precision (the error is less than a human step).

Test 2 results

In Test 2, the entire surface of TE2 has been used, in order to have a coverage area of $370m^2$ (compared to the previous test, few anchors have been moved to expand it). During the test, in some zones an error between $30cm$ and $50cm$ was found, but it was furthermore possible to notice that there were some shadow areas, which means that it was difficult to precisely follow precisely the real track of the user. As a consequence, we can say that it was always possible to determine precisely only the zone of the user. This highlight that four anchors can be enough for an approximate position (within $1m$ of accuracy), but also that more than 4 anchors are required to obtain a

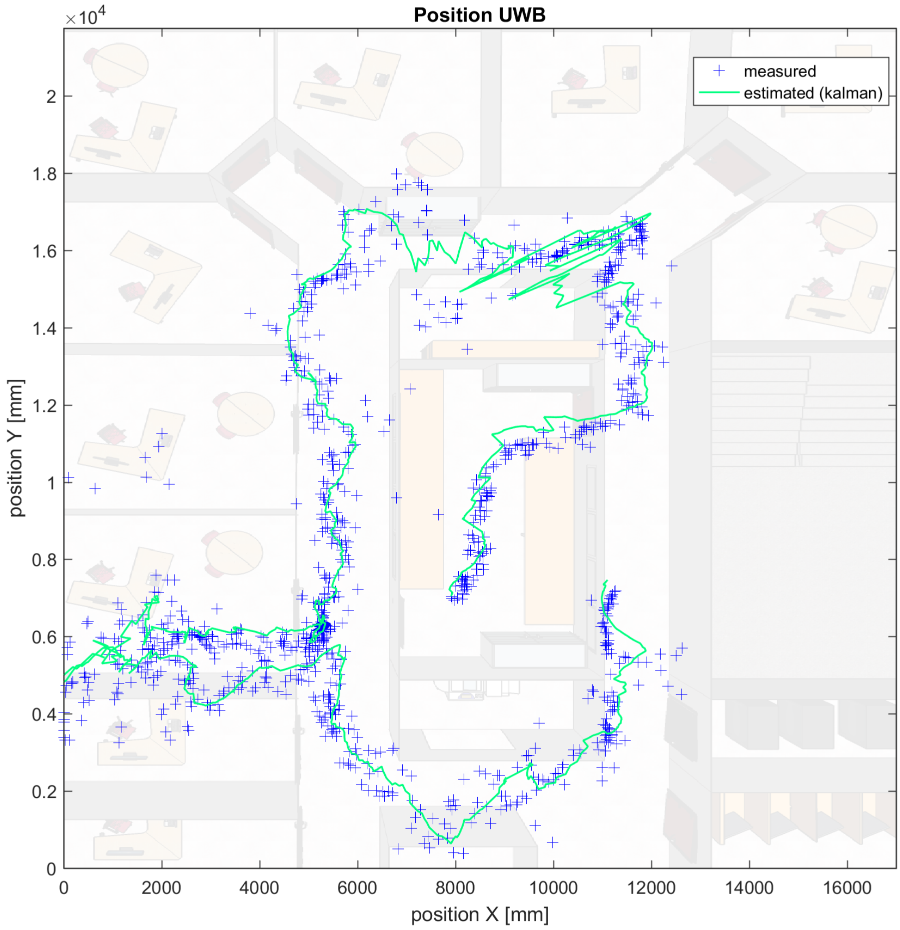

FIGURE 3.19: Test 2 results of UWB dynamic positioning in TE2

precise position.

3.3.10 Conclusions

The purpose of this part was to verify if UWB technology could be used to have a good dynamic estimation of position in an indoor environment. There have been many steps, which have firstly involved the search of better settings and the limits of the technology. Tests in medium-small environment have returned good results and have provided an acceptable estimation of the position. In order to cover a bigger environment (with obstacles like walls, wardrobes or windows) and to maintain an average error under the threshold of $50cm$, more than four anchors are required. For example, in TE2, which has an area of $370m^2$, a good choice could be six anchors. This would allow to eliminate shadow zones and reach better results. It is important to keep in mind that the number of anchors is related to the required degree of precision to be achieved and to the final uses of the technology. For example, if it is necessary to identify an user inside a building, four anchors for $370m^2$ are a good choice. But if it is required to accurately lead an user, then approximately at least six anchors are needed.

3.4 Comparison

To summarize what has been analysed in previous sections, the following table sums up the main differences of the two technologies tested to obtain best performances in Layer 2.

TABLE 3.2: Estimote beacon and Pozyx UWB technologies comparison

Features	**Estimote beacon**	**Pozyx UWB**
Number of devices	3 Location beacon	4 anchors and 1 tag
Maximum coverage area	~5m^2	~370m^2
Dependence on line-of-sight	High, due to the increasing of interferences with more obstacles	High dependency only for very accurate measurements (less than 10cm)
Power supply	Integrated battery (with five years of life)	Both anchors and tag must be powered by an external source (through microUSB or 2.1mm jack)
Physical installation of hardware devices	Very easy: just stick beacons wherever you want	Quite complex: screws, long cables and a way to powered all the devices are needed
Dependence on height position	High dependency related to noises. At the top of a room the error quite decrease	Best performances are obtained with height over than 1.60m (~human height)
Signal range	100m	70m

Noises tolerance	Very low tolerance (subjected to humans and materials interferences)	High tolerance, limited by distance and reinforced walls
Configuration adaptability	A lot of customizable parameters through web gui (protocol, broadcasting power, frequency,...)	Each device can be dynamically updated from software at runtime (channel, networkid, preamble length, ...)
Packet rate	From 100ms to 10s	From 250ms to 330ms
Availability of additional sensors	Motion, temperature, ambient light, magnetometer, pressure	Magnetometer, accelerometer, gyroscope, altimeter
Cost	90 euro	599 euro
Communication interface	Communicates over BLE channel	It can communicates on a serial port or through the connection pin of Arduino
Elaboration of position	Position must be elaborated after collecting RSSI values from all the beacons	Tag device already provides an elaborated position
Precision	error up to 1.30m in 2m^2 area	~50cm in 370m^2 area ~4cm in 20m^2 area

Chapter 4

Data Filtering

In the previous chapter the analysis was focused on searching for the best technology to adopt. Though data result to be accurate, the implementation of a software application for a real use still needs to have a further level of data transformation.

On the basis of the work of (Simon, 2001) and according to (Calabrese, 2015), this chapter will explains how the second layer of the model described in Section 1.2 was composed.

4.1 Filtering problem

Despite the advances in digital instrumentation design, a perfect accuracy has yet not been reached. Digital signals retrieved from sensors may be corrupted by a wide range of noises. The information presented by instruments can be used to a certain extent, but we cannot afford to grant it our total trust. To overcome this problem, it is necessary to apply some kind of filtering techniques: in optimal condition, a filter extract from the (noised) raw data the useful information.

For the purpose of this thesis, applyng a filter to position coordinates is essential to reach good performance. In previous chapters the analysis was focused on evaluation of positioning error, without taking into consideration the the impact of signal noise and how to exclude it. Since data are retrieved from UWB chip, signal may suffer by unwanted noises during capture, transmission, processing or conversion operation. In literature there's a huge amount of materials concerning this topic. In particular a lot of academics papers deal with the problem of object navigation in single or multi-dimensional environment and how to filter signals retrieved from instruments. As filtering is desirable in many situations in engineering and embedded systems, different mathematical models are available. For this work a good filter is the Kalman filter (*KF*).

4.2 Kalman filter

The *KF* uses a system's dynamics model (e.g., physical laws of motion), known control inputs to that system, and multiple sequential measurements (retrieved from sensors) to form an estimate of the system's varying quantities (its state) that is better than the estimate obtained by using only one measurement alone. As such, it is a common sensor fusion and data fusion algorithm.

KF was first developed to solve the specific problem of spacecraft navigation for the Apollo space program. Since then, the Kalman filter has found applications in hundreds of diverse areas, including all forms of navigation (aerospace, land and marine), nuclear power plant instrumentation and detection of underground radioactivity.

KF is one of the most famous Gaussian filters. Main property of this category is the representation of system's knowledge status through a *Gaussian probability distribution*, considering measurement sequence composed by a discrete signal and a random noise with expected value equal to null. This is possible with the assumption that the difference of average value of measured data and real data is very small. As a consequence, the signal retrieved from sensor is not intended as true data, but instead is categorized as an estimation with a certain likelihood degree.

With reference to the algorithm implementation discussed in the next chapters, it is important to point out some theoretical aspects of Gaussian filters:

- A Gaussian distribution can be represented in an easy way. Indeed it is composed by only two parameters: mean (μ), who represents the estimation with highest probability, and variance (σ^2) who indicates the uncertainty level.

- After each new observation, the distribution can be updated and the result will still be a Gaussian distribution.

- It is sufficient to keep in memory just the parameters which characterize the system, i.e. the distribution, instead of all the measurement history (potentially an infinite number of data).

In practice, variance with a high value implies a low sensor reliability, and vice-versa. As a consequence, next signal retrieved from sensor will be weighted during the update process of mean value.

In mathematical terms, it can be shown that among all possible filters, *KF* is the one that minimizes the variance of the estimation error. It will not go deeper in algebraic demonstrations; for this work it will begin to describe *KF* only starting from *control theory*.

4.2.1 Dynamic systems

Many physical processes, such as objects 2D tracking, can be approximated as dynamical systems.

A dynamical system is a mathematical model of a physical object which interacts with real world through two time-dependent vectors of variables.

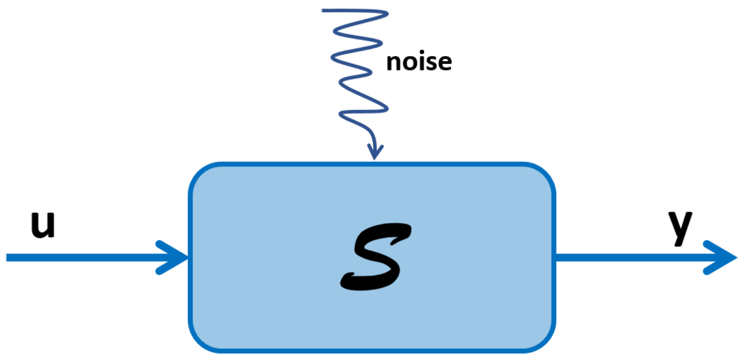

FIGURE 4.1: Linear dynamical system

In particular, to build up the *KF* model, the system must be:

- **Linearised**: system whose evaluation functions are linear.
- **Multiple Input Multiple Output**: system has more than one input and output.
- **Discretized in the time domain**, according to digital instrumentations needs.

Such linear system is represented by:

$$\begin{cases} \vec{x}(k+1) = A\vec{x}(k) + B\vec{u}(k) + \vec{w}(k) \\ \vec{y}(k) = C\vec{x}(k) + D\vec{u}(k) + \vec{z}(k) \end{cases} \tag{4.1}$$

where k is the time index; A,B,C and D are matrices.

The different composing elements (4.1) are:

- $\vec{\mathbf{x}}$ is the *state vector*. It contains all the information about present state of the system.
- $\vec{\mathbf{y}}$ is the *output vector*. It contains all the output data of the system.
- $\vec{\mathbf{u}}$ is the *input vector*. It contains all the input data (coming from external world).
- **A** is the *state matrix*, whose product with the state vector at an initial time t_0 gives $\vec{x}$ at a later time t.
- **B** is the *control matrix*. It determines how the system input affects the state change.
- **C** is the *output matrix*. It determines the relationship between the system state and the system output.
- **D** is the *feed-forward matrix*. It allows for the system input to affect the system output directly.
- $\vec{\mathbf{w}}$ and $\vec{\mathbf{z}}$ are the *process noise* and the *measurement noise* respectively.

To better understand how to initialize the components described above, it's important to start from the physics of objects tracking.

4.2.2 Physical model

Starting from elementary physics laws of motion, we can generically describe the movement of an object on a straight line as (at continuous time):

$$\begin{cases} \vec{p}(t) = \vec{p}(0) + \vec{v}(0)t + \frac{1}{2}\vec{a}t^2 \\ \vec{v}(t) = \vec{v}(0) + \vec{a}(0)t \end{cases} \tag{4.2}$$

where $\vec{p}$ represents position, $\vec{v}$ represents the velocity, $\vec{a}$ the acceleration and t is the time index.

Dealing with digital instruments like UWB chip and accelerometer, we can retrieve sensors data only every T seconds (sampling time), so the (4.2) has to be discretized as follow:

$$\begin{cases} \vec{p}(k+1) = \vec{p}(k) + \vec{v}(k)T + \dfrac{1}{2}\vec{a}(k)T^2 \\ \vec{v}(k+1) = \vec{v}(k) + \vec{a}(k)T \end{cases} \tag{4.3}$$

That is, the position $\vec{p}$ and the velocity $\vec{v}$ one time-step from now (T seconds from now). The first equation of (4.3) is the sum of the current position plus the contribute of velocity and acceleration, the second one is composed by the present velocity plus the acceleration multiplied by T. But the (4.3) does not gives a precise value for $\vec{p}_{k+1}$ and $\vec{v}_{k+1}$. Indeed, in the real world the measurement most likely will be perturbed by a wide range of noises. So a more realistic equations model must also include two others components: $\widetilde{p_k}$ is the module of position noise, $\widetilde{v_k}$ is the module of velocity noise.

To sum up, a more precise model for the problem becomes:

$$\begin{cases} p_x(k+1) = p_x(k) + v_x(k)T + \dfrac{1}{2}a_x(k)T^2 + \widetilde{p_x}(k) \\ v_x(k+1) = v_x(k) + a_x(k)T + \widetilde{v_x}(k) \\ p_y(k+1) = p_y(k) + v_y(k)T + \dfrac{1}{2}a_y(k)T^2 + \widetilde{p_y}(k) \\ v_y(k+1) = v_y(k) + a_y(k)T + \widetilde{v_y}(k) \end{cases} \tag{4.4}$$

Notice that in (4.4) the vectors are decomposed into their portions along x-axis and y-axis, according to 2-dimensional cartesian space requirements.

4.2.3 Components initialization

Starting from (4.4), it is quite easy to highlight the different equations coefficients which compose (4.1) matrices:

- $\vec{x}$ is a 4x1 vector containing position and velocity data of both axes

$$\vec{x} = \begin{bmatrix} p_x \\ v_x \\ p_y \\ v_y \end{bmatrix}$$

- $\vec{y}$ is a 2x1 vector containing the output of the system (position data of both axes)

$$\vec{y} = \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

- $\vec{u}$ is a 2x1 vector containing the input acceleration of both axes

$$\vec{u} = \begin{bmatrix} a_x \\ a_y \end{bmatrix}$$

- **A** is a 4x4 matrix containing position and velocity coefficients of (4.4) equations

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- **B** is a 4x2 matrix containing acceleration coefficient of (4.4) equations

$$B = \begin{bmatrix} \frac{T^2}{2} & 0 \\ T & 0 \\ 0 & \frac{T^2}{2} \\ 0 & T \end{bmatrix}$$

- **C** is a 2x4 matrix which transfer to the output only the required data (it cut out the velocity because it's useless)

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

- **D** is set to 0 because the acceleration cannot affect the system output directly

$$D = 0$$

- $\vec{\mathbf{w}}$ and $\vec{\mathbf{z}}$ must be initialized after some simulation on real implementation environment.

Remember that as per Gaussian distribution, it was assumed that the average value of $\vec{\mathbf{w}}$ and $\vec{\mathbf{z}}$ is zero. Furthermore, no correlation must exists between these two vectors. Then the noise covariance matrices is defined as:

$$S_z = E(\vec{z}_k \vec{z}_k^T) = \begin{bmatrix} |\vec{z}_k|^2 & 0 \\ 0 & |\vec{z}_k|^2 \end{bmatrix} \tag{4.5}$$

called *measurement noise covariance matrix*;

$$S_w = E(\vec{w}_k \vec{w}_k^T) = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} & 0 & 0 \\ \frac{T^3}{2} & T^4 & 0 & 0 \\ 0 & 0 & \frac{T^4}{4} & \frac{T^3}{2} \\ 0 & 0 & \frac{T^3}{2} & T^4 \end{bmatrix} \tag{4.6}$$

called *process noise covariance matrix*. Notice that *E(·)* means the expected value.

4.2.4 Kalman equations

Suppose we have a linear system model as described previously. We want to use the available measurements $\vec{y}$ to estimate the state of the system $\vec{x}$. Knowing how the system behaves according to the state equation, and having measurements of the position (coming from sensors), how can the best estimate of the state *x* be determined? An estimator that gives an accurate estimate of the true state - even though it cannot directly be measured – is required.
There are two obvious requirements that this estimator should satisfy. First, the average value of the state estimate should be equal to the average value of the true state (mathematically, the expected value of the estimate should be equal to the expected value of the state). That is, the estimation may not be biased. Second, the estimated state has to vary from the true state as little as possible. That is, not only the average of the estimated state should be equal to the average of the true state, but also an estimator that results in the smallest possible variation of the state estimate is required (mathematically, the estimator with the smallest possible error variance). *KF*, with the assumptions and initializations of the previous sections, is the estimator that satisfies these two criteria.

Based on the model built up so far, here below are reported the Kalman equations:

$$\begin{aligned} K_k &= AP_kC_T(CP_kC^T + S_z)^{-1} \\ \hat{x}_{k+1} &= (A\hat{x}_k + Bu_k) + K_k(y_{k+1} - C\hat{x}_k) \\ P_{k+1} &= AP_kA^T + S_w - AP_kC^TS_z^{-1}CP_kA^T \end{aligned} \tag{4.7}$$

with $P_0 = S_w$ and $\hat{x}_0 = 0$.
The *K* matrix is called the *Kalman gain*, and the *P* matrix is called the *estimation error covariance*. The state estimate ($\hat{x}$) equation is fairly intuitive. In

the first term is recognizable the equation of (4.1). This would be the estimated state if there was no measurement. The second term in the $\hat{x}$ equation is called the *correction term* and it represents the amount by which to correct the propagated state estimate due to the measurement.
As the composition of *KF* equations is out of the scope of this thesis, no more discussion will be performed on this topic.

To perform some tests, an implementation of the model was made using Matlab; the code is reported in Appendix A.

4.3 Prototype

To evaluate how the positioning technology could be used in a real world implementation, this part will describe a prototype that has been built on top of the UWB technology. Such prototype aims to elaborate an accurate position of people and supply it to a software application, possibly running on a mobile platform. Unfortunately, there are no mobile devices (smartphone or tablet) equipped natively with an UWB antenna, nor there is availability of external accessories with this technology, so it is not possible to simply track it directly. A feasible solution might be to use a multi-component system: a *user device* where the application is shown and a *tag device* (containing UWB antenna) which will be localized. The communication between the two components can be achieved through the WiFi or Bluetooth channel. The whole system is thought to be a Use Case test; in the next sections – and for the sake of argument – we will suppose that the *user* (a person on the move) is keeping the tag device in his hands and has a real-time position feedback on his user device, that is, a smartphone.

4.3.1 System description

The logic behaviour of the system is represented in Figure 4.2. Inside the two types of devices are represented blocks which refer to the different logical functions that are present in the execution flow (which will start from the top of the image). The dashed lines represent an optional block.

Tag device

Tag device is the element which is tracked while it's moving inside the virtual designated Cartesian space. It is designed to be kept attached to the user.

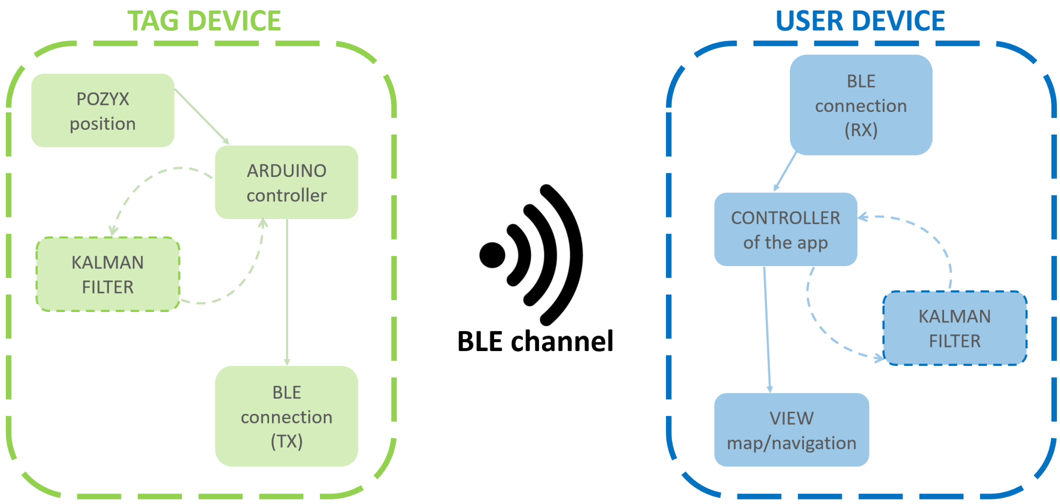

FIGURE 4.2: Logic description of the prototype system

As represented in Figure 4.3, the prototype is assembled using the following components:

a) **Arduino UNO rev3 board**[1]: an open source microcontroller for building digital devices. It will be used as central unit when executing custom code.

b) **RedBear BLE shield**[2]: a Bluetooth communication shield which integrates *Nordic nRF8001 Bluetooth Low Energy IC*[3]. It is designed to work with Arduino boards and allows to establish a BLE connection to the user's device.

c) **Pozyx tag**[4]: this is the board already described and used in the previous chapters to perform positioning.

Clearly, b) and c) contain the technology for transmission and positioning respectively, instead on a) is executed the code that act as controller.

During the test phase, because of the little amount of SRAM available on Arduino (2 KB), it appears necessary to optimize the code running on it. As a consequence, all the static variables (such as string, general settings, constants) are moved from SRAM to internal flash memory using the *PROGMEM*[5] and *F()*[6] macros. Despite these adjustments, the amount of free SRAM

[1] developed by *Arduino*, `https://www.arduino.cc`
[2] developed by *RedBearLab*, `http://redbearlab.com/bleshield`
[3] developed by *Nordic Semiconductor*, `https://www.nordicsemi.com/eng/Products/Bluetooth-low-energy/nRF8001`
[4] developed by *Pozyx*, `https://www.pozyx.io`
[5] `https://www.arduino.cc/en/Reference/PROGMEM`
[6] `http://playground.arduino.cc/Learning/Memory`

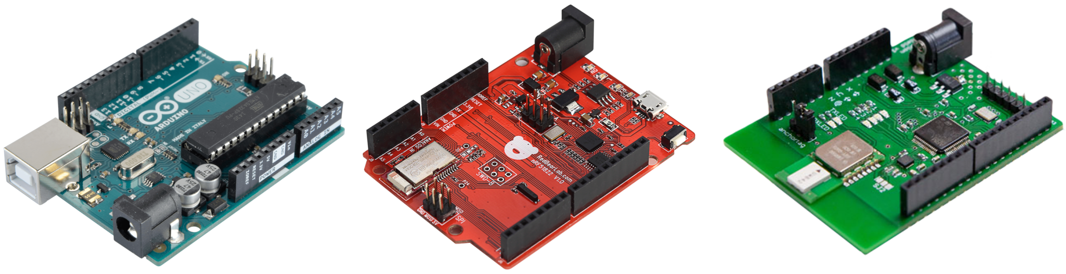

FIGURE 4.3: Components which compose Tag device. From left to right: Arduino UNO, RedBear BLE shield, Pozyx tag

at runtime is not yet enough to reach the final goal. This is due to the fact that about 75% of memory is used by BLE shield and Pozyx tag libraries; in fact the two environments must be initialized during the boot up phase and cannot be deallocated without losing the pair with user device. Unfortunately, with the remaining 25% of memory (approximately 500 bytes) there's no way to perform Kalman Filter operations, because its equations rely heavily on matrix algebra. Therefore, this prototype will be developed considering two working modality:

1) **R-mode**: values retrieved from Pozyx are sent through bluetooth as raw data.

2) **K-mode**: values retrieved from Pozyx are elaborated at runtime with Kalman Filter and the resulted estimate values are sent through bluetooth. This function mode is not supported by Arduino UNO, but can be easily executed on board with an higher amount of SRAM (for example the Arduino MEGA, which has 8 KB of available memory).

Based on the working mode, a single packet sent over BLE channel can be (Listing 4.1):

LISTING 4.1: Prototype: possible packets

```
//case of R-mode
packet = "R" + timestamp + "," + measured_coordX + "," +
    measured_coordY + "," + accelerationX + "," +
    accelerationY + "," + orientation
//case of K-mode
packet = "K" + timestamp + "," + estimated_coordX + "," +
    estimated_coordY + "," + orientation
```

where *orientation* is a variable containing, with reference to Cartesian space, the tag device angle rotation starting from the origin. Notice that, in the case of *R-packet*, also raw data position and acceleration are sent; this is done to allow user device to perform Kalman filtering independently. Also notice that in the current prototype implementation, the choice of one rather than the other modality must be done at compile time, but there would be nothing wrong in making these options interchangeable at runtime.

To quickly describe how the code works, a generic pseudo-code implementation is reported in Listing 4.2. The complete Arduino code is instead available in the thesis resources.

LISTING 4.2: Prototype: Tag device pseudocode

```
float data[];

initial_resources_setup(&data);

while(arduino_is_powered_on){

  retrieveDataFromPozyx(&data);
  if(WORKING_MODALITY == 'K') applyKalmanFilter(&data);
  if(user_device_is_connected) sendDataOverBLEchannel(data);
}
```

As far as the packets rate, this is surely influenced by the working mode. In fact, using the K-mode implies a greater elaboration time: approximately a supposedly rate of *3 packets/second*, which raise to *4 packet/second* if only raw data are sent.

User device

With reference to Figure 4.2, user device is intended as whatever software platform (for desktop, mobile or server) which is able to receive data from tag device through BLE channel. The following analysis are focused on mobile application because, reasonably, smartphone (or tablet) would be one of the main used devices. As consequence of the two possible work modality described previously, the application controller must support the decoding of packets described in Listing 4.1. It is therefore here also implemented the Kalman filter, to perform filtering when tag device sends only raw data. After receiving the BLE transmission, a few small steps must be taken in order to obtain the correct values; in particular the received bytes must be cast to *type float*, with particular attention to the necessity of measurement unit conversion, for example $milligal$ to $millimeters/second^2$ for acceleration.

When controller finish to elaborate the packet, it proceeds with the update of *application view*, where a map of the environment is displayed (see Figure 4.4) and position is marked down by an arrow. To bridge the gap between

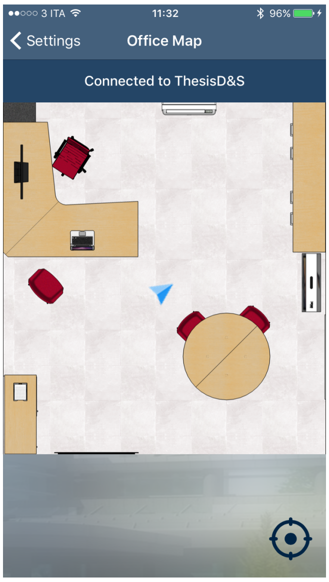

FIGURE 4.4: Prototype application view

positioning given by digital system (discrete function) and people movement simulation (continuous function) the application of a graphical interpolation is needed, i.e. a computer graphics animation which fills in frames between the key frames. This, together with a high value of packet rate, makes the motion more fluid and real-time representation more realistic. Unfortunately, this choice affects the speed of position update: the more fluidity is achieved, the less position update quickness is obtained.

Finally, smartphone performance is to consider. In particular, move Kalman filter elaboration from tag device to the user device adds an expensive continuous task to the mobile platform, with the natural consequences of intensive elaboration, such as battery consumption and heat production. For this reason, using K-mode would be more energy efficient.

4.3.2 Conclusions

The developed prototype has been tested and used for the experiments needed by the navigation system (see next chapter). It has permitted to receive, elaborate and transmit useful and accurate data from antennas to the user device. The performances have been good, with a stable BLE connection which has allowed the user to make the pairing phase quick and transparent. In fact, the connection between tag and user device is done automatically in background. This prototype has been conceived to let the user move freely inside the environment, in order to obtain a wide range of possible tests. The before mentioned experiments have shown the need for an external source of power to be a limit (in this work it has been used a Power Bank) and that the antenna needs to be turned upwards. Furthermore, the multiple tests have highlighted, from a user's perspective, that an external device is uncomfortable; thus, for a real use of the prototype, a more compact solution would be preferred.

Chapter 5

Contextual adaptation

In the previous chapters, the goal was limited to achieve a system able to identify the position of a person in an indoor environment. Now, the accurate results obtained in *Layer 1* and *Layer 2* have permitted to search for a solution where Pozyx tag and anchors are used in order to put together a navigation system, which could be useful in different contexts. For example, blind people could set-up their home in order to move easily inside it, or public buildings could be mapped to help external visitors. Despite the precision of positioning, to obtain an high level user experience it is imperative to contextualize the positioning information retrieved in order to provide better indications to the final user during the navigation. For this reason, the following sections present the description of the *contextual adaptation* with reference to a navigation system, which compose the *Layer 3*.

5.1 A third filtering layer

Kalman filter has permitted to extract clean position data from the UWB noised signals (see Chapter 4). On average, in a big environment (where "big" is intended as an area of more than $300m^2$) the measured error may vary between $30cm$ and $50cm$. It can be considered as a good result, but in real-life uses this could lead to some misleading information: in fact, for a person walking in a room near a partition wall, this kind of error is not small enough to let understand if the user is precisely located inside or outside the room. This is an important issue for a navigation system, as it may cause a wrong room association for actual position of the user and therefore further problems like useless path recalculation and instability. Furthermore, as people can't move through the walls, if they are following a path in a specific environment of the building, it could not be possible for them to go outside passing through obstacles. Thus, it becomes very important to stabilize the estimated location, avoiding flickering of the displayed position. Based on these problems, the necessity is evident to develop a third filtering layer, which has the task to further adapt the data obtained by *KF* to the reference context. This new software layer, which will be implemented inside the navigation application, can be divided into two phases.

First phase

A first step tries to cut off outliers. Although data retrieved from Pozyx are quite precise (in order of centimeters), sometimes there are some measurements that are clearly far away from the real position. In such cases, the average error is more than one meter, so it is easy to identify a cluster of correct points and, consequently, to exclude the outliers. This process aims to eliminate misleading points to improve stability: a buffer memorize the last five positions of the user and the variance is calculated on these values. If the variance is greater than a fixed threshold x, then the information of the last packet is marked as outlier and later cut off. Concerning the choice of the optimal value for x, it has to be considered that each packet is sent over BLE channel at a frequency of three/four packets per second. This means that a buffer of five elements is able to memorize about two seconds of position history. In this time frame, a person that walks usually cover a distance of two/three meters. This distance could be a good value of x, but also the average error between real and estimated position (about $50cm$) has to be considered, thus x should be greater or equal to $3.5m$. After some tests, according to *TE1* and *TE2*, it has been identified a good choice in $x = 4m$.

Second phase

The second step of Layer 2 focuses on a particular aspect: starting from an example, it can be assumed that if an user use a navigator while he walks in an hallway, he is not interested in knowing at what point of width it is located (left side, right side, centre, ...), instead he just want to know if he has taken the correct hallway and if he is going in the right direction. As consequence, it can be reasonable to further abstracting the coordinates of the user position. As will be described in the following sections, when a person launch the navigator from a starting point to a destination, his position and the optimal path that he should follow is known. The idea is to search a point (belonging to the optimal path) that is the closest to user position (the estimated location coordinates retrieved by *KF*) and to mark this point as the current user position. This approach allows to solve another problem mentioned at the beginning of the chapter: an average position error of $e = 50cm$ imply that a person could be in a ray of e around the estimated position provided by *KF*; so, if the user walks near a wall that divides two rooms, there is uncertainty about the room of environment where he is located. To overcome this problem, knowing the initial position and the optimal path, the route that user should follows in terms of sequence of adjacent nodes become clear. Therefore, by attaching the position to the closest node, a good approximation of the exact room can be obtained (even if still not error free). Obviously, the closest point of the path must be at a reasonable maximum distance from the user: a good delta is $2m$. If the difference is greater than this value, then the optimal path must be recalculated because the user may be considered as far away from the original optimal route. This approach also allows to further stabilize the position drawn on the map avoiding flickering.

5.2 Navigation

To accomplish the features mentioned in the previous section, an efficient data structure is required. It has to keep information about each Point Of Interest of the environment (in particular its coordinates on the map) and all the links between them. As literature suggests, the best choice is to use a *graph*, that is a structure composed of vertices (which identify the Point Of Interest) and edges (which represent the connections between pairs of vertices). Furthermore, the graph has to be *connected*: this involves that, once a vertex is chosen, it is possible to reach each other node of the graph following one or more of its edges. It is an essential condition for a navigation system, because users must have the opportunity to move freely in every area of the environment starting from any point. The used model for graph representation is a list of successors: each vertex of the graph memorizes an array of edges which connects itself to another vertex. In this way, it is possible to discover the neighbours of a vertex just following the edges contained in the array.

5.2.1 Optimal path research

The key point of a navigation application is the search for an optimal path, given a starting and an ending point. A lot of studies have been done in the

field of *Operations Research* and in particular about the use of graphs to map cities or whatever. These studies have lead to the formulation of different algorithms based on *network flows*. Among them, and according to (R.K. Ahuja, 1993), one of the most efficient is *Dijkstra's algorithm*: it uses *directed* graphs (directed means that an edge can be travelled only in one direction) and a *cost* assigned at each edge, which could represents for example the travelling time or the length between two vertices.
The algorithm can be defined as follow:

> *Given a directed graph $G = (N, E)$ with a cost $c_{ij} > 0$ for each $edge(i, j)$ and two nodes s and t, it permits to determine a minimum cost (shortest) path from s to t*

To each node $j \in N$ is assigned a label $L[j]$ (which corresponds, at the end of the algorithm, to the cost of a minimum cost path from *s* to *j*) and a label $pred[j]$ (which is the predecessor vertex of *j* in the shortest path from s to *j*). Furthermore, a subset $S \in N$ of vertices is needed; initially it will contain only the starting point vertex. After the setting of these elements, the optimal path can be constructed iteratively. At each step all the edges with starting vertex $\in S$ and with ending vertex $\notin S$ are considered. The vertex which has an edge with the minimum cost among those considered is inserted in set S and so on iteratively until the ending node will be reached. At the end of the process, a set of paths is retrieved. Among all the returned paths, only the one which has the destination node as final vertex is to be considered. With reference to the selected path, to obtain the optimal route from *s* to *t*, simply follows the chain of predecessors vertices.

As mentioned before, Dijkstra's algorithm is very efficient. At each step, all edges are scanned and those without a starting point in *S* and an ending point in $N \setminus S$ are discarded. So, the overall complexity would be $\mathcal{O}(nm)$ (in *big $\mathcal{O}$* notation, with *n* number of vertices, *m* number of edges). Besides, for a dense graph (a graph in which the number of edges is close to the maximal number of edges), it can be said that:

$$m = n(n-1) \tag{5.1}$$

So, after a substitution, the complexity can be expressed as $\mathcal{O}(n^3)$. Dijkstra's algorithm is also exact, so it provides an optimal solution for every instance. In Listing 5.1 its pseudocode is reported.

LISTING 5.1: Dijkstra's algorithm pseudocode

```
G = (N,E);
```

```
n = |N|;
m = |E|;
s in N;
c_{ij} >= 0 foreach (i,j) in A;

BEGIN
  S = {s};
  L[s] = 0;
  pred[s] = s;
  WHILE |S| < n DO
    SELECT (v,h) in S+ (outgoing edges from S) = { (i,j) :
       (i,j) in E , i in S , j notin S } such that L[v] +
       c_{vh} = min { L[i] + c_{ij} : (i,j) in S+ };
    L[h] = L[v] + c_{vh};
    pred[h] = v;
    S = S + h;
  END-WHILE
END
```

5.3 Navigator implementation

Regarding the practical implementation of a navigator with features as described in the previous section, the following lines will focus on the main aspects to consider for a (non exhaustive) real usage.

5.3.1 Definition of edge cost

Ideally, the distance between each pair of vertices could be used as edge cost. In fact, it is reasonable to assume that each user wants to follow the shortest route to reach its destination. In addition to this feature, to meet the purpose expressed in the introduction of this work, it has been chosen to design a bit more sophisticated algorithm in order to help, for example, blind and disabled people to move inside big environments easily. According to this, the necessity to consider (during the path calculation) all the obstacles arised; for this reason, the cost of an edge between a vertex i and a vertex j has been composed as in Equation (5.2):

$$c_{ij} = d_{ij} + k_{ij} \tag{5.2}$$

where d_{ij} indicates the distance between i and j, while k_{ij} is a value which indicates the presence of any kind of obstacle on the specific part of the path (this value could be eventually intended as an indicator of "obstacle importance", with $k_{ij} = 0$ if there aren't obstacles). Applying *Dijkstra's algorithm*, due to the presence of the variable k_{ij}, the paths with obstacles will be penalized. With this configuration, Equation (5.2) might also be dynamically initialized, thus providing a personalization of the map according to different user profiles; as a consequence, the best path could be calculated according to the user characteristics, such as being blind or being a person with wheelchair.

5.3.2 Service vertices

As explained before, the problem of position oscillations can be overcome with the approximation at the nearest point along the optimal path. In practice, to obtain a quite good result, some adjustments on the graph model must be applied: first of all, the set N of vertices should not be merely composed by *Point Of Interest* nodes; each edge which connects two vertices has in fact been divided into more edges by inserting multiple intermediate nodes, called Service nodes, respecting a distance of about 1m each other. So, if two nodes, A and B, are connected and at a distance of $10m$, 9 vertices are entered between them and multiple edges will link these nodes. Figure 5.1 represents both Point of Interest node (in red) and Service node (in orange). This graph

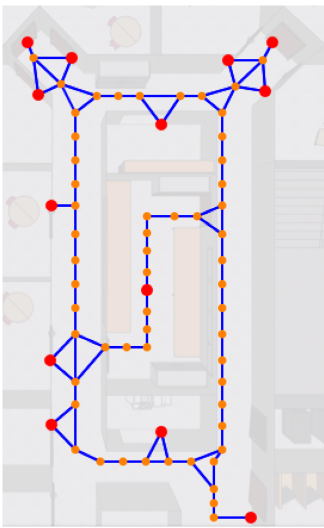

FIGURE 5.1: Graphical representation of graph model representing TE2 map

structure upgrade yields two advantages:

First, some mathematical operations used to couple estimated user position with optimal path point become useless and can be removed (with a consequent decreasing on algorithm's computational complexity). Indeed, it is sufficient only to verify the distance between user location and vertices of the optimal path (denoted as $P \in N$) to select the closest node. In the thesis prototype, a closest node was chosen only if it was within a ray of $2m$; the experiments have demonstrated that the distance of $1m$ between each vertex has brought to obtain a better estimation of the user position.
The second advantage is the possibility to have a greater granularity, with more options to customize map description, in particular while highlighting obstacles on the path. In fact, each Service node can store alerts about an obstacle which is near its position and it is possible to retrieve data concerning this with a precision of $1m$. In the original data structure with only Point Of Interest vertices, if two of them were at a distance of $20m$, the precision was too low and it was only possible to report an alert for a macro-zone, not for a specific point.

5.3.3 Navigation and voice commands

Another important aspect is how to provide navigation information to the user. When an optimal route is calculated, a subset of vertices which compose the path is retrieved. Each vertex contains the information about which edge to follow to reach the next vertex. To build up the navigation system, each edge provides the direction (left, right, straight) to reach all the nearest nodes. So, when the user is located on a node of the optimal route, its next edge to follow is retrieved and the correct direction is elaborated.
Navigation informations are provided in two different (but not concurrent) ways:

- **Through textual commands:** at the bottom of the screen, textual information are provided, supported by road signals contextualized according to the direction to follow. Some examples can be seen in Figure 5.2.
- **Through voice commands:** to improve accessibility, as helping blind people to move easily in a building, voice commands are also supported. At each change of direction, a voice provides the new direction to follow.

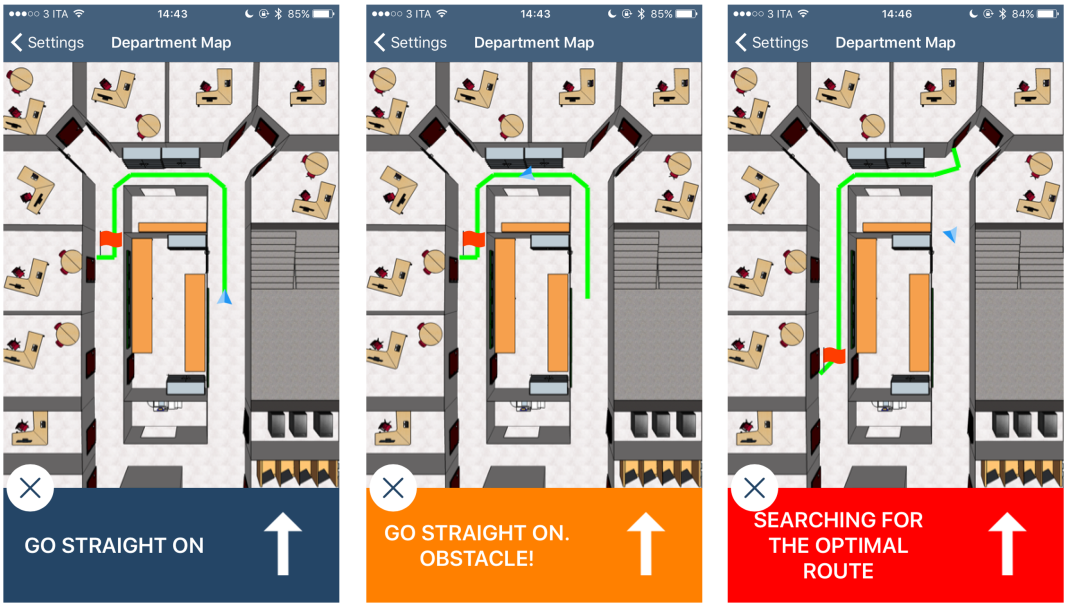

FIGURE 5.2: Three different examples of informations displayed during navigation

When the user is near an obstacle, the navigation interface change its colour (it becomes orange) and both the voice and textual commands provide information about the obstruction in the path. A warning for a wrong path (and subsequently path recalculation) is also implemented (interface becomes red).

In the next chapter will be presented a mobile application (*Medical Center*) which integrates this functionality, in order to provide a real example of implementation and use case.

Chapter 6

Medical Center Mobile Application

In order to test the functionality of the system, a demo mobile application has been developed. The app was created having in mind a medical center whose patients could use throughout their hospitalization, navigating inside the hospital buildings by an integrated navigator.
The organization of a hospital and the needs of the patients have been taken into account to understand how to optimize the workflow of doctors and nurses and to improve the quality of patients' life. The idea is to present a mobile application that helps patients to handle the hospitalization phase. In particular, four areas of interest have been taken into account.
Nowadays an hospitalized patient isn't able to control his situation on his own. Instead, he has to ask doctors or nurses in order to have information about his next examination or prescription. This service application will indeed help patients to be able to have control over their own hospitalization.
The purpose of *Medical Center* is to have an overview on the next examinations and prescriptions of each patient. Furthermore, the user can manage lunch and dinner menus or browse within the Points Of Interest of the structure, in order to easily found the right place and move to it through the integrated navigator. Thanks to the map, the patient can also know where exactly he is located while the news section provide useful information. With these features, *Medical Center* aims to be a solution to the digital problem in a public structure such a hospital.
To enable localization functionalities, each patient is assigned a Tag Device (the component described in previously chapters).

In order to provide a clear description of the system which has been developed, in the following sections functional and nonfunctional requirements will be analysed, beside the goals to reach, the real world (in order to model

the constraints) and the software architectures.
In addition, further specific technical documentation is available in Appendix B and in the thesis resources.

6.1 Overview of functionalities

The scope of the project is to develop a system able to manage and optimize the hospitalization phase of patients. In particular, a patient can use a mobile application to have different kinds of information. *Medical Center* is divided in the following six sections:

- **Next Examination Section:** it provides a list of examinations divided into two further sections: *Today*, which provides all the examinations scheduled for the current day, and *Next Days*, which provides all the future examinations. If the user enables notifications, he will receive a reminder two hours before the beginning of the exam.

- **Prescriptions Section:** it provides all the drugs that the specific patient must take. Date and time are specified for each prescription. If the user enables notifications, he will receive a reminder when he has to take the medicine.

- **Cafeteria Section:** it is a dynamic view which shows lunch and dinner menus, that are different for each day. Patients can select first and second dish and confirm their choice, sending information directly to the cafeteria.

- **Places Section:** it provides the list of all the Points Of Interest of the hospital. Each of them is described by a list of information such as description, place, phone number, with the possibility to start a call or to start the navigation. The latter functionality leads the patient from a starting point to the destination, through textual and vocal commands, giving the best route to reach the final point and information about obstacles along the path. User position is filtered using Kalman Filter in order to improve the precision of retrieved position.

- **Map Section:** it shows the map of the hospital and the position of the user inside it.

- **News Section:** : it provides the latest news of the hospital, directly available in mobile version.

To use the system, the patient must be registered and logged. The registration phase is not managed by the final user, but username and password are provided by the reception of the structure.

6.2 Functional requirements

The following are constraints that the mobile application must satisfy:

- **Registration mandatory:** the registration is mandatory to use the service.
- **Unique username:** two different users cannot have the same username.
- **Hospitalized patient:** the application can be used only by hospitalized patients.
- **Parallelism:** the service must manage multiple requests from different customers at the same time.
- **External device for localization:** to be able to use localization and navigation functionalities, an external device which sends information about the user position to the mobile application is required. It is nowadays a required condition, since mobile devices as smartphone or tablet don't include an UWB antenna, the technology used to estimate the position of an user.
- **Credentials at each access:** each time a user wants to use the service, he must insert username and password. If the device is equipped with touch ID, the patient can access to the app using it.

6.3 Non-Functional requirements

All functional requirements of the application – i.e. functionalities that the system provides to the final user – have been listed in the previous sections. But non-functional requirements are also important, since they permit to guarantee full functionalities for the system:

Design Constraints

For the development of the app, Swift language has been chosen. That means that, in the first release, it will be compatible with iOS devices only (iPhone, iPad, iPod Touch). To guarantee that *Medical Center* works well on different devices, *auto-layout* is used, so the user interface can adapt itself automatically according to the device in use. This guarantees a good user experience. The architectural design pattern chosen is Model-View-Controller (MVC).

Performance

Information about examinations, prescriptions and cafeteria must be retrieved from the hospital server. The system must answer each request in the shortest possible time (a couple of seconds). So, to guarantee this requirement, data are exchanged using JSON files. JSON is a format to quickly transmit data objects through the web.

Security

All the information provided by the application are confidential. The patient can access to his own data only, so he must log in to the system at each access either through username and password or using Touch ID (if available on the device).

6.4 Architecture

This section lists the hardware components used in *Medical Center* and the way they interact with each other. Some diagrams show static and dynamic behaviour of the components. At the end of this section the architectural design pattern used is shown, along with a schema to better understand how it works.

6.4.1 Architectural structure

With *Medical Center* system the following hardware components needed to meet the identified goal have been defined:

- **Database Server:** this component stores all the data used by the system. Here are stored all the information about each patient (examinations, prescriptions, selected menus), which could be accessible through an HTTPS call that returns a JSON file with all the requested information.

- **Client (mobile application):** this component retrieves JSON files from the server, display them in the application view and sends data to the server with all the information about the chosen menu of the patient. It also manages all the notifications that must be sent to the user device when there is a new examination or prescription.

- **Tag Device:** this component sends packets to the user about its position in the hospital through a bluetooth connection.

- **News API:** this component is an API that returns JSON metadata for the headlines currently published on a range of news sources and blogs. This API is used to retrieve the information about the medical news to show in News section.

6.4.2 Architectural patterns

MVC is the chosen architectural pattern. It provides a separation between the user interface of the application (the view, which is shown on the client), the controller (the logic of the software, also present on the client) and all the data (model), that can be retrieved by different sources. In particular, all the information of each patient can be retrieved from a database through an HTTPS call, the news are retrieved through an HTTPS to *newsAPI* service while the position can be retrieved through bluetooth from an external device (Tag device), that is an object assigned to the patient when hospitalized.

6.5 External services and libraries

Some external services and libraries have been used to implement the functionalities of *Medical Center*: Some of them are needed to perform logical operations, others instead are used to retrieve information from external sources.

Library: SwiftyJSON

All the information retrieved from the database or external services are formatted in a JSON format. So, there is the necessity to parse this kind of data structure. SwiftyJSON is a simplified JSON parsing library that gives a clearer syntax respect to the built-in iOS libraries. It doesn't handle HTTPS request to retrieve JSON (this operation is performed using internal Swift library) but it manages only the parsing of the files.

Library: Upsurge

Upsurge is a math utilities library. It provides support for linear operations on vectors and matrices, and slicing of higher-dimensional tensors. It relies on Accelerate, which is a framework that provides high-performance functions for matrix math, digital signal processing, and image manipulation. This library is useful to perform some mathematical operations to apply Kalman Filter on user position retrieved from tag device.

Service: News API

Medical Center provides news about the hospital. Actually, it is not possible to have real news, so an external system has been used to provide this kind of information. Medical news are retrieved in JSON format from a news website and then are parsed and formatted on the view of the application. *News API* is a service which provides API and returns JSON metadata for the headlines and the links currently published on a range of news sources and blogs. In particular, new-scientist.com has been chosen as source.

6.6 User interface

Nowadays, the user interface is essential for a mobile application. There are many apps on the market, so what characterises an application respect to another is the user-friendliness. For this reason, a lot of effort has been made on the usability and the graphics of the application. No external graphics library have been used to develop the views. The intention was to avoid confusion for the end-user, using the same graphic elements as the operating system. So, iOS Human Interface Guidelines[1] have been used to design and implement an iOS application that works both on iPhone and iPad (Portrait and Landscape orientations are supported). In this section will be shown and explained the relevant views of *Medical Center* app. To navigate between each view it has been implemented a tab bar controller, which shows on the bottom all the views implemented in the application.

[1] https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/

6.6.1 Login View

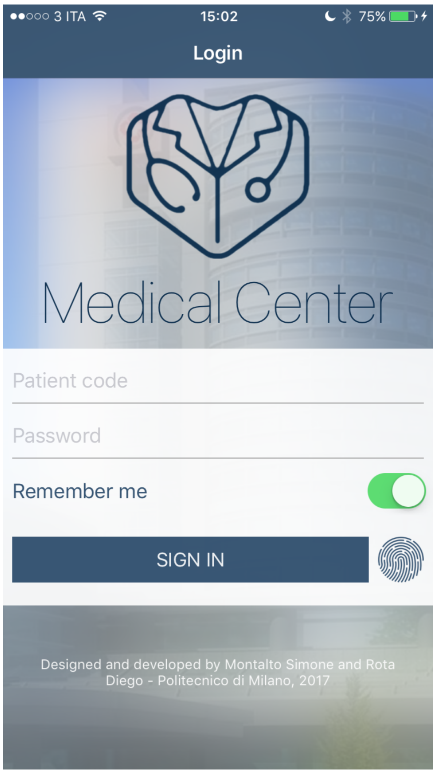

FIGURE 6.1: Login view of Medical Center

This is the first view that is shown when a user tap on the app icon. It permits to do the login with patient credentials. There isn't a *register* button because username and password are provided by the reception when a patient is hospitalized. Two labels are needed to insert the patient code and the password (password field is obscured in order to avoid that external people could see the password during the typing). A switch can be enabled to memorize the password. If the iOS device has Touch ID sensor, enabling Remember me field it is possible to access to the app using the fingerprint. Finally, a button associated to the Touch ID is present to show up the view for login with the fingerprint.

6.6.2 Home View

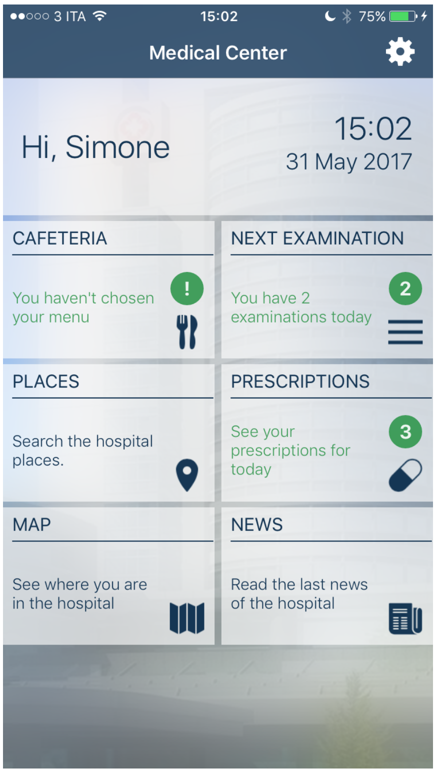

FIGURE 6.2: Home view of Medical Center

The homepage is a dynamic view. It is composed by seven tiles that show different information. The first tile is dedicated to the user's name, with date and hour which update itself automatically. Then, a tile has been dedicated for each section of the app, which is tapped to open the relative view. Below the title of each tile there's a brief description of its content or a quick view of relevant information for the user. For example, the *Next Examination* tile inform the patient if there are new examination for the day or not. A coloured badge is shown if a tile must attract the attention of the user.

6.6.3 Cafeteria View

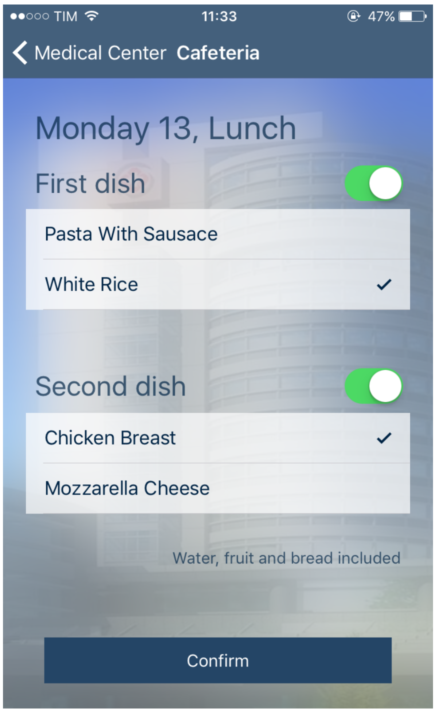

FIGURE 6.3: Cafeteria view of Medical Center

Cafeteria View shows the menu of the day. Each patient can select a personalized meal, with the possibility to deselect first or second dish if he want only one course. At the end of the selection, he can tap on *Confirm* button to confirm its selection. Until a new menu isn't inserted, the user can modify his choice tapping on *Modify* button.

6.6.4 Next Examination View

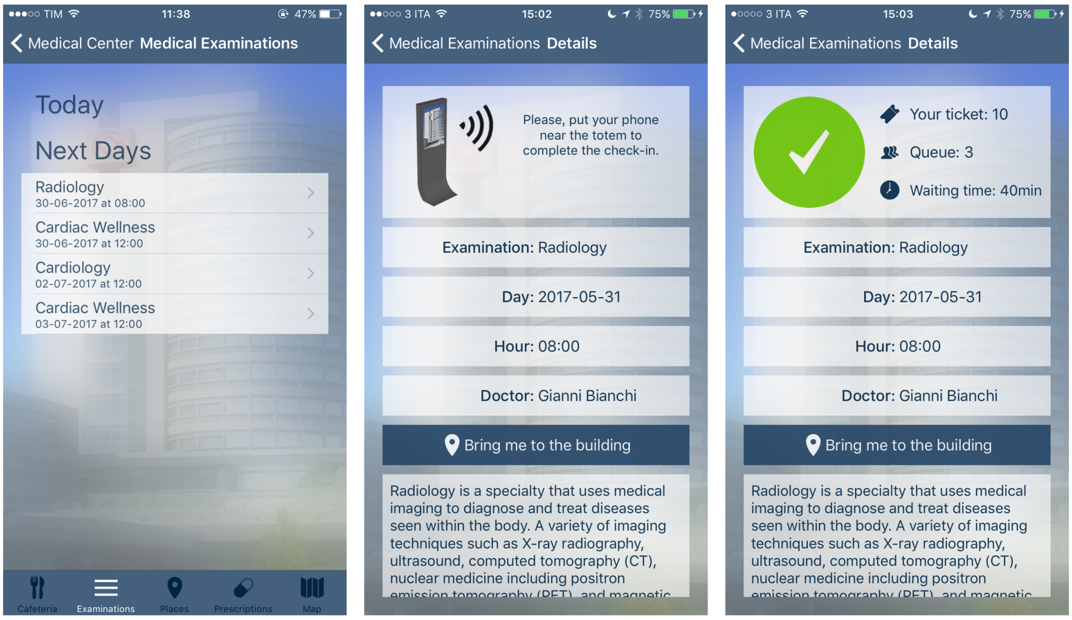

FIGURE 6.4: Next examination views of Medical Center

Medical examinations are presented in two views: the first one is a list of examinations, divided into *Today* and *Next Days*. Tapping on each medical examination, a further view is opened, with its description and some details. This view implements beacons technology in order to complete the check-in when a patient goes to the place of the visit. User can put the phone near a totem (equipped with beacons) the day of the examination to confirm his presence and do the check-in. Then, some information are provided, such as the number of his ticket, the number of people in queue and the waiting time. Also a button can be pressed to start the navigation inside the hospital to reach the place of the examination.

6.6.5 Places View

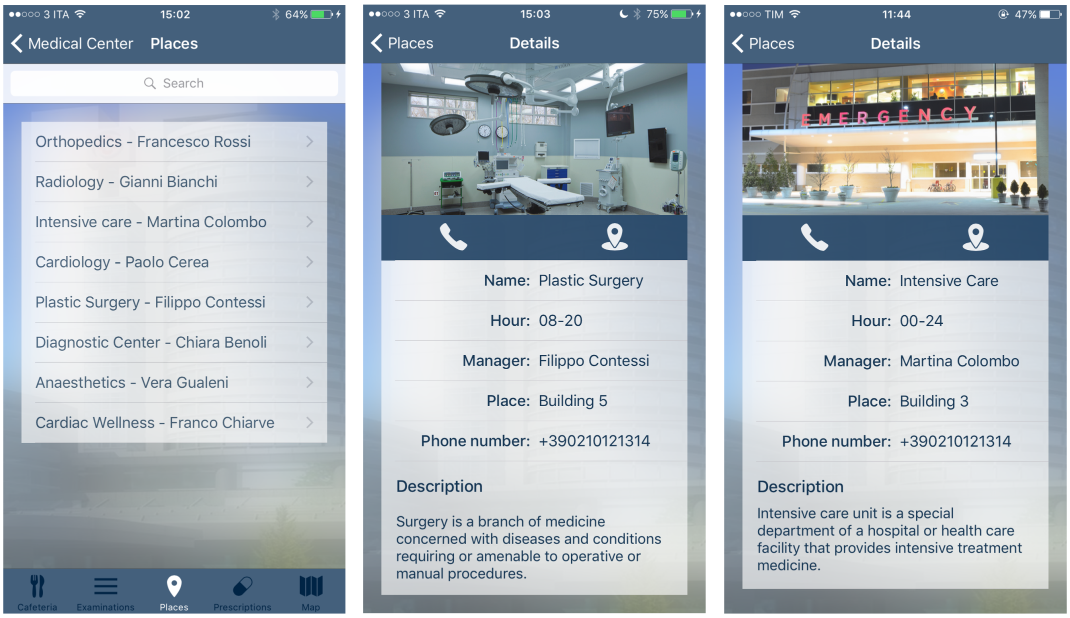

FIGURE 6.5: Places views of Medical Center

Also places are divided into two views, the first one being a list of all the Point Of Interest of the Hospital. This view also features a search bar to perform a quick research between all the places of the hospital while the details view ipresents some information of the place with quick buttons to call the reception or start a navigation to the Point Of Interest.

6.6.6 Prescriptions View

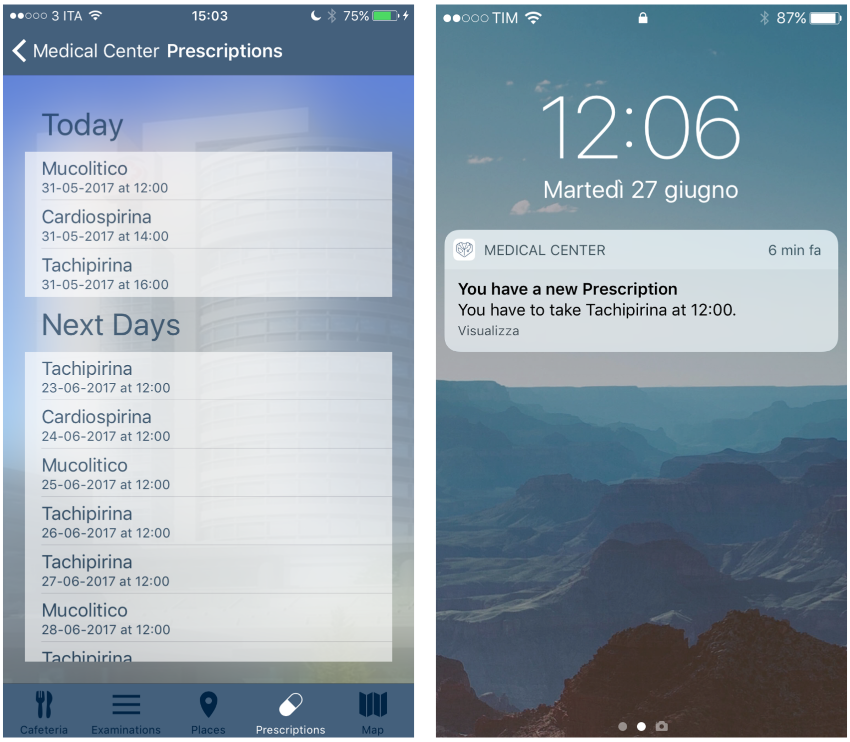

FIGURE 6.6: Prescriptions views of Medical Center

Prescriptions view show all medicines that the patient has to take. They are divided into *Today* and *Next Days*, presenting the name of the prescription and day and hour when the user has to take it.

6.6.7 Map View

FIGURE 6.7: Map view of Medical Center

This view presents the map of the hospital. When the smartphone is connected to a Pozyx Tag, his position and orientation are also shown on the map. At the bottom of the view, users can choose a destination point between all the Point Of Interest of the Hospital. Then, tapping on start button, they can start navigation to the selected place. After pressing the button, the optimal path is calculated and shown on the map. The bottom part of the view changes, showing textual indications place side by side by an arrow that shows the direction to follow. This view is dynamic, because since it updates itself reflecting new directions or possible obstacles on the path. The view also change colour if during the route an obstacle is nearby or if the user leaves the optimal path. Vocal commands to drive the user during the navigation have also been implemented. On the top of the view, some information about the connection state of Pozyx tag are shown in a box, that is hidden when there aren't new communications.

6.6.8 News View

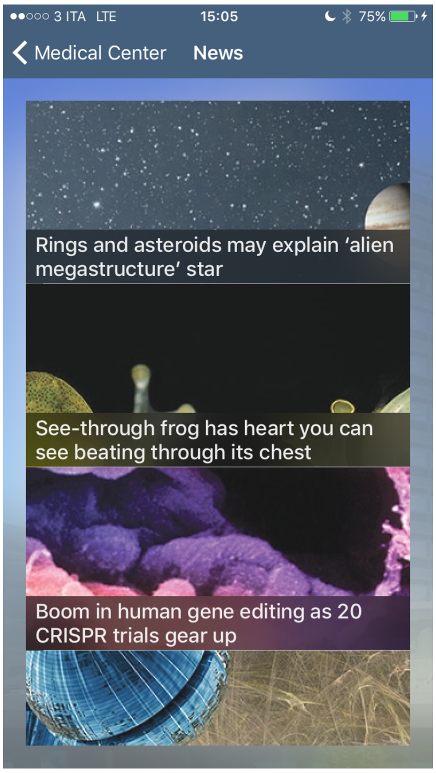

FIGURE 6.8: News view of Medical Center

News view retrieves new information through an HTTPS request and presents them in a collection view with the image of the news and its title. Tapping on each news, a Safari View Controller is created loading the URL of the chosen item.

6.6.9 Settings View

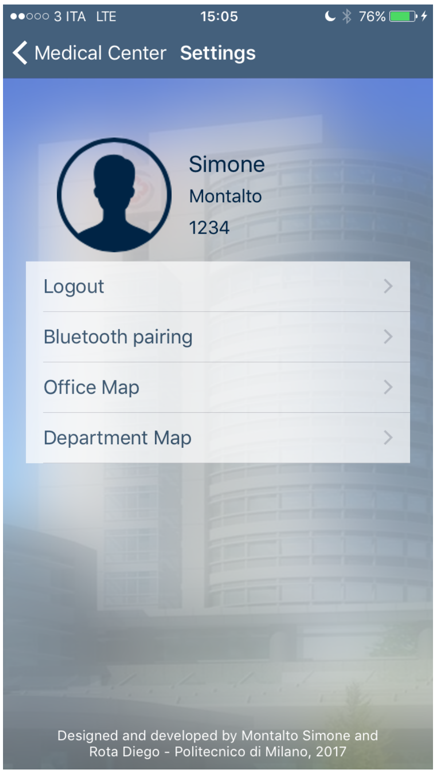

FIGURE 6.9: Settings view of Medical Center

Through the settings view, the user can show his information (name, surname and patient code), logout from his account and perform the bluetooth pairing with the Pozyx Tag.

Chapter 7

Conclusions

7.1 Conclusions

This work shows that is possible to realize an indoor positioning system and a navigation system to help people to move inside buildings. This result has been achieved because some experimental results have shown that, in an environment of $20m^2$, there is an average error of $4cm$ on the estimation of the static position. This is an optimal result which has permitted to estimate, with a high level of precision, the position of a person inside a room. The same instrumentations of the first test have been used to cover an area of $370m^2$. In these conditions, the average error increases up to $50cm$, but it is still acceptable to perform an accurate positioning. Also dynamics results have shown a good level of accuracy, which have permitted to create a system integrated with real world in order to perform further tests.

For this purpose, a prototype application has been designed, firstly to permit to blind people to move easily in an indoor environment but extendible too in order to include other kinds of applications. To deploy the systems, Ultra-Wide-Band technology has been preferred to Beacon, because it allows to have a low average error and a better signal propagation in environment with a lot of noises and obstacles. On top of the logical hardware layer, it has been necessary to implement some filtering algorithms to have a better measurement of the estimated position. In order to achieve this result, two algorithms have been implemented: the first one is a filter based on Kalman model, that is useful to keep track of noises and outliers and to cut-off them; the second one, instead, is an algorithm able to contextualize the estimated user position referring to current use case.

To perform the tests, half of DEIB department (Politecnico di Milano) has been mapped, by creating upon it a graph that has been integrated inside the navigator, in order to have a model where to calculate the optimal route path to the user destination (through Dijkstra's algorithm).

7.2 Future developments

Since the absence of Ultra-Wide-Band antenna inside technological devices is a big limit, it won't be an exaggeration to imagine, in the near future and as announced in (PozyxLabsNews, 2017), a *miniaturized* version of the tag device. This would allow to integrate UWB technology inside a wide range of possible accessories, such as a smartphone cover for example, with the result of enabling a more easy management of the system and reducing technological barriers to entry.

Another realistic improvement can be identified in the integration with *augmented reality*. In fact, it is possible to think at a most accurate realization of the environment map, relying not only on a graphical sketch (as used in navigation application of this work) but also on a live indirect (or direct) view of a physical, real-world environment whose elements are augmented by computer-generated sensory input such as sound, video, graphics or others data. The same *TE2*, described for tests in previous chapters, can be replaced with its augmented reality version.

The migration from a virtual reality (which replaces the real world with a simulated one) to an augmented reality, can enable a further layer of development focused on improving a lot the interaction with the user.
An example in this direction is the London Gatwick Airport (the second busiest airport in the United Kingdom) which has declared the intention to power an indoor navigation system as part of a wider, multi-year transformation program (Lomas/TechCrunch, 2017). On May 25, 2017 it was announced that it's now finished kitting out its two terminals with around 2,000 battery powered beacons so that digital map users will get a more accurate blue dot as they wander around. The beacon system will also be used to power an augmented reality wayfinding tool — so that mobile users will be able to be guided to specific locations within the terminals via on-screen arrows (Figure 7.1). The beacon system is slated as supporting positioning with $3m$ accuracy, which is enough for this use case.

So, it's quite reasonable to think to extend the UWB-based navigator software to an augmented version of it, maintaining its high precision. Furthermore, to take the advantages of beacons (such as easily installation without cable), a union of the two technologies might be considered; Estimote has

FIGURE 7.1: London Gatwick Airport augmented reality navigation system

announced the release of a new type of beacon (which integrates both BLE and UWB) in the near future (The-Estimote-Team-Blog, 2017).

The results obtained in the Machine Learning field could permit to extend the navigator to help blind people to search objects inside an environment. As an example, for a navigation system of a mall, users could be able to choose an object to buy. An application could lead them to the shelf where the object is located and, finally, user can frame the shelf using the smartphone's camera. A voice will indicate if the framed object is the chosen one at the beginning of the navigation, helping blind people to take the right object without others assistance.

Finally, further work could be done concerning filtering algorithm, by improving and best tuning KF model, exploring other filtering technique or developing a highly accurate software to auto-map environment, with a focus on best anchors placement.

Appendix A

Kalman filter Matlab implementation

Here is reported the Matlab implementation of the Kalman filter model. Line 1 to 13 contain the initialization of components (according to what discussed in the thesis). Afterwards, a loop simulates each new incoming measurement from sensors, which is composed by position and acceleration on both axes. The estimated position after filtering is contained in *xhat[1]* (x-axis) and *xhat[3]* (y-axis).

```
1  z = 40;  %measurement noise
2  w = 2;   %process noise
3  T = 0.5;  %sampling time
4
5  A = [1 T 0 0; 0 1 0 0; 0 0 1 T; 0 0 0 1]; % state matrix
6  B = [T^2/2 0; T 0; 0 T^2/2; 0 T]; % control matrix
7  C = [1 0 0 0; 0 0 1 0]; % output matrix
8  xhat = [0; 0; 0; 0]; % initial state vector x(0)
9
10 Sz = [z^2 0; 0 z^2]; % measurement noise covariance matrix
11 Sw = w^2 * [(T^2/2)^2 (T^2/2)*T 0 0; (T^2/2)*T T^2 0 0; 0 0 (T^2/2)^2 (T^2/2)*T; 0 0 (T^2/2)*T T^2]; % process noise covariance matrix
12
13 P = Sw; % initial estimation covariance
14
15 for each_new_incoming_measurement
16 u = [acceleration_x; acceleration_y];     %compose input vector (with acceleration data on both axes coming from sensors)
17 y = [position_x; position_y]; %compose output vector (assign (noisy) measured data)
18
19 %Apply Kalman filter equations
20 xhat = A * xhat + B * u;  % Extrapolate the most recent state estimate to the present time.
21 Inn = y - C * xhat; % Form the Innovation vector.
22 s = C * P * C' + Sz; % Compute the covariance of the Innovation.
23 K = A * P * C' * inv(s);  % Form the Kalman Gain matrix.
24 xhat = xhat + K * Inn;  % Update the state estimate.
25 P = A * P * A' - A * P * C' * inv(s) * C * P * A' + Sw;  % Compute the covariance of the estimation error.
26
27 %Display results
28 disp(['Position estimated, x-axis:' xhat(1)]);
29 disp(['Position estimated, y-axis:' xhat(3)]);
30 end
```

Appendix B

Medical Center Technical Documentation

B.1 Data model

This section will mention all the relationship between the various entity of the system. These will define the database, which data can be exchanged with the application through JSON files.

B.1.1 ER Diagram

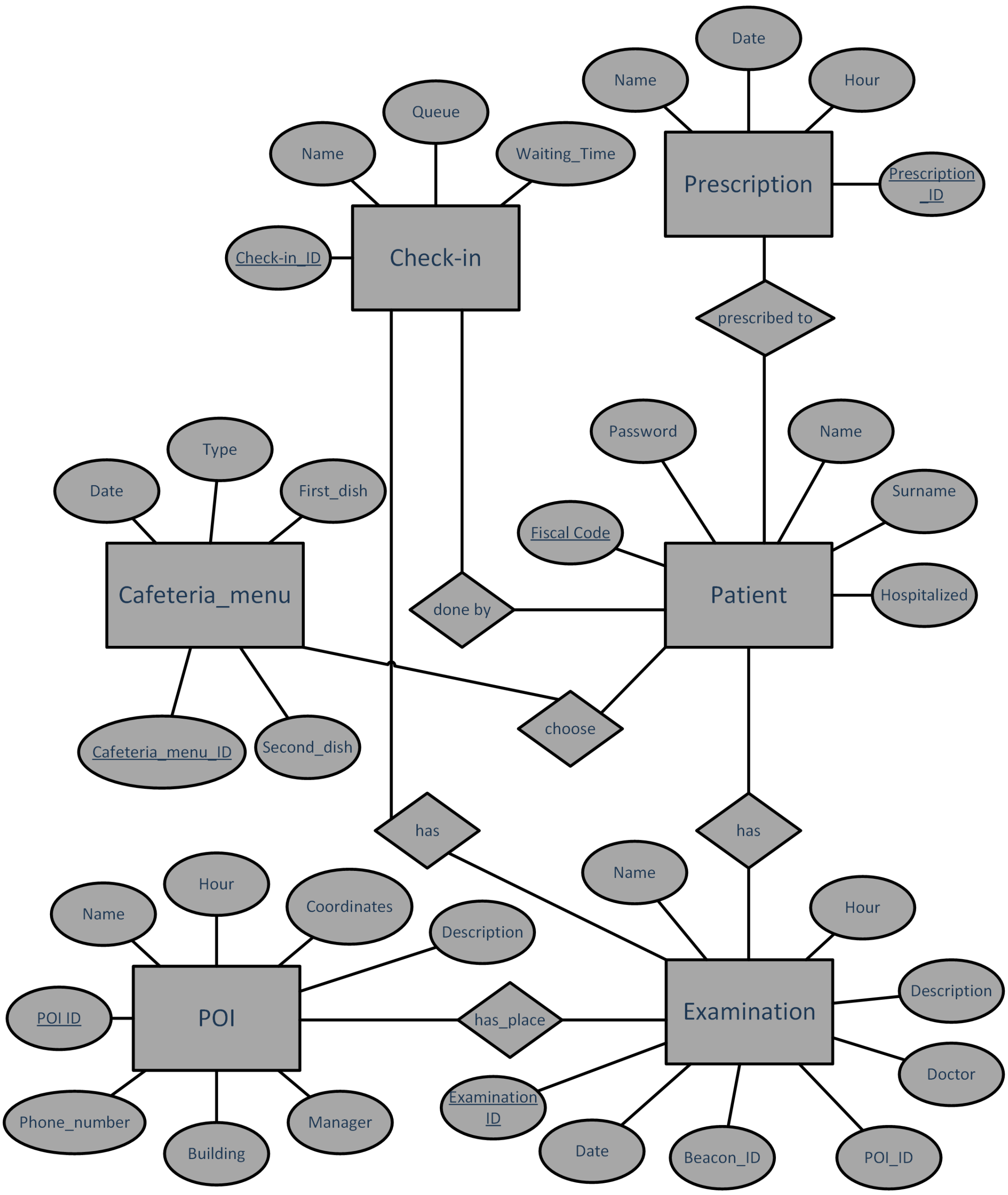

FIGURE B.1: Entity-Relationship Diagram

B.1.2 Logical-Conceptual design

All the tables of the database are listed in the following sections. For each table there's a key value which identify each instance of an entity and is useful to connect two different tables.

Patient table

TABLE B.1

Field	**Type**	**Description**
Fiscal Code	String	It is the key of the table and represents the fiscal code of each user.
Password	String	It the password to access to the application.
Name	String	It is the name of the patient.
Surname	String	It is the surname of the patient.
Hospitalized	Boolean	It is a boolean that is true if the patient is hospitalized, false otherwise.

Examination table

TABLE B.2

Field	**Type**	**Description**
Examination ID	String	It is the key of the table.
Name	String	It the name of the examination.
Date	Date	It is the date when the examination will be done.

Hour	String	It is the hour when the examination will be performed.
Description	Boolean	It is a description of the examination.
Doctor	String	It is the name of the doctor who will perform the examination.
POI ID	String	It is an identifier of the place where the examination will be taken.
Fiscal code	String	It is the identifier of the patient interested in the visit.
Beacon ID	String	It is the ID of the beacon uses to perform the check-in.
Check-In ID	Boolean	It is the ID of the Check-in table.

Prescription table

TABLE B.3

Field	**Type**	**Description**
Prescription ID	String	It is the key of the table.
Name	String	It the name of the prescription.
Date	Date	It is the date when the prescription must be taken.
Hour	String	It is the hour when the prescription must be taken.

Fiscal code	String	It is the identifier of the patient interested in the prescription.

Point of Interest table

TABLE B.4

Field	**Type**	**Description**
POI ID	String	It is the key of the table.
Name	String	It the name of the place.
Hour	String	It is the hour when the place is open.
Coordinates	String	It represents the coordinates of the place.
Description	String	It is the description of the place.
Manager	String	It is the name of the person who manages the place.
Building	String	It is the name of the building where the Point of Interest is.
Phone number	String	It is the phone number to contact the Point Of Interest.

Check-in table

TABLE B.5

Field	**Type**	**Description**
Check-in ID	String	It is the key of the table.
Ticket	Integer	It is the number of the ticket.
Queue	Integer	It is the number of people in queue before the user.
Waiting time	Integer	It is the waiting time for the user in minutes.

Cafeteria table

TABLE B.6

Field	**Type**	**Description**
Cafeteria menu ID	String	It is the key of the table.
Date	Date	It is the date of the menu.
Type	String	It identifies if the menu is for lunch or dinner.
First Dish	String	It is a list of first dishes.
Second Dish	String	It is a list of second dishes.
Fiscal Code	String	It is the ID of the user who chose the menu.

B.2 UML diagrams

The following sections show some diagrams that helps to understand how all the part interact one another. In particular, there are Class Diagram and Use Case.

B.2.1 Class diagram

To have a general idea of the implementation of the app, Figure B.2 shows the class diagram of *Medical Center*. It represents the classes of the application, with all the relevant connections.

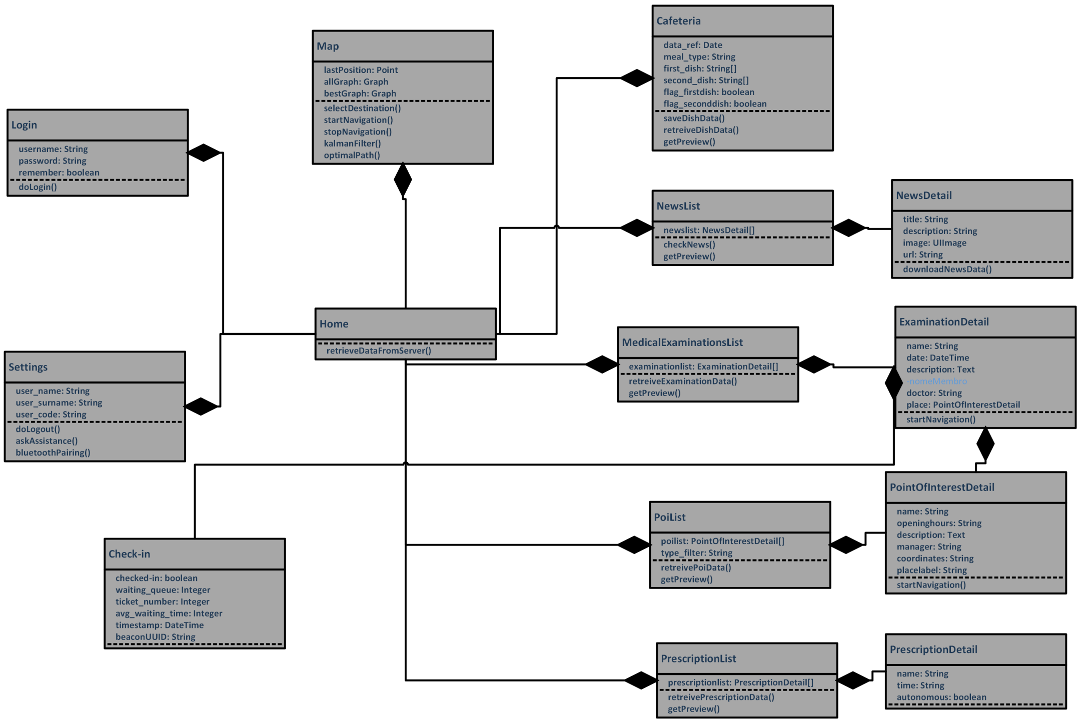

FIGURE B.2: Medical Center Class Diagram

B.2.2 Use case diagrams

In this section will be shown the most relevant use cases of the application. They are an overview of the usage requirements for the system.

Login

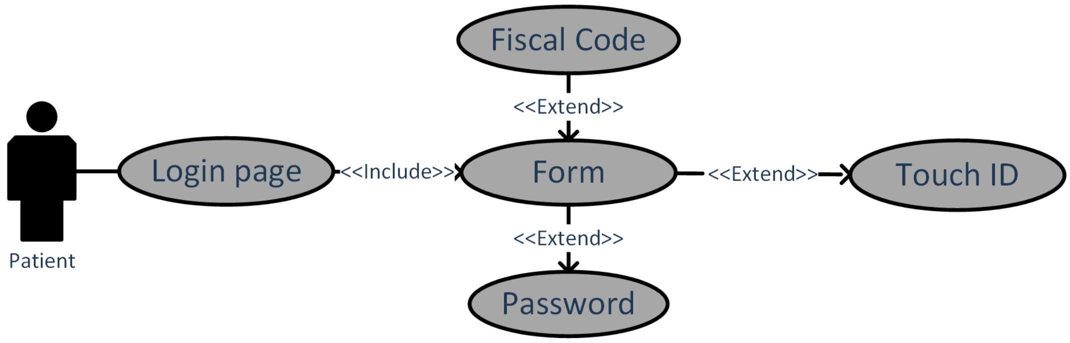

FIGURE B.3: Login Use case

Settings

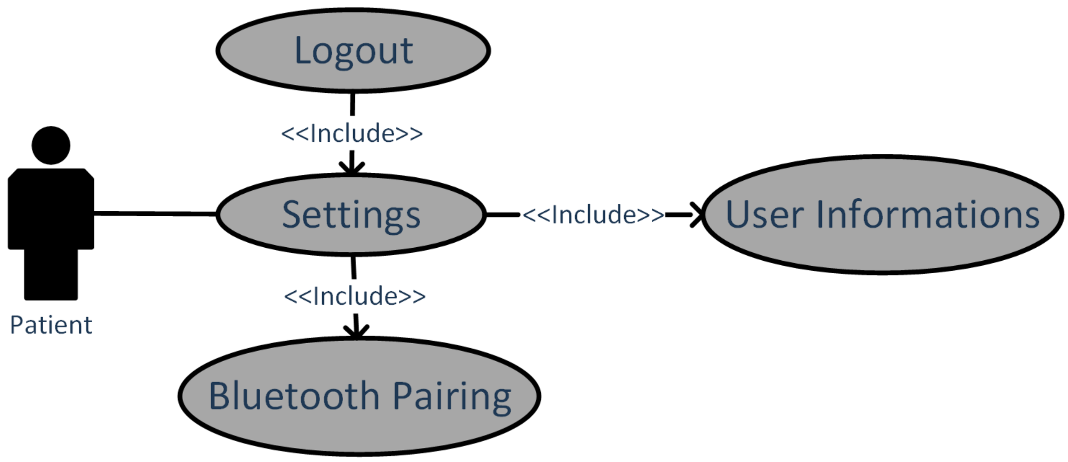

FIGURE B.4: Settings Use case

Homepage

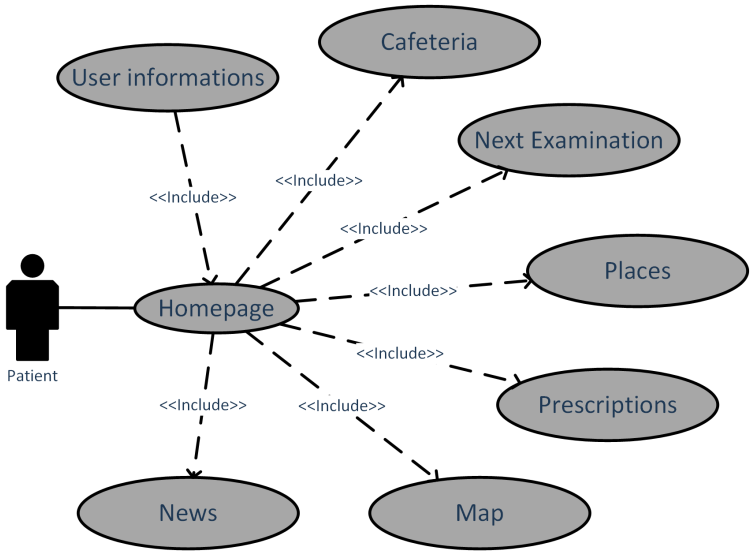

FIGURE B.5: Homepage Use case

Cafeteria

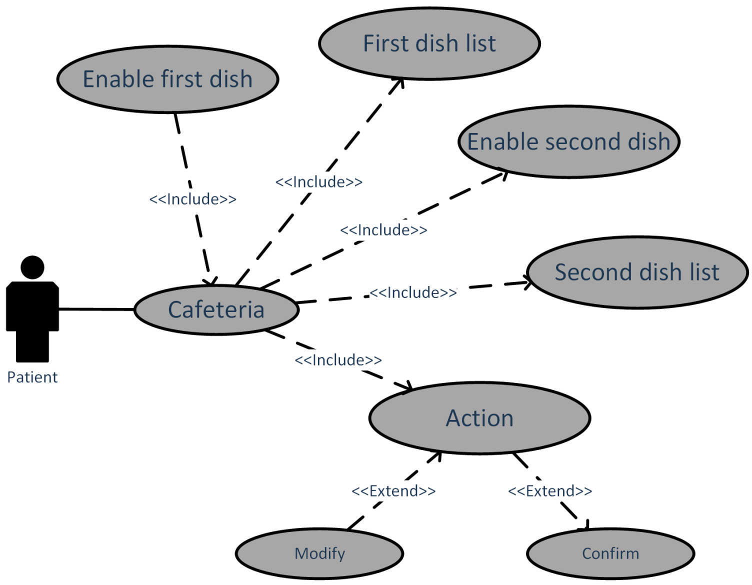

FIGURE B.6: Cafeteria Use case

Medical Examination

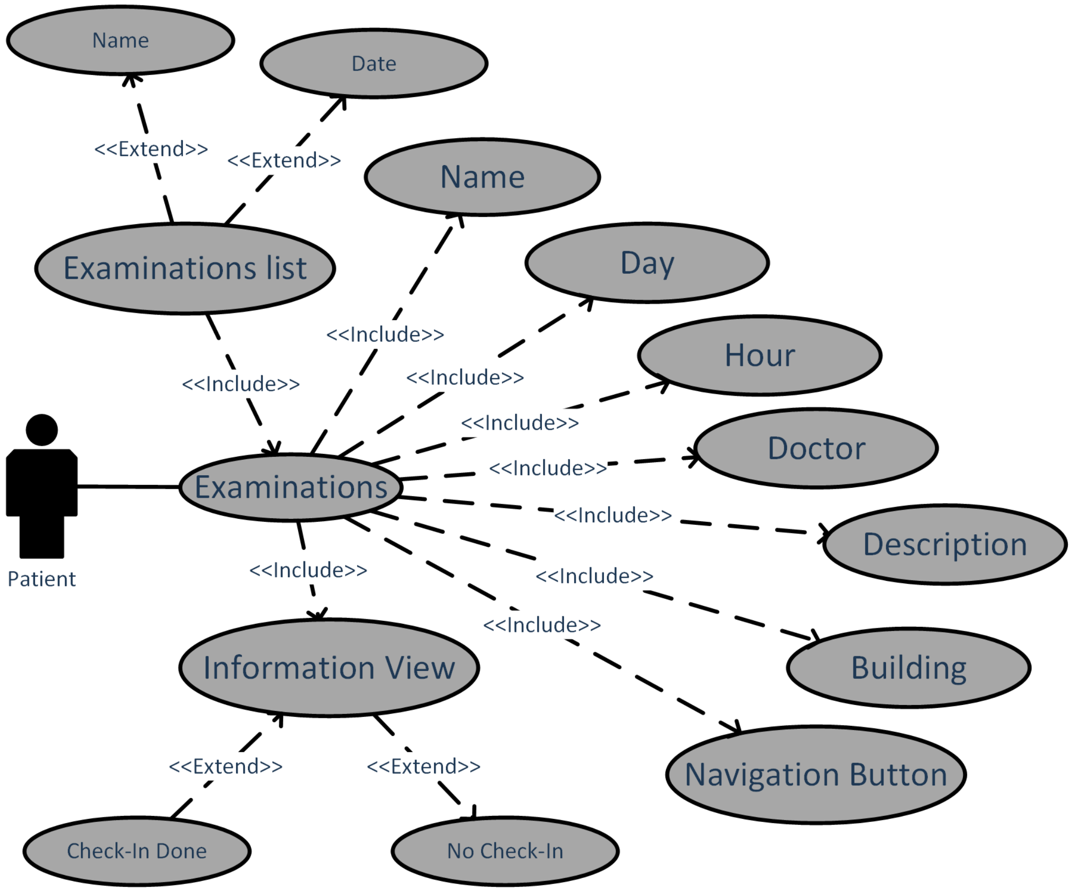

FIGURE B.7: Medical Examination Use case

Places

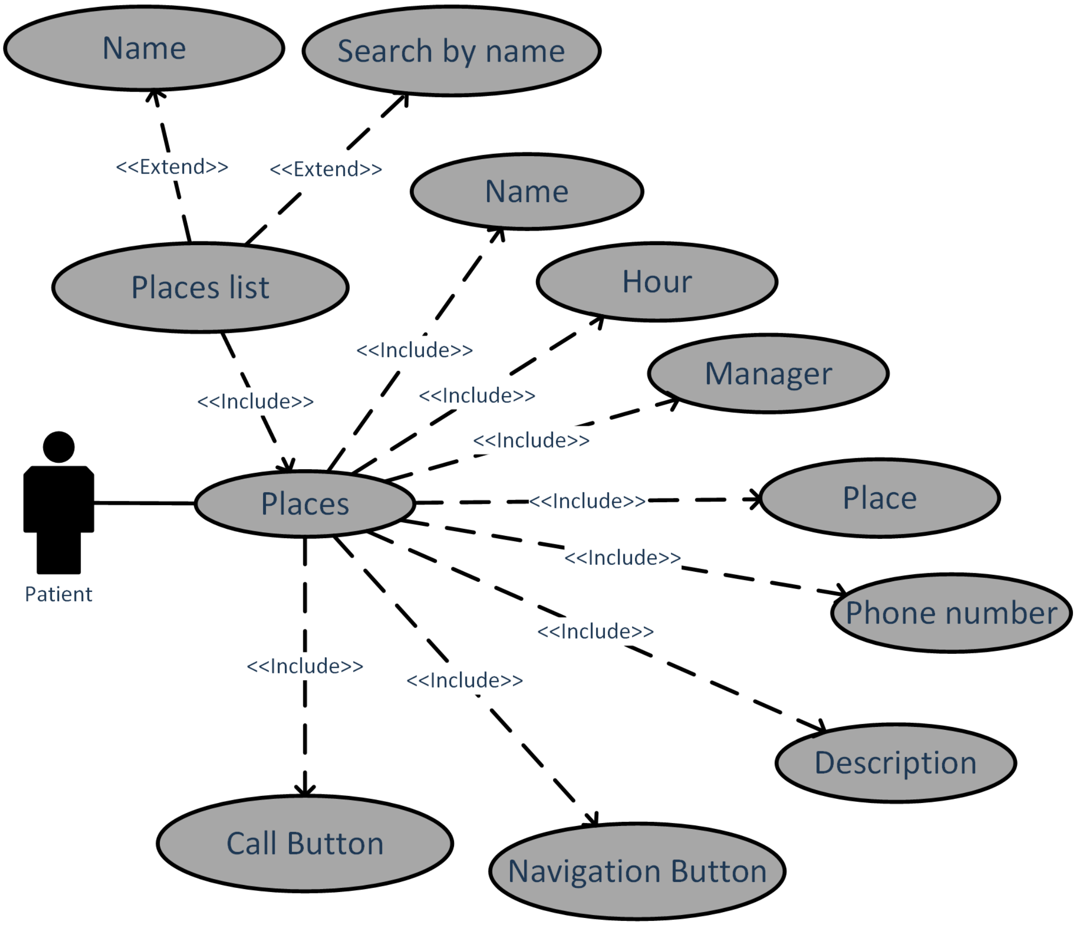

FIGURE B.8: Places Use case

Map

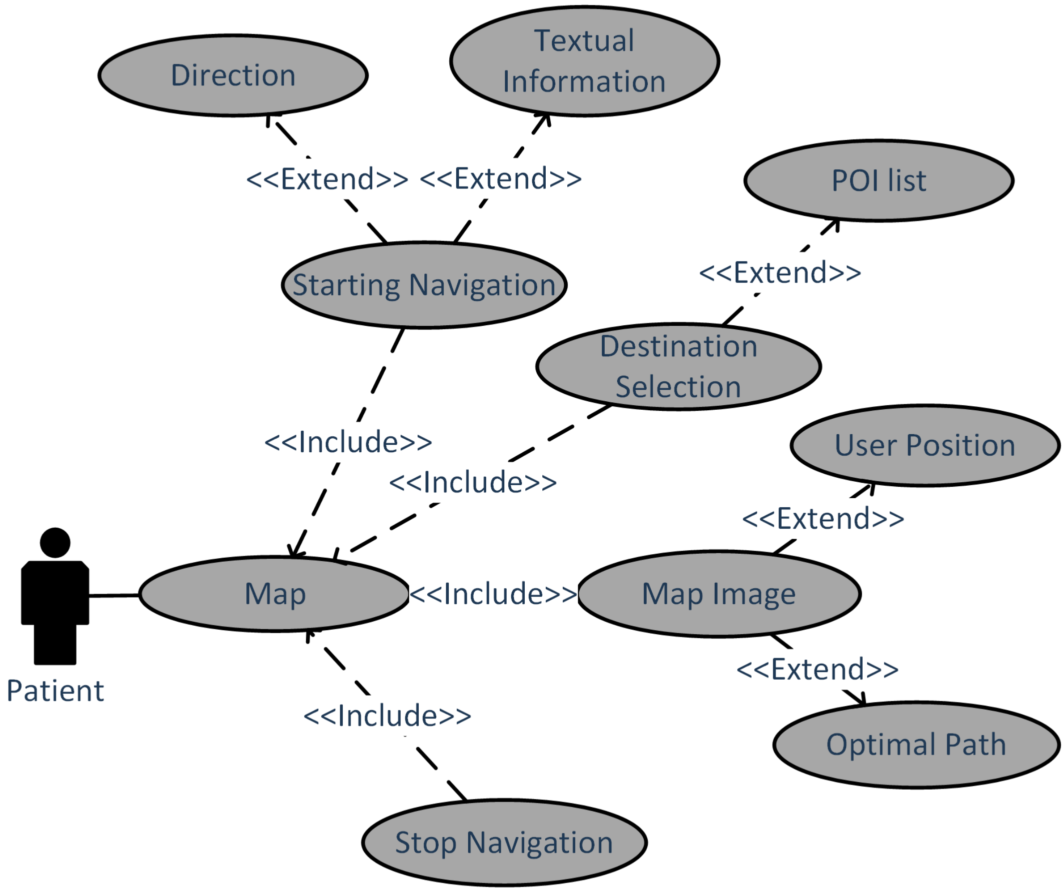

FIGURE B.9: Map Use case

Prescriptions

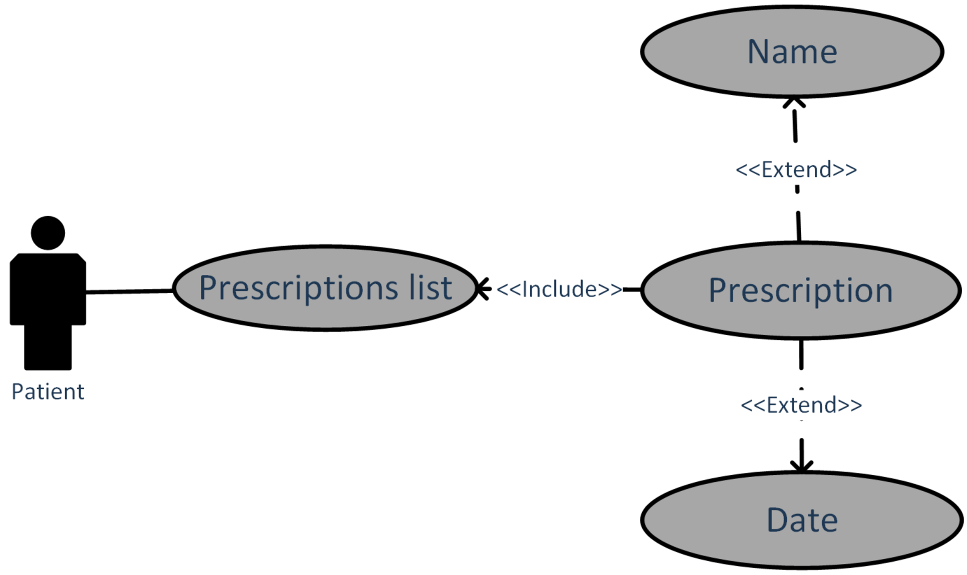

FIGURE B.10: Prescriptions Use case

News

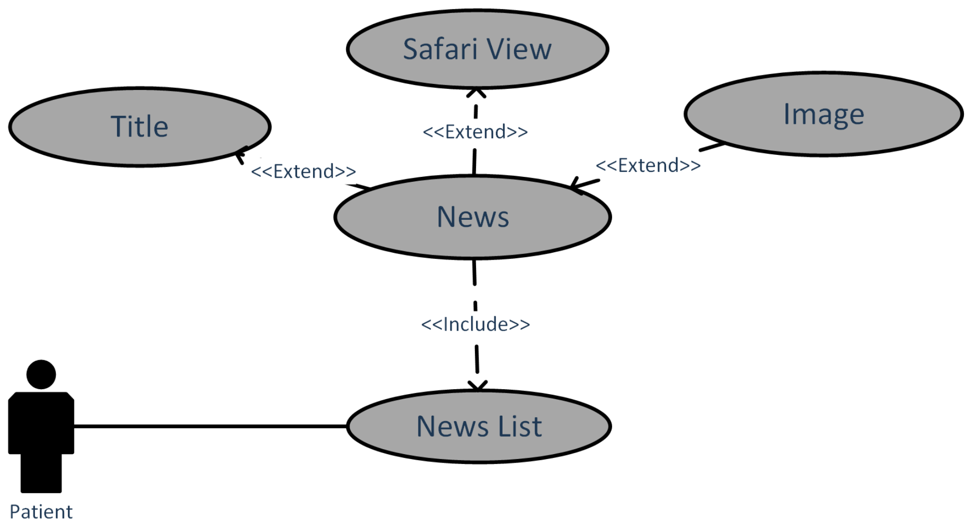

FIGURE B.11: News Use case

Software libraries

pozyx®
ACCURATE POSITIONING

https://github.com/pozyxLabs

RedBearLab

https://github.com/RedBearLab

ARDUINO™

https://github.com/maniacbug/MemoryFree
https://github.com/tomstewart89/BasicLinearAlgebra

Swift

https://developer.apple.com/reference/corelocation
https://github.com/aleph7/Upsurge
https://github.com/SwiftyJSON/SwiftyJSON

Thesis resources

The resources of this thesis, including documentation and source code, are available at:
`https://github.com/diegorota`
`https://github.com/simonemontalto`

A copy of this work is also available at:
`https://www.politesi.polimi.it/?locale=en`

Bibliography

Abu Doush, Iyad et al. (2017). "ISAB: Integrated Indoor Navigation System for the Blind". In: *Interacting with Computers* 29.2, p. 181. DOI: 10.1093/iwc/iww016. eprint: /oup/backfile/content_public/journal/iwc/29/2/10.1093_iwc_iww016/1/iww016.pdf. URL: +http://dx.doi.org/10.1093/iwc/iww016.

Ahmetovic, Dragan et al. (2017). "Achieving Practical and Accurate Indoor Navigation for People with Visual Impairments". In:

Alnafessah, Ahmad et al. (2016). "Developing an Ultra Wideband Indoor Navigation System for Visually Impaired People". In: *International Journal of Distributed Sensor Networks* 12.7, p. 6152342.

ARMmbed (2015). *Understanding the different types of BLE Beacons*. URL: https://developer.mbed.org/blog/entry/BLE-Beacons-URIBeacon-AltBeacons-iBeacon/.

Caicedo, D. and A. Pandharipande (2012). "Ultrasonic array sensor for indoor presence detection". In: *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pp. 175–179.

Calabrese, Chiara (2015). "Realtà aumentata e applicazioni mobili a supporto della pianificazione urbanistica". Available at http://hdl.handle.net/10589/106730. MA thesis. Politecnico di Milano.

Cellar, Circuit (2014). *Triangulation, Trilateration, or Multilateration?* URL: http://circuitcellar.com/ee-tips/triangulation-trilateration-or-multilateration-ee-tip-125/.

Dao, T. K. et al. (2016). "Indoor navigation assistance system for visually impaired people using multimodal technologies". In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1–6. DOI: 10.1109/ICARCV.2016.7838771.

Ersan Gunes, Gonzalo Ibarra (2017). "IntraNav – Low-Cost Indoor Localization System and IoT Platform". In:

Granby, John (2012). *What Are Some Signal Interference Issues for the Bluetooth Technology?* URL: http://smallbusiness.chron.com/signal-interference-issues-bluetooth-technology-58273.html.

Hauschildt, Daniel and Nicolaj Kirchhof (2010). “Advances in thermal infrared localization: Challenges and solutions”. In: *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*.

Hnat, Timothy W. et al. (2012). “Doorjamb: Unobtrusive Room-level Tracking of People in Homes Using Doorway Sensors”. In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. SenSys ’12. Toronto, Ontario, Canada: ACM, pp. 309–322. ISBN: 978-1-4503-1169-4. DOI: `10.1145/2426656.2426687`. URL: `http://doi.acm.org/10.1145/2426656.2426687`.

Jiang Xiao Zimu Zhou, Youwen Yi (2016). “A Survey on Wireless Indoor Localization from the Device Perspective”. In:

Kota Kikuchi Ryota Tazawa, Naoki Honma (2017). “Implementing an iBeacon Indoor Positioning System using Ensemble Learning Algorithm”. In:

Kuan-Wu Su He-Yen Hsieh, Jen-Chieh Hsu (2017). “Implementing an iBeacon Indoor Positioning System using Ensemble Learning Algorithm”. In:

Kul, Gökhan, Tansel Özyer, and Bülent Tavli (2014). “IEEE 802.11 WLAN based Real Time Indoor Positioning: Literature Survey and Experimental Investigations”. In: *Procedia Computer Science* 34. The 9th International Conference on Future Networks and Communications (FNC’14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC’14)/Affiliated Workshops, pp. 157 –164. ISSN: 1877-0509. DOI: `http://dx.doi.org/10.1016/j.procs.2014.07.078`. URL: `http://www.sciencedirect.com/science/article/pii/S1877050914009338`.

Lomas/TechCrunch, Natasha (2017). *Gatwick Airport now has 2,000 beacons for indoor navigation*. URL: `https://techcrunch.com/2017/05/25/gatwick-airport-now-has-2000-beacons-for-indoor-navigation/`.

PozyxLabsNews (2017). *Pozyx in European drone project*. URL: `https://www.pozyx.io/Welcome/news#droneproject`.

Priyantha, Nissanka B., Anit Chakraborty, and Hari Balakrishnan (2000). “The Cricket Location-support System”. In: *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. MobiCom ’00. Boston, Massachusetts, USA: ACM, pp. 32–43. ISBN: 1-58113-197-6. DOI: `10.1145/345910.345917`. URL: `http://doi.acm.org/10.1145/345910.345917`.

RadiusNetwork (2014). *Fundamentals of Beacon Ranging*. URL: http://developer.radiusnetworks.com/2014/12/04/fundamentals-of-beacon-ranging.html.

R.K. Ahuja T.L. Magnanti, J.B. Orlin (1993). *Network flows*. Prentice Hall, pp. 108–112.

Rouse, Margaret (2008). *Ultra wideband*. URL: http://whatis.techtarget.com/definition/ultra-wideband.

Safaai-Jazi, Ahmad et al. (2002). "Ultra-wideband propagation measurements and channel modeling". In: *Report on Through-the-Wall Propagation and Material Characterization*.

Seco, F. et al. (2010). "Improving RFID-based indoor positioning accuracy using Gaussian processes". In: *2010 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–8. DOI: 10.1109/IPIN.2010.5647095.

Simon, Dan (2001). "Kalman filtering". In: *Embedded systems programming* 14.6, pp. 72–79.

The-Estimote-Team-Blog (2017). *Estimote Beacons with UWB can now automatically create floor plans*. URL: http://blog.estimote.com/post/154460651570/estimote-beacons-with-uwb-can-now-automatically.

Yang, Y. and A. E. Fathy (2009). "Development and Implementation of a Real-Time See-Through-Wall Radar System Based on FPGA". In: *IEEE Transactions on Geoscience and Remote Sensing*.

Yang, Yunqiang, Y Wang, and Aly E Fathy (2008). "Design of compact Vivaldi antenna arrays for UWB see through wall applications". In: *Progress In Electromagnetics Research*.

Youssef, Moustafa, Matthew Mah, and Ashok Agrawala (2007). "Challenges: Device-free Passive Localization for Wireless Environments". In: *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking*. MobiCom '07. Montreal, Quebec, Canada: ACM, pp. 222–229. ISBN: 978-1-59593-681-3. DOI: 10.1145/1287853.1287880. URL: http://doi.acm.org/10.1145/1287853.1287880.